Quelques contributions à la théorie de l'apprentissage profond : optimisation, robustesse et approximation

El Mehdi Achour

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *08/12/2022* **par :**

El Mehdi ACHOUR

Quelques contributions à la théorie de l'apprentissage profond: optimisation, robustesse, et approximation

---

JURY

Francis BACH	INRIA-ENS Paris	Examinateur
Jérôme BOLTE	Université Toulouse 1	Président du jury
Stéphane CHRETIEN	Université Lyon 2	Rapporteur
Sébastien GERCHINOVITZ	IRT Toulouse	Directeur de thèse
Rémi GRIBONVAL	INRIA-ENS Lyon	Examinateur
Clément HONGLER	EPFL Lausanne	Rapporteur
Gitta Kutyniok	LMU Munich	Examinatrice
François MALGOUYRES	Université Toulouse 3	Directeur de thèse

École doctorale et spécialité :
MITT : Domaine Mathématiques : Mathématiques appliquées
Unité de Recherche :
Institut de Mathématiques de Toulouse (UMR 5219)
Directeur(s) de Thèse :
François MALGOUYRES et *Sébastien GERCHINOVITZ*
Rapporteurs :
Stéphane CHRETIEN et *Clément HONGLER*

Remerciements

Tout d'abord, je tiens à remercier chaleureusement mes directeurs de thèse, François et Sébastien qui m'ont initié au monde la recherche à travers mon stage et m'ont accompagné sur tous les plans durant ces trois années de thèse. Merci pour vos conseils et vos instructions qui m'ont permis de grandir tant sur le plan humain que scientifique. C'était toujours un plaisir d'échanger avec vous à propos de mathématiques ou autres. Merci aussi d'avoir été disponibles durant la période du Covid. J'ai beaucoup apprécié mon temps avec vous et j'espère que notre collaboration pourra continuer dans le futur.

Je tiens aussi à remercier mon jury. En commençant par les rapporteurs, Stéphane Chrétien et Clément Hongler. Merci pour votre implication et votre relecture attentive de mon manuscrit. Merci Jérôme d'avoir accepté de faire partie de mon jury, ainsi que pour tous nos échanges durant ces trois années. Merci aussi à Francis Bach, Rémi Gribonval et Gitta Kutyniok d'avoir accepté de faire partie de mon jury.

Merci Franck pour notre collaboration, c'était un plaisir de travailler et d'échanger avec toi pendant tout ce temps.

Cette thèse aurait été différente sans la présence de plusieurs autres doctorants. Tout d'abord, mes cobureaux, anciens et actuels : Jordan, merci pour toutes ces discussions et j'attends impatiemment notre combat de boxe. Clément, on a commencé ensemble et on finit ensemble, ce fut un grand plaisir de partager des moments inoubliables avec toi au bureau et en dehors, je pense notamment parmi tant de choses à nos séances de tir où tu m'as vite fait comprendre que ça va pas le faire pour les JO de Paris. Merci à Fanny qui nous a rejoint au milieu et qui a apporté sa touche d'humour et de partage. Enfin merci à Chifaa qui en si peu de temps a su s'imposer comme un élément indispensable au sein du bureau, de par sa joie de vivre et ses blagues bien élaborées.

Enfin, j'ai interagi avec plusieurs autres doctorants que je tiens à remercier. Merci Armand, nos aventures sont nombreuses et ne peuvent pas se résumer en quelques lignes. Merci d'avoir été un collaborateur formidable. Merci Joachim pour tous les événements qu'on a fait ensemble, ce fut un plaisir de te cotoyer. Merci Florian de m'avoir accompagné et guidé dès mon stage, ainsi que pour ces soirées foot où on sait qui paie. Michèle ton sourire et tes ondes positives n'ont pas d'égal. Anthony Mur, ta rapidité nous manque depuis ton départ. Laetitia, désolé pour toutes les questions administratives que j'ai dû te poser alors que la réponse était pas loin. Merci d'avoir rendu toutes mes tâches plus faciles. Mahmoud, les débats avec toi sont toujours intéressants. Paola, merci pour tes bons conseils en voyage. Alain j'espère que tu vas réaliser ton rêve d'aller aux US, n'oublie pas de m'inviter quand tu seras prof là-bas. Louis, merci pour tes énigmes qui sont pour le coup très très durs. William, merci pour toutes les leçons infligées au badminton, prépare-toi, bientôt j'aurais ma première victoire. Dominique, merci pour les échanges ainsi que tes conseils pour le postdoc. Merci Pierre d'avoir commencé avec moi cette route vers Honolulu. Merci aussi à Joe, Clément, Perla, Virgile, Sofiane, Nicolas, Sophia, Lucas, Fu-Hsuan, Sara, Baptiste, Joachim, Matthias, Eva, Alexis, Etienne, Viviana, Alexandre, Anthony, Corentin, Alberto, Fabien, J-M, Dimitri, Maxime, Erfan, Javi, Benjamin et à toute autre personne avec qui j'ai pu interagir au sein du laboratoire durant mes années ici. Merci à tout le personnel de l'IMT, ces trois ans étaient magnifiques à vos côtés. Merci aux gens de l'IRT avec qui j'ai

pu échanger sur des sujets intéressants. Merci au personnel de l'Upsidum qui nous ont tout le temps accueilli avec le sourire.

Merci à mes élèves qui m'ont fait aimer l'enseignement. Merci à mes amis de Toulouse, grâce à qui j'ai aimé cette ville. Merci aussi à tous mes amis de partout qui m'ont soutenu durant cette période de ma vie. Sans vous tout cela n'aurait été possible.

Enfin rien de cela n'aurait été possible sans l'immense aide de ma famille. Les mots ne suffisent pas pour exprimer ma gratitude envers tout ce que vous avez fait pour moi durant ces trois ans en particulier et durant toute ma vie.

Contents

CHAPITRE 1

Introduction en français

1.1 L'apprentissage machine

"AI is the new electricity", tels sont les propos d'Andrew Ng, célèbre enseignant-chercheur à l'université de Stanford dont les cours sur Coursera de Machine Learning et de Deep Learning ont battus tous les records d'inscription sur la plateforme. Ceci révèle l'importance de l'intelligence artificielle, en particulier l'apprentissage machine (Machine Learning) ainsi que l'apprentissage profond (Deep Learning) dans notre ère. En effet, on retrouve ces algorithmes partout de nos jours, si vous ouvrez votre navigateur et que vous cherchez quoique ce soit sur un moteur de recherche, il y a un algorithme derrière qui utilise des réseaux de neurones artificiels ou ce qu'on appelle plus communément le deep learning.

Le deep learning en particulier a connu une explosion depuis 2012 et le fameux papier du groupe de Hinton qui a battu de loin l'état de l'art en reconnaissance d'images grâce au réseau de neurones qu'ils ont entrainé. Suite à ceux-ci plusieurs chercheurs se sont retournés une nouvelle fois vers ces réseaux de neurones qui étaient déjà étudiés théoriquement dans les années 80. Toutefois, leurs performances n'étaient pas assez compétitifs avec les autres méthodes vu le manque de données et de puissance de calcul. En 10 ans le deep learning a connu une avancée phénomenale et il est inconcevable de nos jours d'imaginer un monde sans ces réseaux de neurones artificiels. En effet, ils sont utilisés en reconnaissances d'images, en traduction, en reconnaissance de voix, pour les recommandations et autres. Tous les ans on leur trouve de nouvelles applications où ils réussisent à battre l'état de l'art et permettent des avancées considérables.

Qu'est-ce que donc un réseau de neurones ? Un réseau de neurones est constitué d'une couche d'entrée, une couche de sortie ainsi que plusieurs couches intermédiaires qu'on appelle couches cachées. Pour passer d'une couche à l'autre on applique une transformation linéaire suivie d'une fonction d'activation. L'apprentissage par réseaux de neurones est appelé apprentissage profond car la profondeur est égale aux nombre de ces couches. En pratique pour les grands réseaux ils sont de l'ordre de dizaines. Une définition plus rigoureuse mathématiquement est présentée à la section suivante.

Ces réseaux de neurones ont donc donné des résultats impressionnants concernant tous les domaines, mais il manque une théorie expliquant pourquoi ils fonctionnent si bien. Et contrairement aux anciens algorithmes de machine learning qui fonctionnaient bien et qu'on utilisait dans les années 2000, tels les SVMs qu'on comprend bien théoriquement, les réseaux de neurones artificiels sont difficiles à étudier et selon la théorie classique il n'y a aucune raison pour que l'apprentissage fonctionne aussi bien. Ainsi, de nombreux chercheurs ont travaillé et continuent à travailler sur cette question, afin d'essayer d'apporter des réponses partielles et d'essayer de réduire l'écart existant entre les résultats extraordinaires que l'on

observe en pratique et les aspects mathématiques qui peuvent les expliquer théoriquement.

1.1.1 Réseau de neurones artificiels

Un réseau de neurones artificiels feedforward est une fonction composée de plusieurs couches. A chaque couche une fonction linéaire est appliqué suivi d'une fonction d'activation. Plus précisément, si on considère un réseau :

- H couches ($H-1$ couches cachées)
- les tailles des différentes couches sont notées : $n_0, n_1, \ldots, n_H \in \mathbb{N}^*$.
- On note $f_h(x)$ le résultat obtenu en calculant le contenu de la couche h pour l'entrée $x \in \mathbb{R}^{n_0}$
- On note $W_h \in \mathbb{R}^{n_h \times n_{h-1}}$ la matrice contenant les poids sur les arcs entre la couche $h-1$ et la couche h
- On note $b_h \in \mathbb{R}^{n_h}$ le biais ajouté à la couche h
- On note σ la fonction d'activation appliquée à chaque couche
  - elle applique la même fonction à chaque entrée d'un vecteur en général
  - peut dépendre de h en général

La sortie du réseau est calculée selon le schéma suivant :

$$\begin{cases} f_0(x) = x \\ f_h(x) = \sigma(W_h f_{h-1}(x) + b_h) \in \mathbb{R}^{n_h} \quad \forall h = 1, \ldots, H \end{cases} \tag{1.1.1}$$

1.1.2 Pourquoi est-il important d'étudier la théorie de l'apprentissage profond ?

Pour l'instant, la théorie de l'apprentissage profond est toujours en retard sur les résultats impressionnants observés en pratique dans les différents domaines où les réseaux de neurones ont été appliqués avec succès. Les résultats concernant les aspects mathématiques des réseaux de neurones sont encore descriptifs, en d'autres termes, nous essayons simplement de comprendre ou d'expliquer les phénomènes que nous observons en pratique et nous utilisons souvent des hypothèses simplificatrices pour les rendre abordables. C'est un bon début, mais l'objectif ultime serait de parvenir à une théorie prescriptive, c'est-à-dire à des résultats théoriques que nous pourrions utiliser pour améliorer les algorithmes ou choisir la bonne architecture pour le bon type de données, au lieu de passer des heures à essayer différentes architectures et combinaisons d'hyperparamètres pour voir ce que nous obtenons. Espérons qu'un jour nous pourrons atteindre ce genre de résultat.

1.2 Apprentissage supervisé

Dans cette section, nous rappelons les bases de l'apprentissage supervisé, qui est le cadre le plus utilisé en apprentissage automatique et en apprentissage profond en particulier.

En apprentissage supervisé, nous donnons à notre modèle un échantillon contenant un ensemble d'exemples (par exemple, des images) avec leurs étiquettes (par exemple, chats ou chiens). L'objectif est qu'après la phase d'apprentissage (qui correspond à l'apprentissage des poids pour un réseau de neurones), notre algorithme puisse étiqueter correctement des exemples qu'il n'a jamais vus auparavant.

1.2.1 Décomposition du risque

— Soit $(f_{\mathbf{w}})_{\mathbf{w}}$ une famille de fonctions paramètrée par $\mathbf{w}$ (par exemple, des réseaux de neurones à architecture et activation de fonction fixées et dont les poids varient)

Pour un échantillon d'entraînement $(x_i, y_i)_{i=1..N}$ i.i.d tiré d'une probabilité $\mathcal{P}$ sur $\mathcal{X} \times \mathcal{Y}$ et une perte $\ell : \mathcal{Y} \times \mathcal{Y} \longmapsto \mathbb{R}$, on pose :

— Le risque :

$$R(f_{\mathbf{w}}) = \mathbb{E}\left(\ell(f_{\mathbf{w}}(X), Y)\right).$$

— Le risque empirique :

$$\widehat{R}(f_{\mathbf{w}}) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_{\mathbf{w}}(x_i), y_i).$$

— $R^* = \inf_g R(g)$ le risque optimal, aussi appelé le risque de Bayes.

— On fixe $\mathbf{w}^*$ et $\widehat{\mathbf{w}}$ tel que :

$$\mathbf{w}^* \in \arg\min_{\mathbf{w}} R(f_{\mathbf{w}}) \qquad \text{and} \qquad \widehat{\mathbf{w}} \in \arg\min_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}}).$$

Si le minimiseur n'existe pas, on considère un ε-minimiseur.

En apprentissage supervisé, généralement, le but est d'essayer de trouver $\mathbf{w}$ qui minimise l'excès de risque, qui est la différence entre le risque pour $f_{\mathbf{w}}$ et le risque optimal R^*, cet excès de risque peut être décomposé en trois erreurs principales :

$$\begin{array}{rrcll}
0 \leq & R(f_{\mathbf{w}}) & - & R^* & \text{(Excès de risque)} \\
= & R(f_{\mathbf{w}}) & - & \widehat{R}(f_{\mathbf{w}}) & \text{(Erreur de généralisation)} \\
+ & \widehat{R}(f_{\mathbf{w}}) & - & \widehat{R}(f_{\widehat{\mathbf{w}}}) & \text{(Erreur d'optimisation)} \\
+ & \widehat{R}(f_{\widehat{\mathbf{w}}}) & - & \widehat{R}(f_{\mathbf{w}^*}) & \leq 0 \\
+ & \widehat{R}(f_{\mathbf{w}^*}) & - & R(f_{\mathbf{w}^*}) & \text{(Erreur de généralisation)} \\
+ & R(f_{\mathbf{w}^*}) & - & R^* & \text{(Erreur d'approximation)}
\end{array}$$

Les différentes erreurs contribuent à l'excès de risque, et l'on souhaite que chacune d'entre elles soit la plus petite possible. Nous allons les décrire en détail dans les prochaines sections.

1.2.2 Erreur d'optimisation

Comme présenté dans la précédente décomposition du risque, l'erreur d'optimisation est liée au risque empirique. Et l'on souhaite la rendre aussi faible que possible sans augmenter

les autres erreurs. En général, nous exécutons un algorithme d'optimisation, par exemple la descente de gradient (stochastique) ou des variantes comme Adam [79] et RMSprop [136], et nous nous arrêtons une fois que l'algorithme a convergé ou si nous passons quelques "epochs" sans diminuer le risque empirique. Une question naturelle se pose donc : vers quels points ces algorithmes peuvent-ils converger ou près de quels points peuvent-ils rester bloqués pendant un certain temps. Si nous nous concentrons sur l'algorithme de descente de gradient par exemple, ces points sont ceux où le gradient s'annule, ils sont appelés points critiques ou stationnaires (du premier ordre).

Par conséquent, une question naturelle est de savoir vers quel type de points critiques les algorithmes peuvent converger et quelles sont leurs propriétés. Existe-t-il des minima locaux, des minima globaux ou des points de selle ? L'erreur d'optimisation est-elle faible à ces points ? Qu'en est-il de l'erreur de généralisation ? Y a-t-il une sorte de biais implicite associé à ces points ? Il est donc nécessaire d'étudier le paysage de la fonction objectif $\hat{R}(w)$ afin de mieux comprendre les performances de l'algorithme d'optimisation.

Notre premier travail s'inscrit dans cette catégorie, une description plus précise peut être trouvée dans la Section 2.3 et le Chapitre 3.

1.2.3 Erreur de généralisation

La deuxième erreur est l'erreur de généralisation, qui est liée à l'efficacité de l'algorithme lorsqu'il prédit l'étiquette d'un nouveau point de données non vu. Classiquement, les gens ont prouvé des bornes uniformes sur cette quantité en la reliant à une notion de complexité de l'espace des fonctions dans lequel nous cherchons une solution. Par exemple, la VC-dimension ou la complexité de Rademacher sont des notions qui interviennent dans ces bornes. Cependant, ces bornes sont lâches pour les réseaux de neurones qui sont utilisés en pratique. En effet, ils ont beaucoup plus de paramètres que le nombre d'exemples de l'échantillon sur lequel ils sont entraînés.

Malgré cela, le réseau de neurones appris réussit à bien généraliser. Récemment, un phénomène "étrange" a été observé dans la pratique, appelé la "double descente". En effet, la théorie classique de l'apprentissage statistique nous dit que l'apprentissage parfait des données est généralement associé à un surapprentissage et donc à une mauvaise erreur de généralisation. Cependant, il a été observé (e.g., [15]) que les réseaux de neurones peuvent apprendre parfaitement les données d'entraînement tout en généralisant bien à des points de données non vus.

Un outil populaire utilisé dans la littérature de la théorie du deep learning pour prouver des résultats de généralisation et d'optimisation pour les réseaux de neurones est le "Neural Tangent Kernel" [67]. Il s'agit d'un noyau qui décrit l'évolution des réseaux de neurones profonds pendant leur entraînement par descente de gradient. En particulier pour les réseaux larges, ceci permet de se ramener à des méthodes de noyaux pour l'analyse des propriétés de réseaux de neurones.

Régularisation implicite : De nombreux travaux récents tentent d'expliquer le succès de l'apprentissage profond par le biais du biais implicite (régularisation) induit par des algorithmes tels que la descente de gradient stochastique (SGD). En effet, il semble que pour

les réseaux de neurones surparamétrés qui peuvent parfaitement apprendre l'échantillon d'entraînement, il existe un grand nombre de solutions qui interpolent l'échantillon d'entraînement. Parmi toutes ces solutions, beaucoup ont de très mauvaises erreurs de généralisation (grande), cependant, les algorithmes simples comme SGD semblent choisir celle avec une complexité minimale qui résulte en une bonne erreur de généralisation (petite). La notion de complexité est difficile à définir globalement, mais elle est généralement considérée comme une norme particulière.

Ce qui est impressionnant, c'est qu'en présence de données bruitées, le réseau de neurones s'ajuste parfaitement aux données et est encore capable d'avoir une erreur de test proche du risque de Bayes. Par conséquent, toutes les analyses menées précédemment pour expliquer la généralisation pour les autres modèles, basées sur la dimension VC ou la complexité de Rademacher par exemple ne suffisent pas à expliquer le succès de l'apprentissage profond. Plusieurs auteurs ont essayé d'expliquer ce phénomène dans différents contextes à travers la notion du surapprentissage bénin (e.g., [15, 58, 13, 22, 139, 165, 92, 40]).

1.2.4 Erreur d'approximation

La dernière erreur apparaissant dans la décomposition précédente est l'erreur d'approximation qui est liée à l'expressivité de l'espace des fonctions dans lequel nous recherchons un bon prédicteur. Plus cet espace est riche, plus l'erreur d'approximation est faible et de nombreux travaux ont établi des bornes supérieures et/ou inférieures sur l'erreur d'approximation dans le pire des cas d'une classe de fonctions par une fonction dans un certain espace d'hypothèses. Lorsque l'espace d'hypothèses est l'ensemble des fonctions implémentées par des réseaux de neurones d'une certaine architecture, plusieurs résultats ont été prouvés. Un aperçu plus approfondi de la littérature peut être trouvé dans la section 2.5.

Notre troisième travail se situe dans cette catégorie, une description plus précise peut être trouvée dans la Section 2.5 et le Chapitre 5.

1.2.5 Robustesse

En plus des trois erreurs que nous avons vues dans les sections précédentes, il existe d'autres aspects que l'on peut vouloir optimiser en raison de certains problèmes que l'on peut rencontrer en pratique lorsqu'on travaille avec des réseaux de neurones. En effet, les attaques adverses constituent un problème commun observé pour de nombreuses architectures d'apprentissage profond utilisées dans la pratique. Il existe de nombreux types d'attaques adverses en apprentissage automatique, l'exemple le plus classique est le suivant : le modèle prédit correctement une voiture pour une image contenant une voiture, mais lorsqu'on ajoute un petit bruit (invisible à l'œil humain), l'image de la voiture est toujours la même pour nous, humains, mais le modèle la prédit comme une autruche. Bien que la perturbation soit à peine perceptible à l'œil humain, elle peut être suffisante pour modifier radicalement la décision du modèle. Cela est dû au fait que la fonction calculée par le réseau de neurones peut être très oscillatoire. Pensez à la situation où une voiture autonome doit identifier un panneau stop et où, à cause d'un petit dessin sur le côté du panneau, elle le prédit comme étant autre chose. Cela peut être problématique et peut coûter des vies au final, c'est pourquoi

nous voulons éviter ces situations. Ainsi il est très important de concevoir des algorithmes robustes.

Réseaux Lipschitz et généralisation avec des exemples adversariaux : Une façon de parvenir à des modèles robustes est de construire des fonctions d'hypothèses Lipschitz de telle sorte que les petits changements en entrée n'affectent pas la sortie. Dans ce contexte d'apprentissage profond, les réseaux de neurones Lipschitz et en particulier les réseaux de neurones 1-Lipschitz ont été étudiés. Pour ce faire, on contraint chaque couche à être 1-Lipschitz et on utilise le fait que pour deux fonctions Lipshitz f et g, la propriété suivante s'applique : $Lip(f \circ g) \leq Lip(f)Lip(g)$, où $Lip(f)$ désigne la constante de Lipschitz par rapport à la norme euclidienne.

On rappelle quelques notions liées à la robustesse et comment la régularité Lipschitz améliore ceci. Ceci est connu et on peut le trouver par exemple dans [27]. On a vu précédemment qu'en général on veut minimiser le vrai risque $R(\mathbf{w}) = \mathbb{E}\left([\ell(f_{\mathbf{w}}(X), Y)]\right)$. Quand il s'agit de créer des modèles robustes, on préfère plutôt minimiser en $\mathbf{w}$ le risque robuste défini comme suit :

$$R(\mathbf{w}, p, \epsilon) = \mathbb{E}\left(\left[\max_{\tilde{x}:\|\tilde{x}-x\|_p \leq \epsilon} \ell(f_{\mathbf{w}}(X), Y)\right]\right),$$

pour p et ε fixés. Par définition, $R(\mathbf{w}) \leq R(\mathbf{w}, p, \epsilon)$ pour tout p et $\epsilon > 0$. Si ℓ est 1-Lipschitz et Λ_p désigne la constante de Lipschitz du réseau de neurones on a

$$R(\mathbf{w}, p, \epsilon) \leq R(\mathbf{w}) + \mathbb{E}\left(\left[\max_{\tilde{x}:\|\tilde{x}-x\|_p \leq \epsilon} |\ell(f_{\mathbf{w}}(\tilde{x}), Y) - \ell(f_{\mathbf{w}}(x), Y)|\right]\right) \leq R(\mathbf{w}) + \Lambda_p \epsilon.$$

Ceci explique pourquoi les réseaux Lipschitz sont plus robustes.

Les réseaux Lipschitz peuvent aussi améliorer la généralisation [147]. Si on note $C_p(\mathcal{X}, \gamma)$ le nombre de recouvrement (covering number) de $\mathcal{X}$ utilisant des boules γ pour $\|.\|_p$. En notant $M = \sup_{x,\mathbf{w},y} \ell(g(x, \mathbf{w}), y)$, [147] implique que pour tout $\delta \in (0, 1)$, avec probabilité $1 - \delta$ sur l'échantillon i.i.d $(x_i, y_i)_{i=1}^m$, on a :

$$R(\mathbf{w}) \leq \frac{1}{m} \sum_{i=1}^{m} \ell(g(x_i, \mathbf{w}), y_i) + \Lambda_p \gamma + M \sqrt{\frac{2Y C_p(\mathcal{X}, \frac{\gamma}{2}) \ln(2) - 2\ln(\delta)}{m}}$$

Notre deuxième travail se situe dans cette catégorie, une description plus précise se trouve dans la Section 2.4 et le Chapitre 4.

1.2.6 Autre domaines de la théorie de l'apprentissage profond

De nombreux autres domaines des aspects mathématiques de l'apprentissage profond ont été et sont encore étudiés.

— Certification : Les réseaux de neurones sont généralement utilisés pour donner un simple nombre ou un vecteur en sortie. Cependant, on peut vouloir quantifier l'incerti-

tude pour une prédiction donnée et donner par exemple un intervalle de confiance pour la prédiction. Ceci est par exemple réalisé à l'aide de réseaux de neurones bayésiens.

— Explicabilité : Les réseaux de neurones sont souvent considérés comme un modèle boîte noire, car ils nous donnent des prédictions difficiles à expliquer contrairement à la plupart des algorithmes classiques comme la régression logistique par exemple. De nombreux auteurs ont essayé d'ouvrir la boîte noire et d'examiner les caractéristiques utilisées par le réseau de neurones pour donner une certaine prédiction. Il s'agit d'un problème important puisque la législation exige que les entreprises soient en mesure d'expliquer leurs décisions. Par exemple un banquier doit expliquer à un client pourquoi sa demande de crédit a été rejetée et ne pas se contenter de dire "notre modèle de réseau de neurones dit non". Certains auteurs ont étudié ce problème en utilisant différentes méthodes, par exemple des méthodes basées sur le gradient, comme l'analyse de sensibilité, des méthodes basées sur la rétroaction, des méthodes basées sur des modèles de substitution ou même des méthodes de théorie des jeux.

1.3 Paysage du risque empirique pour les réseaux de neurones

Comme nous l'avons vu précédemment, l'erreur d'optimisation est l'une des trois erreurs de la décomposition du risque que l'on doit essayer de rendre faible. En pratique, les méthodes basées sur le gradient semblent faire du bon travail malgré une fonction de perte fortement non convexe.

Une des directions de recherche pour tenter d'expliquer ce phénomène est d'étudier le paysage de la fonction objectif des réseaux de neurones profonds en caractérisant leurs points critiques.

Nous commençons par rappeler quelques définitions clés relatives au paysage d'une fonction.

1.3.1 Rappel : minimiseurs, points critiques d'ordre 1 ou 2, points-selles stricts et non stricts

Rappelons les définitions des structures locales du paysage du risque empirique, qui sont importantes du point de vue statistique et de celui de l'optimisation.

Pour $\boldsymbol{w} \in \mathbb{R}^n$, on désigne par $\boldsymbol{w} \longmapsto L(\boldsymbol{w})$ la fonction que nous voulons minimiser. Supposons que $\boldsymbol{w} \longmapsto L(\boldsymbol{w})$ est C^2, et on désigne par ∇L et $\nabla^2 L$ son gradient et sa Hessienne. [1] On écrit également $A \succeq 0$ pour dire qu'une matrice $A \in \mathbb{R}^{n \times n}$ est semi-définie positive. Rappelons les quatre définitions suivantes, qui sont imbriquées :

- $\boldsymbol{w}^\star$ est un **minimiseur global** si et seulement si $\forall \boldsymbol{w} \in \mathbb{R}^n,\ L(\boldsymbol{w}^*) \leq L(\boldsymbol{w})$.
- $\boldsymbol{w}^\star$ est un **minimiseur local** si et seulement si il existe un voisinage $\mathcal{O} \subset \mathbb{R}^n$ de $\boldsymbol{w}^\star$ tel que $\forall \boldsymbol{w} \in \mathcal{O},\ L(\boldsymbol{w}^*) \leq L(\boldsymbol{w})$.

1. Lorsque le paramètre d'entrée n'est pas un vecteur, mais, par exemple, une séquence de matrices, les mêmes définitions s'appliquent, où le gradient et la Hessienne sont calculés par rapport à la version vectorisée des paramètres d'entrée.

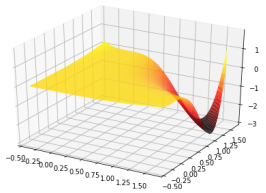

FIGURE 1.1 – Exemple de paysage avec un plateau (point-selle non-strict) et un minimiseur global.

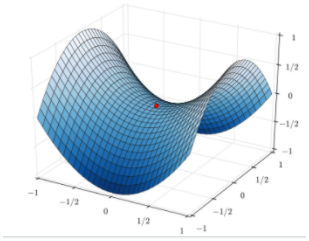

FIGURE 1.2 – Exemple de paysage avec un point-selle strict en (0,0).

- $\boldsymbol{w}^*$ est un **point critique du second ordre** si et seulement si $\nabla L(\boldsymbol{w}^*) = 0$ et $\nabla^2 L(\boldsymbol{w}^*) \succeq 0$. Si, au contraire, la Hessienne a une valeur propre négative, on dit que le point a une courbure négative.
- $\boldsymbol{w}^*$ est un **point critique du premier ordre** si et seulement si $\nabla L(\boldsymbol{w}^*) = 0$.

  Nous pouvons également distinguer un type spécifique de point critique du premier ordre : les points-selles. Comme nous le verrons plus loin, ils peuvent être des points critiques du second ordre ou pas.
- $\boldsymbol{w}^*$ est un **point-selle** si et seulement si c'est un point critique du premier ordre qui n'est ni un minimiseur local, ni un maximiseur local.
  - Un point-selle $\boldsymbol{w}^*$ est **strict** si et seulement si ce n'est pas un point critique du second ordre (c'est-à-dire que la Hessienne $\nabla^2 L(\boldsymbol{w}^*)$ a une valeur propre négative). La Figure 1.2 en donne un exemple.
  - Un point-selle $\boldsymbol{w}^*$ est **non-strict** si et seulement si c'est un point critique du second ordre. Dans ce cas, la Hessienne $\nabla^2 L(\boldsymbol{w}^*)$ est semi-définie positive et possède au moins une valeur propre égale à zéro. Typiquement, dans la direction des vecteurs propres correspondants, un terme d'ordre supérieur en fait un point-selle (par exemple, $L(\boldsymbol{w}) = \sum_{i=1}^{n} w_i^3$ en $\boldsymbol{w}^* = 0$). La figure 1.1 en donne un exemple.

1.3.2 Algorithmes basés sur le gradient

En matière d'apprentissage automatique, l'objectif est de minimiser le risque empirique en espérant que le risque réel sera proche du risque empirique, donc également faible. L'algorithme le plus basique utilisé est la descente de gradient, qui consiste à suivre la direction de plus grande descente, celle donnée par l'opposé du gradient à ce point. Plus formellement, la descente de gradient est un algorithme du premier ordre qui est utilisé en optimisation pour tenter de trouver un point critique (stationnaire) d'une fonction différentiable L paramétrée par θ, c'est-à-dire un point où le gradient de la fonction s'annule ($\nabla_\theta L(\theta) = 0$). Les itérations de l'algorithme sont caractérisées par l'équation suivante :

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t) \tag{1.3.1}$$

où $\eta > 0$ est un taux d'apprentissage (learning rate) qui peut varier pendant le processus d'optimisation. Lorsqu'elle est appliquée à de très grands ensembles de données, la descente de gradient peut être très lente.

Rappelons que le risque empirique que nous voulons optimiser est défini par $\widehat{R}(f_{\mathbf{w}}) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_{\mathbf{w}}(x_i), y_i)$. Lorsqu'il y a plusieurs millions d'exemples, l'approche de descente de gradient peut prendre beaucoup de temps puisque chaque itération nécessite une prédiction pour chaque occurrence dans l'ensemble de données d'apprentissage. Par conséquent, nous pouvons utiliser une variante appelée descente de gradient stochastique (SGD). Au lieu de mettre à jour le paramètreaprès avoir parcouru l'ensemble des données, la SGD le fait pour chaque donnée d'entraînement, et on effectue généralement plusieurs passages sur les données de formation.

En pratique, plutôt que d'utiliser l'ensemble des exemples ou un seul exemple à la fois, nous prenons généralement un mini-batch composé de quelques exemples pour chaque itération de l'algorithme.

D'autres heuristiques peuvent être ajoutées au processus pour rendre l'optimisation plus rapide, par exemple les taux d'apprentissage (learning rate) adaptatifs (par exemple Adagrad, RMSProp), qui consistent à modifier le taux d'apprentissage pendant le processus de formation en fonction des gradients précédents. Un autre algorithme qui a gagné en popularité parmi les praticiens de l'apprentissage profond est Adam. Adam peut être considéré comme une combinaison de taux d'apprentissage adaptatif et de momentum. Nous ne donnons pas les formulations mathématiques des algorithmes mais elles peuvent être trouvées par exemple dans [47].

1.3.3 Importance de l'analyse du paysage à l'ordre 2

Dans cette section, nous commençons par expliquer l'importance d'étudier le paysage de la fonction objective à l'ordre 2, puis nous illustrons ceci sur quelques exemples simples.

1.3.3.1 Motivation

Quand la fonction que nous essayons de minimiser est lisse, convexe, et possède un minimiseur global, l'algorithme de descente de gradient avec un taux d'apprentissage bien choisi converge vers un point critique du premier ordre, et ce point critique est un minimiseur global [105]. Cependant, en général, la recherche d'un optimum global d'une fonction non convexe est un problème NP-complet [104] ; c'est notamment le cas pour un simple réseau de neurones à 3 noeuds [20]. Malgré cela, lors de l'optimisation des réseaux de neurones, la pratique courante reste l'utilisation d'algorithmes basés sur le gradient.

On sait depuis des décennies que, même dans un cadre non convexe, les algorithmes basés sur le gradient convergent vers un point critique du premier ordre, dans le sens où les itérés produits par l'algorithme peuvent atteindre un gradient arbitrairement petit après un nombre fini (polynomial) d'itérations [105]. En ajoutant des conditions de régularité, des travaux récents ont montré que les algorithmes classiques du premier ordre échappent aux points-selles stricts à long terme [87, 85], et que certains d'entre eux peuvent atteindre en un temps polynomial un point critique presque du second ordre. Plus précisément, il est

généralement démontré que, avec une probabilité élevée, ces algorithmes peuvent atteindre en un temps polynomial un point avec un gradient arbitrairement petit et une Hessiene semi-définie quasi-positive. [71, 73, 29, 72]. Cependant, rien n'empêche ces algorithmes de converger vers des points-selles non stricts ou de passer de nombreuses "epochs" dans leur voisinage, ce qui se traduit par un long plateau pendant l'apprentissage. Nous donnons en Section 2.3.3.2 une intuition des problèmes rencontrés avec ce type de points critiques avec des fonctions simples.

Pour voir que ce comportement se produit réellement en pratique, considérons l'expérience simple dont les résultats sont montrés dans les Figures 1.3 et 1.4 (plus de détails dans le Chapitre 3, Annexe 3.G). Pour chaque exécution de cette expérience, les paramètres d'un réseau de neurones linéaire de profondeur 5 sont optimisés pour s'adapter à des paires d'entrée/sortie aléatoires. L'écart est mesuré avec la perte des moindres carrés et nous utilisons l'optimiseur ADAM. Selon l'exécution, l'algorithme est initialisé au voisinage d'un point-selle strict (en rouge) ou d'un point-selle non strict (en bleu). La distance entre l'itération initiale aléatoire et le point-selle n'est volontairement pas négligeable : elle est fixée à environ 10% de la norme du point-selle. La figure 1.3 montre l'évolution typique de la perte pour les deux cas. Nous pouvons voir qu'ADAM s'échappe rapidement du point-selle strict, mais qu'il lui faut de nombreuses "epochs" pour s'échapper du plateau à proximité du point-selle non strict. La figure 1.4 montre que cette observation se généralise à la plupart des exécutions. Nous comparons les distributions empiriques d'un moment aléatoire (appelé *"epoch" d'échappement*) défini comme l'epoch à laquelle la perte a significativement diminué par rapport à sa valeur initiale. Lorsqu'il est initialisé à proximité de points-selles non stricts, l'algorithme souffre d'une "epoch" d'échappement souvent importante et peut s'y arrêter, sans qu'il soit possible de distinguer ce point-selle non strict d'un minimum local. Améliorer l'analyse au-delà des minimiseurs locaux et caractériser les points-selles stricts et non stricts est donc essentiel pour comprendre la dynamique de la descente de gradient et la régularisation implicite.

Au-delà des réseaux de neurones, l'étude du paysage des fonctions de pertes de problèmes d'optimisation non convexes spécifiques a révélé qu'ils sont traitables : récupération de phase [130], apprentissage de dictionnaire [131], décomposition tensorielle [42, 43, 39] et autres [162, 103]. En effet, une propriété du paysage qui est partagée par la plupart de ces problèmes est que chaque point critique est soit un minimiseur global, ou bien il possède une courbure négative. En d'autres termes, chaque point critique du second ordre est un minimiseur global. Pour ces problèmes, il existe des algorithmes du premier ordre qui convergent de manière prouvée vers des minimiseurs globaux.

La compréhension générale du paysage n'est pas aussi bonne pour les réseaux de neurones. Un régime qui a été largement étudié est le régime surparamétré (voir [133] et [132] pour une revue), où il a été prouvé sous certaines hypothèses que pour un large réseau de neurones non linéaires entièrement connectés, presque tous les minima locaux sont des minima globaux [108], ou qu'il n'y a pas de vallées sous-optimale [107].

De nombreux travaux récents se sont concentrés sur les réseaux de neurones linéaires, malgré le fait qu'ils sont rarement utilisés pour résoudre des applications du monde réel. Ils calculent en effet une application linéaire entre les espaces d'entrée et de sortie. La motivation de ces études est que le risque empirique des réseaux linéaires est hautement non

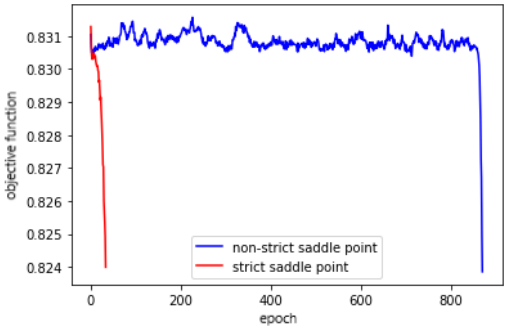

FIGURE 1.3 – La fonction objective pendant le processus itératif, lorsqu'on initialise autour d'un point-selle strict (en rouge) ou d'un point-selle non strict (en bleu).

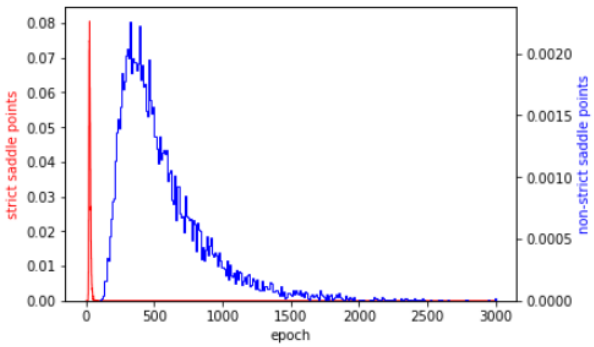

FIGURE 1.4 – Histogramme des epochs d'échappement, lorsqu'on initialise autour d'un point-selle strict (en rouge) ou non-strict (en bleu). Pour plus de clarté, l'axe y est doté de deux échelles. L'axe de droite correspond à la courbe bleue et celui de gauche à la courbe rouge.

convexe et partage des propriétés similaires à celui des réseaux de neurones non linéaires utilisés en pratique. Une revue complète des publications scientifiques sur le paysage des réseaux linéaires est donnée dans la section 2.3.4 mais nous voudrions souligner à ce stade que de nombreux résultats sont fortement liés aux propriétés du paysage à l'ordre 2.

Comme l'a montré [120], les réseaux linéaires présentent des phénomènes d'apprentissage non linéaires similaires à ceux observés lors de l'optimisation des réseaux non linéaires, notamment de longs plateaux suivis de transitions rapides vers des solutions de plus faible erreur, comme sur la figure 2.3. Plusieurs travaux ont suivi (par exemple, [44, 45]) avec des preuves formelles dans plusieurs cas particuliers. Ils expriment que pour des initialisations particulières, la trajectoire des itérations passe un certain temps sur des plateaux dont l'emplacement définit une régularisation implicite, si l'algorithme est arrêté tôt.

D'autres auteurs fournissent des initialisations et/ou des architectures bien choisies pour lesquelles la convergence vers un minimiseur global peut être établie : [5, 11, 36, 33].

La différence entre la régularisation implicite et la convergence globale dépend du fait que la trajectoire des itérés considérée dans ces articles évite ou non les points-selles. Pour mieux comprendre les propriétés de convergence des algorithmes du premier ordre, pour donner quelques indices sur leur dynamique en temps fini et pour mieux comprendre la régularisation implicite, une analyse détaillée du paysage empirique des risques à l'ordre 2 est nécessaire.

En outre, bien qu'il ait été prouvé sous des conditions peu contraignantes que tout minimiseur local est un minimiseur global et qu'il n'existe pas de maximiseur local [75, 156], on sait très peu de choses sur les points-selles et le paysage à l'ordre 2. Pour les réseaux linéaires à une couche cachée, un fait prouvé est que chaque point-selle est strict, ce qui conduit à la propriété que chaque point critique d'ordre 2 est un minimiseur global [164,

75]. Malheureusement, ce n'est pas le cas pour les réseaux linéaires à deux couches cachées ou plus. Pour de tels réseaux de neurones, il a seulement été noté par [75] qu'il existe des points-selles non stricts (par exemple, lorsque toutes les matrices de poids sont égales à 0).

Dans le Chapitre 3, nous faisons le pas additionnel manquant et caractérisons complètement le paysage du risque empirique à l'ordre 2, pour la perte des moindres carrés et les réseaux linéaires profonds. En particulier, nous dérivons une condition simple nécessaire et suffisante pour qu'un point critique du risque empirique au premier ordre soit un minimiseur global, un point-selle strict ou un point-selle non strict.

1.3.3.2 Points-selles stricts et non stricts : quelques exemples simples

Les points-selles stricts ne sont pas des attracteurs Dans ce paragraphe, nous donnons un exemple simple d'une fonction avec un point-selle strict, le but est de prouver que la descente de gradient ne converge pas vers le point-selle strict pour presque toutes les initialisations ([86]) dans un cas simple pour avoir une intuition. Nous allons prouver que la trajectoire du flot de gradient (gradient flow) évite les points-selles stricts. Pour une fonction $f : w \to f(w)$, le flot de gradient (gradient flow), qui est la variante continue de la descente de gradient lorque le pas est infinitésimale, est défini par l'équation suivante :

$$\frac{dw}{dt} = -\frac{\partial f}{\partial w}(w), \tag{1.3.2}$$

où nous initialisons l'algorithme à un point $w(0) = w_0$.

Considèrons la fonction $f(x, y) = x^2 - y^2$. On a $\frac{\partial f}{\partial x}(x, y) = 2x$ et $\frac{\partial f}{\partial y}(x, y) = -2y$. Aussi, $\frac{\partial^2 f}{\partial x^2}(x, y) = 2$, $\frac{\partial^2 f}{\partial y^2}(x, y) = -2$ et $\frac{\partial^2 f}{\partial x \partial y}(x, y) = 0$. Par conséquent, le seul point critique est $(0, 0)$, et d'après la Hessienne, c'est un point-selle strict, car les valeurs propres de la matrice Hessienne sont -2 et 2.

Ains les équations du flot de gradient sont

$$\begin{cases} \frac{dx}{dt} = -\frac{\partial f}{\partial x} = -2x \\ \frac{dx}{dt} = -\frac{\partial f}{\partial y} = 2y \end{cases}$$

où on pose $x(0) = x_0$ et $y(0) = y_0$. Ainsi

$$\begin{cases} x(t) = x_0 \exp(-2t) \\ y(t) = y_0 \exp(2t) \end{cases}$$

On a $\lim_{t\to\infty} x(t) = 0$. Si $y_0 = 0$, alors $\forall t \geq 0, y(t) = 0$. Si $y_0 \neq 0$, alors $\lim_{t\to\infty} y(t) = \infty$. Par conséquent, les seules initialisations qui conduisent à une convergence vers le point-selle strict sont les points situés sur l'axe des x. Par conséquent, le bassin d'attraction de ce point-selle strict est de mesure 0. On peut facilement généraliser les résultats précédents au cas où la fonction est de plus de 2 variables.

Exemple d'un point-selle non-strict attracteur : Contrairement aux points-selles stricts, les points-selles non stricts peuvent avoir un bassin d'attraction de mesure de Lebesgue

positive. Dans cette section, nous donnons un exemple simple où cela peut se produire. Considérons la fonction $g(x) = x^3$. Nous avons $g'(x) = 3x^2$ et $g''(x) = 6x$. Par conséquent, le seul point critique de cette fonction est le point 0 et c'est un point-selle non strict. L'équation du flot de gradient devient

$$\frac{dx}{dt} = -\frac{\partial g}{\partial x} = -3x^2$$

où nous fixons $x(0) = x_0 \neq 0$.

Ainsi, nous avons

$$-\frac{dx}{x^2} = 3dt \implies \frac{1}{x(t)} = 3t + \frac{1}{x_0} \implies x(t) = \frac{1}{3t + \frac{1}{x_0}}.$$

Par conséquent, $\lim_{t\to\infty} x(t) = 0$. Ainsi, le flot de gradient converge vers le point-selle 0 pour toutes les initialisations $x_0 > 0$. Par conséquent, le bassin d'attraction du point-selle 0 non strict est de mesure de Lebesgue positive.

Ralentissement du flot de gradient : Dans les paragraphes précédents, nous avons illustré sur des exemples que le flot de gradient évite presque sûrement les points-selles stricts, mais que pour les points-selles non stricts, ce n'est pas toujours le cas.

Un autre aspect lié aux points-selles est que, puisqu'il s'agit de points où le gradient s'annule, l'algorithme du flot du gradient peut être ralenti à proximité de ces points. Dans cette section, nous allons explorer ce phénomène dans des situations simples et voir comment ce ralentissement se compare entre les points-selles stricts et non stricts.

Nous considérerons deux fonctions et comparerons le temps que prend le flot de gradient pour s'échapper du voisinage de chaque point-selle.

Considèrons d'abord la fonction $f(x, y) = x^2 - y^2$. Rappelons que $(0, 0)$ est un point-selle strict de f. On a vu précedemment que le flot de gradient donne

$$\begin{cases} x_f(t) = x_0 \exp(-2t) \\ y_f(t) = y_0 \exp(2t) \end{cases}$$

où $x_f(0) = x_0$ et $y_f(0) = y_0 > 0$.

La deuxième fonction qu'on considère est $g(x, y) = x^2 - y^4$. Onpeut facilement vérifier que le point critique $(0, 0)$ est un point-selle non-strict.

Dans ce cas, le flot de gradient (avec $x_g(0) = x_0$ et $y_g(0) = y_0$) donne $x_g(t) = x_0 \exp(-2t)$ et

$$\frac{dy_g}{dt} = -\frac{\partial g}{\partial y} = 4y^3 \implies -\frac{y_g'(t)}{y_g^3} = -4 \implies \frac{1}{y_g^2(t)} = -8t + \frac{1}{y_0^2} \implies y_g(t) = \frac{1}{\sqrt{-8t + \frac{1}{y_0^2}}}$$

On a $x_f(t) = x_g(t)$ et $\lim_{t\to\infty} x_f(t) = 0$. Ainsi, il suffit de comparer ce qui se passe au niveau de l'axe des y. On considère que le flot de gradient s'est échappé du point-selle quand il

atteint $y = 1$. Ceci a du sens puisque dans ce cas $y^2 = y^4 = 1$.

On a $y_f(t_f) = 1 \iff y_0 \exp(2t_f) = 1 \iff t_f = \frac{1}{2}\ln(\frac{1}{y_0})$.

D'un autre côté $y_g(t_g) = 1 \iff \frac{1}{\sqrt{-8t_g + \frac{1}{y_0^2}}} = 1 \iff t_g = \frac{1}{8}(\frac{1}{y_0^4} - 1)$.

Dans les deux cas, plus y_0 est proche de 0, plus il est difficile de s'échapper du point-selle et nous avons $\lim_{y_0 \to 0} t_g = \lim_{y_0 \to 0} t_f = \infty$. Cependant, lorsque nous initialisons près du point-selle, le temps d'échappement pour un point-selle strict est logarithmique en $\frac{1}{y_0}$, alors qu'il est polynomial en $\frac{1}{y_0}$ lorsqu'il s'agit de points-selles non stricts, et nous avons $\lim_{y_0 \to 0} \frac{t_g}{t_f} = \infty$. Par conséquent, il faut beaucoup plus de temps pour s'échapper du voisinage du point-selle non strict que du voisinage du point-selle strict. L'effet de ralentissement du point-selle non strict est plus fort que celui du point-selle strict.

1.3.4 Revue de littérature

L'étude des réseaux de neurones linéaires peut être divisée en deux catégories. La première ligne de recherche concerne la géométrie du paysage du risque empirique pour les réseaux de neurones linéaires, tandis que la deuxième ligne concerne la trajectoire de la dynamique de descente de gradient pour les réseaux linéaires. Notre travail se situe dans la première catégorie.

Paysage géométrique pour les réseaux linéaires : Tout a commencé avec [8]. Les auteurs ont prouvé que pour un réseau linéaire à une couche cachée, sous certaines conditions sur les matrices de données, et pour la perte des moindres carrés, tout minimiseur local est un minimiseur global. [75] a par la suite généralisé et étendu ce résultat aux réseaux de neurones linéaires profonds sous certaines conditions et a de nouveau prouvé que chaque minimiseur local est un minimiseur global (cette partie a été prouvée plus tard par [93] avec des hypothèses plus faibles sur les données et des preuves plus simples). Cet auteur a également prouvé que tout autre point critique est un point-selle, que pour un réseau linéaire à une couche cachée, tous les points-selles sont stricts, tandis que pour les réseaux plus profonds, il existe des points-selles non stricts ([75] présente un espace de points-selles non stricts où toutes les matrices de poids sauf une sont égales à zéro). [156] a donné une condition pour qu'un point critique soit un minimiseur global ou un point-selle. [163] a supprimé toutes les hypothèses sur les données et a donné des formes analytiques pour les points critiques du risque empirique. Dans la caractérisation, les matrices de poids sont définies de manière récursive et peuvent être trouvées en résolvant des équations ; en particulier, ils ont donné une caractérisation des minimiseurs globaux. [109] ont montré qu'en utilisant des hypothèses uniquement sur la largeur des couches que tout minimiseur local est un minimiseur global. Ils prouvent que cette hypothèse sur l'architecture est forte dans le sens où sans elle, et si nous ne faisons pas d'hypothèses sur les matrices de données comme dans les travaux précédents, alors il existe un minimiseur local non-global. [164] a utilisé des hypothèses uniquement sur la matrice des données d'entrée, pour prouver que pour un réseau linéaire à une couche cachée, tout minimiseur local est un minimiseur global et tout autre point critique a une courbure négative. [83] a prouvé pour différentes pertes

convexes générales que, sous des hypothèses sur l'architecture, tous les minima locaux sont globaux. Enfin, [137] et [98] ont utilisé des résultats de géométrie algébrique pour donner d'autres propriétés sur les points critiques des réseaux linéaires.

La plupart des travaux précédents se concentrent sur les minimiseurs locaux. Aucun de ces travaux ne fournit de conditions nécessaires et suffisantes simples pour qu'un point-selle soit strict ou non. [2] En particulier, dans le cas de plus de deux couches cachées, seuls des exemples très spécifiques de points-selles non stricts ont été décrits. En outre, les minimiseurs globaux ont été caractérisés mais pas explicitement paramétrés. Voir le Chapitre 3. Section 3.3.4 pour plus de détails.

Dynamique des gradients et régularisation implicite pour les réseaux linéaires : Dans cet axe de recherche, les auteurs étudient la dynamique des algorithmes du premier ordre pour les réseaux linéaires, qu'ils combinent parfois avec des résultats sur le paysage. [5] a prouvé que la descente de gradient converge vers un minimum global avec une vitesse linéaire, sous des hypothèses sur la largeur des couches, l'itéré initial, et la perte à l'initialisation. D'autres travaux ont également prouvé des résultats similaires avec des hypothèses différentes [36, 11, 144]. Cependant, comme le note [123], ces travaux considèrent des hypothèses fortes sur la perte à l'initialisation. En effet, [123] a donné un résultat négatif sur un réseau linéaire profond de largeur 1, en prouvant que pour des initialisations standard, la descente de gradient peut prendre un temps exponentiel pour converger vers le minimiseur global. L'auteur a également fourni des exemples empiriques du même phénomène se produisant pour des largeurs plus importantes. D'autre part, [33] a prouvé que si les couches sont suffisamment larges, la convergence vers un minimiseur global peut être obtenue en temps polynomial en utilisant une initialisation gaussienne aléatoire classique indépendante des données (connue sous le nom d'initialisation de Xavier). La largeur minimale requise du réseau dépend de la norme d'un minimiseur global du problème de régression linéaire. Comme nous le verrons dans la section 3.3.4, ce résultat de convergence globale peut être réinterprété en termes de paysage de pertes à l'ordre 2.

Dans une ligne de recherche similaire, [24] a prouvé, en utilisant des hypothèses sur l'architecture du réseau et les matrices de données, que le flot de gradient converge presque sûrement vers un minimiseur global pour un réseau linéaire à une couche cachée. Plus tard, [7] a prouvé le même résultat sous des hypothèses plus faibles sur les matrices de données. Ils ont également prouvé que, dans les réseaux linéaires profonds, le flot de gradient converge presque sûrement vers l'un des minimiseurs globaux du problème de régression linéaire avec contrainte sur le rang.

Ceci est lié à une autre conséquence des propriétés du paysage : la régularisation implicite. [6] a montré que, pour la récupération de matrices, les réseaux linéaires profonds convergent vers des solutions de faible rang même lorsque toutes les couches cachées ont une taille supérieure ou égale à celle des entrées et des sorties. [117] ont prouvé que, dans la factorisation matricielle profonde, la régularisation implicite peut ne pas être explicable par les normes, car toutes les normes peuvent aller à l'infini. Ils suggèrent plutôt de voir

2. Par "simple", nous entendons une condition plus facile à exploiter que la simple recherche de la plus petite valeur propre de la Hessienne.

la régularisation implicite comme une minimisation du rang. [119] et [44] ont prouvé, avec différentes hypothèses sur les données et une initialisation qui tend vers zero que la dynamique du gradient discret ou continu apprend séquentiellement les solutions d'un problème de régression linéaire avec contrainte sur le rang dont le rang augmente progressivement. Enfin, [45] a prouvé pour un modèle jouet que cet apprentissage incrémental se produit plus souvent (avec une plus grande initialisation), lorsque la profondeur du réseau augmente. Comme nous le verrons dans le Chapitre 3, Section 3.3.4, ces résultats peuvent être réinterprétés à la lumière du paysage à l'ordre 2.

1.4 Couches convolutives orthogonales

1.4.1 Motivation

La contrainte d'orthogonalité a d'abord été envisagée pour les réseaux de neurones entièrement connectés [4]. Pour les réseaux de neurones convolutionnels [84, 81, 161], l'introduction de la contrainte d'orthogonalité est un moyen d'améliorer le réseau de neurones à plusieurs égards. Tout d'abord, malgré des solutions bien établies [59, 66], l'entraînement de réseaux convolutifs très profonds reste difficile. Ceci est notamment dû à des problèmes d'explosion/atténuation de gradient [61, 16]. En conséquence, la capacité expressive des couches convolutionnelles n'est pas pleinement exploitée [66]. Cela peut conduire à des performances plus faibles sur les tâches d'apprentissage automatique. De plus, l'absence de contrainte sur la couche convolutive conduit souvent à des prédictions irrégulières qui sont sujettes à des attaques adverses [134, 106]. L'évitement de l'explosion/atténuation du gradient, la robustesse intégrée et de meilleures capacités de généralisation sont les principaux objectifs de l'introduction des contraintes de Lipschitz [**tsuzuku2018Lipschitz**, 134, 115, 48, 121] et d'orthogonalité aux couches convolutives [146, 27, 65, 158, 91, 54, 114, 142, 138, 70, 90, 64, 70, 9, 145]. Les réseaux convolutifs orthogonaux ont été appliqués avec succès dans diverses applications, telles que la classification, la segmentation, la retouche [142, 159, 82], ou récemment dans le "few-shot learning" [110]. L'orthogonalité est également proposée pour les réseaux adversariaux génératifs [102], ou même requise pour l'estimation de la distance de Wasserstein, comme dans le Wasserstein-GAN [3, 52], et le classificateur basé sur le transport optimal [122].

Les réseaux convolutifs orthogonaux sont constitués de plusieurs couches convolutives orthogonales. Cela signifie que, lorsque l'on exprime le calcul effectué par la couche sous la forme d'une matrice, celle-ci est orthogonale. Le terme "orthogonal" s'applique aussi bien aux matrices carrées qu'aux matrices non carrées [3]. Dans ce dernier cas, il inclut deux notions communément distinguées mais liées : l'orthogonalité de ligne et l'orthogonalité de colonne. Le Chapitre 4 se concentre sur les propriétés théoriques des couches convolutionnelles orthogonales. De plus, comme les couches de déconvolution (également appelées convolution transposée) sont définies à l'aide de couches de convolution, les résultats peuvent également être appliqués aux couches de déconvolution orthogonales. Nous considérerons l'architecture d'une couche convolutive caractérisée par (M, C, k, S),

3. Cette propriété est des fois appelée 'semi-orthogonal'.

où M est le nombre de canaux de sortie, C de canaux d'entrée, les noyaux de convolution sont de taille $k \times k$ et le paramètre de stride est S. Sauf indication contraire, nous considérons les convolutions avec des conditions limites circulaires[4]. Ainsi, appliquées sur canaux d'entrée de taille $SN \times SN$, les M canaux de sortie sont de taille $N \times N$. Nous désignons par $\mathbf{K} \in \mathbb{R}^{M \times C \times k \times k}$ le tenseur noyau et par $\mathcal{K} \in \mathbb{R}^{MN^2 \times CS^2N^2}$ la matrice qui applique la couche convolutive d'architecture (M, C, k, S) à C canaux vectorisés de taille $SN \times SN$.[5]

Nous allons d'abord répondre aux questions importantes :

— **Existence :** Quelle est la condition nécessaire et suffisante sur (M, C, k, S) et N pour qu'il existe une couche convolutive orthogonale (i.e. $\mathcal{K}$ orthogonale) pour cette architecture ? Comment les conditions aux limites "valid" et "same" limitent-elles l'existence de l'orthogonalité ?

En outre, nous nous appuierons sur des articles récemment publiés : [142, 114] qui caractérisent les couches convolutionnelles orthogonales comme l'ensemble de niveau zéro d'une fonction particulière appelée L_{orth} dans [142] (voir le Chapitre 4, Section 4.1.1.2, pour plus de détails). Formellement, $\mathcal{K}$ est orthogonal si et seulement si $L_{orth}(\mathbf{K}) = 0$. Ils utilisent L_{orth} comme terme de régularisation et obtiennent des performances impressionnantes sur plusieurs tâches d'apprentissage automatique (voir [142]). La régularisation est ensuite appliquée avec succès à la segmentation d'images médicales [159], à la retouche d'images [82] et au "few-shot learning" [110].

Dans le Chapitre 4, nous étudions les questions théoriques suivantes :

— **Stabilité par rapport aux erreurs de minimisation :** Est-ce que $\mathcal{K}$ a toujours de bonnes "propriétés d'orthogonalité approximative" lorsque $L_{orth}(\mathbf{K})$ est petit mais non nul ? Sans cette garantie, il pourrait arriver que $L_{orth}(\mathbf{K}) = 10^{-9}$ et $\|\mathcal{K}\mathcal{K}^T - Id\|_2 = 10^9$. Cela rendrait la régularisation avec L_{orth} inutile, à moins que l'algorithme n'atteigne $L_{orth}(\mathbf{K}) = 0$.

— **Scalabilité et stabilité par rapport à N :** Remarquons que, pour un tenseur noyau $\mathbf{K}$ donné, $L_{orth}(\mathbf{K})$ est indépendant de N mais la matrice de transformation de la couche $\mathcal{K}$ dépend de N : Lorsque $L_{orth}(\mathbf{K})$ est petit, est-ce que $\mathcal{K}$ reste approximativement orthogonal et isométrique lorsque N croît ? Si oui, la régularisation avec L_{orth} reste efficace même pour de très grands N.

— **Optimisation :** Le paysage de L_{orth} se prête-t-il à une optimisation globale ?

1.4.2 Revue de littérature

Les matrices orthogonales forment la variété de Stiefel et ont été étudiées dans [34]. En particulier, la variété de Stiefel est compacte, régulière et de dimension connue. Elle est constituée de plusieurs composantes connexes. Ceci peut être un problème numérique puisque la plupart des algorithmes ont des difficultés à changer de composantes connexes pendant l'optimisation. La variété de Stiefel possède de nombreuses autres propriétés intéressantes qui la rendent adaptée à l'optimisation Riemannienne (locale) [89, 90]. Les

4. Avant de calculer une convolution, les canaux d'entrée sont rendus périodiques en dehors de leur support réel.

5. voir Chapitre 4, Annexe 4.B pour la formule de $\mathcal{K}$.

couches convolutionnelles orthogonales sont une sous-partie de cette variété de Stiefel. À notre connaissance, la compréhension des couches convolutionnelles orthogonales est faible. Il n'y a pas d'article qui se concentre sur les propriétés théoriques des couches convolutionnelles orthogonales.

De nombreux articles [**scaman2018Lipschitz**, 148, 27, 129, 70, 48, 37] se concentrent sur les contraintes de Lipschitz et d'orthogonalité des couches du réseau de neurones d'un point de vue statistique, en particulier dans le contexte des attaques adversariales.

De nombreux articles récents ont étudié le problème numérique de l'optimisation d'un tenseur noyau $\mathbf{K}$ sous la contrainte que $\mathcal{K}$ est orthogonal ou approximativement orthogonal. Ils fournissent également des arguments de modélisation et des expériences en faveur de cette contrainte. On peut distinguer deux stratégies principales : **orthogonalité du noyau** [146, 27, 65, 158, 54, 70, 90, 64, 70, 9, 122] et **orthogonalité des couches convolutionnelles**. [91, 114, 142, 138]. Cette dernière a été introduite plus récemment.

Nous désignons l'entrée de la couche par $X \in \mathbb{R}^{C\times SN\times SN}$ et sa sortie par $Y = \mathbf{conv}(\mathbf{K}, X) \in \mathbb{R}^{M\times N\times N}$.

— **Orthogonalité du noyau :** Cette classe de méthodes considère la convolution comme une multiplication entre une matrice $\overline{\mathbf{K}} \in \mathbb{R}^{M\times Ck^2}$ formée en remodelant le tenseur du noyau $\mathbf{K}$ (voir, par exemple, [27, 142] pour plus de détails), et la matrice $U(X) \in \mathbb{R}^{Ck^2\times N^2}$ dont les colonnes contiennent la concaténation des C patchs vectorisés de X nécessaires au calcul des M canaux de sortie à une position spatiale donnée (voir [60, 149]). Nous avons donc, $\text{Vect}\,(Y) = \text{Vect}\,\big(\overline{\mathbf{K}}U(X)\big)$. La stratégie d'orthogonalité du noyau renforce l'orthogonalité de la matrice $\overline{\mathbf{K}}$.

— **Orthogonalité des couches convolutionnelles :** Cette classe de méthodes relie directement l'entrée et la sortie de la couche en écrivant $\text{Vect}\,(Y) = \mathcal{K}\,\text{Vect}\,(X)$ et renforce l'orthogonalité de $\mathcal{K}$. La difficulté de cette méthode est que la taille de la matrice $\mathcal{K} \in \mathbb{R}^{MN^2\times CS^2N^2}$ dépend de N et peut être très grande.

L'orthogonalité du noyau fournit une stratégie numérique dont la complexité est indépendante de N. Cependant, l'orthogonalité du noyau n'implique pas que $\mathcal{K}$ soit orthogonal. En bref, le problème est que la composition d'un plongement orthogonal et d'une réduction de dimension orthogonale n'a aucune raison d'être orthogonale. Ce phénomène a été observé empiriquement dans [91] et [70]. Les auteurs de [142] et [114] affirment également que, lorsque $\mathcal{K}$ a plus de colonnes que de lignes (orthogonalité des lignes), l'orthogonalité de $\overline{\mathbf{K}}$ est nécessaire mais pas suffisante pour garantir l'orthogonalité de $\mathcal{K}$. L'orthogonalité du noyau et l'orthogonalité de la couche convolutive sont différentes, cette dernière permettant de mieux éviter l'atténuation du gradient et la corrélation des features.

On peut distinguer deux manières numériques d'imposer l'orthogonalité pendant l'entraînement :

— **Orthogonalité rigide :** Cette méthode consiste à maintenir la matrice d'intérêt orthogonale pendant tout le processus d'apprentissage. Cela peut être fait soit en optimisant sur la variété de Stiefel, soit en considérant une paramétrisation d'un sous-ensemble de matrices orthogonales (par exemple, [90, 91, 138, 128, 65, 158]). Notez que certaines méthodes d'orthogonalité des couches convolutionnelles rigides considèrent des

mappings de $\mathcal{K}$, ce qui entraîne des convolutions avec des noyaux de taille supérieure à $k \times k$.

— **Orthogonalité souple :** Une autre méthode pour imposer l'orthogonalité des matrices pendant l'optimisation consiste à ajouter une régularisation du type $\|WW^T - I\|^2$ à la perte de la tâche spécifique. Cette régularisation pénalise les matrices loin d'être orthogonales (par exemple, [9, 27, 114, 142, 146, 54, 70, 64]).

Notez que, contrairement à l'orthogonalité de noyau, l'orthogonalité de couche convolutive traite directement de $\mathcal{K}$, et a donc une complexité qui dépend généralement de N. Cependant, dans le contexte de l'orthogonalité souple des couches convolutionnelles, les auteurs de [114, 142] introduisent le régularisateur L_{orth} qui est indépendant de N (voir Chapitre 4, Section 4.1.1.2, pour plus de détails), comme substitut à $\|\mathcal{K}\mathcal{K}^T - \mathrm{Id}_{MN^2}\|_F^2$ et $\|\mathcal{K}^T\mathcal{K} - \mathrm{Id}_{CS^2N^2}\|_F^2$. Dans [142], les couches convolutionnelles orthogonales avec stride sont considérées pour la première fois.

1.5 Approximation par des réseaux de neurones

Rappelons que l'erreur d'approximation qui concerne l'expressivité de la classe des réseaux de neurones est donnée par la quantité $R(f_{\mathbf{w}^*}) - R^*$. Supposons qu'il existe g tel que $R(g) = min_h R(h)$. L'erreur d'approximation peut s'écrire

$$R(g) - R(f_{\mathbf{w}^*}) = \int \left[\ell(g(x), y) - \ell(f_{\mathbf{w}^*}(x), y)\right] dP(x, y).$$

Si ℓ est 1-Lipschitz par rapport à sa seconde variable alors on peut écrire :

$$\int \left[\ell(g(x), y) - \ell(f_{\mathbf{w}^*}(x), y)\right] dP(x, y) \leq \int |g(x) - f_{\mathbf{w}^*}(x)| dP(x, y).$$

Par conséquent, si nous pouvons contrôler la norme sup ou la norme $L_1(P)$ de la différence entre f et g, nous pouvons contrôler l'erreur d'approximation de la décomposition du risque. Il est donc intéressant d'avoir des bornes précises en norme L^p.

1.5.1 Approximation universelle et bienfaits de la profondeur

Les réseaux de neurones sont connus pour être des approximateurs universels. En effet, pour divers espaces de fonctions, il a été prouvé que l'on peut approximer une fonction cible d'intérêt en utilisant un réseau de neurones à une profondeur arbitraire. Mathématiquement, nous avons le théorème classique suivant qui peut être trouvé avec diverses conditions sur les activations et les espaces de fonctions en [28, 62, 88, 76].

Theorem 1. Pour $\varepsilon > 0$, pour toute fonction continue f et une fonction d'activation σ qui n'est pas polynomiale, il existe un réseau de neurones à une couche cachée g tel que

$$\sup_{x \in [0,1]^d} |f(x) - g(x)| \leq \varepsilon$$

Ce théorème est très intéressant et nous indique que les réseaux de neurones sont assez expressifs et que l'on peut les utiliser comme approximateurs. Cependant, il n'explique pas pourquoi la profondeur est utilisée dans la pratique et ne nous donne pas non plus un taux d'approximation (par exemple, le nombre de poids nécessaires pour approximer une classe de fonction avec une erreur de ε).

Ainsi, deux directions de recherche ont suivi ce théorème, la première concerne les résultats de séparation de profondeur, qui stipulent typiquement qu'il existe certaines fonctions qui peuvent être approximées en un nombre polynomial de poids en ε avec un réseau profond alors qu'un réseau peu profond a besoin d'un nombre exponentiel de poids pour l'approximer avec la même erreur. Un exemple de tels résultats sépare le réseau de profondeur 2 du réseau de profondeur 3 et peut être trouvé dans [135] :

Theorem 2. Pour toute profondeur L, il existe un réseau ReLU f avec $\mathcal{O}(L^2)$ couches et noeuds tel que pour tout réseau ReLU g avec au plus L couches et au plus 2^L noeuds nous avons

$$\int_0^1 |f(x) - g(x)| \geq \frac{1}{32}.$$

L'autre direction, qui est celle sur laquelle nous allons nous concentrer, est celle des taux quantitatifs d'approximation des espaces de fonctions par les réseaux de neurones.

1.5.2 Quantification du taux d'approximation

Une façon de quantifier le pouvoir d'expression des réseaux de neurones est de résoudre le problème suivant. Soit G l'ensemble de toutes les fonctions $g_{\mathbf{w}} : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ qui peuvent être représentées en réglant les poids $\mathbf{w} \in \mathbb{R}^W$ d'un réseau de neurones feed-forward avec une architecture fixe, et soit F un ensemble quelconque de fonctions à valeurs réelles sur $\mathcal{X}$. Une question naturelle se pose : dans quelle mesure les fonctions $f \in F$ peuvent-elles être approximées par des fonctions $g_{\mathbf{w}} \in G$? Plus précisément, étant donné une norme $\|\cdot\|$ sur les fonctions, quel est l'ordre de grandeur de l'*erreur d'approximation de F par G* (dans le pire des cas) définie par :

$$\sup_{f \in F} \inf_{g_{\mathbf{w}} \in G} \|f - g_{\mathbf{w}}\| , \tag{1.5.1}$$

et quelle est son ordre de grandeur en fonction des nombres W, L de poids et de couches, et de certaines propriétés de F ?

Les bornes inférieures de l'erreur d'approximation (1.5.1) peuvent être utiles de plusieurs façons. Elles fournissent une limite à la meilleure précision d'approximation que l'on peut espérer atteindre si le nombre de poids ou de couches du réseau est contraint, et aident à concevoir des architectures optimales sous ces contraintes. Elles impliquent également une borne inférieure sur le nombre minimal de poids ou de couches à inclure dans un réseau afin d'approximer toute fonction dans F avec une précision donnée ε.

Le cas où $\|\cdot\|$ est la norme sup (définie par $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)|$) est assez bien compris, du moins dans certains cas particuliers. Par exemple, lorsque F est une boule de Hölder de régularité $s > 0$ et que le réseau utilise la fonction d'activation ReLU, Yarotsky

[152] a dérivé une borne inférieure sur (1.5.1) de l'ordre de $W^{-2s/d}$, raffinée ensuite à $(LW)^{-s/d}$ (à des facteurs logarithmiques près) par [153, 154] lorsque la profondeur du réseau varie de $L = 1$ à $L \approx W$. En utilisant la technique d'extraction de bits, ces auteurs ont montré que ces bornes inférieures sont réalisables (à des facteurs logarithmiques près) avec une architecture de réseau ReLU soigneusement conçue. Des résultats raffinés en termes de largeur et de profondeur ont été obtenus par [125] lorsque $s \leq 1$, tandis que d'autres fonctions d'activation ont également été étudiées dans [154].

Les bornes supérieures sont généralement prouvées en décomposant d'abord les fonctions cibles dans une certaine base (par exemple, développement de Taylor), puis en approchant chaque vecteur de base avec un réseau de neurones (par exemple, en approchant $(x, y) \to xy$). Pour les bornes inférieures, la technique la plus utilisée est basée sur la borne inférieure de la VC-dimension, puis sur l'utilisation de bornes supérieures connues sur la VC-dimension des réseaux de neurones par rapport à W et L pour conclure.

Nous donnons dans la section suivante une preuve typique d'une borne inférieure sur l'erreur d'approximation lorsque la norme est la norme sup. Notez que ce type de preuve peut être trouvé aussi par exemple dans [152, 125, 31].

1.5.3 Bornes inférieures d'approximation en norme sup

Dans cette section, nous prouvons une borne inférieure sur l'erreur d'approximation définie dans (1.5.1) lorsque la norme est la norme sup, et que G est un espace de réseaux de neurones avec une architecture fixe et une activation polynomiale par morceaux. Nous utilisons la preuve typique et expliquons ensuite pourquoi elle ne peut pas être appliquée directement pour prouver des bornes en norme L^p. Pour cet exemple, nous avons choisi de le faire lorsque F est l'espace des fonctions monotones. Nous désignons par $\mathcal{M}_d$ l'espace des fonctions croissantes en chacune des variables. Nous désignons par $H_{\mathcal{A}}$ l'espace des fonctions générées par des réseaux de neurones d'architecture fixe $\mathcal{A}$ et où les poids varient. Dans le chapitre 5, nous reverrons cet espace et prouverons des bornes en norme L^p.

Proposition 1.5.1. Soit $d \in \mathbb{N}^*$ et soit $\mathcal{A}$ une architecture de réseau de neurones avec activation polynomiale par morceaux et W paramètres. Il existe une constante $c_d > 0$ dépendant uniquement de d et une fonction $f \in \mathcal{M}_d$ telle que pour tout $g \in H_{\mathcal{A}}$, $\|f - g\|_\infty \geq c_d W^{-2/d}$. En d'autres termes

$$\sup_{f \in \mathcal{M}_d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_\infty \geq c_d W^{-2/d}. \tag{1.5.2}$$

Démonstration. Soit $\mathcal{A}$ une architecture de réseau de neurones à activation polynomiale par morceaux à $W \in \mathbb{N}^*$ paramètres. Si $\sup_{f \in \mathcal{M}_d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_\infty \geq \frac{1}{6d}$, alors le résultat est immédiat en considérant $c_d = \frac{1}{6d}$. Soit $\varepsilon > \sup_{f \in \mathcal{M}_d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_\infty$ tel que $\varepsilon < \varepsilon_0 = \frac{1}{6d}$, de telle façon à ce que pour tout $f \in \mathcal{M}_d$, il existe $g \in H_{\mathcal{A}}$ tel que $\|f - g\|_\infty \leq \varepsilon$. Tout d'abord, nous cherchons à borner inférieurement W par rapport à ε et nous le faisons en bornant inférieurement la pseudo-dimension de $H_{\mathcal{A}}$.

Soit $N := \left\lceil \frac{1}{3d\varepsilon} \right\rceil$. Puisque $\varepsilon < \varepsilon_0 = \frac{1}{6d}$, on a $\frac{1}{3d\varepsilon} \leq N \leq \frac{1}{2d\varepsilon}$, ainsi $\varepsilon \leq \frac{1}{2dN}$.

On divise $[0,1]^d$ en un grille de N^d cubes $\mathcal{C}_{\mathbf{k}} = \prod_{i=1}^d \left(\frac{k_i}{N}, \frac{k_i+1}{N}\right]$, $\mathbf{k} = (k_1, \ldots, k_d) \in \{0, \ldots, N-1\}^d$. Pour $\sigma := (\sigma_{\mathbf{k}})_{\mathbf{k}\in\{0,\ldots,N-1\}^d} \in \{-1,1\}^{N^d}$, on définit $f_\sigma : [0,1]^d \to [0,1]$ comme suit : pour tout $\mathbf{k} \in \{0, \ldots, N-1\}^d$, pour tout $x \in \mathcal{C}_{\mathbf{k}}$

$$f_\sigma(x) = \frac{\sum_{i=1}^d k_i}{Nd} + \frac{1}{2Nd} + \frac{\sigma_{\mathbf{k}}}{2Nd}.$$

Il est facile de vérifier que pour tout σ, f_σ peut être naturellement étendu à une fonction sur $[0,1]^d$ à valeurs dans $[0,1]$; nous considérons ces extensions sans changer de notations. On voit donc que $\mathcal{F} := \{f_\sigma, \sigma \in \{-1,1\}^{N^d}\}$ est un sous-ensemble de $\mathcal{M}_d$ (tout f_σ est croissante de $[0,1]^d$ dans $[0,1]$). Il s'en suit que pour tout $f \in \mathcal{F}$, il existe $g \in H_{\mathcal{A}}$ tel que $\|f - g\|_\infty \leq \varepsilon$.

Pour montrer que $Pdim(H) \geq N^d$, on construit un ensemble de N^d points dont on prouve qu'ils sont pseudo-éclatés par $H_{\mathcal{A}}$, en utilisant des fonctions dans $H_{\mathcal{A}}$ qui approchent $\mathcal{F}$. Pour tout $\mathbf{k} \in \{0, \ldots, N-1\}^d$, soit

- $x^{\mathbf{k}} := \left(k_1 + \frac{1}{2N}, \ldots, k_d + \frac{1}{2N}\right)$,
- $r_{\mathbf{k}} := \frac{\sum_{i=1}^d k_i}{Nd} + \frac{1}{2Nd}$,

et soit $S = (x^{\mathbf{k}})_{\mathbf{k}\in\{0,\ldots,N-1\}^d}$. Les $x^{\mathbf{k}}$ sont les points au centre de chaque cube de notre grille, il y en a donc N^d. Ce sont ces points que nous allons pseudo-éclater avec la séquence de seuils $(r_{\mathbf{k}})_{\mathbf{k}\in\{0,\ldots,N-1\}^d}$. Soit $z := (z_{\mathbf{k}})_{\mathbf{k}\in\{0,\ldots,N-1\}^d} \in \{-1,1\}^{N^d}$ une séquence de labels associés à S. On choisit $f_\sigma \in \mathcal{F}$ avec $\sigma_{\mathbf{k}} = z_{\mathbf{k}}$ pour tout $\mathbf{k}$. Par définition de ε et puisque f_σ est dans $\mathcal{M}_d$, il existe $g \in H_{\mathcal{A}}$ tel que g approche f_σ avec une erreur d'au plus ε.

En utilisant $\varepsilon \leq \frac{1}{2dN}$, on a,

$$\|f_\sigma - g_\sigma\|_\infty \leq \varepsilon \implies |f_\sigma(x^{\mathbf{k}}) - g_\sigma(x^{\mathbf{k}})| \leq \varepsilon \leq \frac{1}{2Nd} = |\frac{\sigma_{\mathbf{k}}}{2Nd}| = |f_\sigma(x^{\mathbf{k}}) - r^{\mathbf{k}}| \tag{1.5.3}$$

pour tout $\mathbf{k} \in \{0, \ldots, N-1\}^d$. Ainsi, si $\sigma_{\mathbf{k}} > 0$, alors $f_\sigma(x^{\mathbf{k}}) - g_\sigma(x^{\mathbf{k}}) \leq |f_\sigma(x^{\mathbf{k}}) - g_\sigma(x^{\mathbf{k}})| \leq f_\sigma(x^{\mathbf{k}}) - r^{\mathbf{k}}$, donc $g_\sigma(x^{\mathbf{k}}) > r^{\mathbf{k}}$. Si $\sigma_{\mathbf{k}} < 0$, alors $g_\sigma(x^{\mathbf{k}}) - f_\sigma(x^{\mathbf{k}}) \leq |f_\sigma(x^{\mathbf{k}}) - g_\sigma(x^{\mathbf{k}})| \leq r^{\mathbf{k}} - f_\sigma(x^{\mathbf{k}})$, donc $g(x^{\mathbf{k}}) < r^{\mathbf{k}}$.

Il s'en suit que l'indicatrice $\mathbb{1}_{\{g_\sigma(x^{\mathbf{k}}) > r_{\mathbf{k}}\}}$ correspond aux labels $z_{\mathbf{k}}$'s ; c'est-à-dire que $H_{\mathcal{A}}$ pseudo-éclate S et $Pdim(H_{\mathcal{A}}) \geq N^d$, ainsi puisque $N = \lceil \frac{1}{3d\varepsilon} \rceil \geq \frac{1}{3d\varepsilon}$,

$$Pdim(H_{\mathcal{A}}) \geq (3d\varepsilon)^{-d}. \tag{1.5.4}$$

Il reste maintenant à établir la borne supérieure de $Pdim(H_{\mathcal{A}})$ en fonction de W, qui, combinée à (1.5.4), fournira la borne inférieure de W en termes de ε. Le théorème 14.1 de [2] fournit une architecture $\mathcal{A}'$ autorisant les skip-connexion et avec un nombre de paramètres $W' \leq W + 2$, telle que

$$Pdim(H_{\mathcal{A}}) \leq VCdim(H_{\mathcal{A}'}).$$

On utilise maintenant le Théorème 8.7 dans [2] qui implique que $VCdim(H_{\mathcal{A}'}) \leq c_1 W'^2$ pour une constante universelle $c_1 > 0$. Puisque $W' \leq W + 2$,

$$Pdim(H_{\mathcal{A}}) \leq c_2 W^2, \tag{1.5.5}$$

pour une cosntante universelle c_2. En combinant les inégalités (1.5.4) et (1.5.5), on obtient $W \geq c_3 \varepsilon^{-d/2}$, où c_3 dépend seulement de d.

Pour finir la preuve, considèrons une séquence $(\varepsilon_n)_{n\geq 1}$ tel que pour tout n

$$\varepsilon_n > \sup_{f \in \mathcal{M}_d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_\infty,$$

et $\lim_{n\to\infty} \varepsilon_n = \sup_{f\in\mathcal{M}_d} \inf_{g\in H_{\mathcal{A}}} \|f - g\|_\infty$. Pour tout n, $H_{\mathcal{A}}$ approche $\mathcal{M}_d$ avec une erreur d'au plus ε_n, par définition. Les inégalités (1.5.4) et (1.5.5) peuvent donc s'appliquer, et on obtient

$$cW^{-2/d} \leq \varepsilon_n, \tag{1.5.6}$$

pour une constante c qui dépend seulement de d. Le résultat en découle en considérant la limite en (1.5.6) quand $n \to +\infty$. □

1.5.4 Approximation en norme L^p

Il est également intéressant d'étudier (1.5.1) avec la norme $L^p(\mu)$, définie par

$$\|f\|_{L^p(\mu)} = \Big(\int_X |f(x)|^p d\mu(x)\Big)^{1/p},$$

pour $p \geq 1$ et une certaine mesure de probabilité μ sur $\mathcal{X}$. Puisque cela correspond à une approximation des fonctions dans F dans un sens plus "moyen" que pour la norme sup, une question naturelle est de savoir si la même précision peut être obtenue avec un réseau plus petit ou non. Malheureusement, les stratégies de preuve derrière les bornes inférieures de [152, 153, 154, 125] sont spécifiques à la norme sup. En effet, la rupture se produit à l'équation (1.5.3), puisque si l'on utilise la norme L^p, on ne peut pas garantir que la différence entre g et f sont inférieurs à ε pour les centres x^k, ou même pour les autres points uniformément, c'est-à-dire, pour chaque f_σ il existe x_σ tel que $|f_\sigma(x_\sigma) - g_\sigma(x_\sigma)| \leq \varepsilon$. Par conséquent, on ne peut pas exhiber directement un ensemble de points qui sont éclatés par g_σ.

DeVore et al. [31] ont en effet commenté : “Quand nous passons au cas $p < \infty$, la situation est encore moins claire [...] nous ne pouvons pas utiliser la théorie de la VC-dimension pour l'approximation de $L^p(\Omega)$. [...] Ce qui manque vis-à-vis du problème 8.13 est de savoir quelles sont les meilleures bornes et comment nous prouvons les bornes inférieures pour les taux d'approximation dans $L^p(\Omega)$, $p \neq \infty$.”

Les bornes inférieures existantes en norme $L^p(\mu)$. Plusieurs articles ont fourni des bornes inférieures dans certains cas particuliers, sous certaines conditions concernant l'ensemble à approximer F, le réseau de neurones, la métrique d'approximation ou la carte d'encodage $f \in F \mapsto \mathbf{w}(f) \in \mathbb{R}^W$.

Lorsque F est un espace de régularité s, un premier résultat basé sur [32] stipule que lorsque l'on impose aux poids de dépendre continuellement de la fonction à approximer, on ne peut pas obtenir un meilleur taux d'approximation que $W^{-\frac{s}{d}}$.

Pour le même F, un autre résultat pour $p = 2$ et pour des fonctions d'activation qui sont continues ([95, 96]) prouve une borne inférieure sur l'approximation de fonctions de régularité s sur un compact de $\mathbb{R}^d$, par des réseaux de neurones à une couche cachée, d'ordre $W^{-\frac{s}{d-1}}$. Une borne supérieure correspondante est prouvée pour une fonction d'activation particulière, qui est sigmoïdale mais pathologique ([97]). Pour cette même fonction d'activation, ils prouvent que contrairement au cas d'une couche cachée, il n'y a pas de borne inférieure dans le cas de réseaux à deux couches cachées. Le résultat est basé sur le théorème de superposition de Kolmogorov-Arnold.

Dans [126], les auteurs étudient l'approximation par des réseaux de neurones peu profonds avec des poids bornés et des activations de la forme ReLU^k pour un entier k. Ils approximent la fermeture de l'enveloppe convexe des réseaux de neurones ReLU^k peu profonds avec des poids contraints. Ils obtiennent des bornes inférieures optimales d'ordre $W^{-\frac{1}{2}-\frac{2k+1}{2d}}$ pour toute norme $\|\cdot\|_X$ où X est un espace de Banach auquel appartiennent les fonctions d'approximation et tel que ces fonctions sont uniformément bornées par rapport à $\|\cdot\|_X$.

Les bornes inférieures d'approximation en norme $L^p(\mu)$, $p \geq 1$, ont également été étudiées dans le cadre des réseaux de neurones quantifiés (réseaux dont les poids sont codés avec un nombre fixe de bits). Dans [111], sous de faibles hypothèses sur la fonction d'activation, les auteurs prouvent une borne inférieure sur le nombre minimal de poids non nuls W qui sont requis pour qu'un réseau puisse approximer une classe de classificateurs binaires avec une erreur L^p au plus égale à ε. Ils montrent que W est au moins de l'ordre de $\varepsilon^{-\frac{p(d-1)}{\beta}} \log_2^{-1}(1/\varepsilon)$, où β est un paramètre de lissage. Des travaux ultérieurs, dont [141, 51], dérivent des bornes inférieures pour l'approximation par des réseaux quantifiés pour diverses normes.

Dans le chapitre 4, nous donnons une borne inférieure d'approximation générale pour l'erreur de norme $L^p(\mu)$ basée sur l'entropie métrique sur l'espace F et la pseudo-dimension de l'espace G. Nous l'appliquons au cas des réseaux de neurones pour résoudre la question ouverte de la borne inférieure des erreurs d'approximation pour les réseaux de neurones non quantifiés avec la norme L^p.

CHAPTER 2

Introduction

2.1 Machine learning

"AI is the new electricity", these are the words of Andrew Ng, a famous professor at Stanford University whose courses on Coursera on Machine Learning and Deep Learning have broken all records of enrollment on the platform. This reveals the importance of artificial intelligence, in particular Machine Learning and Deep Learning, in our era. Indeed, these algorithms are everywhere nowadays, if you open your browser and search for anything on a search engine, there is an algorithm behind it that uses artificial neural networks or what is more commonly called deep learning.

Deep learning in particular has seen an explosion since 2012 and the famous paper from Hinton's group that beat the state of the art in image recognition by far thanks to the neural network they trained. Following this, several researchers have returned to these neural networks which were already studied theoretically in the 80s but whose performance was not competitive enough with other methods due to the lack of data and computing power. In 10 years, deep learning has experienced phenomenal progress and it is inconceivable nowadays to imagine a world without these artificial neural networks. Indeed, they are used in image recognition, translation, voice recognition, recommendations, and others. Every year we find new applications where they succeed in beating the state of the art and allow considerable advances.

So what is a neural network? A neural network consists of an input layer, an output layer and several intermediate layers called hidden layers. To go from one layer to the other, a linear transformation is applied followed by a non-linear activation function. It is called deep learning because the depth is equal to the number of these layers and in practice for large networks they are of the order of tens.

These neural networks have thus given impressive results in all domains, but there is no theory explaining why they work so well. And contrary to the old machine learning algorithms that worked well and that were used in the 2000s, such as Support Vector Machines, which are well understood theoretically, artificial neural networks are difficult to study and according to the classical theory, there is no reason for learning to work so well. Thus, many researchers have worked and continue to work on this question, in order to try to bring partial answers and to try to reduce the gap between the extraordinary results that we observe in practice and the mathematical aspects that can explain them theoretically.

2.1.1 Definition of a neural network

A feedforward neural network is a function composed of several layers. To each layer, a linear function is applied followed by an activation function. More precisely, it can be seen

as follows:

— Consider H layers ($H-1$ hidden layers).

— We denote by: $n_0, n_1, \dots, n_H \in \mathbb{N}^*$ the width of the different layers.

— We denote by $f_h(x)$ the result obtained when computing the output of layer h for $x \in \mathbb{R}^{n_0}$.

— We denote by $W_h \in \mathbb{R}^{n_h \times n_{h-1}}$ the matrix of the weights on the edges between layer $h-1$ and layer h.

— We denote by $b_h \in \mathbb{R}^{n_h}$ the additive bias for layer h.

— We denote by σ the activation function. It is generally applied component-wise to each layer, and it can depend on h.

The prediction is computed according to the following scheme.

$$\begin{cases} f_0(x) = x \\ f_h(x) = \sigma(W_h f_{h-1}(x) + b_h) \in \mathbb{R}^{n_h} \quad \forall h = 1, \dots, H \end{cases} \tag{2.1.1}$$

2.1.2 Why is it important to study deep learning theory ?

For now, the theory of deep learning is still lagging on behind the impressive results observed in practice in the different domains where neural networks have been applied successfully. The results on the mathematical aspects of neural networks are still descriptive, in other words, we are just trying to understand or explain phenomena that we see in practice and we often use simplifying assumptions to make them tractable. This is a good beginning but the ultimate goal would be to come up with a prescriptive theory, in other words, some theoretical results that we can use to actually make algorithms better or choose the right architecture for the right type of data instead of spending hours trying different architectures and combination of hyperparameters and see what we get. Hopefully one day we can reach this kind of result.

2.2 Supervised learning framework

In this section, we recall the basics of the supervised learning setting which is the most used in machine learning and in deep learning in particular.

In supervised learning, we give our model a sample containing a set of examples (e.g. images) with their labels (e.g. cats or dogs). The goal is that after the training phase (which corresponds to weight learning for a neural network), our algorithm can correctly labels examples that it has never seen before.

2.2.1 Risk decomposition

— Let $(f_{\mathbf{w}})_{\mathbf{w}}$ be a family of functions parameterized by $\mathbf{w}$ (e.g. neural networks, with a fixed architecture and activation function)

For an i.i.d training sample $(x_i, y_i)_{i=1..N}$ drawn from a distribution $\mathcal{P}$ on $\mathcal{X} \times \mathcal{Y}$, and a loss $\ell : \mathcal{Y} \times \mathcal{Y} \longmapsto \mathbb{R}$, we set

— The risk :

$$R(f_{\mathbf{w}}) = \mathbb{E}\left(\ell(f_{\mathbf{w}}(X), Y)\right)$$

— The empirical risk :

$$\widehat{R}(f_{\mathbf{w}}) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_{\mathbf{w}}(x_i), y_i)$$

— $R^* = \inf_g R(g)$ the optimal risk, also called the Bayes risk

— We fix $\mathbf{w}^*$ and $\widehat{\mathbf{w}}$ such that :(if the minimizer does not exist we just consider ε-minmizers)

$$\mathbf{w}^* \in \arg\min_{\mathbf{w}} R(f_{\mathbf{w}}) \qquad \text{and} \qquad \widehat{\mathbf{w}} \in \arg\min_{\mathbf{w}} \widehat{R}(f_{\mathbf{w}})$$

In supervised learning, usually, the goal is to try to find $\mathbf{w}$ that minimizes the excess risk, which is the difference between the risk for $f_{\mathbf{w}}$ and the optimal risk R^*, this excess risk can be decomposed into three main errors

$$\begin{array}{rrcll}
0 \leq & R(f_{\mathbf{w}}) & - & R^* & \text{(Excess Risk)} \\
= & R(f_{\mathbf{w}}) & - & \widehat{R}(f_{\mathbf{w}}) & \text{(Generalization Error)} \\
+ & \widehat{R}(f_{\mathbf{w}}) & - & \widehat{R}(f_{\widehat{\mathbf{w}}}) & \text{(Optimization Error)} \\
+ & \widehat{R}(f_{\widehat{\mathbf{w}}}) & - & \widehat{R}(f_{\mathbf{w}^*}) & \leq 0 \\
+ & \widehat{R}(f_{\mathbf{w}^*}) & - & R(f_{\mathbf{w}^*}) & \text{(Generalization Error)} \\
+ & R(f_{\mathbf{w}^*}) & - & R^* & \text{(Approximation Error)}
\end{array}$$

The different errors contribute to the excess risk, and one wants to make each one of them as small as possible. We will describe them in detail in the next few sections.

2.2.2 Optimization

As presented in the previous decomposition of the risk, the optimization error is related to the empirical risk. And one would want to make it as small as possible without increasing the other errors. Typically we run an optimization algorithm, for instance (stochastic) gradient descent or variants like Adam [79] and RMSprop [136], and we stop once the algorithm has converged or if we spend a few epochs without decreasing the empirical risk. Therefore a natural question that occurs is: to which points can these algorithms converge or close to which points can they get stuck for a while. If we focus on the gradient descent algorithm for example, those points are the ones where the gradient vanishes, they are called (first-order) critical or stationary points.

Hence, a natural question is what kind of critical points algorithms can converge to and what are their properties. Are there local minima, global minima, or saddle points? Is the optimization error small at these points? What about their generalization error? Is there some kind of implicit bias associated with them? It is therefore necessary to study the landscape of the optimization error to be able to have a better picture of the optimization process and to better understand the gradient dynamics.

Our first work lies within this category, a more precise description can be found in Section 2.3 and Chapter 3.

2.2.3 Generalization

The second error is the generalization error, which is related to how well the algorithm does when predicting the target of a new unseen data point. Classically, people have been proving bounds on the generalization error by considering the whole function space we are searching, ie, bounding the quantity $\sup_{\mathbf{w}\in\mathcal{W}} |R(f_{\mathbf{w}}) - \widehat{R}(f_{\mathbf{w}})|$ by relating it to a complexity notion of the space. For example, the VC-dimension or the Rademacher complexity are notions that intervene in those bounds. However, these bounds are loose for neural networks that are used in practice. Indeed, they have many more parameters than the number of examples in the sample they are trained on. Despite that, the learned neural network still generalizes well. Recently, a "strange" phenomenon has been observed in practice, called the "double descent". Indeed, classical statistical learning theory tells us that fitting perfectly the data is usually associated with overfitting and therefore a bad generalization error. However, it has been observed (e.g., [15]) that neural networks can fit perfectly the data and still generalize well to unseen data points.

A popular tool used in the deep learning theory literature to prove generalization and optimization results for neural networks is the "Neural Tangent Kernel" [67]. This is a kernel that describes the evolution of deep neural networks during their training by gradient descent. Especially for wide networks, this allows the use of kernel methods for the analysis of neural network properties.

Implicit regularization: A lot of recent work tries to explain the success of deep learning via the implicit bias (regularization) induced by algorithms like Stochastic Gradient Descent. Indeed it seems that for overparameterized neural networks that can perfectly fit the training data, there are a lot of solutions that interpolate the training data. Among all these solutions, a lot have very bad generalization errors, however, simple algorithms like SGD seem to choose the one with minimal complexity which results in a good generalization error. The notion of complexity is hard to define globally but in this line of work, it is mostly seen as some norm.

What is impressive is that, in the presence of noisy data, the neural network fits the data perfectly and is still able to have a test error close to the Bayes risk. Therefore all the analyses previously conducted to explain generalization for the other models, based on the VC dimension or the Rademacher complexity for example do not suffice to explain the success of deep learning. A lot of works tried to explain this phenomenon in different context via the notion of benign overfitting (e.g., [15, 58, 13, 22, 139, 165, 92, 40]).

2.2.4 Approximation

The last error occurring in the previous decomposition is the approximation error which is related to the expressivity of the space of functions in which we are searching for a good predictor. The richer the hypothesis space the lower the approximation error and many

works have established upper and/or lower bounds on the worst-case approximation error of a class of functions by a function in a certain hypothesis space. When the hypothesis space is chosen to be functions implemented by neural networks of a certain architecture, several results have been proved and a deeper look at the literature can be found in Section 2.5.

Our third work lies within this category, a more precise description can be found in Section 2.5 and Chapter 5.

2.2.5 Robustness

In addition to the three errors we have seen in the previous sections, there are other aspects that one may want to optimize due to some problems that one may encounter in practice when working with neural networks. In fact, one common problem that has been observed for many of the deep learning architectures used in practice is adversarial attacks. They are many types of adversarial attacks in machine learning, the most classical example is the following: the model predicts correctly a car for an image containing a car, however when adding a small noise (invisible to the human eye) the picture of the car still looks the same to us humans, but the model predicts it as an ostrich. Although the perturbation is hardly perceptible to the human eye, this can be sufficient to drastically change the decision of the model. This is because the function computed by the neural network can be very oscillatory. Think of the situation where a self-driving car needs to identify a stop sign and because of some little drawing on the side of the sign it predicts it as something else. This might be problematic and it may cost lives at the end, hence we want to avoid these situations. That is why it is very important to design robust algorithms.

Lipschitz networks and generalization with adversarial examples: One way to do this is to build Lipschitz hypothesis functions such that the little changes in the input do not affect the output. In this context in deep learning, Lipschitz neural networks and in particular 1-Lipschitz neural networks have been investigated. This is done by constraining each layer to be Lipschitz and as we know, for two Lipschitz continuous functions f and g, the following property holds: $Lip(f \circ g) \leq Lip(f)Lip(g)$, where $Lip(f)$ denotes the Lipschitz constant with respect to the Euclidean norm.

We recall some notions related to robustness and how does Lipschitzness help with that. This can be found for example in [27]. We have seen previously that we are in general interested in minimizing the true risk $R(\mathbf{w}) = \mathbb{E}\left([\ell(f_{\mathbf{w}}(X), Y)]\right)$. When it comes to designing robust models, one would like to minimize in $\mathbf{w}$ the robust risk defined as follows

$$R(\mathbf{w}, p, \epsilon) = \mathbb{E}\left(\left[\max_{\tilde{x}:\|\tilde{x}-x\|_p \leq \epsilon} \ell(f_{\mathbf{w}}(X), Y)\right]\right),$$

for a fixed p and ε. By definition, $R(\mathbf{w}) \leq R(\mathbf{w}, p, \epsilon)$ for all p and $\epsilon > 0$. If ℓ is 1-Lipschitz and $\Lambda_p = \sup_{x \neq y} \frac{|f_{\mathbf{w}}(x) - f_{\mathbf{w}}(y)|}{\|x-y\|_p}$ denotes the Lipschitz constant of the neural network we have

$$R(\mathbf{w}, p, \epsilon) \leq R(\mathbf{w}) + \mathbb{E}\left(\left[\max_{\tilde{x}:\|\tilde{x}-x\|_p \leq \epsilon} |\ell(f_{\mathbf{w}}(\tilde{x}), Y) - \ell(f_{\mathbf{w}}(x), Y)|\right]\right) \leq R(\mathbf{w}) + \Lambda_p \epsilon.$$

This explains why Lipschitz networks are more robust.

Note that Lipschitz network can also help with generalization as it is proven in [147]. If we denote by $C_p(\mathcal{X}, \gamma)$ the covering number of $\mathcal{X}$ using γ-balls for $\|.\|_p$. Using $M = \sup_{x,W,y} \ell(g(x, W), y)$, [147] implies that for every $\delta \in (0, 1)$, with probability $1 - \delta$ over the i.i.d sample $(x_i, y_i)_{i=1}^m$, we have:

$$R(W) \leq \frac{1}{m} \sum_{i=1}^{m} \ell(g(x_i, W), y_i) + \Lambda_p \gamma + M \sqrt{\frac{2Y C_p(\mathcal{X}, \frac{\gamma}{2}) \ln(2) - 2 \ln(\delta)}{m}}$$

Our second work lies within this category, a more precise description can be found in Section 2.4 and Chapter 4.

2.2.6 Other deep learning theory areas

Many other areas of the mathematical aspects of deep learning have been and are still being investigated.

— Certification : Neural networks are usually used to give just a number or a vector as an output. However, one may want to quantify the uncertainty for a given prediction and give for example confidence interval for the prediction. This is for example done using Bayesian neural networks.

— Explainability : Neural networks are often seen as a black-box model, as it gives us predictions which are hard to explain contrary to most classical algorithms like logistic regression for instance. Many authors have tried opening the black-box and looking at the features used by the neural network to give a certain prediction. This is an important problem since legislation requires companies to be able to explain their decisions. For example, a banker needs to explain to a client why his credit application has been rejected and not just say "our neural network model says no". Some authors have investigated this using different methods for example gradient-based methods like sensitivity analysis, backprop based methods surrogate models methods or even game-theoretic methods.

2.3 Loss landscape of neural networks

As we have seen before, the optimization error is one of the three errors in the risk decomposition that one should try to make small. In practice, gradient-based methods seem to do a good job despite having a highly non-convex loss function.

One of the research directions to try to explain this phenomena is to study the loss landscape of deep neural networks and characterize their critical points.

We start by recalling some key definitions related to the landscape of a function.

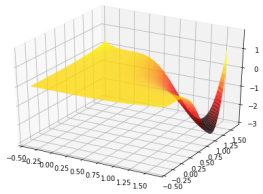

Figure 2.1 – Example of a landscape with a plateau (non-strict saddle point) and a global minimizer.

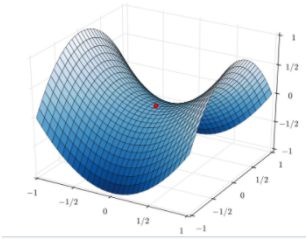

Figure 2.2 – Example of a landscape with a strict saddle point at (0,0).

2.3.1 Reminder : minimizers, critical points of order 1 or 2, strict and non-strict saddle points

Let us recall the definitions of local structures of the landscape of the empirical risk, which are important from the statistical and optimization points of view.

For $\boldsymbol{w} \in \mathbb{R}^n$, denote by $\boldsymbol{w} \longmapsto L(\boldsymbol{w})$ the function we want to minimize. Assume that $\boldsymbol{w} \longmapsto L(\boldsymbol{w})$ is C^2, and denote by ∇L and $\nabla^2 L$ its gradient and its Hessian. [1] We also write $A \succeq 0$ to say that a matrix $A \in \mathbb{R}^{n \times n}$ is positive semi-definite. Recall the following four definitions, which are nested:

- $\boldsymbol{w}^*$ is a **global minimizer** if and only if $\forall \boldsymbol{w} \in \mathbb{R}^n,\ L(\boldsymbol{w}^*) \leq L(\boldsymbol{w})$.
- $\boldsymbol{w}^*$ is a **local minimizer** if and only if there exists a neighbourhood $\mathcal{O} \subset \mathbb{R}^n$ of $\boldsymbol{w}^*$ such that $\forall \boldsymbol{w} \in \mathcal{O},\ L(\boldsymbol{w}^*) \leq L(\boldsymbol{w})$.
- $\boldsymbol{w}^*$ is a **second-order critical point** if and only if $\nabla L(\boldsymbol{w}^*) = 0$ and $\nabla^2 L(\boldsymbol{w}^*) \succeq 0$. If, on the contrary, the Hessian has a negative eigenvalue, we say that the point has a negative curvature.
- $\boldsymbol{w}^*$ is a **first-order critical point** if and only if $\nabla L(\boldsymbol{w}^*) = 0$.

  We can also distinguish a specific type of first-order critical point: saddle points. As discussed below, they can be second-order critical points or not.
- $\boldsymbol{w}^*$ is a **saddle point** if and only if it is a first-order critical point which is neither a local minimizer, nor a local maximizer.
  - — A saddle point $\boldsymbol{w}^*$ is **strict** if and only if it is not a second-order critical point (i.e., the Hessian $\nabla^2 L(\boldsymbol{w}^*)$ has a negative eigenvalue). Figure 2.2 gives an example.
  - — A saddle point $\boldsymbol{w}^*$ is **non-strict** if and only if it is a second-order critical point. In that case, the Hessian $\nabla^2 L(\boldsymbol{w}^*)$ is positive semi-definite and has at least one eigenvalue equal to zero. Typically, in the direction of the corresponding eigenvectors a higher-order term makes it a saddle point (e.g., $L(\boldsymbol{w}) = \sum_{i=1}^n w_i^3$ at $\boldsymbol{w}^* = 0$). Figure 2.1 gives an example.

1. When the input parameter is not a vector, but, e.g., a sequence of matrices, the same definitions hold, where the gradient and the Hessian are computed with respect to the vectorized version of the input parameters.

2.3.2 Gradient-based algorithms

When it comes to machine learning, the objective is to minimize the empirical risk with the hope that the true risk will be close to the empirical one, therefore also small. The most basic algorithm used is gradient descent, which consists in following the steepest direction downhill pointed by the opposite of the gradient at this point. More formally, Gradient descent is a first-order algorithm that is used in optimization to try to find a critical (stationary) point of a differentiable function L parameterized by θ, i.e., a point where the gradient of the function vanishes ($\nabla_\theta L(\theta) = 0$). The iterates of the algorithm are characterized by the following equation:

$$\theta_t = \theta_{t-1} - \eta \nabla_\theta f(\theta_{t-1}) \tag{2.3.1}$$

where $\eta > 0$ is a learning rate that can vary during the optimization process. When applied to very big datasets, gradient descent can be very slow.

Recall that the empirical risk that we want to optimize is defined by $\widehat{R}(f_{\mathbf{w}}) = \frac{1}{N}\sum_{i=1}^{N} \ell(f_{\mathbf{w}}(x_i), y_i)$. When there are several millions of examples, the gradient descent approach might take a long time since each iteration requires a forecast for every occurrence in the training dataset. Therefore, we can use a variant which is called Stochastic Gradient Descent (SGD). Instead of updating the parameter at the end of the batch of cases, SGD does so for each training instance, and one usually perform multiple passes over the training data.

In practice, rather than using the full batch of examples or one example at once, we usually take a mini-batch consisting of a few examples for each iteration of the algorithm.

Other heuristics can be added to the process to help make the optimization faster, for example adaptive learning rates (e.g. Adagrad, RMSProp), which consists of changing the learning rate during the training process depending on the previous gradients. Another algorithm that has gained popularity among deep learning practitioners is Adam. Adam can be seen as a combination of adaptive learning rates and momentum. We do not give the mathematical formulations of the algorithms but they can be found for example in [47].

2.3.3 On the importance of a landscape analysis at order 2

In this section, we first explain the importance of studying the objective function landscape at order 2, and then illustrate this on some simple examples.

2.3.3.1 Motivation

When the function we are trying to minimize is smooth, convex, and has a global minimizer, the gradient descent algorithm with a well-chosen learning rate converges to a first-order critical point, and this critical point is a global minimizer [105]. However, in general, finding a global optimum of a non-convex function is an NP-complete problem [104]; this is in particular the case for a simple 3-node neural network [20]. Despite that, when optimizing neural networks, the current practice is still to use gradient-based algorithms.

It has been known for decades that, even in the non-convex setting, under mild conditions gradient-based algorithms converge to a first-order critical point, in the sense that the iterates produced by the algorithm can reach an arbitrary small gradient after a finite (polynomial) number of iterations [105]. Adding smoothness conditions, recent works have shown that classical first-order algorithms escape strict saddle points in the long run [87, 85], and that some of them can be stopped in polynomial time at a nearly second-order critical point [2] [71, 73, 29, 72]. However, nothing prevents these algorithms to converge to non-strict saddle points or to spend many epochs in their vicinity, which translates into a long plateau during training. We give in Section 2.3.3.2 an intuition of the problems encountered with this kind of critical points with simple functions.

To see that this behavior actually occurs in practice, consider the simple experiment whose results are shown in Figures 2.3 and 2.4 (more details in Chapter 3, Appendix 3.G). For each run of this experiment, the parameters of a linear neural network of depth 5 are optimized to fit random input/output pairs. Discrepancy is measured with the square loss and we use the ADAM optimizer. Depending on the run, the algorithm is initialized in the vicinity either of a strict saddle point (in red) or of a non-strict saddle point (in blue). The distance between the random initial iterate and the saddle point is purposely not negligible: it is fixed to around 10% of the norm of the saddle point. Figure 2.3 shows the typical loss evolution for both cases. We can see that ADAM rapidly escapes from the strict saddle point but needs many epochs to escape the plateau in the vicinity of the non-strict saddle point. Figure 2.4 shows that this observation generalizes to most runs. We compare the empirical distributions of a random time (called *escape epoch*) defined as the epoch at which the loss has significantly decreased from its initial value. When initialized in the vicinity of non-strict saddle points, the algorithm suffers from an often large escape epoch and might be stopped there, without the possibility to distinguish this non-strict saddle point from a local minimum. Improving the analysis beyond local minimizers and characterizing strict and non-strict saddle points are therefore key to understand gradient descent dynamics and implicit regularization.

Beyond neural networks, the study of the loss landscape of specific non-convex optimization problems has revealed that they are tractable: phase retrieval [130], dictionary learning [131], tensor decomposition [42, 43, 39] and others [162, 103]. In fact, a landscape property which is shared by most of these problems is that every critical point is either a global minimizer or has a negative curvature. In other words, every second-order critical point is a global minimizer. For such problems, there are first-order algorithms which provably converge to global minimizers.

The general understanding of the landscape is not as good for neural networks. A regime which has been widely studied is the overparameterized regime (see [133] and [132] for a review), where it has been proved under some assumptions that for a wide non-linear fully connected neural network almost all local minima are global minima [108], or that there are no spurious valleys [107].

Many recent works have focused on linear neural networks, despite the fact that they are

2. More precisely, it is typically shown that, with high probability, such algorithms can be stopped in polynomial time and output a point with arbitrarily small gradient and nearly-positive semi-definite Hessian.

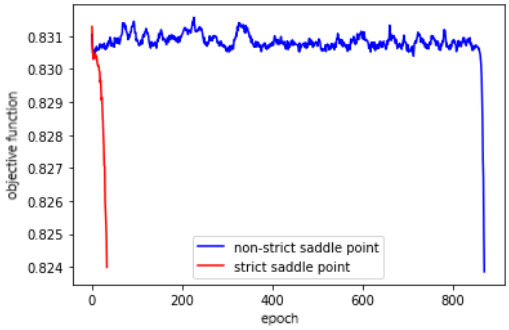

Figure 2.3 – The loss function during the iterative process, when initialized around a strict saddle point (in red) or a non-strict saddle point (in blue).

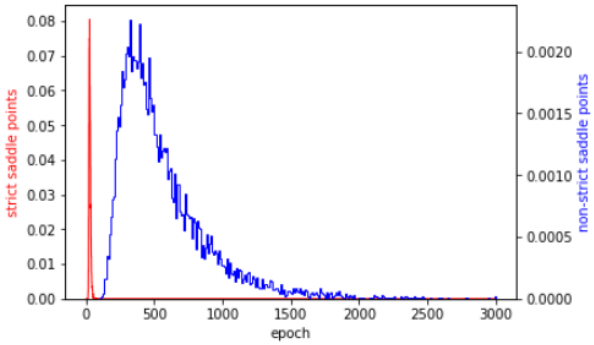

Figure 2.4 – Histogram of escape epochs, when initialized around a strict (in red) or a non-strict saddle point (in blue). For clarity, the y-axis is endowed with two scales. The right axis corresponds to the blue curve and the left to the red one.

rarely used to solve real-world applications. They indeed compute a linear map between the input and output spaces. The motivation for these studies is that the empirical risk of linear networks is highly non-convex and shares similar properties to that of practical non-linear neural networks. A complete review of the scientific publications on the landscape for linear networks is given in Section 2.3.4 but we would like to emphasize at this point that many results are strongly related to the properties of the landscape at order 2.

As shown by [120], linear networks exhibit nonlinear learning phenomena similar to those seen during the optimization of nonlinear networks, including long plateaus followed by rapid transitions to lower error solutions, as in Figure 2.3. Several works followed (e.g., [44, 45]) with formal proofs in several special cases. They express that for particular initializations the iterates trajectory spends some time on plateaus whose location defines an implicit regularization, if the algorithm is stopped early.

Other authors provide well-chosen initilizations and/or architectures for which the convergence to a global minimizer can be established [5, 11, 36, 33].

The difference between implicit regularization and global convergence depends on whether the iterates trajectory considered in these articles avoids saddle points or not. To better understand convergence properties of first-order algorithms, to give some hints on their finite-time dynamics and better understand the implicit regularization, a detailed analysis of the empirical risk landscape at order 2 is needed.

Furthermore, although it has been proved under mild conditions that every local minimizer is a global minimizer and that there exists no local maximizer [75, 156], very little is known concerning saddle points and the landscape at order 2. For 1-hidden layer linear networks, a proved fact is that every saddle point is strict, therefore leading to the property that every second-order critical point is a global minimizer [164, 75]. Unfortunately, this is not the case for linear networks with two hidden layers or more. For such neural networks, it has only been noted by [75] that there exist non-strict saddle points (e.g., when all the

weight matrices are equal to 0).

In Chapter 3, we make the missing additional step and completely characterize the landscape of the empirical risk at order 2, for the square loss and deep linear networks. In particular, we derive a simple necessary and sufficient condition for a first-order critical point of the empirical risk to be either a global minimizer, a strict saddle point or a non-strict saddle point.

2.3.3.2 Strict vs non-strict saddle points: some simple examples

Strict saddle points are not attractors: In this paragraph, we give a simple example of a function with a strict saddle point, the goal is to prove that gradient descent does not converge to the strict saddle point for almost all initializations ([86]) in a simple case to have an intuition. We will prove that the gradient flow trajectory avoids strict saddle points. For a function $f : w \to f(w)$, the gradient flow is defined by the following equation :

$$\frac{dw}{dt} = -\frac{\partial f}{\partial w}(w), \tag{2.3.2}$$

where we initialize the algorithm at a point $w(0) = w_0$.
Consider the function $f(x, y) = x^2 - y^2$. We have $\frac{\partial f}{\partial x}(x, y) = 2x$ and $\frac{\partial f}{\partial y}(x, y) = -2y$. Also, $\frac{\partial^2 f}{\partial x^2}(x, y) = 2$, $\frac{\partial^2 f}{\partial y^2}(x, y) = -2$ and $\frac{\partial^2 f}{\partial x \partial y}(x, y) = 0$. Hence the only critical point is $(0, 0)$, and according to the Hessian, this is a strict saddle point, because the eigenvalues of the Hessian matrix are -2 and 2.

Therefore the equations for the gradient flow becomes

$$\begin{cases} \frac{dx}{dt} = -\frac{\partial f}{\partial x} = -2x \\ \frac{dx}{dt} = -\frac{\partial f}{\partial y} = 2y \end{cases}$$

where we set $x(0) = x_0$ and $y(0) = y_0$. hence

$$\begin{cases} x(t) = x_0 \exp(-2t) \\ y(t) = y_0 \exp(2t) \end{cases}$$

We have $\lim_{t\to\infty} x(t) = 0$. If $y_0 = 0$, then $\forall t \geq 0, y(t) = 0$. If $y_0 \neq 0$, then $\lim_{t\to\infty} y(t) = \infty$. Therefore the only initializations which lead to a convergence to the strict saddle point are the points in the x-axis. Hence, the basin of attraction of this strict saddle point is of measure 0.
One can easily generalize the previous results to the case where the function is of more than 2 variables.

Example of an attractor non-strict saddle point Contrary to strict saddle points, non-strict saddle points can have a basin of attraction of positive measure. In this section, we give a simple example when this can happen. Consider the function $g(x) = x^3$. We have $g'(x) = 3x^2$ and $g''(x) = 6x$. Therefore the only critical point of this function is the point

0 and it is a non-strict saddle point. The gradient flow equation becomes

$$\frac{dx}{dt} = -\frac{\partial g}{\partial x} = -3x^2$$

where we set $x(0) = x_0 \neq 0$. Therefore, we have

$$-\frac{dx}{x^2} = 3dt \implies \frac{1}{x(t)} = 3t + \frac{1}{x_0} \implies x(t) = \frac{1}{3t + \frac{1}{x_0}}.$$

Hence $\lim_{t\to\infty} x(t) = 0$. Therefore, the gradient flow converges to the saddle point 0 for all initializations $x_0 > 0$. Hence the basin of attraction of the non-strict saddle point 0 is of positive measure.

Slowing down the gradient flow In the previous paragraphs, we have illustrated on examples that the gradient flow almost surely avoids the strict saddle points, but that for non-strict saddle points, this does not always hold.

Another aspect related to the saddle points is that since they are points where the gradient vanishes, the gradient flow algorithm can slow down close to them. In this section, we will explore this phenomenon in simple settings and see how this slowing compares between strict and non-strict saddle points.

We will consider two functions and compare the time that the gradient flow takes to escape the vicinity of each saddle point.

Consider first the function $f(x, y) = x^2 - y^2$. Recall that $(0, 0)$ is a strict saddle point of f. We have seen previously that the gradient flow leads to

$$\begin{cases} x_f(t) = x_0 \exp(-2t) \\ y_f(t) = y_0 \exp(2t) \end{cases}$$

where $x_f(0) = x_0$ and $y_f(0) = y_0 > 0$.

The second function we consider here is $g(x, y) = x^2 - y^4$. One can verify that the critical point $(0, 0)$ is a non-strict saddle point.

In this case the gradient flow (with $x_g(0) = x_0$ and $y_g(0) = y_0$) leads to $x_g(t) = x_0 \exp(-2t)$ and

$$\frac{dy_g}{dt} = -\frac{\partial g}{\partial y} = 4y^3 \implies -\frac{y_g'(t)}{y_g^3} = -4 \implies \frac{1}{y_g^2(t)} = -8t + \frac{1}{y_0^2} \implies y_g(t) = \frac{1}{\sqrt{-8t + \frac{1}{y_0^2}}}$$

We have $x_f(t) = x_g(t)$ and $\lim_{t\to\infty} x_f(t) = 0$. Therefore, it is sufficient to compare what happens with respect to the y-axis. We consider that the Gradient Flow has escaped the point when it reaches $y = 1$. It makes sense because in this case $y^2 = y^4 = 1$.

We have $y_f(t_f) = 1 \iff y_0 \exp(2t_f) = 1 \iff t_f = \frac{1}{2}\ln(\frac{1}{y_0})$.

On the other hand $y_g(t_g) = 1 \iff \frac{1}{\sqrt{-8t_g + \frac{1}{y_0^2}}} = 1 \iff t_g = \frac{1}{8}(\frac{1}{y_0^4} - 1)$.

Note that, in both cases, the closer y_0 is to 0, the harder it is to escape the saddle point and we have $\lim_{y_0 \to 0} t_g = \lim_{y_0 \to 0} t_f = \infty$. However when we initialize close to the saddle point, the escape time for a strict saddle point is logarithmic in $\frac{1}{y_0}$, while it is polynomial in $\frac{1}{y_0}$ when it comes to non-strict saddle points, and we have $\lim_{y_0 \to 0} \frac{t_g}{t_f} = \infty$. Hence, it takes a lot more time to escape from the vicinity of the non-strict saddle point than from the vicinity of the strict saddle point. The slow-down effect of the non-strict saddle point is stronger than the one of the strict saddle point.

2.3.4 Related Works

The study of linear neural networks can be divided into two categories. The first line of research is about the geometric landscape of the empirical risk for linear neural networks, while the second line is about the trajectory of gradient descent dynamics in linear networks. Our work lies within the first category.

Geometric landscape for linear networks: This first started with [8]. They proved that for a 1-hidden layer linear network, under some conditions on the data matrices, and for the square loss, every local minimizer is a global minimizer. [75] later generalized and extended this result to deep linear neural networks under mild conditions and again proved that every local minimizer is a global minimizer (this part has been proved later by [93] with weaker assumptions on the data and simpler proofs). This author also proved that every other critical point is a saddle point, that for a 1-hidden layer linear network all saddle points are strict, while for deeper networks, there exist non-strict saddle points ([75] exhibits a space of non-strict saddle points where all but one weight matrix are equal to zero). [156] gave a condition for a critical point to be either a global minimizer or a saddle point. [163] removed all assumptions on the data and gave analytical forms for the critical points of the empirical risk. In the characterization, the weight matrices are defined recursively and can be found by solving equations; in particular they gave a characterization of global minimizers. [109] showed using assumptions only on the width of the layers that every local minimizer is a global minimizer. They prove that this assumption on the architecture is sharp in the sense that without it, and if we do not make assumptions on the data matrices as in previous works, then there exists a poor local minimizer. [164] used assumptions only on the input data matrix, to prove that for a 1-hidden layer linear network, every local minimizer is a global minimizer and every other critical point has a negative curvature. [83] proved for different general convex losses that, under assumptions on the architecture, all local minima are global. Finally, [137] and [98] used results from algebraic geometry to give other properties about critical points of linear networks.

Most of the previous works focus on local minimizers. None of these works provide simple necessary and sufficient conditions for a saddle point to be strict or not. [3] In particular, in the case of more than two hidden layers, only very specific examples of non-strict saddle points were described. Furthermore, global minimizers were characterized but not explicitly

3. By "simple", we mean an easier-to-exploit condition than just looking at the smallest eigenvalue of the Hessian.

parameterized. See Chapter 3, Section 3.3.4, for more details.

Gradient dynamics and implicit regularization for linear networks: In this line of research, authors study the dynamics of first-order algorithms for linear networks, which they sometimes combine with results about the loss landscape. [5] proved that gradient descent converges to a global minimum at a linear rate, under assumptions on the width of the layers, the initial iterate, and the loss at initialization. Other works also proved similar results with different assumptions [36, 11, 144]. However, as noted by [123], these works consider strong assumptions on the loss at initialization. Indeed, [123] gave a negative result on a deep linear network of width 1, by proving that for standard initializations, gradient descent can take exponential time to converge to the global minimizer. The author also provided empirical examples of the same phenomenon happening for larger widths. On the other hand, [33] proved that if the layers are wide enough, convergence to a global minimimum can be achieved in polynomial time using a classical data-independent random Gaussian initialization (known as Xavier initialization). The required minimum width of the network depends on the norm of a global minimizer of the linear regression problem. As we will see in Section 3.3.4 this global convergence result can be re-interpreted in terms of the loss landscape at order 2.

On a similar line of research, [24] proved using assumptions on the architecture of the network and the data matrices that gradient flow almost surely converges to a global minimizer for a 1-hidden layer linear network. Later, [7] proved the same result under weaker assumptions on the data matrices. They also proved that, in deep linear networks, the gradient flow almost surely converges to global minimizers of the rank-constrained linear regression problem. [68] conjectured that the gradient flow for deep linear networks, when initialized with a small variance close to the origin, exhibits asymptotically a saddle-to-saddle dynamics where the rank increases with each saddle.

This is related to another consequence of the landscape properties: implicit regularization. [6] showed that, for matrix recovery, deep linear networks converge to low-rank solutions even when all the hidden layers are of size larger than or equal to the input and output sizes. [117] proved that, in deep matrix factorization, implicit regularization may not be explainable by norms, as all norms may go to infinity. They rather suggest seeing implicit regularization as a minimization of the rank. [119] and [44] proved with different assumptions on the data and a vanishing initialization that both gradient flow and discrete gradient dynamics sequentially learn solutions of a rank-constrained linear regression problem with a gradually increasing rank. Finally, [45] proved for a toy model that this incremental learning happens more often (with larger initialization), when the depth of the network increases. As we will see in Chapter 3, Section 3.3.4, these results can be re-interpreted in the light of the landscape at order 2.

2.4 Orthogonal convolutional layers

2.4.1 Motivation

Orthogonality constraint has first been considered for fully connected neural networks [4]. For Convolutional Neural Networks [84, 81, 161], the introduction of the orthogonality constraint is a way to improve the neural network in several regards. First, despite well-established solutions [59, 66], the training of very deep convolutional networks remains difficult. This is in particular due to vanishing/exploding gradient problems [61, 16]. As a result, the expressive capacity of convolutional layers is not fully exploited [66]. This can lead to lower performances on machine learning tasks. Also, the absence of constraint on the convolutional layer often leads to irregular predictions that are prone to adversarial attacks [134, 106]. Gradient vanishing/exploding avoidance, built-in robustness and better generalization capabilities are the main aims of the introduction of Lipschitz [**tsuzuku2018Lipschitz**, 134, 115, 48, 121] and orthogonality constraints to convolutional layers [146, 27, 65, 158, 91, 54, 114, 142, 138, 70, 90, 64, 70, 9, 145]. Orthogonal convolutional networks have been applied successfully in diverse applications, such as classification, segmentation, inpainting [142, 159, 82], or recently in few-shot learning [110]. Orthogonality is also proposed for Generative Adversarial Networks (GAN)[102], or even required for Wasserstein distance estimation, such as in Wasserstein-GAN [3, 52], and Optimal Transport based classifier [122].

Orthogonal convolutional networks are made of several orthogonal convolutional layers. This means that, when expressing the computation performed by the layer as a matrix, the matrix is orthogonal. The term 'orthogonal' applies both to square and non-square matrices [4]. In the latter case, it includes two commonly distinguished but related notions: row-orthogonality and column-orthogonality. Chapter 4 focuses on the theoretical properties of orthogonal convolutional layers. Furthermore, since deconvolution (also called transposed convolution) layers are defined using convolution layers, the results can also be applied to orthogonal deconvolution layers. We will consider the architecture of a convolutional layer as characterized by (M, C, k, S), where M is the number of output channels, C of input channels, convolution kernels are of size $k \times k$ and the stride parameter is S. Unless we specify otherwise, we consider convolutions with circular boundary conditions [5]. Thus, applied on input channels of size $SN \times SN$, the M output channels are of size $N \times N$. We denote by $\mathbf{K} \in \mathbb{R}^{M \times C \times k \times k}$ the kernel tensor and by $\mathcal{K} \in \mathbb{R}^{MN^2 \times CS^2N^2}$ the matrix that applies the convolutional layer of architecture (M, C, k, S) to C vectorized channels of size $SN \times SN$. [6]

We will first answer the important questions:

— **Existence:** What is a necessary and sufficient condition on (M, C, k, S) and N such that there exists an orthogonal convolutional layer (i.e. $\mathcal{K}$ orthogonal) for this architecture? How do the 'valid' and 'same' boundary conditions restrict the orthogonality existence?

4. The same property is sometimes called 'semi-orthogonal'.
5. Before computing a convolution the input channels are made periodic outside their genuine support.
6. See Chapter 4, Appendix 4.B for the formula of $\mathcal{K}$.

Besides, we will rely on recently published papers [142, 114] which characterize orthogonal convolutional layers as the zero level set of a particular function that is called L_{orth} in [142] (see Chapter 4, Section 4.1.1.2, for details). Formally, $\mathcal{K}$ is orthogonal if and only if $L_{orth}(\mathbf{K}) = 0$. They use L_{orth} as a regularization term and obtain impressive performances on several machine learning tasks (see [142]). The regularization is later successfully applied for medical image segmentation [159], inpainting [82] and few-shot learning [110].

In Chapter 4, we investigate the following theoretical questions:

— **Stability with regard to minimization errors:** Does $\mathcal{K}$ still have good 'approximate orthogonality properties' when $L_{orth}(\mathbf{K})$ is small but non zero? Without this guarantee, it could happen that $L_{orth}(\mathbf{K}) = 10^{-9}$ and $\|\mathcal{K}\mathcal{K}^T - Id\|_2 = 10^9$. This would make the regularization with L_{orth} useless, unless the algorithm reaches $L_{orth}(\mathbf{K}) = 0$.

— **Scalability and stability with regard to N:** Remarking that, for a given kernel tensor $\mathbf{K}$, $L_{orth}(\mathbf{K})$ is independent of N but the layer transform matrix $\mathcal{K}$ depends on N: When $L_{orth}(\mathbf{K})$ is small, does $\mathcal{K}$ remain approximately orthogonal and isometric when N grows? If so, the regularization with L_{orth} remains efficient even for very large N.

— **Optimization:** Does the landscape of L_{orth} lend itself to global optimization?

2.4.2 Related work

Orthogonal matrices form the Stiefel Manifold and were studied in [34]. In particular, the Stiefel Manifold is compact, smooth and of known dimension. It is made of several connected components. This can be a numerical issue since most algorithms have difficulty changing connected components during optimization. The Stiefel Manifold has many other nice properties that make it suitable for (local) Riemannian optimization [89, 90]. Orthogonal convolutional layers are a subpart of this Stiefel Manifold. To the best of our knowledge, the understanding of orthogonal convolutional layers is weak. There is no paper focusing on the theoretical properties of orthogonal convolutional layers.

Many articles [**scaman2018Lipschitz**, 148, 27, 129, 70, 48, 37] focus on Lipschitz and orthogonality constraints of the neural network layers from a statistical point of view, in particular in the context of adversarial attacks.

Many recent papers have investigated the numerical problem of optimizing a kernel tensor $\mathbf{K}$ under the constraint that $\mathcal{K}$ is orthogonal or approximately orthogonal. They also provide modeling arguments and experiments in favor of this constraint. We can distinguish two main strategies: **kernel orthogonality** [146, 27, 65, 158, 54, 70, 90, 64, 70, 9, 122] and **convolutional layer orthogonality** [91, 114, 142, 138]. The latter has been introduced more recently.

We denote the input of the layer by $X \in \mathbb{R}^{C\times SN\times SN}$ and its output by $Y = \mathbf{conv}(\mathbf{K}, X) \in \mathbb{R}^{M\times N\times N}$.

— **Kernel Orthogonality:** This class of methods views the convolution as a multiplication between a matrix $\overline{\mathbf{K}} \in \mathbb{R}^{M\times Ck^2}$ formed by reshaping the kernel tensor $\mathbf{K}$ (see,

for instance, [27, 142] for more details), and the matrix $U(X) \in \mathbb{R}^{Ck^2 \times N^2}$ whose columns contain the concatenation of the C vectorized patches of X needed to compute the M output channels at a given spatial position (see [60, 149]). We therefore have, $\text{Vect}(Y) = \text{Vect}\left(\overline{\mathbf{K}}U(X)\right)$. The kernel orthogonality strategy enforces the orthogonality of the matrix $\overline{\mathbf{K}}$.

— **Convolutional Layer Orthogonality:** This class of methods connects the input and the output of the layer directly by writing $\text{Vect}(Y) = \mathcal{K}\,\text{Vect}(X)$ and enforces the orthogonality of $\mathcal{K}$. The difficulty of this method is that the size of the matrix $\mathcal{K} \in \mathbb{R}^{MN^2 \times CS^2N^2}$ depends on N and can be very large.

Kernel orthogonality provides a numerical strategy whose complexity is independent of N. However, kernel orthogonality does not imply that $\mathcal{K}$ is orthogonal. In a nutshell, the problem is that the composition of an orthogonal embedding [7] and an orthogonal dimensionality reduction has no reason to be orthogonal. This phenomenon has been observed empirically in [91] and [70]. The authors of [142] and [114] also argue that, when $\mathcal{K}$ has more columns than rows (row orthogonality), the orthogonality of $\overline{\mathbf{K}}$ is necessary but not sufficient to guarantee $\mathcal{K}$ orthogonal. Kernel orthogonality and convolutional layer orthogonality are different, the latter better avoids gradient vanishing and feature correlation.

We can distinguish between two numerical ways of enforcing orthogonality during training:

— **Hard Orthogonality:** This method consists in keeping the matrix of interest orthogonal during the whole training process. This can be done either by optimizing on the Stiefel Manifold, or by considering a parameterization of a subset of orthogonal matrices (e.g., [90, 91, 138, 128, 65, 158]). Note that some hard convolutional layer orthogonality methods consider mappings of $\mathcal{K}$, therefore resulting in convolutions with kernels of size larger than $k \times k$.

— **Soft Orthogonality:** Another method to impose orthogonality of matrices during the optimization is to add a regularization of the type $\|WW^T - I\|^2$ to the loss of the specific task. This regularization penalizes the matrices far from orthogonal (e.g., [9, 27, 114, 142, 146, 54, 70, 64]).

Note that, unlike Kernel Orthogonality, Convolutional Layer Orthogonality deals directly with $\mathcal{K}$, and thus has a complexity that generally depends on N. However, in the context of Soft Convolutional Layer Orthogonality, the authors of [114, 142] introduce the regularizer L_{orth} which is independent of N (see Chapter 4, Section 4.1.1.2, for details), as a surrogate to $\|\mathcal{K}\mathcal{K}^T - \text{Id}_{MN^2}\|_F^2$ and $\|\mathcal{K}^T\mathcal{K} - \text{Id}_{CS^2N^2}\|_F^2$. In [142], orthogonal convolutional layers involving a stride are considered for the first time.

2.5 Approximation with neural networks

Recall that the approximation error which relates to the expressivity of the class of neural networks is given by the quantity $R(f_{\mathbf{w}^*}) - R^*$. Suppose that there exists g such that

7. Up to a re-scaling, when considering circular boundary conditions, the mapping U is orthogonal.

$R(g) = min_h R(h)$. The approximation error can be written

$$R(g) - R(f_{\mathbf{w}^*}) = \int [\ell(g(x), y) - \ell(f_{\mathbf{w}^*}(x), y)]\, dP(x, y).$$

If ℓ is 1-Lipschitz with respect to its second variable then we can write :

$$\int [\ell(g(x), y) - \ell(f_{\mathbf{w}^*}(x), y)]\, dP(x, y) \leq \int |g(x) - f_{\mathbf{w}^*}(x)| dP(x, y).$$

Therefore, if we can control the sup norm or the $L_1(P)$ norm of the difference between f and g we can control the approximation error of the risk decomposition. Hence it is interesting to have tight bounds on the L^p norm.

2.5.1 Universal approximation and benefits of depth

Neural networks are known to be universal approximators. Indeed for various functions spaces it has been proved that one can approximate a target function of interest using a neural network to an arbitrary depth, mathematically we have the following classical theorem which can be found with various conditions on the activations and the function spaces in [28, 62, 88, 76].

Theorem 3. For $\varepsilon > 0$, for any continuous function f and an activation function σ which is not polynomial, there exist a one hidden-layer neural network g such that

$$\sup_{x \in [0,1]^d} |f(x) - g(x)| \leq \varepsilon$$

This theorem is very interesting and tells us that neural networks are quite expressive and one can use them as approximators. However, it does not explain why depth is used in practice, and neither does it give us a rate of approximation (e.g., the number of weights needed to approximate a class of function within an error of ε).

Thus, two research directions followed after this theorem, the first one is about depth separation results, which state typically that there exist certain functions that can be approximated in a polynomial number of weights in ε with a deep network while a shallow network needs an exponential number of weights to approximate it within the same error. An example of such results separates depth 2 with depth 3 network and can be found in [135]:

Theorem 4. For any depth L, there exists a ReLU network f with $\mathcal{O}(L^2)$ layers and nodes such that for any ReLU network g with at most L layers and at most 2^L nodes we have

$$\int_0^1 |f(x) - g(x)| \geq \frac{1}{32}.$$

The other direction, which is the one we are going to focus on, is the quantitative rates of approximation of function spaces by neural networks.

2.5.2 Quantifying the approximation rate

One way to quantify the expressive power of neural networks is through the following problem. Let G be the set of all functions $g_{\mathbf{w}} : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ that can be represented by tuning the weights $\mathbf{w} \in \mathbb{R}^W$ of a feed-forward neural network with a fixed architecture, and let F be any set of real-valued functions on $\mathcal{X}$. A natural question is: how well functions $f \in F$ can be approximated by functions $g_{\mathbf{w}} \in G$? More precisely, given a norm $\|\cdot\|$ on functions, what is the order of magnitude of the (worst-case) *approximation error of F by G* defined by:

$$\sup_{f \in F} \inf_{g_{\mathbf{w}} \in G} \|f - g_{\mathbf{w}}\| , \tag{2.5.1}$$

and how small can it be given the numbers W, L of weights and layers, and some properties of F?

Lower bounds on the approximation error (2.5.1) can be useful in several ways. They provide a limit to the best approximation accuracy that one can hope to achieve if the number of weights or layers of the network is constrained, and help design optimal architectures under these constraints. They also imply a lower bound on the minimal number of weights or layers to include in a network in order to approximate any function in F with a given accuracy ε.

The case when $\|\cdot\|$ is the sup norm (defined by $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)|$) is rather well understood at least in some special cases. For example, when F is a Hölder ball of smoothness $s > 0$ and the network uses the ReLU activation function, Yarotsky [152] derived a lower bound on (2.5.1) of the order of $W^{-2s/d}$, later refined to $(LW)^{-s/d}$ (up to log factors) by [153, 154] when the depth of the network varies from $L = 1$ to $L \approx W$. Using the bit extraction technique, these authors showed that these lower bounds are achievable (up to log factors) with a carefully designed ReLU network architecture. Refined results in terms of width and depth were obtained by [125] when $s \leq 1$, while some other activation functions were also studied in [154].

The upper bounds are usually proven by first decomposing the target functions in a certain basis (e.g., Taylor expansion) and then approaching each basis vector with a neural network (e.g., approaching $(x, y) \to xy$). For the lower bounds the most used technique is based on lower bounding the VCdimension and then using known upper bounds on the VCdimension of neural networks with respect to W and L to conclude.

We give in the next section a typical proof of a lower bound on the approximation error when the norm is the sup norm. Note that this kind of proofs can be found also for example in [152, 125, 31].

2.5.3 Lower bounds in sup norm

In this section, we prove a lower bound on the approximation error defined in (2.5.1) when the norm is the sup norm, and G is a space of neural networks with a fixed architecture and polynomial activation. We use the typical proof and explain after why this cannot be applied directly to prove bounds in the L^p norm. For this example, we chose to do it when F is the space of monotonic function. We denote by $\mathcal{M}_d$ the space of functions non-decreasing

in each of the variables. We denote by $H_{\mathcal{A}}$ the space of functions generated by neural networks of fixed architecture $\mathcal{A}$ and where the weights vary. In Chapter 5, we will again see this space and prove bounds with respect to the L^p norm.

Proposition 2.5.1. Let $d \in \mathbb{N}^*$ and let $\mathcal{A}$ be a neural network architecture with piecewise-polynomial activation and W parameters. There exists a constant $c_d > 0$ depending only on d and a function $f \in \mathcal{M}_d$ such that for any $g \in H_{\mathcal{A}}$, $\|f - g\|_\infty \geq c_d W^{-2/d}$. In other words

$$\sup_{f \in \mathcal{M}_d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_\infty \geq c_d W^{-2/d}. \tag{2.5.2}$$

Proof. Let $\mathcal{A}$ be a piecewise-polynomial activation neural network architecture with $W \in \mathbb{N}^*$ parameters. If $\sup_{f \in \mathcal{M}_d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_\infty \geq \frac{1}{6d}$, then the result is straightforwarding by considering $c_d = \frac{1}{6d}$. Let $\varepsilon > \sup_{f \in \mathcal{M}_d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_\infty$ such that $\varepsilon < \varepsilon_0 = \frac{1}{6d}$, so that for any $f \in \mathcal{M}_d$, there exists $g \in H_{\mathcal{A}}$ such that $\|f - g\|_\infty \leq \varepsilon$. Firstly, we aim to lower bound W with respect to ε and we do so by bounding from below the pseudo-dimension of $H_{\mathcal{A}}$.

Let $N := \lceil \frac{1}{3d\varepsilon} \rceil$. Since $\varepsilon < \varepsilon_0 = \frac{1}{6d}$, we have $\frac{1}{3d\varepsilon} \leq N \leq \frac{1}{2d\varepsilon}$, therefore $\varepsilon \leq \frac{1}{2dN}$. We divide $[0,1]^d$ into a grid of N^d cubes $\mathcal{C}_{\mathbf{k}} = \prod_{i=1}^d \left(\frac{k_i}{N}, \frac{k_i+1}{N} \right]$, $\mathbf{k} = (k_1, \ldots, k_d) \in \{0, \ldots, N-1\}^d$. For $\sigma := (\sigma_{\mathbf{k}})_{\mathbf{k} \in \{0,\ldots,N-1\}^d} \in \{-1,1\}^{N^d}$, we define $f_\sigma : [0,1]^d \to [0,1]$ in the following way: for any $\mathbf{k} \in \{0, \ldots, N-1\}^d$, for any $x \in \mathcal{C}_{\mathbf{k}}$

$$f_\sigma(x) = \frac{\sum_{i=1}^d k_i}{Nd} + \frac{1}{2Nd} + \frac{\sigma_{\mathbf{k}}}{2Nd}.$$

It is straightforward to check that for any σ, f_σ can be naturally extended to a function over $[0,1]^d$ that has values in $[0,1]$; we consider these extensions without changing notations. We see then that $\mathcal{F} := \{f_\sigma, \sigma \in \{-1,1\}^{N^d}\}$ is a subset of $\mathcal{M}_d$ (all f_σ are non-decreasing from $[0,1]^d$ to $[0,1]$). It follows that for all $f \in \mathcal{F}$, there exists $g \in H_{\mathcal{A}}$ such that $\|f - g\|_\infty \leq \varepsilon$.

To show that $Pdim(H) \geq N^d$, we now construct a set of N^d points that we prove to be pseudo-shattered by $H_{\mathcal{A}}$, by using functions in $H_{\mathcal{A}}$ that approach $\mathcal{F}$. For all $\mathbf{k} \in \{0, \ldots, N-1\}^d$, let

- $x^{\mathbf{k}} := \left(k_1 + \frac{1}{2N}, \ldots, k_d + \frac{1}{2N} \right)$,
- $r_{\mathbf{k}} := \frac{\sum_{i=1}^d k_i}{Nd} + \frac{1}{2Nd}$,

and let $S = (x^{\mathbf{k}})_{\mathbf{k} \in \{0,\ldots,N-1\}^d}$. The $x^{\mathbf{k}}$'s are the points at the center of each cube in our grid, there are thus N^d of them. These are the points that we are going to pseudo-shatter with the sequence of thresholds $(r_{\mathbf{k}})_{\mathbf{k} \in \{0,\ldots,N-1\}^d}$. Let $z := (z_{\mathbf{k}})_{\mathbf{k} \in \{0,\ldots,N-1\}^d} \in \{-1,1\}^{N^d}$ be a sequence of labels associated to S. We choose $f_\sigma \in \mathcal{F}$ with $\sigma_{\mathbf{k}} = z_{\mathbf{k}}$ for all $\mathbf{k}$. By definition

of ε and since f_σ is in $\mathcal{M}_d$, there exists $g \in H_{\mathcal{A}}$ such that g approaches f_σ with error less than ε.

Using $\varepsilon \leq \frac{1}{2dN}$, we have,

$$\|f_\sigma - g_\sigma\|_\infty \leq \varepsilon \implies |f_\sigma(x^{\mathbf{k}}) - g_\sigma(x^{\mathbf{k}})| \leq \varepsilon \leq \frac{1}{2Nd} = |\frac{\sigma_{\mathbf{k}}}{2Nd}| = |f_\sigma(x^{\mathbf{k}}) - r^{\mathbf{k}}| \tag{2.5.3}$$

where the right-hand side holds for all $\mathbf{k} \in \{0, \dots, N-1\}^d$. Therefore, if $\sigma_{\mathbf{k}} > 0$, then $f_\sigma(x^{\mathbf{k}}) - g_\sigma(x^{\mathbf{k}}) \leq |f_\sigma(x^{\mathbf{k}}) - g_\sigma(x^{\mathbf{k}})| \leq f_\sigma(x^{\mathbf{k}}) - r^{\mathbf{k}}$, hence $g_\sigma(x^{\mathbf{k}}) > r^{\mathbf{k}}$. If $\sigma_{\mathbf{k}} < 0$, then $g_\sigma(x^{\mathbf{k}}) - f_\sigma(x^{\mathbf{k}}) \leq |f_\sigma(x^{\mathbf{k}}) - g_\sigma(x^{\mathbf{k}})| \leq r^{\mathbf{k}} - f_\sigma(x^{\mathbf{k}})$, hence $g(x^{\mathbf{k}}) < r^{\mathbf{k}}$.

It follows that the indicator $\mathbb{1}_{\{g_\sigma(x^{\mathbf{k}}) > r_{\mathbf{k}}\}}$ retrieves the labels $z_{\mathbf{k}}$'s ; that is $H_{\mathcal{A}}$ pseudo-shatters S and $Pdim(H_{\mathcal{A}}) \geq N^d$, or

$$Pdim(H_{\mathcal{A}}) \geq (3d\varepsilon)^{-d}. \tag{2.5.4}$$

It remains now to upper bound $Pdim(H_{\mathcal{A}})$ with respect to W, which combined with (2.5.4) will provide the lower bound for W in terms of ε. Theorem 14.1 in [2] provides an architecture $\mathcal{A}'$ allowing skip-connexions and with a number of parameters $W' \leq W + 2$, such that

$$Pdim(H_{\mathcal{A}}) \leq VCdim(H_{\mathcal{A}'}).$$

We now use Theorem 8.7 in [2] which implies that $VCdim(H_{\mathcal{A}'}) \leq c_1 W'^2$ for an universal constant $c_1 > 0$. Since $W' \leq W + 2$,

$$Pdim(H_{\mathcal{A}}) \leq c_2 W^2, \tag{2.5.5}$$

for an universal constant c_2. Combining inequalities (2.5.4) and (2.5.5) yield the result $W \geq c_3 \varepsilon^{-d/2}$, where c_3 depends only on d.

To finish the proof, let us consider a sequence $(\varepsilon_n)_{n \geq 1}$ such that for all n

$$\varepsilon_n > \sup_{f \in \mathcal{M}_d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_\infty,$$

and $\lim_{n \to \infty} \varepsilon_n = \sup_{f \in \mathcal{M}_d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_\infty$. For any n, $H_{\mathcal{A}}$ approaches $\mathcal{M}_d$ with error less than ε_n, by definition. Inequalities (2.5.4) and (2.5.5) thus apply, hence

$$cW^{-2/d} \leq \varepsilon_n, \tag{2.5.6}$$

for a constant c depending only on d. The result follows by considering the limit in (2.5.6) when $n \to +\infty$. □

2.5.4 Approximation in L^p norm

It is also interesting to study (2.5.1) with the $L^p(\mu)$ norm, defined by

$$\|f\|_{L^p(\mu)} = \Big(\int_X |f(x)|^p d\mu(x)\Big)^{1/p},$$

for $1 \leq p < +\infty$ and some probability measure μ on $\mathcal{X}$. There is a qualitative difference between measuring the error in sup norm or in $L^p(\mu)$ norm, $p < +\infty$. In the former case, the error is small only if the approximation is good over the whole domain. In the latter case, the error can be small even if the approximation is inaccurate over a small portion of the domain. Since the $L^p(\mu)$ approximation problem corresponds to approximating functions in F in a more "average" sense than in sup norm, a natural question is whether the same accuracy can be achieved with a smaller network or not. Unfortunately, however, the proof strategies behind the lower bounds of [152, 153, 154, 125] are specific to the sup norm. Indeed, the break happens at equation (2.5.3), since if we use L^p norm, we cannot guarantee that the difference between g and f are less than ε for the centers x^k, or even for other points uniformly, i.e., for each f_σ there exists x_σ such that $|f_\sigma(x_\sigma) - g_\sigma(x_\sigma)| \leq \varepsilon$. Hence, one cannot directly exhibit a set of points that are shattered by g_σ. DeVore et al. [31] indeed commented: "When we move to the case $p < \infty$, the situation is even less clear [...] we cannot use the VC dimension theory for $L^p(\Omega)$ approximation. [...] What is missing vis-à-vis Problem 8.13 is what the best bounds are and how we prove lower bounds for approximation rates in $L^p(\Omega)$, $p \neq \infty$."

Existing lower bounds in $L^p(\mu)$ norm. Several papers provided lower bounds in some special cases, under some restrictions on the set to approximate F, the neural network, the approximation metric, or the encoding map $f \in F \mapsto \mathbf{w}(f) \in \mathbb{R}^W$.

When F is a space of smoothness s, a first result which is based on [32] states that when imposing the weights to depend continuously on the function to be approximated, one can not achieve a better approximation rate than $W^{-\frac{s}{d}}$.

For the same F, another result for $p = 2$ and for activation functions which are continuous ([95, 96]) proves a lower bound on the approximation of functions of smoothness s on a compact of $\mathbb{R}^d$, by one hidden-layer neural networks, of order $W^{-\frac{s}{d-1}}$. A matching upper bound is proven for a particular activation function, which is sigmoidal but pathological ([97]). For this same activation function, they prove that contrary to the one-hidden-layer case, there is no lower bound in the case of two-hidden-layer networks. The result is based on the Kolmogorov-Arnold superposition theorem.

In [126], the authors study approximation by shallow neural networks with bounded weights and activations of the form ReLU^k for an integer k. They approximate the closure of the convex hull of shallow ReLU^k-neural networks with constrained weights. They obtain optimal lower bounds of order $W^{-\frac{1}{2}-\frac{2k+1}{2d}}$ in any norm $\|\cdot\|_X$, where X is a Banach space to which the approximation functions belong and such that these functions are uniformly bounded w.r.t. $\|\cdot\|_X$.

Approximation lower bounds in $L^p(\mu)$ norm, $p \geq 1$, have also been studied in the quantized neural networks setting (networks with weights encoded with a fixed number of bits). In [111], under weak assumptions on the activation function, the authors prove a

lower bound on the minimal number of nonzero weights W that are required for a network to approximate a class of binary classifiers with L^p error at most ε. They show that W is at least of the order $\varepsilon^{-\frac{p(d-1)}{\beta}} \log_2^{-1}(1/\varepsilon)$, where β is a smoothness parameter. Later works including [141, 51] derive lower bounds for approximation by quantized networks in various norms.

In Chapter 5, we give a general approximation lower bound for the L^p norm error based on the metric entropy on the space F and the pseudo-dimension of the space G. And we apply it to the case of neural networks to solve the open question of lower bounding the approximation errors for non-quantized neural networks with the L^p norm.

CHAPTER 3

The loss landscape of deep linear neural networks: a second-order analysis

Abstract

We study the optimization landscape of deep linear neural networks with the square loss. It is known that, under weak assumptions, there are no spurious local minima and no local maxima. However, the existence and diversity of non-strict saddle points, which can play a role in first-order algorithms' dynamics, have only been lightly studied. We go a step further with a full analysis of the optimization landscape at order 2. We characterize, among all critical points, which are global minimizers, strict saddle points, and non-strict saddle points. We enumerate all the associated critical values. The characterization is simple, involves conditions on the ranks of partial matrix products, and sheds some light on global convergence or implicit regularization that have been proved or observed when optimizing linear neural networks. In passing, we provide an explicit parameterization of the set of all global minimizers and exhibit large sets of strict and non-strict saddle points.

This chapter is based on joint work with François Malgouyres and Sébastien Gerchinovitz.

Contents

3.1 Introduction

Deep learning has been widely used recently due to its good empirical performances in image recognition, natural language processing, speech recognition, among other fields. However, there is still a gap between theory and practice. One of the aspects that are partially missing in the picture is why gradient-based algorithms can achieve low training error despite a highly non-convex function. Another partially open question is why they generalize well to unseen data despite many more parameters than the number of points in the training set, and how implicit regularization can help with that.

Various research directions have been explored to answer these questions, including double-descent and benign overfitting (e.g., [15, 58, 13]), gradient flow dynamics in Wasserstein space (e.g., [25, 99]), landscape analysis of the empirical risk (e.g., [26, 75, 57, 56]). In this chapter, we follow the latter direction, by better characterizing the local structures around critical points of the empirical risk, in the case of deep linear neural networks with the square loss.

We refer the reader to Chapter 2, Section 2.3, for the context and motivation of this chapter.

3.1.1 Summary of our contributions

Our contributions on the optimization landscape of deep linear networks can be summarized as follows.

- We characterize the square loss landscape of deep linear networks at order 2 (see Theorem 5 and Figure 3.2). That is, under some classical and weak assumptions on the data, we characterize, among all first-order critical points, which are global minimizers, strict saddle points, and non-strict saddle points. The characterization is simple and involves conditions on the ranks of partial matrix products. To the best of our knowledge, this is the first work that gives a simple necessary and sufficient condition for a saddle point to be strict or non-strict.
- Several results follow from the characterization: under the same assumptions,
  - — we first immediately recover the fact that all saddle points are strict for one-hidden layer linear networks;
  - — more importantly, for deeper networks, when proving that all cases considered in the characterization can indeed occur, we exhibit large sets of strict and non-strict saddle points (see Proposition 4 and its proof in Appendix 3.B.8);
  - — we show that the non-strict saddle points are associated with r_{max} plateau values of the empirical risk, where r_{max} is the size of the thinnest layer of the network (see Theorem 5). Typically these are values of the empirical risk that first-order algorithms can take for some time, as in Figure 2.3, and which might be confused with a global minimum.
- As a by-product of our analysis, we obtain explicit parameterizations of sets containing or included in the set of all first-order critical points (see Propositions 5 and 6). We also derive an explicit parameterization of the set of all global minimizers (see Proposition 7).

The above results are compared in details with previous works in Section 3.3.4. In particular, our second-order characterization sheds some light on two phenomena:

- Implicit regularization: we recover the fact that every non-strict saddle point corresponds to a global minimizer of the rank-constrained linear regression problem, as shown in [7, Proposition 35]. Our characterization additionally shows that only a fraction of the critical points corresponding to rank-constrained solutions are non-strict saddle points. The others are strict saddle points. Given the differences in the behavior of first-order algorithms in the vicinity of strict and non-strict saddle points as illustrated on Figures 2.3 and 2.4, our results open new research directions related to the very nature of implicit regularization and its stability.
- Our characterization can also be useful to understand recent global convergence results in terms of the loss landscape at order 2. In particular, we show how to re-interpret a proof of [33] to see that gradient descent with Xavier initialization on wide enough deep linear networks meets no non-strict saddle points on its trajectory.

3.1.2 Outline of the chapter

The chapter is organized as follows. We define the setting in Section 3.2 and state our results in Section 3.3. We prove our main result (Theorem 5) in Section 3.4. More precisely, we detail the proof structure and main arguments but defer all technical derivations to the appendix. We finally conclude our work in Section 3.5.

Most technical details can be found in the appendix, which is organized as follows. Section 3.A contains additional notation and lemmas that will be useful in all subsequent sections. In Section 3.B we provide proofs of propositions and lemmas related to first-order critical points, while Section 3.C gathers the proofs for the parameterization of first-order critical points and global minimizers. Sections 3.D, 3.E, and 3.F contain proofs corresponding to each subsection of Section 3.4. Finally, in Section 3.G, we describe in more details the illustrative experiment underlying Figures 2.3 and 2.4.

3.2 Setting

In this section we formally define our setting (deep linear networks with square loss), set some notation, and describe our assumptions on the data.

Model and notation: We consider a fully-connected linear neural network of depth $H \geq 2$. The neural network consists of H layers and maps any input $x \in \mathbb{R}^{d_x}$ to an output $W_H \cdots W_1 x \in \mathbb{R}^{d_y}$, where $W_H \in \mathbb{R}^{d_y \times d_{H-1}}, \ldots, W_h \in \mathbb{R}^{d_h \times d_{h-1}}, \ldots, W_1 \in \mathbb{R}^{d_1 \times d_x}$, are the matrices associated with the H layers (d_h is the width of layer h). We set $d_H = d_y$ and $d_0 = d_x$. The input layer is of size d_x and the output layer is of size d_y. We also define the smallest width of the layers as $r_{max} = \min(d_H, \ldots, d_0)$. [1] We denote the parameters of the model by $\mathbf{W} = (W_H, \ldots, W_1)$.

Let $(x_i, y_i)_{i=1..m}$ with $x_i \in \mathbb{R}^{d_x}$ and $y_i \in \mathbb{R}^{d_y}$, be the training set that we gather column-wise in matrices $X \in \mathbb{R}^{d_x \times m}$ and $Y \in \mathbb{R}^{d_y \times m}$. We consider the empirical risk L defined by:

$$L(\mathbf{W}) = \sum_{i=1}^{m} \|W_H W_{H-1} \cdots W_2 W_1 x_i - y_i\|_2^2 = \|W_H \cdots W_1 X - Y\|^2 \,,$$

where $\|.\|_2$ is the Euclidean norm and $\|.\|$ denotes the Frobenius norm of a matrix.
We set:

$$\Sigma_{XX} = \sum_{i=1}^{m} x_i x_i^T = XX^T \in \mathbb{R}^{d_x \times d_x} \quad , \quad \Sigma_{YY} = \sum_{i=1}^{m} y_i y_i^T = YY^T \in \mathbb{R}^{d_y \times d_y},$$

$$\Sigma_{XY} = \sum_{i=1}^{m} x_i y_i^T = XY^T \in \mathbb{R}^{d_x \times d_y} \quad , \quad \Sigma_{YX} = \sum_{i=1}^{m} y_i x_i^T = YX^T \in \mathbb{R}^{d_y \times d_x},$$

where, A^T denotes the transpose of A.

1. The notation r_{max} comes from the fact that it is the maximum possible rank of the product $W_H \cdots W_1$.

Assumption ($\mathcal{H}$). Throughout the chapter, we assume that $d_y \leq d_x \leq m$, that Σ_{XX} is invertible, and that Σ_{XY} is of full rank d_y. We define $\Sigma^{1/2} = \Sigma_{YX}\Sigma_{XX}^{-1}X \in \mathbb{R}^{d_y \times m}$ and $\Sigma = \Sigma^{1/2}(\Sigma^{1/2})^T = \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY} \in \mathbb{R}^{d_y \times d_y}$. We assume that the singular values of $\Sigma^{1/2}$ are all distinct (i.e., that Σ has d_y distinct eigenvalues).

These assumptions are exactly the ones considered in [75]. Note that we do not make any assumption on the width of the hidden layers. As noted by [8], full rank matrices are dense, and deficient-rank matrices are of measure 0. In general, $m \geq d_x \geq d_y$, which is the classical learning regime, is essentially sufficient to have the other assumptions verified, due to the randomness of the data.
Let

$$\Sigma^{1/2} = U\Delta V^T \tag{3.2.1}$$

be a singular value decomposition of $\Sigma^{1/2}$, where $U \in \mathbb{R}^{d_y \times d_y}$ and $V \in \mathbb{R}^{m \times m}$ are orthogonal, and the diagonal elements of $\Delta \in \mathbb{R}^{d_y \times m}$ are in decreasing order.
Since $\Sigma = \Sigma^{1/2}(\Sigma^{1/2})^T$, Σ can be diagonalized as $\Sigma = U\Lambda U^T$ where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_{d_y})$, with $\lambda_1 > \cdots > \lambda_{d_y} \geq 0$. Moreover, a consequence of Assumption ($\mathcal{H}$) is that Σ is positive definite (see Lemma 4); therefore, we have $\lambda_{d_y} > 0$.

Additional notation: We list below some notation and conventions that will use throughout the chapter.
For all integers $a \leq b$, we denote by $[\![a, b]\!]$ the set of integers between a and b (including a and b). If $a > b$, $[\![a, b]\!]$ is the empty set (e.g. $[\![1, 0]\!] = \emptyset$).
If $\mathcal{S} = \emptyset$, then $\sum_{i \in \mathcal{S}} \lambda_i = 0$.
Given a matrix $A \in \mathbb{R}^{p \times q}$, $\operatorname{col}(A)$, $\operatorname{Ker}(A)$ and $\operatorname{rk}(A)$, denote respectively the column space, the null space and the rank of A.
For a matrix $A \in \mathbb{R}^{p \times q}$, we write $A_i \in \mathbb{R}^p$ for the i-th column of A and $A_{\mathcal{J}} \in \mathbb{R}^{p \times |\mathcal{J}|}$ for the sub-matrix obtained by concatenating the column vectors A_i, for $i \in \mathcal{J}$. The identity matrix of size p will be denoted by I_p.
When we write $W_h \cdots W_{h'}$ for $h > h'$, the expression denotes the product of all W_j from $j = h$ to $j = h'$. For notational compactness, we allow two additional cases: when $h = h'$, the expression simply denotes W_h, and when $h' = h + 1$, it stands for the identity matrix $I_{d_h} \in \mathbb{R}^{d_h \times d_h}$.

Considering submatrices of compatible sizes, we define a block matrix by one of the three following ways:

— $[A, B]$ is the horizontal concatenation of the matrices A and B;

— $\begin{bmatrix} G \\ H \end{bmatrix}$ is the vertical concatenation of G and H;

— $\begin{bmatrix} C & D \\ E & F \end{bmatrix}$ is a 2×2 block matrix.

By convention, in block matrices, some blocks can have 0 lines or 0 columns; this means that such blocks do not exist. However if we have a product between two matrices that have 0 as the common size (the number of columns for the first matrix, of the lines for the second matrix), then their product equals a zero matrix, of the right size. More formally, if $A \in \mathbb{R}^{n\times 0}$ and $B \in \mathbb{R}^{0\times p}$, then, by convention, $AB = 0_{n\times p}$. Note that the product of block matrices is compatible with this convention (e.g., $[A\ ,\ B]\begin{bmatrix} C \\ D \end{bmatrix} = AC + BD$ is still true if $B \in \mathbb{R}^{n\times 0}$ and $D \in \mathbb{R}^{0\times p}$).
Further notation that are used in the appendix can be found at the beginning of Appendix 3.A.

3.3 Main results

In this section, we state the main results of this chapter. We start with a necessary condition for being a first-order critical point of L (Proposition 1), to which we give a light reciprocal (Proposition 2). We then move to our main result (Theorem 5), which is a second-order classification of all first-order critical points. It distinguishes between global minimizers, strict saddle points and non-strict saddle points. Finally, the third result is a necessary parameterization for critical points (Proposition 5) and an explicit parameterization of all global minimizers (Proposition 7). These results are compared with previous works in Section 3.3.4. All the proofs can be found in Section 3.4 or in the appendix, where most technical derivations are deferred.

3.3.1 First-order critical points: preliminary results

In the next proposition we restate in our framework a necessary condition for being a first-order critical point, which was already present in [8] and most of the papers in this line of research. This proposition will serve later to distinguish between different types of critical points.

Proposition 1 (Global map and critical values). Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L and set $r = \mathrm{rk}(W_H \cdots W_1) \in [\![0, r_{max}]\!]$.
There exists a unique subset $\mathcal{S} \subset [\![1, d_y]\!]$ of size r such that:

$$W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1},$$

where U was defined in (3.2.1). We say that the critical point $\mathbf{W}$ is *associated with* $\mathcal{S}$. The associated critical value is

$$L(\mathbf{W}) = \mathrm{tr}(\Sigma_{YY}) - \sum_{i\in\mathcal{S}} \lambda_i.$$

The proof can be found in Appendix 3.B.2. The result is true even for $r = 0$, using the conventions from Section 3.2 (in this case, $\mathcal{S} = \emptyset$).
Note that $\Sigma_{YX}\Sigma_{XX}^{-1}$ corresponds to the solution of the classical linear regression problem.

Therefore, we can see that for every critical point $\mathbf{W}$ of L, the product $W_H \cdots W_1$ is the projection of this least-squares estimator onto a subspace generated by a subset of the eigenvectors of Σ. Note that $\mathrm{tr}(\Sigma_{YY}) = \|Y\|^2$.

The following proposition is a light reciprocal to Proposition 1, by showing that all subsets $\mathcal{S}$ and the corresponding critical values $\mathrm{tr}(\Sigma_{YY}) - \sum_{i\in\mathcal{S}} \lambda_i$ are associated to an existing critical point. In particular the largest critical value is reached for $\mathcal{S} = \emptyset$ and the smallest critical value for $\mathcal{S} = [\![1, r_{max}]\!]$.

Proposition 2. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. For any $\mathcal{S} \subset [\![1, d_y]\!]$ of size $r \in [\![0, r_{max}]\!]$, there exists a first-order critical point $\mathbf{W}$ associated with $\mathcal{S}$.

The proof of Proposition 2 is deferred to Appendix 3.B.6.

3.3.2 Second-order classification of the critical points of L

The main result of this section is Theorem 1 below, where we classify all first-order critical points into global minimizers, strict saddle points and non-strict saddle points. To state Theorem 5 we first need to introduce some definitions.

Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point of L. Below, we introduce the notions of *complementary block*, *tightened pivot* and *tightened critical point* that are key to the main results. Consider the sequence of H matrices $W_H, \ldots, W_2, W_1$ and connect them by plugging Σ_{XY} between W_1 and W_H so as to form a cycle as on Figure 3.1. Note that the dimensions of these matrices allow us to consider any product of consecutive matrices on this cycle, e.g., $W_H W_{H-1} W_{H-2}$ or $W_2 W_1 \Sigma_{XY} W_H$ (the matrix Σ_{XY} between W_1 and W_H is key here). Such products of consecutive matrices in the cycle are what we call "**blocks**". In the sequel, we call "**pivot**" any pair of indices $(i, j) \in [\![1, H]\!]$, with $i > j$, and we consider blocks around a pivot (i, j), as defined formally below.

Definition 1 (Complementary blocks). Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point of L.
For any pivot $(i, j) \in [\![1, H]\!]$, $(i > j)$, we define the two complementary blocks to (i, j) as:

$$W_{j-1} \cdots W_1 \Sigma_{XY} W_H \cdots W_{i+1} \qquad \text{and} \qquad W_{i-1} \cdots W_{j+1}.$$

The general case is represented on Figure 3.1.
Note that, when $i = j + 1$, the second complementary block is $W_j W_{j+1}$, which using the convention in Section 3.2 is I_{d_j}. Similarly, if $i = H$ and $j = 1$, the first complementary block is Σ_{XY}. First we state a proposition about the ranks of the complementary blocks which is key to our analysis.

Proposition 3. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point of L and $r = \mathrm{rk}(W_H \cdots W_1)$. For any pivot (i, j), the rank of each of the two complementary blocks is larger than or equal to r.

The proof is in Appendix 3.B.7. The boundary case when at least one of the two ranks is equal to r plays a special role in the loss landscape at order 2.

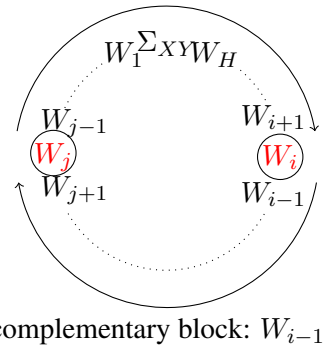

Figure 3.1 – Complementary blocks to the pivot (i, j) .

Definition 2 (Tightened pivot). Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point of L and let $r = \mathrm{rk}(W_H \cdots W_1)$.
We say that a pivot (i, j) is **tightened** if and only if at least one of the two complementary blocks to (i, j) is of rank r.

Definition 3 (Tightened critical point). Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point of L. We say that $\mathbf{W}$ is tightened if and only if every pivot (i, j) is tightened.

Note that a sufficient condition for a first-order critical point $\mathbf{W}$ to be tightened is the existence of three weight matrices W_{h_1}, W_{h_2} and W_{h_3} of rank $r = \mathrm{rk}(W_H \cdots W_1)$.
Note that when $H = 2$, there is no tightened critical point with $r < r_{max}$, because the pivot $(2, 1)$ is not tightened (both complementary blocks Σ_{XY} and I_{d_1} are of full rank, which is larger than or equal to $r_{max} = \min\{d_y, d_1, d_x\}$).

We can now state our main theorem, which characterizes the nature of any first-order critical point $\mathbf{W}$ in terms of the associated index set $\mathcal{S}$ and of tightening conditions. The corresponding classification, which is illustrated on Figure 3.2, complements the result of [75] stating that every critical point is a global minimizer or a saddle point. We recall that $r_{max} = \min(d_H, \ldots, d_0)$ is the width of the thinnest layer, and that U corresponds to the eigenvectors of Σ (see (3.2.1)).

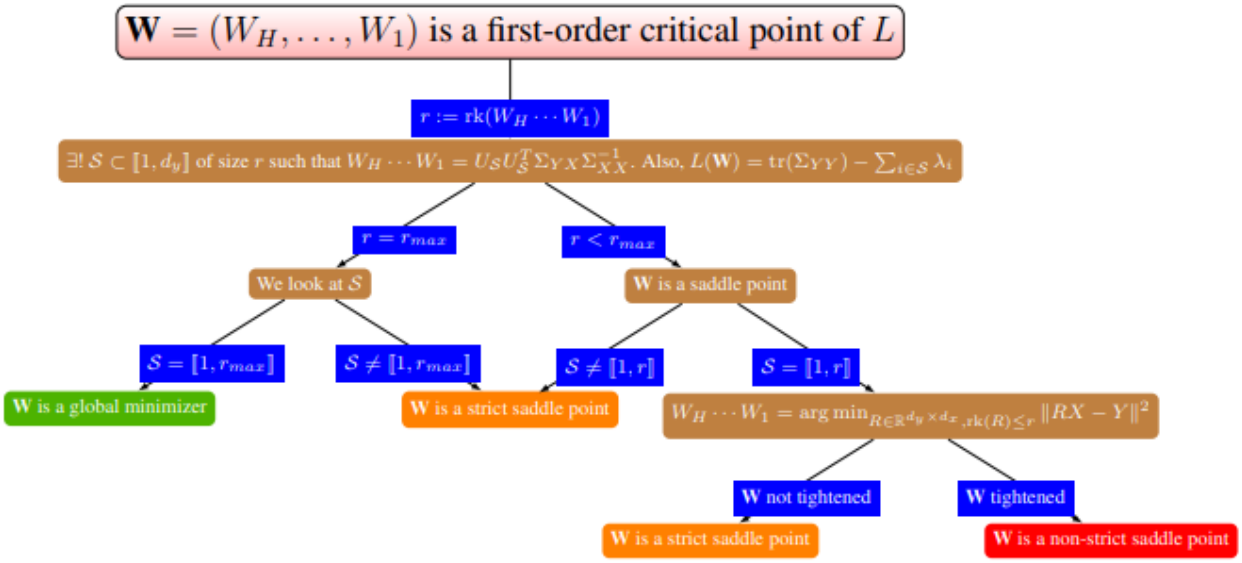

Figure 3.2 – Second-order classification of the critical points of L.

Theorem 5 (Classification of the critical points of L). Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true.

Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point of L and set $r = \mathrm{rk}(W_H \cdots W_1) \leq r_{max}$. Following Proposition 1, we consider the index set $\mathcal{S}$ associated with $\mathbf{W}$.

- When $r = r_{max}$:
  - if $\mathcal{S} = [\![1, r_{max}]\!]$, then $\mathbf{W}$ is a global minimizer.
  - if $\mathcal{S} \neq [\![1, r_{max}]\!]$, then $\mathbf{W}$ is not a second-order critical point ($\mathbf{W}$ is a strict saddle point).
- When $r < r_{max}$: $\mathbf{W}$ is a saddle point.
  - if $\mathcal{S} \neq [\![1, r]\!]$, then $\mathbf{W}$ is not a second-order critical point ($\mathbf{W}$ is a strict saddle point).
  - if $\mathcal{S} = [\![1, r]\!]$: we have $W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \in \arg\min_{R \in \mathbb{R}^{d_y \times d_x}, \mathrm{rk}(R) \leq r} \|RX - Y\|^2$.
    - if $\mathbf{W}$ is not tightened, then $\mathbf{W}$ is not a second-order critical point ($\mathbf{W}$ is a strict saddle point).
    - if $\mathbf{W}$ is tightened, then $\mathbf{W}$ is a second-order critical point ($\mathbf{W}$ is a non-strict saddle point).

The proof of Theorem 5 is given in Section 3.4, with most technical derivations deferred to the appendix. We now make two remarks. Note from the above that every non-strict saddle point corresponds to a global minimizer of the rank-constrained linear regression problem, as already shown by [7, Proposition 35]. In Remark 1 below we explain why

our characterization sheds a new light on implicit regularization, and opens up research questions.

The next proposition shows the existence of both tightened and non-tightened critical points for $H \geq 3$ (there are no tightened critical points when $H = 2$ and $r < r_{max}$). Combining this result with Proposition 2 indicates that all conclusions of Theorem 5 can be observed.

Proposition 4. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. For $H \geq 3$, for every $\mathcal{S} = [\![1, r]\!]$ with $0 \leq r < r_{max}$, there exist both a tightened critical point and a non-tightened critical point associated with $\mathcal{S}$.

The proof is postponed to Appendix 3.B.8. It is constructive: we exhibit large sets of tightened and non-tightened critical points.
We can draw additional consequences from Theorem 5 and Propositions 2 and 4:

- For $H = 2$, for any $r < r_{max}$, there exist strict saddle points satisfying $W_H \cdots W_1 \in \arg\min_{R \in \mathbb{R}^{d_y \times d_x}, \mathrm{rk}(R) \leq r} \|RX - Y\|^2$.
- For $H \geq 3$, for any $r < r_{max}$, there exist both strict and non-strict saddle points satisfying $W_H \cdots W_1 \in \arg\min_{R \in \mathbb{R}^{d_y \times d_x}, \mathrm{rk}(R) \leq r} \|RX - Y\|^2$.
- In the special case $r = 0$, we have $\mathcal{S} = \emptyset$ and $\emptyset = [\![1, r]\!]$ by convention (see Section 3.2), so that $\mathcal{S} = [\![1, r]\!]$. In this case, Theorem 5 and Proposition 4 together imply that their exist both strict and non-strict saddle points $\mathbf{W}$ such that $W_H \cdots W_1 = 0$ when $H \geq 3$.

Remark 1 (Implicit regularization). As detailed in Sections 2.3.4 and 3.3.4, by analyzing the gradient dynamics in well-chosen settings, it has been established that the iterates trajectory passes in the vicinity of critical points $\mathbf{W}$ such that $W_H \cdots W_1 = \arg\min_{R \in \mathbb{R}^{d_y \times d_x}, \mathrm{rk}(R) \leq r} \|RX - Y\|^2$, for increasing $r \in [\![0, r_{max}]\!]$. In such settings, the gradient dynamics sequentially finds the best linear regression predictor in

$$\mathcal{D}_r = \{R \in \mathbb{R}^{d_y \times d_x} \ , \ \mathrm{rk}(R) \leq r\},$$

for increasing r. The subset $\mathcal{D}_r \subset \mathbb{R}^{d_y \times d_x}$ is independent of X, Y and the network architecture, and plays the role of a regularization constraint in the predictor space.

From Theorem 5, we know that the critical points such that $W_H \cdots W_1 = \arg\min_{R \in \mathbb{R}^{d_y \times d_x}, \mathrm{rk}(R) \leq r} \|RX - Y\|^2$ can be either strict saddle points or non-strict saddle points. From Proposition 4 we know that both cases exist. To the best of our knowledge, whether the saddle points approached by the iterates trajectory are strict or non-strict and the impact of this property on the implicit regularization phenomenon have not been studied. Though this study goes beyond the scope of this chapter, let us sketch the main trends that we can anticipate from our results.

First, the experiments on Figures 2.3 and 2.4 suggest that, given a distance between the initial iterate and a saddle point, the number of iterations spent by a first-order algorithm in the vicinity of this saddle point is typically larger when the latter is non-strict. This suggests implicit regularization should be more easily observed around such points in practice. On the other hand, looking at the rank constraint in Definition 2 (which corresponds to the very last item of Theorem 5), we anticipate that there are much fewer non-strict saddle points

than strict saddle points. Understanding where exactly implicit regularization typically occurs in realistic settings is a challenging question for future works.

3.3.3 Parameterization of first-order critical points and global minimizers

We now turn back to first-order critical points, and state all new related results. In our analysis, these results precede the proof of Theorem 5. The presentation has been reversed in Section 3.3 to highlight the main contribution of the chapter.
The next proposition provides an explicit parameterization of first-order critical points. Note that this is only a necessary condition.

Proposition 5. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L associated with $\mathcal{S}$ (cf Proposition 1), and let $Q = [\![1, d_y]\!] \backslash \mathcal{S}$. Then, there exist invertible matrices $D_{H-1} \in \mathbb{R}^{d_{H-1} \times d_{H-1}}, \dots, D_1 \in \mathbb{R}^{d_1 \times d_1}$ and matrices $Z_H \in \mathbb{R}^{(d_y - r) \times (d_{H-1} - r)}$, $Z_1 \in \mathbb{R}^{(d_1 - r) \times d_x}$ and $Z_h \in \mathbb{R}^{(d_h - r) \times (d_{h-1} - r)}$ for $h \in [\![2, H-1]\!]$ such that if we denote $\widetilde{W}_H = W_H D_{H-1}$, $\widetilde{W}_1 = D_1^{-1} W_1$ and $\widetilde{W}_h = D_h^{-1} W_h D_{h-1}$, for all $h \in [\![2, H-1]\!]$, then we have

$$\widetilde{W}_H = [U_{\mathcal{S}}, U_Q Z_H] \tag{3.3.1}$$

$$\widetilde{W}_1 = \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \tag{3.3.2}$$

$$\widetilde{W}_h = \begin{bmatrix} I_r & 0 \\ 0 & Z_h \end{bmatrix} \quad \forall h \in [\![2, H-1]\!] \tag{3.3.3}$$

$$\widetilde{W}_H \cdots \widetilde{W}_2 = [U_{\mathcal{S}}, 0]\,. \tag{3.3.4}$$

The proposition is proved in Appendix 3.C.1, and will be key to prove the last statement of Theorem 5.
Next, we give a sufficient condition for any $\mathbf{W}$ satisfying (3.3.1), (3.3.2) and (3.3.3), to be a first-order critical point of L.

Proposition 6. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathcal{S} \subset [\![1, d_y]\!]$ of size $r \in [\![0, r_{max}]\!]$ and $Q = [\![1, d_y]\!] \setminus \mathcal{S}$. Let $D_{H-1} \in \mathbb{R}^{d_{H-1} \times d_{H-1}}, \dots, D_1 \in \mathbb{R}^{d_1 \times d_1}$ be invertible matrices and let $Z_H \in \mathbb{R}^{(d_y - r) \times (d_{H-1} - r)}$, $Z_1 \in \mathbb{R}^{(d_1 - r) \times d_x}$ and $Z_h \in \mathbb{R}^{(d_h - r) \times (d_{h-1} - r)}$ for $h \in [\![2, H-1]\!]$. Let the parameter of the network $\mathbf{W} = (W_H, \dots, W_1)$ be defined as follows:

$$\begin{aligned} W_H &= [U_{\mathcal{S}}, U_Q Z_H] D_{H-1}^{-1} \\ W_1 &= D_1 \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \\ W_h &= D_h \begin{bmatrix} I_r & 0 \\ 0 & Z_h \end{bmatrix} D_{h-1}^{-1} \quad \forall h \in [\![2, H-1]\!]\,. \end{aligned}$$

If $r = r_{max}$ or if there exist $h_1 \neq h_2$ such that $Z_{h_1} = 0$ and $Z_{h_2} = 0$, then, $\mathbf{W}$ is a first-order critical point of L associated with $\mathcal{S}$.

The proof of Proposition 6 is in Appendix 3.B.5.
Note that, combining Propositions 5 and 6, we obtain an explicit parameterization of all critical points $\mathbf{W}$ with a global map $W_H \cdots W_1$ of maximum rank r_{max}. In particular, it yields the next proposition, which provides an explicit parameterization of all the global minimizers of L.

Proposition 7 (Parameterization of all global minimizers). Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Set $\mathcal{S}_{max} = [\![1, r_{max}]\!]$ and $Q_{max} = [\![1, d_y]\!] \setminus \mathcal{S}_{max} = [\![r_{max}+1, d_y]\!]$. Then, $\mathbf{W} = (W_H, \dots, W_1)$ is a global minimizer of L if and only if there exist invertible matrices $D_{H-1} \in \mathbb{R}^{d_{H-1} \times d_{H-1}}, \dots, D_1 \in \mathbb{R}^{d_1 \times d_1}$, and matrices $Z_H \in \mathbb{R}^{(d_y - r_{max}) \times (d_{H-1} - r_{max})}$, $Z_h \in \mathbb{R}^{(d_h - r_{max}) \times (d_{h-1} - r_{max})}$ for $h \in [\![2, H-1]\!]$, and $Z_1 \in \mathbb{R}^{(d_1 - r_{max}) \times d_x}$ such that:

$$
\begin{aligned}
W_H &= [U_{\mathcal{S}_{max}}, U_{Q_{max}} Z_H] D_{H-1}^{-1} \\
W_1 &= D_1 \begin{bmatrix} U_{\mathcal{S}_{max}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \\
W_h &= D_h \begin{bmatrix} I_{r_{max}} & 0 \\ 0 & Z_h \end{bmatrix} D_{h-1}^{-1} \qquad \forall h \in [\![2, H-1]\!] \, .
\end{aligned}
$$

The proof is in Appendix 3.C.2. See in particular a remark in the same appendix on how to interpret the above formulas precisely (some blocks Z_h have 0 lines or columns).

3.3.4 Comparison with previous works

Next we further detail our contributions in light of earlier works.

Parameterization of global minimizers. To the best of our knowledge, Proposition 7 is the first explicit parameterization of the set of all global minimizers for deep linear networks and the square loss. For $H \geq 2$, it had been previously noted by [156] that a critical point $\mathbf{W}$ is a global minimizer if and only if $\mathrm{rk}(W_H \cdots W_1) = r_{max}$ and $\mathrm{col}(W_H \cdots W_{d_{p+1}}) = \mathrm{col}(U_{\mathcal{S}_{max}})$, where $\mathcal{S}_{max} = [\![1, r_{max}]\!]$ and where p is any layer with the smallest width r_{max}. This is an implicit characterization.

Another previous work which characterized global minimizers is [163], but their characterization is not explicit: the weight matrices are defined recursively and should satisfy some equations, while in Proposition 7 the weight matrices are given explicitly. The same remark holds for their characterization of first-order critical points.

Saddle points. Among saddle points, we give a characterization of those that are strict and those that are not.
Previously, for $H \geq 3$, it had been noted by [75] that $(0, \dots, 0)$ is a non-strict saddle point. This result also follows from Theorem 1 since any critical point is tightened whenever at least 3 weight matrices are of rank $r = \mathrm{rk}(W_H \cdots W_1)$ (which is the case for $(0, \dots, 0)$ with $r = 0$).

Also, Theorem 5 generalizes two results from [75] and [24] about sufficient conditions for strict saddle points. Indeed, [75] proved that, if $\mathbf{W}$ is a saddle point such that

$\mathrm{rk}(W_{H-1}\cdots W_2) = r_{max}$, then $\mathbf{W}$ is a strict saddle point. [24] proved under further assumptions on the data and the architecture that a sufficient condition for a saddle point to be strict is that $\mathrm{rk}(W_{H-1}\cdots W_2) > r = \mathrm{rk}(W_H\cdots W_1)$. Note that both results are special cases of Theorem 5, with the pivot $(H,1)$. More precisely, assume that $\mathbf{W}$ is a saddle point such that either $\mathrm{rk}(W_{H-1}\cdots W_2) = r_{max} = r = \mathrm{rk}(W_H\cdots W_1)$ or $\mathrm{rk}(W_{H-1}\cdots W_2) > r = \mathrm{rk}(W_H\cdots W_1)$ (which includes both conditions above). Then, if $\mathcal{S} \neq [\![1,r]\!]$ (whether $r = r_{max}$ or not), by Theorem 5, $\mathbf{W}$ is a strict saddle point without any condition on $\mathbf{W}$. But if $\mathcal{S} = [\![1,r]\!]$ with $r < r_{max}$, our assumption above implies that the pivot $(H,1)$, and therefore $\mathbf{W}$, is not tightened (recall that $\mathrm{rk}(\Sigma_{XY}) = d_y \geq r_{max} > r$). In any case, $\mathbf{W}$ is a strict saddle point.

Finally, Theorem 5 generalizes another result of [75] stating that all saddle points are strict for one-hidden layer linear networks. Indeed, let $H = 2$ and assume that we have a saddle point associated with $\mathcal{S} = [\![1,r]\!]$ for $r < r_{max}$ (the only case where we can expect to see non-strict saddle points, by Theorem 5). Since $H = 2$, there is only one pivot which is $(2,1)$; this pivot is not tightened because the complementary blocks are I_{d_1} and Σ_{XY} and both are of rank larger than or equal to r_{max}. Therefore, by Theorem 5, when $H = 2$ (and under Assumption $\mathcal{H}$), all saddle points are strict.

Convergence to global minimizer: an example where gradient descent meets no non-strict saddle points. Some recent works on deep linear networks proved under assumptions on the data, the initialization, or the minimum width of the network, that gradient descent or variants converge to a global minimum in polynomial time (e.g., [5, 11, 36, 33]). Since for general non-convex functions, gradient descent may get stuck at a non-strict saddle point, and since non-strict saddle points exist for any linear neural network of depth $H \geq 3$, it seemed impossible to deduce convergence to a global minimum using landscape results only. Instead, papers such as [33] chose to "directly analyze the trajectory generated by [...] gradient descent".

It turns out that our characterization of strict saddle points can help re-interpret such global convergence results. Consider for instance the work of [33], who proved that with high probability gradient descent with Xavier initialization converges to a global minimum for any deep linear network which is wide enough. They analyse a network where all hidden layers have a width d_{hidden} at least proportional to the number H of layers and to other quantities depending on the data X, Y, the output dimension d_y, and the desired probability level. In their analysis, [33, Section 7] prove that with high probability, a condition $\mathcal{B}(t)$ holds at every iteration t. *Importantly, this condition implies that the point $\mathbf{W}$ output by gradient descent at iteration t cannot be a non-strict saddle point.* Indeed, using our notation, the condition $\mathcal{B}(t)$ yields the lower-bound $\sigma_{\min}(W_H\cdots W_2) \geq \frac{3}{4} d_{\text{hidden}}^{(H-1)/2} > 0$, which in particular entails that the matrix product $W_H\cdots W_2$ is of full rank $\min\{d_{\text{hidden}}, d_y\} \geq r_{max}$. Let us check that if $\mathbf{W}$ is a saddle point, then it is necessarily strict. By Theorem 5, either $r = \mathrm{rk}(W_H\cdots W_1)$ is equal to r_{max}, in which case the saddle point $\mathbf{W}$ is indeed strict, or $r < r_{max}$, in which case the pivot $(H,1)$ is not tightened (since the two blocks Σ_{XY} and $W_{H-1}\cdots W_2$ are of rank at least r_{max}), so that the saddle point $\mathbf{W}$ is strict, as previously claimed.

As a consequence, our characterization of strict saddle points in Theorem 5 helps

re-interpret the analysis of [33, Section 7]: under Assumption $\mathcal{H}$, and for wide enough deep linear networks, gradient descent with Xavier initialization meets no non-strict saddle points on its trajectory.

Implicit regularization As previously explained, [85, 71, 73] and others proved that gradient-based algorithms can escape strict saddle points and be stopped at approximate second-order critical points after a number of iterations which is at most polynomial in the desired accuracy. The works [5, 11, 36, 33] mentioned above show that convergence to a global minimum can even be guaranteed under assumptions on the data, the initialization, or the minimum width of the network. Though global convergence outside these assumptions has been conjectured (e.g., [24] for gradient flow), when $H \geq 3$, we cannot yet rule out the possibility that non-stochastic gradient-based algorithms remain on a plateau around one of the r_{max} non-strict saddle points that we identified in Theorem 5 and Proposition 4. Since non-strict saddle points $\mathbf{W}$ satisfy $W_H \cdots W_1 = \arg\min_{R \in \mathbb{R}^{d_y \times d_x}, \mathrm{rk}(R) \leq r} \|RX - Y\|^2$, this can be seen as a form of long-term implicit regularization.

Furthermore, even under assumptions guaranteeing the convergence to a global minimum, gradient-based algorithms can spend some time around non-strict saddle points before convergence (as in Figures 2.3 and 2.4), which would correspond to a form of short-term implicit regularization.

We now outline some relationships between our characterization and several earlier results on implicit regularization. [7] proved that gradient flow converges to global minimizers of the rank-constrained linear regression problem. In fact, what they proved can be understood as follows: "the only critical points the gradient flow almost surely converges to are the global minimizer or non-strict saddle points. The latter lead to global minimizers of the rank-constrained linear regression problem". In Theorem 5 and Proposition 4, we prove the existence and characterize such points, and in addition to non-strict saddle points we prove that some $\mathbf{W}$ leading to the solution of the rank-constrained linear regression problem are strict saddle points. Doing so, we characterize and drastically reduce the implicit regularization set.

[6] found for small initializations and step size that for matrix recovery, deep matrix factorization favors solutions of low-rank. In the same context, [117] stated that, implicit regularization in deep linear networks should be seen as a minimization of rank rather than norms. Again, Theorem 5 and Proposition 4 specify the implicit regularization happening there.

The next two papers showcased the implicit regularization phenomenon for specific gradient flow or gradient descent trajectories. Our landscape analysis can help read their results from a new perspective. [44] proved that for $H = 2$, for a vanishing initialization and a small enough step size, the discrete gradient dynamics sequentially learns solutions of the rank-constrained linear regression problem with a gradually increasing rank. More precisely, the algorithm avoids all critical points associated with $\mathcal{S} \neq [\![1, r]\!]$, but comes close to a critical point associated with $\mathcal{S} = [\![1, r]\!]$, spends some time around it and decreases again. In the light of our work, we know that for $H = 2$ all saddle points are strict, but for $H \geq 3$, we know that there exist non-strict saddle points associated with $\mathcal{S} = [\![1, r]\!]$. If we could extend [44] to $H \geq 3$, we expect that the gradient dynamics converges to

a non-strict saddle point or spends much more time than for $H = 2$ around non-strict saddle points associated with $\mathcal{S} = [\![1, r]\!]$. In both cases these facts would show that the implicit regularization outlined by [44] for $H = 2$ intensifies with depth. In fact, this is the result presented in [45] for a toy linear network, as they proved that, for $H = 2$, the algorithms need an exponentially vanishing initialization for this incremental learning to occur, while for $H \geq 3$, a polynomially vanishing initilization is enough. This indicates that this incremental learning arises more naturally in deep networks.

Perspectives. Even if gradient-based algorithms, and in particular SGD, could escape non-strict saddle points in infinite time with probability one in all cases, can we better formalize that the vicinity of these saddle points is finite-time stable? This way, early stopping would provably be a source of short-term implicit regularization and might explain good generalization guarantees as in [21, 160, 113, 116].

A related and interesting project would be to assess the dimensions and sizes of the implicit regularization sets for the different values of r. This should be strongly related to the expected number of epochs spent by a stochastic algorithm on the corresponding plateau.

Finally, another interesting future research direction is to determine the basin of attraction of the non-strict saddle points for non-stochastic gradient-based algorithms, for various initializations and various dimensions of the deep linear networks. Do they have zero Lebesgue measure in general, so that gradient dynamics would almost surely converge to a global minimizer, as conjectured by [24] for gradient flow? or do they have positive measure for some networks and some data, hence implying a long-term implicit regularization?

3.4 Proof of Theorem 5

The proof of Theorem 5 proceeds in several steps. In the end (see page 68), it will directly follow from Propositions 8, 9, 10 below and from Lemma 5 in Appendix 3.A. In this section, we outline the overall proof structure and state the main intermediate results. We also provide proof sketches for these intermediate results, but defer many technical details to the appendix.

In our proofs, we will not compute the Hessian $\nabla^2 L(\mathbf{W})$ explicitly since this might be quite tedious. To show that a point $\mathbf{W}$ is (or is not) a second-order critical point of L, we will instead Taylor-expand $L(\mathbf{W} + t\mathbf{W}')$ along any direction $\mathbf{W}'$ and use the following lemma. Its proof follows directly from Taylor's theorem.

Lemma 1 (Characterization of first-order and second-order critical points). Let $\mathbf{W} = (W_H, \dots, W_1)$. Assume that, for all $\mathbf{W}' = (W'_H, \dots, W'_1)$, the loss $L(\mathbf{W} + t\mathbf{W}')$ admits the following asymptotic expansion when $t \to 0$:

$$L(\mathbf{W} + t\mathbf{W}') = L(\mathbf{W}) + c_1(\mathbf{W}, \mathbf{W}')t + c_2(\mathbf{W}, \mathbf{W}')t^2 + o(t^2). \tag{3.4.1}$$

Then:

— $\mathbf{W}$ is a first-order critical point of L iff $c_1(\mathbf{W}, \mathbf{W}') = 0$ for all $\mathbf{W}'$.

— $\mathbf{W}$ is a second-order critical point of L iff $c_1(\mathbf{W}, \mathbf{W}') = 0$ and $c_2(\mathbf{W}, \mathbf{W}') \geq 0$ for all $\mathbf{W}'$.
Therefore if for a first-order critical point $\mathbf{W}$, we can exhibit a direction $\mathbf{W}'$ such that $c_2(\mathbf{W}, \mathbf{W}') < 0$, then $\mathbf{W}$ is not a second-order critical point.

We divide the proof of Theorem 5 into three parts. Recall that from [75], we know that all first-order critical points are either global minimizers or saddle points (that is, there is no local extrema apart from global minimizers). We refine this classification.

3.4.1 Global minimizers and 'simple' strict saddle points

In this section, we start by identifying simple sufficient conditions on the support $\mathcal{S}$ associated to a first-order critical point $\mathbf{W}$ which guarantee that $\mathbf{W}$ is either a global minimizer or a strict saddle point. More subtle strict saddle points and non-strict saddle points will be addressed in Sections 3.4.2 and 3.4.3.

Proposition 8. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L associated with $\mathcal{S}$ and set $r = \mathrm{rk}(W_H \cdots W_1) \leq r_{max}$.

— When $r = r_{max}$:

  — if $\mathcal{S} = [\![1, r_{max}]\!]$, then $\mathbf{W}$ is a global minimizer.

  — if $\mathcal{S} \neq [\![1, r_{max}]\!]$, then $\mathbf{W}$ is not a second-order critical point ($\mathbf{W}$ is a strict saddle point).

— When $r < r_{max}$: $\mathbf{W}$ is a saddle point.

  — if $\mathcal{S} \neq [\![1, r]\!]$, then $\mathbf{W}$ is not a second-order critical point ($\mathbf{W}$ is a strict saddle point).

The proof is postponed to Appendix 3.D. To prove that $\mathbf{W}$ associated with $\mathcal{S} \neq [\![1, r]\!]$, $r \leq r_{max}$ is not a second-order critical point, we explicitly exhibit a direction $\mathbf{W}'$ such that the second-order coefficient $c_2(\mathbf{W}, \mathbf{W}')$ in the Taylor expansion of $L(\mathbf{W} + t\mathbf{W}')$ around $t = 0$, in (3.4.1), is negative. Using Lemma 1, we conclude that $\mathbf{W}$ is not a second-order critical point.
Recall from Proposition 1 that the loss at any first-order critical point is given by $tr(\Sigma_{YY}) - \sum_{i \in \mathcal{S}} \lambda_i$. The spirit of the proof is that critical points associated with $\mathcal{S} \neq [\![1, r]\!]$ capture a smaller singular value λ_j instead of a larger one λ_i with $i < j$. Thus, to see that the loss can be further decreased at order 2 (and is therefore not a second-order critical point by Lemma 1), a natural proof strategy is to perturb the singular vector corresponding to λ_j along the direction of the singular vector corresponding to λ_i. This part of the proof is an adaption of the proof of [8].

3.4.2 Strict saddle points associated with $\mathcal{S} = [\![1, r]\!],\ r < r_{max}$

We now address situations that to our knowledge, have never been addressed, in the literature. We prove the following.

Proposition 9. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L associated with $\mathcal{S} = [\![1, r]\!]$, with $0 \leq r < r_{max}$.
If $\mathbf{W}$ is not tightened, then $\mathbf{W}$ is not a second-order critical point ($\mathbf{W}$ is a strict saddle point).

We sketch the main arguments below. We will again construct a direction $\mathbf{W}'$ such that the second-order coefficient $c_2(\mathbf{W}, \mathbf{W}')$ in the asymptotic expansion of $L(\mathbf{W} + t\mathbf{W}')$ around $t = 0$, in (3.4.1), is negative.
More precisely, for a first-order critical point $\mathbf{W}$, for any $\beta \in \mathbb{R}$, we will consider a well-chosen $\mathbf{W}'_\beta$ such that $c_2(\mathbf{W}, \mathbf{W}'_\beta) = a\beta^2 + c\beta$ for some constants a, c (possibly depending on $\mathbf{W}$) such that $a \geq 0$ and $c \neq 0$. Taking

$$\beta = \begin{cases} -c & \text{if} \quad a = 0 \\ -\frac{c}{2a} & \text{if} \quad a > 0 \end{cases} \tag{3.4.2}$$

we obtain

$$c_2(\mathbf{W}, \mathbf{W}'_\beta) = \begin{cases} -c^2 & \text{if} \quad a = 0 \\ -\frac{c^2}{4a} & \text{if} \quad a > 0 \end{cases}$$

and therefore

$$c_2(\mathbf{W}, \mathbf{W}'_\beta) < 0.$$

Using Lemma 1, we can conclude that $\mathbf{W}$ is not a second-order critical point.
We now provide intuitions on how to choose $\mathbf{W}'$. Since $\mathbf{W}$ is not tightened, there exists a pivot (i, j), with $i > j$, which is not tightened. Depending on the values of i and j we will construct $\mathbf{W}'$ differently. However, the strategy for constructing $\mathbf{W}'$ is the same in all cases.
Recall again that from Proposition 1, at any first-order critical point $\mathbf{W}$, the value of the loss is given by $tr(\Sigma_{YY}) - \sum_{i \in \mathcal{S}} \lambda_i$. Contrary to the previous section, since $\mathcal{S} = [\![1, r]\!]$ there is no immediate way to decrease the loss (at order 2) without increasing the rank of the product of the weight matrices. Indeed, we have $W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \in \arg\min_{\mathrm{rk}(R) \leq r} \|RX - Y\|^2$.
Therefore, to be able to decrease the value of the loss, we need to perturb $\mathbf{W}$ in a way that the product of the perturbed parameter weight matrices becomes of rank strictly larger than r. Also, to prove that $\mathbf{W}$ is not a second-order critical point, we need to decrease the loss at order 2. This is possible when $\mathbf{W}$ is not tightened. For the non-tightened pivot (i, j), we choose a perturbation $\mathbf{W}'$ with all $W'_h = 0$ except for W'_i and W'_j. Furthermore, our construction of W'_i and W'_j depends on whether i and/or j are on the boundary $\{1, H\}$. This is due to the fact that H and 1 play a special role in the product of the perturbed weights $(W_H + tW'_H) \cdots (W_1 + tW'_1)$. This is why we distinguish the four cases below:

— 1st case: $i \in [\![2, H-1]\!]$ and $j = 1$. This case is treated in Appendix 3.E.1.

— 2nd case: $i = H$ and $j = 1$. This case is treated in Appendix 3.E.2.

— 3rd case: $i = H$ and $j \in [\![2, H-1]\!]$. This case is treated in Appendix 3.E.3.

— 4th case: $i, j \in [\![2, H-1]\!]$ with $i > j$. This case is treated in Appendix 3.E.4.

3.4.3 Non-strict saddle points

We now provide a sketch of the proof for the converse of Proposition 9, as stated in Proposition 10 below. All the proofs related to this section are deferred to Appendix 3.F.

Proposition 10. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L associated with $\mathcal{S} = [\![1, r]\!]$, $0 \leq r < r_{max}$.
If $\mathbf{W}$ is tightened, then $\mathbf{W}$ is a second-order critical point ($\mathbf{W}$ is a non-strict saddle point).

To prove Proposition 10, we first state a proposition which indicates that multiplications by invertible matrices do not change the nature of the critical point.

Lemma 2. For all $h \in [\![1, H-1]\!]$, let $D_h \in \mathbb{R}^{d_h \times d_h}$ be an invertible matrix. We define $\widetilde{W}_H = W_H D_{H-1}$, $\widetilde{W}_1 = D_1^{-1} W_1$ and $\widetilde{W}_h = D_h^{-1} W_h D_{h-1}$, for all $h \in [\![2, H-1]\!]$. Then

— $\mathbf{W} = (W_H, \dots, W_1)$ is a first-order critical point of L if and only if $\widetilde{\mathbf{W}} = (\widetilde{W}_H, \dots, \widetilde{W}_1)$ is a first-order critical point of L.

— $\mathbf{W} = (W_H, \dots, W_1)$ is a second-order critical point of L if and only if $\widetilde{\mathbf{W}} = (\widetilde{W}_H, \dots, \widetilde{W}_1)$ is a second-order critical point of L.

The lemma is proved in Appendix 3.B.4.

Proposition 10 is then obtained using Proposition 5 (note that when $\mathbf{W}$ is tightened, $\widetilde{\mathbf{W}}$ is also tightened since the rank of a matrix does not change when multiplied by invertible matrices), by showing that $\widetilde{\mathbf{W}} = (\widetilde{W}_H, \dots, \widetilde{W}_1)$ as given by Proposition 5 is a second-order critical point of L and using Lemma 2 to conclude that $\mathbf{W}$ is a second-order critical point. This is easier since $\widetilde{\mathbf{W}}$ has a simpler form.

More precisely, we have the following result, from which Proposition 10 follows (see Appendix 3.F.2 for details).

Proposition 11. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L associated with $\mathcal{S} = [\![1, r]\!]$ with $0 \leq r < r_{max}$ such that there exist matrices $Z_H \in \mathbb{R}^{(d_y - r) \times (d_{H-1} - r)}$, $Z_1 \in \mathbb{R}^{(d_1 - r) \times d_x}$ and $Z_h \in \mathbb{R}^{(d_h - r) \times (d_{h-1} - r)}$ for $h \in [\![2, H-1]\!]$ with

$$W_H = [U_{\mathcal{S}}, U_Q Z_H] \tag{3.4.3}$$

$$W_1 = \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \tag{3.4.4}$$

$$W_h = \begin{bmatrix} I_r & 0 \\ 0 & Z_h \end{bmatrix} \quad \forall h \in [\![2, H-1]\!] \tag{3.4.5}$$

$$W_H \cdots W_2 = [U_{\mathcal{S}}, 0]\,, \tag{3.4.6}$$

where $Q = [\![1, d_y]\!] \setminus \mathcal{S}$.
If $\mathbf{W}$ is tightened, then $\mathbf{W}$ is a second-order critical point of L.

Proposition 11 is proved in details in Section 3.F.1. We provide a proof sketch below.
We denote, for t in the neighborhood of 0, and $h \in [\![1, H]\!]$, $W_h(t) = W_h + tW_h'$ where $W_h' \in \mathbb{R}^{d_h \times d_{h-1}}$ is arbitrary.
We define $\mathbf{W}(t) := (W_H(t), \dots, W_1(t))$ and $W(t) := W_H(t) \cdots W_1(t)$. As in the previous

two sections, we use Lemma 1. However, this time, we show that the second-order coefficient $c_2(\mathbf{W}, \mathbf{W}')$ is non-negative for all directions $\mathbf{W}'$.
To compute the loss $\|W(t)X - Y\|^2$, we expand

$$\begin{aligned} W(t) &= W_H(t) \cdots W_1(t) \\ &= (W_H + tW'_H) \cdots (W_1 + tW'_1) \\ &= W_H \cdots W_1 + t \sum_{i=1}^{H} W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_1 \\ &\quad + t^2 \sum_{H \geq i > j \geq 1} W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_{j+1} W'_j W_{j-1} \cdots W_1 + o(t^2) \ . \end{aligned}$$

Therefore,

$$\begin{aligned} L(\mathbf{W}(t)) = \Bigg\| & W_H \cdots W_1 X - Y + t \sum_{i=1}^{H} W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_1 X \\ & + t^2 \sum_{H \geq i > j \geq 1} W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_{j+1} W'_j W_{j-1} \cdots W_1 X + o(t^2) \Bigg\|^2 \ . \end{aligned}$$

We can now easily calculate the second-order coefficient $c_2(\mathbf{W}, \mathbf{W}')$ in the Taylor expansion of $L(\mathbf{W}(t))$ around $t = 0$ (in (3.4.1)).
Recalling that $c_2(\mathbf{W}, \mathbf{W}')$ is such that $L(\mathbf{W}(t)) = L(\mathbf{W}) + c_2(\mathbf{W}, \mathbf{W}')t^2 + o(t^2)$ (since $\mathbf{W}$ is a first-order critical point), we have

$$\begin{aligned} c_2(\mathbf{W}, \mathbf{W}') = & \left\| \sum_{i=1}^{H} W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_1 X \right\|^2 \\ & + 2 \left\langle \sum_{H \geq i > j \geq 1} W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_{j+1} W'_j W_{j-1} \cdots W_1 X \ , \ W_H \cdots W_1 X - Y \right\rangle \ , \end{aligned}$$

where $\langle A, B \rangle = \mathrm{tr}(AB^T)$. In order to simplify the notation and equations, we define, for all $i \in [\![1, H]\!]$,

$$T_i = W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_1 X \ , \tag{3.4.7}$$

and for all $i, j \in [\![1, H]\!]$ with $i > j$:

$$T_{i,j} = \left\langle W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_{j+1} W'_j W_{j-1} \cdots W_1 X \ , \ W_H \cdots W_1 X - Y \right\rangle \ . \tag{3.4.8}$$

Then we set

$$FT = \left\| \sum_{i=1}^{H} T_i \right\|^2 , \tag{3.4.9}$$

and

$$ST = 2 \sum_{H \geq i > j \geq 1} T_{i,j} . \tag{3.4.10}$$

The coefficient becomes

$$c_2(\mathbf{W}, \mathbf{W}') = \left\| \sum_{i=1}^{H} T_i \right\|^2 + 2 \sum_{H \geq i > j \geq 1} T_{i,j} = FT + ST .$$

Using the fact that $\mathbf{W}$ is tightened, some weight products become simple (see Lemma 14) and we can simplify T_i and $T_{i,j}$ (see Lemmas 21 and 22 in Appendix 3.F).
This allows us to establish that, for any $\mathbf{W}'$, there exist matrices A_2, A_3, A_4 and a non-negative scalar a_1 such that $FT = a_1 + \|A_2\|^2 + \|A_3\|^2 + \|A_4\|^2$ (see Appendix 3.F.1.2) and $ST = -2 \langle A_3, A_4 \rangle$ (see Appendix 3.F.1.3). Therefore

$$c_2(\mathbf{W}, \mathbf{W}') = FT + ST = a_1 + \|A_2\|^2 + \|A_3 - A_4\|^2 \geq 0 ,$$

and using Lemma 1 we conclude that $\mathbf{W}$ is a second-order critical point.

We are now in a position to prove Theorem 5 as a direct corollary from the above results.

Proof of Theorem 5. The classification into global minimizers, strict saddle points, and non-strict saddle points follows directly from Propositions 8, 9, and 10 above. As for the fact that

$$W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \in \underset{R \in \mathbb{R}^{d_y \times d_x}, \mathrm{rk}(R) \leq r}{\arg\min} \|RX - Y\|^2$$

when $\mathcal{S} = [\![1, r]\!]$, it follows from Proposition 1 above and from Lemma 5 in Appendix 3.A. □

3.5 Conclusion

We studied the optimization landscape of linear neural networks of arbitrary depth with the square loss. We first derived a necessary condition for being a first-order critical point by associating any of them with a set of eigenvectors of a data-dependent matrix. We then provided a complete characterization of the landscape at order 2 by distinguishing between global minimizers, strict saddle points, and non-strict saddle points. As a by-product of this analysis, we exhibited large sets of strict and non-strict saddle points and derived an explicit parameterization of all global minimizers. Our second-order characterization also sheds

some light on the implicit regularization that may be induced by first-order algorithms, by proving that non-strict saddle points and some strict saddle points are among the global minimizers of the rank-constrained linear regression problem. It also helps re-interpret a recent convergence result, stating that gradient descent with Xavier initialization converges to a global minimum for any wide enough deep linear network.

Appendix

3.A Notation and useful properties

In this section, we define some additional notation and terminology that will be used through all subsequent appendices. We also state simple linear algebra facts (Section 3.A.2), together with some properties about the Moore-Penrose inverse (Section 3.A.3). Since most of the proofs rely on linear algebra, we recommend the unfamiliar reader to check classical textbooks.

Additional notation: If a matrix A has already a subscript like W_H for example, we denote by $(W_H)_{.,i}$ the i-th column and by $(W_H)_{.,J}$ the sub-matrix obtained by concatenating the column vectors $(W_H)_{.,i}$, for all $i \in \mathcal{J}$. Also $(W_H)_{i,.}$ denotes the i-th row of W_H and $(W_H)_{\mathcal{I},.}$ the sub-matrix obtained by concatenating the line vectors $(W_H)_{i,.}$, for all $i \in \mathcal{I}$. More generally $(W_H)_{\mathcal{I},\mathcal{J}}$ denotes the matrix W_H restricted to the index set $\mathcal{I} \times \mathcal{J}$. For instance, $(W_H)_{1:r,r+1:d_{H-1}} \in \mathbb{R}^{r \times (d_{H-1}-r)}$ is the matrix formed from W_H by keeping the rows from 1 to r and the columns from $r+1$ to d_{H-1}. The symbol $\delta_{i,j}$ denotes the Kronecker index which equal to 0 if $i \neq j$ and 1 if $i = j$.

Also, we define the partial gradients with respect to each weight matrix as follows.

3.A.1 Partial gradients

Definition 4 (gradient and partial gradients of L). Since the input $\mathbf{W} = (W_H, \dots, W_1)$ of $L(\mathbf{W})$ is not a vector but a sequence of matrices, we define the gradient $\nabla L(\mathbf{W})$ of L at $\mathbf{W}$ with a similar format :

$$\nabla L(\mathbf{W}) = (\nabla_{W_H} L(\mathbf{W}), \dots, \nabla_{W_1} L(\mathbf{W})) \ ,$$

where each partial gradient $\nabla_{W_h} L(\mathbf{W}) \in \mathbb{R}^{d_h \times d_{h-1}}$ is the matrix whose entries are the partial derivatives $\frac{\partial L}{\partial (W_h)_{i,j}}$ for $i = 1, .., d_h$ and $j = 1, .., d_{h-1}$

The next lemma provides explicit formulas for the partial gradients of L. A proof can be found at the end of [156].

Lemma 3. Let $h \in [\![2, H-1]\!]$. The partial gradient of L with respect to W_h is:

$$\nabla_{W_h} L(\mathbf{W}) = 2(W_H \cdots W_{h+1})^T (W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX})(W_{h-1} \cdots W_1)^T \ .$$

We also have the partial gradient with respect to W_H:

$$\nabla_{W_H} L(\mathbf{W}) = 2(W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX})(W_{H-1} \cdots W_1)^T \ .$$

Finally, the partial gradient with respect to W_1 is:

$$\nabla_{W_1} L(\mathbf{W}) = 2(W_H \cdots W_2)^T (W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX}) \ .$$

3.A.2 Simple linear algebra facts

Recall that $\Sigma^{1/2} = \Sigma_{YX}\Sigma_{XX}^{-1}X$ and $\Sigma = \Sigma^{1/2}(\Sigma^{1/2})^T = \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$. Recall also from (3.2.1) that $\Sigma^{1/2} = U\Delta V^T$ is a Singular Value Decomposition, where $U \in \mathbb{R}^{d_y \times d_y}$ and $V \in \mathbb{R}^{m\times m}$ are orthogonal matrices.

Lemma 4. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Then Σ is invertible.

Proof. Given the definition of $\Sigma^{1/2}$, it is a standard fact of linear algebra that $\mathrm{rk}(\Sigma^{1/2}) = \mathrm{rk}(\Sigma_{YX}\Sigma_{XX}^{-1}X) \leq \mathrm{rk}(\Sigma_{YX})$. On the other hand, $\mathrm{rk}(\Sigma^{1/2}) = \mathrm{rk}(\Sigma_{YX}\Sigma_{XX}^{-1}X) \geq \mathrm{rk}(\Sigma_{YX}\Sigma_{XX}^{-1}XX^T) = \mathrm{rk}(\Sigma_{YX})$ since $\Sigma_{XX} = XX^T$. Therefore $\mathrm{rk}(\Sigma^{1/2}) = \mathrm{rk}(\Sigma_{YX}) = d_y$ by Assumption $\mathcal{H}$. Finally, using another fact of linear algebra we have $\mathrm{rk}(\Sigma) = \mathrm{rk}(\Sigma^{1/2}(\Sigma^{1/2})^T) = \mathrm{rk}(\Sigma^{1/2})$, and therefore $\mathrm{rk}(\Sigma) = d_y$. Hence, Σ is invertible. □

The next lemma is about global minimizers of the rank-constrained linear regression problem.

Lemma 5. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathcal{S} = [\![1, r]\!]$. We have

$$U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1} \in \underset{R\in\mathbb{R}^{d_y\times d_x},\mathrm{rk}(R)\leq r}{\arg\min} \|RX - Y\|^2.$$

Proof. A proof of can be found in [156]. □

We now present a lemma with elementary properties that we will use frequently and that are related to the orthogonality of U. The proof is straightforward.

Lemma 6. We have the following properties related to the orthogonality of the matrix U:

- We have $I_{d_y} = UU^T = U^TU$.
- For any $i, j \in [\![1, d_y]\!]$, we have $U_i^TU_j = \delta_{i,j}$.
- For any $I, J \subset [\![1, d_y]\!]$ such that $I \cap J = \emptyset$, we have $U_I^TU_J = 0_{|I|\times|J|}$.
- For any $I, J \subset [\![1, d_y]\!]$ such that $I \cap J = \emptyset$ and $I \cup J = [\![1, d_y]\!]$, we have $I_{d_y} = U_IU_I^T + U_JU_J^T$.
- For any $J \subset [\![1, d_y]\!]$, we have $U_J^TU_J = I_{|J|}$ and $\mathrm{rk}(U_JU_J^T) = |J|$.

Note that the same applies also to the other orthogonal matrix $V \in \mathbb{R}^{m\times m}$ appearing in the Singular Value Decomposition of $\Sigma^{1/2}$ (we only replace d_y by m).

Another useful lemma is the following:

Lemma 7. For any $I, J \subset [\![1, d_y]\!]$ such that $I \cap J = \emptyset$, we have

$$U_I^T\Sigma U_J = 0_{|I|\times|J|}.$$

In particular, for any $\mathcal{S} \subset [\![1, d_y]\!]$ and $Q = [\![1, d_y]\!] \setminus \mathcal{S}$, we have $U_{\mathcal{S}}^T\Sigma U_Q = 0$.

Proof. We have, for any $k \in [\![1, d_y]\!]$, $\Sigma U_k = \lambda_k U_k$. Hence for $j \neq k$ we have $U_j^T\Sigma U_k = \lambda_k U_j^TU_k = 0$ since U is orthogonal. Therefore, if we take two disjoint sets $J = \{j_1, \ldots, j_p\}, K = \{k_1, \ldots, k_n\} \subset [\![1, d_y]\!]$, the coefficient in the position (l, m) of the matrix $U_J^T\Sigma U_K$ is equal to $U_{j_l}\Sigma U_{k_m}$ which is zero, since $j_l \neq k_m$. Therefore, $U_J^T\Sigma U_K = 0$. In particular, $U_{\mathcal{S}}^T\Sigma U_Q = 0$. □

3.A.3 The Moore-Penrose inverse and its properties

The Moore-Penrose inverse is the most known and used generalized inverse [2]. It is defined as follows:
For $A \in \mathbb{R}^{m\times n}$, the pseudo-inverse of A is defined as the matrix $A^+ \in \mathbb{R}^{n\times m}$ which satisfies the 4 following criteria known as the Moore-Penrose conditions:

1. $AA^+A = A$.
2. $A^+AA^+ = A^+$.
3. $(AA^+)^T = AA^+$.
4. $(A^+A)^T = A^+A$.

A^+ exists for any matrix A and is unique. We also have the following properties:

(i) $A^+ = \left(A^TA\right)^+ A^T$.

(ii) $\mathrm{rk}(A) = \mathrm{rk}(A^+) = \mathrm{rk}(AA^+) = \mathrm{rk}(A^+A)$.

(iii) If the linear system $Ax = b$ has any solutions, they are all given by

$$x = A^+b + (I - A^+A)w$$

for arbitrary vector w. This is equivalent to

$$x = A^+b + u$$

for arbitrary $u \in \mathrm{Ker}(A)$.

(iv) $P_A := AA^+$ is the orthogonal projection onto the range of A, and is therefore symmetric ($P_A^T = P_A$) (follows from 3) and idempotent ($P_A^2 = P_A$) (follows from 1).

(v) $I_n - A^+A$ is the orthogonal projector onto the kernel of A.

3.B Propositions and lemmas for first-order critical points

In this section, we prove all lemmas about first-order critical points. We start by stating some preliminary results.

3.B.1 Preliminaries

The following lemma gives a necessary condition for $\mathbf{W}$ to be a first-order critical point. It also provides the global map of the network, defined by $W_H \cdots W_1$. Finally, it states that the projection matrix P_K and Σ commute, where $K = W_H \cdots W_2$. This is key in the rest of the analysis.

Lemma 8. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point of L. We define $K = W_H \cdots W_2$ and $W = W_HW_{H-1}\cdots W_1 = KW_1$. Then, we have

$$W_1 = K^+\Sigma_{YX}\Sigma_{XX}^{-1} + M \ ,$$

2. en.wikipedia.org/wiki/Moore-Penrose_inverse

where $M \in \mathbb{R}^{d_1 \times d_x}$ is such that $KM = 0$ and K^+ is the Moore-Penrose inverse of K (see Appendix 3.A.3). As a consequence,

$$\begin{cases} W = P_K \Sigma_{YX} \Sigma_{XX}^{-1} \\ \mathrm{rk}(W) = \mathrm{rk}(P_K) = \mathrm{rk}(K) \end{cases}$$

where we recall that $P_K = KK^+ \in \mathbb{R}^{d_y \times d_y}$ is the matrix of the orthogonal projection onto the range of K. Finally,

$$\Sigma P_K = P_K \Sigma \, .$$

Note that $\Sigma_{YX}\Sigma_{XX}^{-1}$ is the global minimizer of the problem with one layer (i.e the classical linear regression problem). Therefore, the global map $W_H \cdots W_1$ of any first-order critical point of L is equal to the global minimizer of the linear regression projected onto the column space of K.

Proof. Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point of L. In particular, the partial gradients of L with respect to W_1 and W_H are equal to zero at $\mathbf{W}$. Using Lemma 3, this implies

$$\begin{cases} (W_H \cdots W_2)^T W_H \cdots W_1 \Sigma_{XX} = (W_H \cdots W_2)^T \Sigma_{YX} \\ W_H \cdots W_1 \Sigma_{XX} (W_{H-1} \cdots W_1)^T = \Sigma_{YX} (W_{H-1} \cdots W_1)^T \, . \end{cases}$$

We substitute in these equations $K = W_H W_{H-1} \cdots W_2$ and $W = W_H W_{H-1} \cdots W_1 = KW_1$. Using that Σ_{XX} is invertible, and multiplying the second equation on the right by W_H^T, we obtain that any critical point of L satisfies

$$\begin{cases} K^T K W_1 = K^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ W \Sigma_{XX} W^T = \Sigma_{YX} W^T \, . \end{cases} \tag{3.B.1}$$

The first equation implies $W_1 = (K^T K)^+ K^T \Sigma_{YX} \Sigma_{XX}^{-1} + M$, where $M \in \mathbb{R}^{d_1 \times d_x}$ is such that $K^T K M = 0$ (see Property (iii) in the reminder on Moore-Penrose inverse in Appendix 3.A.3).
We have $(K^T K)^+ K^T = K^+$(see Property (i) in Appendix 3.A.3) and a standard fact of linear algebra is that $\mathrm{Ker}(K^T K) = \mathrm{Ker}(K)$.
Therefore, using these properties, we obtain $W_1 = K^+ \Sigma_{YX} \Sigma_{XX}^{-1} + M$, where $KM = 0$. This proves the first statement of the lemma. We then have,

$$W = KW_1 = KK^+ \Sigma_{YX} \Sigma_{XX}^{-1} + KM = P_K \Sigma_{YX} \Sigma_{XX}^{-1} \, . \tag{3.B.2}$$

where $P_K = KK^+$ is the orthogonal projection matrix onto the column space of K (see Appendix 3.A.3). Using Assumption $\mathcal{H}$, we have that $\Sigma_{YX}\Sigma_{XX}^{-1}$ is of full row rank, hence

$$\mathrm{rk}(W) = \mathrm{rk}(P_K \Sigma_{YX} \Sigma_{XX}^{-1}) = \mathrm{rk}(P_K) = \mathrm{rk}(K) \, , \tag{3.B.3}$$

where the last equality comes from the property (ii) in Section 3.A.3. Therefore, (3.B.1) and (3.B.3) prove the second statement of the lemma.

To prove that $\Sigma P_K = P_K \Sigma$, we remark that, using the second equation in (3.B.1), $\Sigma_{YX} W^T = W \Sigma_{XX} W^T$ and since $W \Sigma_{XX} W^T$ is symmetric and $(\Sigma_{YX})^T = \Sigma_{XY}$, we have

$$\Sigma_{YX} W^T = W \Sigma_{XY} \, .$$

Substituting the expression of W from (3.B.2), and since P_K and Σ_{XX}^{-1} are symmetric, we have

$$\Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} P_K = P_K \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} \, .$$

Using the definition of Σ, this can be rewritten as

$$\Sigma P_K = P_K \Sigma \, ,$$

which concludes the proof. □

Lemma 9. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L. We set $K = W_H \cdots W_2$ and $r = \mathrm{rk}(W_H \cdots W_1)$.
There exists a unique subset $\mathcal{S} \subset [\![1, d_y]\!]$ of size r such that:

$$P_K = U \mathcal{I}^{\mathcal{S}} U^T = U_{\mathcal{S}} U_{\mathcal{S}}^T ,$$

where $\mathcal{I}^{\mathcal{S}} \in \mathbb{R}^{d_y \times d_y}$ is the diagonal matrix such that, for all $i \in [\![1, d_y]\!]$, $(\mathcal{I}^{\mathcal{S}})_{i,i} = 1$ if $i \in \mathcal{S}$ and 0 otherwise.

Proof. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L. Using Lemma 8, we have $\Sigma P_K = P_K \Sigma$. Substituting the diagonalization of Σ from Section 3.2, this becomes $U \Lambda U^T P_K = P_K U \Lambda U^T$. Since U is orthogonal, multiplying by U^T on the left and by U on the right we obtain $\Lambda U^T P_K U = U^T P_K U \Lambda$. Hence, $U^T P_K U$ commutes with a diagonal matrix whose diagonal elements are all distinct. Therefore, $\Gamma := U^T P_K U$ is diagonal, and $P_K = U \Gamma U^T$ is a diagonalization of P_K. From Lemma 8, we also have $r = \mathrm{rk}(P_K)$. But, we know that $P_K = K K^+ \in \mathbb{R}^{d_y \times d_y}$ is the matrix of an orthogonal projection. Therefore, its eigenvalues are 1 with multiplicity r and 0 with multiplicity $d_y - r$.

Therefore, there exists an index set $\mathcal{S} \subset [\![1, d_y]\!]$ of size r such that $\Gamma = \mathcal{I}^{\mathcal{S}}$ where $\mathcal{I}^{\mathcal{S}} \in \mathbb{R}^{d_y \times d_y}$ is the diagonal matrix such that, for all $i \in [\![1, d_y]\!]$, $(\mathcal{I}^{\mathcal{S}})_{i,i} = 1$ if $i \in \mathcal{S}$ and 0 otherwise.
Therefore,

$$P_K = U \mathcal{I}^{\mathcal{S}} U^T = U \mathcal{I}^{\mathcal{S}} \mathcal{I}^{\mathcal{S}} U^T = U_{\mathcal{S}} U_{\mathcal{S}}^T .$$

If there exist $\mathcal{S}'$ such that $\Gamma = \mathcal{I}^{\mathcal{S}'}$, we get $P_K = U \mathcal{I}^{\mathcal{S}} U^T = U \mathcal{I}^{\mathcal{S}'} U^T$ which implies $\mathcal{I}^{\mathcal{S}} = \mathcal{I}^{\mathcal{S}'}$, hence $\mathcal{S} = \mathcal{S}'$. Therefore, $\mathcal{S}$ is unique. □

3.B.2 Proof of Proposition 1

In this proof, we use Lemmas 8 and 9 stated and proved in the previous section.
Recall that $\lambda_1 > \cdots > \lambda_{d_y}$ are the eigenvalues of $\Sigma = \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} \in \mathbb{R}^{d_y \times d_y}$.

Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L. We set $K = W_H \cdots W_2$, $r = \mathrm{rk}(W_H \cdots W_1)$. Using Lemma 9, there exists a unique subset $\mathcal{S} \subset [\![1, d_y]\!]$ of size r such that:

$$P_K = U_{\mathcal{S}} U_{\mathcal{S}}^T.$$

Therefore, using Lemma 8,

$$W_H \cdots W_1 = P_K \Sigma_{YX} \Sigma_{XX}^{-1} = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}.$$

This proves the first statement of Proposition 1.
To prove the second statement, notice that we have

$$\begin{aligned}
L(\mathbf{W}) &= \|WX - Y\|^2 \\
&= \|WX\|^2 - 2\langle WX\ ,\ Y\rangle + \|Y\|^2 \\
&= \mathrm{tr}(W\Sigma_{XX}W^T) - 2\,\mathrm{tr}(W\Sigma_{XY}) + \mathrm{tr}(\Sigma_{YY}) \\
&= \mathrm{tr}(U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XX}\Sigma_{XX}^{-1}\Sigma_{XY}U_{\mathcal{S}}U_{\mathcal{S}}^T) - 2\,\mathrm{tr}(U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}) + \mathrm{tr}(\Sigma_{YY}) \\
&= \mathrm{tr}(U_{\mathcal{S}}U_{\mathcal{S}}^TU_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma) - 2\,\mathrm{tr}(U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma) + \mathrm{tr}(\Sigma_{YY})
\end{aligned}$$

Since $U_{\mathcal{S}}^T U_{\mathcal{S}} = I_r$ (see Lemma 6), using Lemma 9 and the fact that U diagonalizes Σ, this becomes

$$\begin{aligned}
L(\mathbf{W}) &= \mathrm{tr}(\Sigma_{YY}) - \mathrm{tr}(U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma) \\
&= \mathrm{tr}(\Sigma_{YY}) - \mathrm{tr}(U\mathcal{I}^{\mathcal{S}}U^TU\Lambda U^T) \\
&= \mathrm{tr}(\Sigma_{YY}) - \mathrm{tr}(\mathcal{I}^{\mathcal{S}}U^TU\Lambda U^TU) \\
&= \mathrm{tr}(\Sigma_{YY}) - \mathrm{tr}(\mathcal{I}^{\mathcal{S}}\Lambda) \\
&= \mathrm{tr}(\Sigma_{YY}) - \sum_{i\in\mathcal{S}} \lambda_i\ .
\end{aligned}$$

This proves the second and last statement of Proposition 1.

3.B.3 Lemma 10

In this section we state and prove a lemma about first-order critical points which will be useful in various proofs. This lemma gives a simpler form for $K = W_H \cdots W_2$ and W_1.

Lemma 10. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L associated with $\mathcal{S}$. We set $r = \mathrm{rk}(W_H \cdots W_1)$.
Then there exists an invertible matrix $D \in \mathbb{R}^{d_1 \times d_1}$, a matrix $M \in \mathbb{R}^{d_1 \times d_x}$ satisfying $W_H \cdots W_2 M = 0$, such that:

$$K = W_H \cdots W_2 = \begin{bmatrix} U_{\mathcal{S}} & 0_{d_y \times (d_1 - r)} \end{bmatrix} D$$

and

$$W_1 = D^{-1} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0_{(d_1-r)\times d_x} \end{bmatrix} + M \ .$$

Note that the result is still true when $r = 0$, provided that $U_\emptyset \in \mathbb{R}^{d_y \times 0}$.

To prove Lemma 10, we use Lemmas 8 and 9 stated and proved in the preliminaries of Appendix 3.B.1. We will also need the following lemma

Lemma 11. Let n be a positive integer and $\emptyset \neq \mathcal{S} \subset [\![1, d_y]\!]$ such that $n \geq r := |\mathcal{S}|$. Let $A \in \mathbb{R}^{d_y \times n}$ such that $AA^+ = U_{\mathcal{S}} U_{\mathcal{S}}^T$. Then there exists an invertible matrix $D \in \mathbb{R}^{n \times n}$ such that

$$A = [U_{\mathcal{S}} \quad 0_{d_y \times (n-r)}] D$$

and

$$A^+ = D^{-1} \begin{bmatrix} U_{\mathcal{S}}^T \\ 0_{(n-r)\times d_y} \end{bmatrix} \ .$$

Proof of Lemma 11. The matrix $I_n - A^+A$ is the orthogonal projection onto Ker(A) (see Appendix 3.A.3), hence

$$\mathrm{rk}(I_n - A^+A) = \dim \mathrm{Ker}(A) = n - \mathrm{rk}(A)$$

But we have (see Property (ii) in Appendix 3.A.3) $\mathrm{rk}(A^+A) = \mathrm{rk}(A) = \mathrm{rk}(AA^+)$ and, using Lemma 6, $\mathrm{rk}(A^+A) = \mathrm{rk}(U_{\mathcal{S}} U_{\mathcal{S}}^T) = r$. Therefore, $\mathrm{rk}(A) = r$ and

$$\mathrm{rk}(I_n - A^+A) = n - r.$$

Let $B \in \mathbb{R}^{n \times (n-r)}$ and $C \in \mathbb{R}^{(n-r)\times n}$ be such that $I_n - A^+A = BC$ (such matrices can be obtained by considering the Singular Value Decomposition of $I_n - A^+A$). Denoting $D = \begin{bmatrix} U_{\mathcal{S}}^T A \\ C \end{bmatrix} \in \mathbb{R}^{n \times n}$, we have

$$[A^+ U_{\mathcal{S}} \ , \ B] D = [A^+ U_{\mathcal{S}} \ , \ B] \begin{bmatrix} U_{\mathcal{S}}^T A \\ C \end{bmatrix} = A^+ U_{\mathcal{S}} U_{\mathcal{S}}^T A + BC = A^+ A A^+ A + I_n - A^+ A \ .$$

Using Criteria 1 in Appendix 3.A.3 we obtain

$$[A^+ U_{\mathcal{S}} \ , \ B] D = A^+ A + I_n - A^+ A = I_n \ .$$

Therefore, D is invertible and $D^{-1} = [A^+ U_{\mathcal{S}} \ , \ B]$. We have

$$[U_{\mathcal{S}} \ , \ 0_{d_y \times (n-r)}] D = [U_{\mathcal{S}} \ , \ 0_{d_y \times (n-r)}] \begin{bmatrix} U_{\mathcal{S}}^T A \\ C \end{bmatrix} = U_{\mathcal{S}} U_{\mathcal{S}}^T A = AA^+A = A \ ,$$

where the last equality follows from Criteria 1 in Appendix 3.A.3. This proves the first equality of Lemma 11 Finally,

$$D^{-1} \begin{bmatrix} U_{\mathcal{S}}^T \\ 0_{(n-r)\times d_y} \end{bmatrix} = [A^+ U_{\mathcal{S}} \ , \ B] \begin{bmatrix} U_{\mathcal{S}}^T \\ 0_{(n-r)\times d_y} \end{bmatrix} = A^+ U_{\mathcal{S}} U_{\mathcal{S}}^T = A^+ A A^+ = A^+ \ ,$$

where the last equality follows again from Criteria 2 in Appendix 3.A.3. This concludes the proof of Lemma 11. □

Now we prove Lemma 10.

Proof of Lemma 10. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L associated with $\mathcal{S}$ and $r = \mathrm{rk}(W_H \cdots W_1)$.
Using Lemma 8, we have $r = \mathrm{rk}(W_H \cdots W_2)$.
If $r = 0$, the conclusion of Lemma 10 is trivial because of the convention $U_\emptyset \in \mathbb{R}^{d_y \times 0}$.
When $r \geq 1$, using Lemma 8 and Proposition 1, we have $W_H \cdots W_1 = P_K \Sigma_{YX} \Sigma_{XX}^{-1} = U_\mathcal{S} U_\mathcal{S}^T \Sigma_{YX} \Sigma_{XX}^{-1}$. Since Σ_{YX} is of full row rank this implies $P_K = KK^+ = U_\mathcal{S} U_\mathcal{S}^T$. Therefore, we can apply Lemma 11 with $n = d_1$ and $A = K$ to conclude that there exists an invertible matrix $D \in \mathbb{R}^{d_1 \times d_1}$ such that

$$K = [U_\mathcal{S} \ , \ 0_{d_y \times (d_1 - r)}] D$$

which is the form of K in Lemma 10. Moreover, Lemma 11 also guarantees that

$$K^+ = D^{-1} \begin{bmatrix} U_\mathcal{S}^T \\ 0_{(d_1 - r) \times d_y} \end{bmatrix} .$$

Using Lemma 8, we have $W_1 = K^+ \Sigma_{YX} \Sigma_{XX}^{-1} + M$ with $KM = 0$. Therefore, $W_1 = D^{-1} \begin{bmatrix} U_\mathcal{S}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0_{(d_1 - r) \times d_x} \end{bmatrix} + M$, with $KM = 0$. This concludes the proof of Lemma 10. □

3.B.4 Proof of Lemma 2

For any $h \in [\![1, H-1]\!]$ let $D_h \in \mathbb{R}^{d_h \times d_h}$ be an invertible matrix. We define $\widetilde{\mathbf{W}} = (\widetilde{W}_H, \dots, \widetilde{W}_1)$ by $\widetilde{W}_H = W_H D_{H-1}$, $\widetilde{W}_1 = D_1^{-1} W_1$ and $\widetilde{W}_h = D_h^{-1} W_h D_{h-1}$ for all $h \in [\![2, H-1]\!]$.
Assume that $\mathbf{W} = (W_H, \dots, W_1)$ is a first-order critical point. Then using Lemma 3 this is equivalent to

$$\begin{cases} \nabla_{W_h} L(\mathbf{W}) = 2(W_H \cdots W_{h+1})^T (W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX})(W_{h-1} \cdots W_1)^T = 0 \quad \forall h \in [\![2, H-1]\!] \\ \nabla_{W_H} L(\mathbf{W}) = 2(W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX})(W_{H-1} \cdots W_1)^T = 0 \\ \nabla_{W_1} L(\mathbf{W}) = 2(W_H \cdots W_2)^T (W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX}) = 0 \ . \end{cases} \tag{3.B.4}$$

Using the definition of $\widetilde{\mathbf{W}}$ above, we have

$$\begin{cases} W_H \cdots W_1 = \widetilde{W}_H \cdots \widetilde{W}_1 \\ W_H \cdots W_{h+1} = \widetilde{W}_H \cdots \widetilde{W}_{h+1} D_h^{-1} \quad \forall h \in [\![1, H-1]\!] \\ W_{h-1} \cdots W_1 = D_{h-1} \widetilde{W}_{h-1} \cdots \widetilde{W}_1 \quad \forall h \in [\![2, H]\!] \ . \end{cases}$$

Therefore (3.B.4) is equivalent to

$$\begin{cases} (D_h^{-1})^T(\widetilde{W}_H\cdots\widetilde{W}_{h+1})^T(\widetilde{W}_H\cdots\widetilde{W}_1\Sigma_{XX}-\Sigma_{YX})(\widetilde{W}_{h-1}\cdots\widetilde{W}_1)^T D_{h-1}^T=0 \quad \forall h\in[\![2,H-1]\!] \\ (\widetilde{W}_H\cdots\widetilde{W}_1\Sigma_{XX}-\Sigma_{YX})(\widetilde{W}_{H-1}\cdots\widetilde{W}_1)^T D_{H-1}^T=0 \\ (D_1^{-1})^T(\widetilde{W}_H\cdots\widetilde{W}_2)^T(\widetilde{W}_H\cdots\widetilde{W}_1\Sigma_{XX}-\Sigma_{YX})=0\,. \end{cases}$$

This is equivalent to

$$\begin{cases} \nabla_{W_h}L(\widetilde{\mathbf{W}})=2(\widetilde{W}_H\cdots\widetilde{W}_{h+1})^T(\widetilde{W}_H\cdots\widetilde{W}_1\Sigma_{XX}-\Sigma_{YX})(\widetilde{W}_{h-1}\cdots\widetilde{W}_1)^T=0 \quad \forall h\in[\![2,H-1]\!] \\ \nabla_{W_H}L(\widetilde{\mathbf{W}})=2(\widetilde{W}_H\cdots\widetilde{W}_1\Sigma_{XX}-\Sigma_{YX})(\widetilde{W}_{H-1}\cdots\widetilde{W}_1)^T=0 \\ \nabla_{W_1}L(\widetilde{\mathbf{W}})=2(\widetilde{W}_H\cdots\widetilde{W}_2)^T(\widetilde{W}_H\cdots\widetilde{W}_1\Sigma_{XX}-\Sigma_{YX})=0\,. \end{cases}$$

which is equivalent to $\nabla_{W_h}L(\widetilde{\mathbf{W}})=0$, for all $h\in[\![1,H]\!]$. Therefore, $\mathbf{W}$ is a first-order critical point if and only if $\widetilde{\mathbf{W}}$ is a first-order critical point. This proves the first part of the proposition.

Now assume that $\mathbf{W}=(W_H,\dots,W_1)$ is a first-order critical point such that it is not a second-order critical point. Note that from the first part of the proof $\widetilde{\mathbf{W}}=(\widetilde{W}_H,\dots,\widetilde{W}_1)$ is also a first-order critical point. Let us prove that $\widetilde{\mathbf{W}}$ is not a second-order critical point. Using Lemma 1, since $\mathbf{W}$ is not a second-order critical point, there exist $\mathbf{W}'=(W_H',\dots,W_1')$ such that, if we denote $\mathbf{W}(t)=\mathbf{W}+t\mathbf{W}'$, the second-order term of $L(\mathbf{W}(t))$ is strictly negative i.e $c_2(\mathbf{W},\mathbf{W}')<0$. We will prove that there exist $\widetilde{\mathbf{W}}'$ such that $c_2(\widetilde{\mathbf{W}},\widetilde{\mathbf{W}}')<0$ and, using again Lemma 1, we conclude.

As already said, we set $W_h(t)=W_h+tW_h'$, for all $h\in[\![1,H]\!]$. We denote

$$\begin{cases} \widetilde{W}_H(t)=\widetilde{W}_H+t\widetilde{W}_H'=\widetilde{W}_H+tW_H'D_{H-1} \\ \widetilde{W}_1(t)=\widetilde{W}_1+t\widetilde{W}_1'=\widetilde{W}_1+tD_1^{-1}W_1' \\ \widetilde{W}_h(t)=\widetilde{W}_h+t\widetilde{W}_h'=\widetilde{W}_h+tD_h^{-1}W_h'D_{h-1} \quad \forall h\in[\![2,H-1]\!] \\ \widetilde{\mathbf{W}}'=(\widetilde{W}_H',\dots,\widetilde{W}_1')\,. \end{cases}$$

Hence, we have (where $\prod_{h=H-1}^{2}A_h$ should read as $A_{H-1}\cdots A_2$)

$$\begin{aligned} \widetilde{W}_H(t)\cdots\widetilde{W}_1(t) &= (W_HD_{H-1}+tW_H'D_{H-1})\prod_{h=H-1}^{2}\left(D_h^{-1}W_hD_{h-1}+tD_h^{-1}W_h'D_{h-1}\right)\,(D_1^{-1}W_1+tD_1^{-1}W_1') \\ &= (W_H+tW_H')\cdots(W_1+tW_1') \\ &= W_H(t)\cdots W_1(t)\,. \end{aligned}$$

Therefore, $L(\widetilde{\mathbf{W}}(t))=L(\mathbf{W}(t))$ and

$$c_2(\widetilde{\mathbf{W}},\widetilde{\mathbf{W}}')=c_2(\mathbf{W},\mathbf{W}')\,.$$

Since by hypothesis $c_2(\mathbf{W},\mathbf{W}')<0$, we conclude that $c_2(\widetilde{\mathbf{W}},\widetilde{\mathbf{W}}')<0$. Hence $(\widetilde{W}_H,\dots,\widetilde{W}_1)$ is not a second-order critical point.

We prove that if $\widetilde{\mathbf{W}}$ is not a second-order critical point then $\mathbf{W}$ is not a second-order critical point in the same way, by changing D_h with D_h^{-1} for all $h \in [\![1, H]\!]$. This proves the second part of the proposition and concludes the proof.

3.B.5 Proof of Proposition 6

Let $\mathcal{S} \subset [\![1, d_y]\!]$ of size $r \in [\![0, r_{max}]\!]$ and $Q = [\![1, d_y]\!] \backslash \mathcal{S}$. Let $Z_H \in \mathbb{R}^{(d_y - r) \times (d_{H-1} - r)}$, $Z_1 \in \mathbb{R}^{(d_1 - r) \times d_x}$ and $Z_h \in \mathbb{R}^{(d_h - r) \times (d_{h-1} - r)}$ for $h \in [\![2, H-1]\!]$. Let the parameter of the network $\mathbf{W} = (W_H, \dots, W_1)$ be defined as follows:

$$\begin{cases} W_H = [U_{\mathcal{S}}, U_Q Z_H] \\ W_1 = \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \\ W_h = \begin{bmatrix} I_r & 0 \\ 0 & Z_h \end{bmatrix} \quad \forall h \in [\![2, H-1]\!] \, . \end{cases} \tag{3.B.5}$$

Note that the above definition of $\mathbf{W}$ does not involve the matrices $D_h \in \mathbb{R}^{d_h \times d_h}$. In fact, using Lemma 2, it suffices to prove that, when $r = r_{max}$ or there exist $h_1 \neq h_2$ such that $Z_{h_1} = 0$ and $Z_{h_2} = 0$, the $\mathbf{W}$ defined above is a first-order critical point to conclude that Proposition 6 holds .
We have

$$\begin{aligned} W_H \cdots W_1 &= [U_{\mathcal{S}}, U_Q Z_H] \begin{bmatrix} I_r & 0 \\ 0 & Z_{H-1} \end{bmatrix} \cdots \begin{bmatrix} I_r & 0 \\ 0 & Z_2 \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \\ &= U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} + U_Q Z_H Z_{H-1} \cdots Z_2 Z_1 \end{aligned}$$

If there exists $h_1 \neq h_2$ such that $Z_{h_1} = 0$ and $Z_{h_2} = 0$, it immediately follows that $W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$.
If $r = r_{max}$, then there exists $h \in [\![0, H]\!]$ such that $r = d_h$.

— If $r = d_H = d_y$, then $U_Q \in \mathbb{R}^{d_y \times 0}$ and $Z_H \in \mathbb{R}^{0 \times (d_{H-1} - r)}$, which, using conventions in Section 3.2, gives

$$U_Q Z_H = 0_{d_y \times (d_{H-1} - r)}. \tag{3.B.6}$$

Therefore, $W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$.

— If $r = d_0 = d_x$, then, since $d_x \geq d_y$, we have $r = d_y$, which we have already treated in the previous item.

— If $r = d_h$ for some $h \in [\![2, H-1]\!]$, then $Z_{h+1} \in \mathbb{R}^{(d_{h+1} - r) \times 0}$ and $Z_h \in \mathbb{R}^{0 \times (d_{h-1} - r)}$, which, using the conventions on Section 3.2, gives

$$Z_{h+1} Z_h = 0_{(d_{h+1} - r) \times (d_{h-1} - r)}. \tag{3.B.7}$$

Therefore, $W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$.

— If $r = d_1$, then $Z_2 \in \mathbb{R}^{(d_2 - r) \times 0}$ and $Z_1 \in \mathbb{R}^{0 \times d_x}$, which, using the conventions on

Section 3.2, gives

$$Z_2 Z_1 = 0_{(d_2-r)\times d_x}. \tag{3.B.8}$$

Therefore, $W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$.

Note that these results still hold if there is more than one layer with the minimum width. Therefore, in all cases, when $r = r_{max}$ or there exist $h_1 \neq h_2$ such that $Z_{h_1} = 0$ and $Z_{h_2} = 0$ we have,

$$W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \ . \tag{3.B.9}$$

Let us prove that the gradient of L at $\mathbf{W}$ is equal to zero.
Recall that from Lemma 3 we have

$$\begin{aligned}
\nabla_{W_h} L(\mathbf{W}) &= 2(W_H \cdots W_{h+1})^T (W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX})(W_{h-1} \cdots W_1)^T \quad \forall h \in [\![2, H-1]\!] \\
\nabla_{W_H} L(\mathbf{W}) &= 2(W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX})(W_{H-1} \cdots W_1)^T \\
\nabla_{W_1} L(\mathbf{W}) &= 2(W_H \cdots W_2)^T (W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX}) \ .
\end{aligned}$$

Using (3.B.9) and Lemma 6, we have

$$\begin{aligned}
W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX} &= U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XX} - \Sigma_{YX} \\
&= (U_{\mathcal{S}} U_{\mathcal{S}}^T - I_{d_y}) \Sigma_{YX} \\
&= -U_Q U_Q^T \Sigma_{YX} \ .
\end{aligned}$$

Also, using (3.B.5), for all $h \in [\![1, H-1]\!]$,

$$\begin{aligned}
W_H \cdots W_{h+1} &= [U_{\mathcal{S}}, U_Q Z_H] \begin{bmatrix} I_r & 0 \\ 0 & Z_{H-1} \end{bmatrix} \cdots \begin{bmatrix} I_r & 0 \\ 0 & Z_{h+1} \end{bmatrix} \\
&= [U_{\mathcal{S}}, U_Q Z_H Z_{H-1} \cdots Z_{h+1}]
\end{aligned}$$

and, for all $h \in [\![2, H]\!]$,

$$\begin{aligned}
W_{h-1} \cdots W_1 &= \begin{bmatrix} I_r & 0 \\ 0 & Z_{h-1} \end{bmatrix} \cdots \begin{bmatrix} I_r & 0 \\ 0 & Z_2 \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \\
&= \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_{h-1} \cdots Z_2 Z_1 \end{bmatrix} \ .
\end{aligned}$$

We have, for all $h \in [\![2, H-1]\!]$,

$$\begin{aligned}
\frac{1}{2} \left(\nabla_{W_h} L(\mathbf{W})\right)^T &= (W_{h-1} \cdots W_1)(W_H \cdots W_1 \Sigma_{XX} - \Sigma_{YX})^T (W_H \cdots W_{h+1}) \\
&= - \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_{h-1} \cdots Z_2 Z_1 \end{bmatrix} (U_Q U_Q^T \Sigma_{YX})^T [U_{\mathcal{S}}, U_Q Z_H Z_{H-1} \cdots Z_{h+1}] \\
&= - \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_{h-1} \cdots Z_2 Z_1 \end{bmatrix} \Sigma_{XY} U_Q U_Q^T [U_{\mathcal{S}}, U_Q Z_H Z_{H-1} \cdots Z_{h+1}]
\end{aligned}$$

$$= - \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY} U_Q \\ Z_{h-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q \end{bmatrix} [U_Q^T U_{\mathcal{S}}, U_Q^T U_Q Z_H Z_{H-1} \cdots Z_{h+1}] \, .$$

Using the definition of Σ, Lemma 6 and Lemma 7, we have

$$\begin{aligned} \frac{1}{2} \left(\nabla_{W_h} L(\mathbf{W}) \right)^T &= - \begin{bmatrix} U_{\mathcal{S}}^T \Sigma U_Q \\ Z_{h-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q \end{bmatrix} [0_{(d_y - r) \times r}, Z_H Z_{H-1} \cdots Z_{h+1}] \\ &= - \begin{bmatrix} 0_{r \times (d_y - r)} \\ Z_{h-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q \end{bmatrix} [0_{(d_y - r) \times r}, Z_H Z_{H-1} \cdots Z_{h+1}] \\ &= - \begin{bmatrix} 0_{r \times r} & 0_{r \times (d_h - r)} \\ 0_{(d_{h-1} - r) \times r} & Z_{h-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1} \cdots Z_{h+1} \end{bmatrix} . \end{aligned}$$

Proceeding similarly, we obtain

$$\frac{1}{2} \left(\nabla_{W_H} L(\mathbf{W}) \right)^T = - \begin{bmatrix} 0_{r \times d_y} \\ Z_{H-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q U_Q^T \end{bmatrix}$$

and

$$\frac{1}{2} \left(\nabla_{W_1} L(\mathbf{W}) \right)^T = - [0_{d_x \times r} \, , \, \Sigma_{XY} U_Q Z_H Z_{H-1} \cdots Z_2] \, .$$

If there exists $h_1 \neq h_2$ such that $Z_{h_1} = 0$ and $Z_{h_2} = 0$, we can easily see that the gradient is equal to zero, i.e., $\mathbf{W}$ is a first-order critical point.
If $r = r_{max}$, then there exists $h' \in [\![1, H]\!]$ such that $r = d_{h'}$. Using the same arguments as above that yielded (3.B.6), (3.B.7) and (3.B.8), we have,

— For $h = 1$,
  — if $r = d_1$, we have $Z_2 \in \mathbb{R}^{(d_2 - r) \times 0}$ and therefore $\Sigma_{XY} U_Q Z_H Z_{H-1} \cdots Z_2 \in \mathbb{R}^{d_x \times 0}$.
  — if $r = d_H$, then $U_Q Z_H = 0_{d_y \times (d_{H-1} - r)}$ and therefore $\Sigma_{XY} U_Q Z_H Z_{H-1} \cdots Z_2 = 0_{d_x \times (d_1 - r)}$.
  — if $r = d_{h'}$ for some $h' \in [\![2, H-1]\!]$, then $Z_{h'+1} Z_{h'} = 0_{(d_{h'+1} - r) \times (d_{h'-1} - r)}$ and therefore $\Sigma_{XY} U_Q Z_H Z_{H-1} \cdots Z_2 = 0_{d_x \times (d_1 - r)}$.

  Hence, in all cases, $\nabla_{W_1} L(\mathbf{W}) = 0$.

— For $h = H$,
  — if $r = d_H = d_y$, then $U_Q U_Q^T = 0_{d_y \times d_y}$ and therefore $Z_{H-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q U_Q^T = 0_{(d_{H-1} - r) \times d_y}$.
  — if $r = d_{H-1}$, then $Z_{H-1} \in \mathbb{R}^{0 \times (d_{H-2} - r)}$ and therefore $Z_{H-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q U_Q^T \in \mathbb{R}^{0 \times d_y}$.
  — if $r = d_{h'}$ for some $h' \in [\![2, H-2]\!]$, then $Z_{h'+1} Z_{h'} = 0_{(d_{h'+1} - r) \times (d_{h'-1} - r)}$ and therefore $Z_{H-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q U_Q^T = 0_{(d_{H-1} - r) \times d_y}$.
  — if $r = d_1$, then $Z_2 Z_1 = 0_{(d_2 - r) \times d_x}$ and therefore $Z_{H-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q U_Q^T = 0_{(d_{H-1} - r) \times d_y}$.

Hence, in all cases, $\nabla_{W_H} L(\mathbf{W}) = 0$.

— For $h \in [\![2, H-1]\!]$,

— if $r = d_{h-1}$, then $Z_{h-1} \in \mathbb{R}^{0\times(d_{h-2}-r)}$ and therefore $Z_{h-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{h+1} \in \mathbb{R}^{0\times(d_h-r)}$.

— if $r = d_h$, then $Z_{h+1} \in \mathbb{R}^{(d_{h+1}-r)\times 0}$ and therefore $Z_{h-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{h+1} \in \mathbb{R}^{(d_{h-1}-r)\times 0}$.

— if $r = d_H$, then $U_Q Z_H = 0_{d_y\times(d_{H-1}-r)}$ and therefore $Z_{h-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{h+1} = 0_{(d_{h-1}-r)\times(d_h-r)}$.

— if $r = d_1$, then $Z_2 Z_1 = 0_{(d_2-r)\times d_x}$ and therefore $Z_{h-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{h+1} = 0_{(d_{h-1}-r)\times(d_h-r)}$.

— if $r = d_{h'}$ for some $h' \in [\![2, H-1]\!] \setminus \{h, h-1\}$, then $Z_{h'+1}Z_{h'} = 0_{(d_{h'+1}-r)\times(d_{h'-1}-r)}$ and therefore $Z_{h-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{h+1} = 0_{(d_{h-1}-r)\times(d_h-r)}$.

Hence, in all cases, $\nabla_{W_h} L(\mathbf{W}) = 0$.

Therefore, when $r = r_{max}$, $\mathbf{W}$ is also a first-order critical point of L.

3.B.6 Proof of Proposition 2

Let $\mathcal{S} \subset [\![1, d_y]\!]$ such that $|\mathcal{S}| = r \leq r_{max}$, and $Q = [\![1, d_y]\!] \setminus \mathcal{S}$.
We define $\mathbf{W} = (W_H, \ldots, W_1)$ by:

$$
\begin{aligned}
W_H &= [U_{\mathcal{S}}, 0_{d_y\times(d_{H-1}-r)}] \\
W_h &= \begin{bmatrix} I_r & 0_{r\times(d_{h-1}-r)} \\ 0_{(d_h-r)\times r} & 0_{(d_h-r)\times(d_{h-1}-r)} \end{bmatrix} \quad \forall h \in [\![2, H-1]\!] \\
W_1 &= \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0_{(d_1-r)\times d_x} \end{bmatrix},
\end{aligned}
$$

By Proposition 6, $\mathbf{W}$ is a first-order critical point of L. Moreover, we have $W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$. Therefore, $\mathbf{W}$ is a first-order critical point associated with $\mathcal{S}$.

3.B.7 Proof of Proposition 3

Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point and $r = \mathrm{rk}(W_H \cdots W_1)$, using Proposition 1 there exists a unique $\mathcal{S} \subset [\![1, d_y]\!]$ of size r such that

$$W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1},$$

which implies

$$W_H \cdots W_1 \Sigma_{XY} = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma.$$

Let $i, j \in [\![1, H]\!]$ such that $i > j$. The complementary blocks are $W_{j-1}\cdots W_1 \Sigma_{XY} W_H \cdots W_{i+1}$ and $W_{i-1}\cdots W_{j+1}$.

Using Lemma 4 and $U_{\mathcal{S}}^T U_{\mathcal{S}} = I_r$, we have, for the second complementary block,

$$\mathrm{rk}(W_{i-1}\cdots W_{j+1}) \geq \mathrm{rk}(W_H \cdots W_1 \Sigma_{XY}) = \mathrm{rk}(U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma) \geq \mathrm{rk}(U_{\mathcal{S}}^T (U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma)\Sigma^{-1} U_{\mathcal{S}}) = \mathrm{rk}(I_r) = r \ .$$

For the first complementary block, using the same arguments, we have

$$\begin{aligned}
\mathrm{rk}(W_{j-1}\cdots W_1 \Sigma_{XY} W_H \cdots W_{i+1}) &\geq \mathrm{rk}(W_H \cdots W_1 \Sigma_{XY} W_H \cdots W_1 \Sigma_{XY}) \\
&= \mathrm{rk}(U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma) \\
&\geq \mathrm{rk}\left(U_{\mathcal{S}}^T (U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma)\Sigma^{-1} U_{\mathcal{S}}\right) \\
&= \mathrm{rk}(U_{\mathcal{S}}^T \Sigma U_{\mathcal{S}}) \ .
\end{aligned}$$

Recall that, from the diagonalization of Σ, we have $\Sigma U = U\Lambda$, hence, $\Sigma U_{\mathcal{S}} = U_{\mathcal{S}} \mathrm{diag}((\lambda_s)_{s\in\mathcal{S}})$

$$\begin{aligned}
\mathrm{rk}(W_{j-1}\cdots W_1 \Sigma_{XY} W_H \cdots W_{i+1}) &\geq \mathrm{rk}(U_{\mathcal{S}}^T U_{\mathcal{S}} \mathrm{diag}((\lambda_s)_{s\in\mathcal{S}})) \\
&= \mathrm{rk}(\mathrm{diag}((\lambda_s)_{s\in\mathcal{S}})) \\
&= r \ .
\end{aligned}$$

This concludes the proof.

3.B.8 Proof of Proposition 4

Let $H \geq 3$, $\mathcal{S} = [\![1, r]\!]$ with $0 \leq r < r_{max}$. We define $\mathbf{W}$ as follows:

$$\begin{cases}
W_H = [U_{\mathcal{S}}, 0] \\
W_h = \begin{bmatrix} I_r & 0 \\ 0 & Z_h \end{bmatrix} \quad \text{for} \quad h \in [\![2, H-1]\!] \\
W_1 = \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0 \end{bmatrix} .
\end{cases} \tag{3.B.10}$$

Using Proposition 6, $\mathbf{W}$ is a first-order critical point associated with $\mathcal{S}$. Let us show that depending on the choice of $(Z_h)_{h=2..H-1}$, $\mathbf{W}$ can be tightened or non-tightened.
Since $H \geq 3$, there exists $h \in [\![2, H-1]\!]$. If we choose $Z_{H-1}, \ldots, Z_2$ such that $Z_{H-1}\cdots Z_2 \neq 0$ (e.g. when only the top left entry of each Z_h is nonzero, which is possible since $r < r_{max} = \min(d_H, \ldots, d_0)$) then $\mathbf{W}$ is non-tightened. Indeed, the pivot $(H, 1)$ is non-tightened because $\mathrm{rk}(\Sigma_{XY}) = d_y > r$ and $\mathrm{rk}(W_{H-1}\cdots W_2) = \mathrm{rk}\left(\begin{bmatrix} I_r & 0 \\ 0 & Z_{H-1}\cdots Z_2 \end{bmatrix}\right) > r$.
If we choose $Z_{H-1}, \ldots, Z_2$ such that $Z_{H-1}\cdots Z_2 = 0$ (e.g. $Z_2 = 0$), then $\mathbf{W}$ is tightened. Indeed, the pivot $(H, 1)$ is tightened because $W_{H-1}\cdots W_2 = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$ is of rank r, and by construction we have $\mathrm{rk}(W_H) = \mathrm{rk}(W_1) = r$. Hence, all the other pivots are tightened because at least one of their complementary blocks includes W_H or W_1, and therefore, using Proposition 3, is of rank r. Therefore, $\mathbf{W}$ is tightened.

3.C Parameterization of first-order critical points and global minimizers

In this section, we prove Propositions 5 and 7 that were stated in Section 3.3.3.

3.C.1 Proof of Proposition 5

Before proving Proposition 5, we introduce and prove two lemmas.

Lemma 12. Let r be a nonnegative integer, and let n and p be two positive integers larger than or equal to r. Let $\mathcal{S} \subset [\![1, d_y]\!]$ of size r and let $Q = [\![1, d_y]\!] \setminus \mathcal{S}$. Let $A \in \mathbb{R}^{d_y \times n}$ and $B \in \mathbb{R}^{n \times p}$ be two matrices such that

$$AB = [U_{\mathcal{S}}, 0] \, .$$

Then, there exist an invertible matrix $D \in \mathbb{R}^{n \times n}$ and two matrices $N \in \mathbb{R}^{(d_y - r) \times (n-r)}$ and $B_{DR} \in \mathbb{R}^{(n-r) \times (p-r)}$ such that

$$AD = [U_{\mathcal{S}}, U_Q N] \tag{3.C.1}$$

$$D^{-1}B = \begin{bmatrix} I_r & 0 \\ 0 & B_{DR} \end{bmatrix} . \tag{3.C.2}$$

In the proof below, we can easily see that the result still holds for $r = 0$ and $r = \min(d_y, n, p)$ with the conventions adopted in Section 3.2.

Proof. Let n and p be non-negative integers such that $n, p \geq r$ and $A \in \mathbb{R}^{d_y \times n}$ and $B \in \mathbb{R}^{n \times p}$ such that

$$AB = [U_{\mathcal{S}}, 0]. \tag{3.C.3}$$

Recall that for any matrix C with n columns we write $C = [C_1, C_2, \ldots, C_n]$ where C_i represents the i-th column of C.
We have from (3.C.3),

$$A[B_1, B_2, \ldots, B_r] = U_{\mathcal{S}} \, . \tag{3.C.4}$$

Since the columns of U are linearly independent, we have

$$\mathrm{rk}(A[B_1, B_2, \ldots, B_r]) = \mathrm{rk}(U_{\mathcal{S}}) = r$$

and $\{B_1, \ldots, B_r\}$ are necessarily also linearly independent. Using the incomplete basis theorem, we complement $(B_1, \ldots, B_r)$ to form a basis $(B_1, \ldots, B_r, E_{r+1}, \ldots, E_n)$. We set $E = [B_1, \ldots, B_r, E_{r+1}, \ldots, E_n] \in \mathbb{R}^{n \times n}$. By construction, the matrix E is invertible.

We now set $A' = AE$ and $B' = E^{-1}B$. In particular $A'B' = AB$.
Also, note that

$$E \begin{bmatrix} I_r \\ 0 \end{bmatrix} = [B_1, \ldots, B_r] \, ,$$

so that

$$E^{-1}[B_1, \dots, B_r] = \begin{bmatrix} I_r \\ 0 \end{bmatrix} .$$

Therefore, we can write

$$B' = E^{-1}B = \begin{bmatrix} I_r & B_{UR} \\ 0 & B_{DR} \end{bmatrix} , \tag{3.C.5}$$

with $B_{UR} \in \mathbb{R}^{r\times(p-r)}$ and $B_{DR} \in \mathbb{R}^{(n-r)\times(p-r)}$ such that

$$\begin{bmatrix} B_{UR} \\ B_{DR} \end{bmatrix} = E^{-1}[B_{r+1}, \dots, B_p] .$$

We define $L \in \mathbb{R}^{r\times(n-r)}$ and $N \in \mathbb{R}^{(d_y-r)\times(n-r)}$ by $\begin{bmatrix} L \\ N \end{bmatrix} = [U_{\mathcal{S}}, U_Q]^{-1}[AE_{r+1}, \dots, AE_n]$. We have

$$[AE_{r+1}, \dots, AE_n] = [U_{\mathcal{S}}, U_Q] \begin{bmatrix} L \\ N \end{bmatrix} = U_{\mathcal{S}}L + U_Q N . \tag{3.C.6}$$

We also define the invertible matrix $F = \begin{bmatrix} I_r & L \\ 0 & I_{n-r} \end{bmatrix} \in \mathbb{R}^{n\times n}$. Using (3.C.4) and (3.C.6) we have

$$\begin{aligned} A' &= AE \\ &= A[B_1, \dots, B_r, E_{r+1}, \dots, E_n] \\ &= [U_{\mathcal{S}}, U_{\mathcal{S}}L + U_Q N] \\ &= [U_{\mathcal{S}}, U_Q N] \begin{bmatrix} I_r & L \\ 0 & I_{n-r} \end{bmatrix} \\ &= [U_{\mathcal{S}}, U_Q N]F . \end{aligned}$$

Therefore, defining the invertible matrix $D = EF^{-1} \in \mathbb{R}^{n\times n}$, we finally have

$$AD = AEF^{-1} = [U_{\mathcal{S}}, U_Q N] . \tag{3.C.7}$$

This proves (3.C.1).
We also have, using (3.C.5) and the definition of F

$$\begin{aligned} D^{-1}B &= FE^{-1}B \\ &= FB' \\ &= \begin{bmatrix} I_r & L \\ 0 & I_{n-r} \end{bmatrix} \begin{bmatrix} I_r & B_{UR} \\ 0 & B_{DR} \end{bmatrix} \\ &= \begin{bmatrix} I_r & B_{UR} + LB_{DR} \\ 0 & B_{DR} \end{bmatrix} . \end{aligned} \tag{3.C.8}$$

However, noticing that, since (3.C.3) holds,

$$(AD)(D^{-1}B) = AB = [U_{\mathcal{S}}, 0] \,,$$

and using (3.C.7) and (3.C.8) we obtain

$$[U_{\mathcal{S}}, U_Q N] \begin{bmatrix} I_r & B_{UR} + LB_{DR} \\ 0 & B_{DR} \end{bmatrix} = [U_{\mathcal{S}}, 0] \,.$$

Therefore $U_{\mathcal{S}}(B_{UR} + LB_{DR}) + U_Q N B_{DR} = 0$. Since $[U_{\mathcal{S}}, U_Q]$ is invertible we get $B_{UR} + LB_{DR} = 0$ and $NB_{DR} = 0$.
Finally, (3.C.8) becomes

$$D^{-1}B = \begin{bmatrix} I_r & 0 \\ 0 & B_{DR} \end{bmatrix} .$$

This proves (3.C.2) and concludes the proof. □

The second lemma states that if the product of two factors takes the format of (3.C.2), then up to the product by an invertible matrix, the two factors have the same format. In the proof of Proposition 5, we will use this property several times to establish (3.3.3).

Lemma 13. Let r, q, n and p be positive integers such that $r \leq \min(q, n, p)$. Let $B \in \mathbb{R}^{q \times n}$, $C \in \mathbb{R}^{n \times p}$ and $P \in \mathbb{R}^{(q-r)\times(p-r)}$ such that

$$BC = \begin{bmatrix} I_r & 0 \\ 0 & P \end{bmatrix} .$$

Then, there exist an invertible matrix $D \in \mathbb{R}^{n \times n}$ and two matrices $B_{DR} \in \mathbb{R}^{(q-r)\times(n-r)}$ and $C_{DR} \in \mathbb{R}^{(n-r)\times(p-r)}$ such that

$$BD = \begin{bmatrix} I_r & 0 \\ 0 & B_{DR} \end{bmatrix} \tag{3.C.9}$$

$$D^{-1}C = \begin{bmatrix} I_r & 0 \\ 0 & C_{DR} \end{bmatrix} . \tag{3.C.10}$$

In the proof below, we can easily see that the result still holds for $r = 0$ and $r = \min(q, n, p)$ with the conventions adopted in Section 3.2.

Proof. Let r, q, n and p be positive integers such that $r \leq \min(q, n, p)$. Let $B \in \mathbb{R}^{q \times n}$, $C \in \mathbb{R}^{n \times p}$ and $P \in \mathbb{R}^{(q-r)\times(p-r)}$ such that

$$BC = \begin{bmatrix} I_r & 0 \\ 0 & P \end{bmatrix} . \tag{3.C.11}$$

We have

$$B[C_1, C_2, \ldots, C_r] = \begin{bmatrix} I_r \\ 0 \end{bmatrix} . \tag{3.C.12}$$

Since the columns of $\begin{bmatrix} I_r \\ 0 \end{bmatrix}$ are linearly independent,

$$\mathrm{rk}(B[C_1, C_2, \ldots, C_r]) = r$$

and the vectors $C_1, \ldots, C_r$ are necessarily also linearly independent. Using the incomplete basis theorem, we complement $(C_1, \ldots, C_r)$ to form a basis $(C_1, \ldots, C_r, E_{r+1}, \ldots, E_n)$. We denote $E = [C_1, \ldots, C_r, E_{r+1}, \ldots, E_n] \in \mathbb{R}^{n \times n}$. By construction, the matrix E is invertible.
We now set $B' = BE$ and $C' = E^{-1}C$. In particular

$$B'C' = BC. \tag{3.C.13}$$

Also notice that

$$E \begin{bmatrix} I_r \\ 0 \end{bmatrix} = [C_1, \ldots, C_r] \,,$$

so that

$$E^{-1}[C_1, \ldots, C_r] = \begin{bmatrix} I_r \\ 0 \end{bmatrix} \,.$$

Therefore, we can write

$$C' = E^{-1}C = \begin{bmatrix} I_r & C_{UR} \\ 0 & C_{DR} \end{bmatrix} \,, \tag{3.C.14}$$

where $C_{UR} \in \mathbb{R}^{r \times (p-r)}$ and $C_{DR} \in \mathbb{R}^{(n-r) \times (p-r)}$ are such that $\begin{bmatrix} C_{UR} \\ C_{DR} \end{bmatrix} = E^{-1}[C_{r+1}, \ldots, C_p]$.
Now notice that, using (3.C.12),

$$\begin{aligned} B' &= BE \\ &= B[C_1, \ldots, C_r, E_{r+1}, \ldots, E_n] \\ &= \begin{bmatrix} I_r & B_{UR} \\ 0 & B_{DR} \end{bmatrix} \,, \end{aligned} \tag{3.C.15}$$

where $B_{UR} \in \mathbb{R}^{r \times (n-r)}$ and $B_{DR} \in \mathbb{R}^{(q-r) \times (n-r)}$ are such that $\begin{bmatrix} B_{UR} \\ B_{DR} \end{bmatrix} = B[E_{r+1}, \ldots, E_n]$.
Plugging (3.C.15), (3.C.14) and (3.C.11) in the equality (3.C.13), we obtain

$$\begin{bmatrix} I_r & B_{UR} \\ 0 & B_{DR} \end{bmatrix} \begin{bmatrix} I_r & C_{UR} \\ 0 & C_{DR} \end{bmatrix} = \begin{bmatrix} I_r & 0 \\ 0 & P \end{bmatrix} \,,$$

which yields

$$\begin{bmatrix} I_r & C_{UR} + B_{UR}C_{DR} \\ 0 & B_{DR}C_{DR} \end{bmatrix} = \begin{bmatrix} I_r & 0 \\ 0 & P \end{bmatrix} \,.$$

Therefore, $C_{UR} + B_{UR}C_{DR} = 0$ or, equivalently ,

$$C_{UR} = -B_{UR}C_{DR} \,. \tag{3.C.16}$$

Define $F = \begin{bmatrix} I_r & -B_{UR} \\ 0 & I_{n-r} \end{bmatrix}$. The matrix F is invertible. Moreover, using (3.C.14) and (3.C.16) we have

$$\begin{aligned} C' &= \begin{bmatrix} I_r & -B_{UR}C_{DR} \\ 0 & C_{DR} \end{bmatrix} \\ &= \begin{bmatrix} I_r & -B_{UR} \\ 0 & I_{n-r} \end{bmatrix} \begin{bmatrix} I_r & 0 \\ 0 & C_{DR} \end{bmatrix} \\ &= F \begin{bmatrix} I_r & 0 \\ 0 & C_{DR} \end{bmatrix} . \end{aligned}$$

Therefore, if we define $D = EF$, D is invertible and

$$D^{-1}C = F^{-1}E^{-1}C = F^{-1}C' = \begin{bmatrix} I_r & 0 \\ 0 & C_{DR} \end{bmatrix} .$$

This proves (3.C.10).
In order to prove (3.C.9), we remark that, using (3.C.15) and the definition of F, we also have

$$\begin{aligned} BD &= BEF \\ &= B'F \\ &= \begin{bmatrix} I_r & B_{UR} \\ 0 & B_{DR} \end{bmatrix} \begin{bmatrix} I_r & -B_{UR} \\ 0 & I_{n-r} \end{bmatrix} \\ &= \begin{bmatrix} I_r & 0 \\ 0 & B_{DR} \end{bmatrix} . \end{aligned}$$

This proves (3.C.9) and concludes the proof. □

Now we prove Proposition 5.

Proof of Proposition 5. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L. Then using Lemma 10 there exist $D \in \mathbb{R}^{d_1 \times d_1}$ invertible and a matrix $M \in \mathbb{R}^{d_1 \times d_x}$ which satisfies $W_H \cdots W_2 M = 0$ such that

$$W_H \cdots W_2 = [U_{\mathcal{S}} \,,\, 0]D \tag{3.C.17}$$

$$W_1 = D^{-1} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0 \end{bmatrix} + M \,. \tag{3.C.18}$$

Denoting $D_1 = D^{-1}$ and using (3.C.17), we have $W_H \cdots W_2 D_1 = [U_{\mathcal{S}}, 0]$. Then applying Lemma 12 with $A = W_H$ and $B = W_{H-1} \cdots W_2 D_1$, there exist an invert-

ible matrix $D_{H-1} \in \mathbb{R}^{d_{H-1} \times d_{H-1}}$, and matrices $Z_H \in \mathbb{R}^{(d_y - r) \times (d_{H-1} - r)}$ and $B_{DR} \in \mathbb{R}^{(d_{H-1}-r)\times(d_1-r)}$ such that

$$\begin{aligned} \widetilde{W}_H := W_H D_{H-1} &= [U_{\mathcal{S}}, U_Q Z_H] \\ D_{H-1}^{-1} W_{H-1} \cdots W_2 D_1 &= \begin{bmatrix} I_r & 0 \\ 0 & B_{DR} \end{bmatrix} . \end{aligned} \tag{3.C.19}$$

The first equality proves (3.3.1).
Then applying Lemma 13 to (3.C.19) with $B = D_{H-1}^{-1} W_{H-1}$ and $C = W_{H-2} \cdots W_2 D_1$ we get the existence of an invertible matrix $D_{H-2} \in \mathbb{R}^{d_{H-2} \times d_{H-2}}$, $C_{DR} \in \mathbb{R}^{(d_{H-2}-r)\times(d_1-r)}$ and $Z_{H-1} \in \mathbb{R}^{(d_{H-1}-r)\times(d_{H-2}-r)}$ such that

$$\widetilde{W}_{H-1} := D_{H-1}^{-1} W_{H-1} D_{H-2} = \begin{bmatrix} I_r & 0 \\ 0 & Z_{H-1} \end{bmatrix} ,$$

and

$$D_{H-2}^{-1} W_{H-2} \cdots W_2 D_1 = \begin{bmatrix} I_r & 0 \\ 0 & C_{DR} \end{bmatrix} .$$

Reiterating the process by using Lemma 13 multiple times with $B = D_h^{-1} W_h$ and $C = W_{h-1} \cdots W_2 D_1$ for h decreasing from $H-2$ to 3, we can conclude that there exist invertible matrices $D_h \in \mathbb{R}^{d_h \times d_h}$ and matrices $Z_h \in \mathbb{R}^{(d_h - r)\times(d_{h-1}-r)}$, for $h \in [\![2, H-1]\!]$, such that

$$\widetilde{W}_h := D_h^{-1} W_h D_{h-1} = \begin{bmatrix} I_r & 0 \\ 0 & Z_h \end{bmatrix} \quad \forall h \in [\![2, H-1]\!] .$$

This entails (3.3.3).
We also have from (3.C.18) that $W_1 = D_1 \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0 \end{bmatrix} + M$ with $W_H \cdots W_2 M = 0$.
Therefore,

$$D_1^{-1} W_1 = \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0 \end{bmatrix} + D_1^{-1} M .$$

Using (3.C.17), $D_1 = D^{-1}$ and $W_H \cdots W_2 M = 0$, we obtain

$$[U_{\mathcal{S}} \ , \ 0] D_1^{-1} M = 0 .$$

Writing $D_1^{-1} M = \begin{bmatrix} Z_0 \\ Z_1 \end{bmatrix}$, where $Z_0 \in \mathbb{R}^{r \times d_x}$ and $Z_1 \in \mathbb{R}^{(d_1 - r) \times d_x}$, we have

$$\begin{aligned} 0 &= [U_{\mathcal{S}} \ , \ 0] D_1^{-1} M \\ &= [U_{\mathcal{S}} \ , \ 0] \begin{bmatrix} Z_0 \\ Z_1 \end{bmatrix} \\ &= U_{\mathcal{S}} Z_0 . \end{aligned}$$

Multiplying on the left by $U_{\mathcal{S}}^T$ we obtain

$$Z_0 = 0.$$

Therefore $D_1^{-1}M = \begin{bmatrix} 0 \\ Z_1 \end{bmatrix}$, which yields

$$\widetilde{W}_1 := D_1^{-1}W_1 = \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX}\Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} .$$

This proves (3.3.2).
Finally we have

$$\begin{aligned} \widetilde{W}_H \cdots \widetilde{W}_2 &= (W_H D_{H-1})(D_{H-1}^{-1} W_{H-1} D_{H-2}) \cdots (D_2^{-1} W_2 D_1) \\ &= W_H \cdots W_2 D_1 \\ &= [U_{\mathcal{S}} \ , \ 0] \ , \end{aligned}$$

where the last equality is due to (3.C.17) and $D_1 = D^{-1}$. This entails (3.3.4) and concludes the proof. □

3.C.2 Proof of Proposition 7

We first make a comment about notational subtleties to help understand the statement of Proposition 7, and then prove the proposition.

Recall that $r_{max} = \min(d_H, \dots, d_0)$, and $d_x = d_0 \geq d_y = d_H$ by assumption. Therefore, in the statement of Proposition 7, some blocks Z_h have 0 lines or 0 columns, and thus do not exist. For example, depending on the value of r_{max}, we have

$$\begin{cases} W_H = U_{\mathcal{S}_{max}} D_{H-1}^{-1} & \text{if } r_{max} = d_{H-1} \\ W_1 = D_1 U_{\mathcal{S}_{max}}^T \Sigma_{YX}\Sigma_{XX}^{-1} & \text{if } r_{max} = d_1 \end{cases}$$

and for $h \in [\![2, H-1]\!]$

$$W_h = \begin{cases} D_h \begin{bmatrix} I_{r_{max}} & 0 \end{bmatrix} D_{h-1}^{-1} & \text{if } r_{max} = d_h < d_{h-1} \\ D_h \begin{bmatrix} I_{r_{max}} \\ 0 \end{bmatrix} D_{h-1}^{-1} & \text{if } r_{max} = d_{h-1} < d_h \\ D_h I_{r_{max}} D_{h-1}^{-1} & \text{if } r_{max} = d_h = d_{h-1} \end{cases}$$

Also, if $r_{max} = d_y$, then $Q_{max} = \emptyset$, hence $U_{Q_{max}} \in \mathbb{R}^{d_y \times 0}$ and $Z_H \in \mathbb{R}^{0 \times (d_{H-1} - r_{max})}$. Then, using the convention in Section 3.2, $U_{Q_{max}} Z_H = 0_{d_y \times (d_{H-1} - r_{max})}$, so that $W_H = [U_{\mathcal{S}_{max}}, 0_{d_y \times (d_{H-1} - r_{max})}] D_{H-1}^{-1} \in \mathbb{R}^{d_y \times d_{H-1}}$.

We are now ready to prove the proposition.

Proof of Proposition 7. Let $\mathcal{S}_{max} = [\![1, r_{max}]\!]$. Let us first prove that $\mathbf{W}$ is a global minimizer of L if and only if $\mathbf{W}$ is a first-order critical point of L associated with $\mathcal{S}_{max}$. From

Lemma 5, we have

$$U_{\mathcal{S}_{max}} U_{\mathcal{S}_{max}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \in \underset{\substack{R \in \mathbb{R}^{d_y \times d_x} \\ \mathrm{rk}(R) \leq r_{max}}}{\arg\min} \|RX - Y\|^2 .$$

Let $\mathbf{W}$ be a first-order critical point associated with $\mathcal{S}_{max}$ (note that from Proposition 2, such $\mathbf{W}$ exist). We have $W_H \cdots W_1 = U_{\mathcal{S}_{max}} U_{\mathcal{S}_{max}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$, hence, for all $\mathbf{W}' = (W'_H, \ldots, W'_1)$, since $\mathrm{rk}(W'_H \cdots W'_1) \leq r_{max}$, we have

$$L(\mathbf{W}') \geq \min_{\substack{R \in \mathbb{R}^{d_y \times d_x} \\ \mathrm{rk}(R) \leq r_{max}}} \|RX - Y\|^2 = \|W_H \cdots W_1 X - Y\|^2 = L(\mathbf{W}) .$$

As a consequence, $\mathbf{W}$ is a global minimizer of L.
Conversely, if $\mathbf{W}$ is a global minimizer of L, then $\mathbf{W}$ is a first-order critical point of L. From Proposition 1, there exist $\mathcal{S} \subset [\![1, d_y]\!]$ of size $r \in [\![0, r_{max}]\!]$ such that $W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1}$, and we have $L(\mathbf{W}) = \mathrm{tr}(\Sigma_{YY}) - \sum_{i \in \mathcal{S}} \lambda_i$. But we have from Assumption $\mathcal{H}$, $\lambda_1 > \ldots > \lambda_{d_y}$, and, since Σ is invertible (see Lemma 4), then $\lambda_{d_y} > 0$. Therefore, using Proposition 2, $\mathbf{W}$ is a global minimizer of L implies that $\mathcal{S} = [\![1, r_{max}]\!] = \mathcal{S}_{max}$. Hence, $\mathbf{W}$ is a global minimizer of L if and only if $\mathbf{W}$ is a first-order critical point of L associated with $\mathcal{S}_{max}$.
Let us now prove Proposition 7.
Let $\mathbf{W} = (W_H, \ldots, W_1)$ be a first-order critical point associated with $\mathcal{S}_{max} = [\![1, r_{max}]\!]$. Using Proposition 5, there exist invertible matrices $D_{H-1} \in \mathbb{R}^{d_{H-1} \times d_{H-1}}, \ldots, D_1 \in \mathbb{R}^{d_1 \times d_1}$, and matrices $Z_H \in \mathbb{R}^{(d_y - r_{max}) \times (d_{H-1} - r_{max})}$, $Z_h \in \mathbb{R}^{(d_h - r_{max}) \times (d_{h-1} - r_{max})}$ for $h \in [\![2, H-1]\!]$, and $Z_1 \in \mathbb{R}^{(d_1 - r_{max}) \times d_x}$ such that:

$$\begin{aligned}
W_H &= [U_{\mathcal{S}_{max}}, U_{Q_{max}} Z_H] D_{H-1}^{-1} \\
W_1 &= D_1 \begin{bmatrix} U_{\mathcal{S}_{max}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \\
W_h &= D_h \begin{bmatrix} I_{r_{max}} & 0 \\ 0 & Z_h \end{bmatrix} D_{h-1}^{-1} \qquad \forall h \in [\![2, H-1]\!] .
\end{aligned}$$

Conversely, consider matrices D_h, for $h \in [\![1, H-1]\!]$ and Z_h, for $h \in [\![1, H]\!]$ as in Proposition 7, and

$$\begin{aligned}
W_H &= [U_{\mathcal{S}_{max}}, U_{Q_{max}} Z_H] D_{H-1}^{-1} \\
W_1 &= D_1 \begin{bmatrix} U_{\mathcal{S}_{max}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \\
W_h &= D_h \begin{bmatrix} I_{r_{max}} & 0 \\ 0 & Z_h \end{bmatrix} D_{h-1}^{-1} \qquad \forall h \in [\![2, H-1]\!] .
\end{aligned}$$

Since $|\mathcal{S}_{max}| = r_{max}$, using Proposition 6, we have that $\mathbf{W}$ is a first-order critical point associated with $\mathcal{S}_{max}$. This concludes the proof. □

3.D Global minimizers and simple strict saddle points (Proof of Proposition 8)

Recall that $r_{max} = \min(d_H, \dots, d_0)$.

Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L associated with $\mathcal{S}$ of size $r = \mathrm{rk}(W_H \cdots W_1) \leq r_{max}$.

Case 1: $\mathcal{S} = [\![1, r_{max}]\!] = \mathcal{S}_{max}$. In this case, using Lemma 5,

$$W_H \cdots W_1 = U_{\mathcal{S}_{max}} U_{\mathcal{S}_{max}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \in \underset{\substack{R \in \mathbb{R}^{d_y \times d_x} \\ \mathrm{rk}(R) \leq r_{max}}}{\arg\min} \|RX - Y\|^2 .$$

Moreover, for all $\mathbf{W}' = (W'_H, \dots, W'_1)$, since $\mathrm{rk}(W'_H \cdots W'_1) \leq r_{max}$, we have

$$L(\mathbf{W}') \geq \min_{\substack{R \in \mathbb{R}^{d_y \times d_x} \\ \mathrm{rk}(R) \leq r_{max}}} \|RX - Y\|^2 = \|W_H \cdots W_1 X - Y\|^2 = L(\mathbf{W}) .$$

As a consequence, $\mathbf{W}$ is a global minimizer of L.

Case 2: In order to prove the two remaining statements, we assume that $\mathcal{S} \neq [\![1, r]\!]$ with $0 < r \leq r_{max}$, and show that $\mathbf{W}$ is not a second-order critical point.
To do this we will find $\mathbf{W}' = (W'_H, \dots, W'_1)$ such that $c_2(\mathbf{W}, \mathbf{W}') < 0$ (see Lemma 1). More precisely, we find a linear trajectory of the form $W_h(t) = W_h + tW'_h$ such that the second-order coefficient of the asymptotic expansion of $L((W_h(t))_{h=1..H})$ around $t = 0$ is negative. This proves that $\mathbf{W}$ is not a second-order critical point.

Since $\mathcal{S} \neq [\![1, r]\!]$, and the eigenvalues $(\lambda_k)_{k \in [\![1, d_y]\!]}$ are distinct and in decreasing order (see Section 3.2), there exist $j \in \mathcal{S}$ and $i \notin \mathcal{S}$ such that

$$\lambda_i > \lambda_j . \tag{3.D.1}$$

We denote by $\mathcal{S} = \{i_1, \dots, i_r\}$, hence there exists $g \in [\![1, r]\!]$ such that $j = i_g$.
Note that,

$$U_{\mathcal{S}} = U \sum_{k=1}^{r} E_{i_k, k}$$

where $E_{l,k} \in \mathbb{R}^{d_y \times r}$ is the matrix whose entries are all 0 except the one in position (l, k) which is equal to 1.
Denote by U_t the matrix formed by replacing in $U_{\mathcal{S}}$ the column corresponding to u_j by $u_j + tu_i$. More precisely, set

$$U_t = U_{\mathcal{S}} + tUE_{i,g} .$$

Set $V = UE_{i,g} \in \mathbb{R}^{d_y \times r}$ and

$$V_t = \sum_{k=1}^{r} E_{i_k,k} + tE_{i,g} \in \mathbb{R}^{d_y \times r}. \tag{3.D.2}$$

Hence we have

$$U_t = U_{\mathcal{S}} + tV = UV_t \ . \tag{3.D.3}$$

Considering $D \in \mathbb{R}^{d_1 \times d_1}$ as provided by Lemma 10, we set

$$\begin{cases} W_1' = D^{-1} \begin{bmatrix} V^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0_{(d_1-r)\times d_x} \end{bmatrix} \\ W_h' = 0 \qquad \forall h \in [\![2, H-1]\!] \\ W_H' = V U_{\mathcal{S}}^T W_H \ . \end{cases}$$

and for all $h \in [\![1, H]\!]$, $W_h(t) = W_h + tW_h'$. Note that

$$W_H(t) = W_H + tW_H' = (I_{d_y} + tVU_{\mathcal{S}}^T)W_H \ ,$$

and therefore

$$K(t) := W_H(t) \cdots W_2(t) = (I_{d_y} + tVU_{\mathcal{S}}^T)K \ ,$$

where $K = W_H \cdots W_2$. Using Lemma 10, there exists $M \in \mathbb{R}^{d_1 \times d_x}$ satisfying $KM = 0$ such that

$$W_1 = D^{-1} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0_{(d_1-r)\times d_x} \end{bmatrix} + M \ .$$

Hence,

$$W_1(t) = D^{-1} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0_{(d_1-r)\times d_x} \end{bmatrix} + M + tW_1' = D^{-1} \begin{bmatrix} (U_{\mathcal{S}}^T + tV^T) \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0_{(d_1-r)\times d_x} \end{bmatrix} + M \ ,$$

where $M \in \mathbb{R}^{d_1 \times d_x}$ is such that $KM = 0$. Therefore

$$\begin{aligned} W_t :&= W_H(t) \cdots W_1(t) \\ &= K(t) W_1(t) \\ &= (I_{d_y} + tVU_{\mathcal{S}}^T) \left(KD^{-1} \begin{bmatrix} (U_{\mathcal{S}}^T + tV^T) \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0_{(d_1-r)\times d_x} \end{bmatrix} + KM \right) \ . \end{aligned}$$

From Lemma 10, using that $KM = 0$ and $K = [U_{\mathcal{S}} \quad 0_{d_y \times (d_1-r)}]D$, this becomes

$$\begin{aligned} W_t &= (I_{d_y} + tVU_{\mathcal{S}}^T)[U_{\mathcal{S}} \quad 0_{d_y \times (d_1-r)}] D D^{-1} \begin{bmatrix} (U_{\mathcal{S}}^T + tV^T) \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0_{(d_1-r)\times d_x} \end{bmatrix} \\ &= (I_{d_y} + tVU_{\mathcal{S}}^T) U_{\mathcal{S}} (U_{\mathcal{S}}^T + tV^T) \Sigma_{YX} \Sigma_{XX}^{-1} \ . \end{aligned}$$

Using that $U_{\mathcal{S}}^T U_{\mathcal{S}} = I_r$ (see Lemma 6), we obtain

$$W_t = (U_{\mathcal{S}} + tV)(U_{\mathcal{S}}^T + tV^T)\Sigma_{YX}\Sigma_{XX}^{-1} = U_t U_t^T \Sigma_{YX}\Sigma_{XX}^{-1} \,. \tag{3.D.4}$$

Recall that our goal is to show that the asymptotic expansion of (3.D.5) around $t = 0$ has a negative second-order coefficient. We calculate

$$\begin{aligned} L((W_h(t))_{h=1..H}) &= \|W_t X - Y\|^2 \\ &= \operatorname{tr}\left(W_t \Sigma_{XX} W_t^T\right) - 2\operatorname{tr}(W_t \Sigma_{XY}) + \operatorname{tr}(\Sigma_{YY}) \,. \end{aligned} \tag{3.D.5}$$

Let us simplify $\operatorname{tr}(W_t \Sigma_{XX} W_t^T)$ first. Using (3.D.4), we have

$$W_t \Sigma_{XX} W_t^T = U_t U_t^T \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XX}\Sigma_{XX}^{-1}\Sigma_{XY} U_t U_t^T = U_t U_t^T \Sigma U_t U_t^T \,.$$

Using (3.D.3), $U^T U = I_{d_y}$, $\Sigma = U\Lambda U^T$ and the cyclic property of the trace, we obtain

$$\operatorname{tr}\left(W_t \Sigma_{XX} W_t^T\right) = \operatorname{tr}\left(U V_t V_t^T U^T U \Lambda U^T U V_t V_t^T U^T\right) = \operatorname{tr}\left(V_t V_t^T \Lambda V_t V_t^T\right) = \operatorname{tr}\left(\left(V_t V_t^T\right)^2 \Lambda\right) \,.$$

We define $(\overline{E}_{k,l})_{k=1..d_y, l=1..d_y}$ the canonical basis of $\mathbb{R}^{d_y \times d_y}$. More precisely, $\overline{E}_{k,l} \in \mathbb{R}^{d_y \times d_y}$ has all its entries equal to 0, except a 1 at position (k,l). Note that for all $a, c \in [\![1, d_y]\!]$ and $b, d \in [\![1, r]\!]$

$$E_{a,b} E_{c,d}^T = \delta_{b,d} \overline{E}_{a,c} \,,$$

where $\delta_{b,d}$ equals 1 if $b = d$ and 0 otherwise. Using the definition of V_t in (3.D.2) and $j = i_g$, for $g \in [\![1, r]\!]$, we have

$$\begin{aligned} V_t V_t^T &= \left(\sum_{k=1}^r E_{i_k,k} + tE_{i,g}\right)\left(\sum_{k'=1}^r E_{i_{k'},k'}^T + tE_{i,g}^T\right) \\ &= \left(\sum_{k=1}^r \overline{E}_{i_k,i_k}\right) + t\overline{E}_{i_g,i} + t\overline{E}_{i,i_g} + t^2\overline{E}_{i,i} \\ &= \left(\sum_{k\in\mathcal{S}} \overline{E}_{k,k}\right) + t\overline{E}_{j,i} + t\overline{E}_{i,j} + t^2\overline{E}_{i,i} \,. \end{aligned} \tag{3.D.6}$$

We also have for all $a, b, c, d \in [\![1, d_y]\!]$

$$\overline{E}_{a,b}\overline{E}_{c,d} = \delta_{b,c}\overline{E}_{a,d} \,.$$

Recalling that $j \in \mathcal{S}$ and $i \notin \mathcal{S}$, we obtain

$$(V_t V_t^T)^2 = \left(\left(\sum_{k\in\mathcal{S}} \overline{E}_{k,k}\right) + t\overline{E}_{j,i} + t\overline{E}_{i,j} + t^2\overline{E}_{i,i}\right)\left(\left(\sum_{k'\in\mathcal{S}} \overline{E}_{k',k'}\right) + t\overline{E}_{j,i} + t\overline{E}_{i,j} + t^2\overline{E}_{i,i}\right)$$

$$
\begin{aligned}
&= \left(\left(\sum_{k\in\mathcal{S}} \overline{E}_{k,k}\right) + t\overline{E}_{j,i} + 0 + 0\right) + (0 + 0 + t^2\overline{E}_{j,j} + t^3\overline{E}_{j,i}) \\
&\quad + (t\overline{E}_{i,j} + t^2\overline{E}_{i,i} + 0 + 0) + (0 + 0 + t^3\overline{E}_{i,j} + t^4\overline{E}_{i,i}) \\
&= \left(\sum_{k\in\mathcal{S}} \overline{E}_{k,k}\right) + t^2(1+t^2)\overline{E}_{i,i} + t^2\overline{E}_{j,j} + t(1+t^2)\overline{E}_{i,j} + t(1+t^2)\overline{E}_{j,i} \ .
\end{aligned}
$$

Finally, since for all $a, b \in [\![1, d_y]\!]$

$$\overline{E}_{a,b}\Lambda = \lambda_b \overline{E}_{a,b} \tag{3.D.7}$$

we have

$$\operatorname{tr}\left(W_t \Sigma_{XX} W_t^T\right) = \operatorname{tr}\left(\left(V_t V_t^T\right)^2 \Lambda\right) = \sum_{k\in\mathcal{S}} \lambda_k + t^2(1+t^2)\lambda_i + t^2\lambda_j \ . \tag{3.D.8}$$

Coming back to (3.D.5), we calculate the other term $\operatorname{tr}(W_t\Sigma_{XY})$. Using (3.D.4), (3.D.3) and $\Sigma = U\Lambda U^T$, we obtain

$$\operatorname{tr}(W_t\Sigma_{XY}) = \operatorname{tr}(U_t U_t^T \Sigma) = \operatorname{tr}(U V_t V_t^T U^T U \Lambda U^T) = \operatorname{tr}(V_t V_t^T \Lambda) \ .$$

Combining with (3.D.6) and (3.D.7), we get

$$\operatorname{tr}(W_t\Sigma_{XY}) = \operatorname{tr}(V_t V_t^T \Lambda) = \sum_{k\in\mathcal{S}} \lambda_k + t^2\lambda_i \ . \tag{3.D.9}$$

Finally, substituting (3.D.8) and (3.D.9) in (3.D.5), we have

$$
\begin{aligned}
L((W_h(t))_{h=1..H}) &= \operatorname{tr}(\Sigma_{YY}) + \sum_{k\in\mathcal{S}} \lambda_k + t^2(1+t^2)\lambda_i + t^2\lambda_j - 2\sum_{k\in\mathcal{S}} \lambda_k - 2t^2\lambda_i \\
&= \operatorname{tr}(\Sigma_{YY}) - \sum_{k\in\mathcal{S}} \lambda_k + t^2(\lambda_j - \lambda_i) + \lambda_i t^4 \ .
\end{aligned}
$$

Using Proposition 1 and recalling (3.D.1), we finally get as $t \to 0$,

$$L((W_h(t))_{h=1..H}) = L(\mathbf{W}) + ct^2 + o(t^2) \quad \text{with} \quad c = \lambda_j - \lambda_i < 0 \ .$$

Therefore, we conclude from Lemma 1 that $\mathbf{W} = (W_H, \ldots, W_1)$ is not a second-order critical point.

3.E Strict saddle points with $\mathcal{S} = [\![1, r]\!],\ r < r_{max}$ (Proof of Proposition 9)

We refer the reader to Section 3.4.2, which introduces the 4 cases proved below. Recall that $\mathcal{S} = [\![1, r]\!]$ and we set $Q = [\![1, d_y]\!] \setminus \mathcal{S} = [\![r+1, d_y]\!]$.
In this section, for each vector space $\mathbb{R}^{d_h}$, we will denote by e_m the m-th element of the

canonical basis of $\mathbb{R}^{d_h}$. That is, the entries of $e_m \in \mathbb{R}^{d_h}$ are all equal to 0 except for the m-th coordinate which is equal to 1. The size of e_m will not be ambiguous, once in context, so we do not include it in the notation.
Remark about $r = 0$**:** Using the conventions of Section 3.2, in this case we have $\mathcal{S} = \emptyset$ and $Q = [\![1, d_y]\!]$. Hence $U_{\mathcal{S}}$ is the matrix with no column, $U_Q = U$, and $U_{\mathcal{S}} U_{\mathcal{S}}^T = 0_{d_y \times d_y}$. For example, we still have $I_{d_y} = U_{\mathcal{S}} U_{\mathcal{S}}^T + U_Q U_Q^T$. We can easily follow the proofs below with these conventions and see that the result still holds.

3.E.1 1st case: $i \in [\![2, H-1]\!]$ and $j = 1$

In this case, the two complementary blocks are $\Sigma_{XY} W_H \cdots W_{i+1}$ and $W_{i-1} \cdots W_2$. Recall that $\mathcal{S} = [\![1, r]\!]$ and $r < r_{max} = \min(d_H, \dots, d_0)$. Note that $\mathrm{rk}(\Sigma_{XY} W_H \cdots W_{i+1}) = \mathrm{rk}(W_H \cdots W_{i+1})$ because Σ_{XY} is of full column rank (see Assumption $\mathcal{H}$, in Section 3.2) . Since the pivot (i, j) is not tightened, using Proposition 3, we have

$$\begin{cases} \mathrm{rk}(W_H \cdots W_{i+1}) > r \\ \mathrm{rk}(W_{i-1} \cdots W_2) > r. \end{cases} \tag{3.E.1}$$

Let us first show that there exists $k \in [\![r+1, d_y]\!]$ and $l \in [\![1, d_i]\!]$ such that

$$U_k^T (W_H \cdots W_{i+1})_{.,l} \neq 0 \ . \tag{3.E.2}$$

Indeed, assume by contradiction that for all $k \in [\![r+1, d_y]\!]$ and $l \in [\![1, d_i]\!]$ we have

$$U_k^T (W_H \cdots W_{i+1})_{.,l} = 0.$$

Recalling that $Q = [\![1, d_y]\!] \setminus \mathcal{S} = [\![r+1, d_y]\!]$, we obtain $U_Q^T W_H \cdots W_{i+1} = 0$. Using from Lemma 6 that $I_{d_y} = U_{\mathcal{S}} U_{\mathcal{S}}^T + U_Q U_Q^T$, we have

$$\begin{aligned} W_H \cdots W_{i+1} &= (U_{\mathcal{S}} U_{\mathcal{S}}^T + U_Q U_Q^T) W_H \cdots W_{i+1} \\ &= U_{\mathcal{S}} U_{\mathcal{S}}^T W_H \cdots W_{i+1}. \end{aligned}$$

Therefore,

$$\mathrm{rk}(W_H \cdots W_{i+1}) = \mathrm{rk}(U_{\mathcal{S}} U_{\mathcal{S}}^T W_H \cdots W_{i+1}).$$

The latter is impossible since $\mathrm{rk}(U_{\mathcal{S}} U_{\mathcal{S}}^T W_H \cdots W_{i+1}) \leq |\mathcal{S}| = r$, which is not compatible with (3.E.1). Therefore (3.E.2) holds.
Since $\mathbf{W}$ is a first-order critical point, using Lemma 10, there exists an invertible matrix $D \in \mathbb{R}^{d_1 \times d_1}$ such that

$$W_H \cdots W_2 = [U_{\mathcal{S}}, 0_{d_y \times (d_1 - r)}] D \tag{3.E.3}$$

and since $\mathbf{W}$ is associated with $\mathcal{S}$, we have

$$W_H \cdots W_1 = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \ . \tag{3.E.4}$$

Using (3.E.1) and D invertible, we have $rk(W_{i-1} \cdots W_2 D^{-1}) = rk(W_{i-1} \cdots W_2) > r$. Hence there exists $g \in [\![r+1, d_1]\!]$ such that

$$(W_{i-1} \cdots W_2 D^{-1})_{.,g} \neq 0 \, .$$

Therefore, there exists $a \in \mathbb{R}^{d_{i-1}}$ such that

$$a^T (W_{i-1} \cdots W_2 D^{-1})_{.,g} = 1 \, . \tag{3.E.5}$$

Recall that k, l satisfy (3.E.2). We define $\mathbf{W}'_\beta = (W_H'^\beta, \dots, W_1'^\beta)$ by

$$\begin{cases} W_i'^\beta = \beta W_i' = \beta e_l a^T \in \mathbb{R}^{d_i \times d_{i-1}}, \text{ where } e_l \in \mathbb{R}^{d_i} \\ W_1'^\beta = W_1' = D^{-1} e_g U_k^T \Sigma_{YX} \Sigma_{XX}^{-1} \in \mathbb{R}^{d_1 \times d_x}, \text{ where } e_g \in \mathbb{R}^{d_1} \\ W_h'^\beta = 0 \quad \forall h \in [\![2, H]\!] \setminus \{i\} \end{cases}$$

We set $\mathbf{W}^\beta(t) = (W_H^\beta(t), \dots, W_1^\beta(t))$ such that $W_h^\beta(t) = W_h + t W_h'^\beta$ for $h \in [\![1, H]\!]$. We have

$$\begin{aligned} W^\beta(t) :&= W_H^\beta(t) \cdots W_1^\beta(t) \\ &= W_H \cdots W_{i+1}(W_i + t\beta W_i') W_{i-1} \cdots W_2 (W_1 + t W_1') \\ &= W_H \cdots W_1 + t(\beta W_H \cdots W_{i+1} W_i' W_{i-1} \cdots W_1 + W_H \cdots W_2 W_1') \\ &\quad + \beta t^2 W_H \cdots W_{i+1} W_i' W_{i-1} \cdots W_2 W_1' \, . \end{aligned}$$

Using (3.E.3) and (3.E.4), we obtain

$$\begin{aligned} W^\beta(t) = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} &+ t(\beta W_H \cdots W_{i+1} W_i' W_{i-1} \cdots W_1 + [U_{\mathcal{S}}, 0] D D^{-1} e_g U_k^T \Sigma_{YX} \Sigma_{XX}^{-1}) \\ &+ \beta t^2 (W_H \cdots W_{i+1})_{.,l} a^T (W_{i-1} \cdots W_2 D^{-1})_{.,g} U_k^T \Sigma_{YX} \Sigma_{XX}^{-1} \, . \end{aligned}$$

Using (3.E.5) and $g \in [\![r+1, d_1]\!]$, we have

$$W^\beta(t) = U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} + t\beta W_H \cdots W_{i+1} W_i' W_{i-1} \cdots W_1 + \beta t^2 (W_H \cdots W_{i+1})_{.,l} U_k^T \Sigma_{YX} \Sigma_{XX}^{-1} \, .$$

Denoting $N = W_H \cdots W_{i+1} W_i' W_{i-1} \cdots W_1$, we have

$$\begin{aligned} L(\mathbf{W}^\beta(t)) &= \|W^\beta(t) X - Y\|^2 \\ &= \|U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X - Y + t\beta N X + \beta t^2 (W_H \cdots W_{i+1})_{.,l} U_k^T \Sigma_{YX} \Sigma_{XX}^{-1} X\|^2 \, . \end{aligned}$$

Expanding the square, the second-order term $c_2(\mathbf{W}, \mathbf{W}'_\beta) t^2$ has a coefficient equal to

$$\begin{aligned} c_2(\mathbf{W}, \mathbf{W}'_\beta) &= \beta^2 \|NX\|^2 + 2\beta \operatorname{tr}((W_H \cdots W_{i+1})_{.,l} U_k^T \Sigma_{YX} \Sigma_{XX}^{-1} X X^T \Sigma_{XX}^{-1} \Sigma_{XY} U_{\mathcal{S}} U_{\mathcal{S}}^T) \\ &\quad - 2\beta \operatorname{tr}((W_H \cdots W_{i+1})_{.,l} U_k^T \Sigma_{YX} \Sigma_{XX}^{-1} X Y^T) \\ &= \beta^2 \|NX\|^2 + 2\beta \operatorname{tr}((W_H \cdots W_{i+1})_{.,l} U_k^T \Sigma U_{\mathcal{S}} U_{\mathcal{S}}^T) - 2\beta \operatorname{tr}((W_H \cdots W_{i+1})_{.,l} U_k^T \Sigma) \\ &= \beta^2 \|NX\|^2 - 2\beta \lambda_k U_k^T (W_H \cdots W_{i+1})_{.,l} \, , \end{aligned}$$

where the last equality follows from Lemma 7 and $k \notin \mathcal{S}$, and $U^T\Sigma = \Lambda U^T$ and the cyclic property of the trace.
Using Lemma 4 and (3.E.2), we have $\lambda_k U_k^T (W_H \cdots W_{i+1})_{.,l} \neq 0$, hence we can choose β according to (3.4.2), such that $c_2(\mathbf{W}, \mathbf{W}'_\beta) < 0$. Therefore, $\mathbf{W}$ is not a second-order critical point.

3.E.2 2nd case: $i = H$ and $j = 1$

In this case, the two complementary blocks are Σ_{XY} and $W_{H-1} \cdots W_2$. We follow again the same lines as above. Since the pivot (i, j) is not tightened, using Proposition 3, we have

$$\mathrm{rk}(W_{H-1} \cdots W_2) > r \; . \tag{3.E.6}$$

Again, since $\mathbf{W}$ is a first-order critical point, using Lemma 10, there exists an invertible matrix $D \in \mathbb{R}^{d_1 \times d_1}$ such that

$$W_H \cdots W_2 = [U_\mathcal{S}, 0_{d_y \times (d_1 - r)}]D \tag{3.E.7}$$

and since $\mathbf{W}$ is associated with $\mathcal{S}$, we have

$$W_H \cdots W_1 = U_\mathcal{S} U_\mathcal{S}^T \Sigma_{YX} \Sigma_{XX}^{-1} \; . \tag{3.E.8}$$

Using (3.E.6) and D invertible, we have $rk(W_{H-1} \cdots W_2 D^{-1}) = rk(W_{H-1} \cdots W_2) > r$. Hence there exists $g \in [\![r+1, d_1]\!]$ such that

$$(W_{i-1} \cdots W_2 D^{-1})_{.,g} \neq 0 \; .$$

Therefore, there exists $a \in \mathbb{R}^{d_{H-1}}$ such that

$$a^T (W_{H-1} \cdots W_2 D^{-1})_{.,g} = 1 \; . \tag{3.E.9}$$

We define $\mathbf{W}'_\beta = (W_H'^\beta, \dots, W_1'^\beta)$ by

$$\begin{cases} W_H'^\beta = \beta W'_H = \beta U_{r+1} a^T \in \mathbb{R}^{d_y \times d_{H-1}} \\ W_1'^\beta = W'_1 = D^{-1} e_g U_{r+1}^T \Sigma_{YX} \Sigma_{XX}^{-1} \in \mathbb{R}^{d_1 \times d_x}, \text{ where } e_g \in \mathbb{R}^{d_1} \\ W_h'^\beta = 0 \quad \forall h \in [\![2, H-1]\!] \; . \end{cases}$$

We set $\mathbf{W}^\beta(t) = (W_H^\beta(t), \dots, W_1^\beta(t))$ such that $W_h^\beta(t) = W_h + tW_h'^\beta$, for all $h \in [\![1, H]\!]$. We have

$$\begin{aligned} W^\beta(t) :&= W_H^\beta(t) \cdots W_1^\beta(t) \\ &= (W_H + t\beta W'_H) W_{H-1} \cdots W_2 (W_1 + tW'_1) \\ &= W_H \cdots W_1 + t(\beta W'_H W_{H-1} \cdots W_1 + W_H \cdots W_2 W'_1) + \beta t^2 W'_H W_{H-1} \cdots W_2 W'_1 \; . \end{aligned}$$

Using (3.E.7) and (3.E.8), then (3.E.9) and $g \in [\![r+1, d_1]\!]$, we obtain

$$\begin{aligned} W^\beta(t) &= U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} + t(\beta W_H' W_{H-1} \cdots W_1 + [U_{\mathcal{S}}, 0] D D^{-1} e_g U_{r+1}^T \Sigma_{YX} \Sigma_{XX}^{-1}) \\ &\quad + \beta t^2 U_{r+1} a^T (W_{H-1} \cdots W_2 D^{-1})_{.,g} U_{r+1}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ &= U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} + t\beta W_H' W_{H-1} \cdots W_1 + \beta t^2 U_{r+1} U_{r+1}^T \Sigma_{YX} \Sigma_{XX}^{-1} \ . \end{aligned}$$

Denoting by $N = W_H' W_{H-1} \cdots W_1$, we have

$$\begin{aligned} L(\mathbf{W}^\beta(t)) &= \|W^\beta(t) X - Y\|^2 \\ &= \|U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X - Y + t\beta N X + \beta t^2 U_{r+1} U_{r+1}^T \Sigma_{YX} \Sigma_{XX}^{-1} X\|^2 \ . \end{aligned}$$

As previously, expanding the square, we can see that the second-order coefficient $c_2(\mathbf{W}, \mathbf{W}'_\beta)$ of the polynomial $L(\mathbf{W}^\beta(t))$ is given by

$$\begin{aligned} c_2(\mathbf{W}, \mathbf{W}'_\beta) &= \beta^2 \|NX\|^2 + 2\beta \operatorname{tr}(U_{r+1} U_{r+1}^T \Sigma_{YX} \Sigma_{XX}^{-1} X X^T \Sigma_{XX}^{-1} \Sigma_{XY} U_{\mathcal{S}} U_{\mathcal{S}}^T) \\ &\quad - 2\beta \operatorname{tr}(U_{r+1} U_{r+1}^T \Sigma_{YX} \Sigma_{XX}^{-1} X Y^T) \\ &= \beta^2 \|NX\|^2 + 2\beta \operatorname{tr}(U_{r+1} U_{r+1}^T \Sigma U_{\mathcal{S}} U_{\mathcal{S}}^T) - 2\beta \operatorname{tr}(U_{r+1} U_{r+1}^T \Sigma) \ . \end{aligned}$$

Using the cyclic property of the trace, $U_{\mathcal{S}}^T U_{r+1} = 0$ (see Lemma 6), and $\Sigma U_{r+1} = \lambda_{r+1} U_{r+1}$, we obtain

$$\begin{aligned} c_2(\mathbf{W}, \mathbf{W}'_\beta) &= \beta^2 \|NX\|^2 - 2\beta \lambda_{r+1} U_{r+1}^T U_{r+1} \\ &= \beta^2 \|NX\|^2 - 2\beta \lambda_{r+1} \ . \end{aligned}$$

Using Lemma 4, we have $\lambda_{r+1} \neq 0$, hence we can choose β according to (3.4.2) such that $c_2(\mathbf{W}, \mathbf{W}'_\beta) < 0$. Therefore $\mathbf{W}$ is not a second-order critical point.

3.E.3 3rd case: $i = H$ and $j \in [\![2, H-1]\!]$

In this case, the two complementary blocks are $W_{j-1} \cdots W_1 \Sigma_{XY}$ and $W_{H-1} \cdots W_{j+1}$. We follow again the same lines as above. Since the pivot (i, j) is not tightened, using Proposition 3, we have

$$\begin{cases} \operatorname{rk}(W_{H-1} \cdots W_{j+1}) > r \\ \operatorname{rk}(W_{j-1} \cdots W_1 \Sigma_{XY}) > r \ . \end{cases} \tag{3.E.10}$$

Let us first show that there exist $k \in [\![r+1, d_y]\!]$ and $l \in [\![1, d_{j-1}]\!]$ such that

$$(W_{j-1} \cdots W_1)_{l,.} \Sigma_{XY} U_k \neq 0 \ . \tag{3.E.11}$$

Indeed, assume by contradiction that for all $k \in [\![r+1, d_y]\!]$ and $l \in [\![1, d_{j-1}]\!]$ we have

$$(W_{j-1} \cdots W_1)_{l,.} \Sigma_{XY} U_k = 0.$$

Recalling that $Q = [\![1, d_y]\!] \setminus \mathcal{S} = [\![r+1, d_y]\!]$, we obtain $W_{j-1} \cdots W_1 \Sigma_{XY} U_Q = 0$, and using, from Lemma 6, that $I_{d_y} = U_{\mathcal{S}} U_{\mathcal{S}}^T + U_Q U_Q^T$, we have

$$\begin{aligned} W_{j-1} \cdots W_1 \Sigma_{XY} &= W_{j-1} \cdots W_1 \Sigma_{XY} (U_{\mathcal{S}} U_{\mathcal{S}}^T + U_Q U_Q^T) \\ &= W_{j-1} \cdots W_1 \Sigma_{XY} U_{\mathcal{S}} U_{\mathcal{S}}^T \ . \end{aligned}$$

Therefore,

$$\mathrm{rk}(W_{j-1} \cdots W_1 \Sigma_{XY}) = \mathrm{rk}(W_{j-1} \cdots W_1 \Sigma_{XY} U_{\mathcal{S}} U_{\mathcal{S}}^T).$$

The latter is impossible since $\mathrm{rk}(W_{j-1} \cdots W_1 \Sigma_{XY} U_{\mathcal{S}} U_{\mathcal{S}}^T) \leq |\mathcal{S}| = r$ is not compatible with (3.E.10). Therefore (3.E.11) holds.

We know that $\mathrm{rk}(W_H \cdots W_{j+1}) \geq \mathrm{rk}(W_H \cdots W_1) = r$. Therefore, depending on the value of $\mathrm{rk}(W_H \cdots W_{j+1})$, we distinguish two situations: either $\mathrm{rk}(W_H \cdots W_{j+1}) > r$ or $\mathrm{rk}(W_H \cdots W_{j+1}) = r$.

When $\mathrm{rk}(W_H \cdots W_{j+1}) > r$, since Σ_{XY} is of full column rank, we have $\mathrm{rk}(\Sigma_{XY} W_H \cdots W_{j+1}) = \mathrm{rk}(W_H \cdots W_{j+1}) > r$. Also, using (3.E.10), we have $\mathrm{rk}(W_{j-1} \cdots W_2) \geq \mathrm{rk}(W_{j-1} \cdots W_1 \Sigma_{XY}) > r$. Hence, in this case, the pivot $(j, 1)$ is not tightened either. We have already proved in Section 3.E.1 (beware that the pivot is denoted $(i, 1)$, not $(j, 1)$, in Section 3.E.1) that, when such a pivot is not tightened, $\mathbf{W}$ is not a second-order critical point. This concludes the proof in the case $\mathrm{rk}(W_H \cdots W_{j+1}) > r$.

In the rest of the section we assume that $\mathrm{rk}(W_H \cdots W_{j+1}) = r$.

Using (3.E.10), we have $\mathrm{rk}(W_{H-1} \cdots W_{j+1}) > r = \mathrm{rk}(W_H \cdots W_{j+1})$. Applying the rank-nullity theorem we obtain

$$\mathrm{Ker}(W_{H-1} \cdots W_{j+1}) \subsetneq \mathrm{Ker}(W_H \cdots W_{j+1}).$$

Therefore there exists $b \in \mathbb{R}^{d_j}$ such that

$$\begin{cases} b \in \mathrm{Ker}(W_H \cdots W_{j+1}) \\ b \notin \mathrm{Ker}(W_{H-1} \cdots W_{j+1}) \ . \end{cases} \tag{3.E.12}$$

Hence, there also exists $a \in \mathbb{R}^{d_{H-1}}$ such that

$$a^T W_{H-1} \cdots W_{j+1} b = 1 \ . \tag{3.E.13}$$

Recall that k, l satisfy (3.E.11). We define $\mathbf{W}'_\beta = (W_H'^\beta, \ldots, W_1'^\beta)$ by

$$\begin{cases} W_H'^\beta = \beta W'_H = \beta U_k a^T \in \mathbb{R}^{d_y \times d_{H-1}} \\ W_j'^\beta = W'_j = b e_l^T \in \mathbb{R}^{d_j \times d_{j-1}}, \text{ where } e_l \in \mathbb{R}^{d_{j-1}} \\ W_h'^\beta = 0 \quad \forall h \in [\![1, H]\!] \setminus \{i, j\} \end{cases}$$

We set $\mathbf{W}^\beta(t) = (W_H^\beta(t), \ldots, W_1^\beta(t))$ such that $W_h^\beta(t) = W_h + t W_h'^\beta$ for $h \in [\![1, H]\!]$. We have

$$W^\beta(t) := W_H^\beta(t) \cdots W_1^\beta(t)$$

$$
\begin{aligned}
&= (W_H + t\beta W'_H)W_{H-1}\cdots W_{j+1}(W_j + tW'_j)W_{j-1}\cdots W_1 \\
&= W_H\cdots W_1 + t(\beta W'_H W_{H-1}\cdots W_1 + W_H\cdots W_{j+1}W'_j W_{j-1}\cdots W_1) \\
&\quad + t^2\beta W'_H\cdots W_{j+1}W'_j W_{j-1}\cdots W_1 \ .
\end{aligned}
$$

Using Proposition 1 and the definition of $\mathbf{W}'_\beta$ above, we obtain

$$
\begin{aligned}
W^\beta(t) &= U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1} + t(\beta W'_H W_{H-1}\cdots W_1 + W_H\cdots W_{j+1}be_l^T W_{j-1}\cdots W_1) \\
&\quad + \beta t^2 U_k a^T W_{H-1}\cdots W_{j+1}b(W_{j-1}\cdots W_1)_{l,\cdot} \\
&= U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1} + t\beta W'_H W_{H-1}\cdots W_1 + \beta t^2 U_k(W_{j-1}\cdots W_1)_{l,\cdot} \ ,
\end{aligned}
$$

where the last equality follows from (3.E.12) and (3.E.13) .
Denoting $N = W'_H W_{H-1}\cdots W_1$, we have

$$
\begin{aligned}
L(\mathbf{W}^\beta(t)) &= \|W^\beta(t)X - Y\|^2 \\
&= \|U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X - Y + t\beta NX + \beta t^2 U_k(W_{j-1}\cdots W_1)_{l,\cdot}X\|^2 \ .
\end{aligned}
$$

Using the cyclic property of the trace, and, since $k \notin \mathcal{S}$, $U_{\mathcal{S}}^T U_k = 0$, we get in this case a second-order coefficient equal to

$$
\begin{aligned}
c_2(\mathbf{W}, \mathbf{W}'_\beta) &= \beta^2\|NX\|^2 + 2\beta\,\mathrm{tr}\left(U_k(W_{j-1}\cdots W_1)_{l,\cdot}XX^T\Sigma_{XX}^{-1}\Sigma_{XY}U_{\mathcal{S}}U_{\mathcal{S}}^T\right) \\
&\quad - 2\beta\,\mathrm{tr}(U_k(W_{j-1}\cdots W_1)_{l,\cdot}\Sigma_{XY}) \\
&= \beta^2\|NX\|^2 - 2\beta(W_{j-1}\cdots W_1)_{l,\cdot}\Sigma_{XY}U_k \ .
\end{aligned}
$$

Since from (3.E.11), $(W_{j-1}\cdots W_1)_{l,\cdot}\Sigma_{XY}U_k \neq 0$, we can choose β according to (3.4.2), such that $c_2(\mathbf{W}, \mathbf{W}'_\beta) < 0$. Therefore $\mathbf{W}$ is not a second-order critical point.

3.E.4 4th case: $i, j \in [\![2, H-1]\!]$, with $i > j$

In this case, the two complementary blocks are $W_{j-1}\cdots W_1\Sigma_{XY}W_H\cdots W_{i+1}$ and $W_{i-1}\cdots W_{j+1}$. We follow again the same lines as above. Since the pivot (i, j) is not tightened, using Proposition 3, we have

$$
\begin{cases}
\mathrm{rk}(W_{i-1}\cdots W_{j+1}) > r \\
\mathrm{rk}(W_{j-1}\cdots W_1\Sigma_{XY}W_H\cdots W_{i+1}) > r \ .
\end{cases}
\tag{3.E.14}
$$

Let us first show that there exist $k \in [\![1, d_i]\!]$ and $l \in [\![1, d_{j-1}]\!]$ such that

$$
(W_{j-1}\cdots W_1)_{l,\cdot}\Sigma_{XY}U_Q U_Q^T(W_H\cdots W_{i+1})_{\cdot,k} \neq 0 \ . \tag{3.E.15}
$$

Indeed, assume by contradiction that, for all $k \in [\![1, d_i]\!]$ and $l \in [\![1, d_{j-1}]\!]$, we have

$$
(W_{j-1}\cdots W_1)_{l,\cdot}\Sigma_{XY}U_Q U_Q^T(W_H\cdots W_{i+1})_{\cdot,k} = 0.
$$

Then $W_{j-1}\cdots W_1\Sigma_{XY}U_Q U_Q^T W_H\cdots W_{i+1}=0$, and so, using $I_{d_y}=U_{\mathcal{S}}U_{\mathcal{S}}^T+U_Q U_Q^T$, we would have

$$\begin{aligned}W_{j-1}\cdots W_1\Sigma_{XY}W_H\cdots W_{i+1} &= W_{j-1}\cdots W_1\Sigma_{XY}I_{d_y}W_H\cdots W_{i+1}\\ &= W_{j-1}\cdots W_1\Sigma_{XY}\left(U_{\mathcal{S}}U_{\mathcal{S}}^T+U_Q U_Q^T\right)W_H\cdots W_{i+1}\\ &= W_{j-1}\cdots W_1\Sigma_{XY}U_{\mathcal{S}}U_{\mathcal{S}}^T W_H\cdots W_{i+1}\ .\end{aligned}$$

Therefore,

$$\mathrm{rk}(W_{j-1}\cdots W_1\Sigma_{XY}W_H\cdots W_{i+1})=\mathrm{rk}(W_{j-1}\cdots W_1\Sigma_{XY}U_{\mathcal{S}}U_{\mathcal{S}}^T W_H\cdots W_{i+1}).$$

The latter is impossible since $\mathrm{rk}(W_{j-1}\cdots W_1\Sigma_{XY}U_{\mathcal{S}}U_{\mathcal{S}}^T W_H\cdots W_{i+1})\leq|\mathcal{S}|=r$ is not compatible with (3.E.14). Therefore (3.E.15) holds.

We know that $\mathrm{rk}(W_H\cdots W_{j+1})\geq\mathrm{rk}(W_H\cdots W_1)=r$. Therefore, depending on the value of $\mathrm{rk}(W_H\cdots W_{j+1})$, we distinguish two situations: either $\mathrm{rk}(W_H\cdots W_{j+1})>r$ or $\mathrm{rk}(W_H\cdots W_{j+1})=r$.
When $\mathrm{rk}(W_H\cdots W_{j+1})>r$, since Σ_{XY} is of full column rank, we have $\mathrm{rk}(\Sigma_{XY}W_H\cdots W_{j+1})=\mathrm{rk}(W_H\cdots W_{j+1})>r$. Also, using (3.E.14), we have $\mathrm{rk}(W_{j-1}\cdots W_2)\geq\mathrm{rk}(W_{j-1}\cdots W_1\Sigma_{XY}W_H\cdots W_{i+1})>r$. Hence, in this case, the pivot $(j,1)$ is not tightened either. We have already proved in Section 3.E.1 (beware that the pivot is denoted $(i,1)$, not $(j,1)$, in Section 3.E.1) that, when such a pivot is not tightened, $\mathbf{W}$ is not a second-order critical point. This concludes the proof when $\mathrm{rk}(W_H\cdots W_{j+1})>r$.
In the rest of the section we assume that $\mathrm{rk}(W_H\cdots W_{j+1})=r$.
Using (3.E.14), we have $\mathrm{rk}(W_{i-1}\cdots W_{j+1})>r=\mathrm{rk}(W_H\cdots W_{j+1})$. Applying the rank-nullity theorem, we obtain

$$\mathrm{Ker}(W_{i-1}\cdots W_{j+1})\subsetneq\mathrm{Ker}(W_H\cdots W_{j+1}).$$

Therefore there exists $b\in\mathbb{R}^{d_j}$ such that

$$\begin{cases}b\in\mathrm{Ker}(W_H\cdots W_{j+1})\\ b\notin\mathrm{Ker}(W_{i-1}\cdots W_{j+1})\ .\end{cases}\tag{3.E.16}$$

Hence, there also exists $a\in\mathbb{R}^{d_{i-1}}$ such that

$$a^T W_{i-1}\cdots W_{j+1}b=1\ .\tag{3.E.17}$$

Recall that k,l satisfy (3.E.15). We define $\mathbf{W}'_\beta=(W_H'^\beta,\dots,W_1'^\beta)$ by

$$\begin{cases}W_i'^\beta=\beta W_i'=\beta e_k a^T\in\mathbb{R}^{d_i\times d_{i-1}}\text{ where }e_k\in\mathbb{R}^{d_i}\\ W_j'^\beta=W_j'=be_l^T\in\mathbb{R}^{d_j\times d_{j-1}}\text{ where }e_l\in\mathbb{R}^{d_{j-1}}\\ W_h'^\beta=0\quad\forall h\in[\![1,H]\!]\setminus\{i,j\}\ .\end{cases}$$

We set $\mathbf{W}^\beta(t) = (W_H^\beta(t), \dots, W_1^\beta(t))$ with $W_h^\beta(t) = W_h + tW_h'^\beta$ for all $h \in [\![1, H]\!]$. We have,

$$\begin{aligned}
W^\beta(t) :&= W_H^\beta(t) \cdots W_1^\beta(t) \\
&= W_H \cdots W_{i+1}(W_i + t\beta W_i')W_{i-1} \cdots W_{j+1}(W_j + tW_j')W_{j-1} \cdots W_1 \\
&= W_H \cdots W_1 + t(\beta W_H \cdots W_{i+1}W_i'W_{i-1} \cdots W_1 + W_H \cdots W_{j+1}W_j'W_{j-1} \cdots W_1) \\
&\quad + \beta t^2 W_H \cdots W_{i+1}W_i'W_{i-1} \cdots W_{j+1}W_j'W_{j-1} \cdots W_1 \ .
\end{aligned}$$

Using Proposition 1 and the definition of $\mathbf{W}'_\beta$ above, we obtain

$$\begin{aligned}
W^\beta(t) &= U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1} + t(\beta W_H \cdots W_{i+1}W_i'W_{i-1} \cdots W_1 + W_H \cdots W_{j+1}be_l^T W_{j-1} \cdots W_1) \\
&\quad + \beta t^2 (W_H \cdots W_{i+1})_{\cdot,k} a^T W_{i-1} \cdots W_{j+1} b (W_{j-1} \cdots W_1)_{l,\cdot} \\
&= U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1} + t\beta W_H \cdots W_{i+1}W_i'W_{i-1} \cdots W_1 + \beta t^2 (W_H \cdots W_{i+1})_{\cdot,k}(W_{j-1} \cdots W_1)_{l,\cdot} \ ,
\end{aligned}$$

where the last equality follows from (3.E.16) and (3.E.17) .
Denoting $N = W_H \cdots W_{i+1}W_i'W_{i-1} \cdots W_1$, we have

$$\begin{aligned}
L(\mathbf{W}^\beta(t)) &= \|W^\beta(t)X - Y\|^2 \\
&= \|U_{\mathcal{S}}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X - Y + t\beta NX + \beta t^2 (W_H \cdots W_{i+1})_{\cdot,k}(W_{j-1} \cdots W_1)_{l,\cdot}X\|^2 \ .
\end{aligned}$$

The second-order coefficient of $L(\mathbf{W}^\beta(t))$ is equal to

$$\begin{aligned}
c_2(\mathbf{W}, \mathbf{W}'_\beta) &= \beta^2\|NX\|^2 + 2\beta \operatorname{tr}\left((W_H \cdots W_{i+1})_{\cdot,k}(W_{j-1} \cdots W_1)_{l,\cdot}XX^T\Sigma_{XX}^{-1}\Sigma_{XY}U_{\mathcal{S}}U_{\mathcal{S}}^T\right) \\
&\quad - 2\beta \operatorname{tr}\left((W_H \cdots W_{i+1})_{\cdot,k}(W_{j-1} \cdots W_1)_{l,\cdot}\Sigma_{XY}\right) \\
&= \beta^2\|NX\|^2 + 2\beta \operatorname{tr}\left((W_H \cdots W_{i+1})_{\cdot,k}(W_{j-1} \cdots W_1)_{l,\cdot}\Sigma_{XY}(U_{\mathcal{S}}U_{\mathcal{S}}^T - I_{d_y})\right) \ .
\end{aligned}$$

Using, from Lemma 6, that $U_{\mathcal{S}}U_{\mathcal{S}}^T - I_{d_y} = -U_Q U_Q^T$, and then the cyclic property of the trace, we obtain

$$\begin{aligned}
c_2(\mathbf{W}, \mathbf{W}'_\beta) &= \beta^2\|NX\|^2 - 2\beta \operatorname{tr}\left((W_H \cdots W_{i+1})_{\cdot,k}(W_{j-1} \cdots W_1)_{l,\cdot}\Sigma_{XY}U_Q U_Q^T\right) \\
&= \beta^2\|NX\|^2 - 2\beta (W_{j-1} \cdots W_1)_{l,\cdot}\Sigma_{XY}U_Q U_Q^T (W_H \cdots W_{i+1})_{\cdot,k} \ .
\end{aligned}$$

Since from (3.E.15), $(W_{j-1} \cdots W_1)_{l,\cdot}\Sigma_{XY}U_Q U_Q^T (W_H \cdots W_{i+1})_{\cdot,k} \neq 0$, we can choose β according to (3.4.2) such that $c_2(\mathbf{W}, \mathbf{W}'_\beta) < 0$. Therefore, $\mathbf{W}$ is not a second-order critical point.

3.F Non-strict saddle points

In this section, we prove the results related to non-strict saddle points (see Section 3.4.3).

3.F.1 Proof of Proposition 11

To prove Proposition 11, we show that for any $\mathbf{W}'$, $c_2(\mathbf{W}, \mathbf{W}') \geq 0$, which is equivalent to say (see Lemma 1) that $\mathbf{W}$ is a second-order critical point. We follow the proof strategy sketched in Section 3.4.3 after the statement of Proposition 11, and use the same notation introduced therein. Note that a first-order critical point can only be tightened if $H \geq 3$. Therefore, in all of this section we make the assumption $H \geq 3$. Recall that m is the number of examples in our sample, $\mathcal{S} = [\![1, r]\!]$, with $r < r_{max}$. We set $Q = [\![r+1, d_y]\!]$.
Recall also that

$$\Sigma^{1/2} = \Sigma_{YX}\Sigma_{XX}^{-1}X \in \mathbb{R}^{d_y \times m}.$$

and

$$\Sigma^{1/2} = U\Delta V^T$$

is a Singular Value Decomposition of $\Sigma^{1/2}$, where $\Delta \in \mathbb{R}^{d_y \times m}$ is such that $\Delta_{ii} = \sqrt{\lambda_i}$ for all $i \in [\![1, d_y]\!]$, and $(\lambda_i)_{i=1..d_y}$ are the eigenvalues of Σ.
We denote

$$\Delta^{(\mathcal{S})} = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_r}) \in \mathbb{R}^{r \times r} \tag{3.F.1}$$

and

$$\Delta^{(Q)} = \text{diag}(\sqrt{\lambda_{r+1}}, \dots, \sqrt{\lambda_{d_y}}) \in \mathbb{R}^{(d_y - r) \times (d_y - r)} \ . \tag{3.F.2}$$

Recall that, from Section 3.4.3, $c_2(\mathbf{W}, \mathbf{W}') = FT + ST$.
In what follows, we are going to present a key lemma, then various quick technical lemmas, then we simplify the expressions of FT and ST and conclude the proof of Proposition 11. Then, we prove all the lemmas of Appendix 3.F.1.
We present a lemma which uses that $\mathbf{W}$ is tightened to simplify some products of weight matrices and lighten further calculations. This is a key lemma as it introduces indices p and q which will be used multiple times in the proof.

Lemma 14. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L verifying the hypotheses of Proposition 11, and r, $\mathcal{S}$, Q, $(Z_h)_{h=1..H}$ as in Proposition 11. If $\mathbf{W}$ is tightened, then, there exist $p \in [\![3, H]\!]$ and $q \in [\![1, \min(p-1, H-2)]\!]$ such that:

$$\forall i \in [\![1, p-1]\!], \qquad W_H \cdots W_{i+1} = [U_{\mathcal{S}} \ , \ 0] \tag{3.F.3}$$

$$\forall i \in [\![p, H]\!], \qquad W_{i-1} \cdots W_2 = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \tag{3.F.4}$$

$$\forall i \in [\![q+1, H]\!], \qquad Z_{i-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q = 0 \tag{3.F.5}$$

$$\forall i \in [\![1, q]\!], \qquad W_{H-1} \cdots W_{i+1} = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \ . \tag{3.F.6}$$

The proof of Lemma 14 is in Appendix 3.F.1.5.

3.F.1.1 Useful technical lemmas

We now present technical lemmas which will be useful in Sections 3.F.1.2, 3.F.1.3 and 3.F.1.4. In all of these Lemmas, we have $\mathcal{S} = [\![1, r]\!]$ and $Q = [\![r+1, d_y]\!]$, and Assumption $\mathcal{H}$ holds true.

Lemma 15. We have

$$\Sigma_{XY} U_Q = X V_Q \Delta^{(Q)} \ .$$

The proof of Lemma 15 is in Appendix 3.F.1.6.

Lemma 16. Let n be a positive integer. For any matrices $A \in \mathbb{R}^{d_y \times n}$ and $B \in \mathbb{R}^{r \times n}$ we have

$$\|A + U_{\mathcal{S}} B\|^2 = \|U_{\mathcal{S}}^T A + B\|^2 + \|U_Q^T A\|^2 \ .$$

The proof of Lemma 16 is in Appendix 3.F.1.7.

Lemma 17. Let n be any positive integer. For any matrices $A \in \mathbb{R}^{n \times r}$ and $B \in \mathbb{R}^{n \times (d_y - r)}$ we have:

$$\left\langle A U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X \ , \ B V_Q^T \right\rangle = 0 \ .$$

The proof of Lemma 17 is in Appendix 3.F.1.8.

Lemma 18. Let n be any positive integer. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point of L verifying the hypotheses of Proposition 11, and r, $\mathcal{S}$, Q, $(Z_h)_{h=1..H}$ as in Proposition 11. If W is tightened, then, for q as in Lemma 14, for any matrices $A \in \mathbb{R}^{n \times (d_q - r)}$ and $B \in \mathbb{R}^{n \times (d_y - r)}$, we have:

$$\left\langle A Z_q \cdots Z_2 Z_1 X \ , \ B V_Q^T \right\rangle = 0 \ .$$

The proof of Lemma 18 is in Appendix 3.F.1.9.

Lemma 19. For any matrix $A \in \mathbb{R}^{(d_y - r) \times r}$ we have

$$\|A U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X\|^2 = \sum_{a=1}^{r} \sum_{b=r+1}^{d_y} (\lambda_a - \lambda_b)(A_{b-r,a})^2 + \|\Delta^{(Q)} A\|^2 \ .$$

The proof of Lemma 19 is in Appendix 3.F.1.10.

Lemma 20. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point associated with $\mathcal{S}$. For any matrix $A \in \mathbb{R}^{d_y \times d_x}$, we have

$$\langle A X \ , \ W_H \cdots W_1 X - Y \rangle = \left\langle A \ , \ -U_Q U_Q^T \Sigma_{YX} \right\rangle \ .$$

The proof of Lemma 20 is in Appendix 3.F.1.11.

3.F.1.2 Simplifying FT

In this section and the next one, we simplify the expressions of FT and ST as defined in (3.4.9) and (3.4.10). In order to decompose $FT = a_1 + \|A_2\|^2 + \|A_3\|^2 + \|A_4\|^2$, with $a_1 \geq 0$, we first simplify the terms T_i, for $i \in [\![1, H]\!]$, defined in (3.4.7). Let us first consider $\mathbf{W}$ tightened satisfying the hypotheses of Proposition 11, and p and q defined as in Lemma 14. The simplification of T_i depends on the position of i with regard to 1 , q, p and H. We define $J_1 = [\![p, H-1]\!]$, $J_2 = [\![q+1, p-1]\!]$ and $J_3 = [\![2, q]\!]$.
Note that, according to the convention in Section 3.2, these sets could be empty.

— if $p = H$, $J_1 = \emptyset$

— if $q = p - 1$, $J_2 = \emptyset$

— if $q = 1$, $J_3 = \emptyset$.

Note also that $\{1\}, J_3, J_2, J_1, \{H\}$ are disjoint and $\{1\} \cup J_3 \cup J_2 \cup J_1 \cup \{H\} = [\![1, H]\!]$.
Depending on the position of i, we need to distinguish four cases, in order to simplify T_i.

Lemma 21. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point satisfying the hypotheses of Proposition 11, and r, $\mathcal{S}$, Q, $(Z_h)_{h=1..H}$ as in Proposition 11. Let $i \in [\![1, H]\!]$. For any $\mathbf{W}' = (W'_H, \dots, W'_1)$, recall that , as defined in (3.4.7),

$$T_i = W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_1 X \ .$$

If $\mathbf{W}$ is tightened, then, for p and q as defined in Lemma 14 and J_1,J_2,J_3 as defined above, we have

— For $i = H$:

$$T_H = (W'_H)_{.,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X \tag{3.F.7}$$

— For $i \in J_1$:

$$T_i = U_{\mathcal{S}} (W'_i)_{1:r,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + U_Q Z_H Z_{H-1} \cdots Z_{i+1} (W'_i)_{r+1:d_i,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X \tag{3.F.8}$$

— For $i \in J_2 \cup J_3$:

$$T_i = U_{\mathcal{S}} (W'_i)_{1:r,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + U_{\mathcal{S}} (W'_i)_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X \tag{3.F.9}$$

— For $i = 1$:

$$T_1 = U_{\mathcal{S}} (W'_1)_{1:r,.} X \tag{3.F.10}$$

The proof of Lemma 21 is in Appendix 3.F.1.12.
We now simplify FT. Substituting the formulas of Lemma 21 in (3.4.9) we have

$$FT = \left\| \sum_{i=1}^{H} T_i \right\|^2$$

$$
\begin{aligned}
&= \big\|(W'_H)_{.,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X \\
&\quad + \sum_{i\in J_1}\big(U_{\mathcal{S}}(W'_i)_{1:r,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X + U_Q Z_H Z_{H-1}\cdots Z_{i+1}(W'_i)_{r+1:d_i,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X\big) \\
&\quad + \sum_{i\in J_2\cup J_3}\big(U_{\mathcal{S}}(W'_i)_{1:r,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X + U_{\mathcal{S}}(W'_i)_{1:r,r+1:d_{i-1}}Z_{i-1}\cdots Z_2Z_1X\big) + U_{\mathcal{S}}(W'_1)_{1:r,.}X\Big\|^2 .
\end{aligned}
$$

FT can be identified with a term as $\|A + U_{\mathcal{S}}B\|^2$ if we take

$$
A = (W'_H)_{.,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X + \sum_{i\in J_1} U_Q Z_H Z_{H-1}\cdots Z_{i+1}(W'_i)_{r+1:d_i,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X \ .
$$

and

$$
\begin{aligned}
B &= \sum_{i\in J_1}(W'_i)_{1:r,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X \\
&\quad + \sum_{i\in J_2\cup J_3}\big((W'_i)_{1:r,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X + (W'_i)_{1:r,r+1:d_{i-1}}Z_{i-1}\cdots Z_2Z_1X\big) + (W'_1)_{1:r,.}X \ .
\end{aligned}
$$

Applying Lemma 16, FT becomes:

$$
\begin{aligned}
FT &= \|U_{\mathcal{S}}^TA + B\|^2 + \|U_Q^TA\|^2 \\
&= \Big\|U_{\mathcal{S}}^T(W'_H)_{.,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X + \sum_{i\in J_1}U_{\mathcal{S}}^TU_Q Z_H Z_{H-1}\cdots Z_{i+1}(W'_i)_{r+1:d_i,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X \\
&\quad + \sum_{i\in J_1}(W'_i)_{1:r,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X \\
&\quad + \sum_{i\in J_2\cup J_3}\big((W'_i)_{1:r,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X + (W'_i)_{1:r,r+1:d_{i-1}}Z_{i-1}\cdots Z_2Z_1X\big) + (W'_1)_{1:r,.}X\Big\|^2 \\
&\quad + \Big\|U_Q^T(W'_H)_{.,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X + \sum_{i\in J_1}U_Q^TU_Q Z_H Z_{H-1}\cdots Z_{i+1}(W'_i)_{r+1:d_i,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X\Big\|^2 .
\end{aligned}
$$

Using Lemma 6, we have $U_{\mathcal{S}}^TU_Q = 0$ and $U_Q^TU_Q = I_{d_y-r}$, hence we can write

$$
FT = FT_1 + FT_2 \ ,
$$

where

$$
FT_1 = \Big\|U_{\mathcal{S}}^T(W'_H)_{.,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X + \sum_{i\in J_1}(W'_i)_{1:r,1:r}U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X
$$

$$+ \sum_{i \in J_2 \cup J_3} \left((W_i')_{1:r,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + (W_i')_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X \right) + (W_1')_{1:r,.} X \Big\|^2 ,$$

and

$$FT_2 = \left\| U_Q^T (W_H')_{.,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + \sum_{i \in J_1} Z_H Z_{H-1} \cdots Z_{i+1} (W_i')_{r+1:d_i,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X \right\|^2 .$$

Let us first simplify FT_1.
Recall that m is the number of examples in our sample, $V \in \mathbb{R}^{m \times m}$ is the orthogonal matrix defined in (3.2.1) and $Q = [\![r+1, d_y]\!]$. We set $\mathcal{S}' = \mathcal{S} \cup [\![d_y+1, m]\!] = [\![1, r]\!] \cup [\![d_y+1, m]\!]$ such that $\mathcal{S}' \cup Q = [\![1, m]\!]$.
Reordering the terms and, since V is orthogonal, using $I_m = VV^T = V_{\mathcal{S}'} V_{\mathcal{S}'}^T + V_Q V_Q^T$, we have

$$\begin{aligned} FT_1 = \Bigg\| & \left(U_{\mathcal{S}}^T (W_H')_{.,1:r} + \sum_{i \in J_1 \cup J_2 \cup J_3} (W_i')_{1:r,1:r} \right) U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X \\ & + \sum_{i \in J_2} (W_i')_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X \\ & + \left(\sum_{i \in J_3} (W_i')_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X + (W_1')_{1:r,.} X \right) \left(V_{\mathcal{S}'} V_{\mathcal{S}'}^T + V_Q V_Q^T \right) \Bigg\|^2 . \end{aligned}$$

Since for $i \in J_2$, we have $i - 1 \geq q$, we denote

$$N := \sum_{i \in J_2} (W_i')_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_{q+1} ,$$

Recall that, using the convention in Section 3.2, for $i - 1 = q$, we have $Z_{i-1} \cdots Z_{q+1} = I_{d_q - r}$.
We also denote

$$\begin{aligned} M &:= \sum_{i \in J_3} (W_i')_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X V_Q + (W_1')_{1:r,.} X V_Q , \\ J &:= \sum_{i \in J_3} (W_i')_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X V_{\mathcal{S}'} + (W_1')_{1:r,.} X V_{\mathcal{S}'} , \\ L &:= U_{\mathcal{S}}^T (W_H')_{.,1:r} + \sum_{i \in J_1 \cup J_2 \cup J_3} (W_i')_{1:r,1:r} . \end{aligned}$$

Therefore, we obtain

$$\begin{aligned} FT_1 &= \left\| L U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + N Z_q \cdots Z_2 Z_1 X + J V_{\mathcal{S}'}^T + M V_Q^T \right\|^2 \\ &= \left\| L U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + N Z_q \cdots Z_2 Z_1 X + J V_{\mathcal{S}'}^T \right\|^2 + \left\| M V_Q^T \right\|^2 \end{aligned}$$

$$+ 2 \left\langle LU_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + N Z_q \cdots Z_2 Z_1 X + J V_{\mathcal{S}'}^T \, , \, M V_Q^T \right\rangle \, .$$

Using Lemma 17 and Lemma 18 and $V_Q^T V_{\mathcal{S}'} = 0$ (since V is orthogonal), the cross-product is equal to zero.
Noting also that since V is orthogonal $\|MV_Q^T\|^2 = \mathrm{tr}(MV_Q^T V_Q M^T) = \mathrm{tr}(MM^T) = \|M\|^2 = \|M^T\|^2$, we have

$$\begin{aligned} FT_1 &= \left\| LU_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + N Z_q \cdots Z_2 Z_1 X + J V_{\mathcal{S}'}^T \right\|^2 + \left\| M^T \right\|^2 \\ &= \|A_2\|^2 + \|A_4\|^2 \end{aligned}$$

where

$$\begin{aligned} A_2 &:= LU_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + N Z_q \cdots Z_2 Z_1 X + J V_{\mathcal{S}'}^T \\ &= U_{\mathcal{S}}^T (W'_H)_{.,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + \sum_{i \in J_1} (W'_i)_{1:r,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X \\ &\quad + \sum_{i \in J_2} \left((W'_i)_{1:r,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + (W'_i)_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X \right) \\ &\quad + \sum_{i \in J_3} \left((W'_i)_{1:r,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + (W'_i)_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X V_{\mathcal{S}'} V_{\mathcal{S}'}^T \right) + (W'_1)_{1:r,.} X V_{\mathcal{S}'} V_{\mathcal{S}'}^T \end{aligned} \tag{3.F.11}$$

$$A_4 := M^T = \left(\sum_{i \in J_3} (W'_i)_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X V_Q + (W'_1)_{1:r,.} X V_Q \right)^T \, . \tag{3.F.12}$$

Let us now simplify FT_2.
We have $FT_2 = \left\| A U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X \right\|^2$, with

$$A := U_Q^T (W'_H)_{.,1:r} + \sum_{i \in J_1} Z_H Z_{H-1} \cdots Z_{i+1} (W'_i)_{r+1:d_i,1:r} \in \mathbb{R}^{(d_y - r) \times r} \, .$$

Hence, using Lemma 19, we have

$$\begin{aligned} FT_2 &= \sum_{a=1}^{r} \sum_{b=r+1}^{d_y} (\lambda_a - \lambda_b)(A_{b-r,a})^2 + \|\Delta^{(Q)} A\|^2 \\ &= \sum_{a=1}^{r} \sum_{b=r+1}^{d_y} (\lambda_a - \lambda_b) \left(U_b^T (W'_H)_{.,a} + \sum_{i \in J_1} (Z_H)_{b-r,.} Z_{H-1} \cdots Z_{i+1} (W'_i)_{r+1:d_i,a} \right)^2 \\ &\quad + \left\| \Delta^{(Q)} \left(U_Q^T (W'_H)_{.,1:r} + \sum_{i \in J_1} Z_H Z_{H-1} \cdots Z_{i+1} (W'_i)_{r+1:d_i,1:r} \right) \right\|^2 \\ &= a_1 + \|A_3\|^2 \, , \end{aligned}$$

where

$$a_1 := \sum_{a=1}^{r} \sum_{b=r+1}^{d_y} (\lambda_a - \lambda_b) \left(U_b^T (W'_H)_{.,a} + \sum_{i \in J_1} (Z_H)_{b-r,.} Z_{H-1} \cdots Z_{i+1} (W'_i)_{r+1:d_i,a} \right)^2 \tag{3.F.13}$$

$$A_3 := \Delta^{(Q)} \left(U_Q^T (W'_H)_{.,1:r} + \sum_{i \in J_1} Z_H Z_{H-1} \cdots Z_{i+1} (W'_i)_{r+1:d_i,1:r} \right) \tag{3.F.14}$$

Finally,

$$\begin{aligned} FT &= FT_1 + FT_2 \\ &= a_1 + \|A_2\|^2 + \|A_3\|^2 + \|A_4\|^2 \,, \end{aligned} \tag{3.F.15}$$

where a_1, A_2, A_3, A_4 are defined in (3.F.13), (3.F.11), (3.F.14), (3.F.12). Notice that, since $\lambda_1 > \cdots > \lambda_{d_y}$, we have

$$a_1 \geq 0 \,. \tag{3.F.16}$$

3.F.1.3 Simplifying ST

In this section, we prove that $ST = -2 \langle A_3, A_4 \rangle$, where ST, A_3 and A_4 are defined in (3.4.10), (3.F.14) and (3.F.12). In order to do so, we first state a lemma that simplifies the terms $T_{i,j}$ defined in (3.4.8). We remind that the sets J_1, J_2 and J_3 are defined at the beginning of Section 3.F.1.2.

Lemma 22. Suppose Assumption $\mathcal{H}$ in Section 3.2 holds true. Let $\mathbf{W} = (W_H, \cdots, W_1)$ be a first-order critical point satisfying the hypotheses of Proposition 11, and r, $\mathcal{S}$, Q, $(Z_h)_{h=1..H}$ defined as in Proposition 11. Let $(i,j) \in [\![1, H]\!]^2$, with $i > j$. For any $\mathbf{W}' = (W'_H, \ldots, W'_1)$, recall that , as defined in (3.4.8),

$$T_{i,j} = \left\langle W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_{j+1} W'_j W_{j-1} \cdots W_1 X \,,\; W_H \cdots W_1 X - Y \right\rangle \,.$$

If $\mathbf{W}$ is tightened, then, for p and q as defined in Lemma 14 and J_1,J_2,J_3 as defined above, we have

— For $i = H$:

  — For $j \in J_3$:

$$T_{H,j} = -\left\langle \Delta^{(Q)} U_Q^T (W'_H)_{.,1:r} \,,\; \left((W'_j)_{1:r,r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q \right)^T \right\rangle \,. \tag{3.F.17}$$

  — For $j = 1$:

$$T_{H,1} = -\left\langle \Delta^{(Q)} U_Q^T (W'_H)_{.,1:r} \,,\; \left((W'_1)_{1:r,.} X V_Q \right)^T \right\rangle \,. \tag{3.F.18}$$

— For $j \in J_1 \cup J_2$:

$$T_{H,j} = 0 \ . \tag{3.F.19}$$

— For $i \in J_1$:

— For $j \in J_3$:

$$T_{i,j} = -\left\langle \Delta^{(Q)} Z_H Z_{H-1} \cdots Z_{i+1} (W_i')_{r+1:d_i,1:r} \ , \ \left((W_j')_{1:r,r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q\right)^T \right\rangle \ . \tag{3.F.20}$$

— For $j = 1$:

$$T_{i,1} = -\left\langle \Delta^{(Q)} Z_H Z_{H-1} \cdots Z_{i+1} (W_i')_{r+1:d_i,1:r} \ , \ ((W_1')_{1:r,.} X V_Q)^T \right\rangle \ . \tag{3.F.21}$$

— For $j \in J_1 \cup J_2$:

$$T_{i,j} = 0 \ . \tag{3.F.22}$$

— For $i \in J_2 \cup J_3$, for all $j < i$, we have

$$T_{i,j} = 0 \ . \tag{3.F.23}$$

The proof of Lemma 22 is in Appendix 3.F.1.13.

Let us now prove that $ST = -2 \langle A_3, A_4 \rangle$. We remind that $[\![1, H]\!] = \{H\} \cup J_1 \cup J_2 \cup J_3 \cup \{1\}$ and separate the sum appearing in (3.4.10) accordingly.

We then substitute the formulas of Lemma 22 in (3.4.10) and obtain

$$\begin{aligned}
ST &= 2 \sum_{H \geq i > j \geq 1} T_{i,j} \\
&= 2 \left(\sum_{j \in J_1 \cup J_2} T_{H,j} + \sum_{j \in J_3} T_{H,j} + T_{H,1} + \sum_{i \in J_1} \sum_{\substack{j \in J_1 \cup J_2, \\ j<i}} T_{i,j} + \sum_{i \in J_1} \sum_{j \in J_3} T_{i,j} + \sum_{i \in J_1} T_{i,1} + \sum_{i \in J_2 \cup J_3} \sum_{j=1}^{i-1} T_{i,j} \right) \\
&= -2 \sum_{j \in J_3} \left\langle \Delta^{(Q)} U_Q^T (W_H')_{.,1:r} \ , \ \left((W_j')_{1:r,r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q\right)^T \right\rangle \\
&\quad - 2 \left\langle \Delta^{(Q)} U_Q^T (W_H')_{.,1:r} \ , \ ((W_1')_{1:r,.} X V_Q)^T \right\rangle \\
&\quad - 2 \sum_{i \in J_1} \sum_{j \in J_3} \left\langle \Delta^{(Q)} Z_H Z_{H-1} \cdots Z_{i+1} (W_i')_{r+1:d_i,1:r} \ , \ \left((W_j')_{1:r,r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q\right)^T \right\rangle \\
&\quad - 2 \sum_{i \in J_1} \left\langle \Delta^{(Q)} Z_H Z_{H-1} \cdots Z_{i+1} (W_i')_{r+1:d_i,1:r} \ , \ ((W_1')_{1:r,.} X V_Q)^T \right\rangle
\end{aligned}$$

$$
\begin{aligned}
&= -2 \left\langle \Delta^{(Q)} U_Q^T (W'_H)_{.,1:r} \ , \ \left(\sum_{j \in J_3} (W'_j)_{1:r, r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q + (W'_1)_{1:r,.} X V_Q \right)^T \right\rangle \\
&\quad - 2 \left\langle \Delta^{(Q)} \sum_{i \in J_1} Z_H Z_{H-1} \cdots Z_{i+1} (W'_i)_{r+1:d_i, 1:r} \ , \right. \\
&\qquad \left. \left(\sum_{j \in J_3} (W'_j)_{1:r, r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q + (W'_1)_{1:r,.} X V_Q \right)^T \right\rangle \\
&= -2 \left\langle \Delta^{(Q)} \left(U_Q^T (W'_H)_{.,1:r} + \sum_{i \in J_1} Z_H Z_{H-1} \cdots Z_{i+1} (W'_i)_{r+1:d_i, 1:r} \right) \ , \right. \\
&\qquad \left. \left(\sum_{j \in J_3} (W'_j)_{1:r, r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q + (W'_1)_{1:r,.} X V_Q \right)^T \right\rangle \\
&= -2 \left\langle A_3, A_4 \right\rangle \ , \qquad \text{(3.F.24)}
\end{aligned}
$$

where we remind that A_3 and A_4 are defined in (3.F.14) and (3.F.12).

3.F.1.4 Concluding the proof of Proposition 11

Using the simplifications (3.F.15) and (3.F.24) above, for any $\mathbf{W}$ satisfying the hypotheses of Proposition 11, if $\mathbf{W}$ is tightened, then for any $\mathbf{W}'$,

$$
\begin{aligned}
c_2(\mathbf{W}, \mathbf{W}') &= FT + ST \\
&= a_1 + \|A_2\|^2 + \|A_3\|^2 + \|A_4\|^2 - 2 \left\langle A_3, A_4 \right\rangle \\
&= a_1 + \|A_2\|^2 + \|A_3 - A_4\|^2 \ .
\end{aligned}
$$

Using (3.F.16), we find $c_2(\mathbf{W}, \mathbf{W}') \geq 0$.
Therefore, $\mathbf{W} = (W_H, \ldots, W_1)$ is a second-order critical point.

3.F.1.5 Proof of Lemma 14

First note that, for $r = 0$, we can easily follow the same proof and see that the result still holds with the conventions adopted in Section 3.2.
Let us prove (3.F.3).
Consider the pivot $(i, j) = (2, 1)$. Its complementary blocks are $\Sigma_{XY} W_H \cdots W_3$ and I_{d_1}.
Since $\mathbf{W}$ is tightened and $\mathrm{rk}(I_{d_1}) = d_1 \geq r_{max} > r$, we have $\mathrm{rk}(\Sigma_{XY} W_H \cdots W_3) = r$.
Since Σ_{XY} is full-column rank, we obtain $\mathrm{rk}(W_H \cdots W_3) = r$.
Let $p \in [\![3, H]\!]$ be the largest index such that

$$
rk(W_H \cdots W_p) = r \ . \qquad \text{(3.F.25)}
$$

Using (3.4.3) and (3.4.5), we have $W_H \cdots W_p = [U_{\mathcal{S}} \ , \ U_Q Z_H Z_{H-1} \cdots Z_p]$.
Since $rk(W_H \cdots W_p) = r$ and since the columns of $U_Q Z_H Z_{H-1} \cdots Z_p$ are in the vector

space spanned by the columns of U_Q (which are orthogonal to the columns of $U_{\mathcal{S}}$), (3.F.25) implies

$$Z_H Z_{H-1} \cdots Z_p = 0 \ .$$

Therefore,

$$W_H \cdots W_p = [U_{\mathcal{S}} \ , \ 0] \ .$$

Using (3.4.5), for all $i \in [\![1, p-1]\!]$,

$$\begin{aligned} W_H \cdots W_{i+1} &= (W_H \cdots W_p)(W_{p-1} \cdots W_{i+1}) \\ &= [U_{\mathcal{S}} \ , \ 0] \begin{bmatrix} I_r & 0 \\ 0 & Z_{p-1} \cdots Z_{i+1} \end{bmatrix} \\ &= [U_{\mathcal{S}} \ , \ 0] \ . \end{aligned}$$

This proves (3.F.3).

Let us prove (3.F.4).

We consider the pivot $(p, 1)$. Its complementary blocks are $\Sigma_{XY} W_H \cdots W_{p+1}$ and $W_{p-1} \cdots W_2$. We have, by definition of p, $\mathrm{rk}(W_H \cdots W_{p+1}) > r$. Therefore, since Σ_{XY} is full-column rank, we have $\mathrm{rk}(\Sigma_{XY} W_H \cdots W_{p+1}) = \mathrm{rk}(W_H \cdots W_{p+1}) > r$. Note that this holds both for $p = H$ and for $p < H$. Hence, since **W** is tightened, the second complementary block is of rank r, i.e.

$$\mathrm{rk}(W_{p-1} \cdots W_2) = r \ .$$

Using (3.4.5), we also have $W_{p-1} \cdots W_2 = \begin{bmatrix} I_r & 0 \\ 0 & Z_{p-1} \cdots Z_2 \end{bmatrix}$.

Then, since $\mathrm{rk}(W_{p-1} \cdots W_2) = r$, we have $Z_{p-1} \cdots Z_2 = 0$ and

$$W_{p-1} \cdots W_2 = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} .$$

Using (3.4.5) again, for all $i \in [\![p, H]\!]$,

$$\begin{aligned} W_{i-1} \cdots W_2 &= (W_{i-1} \cdots W_p)(W_{p-1} \cdots W_2) \\ &= \begin{bmatrix} I_r & 0 \\ 0 & Z_{i-1} \cdots Z_p \end{bmatrix} \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} . \end{aligned}$$

This proves (3.F.4).

Let us now prove (3.F.5).

Using Proposition 1, Lemma 4 and Lemma 6, we have

$$\mathrm{rk}(W_{p-1} \cdots W_1 \Sigma_{XY}) \geq \mathrm{rk}(W_H \cdots W_1 \Sigma_{XY}) = \mathrm{rk}(U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma) \geq \mathrm{rk}(U_{\mathcal{S}}^T (U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma) \Sigma^{-1} U_{\mathcal{S}}) = \mathrm{rk}(I_r) = r \ .$$

Using (3.F.4) for $i = p$, we also have $\mathrm{rk}(W_{p-1}\cdots W_1\Sigma_{XY}) \leq \mathrm{rk}(W_{p-1}\cdots W_2) = r$. Hence, $\mathrm{rk}(W_{p-1}\cdots W_1\Sigma_{XY}) = r$.
Notice that, considering the tightened pivot $(H, H-1)$, since $\mathrm{rk}(I_{d_{H-1}}) = d_{H-1} \geq r_{max} > r$, we obtain $\mathrm{rk}(W_{H-2}\cdots W_1\Sigma_{XY}) = r$.
We consider $q \in [\![1, \min(p-1, H-2)]\!]$ the smallest index such that *rk*$(W_q\cdots W_1\Sigma_{XY}) = r$.
Using (3.4.5) and (3.4.4), we have

$$\begin{aligned} W_q\cdots W_1\Sigma_{XY} &= \begin{bmatrix} U_S^T\Sigma \\ Z_q\cdots Z_2Z_1\Sigma_{XY} \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1 U_1^T \\ \vdots \\ \lambda_r U_r^T \\ Z_q\cdots Z_2Z_1\Sigma_{XY} \end{bmatrix} . \end{aligned}$$

Since $\mathrm{rk}(W_q\cdots W_1\Sigma_{XY}) = r$, every row of $Z_q\cdots Z_2Z_1\Sigma_{XY}$ lies in $\mathrm{Vec}(U_1^T, \ldots, U_r^T)$, hence we have

$$Z_q\cdots Z_2Z_1\Sigma_{XY}U_Q = 0 .$$

Finally, we conclude that, for all $i \in [\![q+1, H]\!]$,

$$\begin{aligned} Z_{i-1}\cdots Z_2Z_1\Sigma_{XY}U_Q &= Z_{i-1}\cdots Z_{q+1}Z_q\cdots Z_2Z_1\Sigma_{XY}U_Q \\ &= Z_{i-1}\cdots Z_{q+1}0 \\ &= 0 . \end{aligned}$$

This proves (3.F.5).
Let us now prove (3.F.6).
Consider the pivot (H, q). Its complementary blocks are $W_{q-1}\cdots W_1\Sigma_{XY}$ and $W_{H-1}\cdots W_{q+1}$. We have, by definition of q, $\mathrm{rk}(W_{q-1}\cdots W_1\Sigma_{XY}) > r$. Hence, since **W** is tightened, the other complementary block is of rank r, i.e. $\mathrm{rk}(W_{H-1}\cdots W_{q+1}) = r$.
Using (3.4.5), we have

$$W_{H-1}\cdots W_{q+1} = \begin{bmatrix} I_r & 0 \\ 0 & Z_{H-1}\cdots Z_{q+1} \end{bmatrix} .$$

Therefore, $Z_{H-1}\cdots Z_{q+1} = 0$ and

$$W_{H-1}\cdots W_{q+1} = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} .$$

Finally, using (3.4.5), for all $i \in [\![1, q]\!]$,

$$W_{H-1}\cdots W_{i+1} = W_{H-1}\cdots W_{q+1}W_q\cdots W_{i+1}$$

$$= \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I_r & 0 \\ 0 & Z_q \cdots Z_{i+1} \end{bmatrix}$$
$$= \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} .$$

This proves (3.F.6) and concludes the proof.

3.F.1.6 Proof of Lemma 15

Recall that $\Sigma^{1/2} = \Sigma_{YX}\Sigma_{XX}^{-1}X$. We have

$$\begin{aligned} \Sigma_{XY} &= XY^T \\ &= XX^T(XX^T)^{-1}XY^T \\ &= X(\Sigma^{1/2})^T . \end{aligned}$$

Using (3.2.1), we obtain

$$\Sigma_{XY} = XV\Delta^T U^T,$$

and, since U is orthogonal, we have

$$\Sigma_{XY}U = XV\Delta^T.$$

Restricting the equality to the columns in Q, we obtain

$$\Sigma_{XY}U_Q = XV_Q\Delta^{(Q)} ,$$

where $\Delta^{(Q)}$ is defined in (3.F.2). This concludes the proof.

3.F.1.7 Proof of Lemma 16

Let $A \in \mathbb{R}^{d_y \times n}$ and $B \in \mathbb{R}^{r \times n}$. We have

$$\begin{aligned} \|A + U_{\mathcal{S}}B\|^2 &= \|A\|^2 + \|U_{\mathcal{S}}B\|^2 + 2\langle A\ ,\ U_{\mathcal{S}}B\rangle \\ &= tr(A^TA) + tr(B^TU_{\mathcal{S}}^TU_{\mathcal{S}}B) + 2\left\langle U_{\mathcal{S}}^TA\ ,\ B\right\rangle . \end{aligned}$$

Using Lemma 6, this becomes

$$\begin{aligned} \|A + U_{\mathcal{S}}B\|^2 &= tr\left(A^T(U_{\mathcal{S}}U_{\mathcal{S}}^T + U_QU_Q^T)A\right) + tr\left(B^TB\right) + 2\left\langle U_{\mathcal{S}}^TA\ ,\ B\right\rangle \\ &= tr(A^TU_QU_Q^TA) + tr(A^TU_{\mathcal{S}}U_{\mathcal{S}}^TA) + tr(B^TB) + 2\left\langle U_{\mathcal{S}}^TA\ ,\ B\right\rangle \\ &= \|U_Q^TA\|^2 + \|U_{\mathcal{S}}^TA\|^2 + \|B\|^2 + 2\left\langle U_{\mathcal{S}}^TA\ ,\ B\right\rangle \\ &= \|U_Q^TA\|^2 + \|U_{\mathcal{S}}^TA + B\|^2 . \end{aligned}$$

3.F.1.8 Proof of Lemma 17

Recall that $\Sigma^{1/2} = \Sigma_{YX}\Sigma_{XX}^{-1}X$ has a Singular Value Decomposition $\Sigma^{1/2} = U\Delta V^T$ (see (3.2.1)). Hence, we have $\Sigma^{1/2}V = U\Delta$ and therefore $\Sigma^{1/2}V_Q = U_Q\Delta^{(Q)}$, where $\Delta^{(Q)}$ is defined in (3.F.2).
As a consequence,

$$\begin{aligned} U_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}XV_Q &= U_{\mathcal{S}}^T\Sigma^{1/2}V_Q \\ &= U_{\mathcal{S}}^TU_Q\Delta^{(Q)} \\ &= 0\ , \end{aligned}$$

where the last equality follows from Lemma 6. Finally, we obtain for any $A \in \mathbb{R}^{n\times r}$, $B \in \mathbb{R}^{n\times(d_y-r)}$

$$\begin{aligned} \left\langle AU_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X\ ,\ BV_Q^T\right\rangle &= tr(AU_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}XV_QB^T) \\ &= 0\ . \end{aligned}$$

3.F.1.9 Proof of Lemma 18

Using Lemma 15, we have $\Sigma_{XY}U_Q = XV_Q\Delta^{(Q)}$, then replacing this formula in (3.F.5) with $i = q+1$, we have

$$Z_q\cdots Z_2Z_1XV_Q\Delta^{(Q)} = 0\ .$$

Since $\Delta^{(Q)}$ is diagonal and its diagonal elements are non-zero, it is invertible, hence

$$Z_q\cdots Z_2Z_1XV_Q = 0\ .$$

Finally, for any matrices $A \in \mathbb{R}^{n\times(d_q-r)}$ and $B \in \mathbb{R}^{n\times(d_y-r)}$, we have

$$\begin{aligned} \left\langle AZ_q\cdots Z_2Z_1X\ ,\ BV_Q^T\right\rangle &= tr(AZ_q\cdots Z_2Z_1XV_QB^T) \\ &= 0\ . \end{aligned}$$

3.F.1.10 Proof of Lemma 19

Recall that $\Delta^{(\mathcal{S})}$ is defined in (3.F.1) and $\Sigma = U\Lambda U^T$. Let $A \in \mathbb{R}^{(d_y-r)\times r}$, we have

$$\begin{aligned} \left\|AU_{\mathcal{S}}^T\Sigma_{YX}\Sigma_{XX}^{-1}X\right\|^2 &= tr(AU_{\mathcal{S}}^T\Sigma U_{\mathcal{S}}A^T) \\ &= tr(A\ \mathrm{diag}(\lambda_1,\dots,\lambda_r)A^T) \\ &= \left\|A\Delta^{(\mathcal{S})}\right\|^2 \\ &= \sum_{a=1}^{r}\sum_{b=r+1}^{d_y}\lambda_a(A_{b-r,a})^2 \\ &= \sum_{a=1}^{r}\sum_{b=r+1}^{d_y}(\lambda_a-\lambda_b)(A_{b-r,a})^2 + \sum_{a=1}^{r}\sum_{b=r+1}^{d_y}\lambda_b(A_{b-r,a})^2 \end{aligned}$$

$$= \sum_{a=1}^{r} \sum_{b=r+1}^{d_y} (\lambda_a - \lambda_b)(A_{b-r,a})^2 + \|\Delta^{(Q)} A\|^2 \ .$$

3.F.1.11 Proof of Lemma 20

Let $\mathbf{W} = (W_H, \dots, W_1)$ be a first-order critical point associated with $\mathcal{S}$ verifying the hypotheses of Proposition 11 and let $A \in \mathbb{R}^{d_y \times d_x}$. Using (3.4.6), (3.4.4), and Lemma 6, we have

$$\begin{aligned}
\langle AX \ , \ W_H \cdots W_1 X - Y \rangle &= \langle A \ , \ W_H \cdots W_1 X X^T - Y X^T \rangle \\
&= \langle A \ , \ U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X X^T - \Sigma_{YX} \rangle \\
&= \langle A \ , \ U_{\mathcal{S}} U_{\mathcal{S}}^T \Sigma_{YX} - \Sigma_{YX} \rangle \\
&= \langle A \ , \ -U_Q U_Q^T \Sigma_{YX} \rangle \ .
\end{aligned}$$

3.F.1.12 Proof of Lemma 21

Let $\mathbf{W} = (W_H, \cdots, W_1)$ be a tightened first-order critical point satisfying the hypotheses of Proposition 11, and r, $\mathcal{S}$, Q, $(Z_h)_{h=1..H}$ defined as in Proposition 11. Since $\mathbf{W}$ satisfies the hypotheses of Proposition 11, we are going to use all the equations (3.4.3), (3.4.4), (3.4.5) and (3.4.6) defined by these hypotheses and (3.F.3), (3.F.4), (3.F.5) and (3.F.6) of Lemma 14.
Let $\mathbf{W}' = (W'_H, \dots, W'_1)$ and $i \in [\![1, H]\!]$. Recall that T_i is defined in (3.4.7) and $J_1 = [\![p, H-1]\!]$, $J_2 = [\![q+1, p-1]\!]$, $J_3 = [\![2, q]\!]$, where p and q are defined as in Lemma 14.

Consider the case $i = H$.
Substituting (3.F.4) and (3.4.4) in (3.4.7), we have

$$\begin{aligned}
T_H &= W'_H (W_{H-1} \cdots W_2) W_1 X \\
&= W'_H \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} X \\
&= W'_H \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0 \end{bmatrix} X \\
&= (W'_H)_{\cdot, 1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X \ .
\end{aligned}$$

This proves (3.F.7).

Consider now the case $i \in J_1$.
Substituting (3.4.3), (3.4.5), (3.F.4) and (3.4.4), in (3.4.7), we have, for $i \in J_1$

$$\begin{aligned}
T_i &= W_H (W_{H-1} \cdots W_{i+1}) W'_i (W_{i-1} \cdots W_2) W_1 X \\
&= [U_{\mathcal{S}} \ , \ U_Q Z_H] \begin{bmatrix} I_r & 0 \\ 0 & Z_{H-1} \end{bmatrix} \cdots \begin{bmatrix} I_r & 0 \\ 0 & Z_{i+1} \end{bmatrix} W'_i \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} X
\end{aligned}$$

$$
\begin{aligned}
&= [U_{\mathcal{S}}\ ,\ U_Q Z_H Z_{H-1} \cdots Z_{i+1}]\, W_i' \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ 0 \end{bmatrix} X \\
&= [U_{\mathcal{S}}\ ,\ U_Q Z_H Z_{H-1} \cdots Z_{i+1}]\, (W_i')_{.,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X \\
&= U_{\mathcal{S}} (W_i')_{1:r,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + U_Q Z_H Z_{H-1} \cdots Z_{i+1} (W_i')_{r+1:d_i,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X\ .
\end{aligned}
$$

Note that the above calculations are still valid in the case $i = H - 1$. In this case using the convention in Section 3.2, $W_{H-1} \cdots W_{i+1} = I_{d_{H-1}}$ and $Z_{H-1} \cdots Z_{i+1} = I_{d_{H-1}-r}$.
This proves (3.F.8).

Consider now the case $i \in J_2 \cup J_3 = [\![2, p-1]\!]$.
Substituting (3.F.3), (3.4.5) and (3.4.4), in (3.4.7), we have, for $i \in J_2 \cup J_3$,

$$
\begin{aligned}
T_i &= (W_H \cdots W_{i+1}) W_i' (W_{i-1} \cdots W_2) W_1 X \\
&= \begin{bmatrix} U_{\mathcal{S}}\ ,\ 0 \end{bmatrix} W_i' \begin{bmatrix} I_r & 0 \\ 0 & Z_{i-1} \end{bmatrix} \cdots \begin{bmatrix} I_r & 0 \\ 0 & Z_2 \end{bmatrix} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} X \\
&= U_{\mathcal{S}} (W_i')_{1:r,.} \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_{i-1} \cdots Z_2 Z_1 \end{bmatrix} X \\
&= U_{\mathcal{S}} (W_i')_{1:r,1:r} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} X + U_{\mathcal{S}} (W_i')_{1:r,r+1:d_{i-1}} Z_{i-1} \cdots Z_2 Z_1 X\ .
\end{aligned}
$$

Note that the above calculations are still valid in the case $i = 2$. In this case, using the conventions of Section 3.2, $W_{i-1} \cdots W_2 = I_{d_1}$ and $Z_{i-1} \cdots Z_2 = I_{d_2-r}$.
This proves (3.F.9).

Consider finally the case $i = 1$.
Substituting (3.F.3) in (3.4.7), we have

$$
\begin{aligned}
T_1 &= (W_H \cdots W_2) W_1' X \\
&= [U_{\mathcal{S}}\ ,\ 0]\, W_1' X \\
&= U_{\mathcal{S}} (W_1')_{1:r,.} X\ .
\end{aligned}
$$

This proves (3.F.10).
Note that, using the conventions of Section 3.2, the proof still holds for $r = 0$. In this case, $T_i = 0, \forall i$.
This concludes the proof.

3.F.1.13 Proof of Lemma 22

Let $\mathbf{W} = (W_H, \cdots, W_1)$ be a tightened first-order critical point satisfying the hypotheses of Proposition 11, and r, $\mathcal{S}$, Q, $(Z_h)_{h=1..H}$ defined as in Proposition 11. Since $\mathbf{W}$ satisfies the hypotheses of Proposition 11, we are going to use all the equations (3.4.3), (3.4.4), (3.4.5) and (3.4.6) defined by these hypotheses and (3.F.3), (3.F.4), (3.F.5) and (3.F.6) of Lemma 14.
Let $\mathbf{W}' = (W_H', \ldots, W_1')$ and $(i, j) \in [\![1, H]\!]^2$, with $i > j$. Recall that $T_{i,j}$ is defined in

(3.4.8) and $J_1 = [\![p, H-1]\!]$, $J_2 = [\![q+1, p-1]\!]$, $J_3 = [\![2, q]\!]$, where p and q are defined as in Lemma 14.

Consider the case $i \in \{H\} \cup J_1$ and $j \in J_1 \cup J_2$ with $i > j$.
Applying Lemma 20 to (3.4.8) and using (3.4.5) and (3.4.4), we obtain

$$\begin{aligned}
T_{i,j} &= \left\langle W_H \cdots W_{i+1} W_i' W_{i-1} \cdots W_{j+1} W_j' W_{j-1} \cdots W_1 X \ , \ W_H \cdots W_1 X - Y \right\rangle \\
&= \left\langle W_H \cdots W_{i+1} W_i' W_{i-1} \cdots W_{j+1} W_j' W_{j-1} \cdots W_1 \ , \ -U_Q U_Q^T \Sigma_{YX} \right\rangle \\
&= -tr\left(W_H \cdots W_{i+1} W_i' W_{i-1} \cdots W_{j+1} W_j' W_{j-1} \cdots W_1 \Sigma_{XY} U_Q U_Q^T \right) \\
&= -tr\left((W_H \cdots W_{i+1} W_i' W_{i-1} \cdots W_{j+1} W_j' \begin{bmatrix} U_S^T \Sigma U_Q \\ Z_{j-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q \end{bmatrix} U_Q^T \right) .
\end{aligned}$$

Using Lemma 7 and since $j \geq q+1$, using (3.F.5), we obtain

$$T_{i,j} = 0.$$

This proves (3.F.19) and (3.F.22).

Consider now the case $i = H$ and $j \in J_3$.
Applying Lemma (20) to (3.4.8) and using (3.F.6), (3.4.5) and (3.4.4), we obtain

$$\begin{aligned}
T_{H,j} &= \left\langle W_H' W_{H-1} \cdots W_{j+1} W_j' W_{j-1} \cdots W_1 X \ , \ W_H \cdots W_1 X - Y \right\rangle \\
&= \left\langle W_H' W_{H-1} \cdots W_{j+1} W_j' W_{j-1} \cdots W_1 \ , \ -U_Q U_Q^T \Sigma_{YX} \right\rangle \\
&= \left\langle W_H' \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} W_j' \begin{bmatrix} U_S^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_{j-1} \cdots Z_2 Z_1 \end{bmatrix} \ , \ U_Q U_Q^T \Sigma_{YX} \right\rangle \\
&= -tr\left(W_H' \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} W_j' \begin{bmatrix} U_S^T \Sigma U_Q U_Q^T \\ Z_{j-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q U_Q^T \end{bmatrix} \right) .
\end{aligned}$$

Using Lemma 15, Lemma 7 and the cyclic property of the trace, we have

$$\begin{aligned}
T_{H,j} &= -tr\left((W_H')_{.,1:r} (W_j')_{1:r,.} \begin{bmatrix} 0 \\ Z_{j-1} \cdots Z_2 Z_1 X V_Q \Delta^{(Q)} U_Q^T \end{bmatrix} \right) \\
&= -tr\left((W_H')_{.,1:r} (W_j')_{1:r,r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q \Delta^{(Q)} U_Q^T \right) \\
&= -tr\left(\Delta^{(Q)} U_Q^T (W_H')_{.,1:r} (W_j')_{1:r,r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q \right) \\
&= -\left\langle \Delta^{(Q)} U_Q^T (W_H')_{.,1:r} \ , \ \left((W_j')_{1:r,r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 X V_Q \right)^T \right\rangle .
\end{aligned}$$

This proves (3.F.17).

Consider now the case $i = H$ and $j = 1$.
Applying Lemma 20 to (3.4.8) and using (3.F.6) and Lemma 15, we obtain

$$T_{H,1} = \left\langle W_H' W_{H-1} \cdots W_2 W_1' X \ , \ W_H \cdots W_1 X - Y \right\rangle$$

$$
\begin{aligned}
&= \langle W'_H W_{H-1} \cdots W_2 W'_1 \, , \, -U_Q U_Q^T \Sigma_{YX} \rangle \\
&= - \left\langle W'_H \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} W'_1 \, , \, U_Q (X V_Q \Delta^{(Q)})^T \right\rangle \\
&= - \left\langle (W'_H)_{.,1:r} (W'_1)_{1:r,.} \, , \, U_Q \Delta^{(Q)} V_Q^T X^T \right\rangle \\
&= - \left\langle \Delta^{(Q)} U_Q^T (W'_H)_{.,1:r} \, , \, V_Q^T X^T \left((W'_1)_{1:r,.} \right)^T \right\rangle \\
&= - \left\langle \Delta^{(Q)} U_Q^T (W'_H)_{.,1:r} \, , \, \left((W'_1)_{1:r,.} X V_Q \right)^T \right\rangle \, .
\end{aligned}
$$

This proves (3.F.18).

Consider now the case $i \in J_1$ and $j \in J_3$.
Applying Lemma 20 to (3.4.8) and using (3.4.3), (3.4.5) and (3.4.4) , we obtain

$$
\begin{aligned}
T_{i,j} &= \langle W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_{j+1} W'_j W_{j-1} \cdots W_1 X \, , \, W_H \cdots W_1 X - Y \rangle \\
&= \langle W_H \cdots W_{i+1} W'_i W_{i-1} \cdots W_{j+1} W'_j W_{j-1} \cdots W_1 \, , \, -U_Q U_Q^T \Sigma_{YX} \rangle \\
&= - \left\langle [U_{\mathcal{S}} \, , \, U_Q Z_H Z_{H-1} \cdots Z_{i+1}] \, W'_i W_{i-1} \cdots W_{j+1} W'_j \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_{j-1} \cdots Z_2 Z_1 \end{bmatrix} \, , \, U_Q U_Q^T \Sigma_{XY} \right\rangle \\
&= -tr \left([U_{\mathcal{S}} \, , \, U_Q Z_H Z_{H-1} \cdots Z_{i+1}] \, W'_i W_{i-1} \cdots W_{j+1} W'_j \begin{bmatrix} U_{\mathcal{S}}^T \Sigma \\ Z_{j-1} \cdots Z_2 Z_1 \Sigma_{XY} \end{bmatrix} U_Q U_Q^T \right) \\
&= -tr \left(\left[U_Q^T U_{\mathcal{S}} \, , \, U_Q^T U_Q Z_H Z_{H-1} \cdots Z_{i+1} \right] W'_i W_{i-1} \cdots W_{j+1} W'_j \begin{bmatrix} U_{\mathcal{S}}^T \Sigma U_Q \\ Z_{j-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q \end{bmatrix} \right) \, .
\end{aligned}
$$

Using Lemma 6 and Lemma 7, we have

$$
\begin{aligned}
T_{i,j} &= -tr \left([0 \, , \, Z_H Z_{H-1} \cdots Z_{i+1}] \, W'_i W_{i-1} \cdots W_{j+1} W'_j \begin{bmatrix} 0 \\ Z_{j-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q \end{bmatrix} \right) \\
&= -tr \left(Z_H Z_{H-1} \cdots Z_{i+1} (W'_i)_{r+1:d_i,.} W_{i-1} \cdots W_{j+1} (W'_j)_{.,r+1:d_{j-1}} Z_{j-1} \cdots Z_2 Z_1 \Sigma_{XY} U_Q \right) \, .
\end{aligned} \tag{3.F.26}
$$

Here, since **W** is tightened, taking the tightened pivot (i, j) we have two possible cases: either $\mathrm{rk}(W_{i-1} \cdots W_{j+1}) = r$ or $\mathrm{rk}(W_{j-1} \cdots W_1 \Sigma_{XY} W_H \cdots W_{i+1}) = r$. We treat the two cases separately.
In the first case, using (3.4.5) we have

$$
\begin{aligned}
W_{i-1} \cdots W_{j+1} &= \begin{bmatrix} I_r & 0 \\ 0 & Z_{i-1} \end{bmatrix} \cdots \begin{bmatrix} I_r & 0 \\ 0 & Z_{j+1} \end{bmatrix} \\
&= \begin{bmatrix} I_r & 0 \\ 0 & Z_{i-1} \cdots Z_{j+1} \end{bmatrix} \, .
\end{aligned}
$$

Hence, $\mathrm{rk}(W_{i-1} \cdots W_{j+1}) = r$ implies $Z_{i-1} \cdots Z_{j+1} = 0$ and we conclude that

$$
W_{i-1} \cdots W_{j+1} = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} \, .
$$

Then, using this last equality, (3.F.26) becomes

$$\begin{aligned} T_{i,j} &= -tr\left(Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,.}\begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}(W_j')_{.,r+1:d_{j-1}} Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q\right) \\ &= -tr\left(Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,1:r}(W_j')_{1:r,r+1:d_{j-1}} Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q\right) . \end{aligned} \tag{3.F.27}$$

In the second case, we have $\mathrm{rk}(W_{j-1}\cdots W_1 \Sigma_{XY} W_H \cdots W_{i+1}) = r$. Let us prove that (3.F.27) also holds in this case. Using (3.4.5), (3.4.4), (3.4.3), Lemma 7 and $\mathcal{S} = [\![1, r]\!]$, we have

$$\begin{aligned} &W_{j-1}\cdots W_1 \Sigma_{XY} W_H \cdots W_{i+1} \\ &= \begin{bmatrix} U_{\mathcal{S}}^T \Sigma \\ Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} \end{bmatrix} [U_{\mathcal{S}}\ ,\ U_Q Z_H Z_{H-1}\cdots Z_{i+1}] \\ &= \begin{bmatrix} U_{\mathcal{S}}^T \Sigma U_{\mathcal{S}} & U_{\mathcal{S}}^T \Sigma U_Q Z_H Z_{H-1}\cdots Z_{i+1} \\ Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_{\mathcal{S}} & Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{i+1} \end{bmatrix} \\ &= \begin{bmatrix} diag(\lambda_1, \cdots, \lambda_r) & 0 \\ Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_{\mathcal{S}} & Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{i+1} \end{bmatrix} . \end{aligned}$$

Therefore since $\mathrm{rk}(W_{j-1}\cdots W_1 \Sigma_{XY} W_H \cdots W_{i+1}) = r$ and for all $i \in [\![1, r]\!]$, $\lambda_i \neq 0$, we must have

$$Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{i+1} = 0 . \tag{3.F.28}$$

Using the above equation, and the cyclic property of the trace, (3.F.26) becomes

$$\begin{aligned} T_{i,j} &= -tr\left(Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,.} W_{i-1}\cdots W_{j+1}(W_j')_{.,r+1:d_{j-1}} Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q\right) \\ &= -tr\left(Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,.} W_{i-1}\cdots W_{j+1}(W_j')_{.,r+1:d_{j-1}}\right) \\ &= 0 . \end{aligned}$$

We can use (3.F.28) again to write the equation $T_{i,j} = 0$ in the format of equation (3.F.27). Indeed, we have

$$\begin{aligned} &- tr\left(Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,1:r}(W_j')_{1:r,r+1:d_{j-1}} Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q\right) \\ &= -tr\left(Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,1:r}(W_j')_{1:r,r+1:d_{j-1}}\right) \\ &= 0 \\ &= T_{i,j} . \end{aligned}$$

Therefore, in both cases we have

$$T_{i,j} = -tr\left(Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,1:r}(W_j')_{1:r,r+1:d_{j-1}} Z_{j-1}\cdots Z_2 Z_1 \Sigma_{XY} U_Q\right) .$$

Using Lemma 15, it becomes

$$\begin{aligned} T_{i,j} &= -tr\left(Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,1:r}(W_j')_{1:r,r+1:d_{j-1}}Z_{j-1}\cdots Z_2 Z_1 X V_Q \Delta^{(Q)}\right) \\ &= -tr\left(\Delta^{(Q)} Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,1:r}(W_j')_{1:r,r+1:d_{j-1}}Z_{j-1}\cdots Z_2 Z_1 X V_Q\right) \\ &= -\left\langle \Delta^{(Q)} Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,1:r}\ ,\ \left((W_j')_{1:r,r+1:d_{j-1}}Z_{j-1}\cdots Z_2 Z_1 X V_Q\right)^T\right\rangle\ . \end{aligned}$$

This proves (3.F.20).

Consider now the case $i \in J_1$ and $j = 1$.
Using Lemma 20 to simplify (3.4.8), we have

$$\begin{aligned} T_{i,1} &= \left\langle W_H\cdots W_{i+1}W_i'W_{i-1}\cdots W_2 W_1' X\ ,\ W_H\cdots W_1 X - Y\right\rangle \\ &= \left\langle W_H\cdots W_{i+1}W_i'W_{i-1}\cdots W_2 W_1'\ ,\ -U_Q U_Q^T \Sigma_{YX}\right\rangle\ . \end{aligned}$$

Using Lemma 15 and substituting (3.4.3), (3.4.5), and since $i \geq p$, using (3.F.4) , this becomes

$$\begin{aligned} T_{i,1} &= -\left\langle [U_{\mathcal{S}}\ ,\ U_Q Z_H Z_{H-1}\cdots Z_{i+1}]\, W_i' \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix} W_1'\ ,\ U_Q (X V_Q \Delta^{(Q)})^T\right\rangle \\ &= -\left\langle \Delta^{(Q)}\left[U_Q^T U_{\mathcal{S}}\ ,\ U_Q^T U_Q Z_H Z_{H-1}\cdots Z_{i+1}\right](W_i')_{.,1:r}(W_1')_{1:r,.}\ ,\ (XV_Q)^T\right\rangle\ . \end{aligned}$$

Using Lemma 6, it becomes

$$\begin{aligned} T_{i,1} &= -\left\langle \Delta^{(Q)}\left[0\ ,\ Z_H Z_{H-1}\cdots Z_{i+1}\right](W_i')_{.,1:r}(W_1')_{1:r,.}\ ,\ (XV_Q)^T\right\rangle \\ &= -\left\langle \Delta^{(Q)} Z_H Z_{H-1}\cdots Z_{i+1}(W_i')_{r+1:d_i,1:r}\ ,\ ((W_1')_{1:r,.}XV_Q)^T\right\rangle\ . \end{aligned}$$

This proves (3.F.21).

Consider now the case $i \in J_2 \cup J_3 = [\![2, p-1]\!]$ and $j < i$.
Applying Lemma 20 to (3.4.8) and , since $i < p$, using (3.F.3), we obtain

$$\begin{aligned} T_{i,j} &= \left\langle W_H\cdots W_{i+1}W_i'W_{i-1}\cdots W_{j+1}W_j'W_{j-1}\cdots W_1 X\ ,\ W_H\cdots W_1 X - Y\right\rangle \\ &= \left\langle W_H\cdots W_{i+1}W_i'W_{i-1}\cdots W_{j+1}W_j'W_{j-1}\cdots W_1\ ,\ -U_Q U_Q^T \Sigma_{YX}\right\rangle \\ &= -tr([U_{\mathcal{S}}\ ,\ 0]W_i'W_{i-1}\cdots W_{j+1}W_j'W_{j-1}\cdots W_1 \Sigma_{XY} U_Q U_Q^T) \end{aligned}$$

The cyclic property of the trace and Lemma 6 lead to

$$\begin{aligned} T_{i,j} &= -tr([U_Q^T U_{\mathcal{S}}\ ,\ 0]W_i'W_{i-1}\cdots W_{j+1}W_j'W_{j-1}\cdots W_1 \Sigma_{XY} U_Q) \\ &= 0\ . \end{aligned}$$

This proves (3.F.23) and concludes the proof.
Note that, with the convention of Section 3.2, the proof still holds for $r = 0$. In this case,

$T_{i,j} = 0, \forall i > j$.

3.F.2 Proof of Proposition 10

Let $\mathbf{W} = (W_H, \dots, W_1)$ be a tightened first-order critical point associated with $\mathcal{S} = [\![1, r]\!]$ with $r < r_{max}$. Then, using Proposition 5 there exist invertible matrices $D_{H-1} \in \mathbb{R}^{d_{H-1} \times d_{H-1}}, \dots, D_1 \in \mathbb{R}^{d_1 \times d_1}$ and matrices $Z_H \in \mathbb{R}^{(d_y - r) \times (d_{H-1} - r)}$, $Z_1 \in \mathbb{R}^{(d_1 - r) \times d_x}$ and $Z_h \in \mathbb{R}^{(d_h - r) \times (d_{h-1} - r)}$ for $h \in [\![2, H-1]\!]$ such that if we denote $\widetilde{W}_H = W_H D_{H-1}$, $\widetilde{W}_1 = D_1^{-1} W_1$ and $\widetilde{W}_h = D_h^{-1} W_h D_{h-1}$ for all $h \in [\![2, H-1]\!]$, and $\widetilde{\mathbf{W}} = (\widetilde{W}_H, \dots, \widetilde{W}_1)$, then

$$\begin{aligned}
\widetilde{W}_H &= [U_{\mathcal{S}}, U_Q Z_H] \\
\widetilde{W}_1 &= \begin{bmatrix} U_{\mathcal{S}}^T \Sigma_{YX} \Sigma_{XX}^{-1} \\ Z_1 \end{bmatrix} \\
\widetilde{W}_h &= \begin{bmatrix} I_r & 0 \\ 0 & Z_h \end{bmatrix} \quad \forall h \in [\![2, H-1]\!] \\
\widetilde{W}_H \cdots \widetilde{W}_2 &= [U_{\mathcal{S}}, 0] \ .
\end{aligned}$$

Then, due to Lemma 2, and since $\mathbf{W}$ is a first-order critical point, we have that $\widetilde{\mathbf{W}}$ is a first-order critical point. We also have $\widetilde{W}_H \cdots \widetilde{W}_1 = W_H \cdots W_1$. Hence, according to Proposition 1 $\widetilde{\mathbf{W}}$ is also associated with $\mathcal{S}$.
Since $\mathbf{W}$ is tightened and multiplication by invertible matrices does not change the rank, $\widetilde{\mathbf{W}}$ is also tightened. Hence, $\widetilde{\mathbf{W}}$ satisfies the hypotheses of Proposition 11 and therefore is a second-order critical point. Finally, using Lemma 2, we conclude that $\mathbf{W}$ is a second-order critical point. Since $r < r_{max}$ and Σ is invertible (Lemma 4), using Proposition 1, we have

$$L(\mathbf{W}) = \operatorname{tr}(\Sigma_{YY}) - \sum_{i=1}^{r} \lambda_i > \operatorname{tr}(\Sigma_{YY}) - \sum_{i=1}^{r_{max}} \lambda_i \ .$$

Therefore, $\mathbf{W}$ is not a global minimizer, hence $\mathbf{W}$ is a non-strict saddle point.

3.G A simple illustrative experiment

Next we provide more details on the experiment whose results were plotted in Figures 2.3 and 2.4. The goal is to illustrate the behavior of the ADAM optimizer in the vicinity of strict or non-strict saddle points.

Experimental setting. We optimize a linear neural network starting in the vicinity either of a strict saddle point (10000 runs in total) or of a non-strict saddle point (10000 runs in total). For each run, the setting is the following:

— Network architecture: $d_x = 10$, $d_y = 4$, $H = 5$ and $d_4 = d_3 = d_2 = d_1 = 10$.

— Data construction: $m = 100$ i.i.d. data points $(x_1, y_1), \dots, (x_m, y_m) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ such that, for all $i = 1, \dots, m$, the points x_i and y_i are drawn independently at

random from the Gaussian distributions $\mathcal{N}(0,\, I_{d_x})$ and $\mathcal{N}(0,\, I_{d_y})$ respectively.

— Initial iterate: we define it as $(W_1, \dots, W_H) = \mathbf{W}^{cp} + (V_1, \dots, V_H)$, for a critical point $\mathbf{W}^{cp}$ (defined later) and a random perturbation $(V_1, \dots, V_H)$ whose components $(V_h)_{i,j}$ are drawn independently from the distributions $\mathcal{N}(0, \sigma_h^2)$, with $\sigma_h = 0.1 \frac{\|W_h^{cp}\|_F}{\sqrt{d_{h-1} d_h}}$. The critical point $\mathbf{W}^{cp}$ is defined as in (3.B.10) in Appendix 3.B.8, for $r = 2$ ($\mathcal{S} = \{1, 2\}$) and

$$Z_h = \begin{cases} I_{d_h - 2} & \text{for all } h \in [\![2, 4]\!], \text{ for runs starting at a strict saddle point;} \\ 0_{(d_h-2)\times(d_h-2)} & \text{for all } h \in [\![2, 4]\!], \text{ for runs starting at a non-strict saddle point.} \end{cases}$$

Since $d_4 = d_3 = d_2 = d_1$, note that the sizes of the above matrices Z_h are consistent with (3.B.10). As explained in Appendix 3.B.8, when $Z_h = I_{d_h-2}$ for all $h \in [\![2, 4]\!]$, the critical point $\mathbf{W}^{cp}$ is non-tightened and therefore Theorem 5 guarantees that it is a strict saddle point. Similarly, when $Z_h = 0_{(d_h-2)\times(d_h-2)}$ for all $h \in [\![2, 4]\!]$, the critical point $\mathbf{W}^{cp}$ is tightened and Theorem 5 guarantees that it is a non-strict saddle point.

— Optimizer: we use the ADAM optimizer of the Keras library, with the default parameters.

Observations. Figure 2.3 in Section 2.3.3 shows the evolution of the loss along the optimization process for two representative runs (initialization near a strict or a non-strict saddle point). We can see that, when initialized in the vicinity of the strict saddle point, ADAM rapidly decreases below the initial value $L(\mathbf{W}^{cp})$. On the contrary, ADAM needs many epochs to exit the plateau at the critical value of the non-strict saddle point.

In order to assess the importance of this phenomenon, we repeated the above experiment 10000 times for both strict saddle points and non-strict saddle points. For each run, we define and compute the *escape epoch* as the first epoch such that $L(\mathbf{W}) < L(\mathbf{W}^{cp}) - \frac{\lambda_3}{2}$ (the average of the critical values associated with $\mathcal{S} = \{1, 2\}$ and $\mathcal{S}' = \{1, 2, 3\}$). On Figure 2.4 (Section 2.3.3) the histograms of the escape epoch are displayed separately for runs corresponding to strict saddle points (in red) or non-strict saddle points (in blue). We can see that, while ADAM quickly escapes from the vicinity of the strict saddle points, it takes many more epochs to escape from the vicinity of the non-strict saddle points. In the last case, the plateau can easily be confused with a global minimum.

CHAPTER 4

Existence, Stability and Scalability of Orthogonal Convolutional Neural Networks

Abstract

Imposing orthogonality on the layers of neural networks is known to facilitate the learning by limiting the exploding/vanishing of the gradient; decorrelate the features; improve the robustness. This chapter studies the theoretical properties of orthogonal convolutional layers.

We establish necessary and sufficient conditions on the layer architecture guaranteeing the existence of an orthogonal convolutional transform. The conditions prove that orthogonal convolutional transforms exist for almost all architectures used in practice for 'circular' padding. We also exhibit limitations with 'valid' boundary conditions and 'same' boundary conditions with zero-padding.

Recently, a regularization term imposing the orthogonality of convolutional layers has been proposed, and impressive empirical results have been obtained in different applications [142]. The second motivation of the present chapter is to specify the theory behind this. We make the link between this regularization term and orthogonality measures. In doing so, we show that this regularization strategy is stable with respect to numerical and optimization errors and that, in the presence of small errors and when the size of the signal/image is large, the convolutional layers remain close to isometric.

The theoretical results are confirmed with experiments and the landscape of the regularization term is studied. Experiments on real datasets show that when orthogonality is used to enforce robustness, the parameter multiplying the regularization term can be used to tune a tradeoff between accuracy and orthogonality, for the benefit of both accuracy and robustness.

Altogether, the study guarantees that the regularization proposed in [142] is an efficient, flexible and stable numerical strategy to learn orthogonal convolutional layers.

This chapter is based on joint work with François Malgouyres and Franck Mamalet (to appear at JMLR).

Contents

4.1 Introduction

As we have seen in Chapter 2, Section 2.2.5, robustness is an important aspect one would like to impose in machine learning problems. One of the ways to make neural networks more robust to adversarial attacks is to impose orthogonality on their layers. We refer the reader to Chapter 2, Section 2.4, for the motivation and the related works for this chapter.

Recall that the architecture of a convolutional layer is characterized by (M, C, k, S), where M is the number of output channels, C of input channels, convolution kernels are of size $k \times k$ and the stride parameter is S. Unless we specify otherwise, we consider convolutions with circular boundary conditions [1]. Thus, applied on input channels of size $SN \times SN$, the M output channels are of size $N \times N$. We denote by $\mathbf{K} \in \mathbb{R}^{M \times C \times k \times k}$ the kernel tensor and by $\mathcal{K} \in \mathbb{R}^{MN^2 \times CS^2N^2}$ the matrix that applies the convolutional layer of architecture (M, C, k, S) to C vectorized channels of size $SN \times SN$.

Recall that we want to answer the important questions:

— **Existence:** What is a necessary and sufficient condition on (M, C, k, S) and N such that there exists an orthogonal convolutional layer (i.e. $\mathcal{K}$ orthogonal) for this architecture? How do the 'valid' and 'same' boundary conditions restrict the orthogonality existence?

Besides, we will rely on recently published papers [142, 114] which characterize orthogonal convolutional layers as the zero level set of a particular function that is called L_{orth} in [142] [2] (see Section 4.1.1.2 for details). Formally, $\mathcal{K}$ is orthogonal if and only if $L_{orth}(\mathbf{K}) = 0$. They use L_{orth} as a regularization term and obtain impressive performances on several machine learning tasks (see [142]). The regularization is later successfully applied for medical image segmentation [159], inpainting [82] and few-shot learning [110].

We also investigate the following theoretical questions:

— **Stability with regard to minimization errors:** Does $\mathcal{K}$ still have good 'approximate orthogonality properties' when $L_{orth}(\mathbf{K})$ is small but non zero? Without this guarantee, it could happen that $L_{orth}(\mathbf{K}) = 10^{-9}$ and $\|\mathcal{K}\mathcal{K}^T - Id\|_2 = 10^9$. This would make the regularization with L_{orth} useless, unless the algorithm reaches $L_{orth}(\mathbf{K}) = 0$.

— **Scalability and stability with regard to N:** Remarking that, for a given kernel tensor $\mathbf{K}$, $L_{orth}(\mathbf{K})$ is independent of N but the layer transform matrix $\mathcal{K}$ depends on N: When $L_{orth}(\mathbf{K})$ is small, does $\mathcal{K}$ remain approximately orthogonal and isometric when N grows? If so, the regularization with L_{orth} remains efficient even for very large N.

— **Optimization:** Does the landscape of L_{orth} lend itself to global optimization?

We give a positive answer to these questions, thus showing theoretical bounds proving that the regularization with L_{orth} is stable (see Theorem 7, Theorem 8 and Section 4.3.1.3), and can be used in most cases to ensure quasi-orthogonality of the convolutional layers (see Section 4.3.1.1 and Section 4.3.1.2).

We give the main elements of context in Section 4.1.1. The theorems constituting the main contributions of the chapter are in Section 4.2. Experiments illustrating the theorems, on the landscape of L_{orth}, as well as experiments showing the benefits of approximate orthogonality on image classification problems are in Section 4.3. In particular, the latter shows that when orthogonality is used to enforce robustness, the regularization parameter λ

1. Before computing a convolution the input channels are made periodic outside their genuine support.
2. The situation is more complex in [142, 114]. One of the contributions of our work is to clarify the situation. We describe here the clarified statement.

multiplying $L_{orth}(\mathbf{K})$ can be used to tune a tradeoff between accuracy and orthogonality, for the benefit of both accuracy and robustness. The code will be made available in the *DEEL.LIP* [3] library.

For clarity, we only consider convolutional layers applied to images (2D) in the introduction and the experiments. But we emphasize that the theorems in Section 4.2 and their proofs are provided for both signals (1D) and images (2D).

The present chapter specifies the theory supporting the regularization with L_{orth} and the construction of orthogonal convolutional layers. We give necessary and sufficient conditions on the architecture for the orthogonal convolutional layers to exist (see Theorem 6); we unify the L_{orth} formulation for both Row-Orthogonality and Column-Orthogonality cases (see Definition 5); and prove that the regularization with L_{orth}: 1/ is stable, i.e. $L_{orth}(\mathbf{K})$ is small $\implies \mathcal{K}\mathcal{K}^T - \mathrm{Id}_{MN^2}$ is small in various senses (see Theorem 7 and Theorem 8); 2/ leads to an orthogonality error that scales favorably when input signal size N grows (see Theorem 8 and Section 4.3.1.3). We empirically show that, in most cases, the landscape of L_{orth} is such that its minimization can be achieved by *Adam* [80], a standard first-order optimizer (see Section 4.3.1.1). We also identify and analyse the problematic cases (see Section 4.3.1.2). We show numerically that approximate orthogonality is preserved when N increases (see Section 4.3.1.3). Finally, we illustrate on Cifar10 and Imagenette datasets how the regularization parameter can be chosen to control the tradeoff between accuracy and orthogonality, for the benefit of both accuracy and robustness (see Section 4.3.2).

4.1.1 Context

In this section, we describe the context of the chapter by defining orthogonality, the regularization function L_{orth} and the Frobenius and spectral norms of the orthogonality residuals, which are two measures of approximate orthogonality. We relate the latter to an approximate isometry property whose benefits are listed in Table 4.1. The main notations defined in this section are reminded in Table 4.2, in Appendix 4.A.1.

4.1.1.1 Orthogonality

Given a kernel tensor $\mathbf{K}$, the convolutional layer transform matrix $\mathcal{K}$ can be written as:

$$\mathcal{K} = \begin{pmatrix} \mathcal{M}(\mathbf{K}_{1,1}) & \dots & \mathcal{M}(\mathbf{K}_{1,C}) \\ \vdots & \vdots & \vdots \\ \mathcal{M}(\mathbf{K}_{M,1}) & \dots & \mathcal{M}(\mathbf{K}_{M,C}) \end{pmatrix} \in \mathbb{R}^{MN^2 \times CS^2N^2},$$

where $\mathcal{M}(\mathbf{K}_{i,j})$ is a matrix that computes a strided convolution for the kernel $\mathbf{K}_{i,j} = \mathbf{K}_{i,j,:,:}$, from the input channel j, to the output channel i (See Appendix 4.B for details). Notice that we use the 'matlab-colon-notation', such that $\mathbf{K}_{i,j,:,:} = (\mathbf{K}_{i,j,m,n})_{0 \leq m,n \leq k-1} \in \mathbb{R}^{k \times k}$.

In order to define orthogonal matrices, we need to distinguish two cases:

— **Row case (RO case).** When the size of the input space of $\mathcal{K} \in \mathbb{R}^{MN^2 \times CS^2N^2}$ is larger than the size of its output space, i.e. $M \leq CS^2$, $\mathcal{K}$ is orthogonal if and only

3. https://github.com/deel-ai/deel-lip

if its rows are normalized and mutually orthogonal. Denoting the identity matrix $\mathrm{Id}_{MN^2} \in \mathbb{R}^{MN^2 \times MN^2}$, this is written

$$\mathcal{K}\mathcal{K}^T = \mathrm{Id}_{MN^2}\,. \tag{4.1.1}$$

In this case, the mapping $\mathcal{K}$ performs a dimensionality reduction.

— **Column case (CO case).** When $M \geq CS^2$, $\mathcal{K}$ is orthogonal if and only if its columns are normalized and mutually orthogonal:

$$\mathcal{K}^T\mathcal{K} = \mathrm{Id}_{CS^2N^2}\,. \tag{4.1.2}$$

In this case, the mapping $\mathcal{K}$ is an embedding.

Both the RO case and CO case are encountered in practice. When $M = CS^2$, the matrix $\mathcal{K}$ is square and if it is orthogonal then both (4.1.1) and (4.1.2) hold. The matrix $\mathcal{K}$ is then orthogonal in the usual sense and both $\mathcal{K}$ and $\mathcal{K}^T$ are isometric.

4.1.1.2 The function $L_{orth}(\mathbf{K})$

In this section, we define a variant of the function $L_{orth} : \mathbb{R}^{M\times C\times k\times k} \longrightarrow \mathbb{R}$ defined in [142, 114]. The purpose of the proposed variant is to unify the properties of L_{orth} in the RO case and CO case.

Reminding that $k \times k$ is the size of the convolution kernel, for any $h, g \in \mathbb{R}^{k\times k}$ and any $P \in \mathbb{N}$, we define $\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = 1) \in \mathbb{R}^{(2P+1)\times(2P+1)}$ as the convolution [4] between h and the zero-padding of g (see Figure 4.1). Formally, for all $i, j \in [\![0, 2P]\!]$,

$$[\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = 1)]_{i,j} = \sum_{i'=0}^{k-1}\sum_{j'=0}^{k-1} h_{i',j'}\bar{g}_{i+i',j+j'},$$

where $\bar{g} \in \mathbb{R}^{(k+2P)\times(k+2P)}$ is defined, for all $(i, j) \in [\![0, k + 2P - 1]\!]^2$, by

$$\bar{g}_{i,j} = \begin{cases} g_{i-P,j-P} & \text{if } (i, j) \in [\![P, P + k - 1]\!]^2, \\ 0 & \text{otherwise.} \end{cases}$$

We define $\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S) \in \mathbb{R}^{(\lfloor 2P/S\rfloor+1)\times(\lfloor 2P/S\rfloor+1)}$, for all integer $S \geq 1$ and all $i, j \in [\![0, \lfloor 2P/S\rfloor]\!]$, by

$$[\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S)]_{i,j} = [\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = 1)]_{Si,Sj}.$$

We denote (in bold) $\mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) \in \mathbb{R}^{M\times M\times(\lfloor 2P/S\rfloor+1)\times(\lfloor 2P/S\rfloor+1)}$ the fourth-order tensor such that, for all $m, l \in [\![1, M]\!]$,

$$\mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S)_{m,l,:,:}$$

4. As is common in machine learning, we do not flip h.

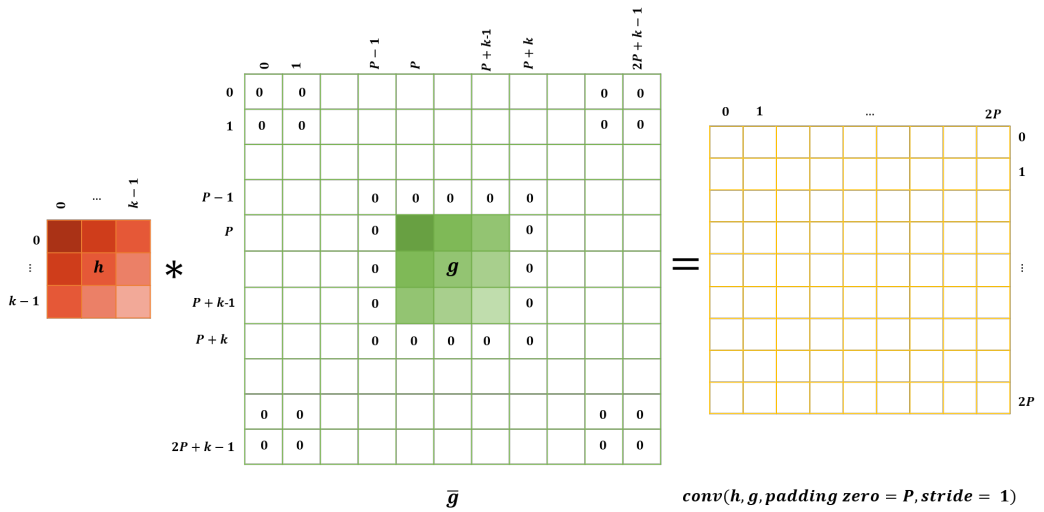

Figure 4.1 – Illustration of $\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = 1)$, in the 2D case.

$$= \sum_{c=1}^{C} \mathrm{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S),$$

where, for all $m \in [\![1, M]\!]$ and $c \in [\![1, C]\!]$, $\mathbf{K}_{m,c} = \mathbf{K}_{m,c,:,:} \in \mathbb{R}^{k \times k}$.

It has been noted in [142] that, in the RO case, when $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$,

$$\mathcal{K} \quad \text{orthogonal} \quad \iff \quad \mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) = \mathbf{I}_{r0}, \tag{4.1.3}$$

where $\mathbf{I}_{r0} \in \mathbb{R}^{M \times M \times (2P/S+1) \times (2P/S+1)}$ is the tensor whose entries are all zero except its central $M \times M$ entry which is equal to an identity matrix: $[\mathbf{I}_{r0}]_{:,:,P/S,P/S} = Id_M$.

Therefore, denoting by $\|.\|_F$ the Euclidean norm in high-order tensor spaces, it is natural to define the following regularization penalty (we justify the CO case right after the definition).

Definition 5 ($\mathbf{L_{orth}}$). We denote by $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$. We define $L_{orth} : \mathbb{R}^{M \times C \times k \times k} \longrightarrow \mathbb{R}_+$ as follows

— In the RO case, $M \leq CS^2$:

$$L_{orth}(\mathbf{K}) = \| \mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) - \mathbf{I}_{r0} \|_F^2 \,. \tag{4.1.4}$$

— In the CO case, $M \geq CS^2$:

$$L_{orth}(\mathbf{K}) = \| \mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) - \mathbf{I}_{r0} \|_F^2 - (M - CS^2) \,.$$

When $M = CS^2$, the two definitions trivially coincide. In the definition, the padding

parameter P is the largest multiple of S strictly smaller than k. The difference with the definitions of L_{orth} in [142, 114] is in the CO case. In this case with $S = 1$, [114, 142] use (4.1.4) with $\mathbf{K}^T$ instead of $\mathbf{K}$. For $S \geq 2$ in the CO case, we can not derive a simple equality as in (4.1.3). In [142], remarking that $\|\mathcal{K}^T\mathcal{K} - \mathrm{Id}_{CS^2N^2}\|_F^2 - \|\mathcal{K}\mathcal{K}^T - \mathrm{Id}_{MN^2}\|_F^2$ is a constant which only depends on the size of $\mathcal{K}$, the authors also argue that, whatever S, one can also use (4.1.4) in the CO case. We alter this in the CO case as in Definition 5 to obtain both in the RO case and the CO case:

$$\mathcal{K} \text{ orthogonal} \qquad \Longleftrightarrow \qquad L_{orth}(\mathbf{K}) = 0.$$

Once adapted to our notations, the authors in [142, 114] propose to regularize convolutional layers parameterized by $(\mathbf{K}_l)_l$ by optimizing

$$L_{task} + \lambda \sum_l L_{orth}(\mathbf{K}_l) \tag{4.1.5}$$

where L_{task} is the original objective function of a machine learning task. The function $L_{orth}(\mathbf{K})$ does not depend on N and can be implemented in a few lines of code with Neural Network frameworks. Its gradient is then computed using automatic differentiation.

Of course, when doing so, even if the optimization is efficient, we expect $L_{orth}(\mathbf{K}_l)$ to be different from 0 but less than ε, for a small ε. We investigate, in this chapter, whether, in this case, the transformation matrix $\mathcal{K}$, still satisfies useful orthogonality properties. To quantify how much $\mathcal{K}$ deviates from being orthogonal, we define the approximate orthogonality criteria and approximate isometry property in the next section. These notions allow to state the stability and scalability theorems (Sections 4.2.3 and 4.2.4) and guarantee that the singular values remain close to 1 when L_{orth} is small, even when N is large. This proves that the benefits related to the orthogonality of the layers, which are presented in Table 4.1, still hold.

4.1.1.3 Approximate orthogonality and Approximate Isometry Property

Perfect orthogonality is an idealization that never happens, due to floating-point arithmetic, and numerical and optimization errors. In order to measure how $\mathcal{K}$ deviates from being orthogonal, we define the **orthogonality residual** by $\mathcal{K}\mathcal{K}^T - \mathrm{Id}_{MN^2}$, in the RO case, and $\mathcal{K}^T\mathcal{K} - \mathrm{Id}_{CS^2N^2}$, in the CO case. Considering both the Frobenius norm $\|.\|_F$ of the orthogonality residual and its spectral norm $\|.\|_2$, we have two criteria:

$$\mathrm{err}_N^F(\mathbf{K}) = \begin{cases} \|\mathcal{K}\mathcal{K}^T - \mathrm{Id}_{MN^2}\|_F & \text{, in the RO case,} \\ \|\mathcal{K}^T\mathcal{K} - \mathrm{Id}_{CS^2N^2}\|_F & \text{, in the CO case,} \end{cases} \tag{4.1.6}$$

and

$$\mathrm{err}_N^s(\mathbf{K}) = \begin{cases} \|\mathcal{K}\mathcal{K}^T - \mathrm{Id}_{MN^2}\|_2 & \text{, in the RO case,} \\ \|\mathcal{K}^T\mathcal{K} - \mathrm{Id}_{CS^2N^2}\|_2 & \text{, in the CO case.} \end{cases} \tag{4.1.7}$$

When $M = CS^2$, the definitions in the RO case and the CO case coincide. The two criteria are of course related since for any matrix $A \in \mathbb{R}^{a\times b}$, the Frobenius and spectral norms are

such that

$$\|A\|_F \leq \sqrt{\min(a,b)}\|A\|_2 \qquad \text{and} \qquad \|A\|_2 \leq \|A\|_F \,. \tag{4.1.8}$$

However, the link is weak, when $\min(a,b)$ is large.

The regularization with $(\mathrm{err}_N^F(\mathbf{K}))^2$ is a natural way to enforce soft-orthogonality of $\mathcal{K}$. However, as mentioned in the introduction, it is not practical because the sizes of $\mathcal{K}$ are too large. We will see in Theorem 7 that $(\mathrm{err}_N^F(\mathbf{K}))^2$ and $L_{orth}(\mathbf{K})$ differ by a multiplicative constant and it will make a clear connection between the regularization with $L_{orth}(\mathbf{K})$ and the regularization with $(\mathrm{err}_N^F(\mathbf{K}))^2$. However, $\mathrm{err}_N^F(\mathbf{K})$ is difficult to interpret, this is why we consider $\mathrm{err}_N^s(\mathbf{K})$ which relates to the *approximate isometry property* of $\mathcal{K}$ as we explain below. The latter has direct consequences on the properties of the layer (see Table 4.1).

Indeed, in the applications, one key property of orthogonal operators is their connection to isometries. It is the property that prevents the gradient from exploding and vanishing [27, 145, 90, 64]. This property also enables to keep the examples well separated, which has an effect similar to the batch normalization [114, 23], and to have a 1-Lipschitz forward pass and therefore improves robustness [142, 27, 91, 138, 70].

We denote the Euclidean norm of a vector by $\|.\|$. To clarify the connection between orthogonality and isometry, we define the 'ε-Approximate Isometry Property' (ε-AIP).

Definition 6. A layer transform matrix $\mathcal{K} \in \mathbb{R}^{MN^2 \times CS^2N^2}$ satisfies the ε-Approximate Isometry Property if and only if

— RO case, $M \leq CS^2$:

$$\begin{cases} \forall x \in \mathbb{R}^{CS^2N^2} & \|\mathcal{K}x\|^2 \leq (1+\varepsilon)\|x\|^2 \\ \forall y \in \mathbb{R}^{MN^2} & (1-\varepsilon)\|y\|^2 \leq \|\mathcal{K}^T y\|^2 \leq (1+\varepsilon)\|y\|^2 \end{cases}$$

— CO case, $M \geq CS^2$:

$$\begin{cases} \forall x \in \mathbb{R}^{CS^2N^2} & (1-\varepsilon)\|x\|^2 \leq \|\mathcal{K}x\|^2 \leq (1+\varepsilon)\|x\|^2 \\ \forall y \in \mathbb{R}^{MN^2} & \|\mathcal{K}^T y\|^2 \leq (1+\varepsilon)\|y\|^2 \end{cases}$$

The following proposition makes the link between $\mathrm{err}_N^s(\mathbf{K})$ and AIP. It shows that minimizing $\mathrm{err}_N^s(\mathbf{K})$ enhances the AIP property.

Proposition 12. Let N be such that $SN \geq k$. We have, both in the RO case and CO case,

$$\mathcal{K} \text{ is } \mathrm{err}_N^s(\mathbf{K})\text{-AIP}.$$

This statement actually holds for any matrix (not only layer transform matrix) and is already stated in [9, 54]. For completeness, we provide proof, in Appendix 4.G.

In Proposition 12 and in Theorem 6 (see the next section), the condition $SN \geq k$ only states that the input width and height are larger than the size of the kernels. This is always the case in practice.

We summarize in Table 4.1 the properties of ε-AIP layers when ε is small, in the different possible scenarios. We remind that a kernel tensor $\mathbf{K}$ can define a convolutional layer or a deconvolution layer. Deconvolution layers are, for instance, used to define layers

Table 4.1 – Properties of a ε-AIP layers (when $\varepsilon \ll 1$), depending on whether $\mathbf{K}$ defines a convolutional or deconvolutional layer. The red crosses indicate when the forward or backward pass performs a dimensionality reduction.

		Forward pass		Backward pass	
		Lipschitz Forward pass	Keep examples separated	Prevent grad. expl.	Prevent grad. vanish.
Convolutional layer	$M < CS^2$	✓	✗	✓	✓
	$M > CS^2$	✓	✓	✓	✗
Deconvolution layer	$M < CS^2$	✓	✓	✓	✗
	$M > CS^2$	✓	✗	✓	✓
Conv. & Deconv.	$M = CS^2$	✓	✓	✓	✓

of the decoder of an auto-encoder or variational auto-encoder [78]. In the convolutional case, $\mathcal{K}$ is applied during the forward pass and $\mathcal{K}^T$ is applied during the backward pass. In a deconvolution layer, $\mathcal{K}^T$ is applied during the forward pass and $\mathcal{K}$ during the backward pass. Depending on whether we have $M < CS^2$, $M > CS^2$ or $M = CS^2$, when $\mathcal{K}$ is ε-AIP with $\varepsilon << 1$, either $\mathcal{K}^T$, $\mathcal{K}$ or both preserve distances (see Table 4.1).

To complement Table 4.1, notice that in the RO case, if $\mathrm{err}_N^F(\mathbf{K}) \leq \varepsilon$, then for any i, j with $i \neq j$, we have $|\mathcal{K}_{i,:}\mathcal{K}_{j,:}^T| \leq \varepsilon$, where $\mathcal{K}_{i,:}$ is the i^{th} line of $\mathcal{K}$. In other words, when ε is small, the features computed by $\mathcal{K}$ are mostly uncorrelated [142].

4.2 Theoretical analysis of orthogonal convolutional layers

This section contains the theoretical contributions of the chapter. In all the theorems in this section, the considered convolutional layers are either applied to a signal, when $d = 1$, or an image, when $d = 2$.

We remind that the architecture of the layer is characterized by (M, C, k, S) where: M is the number of output channels; C is the number of input channels; $k \geq 1$ is an odd positive integer and the convolution kernels are of size k, when $d = 1$, and $k \times k$, when $d = 2$; the stride parameter is S.

We want to highlight that the theorems of Sections 4.2.1, 4.2.3 and 4.2.4 are for convolution operators defined with circular boundary conditions (see Appendix 4.B for details). We point out in Section 4.2.2 restrictions for the 'valid' and 'same' zero-padding boundary conditions (see Appendix 4.D.1 and Appendix 4.D.2 for details).

With circular boundary conditions, all input channels are of size SN, when $d = 1$, $SN \times SN$, when $d = 2$. The output channels are of size N and $N \times N$, respectively when $d = 1$ and 2. When $d = 1$, the definitions of L_{orth}, err_N^F and err_N^s are in Appendix 4.A.2.

In Section 4.2.1, we state a theorem that provides the necessary and sufficient conditions on the architecture for an orthogonal convolutional layer to exist. In Section 4.2.2, we describe restrictions for the 'valid' and 'same' boundary conditions. In Section 4.2.3, we

state a theorem that provides a relation between the Frobenius norm of the orthogonality residual and the regularization penalty L_{orth}. Finally, in Section 4.2.4, we state a theorem that provides an upper bound of the spectral norm of the orthogonality residual using the regularization penalty L_{orth}.

4.2.1 Existence of orthogonal convolutional layers

The next theorem gives a necessary and sufficient condition on the architecture (M, C, k, S) and N for an orthogonal convolutional layer transform to exist. To simplify notations, we denote, for $d = 1$ or 2, the space of all the kernel tensors by

$$\mathbb{K}_d = \begin{cases} \mathbb{R}^{M\times C\times k} & \text{when } d = 1, \\ \mathbb{R}^{M\times C\times k\times k} & \text{when } d = 2. \end{cases}$$

We also denote, for $d = 1$ or 2,

$$\mathbb{K}_d^{\perp} = \{\mathbf{K} \in \mathbb{K}_d |\ \mathcal{K} \text{ is orthogonal}\}.$$

Theorem 6. Let N be such that $SN \geq k$ and $d = 1$ or 2.

— RO case, i.e. $M \leq CS^d$: $\mathbb{K}_d^{\perp} \neq \emptyset$ if and only if $M \leq Ck^d$.

— CO case, i.e. $M \geq CS^d$: $\mathbb{K}_d^{\perp} \neq \emptyset$ if and only if $S \leq k$.

Theorem 6 is proved in Appendix 4.C. Again, the conditions coincide when $M = CS^d$.

When $S \leq k$, which is by far the most common situation, there exist orthogonal convolutional layers in both the CO case and the RO case. Indeed, in the RO case, when $S \leq k$, we have $M \leq CS^d \leq Ck^d$.

However, skip-connection layers (also called shortcut connection) with stride in Resnet [59] for instance, usually have an architecture $(M, C, k, S) = (2C, C, 1, 2)$, where C is the number of input channels. The kernels are of size 1×1. In that case, $M \leq CS^d$ and $M > Ck^d$. Theorem 6 says that there is no orthogonal convolutional layer for this type of layer.

To conclude, the main consequence of Theorem 6 is that, with circular boundary conditions and for most of the architecture used in practice (with an exception for the skip-connections with stride), there exist orthogonal convolutional layers.

4.2.2 Restrictions due to boundary conditions

In Sections 4.2.1, 4.2.3 and 4.2.4, we consider convolutions defined with circular boundary conditions. This choice is neither for technical reasons nor to enable the use of the Fourier basis. We illustrate in the next two propositions that, for convolutions defined with the 'valid' condition, or the 'same' condition with zero-padding, hard-orthogonality is in many situations too restrictive. We consider in this section an unstrided convolution.

Before stating the next proposition, we remind that with the 'valid' boundary conditions, only the entries of the output such that the support of the translated kernel is entirely included in the input's domain are computed. The size of each output channel is smaller than the

size of the input channels. The formal definition of the 'valid' boundary conditions is at the beginning of Appendix 4.D.1.

Proposition 13. Let $N \geq 2k-1$. With the 'valid' condition, there exists no orthogonal convolutional layer in the CO case.

This proposition holds in the 1D and 2D cases. We give its proof only in the 1D case in Appendix 4.D.1.

Before stating the next proposition, we remind that to compute a convolution with the zero-padding 'same' boundary conditions, we first extend each input channel with zeros and then compute the convolutions such that each output channel has the same support as the input channels, before padding/extension. A formal definition of the zero-padding 'same' boundary condition is at the beginning of Appendix 4.D.2.

Let $(e_{i,j})_{i=0..k-1,j=0..k-1}$ be the canonical basis of $\mathbb{R}^{k\times k}$. For the zero-padding 'same', we have the following proposition.

Proposition 14. Let $N \geq k$. For $\mathbf{K} \in \mathbb{R}^{M\times C\times k\times k}$, with the zero-padding 'same' and $S=1$, both in the RO case and CO case, if $\mathcal{K}$ is orthogonal then there exist $(\alpha_{m,c})_{m=1..M,c=1..C} \in \mathbb{R}^{M\times C}$ such that for all $(m,c) \in [\![1,M]\!] \times [\![1,C]\!]$, $\mathbf{K}_{m,c} = \alpha_{m,c} e_{r,r}$, where r satisfies $k = 2r+1$. As a consequence

$$\mathcal{K} = \begin{pmatrix} \alpha_{1,1}Id_{N^2} & \dots & \alpha_{1,C}Id_{N^2} \\ \vdots & \vdots & \vdots \\ \alpha_{M,1}Id_{N^2} & \dots & \alpha_{M,C}Id_{N^2} \end{pmatrix} \in \mathbb{R}^{MN^2\times CN^2} .$$

This proposition holds in the 1D and 2D cases. We give its proof only in the 1D case in Appendix 4.D.2.

To recapitulate, the results state that with padding 'valid', no orthogonal convolution can be built in the CO case and that for zero-padding 'same', the orthogonal convolution layers are trivial transformations.

Note that, the propositions do not exclude the existence of sufficiently expressive sets of 'approximately orthogonal' convolutional layers with these boundary conditions. A strategy based on soft-orthogonality may still enjoy most of the benefits of orthogonality for these boundary conditions.

4.2.3 Frobenius norm stability

We recall that the motivation behind this is the following : The authors of [142, 114] argue that $L_{orth}(\mathbf{K}) = 0$ is equivalent to $\mathcal{K}$ being orthogonal. However, they do not provide stability guarantees. Without this guarantee, it could happen that $L_{orth}(\mathbf{K}) = 10^{-9}$ and $\|\mathcal{K}\mathcal{K}^T - Id\|_F = 10^9$. This would make the regularization with L_{orth} useless, unless the algorithm reaches $L_{orth}(\mathbf{K}) = 0$.

The following theorem proves that this cannot occur. Therefore, if $L_{orth}(\mathbf{K})$ is small, $\mathrm{err}_N^F(\mathbf{K})$ is small at least for moderate signal sizes. Also, a corollary is that adding L_{orth} as a penalty regularization is equivalent to adding the Frobenius norm of the orthogonality residual.

Theorem 7. Let N be such that $SN \geq 2k-1$ and $d = 1$ or 2. For a convolutional layer defined using circular boundary conditions, we have, both in the RO case and CO case,

$$(\mathrm{err}_N^F(\mathbf{K}))^2 = N^d L_{orth}(\mathbf{K}) \ ,$$

where $\mathrm{err}_N^F(\mathbf{K})$ is defined in (4.1.6).

Theorem 7 is proved, in Appendix 4.E. We remind that $L_{orth}(\mathbf{K})$ is independent of N. The theorem formalizes for circular boundary conditions and for both the CO case and the RO case, the reasoning leading to the regularization with L_{orth} in [142].

Using Theorem 7, we find that (4.1.5) becomes

$$L_{task} + \sum_l \frac{\lambda}{N_l^d} (\mathrm{err}_{N_l}^F(\mathbf{K}_l))^2.$$

Once the parameter λ is made dependent on the input size of layer l, the regularization term λL_{orth} is equal to the Frobenius norm of the orthogonality residual. This justifies the use of L_{orth} as a regularizer.

We can also see from Theorem 7 that, for both the RO case and the CO case, when $L_{orth}(\mathbf{K}) = 0$, $\mathcal{K}$ is orthogonal, independently of N. This recovers the result stated in [114] for $S = 1$, and the result stated in [142] in the RO case for any S .

Considering another signal size N' and applying Theorem 7 with the sizes N and N', we find

$$(\mathrm{err}_{N'}^F(\mathbf{K}))^2 = \frac{(N')^d}{N^d} (\mathrm{err}_N^F(\mathbf{K}))^2.$$

To the best of our knowledge, this equality is new. This could be of importance in situations when N varies. For instance when the neural network is learned on a dataset containing signals/images of a given size, but the inference is done for signals/images of varying size [118, 124, 69].

Finally, using (4.1.8) and Proposition 12, $\mathcal{K}$ is ϵ-AIP with ϵ scaling like the square root of the signal/image size. This might not be satisfactory. We exhibit in the next section a tighter bound on ϵ, independent of the input size N.

4.2.4 Spectral norm stability and scalability

We prove in Theorem 8 that $\mathrm{err}_N^s(\mathbf{K})^2$ is sandwiched between two quantities proportional to $L_{orth}(\mathbf{K})$. The multiplicative factors do not depend on N. Hence, when $L_{orth}(\mathbf{K})$ is small, $\mathrm{err}_N^s(\mathbf{K})^2$ is also small for all N. As a consequence, as long as $L_{orth}(\mathbf{K}) \ll 1$ even if the algorithm does not reach $L_{orth}(\mathbf{K}) = 0$, regularizing with $L_{orth}(\mathbf{K})$ permits to construct nearly orthogonal and isometric convolutional layers independently of N.

Moreover, combined with Proposition 12 this ensures that, if $L_{orth}(\mathbf{K})$ is small, $\mathcal{K}$ is ε-AIP with ε small. Using Table 4.1, we see that this property leads to more robustness and avoids gradient vanishing/exploding. This is in line with the empirical results observed in [142, 114].

Theorem 8. Let N be such that $SN \geq 2k-1$ and $d = 1$ or 2. For a convolutional layer

defined using circular boundary conditions, we have

$$\alpha' \, L_{orth}(\mathbf{K}) \;\leq\; (\mathrm{err}_N^s(\mathbf{K}))^2 \;\leq\; \alpha \, L_{orth}(\mathbf{K})$$

where $\mathrm{err}_N^s(\mathbf{K})$ is defined in (4.1.7), for $\alpha' = \frac{1}{\min(M, CS^2)}$ and

$$\alpha = \begin{cases} \left(2\left\lfloor \frac{k-1}{S} \right\rfloor + 1\right)^d M & \text{in the RO case } (M \leq CS^d), \\ (2k-1)^d C & \text{in the CO case } (M \geq CS^d). \end{cases}$$

Theorem 8 is proved, in Appendix 4.F. When $M = CS^d$, the two inequalities hold and it is possible to take the minimum of the two α values.

As we can see from Theorem 8, unlike with the Frobenius norm, the spectral norm of the orthogonality residual is upper-bounded by a quantity that does not depend on N. The lower-bound of Theorem 8 guarantees that the upper-bound is tight up to a multiplicative constant. However, we cannot expect much improvement in this regard since the multiplicative constant $\sqrt{\alpha}$ is usually moderately large [5]. For instance, with $(M, C, k, S) = (128, 128, 3, 2)$, for images, $\sqrt{\alpha} \leq 34$. If as is common in practice the optimization algorithm reaches $L_{orth}(\mathbf{K}) \leq 10^{-6}$, Theorem 8 guarantees that, independently of N,

$$\mathrm{err}_N^s(\mathbf{K}) \leq \sqrt{\alpha L_{orth}(\mathbf{K})} \leq 0.034.$$

Using Proposition 12, independently of N, a convolutional layer defined with $\mathbf{K}$ is ϵ-AIP, for $\epsilon \leq 0.034$, and we have, using Definition 6,

$$\begin{cases} \forall x \in \mathbb{R}^{CS^2N^2} & \|\mathcal{K}x\| \leq \sqrt{1+\varepsilon}\|x\| \leq 1.017\|x\| \\ \forall y \in \mathbb{R}^{MN^2} & 0.982\|y\| \leq \sqrt{1-\varepsilon}\|y\| \leq \|\mathcal{K}^T y\| \leq 1.017\|y\| \end{cases}$$

The layer benefits from the properties described in Table 4.1.

The development done for the above example can be repeated as soon as $L_{orth}(\mathbf{K}) \ll 1$, both in the RO case and CO case. Experiments that confirm this behavior are in Section 4.3.

4.3 Experiments

Before illustrating the benefits of approximate orthogonality to robustly classify images in Section 4.3.2, we conduct several synthetic experiments in Section 4.3.1. The synthetic experiments empirically evaluate the landscape of L_{orth} in Section 4.3.1.1, 4.3.1.2 and illustrate the theorems of Section 4.2, in Section 4.3.1.3.

Both in Section 4.3.1 and Section 4.3.2, to evaluate how close $\mathcal{K}$ is to being orthogonal, we compute some of its singular values σ for different input sizes $SN \times SN$. When $S = 1$, we compute all the singular values of $\mathcal{K}$ with the Algorithm 1, Appendix 4.I, from [121]. For convolutions with stride, $S > 1$, there is no known practical algorithm to compute all the singular values and we simply apply the well-known power iteration algorithm associated with a spectral shift, to retrieve the smallest and largest singular values

5. For usual architectures, $\sqrt{\alpha}$ is always smaller than 200.

$(\sigma_{min}, \sigma_{max})$ of $\mathcal{K}$ (see Algorithm 2 in Appendix 4.I). We remind that $\mathcal{K}$ orthogonal is equivalent to $\sigma_{min} = \sigma_{max} = 1$.

4.3.1 Synthetic experiments

Section 4.3.1.1, 4.3.1.2 and 4.3.1.3 report on results of the massive experiment that is described below.

In order to avoid interaction with other objectives, we train a single 2D convolutional layer with circular padding. We explore all the architectures such that $\mathbb{K}_2^{\perp} \neq \emptyset$, for $C \in [\![1, 64]\!]$, $M \in [\![1, 64]\!]$, $S \in \{1, 2, 4\}$, and $k \in \{1, 3, 5, 7\}$. This leads to 44924 architectures for which an orthogonal convolutional layer exists (among 49152 architectures in total).

For each architecture, the model is trained using a *Glorot uniform* initializer and an *Adam* optimizer [80] with fixed learning rate [6] 0.01 on a null loss ($L_{task}(X, Y, \mathbf{K}) = 0$, for all input X, target Y, and kernel tensor $\mathbf{K}$) and the $L_{orth}(\mathbf{K})$ regularization (see Definition 5) during 3000 steps [7].

We report below implementation details that have no influence on the results, since $L_{task} = 0$. No data are involved in the synthetic experiments and the input of the layer contains a null input of size $(C, 64, 64)$. Other input sizes from 8 to 256 were tested but not reported, leading to the same conclusions. Batch size (which thus has no influence on the results) is set to one.

4.3.1.1 Optimization landscape

For each architecture, we plot on Figure 4.2 the values of σ_{min} and σ_{max} for the obtained $\mathcal{K}$ and $SN \times SN = 64 \times 64$. The experiment for a given architecture (M, C, k, S) is represented by two points: σ_{max}, in blue, and σ_{min}, in orange. For each point (x, y) in Figure 4.2, the first coordinate x corresponds to the ratio $\frac{M}{CS^2}$ of the considered architecture, and the second coordinate y equals the singular value (σ_{min} or σ_{max}) of the obtained $\mathcal{K}$. The points with $x \leq 1$ correspond to the artchitecture in the RO case ($\mathcal{K}$ is a fat matrix), and the others correspond to the architectures in the CO case ($\mathcal{K}$ is a tall matrix).

The right plot of Figure 4.2 shows that all configurations where $M \neq CS^2$ are trained very accurately to near-perfect orthogonal convolutions. These configurations represent the vast majority of cases found in practice. However, the left plot of Figure 4.2 points out that some architectures, with $M = CS^2$, might not fully benefit of the regularization with L_{orth}. These architectures, corresponding to a square $\mathcal{K}$, can mostly be found when $M = C$ and $S = 1$, for instance in VGG [127] and Resnet [59]. We have conducted experiments that we do not report here in detail, and it seems that this is specific to the convolutional case. Fully-connected layers optimized to be orthogonal do not suffer from this phenomenon.

6. We do not report experiments for other tested learning rates $1e-1, 1e-3, 1e-4, 1e-5$ because they lead to the same conclusions.

7. Increasing the number of steps leads to the same conclusions.

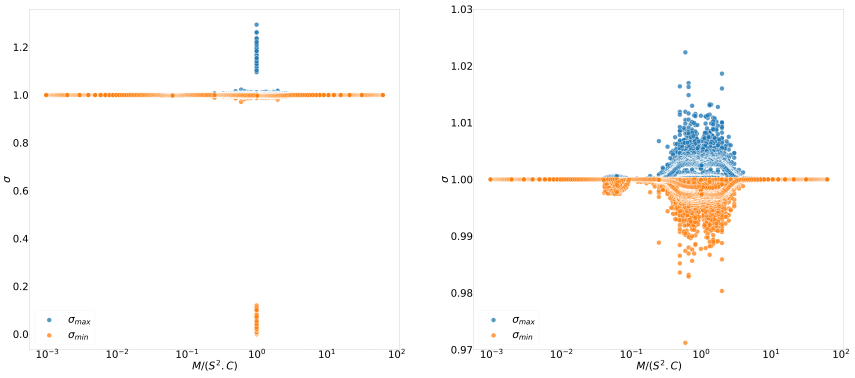

Figure 4.2 – **Optimization of $\mathbf{L_{orth}}$.** Minimization of $L_{orth}(\mathbf{K})$ for a kernel tensor of architecture (M, C, k, S) among the 44924 possible configurations satisfying $\mathbb{K}_2^{\perp} \neq \emptyset$. For each architecture the resulting trained kernel is represented by a blue dot and an orange dot corresponding respectively to its largest and smallest singular values σ_{max} and σ_{min}. The x-axis represents M/CS^2 in log scale. (left) All configurations; (right) All configurations for which $M \neq CS^2$. On the right final convolutions are nearly orthogonal ($\sigma_{max} = \sigma_{min} \approx 1$), but some configurations on the left (where $M = CS^2$) have σ_{max} larger that one, and σ_{min} close to zero.

4.3.1.2 Analysis of the $M = CS^2$ cases

Since we know that $\mathbb{K}_2^{\perp} \neq \emptyset$, the explanation for the failure cases (when σ_{max} or σ_{min} significantly differ from 1) is that the optimization was not successful. We tried many learning rate schemes and iteration numbers but obtained similar results [8].

To evaluate the proportion of successful optimizations when $M = CS^2$, we run 100 training experiments, with independent initialization, for each configuration when $M = CS^2$. In average, after convergence, we found $\sigma_{min} \sim 1 \sim \sigma_{max}$ for 14% of runs, proving that the minimizer can be reached. The explanation of this phenomenon and the evaluation of its impact on applications are open questions that we keep for future research. A contribution of the chapter is to empirically identify these problematic cases.

We display on Figure 4.3 the singular values of $\mathcal{K}$ defined for $S = 1$ and $N \times N = 64 \times 64$ for two experiments where $M = C$. In the experiment on the left, the optimization is successful and the singular values are very accurately concentrated around 1. On the right, we see that only a few of the singular values significantly differ from 1.

Figure 4.3 shows that even if σ_{min} and σ_{max} are not close to 1, as shown in Figure 4.2, most of the singular values are close to 1. This probably explains why the landscape problem does not alter the performance on real datasets in [142] and [114]. Notice that [142] display a curve similar to Figure 4.3 when used for a real dataset.

8. See the description of the experiments at the beginning of Section 4.3.1 and the related footnotes.

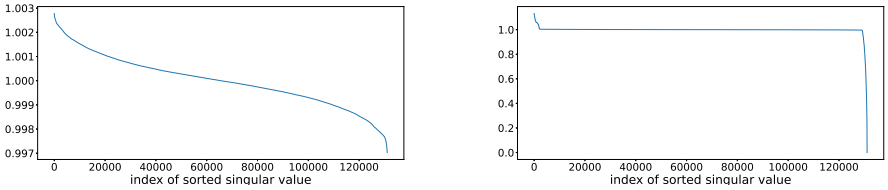

Figure 4.3 – **Singular values of** $\mathcal{K}$, when $C = M$ and $S = 1$. Optimization is (Left) successful, $L_{orth}(\mathbf{K}) \ll 1$, (Right) Unsuccessful, $L_{orth}(\mathbf{K}) \geq 0.1$. On the left, we see that orthogonal convolutions can be reached even when $C = M$ and $S = 1$. On the right we see that, even for unsuccessful optimization, most of the singular values are very close to one.

4.3.1.3 Stability of $(\sigma_{min}, \sigma_{max})$ when N varies

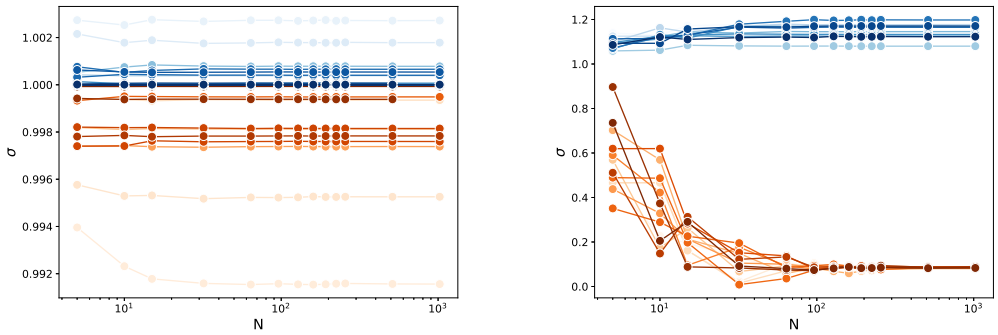

Figure 4.4 – **Evolution of** $\sigma_{\mathbf{min}}$ **and** $\sigma_{\mathbf{max}}$ **according to input image size** (x-axis: N in log-scale). Each line (transparency) represents the singular values σ_{max} (in blue) and σ_{min} (in orange) of $\mathcal{K}$ for different N and a fixed $\mathbf{K}$. (Left) $\mathbf{K}$ is such that $L_{orth}(\mathbf{K}) \ll 1$, singular values remain close to one whatever N. (Right) $\mathbf{K}$ is such that $L_{orth}(\mathbf{K}) \not\ll 1$, the largest (resp. smallest) singular values increase (resp. decrease) when N grows.

In this experiment, we evaluate how the singular values σ_{min} and σ_{max} of $\mathcal{K}$ vary when the parameter N defining the size $SN \times SN$ of the input channels varies, for $\mathbf{K}$ fixed. This is important for applications [124, 69, 118] using fully convolutional networks, or for transfer learning using pre-learnt convolutional feature extractor.

To do so, we randomly select 50 experiments for which the optimization was successful ($L_{orth}(\mathbf{K}) \leq 0.001$) and 50 experiments for which it was unsuccessful ($L_{orth}(\mathbf{K}) \geq 0.02$). They are respectively used to construct the figures on the left and the right side of Figure 4.4. For a given $\mathbf{K}$, we display the singular values σ_{min} and σ_{max} of $\mathcal{K}$ for $N \in \{5, 12, 15, 32, 64, 128, 256, 512, 1024\}$, as orange and blue dots. The dots corresponding to the same $\mathbf{K}$ are linked by a line.

We see, on the left of Figure 4.4, that for successful experiments ($L_{orth}(\mathbf{K}) \ll 1$), the singular values are very stable when N varies. This corresponds to the behavior described in Theorem 8 and Proposition 12. We also point out, on the right of Figure 4.4, that for unsuccessful optimization ($L_{orth} \not\ll 1$), σ_{min} (resp. σ_{max}) values decrease (resp. increase)

rapidly when N increases.

4.3.2 Datasets experiments

In this section we compare, on Cifar10 and Imagenette datasets, the performance, robustness, spectral properties and processing time of three networks: standard convolutional neural networks with unconstrained convolutions called *Conv2D*, the same network architectures with convolutions regularized with L_{orth} and the same network architectures with convolutions constrained with a method that we call *Cayley*, a hard convolutional layer orthogonality method [9] based on the Cayley transform [138]. The latter builds convolutions parameterized by $k \times k$ parameters but, because a mapping is applied to obtain orthogonality, the convolution kernels are of size $N \times N$. In comparison, L_{orth} regularization provides convolutions kernels of size $k \times k$, as is standard. The methods are therefore not expected to provide the same results which makes the comparison a bit complicated. This comparison is also somewhat unfair since the regularization with L_{orth} enjoys a parameter λ. We show results for a wide range of λ but assume, when interpreting the results, that an optimal λ is chosen, for instance using cross-validation.

The design of the experiments aims at simultaneously obtaining good accuracy and robustness. Therefore, for the purpose of robustness, we only use isometric activations and nearly orthogonal convolutional layers. We cannot expect, with this robustness constraint, to obtain clean accuracies as good as those reported in [142].

On Cifar10, we use a VGG-like architecture [127] with nine convolutional layers and a single dense output layer with ten 1-Lipschitz neurons. In all experiments, for a fair comparison, we use invertible downsampling emulation as in [138]. In order to avoid problematic configurations described in Section 4.3.1.2, we alternate channels numbers $C, C+2, C$ within each VGG block (see Table 4.3).

The network is trained during 400 epochs with a batch size of 128, using cross-entropy loss with temperature, Adam optimizer [80] with a decreasing learning rate, and standard data augmentation. A full description of hyperparameters is given in Appendix 4.H.1. The optimized network achieves 91% accuracy on the Cifar10 test set, for the $Conv2D$ classical network.

As already mentioned, three configurations are compared: $Conv2D$: classical convolutions (i.e. no regularization), $Cayley$: convolutions constrained by Cayley method [138], and L_{orth}: the regularization with L_{orth}. For the L_{orth} regularization, we investigate the properties of the solution obtained when minimizing (4.1.5) for $\lambda \in \{10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$.

After training, σ_{max} and σ_{min} values are computed for each convolutional layer using the method described in Appendix 4.I. Each configuration is learnt 10 times to provide mean and standard deviation for the following metrics:

— *Acc. clean*: Classical accuracy on a clean test set

— $\Sigma_{max} = \max_l(\sigma_{max}(\mathcal{K}_l))$: the largest singular value among all the convolutional layers's singular values.

9. Our experiments complement the comparison of L_{orth} with kernel orthogonality methods in [142].

— $\Sigma_{min} = \min_l(\sigma_{min}(\mathcal{K}_l))$: the smallest singular value among all the convolutional layers's singular values.

— E_{lip} : Empirical local Lipschitz constants of the network computed using the PGD-like method proposed by [150].

— E_{rob} : The empirical robustness accuracy, i.e. the proportion of test samples on which a vanilla Projected Gradient Descent (PGD) attack [94] failed (for a robustness radius $\epsilon = 36/255$). PGD attack is applied with 10 iterations and a factor $\alpha = \epsilon/4.0$.

— T_{epoch}: the average epoch processing time.

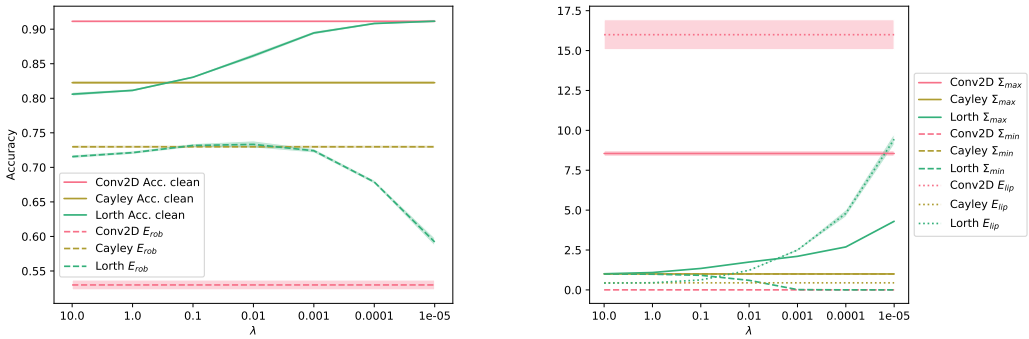

Figure 4.5 – **Cifar10**: Mean evolution of metrics (and error band in shadow) according to λ parameter for L_{orth} method, and comparison with $Conv2D$ and $Cayley$ configurations (constant values): (Left) Clean accuracy and empirical robustness for $\epsilon = 36/255$, (Right) Σ_{max}, Σ_{min} and E_{lip} metrics. The parameter λ permits tune a tradeoff between accuracy and orthogonality, for the benefit of a better accuracy and a better robustness.

Figure 4.5 shows that the regularization parameter λ, in (4.1.5), provides a way to tune a tradeoff between robustness (E_{rob}) and clean accuracy ($Acc.clean$), by controlling the singular values of the layers (Σ_{max} and Σ_{min}). On the contrary $Cayley$ or $Conv2D$ each provide a single tradeoff (shown with constant value in figures). The configurations $\lambda = 1.00 \times 10^{-1}$ and 1.00×10^{-2} achieve better clean accuracy and similar empirical robustness performances as the $Cayley$ method. Furthermore, their empirical Lipschitz constants are very close to one. Finally, error bands for $Cayley$ and L_{orth} methods are very narrow.

Processing time T_{epoch} for regularizing with L_{orth} is only 5% slower than the reference network $Conv2D$, but 2.2 times faster than the one for the $Cayley$ method. It is not reported here in detail but the convergence speeds, in number of epochs, are similar. Moreover, L_{orth} provides classical convolution at inference. On the contrary, the $Cayley$ method provides orthogonal convolutions of size $N \times N$ obtained using a mapping that involves Fourier transforms, which leads to higher computational complexity even at inference. The change of support can also explain the slight difference in $Acc.clean$ between the $Cayley$ method and the strong regularization $\lambda = 10$ for L_{orth} method.

Figure 4.6 presents the same experiments on the Imagenette dataset [63]. The latter is a 10-class subset of Imagenet dataset [30] with 160×160 images. The architecture is also a

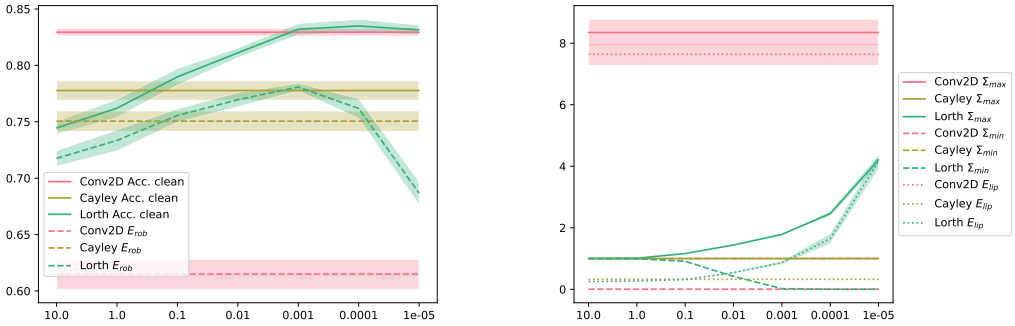

Figure 4.6 – **Imagenette**: Mean evolution of metrics (and error band in shadow) according to λ parameter for L_{orth} method, and comparison with $Conv2D$ and $Cayley$ configurations (constant values): (Left) Clean accuracy and empirical robustness for $\epsilon = 36/255$, (Right) Σ_{max}, Σ_{min} and E_{lip} metrics. The parameter λ permits to tune a tradeoff between accuracy and orthogonality, for the benefit of a better accuracy and a better robustness.

VGG-like one but with 15 convolutional layers. We trained 10 times during 400 epochs with a batch size of 64, using the same loss and optimizer as for the Cifar10 experiments. The architecture and hyperparameters are described in Section 4.H.2. The average performance of the unconstrained $Conv2D$ networks is about 83%.

Interestingly, even if the image size is larger on the Imagenette dataset, L_{orth} regularization shows the same profile as for Cifar10, when λ decreases, ranging from strong orthogonality to high clean accuracy (equivalent to Conv2D configuration), with the best compromise for $\lambda = 1.00 \times 10^{-3}$. Notice that for λ decreasing from 10 to 0.01 the loss of orthogonality permits to obtain a significantly better accuracy but does not significantly favor attacks. Altogether, this leads to an increase in the empirical robustness accuracy. The phenomenon is present but less visible on Cifar10.

Besides, because L_{orth} does not depend on the size parameter N of the input channels, the processing time for the L_{orth} regularization is only 1.1 times slower than for the non-constrained convolution $Conv2D$. In comparison, the $Cayley$ method is 6.5 slower than $Conv2D$.

4.4 Conclusion

This chapter provides a necessary and sufficient condition on the architecture for the existence of an orthogonal convolutional layer with circular padding. The conditions prove that orthogonal convolutional layers exist for most relevant architectures. We show that the situation is less favorable with 'valid' and 'same' zero-paddings. We also prove that the minimization of the surrogate L_{orth} enables constructing orthogonal convolutional layers in a stable manner, that also scales well with the input size parameter N. The experiments confirm that this is practically the case for most of the configurations, except when $M = CS^2$ for which interrogations remain.

Altogether, the study guarantees that the regularization with L_{orth} is an efficient, stable numerical strategy to learn orthogonal convolutional layers. It can safely be used even when the signal/image size is very large. The regularization parameter λ is chosen depending on the tradeoff we want between accuracy and orthogonality, for the benefit of both accuracy and robustness.

Let us mention three open questions related to this chapter. First, a better understanding of the landscape problem as well as solutions to this problem when $M = CS^2$ could be useful. Also, as initiated in [77, 38], the extension of Lipschitz and orthogonal constraints and regularization to the attention-based networks is a natural and relevant open question. Finally, a clean adaptation of the regularization with L_{orth} for the 'valid' and 'same' boundary conditions is needed. As shown in Section 4.2.2, approximate orthogonality seems to be key with these boundary conditions.

Appendix

4.A Notation and definitions

4.A.1 Notation

We summarize the notations specific to our problem in Table 4.2. We then describe mathematical notations, their adaptation to our context and notations for the canonical bases of matrix spaces that appear in the proofs.

notation	domain/type	description
M	$\mathbb{N}$	number of output channels
C	$\mathbb{N}$	number of input channels
k	$\mathbb{N}$, odd	the convolution kernel is of support k for signals, $k \times k$ for images
d	{1,2}	1 when the layer applies to signals; 2 for images
S	$\mathbb{N}$	stride/sampling parameter
N	$\mathbb{N}$	input channels are of size SN for signals, $SN \times SN$ for images output channels are of size N for signals, $N \times N$ for images
$\mathbb{K}_d$	vector space	equal to $\mathbb{R}^{M\times C\times k\times k}$ for images or $\mathbb{R}^{M\times C\times k}$ for signals
$\mathbf{K}$	$\mathbb{K}_d$	kernel tensor that contains all weights defining the layer
$\mathcal{K}$	$\mathbb{R}^{MN^2\times CS^2N^2}$ or $\mathbb{R}^{MN\times CSN}$	the matrix that applies the convolutional layer defined by $\mathbf{K}$ to inputs of size defined by N
$\mathbb{K}_d^{\perp}$	subset of $\mathbb{K}_d$	kernel tensors $\mathbf{K}$ such that $\mathcal{K}$ is orthogonal
$\mathbf{K}_{i,j}$	$\mathbb{R}^{k\times k}$ or $\mathbb{R}^{k}$	weights of the convolution from input channel j to output channel i
$\mathcal{M}(\mathbf{K}_{i,j})$	$\mathbb{R}^{N^2\times S^2N^2}$ or $\mathbb{R}^{N\times SN}$	matrix that applies the strided convolution defined by $\mathbf{K}_{i,j}$ to inputs of size defined by N
$L_{orth}(\mathbf{K})$	$\mathbb{R}_+$	regularization applied to $\mathbf{K}$ and enforcing orthogonality of $\mathcal{K}$, see Definition 5
$\mathrm{err}_N^F(\mathbf{K})$	$\mathbb{R}_+$	measures, in Frobenius norm, how matrix $\mathcal{K}$ for the signal size N deviates from being orthogonal,
$\mathrm{err}_N^s(\mathbf{K})$	$\mathbb{R}_+$	same as above for the spectral norm
$\mathbf{I}_{r0}$	tensor	tensor appearing in the definition of L_{orth}

Table 4.2 – Summary of the main notations.

The floor of a real number will be denoted by $\lfloor . \rfloor$. For two integers a and b, $[\![a, b]\!]$ denotes the set of integers n such that $a \leq n \leq b$. We also denote by $a\%b$ the rest of the euclidean division of a by b, and $[\![a, b]\!]\%n = \{x\%n | x \in [\![a, b]\!]\}$. We denote by $\delta_{i=j}$, the Kronecker symbol, which is equal to 1 if $i = j$, and 0 if $i \neq j$.

We denote by 0_s the null vector of $\mathbb{R}^s$. For a matrix $A \in \mathbb{R}^{m\times n}$, $\sigma_{max}(A)$ denotes the largest singular value of A and $\|A\|_2 = \sigma_{max}(A)$ is its spectral norm. We also have $\|A\|_1 = \max_{0\leq j\leq n-1} \sum_{i=0}^{m-1} |A_{i,j}|$ and $\|A\|_\infty = \max_{0\leq i\leq m-1} \sum_{j=0}^{n-1} |A_{i,j}|$. We denote by $\mathrm{Id}_n \in \mathbb{R}^{n\times n}$ the identity matrix of size n. We use 'Matlab colon notation' as index of matrices and tensors. For instance, $A_{i,:}$ is the i^{th} line of A.

Recall that $\|.\|_F$ denotes the norm which, to any tensor of order larger than or equal to 2, associates the square root of the sum of the squares of all its elements (e.g., for a matrix it corresponds to the Frobenius norm).

Recall that S is the stride parameter, $k = 2r + 1$ is the size of the 1D kernels. SN is the size of the input channels and N is the size of the output channels.
For a vector space $\mathcal{E}$, we denote by $\mathcal{B}(\mathcal{E})$ its canonical basis. We set

$$\begin{cases} (e_i)_{i=0..k-1} = \mathcal{B}(\mathbb{R}^k) \\ (f_i)_{i=0..SN-1} = \mathcal{B}(\mathbb{R}^{SN}) \\ (E_{a,b})_{a=0..N-1,b=0..SN-1} = \mathcal{B}(\mathbb{R}^{N\times SN}) \\ (\overline{E}_{a,b})_{a=0..SN-1,b=0..N-1} = \mathcal{B}(\mathbb{R}^{SN\times N}) \\ (F_{a,b})_{a=0..SN-1,b=0..SN-1} = \mathcal{B}(\mathbb{R}^{SN\times SN}) \\ (G_{a,b})_{a=0..N-1,b=0..N-1} = \mathcal{B}(\mathbb{R}^{N\times N}) \ . \end{cases} \tag{4.A.1}$$

Note that the indices start at 0, thus we have for example $e_0 = \begin{bmatrix} 1 \\ 0_{k-1} \end{bmatrix}$, $e_{k-1} = \begin{bmatrix} 0_{k-1} \\ 1 \end{bmatrix}$, and for all $i \in [\![1, k-2]\!]$, $e_i = \begin{bmatrix} 0_i \\ 1 \\ 0_{k-i-1} \end{bmatrix}$.
To simplify the calculations, the definitions are extended for a, b outside the usual intervals, it is done by periodization. Hence, for all $a, b \in \mathbb{Z}$, denoting by $\hat{a} = a\%SN$, $\tilde{a} = a\%N$, and similarly $\hat{b} = b\%SN$, $\tilde{b} = b\%N$, we set

$$\begin{cases} e_a = e_{a\%k}, \qquad f_a = f_{\hat{a}} \\ E_{a,b} = E_{\tilde{a},\hat{b}}, \qquad \overline{E}_{a,b} = \overline{E}_{\hat{a},\tilde{b}}, \qquad F_{a,b} = F_{\hat{a},\hat{b}}, \qquad G_{a,b} = G_{\tilde{a},\tilde{b}}. \end{cases} \tag{4.A.2}$$

Therefore, for all $a, b, c, d \in \mathbb{Z}$, we have

$$\begin{cases} E_{a,b}F_{c,d} = \delta_{\hat{b}=\hat{c}}E_{a,d}, \qquad E_{a,b}\overline{E}_{c,d} = \delta_{\hat{b}=\hat{c}}G_{a,d} \\ \overline{E}_{a,b}E_{c,d} = \delta_{\tilde{b}=\tilde{c}}F_{a,d}, \qquad F_{a,b}\overline{E}_{c,d} = \delta_{\hat{b}=\hat{c}}\overline{E}_{a,d}. \end{cases} \tag{4.A.3}$$

Note also that

$$E_{a,b}^T = \overline{E}_{b,a} \ . \tag{4.A.4}$$

4.A.2 Corresponding 1D definitions

In this section, we give the definitions for signals (1D case), of the objects defined in the introduction for images (2D case).

4.A.2.1 Orthogonality

As in Section 4.1.1.1, we denote by $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$ the kernel tensor and $\mathcal{K} \in \mathbb{R}^{MN\times CSN}$ the matrix that applies the convolutional layer of architecture (M, C, k, S)

to C vectorized channels of size SN. Note that, in the 1D case, we need to compare M with CS instead of CS^2.
RO case: When $M \leq CS$, $\mathcal{K}$ is orthogonal if and only if $\mathcal{K}\mathcal{K}^T = Id_{MN}$.
CO case: When $M \geq CS$, $\mathcal{K}$ is orthogonal if and only if $\mathcal{K}^T\mathcal{K} = Id_{CSN}$.

4.A.2.2 The function L_{orth}

We define L_{orth} similarly to the 2D case (see Section 4.1.1.2 and Figure 4.1). Formally, for $P \in \mathbb{N}$, and $h, g \in \mathbb{R}^k$, we define

$$\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = 1) \in \mathbb{R}^{2P+1} \tag{4.A.5}$$

such that for all $i \in [\![0, 2P]\!]$,

$$[\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = 1)]_i = \sum_{i'=0}^{k-1} h_{i'} \bar{g}_{i'+i} \ , \tag{4.A.6}$$

where $\bar{g}$ is defined for $i \in [\![0, 2P + k - 1]\!]$ as follows

$$\bar{g}_i = \begin{cases} g_{i-P} & \text{if } i \in [\![P, P + k - 1]\!], \\ 0 & \text{otherwise.} \end{cases} \tag{4.A.7}$$

Note that, for $P' \leq P$, we have, for all $i \in [\![0, 2P']\!]$,

$$\begin{aligned}[\mathrm{conv}(h, g, \text{padding zero} = P', \text{stride} = 1)]_i \\ = [\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = 1)]_{i+P-P'}.\end{aligned} \tag{4.A.8}$$

The strided version will be denoted by $\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S) \in \mathbb{R}^{\lfloor 2P/S \rfloor + 1}$ and is defined as follows: For all $i \in [\![0, \lfloor 2P/S \rfloor]\!]$

$$[\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S)]_i = [\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = 1)]_{Si}. \tag{4.A.9}$$

Finally, reminding that for all $m \in [\![1, M]\!]$ and $c \in [\![1, C]\!]$, $\mathbf{K}_{m,c} \in \mathbb{R}^k$, we denote by

$$\mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) \in \mathbb{R}^{M \times M \times (\lfloor 2P/S \rfloor + 1)}$$

the third-order tensor such that, for all $m, l \in [\![1, M]\!]$,

$$\begin{aligned}\mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S)_{m,l,:} \\ = \sum_{c=1}^{C} \mathrm{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S).\end{aligned} \tag{4.A.10}$$

From now on, we take $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$ and $\mathbf{I}_{r0} \in \mathbb{R}^{M \times M \times (2P/S+1)}$ the tensor whose entries are all zero except its central $M \times M$ entry which is equal to an identity matrix: $[\mathbf{I}_{r0}]_{:,:,P/S} =$

Id_M. Put differently, we have for all $m, l \in [\![1, M]\!]$,

$$[\mathbf{I}_{r0}]_{m,l,:} = \delta_{m=l} \begin{bmatrix} 0_{P/S} \\ 1 \\ 0_{P/S} \end{bmatrix} . \tag{4.A.11}$$

And L_{orth} for 1D convolutions is defined as follows:

— In the RO case:

$$L_{orth}(\mathbf{K}) = \| \mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) - \mathbf{I}_{r0} \|_F^2 .$$

— In the CO case:

$$L_{orth}(\mathbf{K}) = \| \mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) - \mathbf{I}_{r0} \|_F^2 - (M - CS) .$$

4.A.2.3 Measures of deviation from orthogonality

The orthogonality errors are defined by

$$\text{err}_N^F(\mathbf{K}) = \begin{cases} \| \mathcal{K}\mathcal{K}^T - \text{Id}_{MN} \|_F & \text{, in the RO case,} \\ \| \mathcal{K}^T\mathcal{K} - \text{Id}_{CSN} \|_F & \text{, in the CO case,} \end{cases}$$

and

$$\text{err}_N^s(\mathbf{K}) = \begin{cases} \| \mathcal{K}\mathcal{K}^T - \text{Id}_{MN} \|_2 & \text{, in the RO case,} \\ \| \mathcal{K}^T\mathcal{K} - \text{Id}_{CSN} \|_2 & \text{, in the CO case.} \end{cases}$$

4.B The convolutional layer as a matrix-vector product

In this section, we write the convolutional layer as a matrix-vector product. In other words, we explicit $\mathcal{K}$ and the ingredients composing it. The notation and preliminary results are useful in the proofs. Note that the results are already known and can be found for example in [121].

4.B.1 1D case

We denote by $S_N \in \mathbb{R}^{N \times SN}$ the sampling matrix (i.e., for $x = (x_0, \dots, x_{SN-1})^T \in \mathbb{R}^{SN}$, we have for all $m \in [\![0, N-1]\!]$, $(S_N x)_m = x_{Sm}$).
Put differently, we have

$$S_N = \sum_{i=0}^{N-1} E_{i,Si} . \tag{4.B.1}$$

Also, note that, using (4.A.3) and (4.A.4), we have $S_N S_N^T = Id_N$ and

$$S_N^T S_N = \sum_{i=0}^{N-1} F_{Si,Si} . \tag{4.B.2}$$

For a vector $x = (x_0, \dots, x_{n-1})^T \in \mathbb{R}^n$, we denote by $C(x) \in \mathbb{R}^{n \times n}$ the circulant matrix defined by

$$C(x) = \begin{pmatrix} x_0 & x_{n-1} & \cdots & x_2 & x_1 \\ x_1 & x_0 & x_{n-1} & & x_2 \\ \vdots & x_1 & x_0 & \ddots & \vdots \\ x_{n-2} & & \ddots & \ddots & x_{n-1} \\ x_{n-1} & x_{n-2} & \cdots & x_1 & x_0 \end{pmatrix} . \tag{4.B.3}$$

In other words, for $x \in \mathbb{R}^n$ and $X \in \mathbb{R}^{n \times n}$, we have

$$X = C(x) \iff \forall m, l \in [\![0, n-1]\!],\ X_{m,l} = x_{(m-l)\%n} . \tag{4.B.4}$$

The notation for the circulant matrix $C(.)$ should not be confused with the number of the input channels C. We also denote by $\tilde{x} \in \mathbb{R}^n$ the vector such that for all $i \in [\![0, n-1]\!]$, $\tilde{x}_i = x_{(-i)\%n}$. Again, the notation $\tilde{x}$, for $x \in \mathbb{R}^n$, should not be confused with $\tilde{a}$, for $a \in \mathbb{Z}$. We have

$$C(x)^T = C(\tilde{x}) . \tag{4.B.5}$$

Also, for $x, y \in \mathbb{R}^n$, we have

$$C(x)C(y) = C(x * y), \tag{4.B.6}$$

where $x * y \in \mathbb{R}^n$, is such that for all $j \in [\![0, n-1]\!]$,

$$[x * y]_j = \sum_{i=0}^{n-1} x_i y_{(j-i)\%n}. \tag{4.B.7}$$

$x * y$ is extended by n-periodicity. Note that here $x * y$ denotes the classical convolution as defined in math (i.e. by flipping the second argument). Note also that $x * y = y * x$ and therefore

$$C(x)C(y) = C(y)C(x) . \tag{4.B.8}$$

Throughout the chapter, the size of a filter is smaller than the size of the signal ($k = 2r + 1 \leq SN$). For $n \geq k$, we introduce an embedding P_n which associates to each $h = (h_0, \dots, h_{2r})^T \in \mathbb{R}^k$ the corresponding vector

$$P_n(h) = (h_r, \dots, h_1, h_0, 0, \dots, 0, h_{2r}, \dots, h_{r+1})^T \in \mathbb{R}^n .$$

Setting $[P_n(h)]_i = [P_n(h)]_{i\%n}$ for all $i \in \mathbb{Z}$, we have the following formula for P_n: for $i \in [\![-r, -r+n-1]\!]$,

$$[P_n(h)]_i = \begin{cases} h_{r-i} & \text{if } i \in [\![-r, r]\!] \\ 0 & \text{otherwise.} \end{cases} \tag{4.B.9}$$

Single-channel case: Let $x = (x_0, \dots, x_{SN-1})^T \in \mathbb{R}^{SN}$ be a 1D signal. We denote by Circular_Conv$(h, x, \text{stride} = 1)$ the result of the circular convolution [10] of x with the kernel $h = (h_0, \dots, h_{2r})^T \in \mathbb{R}^k$. We have

$$\text{Circular\_Conv}(h, x, \text{stride} = 1) = \left(\sum_{i'=0}^{k-1} h_{i'} x_{(i'+i-r)\%SN} \right)_{i=0..SN-1} .$$

Written as a matrix-vector product, this becomes

$$\begin{pmatrix} h_0 & \cdots & h_{2r} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & h_0 & \cdots & h_{2r} \end{pmatrix} \begin{pmatrix} x_{SN-r} \\ \vdots \\ x_{SN-1} \\ x_0 \\ \vdots \\ x_{SN-1} \\ x_0 \\ \vdots \\ x_{r-1} \end{pmatrix} \in \mathbb{R}^{SN}$$

$$= \begin{pmatrix} h_r & h_{r+1} & \cdots & h_{2r} & 0 & \cdots & 0 & h_0 & \cdots & h_{r-1} \\ h_{r-1} & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & h_0 \\ h_0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & h_{2r} \\ h_{2r} & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & h_{r+1} \\ h_{r+1} & \cdots & h_{2r} & 0 & \cdots & 0 & h_0 & \cdots & h_{r-1} & h_r \end{pmatrix} x$$

$$= C(P_{SN}(h))x .$$

The strided convolution is

$$\text{Circular\_Conv}(h, x, \text{stride} = S) = S_N C(P_{SN}(h))x \in \mathbb{R}^N . \tag{4.B.10}$$

Notice that $S_N C(P_{SN}(h)) \in \mathbb{R}^{N \times SN}$.

Multi-channel convolution: Let $X \in \mathbb{R}^{C \times SN}$ be a multi-channel 1D signal. We denote by Circular_Conv$(\mathbf{K}, X, \text{stride} = S)$ the result of the strided circular convolutional layer of kernel $\mathbf{K} \in \mathbb{R}^{M \times C \times k}$ applied to X. Using (4.B.10) for all the input-output channel

10. as defined in machine learning (we do not flip h).

correspondences, we have $Y = \text{Circular\_Conv}(\mathbf{K}, X, \text{stride} = S) \in \mathbb{R}^{M\times N}$ if and only if

$$\text{Vect}(Y) = \begin{pmatrix} S_N C(P_{SN}(\mathbf{K}_{1,1})) & \dots & S_N C(P_{SN}(\mathbf{K}_{1,C})) \\ \vdots & \vdots & \vdots \\ S_N C(P_{SN}(\mathbf{K}_{M,1})) & \dots & S_N C(P_{SN}(\mathbf{K}_{M,C})) \end{pmatrix} \text{Vect}(X) \ ,$$

where $\mathbf{K}_{i,j} = \mathbf{K}_{i,j,:} \in \mathbb{R}^k$. Therefore,

$$\mathcal{K} = \begin{pmatrix} S_N C(P_{SN}(\mathbf{K}_{1,1})) & \dots & S_N C(P_{SN}(\mathbf{K}_{1,C})) \\ \vdots & \vdots & \vdots \\ S_N C(P_{SN}(\mathbf{K}_{M,1})) & \dots & S_N C(P_{SN}(\mathbf{K}_{M,C})) \end{pmatrix} \in \mathbb{R}^{MN\times CSN} \tag{4.B.11}$$

is the layer transform matrix associated to kernel $\mathbf{K}$.

4.B.2 2D case

Notice that, since they are very similar, the proofs and notation are detailed in the 1D case, but we only provide a sketch of the proof and the main equations in 2D. In order to distinguish between the 1D and 2D versions of $C(.)$, P_n and S_N, we use calligraphic symbols in the 2D case. We denote by $\mathcal{S}_N \in \mathbb{R}^{N^2\times S^2N^2}$ the sampling matrix in the 2D case (i.e., for a matrix $x \in \mathbb{R}^{SN\times SN}$, if we denote by $z \in \mathbb{R}^{N\times N}$, such that for all $i, j \in [\![0, N-1]\!]$, $z_{i,j} = x_{Si,Sj}$, then $\text{Vect}(z) = \mathcal{S}_N \text{Vect}(x)$).
For a matrix $x \in \mathbb{R}^{n\times n}$, we denote by $\mathcal{C}(x) \in \mathbb{R}^{n^2\times n^2}$ the doubly-block circulant matrix defined by

$$\mathcal{C}(x) = \begin{pmatrix} C(x_{0,:}) & C(x_{n-1,:}) & \cdots & C(x_{2,:}) & C(x_{1,:}) \\ C(x_{1,:}) & C(x_{0,:}) & C(x_{n-1,:}) & & C(x_{2,:}) \\ \vdots & C(x_{1,:}) & C(x_{0,:}) & \ddots & \vdots \\ C(x_{n-2,:}) & & \ddots & \ddots & C(x_{n-1,:}) \\ C(x_{n-1,:}) & C(x_{n-2,:}) & \cdots & C(x_{1,:}) & C(x_{0,:}) \end{pmatrix} .$$

For $n \geq k = 2r+1$, we introduce the operator $\mathcal{P}_n$ which associates to a matrix $h \in \mathbb{R}^{k\times k}$ the corresponding matrix

$$\mathcal{P}_n(h) = \begin{pmatrix} h_{r,r} & \cdots & h_{r,0} & 0 & \cdots & 0 & h_{r,2r} & \cdots & h_{r,r+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{0,r} & \cdots & h_{0,0} & 0 & \cdots & 0 & h_{0,2r} & \cdots & h_{0,r+1} \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ h_{2r,r} & \cdots & h_{2r,0} & 0 & \cdots & 0 & h_{2r,2r} & \cdots & h_{2r,r+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{r+1,r} & \cdots & h_{r+1,0} & 0 & \cdots & 0 & h_{r+1,2r} & \cdots & h_{r+1,r+1} \end{pmatrix} \in \mathbb{R}^{n\times n} .$$

Setting $[\mathcal{P}_n(h)]_{i,j} = [\mathcal{P}_n(h)]_{i\%n,j\%n}$ for all $i,j \in \mathbb{Z}$, we have the following formula for $\mathcal{P}_n$: for $(i,j) \in [\![-r, -r+n-1]\!]^2$,

$$[\mathcal{P}_n(h)]_{i,j} = \begin{cases} h_{r-i,r-j} & \text{if } (i,j) \in [\![-r,r]\!]^2 \\ 0 & \text{otherwise.} \end{cases}$$

Single-channel case: Let $x \in \mathbb{R}^{SN \times SN}$ be a 2D image. We denote by Circular_Conv$(h, x, \text{stride} = 1)$ the result of the circular convolution of x with the kernel $h \in \mathbb{R}^{k \times k}$. As in the 1D case, we have

$$y = \text{Circular\_Conv}(h, x, \text{stride} = 1) \iff \text{Vect}(y) = \mathcal{C}(\mathcal{P}_{SN}(h))\,\text{Vect}(x)$$

and the strided circular convolution

$$y = \text{Circular\_Conv}(h, x, \text{stride} = S) \iff \text{Vect}(y) = \mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(h))\,\text{Vect}(x)\ .$$

Notice that $\mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(h)) \in \mathbb{R}^{N^2 \times S^2N^2}$.

Multi-channel convolution : Let $X \in \mathbb{R}^{C \times SN \times SN}$ be a multi-channel 2D image. We denote by Circular_Conv$(\mathbf{K}, X, \text{stride} = S)$ the result of the strided circular convolutional layer of kernel $\mathbf{K} \in \mathbb{R}^{M \times C \times k \times k}$ applied to X. We have $Y = \text{Circular\_Conv}(\mathbf{K}, X, \text{stride} = S) \in \mathbb{R}^{M \times N \times N}$ if and only if

$$\text{Vect}(Y) = \begin{pmatrix} \mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(\mathbf{K}_{1,1})) & \dots & \mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(\mathbf{K}_{1,C})) \\ \vdots & \vdots & \vdots \\ \mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(\mathbf{K}_{M,1})) & \dots & \mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(\mathbf{K}_{M,C})) \end{pmatrix} \text{Vect}(X)\ ,$$

where $\mathbf{K}_{i,j} = \mathbf{K}_{i,j,:,:} \in \mathbb{R}^{k \times k}$. Therefore,

$$\mathcal{K} = \begin{pmatrix} \mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(\mathbf{K}_{1,1})) & \dots & \mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(\mathbf{K}_{1,C})) \\ \vdots & \vdots & \vdots \\ \mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(\mathbf{K}_{M,1})) & \dots & \mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(\mathbf{K}_{M,C})) \end{pmatrix} \in \mathbb{R}^{MN^2 \times CS^2N^2}$$

is the layer transform matrix associated to kernel $\mathbf{K}$.

4.C Proof of Theorem 6

As the proofs are very similar in the 1D and 2D cases, we give the full proof in the 1D case, in Section 4.C.1, and we only give a sketch of the proof in the 2D case, in Section 4.C.2.

We first prove the result in the RO case, then in the CO case. In each case, we prove separately the statement when an orthogonal convolutional layer exists and when no orthogonal convolutional layer exists. When the architecture places us in the former case, to prove an orthogonal convolutional layer exists, we exhibit an explicit kernel tensor $\mathbf{K}$ and do the calculations to prove that $\mathcal{K}$ is orthogonal. The calculations are based on Lemma

24, in the RO case, and Lemma 25, in the CO case. Lemma 23 synthesize the result of a calculation that is used to prove both Lemma 24 and Lemma 25. When on the contrary the architecture is such that there does not exist any orthogonal convolutional layer, we prove that for all $\mathbf{K}$ the architecture condition implies $\mathrm{rk}(\mathcal{K}\mathcal{K}^T) < \mathrm{rk}(Id_{MN})$, in the RO case, and $\mathrm{rk}(\mathcal{K}^T\mathcal{K}) < \mathrm{rk}(Id_{CSN})$, in the CO case. This proves that no orthogonal convolutional layer exists.

4.C.1 Proof of Theorem 6, for 1D convolutional layers

We start by stating and proving three intermediate lemmas. Recall that $k = 2r + 1$ and from (4.A.1), that $(e_i)_{i=0..k-1} = \mathcal{B}(\mathbb{R}^k)$ and $(E_{a,b})_{a=0..N-1,b=0..SN-1} = \mathcal{B}(\mathbb{R}^{N\times SN})$.

Lemma 23. Let $j \in [\![0, k-1]\!]$. We have

$$S_N C\left(P_{SN}\left(e_j\right)\right) = \sum_{i=0}^{N-1} E_{i,Si+j-r} \ .$$

Proof. Let $j \in [\![0, k-1]\!]$. Using (4.B.9), (4.A.1), (4.A.2) and (4.B.3), we have

$$C(P_{SN}(e_j)) = C(f_{r-j}) = \sum_{i=0}^{SN-1} F_{i,i-(r-j)} = \sum_{i=0}^{SN-1} F_{i,i+j-r} \ .$$

Using (4.B.1) and (4.A.3), we have

$$S_N C\left(P_{SN}\left(e_j\right)\right) = \left(\sum_{i=0}^{N-1} E_{i,Si}\right)\left(\sum_{i'=0}^{SN-1} F_{i',i'+j-r}\right) = \sum_{i=0}^{N-1} E_{i,Si+j-r} \ .$$

□

Lemma 24. Let $k_S = \min(k, S)$ and $j, l \in [\![0, k_S - 1]\!]$. We have

$$S_N C(P_{SN}(e_j)) C(P_{SN}(e_l))^T S_N^T = \delta_{j=l} Id_N \ .$$

Proof. Let $j, l \in [\![0, k_S - 1]\!]$. Since $k_S \leq k$, using Lemma 23 and (4.A.4),

$$\begin{aligned} S_N C\left(P_{SN}\left(e_j\right)\right) C\left(P_{SN}\left(e_l\right)\right)^T S_N^T &= \left(\sum_{i=0}^{N-1} E_{i,Si+j-r}\right)\left(\sum_{i'=0}^{N-1} E_{i',Si'+l-r}\right)^T \\ &= \left(\sum_{i=0}^{N-1} E_{i,Si+j-r}\right)\left(\sum_{i'=0}^{N-1} \overline{E}_{Si'+l-r,i'}\right) \ . \end{aligned} \quad (4.C.1)$$

We know from (4.A.3) that $E_{i,Si+j-r}\overline{E}_{Si'+l-r,i'} = \delta_{\widehat{Si+j-r}=\widehat{Si'+l-r}} G_{i,i'}$. But for $i, i' \in [\![0, N-1]\!]$ and $j, l \in [\![0, k_S - 1]\!]$, since $k_S \leq S$, we have

$$-r \leq Si + j - r \leq S(N-1) + k_S - 1 - r \leq SN - 1 - r.$$

Similarly, $Si' + l - r \in [\![-r, SN - 1 - r]\!]$. Therefore, $Si + j - r$ and $Si' + l - r$ lie in the same interval of size SN, hence

$$\widehat{Si + j - r} = \widehat{Si' + l - r} \iff Si + j - r = Si' + l - r \iff Si + j = Si' + l \ .$$

If $Si + j = Si' + l$, then

$$|S(i - i')| = |j - l| < k_S \leq S.$$

Since $|i - i'| \in \mathbb{N}$, the latter inequality implies $i = i'$ and, as a consequence, $j = l$. Finally,

$$\widehat{Si + j - r} = \widehat{Si' + l - r} \iff i = i' \text{ and } j = l \ .$$

Hence, using (4.A.3), the equality (4.C.1) becomes

$$S_N C(P_{SN}(e_j))C(P_{SN}(e_l))^T S_N^T = \delta_{j=l} \sum_{i=0}^{N-1} G_{i,i} = \delta_{j=l} Id_N \ .$$

□

Lemma 25. Let $S \leq k$. We have

$$\sum_{z=0}^{S-1} C(P_{SN}(e_z))^T S_N^T S_N C(P_{SN}(e_z)) = Id_{SN} \ .$$

Proof. Let $z \in [\![0, S-1]\!]$. Since $S \leq k$, we have $z \in [\![0, k-1]\!]$. Hence using Lemma 23, then (4.A.4) and (4.A.3), we have

$$\begin{aligned} C\left(P_{SN}\left(e_z\right)\right)^T S_N^T S_N C\left(P_{SN}\left(e_z\right)\right) &= \left(\sum_{i=0}^{N-1} E_{i,Si+z-r}\right)^T \left(\sum_{i'=0}^{N-1} E_{i',Si'+z-r}\right) \\ &= \left(\sum_{i=0}^{N-1} \overline{E}_{Si+z-r,i}\right) \left(\sum_{i'=0}^{N-1} E_{i',Si'+z-r}\right) \\ &= \sum_{i=0}^{N-1} F_{Si+z-r,Si+z-r} \ . \end{aligned}$$

Hence

$$\sum_{z=0}^{S-1} C(P_{SN}(e_z))^T S_N^T S_N C(P_{SN}(e_z)) = \sum_{z=0}^{S-1} \sum_{i=0}^{N-1} F_{Si+z-r,Si+z-r} \ .$$

But, for $z \in [\![0, S-1]\!]$ and $i \in [\![0, N-1]\!]$, $Si+z-r$ traverses $[\![-r, SN-1-r]\!]$. Therefore,

using (4.A.2)

$$\sum_{z=0}^{S-1} C(P_{SN}(e_z))^T S_N^T S_N C(P_{SN}(e_z)) = \sum_{i=-r}^{SN-1-r} F_{i,i} = \sum_{i=0}^{SN-1} F_{i,i} = Id_{SN} \ .$$

□

Proof of Theorem 6. Let N be a positive integer such that $SN \geq k$.
We start by proving the theorem in the RO case.
Suppose $CS \geq M$ **and** $M \leq Ck$**:**
Let us exhibit $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$ such that $\mathcal{K}\mathcal{K}^T = Id_{MN}$.
Let $k_S = \min(k, S)$. Since $M \leq CS$ and $M \leq Ck$, we have $1 \leq M \leq Ck_S$. Therefore, there exist a unique couple $(i_{max}, j_{max}) \in [\![0, k_S-1]\!]\times[\![1, C]\!]$ such that $M = i_{max}C+j_{max}$. We define the kernel tensor $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$ as follows: For all $(i, j) \in [\![0, k_S - 1]\!] \times [\![1, C]\!]$ such that $iC + j \leq M$, we set $\mathbf{K}_{iC+j,j} = e_i$, and $\mathbf{K}_{u,v} = 0$ for all the other indices. Put differently, if we write $\mathbf{K}$ as a 3rd order tensor (where the rows represent the first dimension, the columns the second one, and the $\mathbf{K}_{i,j} \in \mathbb{R}^k$ are in the third dimension) we have :

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{1,1} & \cdots & \mathbf{K}_{1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{C,1} & \cdots & \mathbf{K}_{C,C} \\ \mathbf{K}_{C+1,1} & \cdots & \mathbf{K}_{C+1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{2C,1} & \cdots & \mathbf{K}_{2C,C} \\ & \vdots & \\ \mathbf{K}_{i_{max}C+1,1} & \cdots & \mathbf{K}_{i_{max}C+1,C} \\ \vdots & \ddots & \vdots \end{bmatrix} = \begin{bmatrix} e_0 & & \\ 0 & \ddots & 0 \\ & & e_0 \\ e_1 & & \\ 0 & \ddots & 0 \\ & & e_1 \\ & \vdots & \\ e_{i_{max}} & & \\ 0 & \ddots & 0 \end{bmatrix} \in \mathbb{R}^{M\times C\times k} \ ,$$

where $e_{i_{max}}$ appears j_{max} times. Therefore, using (4.B.11), we have

$$\mathcal{K} = \begin{bmatrix} S_N C(P_{SN}(e_0)) & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_0)) \\ S_N C(P_{SN}(e_1)) & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_1)) \\ & \vdots & \\ S_N C(P_{SN}(e_{i_{max}})) & & \\ 0 & \ddots & 0 \end{bmatrix} \in \mathbb{R}^{MN\times CSN} \ ,$$

where $S_N C(P_{SN}(e_{i_{max}}))$ appears j_{max} times. We have $\mathcal{K} = D_{1:MN,:}$, where we set

$$D = \begin{bmatrix} S_N C(P_{SN}(e_0)) & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_0)) \\ S_N C(P_{SN}(e_1)) & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_1)) \\ & \vdots & \\ S_N C(P_{SN}(e_{k_S-1})) & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_{k_S-1})) \end{bmatrix} \in \mathbb{R}^{k_S CN \times CSN} .$$

But, for $j, l \in [\![0, k_S - 1]\!]$, the (j, l)-th block of size (CN, CN) of DD^T is :

$$\begin{bmatrix} S_N C(P_{SN}(e_j)) & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_j)) \end{bmatrix} \begin{bmatrix} C(P_{SN}(e_l))^T S_N^T & & \\ 0 & \ddots & 0 \\ & & C(P_{SN}(e_l))^T S_N^T \end{bmatrix},$$

which is equal to

$$\begin{bmatrix} S_N C(P_{SN}(e_j)) C(P_{SN}(e_l))^T S_N^T & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_j)) C(P_{SN}(e_l))^T S_N^T \end{bmatrix}.$$

Using Lemma 24, this is equal to $\delta_{j=l} Id_{CN}$. Hence, $DD^T = Id_{k_S CN}$, and therefore,

$$\mathcal{K}\mathcal{K}^T = D_{1:MN,:}(D_{1:MN,:})^T = (DD^T)_{1:MN,1:MN} = Id_{MN} .$$

This proves the first implication in the RO case, i.e., if $M \leq Ck$, then $\mathbb{K}_1^\perp \neq \emptyset$.

Suppose $CS \geq M$ and $M > Ck$:
We need to prove that for all $\mathbf{K} \in \mathbb{R}^{M \times C \times k}$, we have $\mathcal{K}\mathcal{K}^T \neq Id_{MN}$.
Since for all $(i, j) \in [\![1, M]\!] \times [\![1, C]\!]$, each of the N rows of $S_N C(P_{SN}(\mathbf{K}_{i,j}))$ has at most k non-zero elements, the number of non-zero columns of $S_N C(P_{SN}(\mathbf{K}_{i,j}))$ is less than or equal to kN. Also, for all $i, i' \in [\![1, M]\!]$, the columns of $S_N C(P_{SN}(\mathbf{K}_{i,j}))$ which can be non-zero are the same as those of $S_N C(P_{SN}(\mathbf{K}_{i',j}))$. Hence, we have for all j, the number of non-zero columns of $\begin{bmatrix} S_N C(P_{SN}(\mathbf{K}_{1,j})) \\ \vdots \\ S_N C(P_{SN}(\mathbf{K}_{M,j})) \end{bmatrix}$ is less than or equal to kN. Therefore, the number of non-zero columns of $\mathcal{K}$ is less than or equal to CkN. Hence, since $Ck < M$, we have $\mathrm{rk}(\mathcal{K}\mathcal{K}^T) \leq \mathrm{rk}(\mathcal{K}) \leq CkN < MN = \mathrm{rk}(Id_{MN})$. Therefore, for all $\mathbf{K} \in \mathbb{R}^{M \times C \times k}$, we have $\mathcal{K}\mathcal{K}^T \neq Id_{MN}$.

This proves that if $CS \geq M$ and $M > Ck$, then $\mathbb{K}_1^{\perp} = \emptyset$. This concludes the proof in the RO case.

Suppose $M \geq CS$ and $S \leq k$:
Let us exhibit $\mathbf{K} \in \mathbb{R}^{M \times C \times k}$ such that $\mathcal{K}^T\mathcal{K} = Id_{CSN}$.
For all $(i,j) \in [\![0, S-1]\!] \times [\![1, C]\!]$, we set $\mathbf{K}_{iC+j,j} = e_i$, and $\mathbf{K}_{u,v} = 0$ for all the other indices. Put differently, if we write $\mathbf{K}$ as a 3rd order tensor, we have

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{1,1} & \cdots & \mathbf{K}_{1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{C,1} & \cdots & \mathbf{K}_{C,C} \\ \mathbf{K}_{C+1,1} & \cdots & \mathbf{K}_{C+1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{2C,1} & \cdots & \mathbf{K}_{2C,C} \\ & \vdots & \\ \mathbf{K}_{(S-1)C+1,1} & \cdots & \mathbf{K}_{(S-1)C+1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{CS,1} & \cdots & \mathbf{K}_{CS,C} \\ \mathbf{K}_{CS+1,1} & \cdots & \mathbf{K}_{CS+1,C} \\ \vdots & \vdots & \vdots \\ \mathbf{K}_{M,1} & \cdots & \mathbf{K}_{M,C} \end{bmatrix} = \begin{bmatrix} e_0 & & \\ 0 & \ddots & 0 \\ & & e_0 \\ e_1 & & \\ 0 & \ddots & 0 \\ & & e_1 \\ & \vdots & \\ e_{S-1} & & \\ 0 & \ddots & 0 \\ & & e_{S-1} \\ & O & \\ & & \end{bmatrix} \in \mathbb{R}^{M \times C \times k},$$

where $O = 0_{(M-CS) \times C \times k}$ denotes the null tensor. Therefore, using (4.B.11), we have

$$\mathcal{K} = \begin{bmatrix} S_N C(P_{SN}(e_0)) & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_0)) \\ S_N C(P_{SN}(e_1)) & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_1)) \\ & \vdots & \\ S_N C(P_{SN}(e_{S-1})) & & \\ 0 & \ddots & 0 \\ & & S_N C(P_{SN}(e_{S-1})) \\ & \mathcal{O} & \end{bmatrix} \in \mathbb{R}^{MN \times CSN},$$

where $\mathcal{O} = 0_{(MN-CSN) \times CSN}$ denotes the null matrix. Hence, $\mathcal{K}^T\mathcal{K}$ equals

$$\begin{bmatrix} \sum_{z=0}^{S-1} C(P_{SN}(e_z))^T S_N^T S_N C(P_{SN}(e_z)) & & 0 \\ & \ddots & \\ 0 & & \sum_{z=0}^{S-1} C(P_{SN}(e_z))^T S_N^T S_N C(P_{SN}(e_z)) \end{bmatrix}.$$

Using Lemma 25, we obtain $\mathcal{K}^T\mathcal{K} = Id_{CSN}$.
This proves that in the CO case, if $S \leq k$, then $\mathbb{K}_1^\perp \neq \emptyset$.

Suppose $M \geq CS$ and $S > k$:
We need to prove that for all $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$, we have $\mathcal{K}^T\mathcal{K} \neq Id_{CSN}$.
Following the same reasoning as in the case $CS \geq M$ and $M > Ck$, we have that the number of non-zero columns of $\mathcal{K}$ is less than or equal to CkN. So, since $k < S$, we have $\mathrm{rk}(\mathcal{K}^T\mathcal{K}) \leq \mathrm{rk}(\mathcal{K}) \leq CkN < CSN = \mathrm{rk}(Id_{CSN})$. Therefore, for all $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$, we have $\mathcal{K}^T\mathcal{K} \neq Id_{CSN}$.
This proves that in the CO case, if $k < S$, then $\mathbb{K}_1^\perp = \emptyset$. This concludes the proof. □

4.C.2 Sketch of the proof of Theorem 6, for 2D convolutional layers

We first set $(e_{i,j})_{i=0..k-1,j=0..k-1} = \mathcal{B}(\mathbb{R}^{k\times k})$. As in the 1D case, we have the following two lemmas

Lemma 26. Let $k_S = \min(k, S)$ and $j, j', l, l' \in [\![0, k_S - 1]\!]$. We have

$$\mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(e_{j,j'}))\mathcal{C}(\mathcal{P}_{SN}(e_{l,l'}))^T\mathcal{S}_N^T = \delta_{j=l}\delta_{j'=l'}Id_{N^2} .$$

Lemma 27. Let $S \leq k$. We have

$$\sum_{z=0}^{S-1}\sum_{z'=0}^{S-1}\mathcal{C}(\mathcal{P}_{SN}(e_{z,z'}))^T\mathcal{S}_N^T\mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(e_{z,z'})) = Id_{S^2N^2} .$$

For $CS^2 \geq M$ and $M \leq Ck^2$:
We set $\overline{e}_{i+kj} = e_{i,j}$ for $i, j \in [\![0, k-1]\!]$.
Let $i_{max}, j_{max} \in [\![0, k_S^2 - 1]\!] \times [\![1, C]\!]$ such that $i_{max}C + j_{max} = M$. We set

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{1,1} & \cdots & \mathbf{K}_{1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{C,1} & \cdots & \mathbf{K}_{C,C} \\ \mathbf{K}_{C+1,1} & \cdots & \mathbf{K}_{C+1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{2C,1} & \cdots & \mathbf{K}_{2C,C} \\ & \vdots & \\ \mathbf{K}_{i_{max}C+1,1} & \cdots & \mathbf{K}_{i_{max}C+1,C} \\ \vdots & \ddots & \vdots \end{bmatrix} = \begin{bmatrix} \overline{e}_0 & & \\ 0 & \ddots & 0 \\ & & \overline{e}_0 \\ \overline{e}_1 & & \\ 0 & \ddots & 0 \\ & & \overline{e}_1 \\ & \vdots & \\ \overline{e}_{i_{max}} & & \\ 0 & \ddots & 0 \end{bmatrix} \in \mathbb{R}^{M\times C\times k\times k} ,$$

where $\overline{e}_{i_{max}}$ appears j_{max} times. Then we proceed as in the 1D case.

For $CS^2 \geq M$ and $M > Ck^2$:
Using the same argument as in 1D, we can conclude that the number of non-zero columns of $\mathcal{K}$ is less than or equal to Ck^2N^2. Hence, $\mathrm{rk}(\mathcal{K}) \leq Ck^2N^2 < MN^2$. Therefore, for all

$\mathbf{K} \in \mathbb{R}^{M\times C\times k\times k}$, we have $\mathcal{K}\mathcal{K}^T \neq Id_{MN^2}$.

For $M \geq CS^2$ and $S \leq k$:
Denoting by $O \in \mathbb{R}^{(M-CS^2)\times C\times k\times k}$ the null 4th order tensor of size $(M-CS^2)\times C\times k\times k$, we set

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{1,1} & \cdots & \mathbf{K}_{1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{C,1} & \cdots & \mathbf{K}_{C,C} \\ \mathbf{K}_{C+1,1} & \cdots & \mathbf{K}_{C+1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{2C,1} & \cdots & \mathbf{K}_{2C,C} \\ & \vdots & \\ \mathbf{K}_{C(S^2-1)+1,1} & \cdots & \mathbf{K}_{C(S^2-1)+1,C} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{CS^2,1} & \cdots & \mathbf{K}_{CS^2,C} \\ \mathbf{K}_{CS^2+1,1} & \cdots & \mathbf{K}_{CS^2+1,C} \\ \vdots & \vdots & \vdots \\ \mathbf{K}_{M,1} & \cdots & \mathbf{K}_{M,C} \end{bmatrix} = \begin{bmatrix} e_{0,0} & & \\ 0 & \ddots & 0 \\ & & e_{0,0} \\ e_{1,0} & & \\ 0 & \ddots & 0 \\ & & e_{1,0} \\ & \vdots & \\ e_{S-1,S-1} & & \\ 0 & \ddots & 0 \\ & & e_{S-1,S-1} \\ & & \\ & O & \\ & & \end{bmatrix} \in \mathbb{R}^{M\times C\times k\times k} .$$

Then we proceed as in the 1D case.

For $M \geq CS^2$ and $S > k$:
By the same reasoning as in the 1D case, we have that the number of non-zero columns of $\mathcal{K}$ is less than or equal to Ck^2N^2. So, since $k < S$, we have $\mathrm{rk}(\mathcal{K}) \leq Ck^2N^2 < CS^2N^2$. Therefore, for all $\mathbf{K} \in \mathbb{R}^{M\times C\times k\times k}$, we have $\mathcal{K}^T\mathcal{K} \neq Id_{CS^2N^2}$.

4.D Restrictions due to boundary conditions

4.D.1 Proof of Proposition 13

Proof. For a single-channel convolution of kernel $h \in \mathbb{R}^k$ with 'valid' padding, the matrix applying the transformation on a signal $x \in \mathbb{R}^N$ has the following form:

$$A_N(h) := \begin{pmatrix} h_0 & \cdots & h_{2r} & & 0 \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ 0 & & & h_0 & \cdots & h_{2r} \end{pmatrix} \in \mathbb{R}^{(N-k+1)\times N} .$$

Hence, for $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$, the layer transform matrix is:

$$\mathcal{K} = \begin{pmatrix} A_N(\mathbf{K}_{1,1}) & \dots & A_N(\mathbf{K}_{1,C}) \\ \vdots & \vdots & \vdots \\ A_N(\mathbf{K}_{M,1}) & \dots & A_N(\mathbf{K}_{M,C}) \end{pmatrix} \in \mathbb{R}^{M(N-k+1)\times CN} .$$

Let us focus on the columns corresponding to the first input channel. To simplify the notation, for $m \in [\![1, M]\!]$ we denote by $a^{(m)} := \mathbf{K}_{m,1} \in \mathbb{R}^k$. By contradiction, suppose that $\mathcal{K}^T\mathcal{K} = Id_{CN}$. In particular, for the first block matrix of size $M(N-k+1)\times N$ of $\mathcal{K}$ (i.e., corresponding to the first input channel), its first column, last column and column of index $2r$ are of norm 1. Since $N \geq 2k-1$, we have

$$\sum_{m=1}^{M} \left(a_0^{(m)}\right)^2 = 1, \qquad \sum_{m=1}^{M} \left(a_{2r}^{(m)}\right)^2 = 1 \qquad \text{and} \qquad \sum_{i=0}^{2r}\sum_{m=1}^{M} \left(a_i^{(m)}\right)^2 = 1 .$$

This is impossible. Therefore, for all $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$, we have $\mathcal{K}^T\mathcal{K} \neq Id_{CN}$.

□

4.D.2 Proof of Proposition 14

Proof. For a single-channel convolution of kernel $h \in \mathbb{R}^k$ with zero-padding 'same', the matrix applying the transformation on a signal $x \in \mathbb{R}^N$ has the following form:

$$A_N(h) := \begin{pmatrix} h_r & \cdots & h_{2r} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ h_0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & h_{2r} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_0 & \cdots & h_r \end{pmatrix} \in \mathbb{R}^{N\times N} .$$

Hence, for $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$, the matrix that applies the convolutional layer is :

$$\mathcal{K} = \begin{pmatrix} A_N(\mathbf{K}_{1,1}) & \dots & A_N(\mathbf{K}_{1,C}) \\ \vdots & \vdots & \vdots \\ A_N(\mathbf{K}_{M,1}) & \dots & A_N(\mathbf{K}_{M,C}) \end{pmatrix} \in \mathbb{R}^{MN\times CN} .$$

Suppose $M \leq C$ (RO case): If $\mathcal{K}$ is orthogonal, then $\mathcal{K}\mathcal{K}^T = Id_{MN}$. Let us fix $m \in [\![1, M]\!]$. Since $\mathcal{K}\mathcal{K}^T = Id_{MN}$, the first row, the last row and the row of index r of the m-th block matrix of size $N \times CN$ of $\mathcal{K}$ are of norm equal to 1, i.e.

$$\|\mathcal{K}_{(m-1)N,:}\|_2^2 = 1, \qquad \|\mathcal{K}_{mN-1,:}\|_2^2 = 1 \qquad \text{and} \qquad \|\mathcal{K}_{(m-1)N+r,:}\|_2^2 = 1 .$$

To simplify the notation, for $c \in [\![1, C]\!]$, we denote by $a^{(c)} := \mathbf{K}_{m,c} \in \mathbb{R}^k$. Since $N \geq k$, the previous equations are equivalent to

$$\sum_{i=r}^{2r}\sum_{c=1}^{C}\left(a_i^{(c)}\right)^2 = 1, \qquad \sum_{i=0}^{r}\sum_{c=1}^{C}\left(a_i^{(c)}\right)^2 = 1 \qquad \text{and} \qquad \sum_{i=0}^{2r}\sum_{c=1}^{C}\left(a_i^{(c)}\right)^2 = 1\ .$$

Substracting the first equality from the third one, and the second equality from the third one, we obtain

$$\sum_{i=0}^{r-1}\sum_{c=1}^{C}\left(a_i^{(c)}\right)^2 = 0, \qquad \sum_{i=r+1}^{2r}\sum_{c=1}^{C}\left(a_i^{(c)}\right)^2 = 0 \qquad \text{and} \qquad \sum_{i=0}^{2r}\sum_{c=1}^{C}\left(a_i^{(c)}\right)^2 = 1\ .$$

This implies that for all $c \in [\![1, C]\!]$, for all $i \in [\![0, 2r]\!] \setminus \{r\}$, $a_i^{(c)} = 0$.
As a conclusion, for any $m \in [\![1, M]\!]$, any $c \in [\![1, C]\!]$, and any $i \in [\![0, 2r]\!] \setminus \{r\}$,

$$\mathbf{K}_{m,c,i} = 0.$$

This proves the result in the RO case.
The proof of the CO case is similar, and we have the same conclusion. □

4.E Proof of Theorem 7

As in Section 4.C, we give the full proof in the 1D case and a sketch of proof in the 2D case.

In the RO case, the proof is based on calculations in which we carefully detail the structure of the matrix $\mathcal{K}\mathcal{K}^T - Id_{MN}$ and identify its constituent with those of $L_{orth}(\mathbf{K})$. The main lemma describing the structure of $\mathcal{K}\mathcal{K}^T - Id_{MN}$ is Lemma 30. It is deduced from Lemma 28 and Lemma 29 which focus on submatrices of $\mathcal{K}\mathcal{K}^T - Id_{MN}$.

The result in the CO case is obtained from the result in the RO case and a known relation between $\|\mathcal{K}\mathcal{K}^T - Id_{MN}\|_F^2$ and $\|\mathcal{K}^T\mathcal{K} - Id_{CSN}\|_F^2$, see for instance Lemma 1 in [142].

4.E.1 Proof of Theorem 7, in the 1D case

Before proving Theorem 7, we first present three intermediate lemmas.

Lemma 28. Let $x \in \mathbb{R}^{SN}$. We have

$$S_N C(x) S_N^T = C(S_N x)\ .$$

Proof. Let $x \in \mathbb{R}^{SN}$, $X = C(x)$ and $Y = S_N X S_N^T \in \mathbb{R}^{N\times N}$. The matrix Y is formed by sampling X, i.e., for all $m, n \in [\![0, N-1]\!]$,

$$Y_{m,n} = X_{Sm,Sn}.$$

Hence, using (4.B.4), $Y_{m,n} = x_{(Sm-Sn)\%SN} = x_{S((m-n)\%N)}$. Setting $y = S_N x$, we have $y_l = x_{Sl}$ for all $l \in [\![0, N-1]\!]$. Therefore, $Y_{m,n} = y_{(m-n)\%N}$, and using (4.B.4), we obtain

$Y = C(y)$. Hence, from the definitions of Y, X and y we conclude that

$$S_N C(x) S_N^T = C(S_N x) \ .$$

This completes the proof of the lemma. □

For N such that $SN \geq 2k-1$, and $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$, we introduce the operator $Q_{S,N}$ which associates to a vector $x = (x_0, \ldots, x_{2\frac{P}{S}})^T \in \mathbb{R}^{2\frac{P}{S}+1}$, the vector

$$Q_{S,N}(x) = (x_{\frac{P}{S}}, \ldots, x_{2\frac{P}{S}}, 0, \ldots, 0, x_0, x_1, \ldots, x_{\frac{P}{S}-1})^T \in \mathbb{R}^N. \tag{4.E.1}$$

Lemma 29. Let S, $k = 2r+1$ and N be positive integers such that $SN \geq 2k-1$. Let $h, g \in \mathbb{R}^k$ and $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$, we have

$$S_N C(P_{SN}(h)) C(P_{SN}(g))^T S_N^T = C(Q_{S,N}(\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S))) \ . \tag{4.E.2}$$

Proof. Let N be such that $SN \geq 2k-1$, and $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$. Let us first detail and analyse the left-hand side of (4.E.2). Recall that by definition $P_{SN}(h)$ is SN-periodic: $[P_{SN}(h)]_i = [P_{SN}(h)]_{i\%SN}$ for all $i \in \mathbb{Z}$. Using (4.B.5), (4.B.6), and (4.B.7), we have

$$\begin{aligned} C(P_{SN}(h)) C(P_{SN}(g))^T &= C(P_{SN}(h)) C(\widetilde{P_{SN}(g)}) \\ &= C\left(\left(\sum_{i=0}^{SN-1} [P_{SN}(h)]_i \left[\widetilde{P_{SN}(g)}\right]_{j-i} \right)_{j=0..SN-1} \right) \\ &= C\left(\left(\sum_{i=0}^{SN-1} [P_{SN}(h)]_i [P_{SN}(g)]_{i-j} \right)_{j=0..SN-1} \right) \ . \end{aligned}$$

Setting $b^{(SN)}[h,g] = \left(\sum_{i=0}^{SN-1} [P_{SN}(h)]_i [P_{SN}(g)]_{i-j} \right)_{j=0..SN-1}$, we have

$$C(P_{SN}(h)) C(P_{SN}(g))^T = C(b^{(SN)}[h,g]) \ . \tag{4.E.3}$$

To simplify the forthcoming notation, we temporarily denote by

$$b := b^{(SN)}[h,g]. \tag{4.E.4}$$

Notice that by definition, b is SN-periodic. Therefore, we can restrict its study to an interval of size SN. We consider $j \in [\![-2r, SN-2r-1]\!]$. From the definition of P_{SN} in (4.B.9), we have, for $i \in [\![-r, -r+SN-1]\!]$,

$$[P_{SN}(h)]_i = \begin{cases} h_{r-i} & \text{if} \quad i \in [\![-r, r]\!] \\ 0 & \text{if} \quad i \in [\![r+1, -r+SN-1]\!] \ . \end{cases} \tag{4.E.5}$$

Hence, since $P_{SN}(h)$ and $P_{SN}(g)$ are periodic, we have

$$\begin{aligned} b_j &= \sum_{i=0}^{SN-1} [P_{SN}(h)]_i [P_{SN}(g)]_{i-j} \\ &= \sum_{i=-r}^{SN-1-r} [P_{SN}(h)]_i [P_{SN}(g)]_{i-j} \\ &= \sum_{i=-r}^{r} [P_{SN}(h)]_i [P_{SN}(g)]_{i-j}. \end{aligned} \tag{4.E.6}$$

The set of indices $i \in [\![-r, r]\!]$ such that $[P_{SN}(h)]_i [P_{SN}(g)]_{i-j} \neq 0$ is included in $[\![-r, r]\!] \cap \{i | (i-j)\%SN \in [\![-r, r]\!]\%SN\}$.
Since $j \in [\![-2r, SN-2r-1]\!]$: We have $-r \leq i \leq r$ and $-2r \leq j \leq SN-2r-1$, then $-SN+r+1 \leq i-j \leq 3r$, but by hypothesis, $SN \geq 2k-1 = 4r+1$, hence $3r < SN-r$ and so $-SN+r < i-j < SN-r$. Therefore, for $i \in [\![-r, r]\!]$ and $j \in [\![-2r, SN-2r-1]\!]$

$$(i-j)\%SN \in ([\![-r, r]\!]\%SN) \iff i-j \in [\![-r, r]\!] \iff i \in [\![-r+j, r+j]\!].$$

As a conclusion, for $j \in [\![-2r, SN-2r-1]\!]$,

$$\Big\{ i \in [\![-r, r]\!] \quad | \quad [P_{SN}(h)]_i [P_{SN}(g)]_{i-j} \neq 0 \Big\} \subset [\![-r, r]\!] \cap [\![-r+j, r+j]\!]. \tag{4.E.7}$$

Let us now analyse the right-side of (4.E.2). We start by considering zero-padding $= k-1$ and stride $= 1$, and we will arrive to the formula with zero-padding $= P$ and stride $= S$ later. Using (4.A.5), we denote by

$$a = \mathrm{conv}(h, g, \text{padding zero} = k-1, \text{stride} = 1) \in \mathbb{R}^{2k-1}. \tag{4.E.8}$$

We have from (4.A.6), for $j \in [\![0, 2k-2]\!]$,

$$a_j = \sum_{i=0}^{k-1} h_i \bar{g}_{i+j} \ .$$

Using (4.A.7) and keeping the indices $i \in [\![0, k-1]\!]$ for which $\bar{g}_{i+j} \neq 0$, i.e. such that $i+j \in [\![k-1, 2k-2]\!]$, we obtain

$$\begin{cases} a_j = \sum_{i=k-1-j}^{k-1} h_i g_{i+j-(k-1)} & \text{if} \quad j \in [\![0, k-2]\!] \ , \\ a_j = \sum_{i=0}^{2k-2-j} h_i g_{i+j-(k-1)} & \text{if} \quad j \in [\![k-1, 2k-2]\!] \ . \end{cases} \tag{4.E.9}$$

In the following, we will connect b with a by distinguishing several cases depending on the value of j.
We distinguish $j \in [\![0, 2r]\!]$, $j \in [\![-2r, -1]\!]$ and $j \in [\![2r+1, -2r+SN-1]\!]$. Recall that $k = 2r+1$.

If $j \in [\![0, 2r]\!]$: then $[\![-r, r]\!] \cap [\![-r+j, r+j]\!] = [\![-r+j, r]\!]$. Using (4.E.7) and (4.E.5), the equality (4.E.6) becomes

$$b_j = \sum_{i=-r+j}^{r} [P_{SN}(h)]_i [P_{SN}(g)]_{i-j} = \sum_{i=-r+j}^{r} h_{r-i} g_{r-i+j} \ .$$

By changing the variable $l = r - i$, and using $k = 2r + 1$, we find

$$b_j = \sum_{l=0}^{2r-j} h_l g_{l+j} = \sum_{l=0}^{k-1-j} h_l g_{l+j} = \sum_{l=0}^{2k-2-(k-1+j)} h_l g_{l+(k-1+j)-(k-1)} \ .$$

When $j \in [\![0, 2r]\!] = [\![0, k-1]\!]$, we have $k - 1 + j \in [\![k-1, 2k-2]\!]$, therefore using (4.E.9), we obtain

$$b_j = a_{k-1+j} \ . \tag{4.E.10}$$

If $j \in [\![-2r, -1]\!]$: then $[\![-r, r]\!] \cap [\![-r+j, r+j]\!] = [\![-r, r+j]\!]$. Using (4.E.7) and (4.E.5), the equality (4.E.6) becomes

$$b_j = \sum_{i=-r}^{r+j} [P_{SN}(h)]_i [P_{SN}(g)]_{i-j} = \sum_{i=-r}^{r+j} h_{r-i} g_{r-i+j} \ .$$

By changing the variable $l = r - i$, and using $k = 2r + 1$, we find

$$b_j = \sum_{l=-j}^{2r} h_l g_{l+j} = \sum_{l=-j}^{k-1} h_l g_{l+j} = \sum_{l=k-1-(k-1+j)}^{k-1} h_l g_{l+(k-1+j)-(k-1)} \ .$$

When $j \in [\![-2r, -1]\!] = [\![-(k-1), -1]\!]$, we have $k - 1 + j \in [\![0, k-2]\!]$, and using (4.E.9), we obtain

$$b_j = a_{k-1+j} \ . \tag{4.E.11}$$

If $j \in [\![2r+1, SN-2r-1]\!]$: then $[\![-r, r]\!] \cap [\![-r+j, r+j]\!] = \emptyset$. The equality (4.E.6) becomes

$$b_j = 0. \tag{4.E.12}$$

Therefore, we summarize (4.E.10), (4.E.11) and (4.E.12): For all $j \in [\![-(k-1), -(k-1) + SN - 1]\!]$,

$$b_j = \begin{cases} a_{k-1+j} & \text{if } j \in [\![-(k-1), k-1]\!], \\ 0 & \text{if } j \in [\![k, SN-k]\!]. \end{cases} \tag{4.E.13}$$

Let us now introduce 'padding zero $= P$' and 'stride $= S$'. We will prove the equality between matrices in (4.E.2) using the equality between vectors in (4.E.13).

Recall that $P = \left\lfloor \frac{k-1}{S} \right\rfloor S \leq k-1$, and let $i \in [\![0, 2P]\!]$. Therefore $i - P \in [\![-P, P]\!] \subset [\![-(k-1), k-1]\!]$, hence using (4.E.8), (4.A.8) and (4.E.13), we have

$$[\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = 1)]_i = a_{k-1+i-P} = b_{i-P} \ .$$

Therefore, using (4.A.9) and $\lfloor 2P/S \rfloor + 1 = 2P/S + 1$

$$\begin{aligned}
&\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S) \\
&= \left(b_{-\left\lfloor \frac{k-1}{S} \right\rfloor S}, \ldots, b_{-2S}, b_{-S}, b_0, b_S, b_{2S}, \ldots, b_{\left\lfloor \frac{k-1}{S} \right\rfloor S} \right)^T \in \mathbb{R}^{2P/S+1} \ .
\end{aligned}$$

Using the definition of $Q_{S,N}$ in (4.E.1), we obtain

$$\begin{aligned}
&Q_{S,N}(\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S)) \\
&= \left(b_0, b_S, b_{2S}, \ldots, b_{\left\lfloor \frac{k-1}{S} \right\rfloor S}, 0, \ldots, 0, b_{-\left\lfloor \frac{k-1}{S} \right\rfloor S}, \ldots, b_{-2S}, b_{-S} \right)^T \in \mathbb{R}^N \ .
\end{aligned}$$

But, using (4.E.12), and since $\left\lfloor \frac{k-1}{S} \right\rfloor S$ is the largest multiple of S less than or equal to $k-1$ and b is SN-periodic, we have

$$\begin{aligned}
S_N b &= \left(b_0, b_S, b_{2S}, \ldots, b_{\left\lfloor \frac{k-1}{S} \right\rfloor S}, 0, \ldots, 0, b_{SN-\left\lfloor \frac{k-1}{S} \right\rfloor S}, \ldots, b_{SN-2S}, b_{SN-S} \right)^T \\
&= \left(b_0, b_S, b_{2S}, \ldots, b_{\left\lfloor \frac{k-1}{S} \right\rfloor S}, 0, \ldots, 0, b_{-\left\lfloor \frac{k-1}{S} \right\rfloor S}, \ldots, b_{-2S}, b_{-S} \right)^T \in \mathbb{R}^N \ .
\end{aligned}$$

Finally, we have

$$S_N b = Q_{S,N}(\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S)) \ .$$

Using (4.E.4), (4.E.3) and Lemma 28, we conclude that

$$\begin{aligned}
S_N C(P_{SN}(h)) C(P_{SN}(g))^T S_N^T &= S_N C(b^{(SN)}[h, g]) S_N^T \\
&= C(S_N b^{(SN)}[h, g]) \\
&= C(Q_{S,N}(\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S))) \ .
\end{aligned}$$

□

Lemma 30. Let M, C, S, $k = 2r+1$ be positive integers, and let $\mathbf{K} \in \mathbb{R}^{M \times C \times k}$. Let N be such that $SN \geq 2k-1$, and $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$. We denote by $z_{P/S} = \begin{bmatrix} 0_{P/S} \\ 1 \\ 0_{P/S} \end{bmatrix} \in \mathbb{R}^{2P/S+1}$.

We have

$$\mathcal{K}\mathcal{K}^T - Id_{MN} = \begin{pmatrix} C(Q_{S,N}(x_{1,1})) & \ldots & C(Q_{S,N}(x_{1,M})) \\ \vdots & \ddots & \vdots \\ C(Q_{S,N}(x_{M,1})) & \ldots & C\left(Q_{S,N}(x_{M,M})\right) \end{pmatrix} \ ,$$

where for all $m, l \in [\![1, M]\!]$,

$$x_{m,l} = \sum_{c=1}^{C} \text{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S) - \delta_{m=l} z_{P/S} \in \mathbb{R}^{2P/S+1} . \tag{4.E.14}$$

Proof. We have from (4.B.11),

$$\mathcal{K} = \begin{pmatrix} S_N C(P_{SN}(\mathbf{K}_{1,1})) & \dots & S_N C(P_{SN}(\mathbf{K}_{1,C})) \\ \vdots & \vdots & \vdots \\ S_N C(P_{SN}(\mathbf{K}_{M,1})) & \dots & S_N C(P_{SN}(\mathbf{K}_{M,C})) \end{pmatrix} \in \mathbb{R}^{MN \times CSN} .$$

Hence, we have that the block $(m, l) \in [\![1, M]\!]^2$ of size (N, N) of $\mathcal{K}\mathcal{K}^T$ is equal to :

$$\begin{pmatrix} S_N C(P_{SN}(\mathbf{K}_{m,1})) & \dots & S_N C(P_{SN}(\mathbf{K}_{m,C})) \end{pmatrix} \begin{pmatrix} C(P_{SN}(\mathbf{K}_{l,1}))^T S_N^T \\ \vdots \\ C(P_{SN}(\mathbf{K}_{l,C}))^T S_N^T \end{pmatrix}$$
$$= \sum_{c=1}^{C} S_N C(P_{SN}(\mathbf{K}_{m,c})) C(P_{SN}(\mathbf{K}_{l,c}))^T S_N^T .$$

We denote by $A_{m,l} \in \mathbb{R}^{N \times N}$ the block $(m, l) \in [\![1, M]\!]^2$ of size (N, N) of $\mathcal{K}\mathcal{K}^T - Id_{MN}$. We want to prove that $A_{m,l} = C(Q_{S,N}(x_{m,l}))$ where $x_{m,l}$ is defined in (4.E.14). Using (4.A.1), (4.B.3), and (4.E.1), we have $Id_N = C\left(\begin{bmatrix} 1 \\ 0_{N-1} \end{bmatrix}\right) = C(Q_{S,N}(z_{P/S}))$, and therefore,

$$A_{m,l} = \sum_{c=1}^{C} S_N C(P_{SN}(\mathbf{K}_{m,c})) C(P_{SN}(\mathbf{K}_{l,c}))^T S_N^T - \delta_{m=l} C(Q_{S,N}(z_{P/S})) .$$

Using Lemma 29, this becomes

$$A_{m,l} = \sum_{c=1}^{C} C(Q_{S,N}(\text{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S))) - \delta_{m=l} C(Q_{S,N}(z_{P/S})) .$$

By linearity of C and $Q_{S,N}$, we obtain

$$A_{m,l} = C\left(Q_{S,N}\left(\sum_{c=1}^{C} \text{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S) - \delta_{m=l} z_{P/S}\right)\right)$$
$$= C\left(Q_{S,N}(x_{m,l})\right) .$$

□

Proof of Theorem 7. Let $M, C, S, k = 2r + 1$ be positive integers, and let $\mathbf{K} \in \mathbb{R}^{M \times C \times k}$. Let N be such that $SN \geq 2k - 1$, and $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$. For all $m, l \in [\![1, M]\!]$, we denote

by $A_{m,l} \in \mathbb{R}^{N\times N}$ the block (m,l) of size (N,N) of $\mathcal{K}\mathcal{K}^T - Id_{MN}$. Using Lemma 30, we have

$$A_{m,l} = C\left(Q_{S,N}\left(\sum_{c=1}^{C} \text{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S) - \delta_{m=l} z_{P/S}\right)\right).$$

Hence, from (4.B.3) and (4.E.1), using the fact that for all $x \in \mathbb{R}^N$, $\|C(x)\|_F^2 = N\|x\|_2^2$, and for all $x \in \mathbb{R}^{2P/S+1}$, $\|Q_{S,N}(x)\|_2^2 = \|x\|_2^2$, we have

$$\begin{aligned}
&\|\mathcal{K}\mathcal{K}^T - Id_{MN}\|_F^2 \\
&= \sum_{m=1}^{M}\sum_{l=1}^{M} \|A_{m,l}\|_F^2 \\
&= \sum_{m=1}^{M}\sum_{l=1}^{M} \left\| C\left(Q_{S,N}\left(\sum_{c=1}^{C} \text{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S) - \delta_{m=l} z_{P/S}\right)\right)\right\|_F^2 \\
&= \sum_{m=1}^{M}\sum_{l=1}^{M} N\left\| Q_{S,N}\left(\sum_{c=1}^{C} \text{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S) - \delta_{m=l} z_{P/S}\right)\right\|_2^2 \\
&= N\sum_{m=1}^{M}\sum_{l=1}^{M} \left\| \sum_{c=1}^{C} \text{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S) - \delta_{m=l} z_{P/S}\right\|_2^2 .
\end{aligned}$$

Therefore, using (4.A.11) and (4.A.10), we obtain for any M, C, S, $k = 2r+1$ and $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$,

$$\|\mathcal{K}\mathcal{K}^T - Id_{MN}\|_F^2 = N\|\,\mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) - \mathbf{I}_{r0}\|_F^2 . \tag{4.E.15}$$

This concludes the proof in the RO case.
In order to prove the theorem in the CO case we use Lemma 1 in [142]. This lemma states that

$$\|\mathcal{K}^T\mathcal{K} - Id_{CSN}\|_F^2 = \|\mathcal{K}\mathcal{K}^T - Id_{MN}\|_F^2 + CSN - MN.$$

Therefore, using that (4.E.15) holds for all M, C and S, we have

$$\|\mathcal{K}^T\mathcal{K} - Id_{CSN}\|_F^2 = N\left(\|\,\mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) - \mathbf{I}_{r0}\|_F^2 - (M - CS)\right) \tag{4.E.16}$$

Hence, using the definitions of err_N^F and L_{orth} in Sections 4.A.2.2 and 4.A.2.3, (4.E.15) and (4.E.16) lead to

$$\left(\text{err}_N^F(\mathbf{K})\right)^2 = NL_{orth}(\mathbf{K}).$$

This concludes the proof of Theorem 7 in the 1D case. □

4.E.2 Sketch of the proof of Theorem 7, in the 2D case

We start by stating intermediate lemmas. First we introduce a slight abuse of notation, for a vector $x \in \mathbb{R}^{N^2}$, we denote by $\mathcal{C}(x) = \mathcal{C}(X)$, where $X \in \mathbb{R}^{N\times N}$ such that $\mathrm{Vect}(X) = x$. The main steps of the proof in the 2D case follow those in the 1D case and are given below.

Lemma 31. Let $X \in \mathbb{R}^{SN\times SN}$. We have

$$\mathcal{S}_N\mathcal{C}(X)\mathcal{S}_N^T = \mathcal{C}(\mathcal{S}_N\,\mathrm{Vect}(X)).$$

Let $\mathcal{Q}_{S,N}$ be the operator which associates to a matrix $x \in \mathbb{R}^{(2P/S+1)\times(2P/S+1)}$ the matrix

$$\begin{pmatrix}
x_{P/S,P/S} & \cdots & x_{P/S,2P/S} & 0 & \cdots & 0 & x_{P/S,0} & \cdots & x_{P/S,P/S-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_{2P/S,P/S} & \cdots & x_{2P/S,2P/S} & 0 & \cdots & 0 & x_{2P/S,0} & \cdots & x_{2P/S,P/S-1} \\
0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & 0 \\
x_{0,P/S} & \cdots & x_{0,2P/S} & 0 & \cdots & 0 & x_{0,0} & \cdots & x_{0,P/S-1} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x_{P/S-1,P/S} & \cdots & x_{P/S-1,2P/S} & 0 & \cdots & 0 & x_{P/S-1,0} & \cdots & x_{P/S-1,P/S-1}
\end{pmatrix} \in \mathbb{R}^{N\times N}.$$

Lemma 32. Let N be such that $SN \geq 2k-1$, $h, g \in \mathbb{R}^{k\times k}$ and $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$, we have

$$\mathcal{S}_N\mathcal{C}(\mathcal{P}_{SN}(h))\mathcal{C}(\mathcal{P}_{SN}(g))^T\mathcal{S}_N^T = \mathcal{C}(\mathcal{Q}_{S,N}(\mathrm{conv}(h, g, \text{padding zero} = P, \text{stride} = S))).$$

Lemma 33. Let M, C, S, $k = 2r+1$ be positive integers, and let $\mathbf{K} \in \mathbb{R}^{M\times C\times k\times k}$. Let N be such that $SN \geq 2k-1$, and $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$. We set $z_{P/S,P/S} \in \mathbb{R}^{(2P/S+1)\times(2P/S+1)}$ such that for all $i, j \in [\![0, 2P/S]\!]$, $[z_{P/S,P/S}]_{i,j} = \delta_{i=P/S}\delta_{j=P/S}$. We have

$$\mathcal{K}\mathcal{K}^T - Id_{MN^2} = \begin{pmatrix} \mathcal{C}(\mathcal{Q}_{S,N}(x_{1,1})) & \dots & \mathcal{C}(\mathcal{Q}_{S,N}(x_{1,M})) \\ \vdots & \ddots & \vdots \\ \mathcal{C}(\mathcal{Q}_{S,N}(x_{M,1})) & \dots & \mathcal{C}(\mathcal{Q}_{S,N}(x_{M,M})) \end{pmatrix},$$

where for all $m, l \in [\![1, M]\!]$,

$$x_{m,l} = \sum_{c=1}^{C} \mathrm{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S) - \delta_{m=l} z_{P/S,P/S}.$$

Then we proceed as in the 1D case.

4.F Proof of Theorem 8

As in Section 4.C and Section 4.E, we give the full proof in the 1D case and a sketch of proof in the 2D case.

The lower-bound is a consequence of Theorem 7.

The proof of the upper-bound is different in the RO case and the CO case. In the RO case, we first express the orthogonality residual using Lemma 30. Then we conclude with calculations based on matrix norm inequalities, properties of circulant matrices and the definition of $L_{orth}(\mathbf{K})$.

The CO case is more difficult since, to use in place of Lemma 30, we first need to establish Lemma 34. We then proceed with calculations to express that $\|\mathcal{K}^T\mathcal{K} - Id_{CSN}\|_2$ is upper-bounded by a quantity independent of N, as long as $N \geq 2k-1$. Then, after calculations, a key argument is to apply Theorem 7 to the matrix $\mathcal{K}$ obtained for the signal size $N' = 2k-1$.

4.F.1 Proof of Theorem 8, in the 1D case

The lower-bound of Theorem 8 is an immediate consequence of (4.1.8) and Theorem 7. We have indeed both in the RO and CO case:

$$(\mathrm{err}_N^s(\mathbf{K}))^2 \geq \frac{1}{\min(M, CS^2)N^2}(\mathrm{err}_N^F(\mathbf{K}))^2 = \frac{1}{\min(M, CS^2)}L_{orth}(\mathbf{K}).$$

We focus from now on on the upper-bound. Let M, C, S, $k = 2r+1$ be positive integers, and let $\mathbf{K} \in \mathbb{R}^{M\times C\times k}$. Let N be such that $SN \geq 2k-1$, and $P = \left\lfloor \frac{k-1}{S} \right\rfloor S$. We denote by $z_{P/S} = \begin{bmatrix} 0_{P/S} \\ 1 \\ 0_{P/S} \end{bmatrix} \in \mathbb{R}^{2P/S+1}$.

RO case ($M \leq CS$): From Lemma 30, we have

$$\mathcal{K}\mathcal{K}^T - Id_{MN} = \begin{pmatrix} C(Q_{S,N}(x_{1,1})) & \dots & C(Q_{S,N}(x_{1,M})) \\ \vdots & \ddots & \vdots \\ C(Q_{S,N}(x_{M,1})) & \dots & C(Q_{S,N}(x_{M,M})) \end{pmatrix}, \tag{4.F.1}$$

where for all $m, l \in [\![1, M]\!]$,

$$x_{m,l} = \sum_{c=1}^{C} \mathrm{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S) - \delta_{m=l} z_{P/S} \in \mathbb{R}^{2P/S+1}. \tag{4.F.2}$$

We set

$$B = \mathcal{K}\mathcal{K}^T - Id_{MN}.$$

Since B is symmetric and due to the well-known properties of matrix norms, we have

$\|B\|_1 = \|B\|_\infty$ and $\|B\|_2^2 \leq \|B\|_1 \|B\|_\infty$. Hence, using the definition of $\|B\|_1$, we have

$$\|B\|_2^2 \leq \|B\|_1 \|B\|_\infty = \|B\|_1^2 = \left(\max_{1 \leq l \leq MN} \sum_{m=1}^{MN} |B_{m,l}| \right)^2 .$$

Using (4.F.1), and (4.B.3), we obtain

$$\|B\|_2^2 \leq \max_{1 \leq l \leq M} \left(\sum_{m=1}^{M} \|Q_{S,N}(x_{m,l})\|_1 \right)^2 .$$

Given the definition of $Q_{S,N}$ in (4.E.1), we have for all $x \in \mathbb{R}^{2P/S+1}$, $\|Q_{S,N}(x)\|_1 = \|x\|_1$, therefore,

$$\|B\|_2^2 \leq \max_{1 \leq l \leq M} \left(\sum_{m=1}^{M} \|x_{m,l}\|_1 \right)^2 .$$

We set $l_0 \in \arg\max_{1 \leq l \leq M} \left(\sum_{m=1}^{M} \|x_{m,l}\|_1 \right)^2$. Using that for all $x \in \mathbb{R}^n$, $\|x\|_1 \leq \sqrt{n}\|x\|_2$, we have

$$\|B\|_2^2 \leq \left(\sum_{m=1}^{M} \|x_{m,l_0}\|_1 \right)^2 \leq (2P/S+1) \left(\sum_{m=1}^{M} \|x_{m,l_0}\|_2 \right)^2 .$$

Using Cauchy-Schwarz inequality, we obtain

$$\|B\|_2^2 \leq (2P/S+1)M \sum_{m=1}^{M} \|x_{m,l_0}\|_2^2 \leq (2P/S+1)M \sum_{m=1}^{M} \sum_{l=1}^{M} \|x_{m,l}\|_2^2 .$$

Using (4.F.2), then (4.A.11) and (4.A.10), we obtain

$$\begin{aligned}
&\|B\|_2^2 \\
&\leq (2P/S+1)M \sum_{m=1}^{M} \sum_{l=1}^{M} \left\| \sum_{c=1}^{C} \mathrm{conv}(\mathbf{K}_{m,c}, \mathbf{K}_{l,c}, \text{padding zero} = P, \text{stride} = S) - \delta_{m=l} z_{P/S} \right\|_2^2 \\
&= (2P/S+1)M \sum_{m=1}^{M} \sum_{l=1}^{M} \left\| [\mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) - \mathbf{I}_{r0}]_{m,l,:} \right\|_2^2 \\
&= (2P/S+1)M \left\| \mathbf{conv}(\mathbf{K}, \mathbf{K}, \text{padding zero} = P, \text{stride} = S) - \mathbf{I}_{r0} \right\|_F^2 \\
&= (2P/S+1)M L_{orth}(\mathbf{K}) .
\end{aligned}$$

This proves the inequality in the RO case.

CO case ($M \geq CS$): First, for $n \geq 2k-1$, let R_n be the operator that associates to

$x \in \mathbb{R}^{2k-1}$, the vector

$$R_n(x) = (x_{k-1}, \dots, x_{2k-2}, 0, \dots, 0, x_0, \dots, x_{k-2})^T \in \mathbb{R}^n \ . \tag{4.F.3}$$

Note that, when $S' = 1$, $N' = SN$, we have in (4.E.1), $P' = k-1$ and

$$Q_{1,SN} = R_{SN}. \tag{4.F.4}$$

Recall from (4.A.1) that $(f_i)_{i=0..SN-1}$ is the canonical basis of $\mathbb{R}^{SN}$. Let $\Lambda_j = C(f_j) \in \mathbb{R}^{SN \times SN}$ be the permutation matrix which shifts down (cyclically) any vector by $j \in [\![0, SN-1]\!]$: for all $x \in \mathbb{R}^{SN}$, for $i \in [\![0, SN-1]\!]$, $(\Lambda_j x)_i = x_{(i-j)\%SN}$. Note that, using (4.B.3), we have for all $x \in \mathbb{R}^{SN}$,

$$[C(x)]_{:,j} = \Lambda_j x. \tag{4.F.5}$$

Recall that $k = 2r+1$, and for all $h \in \mathbb{R}^k$,

$$P_{SN}(x) = (h_r, \dots, h_0, 0, \dots, 0, h_{2r}, \dots, h_{r+1})^T \in \mathbb{R}^{SN}.$$

For $j \in [\![0, SN-1]\!]$, for $x \in \mathbb{R}^k$, we denote by

$$P^{(j)}_{SN}(x) = \Lambda_j P_{SN}(x) \tag{4.F.6}$$

and for $x \in \mathbb{R}^{2k-1}$, we denote by

$$R^{(j)}_{SN}(x) = \Lambda_j R_{SN}(x). \tag{4.F.7}$$

By assumption $SN \geq 2k-1$, hence $R_{SN}(x)$ is well-defined and we have for all $j \in [\![0, SN-1]\!]$, for all $x \in \mathbb{R}^{2k-1}$,

$$\begin{cases} \|R^{(j)}_{SN}(x)\|_1 = \|x\|_1, \\ \|R^{(j)}_{SN}(x)\|_2 = \|x\|_2. \end{cases} \tag{4.F.8}$$

We first start by introducing the following Lemma.

Lemma 34. Let $h, g \in \mathbb{R}^k$. There exist S vectors $x_0, \dots, x_{S-1} \in \mathbb{R}^{2k-1}$ such that for all N satisfying $SN \geq 2k-1$, we have for all $j \in [\![0, SN-1]\!]$,

$$\left[C(P_{SN}(h))^T S_N^T S_N C(P_{SN}(g))\right]_{:,j} = R^{(j)}_{SN}(x_{j\%S}) \ .$$

Proof. Recall that from (4.B.1) and (4.B.2), we have $S_N = \sum_{i=0}^{N-1} E_{i,Si}$ and $A_N := S_N^T S_N = \sum_{i=0}^{N-1} F_{Si,Si}$. When applied to a vector $x \in \mathbb{R}^{SN}$, A_N keeps unchanged the components of x whose indices are multiples of S, while the other components of $A_N x$ are equal to zero. We know from (4.F.5) and (4.F.6) that, for $j \in [\![0, SN-1]\!]$, the j-th column of $C(P_{SN}(g))$ is equal to $P^{(j)}_{SN}(g)$. Therefore, when applying A_N, this becomes $A_N P^{(j)}_{SN}(g) = P^{(j)}_{SN}(g^j)$, where $g^j \in \mathbb{R}^k$ is formed from g by putting zeroes in the place of the elements that have been replaced by 0 when applying A_N. But since A_N

preserves the component whose index is a multiple of S, we have that the j-th column of $A_N C(P_{SN}(g))$ has the same elements as its $j\%S$-th column, shifted down by $(j - j\%S)$ indices. More precisely, $A_N P_{SN}^{(j)}(g) = \Lambda_{j-j\%S} A_N P_{SN}^{(j\%S)}(g)$, hence $P_{SN}^{(j)}\left(g^j\right) = \Lambda_{j-j\%S} P_{SN}^{(j\%S)}\left(g^{j\%S}\right) = P_{SN}^{(j)}\left(g^{j\%S}\right)$. This implies that $g^j = g^{j\%S}$. Note that, using (4.B.9), we can also derive the exact formula of g^j, in fact for all $i \in [\![0, 2r]\!]$,

$$\left[g^j\right]_i = \begin{cases} g_i & \text{if } (i - r - j)\%S = 0, \\ 0 & \text{otherwise.} \end{cases}$$

We again can see that $g^j = g^{j\%S}$. Therefore, using (4.F.5) and (4.F.6), we have

$$A_N[C(P_{SN}(g))]_{:,j} = A_N P_{SN}^{(j)}(g) = P_{SN}^{(j)}\left(g^j\right) = P_{SN}^{(j)}\left(g^{j\%S}\right) = \left[C\left(P_{SN}\left(g^{j\%S}\right)\right)\right]_{:,j}.$$

Therefore, we have, for all $j \in [\![0, SN - 1]\!]$,

$$[C(P_{SN}(h))^T A_N C(P_{SN}(g))]_{:,j} = \left[C(P_{SN}(h))^T C(P_{SN}(g^{j\%S}))\right]_{:,j}.$$

Using the fact that the transpose of a circulant matrix is a circulant matrix and that two circulant matrices commute with each other (see (4.B.5) and (4.B.8)), we conclude that the transpose of any circulant matrix commutes with any circulant matrix, therefore

$$[C(P_{SN}(h))^T A_N C(P_{SN}(g))]_{:,j} = \left[C(P_{SN}(g^{j\%S}))C(P_{SN}(h))^T\right]_{:,j}.$$

Using Lemma 29 with $S' = 1$ and $N' = SN$, and noting that,when $S' = 1$, the sampling matrix $S_{N'}$ is equal to the identity, we have

$$\begin{aligned} &C(P_{SN}(g^{j\%S}))C(P_{SN}(h))^T \\ &= Id_{N'} C(P_{N'}(g^{j\%S}))C(P_{N'}(h))^T Id_{N'}^T \\ &= C(Q_{S',N'}(\text{conv}(g^{j\%S}, h, \text{padding zero} = \left\lfloor \frac{k-1}{S'} \right\rfloor S', \text{stride} = S'))) \\ &= C(Q_{1,SN}(\text{conv}(g^{j\%S}, h, \text{padding zero} = k - 1, \text{stride} = 1))) \end{aligned}$$

To simplify, we denote by $x_{j\%S} = \text{conv}(g^{j\%S}, h, \text{padding zero} = k - 1, \text{stride} = 1) \in \mathbb{R}^{2k-1}$. Using (4.F.4), we obtain

$$C(P_{SN}(g^{j\%S}))C(P_{SN}(h))^T = C(Q_{1,SN}(x_{j\%S})) = C(R_{SN}(x_{j\%S})).$$

Using (4.F.5) and (4.F.7), we obtain

$$[C(P_{SN}(h))^T A_N C(P_{SN}(g))]_{:,j} = [C(R_{SN}(x_{j\%S}))]_{:,j} = \Lambda_j R_{SN}(x_{j\%S}) = R_{SN}^{(j)}(x_{j\%S}).$$

Therefore, we have for all $j \in [\![0, SN - 1]\!]$,

$$\left[C(P_{SN}(h))^T S_N^T S_N C(P_{SN}(g))\right]_{:,j} = R_{SN}^{(j)}(x_{j\%S}).$$

This concludes the proof of the lemma. □

Let us go back to the main proof.

Using (4.B.11), we have that the block $(c, c') \in [\![1, C]\!]^2$ of size (SN, SN) of $\mathcal{K}^T\mathcal{K}$ is equal to :

$$\begin{pmatrix} C(P_{SN}(\mathbf{K}_{1,c}))^T S_N^T & \dots & C(P_{SN}(\mathbf{K}_{M,c}))^T S_N^T \end{pmatrix} \begin{pmatrix} S_N C(P_{SN}(\mathbf{K}_{1,c'})) \\ \vdots \\ S_N C(P_{SN}(\mathbf{K}_{M,c'})) \end{pmatrix}$$

$$= \sum_{m=1}^{M} C(P_{SN}(\mathbf{K}_{m,c}))^T S_N^T S_N C(P_{SN}(\mathbf{K}_{m,c'})) \ . \tag{4.F.9}$$

For any $(m, c, c') \in [\![1, M]\!] \times [\![1, C]\!]^2$, we denote by $(x_{m,c,c',s})_{s=0..S-1}$ the S vectors of $\mathbb{R}^{2k-1}$ obtained when applying Lemma 34 with $h = \mathbf{K}_{m,c}$, and $g = \mathbf{K}_{m,c'}$. Hence, we have, for all $j \in [\![0, SN-1]\!]$,

$$[C(P_{SN}(\mathbf{K}_{m,c}))^T S_N^T S_N C(P_{SN}(\mathbf{K}_{m,c'}))]_{:,j} = R_{SN}^{(j)}(x_{m,c,c',j\%S}) \ . \tag{4.F.10}$$

Let $\overline{f}_{k-1} = \begin{bmatrix} 0_{k-1} \\ 1 \\ 0_{k-1} \end{bmatrix} \in \mathbb{R}^{2k-1}$. For all $s \in [\![0, S-1]\!]$, we denote by

$$x_{c,c',s} = \sum_{m=1}^{M} x_{m,c,c',s} - \delta_{c=c'} \overline{f}_{k-1} \in \mathbb{R}^{2k-1}. \tag{4.F.11}$$

Note that, from (4.A.1), (4.F.3), and (4.F.7), we have for all $j \in [\![0, SN-1]\!]$, $f_j = R_{SN}^{(j)}(\overline{f}_{k-1})$. Therefore, $Id_{SN} = (f_0, \dots, f_{SN-1}) = \left(R_{SN}^{(0)}(\overline{f}_{k-1}), \dots, R_{SN}^{(SN-1)}(\overline{f}_{k-1})\right)$. We set

$$B_N = \mathcal{K}^T\mathcal{K} - Id_{CSN} \ .$$

We denote by $A_{c,c'}^N \in \mathbb{R}^{SN \times SN}$ the block $(c, c') \in [\![1, C]\!]^2$ of size (SN, SN) of B_N. Using (4.F.9), (4.F.10), and (4.F.11), we have, for all $j \in [\![0, SN-1]\!]$,

$$\begin{aligned} \left[A_{c,c'}^N\right]_{:,j} &= \left[\sum_{m=1}^{M} C(P_{SN}(\mathbf{K}_{m,c}))^T S_N^T S_N C(P_{SN}(\mathbf{K}_{m,c'})) - \delta_{c=c'} Id_{SN}\right]_{:,j} \\ &= \sum_{m=1}^{M} R_{SN}^{(j)}(x_{m,c,c',j\%S}) - \delta_{c=c'} R_{SN}^{(j)}(\overline{f}_{k-1}) \\ &= R_{SN}^{(j)}(x_{c,c',j\%S}) \ . \end{aligned} \tag{4.F.12}$$

We then proceed in the same way as in the RO case. Since B_N is clearly symmetric, we

have

$$\|B_N\|_2^2 \leq \|B_N\|_1 \|B_N\|_\infty = \|B_N\|_1^2 = \left(\max_{1\leq j\leq CSN} \sum_{i=1}^{CSN} |(B_N)_{i,j}| \right)^2$$
$$= \max_{1\leq c'\leq C\ ,\ 0\leq j\leq SN-1} \left(\sum_{c=1}^{C} \| \left[A^N_{c,c'}\right]_{:,j} \|_1 \right)^2 .$$

Using (4.F.12) and (4.F.8), this becomes

$$\|B_N\|_2^2 \leq \max_{\substack{1\leq c'\leq C \\ 0\leq j\leq SN-1}} \left(\sum_{c=1}^{C} \| R^{(j)}_{SN}\left(x_{c,c',j\%S}\right) \|_1 \right)^2 = \max_{\substack{1\leq c'\leq C \\ 0\leq s\leq S-1}} \left(\sum_{c=1}^{C} \|x_{c,c',s}\|_1 \right)^2 .$$

We set $(c'_0, s_0) \in \arg\max_{\substack{1\leq c'\leq C \\ 0\leq s\leq S-1}} \left(\sum_{c=1}^{C} \|x_{c,c',s}\|_1 \right)^2$. Using that for all $x \in \mathbb{R}^n$, $\|x\|_1 \leq \sqrt{n}\|x\|_2$, we have

$$\|B_N\|_2^2 \leq \left(\sum_{c=1}^{C} \|x_{c,c'_0,s_0}\|_1 \right)^2 \leq (2k-1) \left(\sum_{c=1}^{C} \|x_{c,c'_0,s_0}\|_2 \right)^2 .$$

Using Cauchy-Schwarz inequality, we obtain

$$\|B_N\|_2^2 \leq (2k-1)C \sum_{c=1}^{C} \|x_{c,c'_0,s_0}\|_2^2 \leq (2k-1)C \sum_{c=1}^{C}\sum_{c'=1}^{C}\sum_{s=0}^{S-1} \|x_{c,c',s}\|_2^2 .$$

Using (4.F.8) in the particular case of $N' = 2k-1$, we obtain

$$\begin{aligned}
\|B_N\|_2^2 &\leq (2k-1)C \sum_{c=1}^{C}\sum_{c'=1}^{C}\sum_{s=0}^{S-1} \|R_{S(2k-1)}\left(x_{c,c',s}\right)\|_2^2 \\
&= C \sum_{c=1}^{C}\sum_{c'=1}^{C}\sum_{s=0}^{S-1} (2k-1) \|R_{S(2k-1)}\left(x_{c,c',s}\right)\|_2^2 \\
&= C \sum_{c=1}^{C}\sum_{c'=1}^{C}\sum_{j=0}^{S(2k-1)-1} \left\| R^{(j)}_{S(2k-1)}\left(x_{c,c',j\%S}\right) \right\|_2^2 .
\end{aligned}$$

Using (4.F.12) for $N' = 2k-1$, we obtain

$$\|B_N\|_2^2 \leq C \sum_{c=1}^{C}\sum_{c'=1}^{C}\sum_{j=0}^{S(2k-1)-1} \left\| \left[A^{2k-1}_{c,c'}\right]_{:,j} \right\|_2^2 = C\|B_{2k-1}\|_F^2 .$$

Using Theorem 7 for $N = 2k-1$, we have $\|B_{2k-1}\|_F^2 = (2k-1)L_{orth}(\mathbf{K})$ and we obtain

$$\|B_N\|_2^2 \leq (2k-1)CL_{orth}(\mathbf{K}) .$$

Therefore, we conclude that, in the CO case

$$\left(\mathrm{err}_N^s(\mathbf{K})\right)^2 \leq (2k-1)CL_{orth}(\mathbf{K}) \ .$$

This concludes the proof in the 1D case.

4.F.2 Sketch of the proof of Theorem 8, for 2D convolutional layers

In the RO case, we proceed as in the 1D case.
In the CO case, we first prove a lemma similar to Lemma 34, then we proceed as in the 1D case.

4.G Proof of Proposition 12

Below, we prove Proposition 12 for a general matrix $A \in \mathbb{R}^{a\times b}$ with $a \geq b$. In order to obtain the statement for a convolutional layer $\mathcal{K} \in \mathbb{R}^{MN\times CSN}$:
In the RO case ($M \leq CS$): we take $A = \mathcal{K}^T$, $a = CSN$, $b = MN$.
In the CO case ($M \geq CS$): we take $A = \mathcal{K}$, $a = MN$, $b = CSN$.

Let $A \in \mathbb{R}^{a\times b}$ such that $a \geq b$. We denote by $\varepsilon = \|A^TA - Id_b\|_2$. Let $x \in \mathbb{R}^b$, we have

$$\begin{aligned}\left|\|Ax\|^2 - \|x\|^2\right| = \left|x^TA^TAx - x^Tx\right| = \left|x^T(A^TA - Id_b)x\right| &\leq \|x^T\|\|A^TA - Id_b\|_2\|x\| \\ &\leq \varepsilon\|x\|^2 \ .\end{aligned}$$

Hence, for all $x \in \mathbb{R}^b$,

$$(1-\varepsilon)\|x\|^2 \leq \|Ax\|^2 \leq (1+\varepsilon)\|x\|^2 \ .$$

This also implies $\sigma_{max}(A)^2 \leq 1+\varepsilon$. But we know that $\sigma_{max}(A^T) = \sigma_{max}(A)$, hence $\sigma_{max}(A^T)^2 \leq 1+\varepsilon$ and therefore, for all $x \in \mathbb{R}^a$,

$$\|A^Tx\|^2 \leq (1+\varepsilon)\|x\|^2 \ .$$

Finally:

— In the RO case, for $\varepsilon = \mathrm{err}_N^s(\mathbf{K}) = \|\mathcal{K}\mathcal{K}^T - Id_{CSN}\|_2$, $\mathcal{K}$ is ε-AIP.

— In the CO case, for $\varepsilon = \mathrm{err}_N^s(\mathbf{K}) = \|\mathcal{K}^T\mathcal{K} - Id_{MN}\|_2$, $\mathcal{K}$ is ε-AIP.

4.H Experiment configurations

4.H.1 Cifar10 experiments

The network architecture used for Cifar10 dataset is described in Table 4.3 (1.1 million parameters). Conv2D layer will depend on the configuration: classical $Conv2D$ for unconstrained reference configuration, $CayleyConv$ for $Cayley$ configuration (we use

[138] implementation), and L_{orth} regularization for L_{orth} configuration (we use our implementation according to Definition 5). Weight initialization is done according to *Glorot uniform*.

Table 4.3 – Cifar10 Neural network architectures: Conv2D, GS2 is GroupSort2, InvDown is InvertibleDownsampling [138]

Layer	Parameters (M, C, k, k)	Output size (M, H, W)
Input		$32 \times 32 \times 3$
Conv2D, GS2	$(64, 3, 3, 3)$	$64 \times 32 \times 32$
Conv2D, GS2	$(66, 64, 3, 3)$	$66 \times 32 \times 32$
InvDown		$264 \times 16 \times 16$
Conv2D, GS2	$(64, 264, 3, 3)$	$64 \times 16 \times 16$
Conv2D, GS2	$(128, 64, 3, 3)$	$128 \times 16 \times 16$
Conv2D, GS2	$(130, 128, 3, 3)$	$130 \times 16 \times 16$
InvDown		$520 \times 8 \times 8$
Conv2D, GS2	$(128, 520, 3, 3)$	$128 \times 8 \times 8$
Conv2D, GS2	$(192, 128, 3, 3)$	$192 \times 8 \times 8$
Conv2D, GS2	$(194, 192, 3, 3)$	$194 \times 8 \times 8$
InvDown		$776 \times 4 \times 4$
Conv2D, GS2	$(192, 776, 3, 3)$	$192 \times 4 \times 4$
Flatten, Dense	$(10, 3072)$	10

Task loss is the classical cross-entropy (CE). As described in [17], 1-lipschitz property of orthogonal neural network may prevent learning with CE and require introducing a temperature, i.e. multiply the network predictions/logits by a factor τ. Experiments for $Cayley$ and L_{orth} are done with $\tau = 20$ (and $\tau = 1$ for classical $Conv2D$).

We use classical data augmentation: random translation ($\pm 10\%$), random rotation (± 15 degree), random horizontal flipping (0.5 probability), random contrast modification ($[0.8, 1.2]$). For affine transformation zero-padding is used when required. The initial learning rate is set to 0.03, and linearly decreased down to $3 \times 1.00 \times 10^{-4}$.

The E_{rob} and E_{lip} metrics are computed using the code provided by [138].

4.H.2 Imagenette experiments

The network architecture used for Imagenette dataset is described in Table 4.4 (1.2 million parameters). Conv2D layer will depend on the configuration: classical $Conv2D$ for unconstrained reference configuration, $CayleyConv$ for $Cayley$ configuration (we use [138] implementation), and L_{orth} regularization for L_{orth} configuration (we use our implementation according to Definition 5). Weight initialization is done according to *Glorot uniform*.

Task loss is the classical cross-entropy (CE). As described in [17], 1-lipschitz property of orthogonal neural network may prevent learning with CE and require introducing a temperature, i.e. multiply the network predictions/logits by a factor τ. Experiments for

Table 4.4 – Imagenette Neural network architectures: Conv2D, GS2 is GroupSort2, InvDown is InvertibleDownsampling [138]

Layer	Parameters (M, C, k, k)	Output size (M, H, W)
Input		$3 \times 160 \times 160$
Conv2D, GS2	$(32, 3, 3, 3)$	$32 \times 160 \times 160$
Conv2D, GS2	$(34, 32, 3, 3)$	$34 \times 160 \times 160$
InvDown		$136 \times 80 \times 80$
Conv2D, GS2	$(32, 136, 3, 3)$	$32 \times 80 \times 80$
Conv2D, GS2	$(64, 32, 3, 3)$	$64 \times 80 \times 80$
Conv2D, GS2	$(66, 64, 3, 3)$	$66 \times 80 \times 80$
InvDown		$264 \times 40 \times 40$
Conv2D, GS2	$(64, 264, 3, 3)$	$64 \times 40 \times 40$
Conv2D, GS2	$(96, 64, 3, 3)$	$96 \times 40 \times 40$
Conv2D, GS2	$(98, 96, 3, 3)$	$98 \times 40 \times 40$
InvDown		$392 \times 20 \times 20$
Conv2D, GS2	$(96, 392, 3, 3)$	$96 \times 20 \times 20$
Conv2D, GS2	$(128, 96, 3, 3)$	$128 \times 20 \times 20$
Conv2D, GS2	$(130, 128, 3, 3)$	$130 \times 20 \times 20$
InvDown		$520 \times 10 \times 10$
Conv2D, GS2	$(128, 520, 3, 3)$	$128 \times 10 \times 10$
Conv2D, GS2	$(160, 128, 3, 3)$	$160 \times 10 \times 10$
Conv2D, GS2	$(162, 160, 3, 3)$	$162 \times 10 \times 10$
InvDown		$648 \times 5 \times 5$
Conv2D, GS2	$(160, 648, 3, 3)$	$160 \times 5 \times 5$
Flatten, Dense	$(10, 4000)$	10

$Cayley$ and L_{orth} are done with $\tau = 20$ (and $\tau = 1$ for classical $Conv2D$).The initial learning rate is set to $5 \times 1.00 \times 10^{-4}$, and linearly decreased down to $5 \times 1.00 \times 10^{-6}$.

Input images are normalized per channel using the recommended mean and std ($[0.485, 0.456, 0.406]$, $[0.229, 0.224, 0.225]$). The only data augmentation used is random horizontal flipping (0.5 probability).

4.I Computing the singular values of $\mathcal{K}$

In this appendix, we describe methods for computing singular values of a 2D layer transform matrix, with or without stride. The codes are provided in the *DEEL.LIP* [11] library.

4.I.1 Computing the singular values of $\mathcal{K}$ when $S = 1$

For convolutional layers without stride, $S = 1$, we use the algorithm described in [121]. We describe the algorithm for 2D convolutional layers in Algorithm 1. The algorithm provides the full list of singular values.

11. https://github.com/deel-ai/deel-lip

Algorithm 1 Computing the list of singular values of $\mathcal{K}$, when $S = 1$, [121].

Input: kernel tensor: $\mathbf{K} \in \mathbb{R}^{M \times C \times k \times k}$, channel size: $N \geq k$
Output: list of the singular values of $\mathcal{K}$: σ

procedure COMPUTESINGULARVALUES($\mathbf{K}$,N)
    transforms = np.fft.fft2($\mathbf{K}$, (N,N), axes=[0, 1]) ▷ np stands for numpy
    sigma = np.linalg.svd(transforms, compute_uv=False)
    return sigma
end procedure

4.I.2 Computing the smallest and the largest singular value of $\mathcal{K}$ for any stride S

For convolutions with stride, $S > 1$, there is no known practical algorithm to compute the list of singular values σ. In this configuration, we use the well-known power iteration algorithm associated with a spectral shift to compute the smallest and the largest singular value ($\sigma_{min}, \sigma_{max}$) of $\mathcal{K}$. We give the principle of the algorithm in Algorithm 2. For clarity, we assume a function 'λ = power_iteration(M,u_{init})', that applies the power iteration algorithm to a symetric matrix M starting from a random vector u_{init}, and returns its largest eigenvalue λ. In practice, of course, we cannot construct M and the implementation must use the usual functions that apply $\mathcal{K}$ and $\mathcal{K}^T$. A detailed python implementation is provided in the *DEEL.LIP* [12] library.

12. https://github.com/deel-ai/deel-lip

Algorithm 2 Computing $(\sigma_{min}, \sigma_{max})$, for any $S \geq 1$.

Input: kernel tensor: $\mathbf{K} \in \mathbb{R}^{M \times C \times k \times k}$, channel size: $N \geq k$, stride parameter: $S \geq 1$
Output: the smallest and the largest singular value of $\mathcal{K}$: $(\sigma_{min}, \sigma_{max})$

procedure COMPUTEMINANDMAXSINGULARVALUES($\mathbf{K}$,N,S)

if $CS^2 \geq M$ **then** ▷ RO case
u = np.random.randn(M,N,N)
lambda_1 = power_iteration($\mathcal{K}\mathcal{K}^T$, u)
bigCste = 1.1* lambda_1
u = np.random.randn(M,N,N)
lambda_2 = power_iteration(bigCste. $\mathrm{Id}_{MN^2} - \mathcal{K}\mathcal{K}^T$, u)
else ▷ CO case
u = np.random.randn(C,SN,SN)
lambda_1 = power_iteration($\mathcal{K}^T\mathcal{K}$, u)
bigCste = 1.1* lambda_1
u = np.random.randn(C,SN,SN)
lambda_2 = power_iteration(bigCste. $\mathrm{Id}_{CS^2N^2} - \mathcal{K}^T\mathcal{K}$, u)

end if
sigma_max = np.sqrt(lambda_1)
sigma_min = np.sqrt(bigCste-lambda_2)
return (sigma_min,sigma_max)
end procedure

CHAPTER 5

A general approximation lower bound in L^p norm, with applications to feed-forward neural networks

Abstract

We study the fundamental limits to the expressive power of neural networks. Given two sets F, G of real-valued functions, we first prove a general lower bound on how well functions in F can be approximated in $L^p(\mu)$ norm by functions in G, for any $p \geq 1$ and any probability measure μ. The lower bound depends on the packing number of F, the range of F, and the fat-shattering dimension of G. We then instantiate this bound to the case where G corresponds to a piecewise-polynomial feed-forward neural network, and describe in details the application to two sets F: Hölder balls and multivariate monotonic functions. Beside matching (known or new) upper bounds up to log factors, our lower bounds shed some light on the similarities or differences between approximation in L^p norm or in sup norm, solving an open question by DeVore et al. [31]. Our proof strategy differs from the sup norm case and uses a key probability result of Mendelson [100].

This chapter is based on joint work with Armand Foucault, Sébastien Gerchinovitz, and François Malgouyres (to appear at NeurIPS 2022).

Contents

5.1 Introduction

Neural networks are known for their great expressive power: in classification, they can interpolate arbitrary labels [157], while in regression they have universal approximation properties [28, 62, 88, 76], with approximation rates that can outperform those of linear approximation methods [153, 31]. Though the approximation problem is often only one part of the underlying learning problem (where generalization and optimization properties are also at stake), understanding the fundamental limits to the approximation properties of neural networks is key, both conceptually and for practical issues such as designing the right network architecture for the right problem.

We refer the reader to Chapter 2, Section 2.5, for the motivation and related works for this chapter. Recall that we are interested in quantifying the worst-case approximation error of F by G defined by

$$\sup_{f\in F}\inf_{g_{\mathbf{w}}\in G}\|f-g_{\mathbf{w}}\|. \tag{5.1.1}$$

Main contributions and outline of the chapter. We prove lower bounds on the approximation error (5.1.1) in any $L^p(\mu)$ norm, for non-quantized networks of arbitrary depth, and general sets F. Our main contributions are the following.

In Section 5.2, we first prove a general lower bound for any two sets F, G of real-valued functions on a set $\mathcal{X}$ (Theorem 9). The lower bound depends on the packing number of F, the range of F, and the fat-shattering dimension of G. We then derive a versatile corollary when G corresponds to a piecewise-polynomial feed-forward neural network (Corollary 1), solving the question by DeVore et al. [31]. Importantly, our proof strategy still relies on VC dimension theory, but differs from the sup norm case in using a key probability result of

Mendelson [100], to relate approximation in $L^p(\mu)$ norm with the fat-shattering dimension of G.

In Sections 5.3–5.4 we apply this corollary to the approximation of two sets: Hölder balls and multivariate monotonic functions. Beside matching (known or new) upper bounds up to log factors, our lower bounds shed some light on the similarities or differences between approximation in L^p norm or in sup norm. In particular, with ReLU networks, Hölder balls are not easier to approximate in L^p norm than in sup norm. On the contrary, the approximation rate for multivariate monotonic functions depends on p. In Section 5.5, we outline several other examples of function sets F and G for which the general lower bound (Theorem 9) can also be easily applied. Finally, some proofs are postponed to the supplement, while some details on other existing lower bound proof strategies are provided in the supplement, in Appendix 5.C.

Additional bibliographical remarks There are many other related results that we did not mention to keep the focus on our specific approximation problem. For instance, depth separation results show that deep neural networks can approximate functions that cannot be as easily approximated by shallower networks (e.g., [135, 140]). Let us also mention the general results of [151], which characterize minimax rates of estimation based on metric entropy conditions. Understanding the precise connections between these statistical results and our general approximation lower bound is an interesting question for the future.

Definitions and notation. We provide below some definitions and notation that will be used throughout the chapter. We denote the set of positive integers $\{1, 2, \dots\}$ by $\mathbb{N}^*$ and let $\mathbb{N} := \mathbb{N}^* \cup \{0\}$. All sets considered in this chapter will be assumed to be nonempty. We will not explicitly mention σ-algebras; for instance, by "Let $\mathcal{X}$ be a measurable space" we mean that $\mathcal{X}$ is a set implicitly endowed with a σ-algebra.

Let $p \in [1, +\infty]$ and $\mathcal{X}$ be any measurable space endowed with a probability measure μ. For any measurable function $f : \mathcal{X} \to \mathbb{R}$, the $L^p(\mu)$ norm of f is defined by $\|f\|_{L^p(\mu)} = \left(\int_{\mathcal{X}} |f(x)|^p d\mu(x)\right)^{1/p}$ (possibly infinite) if $p < +\infty$, and $\|f\|_{L^\infty(\mu)} = \operatorname{ess\,sup}_{x\in\mathcal{X}} |f(x)|$. We will write λ for the Lebesgue measure on $[0, 1]^d$.

For any $\varepsilon > 0$, two functions f_1, f_2 are said to be *ε-distant* in $\|\cdot\|$ if $\|f_1 - f_2\| > \varepsilon$. Let F be a set of functions from $\mathcal{X}$ to $\mathbb{R}$. A set $\{f_1, \dots, f_N\} \subset F$ is said to be an ε-packing of F in $\|\cdot\|$ (or just an ε-packing for short) if for any $i \neq j \in \{1, \dots, N\}$, f_i and f_j are ε-distant in $\|\cdot\|$. The ε-packing number $M(\varepsilon, F, \|\cdot\|)$ is the largest cardinality of ε-packings (possibly infinite).

For $\gamma > 0$, we say that a set $S = \{x_1 \dots, x_N\} \subset \mathcal{X}$ is *γ-shattered* by F if there exists $r : S \to \mathbb{R}$ such that for any $E \subset S$, there exists $f \in F$ satisfying for all $i = 1, \dots, N$, $f(x_i) \geq r(x_i) + \gamma$ if $x_i \in E$, and $f(x_i) \leq r(x_i) - \gamma$ if $x_i \notin E$. The *γ-fat-shattering dimension of* F, denoted by $\operatorname{fat}_\gamma(F)$, is the largest number $N \geq 1$ for which there exists $S \subset \mathcal{X}$ of cardinality N that is γ-shattered by F (by convention, $\operatorname{fat}_\gamma(F) = 0$ if no such set S exists, while $\operatorname{fat}_\gamma(F) = +\infty$ if there exist sets S of unbounded cardinality N). Similarly, we say that S is *pseudo-shattered* by F if there exists $r : S \to \mathbb{R}$ such that for any $E \subset S$, there exists $f \in F$ satisfying for all $i = 1, \dots, N$, $f(x_i) \geq r(x_i)$ if $x_i \in E$, and $f(x_i) < r_i$

if $x_i \notin E$. The *pseudo-dimension* $\mathrm{Pdim}(F)$ is the largest number $N \geq 1$ for which there exists $S \subset \mathcal{X}$ of cardinality N that is pseudo-shattered by F (same conventions). [1]

A formal definition of feed-forward neural networks is recalled in Appendix 5.A. In short, in this chapter, a *feed-forward neural network architecture* $\mathcal{A}$ of depth $L \geq 1$ is a directed acyclic graph with $d \geq 1$ input neurons, $L-1$ hidden layers (if $L \geq 2$), and an output layer with only one neuron. Skip connections are allowed, i.e., there can be connections between non-consecutive layers. Given an activation function $\sigma : \mathbb{R} \to \mathbb{R}$, a feed-forward neural network architecture $\mathcal{A}$, and a vector $\mathbf{w} \in \mathbb{R}^W$ of weights assigned to all edges and non-input neurons (linear coefficients and biases), the network computes a function $g_{\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}$ defined by recursively computing affine transformations for each hidden or output neuron, and then applying the activation function σ for hidden neurons only (see Appendix 5.A for more details). Finally, we define $H_{\mathcal{A}} := \{g_{\mathbf{w}} : \mathbf{w} \in \mathbb{R}^W\}$ to be the set of all functions that can be represented by tuning all the weights assigned to the network.

A function $\sigma : \mathbb{R} \to \mathbb{R}$ is *piecewise-polynomial* on $K \geq 2$ pieces, with maximal degree $\nu \in \mathbb{N}$, if there exists a partition $I_1, \ldots, I_K$ of $\mathbb{R}$ into K nonempty intervals, such that σ restricted on each I_j is polynomial with degree at most ν (in particular, σ can be discontinuous).

5.2 A general approximation lower bound in $L^p(\mu)$ norm

In this section, we provide our two main results: a general lower bound on the $L^p(\mu)$ approximation error of F by G, i.e., $\sup_{f\in F} \inf_{g\in G} \|f-g\|_{L^p(\mu)}$, and a corollary when G corresponds to a feed-forward neural network with a piecewise-polynomial activation function. The weak assumptions on F make the last result applicable to a wide range of cases of interest, as shown in Sections 5.3–5.5.

5.2.1 Main results

Our generic lower bound reads as follows, and is proved in Section 5.2.2. We follow the conventions $0 \times \log^2(0) = 0$ and $P^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(P) = +\infty$ when $P = 1$.

Theorem 9. Let $1 \leq p < +\infty$ and $\mathcal{X}$ be a measurable space endowed with a probability measure μ. Let F, G be two sets of measurable functions from $\mathcal{X}$ to $\mathbb{R}$, such that all functions in F have the same range $[a,b]$ for some $a < b$, and such that $\mathrm{fat}_\gamma(G) < +\infty$ for all $\gamma > 0$. Then, there exists a constant $c > 0$ depending only on p such that

$$\sup_{f\in F} \inf_{g\in G} \|f-g\|_{L^p(\mu)} \geq \inf\left\{\varepsilon > 0 : \log M\big(3\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big) \leq c\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G) \log^2\left(\frac{2\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)}{\varepsilon/(b-a)}\right)\right\}. \tag{5.2.1}$$

1. By definition, note that $\gamma \mapsto \mathrm{fat}_\gamma(F)$ is non-increasing and that $\mathrm{fat}_\gamma(F) \leq \mathrm{Pdim}(F)$ for all $\gamma > 0$.

In particular, if $\log M\big(\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big) \geq c_0\varepsilon^{-\alpha}$ for some $c_0, \varepsilon_0, \alpha > 0$ and all $\varepsilon \leq \varepsilon_0$, and if $\mathrm{Pdim}(G) < +\infty$, then there exist constants $c_1, \varepsilon_1 > 0$ depending only on $b-a$, p, c_0, ε_0 and α such that

$$\sup_{f\in F}\inf_{g\in G}\|f-g\|_{L^p(\mu)} \geq \min\left\{\varepsilon_1,\ c_1\,\mathrm{Pdim}(G)^{-\frac{1}{\alpha}}\log^{-\frac{2}{\alpha}}\big(\mathrm{Pdim}(G)\big)\right\}. \tag{5.2.2}$$

The first lower bound (5.2.1) is generic but requires solving an inequation.[2] In (5.2.2) we solve this inequation when $\log M\big(\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big)$ grows at least polynomially in $1/\varepsilon$ (which is typical of nonparametric sets) and when G has finite pseudo-dimension $\mathrm{Pdim}(G)$. Though we will restrict our attention to such cases in all subsequent sections, we stress that the first bound should have broader applications. A first example is when $\mathrm{Pdim}(G) = +\infty$ but $\mathrm{fat}_\gamma(G) < +\infty$ for all $\gamma > 0$ (e.g., for RKHS [14]). The first bound should also be useful to prove (slightly) tighter lower bounds when $\log M\big(\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big)$ has a (slightly) different dependency on $1/\varepsilon$ (e.g., of the order of $\varepsilon^{-\alpha}\log^\beta(1/\varepsilon)$ as when F is the set of all multivariate cumulative distribution functions [19]).

In the rest of the chapter, we focus on the important special case when the approximation set G is the set $H_{\mathcal{A}}$ of all real-valued functions that can be represented by tuning the weights of a feed-forward neural network with fixed architecture $\mathcal{A}$ and a piecewise-polynomial activation function. By combining Theorem 9 with known bounds on the pseudo-dimension [12], we obtain the following corollary, which bounds the approximation error in terms of the number W of weights and the depth L (i.e., the number of hidden and output layers). The proof is postponed to Appendix 5.B.4.

Corollary 1. Let $1 \leq p < +\infty$, $d \geq 1$ and $\mathcal{X}$ be a measurable subset of $\mathbb{R}^d$ endowed with a probability measure μ. Let F be a set of measurable functions from $\mathcal{X}$ to $[a, b]$ (for some real numbers $a < b$), such that $\log M\big(\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big) \geq c_0\varepsilon^{-\alpha}$ for some $c_0, \varepsilon_0, \alpha > 0$ and all $\varepsilon \leq \varepsilon_0$.

Let $\sigma : \mathbb{R} \to \mathbb{R}$ be any piecewise-polynomial activation function of maximal degree $\nu \in \mathbb{N}$ on $K \geq 2$ pieces. Then, there exist $W_{\min} \in \mathbb{N}^*$ and $c_1, c_2, c_3 > 0$ such that, for any $W \geq W_{\min}$, any $L \geq 1$, and any fixed feed-forward neural network architecture $\mathcal{A}$ of depth L with W weights, the set $H_{\mathcal{A}}$ of all real-valued functions on $\mathcal{X}$ that can be represented by the network (cf. Section 5.1) satisfies

$$\sup_{f\in F}\inf_{g\in H_{\mathcal{A}}}\|f-g\|_{L^p(\mu)} \geq \begin{cases} c_1 W^{-\frac{2}{\alpha}}\log^{-\frac{2}{\alpha}}(W) & \text{if } \nu \geq 2\,, \\ c_2 (LW)^{-\frac{1}{\alpha}}\log^{-\frac{3}{\alpha}}(W) & \text{if } \nu = 1\,, \\ c_3 W^{-\frac{1}{\alpha}}\log^{-\frac{3}{\alpha}}(W) & \text{if } \nu = 0\,. \end{cases} \tag{5.2.3}$$

There are equivalent ways to write the above corollary. For example, given a target accuracy $\varepsilon > 0$ and a depth $L \geq 1$, (5.2.3) yields a lower bound on the minimum number W of weights that are needed to get $\sup_{f\in F}\inf_{g\in H_{\mathcal{A}}}\|f-g\|_{L^p(\mu)} \leq \varepsilon$. Some earlier

2. Note that any $\varepsilon \geq (b-a)/3$ is a solution to this inequation, since $\log M\big(3\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big) = \log(1) = 0$ (because all functions in F are $[a,b]$-valued) and $c\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G) \geq 0$. Therefore, the right-hand side of (5.2.1) is at most $(b-a)/3$.

approximation results were written this way (e.g., [152, 111]).

5.2.2 Proof of Theorem 9

In order to prove Theorem 9, we need two inequalities. The first one is straightforward (and appeared within proofs, e.g., in [154]), but formalizes the key idea that if G approximates F with error ε, then G has to be at least as large as F. We use the conventions $\log(+\infty) = +\infty$ and $+\infty \leq +\infty$.

Lemma 35. Let $p \geq 1$ and $\mathcal{X}$ be a measurable space endowed with a probability measure μ. Let F, G be two sets of measurable functions from $\mathcal{X}$ to $\mathbb{R}$. If $\sup_{f\in F} \inf_{g\in G} \|f-g\|_{L^p(\mu)} < \varepsilon$, then

$$\log M\big(3\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big) \leq \log M\big(\varepsilon, G, \|\cdot\|_{L^p(\mu)}\big) \ .$$

Proof. Let $P_F = \{f_1, \dots, f_N\}$ be a 3ε-packing of F, with $N \geq 1$. Let $P_G = \{g_1, \dots, g_N\}$ be a subset of G such that $\|f_i - g_i\|_{L^p(\mu)} \leq \varepsilon$ for all i. Note that the existence of such a P_G is guaranteed by the assumption $\sup_{f\in F} \inf_{g\in G} \|f - g\|_{L^p(\mu)} < \varepsilon$. Since the f_i's are pairwise 3ε-distant in $L^p(\mu)$, the triangle inequality entails that the g_i's are also at least pairwise ε-distant in $L^p(\mu)$. Therefore, P_G is an ε-packing of G, and the result follows. □

The next inequality is a fundamental probability result due to Mendelson [100]. It bounds from above the ε-packing number in $L^p(\mu)$ norm of any uniformly bounded function set in terms of its fat-shattering dimension. Crucially, the inequality holds for finite $p \geq 1$, as opposed to the lower bound strategy of Yarotsky [152, 153] (see also [31]), that relates the VC-dimension with the approximation error in sup norm. The next statement is a slight generalization of a result of [100] initially stated for $[a, b] = [0, 1]$ and for Glivenko-Cantelli classes G (see Appendix 5.B.1 for details).

Proposition 5.2.1 ([100], Corollary 3.12). Let G be a set of measurable functions from a measurable space $\mathcal{X}$ to $[a, b]$ (for some real numbers $a < b$), and such that $\mathrm{fat}_\gamma(G) < +\infty$ for all $\gamma > 0$. Then for any $1 \leq p < +\infty$, there exists $c > 0$ depending only on p such that for every probability measure μ on $\mathcal{X}$ and every $\varepsilon > 0$,

$$\log M\big(\varepsilon, G, \|\cdot\|_{L^p(\mu)}\big) \leq c\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G) \log^2\left(\frac{2(b-a)\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)}{\varepsilon}\right) . \tag{5.2.4}$$

Refinements of this inequality were proved in specific cases such as the $L^2(\mu)$ norm [101] (see also [50] for empirical $L^p(\mu_n)$ norms). However, using the result of [101] when $p = 2$ would only yield a minor logarithmic improvement in the lower bound of Theorem 9.

Proof (of Theorem 9). **Part 1.** We start by proving (5.2.1), using Proposition 5.2.1 as a key argument. Since functions in G are not necessarily uniformly bounded, we will apply Proposition 5.2.1 to the "clipped version of G". More precisely, for any function $g \in G$, we define its clipping (truncature) to $[a, b]$ as the function $\tilde{g} : \mathcal{X} \to \mathbb{R}$ given by $\tilde{g}(x) = \min(\max(a, g(x)), b)$ for all $x \in \mathcal{X}$. We then set $G_{[a,b]} = \{\tilde{g} : g \in G\}$, which by construction consists of functions that are all $[a, b]$-valued.

Noting that clipping can only help since elements of F are $[a,b]$-valued (see Lemma 38 in the supplement, Appendix 5.B.2), we have

$$\sup_{f\in F}\inf_{g\in G}\|f-g\|_{L^p(\mu)} \geq \sup_{f\in F}\inf_{\tilde{g}\in G_{[a,b]}}\|f-\tilde{g}\|_{L^p(\mu)}\,. \tag{5.2.5}$$

Setting $\Delta := \sup_{f\in F}\inf_{\tilde{g}\in G_{[a,b]}}\|f-\tilde{g}\|_{L^p(\mu)}$, we now show that Δ is bounded from below by the right-hand side of (5.2.1). To that end, it suffices to show that every $\varepsilon > \Delta$ is a solution to the inequation

$$\log M\big(3\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big) \leq c\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)\log^2\left(\frac{2(b-a)\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)}{\varepsilon}\right)\,. \tag{5.2.6}$$

The last inequality is true whenever $\varepsilon \geq (b-a)/3$ (see Footnote 2). We only need to prove (5.2.6) when $\Delta < \varepsilon < (b-a)/3$. In this case, by definition of Δ and by Lemma 35 applied to $G_{[a,b]}$, we have

$$\begin{aligned}\log M\big(3\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big) &\leq \log M\big(\varepsilon, G_{[a,b]}, \|\cdot\|_{L^p(\mu)}\big)\\ &\leq c\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G_{[a,b]})\log^2\left(\frac{2(b-a)\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G_{[a,b]})}{\varepsilon}\right)\\ &\leq c\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)\log^2\left(\frac{2(b-a)\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)}{\varepsilon}\right)\,,\end{aligned} \tag{5.2.7}$$

where the second inequality follows from Proposition 5.2.1 (note from Lemma 37 in the supplement, Appendix 5.B.2 that $\mathrm{fat}_\gamma(G_{[a,b]}) \leq \mathrm{fat}_\gamma(G)$ for all $\gamma > 0$, which is finite by assumption), and where (5.2.7) follows from the next remark. Either $\mathrm{fat}_{\frac{\varepsilon}{32}}(G_{[a,b]}) = 0$, and (5.2.7) is true by the convention $0\times\log^2(0) = 0$ and $c\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G) \geq 0$. Either $\mathrm{fat}_{\frac{\varepsilon}{32}}(G_{[a,b]}) \geq 1$, and (5.2.7) follows from $t \mapsto ct\log^2\big(\frac{2(b-a)t}{\varepsilon}\big)$ being non-decreasing on $[\varepsilon/(2(b-a)), +\infty)$ and $\varepsilon/(2(b-a)) \leq 1/6 \leq 1 \leq \mathrm{fat}_{\frac{\varepsilon}{32}}(G_{[a,b]}) \leq \mathrm{fat}_{\frac{\varepsilon}{32}}(G)$. To conclude, every $\varepsilon > \Delta$ satisfies (5.2.6), which implies that Δ is bounded from below by the right-hand side of (5.2.1). Combining with (5.2.5) concludes the proof of (5.2.1).

Part 2. Set $\varepsilon_1' = \min\big\{\frac{\varepsilon_0}{3}, 2(b-a)\big\}$. We now derive (5.2.2) from (5.2.1). To that end, setting $P = \mathrm{Pdim}(G)$, we show that every $\varepsilon > 0$ satisfying (5.2.6) is such that $\varepsilon \geq \min\big\{\varepsilon_1,\ c_1 P^{-\frac{1}{\alpha}}\log^{-\frac{2}{\alpha}}(P)\big\}$, where $\varepsilon_1 \in (0, \varepsilon_1']$ and $c_1 > 0$ will be defined later. Since the claimed lower bound on ε is true when $\varepsilon \geq \varepsilon_1'$, in the sequel we consider any solution ε to (5.2.6) such that $0 < \varepsilon < \varepsilon_1'$ (if such a solution exists).

By the assumption on $\log M\big(u, F, \|\cdot\|_{L^p(\mu)}\big)$ for $u = 3\varepsilon \leq \varepsilon_0$, and then using (5.2.6), we have, setting $r = 2(b-a)$,

$$c_0(3\varepsilon)^{-\alpha} \leq \log M\big(3\varepsilon, F, \|\cdot\|_{L^p(\mu)}\big) \leq c\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)\log^2\left(\frac{r\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)}{\varepsilon}\right) \leq cP\log^2\left(\frac{rP}{\varepsilon}\right)\,,$$

where the last inequality is because $t \mapsto ct\log^2\big(\frac{rt}{\varepsilon}\big)$ is non-decreasing on $[\varepsilon/r, +\infty)$, with $\varepsilon/r \leq 1$, and $1 \leq \mathrm{fat}_{\frac{\varepsilon}{32}}(G) \leq \mathrm{Pdim}(G) = P$ (the lower bound of 1 follows from

$c_0(3\varepsilon)^{-\alpha} > 0$).

Solving the inequation $c_0(3\varepsilon)^{-\alpha} \leq cP\log^2(rP/\varepsilon)$ for ε (see Appendix 5.B.3 for details), we get

$$\varepsilon \geq \min\left\{\varepsilon_1'',\ c_1 P^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}} P\right\}, \tag{5.2.8}$$

for some constants $\varepsilon_1'', c_1 > 0$ depending only on p, c_0, $b-a$ and α. Setting $\varepsilon_1 = \min\{\varepsilon_1'', \varepsilon_1'\}$ and noting that ε_1' only depends on ε_0 and $b-a$, we conclude the proof. □

5.3 Approximation of Hölder balls by feed-forward neural networks

In this section, we apply Corollary 1 to establish nearly-tight lower bounds for the approximation of unit Hölder balls by feed-forward neural networks. Our main result is Proposition 5.3.2, which solves an open question by [31].

Throughout the section, for any $s > 0$, we denote by n and α the unique members of the decomposition $s = n + \alpha$ such that $n \in \mathbb{N}$ and $0 < \alpha \leq 1$.

For a set $\mathcal{X} \subset \mathbb{R}^d$, we follow [154] and define the Hölder space $\mathcal{C}^{n,\alpha}(\mathcal{X})$ as the space of n times continuously differentiable functions with finite norm

$$\|f\|_{\mathcal{C}^{n,\alpha}} = \max\left\{\max_{\mathbf{n}:|\mathbf{n}|\leq n} \|D^{\mathbf{n}} f\|_\infty,\ \max_{\mathbf{n}:|\mathbf{n}|=n} \sup_{x\neq y} \frac{|D^{\mathbf{n}} f(x) - D^{\mathbf{n}} f(y)|}{\|x-y\|_2^\alpha}\right\},$$

where, for $\mathbf{n} = (n_1, \cdots, n_d) \in \mathbb{N}^d$, $D^{\mathbf{n}} f = \left(\frac{\partial}{\partial x_1}\right)^{n_1} \cdots \left(\frac{\partial}{\partial x_d}\right)^{n_d} f$ denotes the $|\mathbf{n}|$-order partial derivative of f. We denote

$$F_{s,d} = \{f \in \mathcal{C}^{n,\alpha}([0,1]^d) : \|f\|_{\mathcal{C}^{n,\alpha}} \leq 1\}.$$

Let λ denote the Lebesgue measure over $[0,1]^d$. In this section, we provide nearly matching upper and lower bounds for the $L^p(\lambda)$ approximation error of elements of $F_{s,d}$ by feed-forward ReLU neural networks. The bounds are expressed in terms of the number of weights of the network.

5.3.1 Known bounds on the sup norm approximation error

[154] gives matching (up to a certain constant) lower and upper bounds of the sup norm approximation error of the elements of $F_{s,d}$ by feed-forward ReLU neural networks.

Proposition 5.3.1 ([154]). Let $d \in \mathbb{N}^*$, $s > 0$, $\gamma \in \left(\frac{s}{d}, \frac{2s}{d}\right]$. Consider $n \in \mathbb{N}$ and $\alpha \in (0,1]$ such that $s = n + \alpha$.

There exist positive constants W_{min} and c_1, depending only on d and n, such that for any integer $W \geq W_{min}$, there exists a feed-forward ReLU neural network architecture $\mathcal{A}$ with $L = O(W^{\gamma\frac{d}{s}-1})$ layers and W weights such that

$$\sup_{f\in F_{s,d}} \inf_{g\in H_{\mathcal{A}}} \|f-g\|_\infty \leq c_1 W^{-\gamma}. \tag{5.3.1}$$

In the meantime, there exists a constant $c_2 > 0$ depending only on d and n such that, for any feed-forward neural network architecture $\mathcal{A}$ with W weights and $L = o(W^{\gamma\frac{d}{s}-1}/\log W)$ layers and for the ReLU activation function,

$$\sup_{f\in F_{s,d}} \inf_{g\in H_{\mathcal{A}}} \|f-g\|_\infty \geq c_2 W^{-\gamma}. \tag{5.3.2}$$

It is worth stressing that, for any probability measure μ on $[0,1]^d$, the upper bound (5.3.1) is automatically generalized to any smaller $L^p(\mu)$ norm, when $1 \leq p < +\infty$. However, the lower bound (5.3.2) does not immediately apply when $\|\cdot\|_\infty$ is replaced with $\|\cdot\|_{L^p(\mu)}$, $1 \leq p < +\infty$. The lower bound of the next subsection shows that, in this setting, approximation in $L^p(\lambda)$ norm is not easier than in sup norm, solving an open question of DeVore et al. [31].

5.3.2 Nearly-matching lower bounds of the $L^p(\lambda)$ approximation error

We first state a lower bound on the packing number of $F_{s,d}$, which is rather classical though hard to find in this specific form (see [18] for the L^∞ norm, or [35] for other Sobolev-type norms). For the sake of completeness, we give a proof of Lemma 36 in the supplement, Appendix 5.D.1.

Lemma 36. Let $s > 0$, $d \in \mathbb{N}^*$ and $1 \leq p < +\infty$. There exist constants $\varepsilon_0, c_0 > 0$ such that for any $0 < \varepsilon \leq \varepsilon_0$,

$$\log M\left(\varepsilon, F_{s,d}, \|\cdot\|_{L^p(\lambda)}\right) \geq c_0 \varepsilon^{-\frac{d}{s}}. \tag{5.3.3}$$

Given Lemma 36, we can use Corollary 1 to establish the next proposition and obtain the lower bound on the $L^p(\lambda)$ approximation error.

Proposition 5.3.2. Let $d \in \mathbb{N}^*$, $s > 0$, $\gamma \in \left(\frac{s}{d}, \frac{2s}{d}\right]$ and $1 \leq p < +\infty$. Consider $n \in \mathbb{N}$ and $\alpha \in (0,1]$ such that $s = n + \alpha$.

Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a piecewise-affine function, and $c > 0$. Then, there exist $c_1 > 0$ and $W_{min} \in \mathbb{N}^*$ (depending only on s, d, p, σ and c) such that for any architecture $\mathcal{A}$ of depth $1 \leq L \leq cW^{\gamma\frac{d}{s}-1}$ with $W \geq W_{min}$ weights, and for the activation σ, the set $H_{\mathcal{A}}$ (cf. Section 5.1) satisfies

$$\sup_{f\in F_{s,d}} \inf_{g\in H_{\mathcal{A}}} \|f-g\|_{L^p(\lambda)} \geq c_1 W^{-\gamma} \log^{-\frac{3s}{d}}(W)\ . \tag{5.3.4}$$

Note that the rate of the lower bound does not depend on p. Note also that, when the activation function is ReLU (which is piecewise-affine), we obtain a lower bound which matches the upper bound of the previous subsection up to logarithmic factors.

Proof. From Lemma 36, there exist $\varepsilon_0, c_0 > 0$ such that $\log M\left(\varepsilon, F_{s,d}, \|\cdot\|_{L^p(\lambda)}\right) \geq c_0\varepsilon^{-\frac{d}{s}}$ for all $0 < \varepsilon \leq \varepsilon_0$. Combining with Corollary 1 and using $L \leq cW^{\gamma\frac{d}{s}-1}$ concludes the proof. □

Remark 2 (Comparison with existing proof strategies in sup norm.). We would like to highlight a key difference between the proof of Proposition 5.3.2 and the lower bound proof strategies of [152, 153, 154, 125] that are specific to the sup norm. Their overall argument is roughly the following: if G can approximate any $f \in F$ in sup norm at accuracy $\varepsilon > 0$, since F contains many "oscillating" functions with oscillation amplitude roughly ε, then so must be the case for G (the sup norm is key here: **all** oscillations of any $f \in F$ are well approximated). Therefore, a small ε implies a large $\mathrm{VCdim}(G)$, which by contrapositive enables to lower bound the approximation error (5.1.1) with a decreasing function of $\mathrm{VCdim}(G)$, and therefore as a function of L and W. In contrast, in the proof of Theorem 9, the key probability result of Mendelson (Proposition 5.2.1) enables us to show that, even if the oscillations of any $f \in F$ are only well approximated **on average** (in $L^p(\mu)$ norm) by G, then $\mathrm{Pdim}(G)$ must be large when ε is small. The conclusion is then the same: the approximation error in $L^p(\mu)$ norm can be lower bounded as a function of $\mathrm{Pdim}(G)$, and therefore in terms of L, W. This solves the question of DeVore et al. [31] mentioned in the introduction, showing in particular that VC dimension theory can (surprisingly) be useful to prove L^p approximation lower bounds.

5.4 Approximation of monotonic functions by feed-forward neural networks

In this section, we consider the problem of approximating the set $\mathcal{M}^d$ of all non-decreasing functions from $[0,1]^d$ to $[0,1]$. These are functions $f : [0,1]^d \to [0,1]$ that are non-decreasing along any line parallel to an axis, i.e., such that, for all $x, y \in [0,1]^d$,

$$x_i \leq y_i,\ \forall i = 1, \dots, d \implies f(x) \leq f(y)\ .$$

Monotonic functions are an interesting case study for at least two reasons. First, they naturally appear in physics or engineering applications (consider for instance the braking distance of a vehicle as a function of variables such as the speed, the total load or the drag coefficient). Second, as will be shown in this section, because their sets of discontinuities can have "complex" shapes in dimension $d \geq 2$, monotonic functions provide a good example for which the approximation by feed-forward neural networks is hopeless in sup norm, but can be achieved in $L^p(\lambda)$ norm.

Next we focus on the approximation of $\mathcal{M}^d$ with Heaviside feed-forward neural networks. After proving an impossibility result for the sup norm, we show that the weaker goal of approximating $\mathcal{M}^d$ in $L^p(\lambda)$ norm is feasible, and derive nearly matching lower and upper bounds. Interestingly, the approximation rates depend on $p \geq 1$, which is in sharp contrast with the case of Hölder balls, that are not easier to approximate in $L^p(\lambda)$ norm than in sup norm (see Section 5.3).

5.4.1 Warmup: an impossibility result in sup norm

We start this section by showing that approximating monotonic functions of $d \geq 2$ variables in sup norm is impossible with Heaviside neural networks.

Proposition 5.4.1. For any neural network architecture $\mathcal{A}$ with the Heaviside activation, the set $H_{\mathcal{A}}$ (cf. Section 5.1) satisfies

$$\sup_{f\in\mathcal{M}^d}\inf_{g\in H_{\mathcal{A}}}\|f-g\|_\infty \geq \frac{1}{2}.$$

The proof of Proposition 5.4.1 is postponed to the supplement, Appendix 5.E.2. We show a slightly stronger result, by exhibiting a single function $f\in\mathcal{M}^d$ such that the lower bound of $\frac{1}{2}$ holds simultaneously for all network architectures.

Next we study the approximation of $\mathcal{M}^d$ in $L^p(\lambda)$ norm.

5.4.2 Lower bound in $L^p(\lambda)$ norm

We start by proving a lower bound, as a direct consequence of Corollary 1 and a lower bound on the packing number due to [41].

Proposition 5.4.2. Let $1\leq p<+\infty$, $d\geq 1$, and let $\alpha=\max\{d,(d-1)p\}$. Let $\sigma:\mathbb{R}\to\mathbb{R}$ be a piecewise-polynomial function having maximal degree $\nu\in\mathbb{N}$. Then, there exist positive constants c_1,c_2,c_3,W_{min} (depending only on d, p, and σ) such that for any architecture $\mathcal{A}$ of depth $L\geq 1$ with $W\geq W_{min}$ weights, and for the activation σ, the set $H_{\mathcal{A}}$ (cf. Section 5.1) satisfies

$$\sup_{f\in\mathcal{M}^d}\inf_{g\in H_{\mathcal{A}}}\|f-g\|_{L^p(\lambda)} \geq \begin{cases} c_1 W^{-\frac{2}{\alpha}}\log^{-\frac{2}{\alpha}}(W) & \text{if } \nu\geq 2\,,\\ c_2(LW)^{-\frac{1}{\alpha}}\log^{-\frac{3}{\alpha}}(W) & \text{if } \nu=1\,,\\ c_3 W^{-\frac{1}{\alpha}}\log^{-\frac{3}{\alpha}}(W) & \text{if } \nu=0\,. \end{cases} \tag{5.4.1}$$

Note that, contrary to the case of Hölder balls (Section 5.3), the rate of the lower bound depends on p through $\alpha=\max\{d,(d-1)p\}$.

Proof. From [41], there exist constants $\varepsilon_0,c_0>0$ such that for $\varepsilon\leq\varepsilon_0$, $\log M\left(\varepsilon,\mathcal{M}_d,\|\cdot\|_{L^p(\lambda)}\right)\geq c_0\varepsilon^{-\alpha}$. Using Corollary 1, we obtain the result. □

5.4.3 Nearly-matching upper bound in $L^p(\lambda)$ norm

To the best of our knowledge, there does not exist any upper-bound of the $L^p(\lambda)$ approximation error of $\mathcal{M}^d$ with feed-forward neural networks. Checking that all the lower-bounds of Proposition 5.4.2 are tight is out of the scope of this chapter and we leave it for future research [3]. However, we establish in the next proposition upper-bounds of the $L^p(\lambda)$ approximation error of $\mathcal{M}^d$ with feed-forward neural networks with the Heaviside activation function. This shows that, for the $L^p(\lambda)$ approximation error, the lower-bound obtained in (5.4.1), for $\nu=0$, is tight up to logarithmic factors. The next proposition follows by reinterpreting a metric entropy upper bound of [41] in terms of Heaviside neural networks. The proof is postponed to Appendix 5.E.1 in the supplement.

3. Obtaining an upper-bound for ReLU networks seems challenging. For example, the bit extraction technique used in [153] to find a sharp upper bound heavily relies on the local smoothness assumption of the function to approximate, which is not satisfied in general for monotonic functions.

Proposition 5.4.3. Let $1 \leq p < +\infty$, $d \in \mathbb{N} \setminus \{0, 1\}$ and let $\alpha = \max\{d, (d-1)p\}$. There exist positive constants W_{min} and c, depending only on d and p, such that for any integer $W \geq W_{min}$, there exists a feed-forward architecture $\mathcal{A}$ with two hidden layers, W weights and the Heaviside activation function such that the set $H_{\mathcal{A}}$ satisfies

$$\sup_{f \in \mathcal{M}^d} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_{L^p(\lambda)} \leq \begin{cases} cW^{-\frac{1}{\alpha}} & \text{if } p(d-1) \neq d\,, \\ cW^{-\frac{1}{d}} \log(W) & \text{if } p(d-1) = d\,. \end{cases} \tag{5.4.2}$$

5.5 Conclusion and other possible applications

We proved a general lower bound on the approximation error of F by G in $L^p(\mu)$ norm (Theorem 9), in terms of generic properties of F and G (packing number of F, range of F, fat-shattering dimension of G). The proof relies on VC dimension theory as in the sup norm case, but uses an additional key probabilistic argument due to Mendelson ([100], see Proposition 5.2.1), solving a question raised by DeVore et al. [31].

In Sections 5.3 and 5.4 we detailed two applications, where Corollary 1 yields nearly optimal approximation lower bounds in L^p norm, and which correspond to two examples where the approximation rate may depend or not depend on p.

Theorem 9 and Corollary 1 can be used to derive approximation lower bounds for many other cases. Corollary 1 only requires a (tight) lower bound on the packing number of F, for which approximation theory provides several examples. For instance, for the *Barron space* introduced in [10], Petersen and Voigtlaender [112] showed a tight lower bound on the log packing number in $L^p(\lambda, [0,1]^d)$ norm, of order $\varepsilon^{-2d/(d+2)}$. Applying Corollary 1, this yields an approximation lower bound of $(LW)^{-\left(\frac{1}{2}+\frac{1}{d}\right)} \log^{-3\left(\frac{1}{2}+\frac{1}{d}\right)}(W)$ for ReLU networks (see Appendix 5.F in the supplement for details). Other examples of sets F for which tight lower bounds on the packing number (or metric entropy) are available include: multivariate cumulative distribution functions [19], multivariate convex functions [53], and functions with other shape constraints [49].

Piecewise-polynomial activation functions are not essential for the current derivation. Indeed, Theorem 9 can also be applied to the case where G corresponds to a neural network with other activation functions such as the sigmoid. In the sigmoid case, the pseudo-dimension is known to be at most of the order of W^4 (see [74, 2]), which we can use to derive an approximation lower bound similar to that of Corollary 1, with a smaller right-hand side for large W. However, to the best of our knowledge, it is not known whether the $\mathcal{O}(W^4)$ VC bound is tight (only a lower bound of the order of W^2 is known), so the resulting approximation lower bound could be loose. We leave this interesting question for future work.

Theorem 9 can also be applied to other approximating sets G, beyond classical feed-forward neural networks, as soon as a (tight) upper bound on the fat-shattering dimension of G is available. For example, upper bounds were derived by [143] on the VC dimension of partially quantized networks, while [14] derived bounds on the fat-shattering dimension of some RKHS. Investigating such applications and whether the obtained approximation lower bounds are rate-optimal is a natural research direction for the future.

Appendix

5.A Feed-forward neural networks: formal definition

In this chapter, we use the following classical graph-theoretic definitions for feed-forward neural networks given, e.g., in [12] (with slightly different terms and notation).

A *feed-forward neural network architecture* $\mathcal{A}$ of depth $L \geq 1$ is a directed acyclic graph (V, E) with $d \geq 1$ nodes with in-degree 0 (also called the *input neurons*), a single node with out-degree 0 (also called the *output neuron*), and such that the longest path in the graph has length L.

We define layers $\ell = 0, 1, \ldots, L$ recursively as follows:

— layer 0 is the set V_0 of all input neurons; we assume that $V_0 = \{1, \ldots, d\}$ without loss of generality.

— for any $\ell = 1, \ldots, L$, layer ℓ is the set V_ℓ of all nodes that have one or several predecessors [4] in layer $\ell - 1$, possibly other predecessors in layers $0, 1, \ldots, \ell - 2$, but no other predecessors.

Layer L consists of a single node: the output neuron. Layers $1, \ldots, L-1$ are called the *hidden layers* (if $L \geq 2$). Note that skip connections are allowed, i.e., there can be connections between non-consecutive layers.

Given a feed-forward neural network architecture $\mathcal{A}$ of depth $L \geq 1$, we associate real numbers $w_e \in \mathbb{R}$ to all edges $e \in E$ and $w_v \in \mathbb{R}$ to all nodes $v \in V_1 \cup \ldots \cup V_L$. These real numbers are called *weights* (they correspond to linear coefficients and biases) and are concatenated in a *weight vector* $\mathbf{w} \in \mathbb{R}^W$, where $W = \mathrm{Card}(E) + \sum_{\ell=1}^{L} \mathrm{Card}(V_\ell)$ is the total number of weights.

Given $\mathcal{A}$, an associated weight vector $\mathbf{w} \in \mathbb{R}^W$, and a function $\sigma : \mathbb{R} \to \mathbb{R}$ (called *activation function*), the network represents the function $g_{\mathbf{w}} : \mathbb{R}^d \to \mathbb{R}$ defined recursively as follows. We write $P_v \subset V$ for the set of all predecessors of any node $v \in V$, and $w_{u \to v}$ for the weight on the edge from u to v. The recursion from layer $\ell = 0$ to layer $\ell = L$ reads: given $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$,

— each input neuron $v \in \{1, \ldots, d\}$ outputs the value $y_v := x_v$;

— for any $\ell = 1, \ldots, L-1$, each neuron $v \in V_\ell$ outputs $y_v := \sigma\big(\sum_{u \in P_v} w_{u \to v} y_u + w_v\big)$;

— the unique output neuron $v \in V_L$ outputs $g_{\mathbf{w}}(x) := \sum_{u \in P_v} w_{u \to v} y_u + w_v$.

Finally, we define $H_{\mathcal{A}} := \{g_{\mathbf{w}} : \mathbf{w} \in \mathbb{R}^W\}$ to be the set of all functions that can be represented by tuning all the weights assigned to the network (the dependency on the activation function σ is not written explicitly).

5.B Main results: technical details

We provide technical details that were missing to establish Proposition 5.2.1, Theorem 9 and Corollary 1.

4. A node $u \in V$ is a predecessor of another node $v \in V$ if there is a directed edge from u to v.

5.B.1 Proof of Proposition 5.2.1

Proposition 5.2.1 is a direct extension of [100, Corollary 3.12] to any range $[a, b]$. We first recall this result but in slightly different terms (see the comments afterwards).

Proposition 5.B.1 (Corollary 3.12 in [100], "almost equivalent" statement). Let G be a set of measurable functions from a measurable space $\mathcal{X}$ to $[0, 1]$, such that $\mathrm{fat}_\gamma(G) < +\infty$ for all $\gamma > 0$. Then, for every $1 \leq p < +\infty$, there is some constant $c_p > 0$ depending only on p such that, for every probability measure μ on $\mathcal{X}$ and every $\varepsilon > 0$,

$$\log M\left(\varepsilon, G, \|\cdot\|_{L^p(\mu)}\right) \leq c_p \,\mathrm{fat}_{\frac{\varepsilon}{32}}(G) \log^2\left(\frac{2\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)}{\varepsilon}\right).$$

To be precise, [100, Corollary 3.12] was stated a little differently. Instead of the assumption on $\mathrm{fat}_\gamma(G)$, there were two conditions on G: (i) G satisfies a weak measurability assumption such as the "image admissible Suslin" property, and (ii) G is a uniform Glivenko-Cantelli class. Fortunately, note that assumption (i) could easily be checked in special cases such as the setting of Corollary 1, and that assumption (ii) is equivalent to $\mathrm{fat}_\gamma(G) < +\infty$ for all $\gamma > 0$ when (i) holds and when G only consists of $[0, 1]$-valued functions (see [1], Theorem 2.5). The two statements are thus "almost equivalent". However, we stress that (i) and (ii) are not necessary (assuming $\mathrm{fat}_\gamma(G) < +\infty$ for all $\gamma > 0$). To see why, it suffices to adapt the proof of [100, Corollary 3.12] as follows: instead of starting from an ε-packing of G in empirical $L^p(\mu_n)$ norm and showing that it is also an ε'-covering of G in $L^p(\mu)$ norm, with $\varepsilon' > \varepsilon$, we can start from an ε-packing of G in $L^p(\mu)$ norm and show that it is also an ε-packing of G in empirical $L^p(\mu_n)$ norm for some large integer n (with positive probability). This last statement directly follows from the Hoeffding inequality: no uniform law of large numbers is required, since we only need to compare empirical averages to their expectations for a finite number of bounded functions. [5]

We now explain how to derive Proposition 5.2.1 (with an arbitrary range $[a, b]$) as a straightforward consequence of Proposition 5.B.1.

Proof (of Proposition 5.2.1). In order to apply Proposition 5.B.1, we reduce the problem from $[a, b]$ to $[0, 1]$ by translating and rescaling every function in G. For $g \in G$, we define $\tilde{g} : \mathcal{X} \to [0, 1]$ by $\tilde{g}(x) = \frac{g(x)-a}{b-a}$, and we set

$$\tilde{G} = \{\tilde{g} \;:\; g \in G\} \;.$$

Note that every $\tilde{g} \in \tilde{G}$ is indeed $[0, 1]$-valued.

We now note that translation does not affect packing numbers nor the fat-shattering dimension, while rescaling only changes the scale ε by a factor of $b - a$. More precisely, we have the following two properties:

Property 1: For all $u > 0$, $\mathrm{fat}_{\frac{u}{b-a}}(\tilde{G}) = \mathrm{fat}_u(G)$.

Property 2: For all $u > 0$, $M\left(\frac{u}{b-a}, \tilde{G}, \|\cdot\|_{L^p(\mu)}\right) = M\left(u, G, \|\cdot\|_{L^p(\mu)}\right)$.

5. In passing, all occurrences of $\mathrm{fat}_{\frac{\varepsilon}{32}}(G)$ could be replaced with $\mathrm{fat}_{\frac{\varepsilon}{8}}(G)$.

Before proving the two properties (see below), we first conclude the proof of Proposition 5.2.1. By Property 1, $\mathrm{fat}_\gamma(\tilde{G}) = \mathrm{fat}_{\gamma(b-a)}(G)$, which by assumption is finite for all $\gamma > 0$. Since every $\tilde{g} \in \tilde{G}$ is $[0,1]$-valued, we can thus apply Proposition 5.B.1. Using it with $\tilde{\varepsilon} = \varepsilon/(b-a)$, we get

$$\log M\Big(\tilde{\varepsilon}, \tilde{G}, \|\cdot\|_{L^p(\mu)}\Big) \leq c_p \,\mathrm{fat}_{\frac{\tilde{\varepsilon}}{32}}(\tilde{G}) \log^2\left(\frac{2\,\mathrm{fat}_{\frac{\tilde{\varepsilon}}{32}}(\tilde{G})}{\tilde{\varepsilon}}\right) .$$

Combining with the two equalities in Properties 1 and 2, we obtain

$$\log M\big(\varepsilon, G, \|\cdot\|_{L^p(\mu)}\big) \leq c_p \,\mathrm{fat}_{\frac{\varepsilon}{32}}(G) \log^2\left(\frac{2(b-a)\,\mathrm{fat}_{\frac{\varepsilon}{32}}(G)}{\varepsilon}\right) ,$$

which concludes the proof of Proposition 5.2.1.

We now prove the two properties.

Proof of Property 1. We first show that $\mathrm{fat}_{\frac{u}{b-a}}(\tilde{G}) \geq \mathrm{fat}_u(G)$. To that end, let $S = \{x_1, \ldots, x_m\}$ and $r : S \to \mathbb{R}$ be such that for any $E \subset S$, there exists $g \in G$ such that $g(x) \geq r(x) + u$ if $x \in E$ and $g(x) \leq r(x) - u$ otherwise. Setting $\tilde{r}(x) = \frac{r(x)-a}{b-a}$, we can see that $\tilde{g}(x) \geq \tilde{r}(x) + \frac{u}{b-a}$ if $x \in E$ and $\tilde{g}(x) \leq \tilde{r}(x) - \frac{u}{b-a}$ otherwise, which proves $\mathrm{fat}_{\frac{u}{b-a}}(\tilde{G}) \geq \mathrm{fat}_u(G)$. The reverse inequality is proved similarly.

Proof of Property 2. Let $\{g_1, \ldots, g_m\}$ be a u-packing of G in $L^p(\mu)$ norm. This means that $\|g_i - g_j\|_{L^p(\mu)} > u$ and therefore $\|\tilde{g}_i - \tilde{g}_j\|_{L^p(\mu)} > \frac{u}{b-a}$ for all $i \neq j \in \{1, \ldots, m\}$, so that $\{\tilde{g}_1, \ldots, \tilde{g}_m\} \subset \tilde{G}$ is a $\frac{u}{b-a}$-packing of $\tilde{G}$. This proves $M\big(\frac{u}{b-a}, \tilde{G}, \|\cdot\|_{L^p(\mu)}\big) \geq M\big(u, G, \|\cdot\|_{L^p(\mu)}\big)$. The reverse inequality is proved similarly. □

5.B.2 Clipping can only help

The next two lemmas indicate that clipping (truncature) to a known range can only help. These are key to apply Proposition 5.2.1 in our setting. In the sequel, for a set G of functions from a set $\mathcal{X}$ to $\mathbb{R}$, and for $a < b$ in $\mathbb{R}$, we denote by $G_{[a,b]}$ the set of all functions in G whose values are truncated (clipped) to the segment $[a,b]$, that is, $G_{[a,b]} = \{\tilde{g} : g \in G\}$, where $\tilde{g} : \mathcal{X} \to \mathbb{R}$ is given by

$$\forall x \in \mathcal{X}, \quad \tilde{g}(x) = \min(\max(a, g(x)), b) .$$

Lemma 37. Let G be a set of functions defined on a set $\mathcal{X}$, and with values in $\mathbb{R}$. Let $G_{[a,b]}$ be defined as above. Then, for any $\gamma > 0$,

$$\mathrm{fat}_\gamma(G) \geq \mathrm{fat}_\gamma(G_{[a,b]}) .$$

Proof. Let $\gamma > 0$. The case when $\mathrm{fat}_\gamma(G_{[a,b]}) = 0$ is straightforward. We thus assume that $\mathrm{fat}_\gamma(G_{[a,b]}) \geq 1$. To prove the result, we show that any subset A of X that is γ-*shattered* by $G_{[a,b]}$ is also γ-*shattered* by G. Let us consider such a subset $A = \{x^1, \ldots, x^N\} \subset X$, with cardinality $N \geq 1$. Hence, there exists $\{r_1, \ldots, r_N\} \subset \mathbb{R}$ such that for any $E \subset A$,

there exists $\tilde{g} \in G_{[a,b]}$ such that $\tilde{g}(x_i) - r_i \geq \gamma$ if $x_i \in E$ and $\tilde{g}(x_i) - r_i \leq -\gamma$ otherwise. Note that this must imply that $r_i \in]a, b[$ for all $i = 1, \dots, N$ (indeed, by choosing E such that $x_i \in E$ or not, we have either $r_i + \gamma \leq \tilde{g}(x_i) \leq b$ or $r_i - \gamma \geq \tilde{g}(x_i) \geq a$). Now fix $i \in \{1, \dots, N\}$ and let us assume $\tilde{g}(x_i) - r_i \geq \gamma$ (by symmetry, the reversed case $\tilde{g}(x_i) - r_i \leq -\gamma$ is treated the same way). Because $r_i > a$, this implies that $\tilde{g}(x_i) > a$ and thus $g(x_i) \geq \tilde{g}(x_i)$ (by definition of $\tilde{g}$), which entails $g(x_i) - r_i \geq \gamma$. It follows that if $G_{[a,b]}$ *γ-shatters* A, then G also *γ-shatters* A, and the result follows. □

The following lemma formalizes the well-known idea that it is easier to approach a function with values in a finite range by a function with values in the same range.

Lemma 38. Let G be a set of measurable functions from a measurable space $\mathcal{X}$ to $\mathbb{R}$, and let $G_{[a,b]}$ be defined as above. Assume F is a set of measurable functions from $\mathcal{X}$ to $[a, b]$. Then, for any probability measure μ on $\mathcal{X}$,

$$\sup_{f \in F} \inf_{g \in G} \|f - g\|_{L^p(\mu)} \geq \sup_{f \in F} \inf_{\tilde{g} \in G_{[a,b]}} \|f - \tilde{g}\|_{L^p(\mu)} .$$

Proof. To prove the above result, it is enough to show that for any $f \in F$ and $g \in G$, the function $\tilde{g}$ is pointwise at least as close to f as g is, which for all $f \in F$ yields $\inf_{g \in G} \|f - g\|_{L^p(\mu)} \geq \inf_{\tilde{g} \in G_{[a,b]}} \|f - \tilde{g}\|_{L^p(\mu)}$. By definition of $G_{[a,b]}$, for any $x \in \mathcal{X}$, if $g(x) \in [a, b]$, then $|f(x) - g(x)| = |f(x) - \tilde{g}(x)|$. And if $g(x) \notin [a, b]$, then $|f(x) - \tilde{g}(x)| < |f(x) - g(x)|$ since $f(x) \in [a, b]$. It follows that the discrepancy $|f - \tilde{g}|$ is everywhere bounded by $|f - g|$, and the result follows. □

5.B.3 Missing details in the proof of Theorem 9

We provide all details that were missing to derive (5.2.8), which is a direct consequence of Lemma 39 below. We follow the convention $aP^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(P) = +\infty$ when $P = 1$.

Lemma 39. Let $P \in \mathbb{N}^*$ and $c, \alpha, r > 0$. There exist constants $a, \varepsilon_1'' > 0$ depending only on c, α and r such that, for all $\varepsilon \in (0, r)$ satisfying

$$\varepsilon^{-\alpha} \leq cP \log^2\left(\frac{rP}{\varepsilon}\right) , \tag{5.B.1}$$

we have

$$\varepsilon \geq \min\left(\varepsilon_1'', aP^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(P)\right) .$$

Proof. Assume $\varepsilon \in (0, r)$ is such that (5.B.1) holds. To show the result, we study the function $f : (1/r, +\infty) \to \mathbb{R}$ defined for all $x > 1/r$ by

$$f(x) = \frac{x^\alpha}{\log^2(rPx)} .$$

Note that (5.B.1) implies that $f(1/\varepsilon) \leq cP$. For all $P \geq 2$, we set

$$\varepsilon_P = P^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(P) . \tag{5.B.2}$$

Let $P_1 \geq 2$ be such that $P_1^{\frac{1}{\alpha}} \log^{\frac{2}{\alpha}}(P_1) \geq \frac{\exp(\frac{2}{\alpha})}{r}$. For all $P \geq P_1$, we have $\frac{1}{\varepsilon_P} \geq \frac{\exp(\frac{2}{\alpha})}{r} > 1/r$ and

$$f\left(\frac{1}{\varepsilon_P}\right) = \frac{P \log^2(P)}{\log^2\left(rP^{1+\frac{1}{\alpha}} \log^{\frac{2}{\alpha}}(P)\right)} .$$

Since

$$\lim_{Q \to +\infty} \frac{\log^2(Q)}{\log^2\left(rQ^{1+\frac{1}{\alpha}} \log^{\frac{2}{\alpha}}(Q)\right)} = \frac{1}{(1+\frac{1}{\alpha})^2} =: c_1 ,$$

there exists P_2 such that for all $Q \geq P_2$, we have $\frac{\log^2(Q)}{\log^2\left(rQ^{1+\frac{1}{\alpha}} \log^{\frac{2}{\alpha}}(Q)\right)} \geq \frac{c_1}{2}$.

Below we distinguish the cases $P \geq \max(P_1, P_2)$ and $P < \max(P_1, P_2)$.

1st case: $P \geq \max(P_1, P_2)$.
We have $f\left(\frac{1}{\varepsilon_P}\right) \geq \frac{c_1 P}{2}$ and $P \geq \frac{1}{c} f\left(\frac{1}{\varepsilon}\right)$ (by (5.B.1)), so that $f\left(\frac{1}{\varepsilon_P}\right) \geq \frac{c_1}{2c} f\left(\frac{1}{\varepsilon}\right)$. We now use Lemma 40 below with $b = \frac{c_1}{2c}$: setting $a := (b/2)^{1/\alpha} = (c_1/(4c))^{1/\alpha}$, there exists $x_1 > \max\left\{\frac{1}{r}, \frac{1}{ar}\right\}$ depending only on r, b, α such that $bf(x) \geq f(ax)$ for all $x \geq x_1$.

Therefore, if $\varepsilon < \frac{1}{x_1} =: \varepsilon_1$, then $\frac{c_1}{2c} f\left(\frac{1}{\varepsilon}\right) \geq f\left(\frac{a}{\varepsilon}\right)$. Therefore $f(\frac{1}{\varepsilon_P}) \geq f\left(\frac{a}{\varepsilon}\right)$.

Recall from (5.B.2) and $P \geq P_1$ that $\frac{1}{\varepsilon_P} \geq \frac{\exp(\frac{2}{\alpha})}{r}$. If $\varepsilon < \frac{ar}{\exp(\frac{2}{\alpha})} =: \varepsilon_2$, then we also have $\frac{a}{\varepsilon} \geq \frac{\exp(\frac{2}{\alpha})}{r}$. Therefore, using Lemma 40 again, $f(\frac{1}{\varepsilon_P}) \geq f(\frac{a}{\varepsilon})$ implies that $\frac{1}{\varepsilon_P} \geq \frac{a}{\varepsilon}$, that is,

$$\varepsilon \geq a\, \varepsilon_P .$$

Summarizing, when $\varepsilon \in (0, r)$ satisfies (5.B.1) and when $P \geq \max(P_1, P_2)$, either $\varepsilon \geq \varepsilon_1$ or $\varepsilon \geq \varepsilon_2$ or $\varepsilon \geq a\, \varepsilon_P$. Put differently,

$$\varepsilon \geq \min(\varepsilon_1, \varepsilon_2, a\, \varepsilon_P) . \tag{5.B.3}$$

2nd case: $P < \max(P_1, P_2) =: P_3$.
Using (5.B.1) and the fact that $t \mapsto ct \log^2\left(\frac{rt}{\varepsilon}\right)$ is non-decreasing on $[\varepsilon/r, +\infty)$, together with $\varepsilon/r \leq 1 \leq P \leq P_3$ yields $\varepsilon^{-\alpha} \leq cP_3 \log^2(rP_3/\varepsilon)$. This entails that, for some $\varepsilon_3 > 0$ depending only on α, c, P_3, r,

$$\varepsilon \geq \varepsilon_3 . \tag{5.B.4}$$

Conclusion: combining the two cases, when $\varepsilon \in (0, r)$ satisfies (5.B.1), whatever $P \in \mathbb{N}^*$, we have (5.B.3) or (5.B.4). Setting $\varepsilon_1'' = \min(\varepsilon_1, \varepsilon_2, \varepsilon_3)$, we obtain

$$\varepsilon \geq \min\left(\varepsilon_1'', a\, P^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(P)\right) .$$

(Note that this is also true in the case $P = 1$, by the convention $aP^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(P) = +\infty$.) Since $\varepsilon_1, \varepsilon_2, \varepsilon_3$ and a only depend on c, α, r, this concludes the proof. □

Lemma 40. Let $\alpha, r > 0$ and $P \in \mathbb{N}^*$. We define $f(x) = \frac{x^\alpha}{\log^2(rPx)}$ for all $x > 1/r$. Then:

i) f is increasing on $I := \left[\frac{\exp(\frac{2}{\alpha})}{r}, +\infty\right)$ and $\lim_{x \to +\infty} f(x) = +\infty$.

ii) for all $b > 0$, setting $a := (b/2)^{1/\alpha}$, there exists $x_1 > \max\{\frac{1}{r}, \frac{1}{ar}\}$ depending only on r, b, α such that,

$$\forall x \geq x_1 \ , \qquad bf(x) \geq f(ax) \ .$$

Proof. **Proof of i):** The fact that $\lim_{x\to+\infty} f(x) = +\infty$ is because $\alpha > 0$. To see why f is increasing on I, note that

$$f'(x) = \frac{\alpha x^{\alpha-1}\log^2(rPx) - x^\alpha 2\log(rPx)\frac{1}{x}}{\log^4(rPx)} = \frac{x^{\alpha-1}\log(rPx)(\alpha\log(rPx) - 2)}{\log^4(rPx)} \ ,$$

so that $f'(x) > 0$ for all $x > \frac{\exp(\frac{2}{\alpha})}{rP}$, and in particular for all $x > \frac{\exp(\frac{2}{\alpha})}{r}$ (since $P \geq 1$). This proves that f is increasing on I.

Proof of ii): Let $b > 0$ and set $a := (b/2)^{1/\alpha}$. Let $x_1 > \max\{\frac{1}{r}, \frac{1}{ar}\}$ depending only on r, b, α such that, for all $u \geq x_1$,

$$\frac{\log^2(ru)}{\log^2(rau)} \leq 2 \ .$$

(Such an x_1 exists since the ratio converges to 1 as $u \to +\infty$, and we can choose x_1 as a function of r, a only.) Now, for all $x \geq x_1$, using the above inequality with $u = Px \geq x$ (since $P \geq 1$), we get

$$\frac{f(ax)}{f(x)} = a^\alpha \frac{\log^2(rPx)}{\log^2(rPax)} \leq 2a^\alpha = b \ ,$$

where the last equality is because $a := (b/2)^{1/\alpha}$. This proves that $bf(x) \geq f(ax)$ for all $x \geq x_1$. □

5.B.4 Proof of Corollary 1

We first recall some definitions and two key bounds on the VC-dimension of piecewise-polynomial feed-forward neural networks, proved by [46] and [12].

For a set F of functions from $\mathcal{X}$ to $\{-1, 1\}$, we say that a set $S = \{x_1 \ldots, x_N\} \subset \mathcal{X}$ is *shattered* by F if for any $E \subset S$, there exists $f \in F$ satisfying for all $i = 1, \ldots, N$, $f(x_i) = 1$ if $x_i \in E$, and $f(x_i) = -1$ if $x_i \notin E$. The VC-dimension of F, denoted by $\mathrm{VCdim}(F)$, is defined as the largest number $N \geq 1$ such that there exists $S \subset \mathcal{X}$ of cardinality N which is shattered by F (by convention, $\mathrm{VCdim}(F) = 0$ if no such set S exists, while $\mathrm{VCdim}(F) = +\infty$ if there exist sets S of unbounded cardinality N).

Let $\mathcal{B}$ be any feed-forward neural network architecture of depth $L \geq 1$ with $W \geq 1$ weights, $d \geq 1$ input neurons, and $U \geq 1$ hidden or output neurons. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be any piecewise-polynomial activation function on $K \geq 2$ pieces, with maximal degree $\nu \in \mathbb{N}$. Denote by $\mathrm{sgn}(H_\mathcal{B}) = \{\mathrm{sgn}(g_\mathbf{w}) : \mathbf{w} \in \mathbb{R}^W\}$ the set of all classifiers obtained by looking at the sign of the network's output, that is, the classifiers defined by $\mathrm{sgn}(g_\mathbf{w})(x) = \mathbb{1}_{\{g_\mathbf{w}(x)>0\}}$ for all $x \in \mathbb{R}^d$.

Goldberg and Jerrum [46] showed that, for some constant $c_1' > 0$ depending only on d,

ν and K, the VC-dimension of $\mathrm{sgn}(H_{\mathcal{B}})$ is bounded as follows (see also Theorem 8.7 in [2]):

$$\mathrm{VCdim}(\mathrm{sgn}(H_{\mathcal{B}})) \leq c_1' W^2 \,. \tag{5.B.5}$$

This bound was refined for piecewise-affine activation functions. Namely, Bartlett et al. [12, Theorem 7] proved that, if $U \geq 3$, then, for some $R \leq U + U(L-1)\nu^{L-1}$,

$$\mathrm{VCdim}(\mathrm{sgn}(H_{\mathcal{B}})) \leq L + \bar{L} W \log_2\Big(4e(K-1)R\log_2\big(2e(K-1)R\big)\Big) \,,$$

where $\bar{L} = 1$ if $\nu = 0$, and $\bar{L} \leq L$ otherwise. Therefore, for some constants $W'_{\min} \geq 1$ and $c_2', c_3' > 0$ depending only on d and K, we have, for all $W \geq W'_{\min}$ (which in particular implies $U \geq 3$),

$$\mathrm{VCdim}(\mathrm{sgn}(H_{\mathcal{B}})) \leq \begin{cases} c_2' L W \log(W) & \text{if } \nu = 1 \,, \\ c_3' W \log(W) & \text{if } \nu = 0 \,. \end{cases} \tag{5.B.6}$$

We are now ready to prove Corollary 1 from Theorem 9.

Proof (of Corollary 1). In order to apply Theorem 9, we first bound $P := \mathrm{Pdim}(H_{\mathcal{A}})$ from above. The bounds (5.B.5) and (5.B.6) were on the VC-dimension of $\mathrm{sgn}(H_{\mathcal{B}})$, for any feed-forward neural network architecture $\mathcal{B}$, while we need a bound on the pseudo-dimension. However, by a well-known trick (e.g., Theorem 14.1 in [2]), the pseudo-dimension of $H_{\mathcal{A}}$ is upper bounded by the VC-dimension of (the sign of) an augmented network architecture of depth L, with $d+1$ input neurons and $W+1$ weights. [6] Therefore, replacing (d, W) with $(d+1, W+1)$ in (5.B.5) and (5.B.6), we get that, for some constants $\tilde{W}_{\min} \geq 1$ and $\tilde{c_1}, \tilde{c_2}, \tilde{c_3} > 0$ depending only on d, ν and K, for all $W \geq \tilde{W}_{\min}$,

$$P \leq \begin{cases} \tilde{c_1} W^2 & \text{if } \nu \geq 2 \,, \\ \tilde{c_2} L W \log(W) & \text{if } \nu = 1 \,, \\ \tilde{c_3} W \log(W) & \text{if } \nu = 0 \,. \end{cases} \tag{5.B.7}$$

Now, by Theorem 9, we have, for some constants $c_1, \varepsilon_1 > 0$ depending only on $b-a$, p, c_0, ε_0, α,

$$\sup_{f \in F} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_{L^p(\mu)} \geq \min\Big\{\varepsilon_1,\ c_1 P^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(P)\Big\} \,. \tag{5.B.8}$$

Noting that $P \mapsto \min\Big\{\varepsilon_1,\ c_1 P^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(P)\Big\}$ is non-increasing and plugging (5.B.7) into (5.B.8), we get, for $W \geq W_{\min}$,

$$\sup_{f \in F} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_{L^p(\mu)} \geq \min\left\{\varepsilon_1, \begin{pmatrix} c_4 W^{-\frac{2}{\alpha}} \log^{-\frac{2}{\alpha}}(W^2) & \text{if } \nu \geq 2 \\ c_5 (LW\log(W))^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(LW\log(W)) & \text{if } \nu = 1 \\ c_6 (W\log(W))^{-\frac{1}{\alpha}} \log^{-\frac{2}{\alpha}}(W\log(W)) & \text{if } \nu = 0 \end{pmatrix}\right\}$$

6. This is because $\mathrm{Pdim}(H_{\mathcal{A}}) = \mathrm{VCdim}\big(\{(x,r) \in \mathbb{R}^d \times \mathbb{R} \mapsto \mathbb{1}_{\{g(x)-r>0\}} : g \in H_{\mathcal{A}}\}\big)$, the output neuron of $\mathcal{A}$ is linear, and we allow skip connections.

for some constants $W_{\min} \geq 1$ and $c_4, c_5, c_6 > 0$ depending only on d, ν, K, $b-a$, p, c_0, ε_0 and α. Taking $W_{\min}$ large enough, the first term ε_1 is always larger than the second term in the above minimum, and the logarithmic terms $\log(W\log(W))$ and $\log(LW\log(W))$ can be upper bounded by a constant times $\log(W)$ (since $L \leq W$). Rearranging concludes the proof. □

5.C Earlier works: two other lower bound proof strategies

Approximation lower bounds in a sense similar to ours have been obtained in other recent works. In the purpose of highlighting the differences between the different approaches, we describe the lower bound proof strategies of Yarotsky [152] and of Petersen and Voigtlaender [111].

5.C.1 Approximation in sup norm of Sobolev unit balls with ReLU networks [152]

Recall that the Sobolev space $\mathcal{W}^{n,\infty}([0,1]^d)$ is defined as the set of functions on $[0,1]^d$ lying in L^∞ along with all their weak derivatives up to order n. We equip this space with the norm

$$\|f\|_{\mathcal{W}^{n,\infty}([0,1]^d)} = \max_{\mathbf{n}\in\mathbb{N}^d:|\mathbf{n}|\leq n} \operatorname*{ess\,sup}_{x\in[0,1]^d} |D^{\mathbf{n}} f(x)|,$$

and we let $F_{n,d}$ be the unit ball of this space.

We first state the sup norm lower bound and then we give a synthesized version of the proof.

Proposition 5.C.1 ([152]). There exists positive constants $W_{\min}, c > 0$ such that for any feed-forward neural network with architecture $\mathcal{A}$, ReLU activation and $W \geq W_{\min}$ weights,

$$\sup_{f\in F_{n,d}} \inf_{g\in H_{\mathcal{A}}} \|f-g\|_\infty \geq cW^{-\frac{2n}{d}}.$$

Details aside, the proof reads as follows. The author assumes that $H_{\mathcal{A}}$ approximates $F_{n,d}$ with error ε. Fixing $N = c_{n,d}(3\varepsilon)^{-1/n}$ for a properly chosen constant $c_{n,d} > 0$, he constructs a set of functions in $F_{n,d}$ that can shatter a grid of N^d points $x_1, \ldots, x_{N^d}$ evenly distributed over $[0,1]^d$. The assumption that $H_{\mathcal{A}}$ approximates $F_{n,d}$ in sup norm with error ε allows to conclude that $H_{\mathcal{A}}$ also shatters $\{x_1, \ldots, x_{N^d}\}$, and hence, $\mathrm{VCdim}(H_{\mathcal{A}}) \geq N^d = c'_{n,d}\varepsilon^{-\frac{d}{n}}$, for a properly chosen constant $c'_{n,d} > 0$. The author concludes using the upper bound on $\mathrm{VCdim}(H_{\mathcal{A}})$ with respect to W from [2] which yields $\mathrm{VCdim}(H_{\mathcal{A}}) \leq c'W^2$ for some constant c'.

It is worth stressing that in this proof, it is paramount to assume that $H_{\mathcal{A}}$ approximates $F_{n,d}$ in sup norm, rather than any L^p norm with $p < +\infty$. The reason is that only this choice of norm allows to bound the discrepancy between $f \in F_{n,d}$ and $g_f \in H_{\mathcal{A}}$ chosen optimally with respect to f at any chosen points. Our proof strategy relying on Proposition 5.2.1 allows to circumvent this issue by relating the pseudo-dimension to the metric entropy with respect to any L^p norm, $1 \leq p < +\infty$.

5.C.2 Approximation in L^p norm of *Horizon functions* with quantized networks [111]

The authors study *quantized* neural networks, that is, networks with weights constrained to be representable with a fixed number of bits. They obtain a lower bound on the minimal number of weights in a quantized neural network that can approximate a set of *Horizon functions* in L^p norm, $p > 0$, with error $\varepsilon > 0$. This lower bound is easily invertible to a bound on the approximation error and is thus comparable to the results we obtain in this chapter.

Textually, the authors introduce the set of horizon functions as follows: "These are $\{0,1\}$-valued functions with a jump along a hypersurface and such that the jump surface is the graph of a smooth function" [111]. Denoting by H the indicator function of the set $[0,+\infty) \times \mathbb{R}^{d-1}$, the set of horizon functions reads as

$$\mathcal{HF}_{\beta,d,B} = \left\{ f \circ T \in L^\infty\left(\left[-\frac{1}{2},\frac{1}{2}\right]^d\right) : \right.$$
$$\left. f(x) = H(x_1 + \gamma(x_2,\dots,x_d), x_2, \dots, x_d), \gamma \in \mathcal{F}_{\beta,d-1,B},\ T \in \Pi(d,\mathbb{R}) \right\},$$

where $\mathcal{F}_{\beta,d-1,B}$ denotes the set of Hölder functions over $[-1/2,1/2]^{d-1}$ whith smoothness parameter β and with norm $\|.\|_{\mathcal{C}^{n,\alpha}}$ bounded by B (see Section 5.3), and $\Pi(d,\mathbb{R})$ denotes the group of d-dimensional permutation matrices.

In the following, for any nonzero integer K and any neural network architecture $\mathcal{A}$, we denote by $H_{\mathcal{A}}^K \subset H_{\mathcal{A}}$ the set of K-quantized functions in $H_{\mathcal{A}}$; namely, the functions in $H_{\mathcal{A}}$ with weights representable using at most K bits. The lower bound in [111] (Theorem 4.2) reads as follow:

Proposition 5.C.2 ([111]). Let $d \geq 2$. Let $p, \beta, B, c_0 > 0$ and let $\sigma : \mathbb{R} \to \mathbb{R}$ be such that $\sigma(0) = 0$. There exist positive constants $\varepsilon_0, c > 0$ depending only on d, p, β, B and c_0 such that, for any $\varepsilon \leq \varepsilon_0$, setting $K = \lceil c_0 \log(1/\varepsilon) \rceil$, for any feed-forward neural network architecture $\mathcal{A}$ with W weights and activation σ such that $H_{\mathcal{A}}^K$ approximates $\mathcal{HF}_{\beta,d,B}$ in L^p norm with error less than ε, we have

$$W \geq c\varepsilon^{-\frac{p(d-1)}{\beta}} \log^{-1}(1/\varepsilon).$$

The proof of this result is based on a lemma giving a lower bound on the minimal number of bits ℓ necessary for a binary encoder-decoder pair to achieve an error less than $\varepsilon > 0$ in approximating $\mathcal{HF} := \mathcal{HF}_{\beta,d,B}$ in L^p norm. Formally, given an integer $\ell > 0$, a binary encoder $E^\ell : \mathcal{HF} \to \{0,1\}^\ell$ and given a decoder $D^\ell : \{0,1\}^\ell \to \mathcal{HF}$, one can measure an approximation error

$$\sup_{f \in \mathcal{HF}} \|f - D^\ell(E^\ell(f))\|_{L^p},$$

which quantifies the loss of information due to the encoding E^ℓ. Clearly, for an optimal

choice of encoder, one can reduce this loss of information by increasing ℓ. In particular, for $\varepsilon > 0$, it is possible to estimate

$$\ell_\varepsilon = \min\left\{\ell > 0 \ : \inf_{E^\ell, D^\ell} \sup_{f\in\mathcal{HF}} \|f - D^\ell(E^\ell(f))\|_{L^p} \leq \varepsilon\right\},$$

with the convention that $\ell_\varepsilon = \infty$ if the above set is empty. The authors show in their Lemma B.3 that for ε small enough (smaller than some $\varepsilon_0 > 0$), it holds that

$$\ell_\varepsilon \geq c\varepsilon^{-\frac{p(d-1)}{\beta}} \tag{5.C.1}$$

for some constant $c > 0$ depending only on d, p, β and B. In other words, one can not achieve a loss of information smaller than ε by encoding functions in $\mathcal{HF}$ over less than $c\varepsilon^{-\frac{p(d-1)}{\beta}}$ bits.

The rest of the proof consists in showing that for an integer $K > 0$, given a neural network architecture $\mathcal{A}$ with W weight that can approximate $\mathcal{HF}$ in L^p norm with error less than $\varepsilon > 0$, one can encode exactly (without loss of information, and for a given activation function) any function in $H^K_{\mathcal{A}}$ over a string of $\ell = c_1 W(K + \lceil \log_2 W\rceil)$ bits. This generates a natural encoder-decoder system where any function $f \in \mathcal{HF}$ is encoded as the bit string of length ℓ associated to $g_f \in H^K_{\mathcal{A}}$ chosen to approximate f. It remains to observe that if we fix K, this automatically yields a lower bound on ℓ using inequality (5.C.1), and thus on W by expressing W through ℓ and K.

Remark. The authors in [111] study the neural network approximation in a setting slightly different from ours, since they focus on the approximation by quantized neural networks. This partly explains why their proof strategy differs from ours. However, it is worth pointing out that the proof of their lower bound on the minimal number of bits required to accurately encode a function in $\mathcal{HF}$ relies on a lower bound of the packing number of $\mathcal{HF}$, just like the lower bound of the packing number of the set to approximate is key in our proof strategy. An interesting question for the future would be to see whether our general lower bound (Theorem 9) yields lower bounds of the same order as those in [111] for quantized neural networks.

5.D Hölder balls

5.D.1 Proof of Lemma 36

Though not necessarily stated this way, many arguments below are classical (see, e.g., Theorem 3.2 by [55] with a similar construction for lower bounds in nonparametric regression).

Let $N \in \mathbb{N}^*$. For $\mathbf{m} = (m_1, \ldots, m_d) \in \{0, \ldots, N-1\}^d$, we let $x_{\mathbf{m}} := \frac{1}{N}(m_1 + 1/2, \ldots, m_d + 1/2)$ and we denote by $C_{\mathbf{m}}$ the cube of side-length $\frac{1}{N}$ centered at $x_{\mathbf{m}}$, with sides parallel to the axes. We see that the N^d cubes $C_{\mathbf{m}}$ decompose the cube $[0,1]^d$ in smaller parts which, up to negligible sets which will not be problematic, form a partition of $[0,1]^d$. We will use this decomposition to construct a packing of $F_{s,d}$. Denoting $\|\cdot\|$ the

sup norm in $\mathbb{R}^d$, we define the C^∞ test function $\phi : \mathbb{R}^d \to \mathbb{R}$ by:

$$\phi(x) = \exp\left(-\frac{\|x\|^2}{1-\|x\|^2}\right),$$

for any $x \in \mathbb{R}^d$ such that $\|x\| < 1$, and $\phi(x) = 0$ otherwise. Recalling that $n \in \mathbb{N}$ and $\alpha \in (0,1]$ are such that $s = n + \alpha$, and since all the high-order partial derivatives of ϕ are uniformly bounded on $[0,1]^d$, $\|\phi\|_{\mathcal{C}^{n,\alpha}}$ is thus finite and is nonzero.

Let $c_s = \frac{1}{2}(2N)^{-s}\|\phi\|^{-1}_{\mathcal{C}^{n,\alpha}}$ and consider, for any tensor of signs $\sigma = (\sigma_{\mathbf{m}})_{\mathbf{m}\in\{0,\cdots,N-1\}^d} \in \{-1,1\}^{N^d}$, the function f_σ defined as follows:

$$f_\sigma(x) = c_s \sum_{\mathbf{m}\in\{0,\ldots,N-1\}^d} \sigma_{\mathbf{m}}\phi\left(2N(x - x_{\mathbf{m}})\right),$$

for all $x \in [0,1]^d$. There are 2^{N^d} different functions f_σ.

Let us prove that, for all $\sigma \in \{-1,1\}^{N^d}$, $f_\sigma \in F_{s,d}$. To do so, we study the constituents of $\|f_\sigma\|_{\mathcal{C}^{n,\alpha}}$ separately and show that they are all bounded by 1. For $\mathbf{m} \in \{0,\cdots,N-1\}^d$, we define the function $g_{\mathbf{m}}(x) = c_s\sigma_{\mathbf{m}}\phi\left(2N(x - x_{\mathbf{m}})\right)$. Note that because ϕ vanishes outside $(-1,1)^d$, we have that $g_{\mathbf{m}}$ vanishes everywhere outside the interior of $C_{\mathbf{m}}$, and the same holds for $D^{\mathbf{n}}g_{\mathbf{m}}$ for all $\mathbf{n} \in \mathbb{N}^d$ such that $|\mathbf{n}| \le n$. For any such $\mathbf{n}$, we have

$$\|D^{\mathbf{n}}g_{\mathbf{m}}\|_\infty = c_s(2N)^{|\mathbf{n}|}\|D^{\mathbf{n}}\phi\|_\infty \le c_s(2N)^s\|\phi\|_{\mathcal{C}^{n,\alpha}} \le \frac{1}{2}.$$

Therefore,

$$\max_{\mathbf{n}:|\mathbf{n}|\le n} \|D^{\mathbf{n}}f_\sigma\|_\infty \le 1.$$

Now for any $\mathbf{n} \in \mathbb{N}^d$ such that $|\mathbf{n}| = n$, any $x, y \in [0,1]^d$, we have

$$\frac{|D^{\mathbf{n}}f_\sigma(x) - D^{\mathbf{n}}f_\sigma(y)|}{\|x-y\|_2^\alpha} = \frac{|D^{\mathbf{n}}g_{\mathbf{m}}(x) - D^{\mathbf{n}}g_{\mathbf{m}'}(y)|}{\|x-y\|_2^\alpha},$$

where $x \in C_{\mathbf{m}}$ and $y \in C_{\mathbf{m}'}$ for some multi-indexes $\mathbf{m}$ and $\mathbf{m}'$. We have to distinguish between the cases $\mathbf{m} = \mathbf{m}'$ and $\mathbf{m} \neq \mathbf{m}'$. In the former case, we have

$$\begin{aligned}\frac{|D^{\mathbf{n}}f_\sigma(x) - D^{\mathbf{n}}f_\sigma(y)|}{\|x-y\|_2^\alpha} &= c_s(2N)^{n+\alpha}\frac{|D^{\mathbf{n}}\phi(2N(x-x_{\mathbf{m}})) - D^{\mathbf{n}}\phi(2N(y-x_{\mathbf{m}}))|}{\|2N(x-x_{\mathbf{m}}) - 2N(y-x_{\mathbf{m}})\|_2^\alpha}\\ &= c_s(2N)^s\frac{|D^{\mathbf{n}}\phi(x') - D^{\mathbf{n}}\phi(y')|}{\|x'-y'\|_2^\alpha}\\ &\le c_s(2N)^s\|\phi\|_{\mathcal{C}^{n,\alpha}} = \frac{1}{2},\end{aligned}$$

where at the second line, we used the changes of variables $x' = 2N(x - x_{\mathbf{m}})$ and $y' = 2N(y - x_{\mathbf{m}})$. In the case $\mathbf{m} = \mathbf{m}'$ (x and y belong to the same cube), we thus have

$$\frac{|D^{\mathbf{n}}f_\sigma(x) - D^{\mathbf{n}}f_\sigma(y)|}{\|x-y\|_2^\alpha} \le 1.$$

In the case $\mathbf{m} \neq \mathbf{m}'$, observe that we have

$$|D^{\mathbf{n}} g_{\mathbf{m}}(x) - D^{\mathbf{n}} g_{\mathbf{m}'}(y)| \leq 2 \max\{|D^{\mathbf{n}} g_{\mathbf{m}}(x)|, |D^{\mathbf{n}} g_{\mathbf{m}'}(y)|\}. \tag{5.D.1}$$

Besides, recall that $D^{\mathbf{n}} g_{\mathbf{m}}$ and $D^{\mathbf{n}} g_{\mathbf{m}'}$ both vanish outside of the interiors of $C_{\mathbf{m}}$ and $C_{\mathbf{m}'}$ respectively. We can thus rewrite (5.D.1) as

$$\begin{aligned}
|D^{\mathbf{n}} g_{\mathbf{m}}(x) - D^{\mathbf{n}} g_{\mathbf{m}'}(y)| &\leq 2 \max\{|D^{\mathbf{n}} g_{\mathbf{m}}(x) - D^{\mathbf{n}} g_{\mathbf{m}}(y)|, |D^{\mathbf{n}} g_{\mathbf{m}'}(x) - D^{\mathbf{n}} g_{\mathbf{m}'}(y)|\} \\
&\leq 2c_s (2N)^n \max\{|D^{\mathbf{n}} \phi(2N(x - x_{\mathbf{m}})) - D^{\mathbf{n}} \phi(2N(y - x_{\mathbf{m}}))|, \\
&\qquad |D^{\mathbf{n}} \phi(2N(y - x_{\mathbf{m}'})) - D^{\mathbf{n}} \phi(2N(y - y_{\mathbf{m}'}))|\}.
\end{aligned}$$

This entails

$$\begin{aligned}
\frac{|D^{\mathbf{n}} f_\sigma(x) - D^{\mathbf{n}} f_\sigma(y)|}{\|x - y\|_2^\alpha} &\leq c_s 2(2N)^s \max\left\{ \frac{|D^{\mathbf{n}} \phi(x') - D^{\mathbf{n}} \phi(y')|}{\|x' - y'\|_2^\alpha}, \frac{|D^{\mathbf{n}} \phi(x'') - D^{\mathbf{n}} \phi(y'')|}{\|x'' - y''\|_2^\alpha} \right\} \\
&\leq c_s 2(2N)^s \|\phi\|_{\mathcal{C}^{n,\alpha}} = 1,
\end{aligned}$$

where $x' = 2N(x - x_{\mathbf{m}})$ and $y' = 2N(y - x_{\mathbf{m}})$, and $x'' = 2N(x - x_{\mathbf{m}'})$ and $y'' = 2N(y - x_{\mathbf{m}'})$.

Summarizing, we showed that for all $\sigma \in \{-1, 1\}^{N^d}$

$$\max_{\mathbf{n}:|\mathbf{n}| \leq n} \|D^{\mathbf{n}} f_\sigma\|_\infty \leq 1 \qquad \text{and} \qquad \max_{\mathbf{n}:|\mathbf{n}|=n} \sup_{x \neq y} \frac{|D^{\mathbf{n}} f_\sigma(x) - D^{\mathbf{n}} f_\sigma(y)|}{\|x - y\|_2^\alpha} \leq 1.$$

We conclude that for all $\sigma \in \{-1, 1\}^{N^d}$

$$\|f_\sigma\|_{\mathcal{C}^{n,\alpha}} \leq 1,$$

and therefore $\{f_\sigma : \sigma \in \{-1, 1\}^{N^d}\} \subset F_{s,d}$.

Let us now evaluate the distance between distinct elements of $\{f_\sigma : \sigma \in \{-1, 1\}^{N^d}\}$. Let σ^1, $\sigma^2 \in \{-1, 1\}^{N^d}$, with $\sigma^1 \neq \sigma^2$, and let $\mathbf{m} \in \{0, \dots, N-1\}^d$ be such that $\sigma^1_{\mathbf{m}} = -\sigma^2_{\mathbf{m}}$. Let us estimate Δ_p the $L^p(\lambda)$ discrepancy between f_{σ^1} and f_{σ^2} on the cube $C_{\mathbf{m}}$, that is

$$\begin{aligned}
\Delta_p^p &= \int_{C_{\mathbf{m}}} |f_{\sigma^1}(x) - f_{\sigma^2}(x)|^p \mathrm{d}x \\
&= 2^p c_s^p \int_{C_{\mathbf{m}}} |\phi\left(2N(x - x_{\mathbf{m}})\right)|^p \mathrm{d}x \\
&= 2^p c_s^p (2N)^{-d} \|\phi\|_{L^p(\lambda)}^p.
\end{aligned}$$

It remains to find a subset among the functions f_σ such that any two functions of this set differ on a significant number of cubes $C_{\mathbf{m}}$. According to the Varshamov-Gilbert Lemma [155], there exists $\Gamma \subset \{-1, 1\}^{N^d}$ with cardinal at least $\exp(N^d/8)$ such that for any $\sigma^1, \sigma^2 \in \Gamma$, such that $\sigma^1 \neq \sigma^2$, σ^1 and σ^2 differ on at least one fourth of their coordinates; i.e., $\sum_{k=1}^{N^d} \mathbb{1}_{\sigma^1_k \neq \sigma^2_k} \geq \frac{N^d}{4}$. We thus fix such a set $\Gamma \subset \{-1, 1\}^{N^d}$. For any σ^1, $\sigma^2 \in \Gamma$, with

$\sigma^1 \neq \sigma^2$,

$$\begin{aligned}\|f_{\sigma^1} - f_{\sigma^2}\|_{L^p(\lambda)}^p &= \sum_{\mathbf{m}:\sigma^1_{\mathbf{m}} \neq \sigma^2_{\mathbf{m}}} \int_{C_{\mathbf{m}}} |f_{\sigma^1}(x) - f_{\sigma^2}(x)|^p \mathrm{d}x \\ &\geq \frac{N^d}{4}\Delta_p^p = \frac{2^{p-d}c_s^p}{4}\|\phi\|_{L^p(\lambda)}^p.\end{aligned}$$

Finally, recalling the definition of c_s, we have for any σ^1, $\sigma^2 \in \Gamma$, with $\sigma^1 \neq \sigma^2$,

$$\|f_{\sigma^1} - f_{\sigma^2}\|_{L^p(\lambda)} \geq 2^{1-\frac{d+2}{p}} \frac{1}{2}(2N)^{-s}\|\phi\|_{\mathcal{C}^{n,\alpha}}^{-1}\|\phi\|_{L^p(\lambda)} = cN^{-s},$$

where $c = 2^{-s-\frac{d+2}{p}} \frac{\|\phi\|_{L^p(\lambda)}}{\|\phi\|_{\mathcal{C}^{n,\alpha}}}$.

It follows that $\{f_\sigma \ : \ \sigma \in \Gamma\}$ is a cN^{-s}-packing of $F_{s,d}$. Given the lower bound on the size of Γ, this implies

$$M\left(cN^{-s}, F_{s,d}, \|\cdot\|_{L^p(\lambda)}\right) \geq \exp(N^d/8),$$

for all $N \in \mathbb{N}^*$.

Set $\varepsilon_0 = c$ and $c_0 = 2^{-d}c^{\frac{d}{s}}/8$. Consider $\varepsilon > 0$, with $\varepsilon \leq \varepsilon_0$. To conclude the proof, we need to show that (5.3.3) holds for ε. To do so, we consider N: the smallest integer such that $cN^{-s} \geq \varepsilon \geq c(2N)^{-s}$. This $N \in \mathbb{N}^*$ exists because $0 < \varepsilon \leq \varepsilon_0 = c$ and $s > 0$. On one side, we have

$$M\left(\varepsilon, F_{s,d}, \|\cdot\|_{L^p(\lambda)}\right) \geq M\left(cN^{-s}, F_{s,d}, \|\cdot\|_{L^p(\lambda)}\right),$$

and on the other side, since $2N \geq c^{\frac{1}{s}}\varepsilon^{-\frac{1}{s}}$,

$$\exp(N^d/8) \geq \exp(2^{-d}c^{\frac{d}{s}}\varepsilon^{-\frac{d}{s}}/8) = \exp(c_0\varepsilon^{-\frac{d}{s}}).$$

Combining the last three inequalities, we finally obtain

$$\log M\left(\varepsilon, F_{s,d}, \|\cdot\|_{L^p(\lambda)}\right) \geq c_0\varepsilon^{-d/s},$$

for all $0 < \varepsilon \leq \varepsilon_0$.

5.E Monotonic functions

This section contains the proofs of the results stated in Section 5.4. More precisely, in Section 5.E.1 we provide the proof of Proposition 5.4.3 and in Section 5.E.2 we provide the proof of Proposition 5.4.1.

5.E.1 Proof of Proposition 5.4.3

The section contains two sub-sections. In the first sub-section, we provide a proposition on the representation of piecewise-constant functions with Heaviside neural-networks.

Section 5.E.1.2 contains the main part of the proof of Proposition 5.4.3.

5.E.1.1 Representing piecewise-constant functions with Heaviside neural networks

We first describe a neural network architecture which, with the Heaviside activation function, is able to represent functions that are piecewise-constant on cubes.

Proposition 5.E.1. Let $d \in \mathbb{N}^*$, $M \in \mathbb{N}^*$. There exists an architecture $\mathcal{A}$ with two-hidden layers, $2(d+1)^2 M$ weights and the Heaviside activation function, such that for any $(\alpha_i)_{1\leq i\leq M} \in \mathbb{R}^M$, any collection $(\mathcal{C}_i)_{1\leq i\leq M}$ of mutually disjoint hypercubes of $\mathbb{R}^d$ the function $\tilde{f}\colon \mathbb{R}^d \to [0,1]$ defined by

$$\forall x \in \mathbb{R}^d, \quad \tilde{f}(x) = \sum_{i=1}^{M} \alpha_i \mathbb{1}_{\mathcal{C}_i}(x)$$

satisfies $\tilde{f} \in H_{\mathcal{A}}$.

Proof. Define $\sigma\colon \mathbb{R} \to \mathbb{R}$ by $\sigma(x) = \mathbb{1}_{x\geq 0}$ for all $x \in \mathbb{R}$.

Let $i \in \{1, \dots, M\}$. The cube $\mathcal{C}_i$ has $2d$ faces. These faces are supported by hyperplanes whose equations are of the form $\langle \mathbf{w}, x\rangle + b = 0$, with $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. We allow the faces to belong to the cube or not. To distinguish them, we denote $J_i \in \{1, \dots, 2d\}$ the number of faces that belong to $\mathcal{C}_i$. We index the J_i faces that belong to the cube from 1 to J_i, and the other faces from $J_i + 1$ to $2d$. Thus, for all $i \in \{1, \dots, M\}$ and all $j \in \{1, \dots, 2d\}$, there exist $\mathbf{w}_j^i \in \mathbb{R}^d, b_j^i \in \mathbb{R}$ such that

$$\mathcal{C}_i = \bigcap_{j=1}^{J_i} \{x \in \mathbb{R}^d\colon \left\langle \mathbf{w}_j^i, x\right\rangle + b_j^i \geq 0\} \cap \bigcap_{j=J_i+1}^{2d} \{x \in \mathbb{R}^d\colon \left\langle \mathbf{w}_j^i, x\right\rangle + b_j^i > 0\}\,.$$

We rewrite:

$$\mathcal{C}_i = \left\{ x \in \mathbb{R}^d\colon \sum_{j=1}^{J_i} \mathbb{1}_{\{\langle \mathbf{w}_j^i, x\rangle + b_j^i \geq 0\}} + \sum_{j=J_i+1}^{2d} \mathbb{1}_{\{\langle \mathbf{w}_j^i, x\rangle + b_j^i > 0\}} \;\geq\; 2d \right\}. \tag{5.E.1}$$

Denoting for all $i \in \{1, \dots, M\}$ and all $j \in \{1, \dots, 2d\}$ and for all $x \in \mathbb{R}^d$,

$$p_j^i(x) = \begin{cases} \sigma\left(\left\langle \mathbf{w}_j^i, x\right\rangle + b_j^i\right) & \text{if } j \leq J_i \\ 1 - \sigma\left(-\left\langle \mathbf{w}_j^i, x\right\rangle - b_j^i\right) & \text{otherwise,} \end{cases}$$

we have, see Figure 5.1 and (5.E.1), for all $x \in \mathbb{R}^d$

$$\begin{aligned} \mathbb{1}_{\mathcal{C}_i}(x) &= \begin{cases} 1 & \text{if } \sum_{j=1}^{2d} p_j^i(x) \geq 2d, \\ 0 & \text{otherwise,} \end{cases} \\ &= \sigma\left(\sum_{j=1}^{2d} p_j^i(x) - 2d\right). \end{aligned}$$

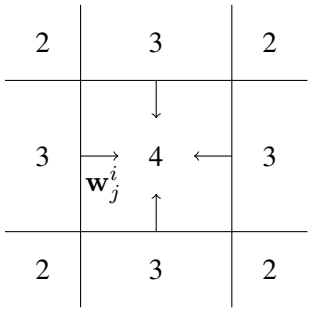

Figure 5.1 – Values of the sum of the perceptrons $p_j^i(x)$ around a hypercube $\mathcal{C}_i$ in dimension 2.

Since the hypercubes are mutually disjoints, for all $x \in [0,1]^d$, we have

$$\begin{aligned}\tilde{f}(x) &= \sum_{i=1}^{M} \alpha_i \sigma \left(\sum_{j=1}^{J_i} \sigma\left(\left\langle \mathbf{w}_j^i, x \right\rangle + b_j^i\right) + \sum_{j=J_i+1}^{2d} \left(1 - \sigma\left(-\left\langle \mathbf{w}_j^i, x \right\rangle - b_j^i\right)\right) \; - \; 2d \right) \\ &= \sum_{i=1}^{M} \alpha_i \sigma \left(\sum_{j=1}^{2d} \varepsilon_j^i \sigma\left(\left\langle \tilde{\mathbf{w}}_j^i, x \right\rangle + \tilde{b}_j^i\right) - J_i \right), \end{aligned} \tag{5.E.2}$$

where

$$\varepsilon_j^i = \begin{cases} +1 & \text{if } j \leq J_i \\ -1 & \text{otherwise,} \end{cases} \qquad \tilde{\mathbf{w}}_j^i = \begin{cases} \mathbf{w}_j^i & \text{if } j \leq J_i \\ -\mathbf{w}_j^i & \text{otherwise,} \end{cases} \qquad \tilde{b}_j^i = \begin{cases} b_j^i & \text{if } j \leq J_i \\ -b_j^i & \text{otherwise.} \end{cases}$$

Equation (5.E.2) is the action of the Heaviside neural network with two hidden layers whose architecture is on Figure 5.2.

It remains to count the weights and biases of $\tilde{f}$:

— the architecture has M edges going to the output layer, due to the α_i;

— it has M biases associated to the neurons of the second hidden layer (they correspond to the terms $-J_i$);

— between the second and the first hidden layer, the architecture has $M \times 2d$ edges (corresponding to the $\varepsilon_{i,j}$);

— it has $M \times 2d$ biases associated to the neurons of the first hidden layer (the $\tilde{b}_j^i$);

— it has $M \times 2d \times d$ edges between the first hidden layer and the entry (the $\tilde{\mathbf{w}}_j^i$).

Thus there are $2M + 2M \times 2d + M \times 2d \times d = 2(d^2 + 2d + 1)M = 2(d+1)^2 M$ weights and biases in total. □

5.E.1.2 Main developments of the proof of Proposition 5.4.3

Let $N \in \mathbb{N}^*$ and $f \in \mathcal{M}^d$. In this section, we partition $[0,1)^d$ into cubes whose sizes depend on the maximal variation of f. Then we use this partition to construct a piecewise

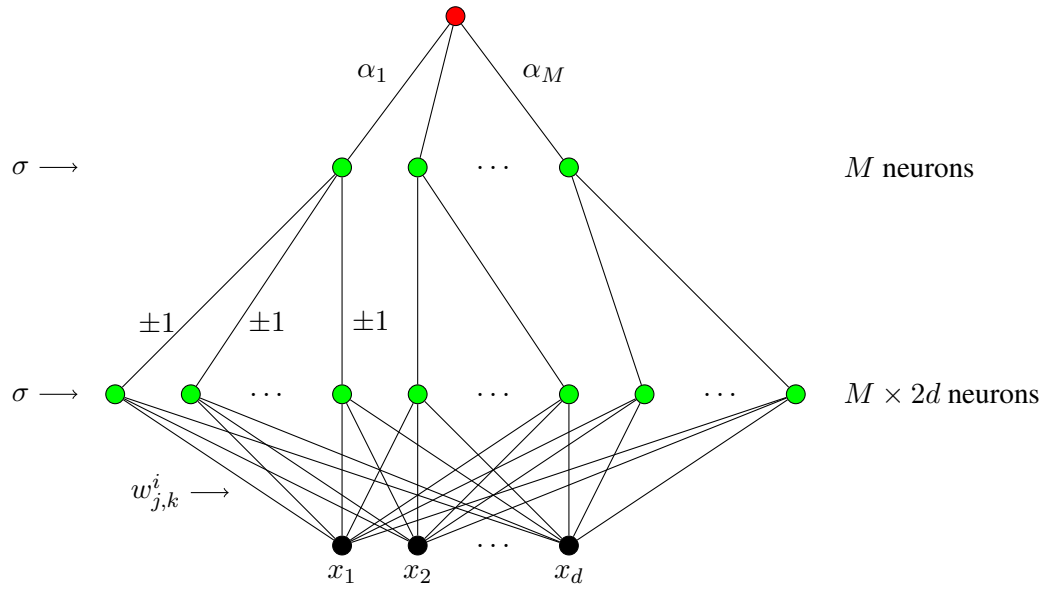

Figure 5.2 – The function $\tilde{f}$ represented as a neural network.

constant approximation $\tilde{f}$ of f; we will bound from above the $L^p(\lambda)$ approximation error $\|f - \tilde{f}\|_{L^p(\lambda)}$ by a function of N. This part is a direct reinterpretation of the proof of Proposition 3.1 in [41]. We then apply Proposition 5.E.1 to $\tilde{f}$ and obtain the announced result.

We first define some notation that will be used in the rest of the section, then we explain the algorithm used to divide $[0,1)^d$ into cubes. We fix the constant $K > 1$ the following way:

$$K := \begin{cases} 2^d & \text{if } p = 1, \\ 2^\beta & \text{otherwise, where } \beta = \frac{1}{2}(d - 1 + \frac{1}{p-1}). \end{cases}$$

We also define an integer l that corresponds to the number of cube decompositions:

$$l := \left\lceil \frac{N \log 2}{\log K} \right\rceil = \begin{cases} \left\lceil \frac{N}{d} \right\rceil & \text{if } p = 1, \\ \left\lceil \frac{N}{\beta} \right\rceil & \text{otherwise .} \end{cases} \tag{5.E.3}$$

It is worth noting that this implies $K^{-l} \leq 2^{-N} < K^{-l+1}$.

Now we partition $[0,1)^d$ into dyadic cubes of the form $[a_1, b_1) \times \cdots \times [a_d, b_d)$. If C is such a cube, we use the following convenient notation:

$$\underline{C} := (a_1, \ldots, a_d) \in \mathbb{R}^d, \qquad \overline{C} := (b_1, \ldots, b_d) \in \mathbb{R}^d,$$

to refer to the smallest and largest vertices of C. The cube decompositions process reads as follow:

— First we partition $[0,1)^d$ into 2^{Nd} cubes of side-length 2^{-N}. We denote by S_0 the set

of these cubes C such that $f(\overline{C}) - f(\underline{C}) \leq K2^{-N}$ and by R_0 the set of the remaining cubes.

— For $1 \leq i < l$, we partition each cube in the set R_{i-1} (the remaining cubes at the step $i-1$) into 2^d cubes of equal size, and we denote by S_i the set of obtained cubes C of side-length $2^{-(i+N)}$ such that

$$f(\overline{C}) - f(\underline{C}) \leq K^{i+1}2^{-N}. \tag{5.E.4}$$

Again, the set of remaining cubes is denoted by R_i.

— Lastly, we partition each cube in the set R_{l-1} into 2^d cubes of equal size, and we denote by S_l the set of obtained cubes of side-length $2^{-(l+N)}$.

Once the algorithm is done, each point in $[0,1)^d$ clearly belongs to one single cube of $\cup_{i=0}^{l} S_i$. For $i \in \{0,\dots,l\}$, we let $\tilde{S}_i = \cup_{C\in S_i} C$.

We now define the piecewise constant approximation of f by

$$\forall x \in [0,1]^d, \quad \tilde{f}(x) = \sum_{C \in \bigcup_{0\leq i\leq l} S_i} f(\underline{C})\mathbb{1}_{x\in C},$$

where $\mathbb{1}_{x\in C}$ denotes the indicator function of the cube C. We do not make the dependence explicit, but $\tilde{f}$ depends on the parameters N, d and p. The number of cubes over which $\tilde{f}$ is constant is $\sum_{i=0}^{l} |S_i|$. This quantity is key when constructing the neural network according to Proposition 5.E.1; in the next lemma, we bound from above $|S_i|$ for all $i = 0,\dots,l$. Then, we will estimate the error $\|f - \tilde{f}\|_{L^p(\lambda)}$.

Lemma 41. With the above notation:

$$\forall i \in \{0,\dots,l\}, \quad |S_i| \leq dK^{-i}2^{i(d-1)+Nd+1}$$

Moreover,

$$\lambda(\tilde{S}_i) \leq \begin{cases} 1 & \text{if } i = 0, \\ 2d(2K)^{-i} & \text{, otherwise.} \end{cases} \tag{5.E.5}$$

Proof. By construction, we have

$$\forall i \in \{1,\dots,l\}, \quad |S_i| + |R_i| = 2^d|R_{i-1}|,$$

since the set $S_i \cup R_i$ contains all the cubes of side-length $2^{-(i+N)}$, that have been constructed from the cubes of R_{i-1}. In particular,

$$\forall i \in \{1,\dots,l\}, \quad |S_i| \leq 2^d|R_{i-1}|. \tag{5.E.6}$$

It remains to bound $|R_{i-1}|$ from above for $i \geq 1$. Define $V := \{\underline{C} \colon C \in R_{i-1}\}$ the set of the smallest vertices of the cubes in R_{i-1}. We consider the classes of these vertices under the "laying on the same extended diagonal" equivalence relation. Since the cubes have side-length $2^{-(i-1+N)}$, there are less than $d2^{(i-1+N)(d-1)}$ equivalence classes. According to the pigeonhole principle, the largest class has at least $\left\lceil \frac{|V|}{d2^{(i-1+N)(d-1)}} \right\rceil$ elements; let us

refer to this class as $\mathcal{D}$. Let $(C_j)_{1\leq j\leq J}$ be the set of cubes in R_{i-1} having a point in $\mathcal{D}$ as lowest vertex. Since f is non-decreasing and according to (5.E.4), we have:

$$1 \geq f(1,\dots,1) - f(0,\dots,0) \geq \sum_{j=1}^{J} f(\overline{C_j}) - f(\underline{C_j}) \geq JK^i 2^{-N}$$
$$\geq \frac{|V|}{d2^{(i-1+N)(d-1)}} K^i 2^{-N} = \frac{|R_{i-1}|}{d2^{(i-1+N)(d-1)}} K^i 2^{-N}.$$

Thus

$$|R_{i-1}| \leq d2^{i(d-1)+Nd+1-d} K^{-i}.$$

The first statement of Lemma 41 follows from (5.E.6).

For $i = 0$, $\lambda(\tilde{S}_0) \leq 1$. For $1 \leq i \leq l$, using the first statement of this lemma, we bound from above the measure of $\tilde{S}_i$:

$$\begin{aligned}\lambda(\tilde{S}_i) = \left(2^{-(i+N)}\right)^d |S_i| &\leq dK^{-i}2^{i(d-1)+Nd+1}2^{-d(i+N)},\\ &= 2d(2K)^{-i}.\end{aligned}$$

□

To show that $\tilde{f}$ is close to f in $L^p(\lambda)$ norm, let us use the fact that $(\tilde{S}_i)_{0\leq i\leq l}$ is a partition of $[0,1)^d$ and decompose the error in three parts:

$$\|f-\tilde{f}\|^p_{L^p(\lambda)} = \int_{\tilde{S}_0} |f(x)-\tilde{f}(x)|^p \mathrm{d}x + \sum_{i=1}^{l-1} \int_{\tilde{S}_i} |f(x)-\tilde{f}(x)|^p \mathrm{d}x + \int_{\tilde{S}_l} |f(x)-\tilde{f}(x)|^p \mathrm{d}x.$$

In the next lemma, we control each term of the above sum to bound from above $\|f-\tilde{f}\|_{L^p(\lambda)}$ by a function of N that is independent of f and tends to 0 when N tends to $+\infty$.

Lemma 42. For any $1 \leq p < +\infty$, there exists a constant $c_{d,p} > 0$ depending only on d and p such that for all $N \in \mathbb{N}^*$

$$\|f-\tilde{f}\|_{L^p(\lambda)} \leq c_{d,p} \begin{cases} 2^{-N} & \text{if } p(d-1) < d, \\ 2^{-N\frac{(1+1/\beta)}{p}} & \text{if } p(d-1) > d, \\ N^{\frac{1}{p}} 2^{-N} & \text{if } p(d-1) = d, \end{cases} \tag{5.E.7}$$

where $\tilde{f}$ is the function constructed for the parameters N, d and p.

Proof. For $0 \leq i < l$, on any cube $C \in S_i$, we have

$$\forall x \in C, \quad |f(x) - \tilde{f}(x)| = |f(x) - f(\underline{C})| \leq f(\overline{C}) - f(\underline{C}) \leq K^{i+1}2^{-N}, \tag{5.E.8}$$

since f is non-decreasing, and by definition of $\tilde{f}$ and S_i.

— Using the fact that $\lambda(\tilde{S}_0) \leq 1$ and by (5.E.8):

$$\int_{\tilde{S}_0} |f(x) - \tilde{f}(x)|^p \mathrm{d}x \leq (2^{-N}K)^p. \tag{5.E.9}$$

— Using (5.E.5) and (5.E.8), we get for all $i \in \{1, \ldots, l-1\}$

$$\int_{\tilde{S}_i} |f(x) - \tilde{f}(x)|^p \mathrm{d}x \leq (K^{i+1}2^{-N})^p 2d(2K)^{-i}. \tag{5.E.10}$$

— On any $C \in S_l$, we have, for all $x \in C$, $|f(x) - \tilde{f}(x)| \leq |f(x) - f(\underline{C})| \leq 1$, and we get, using (5.E.5):

$$\int_{\tilde{S}_l} |f(x) - \tilde{f}(x)|^p \mathrm{d}x \leq 2d(2K)^{-l}. \tag{5.E.11}$$

Combining (5.E.9), (5.E.10) and (5.E.11) we get:

$$\begin{aligned} \|f - \tilde{f}\|^p_{L^p(\lambda)} &\leq (2^{-N}K)^p + \sum_{i=1}^{l-1} (K^{i+1}2^{-N})^p 2d(2K)^{-i} + 2d(2K)^{-l} \\ &\leq (2^{-N}K)^p + 2^{1-Np}K^p d \sum_{i=1}^{l-1} \left(\frac{K^{p-1}}{2}\right)^i + 2d(2K)^{-l}. \end{aligned} \tag{5.E.12}$$

It remains to bound the right-hand side of (5.E.12), depending on the value of p and d. Note that the behavior of this term depends on whether $\frac{K^{p-1}}{2}$ is larger or smaller than 1.

— Suppose that $p(d-1) < d$. In this case, we can have $p = 1$ or $p > 1$. If $p = 1$, we have $\frac{K^{p-1}}{2} = \frac{1}{2} < 1$ and $\frac{1}{2K} < K^{-p}$. If $p > 1$, we have:

$$p(d-1) < d \iff dp - p - d + 1 < 1 \iff d - 1 < \frac{1}{p-1}.$$

Thus, β being the arithmetic mean of $d-1$ and $\frac{1}{p-1}$, we have $d - 1 < \beta < \frac{1}{p-1}$. Then $K = 2^\beta < 2^{1/(p-1)}$ and hence $\frac{K^{p-1}}{2} < 1$ and $\frac{1}{2K} < K^{-p}$. Therefore, both for $p = 1$ and $p > 1$,

$$\sum_{i=1}^{l-1} \left(\frac{K^{p-1}}{2}\right)^i \leq \frac{K^{p-1}}{2 - K^{p-1}} \quad \text{and} \quad (2K)^{-l} \leq K^{-pl}.$$

Since $K^{-l} \leq 2^{-N}$, this leads to

$$\begin{aligned} \|f - \tilde{f}\|^p_{L^p(\lambda)} &\leq (2^{-N}K)^p + 2^{1-Np}K^p d \frac{K^{p-1}}{2 - K^{p-1}} + 2dK^{-pl} \\ &\leq \left(K^p + 2K^p d \frac{K^{p-1}}{2 - K^{p-1}} + 2d\right) 2^{-Np}. \end{aligned}$$

We thus have, setting $c_1 := \left(K^p + 2K^p d\frac{K^{p-1}}{2-K^{p-1}} + 2d\right)^{\frac{1}{p}}$,

$$\|f - \tilde{f}\|_{L^p(\lambda)} \leq c_1 2^{-N}.$$

Notice c_1 only depends on d and p.

— Suppose that $p(d-1) > d$. We have $p > 1$ and $d - 1 > \beta > \frac{1}{p-1}$. Then $K = 2^\beta > 2^{1/(p-1)}$ and hence $\frac{K^{p-1}}{2} > 1$, which entails using (5.E.12)

$$\begin{aligned}\|f - \tilde{f}\|^p_{L^p(\lambda)} &\leq (2^{-N}K)^p + 2^{1-Np}K^p d\frac{(K^{p-1}/2)^l}{K^{p-1}/2-1} + 2d(2K)^{-l} \\ &\leq 2^{-Np}K^p + 2^{-Np}K^{pl}\frac{2K^p d}{K^{p-1}/2-1}(2K)^{-l} + 2d(2K)^{-l}.\end{aligned}$$

Since $p > 1 + \frac{1}{\beta}$, we have $2^{-Np} \leq 2^{-N(1+\frac{1}{\beta})}$. Also, since $K = 2^\beta$, $(2K)^{-l} = 2^{-l(\beta+1)}$, and since $l \geq \frac{N\log(2)}{\log(K)} = \frac{N}{\beta}$, we have $(2K)^{-l} \leq 2^{-\frac{N}{\beta}(\beta+1)} = 2^{-N(1+\frac{1}{\beta})}$. Finally, since $2^{-N}K^l < K$,

$$\|f - \tilde{f}\|^p_{L^p(\lambda)} \leq \left(K^p + K^p\frac{2K^p d}{K^{p-1}/2-1} + 2d\right)2^{-N(1+1/\beta)}$$

We thus have, setting $c_2 := \left(K^p + \frac{2K^{2p}d}{K^{p-1}/2-1} + 2d\right)^{\frac{1}{p}}$,

$$\|f - \tilde{f}\|_{L^p(\lambda)} \leq c_2 2^{-\frac{N(1+1/\beta)}{p}}.$$

Notice c_2 only depends on d and p.

— Suppose that $p(d-1) = d$. It implies $p > 1$ and $p - 1 = \frac{1}{d-1}$, then $\beta = d - 1$. We thus have $K^{p-1} = 2^{(d-1)(p-1)} = 2$. Therefore, (5.E.12) becomes

$$\|f - \tilde{f}\|^p_{L^p(\lambda)} \leq 2^{-Np}K^p + 2^{-Np}2K^p d(l-1) + 2d(K^p)^{-l}.$$

On the one hand, we have $K^{-l} \leq 2^{-N}$. On the other, we have $2^{-N} < K^{-l+1}$, so $l - 1 < N\frac{\log 2}{\log K} = \frac{N}{d-1}$. Putting it all together, we get

$$\begin{aligned}\|f - \tilde{f}\|^p_{L^p(\lambda)} &\leq 2^{-Np}K^p + 2^{-Np}2K^p d(l-1) + 2d2^{-Np} \\ &\leq \left(K^p + 2K^p\frac{d}{d-1} + 2d\right)N2^{-Np}.\end{aligned}$$

We thus have, setting $c_3 := \left(K^p + 2K^p\frac{d}{d-1} + 2d\right)^{\frac{1}{p}}$,

$$\|f - \tilde{f}\|_{L^p(\lambda)} \leq c_3 N^{\frac{1}{p}}2^{-N}.$$

Notice c_3 only depends on d and p.

Letting $c_{d,p} = \max\{c_1, c_2, c_3\}$ yields the result. □

According to Proposition 5.E.1, the function $\tilde{f}$ constructed for a given $N \in \mathbb{N}^*$ can be implemented by a Heaviside neural network with two hidden layers and $W = 2(d+1)^2 \sum_{i=0}^{l} |S_i|$ weights. Using Lemma 41, we obtain

$$W = 2(d+1)^2 \sum_{i=0}^{l} |S_i| \leq 2(d+1)^2 \sum_{i=0}^{l} dK^{-i} 2^{i(d-1)+Nd+1}$$
$$= 2^{Nd+2} d(d+1)^2 \sum_{i=0}^{l} \left(\frac{2^{d-1}}{K}\right)^i .$$

We let, for all $N \in \mathbb{N}^*$,

$$W_N := 2^{Nd+2} d(d+1)^2 \sum_{i=0}^{l} \left(\frac{2^{d-1}}{K}\right)^i . \tag{5.E.13}$$

Although we do not make the dependence explicit, W_N also depends on d and p. Observe that for all $d \geq 1$: $(W_N)_{N\in\mathbb{N}^*}$ is non-decreasing and $\lim_{N\to+\infty} W_N = +\infty$.

Lemma 43. With the above notation: For any $+\infty > p \geq 1$, there exist constants $W'_{\min}, c'_{d,p} > 0$ depending only on d and $p \geq 1$ such that for all N satisfying $W_N \geq W'_{\min}$

$$\|f - \tilde{f}\|_{L^p(\lambda)} \leq c'_{d,p}\, g(W_{N+1})$$

where $\tilde{f}$ is constructed for the parameters N, p and d, and where for all $W \geq 1$,

$$g(W) = \begin{cases} W^{-1/d} & \text{if } (d-1)p < d, \\ W^{-\frac{1}{p(d-1)}} & \text{if } (d-1)p > d, \\ W^{-1/d} \log W & \text{if } (d-1)p = d. \end{cases}$$

Proof. Again, we distinguish three cases depending on the values of p and d.

— Suppose that $p(d-1) < d$: if $p = 1$, $\frac{2^{d-1}}{K} = \frac{1}{2} < 1$; if $p > 1$, since $\frac{1}{p-1} > d-1$, $\beta > d-1$ and $\frac{2^{d-1}}{K} = 2^{d-1-\beta} < 1$. Thus, in both cases $\frac{2^{d-1}}{K} < 1$ and for all $N \geq 1$,

$$W_N \leq 2^{Nd} \left(\frac{4d(d+1)^2}{1 - 2^{d-1-\beta}}\right) =: 2^{Nd} c''_{d,p}.$$

Writing the inequality for $N+1$, we obtain

$$W_{N+1} \leq 2^{Nd} 2^d c''_{d,p} .$$

That is: $2^{-N} \leq 2\left(\frac{c''_{d,p}}{W_{N+1}}\right)^{1/d}$. Combined with (5.E.7), this provides

$$\|f - \tilde{f}\|_{L^p(\lambda)} \leq 2c_{d,p}\left(\frac{c''_{d,p}}{W_{N+1}}\right)^{1/d} = d_{d,p}W_{N+1}^{-1/d},$$

for $d_{d,p} = 2c_{d,p}(c''_{d,p})^{1/d}$ and all $N \in \mathbb{N}^*$.

— If $p(d-1) > d$, then $\beta < d-1$ and $\frac{2^{d-1}}{K} = 2^{d-1-\beta} > 1$. Thus, reminding the definition of l in (5.E.3), we have for all $N \geq 1$

$$\begin{aligned} W_N &\leq 2^{Nd}2^{(d-1-\beta)(l+1)}\left(\frac{4d(d+1)^2}{2^{d-1-\beta}-1}\right) \leq 2^{Nd}2^{(d-1-\beta)(N/\beta+2)}\left(\frac{4d(d+1)^2}{2^{d-1-\beta}-1}\right) \\ &= 2^{N(d+(d-1)/\beta-1)}\left(\frac{4d(d+1)^2 2^{2(d-1-\beta)}}{2^{d-1-\beta}-1}\right) =: 2^{N(1+\frac{1}{\beta})(d-1)}c''_{d,p}, \end{aligned}$$

for a different constant $c''_{d,p}$. Writing again this inequality for $N+1$, we obtain

$$W_{N+1} \leq c''_{d,p}2^{(1+\frac{1}{\beta})(d-1)}\, 2^{N(1+\frac{1}{\beta})(d-1)},$$

which we can write $2^{-N(1+\frac{1}{\beta})} \leq 2^{(1+\frac{1}{\beta})}\left(\frac{c''_{d,p}}{W_{N+1}}\right)^{\frac{1}{d-1}}$. This provides

$$2^{-N\frac{(1+1/\beta)}{p}} \leq 2^{\frac{(1+1/\beta)}{p}}\left(\frac{c''_{d,p}}{W_{N+1}}\right)^{\frac{1}{p(d-1)}}.$$

Therefore, using (5.E.7), we obtain

$$\|f - \tilde{f}\|_{L^p(\lambda)} \leq c_{d,p}2^{\frac{(1+1/\beta)}{p}}\left(\frac{c''_{d,p}}{W_{N+1}}\right)^{\frac{1}{p(d-1)}} = d'_{d,p}W_{N+1}^{-\frac{1}{p(d-1)}},$$

for $d'_{d,p} = c_{d,p}\, 2^{\frac{(1+1/\beta)}{p}}\, (c''_{d,p})^{\frac{1}{p(d-1)}}$ and all $N \in \mathbb{N}^*$.

— If $p(d-1) = d$, then $\beta = d-1$ and $\frac{2^{d-1}}{K} = 1$. Thus, reminding the definition of l in (5.E.3), we have for all $N \geq 1$

$$\begin{aligned} W_N &= 2^{Nd+2}d(d+1)^2(l+1) \leq 2^{Nd+2}d(d+1)^2\left(\frac{N}{\beta}+2\right) \\ &= 2^{Nd}\left(\frac{N}{\beta}+2\right)\left(4d(d+1)^2\right) =: 2^{Nd}\left(\frac{N}{d-1}+2\right)c''_{d,p} \\ &\leq 2^{d(d-1)(\frac{N}{d-1}+2)}\left(\frac{N}{d-1}+2\right)c''_{d,p} \\ &= \exp\left(d(d-1)\left(\frac{N}{d-1}+2\right)\log 2\right)\left(\frac{N}{d-1}+2\right)c''_{d,p} \end{aligned} \tag{5.E.14}$$

where $c''_{d,p} = 4d(d+1)^2$. Setting

$$\tilde{W}_N := \frac{d(d-1)W_N \log 2}{c''_{d,p}} \qquad \text{and} \qquad \tilde{N} := d(d-1)\left(\frac{N}{d-1} + 2\right)\log 2,$$

we can rewrite (5.E.14) as:

$$\tilde{W}_N \leq \tilde{N}\exp(\tilde{N})\ . \tag{5.E.15}$$

Since $d \geq 2$, $c''_{d,p} > 0$, $(W_N)_{N\in\mathbb{N}^*}$ is non-decreasing and $\lim_{N\to+\infty} W_N = +\infty$, there exists W'_{min} such that, for all N satisfying $W_N \geq W'_{min}$, we have the following:

$$\begin{cases} \log(\tilde{W}_N) > 1 \\ \log(\tilde{W}_{N+1}) > 2\ \log(2)\ d(d-1) \\ \frac{\log(\tilde{W}_{N+1})}{d\log(2)} - \frac{\log\log(\tilde{W}_{N+1})}{d\log(2)} - 2(d-1) > \frac{1}{p\log(2)} \\ \log W_{N+1} \geq \log\left(\frac{d(d-1)\log 2}{c''_{d,p}}\right). \end{cases} \tag{5.E.16}$$

These inequalities will be used latter in the proof and, from now on, we always consider N such that $W_N \geq W'_{min}$.

Let us first show by contradiction that, for all N satisfying $W_N \geq W'_{min}$, (5.E.15) implies that

$$\tilde{N} \geq \log\tilde{W}_N - \log\log\tilde{W}_N. \tag{5.E.17}$$

Indeed, if the latter does not hold

$$\tilde{N} < \log\tilde{W}_N - \log\log\tilde{W}_N,$$

$$\exp(\tilde{N}) < \frac{\tilde{W}_N}{\log\tilde{W}_N},$$

and therefore, multiplying the two inequalities, since (5.E.16) implies that $\tilde{W}_N > 0$, $\log\tilde{W}_N > 0$ and $\log(\log(\tilde{W}_N)) > 0$,

$$\tilde{N}\exp(\tilde{N}) < \tilde{W}_N.$$

The latter being in contradiction with (5.E.15), we have proved that, for all N satisfying $W_N \geq W'_{min}$, (5.E.17) holds. Using the definition of $\tilde{N}$, we deduce

$$\begin{aligned} N &\geq \left(\frac{\log\tilde{W}_N - \log\log\tilde{W}_N}{d(d-1)\log(2)} - 2\right)(d-1) \\ &= \frac{\log(\tilde{W}_N)}{d\log(2)} - \frac{\log\log\tilde{W}_N}{d\log(2)} + c, \end{aligned}$$

for the constant $c = -2(d-1) < 0$. Since $(W_N)_{N\in\mathbb{N}}$ is non-decreasing, for all N satisfying $W_N \geq W'_{min}$, $W_{N+1} \geq W'_{min}$ and the inequality also holds for $N+1$.

That is

$$N+1 \geq \frac{\log(\tilde{W}_{N+1})}{d\log(2)} - \frac{\log\log\tilde{W}_{N+1}}{d\log(2)} + c. \tag{5.E.18}$$

Using (5.E.7), we obtain:

$$\|f-\tilde{f}\|_{L^p(\lambda)} \leq c_{d,p} N^{\frac{1}{p}} 2^{-N} \leq 2c_{d,p}(N+1)^{\frac{1}{p}} 2^{-(N+1)}.$$

Since, for $t > \frac{1}{p\log(2)}$, the function $t \longmapsto t^{\frac{1}{p}}2^{-t}$ is non-increasing, using (5.E.18) and (5.E.16) and the fact that $-\frac{\log\log\tilde{W}_{N+1}}{d\log(2)} + c < 0$, we obtain

$$\begin{aligned}
\|f-\tilde{f}\|_{L^p(\lambda)} &\leq 2c_{d,p}\left(\frac{\log(\tilde{W}_{N+1})}{d\log(2)}\right)^{\frac{1}{p}} 2^{-\frac{\log(\tilde{W}_{N+1})}{d\log(2)}} 2^{\frac{\log\log\tilde{W}_{N+1}}{d\log(2)}} 2^{-c}, \\
&= \left(\frac{2^{1-c}c_{d,p}}{(d\log(2))^{1/p}}\right) (\log\tilde{W}_{N+1})^{\frac{1}{p}+\frac{1}{d}}\ \tilde{W}_{N+1}^{-\frac{1}{d}} \\
&= \left(\frac{2^{1-c}c_{d,p}}{(d\log(2))^{1/p}}\right) \tilde{W}_{N+1}^{-\frac{1}{d}}\ \log\tilde{W}_{N+1},
\end{aligned}$$

since $p(d-1) = d$ implies $\frac{1}{p} + \frac{1}{d} = 1$. Finally, using the definition of $\tilde{W}_N$ and (5.E.16), we obtain

$$\|f-\tilde{f}\|_{L^p(\lambda)} \leq d''_{d,p} W_{N+1}^{-\frac{1}{d}}\ \log W_{N+1},$$

for the constant $d''_{d,p} = 2\left(\frac{2^{1-c}c_{d,p}}{(d\log(2))^{1/p}}\right)\left(\frac{d(d-1)\log 2}{c''_{d,p}}\right)^{-1/d}$ and all $N \in \mathbb{N}^*$ such that $W_N \geq W'_{min}$. Notice $d''_{d,p}$ only depends on d and p.

Taking $c'_{d,p} = \max(d_{d,p}, d'_{d,p}, d''_{d,p})$ provides the announced statement. □

Proof of Proposition 5.4.3. Take $W_{\min} = \max(W'_{\min}, W_1)$ and $c = c'_{d,p}$, where $W'_{\min}$ and $c'_{d,p}$ are from Lemma 43 and W_1 is defined in (5.E.13). Let $W \geq W_{\min}$, there exists $N \in \mathbb{N}^*$ such that

$$W_N \leq W < W_{N+1}.$$

Consider the architecture $\mathcal{A}$ with W weights, as in Proposition 5.E.1, which allows to represent piecewise-constant functions with less than $\frac{W}{2(d+1)^2}$ cubic pieces. It can represent piecewise-constant functions with $\frac{W_N}{2(d+1)^2}$ pieces.

For any $f \in \mathcal{M}^d$, the function $\tilde{f}$ obtained for the parameter N is a piecewise-constant function with at most $\frac{W_N}{2(d+1)^2}$ pieces, therefore we have $\tilde{f} \in H_{\mathcal{A}}$ and, according to Lemma 43, $\tilde{f}$ satisfies

$$\|f-\tilde{f}\|_{L^p(\lambda)} \leq c'_{d,p} g(W_{N+1}).$$

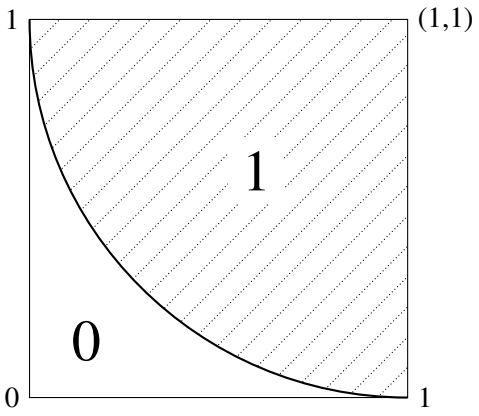

Figure 5.3 – The set $\mathcal{C}$, the set $\partial\mathcal{C} \cap (0,1)^2$ and the indicator function f.

Moreover, since g is non-increasing, we have using $c = c'_{d,p}$

$$\|f - \tilde{f}\|_{L^p(\lambda)} \leq c\, g(W).$$

Therefore, for any $f \in \mathcal{M}^d$,

$$\inf_{g \in H_{\mathcal{A}}} \|f - g\|_{L^p(\lambda)} \leq c\, g(W)$$

and so does the supremum over f in $\mathcal{M}^d$.

This concludes the proof of Proposition 5.4.3.

5.E.2 Proof of Proposition 5.4.1

Step 1: we prove the result in dimension $d = 2$.

We consider the closed disk of radius 1, centered at $(1,1)$,

$$\mathcal{C} = \left\{ x \in \mathbb{R}^2 : \sum_{i=1}^{2} (x_i - 1)^2 \leq 1 \right\} .$$

The intersection between $(0,1)^2$ and the topological boundary $\partial\mathcal{C}$ of $\mathcal{C}$ is the quarter of circle:

$$\partial\mathcal{C} \cap (0,1)^2 = \left\{ x \in (0,1)^2 : \sum_{i=1}^{2} (x_i - 1)^2 = 1 \right\} .$$

We denote by $f : [0,1]^2 \to \{0,1\}$ the indicator function of the set $\mathcal{C} \cap [0,1]^2$. The set $\mathcal{C} \cap [0,1]^2$, the set $\partial\mathcal{C} \cap (0,1)^2$ and the function f are represented on Figure 5.3.

Since no point in $\mathcal{C}^c \cap [0,1]^2$ has all its coordinates strictly larger than those of a point in $\mathcal{C}$, we have $f \in \mathcal{M}^2$ (monotonic functions of 2 variables). We consider an arbitrary neural network architecture $\mathcal{A}$ and $g \in H_{\mathcal{A}}$.

Let $W \geq 1$ be the number of weights in the architecture $\mathcal{A}$. As is well known for Heaviside neural networks, there exist $K \in \mathbb{N}$ with $K \leq 2^W$, reals α_j and polygons $A_j \subset [0,1]^2$, for $j \in \{1, \dots, K\}$, such that for all $x \in [0,1]^2$

$$g(x) = \sum_{j=1}^{K} \alpha_j \mathbb{1}_{A_j}(x).$$

Moreover, $(A_j)_{1 \leq j \leq K}$ form a partition of $[0,1]^2$.

The proof relies on the fact (proved afterwards) that, if $\|f - g\|_\infty < \frac{1}{2}$ then $\partial\mathcal{C} \cap (0,1)^2$ is finite. The latter being false, we conclude that $\|f - g\|_\infty \geq \frac{1}{2}$.

Assume from now on that $\|f - g\|_\infty < \frac{1}{2}$. This implies that $g > \frac{1}{2}$ on $\mathcal{C}$, and $g < \frac{1}{2}$ elsewhere. Let us first show that we then have

$$\partial\mathcal{C} \cap (0,1)^2 \ \subset \ \bigcup_{j=1}^{K} \partial A_j.$$

Indeed, if the latter were not true, then there would exist $x \in \partial\mathcal{C} \cap (0,1)^2$ and $j \in \{1, \dots, K\}$ such that $x \in \mathring{A}_j$. Since $\mathcal{C}$ is closed, $x \in \mathcal{C}$. Let $\epsilon > 0$ be such that $B(x, \epsilon) \subset \mathring{A}_j$. We have $B(x, \epsilon) \not\subset \mathcal{C}$ (otherwise, x belongs to the interior of $\mathcal{C}$ which contradicts $x \in \partial\mathcal{C}$). Thus there exists $z \in B(x, \epsilon) \setminus \mathcal{C}$. Since $g > \frac{1}{2}$ on $\mathcal{C}$, and $g < \frac{1}{2}$ elsewhere, we have

$$g(z) < \frac{1}{2} < g(x).$$

This is not possible since $x, z \in \mathring{A}_j$ and g is constant on A_j. This concludes the proof of the following fact: if $\|f - g\|_\infty < \frac{1}{2}$ then $\partial\mathcal{C} \cap (0,1)^2 \subset \bigcup_{1 \leq j \leq K} \partial A_j$.

Since the A_j are polygons (recall that we work in dimension 2), their boundaries are finite unions of closed line segments. Then $\partial\mathcal{C} \cap (0,1)^2$ is included in a finite union of closed line segments which we denote S_m, for $m \in \{1, \dots, M\}$. The reader may already see that this is in contradiction with the fact that $\partial\mathcal{C} \cap (0,1)^2$ is a quarter circle. To detail this argument and complete the announced proof, we show that $\partial\mathcal{C} \cap (0,1)^2 \subset \bigcup_{m=1}^{M} S_m$ implies that $\partial\mathcal{C} \cap (0,1)^2$ is finite.

To do so, since when $\partial\mathcal{C} \cap (0,1)^2 \subset \bigcup_{m=1}^{M} S_m$ we have

$$\bigcup_{m=1}^{M} \left(\partial\mathcal{C} \cap (0,1)^2 \cap S_m\right) = \partial\mathcal{C} \cap (0,1)^2,$$

it suffices to prove that the intersection of any closed line segment S with $\partial\mathcal{C} \cap (0,1)^2$ contains at most 2 points.

Denote by S a closed line segment: $\mathcal{C}$ and S are convex and hence connected, thus $\mathcal{C} \cap S$ is either empty, a singleton or a line segment, as a connected compact subset of S. If it is empty, then *a fortiori*, $\partial\mathcal{C} \cap (0,1)^2 \cap S = \emptyset$. If it is not, denote by y and z its extremities (assuming $z = y$ in the case of a singleton). By strict convexity of the function $x \mapsto \sum_{i=1}^{2} (x_i - 1)^2$, the open line segment (y, z) is included in $\mathring{\mathcal{C}}$ ($(y, z) = \emptyset$ in the case

of a singleton), hence

$$\partial\mathcal{C} \cap (0,1)^2 \cap S \ \subset \ [y,z] \setminus \mathring{\mathcal{C}} \ \subset \ \{y,z\}.$$

In any case, we have $|\partial\mathcal{C} \cap (0,1)^2 \cap S| \leq 2$.

This concludes the proof of the fact: if $\|f-g\|_\infty < \frac{1}{2}$ then $\partial\mathcal{C} \cap (0,1)^2$ is finite and concludes the proof in the case $d=2$.

Step 2: we prove the result in any dimension $d \geq 2$, by a reduction to dimension 2.

We define

$$\mathcal{C} = \left\{ x \in \mathbb{R}^d : \sum_{i=1}^{d} (x_i - 1)^2 \leq 1 \right\} ,$$

and the function $f : [0,1]^d \to \mathbb{R}$ by

$$f(x_1, \dots, x_d) = \mathbb{1}_{(x_1,\dots,x_d)\in\mathcal{C}} \, .$$

Consider an arbitrary neural network architecture $\mathcal{A}$ and $g \in H_{\mathcal{A}}$. That is, g can be represented by a Heaviside neural network with d input neurons. Note that

$$\begin{aligned} \sup_{x_1,x_2,x_3\dots,x_d\in[0,1]} & |f(x_1,x_2,x_3\dots,x_d) - g(x_1,x_2,x_3\dots,x_d)| \\ &\geq \sup_{x_1,x_2\in[0,1]} |f(x_1,x_2,1\dots,1) - g(x_1,x_2,1\dots,1)| \\ &\geq \frac{1}{2} , \end{aligned}$$

where the last inequality is by the result of Step 1, since $(x_1,x_2) \in [0,1]^2 \mapsto f(x_1,x_2,1\dots,1)$ is the indicator function of Step 1, and $(x_1,x_2) \in [0,1]^2 \mapsto g(x_1,x_2,1\dots,1)$ can be represented by a Heaviside neural network with 2 input neurons. This concludes the proof.

Remark. Note from the above proof that, though we only stated the impossibility result for piecewise-constant activation functions, an analogue statement in fact holds more generally for piecewise-affine activation functions.

5.F Barron space

In Section 5.5 we mentioned that the Barron space introduced in [10] is one among several examples for which approximation theory provides ready-to-use lower bounds on the packing number. This space has received renewed attention recently in the deep learning community, in particular because its "size" is sufficiently small to avoid approximation rates depending exponentially on the input dimension d. Next we detail how to apply Corollary 1 in this case.

Definition of the Barron space. We start by introducing the Barron space, as defined in [112]. Let $d \in \mathbb{N}^*$. For any constant $C > 0$, the Barron space $B_d(C)$ is the set of all

functions $f : [0,1]^d \to [0,1]$ for which there exist a measurable function $F : \mathbb{R}^d \to \mathbb{C}$ and some $c \in [-C, C]$ such that, for all $x \in [0,1]^d$,

$$f(x) = c + \int_{\mathbb{R}^d} (e^{ix\cdot\xi} - 1)F(\xi)\mathrm{d}\xi \qquad \text{and} \qquad \int_{\mathbb{R}^d} \|\xi\|_2 |F(\xi)|\mathrm{d}\xi \leq C,$$

where $x \cdot \xi$ denotes the standard scalar product in between x and ξ.

Known lower bound on the packing number. Petersen and Voigtlaender [112] showed a tight lower bound on the log packing number in $L^p(\lambda, [0,1]^d)$ norm, which we recall below.

Proposition 5.F.1 (Proposition 4.6 in [112]). Let $1 \leq p \leq +\infty$. There exist constants $\varepsilon_0, c_0 > 0$ depending only on d and C such that for any $\varepsilon \leq \varepsilon_0$,

$$\log M(\varepsilon, B_d(C), \|\cdot\|_{L^p}) \geq c_0 \varepsilon^{-1/(\frac{1}{2}+\frac{1}{d})}. \tag{5.F.1}$$

Consequence on the approximation rate by piecewise-polynomial neural networks. Plugging the lower bound of Proposition 5.F.1 in Corollary 1, we obtain the following lower bound on the approximation error of the Barron space by piecewise-polynomial neural networks.

Proposition 5.F.2. Let $1 \leq p < +\infty$, $d \geq 1$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a piecewise-polynomial function on $K \geq 2$ pieces, with maximal degree $\nu \in \mathbb{N}$. Consider the Barron space $B_d(C)$ defined above, with $C > 0$. There exist positive constants c_1, c_2, c_3, W_{min} depending only on d, p, C, K and ν such that, for any architecture $\mathcal{A}$ of depth $L \geq 1$ with $W \geq W_{min}$ weights, and for the activation σ, the set $H_{\mathcal{A}}$ (cf. Section 5.1) satisfies

$$\sup_{f \in B_d(C)} \inf_{g \in H_{\mathcal{A}}} \|f - g\|_{L^p(\lambda)} \geq \begin{cases} c_1 W^{-1-\frac{2}{d}} \log^{-1-\frac{2}{d}}(W) & \text{if } \nu \geq 2\,, \\ c_2 (LW)^{-\frac{1}{2}-\frac{1}{d}} \log^{-\frac{3}{2}-\frac{3}{d}}(W) & \text{if } \nu = 1\,, \\ c_3 W^{-\frac{1}{2}-\frac{1}{d}} \log^{-\frac{3}{2}-\frac{3}{d}}(W) & \text{if } \nu = 0\,. \end{cases} \tag{5.F.2}$$

Bibliography

[1] Noga Alon, Shai Ben-David, Nicolò Cesa-Bianchi, and David Haussler. "Scale-Sensitive Dimensions, Uniform Convergence, and Learnability". In: *J. ACM* 44.4 (1997), pp. 615–631. ISSN: 0004-5411. DOI: 10.1145/263867.263927.

[2] Martin Anthony and Peter L. Bartlett. *Neural Network learning: Theoretical Foundations*. Cambridge University Press, Cambridge, 1999.

[3] Martin Arjovsky and Léon Bottou. "Towards principled methods for training generative adversarial networks". In: *International Conference on Learning Representation, ICLR'17*. 2017.

[4] Martin Arjovsky, Amar Shah, and Yoshua Bengio. "Unitary evolution recurrent neural networks". In: *International Conference on Machine Learning, ICML'16*. 2016.

[5] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. "A Convergence Analysis of Gradient Descent for Deep Linear Neural Networks". In: *International Conference on Learning Representations*. 2019.

[6] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. "Implicit regularization in deep matrix factorization". In: *Advances in Neural Information Processing Systems*. 2019, pp. 7413–7424.

[7] Bubacarr Bah, Holger Rauhut, Ulrich Terstiege, and Michael Westdickenberg. "Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers". In: *Information and Inference: A Journal of the IMA* (2021). 10.1093/imaiai/iaaa039. ISSN: 2049-8772. DOI: 10.1093/imaiai/iaaa039. eprint: https://academic.oup.com/imaiai/advance-article-pdf/doi/10.1093/imaiai/iaaa039/36213130/iaaa039.pdf.

[8] P. Baldi and K. Hornik. "Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima". In: *Neural Netw.* 2.1 (1989), pp. 53–58.

[9] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. "Can We Gain More from Orthogonality Regularizations in Training Deep Networks?" In: *Advances in Neural Information Processing Systems, NeurIPS'18* 31 (2018), pp. 4261–4271.

[10] Andrew Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information Theory* 39 (1993), pp. 930–945.

[11] Peter Bartlett, Dave Helmbold, and Philip Long. "Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks". In: *International conference on machine learning*. 2018, pp. 521–530.

[12] Peter L. Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. "Nearly-tight VC-dimension and Pseudodimension Bounds for Piecewise Linear Neural Networks". In: *Journal of Machine Learning Research* 20.63 (2019), pp. 1–17.

[13] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. "Benign overfitting in linear regression". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30063–30070. eprint: https://www.pnas.org/content/117/48/30063.full.pdf.

[14] Mikhail Belkin. "Approximation beats concentration? An approximation view on inference with smooth radial kernels". In: *Proceedings of the 31st Conference On Learning Theory*. Ed. by Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet. Vol. 75. Proceedings of Machine Learning Research. PMLR, 2018, pp. 1348–1361.

[15] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854. eprint: https://www.pnas.org/content/116/32/15849.full.pdf.

[16] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[17] Louis Béthune, Alberto González-Sanz, Franck Mamalet, and Mathieu Serrurier. "Pay attention to your loss: understanding misconceptions about 1-Lipschitz neural networks". In: *arXiv preprint arXiv:2104.05097* (2022).

[18] Mikhail Š Birman and M Z Solomjak. "Piecewise-Polynomial Approximation of Functions of the Classes W_p^α". In: *Mathematics of The Ussr-sbornik* 2.3 (1967), pp. 295–317. DOI: 10.1070/sm1967v002n03abeh002343.

[19] Ron Blei, Fuchang Gao, and Wenbo V. Li. "Metric Entropy of High Dimensional Distributions". In: *Proceedings of the American Mathematical Society* 135.12 (2007), pp. 4009–4018. ISSN: 00029939, 10886826.

[20] Avrim Blum and Ronald L Rivest. "Training a 3-node neural network is NP-complete". In: *Advances in neural information processing systems*. 1989, pp. 494–501.

[21] Peter Bühlmann and Bin Yu. "Boosting With the L2 Loss". In: *Journal of the American Statistical Association* 98.462 (2003), pp. 324–339. DOI: 10.1198/016214503000125. eprint: https://doi.org/10.1198/016214503000125.

[22] Emmanuel Caron and Stéphane Chrétien. "A finite sample analysis of the benign overfitting phenomenon for ridge function estimation". In: *arXiv preprint arXiv:2007.12882* (2020).

[23] Elaina Chai, Mert Pilanci, and Boris Murmann. "Separating the effects of batch normalization on cnn training speed and stability using classical adaptive filter theory". In: *Asilomar Conference on Signals, Systems, and Computers*. IEEE. 2020, pp. 1214–1221.

[24] Yacine Chitour, Zhenyu Liao, and Romain Couillet. "A geometric approach of gradient descent algorithms in neural networks". In: *arXiv preprint arXiv:1811.03568* (2018).

[25] Lénaïc Chizat and Francis Bach. "On the global convergence of gradient descent for over-parameterized models using optimal transport". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 2018, pp. 3040–3050.

[26] Anna Choromanska, MIkael Henaff, Michael Mathieu, Gerard Ben Arous, and Yann LeCun. "The Loss Surfaces of Multilayer Networks". In: *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. Vol. 38. 2015, pp. 192–204.

[27] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. "Parseval networks: improving robustness to adversarial examples". In: *Proceedings of the 34th International Conference on Machine Learning, ICML'17*. 2017, pp. 854–863.

[28] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signal and Systems* 2 (1989), pp. 303–314. DOI: https://doi.org/10.1007/BF02551274.

[29] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. "Escaping Saddles with Stochastic Gradients". In: *International Conference on Machine Learning*. 2018, pp. 1155–1164.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*. 2009.

[31] Ronald DeVore, Boris Hanin, and Guergana Petrova. "Neural network approximation". In: *Acta Numerica* 30 (2021), pp. 327–444.

[32] Ronald A DeVore, Ralph Howard, and Charles Micchelli. "Optimal nonlinear approximation". In: *Manuscripta mathematica* 63.4 (1989), pp. 469–478.

[33] Simon Du and Wei Hu. "Width Provably Matters in Optimization for Deep Linear Neural Networks". In: *International Conference on Machine Learning*. 2019, pp. 1655–1664.

[34] Alan Edelman, Tomás A Arias, and Steven T Smith. "The geometry of algorithms with orthogonality constraints". In: *SIAM journal on Matrix Analysis and Applications* 20.2 (1998), pp. 303–353.

[35] David E. Edmunds and Hans Triebel. *Function Spaces, Entropy Numbers, Differential Operators*. Cambridge Tracts in Mathematics. Cambridge University Press, 1996. DOI: 10.1017/CBO9780511662201.

[36] Armin Eftekhari. "Training Linear Neural Networks: Non-Local Convergence and Complexity Results". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. 2020, pp. 2836–2847.

[37] Farzan Farnia, Jesse Zhang, and David Tse. "Generalizable Adversarial Training via Spectral Normalization". In: *International Conference on Learning Representations, ICLR'18*. 2018.

[38] Yanhong Fei, Yingjie Liu, Xian Wei, and Mingsong Chen. *O-ViT: Orthogonal Vision Transformer*. 2022.

[39] Abraham Frandsen and Rong Ge. "Optimization landscape of Tucker decomposition". In: *Mathematical Programming* (2020), pp. 1–26.

[40] Spencer Frei, Niladri S. Chatterji, and Peter L. Bartlett. *Benign Overfitting without Linearity: Neural Network Classifiers Trained by Gradient Descent for Noisy Linear Data*. 2022. DOI: 10.48550/ARXIV.2202.05928.

[41] Fuchang Gao and Jon A. Wellner. "Entropy estimate for high-dimensional monotonic functions". In: *Journal of Multivariate Analysis* 98.9 (2007), pp. 1751–1764.

[42] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. "Escaping from saddle points—online stochastic gradient for tensor decomposition". In: *Conference on Learning Theory*. 2015, pp. 797–842.

[43] Rong Ge and Tengyu Ma. "On the optimization landscape of tensor decompositions". In: *Mathematical Programming* (2020), pp. 1–47.

[44] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. "Implicit regularization of discrete gradient dynamics in linear neural networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 3202–3211.

[45] Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. "The Implicit Bias of Depth: How Incremental Learning Drives Generalization". In: *International Conference on Learning Representations*. 2019.

[46] Paul W. Goldberg and Mark Jerrum. "Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers". In: *Machine Learning* 18 (1995), pp. 131–148.

[47] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT press Cambridge, 2016.

[48] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. "Regularisation of neural networks by enforcing lipschitz continuity". In: *Machine Learning* 110.2 (2021), pp. 393–416.

[49] Piet Groeneboom and Geurt Jongbloed. *Nonparametric Estimation under Shape Constraints: Estimators, Algorithms and Asymptotics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2014. DOI: 10.1017/CBO9781139020893.

[50] Yann Guermeur. "Lp-norm Sauer–Shelah lemma for margin multi-category classifiers". In: *Journal of Computer and System Sciences* 89 (2017), pp. 450–473. ISSN: 0022-0000. DOI: https://doi.org/10.1016/j.jcss.2017.06.003.

[51] Ingo Gühring and Mones Raslan. "Approximation rates for neural networks with encodable weights in smoothness spaces". In: *Neural Networks* 134 (2020), pp. 107–130.

[52] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. "Improved training of Wasserstein GANs". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems, NeurIPS'17*. 2017, pp. 5769–5779.

[53] Adityanand Guntuboyina and Bodhisattva Sen. "Covering Numbers for Convex Functions". In: *IEEE Transactions on Information Theory* 59.4 (2013), pp. 1957–1965. DOI: 10.1109/TIT.2012.2235172.

[54] Pei-Chang Guo and Qiang Ye. "On the regularization of convolutional kernel tensors in neural networks". In: *Linear and Multilinear Algebra* 0.0 (2020), pp. 1–13.

[55] László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Series in Statistics. New York: Springer-Verlag, 2002, pp. xvi+647. ISBN: 0-387-95441-4.

[56] Benjamin D Haeffele and René Vidal. "Global optimality in neural network training". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7331–7339.

[57] Moritz Hardt and Tengyu Ma. "Identity Matters in Deep Learning". In: *5th International Conference on Learning Representations*. 2017.

[58] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. *Surprises in High-Dimensional Ridgeless Least Squares Interpolation*. arXiv:1903.08560. 2020. arXiv: 1903.08560 [math.ST].

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, CVPR'16*. 2016, pp. 770–778.

[60] Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. "Fast and flexible convolutional sparse coding". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR'15*. 2015.

[61] Sepp Hochreiter. "Untersuchungen zu dynamischen neuronalen Netzen". In: *Diploma, Technische Universität München* 91.1 (1991).

[62] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2 (1991), pp. 251–257. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(91)90009-T.

[63] Jeremy Howard. *Imagenette*. 2020.

[64] Lei Huang, Li Liu, Fan Zhu, Diwen Wan, Zehuan Yuan, Bo Li, and Ling Shao. "Controllable orthogonalization in training dnns". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'20*. 2020, pp. 6429–6438.

[65] Lei Huang, Xianglong Liu, Bo Lang, Adams Yu, Yongliang Wang, and Bo Li. "Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks". In: *Proceedings of the Conference on Artificial Intelligence, AAAI'18*. Vol. 32. 2018.

[66] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *International Conference on Machine Learning, ICML'15*. PMLR. 2015, pp. 448–456.

[67] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems* 31 (2018).

[68] Arthur Jacot, François Ged, Berfin Şimşek, Clément Hongler, and Franck Gabriel. *Saddle-to-Saddle Dynamics in Deep Linear Networks: Small Initialization Training, Symmetry, and Sparsity*. 2021. DOI: 10.48550/ARXIV.2106.15933.

[69] Xu Ji, João F Henriques, and Andrea Vedaldi. "Invariant information clustering for unsupervised image classification and segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV'19*. 2019, pp. 9865–9874.

[70] Kui Jia, Shuai Li, Yuxin Wen, Tongliang Liu, and Dacheng Tao. "Orthogonal deep neural networks". In: *IEEE transactions on Pattern Analysis and Machine Intelligence, TPAMI'19* (2019).

[71] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. "How to escape saddle points efficiently". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017, pp. 1724–1732.

[72] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan. "On Nonconvex Optimization for Machine Learning: Gradients, Stochasticity, and Saddle Points". In: *J. ACM* 68.2 (2021). ISSN: 0004-5411. DOI: 10.1145/3418526.

[73] Chi Jin, Praneeth Netrapalli, and Michael I Jordan. "Accelerated gradient descent escapes saddle points faster than gradient descent". In: *Conference On Learning Theory*. 2018, pp. 1042–1085.

[74] Marek Karpinski and Angus Macintyre. "Polynomial Bounds for VC Dimension of Sigmoidal and General Pfaffian Neural Networks". In: *Journal of Computer and System Sciences* 54.1 (1997), pp. 169–176. ISSN: 0022-0000.

[75] Kenji Kawaguchi. "Deep Learning without Poor Local Minima". In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 586–594.

[76] Patrick Kidger and Terry Lyons. "Universal Approximation with Deep Narrow Networks". In: *Proceedings of Thirty Third Conference on Learning Theory*. Ed. by Jacob Abernethy and Shivani Agarwal. Vol. 125. Proceedings of Machine Learning Research. PMLR, 2020, pp. 2306–2327.

[77] Hyunjik Kim, George Papamakarios, and Andriy Mnih. "The Lipschitz Constant of Self-Attention". In: *Proceedings of the 38th International Conference on Machine Learning*. Proceedings of Machine Learning Research. 2021.

[78] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". In: *International Conference on Learning Representations, ICLR'14*. 2014.

[79] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *ICLR (Poster)*. 2015.

[80] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.

[81] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems, NIPS'12* 25 (2012), pp. 1097–1105.

[82] Agostina J Larrazabal, César Martinez, Jose Dolz, and Enzo Ferrante. "Orthogonal ensemble networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2021, pp. 594–603.

[83] Thomas Laurent and James von Brecht. "Deep Linear Networks with Arbitrary Loss: All Local Minima Are Global". In: *ICML*. 2018, pp. 2908–2913.

[84] Yann LeCun and Yoshua Bengio. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995).

[85] Jason D Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I Jordan, and Benjamin Recht. "First-order methods almost always avoid strict saddle points". In: *Mathematical programming* 176.1-2 (2019), pp. 311–337.

[86] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. "Gradient Descent Converges to Minimizers". In: *University of California, Berkeley* (2016).

[87] Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. "Gradient Descent Only Converges to Minimizers". In: *Proceedings of the 29th Conference on Learning Theory*. Vol. 49. 2016, pp. 1246–1257.

[88] Moshe Leshno, Vladimir Y. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(05)80131-5.

[89] Mario Lezcano-Casado and David Martınez-Rubio. "Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group". In: *International Conference on Machine Learning, ICML'19*. PMLR. 2019, pp. 3794–3803.

[90] Jun Li, Fuxin Li, and Sinisa Todorovic. "Efficient Riemannian Optimization on the Stiefel Manifold via the Cayley Transform". In: *International Conference on Learning Representations, ICLR'19*. 2019.

[91] Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger B Grosse, and Jörn-Henrik Jacobsen. "Preventing gradient attenuation in lipschitz constrained convolutional networks". In: *Advances in Neural Information Processing Systems, NeurIPS'19*. 2019.

[92] Zhu Li, Weijie J Su, and Dino Sejdinovic. "Benign overfitting and noisy features". In: *Journal of the American Statistical Association* (2022), pp. 1–13.

[93] Haihao Lu and Kenji Kawaguchi. "Depth creates no bad local minima". In: *arXiv preprint arXiv:1702.08580* (2017).

[94] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards Deep Learning Models Resistant to Adversarial Attacks". In: *International Conference on Learning Representations, ICLR'18*. 2018.

[95] Vitaly E. Maiorov. "On Best Approximation by Ridge Functions". In: *Journal of Approximation Theory* 99.1 (1999), pp. 68–94. ISSN: 0021-9045. DOI: https://doi.org/10.1006/jath.1998.3304.

[96] Vitaly E. Maiorov, Ron Meir, and Joel Ratsaby. "On the Approximation of Functional Classes Equipped with a Uniform Measure Using Ridge Functions". In: *Journal of Approximation Theory* 99 (1999), pp. 95–111.

[97] Vitaly E. Maiorov and Allan Pinkus. "Lower bounds for approximation by MLP neural networks". In: *Neurocomputing* 25.1 (1999), pp. 81–91. ISSN: 0925-2312. DOI: https://doi.org/10.1016/S0925-2312(98)00111-8.

[98] Dhagash Mehta, Tianran Chen, Tingting Tang, and Jonathan Hauenstein. "The Loss Surface Of Deep Linear Networks Viewed Through The Algebraic Geometry Lens". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). 10.1109/TPAMI.2021.3071289.

[99] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. "A mean field view of the landscape of two-layer neural networks". In: *Proceedings of the National Academy of Sciences* 115.33 (2018), E7665–E7671. eprint: https://www.pnas.org/content/115/33/E7665.full.pdf.

[100] Shahar Mendelson. "Rademacher averages and phase transitions in Glivenko-Cantelli classes". In: *IEEE Transactions on Information Theory* 48 (2002).

[101] Shahar Mendelson and Roman Vershynin. "Entropy and the combinatorial dimension". In: *Inventiones mathematicae* 152 (2003), pp. 37–55. DOI: https://doi.org/10.1007/s00222-002-0266-3.

[102] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. "Spectral Normalization for Generative Adversarial Networks". In: *International Conference on Learning Representations, ICLR'18*. 2018.

[103] Igor Molybog, Somayeh Sojoudi, and Javad Lavaei. "Role of sparsity and structure in the optimization landscape of non-convex matrix sensing". In: *Mathematical Programming* (2020), pp. 1–37.

[104] Katta G Murty and Santosh N Kabadi. "Some NP-complete problems in quadratic and nonlinear programming". In: *Mathematical Programming* 39.2 (1987), pp. 117–129.

[105] Yurii Nesterov. "Introductory lectures on convex programming volume i: Basic course". In: *Lecture notes* 3.4 (1998), p. 5.

[106] Anh Nguyen, Jason Yosinski, and Jeff Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, CVPR'15*. 2015, pp. 427–436.

[107] Quynh Nguyen. "On connected sublevel sets in deep learning". In: *International Conference on Machine Learning*. 2019, pp. 4790–4799.

[108] Quynh Nguyen and Matthias Hein. "The loss surface of deep and wide neural networks". In: *International conference on machine learning*. 2017, pp. 2603–2612.

[109] Maher Nouiehed and Meisam Razaviyayn. "Learning Deep Models: Critical Points and Local Openness". In: *6th International Conference on Learning Representations*. 2018.

[110] Uche Osahor and Nasser M Nasrabadi. "Ortho-Shot: Low Displacement Rank Regularization with Data Augmentation for Few-Shot Learning". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 2200–2209.

[111] Philipp Petersen and Felix Voigtlaender. "Optimal approximation of piecewise smooth functions using deep ReLU neural networks". In: *Neural Networks* 108 (2018), pp. 296–330. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2018.08.019.

[112] Philipp Petersen and Felix Voigtlaender. *Optimal learning of high-dimensional classification problems using deep neural networks*. arXiv:2112.12555. 2021. DOI: 10.48550/ARXIV.2112.12555.

[113] Lutz Prechelt. "Early stopping-but when?" In: *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[114] Haozhi Qi, Chong You, Xiaolong Wang, Yi Ma, and Jitendra Malik. "Deep isometric learning for visual recognition". In: *International Conference on Machine Learning, ICML'20*. 2020.

[115] Haifeng Qian and Mark N Wegman. "L2-Nonexpansive Neural Networks". In: *International Conference on Learning Representations, ICLR'18*. 2018.

[116] G. Raskutti, M. J. Wainwright, and B. Yu. "Early stopping for non-parametric regression: An optimal data-dependent stopping rule". In: *49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 2011, pp. 1318–1325. DOI: 10.1109/Allerton.2011.6120320.

[117] Noam Razin and Nadav Cohen. "Implicit Regularization in Deep Learning May Not Be Explainable by Norms". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 21174–21187.

[118] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: towards real-time object detection with region proposal networks". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems, NeurIPS'15*. 2015, pp. 91–99.

[119] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. "A mathematical theory of semantic development in deep neural networks". In: *Proceedings of the National Academy of Sciences* 116.23 (2019), pp. 11537–11546.

[120] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks". In: *2nd International Conference on Learning Representations*. 2014.

[121] Hanie Sedghi, Vineet Gupta, and Philip M Long. "The Singular Values of Convolutional Layers". In: *International Conference on Learning Representations, ICLR'18*. 2018.

[122] Mathieu Serrurier, Franck Mamalet, Alberto González-Sanz, Thibaut Boissin, Jean-Michel Loubes, and Eustasio Del Barrio. "Achieving robustness in classification using optimal transport with hinge regularization". In: *Conference on Computer Vision and Pattern Recognition (CVPR'21)*. 2021.

[123] Ohad Shamir. "Exponential convergence time of gradient descent for one-dimensional deep linear neural networks". In: *Conference on Learning Theory*. 2019, pp. 2691–2713.

[124] Evan Shelhamer, Jonathan Long, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence, TPAMI'17* 39.4 (2017), pp. 640–651.

[125] Zuowei Shen, Haizhao Yang, and Shijun Zhang. "Optimal approximation rate of ReLU networks in terms of width and depth". In: *Journal de Mathématiques Pures et Appliquées* 157 (2022), pp. 101–135. ISSN: 0021-7824. DOI: https://doi.org/10.1016/j.matpur.2021.07.009.

[126] Jonathan W. Siegel and Jinchao Xu. "Sharp Bounds on the Approximation Rates, Metric Entropy, and n-widths of Shallow Neural Networks". In: 2021.

[127] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations, ICLR'15*. 2015.

[128] Sahil Singla and Soheil Feizi. "Skew Orthogonal Convolutions". In: *Proceedings of the 38th International Conference on Machine Learning, ICML'21*. 2021.

[129] Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. "Robust large margin deep neural networks". In: *IEEE Transactions on Signal Processing* 65.16 (2017), pp. 4265–4280.

[130] Ju Sun, Qing Qu, and John Wright. "A geometric analysis of phase retrieval". In: *Foundations of Computational Mathematics* 18.5 (2018), pp. 1131–1198.

[131] Ju Sun, Qing Qu, and John Wright. "Complete dictionary recovery over the sphere I: Overview and the geometric picture". In: *IEEE Transactions on Information Theory* 63.2 (2016), pp. 853–884.

[132] Ruoyu Sun. "Optimization for deep learning: theory and algorithms". In: *arXiv preprint arXiv:1912.08957* (2019).

[133] Ruoyu Sun, Dawei Li, Shiyu Liang, Tian Ding, and Rayadurgam Srikant. "The global landscape of neural networks: An overview". In: *IEEE Signal Processing Magazine* 37.5 (2020), pp. 95–108.

[134] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. "Intriguing properties of neural networks". In: *International Conference on Learning Representations, ICLR'14*. 2014.

[135] Matus Telgarsky. "Benefits of depth in neural networks". In: *29th Annual Conference on Learning Theory*. Vol. 49. Proceedings of Machine Learning Research. 2016. arXiv: 1602.04485 [cs.LG].

[136] Tijmen Tieleman and Geoffrey Hinton. "Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning". In: *COURSERA Neural Networks Mach. Learn* (2012).

[137] Matthew Trager, Kathlén Kohn, and Joan Bruna. "Pure and Spurious Critical Points: a Geometric Study of Linear Networks". In: *International Conference on Learning Representations*. 2020.

[138] Asher Trockman and J Zico Kolter. "Orthogonalizing Convolutional Layers with the Cayley Transform". In: *International Conference on Learning Representations, ICLR'21*. 2021.

[139] Alexander Tsigler and Peter L Bartlett. "Benign overfitting in ridge regression". In: *arXiv preprint arXiv:2009.14286* (2020).

[140] Gal Vardi, Daniel Reichman, Toniann Pitassi, and Ohad Shamir. *Size and Depth Separation in Approximating Benign Functions with Neural Networks*. 2021. arXiv: 2102.00314 [cs.LG].

[141] Felix Voigtlander and Philipp Petersen. "Approximation in $L^p(\mu)$ with deep ReLU neural networks". In: *13th International conference on Sampling Theory and Applications (SampTA)*. arXiv:1904.04789. 2019.

[142] Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X Yu. "Orthogonal convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'20*. 2020, pp. 11505–11515.

[143] Yutong Wang and Clayton D. Scott. "VC dimension of partially quantized neural networks in the overparametrized regime". In: *International Conference on Learning Representations*. 2022.

[144] Lei Wu, Qingcan Wang, and Chao Ma. "Global convergence of gradient descent for deep linear residual networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 13389–13398.

[145] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. "Dynamical Isometry and a Mean Field Theory of CNNs: How to Train 10,000-Layer Vanilla Convolutional Neural Networks". In: *Proceedings of the 35th International Conference on Machine Learning,ICML'18*. 2018.

[146] Di Xie, Jiang Xiong, and Shiliang Pu. "All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'17*. 2017, pp. 6176–6185.

[147] Huan Xu and Shie Mannor. "Robustness and Generalization". In: *CoRR* abs/1005.2243 (2010). arXiv: 1005.2243.

[148] Huan Xu and Shie Mannor. "Robustness and generalization". In: *Machine learning* 86.3 (2012), pp. 391–423.

[149] Keiji Yanai, Ryosuke Tanno, and Koichi Okamoto. "Efficient Mobile Implementation of A CNN-Based Object Recognition System". In: *Proceedings of the 24th ACM International Conference on Multimedia, ACMMM'16*. 2016, pp. 362–366.

[150] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. "A Closer Look at Accuracy vs. Robustness". In: *arXiv:2003.02460 [cs, stat]* (2020).

[151] Yuhong Yang and Andrew Barron. "Information-theoretic determination of minimax rates of convergence". In: *The Annals of Statistics* 27.5 (1999), pp. 1564–1599. DOI: 10.1214/aos/1017939142.

[152] Dmitry Yarotsky. "Error bounds for approximations with deep ReLU networks". In: *Neural Networks* 94 (2017), pp. 103–114.

[153] Dmitry Yarotsky. "Optimal approximation of continuous functions by very deep ReLU networks". In: *Proceedings of the 31st Conference On Learning Theory*. Ed. by Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet. Vol. 75. Proceedings of Machine Learning Research. 2018, pp. 639–649.

[154] Dmitry Yarotsky and Anton Zhevnerchuk. "The phase diagram of approximation rates for deep neural networks". In: *Advances in neural information processing systems* 33 (2020), pp. 13005–13015.

[155] Bin Yu. "Assouad, Fano, and Le Cam". In: *Festschrift for Lucien Le Cam: Research Papers in Probability and Statistics*. Ed. by David Pollard, Erik Torgersen, and Grace L. Yang. New York, NY: Springer New York, 1997, pp. 423–435. ISBN: 978-1-4612-1880-7. DOI: 10.1007/978-1-4612-1880-7_29.

[156] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. "Global Optimality Conditions for Deep Neural Networks". In: *International Conference on Learning Representations*. 2018.

[157] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning (still) requires rethinking generalization". In: *Communications of the ACM* 64.3 (2021), pp. 107–115.

[158] Guoqiang Zhang, Kenta Niwa, and W Bastiaan Kleijn. "Approximated Orthonormal Normalisation in Training Neural Networks". In: *arXiv preprint arXiv:1911.09445* (2019).

[159] Junming Zhang, Ruxian Yao, Wengeng Ge, and Jinfeng Gao. "Orthogonal convolutional neural networks for automatic sleep stage classification based on single-channel EEG". In: *Computer Methods and Programs in Biomedicine* (2020).

[160] Tong Zhang and Bin Yu. "Boosting with early stopping: Convergence and consistency". In: *Ann. Statist.* 33.4 (2005), pp. 1538–1579. DOI: 10.1214/009053605000000255.

[161] Xiang Zhang, Junbo Zhao, and Yann Lecun. "Character-level convolutional networks for text classification". In: *Advances in Neural Information Processing Systems, NIPS'15* 2015 (2015), pp. 649–657.

[162] Yuqian Zhang, Qing Qu, and John Wright. "From symmetry to geometry: Tractable nonconvex problems". In: *arXiv preprint arXiv:2007.06753* (2020).

[163] Yi Zhou and Yingbin Liang. "Critical Points of Linear Neural Networks: Analytical Forms and Landscape Properties". In: *International Conference on Learning Representations*. 2018.

[164] Zhihui Zhu, Daniel Soudry, Yonina C Eldar, and Michael B Wakin. "The global optimization geometry of shallow linear neural networks". In: *Journal of Mathematical Imaging and Vision* 62.3 (2020), pp. 279–292.

[165] Difan Zou, Jingfeng Wu, Vladimir Braverman, Quanquan Gu, and Sham Kakade. "Benign overfitting of constant-stepsize sgd for linear regression". In: *Conference on Learning Theory*. PMLR. 2021, pp. 4633–4635.

Quelques contributions à la théorie de l'apprentissage profond: optimisation, robustesse, et approximation

Résumé: Dans cette thèse, nous étudions différents aspects théoriques de l'apprentissage profond, en particulier l'optimisation, la robustesse et l'approximation. Optimisation: Nous étudions le paysage d'optimisation du risque empirique des réseaux neuronaux linéaires profonds avec la perte des moindres carrées. Il est connu que, sous des hypothèses faibles, il n'y a pas de minimiseurs locaux non-globaux et pas de maximiseurs locaux. Cependant, l'existence et la diversité des points selle non-stricts, qui peuvent jouer un rôle dans la dynamique des algorithmes du premier ordre, n'ont été que peu étudiées. Nous fournissons une analyse complète du paysage d'optimisation à l'ordre 2. Nous caractérisons, parmi tous les points critiques, les minimiseurs globaux, les points-selles stricts et les points-selles non stricts. Nous énumérons toutes les valeurs critiques associées. La caractérisation est simple, elle implique des conditions sur les rangs des produits partiels de matrices, et éclaire la convergence globale ou la régularisation implicite qui ont été prouvées ou observées lors de l'optimisation de réseaux neuronaux linéaires. Au passage, nous fournissons une paramétrisation explicite de l'ensemble de tous les minimiseurs globaux et exposons de grands ensembles de points selle stricts et non stricts. Robustesse: Nous étudions les propriétés théoriques des couches convolutives orthogonales. Nous établissons des conditions nécessaires et suffisantes sur l'architecture de la couche garantissant l'existence d'une transformée convolutive orthogonale. Ces conditions prouvent que les transformées convolutives orthogonales existent pour presque toutes les architectures utilisées en pratique pour le padding "circulaire". Nous montrons également des limitations avec des conditions aux bords "valid" et des conditions aux bords "same" avec un zero-padding. Récemment, un terme de régularisation imposant l'orthogonalité des couches convolutives a été proposé, et des résultats empiriques impressionnants ont été obtenus dans différentes applications : (Wang et al. 2020). Nous faisons le lien entre ce terme de régularisation et les mesures d'orthogonalité. Ce faisant, nous montrons que cette stratégie de régularisation est stable vis-à-vis des erreurs numériques et d'optimisation et que, en présence de petites erreurs et lorsque la taille du signal/de l'image est grande, les couches convolutives restent proches de l'isométrie. Les résultats théoriques sont confirmés par des expériences et le paysage du terme de régularisation est étudié. Les expériences sur des jeux de données réels montrent que lorsque l'orthogonalité est utilisée pour renforcer la robustesse, le paramètre multipliant le terme de régularisation peut être utilisé pour régler un compromis entre la précision et l'orthogonalité, au profit de la précision et de la robustesse. Approximation: Nous étudions les limites fondamentales du pouvoir expressif des réseaux de neurones. Étant donné deux ensembles F, G de fonctions à valeurs réelles, nous prouvons d'abord une limite inférieure générale sur la façon dont les fonctions de F peuvent être approximées en norme L^p par des fonctions de G. La borne inférieure dépend du "packing number" de F, de l'étendue de F, et de la "fat-shattering dimension" G. Nous instancions ensuite cette borne au cas où G correspond à un réseau de neurones feedforward dont la fonction d'activation est polynomiale par morceaux, et décrivons en détail l'application à deux ensembles F : les boules de Hölder et les fonctions monotones multivariées. En plus de correspondre aux limites supérieures (connues ou nouvelles) à des facteurs logarithmiques près, nos limites inférieures éclairent les similitudes ou les différences entre l'approximation en norme L^p et en norme sup, résolvant ainsi une question ouverte par (DeVore et al. 2021).

Some contributions to deep learning theory: optimization, robustness, and approximation

Abstract: In this thesis, we study different theoretical aspects of deep learning, in particular optimization, robustness, and approximation.

Optimization: We study the optimization landscape of deep linear neural networks with the square loss. It is known that, under weak assumptions, there are no spurious local minima and no local maxima. However, the existence and diversity of non-strict saddle points, which can play a role in first-order algorithms' dynamics, have only been lightly studied. We go a step further with a full analysis of the optimization landscape at order 2. We characterize, among all critical points, which are global minimizers, strict saddle points, and non-strict saddle points. We enumerate all the associated critical values. The characterization is simple, involves conditions on the ranks of partial matrix products, and sheds some light on global convergence or implicit regularization that have been proved or observed

when optimizing linear neural networks. In passing, we provide an explicit parameterization of the set of all global minimizers and exhibit large sets of strict and non-strict saddle points.

Robustness: We study the theoretical properties of orthogonal convolutional layers. We establish necessary and sufficient conditions on the layer architecture guaranteeing the existence of an orthogonal convolutional transform. The conditions prove that orthogonal convolutional transforms exist for almost all architectures used in practice for 'circular' padding. We also exhibit limitations with 'valid' boundary conditions and 'same' boundary conditions with zero-padding. Recently, a regularization term imposing the orthogonality of convolutional layers has been proposed, and impressive empirical results have been obtained in different applications [142]. The second motivation is to specify the theory behind this. We make the link between this regularization term and orthogonality measures. In doing so, we show that this regularization strategy is stable with respect to numerical and optimization errors and that, in the presence of small errors and when the size of the signal/image is large, the convolutional layers remain close to isometric. The theoretical results are confirmed with experiments and the landscape of the regularization term is studied. Experiments on real datasets show that when orthogonality is used to enforce robustness, the parameter multiplying the regularization term can be used to tune a tradeoff between accuracy and orthogonality, for the benefit of both accuracy and robustness. Altogether, the study guarantees that the regularization proposed in [142] is an efficient, flexible and stable numerical strategy to learn orthogonal convolutional layers.

Approximation: We study the fundamental limits to the expressive power of neural networks. Given two sets F, G of real-valued functions, we first prove a general lower bound on how well functions in F can be approximated in $L^p(\mu)$ norm by functions in G, for any $p \geq 1$ and any probability measure μ. The lower bound depends on the packing number of F, the range of F, and the fat-shattering dimension of G. We then instantiate this bound to the case where G corresponds to a piecewise-polynomial feed-forward neural network, and describe in details the application to two sets F: Hölder balls and multivariate monotonic functions. Beside matching (known or new) upper bounds up to log factors, our lower bounds shed some light on the similarities or differences between approximation in L^p norm or in sup norm, solving an open question by DeVore et al. [31]. Our proof strategy differs from the sup norm case and uses a key probability result of Mendelson [100].