HAL
open science

Active learning and input space analysis for deep networks

Mélanie Ducoffe

To cite this version:

Mélanie Ducoffe. Active learning and input space analysis for deep networks. Neural and Evolutionary Computing [cs.NE]. COMUE Université Côte d'Azur (2015 - 2019), 2018. English. NNT : 2018AZUR4115 . tel-02271840

HAL Id: tel-02271840

https://theses.hal.science/tel-02271840v1

Submitted on 27 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Active Learning et Visualisation des données d'apprentissage pour les réseaux de neurones profonds

Mélanie DUCOFFE

CNRS, LABORATOIRE I3S, SOPHIA-ANTIPOLIS

PRÉSENTÉE EN VUE DE L'OBTENTION DU GRADE DE DOCTEUR EN INFORMATIQUE DE UNIVERSITÉ CÔTE D'AZUR

DIRIGÉE PAR: FREDERIC PRECIOSO

SOUTENUE LE : 12 DÉCEMBRE 2018

DEVANT LE JURY, COMPOSÉ DE:

FREDERIC PRECIOSO,
UNIVERSITÉ CÔTE D'AZUR
DIRECTEUR

KARIM LOUNICI,
UNIVERSITÉ CÔTE D'AZUR
PRÉSIDENT

PATRICK PEREZ,
VALEO
RAPPORTEUR

BJORN SCHULLER,
UNIVERSITY OF AUGSBURG
RAPPORTEUR

STÉPHANE CANU,
INSA ROUEN
EXAMINATEUR

JAKOB VERBEEK,
INRIA GRENOBLE
EXAMINATEUR

MEMBRE DE UNIVERSITÉ CÔTE D'AZUR

Active Learning et Visualisation des données d'apprentissage pour les réseaux de neurones profonds

Jury

Président du jury

- Karim LOUNICI, Université Côte d'Azur

Rapporteurs

- Patrick PEREZ, VALEO
- Bjorn SCHULLER, University of Augsburg

Examinateurs

- Stéphane CANU, INSA Rouen
- Jakob VERBEEK, INRIA Grenoble

Titre : *Active Learning et Visualisation des données d'apprentissage pour les Réseaux de Neurones Profonds*

Résumé

Notre travail est présenté en trois parties indépendantes.

Tout d'abord, nous proposons trois heuristiques d'apprentissage actif pour les réseaux de neurones profonds :

• Nous mettons à l'échelle le 'query by committee' , qui agrège la décision de sélectionner ou non une donnée par le vote d'un comité. Pour se faire nous formons le comité à l'aide de différents masque de dropout.

• Un autre travail se base sur la distance des exemples à la marge. Nous proposons d'utiliser les exemples adversaires comme une approximation de la dite distance. Nous démontrons également des bornes de convergence de notre méthode dans le cas de réseaux linéaires.

• Puis, nous avons formulé une heuristique d'apprentissage actif qui s'adapte tant au CNNs qu'aux RNNs. Notre méthode sélectionne les données qui minimisent l'énergie libre variationnelle.

Dans un second temps, nous nous sommes concentrés sur la distance de Wasserstein. Nous projetons les distributions dans un espace où la distance euclidienne mimique la distance de Wasserstein. Pour se faire nous utilisons une architecture siamoise. Également, nous démontrons les propriétés sous-modulaires des prototypes de Wasserstein et comment les appliquer à l'apprentissage actif.

Enfin, nous proposons de nouveaux outils de visualisation pour expliquer les prédictions d'un CNN sur du langage naturel. Premièrement, nous détournons une stratégie d'apprentissage actif pour confronter la pertinence des phrases sélectionnées aux techniques de phraséologie les plus récentes. Deuxièmement, nous profitons des algorithmes de déconvolution des CNNs afin de présenter une nouvelle perspective sur l'analyse d'un texte.

Mots clés

Apprentissage Actif, Wasserstein, Linguistique, Réseaux de neurones profonds, Déconvolution, Automatisation, Exemple Adversaire

Title : Active Learning and Input Space Analysis for Deep Networks

Abstract

Our work is presented in three separate parts which can be read independently.

Firstly we propose three active learning heuristics that scale to deep neural networks:

• We scale query by committee, an ensemble active learning methods. We speed up the computation time by sampling a committee of deep networks by applying dropout on the trained model.

• Another direction was margin-based active learning. We propose to use an adversarial perturbation to measure the distance to the margin. We also establish theoretical bounds on the convergence of our Adversarial Active Learning strategy for linear classifiers.

• We also derive an active learning heuristic that scales to both CNN and RNN by selecting the unlabeled data that minimize the variational free energy.

Secondly, we focus our work on how to fasten the computation of Wasserstein distances. We propose to approximate Wasserstein distances using a Siamese architecture. From another point of view, we demonstrate the submodular properties of Wasserstein medoids and how to apply it in active learning.

Eventually, we provide new visualization tools for explaining the predictions of CNN on a text. First, we hijack an active learning strategy to confront the relevance of the sentences selected with active learning to state-of-the-art phraseology techniques. These works help to understand the hierarchy of the linguistic knowledge acquired during the training of CNNs on NLP tasks. Secondly, we take advantage of deconvolution networks for image analysis to present a new perspective on text analysis to the linguistic community that we call Text Deconvolution Saliency.

Keywords

Deep Learning, Active Learning, Wasserstein, Linguistic, NLP, CNN, Deconvolution, Adversarial example

Remerciements

Je remercie d'abord les membres de mon jury de thèse. Un grand merci en particulier à Patrick Perez et Björn Schuller pour avoir accepté de rapporter ma thèse. Mais également je remercie tous les membres du jury, Karim Lounici, Stéphane Canu, et Jakob Verbeek. C'est un honneur pour moi de présenter mes travaux devant des chercheurs dont l'excellence de leur recherche, mais également leur bienveillance ont été des sources d'inspirations durant mes trois années de doctorat. Je n'oublie pas également d'adresser un grand merci à Igor Litovsky, Johnny Bond et Christophe Papazian qui furent les premiers à me pousser vers la recherche. Je remercie également l'ENS Cachan-Rennes et les directeurs de leur département informatique, David Pichardie et Luc Bougé qui ont permis le financement de cette thèse. Mes remerciements vont de même à Nicolas Nisse, Rémi Gribonval, Anatole Lécuyer et Yoshua Bengio pour m'avoir encadré dans mes stages de recherche. Dans ce contexte, je remercie tout particulièrement Yoshua Bengio et Pascal Vincent, professeur au MILA pour m'avoir enseignée énormément de concepts sur l'apprentissage automatique. Si je suis capable aujourd'hui de soutenir ma thèse, c'est en partie grâce à ce que j'ai appris à leurs côtés.

Un très grand merci du fond du cœur à mon encadrant, Frédéric Precioso, pour m'avoir fait confiance. Merci à toi pour t'être toujours rendu disponible malgré un emploi du temps plus que chargé et pour tes conseils judicieux, tant humains que scientifiques. Merci également pour les nombreuses opportunités auxquels tu m'as fait accéder. Aucun remerciement ne pourra correctement exprimer ma reconnaissance pour ton investissement.

Merci aux collaborateurs de cette thèse, en particulier, Rémi Flamary, Nicolas Courty, Damon Mayaffre et Laurent Vanni. Rémi a été la plus belle rencontre que j'ai pu faire pendant cette thèse. Il a choisi de me faire confiance et de m'enseigner tout un pan du machine learning qui m'était alors inconnu. Toi et Nicolas êtes deux chercheurs et deux mentors incroyables, et je croise les doigts pour que dans l'avenir nous puissions encore travailler ensemble ! Damon, je n'aurais jamais une prose aussi élaborée que la tienne pour te remercier, mais le cœur y est. Collaborer avec toi et Laurent fût extrêmement enrichissant, et j'espère pouvoir continuer à travailler dans l'interdisciplinarité avec vous deux et Frédéric.

De manière plus indirecte, je tiens à remercier Mireille Borne-fontaine, Lucile Sassatelli, Diane Lingrand, Anne-Marie Pinna-Dery, Asja Fischer, Samira Shabanian, Negar Rostamzadeh et Joëlle Pineau. Être une femme dans un milieu majoritairement composé d'hommes n'est jamais quelque chose d'aisé, et certaines anecdotes à ce sujet auraient pu me détourner de ce milieu si jamais je n'avais pu me rassurer par votre réussite professionnelle et votre persévérance à contrer les tendances. Telle n'était pas votre intention, mais vous m'avez servi de modèles et je vous en remercie.

Que serait une thèse académique sans les collègues doctorants et chercheurs dont la gentillesse et la bonhomie ont largement contribué à me faire garder le sourire. En toute honnêteté, je devrais aussi remercier les producteurs de café et

de sucreries qui furent ma principale source d'alimentation. Je remercie mes co-bureaux: Xheva, qui m'a accueilli avec joie en pleine rédaction de thèse, puis Melissa qui a fait passer les heures de travail plus vite grâce à nos discussions. Merci au trio infernal Romaric, Yoann et Stéphanie, les irréductibles de la pause-déjeuner et la pause goûter, mais également à tous ceux qui m'ont tenu compagnie quand je pointais le bout de mon nez hors du bureau: Geoffrey, Gérald, Thomas, Franck(s), Ben, Philippe(s), Stéphane, Jean-Yves, Anne Marie, Marco, Sébastien, Sami, Mehdi, Claude, Hélène, et à tous les autres que j'aurai oublié de mentionner... Merci à Ali Beikbaghban qui depuis mon entrée à Polytech a toujours été bienveillant avec moi et le reste de la tribu Ducoffe. Je remercie également la merveilleuse Magali Richir, qui m'a toujours aidé avec diligence et gentillesse dans l'enfer administratif qui pave la vie d'un doctorant. Merci également aux collègues étrangers - entre autres Chiheb Trabelsi, Thomas Mesnard, Guillaume Alain, Benjamin Scellier - dont les rencontres annuelles aux conférences et écoles d'été m'ont fourni une motivation supplémentaire pour mes publications. Des remerciements particuliers à Christelle et Denis, mes 'parents adoptifs'. Même si cette expression relevait avant tout de la blague, vous êtes un couple merveilleux et très attachant, je n'aurai pas été peu fière d'avoir fait partie de votre famille. De manière plus générale, je remercie tous les membres de l'équipe SPARKS. Merci à tous pour les moments de joie, de partage et de complicité !

Parmi les supers enseignants avec qui j'ai eu la chance de travailler, je remercie tout particulièrement Diane Lingrand. Merci à toi pour ton implication et ta considération, mais aussi pour avoir su allier travail et humour. Travailler à tes côtés fut extrêmement gratifiant d'un point de vue professionnel, mais aussi personnel.

Un immense merci à mes proches pour leur affection et leur soutien indéfectible. Des remerciements très spéciaux à mon frère et ma soeur, Guillaume et Anaïs, avec qui j'ai la chance d'être très proches, et qui m'ont notamment consolée après les rebuttals difficiles. Vous êtes les meilleurs frères et soeurs qu'on puisse avoir ! Guillaume, tu as été un moteur pendant cette thèse, ta passion pour la science et ton acharnement m'ont toujours poussé à donner le meilleur de moi-même. J'espère un jour être à ta hauteur ! Merci à Clément pour m'avoir fait réaliser que je n'étais pas mariée à ma thèse, et que la passion de la recherche n'est rien sans une vie personnelle épanouissante. Je souhaite qu'elle le soit longtemps à tes côtés. Je remercie également ma soeur de coeur Félicia, et mes amis proches Emilie, Axelle, Marie, Laura, Arnaud, Raphaël et Claudia pour avoir toujours cru en moi. Un grand merci au duo de choc Heck, David et Jonathan, pour leur humour et pour avoir introduit les soirées jeux. Obéron a tenu trois ans en thèse grâce à vous ! Merci à Jipsy, ma bouillotte personnelle qui dort tous les soirs sur mes jambes, et pour tous les ronronnements de mes chats, Mistoufle, Pelote, Kiki et Lolita. Merci bien sûr à mes parents, Dominique et Evelyne. Je n'ai réalisé que trop tardivement les sacrifices et les concessions que vous avez faits pour nos avenirs. Je ne mérite rien que vous ne m'ayez appris ou offert et je vous suis reconnaissante pour tout ce que vous avez fait pour moi. Enfin je fais un merci général à ma grand-mère, à ma belle-soeur Adriana, ainsi qu'au reste de mes amis et de ma famille.

De manière plus anecdotique, je remercie Cédric Villani dont les actions politiques cherchent à pérenniser l'avenir de l'apprentissage artificiel en France. J'aime mon pays et je suis heureuse de voir émerger de nouvelles opportunités qui me permettent de rester.

Contents

1 Introduction **1**
1.1 Contributions . . . 1
1.1.1 Active Learning for Deep Networks . . . 1
1.1.2 Learning Wasserstein Core-Sets . . . 2
1.1.3 Visualization and Active Analysis for Deep Networks on Linguistic tasks . . . 2
1.2 List of publications . . . 2

I Active Learning for Deep Networks **5**

2 Introduction **7**
2.1 Motivations . . . 7
2.2 Definitions . . . 8
2.2.1 Active Learning . . . 8
2.2.2 Related Research Areas . . . 11
2.3 Litterature . . . 12
2.3.1 Uncertainty . . . 12
2.3.2 Query-By-Committee . . . 13
2.3.3 Optimal Experimental Design . . . 13
2.3.4 Core-Set . . . 15
2.3.5 Expected Model Change . . . 15
2.3.6 Batch Active Learning . . . 17
2.4 Theoretical Justification of Active Learning for Deep Networks . . . 18

3 Dropout Query-By-Committee **23**
3.1 Introduction . . . 23
3.2 Sampling a committee with Dropout . . . 25
3.3 Disagreement Scoring Function . . . 27
3.4 Empirical Validation . . . 28
3.5 Conclusion . . . 29

4 Adversarial Active Learning **31**
4.1 Introduction . . . 32
4.2 Margin-Based Active Learning for Deep Networks . . . 34
4.3 Empirical Validation . . . 36
4.3.1 Dataset and hyperparameters . . . 36
4.3.2 Evaluation . . . 37
4.3.3 Comparative study between DFAL and CORE-SET . . . 40
4.4 Transferability . . . 41

4.5 Discussion . . . 44
4.5.1 Theoretical motivations . . . 44
4.5.2 DFAL does not select random samples in the first runs . . . 44
4.6 Adversarial Active Learning for Linear Classifiers . . . 46
4.6.1 Transferable adversarial attacks . . . 46
4.6.2 Label Complexity on the unit ball . . . 49
4.7 Conclusion . . . 50

5 Perspective: Bayesian Active Learning through Laplace Approximation 51
5.1 Introduction . . . 51
5.1.1 Active Learning under the light of Variational Inference . . . 53
5.2 Covering . . . 56
5.2.1 Optimal Experimental Design . . . 56
5.2.2 Increasing the diversity . . . 57
5.3 Application to CNN . . . 58
5.4 Application to RNN . . . 59
5.5 Empirical Validation . . . 61
5.6 Future work . . . 63
5.7 Conclusion . . . 64

6 Conclusion 65

II Learning Wasserstein Core-Sets 67

7 Introduction 69
7.1 Motivations . . . 69
7.2 Definitions . . . 70
7.3 Litterature . . . 71
7.3.1 Fast approximation of the exact Wasserstein distance . . . 71
7.3.2 Metric embedding . . . 73
7.3.3 Domain adaptation . . . 73
7.3.4 Wasserstein Core-Sets for Lipschitz Costs . . . 74
7.3.5 Herding . . . 75

8 Learning Wasserstein embeddings 77
8.1 Introduction . . . 77
8.2 Wasserstein learning and reconstruction with siamese networks . . . 78
8.3 Empirical Validation . . . 80
8.3.1 Wasserstein data mining in the embedded space . . . 82
8.4 Future work: Wasserstein for Text Mining . . . 87
8.4.1 Information Retrieval: Fast computing of WMD at large scale 88
8.4.2 Text Generation given Wasserstein Distance . . . 90
8.5 Conclusion . . . 92

9 Perspective: Wasserstein prototypes **95**
9.1 Introduction . . . 95
9.2 Approximate Submodularity for Wasserstein distance . . . 96
9.2.1 Greedy selection of Prototypes . . . 98
9.3 Empirical Validation . . . 99
9.4 Active Learning . . . 102
9.5 Conclusion . . . 103

10 Conclusion **105**

III Understanding the behavior of Neural Networks on Linguistic tasks **107**

11 Introduction **109**
11.1 Motivation . . . 109
11.2 Litterature . . . 110
11.2.1 CNNs for Text classification . . . 110
11.2.2 Visualization of Deep network . . . 111
11.2.3 Model Agnostic Explanation . . . 112
11.2.4 Adversarial example for NLP . . . 112

12 Deconvolution for Text Analysis **115**
12.1 Introduction . . . 115
12.2 CNNs for Text Classification . . . 116
12.3 Deconvolution . . . 117
12.4 Experiments . . . 118
12.5 Z-score Versus Activation-score . . . 118
12.5.1 Dataset: English . . . 120
12.5.2 Dataset: French . . . 121
12.5.3 Dataset: Latin . . . 123
12.6 Conclusion . . . 125

13 Perspective: Active Learning for Linguistic Analysis **127**
13.1 Introduction . . . 127
13.2 Methodology . . . 128
13.3 Analysis under the light of Phraseology expertise . . . 129
13.4 Future works: Analysis with Model Agnostic Explanations . . . 131
13.5 Conclusion . . . 131

14 Conclusion **133**

IV Conclusion **135**
14.1 Perspectives . . . 137

14.2 Conclusion 137

A Appendix **147**
A.1 Dataset 148
A.2 Hyperparameters 151
A.2.1 Dropout Query-By-Committee 151
A.2.2 Adversarial Active Learning 151
A.2.3 Bayesian Active Learning through Laplace Approximation . . 152
A.2.4 Learning Wasserstein embeddings 152
A.2.5 Deconvolution for Text Analysis 153
A.3 Proofs 154
A.3.1 Adversarial Active Learning 154
A.3.2 Bayesian Active Learning through Laplace Approximation . . 157
A.3.3 Monotony 158
A.3.4 Wasserstein prototypes 160

Bibliography **165**

Chapter 1

Introduction

Contents

1.1 Contributions **1**

1.1.1 Active Learning for Deep Networks 1

1.1.2 Learning Wasserstein Core-Sets 2

1.1.3 Visualization and Active Analysis for Deep Networks on Linguistic tasks 2

1.2 List of publications **2**

1.1 Contributions

Our work is presented in three separate parts which can be read independently. We present their content in Parts 1.1.1, 1.1.2, 1.1.3 respectively. For each topic, we dedicate a thorough introduction. Full papers and code can be found online[1], and are also referred in Section 1.2.

1.1.1 Active Learning for Deep Networks

Part I is adressing the question of the annotation cost when training deep neural networks. Considering the cost of gathering relevant annotations for huge datasets such as ImageNet, the interest in methods requiring smaller training sets is increasing. One possible direction to improve a training set while reducing its size is to rely on active learning. In active learning, the goal is to train a classifier with as few as possible training samples while reaching the same accuracy as if an unlimited number of training samples were available (i.e., at most the whole dataset). The challenge lies in selecting a small subset of data, without supervision, which is informative enough to reach the best possible accuracy. In Sections 3 to 4, we scale active learning methods mostly designed for Convolutional Neural Network (CNN).

First, active query strategies may be handled by ensembling deep networks; either by disagreement over the models (*Query By Committee*: [Ducoffe 2015], Chapter 3), or by assuming some weight's distribution and sample a committee according to this distribution (*Bayesian Active Learning*: [Ducoffe 2016c], Chapter 5).

[1] github/mducoffe

Another direction is to rely on the distance to the decision boundary, namely margin-based active learning. In Chapter 4, we exploit the geometric distances of samples to the decision boundaries for querying new samples. We propose to use an adversarial perturbation to measure the distance to a CNN's decision boundary in [Ducoffe 2018]. We also establish theoretical bounds on the convergence of our Adversarial Active Learning strategy for linear classifiers.

1.1.2 Learning Wasserstein Core-Sets

Part II is focusing on Wasserstein distance. Wasserstein is a distance between distributions derived from the field of optimal transport. It has received a lot of attention in machine learning recently, notably with Wasserstein based generative models. However, its complexity limits the usage of Wasserstein in new applications. To alleviate the cost of computing pairwise Wasserstein distance on discrete distributions, we propose in Chapter 8 to approximate Wasserstein distances using a Siamese architecture ([Courty 2017b]). From another point of view, we demonstrate the submodular properties of Wasserstein medoids in Chapter 9 and how to apply it in active learning in Section 5.2.

1.1.3 Visualization and Active Analysis for Deep Networks on Linguistic tasks

We dedicate Part III to the conception of new visualization tools for the underlying information captured by a CNN on a text.

First, we hijack our active learning strategy from Chapter 3 to confront the relevance of the sentences selected with active learning to state-of-the-art phraseology techniques [Ducoffe 2016a, Mayaffre 2017]. These works help to understand the hierarchy of the linguistic knowledge acquired during the training of CNNs on Natural Language Processing (NLP) tasks.

Secondly, [Vanni 2018] confronts Textual Data Analysis and Convolutional Neural Networks for text analysis. We take advantage of deconvolution networks for image analysis to present a new perspective on text analysis to the linguistic community that we call Text Deconvolution Saliency (TDS), in Chapter 12.

1.2 List of publications

1. **QBDC: Query by dropout committee for training deep supervised architecture** [Ducoffe 2015]

2. **Adversarial Active Learning for Deep Networks: a Margin Based Approach** [Ducoffe 2018]

3. **Learning Wasserstein Embeddings** [Courty 2018]

4. **Introducing Active Learning for CNN under the light of Variational Inference** [Ducoffe 2016c]

5. **Machine Learning under the light of Phraseology expertise** [Ducoffe 2016a]

6. **Textual Deconvolution Saliency (TDS): a deep tool box for linguistic analysis** [Vanni 2018]

7. **Les mots des candidats, de "allons" à "vertu"** [Mayaffre 2017]

8. **Scalable batch mode Optimal Experimental Design for Deep Networks** [Ducoffe 2016b]

Part I

Active Learning for Deep Networks

Chapter 2

Introduction

Contents

2.1 Motivations **7**
2.2 Definitions **8**
2.2.1 Active Learning 8
2.2.2 Related Research Areas 11
2.3 Litterature **12**
2.3.1 Uncertainty 12
2.3.2 Query-By-Committee 13
2.3.3 Optimal Experimental Design 13
2.3.4 Core-Set 15
2.3.5 Expected Model Change 15
2.3.6 Batch Active Learning 17
2.4 Theoretical Justification of Active Learning for Deep Networks **18**

2.1 Motivations

Larger deep architectures fed with more data provide better results in error rate. This widely acknowledged idea has been confirmed all along the recent years when analyzing, for instance, the results at Imagenet Large Scale Visual Recognition Challenge [Russakovsky 2015]. Indeed, in 2012, the winner was the SuperVision team [Krizhevsky 2012] using a deep convolutional neural network with 60 million parameters and making a momentous breakthrough in the image classification task. The huge step forward from SuperVision team has profoundly impacted the following contributions to ILSVRC after 2012. In 2014, Simonyan *et al.* [Simonyan 2014] also proposed to use a CNN architecture from 11 up to 19 layers with 133 up to 144 million of parameters. Owing to the considerable amount of parameters involved which needs to be learned, the training set needs to be huge as well. Nevertheless, state of the art results using deep networks are known on a large training set.

However, a lot of real-life scenarios typically do not come with millions of labeled data available to train a model. Labeling appears to be one of the main bottlenecks towards wide spreading deep networks to a new area: gathering and annotating

massive dataset for supervised learning may prohibit the expansion of deep networks towards new fields such as chemistry or medicine [Smith 2018, Hoi 2006].

Labeling data may sound like a trivial task, but in many cases, it requires expert knowledge. For example, the creation of the Penn Treebank dataset, a benchmark when considering part-of-speech tagging, took more than seven years of collaborations with linguistic experts [Taylor 2003]. Scaling the labeling process is not always practical as it requires the intervention of many specific human operators. Usually, labeling can only be solved with coffee and patience.

A more plausible solution is to reduce the compelling need for labeled training samples to train deep networks. In Section 2.2, we highlight the different families of methods seeking to solve this type of problem and detail the settings on which they are better suited. In particular, we will focus on Active Learning (**AL**) that seeks to optimize the training set automatically for the task at hand to limit the need for human annotations.

2.2 Definitions

2.2.1 Active Learning

Given a large set of unlabeled samples, **AL** tries to guess which ones should be labeled and added to the training set to increase at best the performance of your model. It operates iteratively, by first requesting new labels from the user, and then updating the model given the new labeled training set. The model can leverage its new knowledge to add queries again. Eventually, only a small fraction of the unlabeled data would be annotated to achieve good classification performance. Firstly, we detail three scenarios in which may occur **AL**. Note that this list is not intended to be exhaustive. Eventually, **AL** may occur in different scenarios:

- **Pool-Based Sampling** assumes that the learner has only access to a fixed pool $\mathcal{P}$ of unlabeled i.i.d samples **(C)** and must query a fixed budget size number of points **(D)** from $\mathcal{P}$. It submits each of those queries to an oracle **(E)** that labels them to add them into the labeled training set **(A)**. The classifier **(B)** can then be re-trained on the incremented labeled training set. Figure 2.1 illustrates the iterative process. Notice that, in our context, we assume that the oracle makes no mistakes when annotating new query. However, noisy oracles have been tackled in the literature. For a survey of noisy oracles, we refer the reader to [Settles 2011].

- **Stream-Based Selective Sampling** considers one unlabeled example at a time and for each of them decide whether to ignore it or ask an oracle to annotate it. Stream-based active learning is attractive in many real-world applications when unlabeled samples are presented sequentially, and their number is far too large to maintain a pool of candidates. For example, stream-based active learning may be suitable for the classification of observations by au-

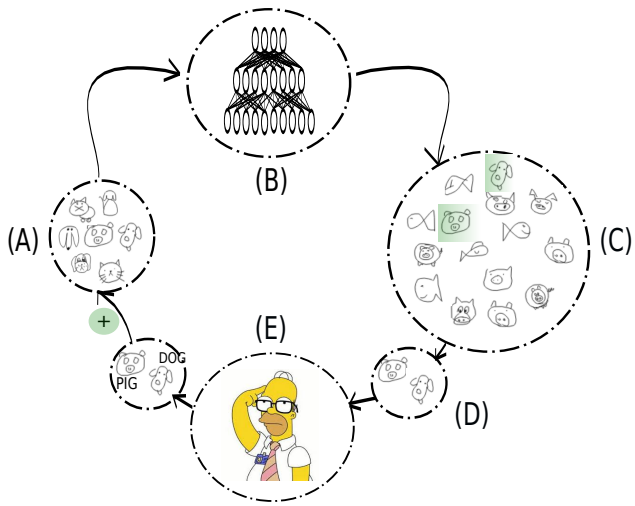
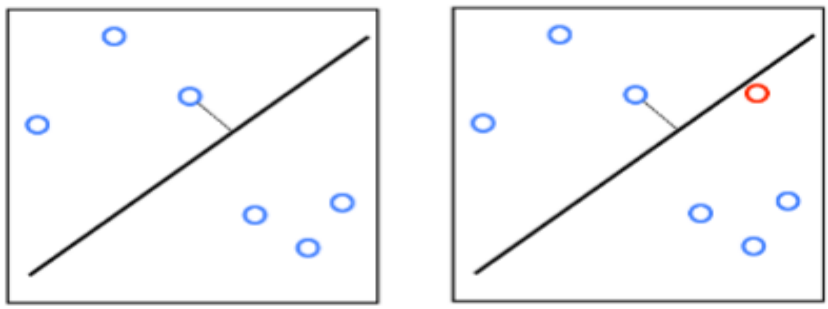


Figure 2.1: **Illustration of Pool-Based Active Learning**: **A**=Labeled training set ; **B**=Classifier ; **C**=Unlabeled set ; **D**=Queries ;**E**=Oracle

tonomous car driving.

- **Query synthesis** consists in generating new unlabeled instances instead of considering a fixed pool of unlabeled samples. Creating de novo the queries may increase the learning speed, as we can optimize the queries according to the query selection. We illustrate this phenomenon with a simple example in Figure 2.2.

  However querying arbitrary instances can be awkward if there are no assumptions on the underlying distribution of the generated samples; in that case, we may generate noisy instances. For example, [Baum 1992] synthesized handwritten digits to train a neural network. However, they obtained poor performance as sometimes their generated queries were not identifiable to the human oracle. Early active query synthesis has encountered some success when considering very low dimensional domains. More recently, [Zhu 2017] proposed to use a pre-trained Generative Adversarial Network (GAN) to generate the queries. So far, they obtain competitive results with pool-based active learning, probably due to a lack of diversity in their criterion. Indeed, when sampling from a finite set, the optimization of the query selection criterion is limited by the number of samples and their distribution. Eventually, one should pay attention not to focus on a subregion of the underlying distribution and create bias in the labeled training set. While their method is interesting, it comes at the price of training a GAN in a preprocessing step, which remains a challenging task. Nevertheless, query synthesis is also used as a form of reinforcement learning to improve dialogue generation: after the training phase, a human oracle scores the generated answer, which helps to



(a) SVM_{active} (b) Query synthesis

Figure 2.2: "*An active learning heuristics for SVM* SVM_{active} *will query data that are the closest to the decision boundary. In that tendency, Query synthesis may help to generate queries that optimize the active learning criterion*" as underlined in 2.2(b) [Zhu 2017]

improve the system in [Asghar 2017].

Active learning is not only motivated by theoretical works demonstrating that one model may perform better using less labeled data if the data are model-crafted [Cohn 1996], but also by its proven efficiency on a wide range of machine learning procedures, including character recognition [Liu 2004], bio-informatics [Sculley 2007, Smith 2010], or classification of medical data [Hoi 2006]. As an example, in Figure 2.3, we illustrate the potential benefit of **AL** on a baby task.

A central challenge in active learning is to define the information required for selecting at best the queries and how to measure it effectively. It happens that **AL** may have a drastic improvement regarding human annotations: in some classification problems, the excess risk of **AL** can converge to zero with an exponential rate comparing with the linear rate of fully supervised classification, also known as *passive learning*. However, the effectiveness of **AL** implies prior knowledge on the data distribution [Willett 2006, Castro 2007]. Eventually, there exists no universal criterion to select the most informative queries. Thus **AL** strategies rely on heuristics to choose these queries. Moreover, several heuristics coexist as it is impossible to obtain a universal active learning strategy effective for any given task [Dasgupta 2005a].

As underlined in the related research areas 2.2.2, applying active learning on deep networks appears promising. Indeed, in peculiar settings, supervised classification on random small labeled training set achieves similar accuracy than state-of-the-art semi-supervised deep algorithms. If one could optimize the labeled training set itself, it is likely that the performance would be even comparable, or even better.

However, transposing directly existing active learning on deep networks is not

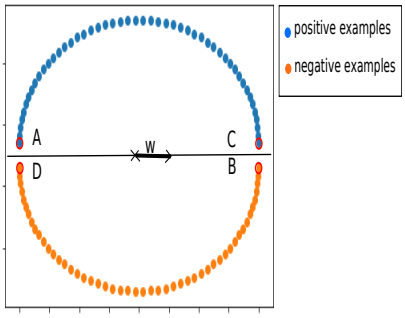


Figure 2.3: **AL on a toy data-set**. We consider a binary classification tasks on n=100 samples (• positive examples, • negative examples) and want to learn an optimal classifier with no bias. Because the data are well separated, there exist an optimal linear classifier (**W**) that can be learnt using at minima four samples: by labelling (**A**, **B**, **C**, **D**). However, the probability of sampling those points at once is of $\frac{1}{\binom{n}{4}} \simeq 10^{-4}$

intuitive. First of all, scaling them to high dimensional parameters networks may turn out to be intractable: some classic active learning methods such as Optimal Experiment Design [Yu 2006] require to inverse the Hessian matrix of the models at each iteration, which would be intractable for current standard CNNs. Secondly, one of the most standard strategies is to rely on the uncertainty measure. Uncertainty in deep networks is usually evaluated through the network's output while this is known to be misleading. Indeed, the discovery of adversarial examples has demonstrated that the way we measure uncertainty may be overconfident. We describe the query selection methods proposed for deep networks in section 2.3 and demonstrate how they compare to one another.

2.2.2 Related Research Areas

We define close related research areas, relevant as well to reduce the effective size of the labeled training set to train deep models. While those topics are out of the scope of our work, they appear complementary to **AL**.

1. **Transfer Learning (TL)** consists in using a solution designed for a related source domain, in order to adapt it to the current problem. Usually, **TL** is used only when few samples from the target domain are available, a.k.a we do not have at our disposal a large set of unlabeled examples; which differs from AL. Nevertheless, **TL** is only applicable when both source and target domains share some relevant information; while defining the type of information required is not intuitive and typically induces several empirical experiments.

   Transfer Learning is popular in deep learning due to a large number of pre-trained networks available online. When it comes to **TL** on deep networks,

it mainly consists in fitting an already trained model to a new classification task on another dataset [Sawada 2017]. Indeed, it appears that deep neural networks trained on image classification tasks, all learn similar and broadly general features in their first layers (visually similar to Gabor filters and color blobs, i.e., biologically receptive fields [Zamir 2018]). Hence, as those features are not dataset-crafted, they may be reused for another task to speed up the training: the scope of solutions would be narrowed by starting from a weight's region different from the common one obtained with random initialization.

2. **Semi-Supervised Learning (SSL)**: combines both a small labeled training set and a larger pool of unlabeled samples. When it comes to deep networks, **SSL** enjoys an extensive literature, ranging from extending autoencoders and generative modeling [Kingma 2014, Gogna 2016], to new regularization schemes [Miyato 2017]. Note that the previous listing is far from being exhaustive.

   However, the new flaws underlined in [Oliver 2018] should leverage the successes of semi-supervised deep algorithms. The first drawback is that when using well-optimized hyperparameters and regularization settings, fully-supervised deep networks are competitive with the current state-of-the-art semi-supervised algorithms, without using any unlabeled samples. Nevertheless, the size of the labeled training set and also, the divergence between both distributions, respectively induced by the labeled samples and the unlabeled samples, profoundly impact the performance of **SSL** on classification tasks. Finally, both drawbacks highlight the necessity of optimizing the labeled training set for the task at hand.

2.3 Litterature

Previous works have shown that a carefully designed query strategy effectively reduces annotation effort required in a variety of tasks for shallow models. The effectiveness of **AL** has been established both theoretically and empirically. Nevertheless, **AL** for shallow models mainly rely on specific model simplifications and closed form solution. Deep Neural Networks, on another side, are inherently complex non-linear functions. Their complexity poses several limitations to scale such existing active learning strategies.

In this section, we establish the range of active learning methods studied for deep networks, starting from the most intuitive setup (*uncertainty estimation*) to the most sophisticated strategies that take into account some properties and specificities involved in the training of deep networks.

2.3.1 Uncertainty

Originally, [Lewis 1994] introduces uncertainty selection. It consists in querying the annotations for the unlabeled samples with the lowest confidence. Thus its cost

is low and its setup simple. Hence, it has been used on deep networks for various tasks, ranging from sentiment classification to visual question answering and Named Entity Recognition [Zhou 2010, Lin 2017a, Yanyao Shen 2018].

The main drawback of uncertainty selection is its tendency to query outliers or other types of noisy instances, such as *adversarial examples*. Tellingly, the apparition of adversarial attacks, which are wrongly predicted with high confidence, empirically demonstrates that the probability of misclassification and the uncertainty are not necessarily correlated.

Uncertainty selection has been improved in a pseudo-labeling method called CEAL [Wang 2016]: CEAL performs uncertainty selection, but also adds highly confident samples into the augmented training set. The labels of these samples are not queried but inferred from the network's predictions. In the case one deals with a highly accurate network, CEAL will improve the generalization accuracy. However, CEAL implies new hyperparameters to threshold the prediction's confidence. If such a threshold is poorly tuned, it will corrupt the training set with mistaken labels.

2.3.2 Query-By-Committee

Uncertainty selection may be also tailored to network ensemble, either by disagreement over the models (*Query-by-committee*, [Seung 1992]) or by sampling through the distribution of the weights (*Bayesian active learning*, [Kapoor 2007]). Query-by-Committee consists in maintaining a committee of models which represent the current set of consistent hypothesis. Whether to label or not a query is decided based on a vote among the committee members. Usually, the vote incorporates some disagreement information on the predicted labels. Figure 2.4 illustrates this process for linear classifiers. Recently, [Gal 2016b] demonstrated that dropout (and other stochastic regularization schemes) is equivalent to perform inference on the posterior distribution of the weights, enabling to leverage the cost of training and updating multiple models. Thus, dropout allows to sample an ensemble of models at test time: to perform *Dropout-Query-By-Committee* (Ducoffe *et al.* , [Ducoffe 2015]) or *Bayesian Active Learning* (Gal *et al.* , [Gal 2016b]). Gal *et al.* proceeded with a comparison of several active learning heuristics: among all the metrics, BALD-which maximizes the mutual information between predictions and model posterior consistently outperforms other metrics.

2.3.3 Optimal Experimental Design

From another theoretical point of view, Optimal Experimental Design (**OED**) is a field which takes interests in the Fisher information. Formally, the Fisher information is the expectation over the partial derivative of the log-likelihood function with respect to the parameters. The Fisher informations $\mathbb{I}$ reads:

$$\mathbb{I}_\Theta = \mathbb{E}_{x,y}\left[\nabla_\Theta l(f_\Theta(x), y)\nabla_\Theta l(f_\Theta(x), y)^T\right] \tag{2.1}$$

This measure is relevant because in a single parameter case, its inverse sets a

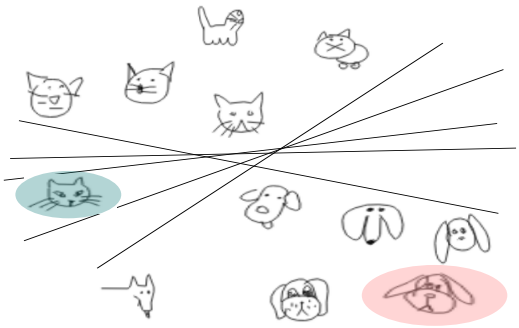


Figure 2.4: **Version space example for linear classifiers on a binary task.** Every hypothesis (—) is consistent with the labeled training set. However, each of them represent a different model in the version space. The unlabeled sample in red is not relevant as a query because every classifier agree in its prediction (*it is a dog*, whereas the green unlabeled sample is interesting as it will shrink the number of consistent classifiers. The more classifiers we discard, the faster QBC converge towards the optimal classifier for the task at hand (*assuming that a linear classifier can solve exactly the problem at hand*)

lower bound on the variance of the model's parameter estimates; this result is known as the Cramer Rao bound [Kagan 2001, Kay 2013].

In other words, to minimize the variance over its parameter estimates, an active learner should select data that maximize the Fisher information or minimize the inverse.

But for multivariate parameters, the Fisher information is a covariance matrix, so its maximization may go through several statistics.

We cite the three most popular scenarios (*other variants exist but, less used by the community, they are left unlisted for the sake of clarity*):

- A-optimality minimizes the trace of the inverse information matrix [Chan 1982]
- D-optimality minimizes the determinant of the inverse information matrix [Chaloner 1995]
- E optimality minimizes the maximum eigenvalue of the information matrix [Flaherty 2005]

Because deep neural networks may involve millions of parameters, computing their Fisher matrix is intractable. Moreover, even relying on approximations is too computationally expensive as one need to update the estimate for every possible query. Hence, **OED** has never been investigated for deep neural networks.

2.3.4 Core-Set

[Ozan Sener 2018] define the batch active learning problem as a covering problem on the output space of the network. In Equ. 2.2, they minimize the population risk of a model learned on a small labeled subset. To do so, they propose an upper bound with a linear combination of the training error, the generalization error and a third term denoted as the *core-set loss*. Notice that in Equ. 2.2, we denote by $\mathbf{s}$ the set of labeled points on which we train the parameter $\mathbf{w}$ of the network. We denote by $l(x, y \mid \mathbf{w}, \mathbf{s})$ the loss of the network over a sample x with label y.

$$\begin{aligned} \textit{Population Risk} &\leq \textit{Generalization Error} + \textit{Training Error} + \textit{CoreSetLoss} \\ \textit{Generalization Error} &\equiv \left|\mathbb{E}_{x,y\sim p_Z}[l(x_i, y_i \mid \mathbf{w}, \mathbf{s})] - \frac{1}{n}\sum_{i\in[n]} l(x_i, y_i \mid \mathbf{w}, \mathbf{s})\right| \\ \textit{Training Error} &\equiv \frac{1}{\mid \mathbf{s} \mid}\sum_{j\in\mathbf{s}} l(x_j, y_j \mid \mathbf{w}, \mathbf{s}) \end{aligned} \tag{2.2}$$

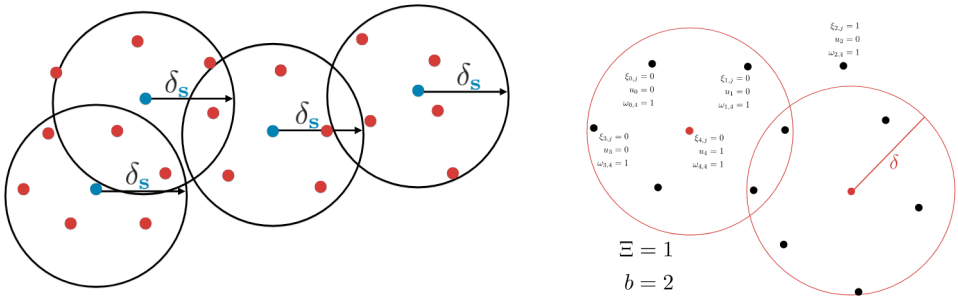
The generalization error is the absolute difference between the expectation of the loss considering every possible sample from the ground-truth distribution, and the expectation given the set of data available (meaning both labeled and unlabeled set, which are indexed by n in Equ. 2.2). On another side, the training error measures how far is the loss over both the unlabeled and labeled points compared to the loss evaluated only on the labeled points. Due to the expressive power of CNNs, the authors argue that the first two terms (training and generalization error) are negligible. Therefore the population risk would mainly be controlled by the core-set loss. Given a labeled training set $\mathbf{s}$, a model $\mathbf{w}$ trained on $\mathbf{s}$, and an unlabeled set of n points, the core-set loss is expressed in equation 2.3

$$\textit{CoreSetLoss} \equiv \left|\frac{1}{n}\sum_{i\in[n]} l(x_i, y_i \mid \mathbf{w}) - \frac{1}{\mid \mathbf{s} \mid}\sum_{j\in\mathbf{s}} l(x_j, y_j \mid \mathbf{w})\right| \tag{2.3}$$

The core-set loss consists in the difference between the average empirical loss over the set of points which are already labeled, and the average empirical loss over the entire dataset including unlabeled points. If not considering the labels, the core-set loss is upper bounded with the covering radius δ_s, as illustrated in Fig. 2.5. Here, we denote by covering radius, the maximum distance in the output space between any labeled sample's prediction and any unlabeled sample's prediction. Finally, Sener *et al.* used a MIP heuristic to minimize at best the covering radius of the training set. We illustrate their method in Fig. 2.5(b). Thanks to their method, they achieve state-of-the-art performance in active learning for image classification.

2.3.5 Expected Model Change

Another direction, rarely explored for deep networks, is to rely on the distance to decision boundaries, namely margin-based active learning. Assuming that the



(a) Covering the uncertainty space (b) MIP Solution

Figure 2.5: Consider the network's predictions of both the labeled training points $\mathbf{s}$ (•) and the unlabeled points (•). Sener *et al.* shows that if the labeled training set covers the space by a distance of at most $\delta_{\mathbf{s}}$ (*as illustrated in 2.5(a)*) then the *core-set loss* is bounded by $\mathcal{O}(\delta_{\mathbf{s}}) + \mathcal{O}(\frac{1}{n})$ with n denoting the number of points available. Eventually the strategy induced by this property implies to query unlabeled points that minimize at best the expected $\delta_{\mathbf{s}}$. To do so, Sener *et al.* developed a MIP strategy to select the samples that cover at best the output space of the network (*as illustrated in 2.5(b)*).

problem is separable with a margin is a reasonable requirement considered for many popular models such as SVM, Perceptron or AdaBoost. When positive and negative data are separable under SVM, [Tong 2001] have demonstrated the efficiency of picking the example which is the closest to the decision boundary. If, exploiting the geometric distances has been relevant for active learning on SVM [Tong 2001, Brinker 2003], it is not intuitive for CNNs since we do not know beforehand the geometrical shape of their decision boundaries. A first trial has been proposed in [Zhang 2017]. The Expected-Gradient-Length strategy (EGL) consists in selecting instances with a high magnitude gradient. Not only such samples will have an impact on the current model parameter estimates, but they will likely modify the shape of the decision boundaries. Their strategy aims to query samples that will impact at most the model. If one knows the ground-truth label in advance, then it would be possible to measure the exact impact of a sample (x_i, y_i) given the current labeled training set $\mathbf{s}$ and the weights $\mathbf{w}$ of the network:

$$x^* = \arg\max_{i \in [n]} ||\nabla l(\mathbf{s} \cup \{(x_i, y_i)\} | \mathbf{w})|| \tag{2.4}$$

However, computing the exact gradient for a given sample is intractable without its ground-truth label. In practice, we can only approximate Eq. 2.4 with the expectation over the gradients conditioned on every possible class assignments:

$$x^* = \arg\max_{i \in [n]} \sum_k P(y_i = k | x_i, \mathbf{w}) ||\nabla l(\mathbf{s} \cup \{(x_i, k)\} | \mathbf{w})|| \tag{2.5}$$

Finally, computing the gradient over the whole batch of data $\mathbf{s} \cup \{(x_i, k)\}$ may not be scalable depending on the size of the labeled data $\mathbf{s}$. Nevertheless, when training $\mathbf{w}$ on $\mathbf{s}$, we expect the magnitude of the gradient over the training set to be close to zero $||\nabla l(\mathbf{s}|\mathbf{w})|| \approx 0$ since the network has converged. Eventually, we can approximate the gradient over the whole set of data by the gradient over the unlabeled samples:

$$x^* = \arg\max_{i \in [n]} \sum_k P(y_i = k | x_i, \mathbf{w}) ||l((x_i, k) | \mathbf{w})|| \tag{2.6}$$

Similarly to uncertainty based selection, EGL may be limited because of an overparameterized network: parameters unused for classification are still taken into account into the EGL score. In that line, Zhang *et al.* argues that EGL should focus on instances that affect specific parameters of the networks, either the embedding space or the final softmax parameters.

2.3.6 Batch Active Learning

In the original setting, **AL** only queries one sample at a time. However, in many practical implementations, it is preferable to query labels for batches of examples in parallel instead of gathering them sequentially. Moreover, the training schemes for deep networks are most of the time working on batches of samples, thus we can expect that adding solely one example in the training set will not have any impact on the accuracy.

A possible solution is to select the samples with the top scores given the active learning heuristics in used. For example [Gal 2016b] selects the samples which maximize the mutual information. But top score is limited because it does not take into account the correlations among the samples. Similar examples will tend to have similar scores, but labeling all of them would not be efficient. To alleviate the sampling bias inherent in active learning heuristics, several works have combined their batch active learning framework with a diversity selection scheme to increase the representativeness of the training set. They either rely on statistical tests to measure the distribution difference, such as Maximum Mean Discrepancy (MMD) ([Wang 2015]), express the data subset selection for specific shallow classifiers as a constrained submodular maximization [Wei 2015, hoi] or rely on core-set approaches [Ozan Sener 2018].

A core-set of a data is a subset of the data, typically denoted as medoids, that are representative of the whole set of data given an informative criterion. It finds its root in computational geometry [Agarwal 2005] and have been widespread to the machine learning community first via importance sampling [Langberg 2010]. We further describe core-set approaches in Section 7.3. Furthermore, we also propose

a new diversity criterion that relies on a Wasserstein based core-set approach, in Chapter 9.

2.4 Theoretical Justification of Active Learning for Deep Networks

Recent works have focused on developing tighter upper bounds on the probability of misclassification of neural networks. From the seminal work of Vapnik and Chervonenkis, it is commonly accepted that the Vapnik Chervonenkis (VC) dimension plays a predominant role in the definition of an upper bound on the generalization error ($\mathbf{GE}(f_\theta)$) of any given classifier, [Blumer 1989]. Indeed, both experimental evidence and learning theory link the generalization of a classifier to the empirical error (*i.e.*, the error made on the training set) and the classifier capacity. When it comes to neural networks, their VC grows with their number of parameters, and highly depends on the number of hidden layers [Bartlett 2003, Bartlett 2017]. Hence, in a context of active learning, the VC dimension would favor shallower networks than the common architectures used in the deep learning community.

However, VC dimension is data independent, and thus may not be a tight upper bound to conclude to the potential benefits of active learning on deep networks. A possible solution to incorporate the nature of the input data is to rely on the Rademacher complexity [Neyshabur 2015]. The empirical Rademacher complexity of a hypothesis class $\mathcal{H}$ on a dataset $\{x_1, ..., x_n\}$ is defined as:

$$\tilde{\mathcal{R}}_n(\mathcal{H}) = \mathbb{E}_\sigma[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i)] \tag{2.7}$$

Where $\sigma_1, ..., \sigma_n \in \{\pm 1\}$ are iid uniform random variables. $\tilde{\mathcal{R}}_n(\mathcal{H})$ measures the ability of the classifiers among $\mathcal{H}$ to fit random binary labels assignment. However Rademacher complexity is not always tractable, and is upper bounded most of the time. For example, a data-independent upper bound has been proposed in [Sokolić 2017]. Given a deep network f_θ with L layers, ReLU activations and trained on m examples, if the spectral norm of the weights of each layer is bounded by some constant $C_F > 0$ then an upper bound on the generalization error is given by:

$$\mathbf{GE}(f_\theta) \leq \frac{1}{\sqrt{m}} 2^{L-1} C_F \tag{2.8}$$

If we analyze the equation 2.8, a deep network with a large number of adjustable parameters and therefore a large capacity is likely to learn the training set without error but exhibit poor generalization. Indeed, the previous formula only provides an upper bound on the generalization error without any notion on how tight is this bound. Empirical analysis tends to confirm the gap between the observed generalization error and such bounds: [Guyon 1993] demonstrated that high-order polynomial classifiers in high dimensional space could be trained with a small amount of training data and yet generalizes better than classifiers with smaller VC dimension. The

generalization error's upper bounds based on the VC dimension or on some approximation of the Rademacher complexity were overly pessimistic. As pointed out by Zhang *et al.* , deep networks exhibit different learning behaviors than shallower classifiers such as SVM. Thus common generalization metrics may not adequately explain the generalization error of neural networks. In [Zhang 2016], they empirically demonstrate that deep networks can still generalize while learning on a training dataset corrupted at some point. Indeed, the authors introduce a certain percentage of random labels in the training set. Despite this noise, not only the networks is able to generalize, but it also overfits on the whole training set. Moreover, introducing regularization scheme does not alter the phenomenon. This is really surprising, as we expect regularization to counter overfitting, as it happens for shallower classifiers. However, as pointed out in [Krueger 2017], even if they are able to do so, deep networks probably don't memorize the training set on natural datasets since the number of epochs required to learn the training set on natural data is less than the ones needed to overfit on a random dataset. In the line of uniform stability [Hardt 2015], this suggests that deep networks are also relying on an inductive bias that suits natural data. Based on the previous empirical observations, we expect that the generalization error has to be understood differently for deep networks, perhaps with new metrics, so that VC dimension and the Rademacher complexity are indeed overly pessimistic for deep networks. This flaw opens exciting opportunities on the effectiveness of active learning for deep networks.

Recent works have refined the existing upper bounds on the generalization error of deep networks. For sake of consistentcy, we will not provide an exhaustive list of those works, as it is outside our scope. Eventually, we will detail the new upper bounds that highlight the potential benefits of active learning for deep networks.

First of all, it has been asserted, through both theoretical and empirical analysis, that regularizing the training with dropout, promotes smaller Rademacher complexity. Initially, dropout was motivated to prevent neurons co-adaptation. Nonetheless, it highly affects the Rademacher complexity of deep networks: dropout is able to reduce exponentially the Rademacher complexity of deep networks [Gao 2016]. Moreover, an upper bound of the Rademacher complexity may be expressed as a function of dropout rates and the weights of a network [Zhai 2018]:

Theorem 4.1: Bounding the empirical Rademacher complexity with Dropout

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the sample matrix with the ith row $x_i \in \mathbb{R}^d$.
Let $p \geq 1$, indexes the L_p norm used $\frac{1}{p} + \frac{1}{q} = 1$.

Consider a network with L layers, and denote W^l the weights at the l-th layer. If we apply a mask of dropout Θ^l (made of Bernouilli parameters) for each layer, then we can upper bound the empirical Rademacher complexity for the network.

Indeed, if we denote $\mathcal{W} = \{W \mid max_j \mid\mid W_j^l \mid\mid_p \leq B^l\}$.

$\forall l \in \{1, 2, ..., L\}$ given Θ, the **empirical Rademacher complexity** $\mathbb{R}$ of the loss for the dropout neural network is bounded by:

$$\tilde{\mathcal{R}}_n(\mathcal{H}) \leq k2^L \sqrt{\frac{2log(2d)}{n}} \left|\left| X \right|\right|_{max} \left(\prod_{l=1}^{L} B^l \left|\left| \Theta^{l-1} \right|\right|^{\frac{1}{q}} \right) \tag{2.9}$$

where k is the number of classes, Θ^l is the k^l dimensional vector of Bernouilli parameters for the dropout random variables in the lth layer and $|| \cdot ||_{max}$ is the matrix norm defined as $|| A ||_{max} = max_{i,j} \mid A_{i,j} \mid$.

While the Rademacher complexity has been really useful to understand *passive* learning, it has also been used in **AL**. Indeed, [Hanneke 2011, Koltchinskii 2010] demonstrated how the Rademacher complexity in **AL** helps to develop strategies extendable to a wide panel of input distributions (*while previous **AL** strategies like the ones proposed in [Balcan 2007, Castro 2007] were data specific*). It turns out that the disagreement set: the set of consistent classifiers for which there are two classifiers whose predictions at point x disagree with each other play an important role in the development of active classification algorithms. When it comes to such a disagreement set, Koltchinskii has shown that the number of samples required to cover this disagreement space can be estimated using Rademacher complexity. Finally establishing a close connection between dropout [Hinton 2012], and Rademacher motivates the usage of dropout in an active learning context, either to measure disagreement over the models (*see Chapter 3*) or by sampling through the distribution of the weights ([Gal 2016b]).

Another line of research, in line with the results of [Schapire 1998], analyzed how the generalization error is correlated to the value of the weights, rather than the number of the weights in a neural network. This theory is at the edge of some well-known weight regularization schemes such as weight decay. In this, Liang *et al.* proposed to use the Fisher Rao norm as an indicator of the generalization performance of a neural network.

Definition 2.4.1: Fisher Rao norm

The Fisher Rao norm is defined as:

$$|| \Theta ||_{FR} = \Theta^T \mathbb{I}_\Theta \Theta \tag{2.10}$$

where $\mathbb{I}_\Theta$ is the Fisher information matrix, based on the weights Θ of the neural network f_Θ, trained on the log loss l:

$$\mathbb{I}_\Theta = \mathbb{E}_{x,y} \left[\nabla_\Theta l(f_\Theta(x), y) \nabla_\Theta l(f_\Theta(x), y)^T \right] \tag{2.11}$$

Regarding deep linear networks, it has been shown in [Liang 2017], that the Rademacher complexity can be bounded by the Fisher Rao Norm. Moreover, Liang *et al.* empirically demonstrate how the Fisher Rao norm correlates with the generalization error. The Fisher matrix is also linked to a wide panel of active learning

strategy called Optimal Experimental Design (*see Section 2.3*). We also investigate the usage of Fisher matrix into a Bayesian active learning framework in Chapter 5.

In similar contexts (i.e. where VC is overly pessimistic) for margin-based classifiers, examples sampled in the margin lead to an optimal improvement of the decision at the next active iteration. Such supporting samples lie close to the decision boundary and define the margin of the classifier w.r.t. some metric d (*the smallest distance in the input space between a sample from the training set and a sample with a different prediction*). The generalization error of a classifier with margin γ is upper bounded by the complexity of the input space $\mathcal{X}$ (neglecting the $log\left(\frac{1}{\gamma}\right)$ term) and the classification margin via what we denote *the covering number* $\mathcal{N}(\mathcal{X}; d, \frac{\gamma}{2})$[1]. $\mathcal{N}_{\mathcal{Y}}$ denotes the number of classes.

$$\mathbf{GE}(f_\theta) \leq \frac{1}{\sqrt{m}}\sqrt{2log(2)\mathcal{N}_{\mathcal{Y}}\mathcal{N}\left(\mathcal{X}; d, \frac{\gamma}{2}\right)} \tag{2.12}$$

[Sokolić 2017] developed further equation 2.12 to demonstrate that the generalization error of a deep network (or any other margin classifier) is inversely proportional to the square root of the margin multiplied by the number of training samples. They assumed that the input distribution is a regular manifold which is in accordance with empirical evidence [Arjovsky 2017b]. Indeed, when assuming that the input distribution $\mathcal{X}$ is a regular manifold, the *covering number* may be approximated given the following expression:

A C_M regular k dimensional manifold where C_M is a constant that captures its intrinsic properties has a covering number upper bounded:

$$\mathcal{N}(\mathcal{X}; d, \rho) \leq \left(\frac{C_M}{\rho}\right)^k \tag{2.13}$$

Several results lend credence to an effective margin-based active learning strategy for deep networks. First of all, [Liu 2016] developed a large margin softmax to encourage intra-class compactness and inter-class separability. Their results highlight the benefit of enhancing a large margin between classes. However, the benefits of margin-based active learning highly depend on the number of decision boundaries drawn by neural networks in the input space. If deep networks split the input space in an exponential number of shattered classification region, one may expect that many samples will lie close to a decision boundary, and thus querying samples close to the margin will be almost like collecting random samples.

Empirical evidence leads to thinking that it is not the case when considering deep networks. Independently of the number of parameters of the network, it has been empirically observed in [Fawzi 2017] that state-of-the-art deep networks learn connected classification regions instead of shattered and disconnected regions. Although such classification regions defined in the input space may suffer from the

[1] Regarding the notation, we have purposely decided to stick to the notation of [Sokolić 2017] in order not to confuse the reader.

curse of dimensionality, [Fawzi 2017] have also observed that few directions interfere with the decision boundaries. Considering now the low dimensional space defined by those impacting directions, it becomes likely that the samples do not suffer anymore from the curse of dimensionality and, thus the distance to the decision boundary will differ among the samples. Eventually, certain samples will lie closer to the decision boundaries of neural networks, and are thus highly uncertain in an active learning context. Our assumption comes in line with other measures of the generalization error based on robustness. Robustness, from the seminal work of Xu *et al.* , [Xu 2012], expresses the correlation between the generalization error and the robustness to perturbations over the training set. [Tom Zahavy 2018] extend their work to demonstrate how deep networks can generalize well when their sensitiveness to adversarial perturbations is bounded in average over the training examples.

Definition 2.4.2: Ensemble Robustness

A randomized algorithm $\mathcal{A}$ is (K, $\bar{\varepsilon}$ (n)) ensemble robust for $K \in \mathbb{N}$, if the sample set $\mathcal{Z}$ can be partitionned into K disjoints sets denoted by $\{C_i\}_{i=1}^{K}$ such that the following holds for any input data $\mathbf{s}$: $\forall\, \mathbf{s} \in \mathcal{Z}$

$$\forall s \in \mathbf{s},\ \forall i \in 1...,K:\ if s \in C_i,\ then\ \mathbb{E}_{\mathcal{A}} max_{z \in C_i} \mid l(\mathcal{A}_{\mathbf{s}}, s) - l(\mathcal{A}_{\mathbf{s}}, z) \mid \leq \bar{\varepsilon}(n) \tag{2.14}$$

Theorem 4.2: Ensemble Robustness

Let $\mathcal{A}$ be a randomized algorithm with (K, $\bar{\varepsilon}$ (n)) ensemble robustness over the training set $\mathbf{s}$, where $\mid \mathbf{s} \mid = n$. Let $\Delta(\mathcal{H})$ denote the output hypothesis distribution of the algorithm $\mathcal{A}$ on the training set $\mathbf{s}$. Suppose the following variance bounds holds:

$$var_{\mathcal{A}} \left[max_{z \tilde{s}_i} \mid l(\mathcal{A}_{\mathbf{s}}, s_i) - l(\mathcal{A}_{\mathbf{s}}, z) \right] \leq \alpha \tag{2.15}$$

Then $\forall\ \delta > 0$ with probability at least $1 - \delta$ with respect to the random draw of the $\mathbf{s}$ and $h \sim \Delta(\mathcal{H})$ the following upper bounds holds:

$$|\mathcal{L}(\mathcal{A}_{\mathbf{s}}) - l_{emp}(\mathcal{A}_{\mathbf{s}})| \leq \bar{\varepsilon}(n) + \frac{\alpha}{\sqrt{2\delta}} + M \sqrt{\frac{2Kln(2) + 2ln(\frac{1}{\delta})}{n}} \tag{2.16}$$

Thus, Equ. 2.16 suggests that controlling the variance of the network positively impact the generalization performance.

In our last work on active learning for deep networks (*see Chapter 4*), we combine adversarial attacks and active learning, based on the insight into the effectiveness of margin-based active learning strategy for deep networks.

Chapter 3

Dropout Query-By-Committee

Contents

3.1 Introduction 23
3.2 Sampling a committee with Dropout 25
3.3 Disagreement Scoring Function 27
3.4 Empirical Validation 28
3.5 Conclusion 29

3.1 Introduction

Summary

- We scale *Query-By-Committee* for deep networks
- We use dropout at test time to sample a committee of neural networks
- We query unlabeled samples which maximize a disagreement score over the committee's members

✓ *Every dataset and parameters used to conduct our experiments are available in the* ***dataset*** *section A.1 and the* ***hyperparameter*** *section A.2.1*

mducoffe/DQBC

In this chapter, we consider an active learning method based on Query-By-Committee (**QBC**). [Seung 1992] have proposed the first algorithm based on Query-By-Committee strategy. They proved two relevant results: first, the generalization error of a linear classifier for random training samples behaves like the inverse power law, $\frac{1}{P^{\frac{1}{N}}}$ with P the number of training samples considered so far and N the dimension of the input space; second, the generalization error of a linear classifier for training samples selected through a query-by-committee strategy, scales like $e^{-\frac{PI}{N}}$ with the constant decay given by the information gain I. Later, [Freund 1997] proved that this property holds for a more general class of learning problems.

Instead of trusting only the current incremental classifier, committee decision relies on defining a space of consistent classifiers (i.e., classifiers whose predictions agree with training set labels) where the optimal learner lies. The aim of the active learning step is then to query a sample which will divide at best the consistent classifier space, also called the *version space*. It will thus reduce the possible solutions to converge towards the optimal classifier. There is no consensus in the literature on an appropriate committee size to consider, even when focusing on a class of learning models or an application. However even small committee sizes, e.g., 2 or 3, work well in practice [Seung 1992, Nigam 1998, Settles 2008]. Some recent works tend to combine active learning and model selection to optimize even further the model design [Ali 2014]. After several iterations, the set of consistent hypotheses will shrink and converge towards the optimal classifier. As the size of the version space might be infinite, **QBC** samples a finite number of classifiers to constitute a committee. Eventually, the query decision relies on the committee: the score assigned to an example is based on the prediction disagreement between all predictions of the classifiers in the committee. In early works describing active learning through committee selection, convergence and better result against random sampling have been proven. However, for those results to hold, each model of the committee has to lie in the current version space defined by the annotated training set. This means that the set of neural networks in the committee should be built from the same "architecture" and should make no prediction error on the current training set. When it comes to large datasets, restricting the selection to one additional training sample at a time is computationally expensive since to maintain the version space we should retrain all the classifiers of the committee on that new training sample [Dasgupta 2005b]. The drawback of **QBC** is the cost of building a representative committee. Our version allows us to get rid of this computational issue by using a version of dropout called *batchwise dropout* [Graham 2015]. Firstly, we sum up our batch active learning strategy in Method 3.1.1.

Method 3.1.1: Dropout Query-By-Committee

We have a pool of unlabeled data $\mathcal{P}$ and start training a CNN with a small set of training samples $\mathcal{A}$. This is the initial state of our active learning training set $\mathcal{A}_0 = \mathcal{A}$. At each iteration t, we aim at selecting the *optimal* batch $\mathcal{B}$ by computing a new loop of the following steps:

1. The network is trained on the current training set $\mathcal{A}_t$ leading to the weights $\mathbf{w}_{t+1}$

2. We build a committee of K networks by applying *batchwise dropout* on $\mathbf{w}_{t+1}$: $\mathcal{C}_{t+1} = \{\tilde{\mathbf{w}}_{t+1}^k\}_{k=1}^K$. The procedure is further described in section 3.2.

3. We search for the *optimal* batch $\mathcal{B}$ to add to the training set, i.e. the batch $\mathcal{B}$ maximizing the disagreement over the committee $\mathcal{C}_{t+1}$.

$$\mathcal{B} = argmax\{Disagreement(x_i \mid \mathcal{C}_{t+1}) \mid x_i \in \mathcal{P}\} \tag{3.1}$$

4. The training set is then augmented by $\mathcal{B}$: $\mathcal{A}_{t+1} = \mathcal{A}_t \cup \mathcal{B}$

3.2 Sampling a committee with Dropout

Before starting, let us define some name convention: For the sake of clarity, we denote by full network, the deep architecture trained on the current labeled training set $\mathcal{A}$ and partial network a CNN member of the committee.

We train the *full* network on the current annotated training set until the prediction error on an independent validation set is not further decreasing. When the training has converged, the *full* network is no longer able to learn more knowledge on the input distribution from the current annotated training set. Thus we apply active learning to query new labeled data and add them to the training set. Eventually the *full* network is retrained from scratch on the new training set(*weights and biases are reset with a random initialization*). When it comes to query-by-committee for deep architectures, the challenges are to define:

1. **Committee design**: developing a computationally lightened building scheme of diverse *partial* CNNs.

2. **Sample selection**: Proposing a relevant sample selection function based on the committee's predictions (*see Section 3.3*).

Here we consider the cost of building a committee. A naive setup would consist in training models in parallel; networks sharing the same architecture with different initialization. However, this framework is not ideal for at least two reasons:

- Training multiple models at the same time is not scalable when considering large dimensional neural networks

- As neural networks solve a high dimensional non-convex problem, we may expect that networks sharing the same architecture but trained independently will vary a lot. This assumption does not always hold, as demonstrated in [Choromanska 2015]. When training networks with the same architecture but initialized differently, they observed empirically that the variance of the test loss shrunk. This suggests that learning a naive ensemble is not a good strategy to obtain a diverse set of classifiers.

Another solution is to consider ensemble from a probabilistic point of view. Indeed, if one may express the posterior distribution of the weights, we are able to sample consistent hypotheses directly from this approximation, and thus design our committee. However, computing the posterior distribution of the weights is generally intractable for deep networks. Note that this **AL** strategy is more thoroughly studied in Chapter 5. Instead, one can rely on a tractable approximation of the weights distribution. Recently, [Gal 2016b] demonstrated that dropout (and other stochastic regularization schemes) is equivalent to infering on the posterior distribution of the weights, thus enabling to leverage the cost of training and updating multiple models. Consequently, dropout allows sampling an ensemble of models at test time.

In the same spirit, we propose a Dropout-based **QBC** strategy that we call *Dropout Query-By-Committee* (**DQBC**). Instead of training an ensemble of networks, we use dropout to sample *partial CNNs* given the weights of the *full CNN*, as illustrated in Figure 3.1. Notice that, independently and after this work, Gal *et al.* also designed a Bayesian active learning framework relying on a dropout committee.

Let us now detail how we build *partial* CNNs in order to form the committee. To initiate a *partial* CNN while getting rid of the computation cost thanks to back-propagation, we apply batchwise dropout [Graham 2015] on our full network. The batchwise dropout [Graham 2015] is a version of dropout where we use a unique bernouilli mask to discard neurons for each sample in the minibatch. Thus the batchwise dropout reduces quadratically in the percentage of preserved neurons, the number of parameters in the architecture. When considering convolutional layers, the batchwise dropout has one advantage over dropout: the latter removes neurons independently given the spatial locations whereas batchwise dropout is spatially dependent, switching on or off filters so to discard neurons obtained through the same filter. Figure 3.1 presents how batchwise dropout preserves the consistency in a CNN architecture which allows us to create our *partial* CNNs.

The main advantage is to obtain a committee whose members contain fewer parameters while sharing the same architecture as the *full* network with zero constraints on several connexions. In order to increase the accuracy of each *partial* CNN, we finetune each member of the committee on the current labeled training

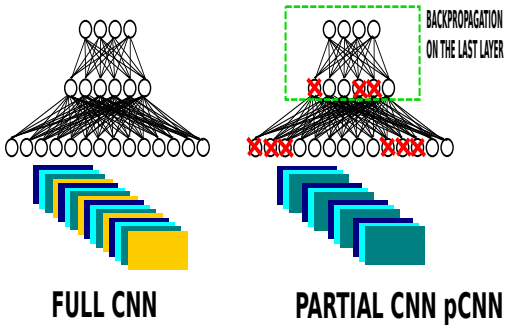


Figure 3.1: applying batchwise dropout to build a *partial* CNN from the *full* CNN

set. This finetuning is not prohibitive as the number of parameters in a partial CNN is drastically much lower than in the full CNN, due to batchwise dropout.

3.3 Disagreement Scoring Function

In the context of **QBC**, a sample is considered as informative based on its ability to reduce the number of current consistent hypotheses. Thus the informativeness of a sample is measured by the quantity of disagreement about the prediction of its label among the *partial* CNNs. We illustrate such disagreement on a baby task in Figure 3.1.

We propose our own metric based on how much a *partial* CNN may change its decision to be in accordance with the majority. In that order, we define a smooth vote on the members of the committee. Let denote the committee as a set of *partial* CNNs: $\mathcal{C} = \{ pCNN_i\}$ with p^i the output probability vector of $pCNN_i$. Given a sample $\mathbf{x}$, we first establish its most probable label based on the committee predictions:

$$\mathbf{LABEL}(\mathbf{x}) = \arg\max_{j} \sum_{pCNN_i} 1_{j=\underset{k}{argmax}} \, p^i(y = k \mid \mathbf{x}) \tag{3.2}$$

We took inspiration from Random Forest margin function [Breiman 2001] in order to produce a ranking of candidates for selection and to have a *soft* pool among the committee. Our point is to take into account the confidence of a *partial* CNN into the score function rg(x) and query the samples with the highest score:

$$rg(\mathbf{x}) = \sum_{pCNN_i} \max_{j} p^i(y = j \mid \mathbf{x}) - \; p^i(y = \mathbf{LABEL}(\mathbf{x}) \mid \mathbf{x}) \tag{3.3}$$

We add minibatches of samples instead of one sample as supposed for active learning technique, both to leverage the computational cost owing to successive runs of active learning and to avoid unbalanced size of minibatch (*in that case an adjustement of the learning rate given the size of the last minibatch would be required*).

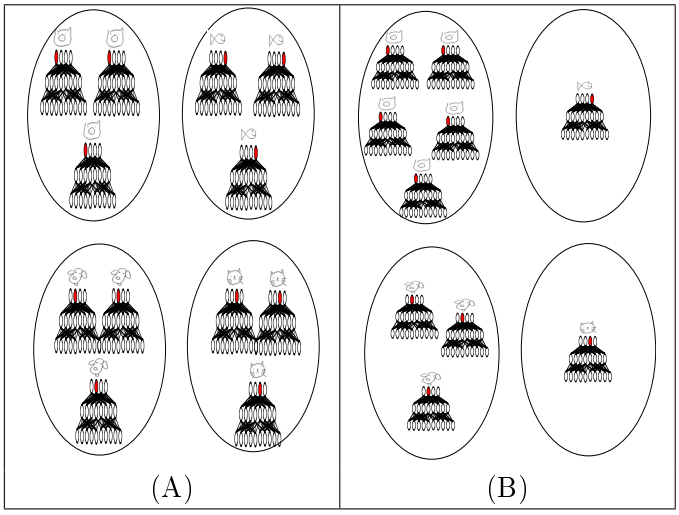


Table 3.1: Example of Sample Selection. The icon outputs by a network represents its most probable label on the unlabeled sample. Samples (A) and (B) are both unlabeled examples. Sample (A) is more appealing as it will necessarily reduce the version space by a ratio of 4. Whereas Sample (B) will reduce at worst the version space by a ratio of 2.5. On another side, if one has complementary information about the networks, such as the confidence of each network in their prediction, sample (B) may become a better choice.

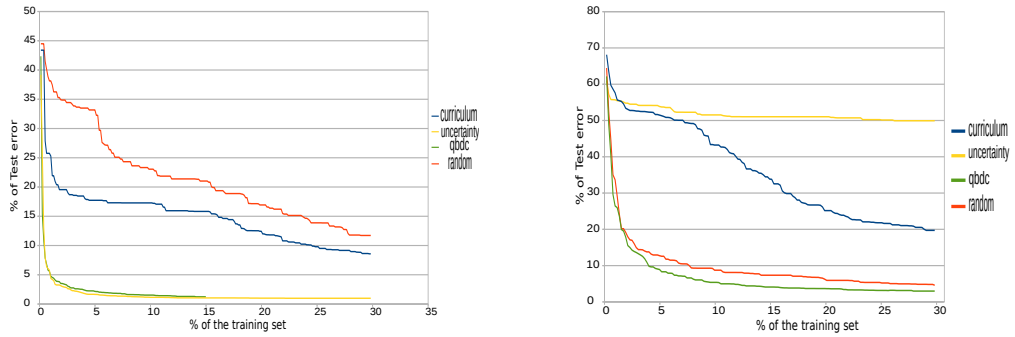
Every dataset and parameters used to conduct our experiments are available in the **dataset** section A.1 the **hyperparameter** section A.2.1

3.4 Empirical Validation

We demonstrate the validity of our approach on two datasets: *MNIST* (batch size of 64) and *USPS* (batch size of 8) both gray scaled digit image datasets. Both CNNs have ReLu. Note that we do not optimize the hyperparameters depending on the size of the current annotated training set. We picked those two similar datasets to judge of the robustness of our method against different size of unlabeled datasets. Finally, our method is efficient on restricted and larger pool of unlabeled samples.

We perform 5 to 10 runs of experiments and record the test error of the best validation error before an active learning iteration. We start from an annotated training set of size one minibatch selected randomly. We stop both sets of experiments when we reach 30% of the training set (15.000 images for *MNIST*, 1255 for *USPS*). We sample 5 *partial* CNNs to form a committee. In Figure 3.2, we compare **DQBC** to uncertainty, curriculum [Bengio 2009] and random selection with a top scoring selection on a convolutional network. We measure both uncertainty and curriculum scores based on the log likelihood of a sample. We use the prediction of the *full* network to approximate the ground-truth label. While uncertainty selects samples with the highest log likelihood, our version of curriculum does the exact contrary. We select the set of possible queries among the unlabeled training data randomly. Its size is set to 30 times the minibatch size. The experiments in (*see Figure 3.2*)

conducted on *MNIST* and *USPS* illustrate that **DQBC** converges faster to the best accuracy achieved without active learning on the whole annotated training set than the other selection methods: for *MNIST* we see that less than 26% of the database is necessary to obtain almost the final accuracy (1.23% on test error instead of 1.1 %). When it comes to *USPS*, larger difference are observed: **DQBC** is the only active method able to achieve the ground-truth accuracy with less than 22% of the training set.



(a) MNIST

(b) USPS

Figure 3.2: **DQBC(qbdc) with top score selection**: Evolution of the test error given the ratio of annotated data over the training set.

3.5 Conclusion

This chapter introduced an adaptation of Query-By-Committee for deep networks based on dropout. It allows to train CNNs on a smaller annotated training set to achieve similar accuracy to the one obtained using a much larger annotated database. Our work bridges the computational gap between active learning for CNNs and other shallow classifiers. The use of a committee allows our active learning heuristic to have the distributive training of its CNNs which is a natural advantage of QBC derived methods.

Chapter 4

Adversarial Active Learning

Contents

4.1 Introduction **32**

4.2 Margin-Based Active Learning for Deep Networks **34**

4.3 Empirical Validation **36**

4.3.1 Dataset and hyperparameters 36

4.3.2 Evaluation 37

4.3.3 Comparative study between DFAL and CORE-SET 40

4.4 Transferability **41**

4.5 Discussion **44**

4.5.1 Theoretical motivations 44

4.5.2 DFAL does not select random samples in the first runs 44

4.6 Adversarial Active Learning for Linear Classifiers **46**

4.6.1 Transferable adversarial attacks 46

4.6.2 Label Complexity on the unit ball 49

4.7 Conclusion **50**

4.1 Introduction

Summary

• We present a new heuristic for margin-based active learning for deep networks, called DeepFool Active Learning (**DFAL**) (*see 4.2*). It queries the unlabeled samples, which are the closest to their adversarial attacks, labels not only the unlabeled sample but its adversarial counterparts as well, using twice the same label. This pseudo-labeling comes for free without introducing any corrupted labels in the training set.
• We empirically demonstrate that **DFAL** labeled data may be used on other networks than the one they have been designed for, while achieving higher accuracy than random selection. To the best of our knowledge, this is the first active learning method for deep networks tested for this property. (*see 4.4*)
• We demonstrate the theoretical gain of our method for linear classifier (*see 4.6*).

✓ *Every dataset and parameters used to conduct our experiments are available in the* ***dataset*** *section A.1 and the* ***hyperparameter*** *section A.2.2*

mducoffe/DFAL PROOF Proofs are available in **Appendix A.3.1**

One of the most standard active learning strategies is to rely on the uncertainty measure. Uncertainty in deep networks is usually evaluated through the network's output. However, this is known to be misleading. Indeed, the discovery of adversarial examples has demonstrated that the way we measure uncertainty may be overconfident. Adversarial examples are inputs modified with small (sometimes not perceptually distinguishable) but specific perturbations which result in an unexpected misclassification despite the strong confidence of the network in the predicted class [Szegedy 2013]. Moreover, their perturbation is often hardly visible (*see Figure 4.1 for an example*).

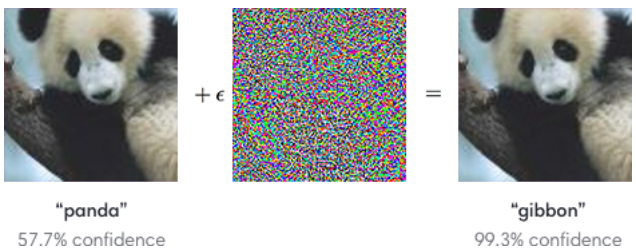


Figure 4.1: *"An adversarial input, overlaid on a typical image, can cause a classifier to miscategorize a panda as a gibbon."* [Goodfellow 2015]

On one hand, the existence of such adversarial examples somehow discards

uncertainty-based selection from being an efficient active learning criterion for deep networks. On the other hand, the magnitude of adversarial attacks does provide a piece of information about how far a sample is from the decision boundariy of a deep network. This information is relevant in active learning and known as margin-based active learning. In a generic margin-based active learning, we assume that the decision boundaries evolve towards the optimal solution as the training set increases. Hence, samples lying the farthest from a decision boundary do not need to be labeled by a human expert, as long as the current model is consistent in its predictions with the optimal solution. To refine the current model, margin-based active learning queries the unlabeled samples lying close to the decision boundary. [Balcan 2007] have demonstrated the significant benefit of margin-based approaches in reducing human annotations: in specific cases, one may obtain an exponential improvement over human labeling. However, it requires computing the distance between a sample and a decision boundary which is not tractable when considering deep networks. Although we can approximate this distance by finding the minimal distance between two samples from different classification regions (i.e., corresponding to two different classes), such an evaluation is computationally expensive, nor it provides a tight upper bound. Eventually, the minimal adversarial perturbation of a sample does provide a better upper bound on how far this sample is from the closest decision boundary.

In this section, we do not consider adversarial examples as a threat but rather as a guidance tool to query new data. Our work focuses on a new active selection criterion based on the sensitiveness of unlabeled examples to adversarial attacks. We depict our method in Method 4.2.1.

4.2 Margin-Based Active Learning for Deep Networks

Method 4.2.1: Adversarial Active Learning

We have a pool of unlabeled data $\mathcal{P}$ and start training our CNN with a small set of training samples $\mathcal{A}$. This is the initial state of our active learning training set $\mathcal{A}_0 = \mathcal{A}$. At each iteration t, we aim at selecting the *optimal* batch $\mathcal{B}$ by following those steps:

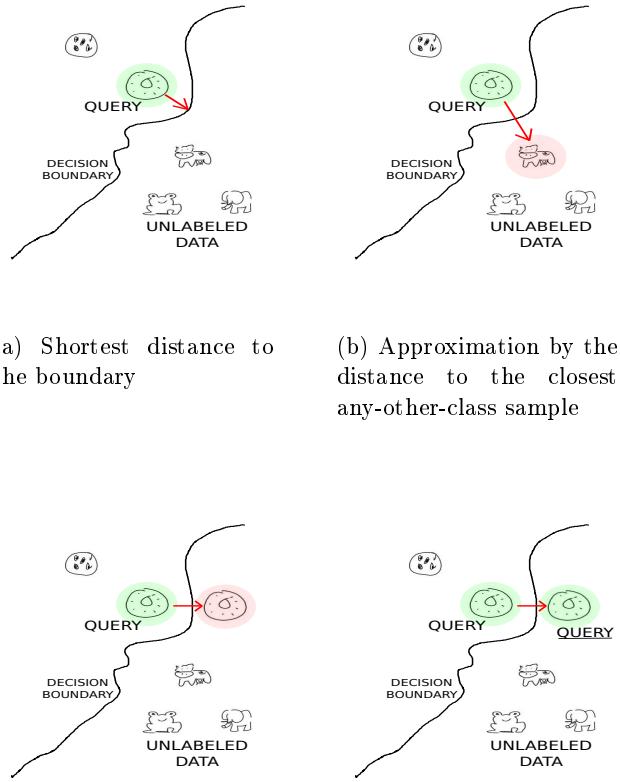
1. The network is trained on the current training set $\mathcal{A}_t$ leading to the weights $\mathbf{w}_{t+1}$
2. We search for the *optimal* batch $\mathcal{B}$ of samples to be added to the training set, i.e. the batch $\mathcal{B}$ whose samples ows the minimal adversarial perturbation

$$\begin{gathered}\text{For } x_i \in \mathcal{P} \setminus \mathcal{A} \\ \#\text{compute adversarial attacks with } L_p \text{ norms} \\ r_i \leftarrow DeepFool(x_i, \mathbf{w}_{t+1}; p) \\ \#\text{ query the labels of the } \mid \mathcal{B} \mid\text{-th smallest perturbation} \\ index_k \leftarrow argsort(< r_i, r_i >_p \mid i = 1..K) \\ \mathcal{B} \leftarrow \{x_j \mid j \in index_k[:\mid \mathcal{B} \mid]\}\end{gathered}$$

3. The training set is then augmented by $\mathcal{B}$: $\mathcal{A}_{t+1} = \mathcal{A}_t \cup \mathcal{B}$

Balcan *et al.* demonstrated the significant benefit of margin-based approaches in reducing human annotations. We illustrate several margin-based active learning heuristics in Figure 4.2: for each scenario, the data underlined in green will be queried. Especially, Figure 4.2(d) describes our contribution. In the original case in Figure 4.2(a), the projection of an unlabeled sample to the decision boundary determines whether or not it is worth to query its label, depending on the distance between the sample and the boundary. Margin-based strategies are effective, but they require to know how to compute the distance to the decision boundary. When such a distance is intractable, a simple approximation consists in computing the distance between the sample of interest and its closest neighboring sample which has a different predicted class.

Approximating the distance between a sample and the decision boundary, by the distance between this same sample and its closest neighboring sample from a different class, is coarse and computationally expensive.
Instead, we propose **DFAL**; a Deep-Fool based Active Learning strategy which selects unlabeled samples with the smallest adversarial perturbation. Indeed, adversarial attacks were initially designed to approximate the smallest perturbation to cross the decision boundary. Hence, in a binary case, the distance between a sample and its smallest adversarial example better approximates the original distance to the decision boundary than the approximation mentioned above, as illustrated in



(a) Shortest distance to the boundary

(b) Approximation by the distance to the closest any-other-class sample

(c) Approximation by the distance to the adversarial example

(d) DFAL Strategy

Figure 4.2: Illustration of different margin-based active learning scenarios in the binary case

Figure 4.2(c). Usually, adversarial attacks which would allow us to design a perturbation requires also to know the target label. However, in a binary case, the target class of the attack is obvious.

In a multi-class context, everything is different: we do not have any prior knowledge on which class the closest adversarial region belongs. Inspired by the strategy done previously in EGL [Zhang 2017], we could design as many perturbations as the number of classes and keep only the smallest perturbation, but this would be time-consuming. We thus have to consider the available techniques of adversarial attacks from the literature [Szegedy 2013, Goodfellow 2015, Carlini 2016] and look for the most laborious procedure to counter since it will provide more information on the margin in more cases and more difficult situations. Indeed, computing the closest adversarial perturbations is a NP-hard problem. Hence we need to rely on heuristics.

To the best of our knowledge, Carlini *et al.* [Carlini 2017b, He 2017, Carlini 2017a] methods are among the hardest attacks to counter. However, it also requires to tune several hyperparameters.

We have thus decided to use the *Deep-Fool* algorithm to compute adversarial attacks for **DFAL** [Moosavi-Dezfooli 2016]. Indeed, *Deep-Fool* is an iterative procedure which alternates between a local linear approximation of the classifier around the source sample and an update of this sample so that it crosses the local linear decision. The algorithm stops when the updated source sample becomes an adversarial sample regarding the initial class of the source sample. When it comes to **DFAL**, *Deep-Fool* holds three main advantages: (i) it is hyperparameter free (especially it does not need target labels which makes it more compliant with multi-class contexts); (ii) it runs fast as we empirically noticed in table 4.3; (iii) it is competitive with state-of-the-art adversarial attacks.

To regularize the network and increase its robustness, we add both the less robust unlabeled samples and their adversarial attacks. Thus, it is more likely that the network will regularize on the adversarial examples added to the training set and become less sensitive to small adversarial perturbations. Unlike CEAL, **DFAL** is hyperparameter-free and cannot corrupt the training set: from the basic definition of adversarial attacks, we know that a sample and its adversarial attack should share the same label.

Finally, **DFAL** improves the robustness of the network by adding at each iteration unlabeled samples at half the cost of reading their right labels (one label amounts to two examples).

4.3 Empirical Validation

4.3.1 Dataset and hyperparameters

We evaluate our algorithm for fully supervised image classification on three datasets that have been considered in recent articles on active learning for Deep Learning [Huijser 2017] (Table 4.1): *MNIST*, *Shoe-Bag*, and *Quick-Draw*. For *Quick-Draw*, we downloaded four classes from the Google Doodle dataset: Cat, Face, Angel, and Dolphin.

	img size	**# classes**	**# Training**	**# Test**
MNIST	(28,28,1)	10	60,000	10,000
Shoe-Bag	(64,64,3)	2	184,792	4,000
Quick-Draw	(28,28,1)	4	444,971	111,246
CIFAR10	(64,64,3)	10	50,000	10,000
Cats & Dogs	(150,150,3)	2	2000	2000

Table 4.1: Summary of the datasets used to evaluate **DFAL**.

We assess the efficiency of our method on two CNNs: LeNet5 and VGG8 (*Adam,*

lr=0.001, batch=32). We use Keras and Tensorflow [Chollet 2015, Abadi 2016]. Note that **DFAL** may be used on any architectures impaired by adversarial attacks.

4.3.2 Evaluation

We compare the evolution of the test accuracy when using **DFAL** against the following baselines:

- BALD: we select on a random subset of the unlabeled training set the first n_{query} samples which are expected to maximize the mutual information with the model parameters. In that order, we sample 10 networks from the approximate posterior of the weights by also applying dropout at test time.
- CEAL: we select on the whole unlabeled training set the first n_{query} samples with the highest entropy on their network's prediction. We also label any unlabeled samples whose entropy is lower than a given threshold (which is set according to the authors' guidelines: 0.05 for *MNIST*, 0.19 for *Shoe-Bag* and 0.08 for *Quick-Draw*). Their labels are not queried but estimated from the network's predictions.
- CORE-SET: we select on a random subset of the unlabeled training set the n_{query} samples which cover at best the training set (labeled and unlabeled data) based on the euclidean distance on the output of the last fully connected layer. To approximate the *covering radius*, we follow the instructions prescribed in [Ozan Sener 2018]: we initialize the selection with the greedy algorithm, and iterate with their mixed integer programming subroutine. We also handle the robustness as prescribed by the authors. We use *or-tools* [1] to reproduce the MIP subroutine.
- EGL: we select from a random subset of the unlabeled training set the first n_{query} samples whose gradients achieves the highest euclidean norm.
- uncertainty: we select from the whole unlabeled training set the first n_{query} samples with the highest entropy on their network's prediction.
- RANDOM: we select randomly from the whole unlabeled training set n_{query} samples.

We average our results over five trials and we plot in Figures 4.3,4.4 the test accuracy achieved by each active learning methods for fixed size training set: ranging from 10 to 1000 labeled samples. We denote as *BASELINE*, the test accuracy obtained when training the network on the full labeled training set. First, an interesting observation is that, independently from networks or datasets, active learning methods originally designed for singleton query (BALD, CEAL, EGL, uncertainty) fail to always compete against random selection (Fig. 4.4). This may result from the correlations among the queries when using one sample at-a-time. When it comes to our method, **DFAL** tends to converge faster than such methods and is always better than random selection, independently from the network or the dataset (Fig.4.3,4.4). Hence our method is more robust to hyperparameter settings than other active learning methods which consider one sample at a time. For various configurations (*Shoe-Bag* with LeNet5 and *Quick-Draw* with VGG8), CEAL is worse than uncertainty selection, hence it selects samples with high entropy but mistaken predictions which

[1] https://developers.google.com/optimization

	Accuracy ≥ 99.04 %	
	# annotations	# labeled data
DFAL	**1210**	2410
CORE-SET	1810	1810
CEAL	≥6000	≥6150

(a) *MNIST* (LeNet5)

	Accuracy ≥ 98.98 %	
	# annotations	# labeled data
DFAL	**980**	1950
CORE-SET	1270	1270
uncertainty	2800	2800

(b) *MNIST* (VGG8)

	Accuracy ≥ 99.70 %	
	# annotations	# labeled data
DFAL	1070	2130
CORE-SET	**860**	860
CEAL	1130	19157

(c) *Shoe-Bag* (LeNet5)

	Accuracy ≥ 99.50 %	
	# annotations	# labeled data
DFAL	530	1050
CORE-SET	**400**	400
CEAL	580	705

(d) *Shoe-Bag* (VGG8)

	Accuracy ≥ 95.46%	
	# annotations	# labeled data
DFAL	**7470**	14930
CORE-SET	≥8590	≥8590
uncertainty	≥10590	≥10590

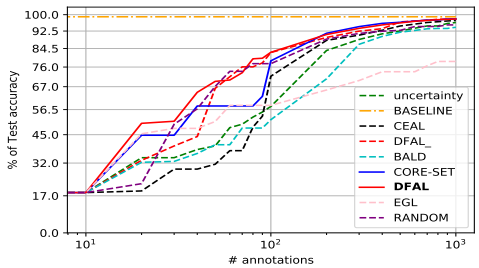
(e) *Quick-Draw* (LeNet5)

	Accuracy ≥ 96.75%	
	# annotations	# labeled data
DFAL	**4810**	9610
CORE-SET	≥6750	≥6750
BALD	5590	5590

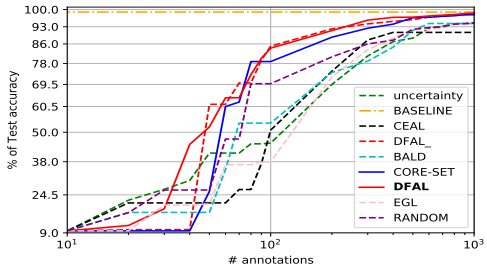
(f) *Quick-Draw* (VGG8)

Table 4.2: Number of annotations to achieve the same test accuracy on LeNet5 and VGG8 as the accuracy obtained on the full training set (BASELINE, ± 0.5 %).

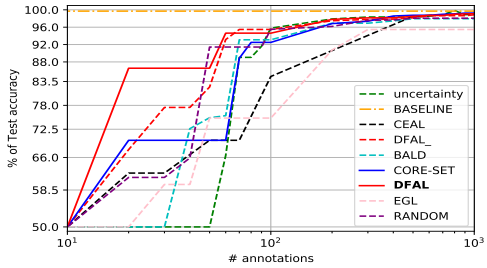
add noise into the training set. Unlike CEAL, whose probability of acquiring extra samples depends on the efficiency of the network, **DFAL** holds a constant number of extra queries, depending only on the number of queries. Moreover **DFAL** creates artificial data which are not part of the pool of data. For example, in Tables 4.2(a) and 4.2(c), CEAL used more than 20% of the training set of *MNIST* and *Shoe-Bag*, while **DFAL** only used at most 2%. Thus, **DFAL** allows more queries, and may also be combined with CEAL. We observe that **DFAL** always remains in the top three of the best performing active learning methods. We define those methods based on the test error rate when the labeled training set reaches 1000 samples. When **DFAL** is outperformed, it is only by a really slight percentage of accuracy (at most 0.15%), either by pseudo-labeling method (*which contributes more to the training set*), or by CORE-SET. Since CORE-SET is designed as a batch active learning strategy, it diminishes the correlations among the queries. In order to outperform CORE-SET, **DFAL** could be extended into a batch setting approach: instead of selecting the top score samples, one could increase the diversity using for example submodular heuristics [Wei 2015]. Finally, Table 4.2 compares the effective number of annotations and real number of data required by active learning to reach the same test accuracy than when training on the full labeled training set. We only compare **DFAL** with the best two active learning methods on 1000 samples. We note that **DFAL** always converges with the smallest number of annotations, on *MNIST* and *Quick-Draw*, for both LeNet5 and VGG8 networks: up to 33% less samples than the current state-of-the-art CORE-SET and up to 80% less samples than CEAL. When it comes to *Shoe-Bag*, **DFAL** remains competitive with CORE-SET and CEAL, overall less than 1% of the training set is needed.



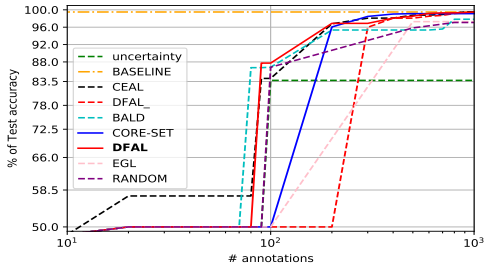
(a) *MNIST* on LeNet5



(b) *MNIST* on VGG8

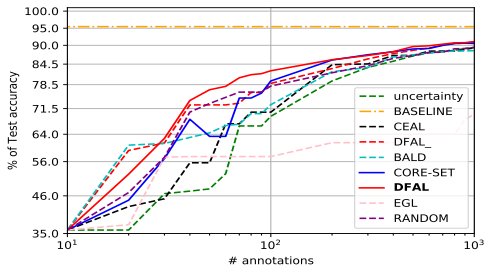


(c) *Shoe-Bag* on LeNet5

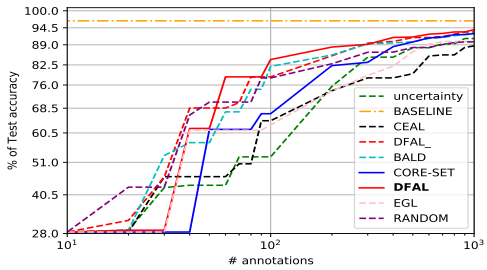


(d) *Shoe-Bag* on VGG8

Figure 4.3: Evolution of the test accuracy achieved by 7 active learning techniques on *MNIST* and *Shoe-Bag* given the number of annotations. We denote by **DFAL_** our active learning method when not adding the adversarial examples. We use a log scale in the x-axis and a linear scale in the y-axis.



(a) *Quick-Draw* on LeNet5



(b) *Quick-Draw* on VGG8

Figure 4.4: Evolution of the test accuracy achieved by 7 active learning techniques on *Quick-Draw* given the number of annotations. We denote by **DFAL_** our active learning method when not adding the adversarial examples. We use a log scale in the x-axis and a linear scale in the y-axis.

MNIST	**DFAL**	CORE-SET (with regularisation)	CORE-SET (no regularisation)
$\mid \mathcal{L} \mid = 100$, $\mid \mathcal{U} \mid = 800$	**126.54**	891.78	784.99
$\mid \mathcal{L} \mid = 1000$, $\mid \mathcal{U} \mid = 800$	**126.54**	3739	3046

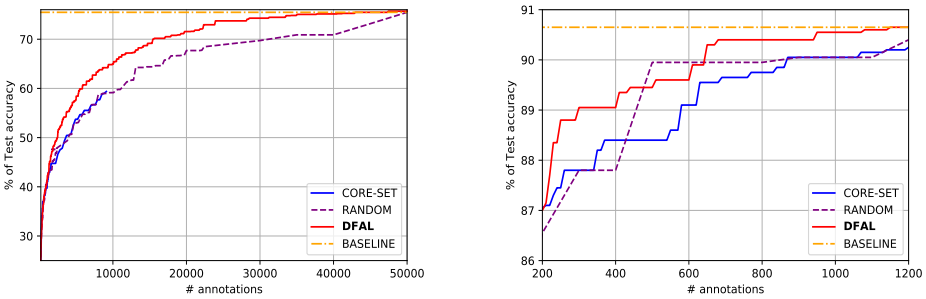
Table 4.3: Average runtime of **DFAL** and CORE-SET on *MNIST*. We denote by $\mathcal{L}$ the labeled training set, and $\mathcal{U}$ the unlabeled set of data; $n_{query} = 10$

4.3.3 Comparative study between DFAL and CORE-SET

In our experiments, **DFAL** is competitive with the current state-of-the-art method, CORE-SET, sometimes outperforming it by a large margin (Tab. 4.2(e),4.2(f)). On the other hand, our method is more interesting than CORE-SET when considering the computational time. DeepFool yields high-performing perturbation vector compared with other state-of-the-art attacks, while being computationally efficient: it converges in a few iterations (less than 3). At each iteration it requires (#classes -1) forward and backward passes. As our DFAL technique uses DeepFool, our active selection criterion is highly efficient compared to the current state-of-the-art CORE-SET. We demonstrate the computational time gap between our method, **DFAL**, and CORE-SET in Table 4.3: we have recorded the average runtime of selecting 10 queries on *MNIST*. For a sake of fairness, we compare **DFAL** running time against the CORE-SET approach, with and without robustness [2]. Note that the runtime performance of **DFAL** is independent from the size of the labeled training set. On the contrary, CORE-SET slows down while we add more and more data to the training set. Eventually, Table 4.3 reports gains of (up to) 24 times faster in running time by our method against CORE-SET. It is worth noting that adversarial attacks are independent, which could easily lead to a parallelized active learning strategy. However, for a fair comparison with CORE-SET, we only consider sequential attack generation.

We investigate further the comparison between **DFAL** and CORE-SET on two experiments. A first experiment studies the behaviour of both active learning methods on a large scale dataset, *CIFAR10*: we train a CNN on *CIFAR10* with 5 layers of convolution, maxpooling and 2 fully connected layers with a dropout rate of 0.25 and no artifical augmentation. In Figure 4.5(a), CORE-SET achieves similar accuracy than RANDOM. Due to the running time of CORE-SET, we could not pursue CORE-SET until convergence. On the other hand, our method **DFAL** converges much faster. The second experiment consists in combining active learning with transfer learning: we use VGG8 as a pretrained network that remained fixed during the training on *Cats & Dogs*. In Figure 4.5(b) we train a CNN with 3 layers of convolutions, maxpooling and 2 dense layers, with a dropout rate of 0.5 and artificial augmentation.

[2] Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz; 64 GB memory and GTX TITAN X



(a) CIFAR (b) Transfer Learning (Cats & Dogs)

Figure 4.5: Evolution of the test accuracy achieved by 3 active learning techniques given the number of annotations

4.4 Transferability

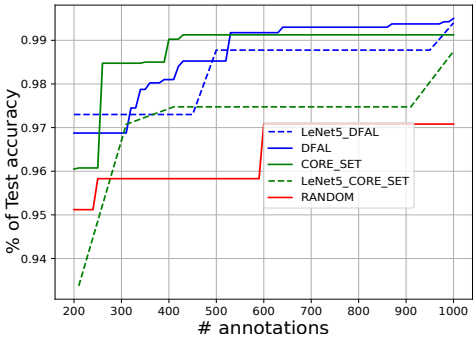


Figure 4.6: Evolution of the test accuracy for (*Shoe-Bag*, VGG8) trained with different labeled training set: we compare the efficiency of **DFAL** and CORE-SET built on LeNet5 (LeNet5_**DFAL** and LeNet5_CORE-SET) and transfered to VGG8.

When faced with a new classification problem, we don't know the hyperparameters that are best suited for the problem. One can argue that a network with high capacity is likely to give high accuracy and is sufficient enough when combined with some human expertise on the problem: several architectures have been handcrafted for specific tasks and are available online [Chollet 2015]. Still, their efficiency is known for large datasets. [Yanyao Shen 2018] pointed out a flaw in active learning: their active learning heuristics perform well if and only if they use it on a lightweight architecture instead of the architecture of reference for Named Entity Recognition (NER) classification. Such an issue is inherent to active learning. Combining model selection with active learning has been investigated for shallow models [Sugiyama 2008]. One of the main issues raised is that multiple hypotheses (i.e. candidate networks) trained in parallel may require labeling different training points.

	DFAL	CORE-SET	RANDOM
LeNet5→ VGG8	**97.80**	96.90	94.46
VGG8→ LeNet5	**97.93**	97.40	95.31

(a) *MNIST*

	DFAL	CORE-SET	RANDOM
LeNet5→ VGG8	**92.87**	91.06	89.94
VGG8→ LeNet5	89.23	89.41	**89.42**

(b) *Quick-Draw*

	DFAL	CORE-SET	RANDOM
LeNet5→ VGG8	**99.40**	99.12	97.08
VGG8→ LeNet5	**98.75**	98.50	98.07

(c) *Shoe-Bag*

Table 4.4: Comparison of the transferability of **DFAL** and CORE-SET with 1000 annotations

Furthermore, [Fawzi 2017] empirically demonstrated a strong correlation between the vulnerability of a network to small adversarial perturbations and an asymmetry in the curvature of its decision boundary: if a model is not robust to an adversarial attack, it is likely that the curvature in that direction is negative and vice-versa. Thus, not only that the decision boundaries would lie close one to another but they would likely share some strong topological properties. Based on those arguments, we assume adversarial queries are useful for a diverse set of architectures, not only for the CNN they have been queried for.

First of all, we assert this assumption by evaluating the classification regions overlap between LeNet5 and VGG8; both trained on the QuickDraw dataset. Results are presented in Figure 4.7. We observe that most of the test samples share the same classification regions (• blue dots) for both networks, LeNet5 and VGG8, while few of them (• red dots) are in different classification regions. Note that, this does not mean that the networks disagree on their prediction on such samples but put them in different classification regions. Thus, it appears than CNNs may have significant overlaps on their classification regions, at least for LeNet5 and VGG8.

When it comes to the transferability, we empirically demonstrate **DFAL**'s potential for a baby task. We compare the test accuracy of **DFAL**and CORE-SETtransferred dataset on 1000 samples in Table 4.4. Surprisingly the transferred queries from CORE-SETperform better than random. However, the transferred queries from **DFAL**outperform CORE-SETand RANDOM.

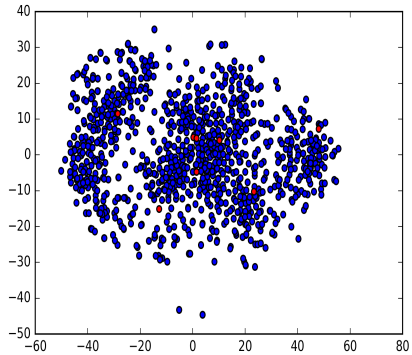


Figure 4.7: **Overlap of the classification regions of LeNet5 and VGG8 trained on the QuickDraw datasets**. Blue dots • are test samples that fall into the same classification regions for both networks, while red dots • do not fall into the same region. We proceed by looking for a convex path so that every point in that path share the same prediction. To do so, we check the validity of the path in the convex combinations of consecutive anchor points, as proposed by Fawzi *et al.* [Fawzi 2017]. Then we check, whether paths exist for both networks and project the test samples in a two-dimensional space using T-SNE [Maaten 2008].

However, it has been shown that under some constraints of similarities between the architectures, adversarial examples of a network A are very likely to be adversarial for a network B. This turns to be a significant advantage for our adversarial active learning strategy since the training set built with **DFAL** for the network A will then be very likely to be a relevant training set for the network B.

When it comes to the transferability, we empirically demonstrate **DFAL** potential on a toy task: in Figure 4.6 we have recorded *Shoe-Bag* adversarial queries for LeNet5 and use them for training VGG8. While the test accuracy achieved is lower than with the adversarial active strategy directly applied for the training of VGG8, the transfered training set achieves better accuracy than RANDOM. When reaching 1000 annotated samples, it is also better than considering other active criteria designed for VGG8. We go further and compare the accuracy on 1000 test samples of DFAL and CORE-SET trained on the transfered training set in Table 4.4. Surprisingly the transfered queries from CORE-SET perform better than RANDOM. However, in almost every case, the transfered queries with **DFAL** outperform CORE-SET and RANDOM. We have therefore shown the relevance of transfering adversarial examples generated within active learning from one architecture to another. This opens up promising perspective for the design of tractable methods to explore network architectures.

4.5 Discussion

4.5.1 Theoretical motivations

It is challenging to demonstrate theoretically the gain in annotations of **DFAL** owing to (i) the high-dimensional space induced by deep networks and (ii) the lack of understanding of the phenomenon of adversarial examples. However, we have lately been able to prove the gain of **DFAL** for linear classifiers theoretically (*see Section 4.6*). Specifically, we demonstrate the theoretical gain in reducing the labeling effort when data are drawn from the unit ball and consistent with a linear separator with no bias. Notice that our proof may be extended to other distributions as long as they are iid along any dimension (such as isotropic Gaussian).

We already know from the theoretical work of [Balcan 2007] that we need to sample the examples from a subregion carefully chosen to obtain an exponential improvement in the label sample complexity. Such subregion is the area along the decision boundary given the current generalization error achieved at iteration k. For the linear case, DeepFool is a natural extension of the well-known attack which consists in adding the perturbation along the gradient direction In the linear case, adversarial attacks directly measure the distance to the decision boundary. Thus, when sampling unlabeled samples with the smallest adversarial perturbation, we sample examples from the low confidence subregion and we are consistent with Balcan's protocol. Our proof goes further, by demonstrating how the adversarial counterparts help reducing up to twice the number of required queries. Our proof goes into two steps: (i) Based on the notion of adversarial strength, [Tanay 2016]; we have demonstrated how to build adversarial attacks that will transfer to any other consistent classifiers; (ii) we have also demonstrated that any sample from the low confidence subregion will lead to adversarial examples also in the low confidence subregion. We further describe the impact of **DFAL** for linear classifiers in Sec 4.6

4.5.2 DFAL does not select random samples in the first runs

DFAL is very promising empirically. However, for complicated network architectures with millions of parameters like VGG8, but trained on a small labeled set, it seems plausible that any example is vulnerable to small adversarial attacks. We clarify this hypothesis and explain why we do not observe such behavior in practice.

Independently of the number of parameters of the network, [Fawzi 2017] have empirically observed that state-of-the-art deep networks learn connected classification regions instead of shattered and disconnected regions. Although such classification regions defined in the input space may suffer from the curse of dimensionality, eventually few directions interfere with the decision boundaries. Considering now the low dimensional space defined by these impacting directions, it becomes likely that the samples do not suffer anymore from the curse of dimensionality and, thus the distance to the decision boundary will differ among the samples. Hence, even in the first iterations of **DFAL**, we expect the magnitude of the smallest adversarial perturbations to be diverse enough so not to select samples randomly.

Finally, we observe in Figure 4.8 that adversarial perturbations are far from being constant. We believe that the underlying topology of classification regions of deep networks explains the efficiency of our method, even in the first runs.

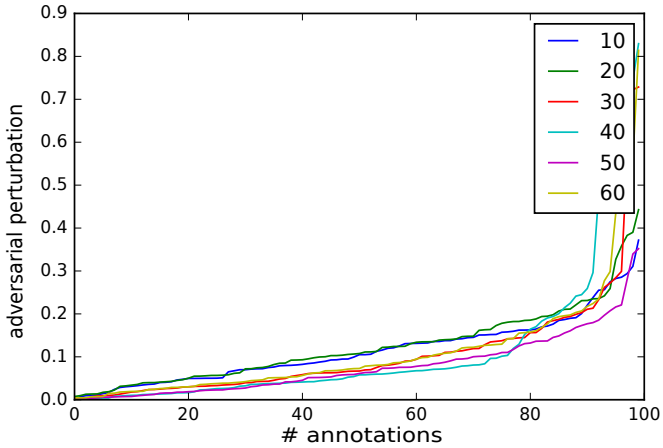


Figure 4.8: range of adversarial perturbations (i.e. distances between samples and their adversary) for VGG8trained on *MNIST*with 10, 20, 30 ... to 100 labeled examples. A curve corresponds to the range of adversarial pertubation found on the unlabeled example, while its color matches the size of the labeled set used to train the network

4.6 Adversarial Active Learning for Linear Classifiers

We point out specific cases in which we can obtain a significant improvement in the labeled data sample complexity using adversarial active learning for linear classifiers. We restrict our case of study to a specific case; which is when the data instances are drawn from the unit ball in $\mathbb{R}^2$ and their labels are drawn from ± 1. Notice that our proof may be extended to other distributions as long as they are uniformly distributed along with any dimension (*such as isotropic gaussian*). Throughout this section, our goal is to find a linear classifier f going through the origin, so that its expected true loss is as small as possible. The error is induced by the classification rule $2\mathbb{I}(f(x) \geq 0) - 1$ where $\mathbb{I}(\cdot)$ is the set of indicator functions. We consider the following classification error loss defined as l(f(x),y)=1 if $yf(x) \leq 0$ and $l(f(x), y) = 0$ otherwise.

Firstly, we detail our strategy when the labels are consistent with a linear separator going through the origin. While we knew already from the literature that active learning is highly beneficial for such a case, ensuring a need of $\tilde{\mathcal{O}}(d \ \ln(\frac{1}{\varepsilon}))$ labeled examples, given ε as the error rate and d the dimension, we will see how adversarial active queries help to diminish the effective numbers of labels queried.

Indeed, [Balcan 2007] demonstrated that to obtain an exponential improvement in the label sampled complexity, one needs to sample the examples from a subregion carefully chosen and not from the entire region of uncertainty. When sampling uniformly along the unit ball, few samples lie in such low confidence regions. Although, to achieve an error rate of $2^{-(k+1)}$ at the k-th iteration, we still need to add $\tilde{\mathcal{O}}(2^k d)$ unlabeled samples[3], we can automatically guess the ground-truth labels of the majority of them. Given the current linear classifier c_k consistent on the labeled examples at iteration k and a given threshold b_k, every unlabeled sample x_k lying further from the decision boundary than b_k is necessarily predicted correctly by the current classifier c_k. This result relies on the assumption made on the data distribution and its separability using a linear classifier [Balcan 2007]. When sampling uniformly queries and considering $b_k = 2^{-k}\pi$, we can estimate the probability for any sample x to be part of the low confidence regions as $p(\mid c_k \cdot x \mid \leq b_k) = \tilde{\mathcal{O}}(2^{-k}\sqrt{d})$. Hence, in the original strategy proposed in [Balcan 2007], a human annotator effectively annotates $\tilde{\mathcal{O}}(d^{\frac{3}{2}})$ unlabeled samples at each iteration to obtain an exponential improvement in the error rate.

Here we argue how adversarial queries may help to reduce the number of effective labels at any iteration $k > 1$.

4.6.1 Transferable adversarial attacks

When it comes to deep networks, their adversarial attacks can transfer across many other models: adversarial examples generated for a specific model will often mislead other unseen networks. Such a property is commonly known as transferability. However, transferability has been mainly observed empirically [Goodfellow 2015].

[3] according to the VC dimension of linear classifiers

Up to our knowledge, how to understand the underlying phenomena and how to defend against them effectively are still open questions. Meanwhile, [Tanay 2016] have investigated the phenomenon of adversarial examples for binary linear classifiers. They proposed a new taxonomy to classify adversarial attacks: they defined the notion of *adversarial strength* and show that it can be reduced to the deviation angle between the classifier considered and the nearest centroid classifier (*i.e the bissecting hyperplane between positive and negative samples*).

The probability of transferability of an adversarial attack directly depends on the level of regularization used; more specifically to the deviation angle between the classifier and the bissecting hyperplane between positive and negative samples.

Based on the notion of *adversarial strength*, we define **weak** adversarial examples. Weak adversarial examples will not transfer to any other consistent classifier, other than the one they have been designed for. They result from a lack of regularization, which can be improved by adding the adversarial sample to the training set. Similarly, as for **DFAL**, we can use twice the same label for any sample and its weak adversarial counterpart. If one is able to design weak adversarial examples given a labeled sample x, then we can increase the training set without corrupting it. Eventually, the weak adversarial sample will have the same label as x.

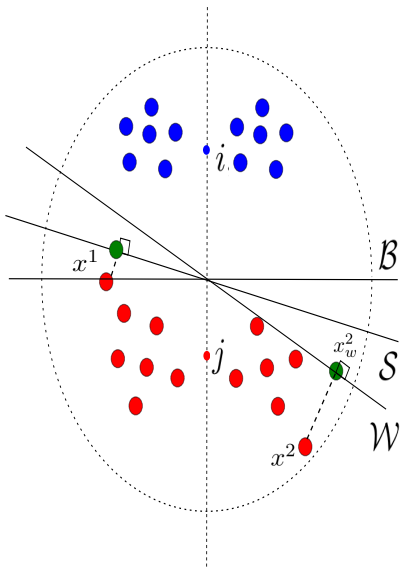


Figure 4.9: Toy problem: learning a linear separator that predicts with no error the labels of positive instances •, and negative instances •. We illustrate the notion of weak adversarial examples • on two samples x^1 and x^2.

We detail the procedure to build weak adversarial attacks for linear classifiers in Theorem 6.1. To build our adversarial attacks, we stick to the standard of the litterature by adding a perturbation along the gradient direction [Goodfellow 2015]. The strength of the adversarial example is directly impacted by the deviation angle and the magnitude of the perturbation. We illustrate our strategy on a 2-dimensional toy problem in Figure 4.9. Consider instances distributed in a circle such that

positive and negative points may be well separated given a linear classifier going through the origin. $\mathcal{B}$, in accordance with Definition 1, is the bissecting line between positive and negative points. For the sake of clarity, we centered $\mathcal{B}$ to go through the origin. Several optimal separators coexist. Among them, we consider the one which maximizes its angle with $\mathcal{B}$ (*i.e* $\mathcal{W}$ *in def 2*), and the one wich minimizes it (*i.e* $\mathcal{S}$ *in Definition 2*). Every other solution necessarily lies between $\mathcal{W}$ and $\mathcal{S}$.

We describe for two points x^1 and x^2 in Figure 4.9 how to build their weak adversarial counterparts, based on Definition 2. Note that a necessary condition is that both points x^1 and x^2 considered are well predicted by our strong and weak classifiers. The mirror projection of x^1 given $\mathcal{S}$ will lie in the hypothesis space (*a.k.a in the area between* $\mathcal{W}$ *and* $\mathcal{S}$). When it comes to x^2, projecting it on $\mathcal{W}$ ensures that every consistent classifier will predict x^2_w as a negative instance.

Definition 4.6.1: Bissecting Hyperplane

According to [Tanay 2016], we define the bissecting hyperplane $\mathcal{B}$ as a unique linear separator of unit vector $\mathbf{b}$ and bias b_0 such that $\mathcal{B}$ reflects the mean of positive instances on the mean of negative instances. Note that $\mathcal{B}$ is not necessarily part of the hypothesis space, nor $\mathcal{B}$ minimizes the error on $X \times Y$.

$$j = i - 2(i \cdot \mathbf{b} + b_0)\mathbf{b} \; s.t \;\; i = \mathbb{E}[X \mid Y = 1], \; j = \mathbb{E}[X \mid Y = -1]$$

Definition 4.6.2: Transferable adversarial attacks

Given $\mathcal{L}(X \times Y)$ the set of optimal classifiers given the task at hand, we define two boundary classifiers: $\mathcal{S}$ the strong linear classifier of unit vector $\mathbf{s}$, and $\mathcal{W}$ the weak linear classifier of unit vector $\mathbf{w}$. $\mathcal{S}$ is consistent with the training set and minimizes the deviation angle with $\mathcal{B}$. $\mathcal{W}$ is consistent with the training set and maximizes the deviation angle with $\mathcal{B}$.

$$\mathcal{L}(X \times Y) = \{\mathcal{C} \mid \forall\, (x, y) \in X \times Y \; y(x \cdot \mathbf{c}) > 0\} \tag{4.1}$$

$$\mathcal{S} = argmin_{\mathcal{S} \in \mathcal{L}(X \times Y)} \mathbf{s} \cdot \mathbf{b} \tag{4.2}$$

$$\mathcal{W} = argmax_{\mathcal{W} \in \mathcal{L}(X \times Y)} \mathbf{w} \cdot \mathbf{b} \tag{4.3}$$

$$\tag{4.4}$$

$\forall (x, y) \in X \times Y$ we define its weak adversarial attack, $\tilde{x}_w$, based on the following:

- if $|\mathbf{w} \cdot x| \leq |\mathbf{s} \cdot x|$: $\tilde{x}_w = x - (x \cdot \mathbf{w})\mathbf{w}$
- if $|\mathbf{s} \cdot x| \leq |\mathbf{w} \cdot x|$: $\tilde{x}_w = x - (x \cdot \mathbf{s})\mathbf{s}$

Theorem 6.1: Weak Adversarial Examples

$\forall (x, y) \in X \times Y$, $y(\tilde{x}_w \cdot \mathbf{c}) \geq 0$

Notice that our definition of adversarial attacks does not match exactly the common definition as we do not restrict our adversarial attacks to be close to their target sample anymore.

4.6.2 Label Complexity on the unit ball

Here we argue how weak adversarial queries help to reduce the number of effective labels at any iteration $k > 1$. Our active learning strategy consists in adding also weak adversarial instances to the training set when it is relevant, as proposed for deep networks with **DFAL**. Thus we will reduce the effective need of queries by a ratio of two at best. Indeed, weak adversarial instances are relevant if and only if the sample queried is already well predicted by the current weak and strong classifiers. In Theorem 6.2, we describe further the expected improvement in terms of human annotations.

A first observation is that projecting the unit ball according to any hyperplane going through the origin corresponds to the identity mapping. Consequently, when adding weak adversarial examples in the training set, we do not modify the underlying distribution of the instance space. Moreover, the main advantage of our adversarial examples is that for any instance lying in the low confidence region, its weak adversarial examples will also lie in that subregion (*Lemma 6.1*). It means that when using adversarial queries, we respect the $i.i.d$ assumption, and query relevant samples, as illustrated in Figure 4.10. Finally the number of artificial queries that can be added mostly depend on the generalization error at the current iteration: when a sample query is correctly predicted, we can add its weak adversarial attacks.

Lemma 6.1: Low confidence region

$\forall (x, y) \in X \times Y$, $\forall \mathcal{C} \in \mathcal{L}(X \times Y)$: $\forall \alpha \in \mathbb{R}^+$ such that $\mid \mathbf{c} \cdot x \mid \leq \alpha$ then we have $\mid \mathbf{c} \cdot \tilde{x}_w \mid \leq \alpha$

Theorem 6.2: Convergence of adversarial queries

Given n $=\tilde{\mathcal{O}}(d^{\frac{3}{2}})$ the effective number of labels to query at iteration k. We denote the generalization error at step k, $p_k = 2^{-(k+1)}$.

Using our adversarial strategy (adding both $\tilde{x}_w$ and $\tilde{x}_s$), we can reduce the effective number m_k of labels with high probability $\delta > 0$ up to:

$$m_k = \min\{m \geq \frac{n}{2} \mid \binom{m-1}{m-\frac{n}{2}}(1-p_k)^{\frac{n}{2}} p_k^{m-\frac{n}{2}} \geq \delta\}$$

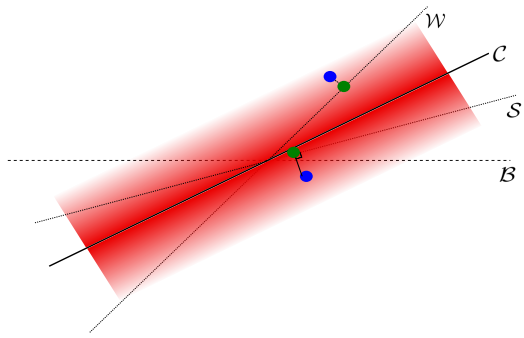


Figure 4.10: Repartition of weak examples • for samples • lying in the low confidence subregion of a consistent classifier $\mathcal{C}$

4.7 Conclusion

In this chapter, we proposed a new active learning heuristic, called **DFAL**, to perform margin-based active learning for CNNs: we approximate the projection of a sample to the nearest decision boundary using its smallest norm adversarial attack. We demonstrate empirically that our **DFAL** strategy is highly efficient for CNNs trained on various image classification benchmarks. We are not only competitive with the state-of-the-art approach CORE-SET, but we also outperform that method for runtime performance. Thanks to the transferability of adversarial attacks, **DFAL** is a promising approach for combining active learning with model selection for deep networks.

Chapter 5

Perspective: Bayesian Active Learning through Laplace Approximation

Contents

5.1 Introduction **51**
5.1.1 Active Learning under the light of Variational Inference . . . 53
5.2 Covering **56**
5.2.1 Optimal Experimental Design 56
5.2.2 Increasing the diversity 57
5.3 Application to CNN **58**
5.4 Application to RNN **59**
5.5 Empirical Validation **61**
5.6 Future work **63**
5.7 Conclusion **64**

5.1 Introduction

Summary

- We propose a new bayesian active learning strategy for Deep Networks using the variational free energy. (*see 5.1.1*).
- We provide a tractable upper bound based on Optimal Experimental Design (**OED**) (*see 5.1.1 and 2.3*)
- We design a diversity regularizer based on Wasserstein distance (*see 5.1.1*)
- We rely on Fisher approximation to scale **OED** to both CNN and Recurrent Neural Networks (RNNs) (*see 5.1.1*)

mducoffe/AL_VFE PROOF Proofs are available in **Appendix A.3.2**

Instead of measuring the uncertainty for only one model, Bayesian Neural Networks offer the possibility to evaluate the uncertainty through an ensemble of models. In a Bayesian context, a neural network is considered as a parametric model which assigns a conditional probability on the observed labeled data $\mathcal{A}$ given a vector of weights $\mathbf{w}$. If the weights follow some prior distribution $\Pr(\mathbf{w} \mid \alpha)$ (*depending on the parameter* α), the posterior distribution of the weights can be written as $\Pr(\mathbf{w} \mid \mathcal{A}, \alpha)$. We are interested in finding the most probable weights that have generated our data, i.e. the posterior over the weights given our observables. Such a probabilistic uncertainty is highly relevant in an active learning setting as it consequently leads to a pertinent exploration of the underlying distribution of the input data. This strategy is similar to **QBC**, with the main difference that the size of the committee might be infinite instead of being discrete as in Section 3.

However computing the posterior distribution of the weights $\Pr(\mathbf{w} \mid \mathcal{A}, \alpha)$ is usually intractable for deep networks. Instead, one can approximate the posterior with a variational distribution $Q(\mathbf{w} \mid \beta)$ whose structure leads to an easier evaluation. Only few works have attempted to estimate the distribution of the weights of a CNN, mainly due to the high dimensional parameter space inherent with such models.

A possible solution consists in considering Bayesian inference as an optimization process and thus by minimizing the variational free energy [Feynman 1972, Neal 1998]. Such an approximation has already been exploited by [Graves 2011] for deep networks [1]. The posterior distribution $\Pr(\mathbf{w} \mid \mathcal{A}, \alpha)$ is approximated with a tractable distribution $Q(\mathbf{w} \mid \beta)$ depending on a new parameter β. The quality of the approximation $Q(\mathbf{w} \mid \beta)$ compared to the true posterior $\Pr(\mathbf{w} \mid \mathcal{A}, \alpha)$ is measured by their Kullback-Leibler divergence, with $\mathcal{L}$ the log-likelihood.

The approximation quality is equivalently measured by the variational free energy $\mathcal{F}$, which can be expressed as a minimum description length loss function:

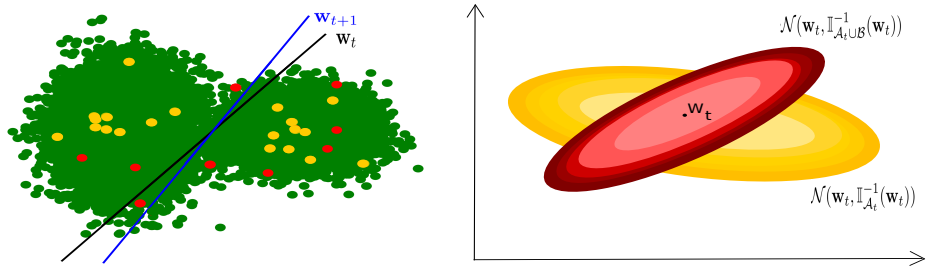
$$\mathcal{F}(\mathcal{A}) = \mathbb{E}_{\mathbf{w} \sim Q(\mathbf{w}|\beta)}\Big(\mathcal{L}(\mathcal{A}; \mathbf{w})\Big) + KL\Big(Q(\mathbf{w} \mid \beta) \mid\mid P(\mathbf{w} \mid \alpha)\Big) \tag{5.1}$$

When it comes to active learning, deducing an approximating variational distribution through Eq. (5.1), is not intuitive. The main drawback lies in the additional hyperparameters(*such as the mean and the covariance matrix if assuming Gaussian distribution*) required by the method, which drastically increases the complexity of the training stage. For example, [Blundell 2015] use a Gaussian distribution for their approximate variational distribution which has doubled the number of parameters in the model without outperforming state-of-the-art performance.

In this section, we propose a Bayesian batch active learning method for CNNs. We derive the use of the variational free energy at test time, to evaluate how the approximating variational distribution generalizes to new unseen data. Eventually, our Bayesian Active Learning method 5.1.1 queries the batch of unlabeled data which maximizes the variational free energy.

[1] For the sake of consistency we stick to his notations

5.1.1 Active Learning under the light of Variational Inference



(a) classification with a shallow logistic classifier

(b) *Laplace approximation* of the prior $P(\mathbf{w}_t \mid \alpha)$ and the posterior distribution of the weights $Q(\mathbf{w}_t \mid \beta)$

Figure 5.1: Illustration of our active learning criterion for linear separators (*see* Eq. 5.11): $\mathcal{A}_t$ ={ • }, $\mathcal{B}$ ={ • }, $\mathcal{P}\backslash\{\mathcal{A} \cup \mathcal{B}\}$ ={ • }. We learn a logistic classifier with no bias on a 2D binary classification task. The data are made of a mixture of two Gaussians (on the left lie positive examples and the right negative examples). The yellow points $\mathcal{A}_t$ are labeled, and $\mathbf{w_t}$ is learn on them with no error. We then select 9 unlabelled data $\mathcal{B}$ to minimize our variational free energy: the set of points $\mathcal{B}$ whose induced posterior distribution $Q(\mathbf{w}_t \mid \beta)$ will diverge at most from the prior distribution $P(\mathbf{w}_t \mid \alpha)$. $\mathbf{w_{t+1}}$ is the classifier trained on $\mathcal{A}_t \cup \mathcal{B}$.

Firstly, we sum up our batch active learning strategy in Method 5.1.1.

Method 5.1.1: Bayesian Active Learning

We have a pool of unlabeled data $\mathcal{P}$ and start training our CNN with a small set of training samples $\mathcal{A}$. This is the initial state of our active learning training set $\mathcal{A}_0 = \mathcal{A}$. At each iteration t, we aim at selecting the *optimal* batch $\mathcal{B}$ by computing a new loop of the following steps:

1. the network is trained on the current training set $\mathcal{A}_t$ leading to the weights $\mathbf{w}_{t+1}$
2. we search for the *optimal* batch $\mathcal{B}$ of samples to be added to the training set, i.e. the batch $\mathcal{B}$ maximizing the *variational free energy*

$$\mathcal{B} = argmax_{\mathcal{B}}\, \mathcal{F}_{\mathbf{w}_t}(\mathcal{A}_t \cup \mathcal{B})$$

3. we select a subset $\tilde{\mathcal{B}} \subset \mathcal{B}$ of fixed size such that $\mathcal{A} \cup \tilde{\mathcal{B}}$ follows at best the unknown input distribution (*see section 5.2*).
4. the training set is then augmented by $\tilde{\mathcal{B}}$: $\mathcal{A}_{t+1} = \mathcal{A}_t \cup \tilde{\mathcal{B}}$

For clarity sake, in the following we rename $\mathcal{F}_{\mathbf{w}_t}$ by $\mathcal{F}$ (keeping in mind that we target the selection of the *optimal* batch $\mathcal{B}$, thus the weights $\mathbf{w}_t$ of the network remain fixed after *step* 1 until *step* 4).

Our **active criterion** thus corresponds to select the batch $\mathcal{B}$ maximizing $\mathcal{F}(\mathcal{A}_t \cup \mathcal{B})$. From Eq. (5.1), it consists in the minimization of the sum of two terms which, in accordance with [Graves 2011], we denote respectively by:

- the *training factor* $\mathbb{E}_{\mathbf{w}\sim Q(\mathbf{w}|\beta)}\Big(\mathcal{L}(\mathcal{A}\cup\mathcal{B};\mathbf{w})\Big)$,
- the *generalization factor* $KL\Big(Q(\mathbf{w} \mid \beta) \mid\mid P(\mathbf{w} \mid \alpha)\Big)$.

Those two terms require to be able to compute both the prior distribution of the weights $P(\mathbf{w} \mid \alpha)$ and the approximation $Q(\mathbf{w} \mid \beta)$ of the posterior distribution $P(\mathbf{w} \mid \mathcal{A}\cup\mathcal{B}, \alpha)$.

Here we consider the *Laplace approximation* [MacKay 1992, Ritter 2018]. It imposes a Gaussian distribution on $Q(\mathbf{w} \mid \beta)$ whose covariance is estimated from the Hessian of the model, evaluated at the variational mode $\mathbf{w}_t$. The covariance corresponds to $\mathbb{I}^{-1}_{\mathcal{A}_t\cup\mathcal{B}}(\mathbf{w}_t)$, a quantity also denoted as the empirical *inverse Fisher information* matrix. When considering our active criterion, the *Laplace approximation* holds two main advantages. First, it allows inferring $Q(\mathbf{w} \mid \beta)$ at test time, without impacting the training phase. Secondly, the assumption of a Gaussian distribution, instead of a Gaussian mixture as in [Gal 2016a], simplifies the variational steps when computing $\mathcal{F}$, so to obtain an analytical expression of our active criterion.

5.1.1.1 BALNet: Batch Active Learning Networks

At the beginning of an active learning step, the current weight distribution given the labeled dataset $\mathcal{A}_t$ defines our prior. The posterior distribution is computed on both the labeled data and the query batch $\mathcal{B}$. The next equations define the formulation of our prior and posterior:

$$P(\mathbf{w}_t \mid \alpha) \sim \mathcal{N}(\mathbf{w}_t, \mathbb{I}^{-1}_{\mathcal{A}_t}(\mathbf{w}_t)) \tag{5.2}$$

$$Q(\mathbf{w}_t \mid \beta) \sim \mathcal{N}(\mathbf{w}_t, \mathbb{I}^{-1}_{\mathcal{A}_t\cup\mathcal{B}}(\mathbf{w}_t)) \tag{5.3}$$

For the sake of clarity, because each Fisher matrices considered are evaluated at $\mathbf{w}$, we skip it from now in the notations.

However, when the variational free energy evaluated on $\mathcal{A}\cup\mathcal{B}$ increases, it induces that, given $\mathcal{B}$, the quality of our posterior approximate, $Q(\mathbf{w} \mid \beta)$, is getting worse to represent the posterior distribution $\Pr(\mathbf{w} \mid \mathcal{A}\cup\mathcal{B})$. Thus, our assumption to consider the weights $\mathbf{w}$ as the variational mode in Eq. (5.3) is not valid anymore. Consequently, it is relevant to add the data maximizing our active criterion, to update the weights $\mathbf{w}$.

Next we define tractable lower bounds for both terms in **BalNet**: the *training factor* and the *generalization factor*.

• **The Training factor** is intractable and is always approximated, generally through sampling [Graves 2011]. However, sampling on the approximate posterior $Q(\mathbf{w} \mid \beta)$ requires to compute the inverse of the Fisher matrix for every possible batch $\mathcal{B}$ (Eq. 5.2). To overcome this computational issue, we opt for a second-order linear approximation of the expectation of the log-likelihood. Note that we evaluate our loss only on the labeled data. For any random vector whose expected mean and covariance are known, the expectation of a quadratic form can be expressed [Mathai 1992]. Eventually our evaluation of the *training factor* becomes:

$$\mathbb{E}_{\mathbf{w}\sim Q(\mathbf{w}|\beta)}\Big(\mathcal{L}(\mathcal{A};\mathbf{w})\Big) \approx \mathcal{L}(\mathcal{A};\mathbf{w}) - \frac{1}{2}\mathbf{w}^T\mathbb{I}_{\mathcal{A}}\mathbf{w} - \frac{1}{2}Tr(\mathbb{I}^{-1}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}}) \tag{5.4}$$

• **The generalization factor** corresponds to the KLdivergence between the approximate posterior $Q(\mathbf{w} \mid \beta)$ and the prior $P(\mathbf{w} \mid \alpha)$. Because both distributions are multivariate Gaussian of N parameters, we have a direct formulation of the KLwhich is always definite since the Fisher matrices are invertible. Moreover, for computational efficiency, we want to discard the determinant as its complexity is cubic in the size of the batch $\mathcal{B}$. Hence, we lower bound the logarithm of the determinant by a function of its trace. Eventually we provide an upper bound on the *generalization factor* in Eq. 5.8. Note that to obtain our upper bound we assume that every Fisher matrices are Hermitian:

$$\begin{aligned} KL\Big(Q(\mathbf{w} \mid \beta) \mid\mid \mathcal{P}_\alpha(\mathbf{w})\Big) &= KL\Big(\mathcal{N}(\mathbf{w}, \mathbb{I}^{-1}_{\mathcal{A}\cup\mathcal{B}}(\mathbf{w})) \mid\mid \mathcal{N}(\mathbf{w}, \mathbb{I}^{-1}_{\mathcal{A}}(\mathbf{w}))\Big) && (5.5)\\ &= \tfrac{1}{2}\Big(Tr(\mathbb{I}^{-1}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}} + (\mathbf{w}-\mathbf{w})^T\mathbb{I}_{\mathcal{A}}(\mathbf{w}-\mathbf{w})^T && (5.6)\\ &\qquad + \log(|\mathbb{I}^{-1}_{\mathcal{A}}\mathbb{I}_{\mathcal{A}\cup\mathcal{B}}|) - N\Big) && (5.7) \end{aligned}$$

We use the following inequality that holds for every Hermitian matrix M:

$$\frac{1}{N}\log(|M^{-1}|) \geq \log(N) - \log(Tr(M))$$

$$KL\Big(Q(\mathbf{w} \mid \beta) \mid\mid \mathcal{P}_\alpha(\mathbf{w})\Big) \geq \frac{1}{2}\Big(Tr(\mathbb{I}^{-1}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}}) + N + N\log(N) - N\log\big(Tr(\mathbb{I}^{-1}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}})\big)\Big) \tag{5.8}$$

• **The variational free energy** is the sum of the *training factor* and the *generalization factor*. Eventually, based on the previous analytic formulations, we are able to upper bound the *variational free energy* by **BalNet**defined in Eq. (5.11). Note that we discard the terms constant w.r.t. $\mathcal{B}$ in our criterion and do not consider the log as it is a strictly increasing function.

$$\begin{aligned} \mathcal{F}_{\mathbf{w}}(\mathcal{A}\cup\mathcal{B}) &= KL\Big(Q(\mathbf{w} \mid \beta) \mid\mid \mathcal{P}_\alpha(\mathbf{w})\Big) + \mathbb{E}_{\mathbf{w}\sim Q(\mathbf{w}|\beta)}\Big(\mathcal{L}(\mathcal{A};\mathbf{w})\Big)\\ &\geq \frac{1}{2}Tr(\mathbb{I}^{-1}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}}) - N\log\big(Tr(\mathbb{I}^{-1}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}})\big) - \frac{1}{2}Tr(\mathbb{I}^{-1}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}}) + Const \end{aligned} \tag{5.9}$$

$$max_{\mathcal{B}}\mathcal{F}_{\mathbf{w}}(\mathcal{A}\cup\mathcal{B}) \geq max_{\mathcal{B}}\Big(- N\log\big(Tr(\mathbb{I}^{-1}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}})\big) + Const\Big) \tag{5.10}$$

According to [Hoi 2006], we thus search for the optimal batch $\mathcal{B}$ such that:

$$\boxed{\mathcal{B} = argmax_{\mathcal{B}}\Big(Tr(\mathbb{I}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}^{-1}_{\mathcal{A}})\Big)} \tag{5.11}$$

5.2 Covering

5.2.1 Optimal Experimental Design

Following a different path of reasonning, our criterion is similar to the theoretical foundations developed by Zhang *et al.* in [Zhang 2000] (*see Section 2.3*), since their goal is to search for a set of examples which can reduce at best the Fisher information matrix: $argmin_{\mathcal{B}}\, Tr(\mathbb{I}^{-1}_{\mathcal{B}}\mathbb{I}_{\mathcal{A}})$. Indeed, [Zhang 2000] proposed a batch mode extension of A-optimality. They studied active learning by looking for the best resampling of the unlabelled input data. They consider as the optimal resampling the one which minimizes the negative expected log likelihood.

It led them to formulate a criterion on the asymptotic expected log likelihood of the resampling Fisher matrix $\mathbb{I}_q$, given q the resampled distribution of p, the original distribution of the input data, with $\mathbb{E}^n$ the expectation over n samples from q:

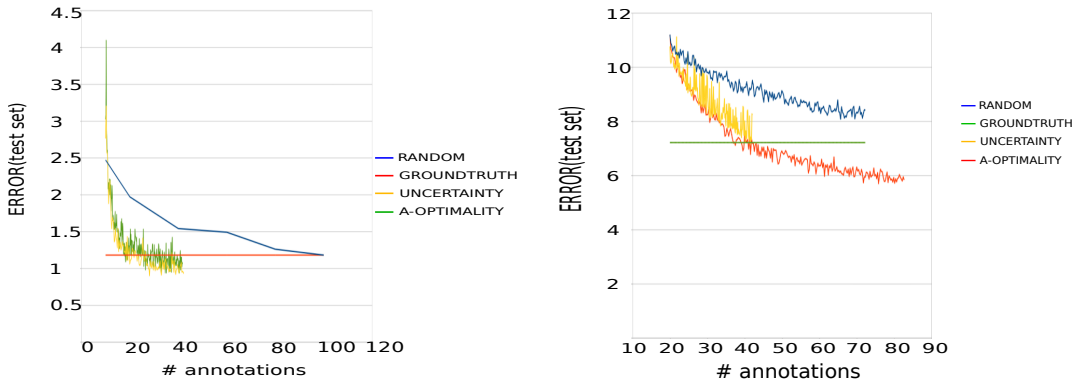
$$\mathbb{E}^n(\theta) = -\frac{1}{2n}Tr(\mathbb{I}_q(\theta)^{-1}\mathbb{I}_p(\theta)) \tag{5.12}$$

As long as we restrict the context to log likelihood, the Cramer Rao bound implies that the MLE parameter Θ which minimizes $\mathbb{E}^n(\theta)$ is the asymptotic most efficient estimator of the optimal parameter among all estimators based on resampling of the input distribution. To apply this result, they proposed to use a good empirical estimate $\hat{\Theta}$ of Θ and then replace the criterion Θ by its approximation in order to estimate the optimal resampled distribution q^*. They estimate $\hat{\Theta}$ by the trained parameters of their algorithm on the currently labeled samples:

$$q^* = argmin_q Tr(\mathbb{I}_q(\hat{\Theta})^{-1}\mathbb{I}_p(\hat{\Theta})) \tag{5.13}$$

More samples can then be drawn so to re-estimate $\hat{\Theta}$ as well as the optimal distribution q^*. However, looking for a subset sampled from the optimal distribution q^* is not feasible as it is exponential in the number of unlabeled samples. Following this path, [Hoi 2006] used this previous criterion and approximated the solution but for logistic functions only.

The fact that our criterion (*see Method 5.1.1*) is approximately upper bounded by A-optimality is interesting because it may highlight some flaws in **OED** and why it failed to perform better to perform better than uncertainty selection in Figure 5.2. In the next Section 5.2.2, we provide details in the needs of regularizing our active learning criterion, using a diversity scheme.



(a) MNIST (b) SVHN

Figure 5.2: **Empirical evaluation of A-optimality on CNNs**: Test error achieved for different percentage of the whole training set. We use *MNIST*5.2(a) and *SVHN*5.2(b). We compare A-optimality (*by approximating the Fisher matrix with KFAC, see Section 5.3*) against random selection and uncertainty selection [Ducoffe 2016b]. We consider 2 CNNs whose hyperparameters are fully described in Appendix A.2.3.

5.2.2 Increasing the diversity

Variational free energy measures the approximating variational distribution and the posterior distribution. It is done on a fixed set of data that represents the distribution. The objective is to preserve the statistics of the input distribution. Indeed, when selecting a batch of data, we assume that our training data are uniformly sampled from the input data distribution.

When digging into the formulation of our criterion, the KLdivergence between our approximate distribution $Q(\mathbf{w}_t \mid \beta)$ and the true posterior distribution $\Pr(\mathbf{w} \mid \mathcal{A} \cup \mathcal{B}, \alpha)$ may be expressed as the sum of both terms:

1. the variational free energy: $\mathcal{F}(\mathcal{A})$
2. the log evidence of the data: $log(\Pr(\mathcal{A}))$

In our context, we expect the log evidence to remain constant along the training. Because, if so, the variations of the variational free energy will directly impact the variations on the KLdivergence, as illustrated in Eq. 5.14. Otherwise, if the underlying distribution of $\mathcal{A}$ mismatches with the ground-truth source distribution, our **AL**method will promote new data to optimize our approximating variational distribution given a biased estimate of the true posterior distribution. Eventually, to guarantee satisfactory performance of our CNN, we need to preserve the data distribution as much as possible in the training set. However, this assumption may not hold in active learning, since we favor instances not complying with it.

$$if\ \forall \mathcal{A}\ log(\Pr(\mathcal{A})) \simeq Const\ \ then\ \mathcal{F}(\mathcal{A}) \searrow \implies KL(Q(\mathbf{w}_t \mid \beta) || \Pr(\mathbf{w} \mid \mathcal{A} \cup \mathcal{B}, \alpha)) \searrow \tag{5.14}$$

To promote the diversity among our training set, we advocate the use of a distance on distributions, the 2-Wasserstein distance. The Wasserstein distance is a powerful tool based on the theory of optimal transport to compare data distributions. It knows a renewed interest thanks to its success in generative modeling and image processing. While, to our knowledge, Wasserstein distance has not yet been investigated in active learning, several of its properties are nevertheless highly relevant in our context. Wasserstein can be applied to distributions with non-overlapping supports and has good out-of-sample performance. Moreover, it is robust to discrete distributions without the need to resort to kernel estimators, and is parameter-free, unlike Maximum Mean Discrepancy [Muandet 2017].

Therefore selecting a more diverse subset among our queries may be solved by minimizing the Wasserstein distance between our queries and the pool of data $\mathcal{P}$ (both labeled and unlabeled). We consider euclidian distance for computing Wasserstein. Formally given a fixed size of data to label, K, we propose in Eq. 5.15 a subset selection $\tilde{\mathcal{B}}$ among our candidates $\mathcal{B}$, to reduce the bias of our active selection scheme.

$$\tilde{\mathcal{B}} = \underset{\tilde{\mathcal{B}} \subset \mathcal{B},\ |\tilde{\mathcal{B}}| = K}{\arg\min}\ \mathbb{W}_2(\mathcal{A} \cup \tilde{\mathcal{B}} \,||\, \mathcal{P}) \tag{5.15}$$

The optimization subroutine we defined previously in Eq. 9.1 is generally not tractable as it is combinatorial given $\mathcal{B}$. However, we can obtain a reasonably good approximation with a greedy selection. We invite the reader to read Chapter 9 for further properties about using greedy search for selecting our queries given Wasserstein.

5.3 Application to CNN

The main limitation of the *Laplace approximation* is that, in most cases, the Fisher matrix may not be stored in memory. To solve this issue, we consider a recent approximation of the Fisher information for CNNs proposed in [Martens 2015] and [Grosse 2016], called Kronecker Factored Approximate Curvature (KFAC). The Fisher information is approximated as a block diagonal matrix. Each block itself is an approximation of the Fisher information related to the weights of a layer. Such a structure enables to capture the correlations between parameters of the same layers. Because the covariance matrix from a single layer may not be even stored in memory, the Fisher information of each layer is approximated by a Kronecker product $\psi \otimes \tau$. A summary of their decomposition is presented in Eq. (5.16)[2].

[2]Note that the previous works do not handle the convolutional biases and batch normalisation parameters. Because the number of parameters involved is limited, we compute the exact observed Fisher matrices for both convolutional biases and batch normalization parameters.

$$\mathbb{I}_{\mathcal{A}} = diag([\psi_{\mathcal{A},l} \otimes \tau_{\mathcal{A},l}]_{l=1}^{L}) \tag{5.16}$$

$$\psi_{\mathcal{A},l} = \frac{1}{|\mathcal{A}|} \sum_{(x_i,y_i)\in\mathcal{A}} \psi_{i,l} \tag{5.17}$$

$$\tau_{\mathcal{A},l} = \frac{1}{|\mathcal{A}|} \sum_{(x_i,y_i)\in\mathcal{A}} \tau_{i,l} \tag{5.18}$$

$$\tag{5.19}$$

Finally, we express the trace $Tr(\mathbb{I}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}}^{-1})$ based on the KFAC approximation of the Fisher matrix: we consider that every Fisher matrix for a CNN is an L diagonal block matrix, with L the number of layers of the CNN. Every block is made of a Kronecker product of two terms ψ and τ. We rely on the properties involved in the choice of this specific matrix topology to obtain in Eq. 5.20 a tractable approximation of $Tr(\mathbb{I}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}}^{-1})$. It allows us to express our **active learning criterion** as a linear function given the Kronecker coefficients of the queries.

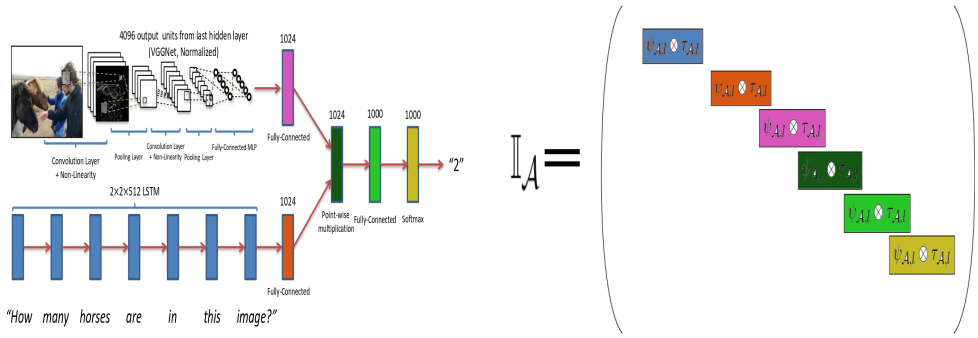
$$Tr(\mathbb{I}_{\mathcal{A}\cup\mathcal{B}}\mathbb{I}_{\mathcal{A}}^{-1}) = \sum_{l=1}^{L} Tr(\psi_{l,\mathcal{A}\cup\mathcal{B}}\psi_{l,\mathcal{A}}^{-1})Tr(\tau_{l,\mathcal{A}\cup\mathcal{B}}\tau_{l,\mathcal{A}}^{-1}) \tag{5.20}$$

One of the main contributions of this approach is that our active learning criterion itself is equivalent to minimize a submodular function [3]. Indeed it consists in a constrained minimization over the sum of submodular functions which is also submodular. Submodular functions are widely used in active learning [Wei 2015]. Although so far, they have been used primarily in a context of maximization, since a greedy selection scheme may efficiently approximate their optimization [Nemhauser 1978]. On the other side, minimizing a submodular function may be solved exactly in strongly polynomial time [Schrijver 2000]. Eventually, we can solve exactly **BalNet**, unlike other state-of-the-art approaches such as DFAL and CORE-SET, which relies on approximations (*of the distance to the decision boundary*, *of the covering radius*). Eventually, maximizing the variational free energy allows to take into account the correlations among the queries, unlike top score approaches used in previous Bayesian active learning techniques for CNNs (*cf. Section 2.3.6*).

5.4 Application to RNN

Initially, the KFAC approximation assumes that each weight matrix is involved in a unique mapping. Thus, the original KFAC approximation cannot be directly applied to the Fisher matrices of RNNs, in which the weights are repeated along the sequence. Next, we propose to exploit the recent work of Martens *et al.* , to adapt our active learning criterion to classification tasks on RNNs [Martens 2018]. Note that few works have undertaken active learning selection for sequence classification on RNNs. Perhaps the most notable works are [Yanyao Shen 2018, Lin 2017b],

[3] the proof is available in appendix A.3.2



(a) VQA network

(b) Approximate Fisher matrix for the VQA network

Figure 5.3: **Illustration of KFAC approximation on VQA network [Antol 2015]**. VQA stands for Visual Question Answering. It is a multimodal classification task that involves a lot of handcrafted descriptions to build the training set. Consequently, active learning is perfectly suited for VQA. While it requires a lot of labeled training data, it is likely than just feeding it with random data will not be enough to improve its performance, as first assumed in [Lin 2017b]: "*As a result with long tail distributions, it will likely result in redundant questions and answers while still having insufficient training data for rare concepts*"

which derived existing active learning techniques to respectively NER classification and Visual Question Answering (VQA). However, up to our knowledge, there exists no active learning strategy for RNNs that take into account the specificities of the architecture.

We illustrate beforehand the topology of the approximation of the Fisher matrix for a recurrent architecture in Figure 5.3.

To proceed with our goal of obtaining a tractable approximation to the Fisher matrix of RNNs, we will follow several approximating assumptions [Martens 2018]:

- independence of the parameters between different layers to approximate the Fisher matrix by a block diagonal matrix. This assumption already exists in the original KFAC approximation

- temporal homogeneity among the gradient contributions from different time-steps. Temporal homogeneity is a pretty mild approximation and is analogous to the frequently used *steady state assumption* from dynamical systems. It assumes that the gradient contributions does not depend on the two times considered but only on their deviation.

Thanks to previous assumptions, Martens *et al.* proposed the following decomposition of the Fisher matrix of a recurrent layer. Hence given T the length of

the sequences, we can approximate the conditional Fisher matrix $\mathbb{I}^T_{\mathcal{A},l}$ by a sum of $2T+1$ Kronecker products. Eventually, the Fisher matrix is the expectation over time steps of the conditional Fisher.

$$\begin{aligned}
\mathbb{I}_{\mathcal{A},l} &= \mathbb{E}_T\left[\mathbb{I}^T_{\mathcal{A},l}\right] \\
\mathbb{I}^T_{\mathcal{A},l} &= \sum_{\substack{d=t-s \\ t-s=\tau,\, t=1\ldots T \\ s=1\ldots T}} (T-\mid d \mid)(\psi_{\mathcal{A},d,l} \otimes \tau_{\mathcal{A},d,l}) \\
\psi^T_{\mathcal{A},d,l} &= \frac{1}{\mid \mathcal{A} \mid} \sum_{(x_i,y_i)\in\mathcal{A}} \psi_{i,d,l} \\
\tau^T_{\mathcal{A},d,l} &= \frac{1}{\mid \mathcal{A} \mid} \sum_{(x_i,y_i)\in\mathcal{A}} \tau_{i,d,l}
\end{aligned} \tag{5.21}$$

The main drawback of the formulation of Eq. 5.21 is that there is no trick to decompose the inverse of the Fisher matrix $\mathbb{I}^{-1}_{\mathcal{A}}$ into a Kronecker product. Such a trick is necessary for our framework, as there exists no property for the product of any matrix with a Kronecker product. Eventually, there is no analytical formulation of $\mathbb{I}_{\mathcal{A}\cup\mathcal{B},l}\mathbb{I}^{-1}_{\mathcal{A},l}$, that we can easily use, based solely on Eq. 5.21.

If you consider sequences of varying size, other assumptions need to be taken into account. Although we can compute the Fisher matrix given the conditional Fisher matrix for different varying time steps, the expectation over the time steps will vary depending on the samples selected. Thus, without assuming any additional structure such as relationships between the various Kronecker factors, this does not appear to be any efficient way to select the optimal batch. Thus it appears that we must make additional approximating assumptions to proceed. Following the approach of Martens *et al.* , we assume that the contributions to the gradients are independent across time or at least uncorrelated [4]. This results in:

$$\forall T, \forall d > 0 \implies \psi^T_{\mathcal{A},d,l} \otimes \tau^T_{\mathcal{A},d,l} = 0$$

Notice that those previous assumptions are not the only conditions imposed in [Martens 2018]. However, Martens *et al.* developed a regularizer based on the Fisher information, which requires to be tractable but also to have a fast computation of its inverse. Eventually, the approximation for the Fisher matrix of a recurrent unit reads:

$$\mathbb{I}_{\mathcal{A},l} = \mathbb{E}_T\left[T\right](\psi^T_{\mathcal{A},0,l} \otimes \tau^T_{\mathcal{A},0,l}) \tag{5.22}$$

Every dataset and parameters used to conduct our experiments are available in the **dataset** section A.1 and the **hyperparameter** section A.2.3

5.5 Empirical Validation

We perform preliminary numerical experiments on two image classification datasets: *MNIST* and *Quick-Draw*.

[4] This assumption is similar to the spatially correlated assumption proposed for the extension of KFAC to CNNs in [Grosse 2016]

We compare the evolution of the test accuracy when using our method- that we denote **BalNet**- against the following baselines:

• DFAL: we select on the whole unlabeled training set the first n_{query} samples with the lowest adversarial perturbation.

• BALD: we select on a random subset of the unlabeled training set the first n_{query} samples which are expected to maximize the mutual information with the model parameters. In that order, we sample 10 networks from the approximate posterior of the weights by also applying dropout at test time.

• CEAL: we select on the whole unlabeled training set the first n_{query} samples with the highest entropy on their network's prediction. We also label any unlabeled samples whose entropy is lower than a given threshold (which is set according to the authors' guidelines: 0.05 for *MNIST*, 0.19 for *Shoe-Bag* and 0.08 for *Quick-Draw*). Their labels are not queried but estimated from the network's predictions.

• CORE-SET: we select on a random subset of the unlabeled training set the n_{query} samples which cover at best the training set (labeled and unlabeled data) based on the euclidean distance on the output of the last fully connected layer. To approximate the cover set problem, we follow the instructions prescribed in [Ozan Sener 2018]: we initialize the selection with the greedy algorithm, and iterate with their mixed integer programming subroutine. We also handle the robustness as prescribed by the authors. We use *or-tools* [5] to reproduce the MIP subroutine.

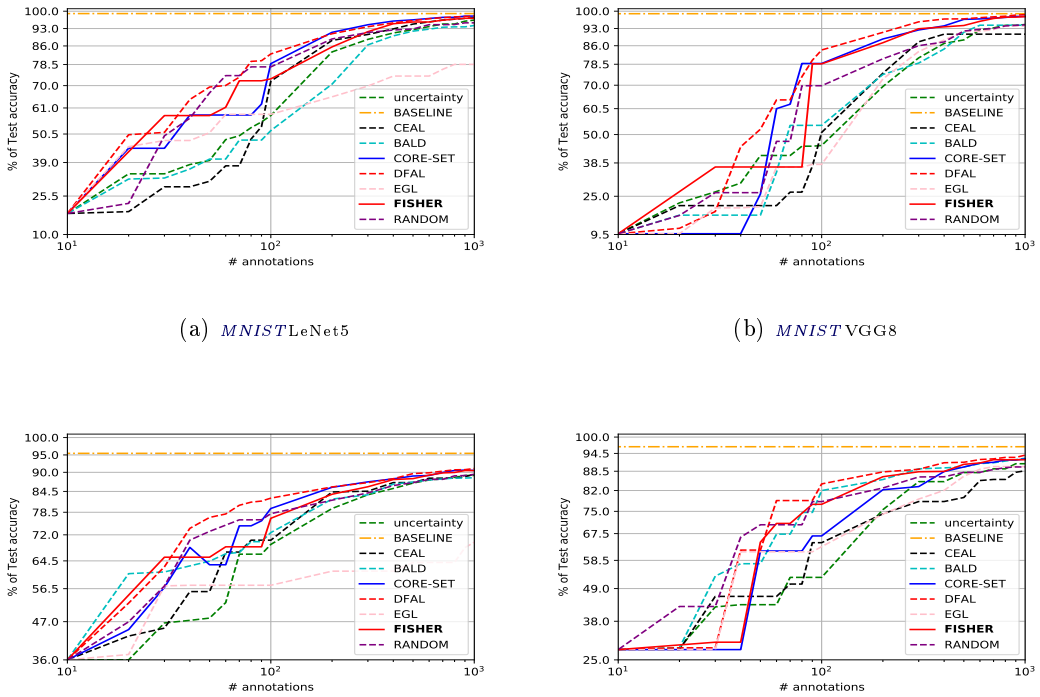
• EGL: we select from a random subset of the unlabeled training set the first n_{query} samples whose gradients achieves the highest euclidean norm.

• uncertainty: we select from the whole unlabeled training set the first n_{query} samples with the highest entropy on their network's prediction.

• RANDOM: we select randomly from the whole unlabeled training set n_{query} samples.

We average our results over five trials and we plot in figures 5.4, the test accuracy achieved by each active learning methods for fixed size training set: with 100, 200, ... to 1000 labeled samples. We denote as *BASELINE*, the test accuracy when training the network on the full labeled training set. Preliminary experiments demonstrate that **BalNet** is competitive with state-of-the-art active learning heuristics for image classification, CORE-SET and **DFAL**.

[5] https://developers.google.com/optimization



(a) *MNIST* LeNet5

(b) *MNIST* VGG8

(c) *Quick-Draw* LeNet5

(d) *Quick-Draw* VGG8

Figure 5.4: Evolution of the test accuracy achieved by 7 active learning techniques on *MNIST* and *Quick-Draw* given the number of annotations. We denote our method by **FISHER**

5.6 Future work

Although **BalNet** appears promising, in the first iterations, we observe in the first iterations that it is underperforming other active learning heuristics. This pattern may be explained due to the usage of KFAC approximation: even as a regularizer, KFAC should be used on large size minibatches. So we assume that the approximation of the Fisher matrices do not hold on small labeled training set. This assumption will be clarified with experiments on large scale dataset in future works. Moreover, using KFAC as a regularizer should be relevant in our active learning method. Note that we use greedy search to select our **BalNet** queries which is sub-optimal for our criterion. Future works should improve the selection of the queries according to the properties of minimizing a submodular function. More comparative studies are required to assert of the relevancy of our fisher criterion:

1. **diversity**: what is the impact of our Wasserstein covering compared to random selection or any other diversity based criterion proposed in active learning

[Wei 2015]

2. **Large Scale Dataset**: how will **BalNet** behave on large scale datasets?

3. **Classification tasks and network architectures**: DFAL and CORE-SET are designed for CNNs and have been asserted on image classification tasks. The performance of **BalNet** should also be tested on other tasks such as Visual Question Answering and architectures such as LSTMs.

5.7 Conclusion

In a nutshell, we proposed a scalable batch active learning framework for deep networks relying on the variational free energy. We deduced a formulation of the posterior and prior distributions of the CNN' s weights using the Laplace hypothesis. Those assumptions, combined with a Kronecker based formulation of the Fisher information matrix for neural networks, lead us to a gradient free active learning heuristic. Eventually, we develop a highly efficient query search for batch active learning thanks to the induced submodularity properties of our criterion.

Our criterion is the first of the kind to scale batch active learning to deep networks, especially CNNs. On different databases, it achieves better test accuracy than random sampling and is scalable with increasing size of queries. It achieves near-optimal test error using a limited percentage of the annotated training set on larger and more reduced dataset. Our works demonstrated the validity of batch mode active learning for deep networks and the promise of the KFAC formulation for deep Fisher matrices for the active learning community. Such a solution is also interesting as a new technique for curriculum learning approach.

Chapter 6

Conclusion

We investigate the impact of **AL** on deep neural architectures. Neural networks need large, even huge, datasets for their training. Otherwise, if the training data set is too small, overfitting may very well occur, in particular when it comes to deep networks with many layers. However, unexpectedly, **AL** is efficient, even on a large network such as VGG8. The reasons underlying the success of **AL** on deep networks are in some part, explained by some understanding of the generalization bounds for deep networks (see Section 2.4).

While dropout has been promoted for active learning, the impact of other well-known regularizers such as batch normalization [Ioffe 2015], have not yet been investigated. Indeed, batch normalization is more indicated on large size minibatch, which is counter-intuitive with a reduced training set. Similarly, while we have proposed to use the KFAC approximation of Fisher matrix in our **AL** criterion, the natural gradient based regularizer that KFAC was designed for, has not been studied in the context of active learning. Indeed, while KFAC is efficient, it is advised for a sizeable minibatch.

Other related research areas, such as domain adaptation and semi-supervised learning, are well suited to act as regularizers in an active learning context. Up to some point, we have started to investigate these factors in the next chapter, using Wasserstein distance. Moreover, while we have focused our work on scaling **AL** for deep networks, other architectures may better handle small datasets, such as Gaussian processes.

Another step towards promoting active learning heuristics is to propose batch query heuristics and think of how it should be incorporated into the algorithm itself: Should it be done in a post-processing step, as we recommended in Section 5.2, or should it be captured by the **AL** criterion itself, like CORE-SET? Although the latest solution sounds more indicated, it also explodes the computational cost of the **AL** heuristic (cf. Table 4.3). Moreover, our latest method **DFAL** suggests that a simple top score selection is competitive with a batch query selection. Nevertheless, our fisher method leads to think that a post-processing selection, while being suboptimal works well in practice.

The definition of active learning protocols and benchmarks is crucial to improve our methods. Nowadays, the field cruelly lacks rules of thumbs which leads to unobtainable results in real life settings: balancing the labels in the initial training set, using an enormous amount of labeled validation set (which is furthermore not counted in the number of annotations), tuning the hyperparameters on the full labeled training set.

We are aware of meta-learning algorithms, based on reinforcement learning, that learns an active learning policy [Contardo 2017]. Up to our knowledge, neither **AL** heuristics nor meta-learning algorithms have yet been compared one to another. However, it sounds relevant as future works to analyze both the performance and computational time of such algorithms regarding state-of-the-art active learning strategies. As meta-algorithms are learning transferable policies, they can be useful to develop hyperparameter search combined with query selection.

Part II

Learning Wasserstein Core-Sets

Chapter 7

Introduction

Contents

7.1 Motivations **69**

7.2 Definitions **70**

7.3 Litterature **71**

7.3.1 Fast approximation of the exact Wasserstein distance 71

7.3.2 Metric embedding 73

7.3.3 Domain adaptation 73

7.3.4 Wasserstein Core-Sets for Lipschitz Costs 74

7.3.5 Herding 75

7.1 Motivations

With the overwhelming success of deep networks and their requrements for large datasets, there is a growing need to assess the consistency of our training dataset. In that aim, one possible solution is to rely on core-sets. A Core-set of a dataset is a subset, typically denoted as medoids, that is representative of the whole set of data given an informative criterion. It takes root in computational geometry [Agarwal 2005] and have been widespread to the machine learning community first via importance sampling [Langberg 2010]. Core-sets provide a first glimpse of the dataset, which can be used in various forms: either to visualize, compact the information or identify bias.

One of the main challenge is to decide which informations should be captured to build such core-sets. We focus on Wasserstein distance, as an informative criterion to build a core-set.

The Wasserstein distance is a powerful tool based on the theory of optimal transport to compare data distributions with wide applications in image processing, computer vision and machine learning [Kolouri 2017]. In a context of machine learning, it has recently found numerous applications, *e.g.* domain adaptation [Courty 2017a], word embedding [Huang 2016a] or generative models [Arjovsky 2017a]. Its power lies in two properties: *i)* it allows to operate on empirical data distributions in a non-parametric way ;*ii)* the geometry of the underlying space can be leveraged to compare the distributions. The space of probability measures equipped with the

Wasserstein distance can be used to construct objects of interest such as barycenters [Agueh 2011] or geodesics [Seguy 2015] that can be used in data analysis and mining tasks.

We dedicate this section to provide fast technique of computing pairwise Wasserstein distance in parallel. Specifically, we propose in Section 8, a neural network whose output embeds the Wasserstein distance, for low dimensional manifolds. Thanks to this scheme, we can naturally approximate data mining tasks in the embedding space, but we can also dedicate this embedding to clustering and core-sets.

Nextly, in Section 9, we highlight some submodularities properties of Wasserstein distance for empirical distributions, and how we can greedily select Wasserstein prototypes to build a core-set.

7.2 Definitions

More formally, let X be a metric space endowed with a metric d_X. Let $p \in (0, \infty)$ and $\mathcal{P}_p(X)$ the space of all Borel probability measures μ on X with finite moments of order p, *i.e.* $\int_X d_X(x, x_0)^p d\mu(x) < \infty$ for all x_0 in X. The p-Wasserstein distance between μ and ν is defined as:

$$W_p(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \iint_{X \times X} d(x, y)^p d\pi(x, y) \right)^{\frac{1}{p}}. \tag{7.1}$$

Here, $\Pi(\mu, \nu)$ is the set of probabilistic couplings π on (μ, ν). As such, for every Borel subsets $A \subseteq X$, we have that $\mu(A) = \pi(X \times A)$ and $\nu(A) = \pi(A \times X)$. It is well known that W_p defines a metric over $\mathcal{P}_p(X)$ as long as $p \geq 1$ (e.g. [Villani 2009], Definition 6.2).

When $p = 1$, W_1 is also known as Earth Mover Distance (EMD) or Monge-Kantorovich distance. The geometry of $(\mathcal{P}_p(X), W_1(X))$ has been thoroughly studied, and there exists several works on computing EMD for point sets in $\mathbb{R}^k$ (e.g. [Shirdhonkar 2008]). However, in a number of applications the use of W_2 (a.k.a *root mean square bipartite matching distance*) is a more natural distance arising in computer vision [Bonneel 2015], computer graphics [Bonneel 2011, de Goes 2012, Solomon 2015a, Bonneel 2016] or machine learning [Cuturi 2014, Courty 2017a]. See [de Goes 2012] for a discussion on the quality comparison between W_1 and W_2.

In the discrete version, where both μ and ν are uniform distributions respectively supported by n and m points, the p-Wasserstein distance can be expressed as a linear programming optimization problem:

$$W_p(\mu, \nu) = \min_{\Gamma \in \Sigma(\mu,\nu)} \left\langle C, \Gamma \right\rangle^p \tag{7.2}$$

Where matrix C is the distance matrix between every pairwise samples of μ and ν, the notation $< \cdot, \cdot >$ denotes the Frobenius dot product and $\Sigma(\mu, \nu) = \{\Gamma \in \mathbb{R}_+^{n,m},\ \Gamma 1_n = \mu,\ \Gamma^T 1_m = \nu\}$ is the set of valid transportation matrices between

both distributions, where 1_n represents the n-dimensional vector of ones. Because the number of variables scales quadratically with the number of samples in the distributions, computing the exact Wasserstein holds a cubical complexity.

7.3 Litterature

7.3.1 Fast approximation of the exact Wasserstein distance

The cost for computing the exact Wasserstein distance for empirical distributions can limit its usage in various applications. Thus, a lot of efforts has been put on alleviating the computation complexity, by either proposing regularized versions of Wasserstein, such as Sinkhorn [Cuturi 2013b], or deducing iterative schemes that will converge to the exact Wasserstein, such as IPOT [Xie 2018].

The Sinkhorn distance allows the fast computation of an entropically regularized Wasserstein distance between two probability distributions supported on a finite metric space of (possibly) high-dimension. The entropic regularization results in an optimization problem that can be solved efficiently by Iterative Bregman projections [Benamou 2015]. It is known to achieve near quadratic complexity. However, the Sinkhorn distance remains an approximation of the exact Wasserstein distance. While some machine learning problems benefit from the Sinkhorn approximation, others do not. In particular, the computation of Wasserstein barycenters requires a tight approximation of the exact Wasserstein, using a small entropic regularization. Indeed, the regularization parameter in the Sinkhorn distance, is a major hyperparameter, which owns huge numerical implications. If the regularization parameter is very small, we can observe numerical instability [Xie 2018]. Nevertheless, the linear convergence rate of the Sinkhorn algorithm is determined by the contraction ratio which tends to 1 as the regularization parameter decreases. Consequently, we observe drastically increase number of iterations for the Sinkhorn method when using small regularization value.

Recently, a new approximation scheme for Wasserstein distance has been proposed, called IPOT. IPOT relies on proximal point methods. Eventually their optimization can be solved by Sinkhorn iteration by updating the distance matrix at each step, instead of keeping it fixed as in the original Sinkhorn algorithm. We detail both pseudo code for Sinkhorn and IPOT in Alg. 1 and 2. Regarding IPOT, unlike Sinkhorn, empirical analysis confirmed that it converges to the exact Wasserstein distance, independently to the choice of the regularization parameter, with linear convergence.

Another line of work [Wang 2013, Kolouri 2016b] also considers the Riemannian structure of the Wasserstein space to provide meaningful linearization by projecting onto the tangent space. By doing so, they notably allows for faster computation of pairwise Wasserstein distances (only N transport computations instead of $N(N-1)/2$ with N the number of samples in the dataset) and allow for statistical analysis of the embedded data. They proceed by specifying a template element and compute, from particle approximations of the data, linear transport plans with this

Algorithm 1 Pseudo code of $SINKHORN(\mu, \nu, C, \beta)$ [Cuturi 2013b]

Require: empirical probability distribution $\{\mu, \nu\}$ respectively on support points $\{x_i\}_{i=1}^n$, $\{y_j\}_{j=1}^m$
Require: distance matrix $C = ||x_i - y_j||$
Require: regularization constant β
  $u^{(0)} = \mathbf{1}_n$
  $G \leftarrow e^{-\frac{C}{\beta}}$
  for i=1,2,3,... **do**
    $v^{(i)} = b \oslash K^T u^{(i-1)}$
    $u^{(i)} = a \oslash K v^{(i)}$
  end for
  $\mathcal{T} \leftarrow diag(u^{(i)}) G diag(b^{(i)})$

Algorithm 2 Pseudo code of $IPOT(\mu, \nu, C, \beta)$ [Xie 2018]

Require: empirical probability distribution $\{\mu, \nu\}$ respectively on support points $\{x_i\}_{i=1}^n$, $\{y_j\}_{j=1}^m$
Require: distance matrix $C = ||x_i - y_j||$
Require: regularization constant β
  $b \leftarrow \frac{1}{m}\mathbf{1}_m$
  $G \leftarrow e^{-\frac{C}{\beta}}$
  $\mathcal{T}^{(1)} \leftarrow \mathbf{1}_{n,m}$
  for t=1,2,3,... **do**
    $Q \leftarrow G \odot \mathcal{T}^{(t)}$
    for l=1,2,3,... **do**
      $a \leftarrow \frac{\mu}{Qb}$
      $b \leftarrow \frac{\nu}{Q^T a}$
    end for
    $\mathcal{T}^{(t+1)} \leftarrow diag(a) Q diag(b)$
  end for

template element, that allow to derive an embedding used for analysis. Seguy and Cuturi [Seguy 2015] also proposed a similar pipeline, based on velocity field, but without relying on an implicit embedding. It is to be noted that for data in 2D, such as images, the use of cumulative Radon transform also allows for an embedding which can be used for interpolation or analysis [Bonneel 2015, Kolouri 2016a], by exploiting the exact solution of the optimal transport in 1D through cumulative distribution functions.

7.3.2 Metric embedding

In Section 8, we proposed to alleviate the cost of computing the exact Wasserstein distance using Metric embedding approach. Our method is fairly new as we do not rely on mathematical grounded approximations.

The question of metric embedding usually arises in the context of approximation algorithms. Generally speaking, one seeks a new representation (embedding) of data at hand in a new space where the distances from the original space are preserved. This new representation should, as a positive side effect, offers computational ease for time-consuming task (e.g. searching for a nearest neighbor), or interpretation facilities (e.g. visualization of high-dimensional datasets). More formally, given two metrics spaces (X, d_X) and (Y, d_y) and $D \in [1, \infty)$, a mapping $\phi : X \to Y$ is an embedding with distortion at most D if there exists a coefficient $\alpha \in (0, \infty)$ such that $\alpha d_X(x, y) \leq d_Y(\phi(x), \phi(y)) \leq D\alpha d_X(x, y)$. Here, the α parameter is to be understood as a global scaling coefficient. The **distortion** of the mapping is the infimum over all possible D such that the previous relation holds. Obviously, the lower the D, the better the quality of the embedding is. It should be noted that the existence of exact (**isometric**) embedding ($D = 1$) is not always guaranteed but sometimes possible. Finally, the embeddability of a metric space into another is possible if there exists a mapping with constant distortion. A good introduction on metric embedding can be found in [Matoušek 2013].

7.3.3 Domain adaptation

Recent works have underlined the usage of Wasserstein into domain adaptation [Courty 2017c, Shen 2018]. In particular, [Lee 2017] provides generalization guarantees for domain adaptation based on the notion of Wasserstein balls, which owns similarity with part of our work, denoted as Wasserstein prototypes (*see Section 9*). They aim to minimize the worst-case risk over a larger ambiguity set containing the original empirical distribution of the training data.

Given an n-tuple $\{x_1, \dots x_n\}$ of iid training examples sampled from the unknown ground-truth distribution $\mathbb{P}$, the objective is to find a hypothesis whose risk is close to the minimum risk with high probability. The risk $R(\mathbb{P}, f)$ is the expectation of f over instances sampled from $\mathbb{P}$. However, because the ground-truth distribution is unknown, the risk is not tractable. One solution to optimize the risk is to minimize an *approximate risk*: the maximal risk given any distribution $\mathbb{Q}$ approximately close

to the true but unknown ground-truth distribution $\mathbb{P}$. That distribution lies in an area denoted as the ambiguity set. They define the ambiguity set $\mathcal{A}(\mathbb{P})$, as the p-Wasserstein ball of radius ε centered around $\mathbb{P}$. Where $\mathbb{Q}$ is a Borel distribution defined on a Polish space. Eventually, we can define the *approximate* risk as the following:

$$\begin{aligned} \mathcal{A}(\mathbb{P}) &= \{\mathbb{Q} \mid W_p(\mathbb{P}, \mathbb{Q}) \leq \varepsilon\} \\ R_{\varepsilon,p}(\mathbb{P}, f) &= \sup_{\mathbb{Q} \in \mathcal{A}(\mathbb{P})} R(\mathbb{Q}, f) \\ \textit{with } R(\mathbb{Q}, f) &= \mathbb{E}_{z \sim \mathbb{Q}}\Big[f(z)\Big] \end{aligned} \tag{7.3}$$

Assuming that the difference between the labelled training set $\mathbb{Q}$ and the ground-truth distribution $\mathbb{P}$ comes only from transformations of the input space (independently from the labels associated to the examples), then we can upper bound the approximate risk $R_{\varepsilon,p}(\mathbb{P}, f)$ given the risk on the labelled training set $R(\mathbb{Q}, f)$:

Theorem 3.1: Minimax Statistical Learning

Suppose that the hypothesis f is L-Lipschitz, if we denote by $\mathbb{P}$ the ground-truth distribution and $\mathbb{Q}$ the discrete distribution induced by sampling along $\mathbb{P}$. We can upper bound the *approximate* risk on $\mathbb{P}$ given the risk on the labelled set and their Wasserstein distance.

$$R_{\varepsilon,p}(\mathbb{P}, f) \leq R(\mathbb{Q}, f) + 2L * W(\mathbb{P}, \mathbb{Q}) \tag{7.4}$$

7.3.4 Wasserstein Core-Sets for Lipschitz Costs

Another line of work, in [Claici 2018] bridges the gap between core-sets with Lipschitz cost and optimal transport. They build an upper bound on what they denote as *measure core-set* with Wasserstein distance:

Definition 7.3.1: Measure Core-Set

Given $\varepsilon \in \mathbb{R}^+$, we call ν a *measure core-set* for μ on the support $\mathcal{X}$, if ν is absolutely continuous with respect to μ and $\forall\ f \in \mathcal{F}$:

$$|cost(\mathcal{X}, f, \nu) - cost(\mathcal{X}, f, \mu)| \leq \varepsilon cost(\mathcal{X}, f, \nu) \tag{7.5}$$

Note that ν always exists since $\nu \equiv \mu$ satisfies the inequality.

For *measure core-set* to be tractable, they restrict their case of study when ν is a uniform empirical distribution over a fixed number of points, whose support is undefined: ν is of the form $\frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$. Finally, they propose to build ν so that ν minimizes what they denote as *Wasserstein Core-Set*, which upper bounds the *measure core-set*:

Definition 7.3.2: Wasserstein Core-Set

When $\mathcal{F} \subset Lip_1(\mathcal{X})$, a sufficient condition for ν to be an *epsilon* core-set given μ and $\mathcal{F}$ is $W_1(\mu, \nu) \leq \varepsilon$

Because W_2 is a way more popular distance to compute baycenters than W_1, and justified by the inequality $W_1(\mu, \nu) \leq W_2(\mu, \nu)$, [Claici 2018] prescribed Eq. 7.6 to construct the n-point *measure core-set*:

$$\underset{x_1, x_2, \cdots, x_n}{\arg\min} \; W_2(\mu, \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}) \tag{7.6}$$

Determining the positions of the points makes the problem highly non convex. They provide a simple optimization strategy based on an iterative optimizations between the different parameters involved.

Wasserstein Core-Sets hold several similarity with one of our recent work that we denote Wasserstein prototypes. We highlight in Section 9 the pros and cons of both of those methods when it comes to our problematic.

7.3.5 Herding

MMD is a measure of the difference between two distributions μ and ν given by the supremum over a function space $\mathcal{F}$. The MMD between μ and ν reads:

$$MMD(\mu, \nu; \mathcal{F}) = sup_{f \in \mathcal{F}} \big(\mathbb{E}_{x \sim \mu}[f(x)] - \mathbb{E}_{x \sim \nu}[f(x)] \big) \tag{7.7}$$

Similarly as our work in Section 9, we can define prototypes according to MMD. We denote by prototypes example-based explanations according to an informative criterion, here MMD. Among the possible usage, prototypes are widely used in the effort to improve the interpretability of highly complex distributions. Formally, we can express MMD prototypes as the solution of Eq. 7.8. Considering the nature of the kernel matrix, MMD prototypes can be fairly well approximated using greedy search, due to inherent submodularity properties, as described in Theorem 3. On the contrary, looking for MMD prototypes comes to maximising a *weakly submodular* function [Huszár 2012]. Although using greedy search to maximize a weakly submodular function works fairly well in practice, there exist no tight upper bound on the quality of the prototypes.

Definition 7.3.3: MMD Prototypes

Formally given an empirical distribution $\mathcal{P}$, a fixed size of data to label, K, we denote **MMD prototypes** a subset $\mathcal{S}_K^* \subset \mathcal{P}$ that minimizes Eq. 7.8.

$$\min_{\mathcal{S}_K^* \subset \mathcal{P},\, |\mathcal{S}_K^*| = K} MMD(\mathcal{S}_K^*, \mathcal{P})^2 \tag{7.8}$$

Theorem 3.2: Monotone Submodularity for MMD prototypes [Kim 2016]

Let the kernel matrix $K \in \mathbb{R}^{n \times n}$ be element-wise non-negative, with equal diagonal terms $k_{i,i} = k^* > 0 \ \forall \ i \in [n]$ ad be diagonally dominant. If the off diagonal terms $k_{i,j}$ satisfies:

$$0 \leq k_{i,j} \leq \frac{k^*}{n^3 + 2n^2 + 2n - 3}$$

Then selecting **MMD prototypes** given Eq. 7.8 consists in maximising a submodular function which can be approximated with greedy search.

Chapter 8

Learning Wasserstein embeddings

Contents

8.1 Introduction **77**

8.2 Wasserstein learning and reconstruction with siamese networks **78**

8.3 Empirical Validation **80**

8.3.1 Wasserstein data mining in the embedded space 82

8.4 Future work: Wasserstein for Text Mining **87**

8.4.1 Information Retrieval: Fast computing of WMD at large scale 88

8.4.2 Text Generation given Wasserstein Distance 90

8.5 Conclusion **92**

8.1 Introduction

Summary

• We present a new way to approximate pairwise Wasserstein distances for examples sampled from a specific distribution such as an image collection. To do so, we train a siamese neural network, denoted as **Deep Wasserstein Embedding** (**DWE**), on which the euclidian loss on the outputs matches the exact Wasserstein distance on the input instances.

• We empirically demonstrate how the learned embedding may be used for computing efficiently optimization problems in the Wasserstein space.

✓ *Every dataset and parameters used to conduct our experiments are available in the* ***dataset*** *section A.1 and the* ***hyperparameter*** *section A.2.4*

mducoffe/LearningWassersteinEmbeddings

The Wasserstein distance received a lot of attention recently in the community of machine learning, especially for its principled way of comparing distributions. It has found numerous applications in several hard problems, such as domain adaptation, dimensionality reduction or generative models. Yet, the deployment

of Wasserstein distances in a wide class of applications is somehow limited, especially because of an heavy computational burden. In the discrete version of the above optimisation problem, the number of variables scale quadratically with the number of samples in the distributions, and solving the associated linear program with network flow algorithms is known to have a cubical complexity. While recent strategies implying slicing technique [Bonneel 2015, Kolouri 2016a], entropic regularization [Cuturi 2013a, Benamou 2015, Solomon 2015b] or involving stochastic optimization [Genevay 2016], have emerged, the cost of computing pairwise Wasserstein distances between a large number of distributions (like an image collection) is prohibitive. This is all the more true if one considers the problem of computing barycenters [Cuturi 2014, Benamou 2015] or population means. A recent attempt by Staib and colleagues [Staib 2017] use distributed computing for solving this problem in a scalable way.

Our goal is to alleviate this problem by providing an approximation mechanism that allows to break its inherent complexity. It relies on the search of an embedding where the Euclidean distance mimics the Wasserstein distance. We show that such an embedding can be found with a siamese architecture associated with a decoder network that allows to move from the embedding space back to the original input space. Once this embedding has been found, computing optimization problems in the Wasserstein space (*e.g.* barycenters, principal directions or even archetypes) can be conducted extremely fast. Numerical experiments supporting this idea are conducted on image datasets, and show the wide potential benefits of our method.

8.2 Wasserstein learning and reconstruction with siamese networks

We propose in this work to learn an Euclidean embedding of distributions where the Euclidean norm approximates the Wasserstein distances. Finding such an embedding enables the use of standard Euclidean methods in the embedded space and significant speed up in pairwise Wasserstein distance computation, or construction of objects of interests such as barycenters. The embedding is expressed as a deep neural network, and is learnt with a strategy similar to those of Siamese networks [Chopra 2005]. We also show that simultaneously learning the inverse of the embedding function is possible and allows a reconstruction of a probability distribution from the embedding. Our work is the first to propose to learn a generic embedding rather than constructing it from explicit approximations/transformations of the data and analytical operators such as Riemannian Logarithm maps. As such, our formulation is generic and adapts to any type of data. Finally, since the mapping to the embedded space is constructed explicitly, handling unseen data does not require to compute new optimal transport plans or optimization, yielding extremely fast computation performances, with similar approximation performances.

We discuss here how our method, coined **DWE** for Deep Wasserstein Embedding works. **DWE** learns in a supervised way a new representation of the data.

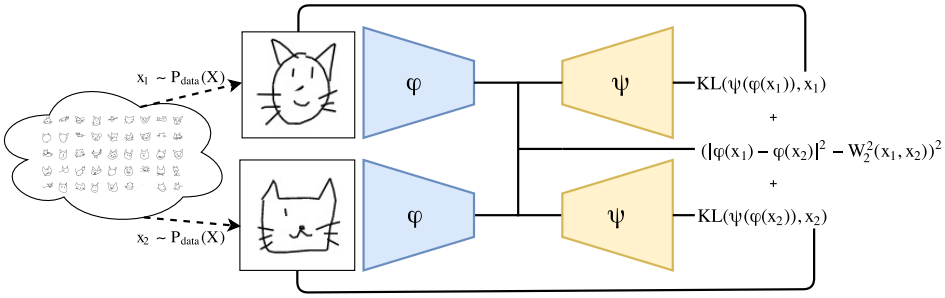


Figure 8.1: Architecture of the Wasserstein Deep Learning: two samples are drawn from the data distribution and set as input of the same network (ϕ) that computes the embedding. The embedding is learnt such that the squared Euclidean distance in the embedding mimics the Wasserstein distance. The embedded representation of the data is then decoded with a different network (ψ), trained with a Kullback-Leibler divergence loss.

To this end we need a pre-computed dataset that consists of pairs of histograms $\{x_i^1, x_i^2\}_{i\in 1,\dots,n}$ of dimensionality d and their corresponding W_2^2 Wasserstein distance $\{y_i = W_2^2(x_i^1, x_i^2)\}_{i\in 1,\dots,n}$. One immediate way to solve the problem would be to concatenate the samples x^1 and x^2 and learn a deep network that predicts y. This would work in theory but it would prevent us from interpreting the Wasserstein space and it is not by default symmetric which is a key property of the Wasserstein distance.

Another way to encode this symmetry and to have a meaningful embedding that can be used more broadly is to use a Siamese neural network [Bromley 1994]. Originally designed for metric learning purpose and similarity learning (based on labels), this type of architecture is usually defined by replicating a network which takes as input two samples from the same learning set, and learns a mapping to new space with a contrastive loss. It has mainly been used in computer vision, with successful applications to face recognition [Chopra 2005] or one-shot learning for example [Koch 2015]. Though its capacity to learn meaningful embeddings has been highlighted in [Weston 2012], it has never been used, to the best of our knowledge, for mimicking a specific distance that exhibits computation challenges. This is precisely our objective here.

We propose to learn an embedding network ϕ that takes as input a histogram and project it in a given Euclidean space of $\mathbb{R}^p$. In practice, this embedding should mirror the geometrical property of the Wasserstein space. We also propose to regularize the computation of this embedding by adding a reconstruction loss based on a decoding network ψ. This has two important impacts. First we observed empirically that it eases the learning of the embedding and improves the generalization performance of the network (*as illustrated in Figure 8.3*) by forcing the embedded representation to catch sufficient information of the input data and thus allowing a good reconstruction. This type of autoencoder regularization loss has been discussed in [Yu 2013] in the different context of embedding learning. Second, the decoder net-

work allows the interpretation of the results, which is of prime importance in several data-mining tasks (*discussed in the next subsection 8.3.1*).

An overall picture depicting the whole process is given in Figure 8.1. The global objective function reads

$$\min_{\phi,\psi} \quad \sum_i \left\| \|\phi(x_i^1) - \phi(x_i^2)\|^2 - y_i \right\|^2 + \lambda \sum_i \Big(\mathrm{KL}(\psi(\phi(x_i^1)), x_i^1) + \mathrm{KL}(\psi(\phi(x_i^2)), x_i^2) \Big) \tag{8.1}$$

where $\lambda > 0$ weights the two data fitting terms and KL(,) is the Kullbach-Leibler divergence. This choice is motivated by the fact that the Wasserstein metric operates on probability distributions.

Every dataset and parameters used to conduct our experiments are available in the **dataset** section A.1 and the **hyperparameter** section A.2.4

Next, we evaluate the performances of our method on grayscale images normalized as histograms. Images are offering a nice testbed because of their dimensionality and because large datasets are frequently available in computer vision. We also operate our method for text mining in Section 8.4.

8.3 Empirical Validation

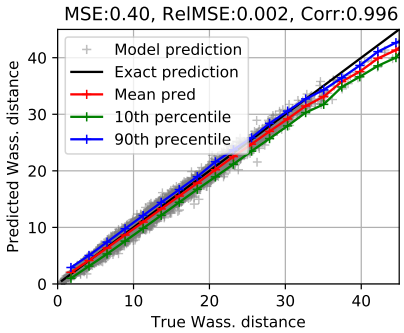
The framework of our approach as shown in Fig 8.1 consists of an encoder ϕ and a decoder ψ organized as a cascade. The encoder produces the representation of input images $h = \phi(x)$. The architecture used for the embedding and the reconstruction consists in convolutional layers with ReLU activations, plus dense layers. In this section, we only consider grayscale images, that are normalized to represent probability distributions. Hence each image is depicted as an histogram. In order to normalize the decoder reconstruction we use a softmax activation for the last layer.

All the datasets considered are handwritten data and hence holds an inherent sparsity. In our case, we cannot promote the output sparsity through a convex L1 regularization because the softmax outputs only positive values and forces the sum of the output to be 1. Instead, we apply a ℓ_p^p pseudo -norm regularization with $p = 1/2$ on the reconstructed image, which promotes sparse output and allows a sharper reconstruction of the images [Gasso 2009].

8.3.0.1 Numerical precision and computational performance

The true and predicted values for the Wasserstein distances are given in Fig. 8.2. We can see that we reach a good precision with a test MSE of 0.4 and a relative MSE of 2e-3. The correlation is of 0.996 and the quantiles show that we have a very small uncertainty with only a slight bias for large values where only a small number of samples is available. This results show that a good approximation of the W_2^2 can be performed by our approach (≈1e-3 relative error).

Now we investigate the ability of our approach to compute W_2^2 efficiently. To this end we compute the average speed of Wasserstein distance computation on test dataset to estimate the number of W_2^2 computations per second in the Table of Fig. 8.2. Note that there are 2 ways to compute the W_2^2 with our approach denoted



Method	W_2^2/sec
LP network flow (1 CPU)	192
DWE Indep. (1 CPU)	3 633
DWE Pairwise (1 CPU)	213 384
DWE Indep. (GPU)	233 981
DWE Pairwise (GPU)	10 477 901

Figure 8.2: Prediction performance on the *MNIST* dataset. (Figure) The test performances are as follows: MSE=0.40, Relative MSE=0.002 and Correlation=0.996. (Table) Computational performance of W_2^2 and **DWE** given as average number of W_2^2 computation per seconds for different configurations.

as Indep. and Pairwise. This comes from the fact that our W_2^2 computation is basically a squared Euclidean norm in the embedding space. The first computation measures the time to compute the W_2^2 between independent samples by projecting both in the embedding and computing their distance. The second computation aims at computing all the pairwise W_2^2 between two sets of samples and this time one only needs to project the samples once and compute all the pairwise distances, making it more efficient. Note that the second approach would be the one used in a retrieval problem where one would just embed the query and then compute the distance to all, or a selection of, the dataset to find a Wasserstein nearest neighbor for instance. The speed up achieved by our method is very impressive even on CPU with speed up of x18 and x1000 respectively for Indep. and Pairwise. But the GPU allows an even larger speed up of respectively x1000 and x500 000 with respect to a state-of-the-art C compiled Network Flow LP solver of the POT Toolbox [Flamary 2017, Bonneel 2011]. Of course this speed-up comes at the price of a time-consuming learning phase, which makes our method better suited for mining large scale datasets and online applications.

Lastly, we discuss the role of the decoder, not only as a matter of interpreting the results, but rather as a regulizer. We train our **DWE** on *MNIST* with and without the decoder and compares the learning curves of the MSE on the validation set. In Figure 8.3, **DWE** achieves a lower MSE with the decoder, which enforces the use of a decoder into our framework.

8.3.0.2 Numerical precision and cross dataset comparison

The numerical performances of the learned models on each of the *Quick-Draw* classes is reported in the diagonal of Table 8.1. Those classes are much more difficult than *MNIST* because they have not been curated and contain a very large variance due to numerous unfinished doodles. An interesting comparison is the cross comparison between datasets where we use the embedding learned on one dataset to compute the

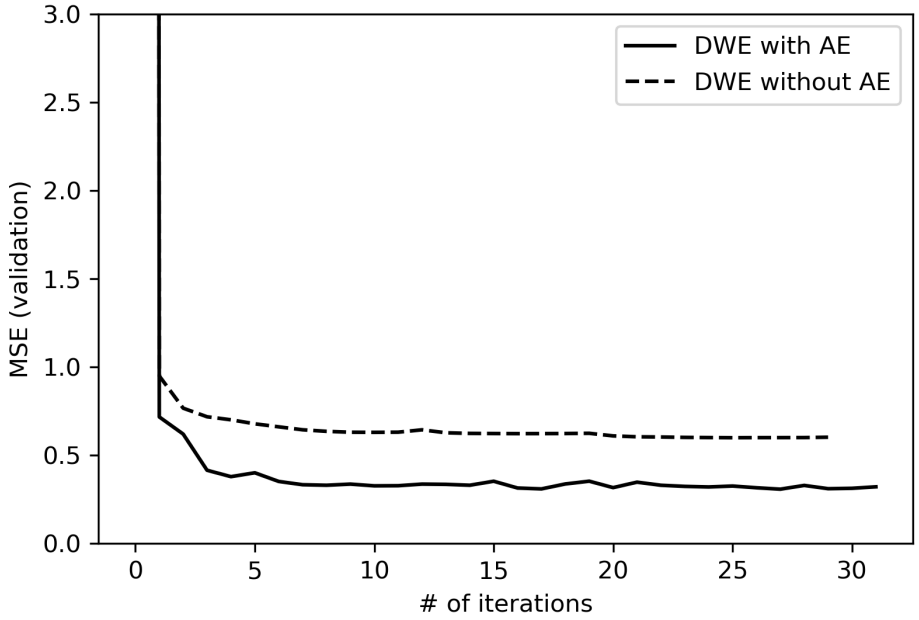


Figure 8.3: MSE of the validation test given the number of epochs (**DWE**).

Network Data	CAT	CRAB	FACE	MNIST
CAT	**1.195**	*1.654*	2.069	12.131
CRAB	*2.621*	**0.815**	3.158	10.881
FACE	*5.025*	5.445	**1.254**	50.526
MNIST	9.118	6.698	*4.68*	**0.412**

Table 8.1: Cross performance between the **DWE** embedding learned on each datasets. On each row, we observe the MSE of a given dataset obtained on the deep network learned on the four datasets (Cat, Crab, Faces and MNIST).

W_2^2 on another. The cross performances is given in Table 8.1 and shows that while there is definitively a loss in accuracy of the prediction, this loss is limited between the classes from the *Quick-Draw* dataset that have all a large diversity. Performance loss across *Quick-Draw* and *MNIST* dataset is larger because the latter is highly structured and one needs to have a representative dataset to generalize well which is not the case with *MNIST*.

8.3.1 Wasserstein data mining in the embedded space

Once the functions ϕ and ψ have been learned, several data mining tasks can be operated in the Wasserstein space. We discuss here the potential applications of our computational scheme and its wide range of applications on problems where the Wasserstein distance plays an important role. Though our method is not an exact

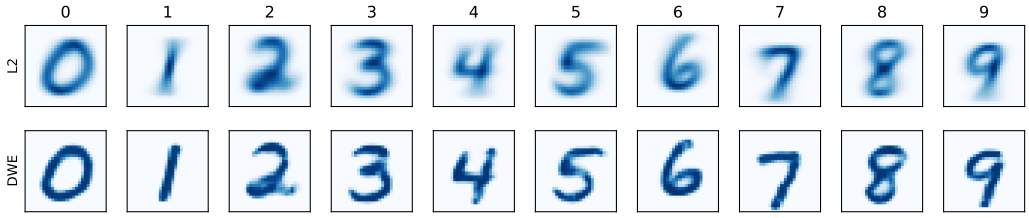


Figure 8.4: Barycenter estimation on each class of the *MNIST* dataset for squared Euclidean distance (L2) and Wasserstein Deep Learning (**DWE**).

Wasserstein estimator, we empirically show in the numerical experiments that it performs very well and competes favorably with other classical computation strategies.

8.3.1.1 Wasserstein barycenters [Agueh 2011, Cuturi 2014, Bonneel 2016].

Barycenters in Wasserstein space were first discussed by Agueh and Carlier [Agueh 2011]. Designed through an analogy with barycenters in a Euclidean space, the Wasserstein barycenters of a family of measures are defined as minimizers of a weighted sum of squared Wasserstein distances. In our framework, barycenters can be obtained as

$$\bar{x} = \arg\min_x \sum_i \alpha_i W(x, x_i) \approx \psi(\sum_i \alpha_i \phi(x_i)), \tag{8.2}$$

where x_i are the data samples and the weights α_i obeys the following constraints: $\sum_i \alpha_i = 1$ and $\alpha_i > 0$. Note that when we have only two samples, the barycenter corresponds to a Wasserstein interpolation between the two distributions with $\boldsymbol{\alpha} = [1-t, t]$ and $0 \leq t \leq 1$ [Santambrogio 2014]. When the weights are uniform and the whole data collection is considered, the barycenter is the Wasserstein population mean, also known as Fréchet mean [Bigot 2017].

Next we evaluate our embedding on the task of computing Wasserstein Barycenters for each class of the *MNIST* dataset. We take 1000 samples per class from the test dataset and compute their uniform weight Wasserstein Barycenter using Eq. 8.2. The resulting barycenters and their Euclidean means are reported in Fig. 8.4. Note that not only those barycenters are sensitive but also preserve most of their sharpness which is a problem that occurs for regularized barycenters [Solomon 2015b, Benamou 2015]. The computation of those barycenters is also very efficient since it requires only 20ms per barycenter (for 1000 samples) and its complexity scales linearly with the number of samples.

We first compute the Wasserstein interpolation between four samples of each datasets in Figure 8.5. Note that these interpolation might not be optimal w.r.t. the objects but we clearly see a continuous displacement of mass that is characteristic of optimal transport. This leads to surprising artefacts for example when the eye of a face fuse with the border while the nose turns into an eye. Also note that there is no reason for a Wasserstein barycenter to be a realistic sample.

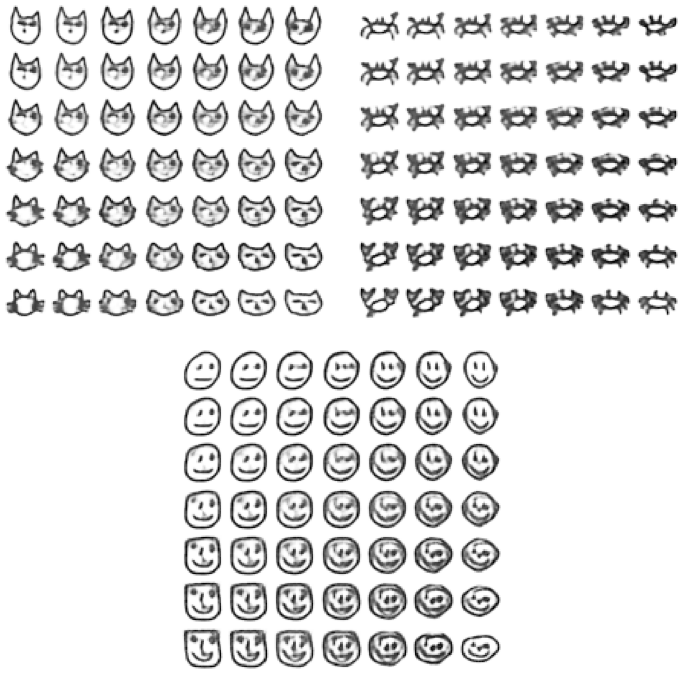


Figure 8.5: Interpolation between four samples of each datasets using **DWE**. (left) cat dataset, (center) Crab dataset (right) Face dataset.

Next we qualitatively evaluate the subspace learned by **DWE** by comparing the Wasserstein interpolation of our approach with the true Wasserstein interpolation estimated by solving the OT linear program and by using regularized OT with Bregman projections [Benamou 2015]. The interpolation results for all those methods and the Euclidean interpolation are available in Fig. 8.6. The LP solver takes a long time (20 sec/interp) and leads to a "noisy" interpolation as already explained in [Cuturi 2016]. The regularized Wasserstein barycenter is obtained more rapidly (4 sec/interp) but is also very smooth at the risk of loosing some details, despite choosing a small regularization that prevents numerical problems. Our reconstruction also looses some details due to the Auto-Encoder error but it is very fast and can be done in real time (4 ms/interp).

8.3.1.2 Principal Geodesic Analysis in Wasserstein space [Seguy 2015, Bigot 2017].

PGA, or Principal Geodesic Analysis, has first been introduced by Fletcher *et al.* [Fletcher 2004]. It can be seen as a generalization of PCA on general Riemannian manifolds. Its goal is to find a set of directions, called geodesic directions or principal geodesics, that best encode the statistical variability of the data. It is possible to define PGA by making an analogy with PCA. Let $x_i \in \mathbb{R}^n$ be a set of elements, the classical PCA amounts to *i)* find $\overline{x}$ the mean of the data and subtract it to all the samples *ii)* build recursively a subspace $V_k = \texttt{span}(v_1, \cdots, v_k)$ by solving

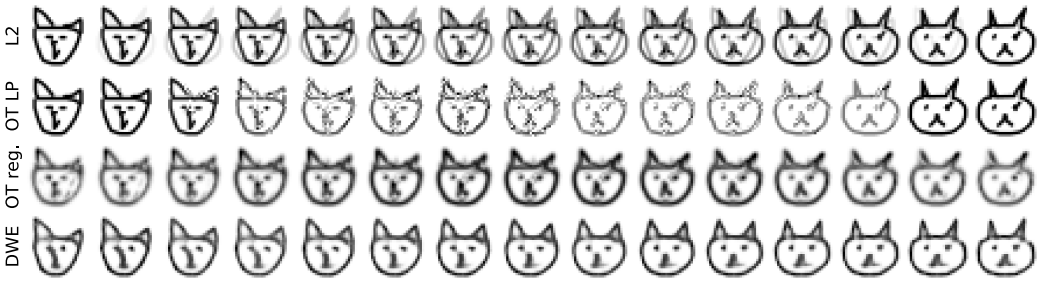


Figure 8.6: Comparison of the interpolation with L2 Euclidean distance (top), LP Wasserstein interpolation (top middle) regularized Wasserstein Barycenter (down middle) and **DWE** (down).

the following maximization problem:

$$v_1 = \text{argmax}_{|v|=1} \sum_{i=1}^{n} (v.x_i)^2, \quad v_k = \text{argmax}_{|v|=1} \sum_{i=1}^{n} \left((v.x_i)^2 + \sum_{j=1}^{k-1} (v_j.x_i)^2 \right). \tag{8.3}$$

Fletcher gives a generalization of this problem for complete geodesic spaces by extending three important concepts: **variance** as the expected value of the squared Riemannian distance from mean, **Geodesic subspaces** as a portion of the manifold generated by principal directions, and a **projection** operator onto that geodesic submanifold. The space of probability distribution equipped with the Wasserstein metric $(\mathcal{P}_p(X), W_2^2(X))$ defines a geodesic space with a Riemannian structure [Santambrogio 2014], and an application of PGA is then an appealing tool for analyzing distributional data. However, as noted in [Seguy 2015, Bigot 2017], a direct application of Fletcher's original algorithm is intractable because $\mathcal{P}_p(X)$ is infinite dimensional and there is no analytical expression for the exponential or logarithmic maps allowing to travel to and from the corresponding Wasserstein tangent space. We propose a novel PGA approximation as the following procedure: *i)* find $\overline{x}$ the approximate Fréchet mean of the data as $\overline{x} = \frac{1}{N}\sum_i^N \phi(x_i)$ and subtract it to all the samples *ii)* build recursively a subspace $V_k = \texttt{span}(v_1, \cdots, v_k)$ in the embedding space (v_i being of the dimension of the embedded space) by solving the following maximization problem:

$$v_1 = \text{argmax}_{|v|=1} \sum_{i=1}^{n} (v.\phi(x_i))^2, \quad v_k = \text{argmax}_{|v|=1} \sum_{i=1}^{n} \left((v.\phi(x_i))^2 + \sum_{j=1}^{k-1} (v_j.\phi(x_i))^2 \right). \tag{8.4}$$

which is strictly equivalent to perform PCA in the embedded space. Any reconstruction from the corresponding subspace to the original space is conducted through ψ. We report in Figure 8.7 the Principal Component Analysis (L2) and Principal Geodesic Analysis (**DWE**) for 3 classes of the *MNIST* dataset. We can see that using Wasserstein to encode the displacement of mass leads to more semantic and

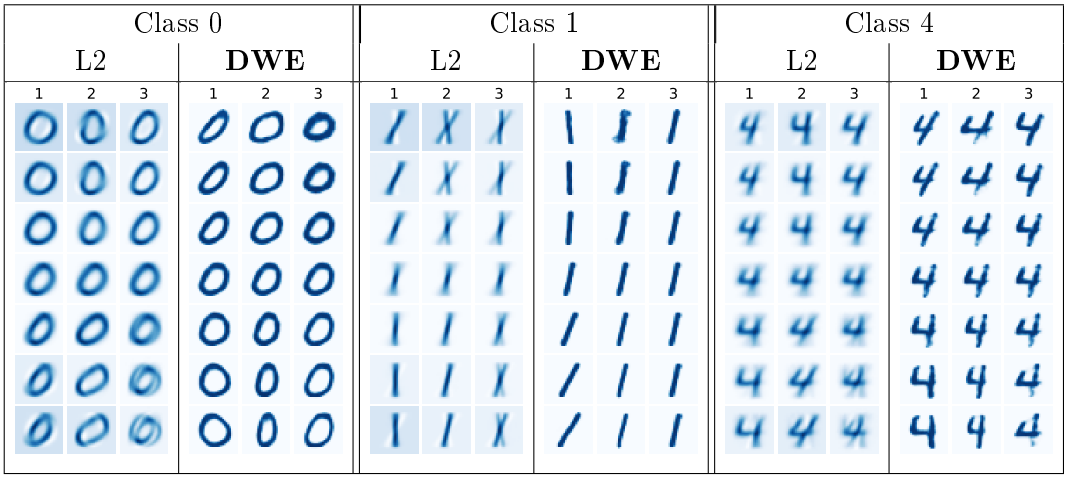


Figure 8.7: Principal Geodesic Analysis for classes 0,1 and 4 from the *MNIST* dataset for squared Euclidean distance (L2) and Wasserstein Deep Learning (**DWE**). For each class and method we show the variation from the barycenter along one of the first 3 principal modes of variation.

nonlinear subspaces such as rotation/width of the stroke and global sizes of the digits. This is well known and has been illustrated in [Seguy 2015]. Nevertheless our method allows for estimating the principal component even in large scale datasets and our reconstruction seems to be more detailed compared to [Seguy 2015] maybe because our approach can use a very large number of samples for subspace estimation.

8.3.1.3 Other possible methods.

As a matter of facts, several other methods that operate on distributions can benefit from our approximation scheme. Most of those methods are the transposition of their Euclidian counterparts in the embedding space. Among them, clustering methods, such as Wasserstein k-means [Cuturi 2014], are readily adaptable to our framework. Recent works have also highlighted the success of using Wasserstein distance in dictionary learning [Rolet 2016] or archetypal Analysis [Wu 2017]. Few works have adressed the question of discriminating distributions by taking into account that the instances are discrete distributions by itself. [Rakotomamonjy 2018] have studied the potential of using Wasserstein distance as a dissimilarity function for empirical distributions. They mostly rely on the previous works of Balcan *et al.* demonstrates that for some learning problem, by using the appropriate divergence function, one can achieve low error linear decision functions with high probability [Balcan 2008]. The approach they advocate for empirical distributions is the following: they first compute exact Wassertstein distances between their training distributions and some fixed distributions that they denote as patterns, and then in a second time, they use such distances as a set of features to learn a classifier. Our method allows a fast computation of pairwise Wasserstein distance which increases

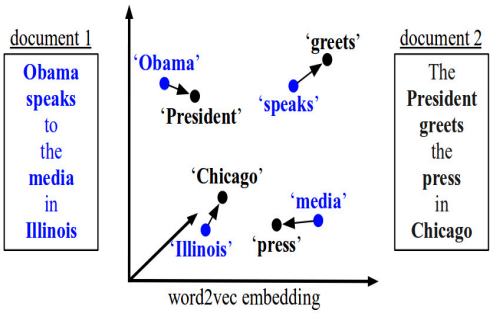


Figure 8.8: Illustration of WMD. Words are embedded with Word2Vec. The distance between the two sentences is the minimum cumulative distance that all words from the first sentence need to *travel* to be transformed into the target sentence.

the speed up for Wasserstein based nearest neighbor classifiers.

8.4 Future work: Wasserstein for Text Mining

Wasserstein distance also denoted as Earth Mover's Distance has been widely adopted for Text mining tasks, ranging from information retrieval, to establish a cross-lingual connection without any supervision [Kusner 2015, Kumar 2017, Balikas 2018]. In particular, [Kusner 2015] proposed Wasserstein distance as a similarity metric for Text document, leveraging on the word embedding used. They denote their method as WMD, standing for Word Mover's Distance. It achieved state-of-the-art error rates in nearest neighbor classification for document. WMD defines the distance between two documents as the Wasserstein distance between the empirical distributions induced by the words from one text to another. The distance between two words is generally defined as the euclidian distance between their embeddings. This holds sense, as the embedding in used in the NLP community, has semantic properties given the euclidian distance [Kusner 2015]. Huang *et al.* extend WMD with supervision [Huang 2016b]: they learn the usefulness of specific words for the classification task. Figure 8.8 illustrates the semantic similarities highlighted by the use of WMD.

When it comes to the use of Wasserstein distance on text, we identify two contributions:

- **information retrieval**: Despite the efficiency of WMD to naturally measure semantic similarities between texts, it scales with a cubic complexity reguarding the dimensionality of the documents. Thus, our first goal is to approximate at best the Wasserstein distance, plus with a lightened architecture that speeds up the computation of pairwise Wasserstein.

- **Text Generation given Wasserstein**: Similarly as other similarity metrics, WMD cannot create new sentences based only on the distance given another

document [1]. In that aim, we propose to extend DWE to text using a sequence to sequence encoder-decoder architecture.

8.4.1 Information Retrieval: Fast computing of WMD at large scale

If the goal is solely information retrieval, one can consider a bag-of-words architecture. Indeed, Wasserstein is not considering the structure of the sentence, but the appearance of the words, independently of their order. To alleviate the lack of structure, futures works could envisage variants of Gromov Wasserstein distance, like the recent one proposed in [Vayer 2018]. Unlike Reccurrent networks that takes naturally in account the position of the words, we could envisage a static architecture taking a bag-of-words as an input. This constraint is highly relevant in the case one wants to approximate at best the Wasserstein distance for any kind of input: on two sentences containing the same words but having a different meaning, our network will output a unique embedding. Thus, a network that would best suit to measuring Wasserstein is a multi-layer perceptron along the embdeding dimension that slides along the words. We decompose our architecture into two sub-networks: a first network ψ encodes each word in the sentence into a vector. Those vectors are then sum together and embedded into a second network τ that will output the embedding that mimics the Wasserstein distance. It is important to use the addition as the pooling function to preserve the information and the occurences. If using the average pooling, then the document "*Stop ! Stop ! Stop*" will have the same internal representation as "*Stop !*". The global function reads:

$$min_{\psi,\tau}||\Big(\tau(\sum_i \psi(x_{1,i})) - \tau(\sum_j \psi(x_{2,j}))\Big)^2 - W_2^2(x_1, x_2)||^2 \tag{8.5}$$

We provide further details about our architecture in Fig. 8.9.

[1] cf Section 11.2

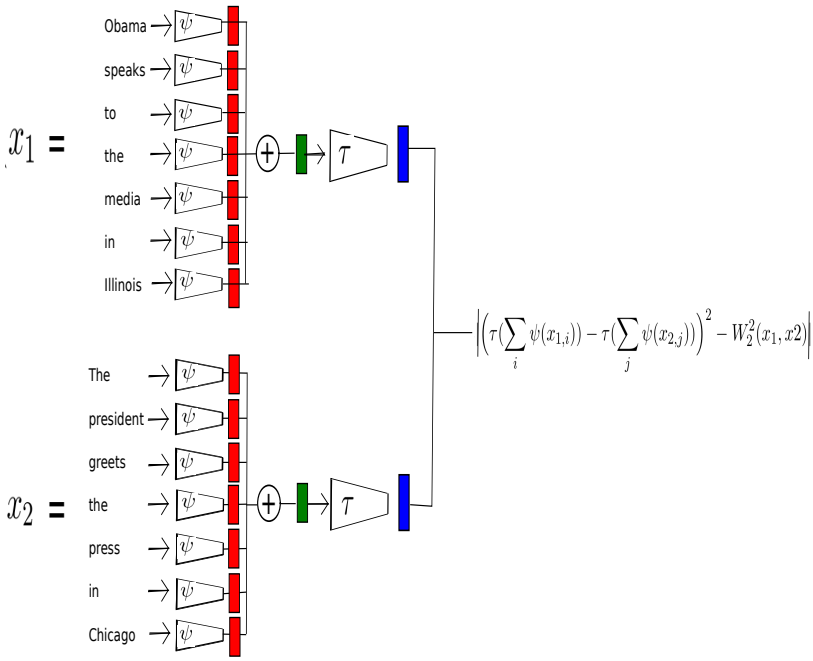


Figure 8.9: Two sentences x_1 and x_2 are sampled from the data distribution. Each of their words are encoded by a first network ψ whose input size matches the dimension of the words' embedding. ψ outputs a vector for each word which are then summed together into a unique vector. This vector is finally encoded by a second network τ thats computes the embedding that mimics the Wasserstein distance.

8.4.1.1 Empirical Evaluation:

We evaluate our approach on two databases:

- the *Twitter*dataset: a set of tweets labeled with sentiments *positive*. The words are embedded with Word2Vec.

- the Visual Question Answeringdataset: we retrieve the questions from the SQUAD dataset. The words are embedded with the Glove representation.

Thanks to the two datasets, we can validate our method independently from the words' representations. We present our ongoing results in Table 8.2. Preliminary results are encouraging, but they need major improvements: up to now, we obtain better results when always predicting the mean Wasserstein distance computed on the training set.

In the next section, we investigate whether reccurent architectures handles better our Wasserstein based embedding for text. Thanks to Reccurent Units, we will be able to promote a decoder and use it for text generation given Wasserstein distance.

	Twitter	Visual Question Answering
MSE	0.084	72.15
rMSE	0.012	0.061

Table 8.2: Mean Squared Error (MSE) and Relative Mean Squared Error obtained on an independent test set, respectively on the *Twitter*and Visual Question Answeringdataset. We can see that **DWE** is actually learning the exact Wasserstein distance. Although, those results should be leveraged by the RMSE obtained when predicting the mean Wasserstein distance between two random sentences of *Twitter* (0.012) and Visual Question Answering (0.036).

8.4.2 Text Generation given Wasserstein Distance

The main limitation of our previous architecture is that it cannot decode the sentence, which limits the use of our model to data mining applications, as done in Section 8.3.1. We formulate our model given Eq. 8.6. Similarly as in **DWE**, a first recurrent network ψ takes as input the ordered sequences of words' embedding, and outputs in its last state the embedding vector. Such embedding should mimic the Wasserstein distance using instead the Euclidian Distance. Finally, we init the hidden state of the decoder τ given the embedding state to generate our input sentence. Note that we do not use a mean squared loss to train the decoder τ, but the categorical cross-entropy loss l.

$$min_{\psi,\tau}||\Big(\psi(x^1)-\psi(x^2)\Big)^2-W_2(x_1,x_2)\ ||^2+l\Big(\psi(\tau(x_1)),x_1\Big)+l\Big(\psi(\tau(x_2)),x_2\Big) \quad (8.6)$$

Our requirements are two folds. First, we need a decoder that can reconstruct the sentence given the last state of the encoder. Secondly, the last state of the encoder should embed Wasserstein distance given the Euclidian distance, as initially proposed in **DWE** for CNNs. Using a decoder would help us to compute optimizations efficiently in the Wasserstein space. Figure 8.10 shows a diagram of our encoder-decoder network. Note that the design of our architecture takes its inspiration from the recurrent networks combined with greedy search originally used to sentence translation.

So far, we have not investigated thoroughly the potential of our **DWE** RNN model on texts. Our first results are encouraging. Without incorporating the decoder into the training, we obtain major improvements of the Wasserstein embedding compared to our previous formulation. On the Visual Question Answering dataset, we obtain a MSE of 15.9 on the test set.

8.4.2.1 Wasserstein based Adversarial example for NLP

Next, we describe the potential of our **DWE** RNN model. Adding a decoder to our model helps to generate instances sampled from the unknown ground-truth distribution, given a certain criterion in the Wasserstein space. For example, we

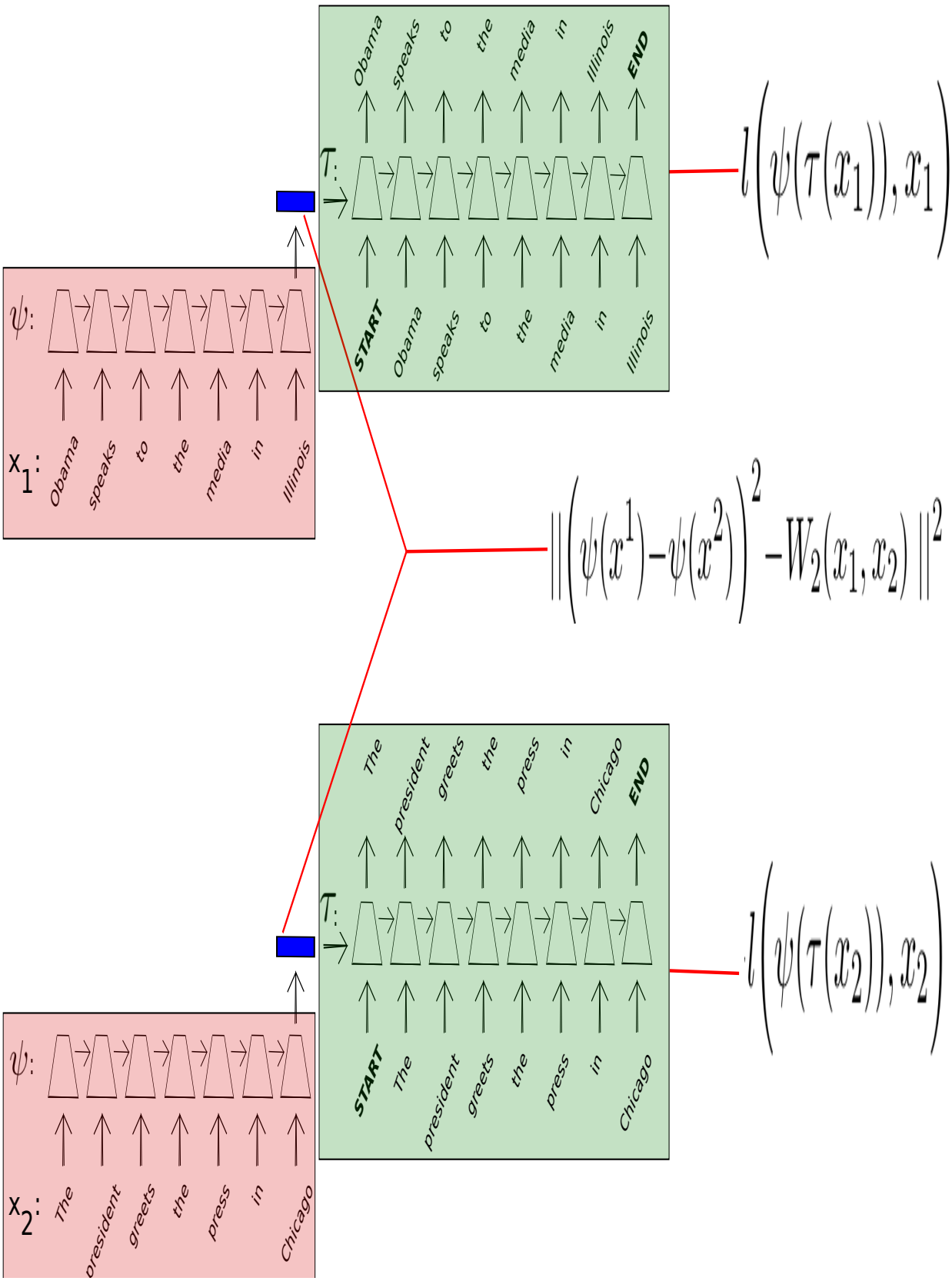


Figure 8.10: Deep Reccurent Wasserstein Embedding for Text

describe our attempts to generate adversarial examples on NLP classification tasks. We detail existing work on creating adversarial tasks for text in Section III.

Indeed, Wasserstein is well indicated to measure similarity between sentences or documents. Thus, if we want to generate an adversarial sample, as close as possible to the source sentence, we can use Wasserstein as a good metric for the distortion between the true and fake sentences.

Given a source sentence x, and a classification system f, generating our adversarial sentence to get f mistaken reads:

$$\tilde{x} = \arg\min \; W_2(x, \tilde{x}) \;\; s.t \; l(f(x)) \neq l(f(\tilde{x})) \tag{8.7}$$

In the previous formulation, one needs to be able to compute the gradient to perform optimization scheme, such as L-BFGS [Szegedy 2013]. However, our sentences lie in discrete space, so computing the gradient is intractable. Moreover, one can imagine an approximate optimization scheme, where we optimize $\tilde{x}$ given only the Wasserstein distance. But, the Wasserstein distance considers the sentences as a bag-of-words, unlike f which generally required an ordered sequence. A possible solution may be to alleviate the formulation by transposing the search into our embedding space.

$$\tilde{x} = \tau(y) \; where \; y = \arg\min ||\psi(x) - y||^2 \; s.t \; l(f(x)) \neq l(f(\tau(y))) \tag{8.8}$$

8.5 Conclusion

In this chapter, we presented a computational approximation of the Wasserstein distance suitable for large scale data mining tasks. Our method finds an embedding of the samples in a space where the Euclidean distance emulates the behavior of the Wasserstein distance. Thanks to this embedding, numerous data analysis tasks can be conducted at a very low computational price. We forecast that this strategy can help in generalizing the use of Wasserstein distance in numerous applications.

However, while our method is very appealing in practice it still raises a few questions about the theoretical guarantees and approximation quality.

First, embedding Wasserstein space in normed metric space is still a theoretical and open questions [Matoušek 2011]. Most of the theoretical guarantees were obtained with W_1. In the simple case where $X = \mathbb{R}$, there exists an isometric embedding with L_1 between two absolutely continuous (*wrt.* the Lebesgue measure) probability measures μ and ν given by their by their cumulative distribution functions F_μ and F_ν, *i.e.* $W_1(\mu, \nu) = \int_{\mathbb{R}} |F_\mu(x) - F_\nu(x)| dx$. This fact has been exploited in the computation of sliced Wasserstein distance [Bonneel 2015, Kolouri 2016c]. Conversely, there is no known isometric embedding for pointsets in $[n]^k = \{1, 2, \ldots, n\}^k$, *i.e.* regularly sampled grids in $\mathbb{R}^k$, but best known distortions are between $O(k \log n)$ and $\Omega(k + \sqrt{\log n})$ [Charikar 2002, Indyk 2003, Khot 2006]. Regarding W_2, recent results [Andoni 2016] have shown there does not exist meaningful embedding over

$\mathbb{R}^3$ with constant approximation. Their results show notably that an embedding of pointsets of size n into L_1 must incur a distortion of $O(\sqrt{\log n})$. Regarding our choice of W_2^2, there does not exist embeddability results up to our knowledge, but we show that, for a population of locally concentrated measures, a good approximation can be obtained with our technique. Moreover it is difficult to foresee from a given network architecture if it is sufficiently (or too much) complex for finding a successful embedding. It can be conjectured that it is dependent on the complexity of the data at hand and also the locality of the manifold where the data live in.

Second, the theoretical existence results on such Wasserstein embedding with constant distortion are still lacking. Future works should consider these questions as well as applications of our approximation strategy on a wider range of ground loss and data mining tasks. Also, the transferability of one database to another to diminish the computational burden of computing Wasserstein distances on numerous pairs for the learning process should be studied.

Chapter 9

Perspective: Wasserstein prototypes

Contents

9.1 Introduction **95**
9.2 Approximate Submodularity for Wasserstein distance . . . **96**
9.2.1 Greedy selection of Prototypes 98
9.3 Empirical Validation **99**
9.4 Active Learning **102**
9.5 Conclusion **103**

9.1 Introduction

Summary

- We guide this work towards selecting prototypes representative of the ground truth distribution, thanks to Wasserstein distances
- We demonstrate the weak submodularity of selecting Wasserstein prototypes
- We propose approximations using SINKHORN and IPOT that can fasten the selection of Wasserstein prototypes.
- We investigate the applications of Wasserstein prototypes in active learning.

✓ *Every dataset and parameters used to conduct our experiments are available in the* ***dataset*** *section A.1 and the* ***hyperparameter***

mducoffe/Greedy_Wass PROOF Proofs are available in **Appendix A.3.4**

In this Chapter, we demonstrate how to select examples representative of their distribution by using Wasserstein distances. Such examples, that we denote Wasserstein prototypes are highly relevant to illustrate the modes of their underlying complex distribution and improve the interpretability of the data. One of the main advantages of our method is that it is parameter-free unlike MMD prototypes

[Huszár 2012], and owns proof of convergence thanks to the well-known convergence results about submodular functions. We are able to select prototypes for any given sparsity level. Due to the cost for computing the exact Wasserstein distance, we also propose two approximations of varying computational cost using the entropic regularization or the inexact proximal point method.

Wasserstein prototypes are highly interesting for a wide set of tasks ranging from visualization (facilitate human understanding and reasoning), interpretability of classification decision, active learning and generative modeling. Numerical experiments supporting these idea are conducted on simulated and real datasets and demonstrate the benefits of our methods. In particular, we illustrate several use-cases with empirical experiments on *MNIST*.

9.2 Approximate Submodularity for Wasserstein distance

Wasserstein prototypes are samples from a known empirical distribution, that are the most informative, considering the Wasserstein distance. Formally Wasserstein prototypes consists in minimizing the Wasserstein distance between the fixed empirical distribution, and Wasserstein prototypes, knowing that they are part of a dataset. The main difference between Wasserstein prototypes and Wasserstein barycenters is that prototypes are part of a discrete and known-beforehand distribution unlike barycenters. Hence, prototypes can be used for visualizing and analyzing the properties of the dataset unlike barycenters, that, while looking similar to the datasets, may not be representative of the statistics of the data. Moreover, in high dimensional space, barycenters may suffer from bluriness, and unrepresent biases [Agueh 2011]. Note that the main difference between our problem and *Wasserstein Core-Set* is that we fix our support, unlike *Wasserstein Core-Set* which creates barycenters.

Definition 9.2.1: Wasserstein prototypes

Formally given an empirical distribution $\mathcal{P}$, a fixed size of data to label, K; and a subset $\mathcal{U} \subset \mathcal{P}$, we denote Wasserstein prototypes a subset $\mathcal{S}_K^* \subset \mathcal{P}$ that minimizes Eq. 9.1.

$$\min_{\mathcal{S}_K^* \subset \mathcal{P},\, |\mathcal{S}_K^*| = K} W_p(\mathcal{S}_K^*, \mathcal{U}) \tag{9.1}$$

The optimization subroutine we define in 9.1, is generally not tractable as it is combinatorial given $\mathcal{P}$. The desiderata for solving Equation 9.1 naturally implies the notion of submodularity. Submodular functions are widely used in the approximation of combinatorial problem as their optimization may be efficiently solved through greedy search. Although, maximizing a submodular function under cardinality constraints is NP-hard, when the function is *non-negative* and *monotone*, [Nemhauser 1978, Kim 2016] demonstrated that a naive greedy selection algorithm provides the best approximation to the optimal solution: the greedy solution is

guaranteed not to differ from the optimal strategy by more than a fixed constant, roughly 63%.

Definition 9.2.2: Submodularity

If $s : 2^{\mathcal{X}} \mapsto \mathbb{R}$ is a **non-negative**, **monotone**, **submodular** function and is used to select greedily the set of prototypes $\mathcal{S}_n$, such that $\mid \mathcal{S}_n \mid= n$ then, if we denote by $\mathcal{S}^*$ the optimal solution, the following upper bound holds:

$$s(\mathcal{S}_n) \geq (1 - \frac{1}{e}) \max_{|\mathcal{S}^*| \leq n} s(\mathcal{S}^*) \tag{9.2}$$

Unfortunaltely, submodularity is a strong property that generally does not hold in prototypical selection [Huszár 2012]. Indeed, Bayesian prototypical selection is known to satisfy a weaker condition of convergence denoted as *weak submodularity*.

Definition 9.2.3: Weak Submodularity

If $s : 2^{\mathcal{X}} \mapsto \mathbb{R}$ is a **weakly submodular** function with constant ε and is used to select greedily the set of prototypes $\mathcal{S}_n$, such that $\mid \mathcal{S}_n \mid= n$ then, if we denote by $\mathcal{S}$ the optimal solution, the following upper bound holds:

$$s(\mathcal{S}_n) \geq (1 - \frac{1}{e}) \max_{|\mathcal{S}| \leq n} s(\mathcal{S}) - n\varepsilon \tag{9.3}$$

The property of Wasserstein prototypes lies in between submodularity and weak submodularity. Indeed, for Wasserstein prototypes, we do not have an additional factor $n\varepsilon$ in our upper bound, that is linearly growing given the number of prototypes, but a constant additional factor. However, unlike MMD, our objective function is not strictly monotone, as required in the previous definition. Thus we cannot obtain an upper bound given the maximum over all possible combinations of size less than k, but for size k only. In fact we provide our own definition of *approximate submodularity* for Wasserstein prototypes, in Th. 2.1. All the proofs are available in Appendix A.3.4.

Theorem 2.1: Approximate Submodularity

Suppose (Ω, ρ) is a metric space and suppose $\mathbb{P}$ is the uniform distribution based on a countable set $\mathcal{P} \subseteq \Omega$. We denote respectively by $Sep(\mathcal{P})$ and $Diam(\mathcal{P})$ the minimum and maximum distance between two distinct samples in $\mathcal{P}$. If we greedily select the set of prototypes $\mathcal{S}_n$, such that $\mid \mathcal{S}_n \mid= n$ then, if we denote by $\mathcal{S}_n^*$ the optimal solution of cardinality n, the following upper bounds holds:

$$W(\mathbb{P}, \mathcal{S}_n) \leq (1 - \frac{1}{e}) W(\mathbb{P}, \mathcal{S}_n^*) + \frac{1}{e} Diam(\mathcal{P}) \tag{9.4}$$

Although it is interesting to underline the approximate submodularity of Wasserstein prototypes, it does not provide strong theoretical guarantees on the quality of the Wasserstein prototypes found with greedy search. Indeed, due to the linear decay of Wasserstein given the number of samples, we know that few random samples are enough to make this upper bound not tight anymore. However, in practice, greedy selection over weakly submodular functions remains highly efficient [Krause 2010]. Moreover, such submodularity properties opens up interesting directions for future research, especially for sparse dictionary selection using Wasserstein distance [Rolet 2016].

9.2.1 Greedy selection of Prototypes

The scalability of Wasserstein prototypesis limited by the computational cost to evaluate the exact Wasserstein distance which is of order $O(n^3 log(n))$ for discrete probability distribution with a support of size n. A first solution to overcome the lack of scalability of Wasserstein prototypesis to rely on entropic regularization, also known as the Sinkhorn algorithm. Thanks to the power iterative method of the Sinkhorn algorithm, relying only on matrix multiplications, computing Wasserstein prototypeswith Sinkhorn is highly scalable and may also be embedded on gpus. We evaluate empirically the validity of such approximation for Wasserstein prototypesin Section 9.3. However, we have no theoretical motivations to use entropic regularization, nor any theoretical assessment.

Nevertheless, as also highlighted in Section 7.3, the regularization scaling is a major hyperparameter that will affect either the quality of our prototypes or the computational speed up of using Sinkhorn distance rather than the exact Wasserstein. Another solution is to rely on IPOT. IPOT is promising due to its similarity with the Sinkhorn distance, it can speed up the computations. Moreover, its regularization parameter is far less controversial than the Sinkhorn distance. Instead of waiting for convergence, we limit the maximum number of iterations by an hyperprameter and fix the number of inner iterations to one (in accordance with the empirical analysis provided in [Xie 2018]).

Every dataset and parameters used to conduct our experiments are available in the **dataset** section A.1 and the **hyperparameter**

9.3 Empirical Validation

We illustrate on three toy datasets (*samples from a circle, samples from 3 univariate gaussians, samples from 3 gaussians with different ratio*) the prototypes selected by our three options, in Figures 9.1 , 9.2, 9.3:

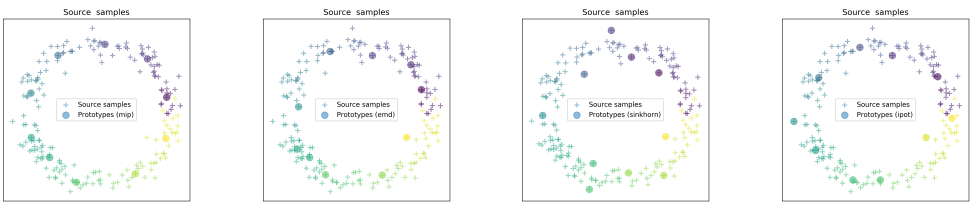
• **Optimal prototypes**: Prototypes given the exact Wasserstein, computed using a Mixed Integer Programming framework

• **EMD prototypes**: Prototypes selected with greedy search given the exact Wasserstein

• **SINKHORN prototypes**: Prototypes selected with greedy search given the Sinkhorn distance ($\varepsilon = 1e-3, iter = 10$)

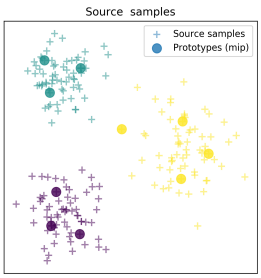
• **IPOT prototypes**: Prototypes selected with greedy search given the IPOT algorithm ($\varepsilon = 1e-3, iter = 10$)

For example, in Figure 9.1, the optimal prototypes are spread uniformly along the inner circle of the distribution. We select greedily the samples given the ones that minimizes their Wasserstein distance with the set of samples in Figure 9.1(b) (EMD prototypes). As expected, Wasserstein prototypes are highly similar to the MIP prototypes. When it comes to both our approximations, Sinkhorn prototypes (Figure 9.1(c)) and IPOT prototypes (Figure 9.1(d)), they underperform EMD prototypes, in particular, the Sinkhorn prototypes are mostly spread out of the distribution.

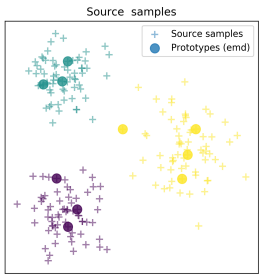


(a) Optimal Prototypes (b) EMD Prototypes (c) Sinkhorn Prototypes (d) IPOT Prototypes

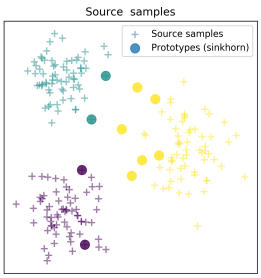
Figure 9.1: *Circle:* Visualization of prototypes selected given different criteration with p=2



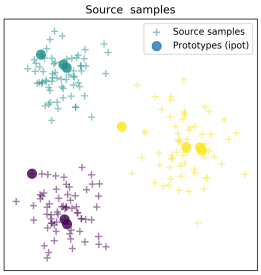
(a) Optimal Prototypes



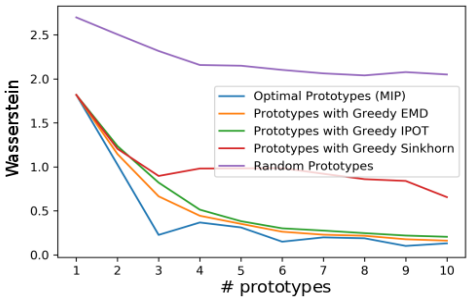
(b) EMD Prototypes



(c) Sinkhorn Prototypes

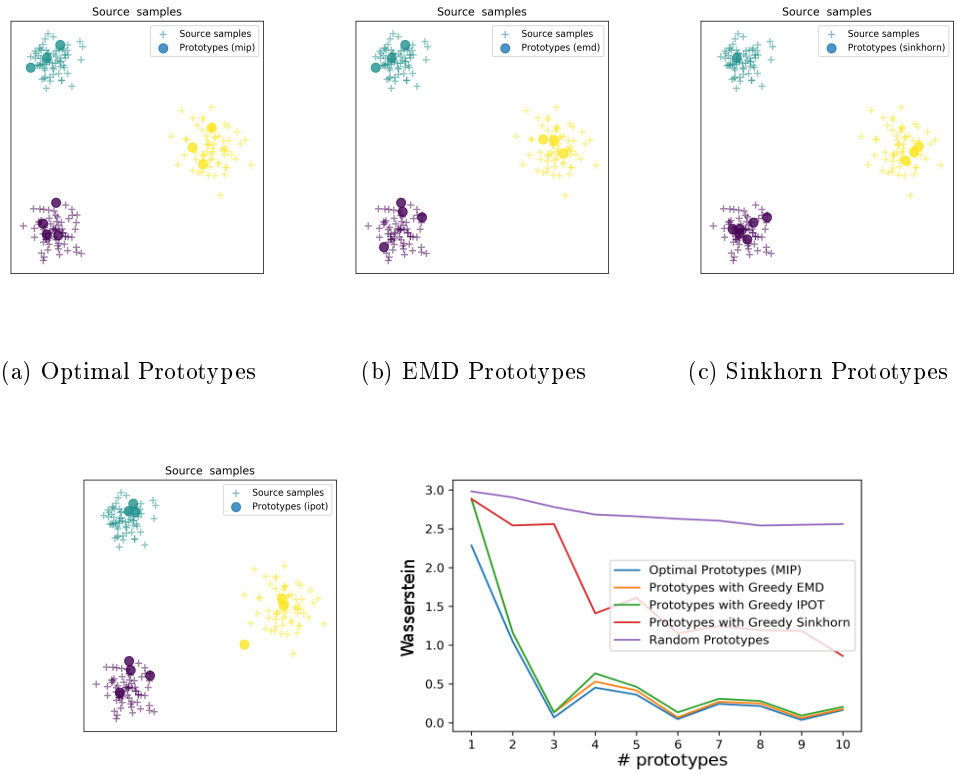


(d) IPOT Prototypes



(e) Evolution of the Wasserstein distance

Figure 9.2: *3 univariate Gaussians:* Visualization of prototypes selected given different criteration with p=2

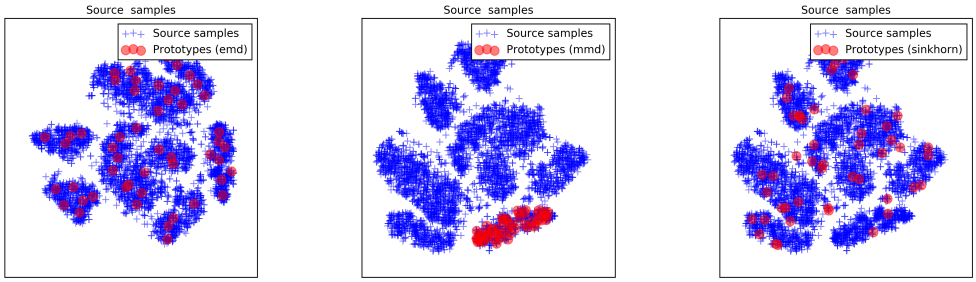


(a) Optimal Prototypes (b) EMD Prototypes (c) Sinkhorn Prototypes

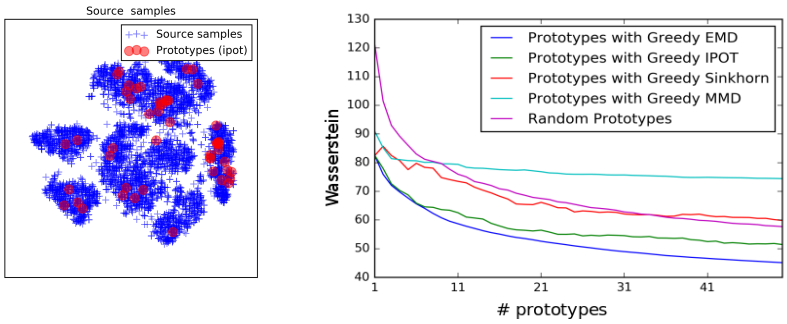
(d) IPOT Prototypes (e) Evolution of the Wasserstein distance

Figure 9.3: *3 Gaussians with different ratio:* Visualization of prototypes selected given different criteration with p=2

We also proceed on higher dimensional dataset: we conduct numerical experiments on *MNIST* in Figure 9.4. Although computing the optimal prototypes was not tractable, we added another method **MMD prototypes**, which adds greedily samples that minimize their MMD score with the distribution. We use tSNE to plot the prototypes selected by our methods and evaluate the evolution of Wasserstein distances along the sequence of prototypes. While IPOT prototypes appear promising, as their evaluation in Figure 9.4(e) remains close to EMD prototypes, on the contrary both SINKHORN and MMD prototypes perform worse than a random selection.



(a) EMD Prototypes (b) MMD Prototypes (c) Sinkhorn Prototypes



(d) IPOT Prototypes (e) Evolution of the Wasserstein distance

Figure 9.4: *3 Gaussians with different ratio:* Visualization of prototypes selected given different criteration with p=2

The usage of Wasserstein prototypes can be dedicated to several machine learning tasks. Among others, we suggest future explorations for active learning. Indeed, selecting wasserstein prototypes may promote more diverse queries, as also hilighted in Section 5.2.

9.4 Active Learning

The previous Part has highlighted the needs for increasing the diversity when querying batches of unlabelled samples in an active learning context. While both KL divergence and MMD have been investigated as potential solutions for batch active learning [Wei 2015, Wang 2015]; to our knowledge, it has never been the case for Wasserstein distance. However Wasserstein distance holds many advantages in an active learning context. Indeed, Wasserstein can be applied to distributions with non-overlapping supports and has good out-of-sample performance. Moreover, it is robust to discrete distributions without the need to resort to kernel estimators, and is parameter-free, unlike MMD [Muandet 2017]. Nevertheless, we have described

in Section 5.2 how Wasserstein may help in upper bounding the *approximate risk* over a classification task 7.3.3. Considering Wasserstein prototypes will help to get closer to the empirical risk, as we will diminish the radius of the Wasserstein ball. This encourage the usage of Wasserstein prototypes as a post-processing step in an active learning heuristic, in order to cover at best the groundtruh distribution.

We have integrated Wasserstein prototypes into our active learning heuristic **BalNet** . If our goal is to label k unlabelled samples, our method will consist in pre-selecting k*M (M being an hyperparameter) potential candidates given the active learning criterion itself, and then selecting k Wasserstein prototypes among the previous pool. Preliminary experiments conducted on *MNIST*in Figure 9.5 indicate that the usage of Wasserstein prototypes speeds up the convergence of **BalNet**.

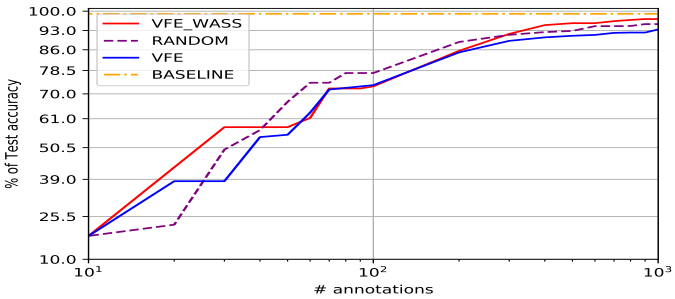


Figure 9.5: Comparison of **BalNet**[Ducoffe 2016c] with (–) and without (–) using Wasserstein prototypes as a post processing step to select the queries (with p=2 for the Wasserstein distance). We measure the test accuracy on *MNIST* trained on LeNet5 given the number of annotations.

9.5 Conclusion

We presented Wasserstein prototypes, a framework to help give insights on complex data distribution, thanks to Wasserstein distances. We demonstrated the weak submodularity of selecting Wasserstein prototypes and develop heuristics to fasten the computation of Wasserstein prototypes, using either SINKHORN or IPOT. So far we have evaluated Wasserstein prototypes under a specific setting which is when we sample the prototypes from the distribution on which we measure the Wasserstein distance: $\mathcal{S} \subset \mathcal{U}$. However, we can also operate on non-overlapping support, as long as it makes sense to represent their samples into a shared space and compute the distance between them. Indeed, the approximate submodularity of Wasserstein prototypes holds also in this setting. By selecting generated prototypes, we can measure their Wasserstein distance along the training and test set. This score provides a glimpse into the overfitting of a generator. Nevertheless, we investigate the applications of Wasserstein prototypes in active learning.

Chapter 10

Conclusion

In summary, we studied new applications of Wasserstein distance to machine learning. In particular, we empirically demonstrated how we could approximate pairwise Wasserstein distances on distributions sampled from a low dimensional manifold. To do this, we successfully use Siamese Neural Network. From our framework results in great opportunities in data mining and text mining tasks. Among the possible applications, we plan to analyze the generation of adversarial attacks based on Wasserstein. Second, we study an innovative direction for the Wasserstein distance with the selection of a subset representative of an empirical distribution. To our knowledge, our work is the first to rely on Wasserstein to generate prototypes. For our future work, we hope to further explore the properties of our Wasserstein prototypes and their effectiveness in active learning.

Part III

Understanding the behavior of Neural Networks on Linguistic tasks

Chapter 11

Introduction

Contents

11.1 Motivation **109**

11.2 Litterature **110**

11.2.1 CNNs for Text classification 110

11.2.2 Visualization of Deep network 111

11.2.3 Model Agnostic Explanation 112

11.2.4 Adversarial example for NLP 112

11.1 Motivation

As in many other fields of data analysis, NLP has been strongly impacted by the recent advances in Machine Learning, more particularly with the emergence of Deep Learning techniques. For example, deep networks have been embedded in various NLP tasks, ranging from machine comprehension to authorship classification, VQA or sentiment analysis [Yu 2018, Ducoffe 2016a, Antol 2015, Glorot 2011]. These techniques outperform state-of-the-art approaches on a wide range of NLP tasks, and so they have been quickly and intensively used in industrial systems. Such systems rely on end-to-end training on massive amounts of data, making few prior assumptions about linguistic structure and focusing on statistically frequent patterns. Thus, they somehow step away from computational linguistics as they learn implicit linguistic information automatically without aiming at explaining or even exhibiting classic linguistic structures underlying the decision. Lately, some works have focused on understanding the black box decisions and the linguistic patterns on which depends the network's decisions. We describe those works in Section 11.2. Indeed, understanding the flaws of deep learning in NLP is a significant task, as Text is one of the most intuitive ways to establish communication protocols between computers and humans. However, if such systems are biased and miscommunicate, not only there is a chance to create some frustration, but also we can jeopardize the users. We illustrate a naive use case of deep network's failures in Fig 11.1.

What color is the tray?	Pink
What colour is the tray?	Green
Which color is the tray?	Green
What color is it?	Green
How color is tray?	Green

Figure 11.1: **Adversarial Questions for state-of-the-art VQA systems [Ribeiro 2018b]**. The authors paraphrase the questions which highly impacts the quality of the answer.

We dedicate this chapter to new machine learning and linguistic analysis to highlight some linguistic observables learned by deep neural networks, in particular CNNs. Highlighting such linguistic patterns hold several goals:

- If we understand the type of linguistic information relevant for learning a specific task, NLP datasets and annotations may benefit from it and contain less bias.
- We can optimize the network's architecture and words embedding
- We can improve our evaluation pipeline
- We can provide new observations tools to linguistic experts to analyze their corpora.

In the next section, we describe recent advances towards understanding deep networks for NLP tasks. As we focus our analysis on text classification, we will mainly present CNNs for NLP tasks.

11.2 Litterature

11.2.1 CNNs for Text classification

CNNs are widely used in the computer vision community for a broad panel of tasks: ranging from image classification, object detection to semantic segmentation. It is a bottom-up approach where we applied an input image, stacked layers of convolutions, non-linearities, and sub-sampling.

Encouraged by the success for vision tasks, researchers applied CNNs to text-related problems [Kalchbrenner 2014, Kim 2014]. The use of CNNs for sentence modeling traces back to [Collobert 2008]. Collobert adapted CNNs for various NLP problems including Part-of-Speech tagging, chunking, Named Entity Recognition and semantic labeling. CNNs for NLP work as an analogy between an image and a text representation. Indeed each word is embedded in vector representation. Then several words build a matrix (concatenation of the vectors). If Recurrent Neural Networks (*mostly GRU and LSTM*) are known to perform well on a broad range

of tasks for text, recent comparisons have confirmed the advantage of CNNs over RNNs when the task at hand is mostly a keyphrase recognition task [Yin 2017].

In Textual Mining, we aim at highlighting linguistics patterns to analyze their contrast: specificities and similarities in a corpus [Feldman, R., and J. Sanger 2007, L. Lebart, A. Salem and L. Berry 1998]. It mostly relies on frequential based methods such as z-scoring. However, such existing methods have so far encountered difficulties in underlining more challenging linguistic knowledge, which has yet been empirically observed, for instance syntactical motifs [Mellet 2009a]. In that context, supervised classification, especially CNNs, may be exploited for corpus analysis. Indeed, CNN learns parameters automatically to cluster similar instances and drive away examples from different categories. Eventually, their prediction relies on features which inferred specificities and similarities in a corpus. Projecting such features in the word embedding will reveal important spots and may automatize the discovery of new linguistic structure as the previously cited, syntactical motifs. Moreover, CNNs hold other advantages for semantic analysis. They are static architectures that, according to specific settings are more robust to vanishing gradient, and thus can also model long-term dependency in a sentence [Dauphin 2017, Wen 2017, Adel 2017]. Such a property may help to detect structures relying on different parts of a sentence.

11.2.2 Visualization of Deep network

All previous works converged to a shared assessment: both CNNs and RNNs provide relevant, but different kinds of information for text classification. However, though several works have studied linguistic structures inherent in RNNs, to our knowledge, none of them have focused on CNNs. The first line of research has extensively studied the interpretability of word embeddings and their semantic representations. When it comes to deep architectures, [Karpathy 2015] used LSTMs on character level language as a testbed. They demonstrate the existence of long-range dependencies on real word data. Their analysis is based on gate activation statistics and is thus global. On another side, [Li 2015] provided new visualization tools for recurrent models. They use decoders, t-SNE, and first derivative saliency, to shed light on how neural models work.

Although the usage of RNNs is more common, there are various visualization tools for CNNs analysis, inspired by the computer vision field. Such works may help us to highlight the linguistic features learned by a CNN. One can either train a decoder network or use backpropagation on the input instance to highlight its most relevant features. While those methods may hold accurate information in their input recovery, they have two main drawbacks: i) they are computationally expensive: the first method requires training a model for each latent representation, and the second relies on backpropagation for each submitted sentence. ii) they are highly hyperparameter dependent and may need some fine tuning depending on the task at hand. On the other hand, Deconvolution Networks, proposed by [Zeiler 2014], provide an off-the-shelf method to project a feature map in the input space. It

consists of inverting each convolutional layer iteratively, back to the input space. The inverse of a discrete convolution is computationally challenging. In response, a coarse approximation may be employed which consists of inverting channels and filter weights in a convolutional layer and then transposing their kernel matrix. More details of the deconvolution heuristic are provided in Section 12. Deconvolution holds several advantages. First, it induces minimal computational requirements compared to previous visualization methods. Also, it has been used with success for semantic segmentation on images: [Noh 2015] demonstrated the efficiency of deconvolution networks to predict segmentation masks to identify pixel-wise class labels. Thus deconvolution can localize meaningful structure in the input space.

11.2.3 Model Agnostic Explanation

Another line of works consists in explaining the decision, independently from the nature of the model itself. Such practices are denoted as Model Agnostic Explanation. For example, LIME [Ribeiro 2016] is a local approximation of a classifier's prediction that approximates the decision boundary around a sample by a hyperplane. Thanks to this choice of approximation, a greedy search can relatively well select the features that contribute the most to the prediction. LIME is a particular case of local approximation with a linear function. Linear functions hold two main advantages: when zooming enough, we can assume that the decision boundary is locally a linear separator, plus it allows to capture features of relative importance easily with greedy search, thanks to the induced submodularity. However, the features highlighted are representative of the local approximation, nor of the model itself. Moreover, a local explanation can be hardly extended to other sentences and does not provide rule of thumbs of how to combine the features to explain the decision. To mitigate such limitations, Ribeiro *et al.* have developed anchor explanations [Ribeiro 2018a]: "*if-then*" rules that are sufficient to explain the decision. The main advantage of anchors is that they apply when the conditions of the rule are met. Moreover, they explain the mechanism involved in the prediction. Listing all the possible anchors is intractable, but it is possible to look for short anchors (anchors with few items) but applicable to a broad set of sentences.

11.2.4 Adversarial example for NLP

While we have highlighted in Section 4, the potential benefits of adversarial examples in active learning, their outcome are mainly for vision applications. When it comes to NLP, generating adversarial examples is already a key challenge. Indeed, as opposed to images, or sound, where the features lie in a continuous space, words are discrete entities. Eventually, it is more difficult to measure and build perturbations into a discrete domain, while also preserving the semantics of the original sentence. Working on the word level and the embeddings used in our applications are not the sole part of the issue. Indeed, character level systems do suffer from adversarial examples: Ebrahimi *et al.* [Ebrahimi 2017], among others, show that networks trained

with characters are overly sensitive to keyboard typos, or unnatural dots or blank space in the sentence.

When it comes to word level system, adversarial perturbations have been designed for a broad panel of tasks, including spam filtering, fake news detection, or sentiment analysis, and also on both CNNs and RNNs. Kuleshov *et al.* designed adversarial attacks by iteratively replacing words by synonyms until it occurs a change of prediction [Kuleshov 2018]. Recently, Ribeiro *et al.* proposed a new system, called SEARS, to develop adversarial examples for NLP with logical rules to generate them [Ribeiro 2018b]. While previous works introduced ways to measure semantic similarity, none of them could detect unnatural sentence, nor create new sentences. SEARS use neural machine translation to generate paraphrase and combine it with semantic similarity. SEARS generates similar rules for a various type of applications, such as VQA and machine comprehension.

Chapter 12

Deconvolution for Text Analysis

Contents

12.1 Introduction **115**
12.2 CNNs for Text Classification **116**
12.3 Deconvolution **117**
12.4 Experiments **118**
12.5 Z-score Versus Activation-score **118**
12.5.1 Dataset: English 120
12.5.2 Dataset: French 121
12.5.3 Dataset: Latin 123
12.6 Conclusion **125**

12.1 Introduction

Summary

• We present our extension on deconvolution on CNN for text. Based on the patterns highlight by the deconvolution, we empirically demonstrate that CNNs encode complex semantic patterns based on co-ocurrences.
• We analyze the deconvolution saliency on three languages: English, French and Latin

✓ *Every dataset and parameters used to conduct our experiments are available in the* ***dataset*** *section A.1 and the* ***hyperparameter*** *section A.2.5*

lvanni/hyperdeep PROOF

We intend to exhibit classic linguistic patterns which are exploited implicitly in deep architectures to lead to higher performances. Do neural networks make use of co-occurrences and other standard features, considered in traditional Textual Data Analysis (TDA)? Do they also rely on some complementary linguistic structure which is invisible to traditional techniques? If so, projecting neural networks features

back onto the input space would highlight new linguistic structures and would lead to improving the analysis of a corpus and a better understanding of where the power of the Deep Learning techniques comes.

We hypothesize that Deep Learning is sensitive to the linguistic units on which the computation of the critical statistical sentences is based as well as to phenomena other than frequency and complex linguistic observables. The TDA has more difficulty taking such elements into account – such as linguistic patterns [Mellet 2009b]. Our contribution confronts TDA and Convolutional Neural Networks for text analysis. We take advantage of deconvolution networks for image analysis to present a new perspective on text analysis to the linguistic community that we call deconvolution saliency. Our deconvolution saliency corresponds to the sum over the word embedding of the deconvolution projection of a given feature map. Such score provides a heat-map of words in a sentence that highlights the pattern relevant to the classification decision. We examine z-scoring and deconvolution saliency in three languages: English, French, and Latin. For all our datasets, deconvolution saliency highlights new linguistic observables, invisible with z-scoring alone.

12.2 CNNs for Text Classification

We propose a deep neural model to capture linguistics patterns in text. This model is based on Convolutional Neural Networks with an embedding layer for word representations, one convolutional with pooling layer and non-linearities. Finally, we have two fully-connected layers. The final output size corresponds to the number of classes. The model is trained with cross-entropy with an Adam optimizer. Figure 12.1 shows the global structure of our architecture. The input is a sequence of words $w_1, w_2 ... w_n$ and the output contains class probabilities (for text classification).

The embedding is built on top of a Word2Vec architecture; here we consider a Skip-gram model. This embedding is also finetuned by the model to attain optimal text-classification accuracy. Notice that we do not use lemmatization, as in [Collobert 2008]. Thus the linguistic material which is automatically detected does not rely on any prior on the part of speech.

In computer vision, we consider images as 2-dimensional isotropic signals. A text representation may also be considered as a matrix: each word is embedded in a feature vector, and their concatenation builds a matrix. However, we cannot assume both dimensions - the sequence of words and their embedding representation - are isotropic. Thus the filters of CNNs for text typically differ from their counterparts designed for images. Consequently, in a text, the width of the filter is usually equal to the dimension of the embedding, as illustrated with the red, yellow, blue and green filters in Figure 12.1

Using CNNs hold another advantage in our context: due to the convolution operators involved, they can be easily parallelized and may also be easily used on CPU, which is a practical solution for avoiding the use of GPUs at test time.

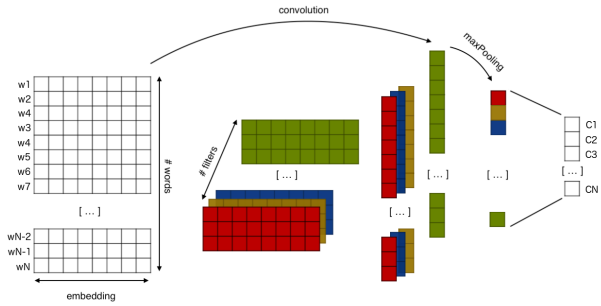


Figure 12.1: CNN model

12.3 Deconvolution

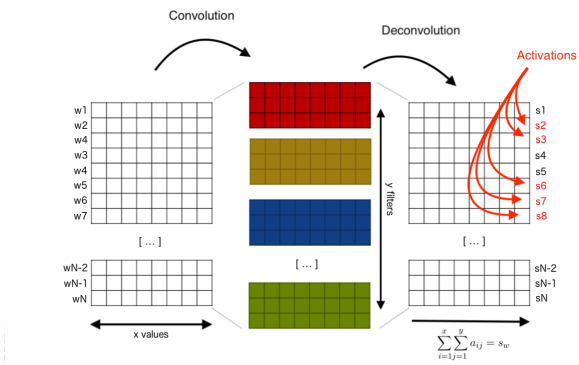


Figure 12.2: Deconvolution model

Extending Deconvolution Networks for text is not straightforward. Usually, in computer vision, the deconvolution is represented by a convolution whose weights depends on the filters of the CNN: we invert the weights of the channels and the filters and then transpose each kernel matrix. When considering deconvolution for text, transposing the kernel matrices is not realistic since we are dealing with nonisotropic dimensions - the word sequences and the filter dimension. Eventually, we do not transpose the kernel matrix.

Another drawback regards the dimension of the features map. We denote by features map; the output of the convolution before applying max pooling. Its shape is the tuple *(# words, # filters)*. Because the filters' width (red, yellow, blue and green in fig 12.1) matches the embedding dimension, the feature maps cannot contain this information. To project the feature map in the embedding space, we need to convolve our feature map with the kernel matrices. In that aim, we upsample the feature map to obtain a 3-dimensional sample of size *(# words, embedding dimension, # filters)*.

To analyze the relevance of a word in a sentence, we only keep one value per

	Z-score	Deep Learning
Latin	84%	93%
French	89%	91%
English	90%	97%

Figure 12.3: Prediction task acuray with Z-score and Deep Learning

words which corresponds to the sum along the embedding axis of the output of the deconvolution. We denote this score as deconvolution saliency.

For the sake of consistency, we sum up our method in Figure 12.2

With this method, we can show a sort of topology of a sequence of words. Eventually, every word in a sentence has a unique deconvolution saliency score whose value is related to the others. In the next section, we analyze the relevance of deconvolution saliency. We thoroughly demonstrate empirically, that the deconvolution saliency encodes complex linguistic patterns based on co-occurrences and possibly also on grammatical and syntaxic analysis.

Every dataset and parameters used to conduct our experiments are available in the **dataset** section A.1 the **hyperparameter** section A.2.5

12.4 Experiments

We conduct our experiments on three datasets, respectively in English, French and Latin.

12.5 Z-score Versus Activation-score

Z-score is one of the most used methods in linguistic statistics. It compares the observed frequency of a word with the frequency expected in the case of a "normal" distribution. This calculation readily gives, for example, the most specific vocabulary of a given author in a contrastive corpus. The highest Z-scores are the most specific words in this case. This is a simple but strong method for analyzing features of a text. It can also be used to classify word sequences according to the global Z-score (sum of the scores) in the sequence. The mean accuracy of this method on our data set is around 87%, which confirms Z-score is, in fact, meaningful on contrastive data. On the other hand, most of the time deep learning attains greater than 90% accuracy in text classification (As shown in Figure 12.3 – a benchmark of our three datasets detailed in the next subsections 12.5.1,12.5.2,12.5.3). The Z-test can be approximated by a normal distribution. The score we obtain by the Z-test is the standard deviation. A low standard deviation indicates that the data points tend to be close to the mean (the expected value). Over 2 this score means there is less than 2

This means that the training methods can also learn on their own some of the linguistic specificities useful in distinguishing between classes of text or authors. We've seen in work on images that this is the role of convolution. It learns an

abstraction of the data to make classification easier. The question is: what is the nature of this abstraction on text? We will see now that deep learning detects words automatically with high Z-score, but this is not the only linguistic structure identified.

To make the two values comparable, we normalize them with a maximum score of around +38 and a minimum of −38. This interval gives two thresholds for the Z-score: over 2 a word is considered as specific and over 5 it is strongly distinct (and the opposite with negative values). For the activation score, it is just a matter of activation strength.

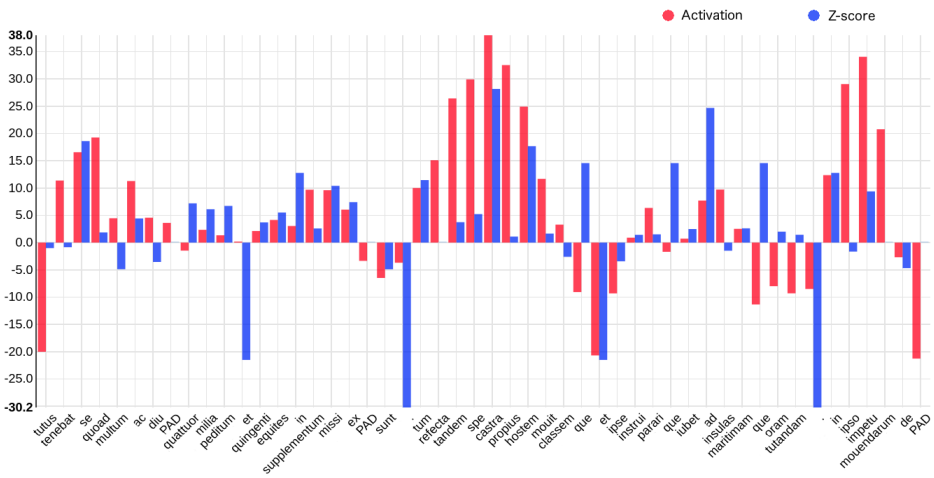


Figure 12.4: Z-score versus Activation-score

The Figure 12.4 shows us a comparison between Z-score and activation-score on a sequence extract from our Latin corpora (Livy Book XXIII Chap. 26). Here it is an example of specific word use by Livy[1]. As we can see, when the Z-score is the highest there is a sort of activation spike (for example around the word *castra*). However, this is not always the case: for example small words as *que*, *ad* and *et* are also high in Z-score, but they do not activate the network at the same level. We saw in (reference ****) that deep learning is more sensitive to long words, but we can see also on Figure 12.4 that words like *tenebat*, *multum* or *propius* are totally uncorrelated. The Pearson[2] correlation coefficient tells us that in this sequence there is no correlation between z-score and activation-score (with a Pearson of 0.38). This example is one of the most correlated examples of our dataset. Thus deep learning seems to learn more than a simple Z-score.

To understand what the real linguistic marks found by deep learning are (the convolution layer), we did several tests on different languages, and our model seems

[1]Titus Livius Patavinus – (64 or 59 BC-AD 12 or 17) – was a Roman historian.

[2]Pearson correlation coefficient measures the linear relationship between the two datasets. It has a value between +1 and −1, where 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative

to have the same behavior in all of them. We used a French web-platform called Hyperbase[3] to perform all the linguistic statistics tests.

12.5.1 Dataset: English

The first dataset we used for our experiments is the well known IMDB Movie review corpus for sentiment classification. It consists of 25,000 reviews labeled by positive or negative sentiment with around 230,000 words. With the default methods given by Hyperbase, we can easily show the specific vocabulary of each class (positive/negative), according to the Z-score. There are for example the words *too*, *bad*, *no* or *boring* as most indicitive of negative sentiment, and the words *and*, *performance*, *powerful* or *best* for positive. Is it enough to detect automatically if a new review is positive or not? Let's see an example excerpted from a review from December 2017 (not in the training set) on the last American blockbuster:

Quote 12.5.1: English Review

*[...] **i enjoyed three moments** in the film in total , **and if i am being honest and** the person **next to me fell asleep** in the middle and started snoring during the slow space chasescenes . **the story failed to** draw me in and entertain **me the way** [...]*

In general, the Z-score is enough to predict the class of this kind of comment. But in this case, deep learning seems to do better, but why? If we sum all the Z-scores (for negative and positive), the positive class obtains a greater score than the negative. The words *film*, *and*, *honest* and *entertain* – with scores 5.38, 12.23, 4 and 2.4 – make this example positive. Deep learning has activated different parts of this sequence (as we show in bold/red in the example). If we take the sub-sequence *and if i am being honest and*, there are two occurrences of *and* but the first one is followed by *if* and Hyperbase give us 0.84 for *and if* as a negative class. This is far from the 12.23 in the positive. And if we go further, we can do a co-occurrence analysis on *and if* on the training set. As we see on Figure 12.5, one of most specific adjectives[4] associated with *and if* is *honest*. Exactly what we found in our example.

In addition, we have the same behavior with the verb *fall*. There is *asleep* next to it. *asleep* alone is not really specific of negative review (Z-score of 1.13). But with the word *fall*, *asleep* becomes one of the most specific (see the co-occurrences analysis - Figure 12.6).

The activation-score here confirms that deep learning seems to focus not only on high Z-score but more complex patterns and maybe detects the lemma or the part of speech linked to each word. While the embedding is modifiable during the learning, it is possible that the final word vectors share this kind of information. We

[3]Hyperbase is an online (*http://hyperbase.unice.fr*) linguistic toolbox, which allows the creation of databases from a textual corpus and the performing of analysis and calculations such Z-score, co-occurrences, PCA, K-Means distance, ...

[4]With Hyperbase we can focus on different part of speech.

Figure 12.5: co-occurrences analysis of *and if* showed by Hyperbase. A layer shows the major co-occurrences for a given word (or lemma or PartOfSpeech). There two layers of cooccurrences, the first one (on top) show the direct co-occurrence and the second (on bottom) show a second level of co-occurrence. This level is given by the context of two words (taken together). The colors and the dotted lines are only used to make it more readable (dotted lines are used for the first level). The width of each line is related to the Z-test score (the bigger the Z-test, the wider the line).

will see now that these observations are still valid for other languages and can even be generalized between different activation spikes.

12.5.2 Dataset: French

The French dataset consists of political speeches. It is a corpus of 2.5 millions of words of French Presidents from 1958 (with C. de Gaulle, the first President of the Fifth Republic) to 2018 with the first speeches by Macron. In this corpus we removed Macron's speech from the 31st of December 2017, to use it as a test data set. In this speech, the deep learning network primarily recognizes E. Macron (the training task was to be able to predict the correct President). To achieve this task, the deep learning network seems to succeed in finding complex patterns specific to E. Macron. For example in this sequence :

Quote 12.5.2: French discourse

[...] notre pays ***advienne à*** *l'école pour nos enfants, au travail pour l' ensemble de* ***nos concitoyens*** *pour le climat pour le quotidien de chacune et chacun d' entre vous .* ***Ces transformations profondes*** *ont commencé et se* ***poursuivront*** *avec la même force le même rythme la même intensité [...]*

The Z-score gives a result statistically closer to de Gaulle than to E. Macron. The error in the statistical attribution can be explained by a Gaullist phraseology,

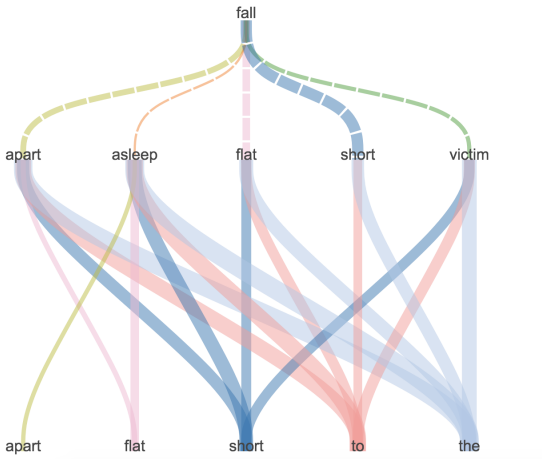


Figure 12.6: co-occurrences analysis of *fall* showed by Hyperbase

and the multiplication of linguistic markers strongly indexed with de Gaulle: for example, de Gaulle had the characteristic of making long and literary sentences articulated around conjunctions of coordination as in *et* (Z-score = 28 for de Gaulle, two occurrences in the excerpt). His speech was also more conceptual than average, and this resulted in an over-use of the articles defined *le*, *la*, *l´*, *les*) very numerous in the excerpt(7 occurrences); especially in the feminine singular (*la république*, *la liberté*, *la nation*, *la guerre*, etc., here we have *la même force*, *la même intensité*.

The best results given by deep learning themselves can surprise the linguist and match perfectly with what is known about the sociolinguistics of Macron's dynamic kind of speeches.

The most important activation zone of the excerpt concerns the nominal syntagm *transformations profondes*. Taken separately, neither of the phrase's two words are very Macronian from a statistical point of view (*transformations* = 1.9 *profondes* = 2.9). Better: the syntagm itself is not attested in the President's learning corpus (0 occurrences). However, it can be seen that the co-occurrence of *transformation* and *profondes* amount to 4.81 at Macron: so it is not the occurrence of one word alone, or the other, which is Macronian but the simultaneous appearance of both in the same window. The second and complementary activation zones of the excerpt thus concern the two verbs *advienne* and *poursuivront*. From a semantic point of view, the two verbs perfectly conspire, after the phrase *transformations profondes*, to give the necessary dynamic to a discourse that advocates change. But it is the verb tenses (borne by the morphology of the verbs) that appear to be the determining factor in the analysis. The calculation of the grammatical codes co-occurring with the word *transformations* thus indicates that the verbs in the subjunctive and the verbs in the future (and also the nouns) are the privileged codes for Macron (Figure 12.8).

More precisely the algorithm indicates that, for Macron, when *transformation*

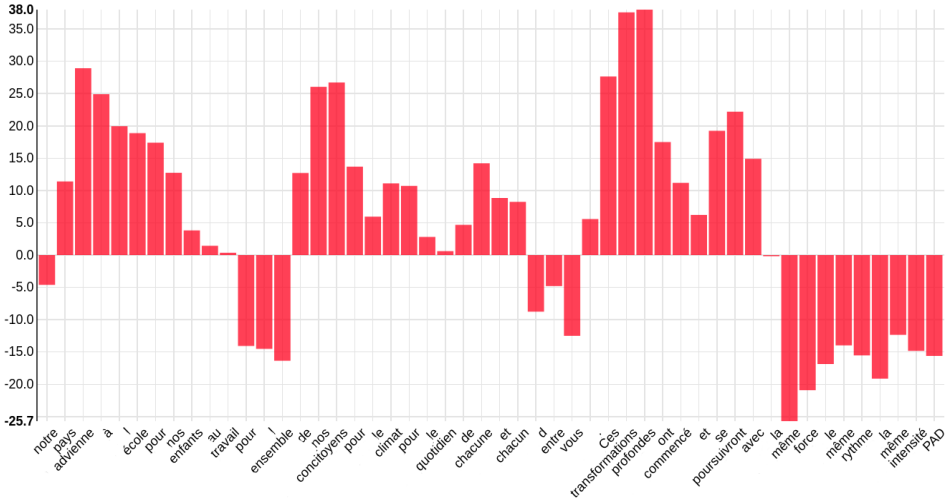


Figure 12.7: Deconvolution on E. Macron speech.

is associated with a verb in the subjunctive (here *advienne*), then there is usually a verb in the future co-present (here *poursuivront*). *transformations profondes*, *advienne* to the subjunctive, *poursuivront* to the future: all these elements together form a speech promising action. Finally, the graph indicates that *transformations* is especially associated with nouns in the President's speeches: in an extraordinary concentration, the excerpt lists 11 (*pays*, *école*, *enfants*, *travail*, *concitoyens*, *climat*, *quotidien*, *transformations*, *force*, *rythme*, *intensité*).

12.5.3 Dataset: Latin

The last dataset we used is in Latin. We assembled a contrastive corpus of 2 million words with 22 principle authors writing in classical Latin. As in the French dataset, the learning task here was to be able to predict each author according to new sequences of words. The next example is an excerpt of chapter 26 of the 23rd book of Livy:

Quote 12.5.3: Latin corpus

[...] tutus tenebat se quoad multum ac diu PAD quattuor milia peditum et quingenti equites in supplementum missi ex PAD sunt. tum refecta tandem spe ***castra propius hostem*** *mouit classem que et ipse instrui parari que iubet ad insulas maritimam que oram tutandam . in* ***ipso impetu*** *mouendarum de [...]*

The statistics here identify this sequence with Caesar[5] but Livy is not far off. As historians, Caesar and Livy share a number of specific words: for example tool

[5]Gaius Julius Caesar, 100 BC - 44 BC, usually called Julius Caesar, was a Roman politician and general and a notable author of Latin prose.

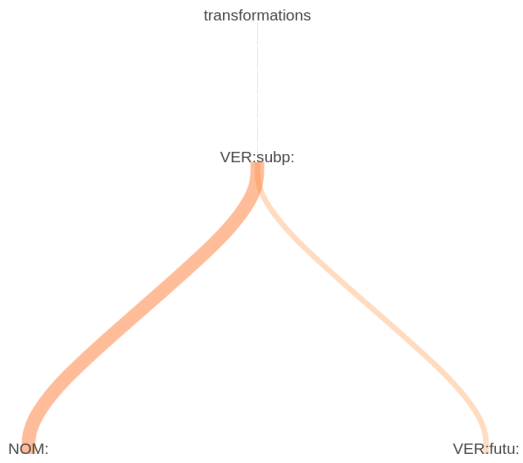


Figure 12.8: Main part-of-speech cooccurrences for *transformations* showed by Hyperbase

words like *se* (reflexive pronoun) or *que* (a coordinator) and prepositions like *in*, *ad*, *ex*, *of*. There are also names like *equites* (cavalry) or *castra* (fortified camp).

The attribution of the sentence to Caesar can not only rely only on Z-score: *que* or *in* or *castra*, with differences thereof equivalent or inferior to Livy. On the other hand, the differences of *se*, *ex*, are greater, as is that of *equites*. Two very Caesarian terms undoubtedly make the difference *iubet* (he orders) and *milia* (thousands).

The greater score of *quattuor* (four), *castra*, *hostem* (the enemy), *impetu* (the assault) in Livy are not enough to switch the attribution to this author.

On the other hand, deep learning activates several zones appearing at the beginning of sentences and corresponding to coherent syntactic structures (for Livy) – *Tandem reflexes spe castra propius hostem mouit* (then, hope returned, he moved the camp closer to the field of the enemy) – despite the fact that *castra* in *hostem mouit* is attested only by Tacitus[6].

There are also *in ipso metu* (in fear itself), while *in* followed by *metu* is counted one time with Caesar and one time also with Quinte-Curce[7].

More complex structures are possibly also detected by deeplearning: the structure *tum* + participates Ablative Absolute (*tum refecta*) is more characteristic of Livy (Z-score 3.3 with 8 occurrences) than of Caesar (Z-score 1.7 with 3 occurrences), even if it is even more specific of Tacitus (Z-score 4.2 with 10 occurrences).

Finally and more likely, the co-occurrence between *castra*, *hostem* and *impetu* may have played a major role: Figure 12.9

With Livy, *impetu* appears as a co-occurrent with the lemmas *HOSTIS* (Z-score

[6]Publius (or Gaius) Cornelius Tacitus, 56 BC - 120 BC, was a senator and a historian of the Roman Empire.

[7]Quintus Curtius Rufus was a Roman historian, probably of the 1st century, his only known and only surviving work being "Histories of Alexander the Great"

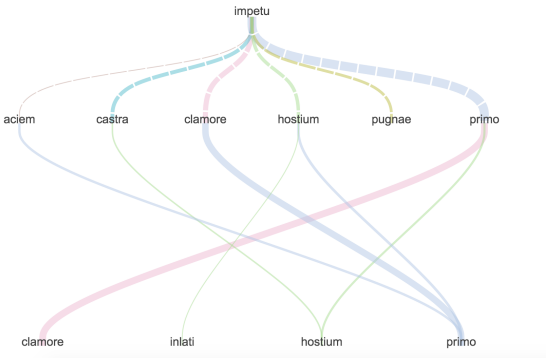


Figure 12.9: Specific co-occurrences between *impetu* and *castra* showed by Hyperbase.

9.42) and *CASTRA* (Z-score 6.75), while *HOSTIS* only has a gap of 3.41 in Caesar and that *CASTRA* does not appear in the list of co-occurrents.

For *castra*, the first co-occurent for Livy is *HOSTIS* (Z-score 22.72), before *CASTRA* (Z-score 10.18), *AD* (Z-score 10.85), *IN* (Z-score 8.21), *IMPETVS* (Z-score 7.35), *QUE* (Z-score 5.86)) while in Caesar, *IMPETVS* does not appear and the scores of all other lemmas are lower except *CASTRA* (Z-score 15.15), *HOSTIS* (8), *AD* (10,35), *IN* (5,17), *QUE* (4.79).

Thus, all is as it should be if the deep learning network manages to simultaneously account for specificity, phrase structure, and co-occurrence networks...

12.6 Conclusion

TDA and deep learning may not be distant continents to each other. This contribution by crossing a statistical approach and neural network allowed us to identify critical passages and perhaps reasons that could feed our textual treatments. If the observables that presided over the detection of key passages by the TDA (the lexical specificities) are known and tested, the zones of activation of the deep learning seem to raise new linguistic observables. Recall that the linguistic matter and the topology of the passages cannot return to chance: the zones of activations make it possible to obtain recognition rates of more than 90 % on the French political speech and 85 % on the corpus of the LASLA ; either rates equivalent to or higher than the rates obtained by the statistical calculation of the key passages. It remains to improve the model and to understand all the mathematical and linguistic outcomes. The first improvement that we now propose to implement is the injection of morphosyntactic information into the network to test more complex linguistic patterns ever.

Chapter 13

Perspective: Active Learning for Linguistic Analysis

Contents

13.1 Introduction 127
13.2 Methodology 128
13.3 Analysis under the light of Phraseology expertise 129
13.4 Future works: Analysis with Model Agnostic Explanations 131
13.5 Conclusion 131

13.1 Introduction

Summary

• We combine active learning heuristics with human expertise to highlight some hierarchy among the linguistic patterns learnt during the training of a CNN.

mducoffe/Active_NLP

Author identification and text genesis have always been a hot topic for the statistical analysis of textual data community. Recent advances in machine learning have seen the emergence of machines competing for state-of-the-art Computational linguistic methods on specific natural language processing tasks (part-of-speech tagging, chunking, and parsing, etc.). In particular, Deep Linguistic Architectures are based on the knowledge of language specificities such as grammar or semantic structure. These models are considered the most competitive thanks to their assumed ability to capture syntax. However, if those methods have proven their efficiency, their underlying mechanisms, both from a theoretical and an empirical analysis points of view, remains hard both to explicit and to maintain stable, which restricts their area of applications. Our work is enlightening mechanisms involved in deep architectures when applied to NLP tasks. Several post training methods have been

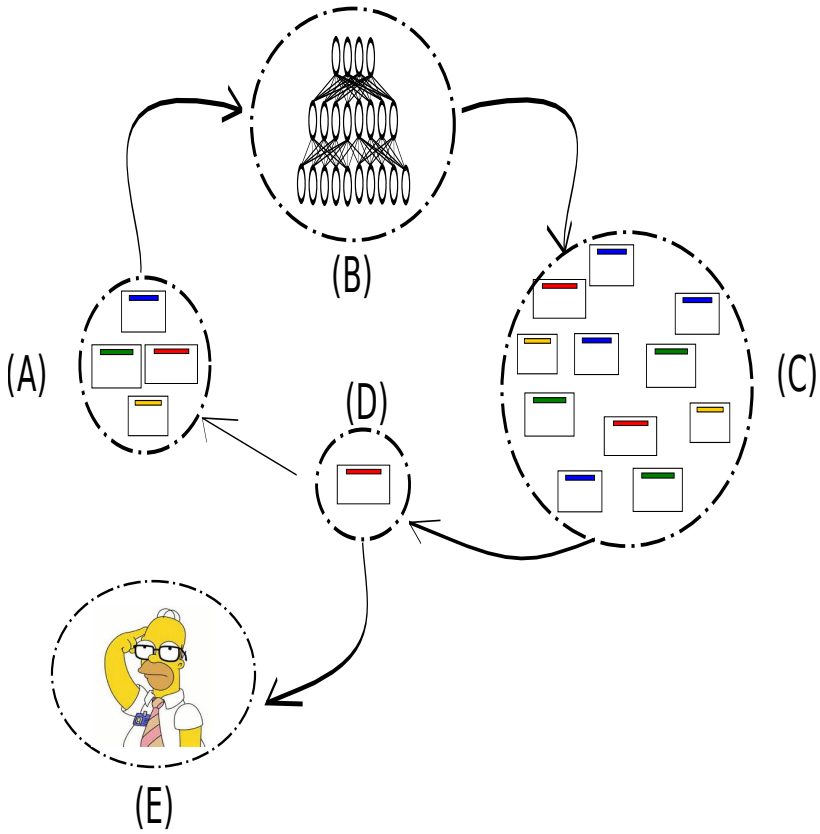


Figure 13.1: We train a network (**B**) on a labeled training set (**A**) made of sentences extracted from political discourses. When the network has converged , we query new sentences (**D**) among a pool of unlabelled samples (**C**), using **DQBC**. The queries are submitted to a human oracle (**E**) to be analyzed using phraseology techniques.

proposed to underline both semantic and syntactic patterns that have been necessary for the network's prediction [Vanni 2018, Li 2015, Karpathy 2015]. However, none of them have highlighted any hierarchy of the patterns learnt during the training. We instead confront the relevance of the sentences chosen by our active strategy to state-of-the-art phraseology techniques. In future works, we will also extend our analysis using model agnostic explanations.

13.2 Methodology

Despite the accuracy achieved, the complexity of deep networks on NLP tasks diminishes their interpretability. Up to now, few works have addressed what kind of knowledge deep networks are relying on: syntax, ontology, semantic or another non-linguistic intuitive information. In this Section, we consider the different type of information acquired by a network while training on specific tasks. We hypothesize that deep networks learn linguistic knowledge by step, to converge to their final

state which combines every semantic rule discovered previously. To have a glance at what kind of information the network is focusing on, we propose to select iteratively the samples which are the most helping the system to improve its accuracy. How to build an indicator function of such sentences is done through **AL**. Active learning is a particular case of learning when the model restricts its learning knowledge to a subset of the data and may gather more data in an online fashion: the model can interactively query new data and then adds them to the current set of training data. The main reason to be for active learning is the difficulty in gathering annotated data, especially when it requires experts. In our case, we are not considering **AL**heuristics to limit human annotations, but as an indicator function of the sentence required to extend the knowledge of a deep network. We illustrate our process in Figure 13.1.

The text analysis with our machine learning approach proceeds through active learning stages by selecting new samples at each iteration to be added to the training set. We compel this selection by a linguistic analysis, driven by linguistic experts, whose understanding helps to clarify which information is relevant in those queried sentences.

Every dataset and parameters used to conduct our experiments are available in the **dataset** section A.1 and the **hyperparameter** section A.2.1

Our method depends on the active learning heuristics used. However, while many active learning strategies coexist, none of them is optimal. Although the heuristic used will introduce some bias, as far as we obtain higher accuracy than random sampling, the queries hold relevant information for classification. However, since we are not interested in the accuracy achieved but on the underlying information hold by the queries, we would rather not use a batch active learning strategy.

13.3 Analysis under the light of Phraseology expertise

In previous works, we proposed an active learning method suitable for deep learning architectures. It is a query by committee based approach which consists in building a set of models trained on the same current labeled database and make each instance vote on the output of queried elements. Eventually, the score of an unlabelled sample is the disagreement it provokes among the members of the committee. Among such methods, **DQBC**[Ducoffe 2015] is an active learning method designed to build a committee of deep architectures with a low computation cost. Our approach is described in Section 3.

We illustrate our analysis on two French dataset extracted from political discourses of former French president. Our results are obtained by a network made of two dense layers whose takes as input a Word2Vec embedding:

- **De Gaulle / Hollande**: This classification tasks is leveraged by the evolution in the presidential discourses during the 80s. Indeed [Mayaffre 2012] have hilighted two main factors. First, the themes, but also the lexicon used in the discourses, have of course evolved during the Fifth Republic, so as the presidential discourse style. Moreover media coverage, which used to rely on

radio and is now essentially based on internet and television, impacts also on the discourses. On this dataset, our networks achieves 85 % of accuracy.

- **Hollande / Sarkozy**: This dataset is more challenging, firstly due that both presidents are contemporary, and also owing to the predominance of the crisis theme and the economic vocabulary in their discourses [Damon 2012]. Our network achieves 71% of accuracy.

We analyze three exercpts considered in early and late active learning stages. The linguistic observes that, in the first active learning phase, the selected sentences are indeed ambiguous for the linguist. For example, we find these two excerpts:

Quote 13.3.1: De Gaulle

C'est cela que les évènements m'ont amené à représenter à travers toutes les tempêtes ...

Quote 13.3.2: Hollande

Je transmettrai ma charge officielle à celui que vous aurez élu pour l'assumer après moi ...

In these two examples Quotes 1, and 2, one can foresee a contradictory linguistic characterization between the lexical level and the grammatical level. The lexical composition would be rather Gaullist with a typical vocabulary "*tempête*", "*êtes*", "*évènements*", "*assumer*", "*charge officielle*" The grammatical structure is rather associated to Hollande with the use of the first person ("*m*", "*je*", "*ma*", "*moi*") and a verbal tone (lots of verbs). In the end, at this stage, the analyst may therefore not be more sure of the paternity of these excerpts than the algorithm.

In the later active learning phase, illustrated by Quote 3, the selected sentences are gradually refined and disambiguated. After three active learning selection, for example, the algorithm remains indeterminate on the following excerpt:

Quote 13.3.3: De Gaulle

Cela dit, l'apparition de l'Algérie dans la situation d'un Etat indépendant coopérant organiquement avec la France ...

The analyst recognizes without difficulty the phraseology, the lexicon and the concerns of De Gaulle period (the issue of "*Algérie*" and "France", the nominal tone). However, we may assume that the introductory words "cela" and "dit" scramble the classification since they do not belong to the phraseology of De Gaulle.

13.4 Future works: Analysis with Model Agnostic Explanations

Future works will envisage systematic and automatic methods to highlight the information contained in the queries. The first tasks will consist in identifying the bias of active learning settings using model agnostic explanations: when reaching a certain accuracy, is the prediction based on the same observations or does it also depends on the choice of annotations. Eventually, model agnostic explanations, such as anchors, will help to express the rules given a set of queries.

13.5 Conclusion

Deep architectures have demonstrated a compelling potential for a better sampling of the target manifold [Bengio 2007] thanks to their *expressive power* [Bengio 2011]. However, the lack of comprehensive understanding (both on a theoretical or a practical aspect) of their underlying mechanisms hampers their broader application to difficult linguistic tasks. We made a step towards understanding the shared linguistic knowledge entailed in both machine and human analysis processes. Indeed, we analyzed the ability of deep learning approaches to cross the different levels of text granularity, vocabulary granularity, and morphosyntactic structure granularity, to encompass all the linguistic knowledge at once. Furthermore, we shed light on the persistent intricacy of the predictive process even for relatively simple classification task from a linguist's point of view.

Chapter 14

Conclusion

We derived existing techniques such as active learning and deconvolution to explain the decision of CNNs on a text. Although such methods have been thoroughly asserted on images dataset, extending them to text data was a challenging task. Indeed, working on Model Explanations for text classification requires the collaboration with linguistic experts, unlike natural image classification whom generally may be explained by any user. Explaining a pattern requires to clearly understand the data and the information held in a sentence, something that neither a non-native speaker (whom may have a poor comprehension of the meaning of a sentence itself), neither a native speaker may provide. Indeed the underlying mechanisms involved in our cognition when analyzing a sentence is still an open research field. Thus we focus our work in understanding the linguistic information kept by a CNN. We hope that our contributions will help in optimizing the design of new architectures and words embedding. Moreover, as a perspective, we would like to pursue future works in strengthening our visualization with automatic tools.

Part IV

Conclusion

14.1 Perspectives

Before concluding this Ph.D. thesis, we would like to underline how the different topics covered in this manuscript intertwine and how they bring perspectives that focus mainly on a better understanding of the mechanisms which are at the heart of the success of deep learning.

Firstly, **AL** methods have been at the core of this thesis, but we have been mainly focusing on reducing the annotations costs for image classification. Future works should also focus on its applications on new fields such as linguistic tasks. Indeed this would bring a broader view of which active learning criterions are efficient for deep networks, independently from the input space. Moreover, this study would open promising opportunities towards applying active learning for deep networks on structural data.

Furthermore, **AL** can be derived to illustrate the learning stage of deep models. If we thoroughly study the resulting patterns of this scenario, we could underline unknown learning strategies developed in the training of deep networks.

When it comes to the perspectives that will result from our contributions on the Wasserstein distance, they are threefold. Of course, we will investigate with more empirical experiments the impact of Wasserstein prototypes in increasing the diversity of active learning queries. But we also wish to dedicate their usage into the evaluation of GANs and transfer learning for deep networks. Indeed, Wasserstein prototypes may underline new statistics of the input space. On another side, the approximations of pairwise Wasserstein distance on text may create a new kind of adversarial attacks (see Section 8.4.2.1). Preventing the flaws underlined by those attacks will help to increase the robustness of our deep models.

Last but not least, our recent success in deriving adversarial attacks for **AL** opens up promising perspectives in the development of new regularizers for active learning on deep networks. Indeed, recent works on robustness to adversarial attacks derived networks to threshold the adversarial attacks [Dvijotham 2018]. Combining their approach with our method could potentially lead to fasten the convergence in terms of annotations.

14.2 Conclusion

The success of deep networks is well established. However, supervised learning requires a large amount of labeled data. A highly relevant question is how to gather this training set. Optimizing its construction and avoiding biases are two main issues. This Ph.D. thesis contributed to this problematic by focusing on two main directions:

Firstly, we reduced the size of the training set by using active learning. Active learning consists in building the training set iteratively. Given a set of unlabeled examples and a deep network, active learning queries the data to be labeled that will improve at best the accuracy. Here the main issue is which data should be labeled to ensure the sparsest training set as possible. Active learning is considered

to be one of the pillars of machine learning, although it has not been the subject of further study on one of the greatest successes of machine learning: deep neural networks. If few algorithms have been interested in the question, it is because we could not scale existing algorithms to deep networks. As a result, we could not use active learning methods on deep networks without increasing the computation time drastically. Our work contributes to new active learning heuristics that scale for CNNs. In that context, we proposed several methods ranging from adapting existing active learning algorithms such as query-by-committee, to create inedite methods that take into account the properties inherent with deep networks. Our contributions open up promising directions towards using active learning in the development of academic or industrial applications.

Secondly, another problem remains; the biases in the training set. These biases are highly relevant as they may alter the predictions of the network, but also comfort the users in their take of decisions. Our solution is to visualize the data with a subset of prototypes. One of the main challenges is to decide what kind of information should be illustrated by those prototypes. With this in mind, we focused on the Wasserstein distance. The Wasserstein distance is used to measure the differences between the two distributions. In our case, it is the distribution induced by the learning data, and the one made up of prototypes. Our work also focused on combining Wasserstein distances and neural networks so to speed up the computation of Wasserstein distance. Such speed up will help to spread the usage of Wasserstein into new applications.

In a nutshell, we extend active learning to deep networks. We also design new tools to improve human understanding of complex data and their predictions, particularly for linguistic tasks. We hope that our contributions will highlight new solutions towards a more robust, explainable and less greedy AI.

List of Figures

2.1 **Illustration of Pool-Based Active Learning**: **A**=Labeled training set ; **B**=Classifier ; **C**=Unlabeled set ; **D**=Queries ;**E**=Oracle . . 9

2.2 "*An active learning heuristics for SVM* SVM_{active} *will query data that are the closest to the decision boundary. In that tendency, Query synthesis may help to generate queries that optimize the active learning criterion*" as underlined in 2.2(b) [Zhu 2017] 10

2.3 **AL on a toy data-set**. We consider a binary classification tasks on n=100 samples (• positive examples, • negative examples) and want to learn an optimal classifier with no bias. Because the data are well separated, there exist an optimal linear classifier (**W**) that can be learnt using at minima four samples: by labelling (**A**, **B**, **C**, **D**). However, the probability of sampling those points at once is of $\frac{1}{\binom{n}{4}} \simeq 10^{-4}$ 11

2.4 **Version space example for linear classifiers on a binary task.** Every hypothesis (—) is consistent with the labeled training set. However, each of them represent a different model in the version space. The unlabeled sample in red is not relevant as a query because every classifier agree in its prediction (*it is a dog*, whereas the green unlabeled sample is interesting as it will shrink the number of consistent classifiers. The more classifiers we discard, the faster QBC converge towards the optimal classifier for the task at hand (*assuming that a linear classifier can solve exactly the problem at hand*) 14

2.5 Consider the network's predictions of both the labeled training points **s** (•) and the unlabeled points (•). Sener *et al.* shows that if the labeled training set covers the space by a distance of at most $\delta_{\mathbf{s}}$ (*as illustrated in 2.5(a)*) then the *core-set loss* is bounded by $\mathcal{O}(\delta_{\mathbf{s}}) + \mathcal{O}(\frac{1}{n})$ with n denoting the number of points available. Eventually the strategy induced by this property implies to query unlabeled points that minimize at best the expected $\delta_{\mathbf{s}}$. To do so, Sener *et al.* developed a MIP strategy to select the samples that cover at best the output space of the network (*as illustrated in 2.5(b)*). 16

3.1 applying batchwise dropout to build a *partial* CNN from the *full* CNN 27

3.2 **DQBC(qbdc) with top score selection**: Evolution of the test error given the ratio of annotated data over the training set. 29

4.1 *"An adversarial input, overlaid on a typical image, can cause a classifier to miscategorize a panda as a gibbon."* [Goodfellow 2015] 32

4.2 Illustration of different margin-based active learning scenarios in the binary case 35

4.3 Evolution of the test accuracy achieved by 7 active learning techniques on *MNIST* and *Shoe-Bag* given the number of annotations. We denote by **DFAL_** our active learning method when not adding the adversarial examples. We use a log scale in the x-axis and a linear scale in the y-axis. 39

4.4 Evolution of the test accuracy achieved by 7 active learning techniques on *Quick-Draw* given the number of annotations. We denote by **DFAL_** our active learning method when not adding the adversarial examples. We use a log scale in the x-axis and a linear scale in the y-axis. 39

4.5 Evolution of the test accuracy achieved by 3 active learning techniques given the number of annotations 41

4.6 Evolution of the test accuracy for (*Shoe-Bag*, VGG8) trained with different labeled training set: we compare the efficiency of **DFAL** and CORE-SET built on LeNet5 (LeNet5_**DFAL** and LeNet5_CORE-SET) and transfered to VGG8. 41

4.7 **Overlap of the classification regions of LeNet5 and VGG8 trained on the QuickDraw datasets**. Blue dots • are test samples that fall into the same classification regions for both networks, while red dots • do not fall into the same region. We proceed by looking for a convex path so that every point in that path share the same prediction. To do so, we check the validity of the path in the convex combinations of consecutive anchor points, as proposed by Fawzi *et al.* [Fawzi 2017]. Then we check, whether paths exist for both networks and project the test samples in a two-dimensional space using T-SNE [Maaten 2008]. 43

4.8 range of adversarial perturbations (i.e. distances between samples and their adversary) for VGG8trained on *MNIST*with 10, 20, 30 ... to 100 labeled examples. A curve corresponds to the range of adversarial pertubation found on the unlabeled example, while its color matches the size of the labeled set used to train the network 45

4.9 Toy problem: learning a linear separator that predicts with no error the labels of positive instances •, and negative instances •. We illustrate the notion of weak adversarial examples • on two samples x^1 and x^2. 47

4.10 Repartition of weak examples • for samples • lying in the low confidence subregion of a consistent classifier $\mathcal{C}$ 50

5.1 Illustration of our active learning criterion for linear separators (*see* Eq. 5.11): $\mathcal{A}_t$ ={ • }, $\mathcal{B}$ ={ • }, $\mathcal{P}\backslash\{\mathcal{A}\cup\mathcal{B}\}$ ={ • }. We learn a logistic classifier with no bias on a 2D binary classification task. The data are made of a mixture of two Gaussians (on the left lie positive examples and the right negative examples). The yellow points $\mathcal{A}_t$ are labeled, and $\mathbf{w_t}$ is learn on them with no error. We then select 9 unlabelled data $\mathcal{B}$ to minimize our variational free energy: the set of points $\mathcal{B}$ whose induced posterior distribution $Q(\mathbf{w}_t \mid \beta)$ will diverge at most from the prior distribution $P(\mathbf{w}_t \mid \alpha)$. $\mathbf{w_{t+1}}$ is the classifier trained on $\mathcal{A}_t \cup \mathcal{B}$. 53

5.2 **Empirical evaluation of A-optimality on CNNs**: Test error achieved for different percentage of the whole training set. We use *MNIST*5.2(a) and *SVHN*5.2(b). We compare A-optimality (*by approximating the Fisher matrix with KFAC, see Section 5.3*) against random selection and uncertainty selection [Ducoffe 2016b]. We consider 2 CNNs whose hyperparameters are fully described in Appendix A.2.3. 57

5.3 **Illustration of KFAC approximation on VQA network [Antol 2015]**. VQA stands for Visual Question Answering. It is a multimodal classification task that involves a lot of handcrafted descriptions to build the training set. Consequently, active learning is perfectly suited for VQA. While it requires a lot of labeled training data, it is likely than just feeding it with random data will not be enough to improve its performance, as first assumed in [Lin 2017b]: "*As a result with long tail distributions, it will likely result in redundant questions and answers while still having insufficient training data for rare concepts*" 60

5.4 Evolution of the test accuracy achieved by 7 active learning techniques on *MNIST* and *Quick-Draw* given the number of annotations. We denote our method by **FISHER** 63

8.1 Architecture of the Wasserstein Deep Learning: two samples are drawn from the data distribution and set as input of the same network (ϕ) that computes the embedding. The embedding is learnt such that the squared Euclidean distance in the embedding mimics the Wasserstein distance. The embedded representation of the data is then decoded with a different network (ψ), trained with a Kullback-Leibler divergence loss. 79

8.2 Prediction performance on the *MNIST*dataset. (Figure) The test performances are as follows: MSE=0.40, Relative MSE=0.002 and Correlation=0.996. (Table) Computational performance of W_2^2 and **DWE** given as average number of W_2^2 computation per seconds for different configurations. 81

8.3 MSE of the validation test given the number of epochs (**DWE**). . . . 82

8.4 Barycenter estimation on each class of the *MNIST* dataset for squared Euclidean distance (L2) and Wasserstein Deep Learning (**DWE**). . . 83
8.5 Interpolation between four samples of each datasets using **DWE**. (left) cat dataset, (center) Crab dataset (right) Face dataset. 84
8.6 Comparison of the interpolation with L2 Euclidean distance (top), LP Wasserstein interpolation (top middle) regularized Wasserstein Barycenter (down middle) and **DWE** (down). 85
8.7 Principal Geodesic Analysis for classes 0,1 and 4 from the *MNIST* dataset for squared Euclidean distance (L2) and Wasserstein Deep Learning (**DWE**). For each class and method we show the variation from the barycenter along one of the first 3 principal modes of variation. . . . 86
8.8 Illustration of WMD. Words are embedded with Word2Vec. The distance between the two sentences is the minimum cumulative distance that all words from the first sentence need to *travel* to be transformed into the target sentence. 87
8.9 Two sentences x_1 and x_2 are sampled from the data distribution. Each of their words are encoded by a first network ψ whose input size matches the dimension of the words' embedding. ψ outputs a vector for each word which are then summed together into a unique vector. This vector is finally encoded by a second network τ thats computes the embedding that mimics the Wasserstein distance. . . . 89
8.10 Deep Reccurent Wasserstein Embedding for Text 91

9.1 *Circle:* Visualization of prototypes selected given different criteration with p=2 99
9.2 *3 univariate Gaussians:* Visualization of prototypes selected given different criteration with p=2 100
9.3 *3 Gaussians with different ratio:* Visualization of prototypes selected given different criteration with p=2 101
9.4 *3 Gaussians with different ratio:* Visualization of prototypes selected given different criteration with p=2 102
9.5 Comparison of **BalNet**[Ducoffe 2016c] with (–) and without (–) using Wasserstein prototypes as a post processing step to select the queries (with p=2 for the Wasserstein distance). We measure the test accuracy on *MNIST* trained on LeNet5 given the number of annotations. 103

11.1 **Adversarial Questions for state-of-the-art VQA systems [Ribeiro 2018b]**. The authors paraphrase the questions which highly impacts the quality of the answer. 110

12.1 CNN model 117
12.2 Deconvolution model 117
12.3 Prediction task acuray with Z-score and Deep Learning 118

12.4 Z-score versus Activation-score 119
12.5 co-occurrences analysis of *and if* showed by Hyperbase. A layer shows the major co-occurrences for a given word (or lemma or PartOf-Speech). There two layers of cooccurrences, the first one (on top) show the direct co-occurrence and the second (on bottom) show a second level of co-occurrence. This level is given by the context of two words (taken together). The colors and the dotted lines are only used to make it more readable (dotted lines are used for the first level). The width of each line is related to the Z-test score (the bigger the Z-test, the wider the line). 121
12.6 co-occurrences analysis of *fall* showed by Hyperbase 122
12.7 Deconvolution on E. Macron speech. 123
12.8 Main part-of-speech cooccurrences for *transformations* showed by Hyperbase 124
12.9 Specific co-occurrences between *impetu* and *castra* showed by Hyperbase. 125

13.1 We train a network (**B**) on a labeled training set (**A**) made of sentences extracted from political discourses. When the network has converged , we query new sentences (**D**) among a pool of unlabelled samples (**C**), using **DQBC**. The queries are submitted to a human oracle (**E**) to be analyzed using phraseology techniques. 128

List of Tables

3.1 Example of Sample Selection. The icon outputs by a network represents its most probable label on the unlabeled sample. Samples (A) and (B) are both unlabeled examples. Sample (A) is more appealing as it will necessarily reduce the version space by a ratio of 4. Whereas Sample (B) will reduce at worst the version space by a ratio of 2.5. On another side, if one has complementary information about the networks, such as the confidence of each network in their prediction, sample (B) may become a better choice. 28

4.1 Summary of the datasets used to evaluate **DFAL**. 36
4.2 Number of annotations to achieve the same test accuracy on LeNet5 and VGG8 as the accuracy obtained on the full training set (BASELINE, $\pm$ 0.5 %). 38
4.3 Average runtime of **DFAL** and CORE-SET on *MNIST*. We denote by $\mathcal{L}$ the labeled training set, and $\mathcal{U}$ the unlabeled set of data; $n_{query} = 10$ 40
4.4 Comparison of the transferability of **DFAL** and CORE-SET with 1000 annotations 42

8.1 Cross performance between the **DWE** embedding learned on each datasets. On each row, we observe the MSE of a given dataset obtained on the deep network learned on the four datasets (Cat, Crab, Faces and MNIST). 82
8.2 Mean Squared Error (MSE) and Relative Mean Squared Error obtained on an independent test set, respectively on the *Twitter*and Visual Question Answeringdataset. We can see that **DWE** is actually learning the exact Wasserstein distance. Although, those results should be leveraged by the RMSE obtained when predicting the mean Wasserstein distance between two random sentences of *Twitter* (0.012) and Visual Question Answering (0.036). 90

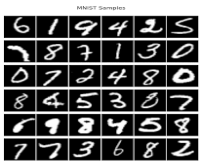
A.1 Set of hyperparameters for the CNN used respectively for *MNIST*and *USPS* . . 151
A.2 LeNet5 151
A.3 VGG8 152

Appendix A

Appendix

A.1 Dataset

Dataset 1.1.1: *MNIST*



28x28 grayscale images from 10 digits classes. The training and test set contains respectively 60,000 and 10,000 samples.

Dataset 1.1.2: *USPS*

16x16 grayscale images from 10 digits classes. The dataset has 7291 train and 2007 test images.

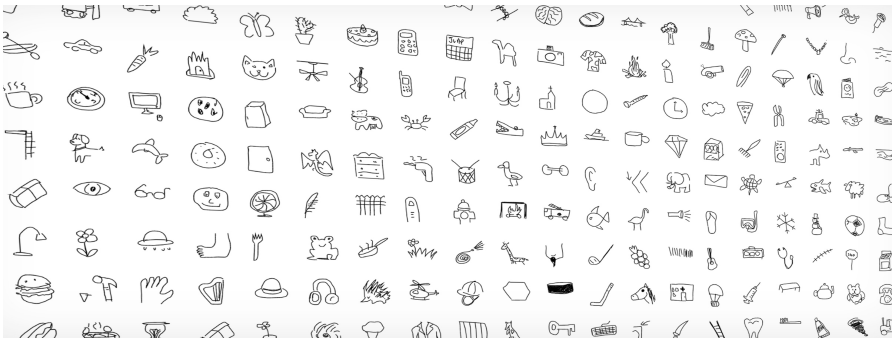
Dataset 1.1.3: *SVHN*

32x32 RGB images from 10 digits classes. The images are centered around a single character (many of the images do contain some distractors at the sides). The dataset contains 73257 digits for training, and 26032 digits for testing.

Dataset 1.1.4: *CIFAR10*

32x32 RGB images from 10 object classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck), with 6000 images per class. There are 50,000 training images and 10,000 test images.

Dataset 1.1.5: *Quick-Draw*



The Google Doodle dataset is a crowd sourced dataset that is freely available from the web [a] and contains 50 million drawings. The data has been collected by asking users to hand draw with a mouse a given object or animal in less than 20 seconds. This lead to a large number of examples for each class but also a lot of noise in the sens that people often get stopped before the end of their drawing .We used the numpy bitmaps format proposed on the quick draw github account. Those are made of the simplified drawings rendered into 28x28 grayscale images. These images are aligned to the center of the drawing's bounding box.

[a] https://quickdraw.withgoogle.com/data

Dataset 1.1.6: *Shoe-Bag*

[Huijser 2017] create the Shoe-Bag dataset of 40,000 train and 14,000 test images by taking subsets from the Handbags dataset [Zhu 2016] and the Shoes dataset [Yu 2014]. They contain tiny RGB images of size 28x28.

Dataset 1.1.7: *Cats & Dogs*

150x150 RGB images of dogs and cats. There are 2000 training images and 2000 testing images sampled from the cats and dogs database which we got from kaggle.

Dataset 1.1.8: *Twitter*

We use the set of tweets labeled with sentiments positive, negative, or neutral [Sanders 2011] (the set is reduced due to the unavailability of some tweets). We use a word2vec embedding trained on the Google News corpus

(*source: github.com/mkusner/wmd*)

Dataset 1.1.9: Visual Question Answering

VQA is a new dataset containing open-ended questions about images. These questions require an understanding of vision, language and commonsense knowledge to answer.
• 265,016 images (COCO and abstract scenes)
• At least 3 questions (5.4 questions on average) per image
• 10 ground-truth answers per question
• 3 plausible (but likely incorrect) answers per question

We only work on the questions that have been preprocessed with a Glove embedding at *github/anantzoid/VQA-Keras-Visual-Question-Answering*

A.2 Hyperparameters

In order to make our experiments reproducible, we detail here all the hyperparameters used.

A.2.1 Dropout Query-By-Committee

dataset	# filters	filter size	pooling size	hidden layers	Test error
MNIST	[20, 20]	[(3,3), (3,3)]	[(2,2), (2,2)]	[200, 200, 50, 10]	1.1
USPS	[20, 20]	[(3,3), (3,3)]	[None, (2,2)]	[300, 50, 10]	3.25

Table A.1: Set of hyperparameters for the CNN used respectively for *MNIST* and *USPS*

A.2.2 Adversarial Active Learning

Layer	**Parameters**
Conv	6 - (5,5) - ReLU
MaxPooling2D	(2,2)
Dropout	0.25
Conv MaxPooling2D	120 - (3,3) -ReLU (2,2)
Dropout	0.25
Dense	84 - ReLU
Dense	10 - ReLU
Dropout	0.5
Dense	10 - Softmax

Table A.2: LeNet5

Layer	Parameters
Zero Padding	(1,1)
Conv	64 - (3,3) - ReLU
MaxPooling2D	(2,2)
Dropout	0.25
Conv	120 - (3,3) -ReLU
Zero Padding	(1,1)
Conv	128 - (3,3) - ReLU
Zero Padding	(1,1)
Conv	256 - (3,3) - ReLU
Zero Padding	(1,1)
Conv	256 - (3,3) - ReLU
MaxPooling2D	(2,2)
Dropout	0.25
Dense	4096 - ReLU
Dropout	0.5
Dense	10 - Softmax

Table A.3: VGG8

A.2.3 Bayesian Active Learning through Laplace Approximation

A.2.4 Learning Wasserstein embeddings

Architecture for DWE between grayscale images The framework of our approach consists of an encoder ϕ and a decoder ψ composed as a cascade. The encoder produces the representation of input images $h = \phi(x)$. The architecture used for the embedding ϕ consists in 2 convolutional layers with ReLU activations: first a convolutional layer of 20 filters with a kernel of size 3 by 3, then a convolutional layer of 5 filters of size 5 by 5. The convolutional layers are followed by two linear dense layers respectively of size 100 and the final layer of size $p = 50$. The architecture for the reconstruction ψ consists in a dense layer of output 100 with ReLU activation, followed by a dense layer of output 5*784. We reshape the layer to map the input of a convolutional layer: the output vector is (5,28,28) 3D-tensor. Eventually, we invert the convolutional layers of ϕ with two convolutional layers: first a convolutional layer of 20 filters with ReLU activation and a kernel of size 5 by 5, followed by a second layer with 1 filter, with a kernel of size 3 by 3. Eventually the decoder outputs a reconstruction image of shape 28 by 28. In this work, we only consider grayscale images, that are normalized to represent probability distributions. Hence each image is depicted as an histogram. In order to normalize the decoder reconstruction we use a softmax activation for the last layer.

CNN for Text Mining The framework of our approach consists of an network ϕ that outpouts a vector for each word'embedding. We aggregate the representation

into a unique vector by summing all the outputs along a sentence. ψ consists in a dense layer of input size, the dimension of the embedding (=300) and output 100 with ReLU activation and no bias. In a second step we use a network ϕ whose output will mimic the Wasserstein distance. ϕ is made of two denses layers of 100 units with Relu activation. We use Adam and batch size 32.

RNN for Text Mining Our framework is made of a LSTM with 50 units, trained with Adam and Tanh activation.

A.2.5 Deconvolution for Text Analysis

The neural network is written in python with the library Keras (and Tensorflow as backend). The embedding uses a Word2Vec implementation given by the gensim Library. Here we use the SkipGram model with a window size of 10 words and output vectors of 128 values (embedding dimension). The textual data are tokenized by a home-made tokensizer (which work on English, Latin and French). The corpus is split into 50 length sequence of words (punctuation is kept) and each word is converted into a vector of size 128. The first layer of our model takes the text sequence (as word vectors) and applies a weight corresponding to our WordToVec values. Those weights are still trainable during model training. The second layer is the convolution, a Conv2D in Keras with 512 filters of size 3128 (filtering three words at a time), with a Relu activation method. Then, there is the Maxpooling (MaxPooling2D). (The deconvolution model is identical until here. We replace the rest of the classification model (Dense) by a transposed convolution (Conv2DTranspose).) The last layers of the model are Dense layers. One hidden layer of 100 neurons with a Relu activation and one final layer of size equal to the number of classes with a softmax activation. All experiments in this paper share the same architecture and the same hyperparameters, and are trained with a cross-entropy method (with an Adam optimizer) with 90% of the dataset for the training data and 10% for the validation. All the tests are done with new data not included in the training dataset.

A.3 Proofs

A.3.1 Adversarial Active Learning

A.3.1.1 Proof of Theorem 6.1

♠ We consider samples from the unit ball from a binary task. The dataset is centered around the origin We also assume that the task at hand is linearly separable by a normalized linear classifier going through the origin. Given i, j the mean of respectively positive and negative points in X; we can deduce the nearest centroid classifier $\mathcal{B}$ with unit vector $\mathbf{b}$ and bias b_0: the bissecting hyperplane separating at best i and j. Notice that $\mathcal{B}$ is not necessarily minimizing the error. However, necessarily, $\mathcal{B}$ predicts i as positive and j as negative.

Since the problem is consistent with linear classifiers without bias, we denote by $\mathcal{L}$ the set of optimal classifiers of norm 1 and going through the origin: $\mathcal{L} = \{\mathcal{W} \mid \forall x \in X,\ y(x)\langle w, x\rangle > 0\}$. Also among those classifiers, we call *weak classifier* $\mathcal{W}$, the classifier minimizing the error with the largest deviation angle given $\mathbf{b}$, in accordance with the Definition 2. Moreover, we denote *strong classifier* $\mathcal{S}$, the classifier minimizing the error with the smallest deviation angle given $\mathbf{b}$.

Firstly, we demonstrate that what we call the *deviation angle* (the angle between a classifier in $\mathcal{L}$ and $\mathcal{B}$) lies in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$, as detailed in Lemma 3.1.

Lemma 3.1: Deviation Angle

$\forall \mathcal{C} \in \mathcal{L}$ we can express its unit vector $\mathbf{c}$ given the unit vector $\mathbf{b}$ of the bissecting angle and a unit vector $\mathbf{b}_c^{\perp}$:

$$\mathbf{c} = cos(\delta_c)\mathbf{b} + sin(\delta_c)\mathbf{b}^{\perp}$$

Thus we obtain $cos(\delta_c) \geq 0$ which implies that the deviation angle lies in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$.

♣ By linearity any optimal classifier predicts i as positive and j as negative: $\mathbf{c} \cdot i \geq 0$, $\mathbf{c} \cdot j \leq 0$. Also, since $\mathcal{B}$ is the bissecting hyperplane, we can express i, and j using $\mathbf{b}$ and the signed distance to the hyperplane $d(\cdot, \mathcal{B})$. Note that since $\mathcal{B}$ predicts i as positive we have $d(i, \mathcal{B}) > 0$.

$$j = i - 2d(i, \mathcal{B})\mathbf{b} \tag{A.1}$$
$$i = j - 2d(j, \mathcal{B})\mathbf{b} \tag{A.2}$$
$$\tag{A.3}$$

Thus, for the orthogonal vector $\mathbf{b}^{\perp}$ to $\mathbf{b}$, we have : $(i \cdot \mathbf{b}^{\perp}) = (j \cdot \mathbf{b}^{\perp})$. Eventually we can express the dot product $c \cdot (i - j)$, which is positive, using $\mathbf{b}$ and $\mathbf{b}_c^{\perp}$.

$$\mathbf{c} \cdot (i - j) = 2d(i, \mathbf{B})\mathbf{c} \cdot \mathbf{b} \tag{A.4}$$
$$= 2d(i, \mathbf{B})(cos(\delta_c)\mathbf{b} \cdot \mathbf{b} + sin(\delta_c)\mathbf{b}_c^{\perp} \cdot \mathbf{b}) \tag{A.5}$$
$$= 2d(i, \mathbf{B})cos(\delta_c) \tag{A.6}$$

Finally, equation A.3.1.1 implies $cos(\delta_c) \geq 0$

Weak adversarial attack: We prove it by contradiction. Assume $\exists\, x$ s.t $\exists\, \mathcal{C} \in \mathcal{L}$ which wrongly predicts x_w. Thus it implies $(\mathbf{c} \cdot x_w)(\mathbf{c} \cdot x) < 0$ Without loss of information, we assume that the *weak classifier* is the closest boundary to x. Thus $x_w = x - (\mathbf{w} \cdot x)\mathbf{w}$.

$$(\mathbf{c} \cdot x_w)(\mathbf{c} \cdot x) < 0$$
$$(\mathbf{c} \cdot x)(\mathbf{c} \cdot x) - (x \cdot \mathbf{w})(\mathbf{c} \cdot \mathbf{w})(\mathbf{c} \cdot x) < 0$$
$$(\mathbf{c} \cdot x)^2 < [cos(\delta_w)cos(\delta_c) + sin(\delta_w)sin(\delta_c)](x \cdot \mathbf{w})(\mathbf{c} \cdot x)$$
$$(\mathbf{c} \cdot x)^2 < cos(\delta_w - \delta_c)(x \cdot \mathbf{w})(\mathbf{c} \cdot x)$$

Because $\mathcal{W}$ and $\mathcal{C}$predict the same label

for x we obtain a necessary condition:

$$cos(\delta_w - \delta_c) \geq \frac{(\mathbf{c} \cdot x)^2}{(x \cdot \mathbf{w})(\mathbf{c} \cdot x)}$$
$$cos(\delta_w - \delta_c) \geq \frac{\mathbf{c} \cdot x}{x \cdot \mathbf{w}}$$

Because we picked **w** instead of **s** as **w** minimizes its distance with the sample x, then $\frac{\mathbf{c} \cdot x}{x \cdot \mathbf{w}} > 1$ which contradicts the previous inequality. Thus any other classifier than $\mathcal{W}$ will predict the same label for both x, x_w. When it comes to $\mathcal{W}$, x_w lies on the boundary, thus it can be assumed to share the same label.

A.3.1.2 Proof of Theorem 6.2

♠ Theorem 6.2 results from the number of successes from a Bernouilli law. If we assume that the probability of misclassification of $\mathcal{W}$ and $\mathcal{S}$ are independent, then we have probability $\frac{p_k}{2}$ to be able to build a query for an unlabeled sample at step k. In this case, we will add two samples to the training set instead of one. Consequently, Theorem 6.2 relies on Lemma 6.1.

Lemma 6.1 Consider a threshold α so that $|\mathbf{c} \cdot x| \leq \alpha$. Without loss of information, we assume that the *weak classifier* is the closest boundary to x. Thus $x_w = x - (\mathbf{w} \cdot x)\mathbf{w}$.

- Without loss of information we assume $(\mathbf{c} \cdot x) \geq 0$

$$
\begin{aligned}
cos(\delta_c)cos(\delta_w)(\mathbf{w} \cdot x) &\geq 0 \\
(\mathbf{c} \cdot x) - cos(\delta_c)cos(\delta_w)(\mathbf{w} \cdot x) &\leq \alpha \\
(\mathbf{c} \cdot x) - (\mathbf{w} \cdot \mathbf{c})(\mathbf{w} \cdot x) &\leq \alpha \\
(\mathbf{c} \cdot x_w) &\leq \alpha
\end{aligned}
$$

A.3.2 Bayesian Active Learning through Laplace Approximation

♠ Definition of the **problem**: Find a subset $\mathcal{B}$ of size k, in a pool of data $\mathcal{P}$ such that $\forall\ \mathcal{A} \subset \mathcal{P}$, $\mathcal{B} \bigcap \mathcal{A} = \emptyset$ $\mathcal{B}$ maximizes the following criterion $g(\cdot)$:

$$g(k; \mathcal{P}, \mathcal{A}) = \underset{\mathcal{B} \subset \mathcal{P},\ \mathcal{B} \bigcap \mathcal{A} = \emptyset,\ |\mathcal{B}| = k}{\arg\max} f(\mathcal{B}; \mathcal{P}, \mathcal{A})$$
$$f(\mathcal{B}; \mathcal{P}, \mathcal{A}) = Tr(\mathbb{I}_{\mathcal{A} \cup \mathcal{B}} \mathbb{I}_{\mathcal{A}}^{-1})$$

For a sake of clarity, we only consider the Fisher matrices along one layer of a network, while our proof may be easily extended to any other depth following the same path of reasoning. We assume that every Fisher matrix is a **positive definite** matrix. Also, according to the previous work of Martens *et al.* , we assume that every Fisher matrix may be decomposed as a kronecker product. Consequently every factor ψ_i and τ_i is a positive definite matrix.

$$\begin{aligned} \mathbb{I}_{\mathcal{A}} &= diag([\psi_{\mathcal{A},l} \otimes \tau_{\mathcal{A},l}]_{l=1}^{L}) \\ \psi_{\mathcal{A},l} &= \tfrac{1}{|\mathcal{A}|} \sum_{(x_i, y_i) \in \mathcal{A}} \psi_{i,l} \\ \tau_{\mathcal{A},l} &= \tfrac{1}{|\mathcal{A}|} \sum_{(x_i, y_i) \in \mathcal{A}} \tau_{i,l} \end{aligned}$$

Due to the properties inherent with the trace and the product of kronecker factors, our criterion reads:

$$f(\mathcal{B}; \mathcal{P}, \mathcal{A}) = Tr(\psi_{\mathcal{A} \cup \mathcal{B}} \psi_{\mathcal{A}}^{-1}) Tr(\tau_{\mathcal{A} \cup \mathcal{B}} \tau_{\mathcal{A}}^{-1})$$

When looking for a fixed size k, $|\mathcal{B}| = k$ then we can discard the mean factors, and our criterion is equivalent to:

$$f(\mathcal{B}; \mathcal{P}, \mathcal{A}) \equiv \Big(\sum_{x_i \in \mathcal{A}} Tr\big(\psi_i \psi_{\mathcal{A}}^{-1}\big) + \sum_{x_j \in \mathcal{B}} Tr\big(\psi_j \psi_{\mathcal{A}}^{-1}\big) \Big) \Big(\sum_{x_i \in \mathcal{A}} Tr\big(\tau_i \tau_{\mathcal{A}}^{-1}\big) + \sum_{x_j \in \mathcal{B}} Tr\big(\tau_j \tau_{\mathcal{A}}^{-1}\big) \Big)$$

It appears intuitive to check whether our objective function is *submodular*. Submodularity is a diminishing returns property: adding an element to a smaller set has larger relative effect than adding it to a larger set. A key result is that we can minimize a **submodular** function in strongly polynomial time []. Thus we can add the optimal query according to our criterion.

In order to be in accordance with minimising a submodular function, our objectives are:

- **minimising** instead of **maximizing**. This is obtained with Eq. A.7

$$g(k; \mathcal{P}, \mathcal{A}) = \underset{\mathcal{B} \subset \mathcal{P},\ \mathcal{B} \bigcap \mathcal{A} = \emptyset,\ |\mathcal{B}| = k}{\arg\min} -f(\mathcal{B}; \mathcal{P}, \mathcal{A}) \tag{A.7}$$

- our function -$f(\cdot)$ must be **decreasing** under cardinality constraint. Consequently, it leads that the minimum will be achieved for a subset of size k. This is obtained in A.3.3

- our function must respect the property of submodular functions. We detail in A.3.3.1 how to proceed.

A.3.3 Monotony

First, we decompose $f(\cdot)$ into sub-functions.

$$f(\mathcal{B}\bigcup\{x\};\mathcal{A}) = f_1(\mathcal{B}\bigcup\{x\};\mathcal{A})f_2(\mathcal{B}\bigcup\{x\};\mathcal{A})$$
$$f(\mathcal{B}\bigcup\{x\};\mathcal{A}) = \big(f_1(\mathcal{B};\mathcal{A}) + f_1(\{x\};\emptyset)\big)\big(f_2(\mathcal{B};\mathcal{A}) + f_2(\{x\};\emptyset)\big)$$
$$f_1(\mathcal{B};\mathcal{A}) = \sum_{i\in\mathcal{A}} \psi_i\psi_{\mathcal{A}}^{-1} + \sum_{j\in\mathcal{B}} \psi_j\psi_{\mathcal{A}}^{-1}$$
$$f_1(\{x\};\emptyset) = \psi_x\psi_{\mathcal{A}}^{-1}$$
$$f_2(\mathcal{B};\mathcal{A}) = \sum_{i\in\mathcal{A}} \tau_i\tau_{\mathcal{A}}^{-1} + \sum_{j\in\mathcal{B}} \tau_j\tau_{\mathcal{A}}^{-1}$$
$$f_2(\{x\};\emptyset) = \tau_x\tau_{\mathcal{A}}^{-1}$$

Because every matrix is, in theory, **symmetric, positive, definite**, consequently their trace are always stricly positive.

$$\begin{aligned} f_1(\mathcal{B};\mathcal{A}) &> 0 \\ f_1(\{x\};\emptyset) &> 0 \\ f_2(\mathcal{B};\mathcal{A}) &> 0 \\ f_2(\{x\};\emptyset) &> 0 \end{aligned}$$

Adding a new term will thus increase $f(\cdot)$. Finally -$f(\cdot)$ is a stricly negative and decreasing function.

A.3.3.1 Submodularity

Nextly we demonstrate the submodularity of $-f(\cdot)$. A necessary and sufficient condition for $-f(\cdot)$ to be submodular is that $\forall\mathcal{B}\subset\mathcal{P}$ and $\forall x_1, x_2 \in \mathcal{P}\backslash\mathcal{B}$ we have:

$$f(\mathcal{B}\cup\{x_1\};\mathcal{A}) + f(\mathcal{B}\cup\{x_2\};\mathcal{A}) \leq f(\mathcal{B};\mathcal{A}) + f(\mathcal{B}\cup\{x_1,x_2\};\mathcal{A}) \tag{A.8}$$

♣ Our proof consists into expressing $f(\mathcal{B}\cup\{x_1,x_2\};\mathcal{A})$ as a linear expression $f(\mathcal{B};\mathcal{A})$, $f(\mathcal{B}\cup\{x_1\};\mathcal{A})$, *and* $f(\mathcal{B}\cup\{x_2\};\mathcal{A})$

$$\begin{aligned} f(\mathcal{B} \cup \{x_1, x_2\}; \mathcal{A}) = & f(\mathcal{B} \cup \{x_1\}; \mathcal{A}) + f(\mathcal{B} \cup \{x_2\}; \mathcal{A}) - f(\mathcal{B}; \mathcal{A}) \\ & + f_1(\{x_1\}; \emptyset) f_2(\{x_2\}; \emptyset) + f_1(\{x_2\}; \emptyset) f_2(\{x_1\}; \emptyset) \end{aligned}$$

Because $f_1(\cdot; \emptyset)$ and $f_2(\cdot; \emptyset)$ are stricly positive functions, Eq. A.8 holds.

A.3.4 Wasserstein prototypes

♠ Definition of the **problem**: Find a subset $\mathcal{S}$ of size k, in a pool of data $\mathcal{P}$ such that $\forall\, \mathcal{U} \subset \mathcal{P}$, $\mathcal{S}$ minimizes its Wasserstein distance with $\mathcal{U}$:

$$\min_{\mathcal{S}\subset\mathcal{P},\, |\mathcal{S}|=K} \mathbb{W}_p(\mathcal{S},\mathcal{U}) \tag{A.9}$$

Note that equ (A.9) is equivalent to maximizing minus of the Wasserstein distance: (A.9)≡ (A.10):

$$\max_{\mathcal{S}\subset\mathcal{P},\, |\mathcal{S}|=K} -\mathbb{W}_p(\mathcal{S},\mathcal{U}) \tag{A.10}$$

Given a uniform empirical distribution μ whose supports is defined on $\mathcal{U} \subset \mathcal{P}$. We define the function $f(\mathcal{S};\mathcal{U}) = \mathbb{W}_p(\nu, \mu)$ with ν, μ respectively the uniform empirical distributions along $\mathcal{S}$ and $\mathcal{U}$:

$$\forall x \in \mathcal{S},\ \nu(x) = \frac{1}{|\mathcal{S}|}$$
$$\forall x \in \mathcal{U},\ \mu(x) = \frac{1}{|\mathcal{U}|}$$

We propose to approximate this problem by a greedy search: we greedily minimize $\mathbb{W}_p(\{x_1, x_2, ..., x_n\},\mathcal{U})$ from equ A.9 by adding pseudo sample x_i one at a time.

It appears intuitive to check whether our objective function is *submodular*. Submodularity is a diminishing returns property: adding an element to a smaller set has larger relative effect than adding it to a larger set. A key result is that greedily **maximising** a **monotone**, **submodular** function is guaranteed not to differ from the optimal strategy by more than a constant factor.

In order to be in accordance with using greedy search for maximising a submodular function, our objectives are:

- greedily **maximizing** instead of greedily minimizing. This is obtained with Eq. A.10
- our function must be **non-negative**. This is obtained in A.3.4.1.
- our function must be **monotone**. We detail in A.3.4.2 how to proceed.
- our function must respect the property of submodular functions. We detail in A.3.4.3 how to proceed

Our proof mostly relies on the following inequality that holds for Wasserstein distance on empirical distributions, like in our case of study.

Theorem 3.1: Upper and lower bounds for Wasserstein []

Suppose (Ω, ρ) is a metric space, and suppose μ and ν are Borel probability

distributions on Ω with countable support: there exists a countable set $\mathcal{P} \subset \Gamma$ such that $\mu(\mathcal{P}) = \nu(\mathcal{P}) = 1$. Then the following inequality holds:

$$Sep(\mathcal{P})^p \sum_{x \in \mathcal{P}} |\mu(x) - \nu(x)| \leq \mathbb{W}_p^p(\mu, \nu) \leq Diam(\mathcal{P})^p \sum_{x \in \mathcal{P}} \mid \mu(x) - \nu(x) \mid \quad \text{(A.11)}$$

which is equivalent to:

$$Sep(\mathcal{P}) \big(\sum_{x \in \mathcal{P}} |\mu(x) - \nu(x)| \big)^{\frac{1}{p}} \leq \mathbb{W}_p(\mu, \nu) \leq Diam(\mathcal{P}) \big(\sum_{x \in \mathcal{P}} \mid \mu(x) - \nu(x) \mid \big)^{\frac{1}{p}} \quad \text{(A.12)}$$

We denote respectively by $Sep(\mathcal{P})$ and $Diam(\mathcal{P})$ the minimum and maximum of distances between two samples from $\mathcal{P}$.

Consider $\mathcal{S} \subset \mathcal{P}$ the support of ν, and denote $F(\mathcal{S}) = \big(\sum_{x \in \mathcal{P}} |\mu(x) - \nu(x)| \big)^{\frac{1}{p}}$.

$F(\mathcal{S})$ can be easily bounded. First we assume $\nu \not\equiv \mu$ so that at least one element of μ (or reciprocally) is not in the support of ν. Thus $\forall \mathcal{S}\ F(\mathcal{S}) \geq \big(\frac{1}{|\mathcal{P}|}\big)^{\frac{1}{p}}$. On the other side, if μ and ν lies in independent support: $\mathcal{S} \bigcap \mathcal{U} = \emptyset$ then, F is bounded by $2^{\frac{1}{p}}$. Eventually, the inequality of $F(\mathcal{S})$ reads:

$$\big(\frac{1}{|\mathcal{P}|}\big)^{\frac{1}{p}} \leq F(\mathcal{S}) \leq 2^{\frac{1}{p}} \quad \text{(A.13)}$$

Moreover, we can upper bound the ratio between any subsets given F: $\forall\ \mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{P}$ then $\frac{F(\mathcal{S}_1)}{F(\mathcal{S}_2)} \leq (2|\mathcal{P}|)^{\frac{1}{p}}$.

Our Wasserstein objective $f(\cdot)$ with idd assumption is not necesarrily monotone, nor submodular. In Proposition 1, we modify our criterion so to make it submodular, non-negative and monotone. We demonstrate in A.3.4.4 that thanks to our new objective function, we can guarantee convergence bounds for $f(\cdot)$.

Proposition 1. *We consider as our objective function*

$$G_\alpha(\mathcal{S}) = \frac{2Diam(\mathcal{P}) - \mathbb{W}_p(\mathcal{S}, \mathcal{U})}{|\mathcal{S}|^\alpha}$$

A.3.4.1 Non-Negative

Consider $C(\mathcal{S}, \mathcal{U})$ the distance matrix between samples from $\mathcal{S}$ and $\mathcal{U}$ and $\Gamma(\mathcal{S}, \mathcal{U}$ the transportation matrix between samples from $\mathcal{S}$ and $\mathcal{U}$. For convenience we denote such matrices respectively by C and Γ. Eventually, we can express $\mathbb{W}_p(\mathcal{S}, \mathcal{U})$ by:

$$\mathbb{W}_p(\mathcal{S}, \mathcal{U}) = \sum_{i,j} C_{i,j} \Gamma_{i,j} \leq \sum_i \frac{1}{|\mathcal{S}|} * C_{i,j} \leq Diam(\mathcal{P})$$

A.3.4.2 Monotone

Here we demonstrate that our function $G_\alpha(\cdot)$ is an increasing function **for α positive but small enough**, and solely given the cardinality of the set.

Lemma 3.2: Monotone

$\exists\, M \in \mathbb{R}^+$ s.t $\forall\, \alpha \in \mathbb{R}^+$ and $\alpha \leq M$ $G_\alpha(\cdot)$ is an increasing function

♣ Our proof consists in defining a lower bound to $G(\mathcal{S} \cup \{x\})$ that is greater than an upper bound of $G(\mathcal{S})$.

$$\min_{x \in \mathcal{P} \backslash \mathcal{S}} G(\mathcal{S} \cup \{x\}) \geq Diam(\mathcal{P}) \frac{2 - \max_x F(\mathcal{S} \cup \{x\})}{(|\mathcal{S}| + 1)^\alpha}$$

$$G(\mathcal{S}) \leq \frac{2Diam(\mathcal{P}) - Sep(\mathcal{P})F(\mathcal{S})}{|\mathcal{S}|^\alpha}$$

$$Diam(\mathcal{P}) \frac{2 - \max_x F(\mathcal{S} \cup \{x\})}{(|\mathcal{S}| + 1)^\alpha} \geq \frac{2Diam(\mathcal{P}) - Sep(\mathcal{P})F(\mathcal{S})}{|\mathcal{S}|^\alpha}$$

$$\big(1 - \frac{1}{|\mathcal{S}| + 1}\big)^\alpha \geq \frac{2 - \frac{Sep(\mathcal{P})}{Diam(\mathcal{P})}\big(\frac{1}{|\mathcal{P}|}\big)^p}{2 - 2^{\frac{1}{p}}}$$

$$\big(Diam(\mathcal{P})\big(2 - \max_x F(\mathcal{S} \cup \{x\})\big)\big)\big((|\mathcal{S}| + 1)^\alpha\big) \geq \big(2Diam(\mathcal{P}) - Sep(\mathcal{P})F(\mathcal{S})\big)\big(|\mathcal{S}|^\alpha\big)$$

If $\forall\ \mathcal{S} \subset \mathcal{P}$ $\big(Diam(\mathcal{P})\big(2 - \max_x F(\mathcal{S} \cup \{x\})\big)\big)\big((|\mathcal{S}| + 1)^\alpha\big) \geq \big(2Diam(\mathcal{P}) - Sep(\mathcal{P})F(\mathcal{S})\big)\big(|\mathcal{S}|^\alpha\big)$, then we can deduce a sufficient (but not necessary) condition to ensure the monotony of $G_\alpha(\cdot)$. Eventually the threshold reads:

$$\alpha \leq \frac{1}{\log\big(1 - \frac{1}{|\mathcal{P}|}\big)} \log\Big(\frac{2 - \frac{Sep(\mathcal{P})}{Diam(\mathcal{P})}\big(\frac{1}{|\mathcal{P}|}\big)^{\frac{1}{p}}}{2 - 2^{\frac{1}{p}}}\Big) \tag{A.14}$$

Notice that both terms $\frac{2 - \frac{Sep(\mathcal{P})}{Diam(\mathcal{P})}\big(\frac{1}{|\mathcal{P}|}\big)^{\frac{1}{p}}}{2 - 2^{\frac{1}{p}}}$ and $1 - \frac{1}{|\mathcal{P}|}$ are smaller than 1, thus α is well defined.

A.3.4.3 Submodularity

A necessary and sufficient condition for $G_\alpha(\cdot)$ to be submodular is that $\forall \mathcal{S} \subset \mathcal{P}$ and $\forall x_1, x_2 \in \mathcal{P} \backslash \mathcal{S}$ we have:

$$G_\alpha(\mathcal{S} \cup \{x_1\}) + G_\alpha(\mathcal{S} \cup \{x_2\}) \geq G_\alpha(\mathcal{S}) + G_\alpha(\mathcal{S} \cup \{x_1, x_2\}) \tag{A.15}$$

Lemma 3.3: Submodular

$\exists\, M \in \mathbb{R}^+$ s.t $\forall\, \alpha\, \mathbb{R}^+\, \alpha \leq M\, G_\alpha(\cdot)$ is submodular

♣ Our proof consists in defining a lower bound to $G_\alpha(\mathcal{S} \cup \{x_1\}) + G_\alpha(\mathcal{S} \cup \{x_2\})$ that is greater than an upper bound of $G_\alpha(\mathcal{S}) + G_\alpha(\mathcal{S} \cup \{x_1, x_2\})$. Consider two samples $\{x_1, x_2\}$ out from the support of ν. Given Theorem 3.1, we obtain the following inequalities. Without loss of information, we assume $F(\mathcal{S} \cup \{x_1\}) \geq F(\mathcal{S} \cup \{x_2\})$.

$$G(\mathcal{S} \cup \{x_1\}) + G(\mathcal{S} \cup \{x_2\}) \geq 2Diam(\mathcal{P})\big[2 - F(\mathcal{S} \cup \{x_1\})\big]\tfrac{1}{(|\mathcal{S}|+1)^\alpha} \tag{A.16}$$

$$G(\mathcal{S}) + G(\mathcal{S} \cup \{x_1, x_2\}) \leq \big[2Diam(\mathcal{P}) - Sep(\mathcal{P})min(F(\mathcal{S}), F(\mathcal{S} \cup \{x\}))\big]\Big(\tfrac{1}{|\mathcal{S}|^\alpha} + \tfrac{1}{(|\mathcal{S}|+2)^\alpha}\Big) \tag{A.17}$$

Nextly, we provide a sufficient condition for the submodularity of $G_\alpha(\cdot)$. Indeed we design M so that Eq. A.15 holds: A.18 $\implies$ 3.3

If $Diam(\mathcal{P})(|\mathcal{S}|+1)^\alpha\big[2 - F(\mathcal{S} \cup \{x_1\})\big] \geq \big[2Diam(\mathcal{P}) - Sep(\mathcal{P})min(F(\mathcal{S}), F(\mathcal{S} \cup \{x\}))\big]\Big(\dfrac{1}{(|\mathcal{S}|)^\alpha}\Big)$

then we can develop the previous inequality to develop a threshold to assert the submodular property $G_\alpha(\cdot)$

$$\Big(1 - \frac{1}{|\mathcal{S}|+1}\Big)^\alpha\Big) \geq \frac{2 - \frac{Sep(\mathcal{P})}{Diam(\mathcal{P})}\big(\frac{1}{|\mathcal{P}|}\big)^{\frac{1}{p}}}{2 - 2^{\frac{1}{p}}} \implies \alpha \leq \frac{1}{\log\big(1 - \frac{1}{|\mathcal{P}|}\big)} \log\Big(\frac{2 - \frac{Sep(\mathcal{P})}{Diam(\mathcal{P})}\big(\frac{1}{|\mathcal{P}|}\big)^{\frac{1}{p}}}{2 - 2^{\frac{1}{p}}}\Big) \tag{A.18}$$

A.3.4.4 Greedy

Here we provide the convergence bounds towards optimality of $f(\cdot)$ using greedy search.

Theorem 3.2: Convergence using greedy search

Suppose (Ω, ρ) is a metric space and suppose $\mathbb{P}$ is the uniform distribution based on a countable set $\mathcal{X} \subseteq \Omega$. We denote respectively by $Sep(\mathcal{X})$ and $Diam(\mathcal{X})$ the minimum and maximum distance between two distinct samples in $\mathcal{X}$. If we greedily select the set of prototypes $\mathcal{A}_n$, such that $\mid \mathcal{A}_n \mid = n$ then, if we denote by $\mathcal{A}_n^*$ the optimal solution of cardinality n, the following upper bounds holds:

$$W(\mathbb{P}, \mathcal{A}_n) \leq (1 - \frac{1}{e})W(\mathbb{P}, \mathcal{A}_n^*) + \frac{1}{e}Diam(\mathcal{X}) \tag{A.19}$$

♣ Our proof consists in developping the optimal bound that comes with maximizing a submodular, monotone function. After selecting n samples, denoting $\mathcal{S}_n$ the set of samples, the following bounds holds:

$$G_\alpha(\mathcal{S}_n) \geq \big(1 - \frac{1}{e}\big) \max_{\mathcal{S}^* \text{ s.t. } |\mathcal{S}^*| \leq n} G_\alpha(\mathcal{S}^*)$$

Because $G_\alpha(\cdot)$ is an increasing function, then $|\mathcal{S}^*| = n$.

$$\frac{2Diam(\mathcal{P}) - \mathbb{W}_p(\mathcal{S},\mathcal{U})}{n^\alpha} \geq \big(1 - \frac{1}{e}\big) \frac{2Diam(\mathcal{P}) - \mathbb{W}_p(\mathcal{S}^*,\mathcal{U})}{n^\alpha}$$

$$\mathbb{W}_p(\mathcal{S},\mathcal{U}) \leq \big(1 - \frac{1}{e}\big)\, \mathbb{W}_p(\mathcal{S}^*,\mathcal{U}) + \frac{2Diam(\mathcal{P})}{e} \tag{A.20}$$

Bibliography

[Abadi 2016] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard*et al.* *Tensorflow: a system for large-scale machine learning.* In OSDI, volume 16, pages 265–283, 2016. (Cit page 37.)

[Adel 2017] Heike Adel et Hinrich Schütze. *Global Normalization of Convolutional Neural Networks for Joint Entity and Relation Classification.* In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1723–1729, 2017. (Cit page 111.)

[Agarwal 2005] Pankaj K Agarwal, Sariel Har-Peled et Kasturi R Varadarajan. *Geometric approximation via coresets.* Combinatorial and computational geometry, vol. 52, pages 1–30, 2005. (Cit pages 17 et 69.)

[Agueh 2011] Martial Agueh et Guillaume Carlier. *Barycenters in the Wasserstein space.* SIAM Journal on Mathematical Analysis, vol. 43, no. 2, pages 904–924, 2011. (Cit pages 70, 83 et 96.)

[Ali 2014] Alnur Ali, Rich Caruana et Ashish Kapoor. *Active Learning with Model Selection.* In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14, pages 1673–1679. AAAI Press, 2014. (Cit page 24.)

[Andoni 2016] A. Andoni, A. Naor et O. Neiman. *Impossibility of Sketching of the 3D Transportation Metric with Quadratic Cost.* In 43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016), volume 55, pages 83:1–83:14, 2016. (Cit page 92.)

[Antol 2015] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick et Devi Parikh. *Vqa: Visual question answering.* In Proceedings of the IEEE International Conference on Computer Vision, pages 2425–2433, 2015. (Cit pages 60, 109 et 141.)

[Arjovsky 2017a] M. Arjovsky, S. Chintala et L. Bottou. *Wasserstein Generative Adversarial Networks.* In Proceedings of the 34th International Conference on Machine Learning, volume 70, pages 214–223, Sydney, Australia, 06–11 Aug 2017. (Cit page 69.)

[Arjovsky 2017b] Martin Arjovsky et Léon Bottou. *Towards principled methods for training generative adversarial networks.* arXiv preprint arXiv:1701.04862, 2017. (Cit page 21.)

[Asghar 2017] Nabiha Asghar, Pascal Poupart, Xin Jiang et Hang Li. *Deep active learning for dialogue generation.* In Proceedings of the 6th Joint Conference

on Lexical and Computational Semantics (* SEM 2017), pages 78–83, 2017. (Cit page 10.)

[Balcan 2007] Maria-Florina Balcan, Andrei Broder et Tong Zhang. *Margin based active learning.* Learning Theory, pages 35–50, 2007. (Cit pages 20, 33, 44 et 46.)

[Balcan 2008] Maria-Florina Balcan, Avrim Blum et Nathan Srebro. *A theory of learning with similarity functions.* Machine Learning, vol. 72, no. 1-2, pages 89–112, 2008. (Cit page 86.)

[Balikas 2018] Georgios Balikas, Charlotte Laclau, Ievgen Redko et Massih-Reza Amini. *Cross-Lingual Document Retrieval Using Regularized Wasserstein Distance.* In European Conference on Information Retrieval, pages 398–410. Springer, 2018. (Cit page 87.)

[Bartlett 2003] Peter L Bartlett et Wolfgang Maass. *Vapnik-Chervonenkis dimension of neural nets.* The handbook of brain theory and neural networks, pages 1188–1192, 2003. (Cit page 18.)

[Bartlett 2017] Peter L Bartlett, Nick Harvey, Chris Liaw et Abbas Mehrabian. *Nearly-tight vcdimension and pseudodimension bounds for piecewise linear neural networks. arXiv preprint.* arXiv, vol. 1703, 2017. (Cit page 18.)

[Baum 1992] Eric B Baum et Kenneth Lang. *Query learning can work poorly when a human oracle is used.* In International joint conference on neural networks, volume 8, page 8, 1992. (Cit page 9.)

[Benamou 2015] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna et Gabriel Peyré. *Iterative bregman projections for regularized transportation problems.* SIAM Journal on Scientific Computing, vol. 37, no. 2, pages A1111–A1138, 2015. (Cit pages 71, 78, 83 et 84.)

[Bengio 2007] Yoshua Bengio, Pascal Lamblin, Dan Popovici et Hugo Larochelle. *Greedy layer-wise training of deep networks.* In Advances in neural information processing systems, pages 153–160, 2007. (Cit page 131.)

[Bengio 2009] Yoshua Bengio, Jérôme Louradour, Ronan Collobert et Jason Weston. *Curriculum learning.* In ICML, pages 41–48. ACM, 2009. (Cit page 28.)

[Bengio 2011] Yoshua Bengio et Olivier Delalleau. *On the expressive power of deep architectures.* In International Conference on Algorithmic Learning Theory, pages 18–36. Springer, 2011. (Cit page 131.)

[Bigot 2017] Jérémie Bigot, Raúl Gouet, Thierry Klein, Alfredo López*et al.* *Geodesic PCA in the Wasserstein space by convex PCA.* In Annales de l'Institut Henri Poincaré, Probabilités et Statistiques, volume 53, pages 1–26. Institut Henri Poincaré, 2017. (Cit pages 83, 84 et 85.)

[Blumer 1989] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler et Manfred K Warmuth. *Learnability and the Vapnik-Chervonenkis dimension.* Journal of the ACM (JACM), vol. 36, no. 4, pages 929–965, 1989. (Cit page 18.)

[Blundell 2015] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu et Daan Wierstra. *Weight uncertainty in neural networks.* arXiv preprint arXiv:1505.05424, 2015. (Cit page 52.)

[Bonneel 2011] N. Bonneel, M. van de Panne, S. Paris et W. Heidrich. *Displacement Interpolation Using Lagrangian Mass Transport.* ACM Transaction on Graphics, vol. 30, no. 6, pages 158:1–158:12, dec 2011. (Cit pages 70 et 81.)

[Bonneel 2015] N. Bonneel, J. Rabin, G. Peyré et H. Pfister. *Sliced and Radon Wasserstein Barycenters of Measures.* Journal of Mathematical Imaging and Vision, vol. 51, no. 1, pages 22–45, Jan 2015. (Cit pages 70, 73, 78 et 92.)

[Bonneel 2016] N. Bonneel, G. Peyré et M. Cuturi. *Wasserstein Barycentric Coordinates: Histogram Regression Using Optimal Transport.* ACM Trans. Graph., vol. 35, no. 4, pages 71:1–71:10, Juillet 2016. (Cit pages 70 et 83.)

[Breiman 2001] Leo Breiman. *Random Forests.* Mach. Learn., vol. 45, no. 1, pages 5–32, oct 2001. (Cit page 27.)

[Brinker 2003] Klaus Brinker. *Incorporating diversity in active learning with support vector machines.* In Proceedings of the 20th international conference on machine learning (ICML-03), pages 59–66, 2003. (Cit page 16.)

[Bromley 1994] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger et Roopak Shah. *Signature verification using a" siamese" time delay neural network.* In Advances in Neural Information Processing Systems, pages 737–744, 1994. (Cit page 79.)

[Carlini 2016] Nicholas Carlini et David Wagner. *Defensive distillation is not robust to adversarial examples.* arXiv preprint arXiv:1607.04311, 2016. (Cit page 35.)

[Carlini 2017a] Nicholas Carlini et David Wagner. *Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods.* In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec '17, pages 3–14, New York, NY, USA, 2017. ACM. (Cit page 36.)

[Carlini 2017b] Nicholas Carlini et David Wagner. *Towards evaluating the robustness of neural networks.* In Security and Privacy (SP), 2017 IEEE Symposium on, pages 39–57. IEEE, 2017. (Cit page 36.)

[Castro 2007] Rui M Castro et Robert D Nowak. *Minimax bounds for active learning.* In International Conference on Computational Learning Theory, pages 5–19. Springer, 2007. (Cit pages 10 et 20.)

[Chaloner 1995] Kathryn Chaloner et Isabella Verdinelli. *Bayesian experimental design: A review.* Statistical Science, pages 273–304, 1995. (Cit page 14.)

[Chan 1982] NN Chan. *A-optimality for regression designs.* Journal of Mathematical Analysis and Applications, vol. 87, no. 1, pages 45–50, 1982. (Cit page 14.)

[Charikar 2002] M.. Charikar. *Similarity Estimation Techniques from Rounding Algorithms.* In Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing, STOC '02, pages 380–388, 2002. (Cit page 92.)

[Chollet 2015] François Chollet*et al.* *Keras*, 2015. (Cit pages 37 et 41.)

[Chopra 2005] S. Chopra, R. Hadsell et Y. LeCun. *Learning a similarity metric discriminatively, with application to face verification.* In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 539–546. IEEE, 2005. (Cit pages 78 et 79.)

[Choromanska 2015] Anna Choromanska, Yann LeCun et Gerard Ben Arous. *Open problem: The landscape of the loss surfaces of multilayer networks.* In Conference on Learning Theory, pages 1756–1760, 2015. (Cit page 26.)

[Claici 2018] Sebastian Claici et Justin Solomon. *Wasserstein Coresets for Lipschitz Costs.* arXiv preprint arXiv:1805.07412, 2018. (Cit pages 74 et 75.)

[Cohn 1996] David A Cohn, Zoubin Ghahramani et Michael I Jordan. *Active learning with statistical models.* Journal of artificial intelligence research, 1996. (Cit page 10.)

[Collobert 2008] Ronan Collobert et Jason Weston. *A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning.* In Proceedings of the 25th International Conference on Machine Learning, ICML '08, pages 160–167, New York, NY, USA, 2008. ACM. (Cit pages 110 et 116.)

[Contardo 2017] Gabriella Contardo, Ludovic Denoyer et Thierry Artières. *A Meta-Learning Approach to One-Step Active Learning.* arXiv preprint arXiv:1706.08334, 2017. (Cit page 66.)

[Courty 2017a] N. Courty, R. Flamary, D. Tuia et A. Rakotomamonjy. *Optimal transport for domain adaptation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. (Cit pages 69 et 70.)

[Courty 2017b] Nicolas Courty, Rémi Flamary et Mélanie Ducoffe. *Learning Wasserstein Embeddings.* arXiv preprint arXiv:1710.07457, 2017. (Cit page 2.)

[Courty 2017c] Nicolas Courty, Rémi Flamary, Devis Tuia et Alain Rakotomamonjy. *Optimal transport for domain adaptation.* IEEE transactions on pattern

analysis and machine intelligence, vol. 39, no. 9, pages 1853–1865, 2017. (Cit page 73.)

[Courty 2018] Nicolas Courty, Rémi Flamary et Mélanie Ducoffe. *Learning Wasserstein Embeddings*. In International Conference on Learning Representations, 2018. (Cit page 3.)

[Cuturi 2013a] M. Cuturi. *Sinkhorn Distances: Lightspeed Computation of Optimal Transportation*. In Advances on Neural Information Processing Systems (NIPS), pages 2292–2300, 2013. (Cit page 78.)

[Cuturi 2013b] Marco Cuturi. *Sinkhorn distances: Lightspeed computation of optimal transport*. In Advances in neural information processing systems, pages 2292–2300, 2013. (Cit pages 71 et 72.)

[Cuturi 2014] M. Cuturi et A. Doucet. *Fast Computation of Wasserstein Barycenters*. In ICML, 2014. (Cit pages 70, 78, 83 et 86.)

[Cuturi 2016] Marco Cuturi et Gabriel Peyré. *A smoothed dual approach for variational Wasserstein problems*. SIAM Journal on Imaging Sciences, vol. 9, no. 1, pages 320–343, 2016. (Cit page 84.)

[Damon 2012] Mayaffre Damon. *Le discours présidentiel sous la V e République*. Chirac, Mitterrand, Giscard, Pompidou, De Gaulle, Paris, Presses de Sciences Po, 2012. (Cit page 130.)

[Dasgupta 2005a] Sanjoy Dasgupta. *Analysis of a greedy active learning strategy*. In Advances in neural information processing systems, pages 337–344, 2005. (Cit page 10.)

[Dasgupta 2005b] Sanjoy Dasgupta. *Analysis of a greedy active learning strategy*. In L. K. Saul, Y. Weiss et L. Bottou, editeurs, Advances in Neural Information Processing Systems 17, pages 337–344. MIT Press, 2005. (Cit page 24.)

[Dauphin 2017] Yann N Dauphin, Angela Fan, Michael Auli et David Grangier. *Language Modeling with Gated Convolutional Networks*. In International Conference on Machine Learning, pages 933–941, 2017. (Cit page 111.)

[de Goes 2012] F. de Goes, K. Breeden, V. Ostromoukhov et M. Desbrun. *Blue Noise Through Optimal Transport*. ACM Trans. Graph., vol. 31, no. 6, pages 171:1–171:11, Novembre 2012. (Cit page 70.)

[Ducoffe 2015] Melanie Ducoffe et Frederic Precioso. *QBDC: Query by dropout committee for training deep supervised architecture*. arXiv preprint arXiv:1511.06412, 2015. (Cit pages 1, 2, 13 et 129.)

[Ducoffe 2016a] Mélanie Ducoffe, Damon Mayaffre, Frédéric Precioso, Frédéric Lavigne, Laurent Vanni et A Tre-Hardy. *Machine Learning under the light of*

Phraseology expertise: use case of presidential speeches, De Gaulle-Hollande (1958-2016). In JADT 2016-Statistical Analysis of Textual Data, volume 1, pages 157–168. Presses de FacImprimeur, 2016. (Cit pages 2, 3 et 109.)

[Ducoffe 2016b] Melanie Ducoffe, Geoffrey Portelli et Frederic Precioso. *Scalable batch mode Optimal Experimental Design for Deep Networks.* In 29th Conference on Neural Information Processing Systems, 2016. (Cit pages 3, 57 et 141.)

[Ducoffe 2016c] Melanie Ducoffe et Frederic Precioso. *Introducing Active Learning for CNN under the light of Variational Inference.* 2016. (Cit pages 1, 3, 103 et 142.)

[Ducoffe 2018] Melanie Ducoffe et Frederic Precioso. *Adversarial Active Learning for Deep Networks: a Margin Based Approach.* arXiv preprint arXiv:1802.09841, 2018. (Cit page 2.)

[Dvijotham 2018] Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy Mann et Pushmeet Kohli. *A dual approach to scalable verification of deep networks.* arXiv preprint arXiv:1803.06567, 2018. (Cit page 137.)

[Ebrahimi 2017] Javid Ebrahimi, Anyi Rao, Daniel Lowd et Dejing Dou. *HotFlip: White-Box Adversarial Examples for NLP.* arXiv preprint arXiv:1712.06751, 2017. (Cit page 112.)

[Fawzi 2017] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard et Stefano Soatto. *Classification regions of deep neural networks.* arXiv preprint arXiv:1705.09552, 2017. (Cit pages 21, 22, 42, 43, 44 et 140.)

[Feldman, R., and J. Sanger 2007] Feldman, R., and J. Sanger. The text mining handbook. advanced approaches in analyzing unstructured data. New York: Cambridge University Press., 2007. (Cit page 111.)

[Feynman 1972] Richard P Feynman. Lectures on statistical mechanics. Addison-Wesley Reading ,MA, 1972. (Cit page 52.)

[Flaherty 2005] Patrick Flaherty, Adam Arkin et Michael I Jordan. *Robust design of biological experiments.* In Advances in neural information processing systems, pages 363–370, 2005. (Cit page 14.)

[Flamary 2017] Rémi Flamary et Nicolas Courty. *POT Python Optimal Transport library.* 2017. (Cit page 81.)

[Fletcher 2004] P. T. Fletcher, C. Lu, S. M. Pizer et S. Joshi. *Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape.* IEEE Trans. Medical Imaging, vol. 23, no. 8, pages 995–1005, aug 2004. (Cit page 84.)

[Freund 1997] Yoav Freund, H. Sebastian Seung, Eli Shamir et Naftali Tishby. *Selective Sampling Using the Query by Committee Algorithm.* Mach. Learn., vol. 28, no. 2-3, pages 133–168, Septembre 1997. (Cit page 23.)

[Gal 2016a] Yarin Gal et Zoubin Ghahramani. *Dropout as a Bayesian approximation: Representing model uncertainty in deep learning.* In international conference on machine learning, pages 1050–1059, 2016. (Cit page 54.)

[Gal 2016b] Yarin Gal, Riashat Islam et Zoubin Ghahramani. *Deep Bayesian Active Learning with Image Data.* In Bayesian Deep Learning workshop, NIPS, 2016. (Cit pages 13, 17, 20 et 26.)

[Gao 2016] Wei Gao et Zhi-Hua Zhou. *Dropout Rademacher complexity of deep neural networks.* Science China Information Sciences, vol. 59, no. 7, page 072104, 2016. (Cit page 19.)

[Gasso 2009] Gilles Gasso, Alain Rakotomamonjy et Stéphane Canu. *Recovering sparse signals with a certain family of nonconvex penalties and DC programming.* IEEE Transactions on Signal Processing, vol. 57, no. 12, pages 4686–4698, 2009. (Cit page 80.)

[Genevay 2016] A. Genevay, M. Cuturi, G. Peyré et F. Bach. *Stochastic optimization for large-scale optimal transport.* In Advances in Neural Information Processing Systems, pages 3432–3440, 2016. (Cit page 78.)

[Glorot 2011] Xavier Glorot, Antoine Bordes et Yoshua Bengio. *Domain adaptation for large-scale sentiment classification: A deep learning approach.* In Proceedings of the 28th international conference on machine learning (ICML-11), pages 513–520, 2011. (Cit page 109.)

[Gogna 2016] Anupriya Gogna et Angshul Majumdar. *Semi Supervised Autoencoder.* In International Conference on Neural Information Processing, pages 82–89. Springer, 2016. (Cit page 12.)

[Goodfellow 2015] I. J. Goodfellow, J. Shlens et C. Szegedy. *Explaining and Harnessing Adversarial Examples.* ICLR 2015, Dmbre 2015. (Cit pages 32, 35, 46, 47 et 139.)

[Graham 2015] Ben Graham, Jeremy Reizenstein et Leigh Robinson. *Efficient batchwise dropout training using submatrices.* arXiv preprint arXiv:1502.02478, 2015. (Cit pages 24 et 26.)

[Graves 2011] Alex Graves. *Practical Variational Inference for Neural Networks.* In Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11, pages 2348–2356, USA, 2011. Curran Associates Inc. (Cit pages 52, 54 et 55.)

[Grosse 2016] Roger Grosse et James Martens. *A Kronecker-factored approximate Fisher matrix for convolution layers.* arXiv preprint arXiv:1602.01407, 2016. (Cit pages 58 et 61.)

[Guyon 1993] Isabelle Guyon, B Boser et Vladimir Vapnik. *Automatic capacity tuning of very large VC-dimension classifiers.* In Advances in neural information processing systems, pages 147–155, 1993. (Cit page 18.)

[Hanneke 2011] S. Hanneke. *Rates of convergence in active learning.* ArXiv e-prints, Mars 2011. (Cit page 20.)

[Hardt 2015] Moritz Hardt, Benjamin Recht et Yoram Singer. *Train faster, generalize better: Stability of stochastic gradient descent.* arXiv preprint arXiv:1509.01240, 2015. (Cit page 19.)

[He 2017] Warren He, James Wei, Xinyun Chen, Nicholas Carlini et Dawn Song. *Adversarial Example Defense: Ensembles of Weak Defenses are not Strong.* In 11th USENIX Workshop on Offensive Technologies (WOOT 17), Vancouver, BC, 2017. USENIX Association. (Cit page 36.)

[Hinton 2012] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever et Ruslan R Salakhutdinov. *Improving neural networks by preventing co-adaptation of feature detectors.* arXiv preprint arXiv:1207.0580, 2012. (Cit page 20.)

[hoi] (Cit page 17.)

[Hoi 2006] Steven C. H. Hoi, Rong Jin, Jianke Zhu et Michael R. Lyu. *Batch Mode Active Learning and Its Application to Medical Image Classification.* ICML '06, pages 417–424, New York, NY, USA, 2006. ACM. (Cit pages 8, 10 et 56.)

[Huang 2016a] G. Huang, C. Guo, M. Kusner, Y. Sun, F. Sha et K. Weinberger. *Supervised Word Mover's Distance.* In Advances in Neural Information Processing Systems, pages 4862–4870, 2016. (Cit page 69.)

[Huang 2016b] Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha et Kilian Q Weinberger. *Supervised Word Mover's Distance.* In Advances in Neural Information Processing Systems, pages 4862–4870, 2016. (Cit page 87.)

[Huijser 2017] Miriam W Huijser et Jan C van Gemert. *Active Decision Boundary Annotation with Deep Generative Models.* arXiv preprint arXiv:1703.06971, 2017. (Cit pages 36 et 149.)

[Huszár 2012] Ferenc Huszár et David Duvenaud. *Optimally-weighted herding is Bayesian quadrature.* arXiv preprint arXiv:1204.1664, 2012. (Cit pages 75, 96 et 97.)

[Indyk 2003] P. Indyk et N. Thaper. *Fast image retrieval via embeddings.* In 3rd International Workshop on Statistical and Computational Theories of Vision, pages 1–15, 2003. (Cit page 92.)

[Ioffe 2015] Sergey Ioffe et Christian Szegedy. *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* arXiv preprint arXiv:1502.03167, 2015. (Cit page 65.)

[Kagan 2001] Abram Kagan. *Another look at the Cramer-Rao inequality.* The American Statistician, vol. 55, no. 3, pages 211–212, 2001. (Cit page 14.)

[Kalchbrenner 2014] Nal Kalchbrenner, Edward Grefenstette et Phil Blunsom. *A Convolutional Neural Network for Modelling Sentences.* In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 655–665, 2014. (Cit page 110.)

[Kapoor 2007] Ashish Kapoor, Kristen Grauman, Raquel Urtasun et Trevor Darrell. *Active learning with gaussian processes for object categorization.* In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8. IEEE, 2007. (Cit page 13.)

[Karpathy 2015] Andrej Karpathy, Justin Johnson et Li Fei-Fei. *Visualizing and understanding recurrent networks.* arXiv preprint arXiv:1506.02078, 2015. (Cit pages 111 et 128.)

[Kay 2013] Steven M Kay. Fundamentals of statistical signal processing: Practical algorithm development, volume 3. Pearson Education, 2013. (Cit page 14.)

[Khot 2006] Subhash Khot et Assaf Naor. *Nonembeddability theorems via Fourier analysis.* Mathematische Annalen, vol. 334, no. 4, pages 821–852, Apr 2006. (Cit page 92.)

[Kim 2014] Yoon Kim. *Convolutional Neural Networks for Sentence Classification.* In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, 2014. (Cit page 110.)

[Kim 2016] Been Kim, Rajiv Khanna et Oluwasanmi O Koyejo. *Examples are not enough, learn to criticize! criticism for interpretability.* In Advances in Neural Information Processing Systems, pages 2280–2288, 2016. (Cit pages 76 et 96.)

[Kingma 2014] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende et Max Welling. *Semi-supervised learning with deep generative models.* In Advances in Neural Information Processing Systems, pages 3581–3589, 2014. (Cit page 12.)

[Koch 2015] G. Koch, R. Zemel et R Salakhutdinov. *Siamese neural networks for one-shot image recognition.* In ICML Deep Learning Workshop, volume 2, 2015. (Cit page 79.)

[Kolouri 2016a] S. Kolouri, S. R. Park et G. K. Rohde. *The Radon Cumulative Distribution Transform and Its Application to Image Classification.* IEEE Transactions on Image Processing, vol. 25, no. 2, pages 920–934, 2016. (Cit pages 73 et 78.)

[Kolouri 2016b] S. Kolouri, A. Tosun, J. Ozolek et G. Rohde. *A continuous linear optimal transport approach for pattern analysis in image datasets.* Pattern Recognition, vol. 51, pages 453 – 462, 2016. (Cit page 71.)

[Kolouri 2016c] S. Kolouri, Y. Zou et G. Rohde. *Sliced Wasserstein Kernels for Probability Distributions.* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5258–5267, 2016. (Cit page 92.)

[Kolouri 2017] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev et G. K. Rohde. *Optimal Mass Transport: Signal processing and machine-learning applications.* IEEE Signal Processing Magazine, vol. 34, no. 4, pages 43–59, July 2017. (Cit page 69.)

[Koltchinskii 2010] Vladimir Koltchinskii. *Rademacher complexities and bounding the excess risk in active learning.* Journal of Machine Learning Research, vol. 11, no. Sep, pages 2457–2485, 2010. (Cit page 20.)

[Krause 2010] Andreas Krause et Volkan Cevher. *Submodular dictionary selection for sparse representation.* In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pages 567–574, 2010. (Cit page 98.)

[Krizhevsky 2012] Alex Krizhevsky, Ilya Sutskever et Geoffrey E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks.* In Proceedings of Neural Information Processing Systems (NIPS), pages 1106–1114, 2012. (Cit page 7.)

[Krueger 2017] David Krueger, Nicolas Ballas, Stanislaw Jastrzebski, Devansh Arpit, Maxinder S Kanwal, Tegan Maharaj, Emmanuel Bengio, Asja Fischer et Aaron Courville. *Deep Nets Don't Learn via Memorization.* 2017. (Cit page 19.)

[Kuleshov 2018] Volodymyr Kuleshov, Shantanu Thakoor, Tingfung Lau et Stefano Ermon. *Adversarial Examples for Natural Language Classification Problems.* 2018. (Cit page 113.)

[Kumar 2017] Sachin Kumar, Soumen Chakrabarti et Shourya Roy. *Earth Mover's Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading.* In Proceedings of the International Joint Conference on Artificial Intelligence, pages 2046–2052, 2017. (Cit page 87.)

[Kusner 2015] Matt Kusner, Yu Sun, Nicholas Kolkin et Kilian Weinberger. *From word embeddings to document distances.* In International Conference on Machine Learning, pages 957–966, 2015. (Cit page 87.)

[L. Lebart, A. Salem and L. Berry 1998] L. Lebart, A. Salem and L. Berry. Exploring textual data. Ed. Springer, 1998. (Cit page 111.)

[Langberg 2010] Michael Langberg et Leonard J Schulman. *Universal ε-approximators for integrals.* In Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms, pages 598–607. SIAM, 2010. (Cit pages 17 et 69.)

[Lee 2017] Jaeho Lee et Maxim Raginsky. *Minimax statistical learning and domain adaptation with Wasserstein distances.* arXiv preprint arXiv:1705.07815, 2017. (Cit page 73.)

[Lewis 1994] David D Lewis et William A Gale. *A sequential algorithm for training text classifiers.* In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, pages 3–12. Springer-Verlag New York, Inc., 1994. (Cit page 12.)

[Li 2015] Jiwei Li, Xinlei Chen, Eduard Hovy et Dan Jurafsky. *Visualizing and understanding neural models in NLP.* arXiv preprint arXiv:1506.01066, 2015. (Cit pages 111 et 128.)

[Liang 2017] Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin et James Stokes. *Fisher-rao metric, geometry, and complexity of neural networks.* arXiv preprint arXiv:1711.01530, 2017. (Cit page 20.)

[Lin 2017a] X. Lin et D. Parikh. *Active Learning for Visual Question Answering: An Empirical Study.* ArXiv e-prints, Novembre 2017. (Cit page 13.)

[Lin 2017b] Xiao Lin et Devi Parikh. *Active Learning for Visual Question Answering: An Empirical Study.* arXiv preprint arXiv:1711.01732, 2017. (Cit pages 59, 60 et 141.)

[Liu 2004] Ying Liu. *Active learning with support vector machine applied to gene expression data for cancer classification.* Journal of chemical information and computer sciences, vol. 44, no. 6, pages 1936–1941, 2004. (Cit page 10.)

[Liu 2016] Weiyang Liu, Yandong Wen, Zhiding Yu et Meng Yang. *Large-Margin Softmax Loss for Convolutional Neural Networks.* In ICML, pages 507–516, 2016. (Cit page 21.)

[Maaten 2008] Laurens van der Maaten et Geoffrey Hinton. *Visualizing data using t-SNE.* Journal of machine learning research, vol. 9, no. Nov, pages 2579–2605, 2008. (Cit pages 43 et 140.)

[MacKay 1992] David JC MacKay. *Bayesian interpolation.* Neural computation, vol. 4, no. 3, pages 415–447, 1992. (Cit page 54.)

[Martens 2015] James Martens et Roger Grosse. *Optimizing neural networks with Kronecker-factored approximate curvature.* arXiv preprint arXiv:1503.05671, 2015. (Cit page 58.)

[Martens 2018] James Martens, Jimmy Ba et Matt Johnson. *Kronecker-factored Curvature Approximations for Recurrent Neural Networks.* In International Conference on Learning Representations, 2018. (Cit pages 59, 60 et 61.)

[Mathai 1992] Arakaparampil M Mathai et Serge B Provost. Quadratic forms in random variables: theory and applications. M. Dekker New York, 1992. (Cit page 55.)

[Matoušek 2011] J. Matoušek et A. Naor. *Open problems on embeddings of finite metric spaces.* available at `http://kam.mff.cuni.cz/~matousek/metrop.ps`, 2011. (Cit page 92.)

[Matoušek 2013] J. Matoušek. *Lecture notes on metric embeddings.* Rapport technique, ETH Zurich, 2013. (Cit page 73.)

[Mayaffre 2012] Damon Mayaffre. *Mesure et démesure du discours. Nicolas Sarkozy (2007-2012)*, 2012. (Cit page 129.)

[Mayaffre 2017] Damon Mayaffre, Camille Bouzereau, Mélanie Ducoffe, Magali Guaresi, Frédéric Precioso et Laurent Vanni. *Les mots des candidats, de "allons" à "vertu"*, 2017. (Cit pages 2 et 3.)

[Mellet 2009a] S. Mellet et D. Longrée. *Syntactical Motifs and Textual Structures.* In Belgian Journal of Linguistics 23, pages 161–173, 2009. (Cit page 111.)

[Mellet 2009b] Sylvie Mellet et Jean-Pierre Barthélemy. *La topologie textuelle: légitimation d'une notion émergente.* Lexicometrica, vol. 7, pages http–www, 2009. (Cit page 116.)

[Miyato 2017] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama et Shin Ishii. *Virtual adversarial training: a regularization method for supervised and semi-supervised learning.* arXiv preprint arXiv:1704.03976, 2017. (Cit page 12.)

[Moosavi-Dezfooli 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi et Pascal Frossard. *Deepfool: a simple and accurate method to fool deep neural networks.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2574–2582, 2016. (Cit page 36.)

[Muandet 2017] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf*et al.* *Kernel mean embedding of distributions: A review and beyond.* Foundations and Trends® in Machine Learning, vol. 10, no. 1-2, pages 1–141, 2017. (Cit pages 58 et 102.)

[Neal 1998] Radford M Neal et Geoffrey E Hinton. *A view of the EM algorithm that justifies incremental, sparse, and other variants*, 1998. (Cit page 52.)

[Nemhauser 1978] George L Nemhauser, Laurence A Wolsey et Marshall L Fisher. *An analysis of approximations for maximizing submodular set functions—I.* Mathematical Programming, vol. 14, no. 1, pages 265–294, 1978. (Cit pages 59 et 96.)

[Neyshabur 2015] Behnam Neyshabur, Ryota Tomioka et Nathan Srebro. *Norm-based capacity control in neural networks.* In Conference on Learning Theory, pages 1376–1401, 2015. (Cit page 18.)

[Nigam 1998] Kamal Nigam et Andrew McCallum. *Pool-based active learning for text classification.* CONALD, 1998. (Cit page 24.)

[Noh 2015] Hyeonwoo Noh, Seunghoon Hong et Bohyung Han. *Learning deconvolution network for semantic segmentation.* In Proceedings of the IEEE International Conference on Computer Vision, pages 1520–1528, 2015. (Cit page 112.)

[Oliver 2018] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D. Cubuk et Ian J. Goodfellow. *Realistic Evaluation of Semi-Supervised Learning Algorithms*, 2018. (Cit page 12.)

[Ozan Sener 2018] Silvio Savarese Ozan Sener. *Active Learning for Convolutional Neural Networks: A Core-Set Approach.* International Conference on Learning Representations, 2018. accepted as poster. (Cit pages 15, 17, 37 et 62.)

[Rakotomamonjy 2018] Alain Rakotomamonjy, Abraham Traore, Maxime Berar, Rémi Flamary et Nicolas Courty. *Wasserstein Distance Measure Machines.* arXiv preprint arXiv:1803.00250, 2018. (Cit page 86.)

[Ribeiro 2016] Marco Tulio Ribeiro, Sameer Singh et Carlos Guestrin. *Model-agnostic interpretability of machine learning.* arXiv preprint arXiv:1606.05386, 2016. (Cit page 112.)

[Ribeiro 2018a] Marco Tulio Ribeiro, Sameer Singh et Carlos Guestrin. *Anchors: High-precision model-agnostic explanations.* In AAAI Conference on Artificial Intelligence, 2018. (Cit page 112.)

[Ribeiro 2018b] Marco Tulio Ribeiro, Sameer Singh et Carlos Guestrin. *Semantically Equivalent Adversarial Rules for Debugging NLP Models.* In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 856–865, 2018. (Cit pages 110, 113 et 142.)

[Ritter 2018] Hippolyt Ritter, Aleksandar Botev et David Barber. *A Scalable Laplace Approximation for Neural Networks.* In International Conference on Learning Representations, 2018. (Cit page 54.)

[Rolet 2016] A. Rolet, M. Cuturi et G. Peyré. *Fast dictionary learning with a smoothed Wasserstein loss.* In AISTATS, pages 630–638, 2016. (Cit pages 86 et 98.)

[Russakovsky 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg et Li Fei-Fei. *ImageNet Large Scale Visual Recognition Challenge.* International Journal of Computer Vision (IJCV), vol. 115, no. 3, pages 211–252, 2015. (Cit page 7.)

[Sanders 2011] Niek J Sanders. *Sanders-twitter sentiment corpus.* Sanders Analytics LLC, 2011. (Cit page 149.)

[Santambrogio 2014] Filippo Santambrogio. *Introduction to optimal transport theory.* Notes, 2014. (Cit pages 83 et 85.)

[Sawada 2017] Yoshihide Sawada, Yoshikuni Sato, Toru Nakada, Kei Ujimoto et Nobuhiro Hayashi. *All-transfer learning for deep neural networks and its application to sepsis classification.* arXiv preprint arXiv:1711.04450, 2017. (Cit page 12.)

[Schapire 1998] Robert E Schapire, Yoav Freund, Peter Bartlett et Wee Sun Lee. *Boosting the margin: A new explanation for the effectiveness of voting methods.* Annals of statistics, pages 1651–1686, 1998. (Cit page 20.)

[Schrijver 2000] Alexander Schrijver. *A combinatorial algorithm minimizing submodular functions in strongly polynomial time.* Journal of Combinatorial Theory, Series B, vol. 80, no. 2, pages 346–355, 2000. (Cit page 59.)

[Sculley 2007] D Sculley. *Online active learning methods for fast label-efficient spam filtering.* In CEAS, volume 7, page 143, 2007. (Cit page 10.)

[Seguy 2015] Vivien Seguy et Marco Cuturi. *Principal geodesic analysis for probability measures under the optimal transport metric.* In Advances in Neural Information Processing Systems, pages 3312–3320, 2015. (Cit pages 70, 73, 84, 85 et 86.)

[Settles 2008] B. Settles, M. Craven et L. Friedland. *Active Learning with Real Annotation Costs.* In Proceedings of the NIPS Workshop on Cost-Sensitive Learning, 2008. (Cit page 24.)

[Settles 2011] Burr Settles. *From theories to queries: Active learning in practice.* In Active Learning and Experimental Design workshop In conjunction with AISTATS 2010, pages 1–18, 2011. (Cit page 8.)

[Seung 1992] H. S. Seung, M. Opper et H. Sompolinsky. *Query by Committee.* COLT '92, pages 287–294, New York, NY, USA, 1992. ACM. (Cit pages 13, 23 et 24.)

[Shen 2018] Jian Shen, Yanru Qu, Weinan Zhang et Yong Yu. *Wasserstein Distance Guided Representation Learning for Domain Adaptation.* In AAAI, 2018. (Cit page 73.)

[Shirdhonkar 2008] S. Shirdhonkar et David W. Jacobs. *Approximate earth mover's distance in linear time.* In CVPR, pages 1 – 8, June 2008. (Cit page 70.)

[Simonyan 2014] Karen Simonyan et Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* CoRR, vol. abs/1409.1556, 2014. (Cit page 7.)

[Smith 2010] Brian C Smith, Burr Settles, William C Hallows, Mark W Craven et John M Denu. *SIRT3 substrate specificity determined by peptide arrays and machine learning.* ACS chemical biology, vol. 6, no. 2, pages 146–157, 2010. (Cit page 10.)

[Smith 2018] J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev et A. E. Roitberg. *Less is more: sampling chemical space with active learning.* ArXiv e-prints, Janvier 2018. (Cit page 8.)

[Sokolić 2017] Jure Sokolić, Raja Giryes, Guillermo Sapiro et Miguel RD Rodrigues. *Generalization error of deep neural networks: Role of classification margin and data structure.* In Sampling Theory and Applications (SampTA), 2017 International Conference on, pages 147–151. IEEE, 2017. (Cit pages 18 et 21.)

[Solomon 2015a] J. Solomon, F. de Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du et L. Guibas. *Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains.* ACM Trans. Graph., vol. 34, no. 4, pages 66:1–66:11, Juillet 2015. (Cit page 70.)

[Solomon 2015b] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du et L. Guibas. *Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains.* ACM Transactions on Graphics (TOG), vol. 34, no. 4, page 66, 2015. (Cit pages 78 et 83.)

[Staib 2017] M. Staib, S. Claici, J. Solomon et S. Jegelka. *Parallel Streaming Wasserstein Barycenters.* CoRR, vol. abs/1705.07443, 2017. (Cit page 78.)

[Sugiyama 2008] Masashi Sugiyama et Neil Rubens. *A batch ensemble approach to active learning with model selection.* Neural Networks, vol. 21, no. 9, pages 1278–1286, 2008. (Cit page 41.)

[Szegedy 2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow et Rob Fergus. *Intriguing properties of neural networks.* arXiv preprint arXiv:1312.6199, 2013. (Cit pages 32, 35 et 92.)

[Tanay 2016] Thomas Tanay et Lewis Griffin. *A boundary tilting persepective on the phenomenon of adversarial examples.* arXiv preprint arXiv:1608.07690, 2016. (Cit pages 44, 47 et 48.)

[Taylor 2003] Ann Taylor, Mitchell Marcus et Beatrice Santorini. *The Penn treebank: an overview.* In Treebanks, pages 5–22. Springer, 2003. (Cit page 8.)

[Tom Zahavy 2018] Alex Sivak Jiashi Feng Huan Xu Shie Mannor Tom Zahavy Bingyi Kang. *Ensemble Robustness and Generalization of Stochastic Deep Learning Algorithms*, 2018. (Cit page 22.)

[Tong 2001] Simon Tong et Daphne Koller. *Support vector machine active learning with applications to text classification.* Journal of machine learning research, vol. 2, no. Nov, pages 45–66, 2001. (Cit page 16.)

[Vanni 2018] Laurent Vanni, Mélanie Ducoffe, Carlos Aguilar, Frederic Precioso et Damon Mayaffre. *Textual Deconvolution Saliency (TDS): a deep tool box for linguistic analysis.* In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 548–557, 2018. (Cit pages 2, 3 et 128.)

[Vayer 2018] Titouan Vayer, Laetitia Chapel, Rémi Flamary, Romain Tavenard et Nicolas Courty. *Optimal Transport for structured data.* arXiv preprint arXiv:1805.09114, 2018. (Cit page 88.)

[Villani 2009] C. Villani. Optimal transport: old and new. Grundlehren der mathematischen Wissenschaften. Springer, 2009. (Cit page 70.)

[Wang 2013] W. Wang, D. Slepčev, S. Basu, J. Ozolek et G. Rohde. *A Linear Optimal Transportation Framework for Quantifying and Visualizing Variations in Sets of Images.* International Journal of Computer Vision, vol. 101, no. 2, pages 254–269, Jan 2013. (Cit page 71.)

[Wang 2015] Zheng Wang et Jieping Ye. *Querying discriminative and representative samples for batch mode active learning.* ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 9, no. 3, page 17, 2015. (Cit pages 17 et 102.)

[Wang 2016] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang et Liang Lin. *Cost-effective active learning for deep image classification.* IEEE Transactions on Circuits and Systems for Video Technology, 2016. (Cit page 13.)

[Wei 2015] Kai Wei, Rishabh Iyer et Jeff Bilmes. *Submodularity in Data Subset Selection and Active Learning.* In International Conference on Machine Learning, pages 1954–1963, 2015. (Cit pages 17, 38, 59, 64 et 102.)

[Wen 2017] Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes et Steve Young. *A Network-based End-to-End Trainable Task-oriented Dialogue System.* In Proceedings

of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, volume 1, pages 438–449, 2017. (Cit page 111.)

[Weston 2012] J. Weston, F. Ratle, H. Mobahi et R. Collobert. *Deep learning via semi-supervised embedding.* In Neural Networks: Tricks of the Trade, pages 639–655. Springer, 2012. (Cit page 79.)

[Willett 2006] Rebecca Willett, Robert Nowak et Rui M Castro. *Faster rates in regression via active learning.* In Advances in Neural Information Processing Systems, pages 179–186, 2006. (Cit page 10.)

[Wu 2017] C. Wu et E. Tabak. *Statistical Archetypal Analysis.* arXiv preprint arXiv:1701.08916, 2017. (Cit page 86.)

[Xie 2018] Yujia Xie, Xiangfeng Wang, Ruijia Wang et Hongyuan Zha. *A Fast Proximal Point Method for Wasserstein Distance.* arXiv preprint arXiv:1802.04307, 2018. (Cit pages 71, 72 et 98.)

[Xu 2012] Huan Xu et Shie Mannor. *Robustness and generalization.* Machine learning, vol. 86, no. 3, pages 391–423, 2012. (Cit page 22.)

[Yanyao Shen 2018] Zachary C. Lipton Yakov Kronrod Animashree Anandkumar Yanyao Shen Hyokun Yun. *Deep Active Learning for Named Entity Recognition.* International Conference on Learning Representations, 2018. accepted as poster. (Cit pages 13, 41 et 59.)

[Yin 2017] Wenpeng Yin, Katharina Kann, Mo Yu et Hinrich Schütze. *Comparative study of cnn and rnn for natural language processing.* arXiv preprint arXiv:1702.01923, 2017. (Cit page 111.)

[Yu 2006] Kai Yu, Jinbo Bi et Volker Tresp. *Active learning via transductive experimental design.* In Proceedings of the 23rd international conference on Machine learning, pages 1081–1088. ACM, 2006. (Cit page 11.)

[Yu 2013] W. Yu, G. Zeng, P. Luo, F. Zhuang, Q. He et Z. Shi. *Embedding with autoencoder regularization.* In ECML/PKDD, pages 208–223. Springer, 2013. (Cit page 79.)

[Yu 2014] Aron Yu et Kristen Grauman. *Fine-grained visual comparisons with local learning.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 192–199, 2014. (Cit page 149.)

[Yu 2018] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi et Quoc V Le. *QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension.* arXiv preprint arXiv:1804.09541, 2018. (Cit page 109.)

[Zamir 2018] Amir R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik et Silvio Savarese. *Taskonomy: Disentangling Task Transfer Learning.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3712–3722, 2018. (Cit page 12.)

[Zeiler 2014] Matthew D Zeiler et Rob Fergus. *Visualizing and understanding convolutional networks.* In European conference on computer vision, pages 818–833. Springer, 2014. (Cit page 111.)

[Zhai 2018] Ke Zhai et Huan Wang. *Adaptive Dropout with Rademacher Complexity Regularization.* In International Conference on Learning Representations, 2018. (Cit page 19.)

[Zhang 2000] Tong Zhang et F Oles. *The value of unlabeled data for classification problems.* In Proceedings of the Seventeenth International Conference on Machine Learning,(Langley, P., ed.), pages 1191–1198. Citeseer, 2000. (Cit page 56.)

[Zhang 2016] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht et Oriol Vinyals. *Understanding deep learning requires rethinking generalization.* arXiv preprint arXiv:1611.03530, 2016. (Cit page 19.)

[Zhang 2017] Ye Zhang, Matthew Lease et Byron Wallace. *Active GAminative Text Representation Learning.* 2017. (Cit pages 16 et 35.)

[Zhou 2010] Shusen Zhou, Qingcai Chen et Xiaolong Wang. *Active deep networks for semi-supervised sentiment classification.* In ACL ICCL, pages 1515–1523, 2010. (Cit page 13.)

[Zhu 2016] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman et Alexei A Efros. *Generative visual manipulation on the natural image manifold.* In European Conference on Computer Vision, pages 597–613. Springer, 2016. (Cit page 149.)

[Zhu 2017] Jia-Jie Zhu et Jose Bento. *Generative Adversarial Active Learning.* arXiv preprint arXiv:1702.07956, 2017. (Cit pages 9, 10 et 139.)