



HAL
open science

Apprentissage en profondeur fiable et explicable pour la classification de signaux acoustiques sous-marins

Thomas Mahiout

► **To cite this version:**

Thomas Mahiout. Apprentissage en profondeur fiable et explicable pour la classification de signaux acoustiques sous-marins. Traitement du signal et de l'image [eess.SP]. Université Côte d'Azur, 2022. Français. NNT : 2022COAZ4034 . tel-04472195

HAL Id: tel-04472195

<https://theses.hal.science/tel-04472195>

Submitted on 22 Feb 2024

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.

THÈSE DE DOCTORAT

Apprentissage en profondeur fiable et
explicable pour la classification de signaux
acoustiques sous-marins

Thomas Mahiout

Laboratoire I3S

**Présentée en vue de l'obtention
du grade de docteur en informatique**
d'Université Côte d'Azur

Dirigée par : Lionel Fillatre

Co-encadrée par : Laurent Deruaz-Pepin

Soutenue le : 23/06/2022

Devant le jury, composé de :

Abdourrahmane Atto, MCF, Polytech Annecy-
Chambéry

André Ferrari, Pr, Université Côte d'Azur

Edouard Oyallon, MCF, Sorbonne Université

François-Xavier Socheleau, Pr, IMT Atlantique

Laurent Deruaz-Pepin, Thales DMS

Lionel Fillatre, Pr, Université Côte d'Azur

Olivier Michel, Pr, Grenoble-INP

Apprentissage en profondeur fiable et explicable pour la classification de signaux acoustiques sous-marins

Jury :

Rapporteurs

François-Xavier Socheleau, Professeur, IMT Atlantique, Département Mathematical and Electrical Engineering

Olivier Michel, Professeur, Grenoble-INP, Grenoble Images Parole Signal Automatique (GIPSA-Lab)

Examineurs

Abdourrahmane Atto, Maître de conférence, Polytech Annecy-Chambéry, Laboratoire d'Informatique, Systèmes, Traitement de l'Information et de la Connaissance (LISTIC)

André Ferrari, Professeur, Université Côte d'Azur, Laboratoire J.-L. Lagrange

Edouard Oyallon, Maître de conférence, Sorbonne Université, Institut des Systèmes Intelligents et de Robotique (ISIR)

Lionel Fillatre, Professeur, Université Côte d'Azur, Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S)

Invités

Laurent Deruaz-Pepin, Manager, Thales DMS Sophia Antipolis, Laboratoire étude amont (LEA)

Résumé de la thèse

La classification acoustique sous-marine (ASM) sur antenne passive, constitue un prolongement direct à la détection. Elle a pour objectif d'identifier les sources potentielles de bruits selon des besoins opérationnels divers, allant de l'évitement des collisions, la localisation des bancs de poissons et jusqu'à l'identification de navires à propulsion mécanique (NPM). Généralement, un système de classification de données ASM peut se décomposer en trois étapes distinctes : l'extraction des caractéristiques du signal acoustique rayonné par une source, la sélection des caractéristiques adaptées au processus de classification et enfin la classification en elle-même par un algorithme spécifique. L'environnement océanique réel est cependant un domaine d'application très complexe pour la classification, et des algorithmes de classification très performants dans d'autres domaines d'application, peuvent s'avérer être inappropriés dans certains contextes opérationnels. Qui plus est, le manque de connaissances préalables sur les informations acoustiques de navires, les variations inattendues dans l'environnement océanique et l'influence des nombreuses réverbérations avant l'arrivée du signal sur capteur, sont autant plus de facteurs qui contribuent à rendre ce problème complexe pour l'apprentissage machine. Aussi, afin de minimiser l'impact de potentielles erreurs de classification en ASM, celle-ci doit être fiable du point de vue de l'utilisateur. Il est donc nécessaire de pouvoir caractériser les propriétés de robustesse des algorithmes employés et de proposer des formes d'explicabilité adaptées à son utilisation dans le contexte ASM.

Dans cette étude, nous effectuons ainsi un état de l'art des méthodes de classification acoustiques, mais également de différentes méthodes d'explicabilité et de robustesse en machine learning, pour finalement proposer des méthodes adaptées aux problématiques de l'ASM. Nous proposons notamment d'utiliser des réseaux convolutifs de scattering transform apprise (LST). En effet, leurs propriétés mathématiques adaptées à l'ASM et le fait qu'elles soient des architectures profondes intrinsèquement explicables de par leur structure mathématique, en font de bons candidats pour certains problèmes de classification ASM. Pour évaluer les performances des algorithmes d'apprentissages sélectionnés, nous les évaluons au sein d'environnements contrôlés et maîtrisés, via des modèles acoustiques de bruit de navires dont on peut dériver le détecteur de Bayes optimal. Dans ce même cadre d'explicabilité, nous proposons une classe d'architectures profondes bayésiennes permettant d'approximer efficacement la structure mathématique même de ce détecteur optimal, en forçant chaque partie de ces réseaux à remplir un rôle particulier dans la structure du détecteur profond.

Mots clés : Apprentissage Profond, Acoustique Sous-Marine, Détection Bayésienne, Explicabilité, Transformée en Ondelettes

Thesis abstract :

Underwater acoustic classification on passive antennas can be seen as a direct extension to detection. Its objective is to identify potential noise sources, according to various operational needs, ranging from collision avoidance, localization of schools of fish, to the identification of mechanically propelled vessels. Generally, an underwater acoustic signal classification system can be broken down into three distinct steps : a phase where characteristics are extracted from the acoustic signal radiated by a source, the selection of suitable characteristics for the classification process and finally the choice of a specific algorithm and the classification itself. The real ocean environment is however a very complex application domain for classification, and classification algorithms that perform well in other application domains may prove inadequate in some operational contexts. What's more, the lack of prior knowledge on ships acoustic information, the unexpected variations in the oceanic environment and the influence of many reverberations before the arrival of signals on sensors, are over factors that contribute to making this a complex problem case for machine learning. Also, in order to minimize the impact of potential classification errors in underwater acoustic, from the user's point of view, there is a need for reliability. It is therefore necessary to be able to characterize the robustness properties of our algorithms and to propose forms of explainability adapted to its use in the underwater acoustic context.

In this work, we study the state of the art of acoustic classification methods, but also look at different explicability and robustness methods in machine learning, to finally propose some methods adapted to underwater acoustic problems. In particular, we propose to use a learned based scattering transform (LST) convolutional networks. Indeed, its mathematical properties, adapted to the analysis of underwater acoustic signals and the fact that its mathematical structure lead to an intrinsically explainable architecture, make it a good candidate for certain applications in underwater acoustic classification. To evaluate the performance of state-of-the-art machine learning algorithms and ours, we evaluate them within a controlled environments provided by a ship noise acoustic models from which we can derive optimal Bayes detectors. Within this same explainability framework, we propose a class of bayesian deep architectures which allows us to efficiently approximate the very mathematical structure of this optimal detector, by forcing each part of these networks to fulfill a particular role in the deep detector structure.

Keywords : Deep Learning, Underwater Acoustic, Bayes Detection, Explainability, Wavelet Transform

Déclaration

Toutes les phrases, passages et illustrations cités dans ce document provenant de travaux d'autres personnes ont été spécifiquement identifiés ou référencés par une citation claire à l'auteur, le travail et la ou les pages correspondantes. Je comprends que faillir à cela équivaut à du plagiat et sera considéré en tant que tel lors de l'évaluation.

Prénom NOM : Thomas Mahiout

Signature : *Thomas Mahiout*

Date : 28/04/2022

Remerciements

Tout d'abord, je tiens à remercier mon directeur de thèse Lionel Fillatre, Professeur à l'Université Côte d'Azur, qui m'a encadré tout au long de cette thèse et m'a fait partager ses intuitions et connaissances sur le sujet de l'apprentissage automatique. Je le remercie aussi pour ses encouragements et ses nombreux conseils, notamment pour la rédaction de nos articles scientifiques tout au long de la thèse. Je remercie également mon tuteur de thèse Laurent Deruaz-Pepin, directeur du laboratoire études amont à Thales DMS Sophia Antipolis, pour sa disponibilité, son aide et les connaissances qu'il m'a permis d'acquérir sur le traitement du signal et les problématiques liées à l'ASM.

De même, je tiens à remercier tous les membres du laboratoire études amont et du service GSS dont notamment Serge Sitbon, pour leur accueil et toutes les questions auxquelles ils ont pu répondre au cours de ces trois ans de thèse. Je tiens tout particulièrement à remercier Xavier Christol pour son aide concernant le choix de modélisation acoustique et toutes les connaissances qu'il a pu partager sur l'ASM. Je remercie également par la même occasion Yves Chocheyras pour m'avoir partagé ses simulations de bruits acoustiques sous-marins et Jackson Shield, ancien stagiaire sur le domaine de la classification, pour toute la documentation qu'il a pu me laisser pour la thèse.

Enfin, je remercie tous les anciens doctorants que j'ai pu rencontrer au sein du laboratoire I3S ainsi les autres membres du laboratoire pour les échanges qu'on a pu avoir au cours de la thèse. Je remercie de même les membres de mon comité de suivi intermédiaire, Le Professeur Olivier Michel et Docteur Maria João Rendas pour leurs conseils sur l'orientation de la thèse.

Table des matières

1	Introduction	1
1	Attentes pour la classification acoustique sous-marine	1
2	Contexte et enjeux	2
2.1	Contexte pour implémentation	2
2.2	Contexte des données	4
3	Méthodologie	5
3.1	Définition des données d'apprentissage	5
3.2	Choix des algorithmes d'apprentissage	6
3.3	Conception d'architectures intrinsèquement explicables pour la détection	7
3.4	Utilisation de représentations apprises explicables	8
4	Contributions et plan de la thèse	9
4.1	Contributions	9
4.2	Plan de la thèse	10
2	État de l'art de la classification	11
1	Introduction	11
2	Classification acoustique	11
2.1	Classification ASM traditionnelle	11
2.2	Classification ASM par apprentissage automatique	12
2.3	Classification acoustique par apprentissage profond	12
2.4	Classification ASM par apprentissage profond	14
2.5	CNN 1D sur signaux acoustiques brutes	15
2.6	Variante des CNN 1D	18
2.7	Classification ASM sur signaux brutes	22
2.8	Structures de décisions alternatives	22
3	Formes d'explicabilité dans l'apprentissage profond	24
3.1	Définition	24
3.2	Taxonomie de l'explicabilité en apprentissage profond	25
3.3	Méthodes de visualisations	27
3.4	Méthodes de description	27
3.5	Méthodes produisant des sorties intrinsèquement explicables	28
3.6	Méthodes d'explicabilité en ASM	29
4	Conclusion	31

3	Théorie de l'apprentissage	33
1	Introduction	33
2	Contrôle du risque	33
2.1	Définition	33
2.2	Consistance	36
2.3	Consistance pour les réseaux de neurones	37
2.4	Théorème No Free-Lunch	39
2.5	Évaluation du classifieur	40
3	Théorie de Vapnik-Tchervonenkis	41
3.1	Introduction	41
3.2	Coefficient de pulvérisation	41
3.3	Dimension de Vapnik-Tchervonenkis	42
3.4	Nombre de recouvrement	43
3.5	Contrôle de capacité par la taille du réseau	44
4	Mesure de complexité	44
4.1	Nécessité de repenser la généralisation pour l'apprentissage profond	44
4.2	Objectifs des mesures de complexité	45
4.3	Complexité de Rademacher	47
4.4	Borne de généralisation PAC-Bayésienne déterministe	48
4.5	Mesure de complexité liée à la norme et à la marge	50
4.6	Mesure de complexité lié à la netteté des minima	50
5	Optimisation par SGD	52
5.1	Propriétés de la SGD	52
5.2	Régularisation implicite de la SGD par mise à jour des poids	52
5.3	Entraînement en deux phases de la SGD et visualisation dans le plan de l'information	53
6	Conclusion	55
4	Modèles acoustiques	57
1	Introduction	57
2	Modèle de bruit de cavitation d'hélice	58
2.1	Schéma de cavitation	58
2.2	Modèle de bruit de cavitation blanc	60
2.3	Modèle de bruit de cavitation coloré	61
2.4	Discussion	62
3	Modèle de bruits acoustiques sous-marins	62
3.1	Chaîne de simulation du signal	63
3.2	Data Augmentation	64
4	Modélisation des bruits de navires	65
4.1	Caractéristiques des bruits de navire	65
4.2	Modèle de bruits de navire	67
5	Modélisation des bruits biologiques	71
5.1	Modèle de bruits de cétacés	71
5.2	Modèle de bruits de crevettes claqueuses	75
5.3	Discussion	78
6	Modélisation du bruit ambiant	78

6.1	Caractéristiques du bruit ambiant	78
6.2	Modèle de bruit ambiant	82
7	Modélisation des phénomènes de propagation	84
7.1	Effet Doppler	84
7.2	Effet d'absorption du milieu	85
7.3	Trajets multiples	86
8	Conclusion	87
5	Détection binaire	89
1	Introduction	89
2	Problème de détection de bruit d'hélice blanc	89
2.1	Énoncé du problème	89
2.2	Apprentissage par réseau de neurones profond	90
2.3	Règle de détection bayésienne optimale	91
3	Apprentissage profond pour la détection de bruit d'hélice blanc	94
3.1	Architecture profonde bayésienne	94
3.2	Module de réseau de neurones d'approximation de fonction	95
3.3	Architecture CNN	98
3.4	Méthodologie pour les expérimentations	99
3.5	Optimisation des modèles profonds	100
3.6	Résultats numériques	100
3.7	Résultats sur la généralisation	102
4	Détection de bruit d'hélice coloré	105
4.1	Énoncé du problème	105
4.2	Règle de détection bayésienne optimale	106
4.3	Règle de détection sous-optimale avec hypothèse de blancheur dans une sous-bande fréquentielle	107
4.4	Règle de détection sous-optimale avec déflexion	107
4.5	Règle de détection avec blanchiment	108
4.6	Règle de détection sous-optimale avec hypothèse de blancheur dans les sous-bandes fréquentielles	108
5	Détection de bruit d'hélice aléatoirement retardé	109
5.1	Énoncé du problème	109
5.2	Règles de détection	110
6	Conclusion	111
6	Détection multi-hypothèses	112
1	Introduction	112
2	Problème de détection de bruit d'hélice blanc avec SNR inconnu	112
2.1	Énoncé du problème	112
2.2	Règle de détection bayésienne optimale	113
2.3	Règle de détection alternative pour le cas indénombrable	114
3	Apprentissage profond pour la détection de bruit d'hélice blanc avec SNR inconnu	115
3.1	Architecture profonde bayésienne	115
3.2	Approximation de l'estimateur optimal avec une architecture profonde bayésienne	116

3.3	Architecture CNN	116
3.4	Méthodologie pour les expérimentations	117
3.5	Optimisation des modèles profonds	118
3.6	Résultats numériques	119
3.7	Résultats sur l'explicabilité	120
3.8	Résultats sur le déséquilibre des sources dans les données	121
4	Problème de classification de bruit d'hélice blanc avec signature inconnue	122
4.1	Énoncé du problème	122
4.2	Règle de détection bayésienne optimale	124
4.3	Règle de détection alternative multi-hypothèses	128
5	Apprentissage profond pour la classification de bruit d'hélice blanc avec signature inconnue	130
5.1	Architecture profonde bayésienne	130
5.2	Méthodologie pour les expérimentations	130
5.3	Optimisation des modèles profonds	132
5.4	Résultats numériques	133
5.5	Résultats sur l'explicabilité	135
6	Conclusion	138
7	Classification par Scattering Transform	139
1	Introduction	139
2	Représentations pour la tâche de classification	139
2.1	Introduction	139
2.2	Représentation dans une base orthogonale	140
3	Description de la scattering transform	142
3.1	Introduction	142
3.2	Transformée d'ondelette analytique	142
3.3	Implémentation de la scattering transform	144
3.4	Similarités avec le spectre fréquentiel Mel	145
4	Propriétés de la scattering transform	146
4.1	Stabilité aux déformations temporelles	146
4.2	Conservation d'énergie et contraction	147
4.3	Mesure d'intervalle de fréquence à partir d'interférence	147
4.4	Analyse du spectre de modulation d'amplitude	148
5	Apprentissage profond de la scattering transform pour la classification de bruit d'hélice blanc avec signature inconnue	150
5.1	Motivations	150
5.2	Architecture profonde bayésienne avec scattering transform apprise	151
5.3	Module de réseau convolutif de scattering transform d'ordre 1	152
5.4	Méthodologie et optimisation des modèles pour les expérimentations	154
5.5	Résultats numériques	155
5.6	Module de réseau convolutif scattering transform d'ordre 2	156
5.7	Algorithme d'apprentissage pour Q filtres par octave	159
5.8	Conclusion	160

8	Expérimentations sur jeux de données	162
1	Introduction	162
2	Expérimentation sur l'architecture convolutive de scattering transform apprise	162
2.1	Expérimentation sur jeux de données réelles	162
2.2	Discussions	164
3	Expérimentations sur simulateur de bruits acoustiques sous-marins	165
3.1	Classification multi classes mono label	165
3.2	Influence du SNR sur l'apprentissage	167
3.3	Influence des paramètres d'apprentissage sur la généralisation	168
3.4	Mesure de complexité sur modèles appris	173
4	Conclusion	173
9	Conclusions et perspectives suite à la thèse	174
1	Conclusions	174
2	Perspectives pour la suite des travaux	175
2.1	Perspectives pour le module LST	175
2.2	Perspectives pour la classification acoustique sous-marine	176
	Appendices	191
A	Appendice A	192
1	Introduction	192
2	Définitions de termes associés aux représentations	192
3	Traitement du signal	193
3.1	Transformée de Fourier	193
3.2	Représentation DEMON	194
3.3	Système auditif humain	196
3.4	Cepstre	197
4	Représentation temps-fréquence	198
4.1	Propriétés des représentations dans le domaine temps-fréquence	198
4.2	Transformée de Fourier fenêtrée/à court terme	200
4.3	Transformée en ondelettes rapide	201
5	Modélisation d'un bruit non gaussien	203
B	Appendice B	206
1	Preuve du théorème 10	206
2	Figures	208

Table des figures

1.1	Schéma de la réception de signaux directionnels dépassant un seuil d'énergie sur une antenne acoustique remorquée.	3
2.1	Application d'un CNN 2D sur une représentation temps-fréquence $x(t, f)$. Les couches de neurones complètement connectées prennent en entrée les caractéristiques "haut niveau" extraites par les deux couches successives de convolution du modèle afin de prendre une décision sur la classe H_m du signal.	13
2.2	Application d'un CNN 1D sur un signal acoustique brut \mathbf{x} . Les couches de neurones complètement connectées prennent en entrée les caractéristiques "haut niveau" extraites par les deux couches successives de convolution temporelles du modèle afin de prendre une décision sur la classe H_m du signal.	16
2.3	Motifs appris par une première couche de convolution. Tiré de l'article de Golik et al. (2015).	17
2.4	Motifs appris par une seconde couche de convolution après avoir réordonné les filtres de la première couche par réponse fréquentielle croissante. Tiré de l'article de Golik et al. (2015).	18
2.5	Entraînement en deux phases d'un CNN 1D et 2D concaténant représentations apprises par un CNN 1D et spectrogramme Mel. Tiré de l'article de Zhu et al. (2018)	19
2.6	Architecture d'un CNN comportant des couches de convolution dilatées. Tiré de l'article de Oord et al. (2016)	19
2.7	Exemples de filtres sous-bandes appris par un modèle de CRBM dans le domaine temporel en haut, et fréquentiel en bas. Tiré de l'article de Sailor et al. (2017).	20
2.8	Utilisation d'un GMM avec $C = 6$ clusters sur $M = 4$ groupes de points suivants des distributions Gaussiennes multivariées de centres différents et de mêmes matrices de covariances.	23
2.9	Schéma de la taxonomie des différents types de méthodes d'interprétabilité. Adapté de l'article de Linardatos et al. (2021).	26
2.10	Modèle linéaire en tant que descripteur local d'un modèle complexe.	28
2.11	Exemple de multi-task learning pour les images. Tiré de l'article de Kendall et al. (2018).	29
2.12	Visualisation GradCam appliquée à du bruit ambiant blanc Gaussien, à gauche et à des clics biologiques dans du bruit ambiant blanc Gaussien, à droite. . . .	30

2.13	Visualisation GradCam appliquée à un bruit d’hélice fort, à gauche et à un bruit d’hélice faible, à droite, les deux dans du bruit ambiant blanc Gaussien.	30
2.14	Visualisation GradCam appliquée à un bruit d’hélice fort et des clics biologiques, à gauche et à un bruit d’hélice faible et du bruit de modulation, à droite, les deux dans du bruit ambiant blanc Gaussien.	31
3.1	Pulvérisation de 3 points appartenant à deux classes différentes par un classifieur linéaire dans \mathbb{R}^2 , et exemple impossible à classifier avec 4 points.	42
3.2	Évolution de l’information mutuelle discrète moyenne des parties encodeur $I(X;T_i)$ et décodeur $I(T_i;Y)$ pour chaque représentation T_i et 50 différents tirages de 5%, 45% et 85% du data set. Tiré de l’article de Shwartz-Ziv & Tishby (2017).	54
3.3	Moyenne normalisée et écart type du gradient stochastique des poids de différentes couches. Tiré de l’article de Shwartz-Ziv & Tishby (2017).	55
4.1	Signaux périodiques de modulation discret $m_{i,\omega}(k)$ (gauche) provenant d’une signature d’hélice η disposant de $p = 5$ pales avec son enveloppe globale de modulation $\sigma_\eta(k)^2$ correspondante (milieu) et les détail de 0 à 100 Hz de sa FFT (droite).	59
4.2	Zone de cavitation de la i -ème pale dans le modèle de bruit d’hélice à côté d’une image de cavitation d’hélice, tirée de theshipyardblog.com.	59
4.3	Évolution de $\sin(\varphi_i(t)) - \sin(\phi_i)$ en fonction de $\varphi_i(t)$ (en rouge).	60
4.4	Signal discret $\mathbf{x}(k)$ reçu avec des bruits de cavitation et ambiants blancs ainsi que la signature $\sigma_\eta(t)^2$ affichée en Fig. 4.1.	61
4.5	Schéma de la chaîne de simulation des signaux pour la classification multi-classes.	63
4.6	Schéma d’un système de propulsion mécanique par combiné de 4 lignes diesels.	66
4.7	Forme générale d’un spectre associé à un bâtiment de surface à droite, avec la forme d’un spectre de cavitation à gauche, et la forme d’un spectre de bruit de machines au centre.	67
4.8	Représentation temporelle, à gauche et fréquentielle, à droite, de la composante machine enregistrée pendant 5 secondes à une fréquence d’échantillonnage $fs = 44100$ Hz.	68
4.9	Spectrogramme de la composante machine associé à la figure 4.8. Pour tout les spectrogrammes, utilisation d’une fenêtre de hanning de 2048 échantillons avec un recouvrement de 50 %.	69
4.10	Représentation temporelle, à gauche et fréquentielle, à droite, de la composante de cavitation enregistrée pendant 5 secondes à une fréquence d’échantillonnage $fs = 44100$ Hz.	69
4.11	Spectrogramme de la composante de cavitation associé à la figure 4.10 en échelle logarithmique.	70
4.12	Représentation fréquentielle enregistrée pendant 5 secondes à une fréquence d’échantillonnage $fs = 44100$ Hz du bruit produit par un navire dans notre modèle.	71
4.13	Représentation temporelle, à gauche et fréquentielle, à droite, de la composante de modulation enregistrée pendant 5 secondes à une fréquence d’échantillonnage $fs = 44100$ Hz.	73

4.14	Spectrogramme de la composante de modulation associé à la figure 4.13	73
4.15	Représentation temporelle, à gauche et fréquentielle, à droite, de la composante de clic enregistrée pendant 5 secondes à une fréquence d'échantillonnage $fs = 44100$ Hz.	74
4.16	Spectrogramme de la composante de modulation associée à la figure 4.15 en échelle logarithmique.	75
4.17	Exemple d'une forme d'onde typique du claquement d'une crevette claqueuse, en haut, et sa réponse spectrale en bas. Tiré de l'article de Au & Banks (1998).	76
4.18	Représentation temporelle, à gauche, temporelle avec zoom, au milieu et fréquentielle à droite, de la composante de claquement enregistrée pendant 5 secondes à une fréquence d'échantillonnage $fs = 44100$ Hz.	77
4.19	Spectrogramme de la composante de claquement associé à la figure 4.18. . . .	78
4.20	Schéma d'une distribution verticale d'un niveau de bruit ambiant typiquement trouvée en ASM, avec (à gauche) et sans prédominance du bruit de trafic (à droite).	79
4.21	Représentation des niveaux limites de bruits ambiants pour des sources continues et intermittentes. Les flèches horizontales en haut de la figure indiquent l'étalement spectral des différentes sources de bruits ambiants : sismique, biologique, précipitations, navires et activités industrielles et bruits de glacier. Différents tracés pour chaque type de bruit sont dessinés pour indiquer qu'il est possible de rencontrer différents niveaux d'intensité dans la nature, en fonction par exemple de la force du vent. Tiré de l'article de Wenz (1962).	81
4.22	Représentation de l'intensité des bruits de surface en fonction de la fréquence, pour différents niveaux de précipitation à faible niveau de vent. Tiré de l'article de Ma et al. (2005).	83
4.23	Représentation des interactions entre vent et pluie sur les bruits de surface pour différents niveaux de vent et pour deux faibles niveaux de précipitations. Tiré de l'article de Ma et al. (2005).	84
4.24	Délai d'arrivée du son émis par une source à vitesse constante vers un récepteur placé sous la trajectoire de la source.	85
4.25	Spectrogramme d'un signal pur non déformé $s(t) = \sin(\omega t + \phi)$ de fréquence $fs = 5000$ Hz (à gauche) et spectrogramme du même signal déformé par l'effet Doppler (à droite).	86
5.1	Architecture du module CNN d'approximation de fonction $\hat{\Phi}_{g,\gamma} : \mathbb{R}^K \rightarrow \mathbb{R}^K$ avec un ensemble de noyaux de convolution équivalant à la fonction $\Phi_{g,\gamma} : \mathbb{R} \rightarrow \mathbb{R}$ distincte en rouge.	97
5.2	Architecture de $f_\theta(\mathbf{x})$ avec la terminologie de la librairie Keras.	99
5.3	Courbes ROCs sur le jeu de test de l'architecture profonde bayésienne proposée (courbes en tirets), du test de Neyman-Pearson du détecteur optimal (courbes continues) et du classifieur de comparaison M11 (courbes en points et tirets).	101
5.4	Exemples de fonctions $\Phi_{p_2,\gamma}$ apprises avec différentes profondeurs L	101
5.5	Exemple de filtre appris φ_k , en bleu, comparé à la forme du filtre optimal b_k , en rouge.	102

5.6	Précision d'entraînement et de test de l'architecture f_θ sans aucune forme de régularisation, à gauche, en fonction de la taille du jeu de données et comparaison à un perceptron prenant \mathbf{x}^2 en entrée à droite. La précision optimale correspond à un test de Monte-Carlo avec le détecteur donné dans l'équation (5.12).	103
5.7	Précision d'entraînement et de test de l'architecture f_θ avec uniquement une contrainte "Maximum kernel" à gauche, et avec uniquement une régularisation L^2 à droite, en fonction de la taille du jeu de données.	104
5.8	Précision d'entraînement et de test de l'architecture f_θ avec les deux formes de régularisation précédentes à gauche, en fonction de la taille du jeu de données et comparaison à une méthode des moindres carrés avec \mathbf{x}^2 en entrée, à droite.	104
6.1	Courbes ROC de trois différents détecteurs pour des distributions de données comportant un unique niveau de bruit d'hélice en orange et vert et un mélange de deux niveaux de bruit d'hélice en bleue.	115
6.2	Architecture de $f_\theta(\mathbf{x})$ avec la terminologie de la librairie Keras.	117
6.3	Précision d'entraînement et de test en fonction de la taille du jeu de données associé pour deux scénarios de SNR respectivement lié au niveaux ($\alpha_1 = a_1, \alpha_2 = a_2$) à gauche et ($\alpha_1 = a_2, \alpha_2 = a_3$) à droite. La précision optimale correspond à un test de Monte-Carlo avec le détecteur (6.8).	119
6.4	Exemple de fonctions apprises par un module CNN Φ_{p_2, γ_1} à gauche et Φ_{e, γ_2} à droite.	120
6.5	Filtre optimal $k \mapsto b_{1,k}$ et son estimation par le réseau profond Bayésien $k \mapsto \varphi_{1,k}$	121
6.6	Coefficients $b_{m,k}$ pour un signal de 2 secondes avec les paramètres de signature η_m présenté en figure 4.1 et une fréquence d'échantillonnage $fs = 256$ Hz (gauche) et $fs = 32$ Hz (droite).	125
6.7	Coefficients c_m en échelle logarithmique pour un problème comportant $M = 16$ signatures différentes.	125
6.8	Signatures d'hélice $\sigma_{\eta_m}^2(k)$ normalisées liées aux contributions individuelles d'une hélice avec des pales similaires, à gauche et d'une hélice avec une pale relativement distincte à droite. L'image du bas est une comparaison entre les courbes COR issues de tests de détection bayésiens binaires pour ces signatures.	126
6.9	Moyennes théoriques (noir) et variances théoriques (rouge) des coefficients $\sum_{k=1}^K b_{l,k} x^2(k) H_m \forall (l, m) \in \{0, 1, \dots, M\}^2$ pour un problème comportant $M = 16$ signatures différentes.	128
6.10	Distribution en échelle logarithmique de l'estimateur $\Lambda^*(\mathbf{x})$ (gauche) et $\Lambda_{\max}(\mathbf{x})$ (droite), pour des signaux pouvant suivre l'hypothèse H_0 ou H_1 et appartenant à 16 signatures différentes.	129
6.11	Courbes COR basées sur les estimateurs $\Lambda^*(\mathbf{x})$ et $\Lambda_{\max}(\mathbf{x})$ pour des signaux appartenant à 16 signatures différentes.	130
6.12	Précision d'entraînement et de test en fonction de la taille du jeu de données associé et des signaux de $K = 512$ échantillons. La précision optimale correspond à un test de Monte-Carlo avec le détecteur (6.30).	134

6.13	Précision d'entraînement et de test en fonction de la taille du jeu de données associé et des signaux de $K = 64$ échantillons. La précision optimale correspond à un test de Monte-Carlo avec le détecteur (6.30).	135
6.14	Exemples de fortes corrélations trouvées entre des filtres appris $\varphi_{m_1,k}$ et des filtres optimaux $b_{m_2,k}$ après l'apprentissage du modèle $f_\theta(\mathbf{x})$ avec $M = 16$ signatures ($K = 512$).	136
6.15	Exemples de filtres appris $\varphi_{m_1,k}$ proches de filtre nuls, comparés à des filtres optimaux $b_{m_2,k}$ après l'apprentissage du modèle $f_\theta(\mathbf{x})$ avec $M = 16$ signatures ($K = 512$).	137
6.16	Exemples de fonction apprises Φ_{p_2,γ_1} à gauche et Φ_{e,γ_2} au milieu et à droite, pour différentes tailles de jeu d'entraînement. Les points affichés sur les fonctions représentent des valeurs présentes dans le jeu de données.	137
7.1	Parties réelles et imaginaires d'une ondelette Morlet. Tiré du livre de Mallat (1999).	143
7.2	Schéma de la scattering transform d'ordre 2. Adapté de l'article de Andén & Mallat (2014).	145
7.3	Architecture du module LST d'ordre 1 $\tilde{\mathcal{S}}_1[\mathbf{x}]$.	153
7.4	Précision d'entraînement et de test en fonction de la taille du jeu de données associé et des signaux de $K = 512$ échantillons. La précision optimale correspond à un test de Monte-Carlo avec le détecteur (6.30).	156
7.5	Exemples de filtres appris ψ_f au sein du module LST d'ordre 1 présent dans l'architecture f_ν .	156
7.6	Architecture du module LST d'ordre 2 $\tilde{\mathcal{S}}_2[\mathbf{x}]$.	157
7.7	Exemples de filtres appris ψ_{f1} au sein d'un module LST d'ordre 2.	158
8.1	Spectrogrammes de signaux du jeu ASM ShipsEar appartenant à la classe H_0 , avec seulement du bruit ambiant (gauche) et H_1 , avec un bruit de navire à passagers (droite).	163
8.2	Matrice de précision de modèles CNN 1D entraînés sur des jeux disposants de différents SNR moyens, puis testés sur ces mêmes différents jeux de données.	168
8.3	Matrice de précision de modèles CNN 1D pré-entraînés sur des jeux disposants de différents SNR moyens, puis re-entraînés et testés sur ces mêmes différents jeux de données.	169
A.1	Exemple de DEMONgram.	195
A.2	Partie A : Coupe longitudinale de l'appareil auditif humain. Partie B : Coupe de la cochlée. Partie C : Propagation des vibrations dans la cochlée. Partie D : Schéma de la cochlée déroulée. Tiré de l'article de Girard & Roi (2013).	196
A.3	Boîte de Heisenberg d'un atome ϕ_γ . Tiré du livre de Mallat (1999).	199
A.4	Dispersion de l'énergie d'une fenêtre \hat{g} . Tiré du livre de Mallat (1999).	201
A.5	Boîte de Heisenberg de deux ondelettes. Des échelles plus petites diminuent l'étalement temporel mais augmentent le support fréquentiel, qui est alors décalé vers des fréquences plus élevées. Tiré du livre de Mallat (1999).	203

B.1 Distributions des coefficients $\sum_{k=1}^K b_{l,k} x^2(k)|H_m \forall l \in \{0, 1, \dots, M\}$ où chaque image correspond à une distribution $\mathbf{x} \sim H_m$ différente. 208

Liste des tableaux

6.1	Tableau des différentes architectures utilisées pour la comparaison, avec leurs titres associés en première ligne. La couche "Conv" représente une couche de convolution avec "same padding". Dans les couches convolutives, les nombres entre crochets indiquent respectivement le nombre de filtres et la taille du noyau de convolution. La stride/le pas est toujours fixée à 1 et le nombre multiplié aux couches Conv est le nombre de fois que la couche est réutilisée. Pour les opérations "Pool", on affiche respectivement la taille et la stride.	118
6.2	Précision d'entraînement de test de modèles entraînés sur des jeux d'entraînements avec différentes probabilités $\pi_m = 1 - \sum_{i \neq m} \pi_i$ d'obtenir un niveau de source α_m dans les données. La précision du détecteur optimal dans pour ces mêmes conditions est donnée en comparaison.	121
6.3	Tableau des différentes architectures utilisées pour la comparaison dans le cas $K = 512$, avec leurs titres associés en première ligne. Le nombre en dessous de l'architecture indique son nombre de paramètres pouvant être entraînés. La stride est toujours fixée à 1 et le nombre multiplié aux parenthèses est le nombre de fois que la couche est réutilisée. La couche BLSTM utilise un dropout de 0,1 comme dans la publication de Sang et al. (2018) et le nombre entre crochets indique le nombre de filtres.	132
6.4	Tableau des différentes architectures utilisées pour la comparaison dans le cas $K = 64$, avec leurs titres associés en première ligne.	133
7.1	Tableau des différentes architectures utilisées pour une comparaison avec des signaux de $K = 512$ échantillons. $LST_1(S, F)$ représente un module LST d'ordre 1 disposant de respectivement S couches et F filtres.	155
8.1	Architecture LST CNN utilisée pour la classification de signaux du jeu de données ShipEar.	164
8.2	Tableau des différentes architectures complètement convolutives utilisées pour une comparaison avec des signaux de bruits acoustiques sous-marins. $LST_2(\Delta K_1, \Delta K_2, F_1, F_2)$ représente un module LST d'ordre 2 disposant respectivement d'une condition d'arrêt à l'ordre 1 ΔK_1 et à l'ordre 2 ΔK_2 , F_1 filtres à l'ordre 1 et F_2 filtres à l'ordre 2.	166

8.3	Précision moyenne de classification de différentes architectures profondes sur le problème de classification de bruits acoustiques sous-marins simulés. Le symbole $\pm \sigma$ dénote l'écart type de performances des différents modèles d'une même architecture.	167
8.4	Tableau des différentes architectures utilisées pour la "GridSearch" sur les paramètres de FCNNs. Le nombre de paramètres affiché est celui d'une des combinaisons possibles. Le symbole +, sur une même couche, se réfère à une concaténation. La valeur s , dans la couche Conv, est le pas ou la stride de convolution employée, qui vaut 1 par défaut.	170
8.5	Précision de classification des architectures f_θ et f_ν pour un nombre de couches différent.	171
8.6	Précision de classification des architectures f_θ et f_ν pour différentes dimensions de filtres de convolution sur la première couche.	171
8.7	Précision de classification des architectures f_θ et f_ν pour différentes dimensions de filtres de convolution avec précision des valeurs obtenues pour chacune des deux architectures séparément.	172
8.8	Précision de classification des architectures f_θ et f_ν pour des tailles de batchs différentes.	172
8.9	Précision de classification des architectures f_θ et f_ν pour un type de pooling différent.	172
A.1	Paramètres caractéristiques de fenêtres de Fourier usuelles.	201

Nomenclature

Liste des abréviations

ASM Acoustique sous-marine

ASR Automatic Speech Recognition

BLSTM Bidirectional Long Short-Term Memory

CNN Convolutional Neural Network ou réseau de neurone convolutif en français. Architecture profonde d'apprentissage automatique.

CPU Central Processing Unit

CRBM Convolutional Restricted Boltzmann Machine

CRNN Convolutional Recurrent Neural Networks

DBN Deep Belief Network

DEMON Demodulation of Envelope Modulation On Noise

DNN Deep Neural Networks

FCNN Fully Convolutional Neural Network ou réseau de neurone entièrement convolutif en français. Réseau CNN ne comportant pas de couches de neurones cachées.

FFNN Feed-Forward Neural Networks

FFT Fast Fourier Transform ou transformée de Fourier fenêtrée rapide en français. Implémentation de la transformée de Fourier discrète au moyen de N transformées de Fourier rapides, en $\mathcal{O}(N^2 \cdot \log_2(N))$ opérations.

FIR Finite Impulse Response

GAN Generative Adversarial Network

GMM Gaussian Mixture Model

GMM Gaussian Mixture Models ou modèle de mélange Gaussien en français.

GPU Graphical Processing Unit

HHT Hilbert-Huang Transform

HMM Hidden Markov Models

KL-divergence Divergence de Kullback Leibler

LIME	Local Interpretable Model-Agnostic Explanations
LSTM	Long Short-Term Memory
MFCC	Mel-Frequency Cepstral Coefficient ou coefficients cepstraux de fréquence mel en français. Coefficients associés à la transformation en cosinus linéaire du logarithme du spectre de puissance d'un signal sur une échelle de fréquence mel (non linéaire).
MLP	Multi-Layer Perceptron ou perceptron multi-couches en français. Algorithme de classification d'apprentissage automatique.
MSE	Mean Squared Error
NPM	Navire à propulsion mécanique
RBM	Restricted Boltzmann Machine ou machine de Boltzmann restreinte en français.
RNN	Recurrent Neural Network ou réseau de neurones récurrents en français.
ROC	Receiver Operating Characteristic
SAI	Stabilized Auditory Images
SGD	Stochastic Gradient Descent
SHAP	Méthode associé aux valeurs de Shapley
SIF	Spectrogram Image Feature
SNR	Signal to Noise Ratio ou signal sur bruit en français. Rapport, généralement exprimé en décibel, entre les puissances du bruit utile et du bruit de fond, information non significatives.
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine ou Machine à vecteur de support en français. Algorithme de classification d'apprentissage automatique.
XAI	Explainable AI

Liste des symboles mathématiques

*	Opération de convolution
\mathbb{C}	Ensemble des nombres complexes
$\lceil \cdot \rceil$	Arrondie à l'entier supérieur
$\delta(\cdot)$	Fonction dirac
$\lfloor \cdot \rfloor$	Arrondie à l'entier inférieur
$\mathbb{E}[\cdot]$	Opérateur d'espérance
$\mathbb{V}[\cdot]$	Opérateur d'espérance
$\mathcal{F}\{\cdot\}$	Transformée de Fourier discrète
$\mathcal{O}(\cdot)$	Notation grand o, pour les relations d'ordre asymptotique

$\mathcal{R}(\cdot)$	Risque réel d'une fonction
$\mathcal{R}_N(\cdot)$	Risque empirique d'une fonction
\odot	Opération de produit terme à terme
\amalg	Notation pour la multiplication
\mathbb{R}	Ensemble des nombre réels
$\rho(\cdot)$	Fonction d'activation
\sum	Notation pour l'addition
f_θ	Fonction définis par une architecture de paramètres θ
K	Désignera généralement le nombre d'échantillons temporels discrets d'un signal
N	Désignera généralement le nombre d'exemple d'entraînement

Liste des unités

dB Décibel, l'échelle logarithmique définit une expression en décibels à partir d'une variable de mesure x de type énergétique :

$$X = 10. \log\left(\frac{x}{x_0}\right),$$

avec X en dB, x une valeur naturelle et x_0 une valeur de référence. Pour les mesures de pression en dB , nous utiliserons généralement une valeur de référence la pression p_0 , à une distance de référence de 1 m de la source et dans une bande de fréquences de largeur 1 Hz :

$$P = 20. \log\left(\frac{p}{p_0}\right).$$

Hz	Hertz
m	Mètre
Pa	Pascal
s	Seconde
W	Watt

Chapitre 1

Introduction

1 Attentes pour la classification acoustique sous-marine

Reprenant l'article Cols Bleus de [Bordes \(1996\)](#), l'un des premiers essais de l'intelligence artificielle dans le domaine de l'ASM, eu lieu au sein de la marine française dès la fin des années 1980, sous la forme d'outils modélisant les connaissances et pratiques des différents opérateurs travaillant sur les sonars. Cette première automatisation de l'analyse acoustique devait ainsi soulager la tâche toujours plus importante attribuée à ces derniers. Une nouvelle méthode de classification mêlant données acoustiques, images des fonds et connaissance de la situation opérationnelle, devait ainsi voir le jour en accord avec la vision d'analystes et experts systèmes. Les ingénieurs de ce projet avaient cependant relevé plusieurs facteurs qui pourraient venir discréditer la fiabilité d'une classification apportée par un tel outil.

Premièrement, les scientifiques avaient conclu à l'époque que les mécanismes cognitifs mis en jeu lors de la reconnaissance des bruits par l'oreille, étaient peu propices à l'écriture de règles assez strictes afin d'obtenir un outil de classification suffisamment fiable pour des applications critiques comme les opérations marines. En effet, la complexité d'une situation opérationnelle ne se résume pas simplement à reconnaître des sonorités telles que le "chuintement métallique" d'une hélice, une attaque des pales du bruit de cavitation ou encore un grincement provoqué par des lignes d'arbres. Ces bruits sont confondus dans des bruits environnementaux tels que les bruits de mers, de pluie, de bancs de crevettes claqueuses, ou encore de travaux côtiers. Si des analystes du domaine acoustiques parviennent à extraire des informations, c'est qu'ils ont développé une "profondeur d'écoute" liée au contexte opérationnel et leur permettant de discerner un bruit faible provenant d'une source proche, d'un bruit fort mais provenant d'une source plus éloignée. Aussi, cette "profondeur d'écoute" est implicitement liée aux hypothèses que font les opérateurs sur la source d'un bruit, des éléments relativement difficiles à modéliser au sein d'un algorithme.

Un deuxième problème pour l'ASM venait du fait qu'en parallèle des avancées sur l'analyse acoustique des bruits émis par les navires, les concepteurs du secteur naval en tiraient des conclusions permettant de rendre leurs bâtiments toujours plus silencieux. On pourra notamment penser plus récemment à des systèmes de propulsion par jets d'eau qui ne permettent plus d'entendre de rythme de pales. Le problème est ainsi qu'on pourrait demander dans un

contexte opérationnel que notre algorithme statue sur des bruits sous des formes encore jamais rencontrées, là où un humain pourrait imaginer de nouvelles possibilités en temps réel grâce à ses expériences acquises depuis l'enfance. Néanmoins certains types de bruit n'ont que très peu évolués depuis les débuts de l'ASM. On peut notamment penser aux bruits moteurs liés aux défauts de la chaîne de transmission, à des bruits transitoires provoqués par un changement d'état du système, à des craquements ou encore des chocs internes présents sur tout type de bâtiments. Ces bruits resteront toujours des indicateurs critiques de la présence d'une source pouvant être un navire, dans une voie spécifique et devront être traités comme tel par un algorithme de classification automatique, bien que d'autres types de bruits puissent être inconnus.

Dès lors, beaucoup de doutes furent émis sur la possibilité de pouvoir mettre en place un tel système de classification. Ses tâches auraient été de capturer le tempo des lignes d'arbres moteurs, de compter le nombre de tours par minute de l'hélice ainsi que le nombre de pales et enfin d'en tirer des conclusions compte tenu des analyses spectrales et des informations cinématiques disponibles. Ces dernières doivent d'ailleurs être inférées grâce aux hypothèses de trajectographies permises par l'évolution des détections énergétiques au sein de différentes pistes, elles-même liées à l'angle de détection du récepteur. Ainsi, selon les dires du capitaine de frégate Jean-Pierre Bordes, "la classification est une décision de commandement qui se construit par un dialogue au cours duquel l'argumentaire technique ne prend sa valeur que s'il s'inscrit de façon cohérente dans le contexte du moment, que s'il peut être reconsidéré en totalité ou simplement nuancé par la prise en compte d'informations nouvelles, acoustiques ou non, comme une interception électromagnétique ou un message de renseignement". Selon lui, une décision de classification ou d'identification doit se faire par le biais d'un avis "exprimé avec conviction et discernement", chose qui semble déjà difficilement compatible avec la décision brute d'un algorithme d'apprentissage machine.

La marine française avait ainsi conclu dans les années 90 que la machine devait se contenter de servir l'individu dans sa tâche plutôt que chercher à le remplacer. L'exploitation des informations par l'humain semblait alors rester le choix privilégié par le commandement pour les années à venir, compte-tenu des aspects complexes et critiques présents dans la classification.

2 Contexte et enjeux

2.1 Contexte pour implémentation

Avec le développement d'antennes acoustiques comportant un nombre croissant de capteurs, de nouveaux algorithmes de traitement d'antenne, ainsi qu'un trafic maritime toujours plus intense, il est prévu que le nombre de détections énergétiques reçues en sortie de traitement d'antenne augmente considérablement dans les années qui viennent. Pour autant, le nombre d'opérateurs présents sur des bâtiments transportant des antennes restera limité et il ne leur sera pas possible de passer plus de temps à analyser ces détections. La capacité des opérateurs à analyser toute l'information mise à leur disposition, tel qu'ils le font aujourd'hui, pourrait ainsi être remise en question. Aussi, le développement d'algorithmes en sortie de détection tend à inclure des fonctionnalités permettant d'aider ces derniers à accéder aux informations critiques plus efficacement, mais ce, dans des cadres de fonctionnement restreints.

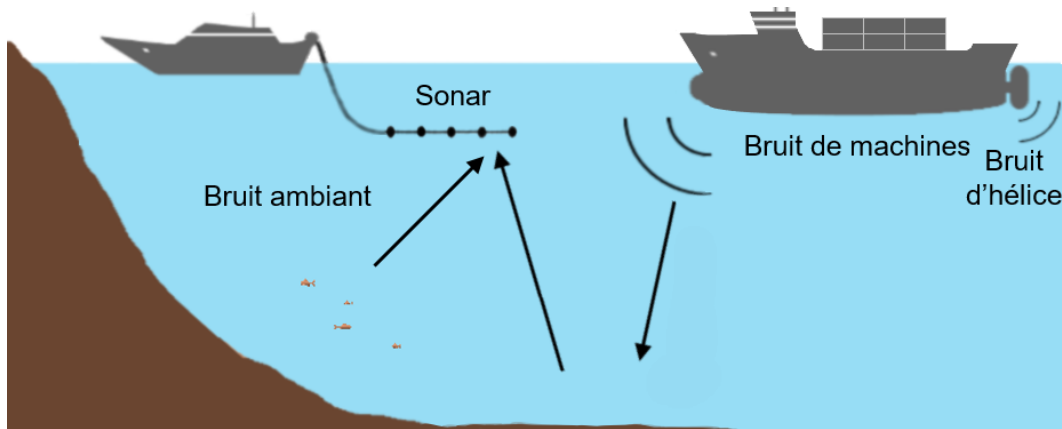


Figure 1.1: Schéma de la réception de signaux directionnels dépassant un seuil d'énergie sur une antenne acoustique remorquée.

À cet égard, une classification automatique pouvant discriminer bruits ambiants et bruits de navire à propulsion mécanique (NPM) avec un faible taux de fausses alarmes, pourrait aider les opérateurs en réduisant le nombre d'interventions nécessaires durant la surveillance des détections. En effet, comme on peut le voir sur la figure 1.1, en plus du bruit ambiant multi-directionnel, provoqué entre autres par la houle, des bruits parasites quelconques peuvent venir se cumuler dans une direction, déclenchant ainsi une détection automatique sur un critère de niveau d'énergie. Ces détections doivent alors être différenciées des sources mécaniques potentielles qui émettent des émissions vers les capteurs d'une antenne avec un trajet direct et d'éventuelles réfléchies sur fond ou surface. L'objectif serait ainsi de trier les trop nombreuses détections basées sur les niveaux d'énergie, afin de permettre aux opérateurs d'analyser uniquement des bruits de navires pouvant constituer un intérêt. Qui plus est, les seuils de détections étant actuellement basés sur des niveaux d'énergie déterminés en prenant en compte l'ensemble de l'équation du sonar, notamment le niveau de bruit ambiant, les pertes de propagation, etc., le fait de trier automatiquement ces détections avec des critères plus complexes que le niveau d'énergie, pourrait également introduire un nouveau gain, ou à l'inverse une nouvelle perte dans l'équation du sonar, ce qui nécessiterait une réévaluation de ces seuils. Ainsi, avant d'envisager d'adopter des méthodes de classification basées sur l'apprentissage profond, il est nécessaire de savoir à la fois s'il est possible de maîtriser la fausse alarme de ces algorithmes, et de savoir quelles sont leurs performances, afin de pouvoir évaluer leur impact potentiel sur la chaîne sonar actuelle. Notre étude des algorithmes de classification pour la détection du bruit de d'hélice/propulsion doit ainsi venir éclaircir ces interrogations, notamment s'il est possible de déterminer quelles sont leur probabilité de détection (PD) sur probabilité de fausse alarme (PfA) pour différents niveau de bruit, ou SNR, et comment leurs performances sont affectées par le bruit.

En dehors des performances de classification, il est également nécessaire pour le cas d'application qu'est l'ASM, d'éclaircir deux aspects d'intérêts de l'apprentissage profond : l'explicabilité et la fiabilité. Concernant l'explicabilité, on peut penser au fait de proposer des structures d'architectures adaptées au problème de classification posé, ce qui nécessite d'effectuer un état de l'art sur ce domaine d'application. Il serait également intéressant de pouvoir

établir un lien théorique fort entre l'architecture d'un modèle et les propriétés de la représentation des signaux qui en découle, que ce soit sa capacité à approximer efficacement une certaine classe de fonction, ou encore ces invariances intrinsèques. Une autre attente de cette thèse serait de proposer une forme d'explicabilité en terme de structure mathématique intrinsèquement explicable, du point de vue des utilisateurs finaux. En effet les architectures profondes récentes sont communément perçues comme des boîtes noires pour leurs utilisateurs et une telle démarche pourrait tirer profit des connaissances disponibles dans le traitement du signal pour se différencier d'une simple application d'algorithmes tirés d'autres domaines.

Comme pour l'explicabilité, la fiabilité d'un algorithme pour la classification ASM n'a pas de définition fixe et peut induire plusieurs critères dont l'explicabilité même de l'architecture. Un des objectifs de cette thèse est ainsi d'établir un lien théorique entre l'architecture d'un réseau profond et la classification/détection optimale ou presque de signaux bruités dans le cadre Bayésien. Dans la théorie de l'apprentissage, la fiabilité peut aussi concerner la robustesse dans le contrôle du risque, soit les pertes d'optimalité liées à l'apprentissage avec l'erreur d'approximation, l'optimisation non convexe et les bornes de généralisation. Il serait alors nécessaire d'étudier l'impact qu'ont les différents paramètres d'une architecture et les paramètres d'apprentissage sur la robustesse. Enfin la fiabilité au sens opérationnel peut exiger de maîtriser les performances de nos algorithmes sur données simulées et le fait de pouvoir déterminer si des entrées réelles sont éloignées de la distribution des données d'apprentissage et ce, afin d'éviter des comportements de classification non envisagés.

2.2 Contexte des données

Reproduire l'analyse auditive d'un sous-marinier exige le développement de techniques de classification passives permettant d'extraire et d'exploiter les différentes caractéristiques des bruits rayonnés. L'une des difficultés majeures rencontrées dans cette tâche est la nécessité de faire ressortir des informations à différentes échelles temporelles au sein d'une même représentation des signaux acoustiques : l'information portée par le bruit de cavitation est étalée sur une bande allant de 1 à 20 kHz et peut être déformée par les effets de propagation tels que l'effet Doppler. Bien que la classification par apprentissage profond soit souvent utilisée dans des conditions de SNR élevées telles que la reconnaissance vocale, abordée par [Deng et al. \(2013\)](#), la classification de sons environnementaux, comme proposé par [Piczak \(2015b\)](#), ou plus récemment la détection d'événements acoustiques, présentée dans les travaux de [Mesaros et al. \(2016\)](#), l'environnement sous-marin impose de traiter des niveaux substantiels de bruits ambiants masquant partiellement les signaux de navires.

Fort heureusement, les signatures de bruit de navires, et plus spécifiquement les bruits d'hélices, présentent certaines formes de cyclostationnarité sous la forme périodicité dans les statistiques du second ordre, comme le soutiennent les publications de [Byun et al. \(2016\)](#), [Chung et al. \(2011\)](#) et [Kirsteins et al. \(2011\)](#). Le bruit ambiant, en revanche, est principalement de nature stationnaire ou impulsif, ce qui permet de potentiellement distinguer les bruits d'hélice dans le bruit ambiant. La cyclostationnarité est un phénomène qui peut également être trouvé dans d'autres domaines, le livre de [Antoni \(2009\)](#) en faisant une bonne synthèse. Parmi ces domaines, on pourra notamment penser à l'analyse du climat ou les communications acoustiques avec porteurs bruyants, étudiées par [Kozlenko et al. \(2020\)](#) ainsi que [Socheleau, Ciblat](#)

& Houcke (2009). Dans notre tâche, ces périodicités cachées sont suivies de près par les opérateurs sonars lors de la classification des enregistrements acoustiques pour compter le nombre de pales et déterminer la vitesse de rotation de l'hélice. Ceci a conduit au développement de techniques de détection passive permettant d'extraire les caractéristiques des bruits rayonnés comme l'analyse DEMON proposée par Lourens (1997) et Hanson et al. (2008). Certaines représentations peuvent également être choisies pour leurs propriétés d'invariances liées à la tâche de classification posée, c'est notamment le cas de la scattering transform de Mallat (2012) qui permet de linéariser des difféomorphismes tels que l'effet Doppler.

La classification en acoustique sous-marine étant un domaine très vaste du fait d'une très grande diversité d'espèces marine, de navires ainsi que de bruits environnementaux, l'objectif de la thèse devra se contraindre à étudier des tâches de classification restreintes comme la classification de bruit de navire. En particulier, l'étude de la détection de bruit de navire se focalisera sur la distinction entre bruits de propulsion et bruits ambiants. Le principal objectif pour cette classification passive devrait alors être l'extraction des périodicités cachées dans le bruit de propulsion, et ce, en présence d'un bruit ambiant bande large et éventuellement de bruits parasitaires.

3 Méthodologie

3.1 Définition des données d'apprentissage

L'accès à des jeux de données ASM "Open Source" suffisamment larges et documentés pour une analyse de nos algorithmes n'étant pas disponible à l'heure actuelle, nous développons dans le chapitre 4 un simulateur de bruit ASM en se basant sur divers modèles de bruits de navires, ambiants et biologiques ainsi que divers modèles de propagations. En particulier, pour des études plus cadrées, nous proposons un nouveau modèle de bruit d'hélice s'efforçant de simuler le plus fidèlement possible les contributions individuelles de chaque pale à la signature globale perçue, à la manière de ce que nous avons proposé dans Mahiout et al. (2020). Contrairement à des modèles plus réalistes tels que celui de Bagheri et al. (2014), nous choisirons de limiter la complexité de notre modèle, en utilisant un bruit gaussien bande large, par défaut blanc, comme représentation simplifiée des bruits ambiants et de cavitation. Ces représentations simplifiées nous permettent cependant d'aborder le problème de détection de bruit de navires connaissant la distribution réelle $\mathcal{D}(\mathbf{x}, y)$ qui relie les signaux d'entrée \mathbf{x} à leurs labels associés y et ce, pour un problème de classification donné. Par conséquent, contrairement aux cas de jeux de données réelles, nous serons en mesure de dériver le test de Bayes optimal pour chacun des problèmes de classification de sources ASM proposés à partir de ce modèle. Ces tests optimaux pourront alors être utilisés comme modèles de comparaison pour des architectures profondes. Par la suite, nous présenterons séparément dans le chapitre 5 plusieurs détecteurs Bayésien optimaux dérivés du modèle de bruit d'hélice et ce, pour différentes hypothèses de problèmes de classification, dont certaines composites qui seront au centre du chapitre 6.

Les implications liées à la présence de certains phénomènes environnementaux et de certaines variabilités environnementales devant être prises en compte pour une implémentation de nos algorithmes en conditions réelles, à défaut d'expérimentations supplémentaires, nous

tiendrons également compte au cours de la thèse de l'impact potentiel sur nos algorithmes des hypothèses de modélisation suivantes :

- Coloration aléatoire du bruit ambiant : permet d'éprouver la capacité d'un algorithme à prendre une décision sur différentes bandes fréquentielles.
- Présence de multi-trajets : permet d'évaluer la robustesse voire l'invariance d'un algorithme à une répétition aléatoirement décalée dans le temps et modifiée du signal source.
- Présence d'effets Doppler : permet d'évaluer la robustesse voire l'invariance d'un algorithme à la déformation d'une signature de bruit d'hélice.

Enfin, bien que le test de Bayes fournisse des performances statistiques optimales pour la distribution $\mathcal{D}(\mathbf{x}, y)$, il est nécessaire pour dériver son expression de connaître cette distribution conjointe, ce qui est rarement le cas dans la réalité où l'environnement n'est pas entièrement caractérisable. Ainsi, avec ses récentes avancées en matière d'algorithmes d'optimisation, d'architectures et de technologie de calcul, l'apprentissage automatique est réapparu comme une possibilité pour aider des opérateurs humains dans leur constante tâche de surveillance. En effet, un modèle d'apprentissage profond n'a pas un accès direct au modèle du problème de classification défini par $\mathcal{D}(\mathbf{x}, y)$, mais doit approximer le détecteur de Bayes $\delta^*(\mathbf{x})$, en s'appuyant uniquement sur un ensemble de signaux labélisés $\mathcal{S}_N = \{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$ où, $\mathbf{x}^{(i)} \in \mathbb{R}^K$ est le signal reçu et $y^{(i)} \in \{0, 1\}$ est son label, $y^{(i)} = k$ signifiant que $\mathbf{x}^{(i)}$ suit l'hypothèse H_k .

3.2 Choix des algorithmes d'apprentissage

Dans la communauté de l'apprentissage automatique, de nombreuses tentatives ont été faites pour résoudre des problèmes de classification ASM grâce à des représentations extraites des signaux. De cette manière, des représentations fréquentielles telles que la transformée de Fourier à court terme ou la transformée en ondelettes, ont d'abord été utilisées dans les années 2000 en combinaison avec des SVMs, HMMs ou MLPs pour classifier des enregistrements acoustiques sous-marins, à l'image des travaux de [Soares-Filho et al. \(2000\)](#), [Azimi-Sadjadi et al. \(2000\)](#) et [Shi & Xu \(2013\)](#). De même, la classification par SVM de données dans le domaine cepstral a fait l'objet de nombreuses publications comme [Das et al. \(2013\)](#) et [Sherin & Supriya \(2015a\)](#), la plupart d'entre elles utilisent cependant des MFCCs, connus pour être moins efficaces pour la classification de bruits de navire. L'émergence d'algorithmes d'apprentissage en profondeur a également eu un grand succès dans le domaine, que ce soit en utilisant des représentations temps-fréquence en combinaison avec un CNN comme le font [Yue et al. \(2017\)](#), [Wu et al. \(2018\)](#) et [Valenti et al. \(2017\)](#), ou alternativement avec des DBN comme le font [Kamal et al. \(2013\)](#). La combinaison de plusieurs représentations expertes s'avère également être une possibilité pour améliorer la précision de classification, comme le montrent [Cao et al. \(2019\)](#). En effet on peut alors concaténer des caractéristiques extraites complémentaires sur différents canaux des données d'entrée.

Néanmoins, bien que l'apprentissage en profondeur soit connu comme un algorithme d'apprentissage automatique de bout en bout, i.e. qui ne nécessite pas nécessairement un pré-

traitement des données, peu de travaux ont évalué les avantages de l'utilisation de signaux acoustiques bruts comme entrée pour la classification ASM. C'est cependant la représentation utilisée en entrée de la plupart des détecteurs optimaux théoriques, si on se réfère au livre de [Poor \(2013\)](#). Qui plus est, les opérateurs sonar se basent principalement sur le signal acoustique brut pour prendre une décision, les autres formes de représentations ne servant qu'à confirmer ou affiner une décision déjà prise avec l'écoute. Comme cela est indiqué dans le chapitre 2, de nombreuses publications telles que [Dai et al. \(2017\)](#), [Zhu et al. \(2018\)](#) et [Tokozume et al. \(2017\)](#) ont en revanche montré que les CNNs 1D, ainsi que leurs variantes, pouvaient être utilisés dans d'autres domaines acoustiques pour extraire directement les caractéristiques de divers types de signaux. En effet, dans ces autres domaines d'application, les architectures profondes prenant en entrée des signaux acoustiques brutes peuvent apprendre des représentations internes adaptées à la tâche de classification demandée, ce qui permet souvent d'obtenir des performances redéfinissant l'état de l'art.

Toutes ces avancées peuvent paraître encourageantes pour une utilisation opérationnelle, mais dans une application critique pour la sécurité telle que la classification sonar, aider des opérateurs nécessite un niveau important de robustesse. La robustesse, comme une forme de fiabilité, doit permettre de prêter plus de confiance aux prédictions, et ainsi de moins fréquemment gêner les utilisateurs pour cause de doutes sur les performances de l'architecture. Dans le chapitre 8 nous testons ainsi plusieurs de ces architectures sur nos jeux de données simulés pour analyser leurs performances de généralisation en plus de leur précision de classification. En particulier nous étudierons comment les performances de généralisation sont affectées par le paramétrage d'un CNN 1D. La fiabilité peut également concerner l'interprétation de la raison d'une décision au travers de différentes méthodes de visualisation, et ce, même si l'algorithme est considéré comme une boîte noire, comme le montrent [Fel & Vigouroux \(2020\)](#). Néanmoins dans les chapitres 5, 6 et 7, nous privilégierons une explicabilité structurelle. En effet, de notre point de vue, il est impossible de justifier complètement l'utilisation d'un réseau profond à un être humain sans lui fournir une explication satisfaisante quant au fonctionnement même de l'algorithme de classification. Aussi, en raison de leurs cascades d'opérations internes non linéaires, les réseaux profonds peuvent difficilement être vus comme des algorithmes intrinsèquement explicables et cela aura toujours un biais cognitif sur l'utilisateur, surtout quand le coup d'une mauvaise classification a de graves conséquences.

3.3 Conception d'architectures intrinsèquement explicables pour la détection

Comme le stipulent [Xie et al. \(2020\)](#), une structure mathématique interne qui soit interprétable pourrait être considérée comme une forme d'explicabilité adaptée à notre problème. Aussi, bien que de nombreuses études, comme celle de [Faragó & Lugosi \(1993\)](#), aient montré que sous certaines hypothèses appropriées le risque d'un réseau de neurones profond pouvait converger vers le risque minimum atteint par le détecteur de Bayes, la convergence de l'architecture d'un réseau de neurones vers la structure même du détecteur de Bayes est un sujet plutôt négligé. Dans les chapitres 5 et 6, nous nous intéresserons donc à l'étude de cette convergence fonctionnelle qui garantirait que la structure d'un réseau de neurones profonds puisse correspondre à la structure du détecteur de Bayes, puisque l'utilisation d'un algorithme boîte noire est difficilement justifiable dans ce domaine. Nous proposerons ainsi dans ces chapitres

une classe d’architectures profondes explicables que l’on nommera simplement architecture profonde bayésienne. Cette classe d’architectures vise à approcher efficacement la structure mathématique d’un détecteur optimal, en imposant que chaque partie du réseau remplisse un rôle particulier dans la prise de décision. Pour la détection de bruit d’hélice en particulier, le détecteur permet d’extraire la cyclostationnarité du signal et base sa décision dessus. Le choix de la structure de ces architectures profondes bayésiennes sera justifié grâce à la théorie de l’approximation, de manière à contrôler leurs complexités et erreurs de généralisation.

Néanmoins, face aux nombreuses contraintes de l’apprentissage machine pratique (algorithme d’optimisation, vanishing gradient, ...), la structure mathématique du détecteur de Bayes peut imposer l’implémentation d’architectures qui ont prouvé être empiriquement mauvaises pour l’apprentissage et ce, malgré de meilleures capacités d’approximation. Un problème notable peut par exemple être une taille de filtres excessifs par rapport aux pratiques habituelles, qu’il faudrait cependant apprendre pour extraire de longues dépendances temporelles ou des signaux haute résolution. En effet, différents rapports sur l’impact des tailles de filtres convolutifs, comme celui fait par [Jiang et al. \(2019\)](#), suggèrent que les modèles disposant de grands filtres ont tendance à être moins performants que ceux incluant des plus petits. Ceci pourrait impliquer que des filtres de la taille du signal discret soient inadéquats pour l’apprentissage d’un modèle dû à une inefficacité en matière de nombre de paramètres et à une fonction de perte moins propice à la convergence vers de bons minima locaux. De plus, comme le stipule [Oyallon \(2017a\)](#), une convolution de n’importe quelle taille peut être réalisée en utilisant sur plusieurs couches des filtres de convolution de tailles réduites. Aussi, pour améliorer les performances de nos architectures profondes bayésiennes, sans augmenter la complexité de leur modèle, au point de perdre la compréhension de leur fonctionnement interne, nous tournerons notre attention vers des méthodes de représentation du signal implémentables dans le cadre de l’apprentissage profond.

3.4 Utilisation de représentations apprises explicables

Une forme d’explicabilité possible pour l’apprentissage profond est de simplement proposer des architectures dont le raisonnement décisionnel pourrait être expliqué aux utilisateurs, comme le suggèrent [Xie et al. \(2020\)](#). Dans ce cadre, une architecture dont les représentations internes sont compréhensibles, se démarquerait déjà des architectures CNN boîtes noires, souvent utilisées avec pour seul objectif d’obtenir de bonnes performances de classification. Les représentations tirées du traitement du signal sont ainsi de bons candidats pour obtenir des structures mathématiques explicables au sein de nos architectures explicables. Nous introduisons ainsi dans le chapitre 7 la scattering transform de [Mallat \(2012\)](#), une transformation non-linéaire consistant à calculer les coefficients spectraux de modulation d’un signal à différents ordres, et notamment une approximation du spectrogramme d’échelle Mel à l’ordre 1. Contrairement aux spectrogrammes linéaires, cette transformée définit une représentation localement invariante aux translations et stable aux déformations temporelles, permettant de réduire les variabilités induites par l’effet Doppler. Les coefficients du second ordre permettent également de caractériser les phénomènes transitoires tels que les attaques et modulation en amplitude, très présentes dans le contexte de la classification acoustique sous-marine et essentiels pour caractériser les signatures de palets.

Les calculs de la scattering transform se faisant à travers des cascades de convolutions d'ondelettes et de l'opérateur de module, cette représentation peut être considérée comme une sous-catégorie de CNN. Comme le suggèrent [Andén & Mallat \(2014\)](#), nous proposons dans le chapitre 7 une adaptation de la scattering transform conservant l'essentiel de sa structure mathématique interne mais basée sur l'apprentissage de différentes ondelettes mères pour obtenir un module de représentation profonde intrinsèquement explicable. Ce module $\mathcal{S}_1[\mathbf{x}](s, f)$ apprend ainsi une banque de petits filtres convolutifs f qui seront appliqués à différentes versions sous-échantillonnées $\mathbf{x}_{\dagger s}$ du signal d'entrée \mathbf{x} à la place d'une famille prédéterminée d'ondelettes basée sur une même ondelette mère, pas forcément adaptées aux données d'entraînement. Ce module, que l'on nomme LST (Learning based Scattering Transform), peut ainsi être vu comme un scalogramme multi canal, où chaque canal est basé sur une différente ondelette mère apprise, d'indice f . Aussi, au lieu de laisser notre réseau apprendre à réaliser des invariances grâce à ses couches de convolutions successives, avec cette approche nous espérons profiter de la connaissance d'une structure mathématique permettant de réaliser ces invariances, pour considérablement réduire le nombre de paramètres et la complexité de notre réseau profond, sans pour autant trop réduire sa capacité d'approximation pour notre problème.

4 Contributions et plan de la thèse

4.1 Contributions

Dans cette sous-section nous résumons brièvement les différentes contributions apportées durant cette thèse :

- Nous effectuons un état de l'art des différentes méthodes d'explicabilité, représentations et architectures communément employées pour la classification acoustique sous-marine.
- Nous faisons un résumé des différentes mesures de complexités permettant de décrire le comportement de généralisation des architectures profondes dans l'apprentissage supervisé.
- Nous proposons un nouveau modèle de bruit de cavitation d'hélice s'efforçant de simuler les contributions individuelles de ses différentes pales.
- Nous proposons un simulateur combinant différents modèles acoustiques et modèles de propagation et l'implémentons.
- Nous dérivons le détecteur de Bayes et certaines variantes de notre modèle acoustique de bruit d'hélice pour différentes hypothèses et l'implémentons pour des simulations de Monte-Carlo.
- Nous proposons l'implémentation d'un module CNN d'approximation de fonction continue.
- Nous proposons une méthode d'explicabilité basée sur des architectures profondes bayésiennes intrinsèquement explicables qui permettent d'apprendre la structure mathématique d'un détecteur de Bayes pour certains problèmes de détection.
- Nous proposons différentes versions d'une implémentation de la structure mathématique de la scattering transform, jusqu'à l'ordre 2, en tant que CNN.

- Nous implémentons différentes architectures citées dans l'état de l'art ainsi qu'une méthode "GridSearch" pour analyser l'influence des paramètres d'architectures profondes.

4.2 Plan de la thèse

Le manuscrit de thèse est organisé tel que suit. Le chapitre 2 effectue un état de l'art sur la classification en acoustique sous-marine et les méthodes d'explicabilité pour l'apprentissage profond. Le chapitre 3 décrit le problème d'apprentissage supervisé en le décomposant en différents types d'erreurs. Il fournit également une liste de différentes bornes ou mesures existant dans la littérature, qui permettent de refléter l'impacte des différents paramètres d'apprentissage des architectures profondes sur leurs performances de généralisation. Le chapitre 4 décrit deux modèles de bruits acoustiques sous-marins, l'un centré sur les bruits de cavitation d'hélice et l'autre plus général et permettant de décrire différentes classes de bruits de navires, bruits parasites et bruit ambiant souvent rencontrés en mer. Le chapitre inclut une description des différentes étapes de simulation pour partir de la sélection des paramètres de la source de type NPM, jusqu'à l'obtention du signal temporel qui prend en compte certains phénomènes de propagation. Le chapitre 5 part du modèle simple de bruit de cavitation d'hélice pour poser un problème de détection binaire. Des tests de Bayes optimaux sont dérivés de ce problème pour différentes hypothèses. Nous proposons par la suite une architecture profonde explicable permettant d'approximer efficacement la structure mathématique de ce détecteur optimal et comparons ses performances à des modèles de comparaison. Le chapitre 6 part une fois de plus du modèle simple de bruit de cavitation d'hélice pour poser cette fois deux problèmes de détection multi-hypothèses. Des tests de Bayes optimaux sont dérivés de ces problèmes et nous proposons d'autres architectures profondes explicables permettant d'approximer leurs structures mathématiques. Le chapitre 7 propose une implémentation de différentes versions de la représentation de scattering transform proposée par [Mallat \(2012\)](#) via une architecture profonde, entraînable de différentes manières. Des expérimentations y sont effectuées pour comparer ses performances à d'autres modèles pour le problème de détection de bruit de cavitation d'hélice multi-hypothèses. Le chapitre 8 propose de tester les performances de différentes architectures profondes citées durant la thèse sur notre jeu de données simulé. Nous proposons également d'évaluer l'influence de différents choix d'architecture et de paramètre d'apprentissage sur les performances d'un CNN 1D. Le chapitre 9 conclut finalement ce manuscrit de thèse et propose certaines perspectives à court terme pour continuer nos travaux, ainsi que des pistes de travaux plus conséquents, qui pourraient améliorer la classification acoustique sous-marine.

Chapitre 2

État de l’art de la classification

1 Introduction

Dans la section 2 de ce chapitre nous effectuons un état de l’art des méthodes de classification en acoustique et acoustique sous-marine, en distinguant plusieurs types d’algorithmes d’apprentissage automatiques et différentes représentations des signaux. La section 3 explore différentes formes d’explicabilité existant dans divers domaines d’application de l’apprentissage profond et conclut quant à une possible utilisation pour l’ASM.

2 Classification acoustique

2.1 Classification ASM traditionnelle

La classification de bruit de navire ASM est généralement considérée comme une tâche difficile à traiter en raison de son aspect crucial pour le bon déroulement d’opérations maritimes. De ce fait, elle exige l’utilisation d’algorithmes robustes pour un haut niveau de fiabilité. Aussi, la classification acoustique sous-marine a historiquement reposé sur l’utilisation de représentations expertes fixes, conjointement à l’écoute d’opérateurs entraînés à cette tâche. L’oreille humaine est en effet bien adaptée à cette tâche, comme cela est décrit en sous-section 3.3 de l’annexe A. L’analyse sonar vient compléter l’écoute par le biais de représentations telles que le LOFAR, un spectre bande étroite utilisé par Zak (2008), ou le DEMON décrit en sous-section 3.2, annexe A. Ces représentations des signaux permettent d’effectuer une analyse des harmoniques et une étude de l’évolution des raies, conjointement avec le reste du contexte opérationnel. Le contexte opérationnel en lui-même est particulièrement important puisqu’il permet de faire des hypothèses sur la nature des sources de bruits, néanmoins il est souvent incompatible, ou tout du moins difficilement retransmissible aux algorithmes d’apprentissage automatique.

Malgré l’importance des opérateurs dans le contexte ASM, il est cependant souhaitable de pouvoir automatiser certaines tâches, en raison de la fatigue des utilisateurs et de la trop grande quantité de données reçues. En outre, si on se réfère aux travaux de Szegedy et al. (2017) dans le domaine de la vision, il est possible d’atteindre, voire de dépasser la précision de classification humaine pour certaines tâches dont le cadre est bien défini. Par la suite nous

effectuons un état de l'art des algorithmes d'apprentissage automatique reposant sur des représentations des signaux, à la fois pour la tâche de classification des signaux acoustiques et d'autres applications proches. En second lieu, les réseaux de neurones profonds ayant la capacité d'apprendre à partir de sources de données abstraites et pouvant apprendre efficacement à partir de données temporelles, nous étudierons leur usage sur des représentations des signaux temporels telles que des spectrogrammes ou coefficients cepstraux de fréquence mel (MFCC). Enfin nous explorerons l'utilisation de réseaux profonds sur signaux acoustiques bruts, une méthode explorée plus en détail durant cette thèse.

2.2 Classification ASM par apprentissage automatique

Pour traiter la tâche de classification sur sonar passif, différentes méthodes de classification par apprentissage automatique ont été proposées, et ce, grâce à une grande variété de techniques d'extraction de caractéristiques et de méthodes de reconnaissance de formes. Ces méthodes d'extraction ont pour but de représenter les signaux sous forme de variables sans dépendances temporelles, afin de pouvoir utiliser des algorithmes d'apprentissage automatique simples tels que des modèles linéaires, comme les SVMs et MLPs, ou plus complexes avec des SVM avec fonction de noyau ou des réseaux de neurones. Dans le domaine spectral, [Soares-Filho et al. \(2000\)](#), utilisent un réseau de neurones pour classifier les bruits rayonnés reçus sur hydrophones grâce aux informations apportées par la moyenne de leurs réponses spectrales. [Azimi-Sadjadi et al. \(2000\)](#), et plus récemment [Shi & Xu \(2013\)](#), utilisent quant à eux une décomposition en paquet d'ondelettes dans laquelle chaque sous-bande, réglée sur une gamme de fréquences, extrait certaines informations tonales du signal en fonction du contenu du signal. Toujours concernant les ondelettes, [Andén & Mallat \(2014\)](#) utilisent les propriétés de la décomposition en ondelettes dans leur Deep Scattering Network afin d'obtenir de nouvelles représentations des données à passer en entrée d'une SVM, mais plutôt pour le domaine de la reconnaissance vocale. [Das et al. \(2013\)](#) utilisent le domaine cepstral, comme caractéristique afin de réduire les effets néfastes des distorsions dues aux trajets multiples dans des eaux peu profondes. La caractéristique cepstrale moyenne permet entre autre de réduire considérablement les effets des trajets multiples variant dans le temps, quand le SNR est élevé. [Sherin & Supriya \(2015b\)](#) utilisent également le domaine cepstral sous forme de MFCC moyennés, et emploient un algorithme génétique pour chercher à optimiser l'apprentissage des paramètres d'optimisation d'une SVM avec fonction de noyau. Ces approches employant des architectures peu profondes sont cependant généralement sujettes à des dégradations de performances et des capacités de généralisations limitées quand appliquées au domaine trop complexe qu'est la classification de signaux sous-marins réels. Ceci peut laisser craindre que les capacités de modélisations de ces architectures ne soient pas capables de donner de bons estimateurs pour des tâches ASM plus complexes.

2.3 Classification acoustique par apprentissage profond

Avec des puissances de calcul toujours plus grandes et accessibles, ainsi que le développement d'algorithmes de parallélisation et d'optimisation capables d'entraîner les CNNs théorisés dès les années 1980 par [Fukushima & Miyake \(1982\)](#), on a pu voir à partir de 2012 l'émergence de l'apprentissage profond avec l'impulsion des travaux de [LeCun et al. \(2015\)](#). Contrairement aux architectures peu profondes, l'apprentissage profond peut générer des ca-

ractéristiques "haut niveau" apprises à partir de combinaisons structurées de caractéristiques "bas niveau", qui sont généralement trop détaillées. Lors de la classification, les couches supérieures permettent alors de renforcer les aspects discriminants des données d'entrée, comme le soulignent Goodfellow et al. (2016). L'utilisation de caractéristiques extraites exploite ainsi le domaine très développé du traitement du signal afin de faciliter l'extraction de caractéristiques informatives dans les signaux. Ces représentations, bien que non adaptatives, permettent de mettre de côté une importante portion des caractéristiques du signal supposées ne présenter aucune information utile pour la classification des données, ce qui peut permettre de faciliter l'apprentissage de modèles complexes si une représentation adaptée au signal a bien été choisie. Dans ses travaux, Abeßer (2020) effectue une liste assez complète des représentations utilisées dans la classification de scènes acoustiques et comprenant entre autres la transformée de Fourier à court terme, le spectrogramme Mel, les MFCCs, les scalogrammes, la scattering transform ainsi que les gammatones. D'autres représentations peuvent également être utilisées comme des moyennes temporelles basées sur des algorithmes de computer vision tels que les modèles binaires locaux (LPB) ou l'histogramme de gradients orientés (HOG).

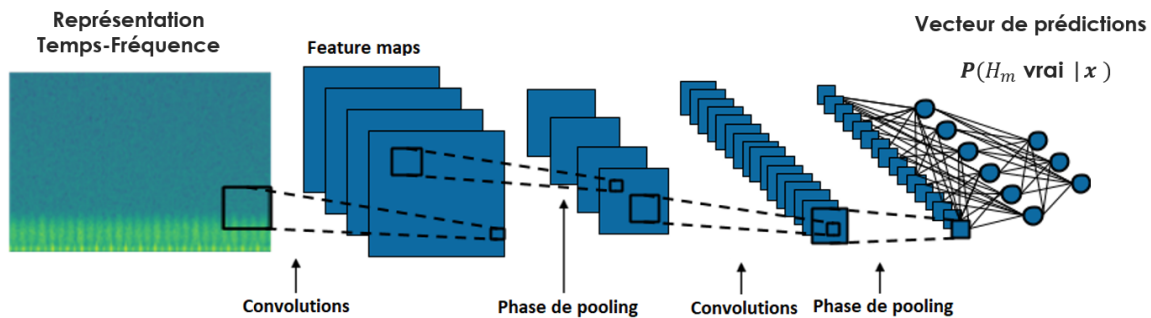


Figure 2.1: Application d'un CNN 2D sur une représentation temps-fréquence $x(t, f)$. Les couches de neurones complètement connectées prennent en entrée les caractéristiques "haut niveau" extraites par les deux couches successives de convolution du modèle afin de prendre une décision sur la classe H_m du signal.

Parmi les différentes applications acoustiques possibles, la classification de sons environnementaux peut présenter des similarités avec l'acoustique sous-marine au regard de la nature de ses sources. Néanmoins, la nature des bruits ambiants environnementaux et leurs intensités sont bien plus éloignées. Les jeux de données ESC-10 et ESC-50 proposés par Piczak (2015b) et, dans une moindre mesure, la base de données TUT proposée par de Mesáros et al. (2016) au sein de la compétition DCASE (Challenge on Detection and Classification of Acoustic Scenes and Events), sont ainsi de bonnes références pour comparer les performances d'algorithmes profonds sur données acoustiques. La majorité des représentations fixes se classant haut dans ces compétitions acoustiques utilisent des Spectrogramme Mel ou des variantes de cette représentation. Sur le jeu de données urbain ESC-50, Piczak (2015a) utilise par exemple un spectrogramme Mel en entrée d'un CNN 2D, comme celui présenté sur la figure 2.1, en argumentant que cela fait correspondre la distribution des réponses des filtres du spectrogramme à celle de l'oreille humaine. Le delta du spectrogramme, i.e. la différence entre les FFT suivantes dans le spectrogramme est également utilisé en entrée du modèle pour capter des modulations dans les signaux. Cette méthode établie une référence de base pour la compétition ESC-50

avec 64,7 % de précision devant toutes les méthodes d'apprentissage machine, mais derrière l'oreille humaine à 81.3 % de précision. Beaucoup de travaux effectuent également une forme de pré-apprentissage ou transfer-learning sur des jeux de données annexes plus riches pour obtenir de meilleures performances à l'image de [Kumar et al. \(2018\)](#) qui obtiennent 83.5 % de précision toujours sur ESC-50 avec des CNN 2D. Par la suite, [Lopez-Meyer et al. \(2021\)](#) comparent différentes formes de pré-apprentissage pour obtenir jusqu'à 92,32 % sur ESC-50. D'autres méthodes peuvent également être utilisées pour améliorer les performances d'une architecture, comme effectuer certaines formes de Data augmentation ou une forme de vote, avec par exemple plusieurs modèles entraînés à partir d'une même architecture. [Nanni et al. \(2020\)](#) montrent ainsi une amélioration de performance significative avec ces méthodes en obtenant 88.65 % sur ESC-50. L'utilisation de plusieurs spectrogrammes, avec des résolutions différentes en entrée du réseau, peut également être une façon d'améliorer les performances de classification comme le montrent [Espí et al. \(2015\)](#). En effet, comme cela est indiqué dans la section 4 de l'annexe A, l'utilisation de fenêtres de convolution larges dans les FTT permet d'obtenir une forte résolution spectrale, au détriment du nombre d'échantillons dans le domaine temporel et inversement. Différentes résolutions temps-fréquence permettent ainsi de chercher plus précisément certaines raies ou des motifs temporelles plus fins comme des transitoires.

Bien que ce soit pour un domaine assez différent de l'ASM, beaucoup de travaux ont démontré l'utilité des MFCCs dans le domaine de la parole. [Hinton et al. \(2012\)](#) utilisent par exemple des MFCCs en combinaison avec des PLPs en entrée d'un processus de Markov caché (HMM) suivie d'un GMM, qu'ils notent GMM-HMM. Les approches séquentielles telles que les HMM et RNN semblent cependant être moins efficaces dans des domaines d'application où de forts niveaux de bruits ambiants et ou parasites sont présents. Dans ce sens, [Zhang et al. \(2015\)](#) dénotent qu'en comparant l'utilisation de MFCCS avec HMM par rapport à des SIFs et SAIs avec CNN 2D, comme proposé par [McLoughlin et al. \(2015\)](#), cette première méthode est sujette à des pertes de performances importantes en présence de bruits parasites pour la classification de sons environnementaux. Expérimentalement, en utilisant des SIFs et CNN pour un SNR de 0 dB, ils obtiennent de meilleures performances de classifications comparativement à d'autres méthodes classiques comme des MFCC avec SVM, Gabors, MFCC et MLP, et SIF avec MLP.

2.4 Classification ASM par apprentissage profond

Parallèlement aux avancées dans les domaines acoustiques, des travaux dans le domaine de l'ASM ont commencé à explorer l'utilisation de représentations temps-fréquence, temps-cepstre ou temps-fréquence-cyclique en combinaison avec des CNNs 2D, comme le font [Yue et al. \(2017\)](#), [Wu et al. \(2018\)](#) et [Valenti et al. \(2017\)](#), ou alternativement avec d'autres architectures profondes telles que des DBNs, à la manière de [Kamal et al. \(2013\)](#) et [Yue et al. \(2017\)](#). Contrairement à la détection d'événements acoustiques urbains, il existe cependant très peu de jeux de données sur des bruits de navires labélisés et leurs tailles sont souvent insuffisantes pour entraîner des architectures profondes. Ceci est souvent dû à des problèmes de confidentialité, le jeu de données "Historic Naval Sound and Video data base" par exemple, ne comprend que des enregistrements acoustiques de navires militaires décommissionnés datant ainsi de plus de 30 ans. Du peu de résultats qu'on peut en tirer, il semble cependant que

l'utilisation d'approche temps fréquence apporte de meilleurs résultats que des approches sur MFCCs, comme en attestent [Yue et al. \(2017\)](#).

La combinaison de plusieurs représentations expertes s'avère cependant être une possibilité pour améliorer la précision de classification. En effet on peut alors concaténer des caractéristiques extraites complémentaires sur différents canaux des données d'entrée. [Cao et al. \(2019\)](#) utilisent par exemple une combinaison de spectrogramme Mel, de composantes énergétiques obtenues à partir de paquets d'ondelettes, ainsi qu'une estimation des fréquences de rotation des hélices, pour obtenir de meilleurs résultats qu'avec simplement un spectrogramme Mel obtenu dans leurs précédents travaux [Cao et al. \(2016\)](#) sur le jeu de données utilisé dans [Kamal et al. \(2013\)](#). Cette amélioration des résultats peut être attribuée au fait que les caractéristiques non apprises peuvent écarter certaines propriétés significatives du signal acoustique. Aussi chaque transformation nécessaire à l'obtention de caractéristique "haut niveau" ne capture qu'un aspect spécifique des propriétés du signal acoustique. Le rôle des architectures profondes est ainsi d'intégrer l'ensemble de ces représentations avec des structures mathématiques complexes, afin de combiner leurs avantages sans pour autant accumuler leurs défauts. [Hong et al. \(2021\)](#) utilisent quant à eux une architecture ResNet, comportant 2 branches disposant respectivement d'une et trois couches de convolution, sur une agrégation de MFCCs, Spectrogramme en échelle Mel et CCTZ (Chroma, Contraste, Tonnetz et Zero-cross) sur 3 canaux. Cette méthode est testée sur le jeu de données ShipsEar, mis en place par [Santos-Domínguez et al. \(2016\)](#) et comportant 91 enregistrements de 11 types de navires regroupés en 4 classes ainsi qu'une classe de bruits ambiants. Aussi, comparativement aux précédentes approches proposées par d'autres auteurs, [Hong et al. \(2021\)](#) parviennent à obtenir de meilleures performances avec 94,3 % de précision de validation. Ces bons résultats, obtenus avec des représentations de faibles dimensionnalités ($\approx (60, 40)$) comparés aux signaux à haute résolution temporelle, semblent indiquer que l'approche consistant à utiliser des représentations fixes, mais choisies de façon pertinente, est une bonne solution pour classifier des signaux pour des jeux de données comportant relativement peu d'exemples d'entraînement.

2.5 CNN 1D sur signaux acoustiques brutes

Bien qu'il existe de nombreux travaux concernant l'utilisation d'architectures profondes pour la reconnaissance acoustique, reprenant souvent des architectures utilisées dans le domaine des images, bien moins de travaux se focalisent sur l'utilisation d'architectures prenant en entrée des signaux acoustiques bruts. Un modèle profond capable d'apprendre des structures internes à partir de signaux acoustiques, demande en effet une architecture plus complexe afin de pouvoir apprendre une représentation interne capable d'extraire les informations fournies par des représentations fixes. Les architectures profondes étant des algorithmes d'apprentissage de bout en bout, ou pourrait cependant s'attendre à ce que leur utilisation sur des signaux acoustiques bruts puisse nous permettre d'obtenir de meilleurs résultats qu'avec des représentations fixes, pas forcément adaptées au problème posé. L'utiliser d'architectures de CNN 1D est ainsi une possibilité pour classifier les signaux acoustiques. Dans la figure 2.2 nous décrivons une architecture CNN 1D disposant de $L = 2$ couches convolutives, ainsi que d'une couche de neurones cachée et une couche de neurones de sortie. En considérant que le zero-padding est utilisé, chaque couche de convolution 1D prend en entrée une représentation intermédiaire ou ensemble de feature maps $X^{(l)} \in \mathbb{R}^{K_l \times F_l}$ et ressort un ensemble de feature

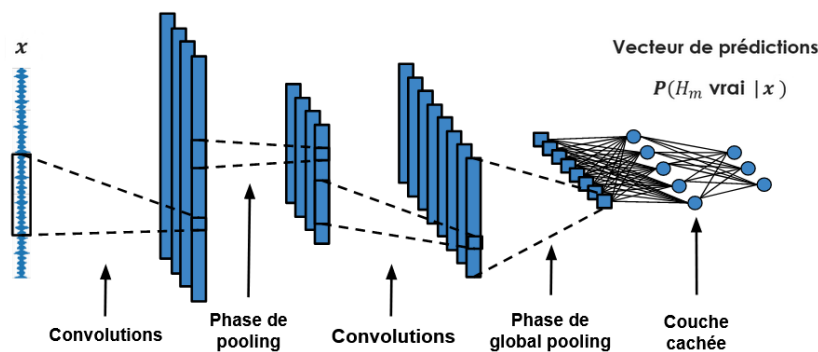


Figure 2.2: Application d'un CNN 1D sur un signal acoustique brut x . Les couches de neurones complètement connectées prennent en entrée les caractéristiques "haut niveau" extraites par les deux couches successives de convolution temporelles du modèle afin de prendre une décision sur la classe H_m du signal.

maps $Y^{(l+1)} \in \mathbb{R}^{K_l \times F_{l+1}}$ en effectuant une convolution temporelle avec des filtres de convolution $W^{(l+1)} \in \mathbb{R}^{p_{l+1} \times F_l \times F_{l+1}}$ et un biais $b^{(l)} \in \mathbb{R}^{1 \times F_{l+1}}$. Une fonction d'activation non linéaire $\rho(\cdot)$ ainsi qu'une opération de pooling sont ensuite appliquées au signal $Y^{(l+1)}$:

$$Y_j^{(l+1)} = W_j^{(l+1)} * X^{(l)} + b_j^{(l+1)} \in \mathbb{R}^{K_l}, \forall j = 1, \dots, F_{l+1} \quad (2.1)$$

$$Z^{(l+1)} = \rho(Y^{(l+1)}) \in \mathbb{R}^{K_l \times F_{l+1}} \quad (2.2)$$

$$X^{(l+1)} = \text{pool}(Z^{(l+1)}) \in \mathbb{R}^{K_{l+1} \times F_{l+1}}, \quad (2.3)$$

avec $Y^{(l+1)} = \{Y_1^{(l+1)}, \dots, Y_{F_{l+1}}^{(l+1)}\}$, $b^{(l+1)} = \{b_1^{(l+1)}, \dots, b_{F_{l+1}}^{(l+1)}\}$ et $\text{pool}(\cdot)$ une opération de pooling effectuant une agrégation locale de l'information à partir d'un opérateur max, average ou une combinaison des deux. La première représentation $X^{(0)}$, qui doit être traitée comme un ensemble de feature maps, est simplement le signal acoustique x qui doit avoir une dimension $K \times 1$ si le signal ne dispose que d'un seul canal. Après la $L^{\text{ème}}$ couche de convolution, une opération de pooling globale est employée pour obtenir un signal $X^{(L+1)} \in \mathbb{R}^{1 \times F_L}$ sous la forme d'un ensemble de variables qui seront l'entrée de couches de neurones complètement connectées.

Les CNNs 1D ont été utilisés dans de nombreux domaines d'application tels que l'analyse de séries temporelles, la détection d'anomalies avec [Zhang et al. \(2018\)](#), l'estimation de fréquences avec [Izacard et al. \(2019\)](#) ou encore les communications électromagnétiques avec les travaux de [O'Shea et al. \(2016\)](#) et acoustiques avec les travaux de [Kozlenko et al. \(2020\)](#). Parmi les différents travaux sur l'utilisation de CNN 1D en acoustique, [Dai et al. \(2017\)](#) mettent en avant que les performances de ce type d'architecture s'améliorent nettement avec le nombre de couches. En particulier, ils proposent d'utiliser des FCNNs, des architectures CNNs ne disposant pas de couches cachées, mais seulement d'une couche de neurones de décision qui prend en entrée le résultat d'une opération average pooling globale appliquée sur la dernière couche de convolution. Durant les tests de performances de ces réseaux très profonds, employant jusqu'à 34 couches, [Dai et al. \(2017\)](#) constatent cependant que la précision ne s'améliore plus après 18 couches. Ceci semble impliquer qu'en pratique, des effets tels que le vanishing gradient rendent l'entraînement de ces architectures très profondes problématiques

à partir d'un certain nombre de couches, un constat partagé avec d'autres types d'architectures profondes. De même, les travaux de [Golik et al. \(2015\)](#) montrent qu'un CNN 1D avec une seule couche de convolution ne parvient pas à obtenir les performances de classification d'un CNN 1D disposant de plus d'une couche convolutive et une couche de neurones cachée et ce, quel que soit le nombre de couches cachées utilisées après cette unique couche de convolution.

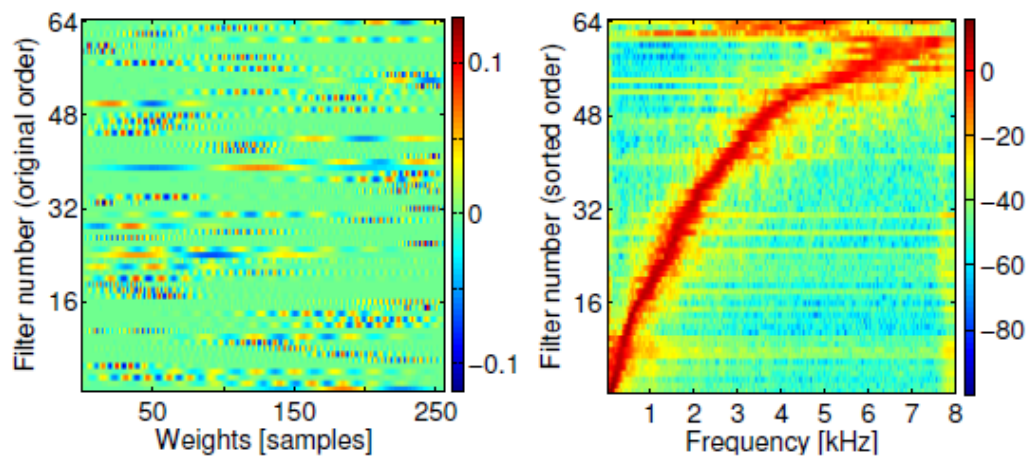


Figure 2.3: Motifs appris par une première couche de convolution. Tiré de l'article de [Golik et al. \(2015\)](#).

La conception d'architectures de CNN 1D semble également converger vers l'utilisation de grands filtres de convolution sur la première couche convolutive, comme le font remarquer [Dai et al. \(2017\)](#), [Golik et al. \(2015\)](#), [Sailor et al. \(2017\)](#). [Zhu et al. \(2018\)](#) proposent également d'utiliser des tailles de filtres différents sur la première couche, afin d'apprendre des formes d'ondes de tailles différentes. Ce type d'architecture semble alors faciliter l'apprentissage d'un banc de filtres adapté au sein de cette première couche de convolution. Les travaux de [Golik et al. \(2015\)](#), dans le domaine de l'acoustique aérienne, se focalisent justement sur les caractéristiques de ces filtres appris. Ils dénotent ainsi qu'environ 95 % des filtres apprennent une banque de filtres passe-bande relativement étroite et distribuée de façon logarithmique sur l'axe des fréquences, comme on peut le voir sur la figure 2.3. La première couche de convolution semble ainsi apprendre une décomposition temps fréquence similaire à un spectrogramme d'échelle Mel, un comportement retrouvé sur nos jeux de données acoustiques simulées.

Pour analyser la seconde couche de convolution, [Golik et al. \(2015\)](#) réordonnent l'index des filtres de la première couche de convolution par réponse fréquentielle croissante, ce qui fait que les canaux voisins ont des fréquences centrales voisines. Parmi les différents motifs appris par la seconde couche de convolution visualisés sur la figure 2.4, certains correspondent à de simples stationnarités en temps ou encore à une non-stationnarité en temps et fréquence. Cependant, certains filtres semblent effectuer un filtrage de type Gaussien sur certaines bandes de fréquences, ce qui commence à ressembler à des traitements classiquement effectués sur spectrogrammes. Enfin, il est également commun pour ce type de méthode d'effectuer un

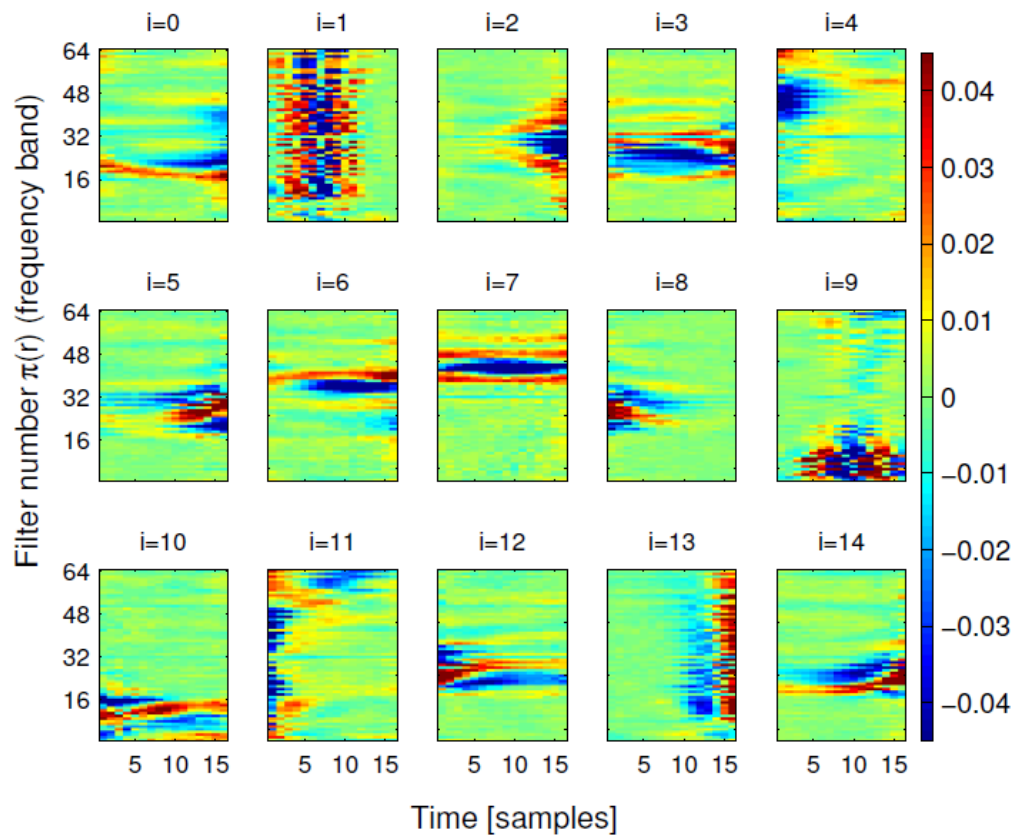


Figure 2.4: Motifs appris par une seconde couche de convolution après avoir réordonné les filtres de la première couche par réponse fréquentielle croissante. Tiré de l'article de [Golik et al. \(2015\)](#).

pré-apprentissage sur des jeux de données plus larges comme Audioset de [Gemmeke et al. \(2017\)](#) et ce, afin d'apprendre des filtres "bas niveau" pouvant être efficaces pour extraire des caractéristiques utiles pour une large variété de signaux.

2.6 Variantes des CNN 1D

Afin d'obtenir de meilleures performances qu'avec l'emploi de CNN 1D, de nombreux travaux ont cherché à apprendre des bancs de filtres 1D en proposant des variantes de l'algorithme d'apprentissage ou de la structure même du CNN 1D. Les travaux de [Tokozume et al. \(2017\)](#) et [Zhu et al. \(2018\)](#), proposent ainsi d'utiliser des filtres de convolutions temporelles 1D, suivis par des couches de convolutions 2D. Les feature map 1D multicanal sont alors redimensionnés en représentation 2D en transformant les canaux, liés aux réponses des différents filtres de convolution 1D, en un nouvel axe en plus du temps. Des architectures CNN 2D, applicables sur des représentations temps-fréquence, peuvent alors être utilisées sur les sorties des premières couches de convolution 1D. [Zhu et al. \(2018\)](#) proposent d'ailleurs de concaténer un spectrogramme Mel dans un second canal, après avoir obtenu cette représentation 2D apprise à partir de filtres 1D. Cette méthode leur permet d'obtenir de meilleurs

résultats qu'avec un spectrogramme Mel ou des couches de convolutions 1D seules, montrant ainsi qu'il peut y avoir une complémentarité entre ces méthodes. L'utilisation de filtres 2D permet alors de limiter le nombre de paramètres dans l'architecture en évitant de connecter toutes les réponses des filtres 1D sur différents canaux aux couches suivantes. Le réseau peut ainsi apprendre un nombre limité de corrélations entre les canaux des feature map en sortie des couches successives. Les auteurs soulignent ainsi, que des filtres de convolution d'index proches ont souvent des réponses fréquentielles similaires.

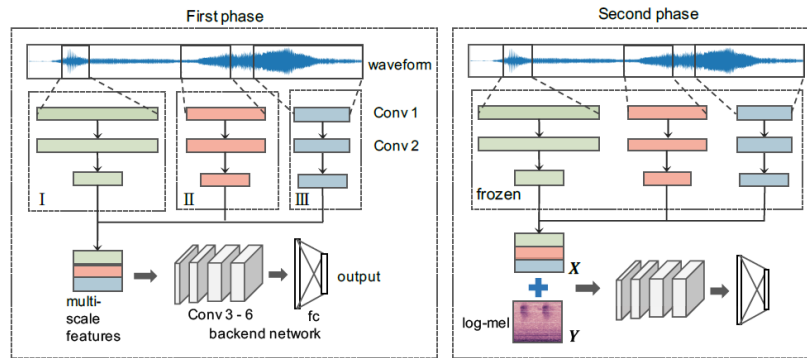


Figure 2.5: Entraînement en deux phases d'un CNN 1D et 2D concaténant représentations apprises par un CNN 1D et spectrogramme Mel. Tiré de l'article de [Zhu et al. \(2018\)](#)

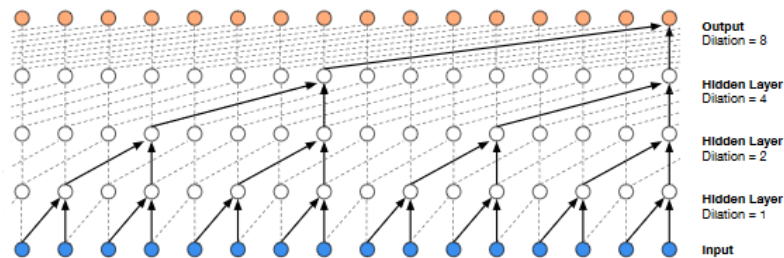


Figure 2.6: Architecture d'un CNN comportant des couches de convolution dilatées. Tiré de l'article de [Oord et al. \(2016\)](#)

Une autre possibilité est l'emploi de convolutions dilatées, une méthode proposée par [Oord et al. \(2016\)](#), et capable d'analyser de très grandes sections de signaux, comme le font [Li, Liu, Drossos & Virtanen \(2020\)](#) dans le domaine de la détection d'évènements acoustiques. Dans ces convolutions dilatées ou "à trous", visualisables sur la figure 2.6, les filtres sont appliqués sur un intervalle plus large que leur longueur en effectuant des sauts avec un pas de taille fixe. C'est ainsi une opération équivalente à une convolution avec un filtre plus large et comportant des zéros, mais nettement plus efficace en terme de coût de calculs.

Alternativement, [Sailor et al. \(2017\)](#) proposent d'employer une machine de Boltzmann restreinte convolutive (CRBM), pour apprendre une banque de filtres de façon non-supervisée. Dans leurs travaux, la CRBM apprend dans un premier temps à modéliser des signaux acous-

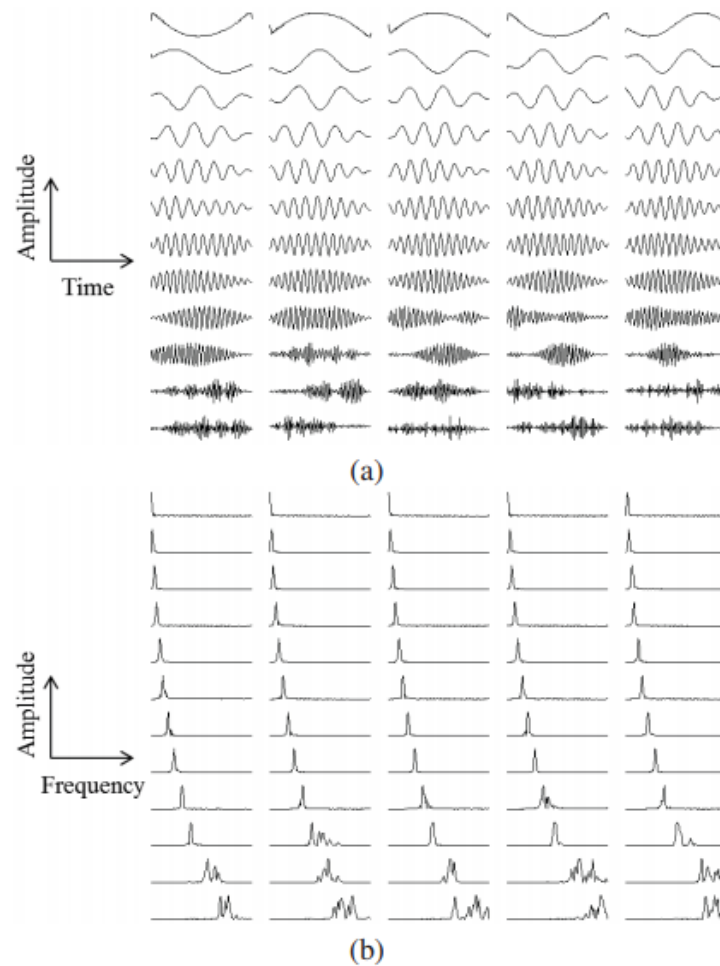


Figure 2.7: Exemples de filtres sous-bandes appris par un modèle de CRBM dans le domaine temporel en haut, et fréquentiel en bas. Tiré de l'article de [Sailor et al. \(2017\)](#).

tiques avec un banc de filtre de convolution 1D, sur le jeu de données ESC-50. Ces filtres appris sont ensuite réutilisés au sein d'un CNN 1D pour un apprentissage supervisé permettant d'obtenir 86.5 % de précision sur le jeu de données ESC-50. Concernant les filtres appris avec cette méthode, que l'on peut visualiser sur la figure 2.7, beaucoup peuvent être assimilés à des atomes de la transformée de Fourier, mais certains sont similaires à des fonctions de type gammatone, notamment à basse fréquence. La distribution des réponses fréquentielles des filtres appris est là aussi similaire à une échelle Mel, similaire à la perception humaine. [Sailor et al. \(2017\)](#) remarquent cependant qu'il y a une densité très importante de filtres entre 1.5 et 8 kHz, gamme de fréquences correspondant aux vocalises d'animaux présents dans le jeu ESC-50. La CRBM permet ainsi d'obtenir une importante résolution spectrale dans des gammes de fréquences hautement caractérisantes du jeu de données.

Du côté des modèles séquentiels, les travaux de [Cakır et al. \(2017\)](#) et [Sang et al. \(2018\)](#)

sont centrés autour de l'utilisation de réseaux de neurones convolutifs récurrents (CRNN), afin d'apprendre de longues dépendances temporelles. C'est en effet un problème pouvant apparaître avec l'utilisation de CNN 1D sur des signaux haute résolution, ou avec l'utilisation de CNN 2D sur des représentations de résolutions temporelles bien plus élevées que celle du second axe. Aussi, avec cette approche, les couches convolutives permettent d'extraire des caractéristiques haut niveau dépendant du temps, tandis que les couches récurrentes du modèle permettent d'effectuer une agrégation efficace des informations temporelles extraites par ces filtres convolutifs. Ces méthodes séquentielles globales peuvent ainsi venir remplacer les opérations de pooling globales, pour apprendre des caractéristiques plus élaborées quand il est nécessaire de détecter plusieurs patterns sur de très longues durées.

Enfin, plus récemment, les différents mécanismes d'attention ont permis d'encore améliorer les performances de classification des architectures profondes, en mettant plus à profits les méthodes séquentielles. L'utilisation de auto-attention en sortie de CNN, comme le font [Kong et al. \(2020\)](#), a notamment permis de redéfinir l'état de l'art dans la classification d'événement acoustiques. De même, [Gong et al. \(2021\)](#) et [Kumar & Ithapu \(2020\)](#) parviennent à obtenir les deux meilleurs scores sur le jeu de données ESC-50 avec des mécanismes d'attention en obtenant 95.7 % et 94.1 % de précision, en utilisant respectivement en entrée des images de spectrogrammes et des données acoustiques brutes. Les mécanismes d'attention peuvent être utilisés de plusieurs manières possibles en raisonnant de façon locale ou globale, ou bien en considérant des échantillons temporels ou les différentes caractéristiques d'une feature map comme des états au sein d'une séquence. [Yang et al. \(2019\)](#) proposent par exemple un CNN 1D disposant d'une couche d'auto-attention de type produit scalaire simple, afin de pondérer toutes les feature maps provenant d'une première couche de convolution 1D. En notant $Z^{(L)} \in \mathbb{R}^{F \times K}$ le vecteur de caractéristiques en sortie de la $L^{\text{ème}}$ couche, un vecteur caractéristique d'agrégation $\mathbf{c} \in \mathbb{R}^{F \times 1}$ est calculé tel que :

$$\mathbf{b} = \rho(\mathbf{w}Z^{(l)}), \quad (2.4)$$

$$\alpha_i = \frac{e^{b_i}}{\sum_{j=1}^K e^{b_j}}, \quad (2.5)$$

$$\mathbf{c} = Z^{(l)}\alpha, \quad (2.6)$$

où $\mathbf{w} \in \mathbb{R}^{1 \times F}$ est un vecteur de paramètres entraînaables, $\rho(\cdot)$ est une fonction d'activation et $\alpha = \{\alpha_1, \dots, \alpha_K\}$ donne les poids d'attention à appliquer à chaque canal ou état $Z_f^{(L)}$ de $Z^{(L)}$. Comme le font [Kumar & Ithapu \(2020\)](#), il est également possible d'apprendre une matrice de paramètres entraînaables $W \in \mathbb{R}^{F \times F}$ pour apprendre $\mathbf{c} \in \mathbb{R}^{F \times 1}$ comme suit :

$$A = \rho(WZ^{(l)}), \quad (2.7)$$

$$O = Z^{(l)} \odot A, \quad (2.8)$$

$$\mathbf{c} = \sum_{k=1}^K O_k, \quad (2.9)$$

où $O = \{O_1, \dots, O_K\} \in \mathbb{R}^{F \times K}$, \odot est le produit terme à terme et A donne les poids d'attention à appliquer à $Z^{(L)}$.

Les 'transformers', présentés pour la première fois par [Vaswani et al. \(2017\)](#), sont quant à eux constitués de plusieurs couches d'attention successives, où chaque état d'une couche prend les informations de tous les états de la couche précédente. Par conséquent, chaque état conserve les informations globales de la séquence d'entrée. Les mécanismes d'auto-attention, souvent utilisés au sein des 'transformers', regrouperont eux les types de mécanismes d'attention permettant de relier différentes positions d'une même séquence afin de calculer une nouvelle représentation de cette séquence i.e. les trois entrées d'une couche d'attention, les clés, valeurs et requêtes sont identiques.

2.7 Classification ASM sur signaux brutes

La classification de signaux acoustiques sous-marins bruts, sous-entendu sans utilisation de représentations expertes, est un sujet bien moins exploré dans la littérature comparé aux méthodes décrites précédemment. Bien que ce ne soit pas directement sur un signal acoustique brut, [Wang et al. \(2021\)](#) utilisent par exemple un CNN 1D sur une représentation DEMON de signaux ASM pour la classification de navires. Dans leurs travaux, [Berg & Hjelmervik \(2018\)](#) et [Hu et al. \(2018\)](#) utilisent des CNN 1D sur signaux acoustiques bruts et parviennent à montrer expérimentalement que ces architectures peuvent bien être utilisées pour extraire des caractéristiques ASM. [Hu et al. \(2018\)](#) obtiennent par exemple de meilleurs résultats avec cette approche comparé à l'utilisation de MFCCs et HTTs. [Tian et al. \(2021\)](#) utilisent quant à eux une architecture CNN 1D inspirée de la structure de l'architecture ResNet, qu'ils nomment CNN multi-échelles résiduel.

2.8 Structures de décisions alternatives

L'apprentissage profond est généralement utilisé pour sa capacité à apprendre des représentations basées sur une hiérarchie de caractéristiques informatives et ce, à partir d'un très grand nombre de données. Au sein de l'apprentissage supervisé, l'emploi d'une couche de neurones de décision est cependant nécessaire pour permettre un apprentissage par descente de gradient. En notant $Z^{(L)} \in \mathbb{R}^{F \times 1}$ le vecteur de caractéristiques de la couche L d'un réseau profond et donc sa représentation apprise, on considère la couche de décision de ce réseau avant seuillage comme étant le vecteur d'éléments

$$[f_{\theta}(\mathbf{x})]_m = W_m \cdot Z^{(L)} + b_m, \quad \forall m = 1, \dots, M \quad (2.10)$$

avec des coefficients W_m appartenant tous à $\mathbb{R}^{F \times 1}$, des biais $b_m \in \mathbb{R}$ et M le nombre de classes. Il n'est cependant pas nécessaire de conserver cette couche de décision pour exploiter la représentation apprise par les L premières couches du modèle appris.

Il est en effet possible d'utiliser un modèle de mixture gaussienne (GMM) pour remplacer la couche de sortie d'un réseau de neurones appris. De cette manière, au lieu de classifier les signaux d'entrée \mathbf{x} en fonction de leur position par rapport aux hyperplans définis par les neurones de chaque classe, les entrées sont classifiées par rapport à leur distance au centre de la distribution définie par chaque groupe au sein du GMM. Un mélange gaussien est une fonction composée de plusieurs gaussiennes identifiées par leur indice $c \in 1, \dots, C$, où le nombre de clusters C peut différer du nombre réel de classes M dans l'ensemble de données, comme cela

est illustré dans la figure 2.8. Chaque gaussienne c dans le mélange est composée d'une probabilité π_c , d'une moyenne μ_c ainsi que d'une matrice de covariance Σ_c , qui seront appris via un algorithme espérance-maximisation de façon non supervisée, bien que certaines variantes supervisées existent. Cette méthode peut ainsi permettre d'éviter de classifier des données

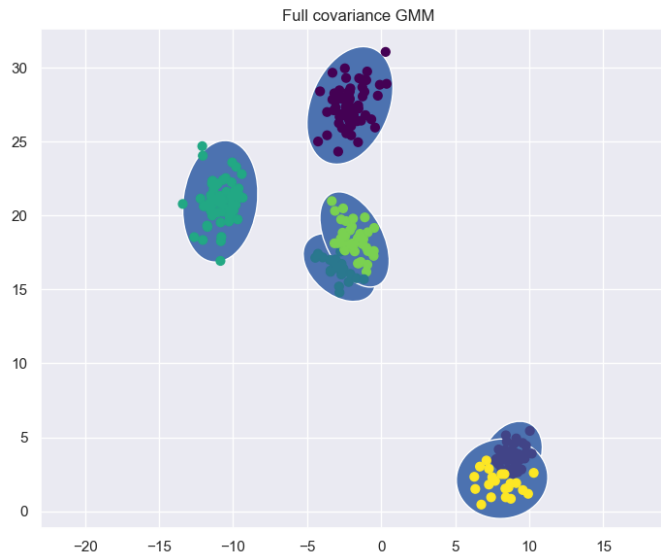


Figure 2.8: Utilisation d'un GMM avec $C = 6$ clusters sur $M = 4$ groupes de points suivants des distributions Gaussiennes multivariées de centres différents et de mêmes matrices de covariances.

éloignées des données d'entraînement, si ces dernières sont également éloignées dans l'espace appris par le réseau de neurones $Z^{(L)}$. Dans un espace de grande dimension, la malédiction de la dimensionnalité tend en effet à rendre insignifiant le volume représenté par les groupes définis par les GMMs. Aussi, bien qu'il n'y ait pas de certitudes quant au fait que la représentation apprise permet d'ignorer des classes de signaux non envisagées, cela peut être vu comme une avancée en terme de fiabilité par rapport à une décision basée sur un hyperplan.

Le fait de changer la structure de la fonction de décision, qui passerait dans ce cas d'un ensemble d'hyperplans, à des groupes locaux, est cependant un défaut important de cette approche. Les couches de convolutions et les couches cachées sont effet entraînées pour une optimisation à partir de cette structure de sortie. Une solution plus élégante est alors d'apprendre directement le modèle profond avec une fonction de décision ainsi qu'une fonction de perte alternative. Cai et al. (2018) proposent par exemple d'apprendre de bout en bout des centres de distribution de classes ainsi que des couches convolutives, grâce à une couche d'encodeur de dictionnaire entraînable et une fonction de perte \mathcal{L} , inspirée de la procédure d'entraînement des GMMs.

Il existe également d'autres structures de décision possibles, comme les réseaux bayésiens ou la structure de marge présente dans les SVMs. Enfin, il est possible d'utiliser la représen-

tation apprise, afin de classifier une entrée $\mathbf{x}^{(i)}$ en cherchant son plus proche voisin au sein d'une base de données, à travers une distance donnée.

3 Formes d'explicabilité dans l'apprentissage profond

3.1 Définition

Les récentes avancées en l'intelligence artificielle et nous concernant, en apprentissage profond, ont abouti à son adoption dans un grand nombre de domaines d'application où ses performances peuvent redéfinir l'état de l'art. Néanmoins, ces meilleures performances s'accompagnent généralement d'une complexification des modèles, tandis que, dans une application critique pour la sécurité telle que l'ASM, aider un utilisateur plutôt que d'être un inconvénient, nécessite de la fiabilité dans les classifications du modèle profond. En effet, cet ajout de complexité transforme les algorithmes de classification en boîtes noires, entraînant ainsi des problèmes pour savoir comment l'algorithme opère pour prendre une décision, que ce soit pour une donnée particulière ou son fonctionnement général. Cette problématique ouvre la voie à un nouveau domaine d'algorithmes d'apprentissage XAI, recentré sur des méthodes permettant d'expliquer ou d'interpréter le fonctionnement d'un modèle. Pour notre cas, nous chercherons plus spécifiquement des méthodes permettant d'éviter des algorithmes fonctionnant comme des boîtes noires et ce, même si leur décisions peuvent être interprétées. Qui plus-est, il sera nécessaire de trouver un bon compromis entre les performances d'un modèle d'apprentissage et sa capacité à produire des prédictions explicables et interprétables, sans forcément opposer les algorithmes d'apprentissage profond, vus comme des boîtes noires (CNN, Random forest, ...) et des algorithmes plus simples, pouvant être considérés comme des boîtes blanches (modèle linéaire, arbre de décision, SVM sans noyau, ...).

Une définition globalement acceptée et donnée par [Doshi-Velez & Kim \(2017\)](#), du terme d'interprétabilité dans le domaine de l'IA, le définit comme étant "la capacité d'expliquer ou de présenter en termes compréhensibles à un humain". Qui plus est, bien qu'il soit difficile de donner une définition mathématique à ce terme, des tentatives ont été faites par [Adadi & Berrada \(2018\)](#) pour refléter le fait que, plus un modèle est interprétable, plus il est facile d'identifier les relations de cause à effet concernant une décision prise sur une entrée. Dans ce sens, la fiabilité pourrait concerner l'interprétation de la raison d'une décision à travers différentes méthodes de visualisation, même si l'algorithme est considéré comme une boîte noire. Selon [Linardatos et al. \(2021\)](#), l'explicabilité est quant à elle associée à la logique et aux mécanismes internes d'un algorithme d'apprentissage automatique : plus un modèle est explicable, plus un être humain peut comprendre les opérations internes ayant lieu pendant que le modèle s'entraîne ou prend des décisions. Un modèle interprétable n'induit ainsi pas nécessairement qu'un humain soit capable de comprendre sa logique interne ou ses processus sous-jacents. Aussi, puisque les réseaux profonds ne sont pas intrinsèquement explicables en raison de leur cascade d'opérations internes, justifier de leur utilisation au sein de l'ASM est une tâche périlleuse. Une structure mathématique interne interprétable pourrait en revanche apporter une explication de fonctionnement satisfaisante et ainsi être une forme d'explicabilité adaptée à notre cas.

Dans leurs travaux de synthèse, [Xie et al. \(2020\)](#) voient l'explicabilité comme un problème

commun à l'utilisation d'algorithmes d'apprentissage machine, bien que ces algorithmes soient souvent des approches génériques dont l'explicabilité n'est pas forcément adaptée pour une situation précise. Chaque domaine exige en effet certaines caractéristiques d'une architecture explicable. Cela peut correspondre à la possibilité de visualiser des activations internes lors d'une décision, d'effectuer des mesures associées à un trait spécifique, ou encore de passer une évaluation par être humain lors d'études d'utilisabilité. Pour [Xie et al. \(2020\)](#), l'explicabilité qui peut être associée à la transparence, à la robustesse ou à la fiabilité, n'est pas intrinsèquement présente dans les architectures profondes. En effet un utilisateur ne peut comprendre les informations émises par un algorithme qui compresse les données d'entrée en une unique valeur par une cascade de sommes pondérées et d'opérations non linéaires. Une telle décision n'est pas fondée sur des évidences raisonnables. Pour croire en une décision il faut pouvoir se reposer sur des raisons qui puissent :

- Être facilement explicables.
- Être pertinentes pour l'utilisateur.
- Connecter la décision à des informations contextuelles proches d'anciens choix ou expériences d'utilisateurs.
- Refléter le raisonnement intermédiaire utilisé par l'utilisateur dans sa prise de décision.

Afin de rationaliser les décisions des architectures profondes, une grande diversité de définitions et de techniques a ainsi été produite sur ce sujet au cours du temps. En particulier, certains traits des architectures, pouvant être des atouts pour la sécurité, peuvent être retrouvés au sein de la littérature. De façon grossière, on pensera au fait de fonctionner systématiquement comme prévu. Une architecture robuste doit également montrer un haut niveau de fiabilité, à la fois pour des conditions de fonctionnement communes et rares, ou à l'inverse, sous condition de présence d'indices inquiétants ou jamais vue dans les données d'entrée, doit se prémunir de tout choix pouvant impacter négativement l'utilisateur ou son environnement. Enfin, il peut aussi être intéressant de fournir des retours utiles à l'utilisateur sur la façon dont les conditions de fonctionnement influencent les décisions. Certains autres sujets d'apprentissage profond peuvent également être associés à l'explicabilité, notamment l'explication des mécanismes d'apprentissage ou les techniques associées aux GAN (Generative Adversarial Network), qui permettent de rechercher les différences entre des activations régulières et inattendues dans la décision du réseau.

3.2 Taxonomie de l'explicabilité en apprentissage profond

Dans leurs travaux, [Gilpin et al. \(2018\)](#) considèrent 3 catégories de méthodes d'explicabilité. Une première englobe celles qui émulent le traitement des données afin d'obtenir des intuitions concernant les connexions entre les entrées et les sorties du modèle. La deuxième catégorie contient des approches qui tentent d'expliquer la représentation des données à l'intérieur d'un réseau, tandis que la dernière catégorie est constituée de réseaux transparents qui s'expliquent d'eux-mêmes. Tout comme dans [Adadi & Berrada \(2018\)](#), ils insistent sur le fait qu'une combinaison de ces méthodes est bénéfique dans un grand nombre de domaines d'application.

Dans une revue sur les méthodes d'interprétabilité pour l'apprentissage profond, [Linardatos et al. \(2021\)](#) proposent une taxonomie permettant de classifier ses différentes méthodes, que l'on représente ici avec la figure 2.9.

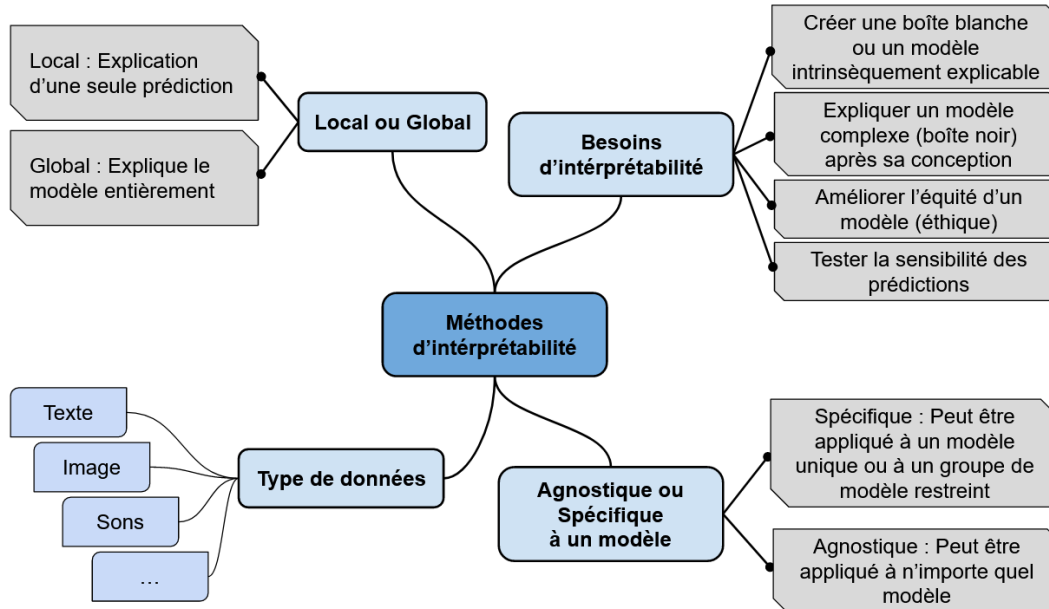


Figure 2.9: Schéma de la taxonomie des différents types de méthodes d'interprétabilité. Adapté de l'article de [Linardatos et al. \(2021\)](#).

[Xie et al. \(2020\)](#) proposent également de regrouper les méthodes d'explicabilité parmi trois sous-groupes, pouvant concerner différents utilisateurs experts :

- **Méthodes de visualisation :** Permettent d'exprimer une explication en mettant en évidence, à travers une visualisation scientifique, les caractéristiques d'une entrée qui influencent fortement la sortie d'un réseau profond.
- **Méthodes de description :** Développement d'un modèle d'apprentissage automatique boîte blanche, qui sera entraîné après une architecture profonde, afin d'imiter le comportement entre les entrées et sorties de ce dernier au sein d'une représentation pouvant être expliquée par un utilisateur. Ce modèle de boîte blanche, qui doit être intrinsèquement explicable comme un modèle linéaire ou un arbre de décision, est destiné à identifier les règles de décision ou les caractéristiques d'entrée influençant les sorties des réseaux profonds.
- **Méthodes produisant des sorties intrinsèquement explicables :** Architectures profondes spécifiquement créées pour restituer pas seulement un label, mais également des informations explicatives en sortie. En conséquence de cette conception, les réseaux profonds donnant des sorties intrinsèquement explicables peuvent optimiser conjointement les performances du modèle et la qualité des explications produites.

Remarque

Par la suite nous distinguerons méthodes produisant des sorties intrinsèquement explicables et architectures intrinsèquement explicables, sous-entendues par design, dans le sens où ces dernières ont une structure mathématique intrinsèquement explicable, ce qui n'est pas forcément le cas des premières.

3.3 Méthodes de visualisations

Parmi les méthodes de visualisation, on pourra notamment penser à des méthodes basées sur la rétro-propagation du gradient d'une prédiction telle que GradCAM, une méthode proposée par [Zhou et al. \(2016\)](#), qui peut être utilisée pour la classification d'images comme le font [Selvaraju et al. \(2017\)](#). Cette méthode permet en effet de visualiser la pertinence d'appartenir à une classe pour chaque zone d'une image et ce, en se basant sur l'intensité qui aurait été transmise par la rétro-propagation du gradient à travers les couches pendant l'entraînement du réseau sur cette image.

Les méthodes de perturbation peuvent également être regroupées parmi les méthodes de visualisation, puisqu'elles consistent à évaluer l'importance d'une caractéristique, ou d'une localisation dans une image, en comparant la sortie d'un modèle avec et sans perturbation. Les méthodes de perturbation peuvent ainsi calculer la pertinence marginale de chaque caractéristique d'une entrée $\mathbf{x} \in \mathcal{X}$, par rapport à la façon dont un réseau répond à une entrée particulière $\mathbf{x}^{(i)}$.

3.4 Méthodes de description

Parmi les méthodes de description, les approximations locales doivent permettre d'apprendre un modèle simple, dont la modélisation des relations entre entrées et sorties doit imiter celle d'une boîte noire pour un sous-ensemble de données. En effet, si on dispose d'une densité de points \mathbf{x} suffisamment élevée localement pour permettre à une fonction linéaire par morceaux d'approximer la distribution des entrées sorties, alors le comportement d'un modèle profond peut être localement décrit par un ensemble de modèles linéaires. C'est le principe de la méthode agnostique LIME, développée par [Ribeiro et al. \(2016\)](#) et qui cherche une fonction g linéaire telle que :

$$\arg \min_g \sum_{\mathbf{x} \in \mathcal{S}} \mathcal{L}(f(\mathbf{x}), g(\mathbf{x})), \quad (2.11)$$

avec \mathcal{L} une fonction de perte pouvant inclure un terme de régularisation sur g , f_θ le modèle profond et \mathcal{S} un sous-ensemble de données.

La méthode SHAP est également une approche locale qui consiste à calculer les valeurs de [Shapley \(1953\)](#), tirées de la théorie des jeux, pour une caractéristique/variable de \mathbf{x} . La contribution d'une variable spécifique au score donné par un réseau, i.e. sa prédiction, est en effet capturée par la valeur de Shapley. Cette dernière représente à quel point la prédiction du modèle change quand une valeur de variable est observée. Calculer l'ensemble de ces valeurs,

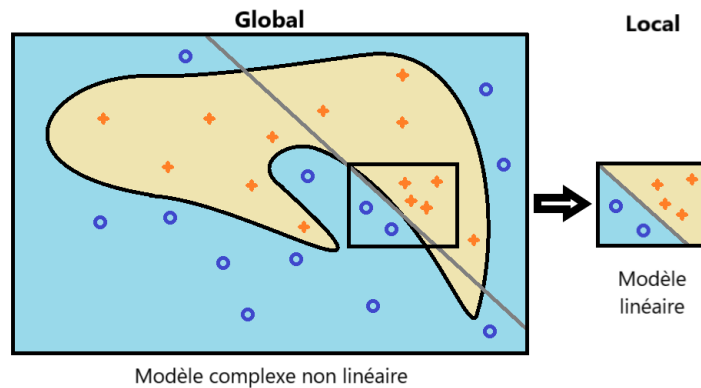


Figure 2.10: *Modèle linéaire en tant que descripteur local d'un modèle complexe.*

pour toutes les variables des entrées $\mathbf{x} \in \mathcal{X}$, permet alors d'obtenir une explication globale compatible avec les explications locales de chaque variable.

3.5 Méthodes produisant des sorties intrinsèquement explicables

Concernant les méthodes produisant des sorties intrinsèquement explicables, qui doivent à la fois fournir un label et une explication de classification, on y retrouve souvent les méthodes basées sur les mécanismes d'attention, dont la visualisation révèle l'explicabilité inhérente, notamment pour les textes et images. Comme le suggèrent [Vaswani et al. \(2017\)](#), l'inclusion de mécanismes d'attention au sein d'un réseau profond permet souvent d'obtenir de meilleures performances qu'avec des réseaux convolutifs ou à propagation avant équivalents, tout en ayant des sorties explicables, en rendant compte des opérations qui ont été effectuées. Le mécanisme d'attention apprend en effet la distribution conditionnelle sur les variables/caractéristiques des entrées, fournissant ainsi un vecteur pondéré contextuel qui sera réutilisé par les couches suivantes. Dans leurs travaux, [Larochelle & Hinton \(2010\)](#) ont mis en avant le fait qu'en regardant différentes parties d'une image, nous pouvons apprendre à accumuler des informations sur une forme, afin de classer cette image en conséquence. Le même principe a ensuite été étendu aux séquences de mots. Nous pouvons en effet regarder tous les différents mots en même temps et apprendre à prêter attention aux bons mots en fonction de la tâche à accomplir. L'attention est ainsi une notion de mémoire, acquise en faisant attention à de multiples entrées à travers le temps. Pour les signaux acoustiques les couches d'attention peuvent également être utilisées sur signaux acoustiques bruts comme le font [Yang et al. \(2019\)](#), mais sur ce type de problème les mécanismes d'attention n'apportent pas une grande contribution à l'explicabilité, bien qu'on puisse facilement visualiser les poids d'attention.

Les méthodes de multi-task learning peuvent également être considérées comme des méthodes produisant des sorties intrinsèquement explicables, puisqu'elles consistent à ajouter une tâche d'explication supplémentaire à la tâche de classification du modèle d'origine. On espère alors que le modèle, dont l'architecture est conçue à l'origine pour une tâche de classification, devrait également être capable d'apprendre une structure mathématique interne capable de fournir de bonnes règles pour les tâches d'explication. Qui plus-est, ces tâches d'explication

peuvent également fournir plus d'informations sur la tâche d'origine en présentant des labels plus riches au modèle durant l'apprentissage conjoint des tâches d'explication et de la tâche d'origine. Dans la littérature, cette méthode peut prendre plusieurs formes comme par exemple l'explication via un texte, qui consiste à fournir des explications sous la forme de phrase en langage naturel concernant ce qu'il se passe dans une image, comme le font [Zellers et al. \(2019\)](#). Les tâches de segmentations d'objet, qui accompagnent souvent la labellisation d'images comme sur la figure 2.11, sont également des formes d'explicabilité données par un apprentissage sur multiples objectifs.

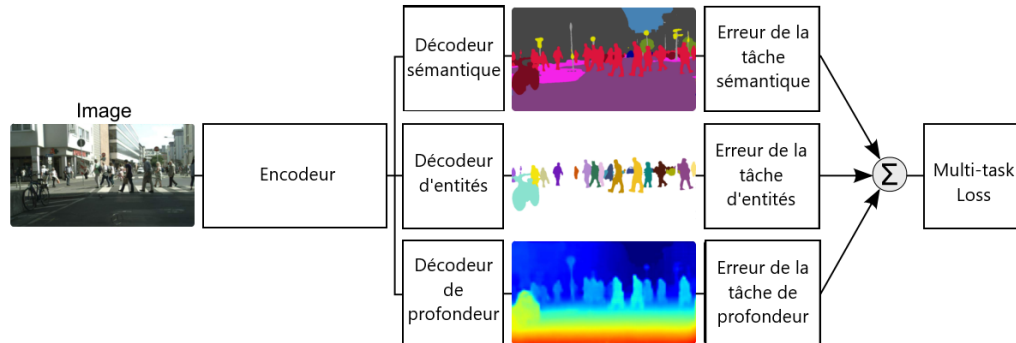


Figure 2.11: Exemple de multi-task learning pour les images. Tiré de l'article de [Kendall et al. \(2018\)](#).

3.6 Méthodes d'explicabilité en ASM

Durant la thèse, plusieurs options ont été envisagées pour apporter des formes d'explicabilité supplémentaires. Le multi-task learning a par exemple été utilisé pour la classification binaire, entre bruits de navire et bruits ambiants, pour apprendre en même temps le nombre de pales et la vitesse de rotation de l'hélice. De même, des tentatives ont été faites pour prédire le SNR d'une source. Ces deux options n'ont cependant pas été fructueuses et ont même abouti à de moins bonnes performances de classification pour l'objectif d'origine. La prédiction de la vitesse de rotation de l'hélice seule n'a en revanche pas vraiment altéré les performances de la tâche de classification et a permis d'apporter des informations supplémentaires en sortie.

Concernant les méthodes de visualisation, elles sont assez bien adaptées pour des représentations des signaux en 2 dimensions, notamment les représentations temps-fréquence, comme en attestent les figures 2.12, 2.13 et 2.14 ci-dessous. Ces figures ont été obtenues avec la méthode GradCam sur des modèles CNNs 2D entraîné avec un jeu de données basé sur les classes présentées dans la section 3.1 du chapitre 4. Sur la figure 2.12, on peut constater que l'absence de signal provenant d'une source ponctuelle entraîne l'activation des couches de convolution de façon chaotique dans l'image, tandis que la présence de clics biologiques, localisés en temps et étalés sur les hautes fréquences, entraîne l'activation du modèle CNN 2D uniquement pour ces zones sur la visualisation heatmap avec GradCam. Sur la figure 2.13, il est possible de visualiser des activations plus régulières. Celles-ci sont localisées en temps et fréquence, une fois sur deux entre 2 pics successifs correspondants au rythme de la rotation

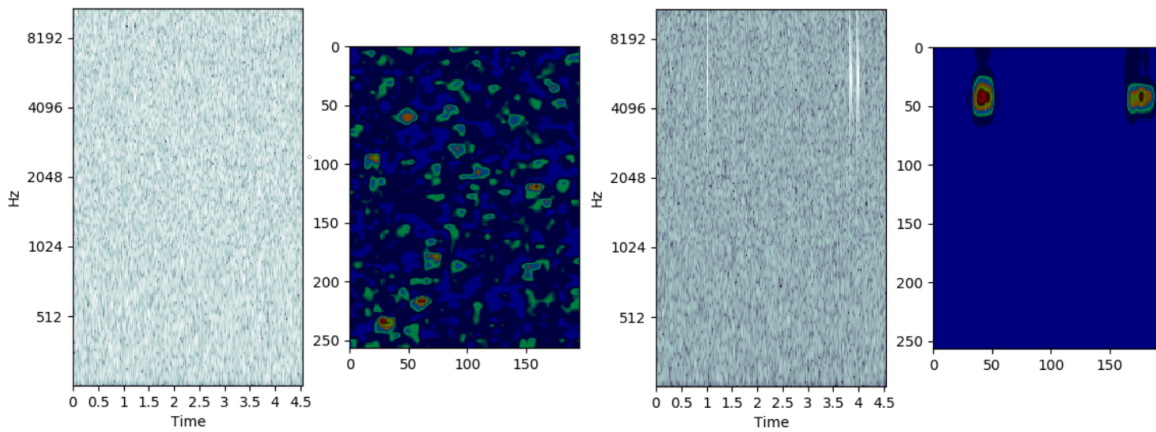


Figure 2.12: Visualisation GradCam appliquée à du bruit ambiant blanc Gaussien, à gauche et à des clics biologiques dans du bruit ambiant blanc Gaussien, à droite.

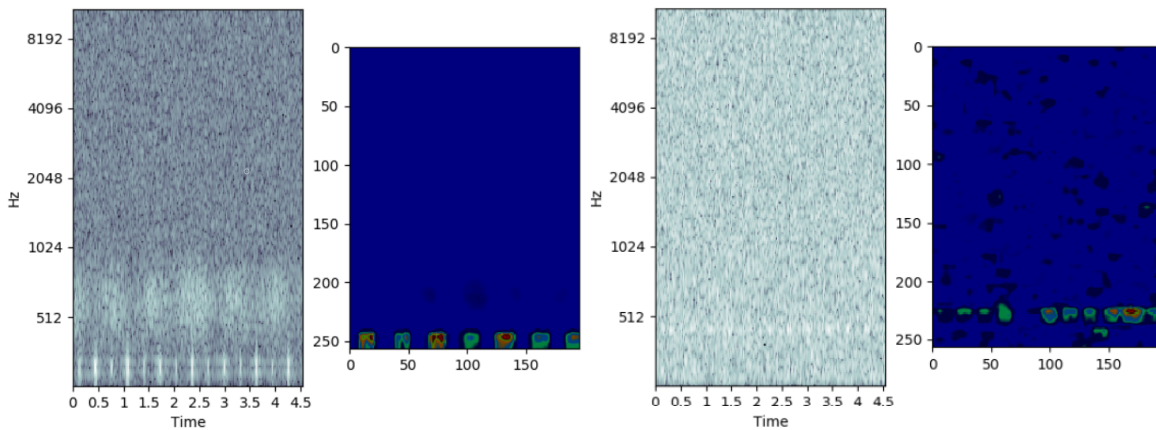


Figure 2.13: Visualisation GradCam appliquée à un bruit d'hélice fort, à gauche et à un bruit d'hélice faible, à droite, les deux dans du bruit ambiant blanc Gaussien.

des pales d'un navire. Pour un SNR moindre, sur l'image de droite, on peut constater que tous les pics ne sont pas détectés par l'architecture CNN 2D. Qui plus est, les performances d'un détecteur dépendant de la durée d'intégration du signal, ces méthodes de visualisation tendent à faire ressortir des activations faibles sur une bande de fréquences étalée sur l'ensemble des valeurs temporelles, comme en attestent les images de la figure 2.14. Aussi, contrairement à des images conventionnelles où les activations sont classiquement restreintes à une zone de l'image, cette visualisation présente des faiblesses pour les signaux stationnaires du fait qu'on puisse classifier un signal avec une grande confiance en intégrant sur de longue durée avec les couches de pooling successives, malgré le fait qu'une visualisation par GradCam montre des activations faibles mais étalées. La visualisation par Gradcam permet également de visualiser plusieurs sources différentes si elles sont séparées en temps et fréquence

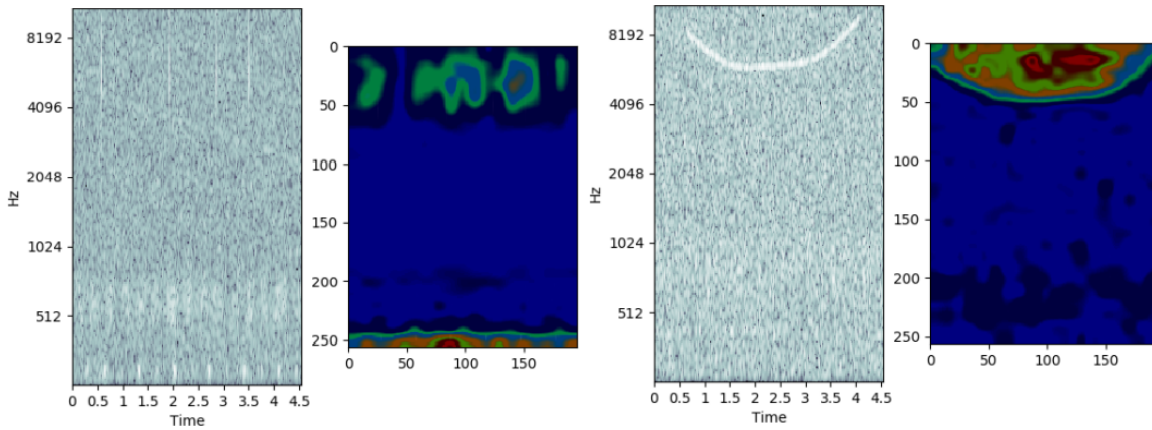


Figure 2.14: *Visualisation GradCam appliquée à un bruit d'hélice fort et des clics biologiques, à gauche et à un bruit d'hélice faible et du bruit de modulation, à droite, les deux dans du bruit ambiant blanc Gaussien.*

L'explicabilité basée sur la visualisation est cependant plus limitée sur des signaux acoustiques brutes. Le cumul d'informations sur une représentation 1D rend en effet son interprétation difficile pour son utilisateur, notamment pour des signaux cyclostationnaires ou dans le cas multi-classe. Néanmoins, cette option peut s'avérer utile pour la détection d'événements courts et spontanés. Il est en revanche envisageable d'utiliser une méthode de visualisation sur des couches intermédiaire d'un CNN 1D, notamment en essayant de transformer une feature map intermédiaire en image 2D, comme le proposent [Tokozume et al. \(2017\)](#). Il est cependant nécessaire que cette représentation apprise puisse être interprétable par un être humain. Aussi, en plus de l'information temporelle, il faut que la deuxième direction, introduite par les différents filtres de convolutions 1D, puisse être pertinente en représentant par exemple des réponses fréquentielles croissantes, comme c'est le cas de la Scattering transform de [Mallat \(2012\)](#), bien que [Tokozume et al. \(2017\)](#) observent tout de même souvent cette propriété dans les filtres appris.

4 Conclusion

Dans ce chapitre nous avons effectué un état de l'art des différentes méthodes de classification dans divers domaines d'application de l'acoustique aérienne et en acoustique sous-marine. Ces méthodes se regroupent généralement en trois grandes catégories : les méthodes utilisant des représentations des signaux sous forme de variables, en combinaison avec des algorithmes intrinsèquement explicables de type boîte blanche, les méthodes utilisant des représentations 2D liées au temps avec des algorithmes de type CNN 2D et enfin, l'utilisation de signaux bruts par le biais d'architectures profondes comportant très souvent des couches de convolutions 1D. Bien que les méthodes sur signaux 2D permettent déjà d'obtenir d'excellentes performances, les architectures sur signaux brutes peuvent potentiellement apporter de meilleurs résultats avec des architectures plus complexes, qui requièrent ainsi souvent plus de données d'entraînement, pour un apprentissage de bout en bout.

Dans la section 3 nous avons ensuite classifié les différentes méthodes d'explicabilité existant pour les architectures profondes. Aussi, parmi les méthodes d'explicabilité i.e. les algorithmes intrinsèquement explicables, les méthodes de visualisation, les méthodes de description et les méthodes produisant des sorties intrinsèquement explicables, toutes ne sont pas applicables à notre problème principal de classification de bruit de navire en ASM. Les méthodes de visualisation peuvent par exemple être utilisées pour les CNNs 2D mais pas forcément pour les CNNs 1D. Pour ces raisons, la suite de la thèse se focalise plutôt sur des algorithmes profonds intrinsèquement explicables, un axe peu présent dans la littérature pour la simple raison que c'est un compromis entre performances et l'utilisation de boîtes blanches. La plupart des travaux dans ce domaine préfèrent en effet utiliser des architectures de pointe complexes en tant que boîte noire, en y apportant des explications interprétables lors de la prise de décision sur une entrée précise de l'algorithme.

Chapitre 3

Théorie de l'apprentissage

1 Introduction

Dans le but de mieux comprendre les processus d'apprentissage, nous abordons dans ce chapitre divers résultats concernant la théorie de l'apprentissage supervisé, en commençant par le contrôle du risque dans la section 2. Dans le cadre d'un apprentissage où les données (\mathbf{x}, y) suivent une distribution conjointe \mathcal{D} , le contrôle du risque doit permettre de décomposer l'erreur entre un modèle $g_\theta(\mathbf{x})$ et le classifieur optimal $\delta^*(\mathbf{x})$ en erreurs d'approximation, de généralisation et d'optimisation. Dans un objectif de robustesse, nous présentons dans la section 3 des aspects de la théorie de Vapnik-Tchervonenkis, qui doit permettre de borner l'erreur de généralisation de modèles appris, en fonction de l'expressivité de l'algorithme qui les définit. Dans la section 4, nous présentons certaines mesures de complexité, censées prédire plus finement le comportement de généralisation de modèles d'apprentissage profond. Enfin, dans la section 5, nous présentons quelques propriétés de la descente de gradient stochastique (SGD) concernant l'erreur de généralisation. Nous ne discutons cependant pas des diverses variantes de la SGD et de leurs impacts sur l'erreur d'optimisation. De même, la théorie de l'apprentissage étant un domaine extrêmement vaste, nous n'aborderons pas explicitement les nombreuses méthodes de régularisation souvent utilisées dans la pratique (dropout, lasso, ridge, ...) dans une section dédiée, afin de plutôt se concentrer sur des résultats théoriques fondamentaux liés aux détecteurs optimaux.

2 Contrôle du risque

2.1 Définition

Dans le domaine de l'apprentissage automatique, le contrôle du risque désigne généralement, l'estimation des limites théoriques des performances de classification d'une famille d'algorithmes d'apprentissage \mathcal{G} donnée. En effet, bien que nous ne puissions pas savoir comment un algorithme se comportera dans la pratique si nous ne connaissons pas la distribution exacte des données sur lesquelles l'algorithme fonctionnera (notion de risque réel \mathcal{R}), nous pouvons cependant mesurer ses performances sur un ensemble de données d'entraînement connu, avec le risque empirique \mathcal{R}_N . Le principe même de l'apprentissage machine consiste alors à trouver le jeu de paramètres optimal θ^* , d'une fonction g_θ appartenant à une famille

de fonctions \mathcal{G} et tel que

$$g_{\theta^*} = \arg \min_{g_{\theta} \in \mathcal{G}} \mathcal{R}(g_{\theta}), \quad (3.1)$$

$$= \arg \min_{g_{\theta} \in \mathcal{G}} \mathbb{E}_{(X,Y) \sim \mathcal{D}}[\mathcal{L}(g_{\theta}(X), Y)], \quad (3.2)$$

où \mathcal{R} est le risque réel au sens Bayésien. Par la suite on définira \mathcal{D} comme étant la distribution des couples (X, Y) , une loi hypothétique liant les entrées aux sorties. Généralement on disposera plutôt d'échantillons $(\mathbf{x}^{(i)}, y^{(i)})$ qui suivent cette loi $\mathcal{D}(\mathbf{x}, y)$. Il est également nécessaire de prendre en considération que dans certains problèmes de classification et en ce qui nous concerne, la détection bruitée, nous n'aurons pas forcément $\forall x \in \mathcal{X}, \exists y \in \mathcal{Y}$ telle que $\Pr(Y = y|X = x) = 1$.

Dans l'apprentissage agnostique où la distribution $\mathcal{D}(\mathbf{x}, y)$ est inconnue pour l'algorithme de classification, on va ainsi calculer le risque empirique en prenant en compte la moyenne de la fonction de perte sur un jeu d'entraînement $\mathcal{S} = \{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$ disposant de N instances/exemples i.i.d. :

$$\mathcal{R}_N(g_{\theta}) := \frac{1}{N} \sum_{i=1}^N \mathcal{L}(g_{\theta}(\mathbf{x}^{(i)}), y^{(i)}), \quad (3.3)$$

Le principe de minimisation du risque empirique de [Vapnik \(1992\)](#) stipule alors que l'algorithme d'apprentissage doit choisir une fonction (ou hypothèse) $g_{\theta'}$, de paramètres θ' , qui minimise le risque empirique :

$$g_{\theta'} = \arg \min_{g_{\theta} \in \mathcal{G}} \mathcal{R}_N(g_{\theta}). \quad (3.4)$$

En notant $\mathcal{R}^* = \min_f \mathcal{R}(f)$, sans aucune restriction sur la classe de fonction de f , pour tout ensemble de paramètre θ , ainsi que θ' et θ^* , on peut décomposer l'excès de risque $\mathcal{E}(g_{\theta}) = \mathcal{R}(g_{\theta}) - \mathcal{R}^*$, puis majorer indépendamment chaque terme :

$$\begin{aligned} 0 \leq & \mathcal{R}(g_{\theta}) - \mathcal{R}^* & = & \text{(excès de risque)} \\ & \mathcal{R}(g_{\theta^*}) - \mathcal{R}^* & + & \text{(erreur d'approximation)} \\ & \mathcal{R}_N(g_{\theta^*}) - \mathcal{R}(g_{\theta^*}) & + & \text{(erreur de généralisation 1)} \\ & \mathcal{R}_N(g_{\theta'}) - \mathcal{R}_N(g_{\theta^*}) & + & \leq 0 \text{ par optimalité} \\ & \mathcal{R}_N(g_{\theta}) - \mathcal{R}_N(g_{\theta'}) & + & \text{(erreur d'optimisation)} \\ & \mathcal{R}(g_{\theta}) - \mathcal{R}_N(g_{\theta}) & + & \text{(erreur de généralisation 2)} \end{aligned}$$

- **Erreur d'approximation** : Concerne la capacité d'approximation de la classe d'algorithme employée. Elle dépend ainsi directement de la richesse de la classe de fonction \mathcal{G} , sans tenir compte d'échantillons aléatoires : par exemple le pouvoir d'expressivité d'un réseau de neurones d'architecture fixée.
- **Erreur de généralisation** : Concerne les conditions nécessaires pour que l'algorithme permette à $\mathcal{R}_N(g_{\theta})$ d'approcher $\mathcal{R}(g_{\theta})$ pour tout ensemble de paramètres θ . Cela concer-

nera principalement le contrôle de

$$\sup_{\theta} \left| \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\mathcal{L}(g_{\theta}(\mathbf{x}), y)] - \frac{1}{N} \sum_{i=1}^N \mathcal{L}(g_{\theta}(\mathbf{x}^{(i)}), y^{(i)}) \right|, \quad (3.5)$$

notamment en fonction du nombre d'échantillons d'entraînement N comme par le biais de la dimension de Vapnik-Tchervonenkis ou la complexité de Rademacher.

- **Erreur d'optimisation** : Concerne le problème d'optimisation même. Pour les réseaux de neurones, les paysages des fonctions d'objectif sont souvent non convexes et on ne sait que peu de choses sur leur régularité. L'optimisation dans l'apprentissage profond pose aussi des problèmes de calculs du fait que le nombre d'échantillons N est conséquent et que les ensembles de paramètres θ peuvent vivre dans des espaces de grande dimension.

Dans une situation idéale, la loi de distribution \mathcal{D} des paires entrées sorties (X, Y) est connue. Alors, le problème de prédiction se résume à un problème d'optimisation : trouver $g_{\theta} \in \mathcal{G}$ tel que le risque $\mathcal{R}(g_{\theta})$ est minimal, où $\mathcal{R} : \mathcal{G} \rightarrow \mathbb{R} \cup \{+\infty\}$ est une fonction connue :

$$\mathcal{R}(f) := \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\mathcal{L}(f(\mathbf{x}), y)]. \quad (3.6)$$

On définit alors le risque de Bayes tel que

$$\mathcal{R}^* = \min_f \mathcal{R}(f) = \mathcal{R}(\delta^*), \quad (3.7)$$

où \mathcal{R}^* est alors la plus petite valeur de risque envisageable pour un prédicteur et ce prédicteur lorsqu'il existe, est le prédicteur de Bayes, ou encore règle de Bayes qu'on notera $\delta^*(x)$. De plus, si \mathcal{L} est à valeurs positives ou nulles, le risque de Bayes le sera également. Dans la plupart des cas une prédiction parfaite est cependant impossible, même en connaissant \mathcal{D} , due à une forme de bruit et le risque de Bayes est alors strictement positif. Une classe de prédicteur optimal pour la distribution \mathcal{D} est alors une classe \mathcal{G} dont le risque minimal est égal au risque de Bayes :

$$g_{\theta^*} = \arg \min_{g_{\theta} \in \mathcal{G}} \mathcal{R}(g_{\theta}) \in \arg \min_f \mathcal{R}(f) \quad (3.8)$$

Remarque

Une autre forme, plus répandue, est la décomposition de l'excès de risque $\mathcal{E}(g_{\theta'}) = \mathcal{R}(g_{\theta'}) - \mathcal{R}^*$.

$$\mathcal{R}(g_{\theta'}) - \mathcal{R}^* = \underbrace{\mathcal{R}(g_{\theta'}) - \mathcal{R}(g_{\theta^*})}_{\text{erreur d'estimation}} + \underbrace{\mathcal{R}(g_{\theta^*}) - \mathcal{R}^*}_{\text{erreur d'approximation}}. \quad (3.9)$$

Cette décomposition met en évidence le compromis fait entre les erreurs d'approximation et d'estimation, l'erreur d'estimation étant la seule partie aléatoire de cette décomposition. En effet, généralement plus la classe de fonction \mathcal{G} sera riche, plus l'erreur d'approximation sera faible au détriment de l'erreur d'estimation, ou tout du moins c'est ce à quoi on s'attend pour l'apprentissage automatique classique.

2.2 Consistance

Pour qu'un algorithme d'apprentissage soit considéré comme efficace, un critère naturel serait d'exiger à ce qu'il fasse aussi bien que le risque de Bayes pour la loi $(X, Y) \sim \mathcal{D}$, lorsque le nombre d'observations disponibles tend vers l'infini. On parle alors de consistance pour un type d'algorithmes et plusieurs définitions sont alors possibles :

Définition : Consistance

Pour une règle d'apprentissage $g_{\theta, N}$ donnée, liée à la fois à une classe de fonction \mathcal{G} , à un algorithme d'apprentissage, ainsi qu'à un ensemble d'apprentissage \mathcal{S} disposant de $N > 1$ variables indépendantes suivant toutes une même loi de distribution \mathcal{D} , on dit que :

g_{θ} est faiblement consistante pour \mathcal{D} quand :

$$\mathbb{E} \left[\mathcal{R}(g_{\theta, N}) \xrightarrow[N \rightarrow +\infty]{} \mathcal{R}^* \right],$$

g_{θ} est fortement consistante pour \mathcal{D} quand :

$$\mathcal{R}(g_{\theta, N}) \xrightarrow[N \rightarrow +\infty]{p.s.} \mathcal{R}^*.$$

On dira également qu'une classe de fonction \mathcal{G} est universellement consistante, si elle est consistante pour toute distribution de couple entrée sortie $\mathcal{D}(X, Y)$.

L'algorithme des k plus proches voisins noté k_N -NN, avec $k_N = \sqrt{N}$, est un exemple de classe de fonction fortement universellement consistante, comme le stipulent [Kulkarni & Harman \(2011\)](#). En effet, à mesure que le nombre d'échantillons $\mathbf{x}^{(i)}$ augmente, les performances de la règle de classification du k_N -NN approchent de plus en plus celles du classifieur optimal de Bayes.

En outre, dans le cas où l'ensemble \mathcal{X} est dénombrable, l'existence d'algorithmes consistants est facilement explicable puisque, comme le montrent [Devroye et al. \(1989\)](#), même s'il n'existe aucune relation entre entrée et sortie, en échantillonnant indépendamment à plusieurs reprises des données de la distribution \mathcal{D} , on pourra voir un nombre croissant de données différentes, pour finir par converger vers l'ensemble des entrées. Ainsi, dans le cas dénombrable, un algorithme qui apprendrait par coeur serait consistant. On peut notamment penser à des algorithmes votant à la majorité.

En revanche, dans le cas où l'ensemble \mathcal{X} n'est pas dénombrable, i.e. $\mathcal{X} = \mathbb{R}$, une hypothèse devient cruciale : pour pouvoir définir une mesure de probabilité \Pr sur \mathcal{X} , on a besoin d'une σ -algèbre ou tribu sur cet espace et typiquement un corps de Borel. On assume ainsi que \Pr est une mesure de Borel. Cela signifie que la topologie de \mathcal{X} joue ici un rôle, et donc, la fonction cible $\delta^*(\mathbf{x})$ sera Borel mesurable. Dans un sens, cela garantit qu'il est possible d'approximer $\delta^*(\mathbf{x})$ à partir de sa valeur, ou au moins une approximation de sa valeur, en un nombre fini

de points. Les algorithmes cohérents sont donc ceux qui utilisent la topologie, dans le sens où ils généralisent des valeurs observées dans un voisinage, par exemple les classificateurs locaux.

Les fonctions noyaux sont ainsi également des algorithmes fortement universellement consistants. En particulier, les travaux de [Steinwart & Christmann \(2008\)](#) sur les SVMs, ont permis de mettre en avant le fait que les SVMs disposant de certaines fonctions noyaux comme le noyau Gaussien $K(\mathbf{x}, \mathbf{x}') = e^{-\gamma \|\mathbf{x} - \mathbf{x}'\|^2}$, étaient universellement consistants :

Consistance universelle des SVM

Théorème 1 Soit $K(\mathbf{x}, \mathbf{x}')$ une fonction noyau universel, utilisée pour une SVM $f_\theta \in \mathcal{F}$ et telle que :

$$\inf_{f_\theta \in \mathcal{F}} \mathcal{R}(f_\theta) = \mathcal{R}^*, \quad (3.10)$$

avec un risque basé sur une fonction de perte Lipschitzienne \mathcal{L}_L telle que $\sup_{y \in \mathcal{Y}} \mathcal{L}_L(0, y) < +\infty$. On définira une fonction de perte Lipschitzienne comme une fonction de perte \mathcal{L} telle que pour tout $y \in \mathcal{Y}$, $\mathcal{L}(\cdot, y)$ est C -Lipschitz, avec C ne dépendant pas de y . En assumant alors que $\sup_{\mathbf{x} \in \mathcal{X}} \sqrt{K(\mathbf{x}, \mathbf{x})} < \infty$ et que \mathcal{X} soit un compact, alors, pour $\lambda = \lambda_N \rightarrow 0$ telle que

$$N \cdot \lambda_N \xrightarrow{N \rightarrow +\infty} +\infty, \quad (3.11)$$

on a pour toute distribution $\mathcal{D}(\mathcal{X}, \mathcal{Y})$:

$$\mathcal{R}(f_{\theta, N}) \xrightarrow[N \rightarrow +\infty]{p.s.} \mathcal{R}^*. \quad (3.12)$$

Cette propriété de consistance universelle peut également être élargie aux SVM multi-classes, comme le montrent [Steinwart & Christmann \(2008\)](#).

2.3 Consistance pour les réseaux de neurones

Concernant la consistance des réseaux de neurones, nous considérerons ici le classifieur défini par un réseau de neurones à propagation avant ne disposant que d'une seule couche cachée :

$$\widehat{\delta}_\theta(\mathbf{x}) = \mathbb{1}_{\{\rho_s(f_\theta(\mathbf{x})) \geq 0.5\}} = \begin{cases} 0 & \text{si } f_\theta(\mathbf{x}) < 0, \\ 1 & \text{si } f_\theta(\mathbf{x}) \geq 0, \end{cases} \quad (3.13)$$

où $\mathbf{x} \in \mathbb{R}^K$, $\mathbb{1}_{\mathcal{A}}$ est la fonction indicatrice, $\rho_s(t)$ est une fonction d'activation de type sigmoïde monotone croissante telle que

$$\lim_{t \rightarrow +\infty} \rho_s(t) = 1, \quad (3.14)$$

$$\lim_{t \rightarrow -\infty} \rho_s(t) = -1 \quad (3.15)$$

et

$$f_\theta(\mathbf{x}) = \sum_{j=1}^W \gamma_j^{(1)} \rho_s \left(\gamma_j^{(0)} \mathbf{x} + b_j^{(0)} \right) + b^{(1)}, \quad (3.16)$$

avec W le nombre de neurones présent dans la couche cachée, $b^{(1)}, \gamma_j^{(1)}, b_j^{(0)} \in \mathbb{R}$, $\gamma_j^{(0)} \in \mathbb{R}^K$ et $\theta = \{\gamma_1^{(1)}, \dots, \gamma_W^{(1)}, b_1^{(0)}, \dots, b_W^{(0)}, \gamma_1^{(0)}, \dots, \gamma_W^{(0)}, b^{(1)}\}$. Dans leur démonstration de la forte consistance universelle de ce type de réseau, [Faragó & Lugosi \(1993\)](#) emploient la fonction de perte idéale "0-1", $\mathcal{L}_{0-1}(z, y) = \mathbb{1}_{\{z=y\}}$ comme erreur et une fonction en escalier comme fonction d'activation :

$$\rho_e(x) = \begin{cases} -1 & \text{si } x \leq 1/2, \\ 1 & \text{sinon.} \end{cases} \quad (3.17)$$

En choisissant un ensemble de paramètres θ' qui minimise le risque empirique $\mathcal{R}_N(g_\theta)$, [Faragó & Lugosi \(1993\)](#) montrent que la règle de décision $g_{\theta'}$ est fortement universellement consistante à condition que $W \rightarrow +\infty$ et $W \cdot \log(N)/N \rightarrow 0$ quand $N \rightarrow +\infty$. Pour y parvenir l'excès de risque $\mathcal{R}(g_{\theta'}) - \mathcal{R}^*$ est décomposé en l'erreur d'approximation $\mathcal{R}(g_{\theta^*}) - \mathcal{R}^*$ et l'erreur d'estimation $\mathcal{R}(g_{\theta'}) - \mathcal{R}(g_{\theta^*})$. [Asadi & Jiang \(2020\)](#) parviennent à un résultat similaire en considérant un réseau disposant de fonction d'activation ReLU et de paramètres initialisés aléatoirement.

Concernant l'erreur d'approximation, le théorème d'approximation de [Cybenko \(1989\)](#) permet de montrer que $\mathcal{R}(g_{\theta^*})$, la probabilité d'erreur du classifieur g_θ disposant des meilleures paramètres possibles, peut approcher de très près le risque de Bayes, si le nombre de neurones sur la couche cachée est suffisamment important :

$$\lim_{W \rightarrow +\infty} \min_{g_\theta \in \mathcal{G}} \mathcal{R}(g_\theta) - \mathcal{R}^* = 0. \quad (3.18)$$

Pour l'erreur d'estimation, qui représente la différence entre la probabilité d'erreur de la règle choisie $g_{\theta'}$ et celle avec les paramètres optimaux g_{θ^*} , on peut se servir d'un lemme selon lequel, pour tout $\epsilon > 0$, $N \geq 2/\epsilon^2$ et avec d la dimension des entrées, on a :

$$\Pr(\mathcal{R}(g_{\theta'}) - \mathcal{R}(g_{\theta^*}) \geq \epsilon) \leq 2^{w+3} (2N + 1)^{(w+1)d} e^{-\frac{N\epsilon^2}{32}}. \quad (3.19)$$

Aussi, si le nombre de neurones W dans l'unique couche cachée d'un réseau de neurones à propagation avant $g_{\theta'} = \arg \min_{g_\theta \in \mathcal{G}} \mathcal{R}_N(g_\theta)$ satisfait $W \rightarrow +\infty$ et $W \cdot \log(N)/N \rightarrow 0$ quand $N \rightarrow +\infty$ alors, $\lim_{N \rightarrow +\infty} \mathcal{R}(g_{\theta'}) = \mathcal{R}^*$, avec une probabilité égal à 1.

Une information intéressante serait alors de savoir à quelle vitesse $\mathcal{R}(g_{\theta'})$ converge vers \mathcal{R}^* , et quel serait un bon choix pour W . [Devroye \(1982\)](#) a démontré qu'il n'était pas possible d'obtenir de borne supérieure sur le taux de convergence sans faire d'hypothèses sur la distribution des couples entrées sorties \mathcal{D} . Aussi, [Faragó & Lugosi \(1993\)](#) utilisent une borne supérieure indépendante de cette distribution pour trouver un taux de convergence de l'erreur d'estimation :

Convergence de l'erreur d'estimation d'un réseau de neurones avec une couche cachée

Théorème 2 *Assumons qu'il existe un ensemble compact $A \subset \mathbb{R}^K$ tel que $\Pr(X \in A) = 1$ et la transformation de Fourier $\widehat{\Pr}_0(\omega)$ de $\Pr_0(x) = \Pr(Y = 0|X = x)$ satisfait $\int_{\mathbb{R}^K} |\omega| |\widehat{\Pr}_0(\omega)| d\omega < +\infty$, alors :*

$$\mathbb{E}[\mathcal{R}(g_{\theta'})] - \mathcal{R}^* = \mathcal{O}\left(\sqrt{\frac{W \cdot K \log(N)}{N}} + \frac{1}{\sqrt{W}}\right). \quad (3.20)$$

Si de plus, le nombre de neurones est choisi tel que $W = \mathcal{O}(\sqrt{N/K \log(N)})$, alors :

$$\mathbb{E}[\mathcal{R}(g_{\theta'})] - \mathcal{R}^* = \mathcal{O}\left(\left(\frac{W \cdot K \log(N)}{N}\right)^{\frac{1}{4}}\right). \quad (3.21)$$

Enfin, il reste à prouver qu'il est possible d'apprendre le jeu de paramètres optimal θ' , malgré le fait qu'optimiser un réseau de neurones disposant d'une couche cachée est un problème non convexe. Bien qu'aujourd'hui nous utilisions des algorithmes de rétro-propagation convergeant généralement vers des minima locaux, il est néanmoins possible d'obtenir le minimum global avec un algorithme à complexité non polynomiale par rapport à l'ensemble des paramètres, l'apprentissage d'un tel réseau étant un problème NP-complet. Ce type d'apprentissage n'est cependant que rarement réalisable dans la pratique. Pour un nombre de neurones W et une dimension des données d'entrée \mathbf{x} fixée, [Farágó & Lugosi \(1993\)](#) avaient cependant proposé un algorithme polynomial par rapport à N . Plus récemment, des articles tels que celui de [Du et al. \(2019\)](#) ou [Li & Liang \(2018\)](#), permettent de montrer que sous certaines conditions relativement contraignantes, telles que $\mathcal{R}^* = 0$, il est possible de faire converger une architecture de réseaux de neurones vers le minimum global, avec une descente de gradient et pour une complexité polynomiale par rapport à N . Leur démarche est ainsi d'étudier la dynamique de l'algorithme d'optimisation pour une architecture de réseau neuronal spécifique. La démonstration de [Du et al. \(2019\)](#) repose notamment sur une hypothèse de sur-paramétrisation des réseaux de neurones profonds et en l'occurrence des ResNet au cours d'une optimisation par descente de gradient non stochastique, ce qui viendrait renforcer l'idée que les réseaux sur-paramétrés généralisent bien dans la pratique.

2.4 Théorème No Free-Lunch

Le théorème "No Free-Lunch" est connu pour transcrire le fait qu'aucun modèle ou algorithme ne fonctionne bien pour tous les problèmes de classification. En d'autres termes, il n'existe pas de modèle d'apprentissage machine capable de résoudre efficacement tout problème imaginable et, si un algorithme a de bonnes performances sur une distribution de données \mathcal{D} , il aura nécessairement de mauvaises performances sur un autre jeu de données.

No Free Lunch

Théorème 3 *Pour tout algorithme, tout jeu de données d'apprentissage de taille N et tout $\epsilon > 0$, il existe une distribution \mathcal{D} tel que $\mathcal{R}^* = 0$ et*

$$\Pr \left(\mathcal{R}(g_\theta) \geq \frac{1}{2} - \epsilon \right) = 1. \quad (3.22)$$

Ce théorème implique ainsi que pour un nombre fixé d'échantillons N , on peut construire un jeu de données qui donne des performances qu'on peut choisir arbitrairement et ce, pour n'importe quel algorithme. Le second théorème montre que pour un algorithme donné, on peut construire un jeu de données pour lequel l'algorithme convergera avec n'importe quelle vitesse souhaitée.

No Free Lunch at all

Théorème 4 *Pour tout algorithme et toute séquence (a_n) convergeant vers 0, il existe une distribution \mathcal{D} tel que $\mathcal{R}^* = 0$ et*

$$\mathcal{R}(g_\theta) \geq a_n. \quad (3.23)$$

2.5 Évaluation du classifieur

Pour sélectionner le modèle le plus fiable parmi un ensemble de classifieurs, il est rare de se contenter de choisir celui qui a la meilleure précision sur un jeu de données d'apprentissage de taille N . Ainsi, il est commun d'essayer d'estimer la borne supérieure de leur erreur d'estimation avec par exemple l'inégalité de Hoeffding tirée de la borne de Chernoff. Généralement trois différents types de bornes peuvent être employées :

- **Complexité des échantillons** : Pour tout $\epsilon > 0$ et $\delta \in]0, 1]$, à condition que $N \geq N_{\mathcal{D}, \mathcal{G}}(\epsilon, \delta)$, on a :

$$\Pr (\mathcal{R}(g_\theta) - \mathcal{R}_N(g_\theta) \geq \epsilon) \leq \delta. \quad (3.24)$$

- **Majoration de la queue de probabilité** : Pour tout $\epsilon > 0$ et $n \in \mathbb{N}$ on a :

$$\Pr (\mathcal{R}(g_\theta) - \mathcal{R}_N(g_\theta) \geq \epsilon) \leq \delta_{\mathcal{D}, \mathcal{G}}(n, \epsilon). \quad (3.25)$$

- **Majoration de l'erreur** : Pour tout $\delta \in]0, 1]$ et $n \in \mathbb{N}$, on a :

$$\Pr (\mathcal{R}(g_\theta) - \mathcal{R}_N(g_\theta) \leq \epsilon_{\mathcal{D}, \mathcal{G}}(n, \delta)) \geq 1 - \delta. \quad (3.26)$$

Il est nécessaire de noter qu'évaluer un classifieur quand le problème est séparable ou défini qu'avec un nombre limité d'exemples, comme c'est le cas avec le jeu d'image CIFAR-10, pose un problème majeur dans le sens où il n'existe pas un unique classifieur tel que $\mathcal{R}(f) = \mathcal{R}^* = 0$.

Pour un tel jeu de données, plusieurs modèles g_θ d'une même architecture profonde sur-paramétrée \mathcal{G} , peuvent atteindre une erreur empirique et de test nul, sans pour autant être mathématiquement équivalents. En d'autres termes, comme le précisent [Poggio & Liao \(2018\)](#), imposer une erreur empirique nulle fournit un système d'équations qui ont un grand nombre de solutions de base dégénérées au sein des différents ensembles de paramètres θ . Cela implique que ces solutions existent dans différents bassins, plat sur un grand nombre d'axes, atteignant le minimum global du paysage défini par la fonction de risque empirique. De tels minima plats sont d'ailleurs susceptibles de mieux généraliser que des minima nets. En d'autres termes, il peut exister plusieurs minima globaux ayant la même erreur empirique mais généralisant de façon différente, ce qui pourrait être une raison pour laquelle les réseaux profonds sur-paramétrés ne semblent pas toujours surapprendre sur de tels jeux de données, en utilisant toutefois certaines méthodes de régularisation.

3 Théorie de Vapnik-Tchervonenkis

3.1 Introduction

Dans l'apprentissage statistique, la théorie de Vapnik et Tchervonenkis ou théorie VC, vise à donner des outils permettant d'analyser un processus d'apprentissage. La théorie VC permet notamment de caractériser la stabilité ou généralisation d'une classe de fonction \mathcal{G} indénombrable, qui serait définie par un algorithme d'apprentissage tel qu'un réseau de neurones disposant d'une certaine architecture. Elle a notamment pour but de caractériser la vitesse de convergence d'un processus d'apprentissage et d'identifier comment contrôler cette vitesse de convergence ainsi que la capacité de généralisation du processus d'apprentissage durant la conception de l'architecture.

3.2 Coefficient de pulvérisation

Quand l'ensemble de fonction \mathcal{G} est indénombrable, en se référant à [Bousquet et al. \(2003\)](#), l'inégalité de Hoeffding n'est pas valable et il faut alors observer l'ensemble de fonctions projeté sur les échantillons d'entraînement, i.e. pour un ensemble de couples entrée sortie $z^{(1)}, \dots, z^{(N)}$, où $z^{(i)} = (\mathbf{x}^{(i)}, y^{(i)})$, on considère :

$$\mathcal{G}_{z^{(1)}, \dots, z^{(N)}} = \{(g_\theta(z^{(1)}), \dots, g_\theta(z^{(N)})) : g_\theta \in \mathcal{G}\}. \quad (3.27)$$

La taille de cet ensemble est le nombre de façons possibles que les données $(z^{(1)}, \dots, z^{(N)})$ peuvent être classifiées. Puisque les fonctions de \mathcal{G} ne peuvent prendre qu'un nombre fini de valeurs, l'ensemble (3.27) est toujours fini, quelle que soit la taille de \mathcal{G} . On définit le coefficient de pulvérisation comme le nombre maximum de façons que N points peuvent être classifiés par une classe de fonction \mathcal{G} :

$$\mathcal{S}_{\mathcal{G}}(N) = \sup_{(z^{(1)}, \dots, z^{(N)})} |\mathcal{G}_{z^{(1)}, \dots, z^{(N)}}|_c, \quad (3.28)$$

avec $|\mathcal{A}|_c$ le cardinal de l'ensemble \mathcal{A} .

Théorème de Vapnik-Tchervonenkis

Théorème 5 En considérant des fonctions binaires $g_\theta \in \mathcal{G}$, pour tout $\delta > 0$, on a, avec une probabilité au moins $1 - \delta$:

$$\forall g_\theta \in \mathcal{G}, \mathcal{R}(g_\theta) \leq \mathcal{R}_N(g_\theta) + 2\sqrt{2 \cdot \frac{\log(\mathcal{S}_{\mathcal{G}}(2N)) + \log(2/\delta)}{N}}. \quad (3.29)$$

3.3 Dimension de Vapnik-Tchervonenkis

Par la suite nous considérerons des fonctions g_θ ne pouvant prendre que $C = 2$ valeurs différentes en sortie, soit une décision binaire. Puisque g_θ ne peut prendre que 2 valeurs différentes, on a $\mathcal{S}_{\mathcal{G}}(N) \leq 2^N$. Dans le cas où $\mathcal{S}_{\mathcal{G}}(N) = 2^N$, il existe un jeu de données de taille N tel que la classe de fonction \mathcal{G} peut générer n'importe quelle classification avec ces points. On dit alors que \mathcal{G} pulvérise l'ensemble d'entraînement. On définit la dimension VC d'une classe \mathcal{G} comme étant le plus grand nombre N tel que :

$$\mathcal{S}_{\mathcal{G}}(N) = 2^N, \quad (3.30)$$

en d'autres termes, la dimension VC d'une classe \mathcal{G} est la taille du plus grand jeu d'entraînement qu'elle peut pulvériser. Dans \mathbb{R}^d par exemple, la classe des hyperplans peut pulvériser jusqu'à $d + 1$ points, mais pas $d + 2$. Si \mathcal{G} une classe de fonction de dimension VC finie h .

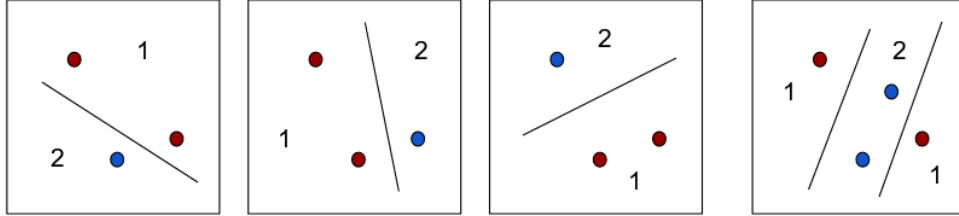


Figure 3.1: Pulvérisation de 3 points appartenant à deux classes différentes par un classifieur linéaire dans \mathbb{R}^2 , et exemple impossible à classifier avec 4 points.

Alors, pour tout $N \in \mathbb{N}$:

$$\mathcal{S}_{\mathcal{G}}(N) = \sum_{i=0}^h \binom{N}{i} \quad (3.31)$$

et pour tout $N \geq h$ (Lemme de Sauer) :

$$\mathcal{S}_{\mathcal{G}}(N) = \left(\frac{eN}{h}\right)^h. \quad (3.32)$$

En reprenant le théorème (5) on obtient alors que si \mathcal{G} a une dimension VC h , on a avec une probabilité au moins $1 - \delta$:

$$\forall g_\theta \in \mathcal{G}, \mathcal{R}(g_\theta) \leq \mathcal{R}_N(g_\theta) + 2\sqrt{2 \cdot \frac{h \log(\frac{2eN}{h}) + \log(\frac{2}{\delta})}{N}}. \quad (3.33)$$

Une interprétation de la dimension VC et du coefficient de pulvérisation est ainsi qu'ils mesurent la taille effective de la classe de fonction, c'est-à-dire la taille de la projection de la classe sur un jeu d'échantillons fini. De plus, cette mesure ne fait pas seulement que compter le nombre de fonctions dans la classe mais dépend de la géométrie de la classe ou plutôt de ses projections. Enfin, la "finitude" de la dimension VC garantit que le risque empirique converge uniformément vers le risque réel, pour la classe de fonctions \mathcal{G} .

Un aspect important de la dimension VC est qu'elle est indépendante de la distribution des $z^{(i)}$, ce qui permet de donner des bornes valables pour n'importe quelle distribution. Il est cependant possible d'obtenir des résultats dépendant de la distribution des données et pour ce faire, on introduit $|\mathcal{G}_{z^{(1)}, \dots, z^{(N)}}|_c$. L'entropie VC est alors définie comme suit :

$$H_{\mathcal{G}}(N) = \log(\mathbb{E}[|\mathcal{G}_{z^{(1)}, \dots, z^{(N)}}|_c]). \quad (3.34)$$

Théorème de l'entropie de Vapnik-Tchervonenkis

Théorème 6 *Pour tout $\delta > 0$, on a, avec une probabilité au moins $1 - \delta$*

$$\forall g_{\theta} \in \mathcal{G}, \mathcal{R}(g_{\theta}) \leq \mathcal{R}_N(g_{\theta}) + 2\sqrt{2 \cdot \frac{H_{\mathcal{G}}(2N) + \log(\frac{2}{\delta})}{N}}. \quad (3.35)$$

3.4 Nombre de recouvrement

On définira par la suite :

$$\mathcal{F} = \{(f_{\theta} : (\mathbf{x}, y) \mapsto \mathbb{1}_{g_{\theta}(\mathbf{x}) \neq y}) : g_{\theta} \in \mathcal{G}\}, \quad (3.36)$$

avec $g_{\theta} \in \mathcal{G}$ une fonction paramétrée par θ et à valeur réelle. On définit à partir de la classe de fonction \mathcal{F} la mesure aléatoire

$$d_{\mathcal{S}}(f, f') = \frac{1}{N} |\{(f(\mathbf{x}^{(i)}) \neq f'(\mathbf{x}^{(i)})) : i = 1, \dots, N\}|_c, \quad (3.37)$$

qui est la distance de Hamming normalisée de la "projection" sur les échantillons. À partir de cette mesure, on dit qu'un ensemble de fonctions $f_{\theta_1}, \dots, f_{\theta_L}$ recouvre \mathcal{F} avec un rayon ϵ_d si :

$$\mathcal{F} \subset \bigcup_{l=1}^L \mathcal{B}(f_{\theta_l}, \epsilon_d), \quad (3.38)$$

avec $\mathcal{B}(\cdot, \cdot)$ une boule. Le nombre de recouvrement de \mathcal{F} , avec un rayon ϵ_d et par rapport à la mesure $d_{\mathcal{S}}$, est dénoté $\mathcal{N}(\mathcal{F}, \epsilon_d, d_{\mathcal{S}})$ et est la taille minimale de recouvrement ayant pour rayon ϵ_d . Pour tout $t > 0$:

$$\mathbb{P}(\exists g_{\theta} \in \mathcal{G}, \mathcal{R}(g_{\theta}) > \mathcal{R}_N(g_{\theta}) + t) \leq 8\mathbb{E}[\mathcal{N}(\mathcal{F}, \epsilon_d, d_{\mathcal{S}})] \cdot e^{-\frac{Nt^2}{128}}. \quad (3.39)$$

Si \mathcal{G} une classe de fonctions de dimension VC h , alors, pour tout $\epsilon_d > 0$, pour tout ensemble d'entraînement \mathcal{S} de taille N , et tout échantillon d'entraînement,

$$\mathcal{N}(\mathcal{G}, \epsilon_d, d_{\mathcal{S}}) \leq C \cdot \frac{h(4e)^h}{\epsilon_d^h}. \quad (3.40)$$

La borne du nombre de recouvrement est une généralisation de la borne d'entropie VC où l'échelle serait adaptée à l'erreur. Il s'avère que ce résultat peut être amélioré en considérant toutes les différentes échelles.

3.5 Contrôle de capacité par la taille du réseau

En se référant à [Neyshabur et al. \(2017\)](#), si nous considérons la capacité statistique d'une classe de fonction en termes du nombre d'exemples N nécessaires pour assurer la généralisation, cette notion est étroitement liée au nombre maximum d'exemples sur lesquels on peut obtenir une petite erreur d'apprentissage même avec des étiquettes aléatoires. Aussi, pour tout modèle, si les paramètres ont une précision finie, on devrait pouvoir borner la capacité du modèle de façon linéaire par rapport au nombre total de paramètres. Même sans faire d'hypothèse sur la précision des paramètres, la dimension VC des réseaux à propagation avant peut être bornée en termes de nombre de paramètres $\dim(\theta)$ comme le montrent [Shalev-Shwartz & Ben-David \(2014\)](#). En particulier, [Harvey et al. \(2017\)](#), à la suite de [Bartlett et al. \(1998\)](#), proposent une limite à un facteur logarithmique près de la capacité des réseaux à propagation avant disposant de fonctions d'activation ReLU :

$$h = \mathcal{O}(L \cdot \dim(\theta)), \quad (3.41)$$

avec L le nombre de couches du réseau et h la dimension VC de la classe \mathcal{G} du réseau. Dans une configuration sur-paramétrée, i.e. où le nombre de paramètres est supérieur au nombre d'échantillons, les mesures de complexité qui dépendent du nombre total de paramètres sont cependant trop larges et ne permettent pas d'expliquer le comportement de généralisation. En effet, les réseaux de neurones utilisés dans la pratique ont souvent beaucoup plus de paramètres que le nombre d'échantillons à disposition. Comme le font remarquer [Zhang et al. \(2021\)](#), ils peuvent parfaitement s'adapter aux labels du jeu d'entraînement et ce, même quand ces labels sont aléatoires, bien que dans ce cas les modèles ne généralisent pas. De plus, la mesure de la complexité en termes de nombre de paramètres ne peut expliquer les réductions de l'erreur de généralisation souvent observées lorsque le nombre de couches cachées augmente.

4 Mesure de complexité

4.1 Nécessité de repenser la généralisation pour l'apprentissage profond

En montrant que les architectures profondes pouvaient obtenir d'excellentes performances de généralisation sur des jeux de données telles que CIFAR-10, tout en pouvant parfaitement classifier des données labélisées aléatoirement, [Zhang et al. \(2017\)](#) ont mis en avant le fait qu'il était nécessaire de repenser la généralisation au sein de l'apprentissage automatique pour l'apprentissage profond. En effet, malgré l'existence de nombreux résultats théoriques qui impliquent de faibles écarts de généralisation pour des hypothèses de complexité limitée, il est fréquent d'obtenir des faibles écarts de généralisation en combinaison avec une grande complexité de l'espace d'hypothèses, une non-robustesse ou encore des minima nets instables aux perturbations. Ainsi, évaluer la complexité en termes de nombre de paramètres ou de mesures uniformes sur toutes les fonctions représentables par une architecture donnée, n'est pas une méthode suffisante dans la pratique pour expliquer la capacité de généralisation de réseaux de neurones entraînés. Aussi, il est nécessaire de voir ces hypothèses comme

des conditions suffisantes mais pas nécessaires. Ceci est facilement compréhensible du fait que l'écart de généralisation est complètement déterminé par le triplet $(\mathcal{D}, \mathcal{S}, g_\theta)$, alors qu'on base la théorie VC uniquement, ou tout du moins principalement, sur la classe de fonction \mathcal{G} .

Malgré les succès des réseaux de neurones profonds, reflétés par leurs excellentes performances empiriques, les raisons de leurs bonnes performances de généralisation restent aujourd'hui encore obscures. Bien que différents travaux aient attribué la raison pour laquelle les réseaux profonds surapprennent moins que des algorithmes d'apprentissage classiques à des régularisations implicites durant leur apprentissage, aujourd'hui encore aucun consensus n'a été trouvé. En effet, beaucoup de corrélations entre mesures de complexité et généralisation ont été observées sans forcément identifier une cause précise. Certains travaux, tels que [Valle-Perez et al. \(2018\)](#), soutiennent quant à eux que l'espace des paramètres de nombreuses architectures profondes devrait être biaisé de manière à converger vers des fonctions simples avec l'utilisation de la SGD. Dans leur article, [Valle-Perez et al. \(2018\)](#) éprouvent cette hypothèse en appliquant une limite de probabilité-complexité très générale, dérivée de la théorie de l'information algorithmique, sur des réseaux complètement connectés et convolutifs.

4.2 Objectifs des mesures de complexité

Comprendre quelles sont les propriétés des réseaux profonds leur permettant de généraliser dans le cadre d'un problème, est un sujet de plus en plus abordé qui pourrait s'avérer utile pour parvenir à concevoir des modèles plus fiables. Dans ce sens, des travaux récents comme ceux de [Jiang et al. \(2019\)](#), ont exploré de nouvelles mesures de complexité qui doivent permettre de quantifier la complexité de la fonction que représente un réseau et être de bons prédicteurs du degré de généralisation de ce dernier. Dans la pratique, ces mesures conçues manuellement doivent refléter certaines particularités des architectures profondes qui contredisent notre compréhension de l'apprentissage automatique conventionnel sur le compromis entre l'expressivité d'un modèle et sa capacité à généraliser. On peut notamment penser à l'amélioration de la généralisation des architectures profondes avec la taille du réseau.

Dans leurs travaux, [Jiang et al. \(2019\)](#) différencient les mesures de complexités en différentes familles : En premier lieu viennent les bornes de généralisation théoriques telles que PAC-Bayes, traitées par [McAllester \(1999\)](#) et [Neyshabur et al. \(2017\)](#), la dimension VC de [Vapnik & Chervonenkis \(2015\)](#) ou encore les bornes basées sur des normes de paramètres appris, proposées par [Neyshabur et al. \(2014\)](#). On peut également considérer des mesures basées sur des observations empiriques, telles que la mesure de netteté des minima présentée par [Keskar et al. \(2016\)](#), une méthode Fisher-Rao de [Liang et al. \(2019\)](#), la distance des poids à leur initialisation proposée par [Nagarajan & Kolter \(2019b\)](#) ou encore la méthode Path-SGD de [Neyshabur et al. \(2017\)](#). Enfin, il est possible de considérer des mesures d'optimisation basées sur la vitesse de l'algorithme d'optimisation, comme l'ont proposé [Hardt et al. \(2016\)](#) ainsi que [Wilson et al. \(2017\)](#), ou encore l'ampleur du bruit de gradient explorée par [Smith & Le \(2017\)](#) et [Chaudhari & Soatto \(2018\)](#). Certains travaux ont également envisagé d'examiner la capacité du modèle en conjonction avec l'algorithme d'apprentissage utilisé en additionnant plusieurs de ces mesures déjà citées.

Dans leur article, [Neyshabur et al. \(2017\)](#) exposent quelques éléments permettant de dire

en quoi une mesure de complexité peut être appropriée pour expliquer la généralisation au sein de l'apprentissage profond. Premièrement, une telle mesure de complexité devrait être suffisante pour assurer la généralisation. Deuxièmement, les réseaux appris ayant de bonnes performances dans la pratique, devraient être de faible complexité pour cette mesure. Cela peut arriver si notre algorithme d'optimisation nous oriente vers des modèles de moindre complexité pour cette mesure et ces modèles de faible complexité sont toujours capables de modéliser des fonctions capturant la distribution des données. Dernièrement, la mesure de complexité doit aider à expliquer plusieurs phénomènes observés empiriquement dans la littérature et qui restent inexplicables par des notions de complexité plus simples :

- Possibilité d'obtenir une erreur d'apprentissage empirique nulle sur des labels aléatoires et ce, en utilisant une architecture pour laquelle l'apprentissage avec étiquettes réelles conduit à une généralisation faible. On s'attendrait alors à ce que des modèles de réseaux appris à l'aide de labels réels, pour lesquels nous avons de bonnes performances de généralisation, aient une complexité beaucoup plus faible que ceux appris à l'aide de labels aléatoires et généralisant mal, sous la mesure suggérée.
- L'augmentation du nombre de couches cachées et ainsi du nombre de paramètres, peut conduire à une diminution de l'erreur de généralisation même lorsque l'erreur d'apprentissage ne diminue pas, comme l'observent [Neyshabur et al. \(2014\)](#). Nous nous attendrions alors à voir la mesure de complexité diminuer à mesure que nous augmentons le nombre d'unités cachées, pour ce type d'application.
- Lors de l'apprentissage d'une même architecture, avec le même ensemble d'apprentissage, en utilisant deux méthodes d'optimisation différentes avec des choix d'algorithmiques ou de paramètres différents, si une méthode entraîne une meilleure généralisation même si les deux conduisent à une erreur d'apprentissage nulle, nous nous attendrions à voir une corrélation entre mesure de complexité et capacité de généralisation parmi ces deux modèles.

[Valle-Pérez & Louis \(2020\)](#) proposent également de vérifier 7 propriétés désirées dans une mesure de complexité pour l'apprentissage supervisé :

- Pouvoir s'adapter correctement à la complexité des données. Dans le domaine acoustique, on pensera notamment au SNR.
- Pouvoir s'adapter correctement à la taille de l'ensemble d'entraînement.
- Capturer les différences entre les architectures.
- Capturer les différences entre les algorithmes d'optimisation.
- Ne pas être trop éloigné de l'erreur de généralisation.
- Pouvoir être calculée de façon efficace. En effet, beaucoup de bornes de généralisation ne sont pas calculables dans la pratique.
- La prédiction doit être rigoureuse, dans le sens où elle est accompagnée d'un théorème qui garantit sa validité pour un ensemble d'hypothèses spécifiées qui doivent être satisfaites pour le domaine d'application.

4.3 Complexité de Rademacher

Si on se réfère à [Bousquet et al. \(2003\)](#), la complexité de Rademacher est une mesure de capacité pouvant donner des bornes plus fines que celles de la théorie VC en effectuant un calcul sur un ensemble de données. En reprenant les notations f_θ introduite dans l'équation (3.36) et en considérant le risque $\mathcal{R}(f_\theta)$ pour la fonction de perte idéale "0-1", $\mathcal{L}_{0-1}(z, y) = \mathbb{1}_{\{z=y\}}$, on s'intéresse à une quantité associée aux processus empiriques souvent étudiés, qui est le supremum de l'excès de risque :

$$\sup_{f_\theta \in \mathcal{F}} \mathcal{R}(f_\theta) - \mathcal{R}_N(f_\theta). \quad (3.42)$$

On définira la mesure empirique signée telle que :

$$R_N(f_\theta) = \frac{1}{N} \sum_{i=1}^N \sigma_i f_\theta(z^{(i)}), \quad (3.43)$$

où σ_i est une valeur aléatoire de Rademacher, i.e. indépendante et prenant des valeurs dans $\{-1, 1\}$ avec une probabilité $\frac{1}{2}$. On dénotera \mathbb{E}_σ l'espérance par rapport aux variables de Rademacher, tandis que \mathbb{E} correspondra à l'espérance par rapport à toutes les variables aléatoires, soit les données d'entraînement et de test ainsi que les variables de Rademacher. Pour toute classe de fonction \mathcal{F} , la complexité de Rademacher ou moyenne de Rademacher est définie telle que suit :

$$R(\mathcal{F}) = \mathbb{E} \left[\sup_{f_\theta \in \mathcal{F}} R_N(f_\theta) \right] \quad (3.44)$$

et la complexité de Rademacher conditionnelle est définie comme :

$$R_N(\mathcal{F}) = \mathbb{E}_\sigma \left[\sup_{f_\theta \in \mathcal{F}} R_N(f_\theta) \right]. \quad (3.45)$$

Borne de la complexité de Rademacher

Théorème 7 *Pour tout $\delta > 0$, on a, avec une probabilité au moins $1 - \delta$*

$$\forall f_\theta \in \mathcal{F}, \mathcal{R}(f_\theta) \leq \mathcal{R}_N(f_\theta) + 2R(\mathcal{F}) + \sqrt{\frac{\log(\frac{1}{\delta})}{2N}} \quad (3.46)$$

et également avec une probabilité au moins $1 - \delta$

$$\forall f_\theta \in \mathcal{F}, \mathcal{R}(f_\theta) \leq \mathcal{R}_N(f_\theta) + 2R_N(\mathcal{F}) + \sqrt{\frac{2 \log(\frac{2}{\delta})}{N}}. \quad (3.47)$$

[Bousquet et al. \(2003\)](#) prouvent ce résultat grâce à une variante de l'inégalité de Hoeffding des classes \mathcal{F} dénombrables. Cette variante est toujours une inégalité de concentration dans le sens

où, lorsque N augmente, la moyenne empirique se concentre autour de l'espérance. Elle permet alors de généraliser ce résultat aux fonctions qui dépendent de variables aléatoires i.i.d. :

Inégalité de McDiarmid

Théorème 8 *Supposons que pour tous échantillons $z^{(i)}, i = 1, \dots, N$ appartenant à \mathcal{S} ,*

$$\sup_{z^{(1)}, \dots, z^{(N)}, z'^{(i)}} |F(z^{(1)}, \dots, z^{(i)}, \dots, z^{(N)}) - F(z^{(1)}, \dots, z'^{(i)}, \dots, z^{(N)})| \leq c, \quad (3.48)$$

avec $F = F(z^{(1)}, \dots, z^{(N)}) = R_N(\mathcal{F})$ et $\mathbb{E}[F] = R(\mathcal{F})$, alors pour tout $\epsilon > 0$,

$$\mathbb{P}(|F - \mathbb{E}[F]| > \epsilon) \leq 2e^{-\frac{2\epsilon^2}{nc^2}}. \quad (3.49)$$

Une signification de ce résultat proposée par [Bousquet et al. \(2003\)](#) est ainsi que, dès lors que l'on a une fonction dépendant de N variables aléatoires indépendantes dont les variations sont bornées lorsqu'une variable est modifiée, cette fonction satisfera une inégalité semblable à celle de Hoeffding.

4.4 Borne de généralisation PAC-Bayésienne déterministe

Le cadre PAC-bayésien, comme il est introduit par [McAllester \(1999\)](#), donne la possibilité d'obtenir une borne de généralisation pour un classifieur stochastique : la variable aléatoire $\tilde{\theta}$ dans l'espace des paramètres a une distribution apprise à partir d'un ensemble d'apprentissage \mathcal{S} de taille N .

Borne de PAC-Bayes

Théorème 9 *Soit \mathcal{D} une distribution à priori dans l'espace des paramètres et choisie indépendamment des données d'apprentissage. Avec l'erreur 0-1 du classifieur stochastique, on a avec une probabilité au moins $1 - \delta$ sur le tirage de l'ensemble d'apprentissage \mathcal{S} que :*

$$\mathbb{E}_{\tilde{\theta}}[\mathcal{R}(f_{\tilde{\theta}})] \leq \mathbb{E}_{\tilde{\theta}}[\mathcal{R}_N(f_{\tilde{\theta}})] + \mathcal{O}\left(\sqrt{\frac{KL(\tilde{\theta}||\tilde{w})}{N}}\right), \quad (3.50)$$

avec $KL(\cdot||\cdot)$ la KL-divergence et $\tilde{\theta}$ ainsi que \tilde{w} , deux variables Gaussiennes de matrice de covariance $\sigma^2 \cdot I_{d_{\tilde{\theta}}}$ ayant respectivement pour moyenne θ , les poids appris sur l'ensemble d'apprentissage et p , les poids d'initialisation.

Dans leurs travaux, [Nagarajan & Kolter \(2019a\)](#) proposent d'étendre cette borne à un classifieur g_{θ} déterministe tel qu'un réseau de neurones. Pour ce faire, il faut remplacer les pertes d'entraînement et de test du classifieur stochastique par celles du classifieur déterministe, ce qui implique d'introduire des termes supplémentaires dans la borne supérieure pour tenir

compte des perturbations subies par les valeurs de perte à cause des perturbations Gaussiennes des paramètres. Obtenir une borne étroite pour ces termes nécessite alors que le réseau soit résilient au bruit sur les données d'entraînement et de test respectivement. Il faut ainsi espérer que quand l'algorithme apprend des poids tels que le réseau peut résister à du bruit sur les données d'entraînement, alors cette résilience au bruit peut être généralisée pour les données test. Pour vulgariser, la résilience au bruit d'un réseau correspond à son taux de réaction aux perturbations aléatoires de ses paramètres, en fonction d'une entrée donnée. Dans leurs travaux, [Nagarajan & Kolter \(2019a\)](#) caractérisent la résilience au bruit à travers des conditions sur la façon dont les poids du modèle interagissent les uns avec les autres, pour une entrée donnée. Par exemple, quand les valeurs de pré-activation des unités cachées de la couche l ont une magnitude supérieure à une constante positive.

Considérons un classifieur pour lequel nous pouvons définir R conditions différentes, qui, lorsqu'elles sont satisfaites pour une entrée donnée, nous aideront à garantir la résilience au bruit du classificateur à cette entrée, c'est-à-dire à limiter la perturbation de la sortie sous des perturbations aléatoires de paramètres. R dans le cas des réseaux profonds peut correspondre à une condition pour chaque couche, impliquant que R augmente avec le nombre de couches L . En particulier, soit la condition r , une borne impliquant un ensemble particulier de propriétés des poids dépendant de l'entrée \mathbf{x} et notée $\rho_{r,1}(\theta, \mathbf{x}, y), \rho_{r,2}(\theta, \mathbf{x}, y)$. On dira ainsi que les poids θ satisfont la condition r sur l'entrée (\mathbf{x}, y) si :

$$\forall q, \rho_{r,q}(\theta, \mathbf{x}, y) > \Delta_{r,q}^*, \quad (3.51)$$

où q est l'indice de l'élément dans l'ensemble. Aussi si les poids θ respectent les R conditions ci-dessus sur l'ensemble d'apprentissage \mathcal{S} alors, pour tout $\delta > 0$, on a avec une probabilité au moins $1 - \delta$ sur le tirage de l'ensemble d'apprentissage \mathcal{S} depuis \mathcal{D} :

$$\mathcal{R}(g_\theta) \leq \mathcal{R}_N(g_\theta) + \mathcal{O} \left(R \sqrt{2 \cdot \frac{KL(\mathcal{N}(\theta, \sigma_\star^2 \cdot I_D) || \mathcal{D}) + \ln(2NR/\delta)}{N-1}} \right), \quad (3.52)$$

où D est la dimension de θ et σ_\star est un écart type précisé dans l'article de [Nagarajan & Kolter \(2019a\)](#). Dans leurs travaux, [Vakanski & Xian \(2021\)](#) proposent également des mesures PAC-bayésiennes μ liées à l'initialisation des poids. En notant $W^{(l)}$ et $W_0^{(l)}$ les poids de la $l^{\text{ème}}$ couche d'un réseau respectivement à la fin de l'entraînement et à l'initialisation, $\theta = \{W^{(1)}, \dots, W^{(L)}\}$, $\theta_0 = \{W_0^{(1)}, \dots, W_0^{(L)}\}$, D la dimension de θ et θ_0 , $\tilde{\theta} = \mathcal{N}(\theta, \sigma^2 \cdot I_D)$ ainsi que $\tilde{\theta}_0 = \mathcal{N}(\theta_0, \sigma^2 \cdot I_D)$, ces mesures PAC bayésiennes emploient alors la KL-divergence $KL(\tilde{\theta} || \tilde{\theta}_0)$.

Remarque

Le cadre PAC-Bayésien a également permis de développer des méthodes de régularisation pour l'apprentissage des architectures de réseau de neurones, comme le font [Guedj \(2019\)](#).

4.5 Mesure de complexité liée à la norme et à la marge

Comme le précisent [Neyshabur et al. \(2017\)](#), la capacité des prédicteurs linéaires peut être contrôlée à travers une régularisation de leurs normes L^2 , indépendamment de leur nombre de paramètres, ce qui a incité d'autres travaux à utiliser ces normes comme mesures de complexité pour les réseaux de neurones à propagation avant f_w , disposant d'activations ReLU. [Neyshabur et al. \(2017\)](#) montrent ainsi que la capacité d'un réseau de L couches peut être bornée grâce aux normes L^1 des poids des couches cachées et est proportionnelle à

$$\prod_{l=1}^L \|w^{(l)}\|_{1,\infty}^2, \quad (3.53)$$

où $\|w^{(l)}\|_{p,q}^2$ est la norme L^q des normes L^p des poids des neurones $w^{(l)}$ de la couche l . Cette borne peut également être étendue pour la plupart des normes L^p , ainsi que $\prod_{l=1}^L \|w^{(l)}\|_{2,2}^2$. En l'absence de contrainte de régularisation, l'entraînement des réseaux de neurones pouvant faire tendre les poids de l'architecture vers l'infini pour minimiser la fonction de perte, il est nécessaire de prendre également en compte l'échelle de valeurs des poids au sein de la mesure de complexité, ce qui est possible avec une "marge" de prédiction. La marge d'un point \mathbf{x} pour le modèle f_w , une fonction de perte \mathcal{L} et au sein d'une donnée (\mathbf{x}, y) , est définie comme étant l'écart maximum entre le score du label correct et le pire label possible :

$$\mathcal{L}(f_w(\mathbf{x}), y) - \max_{c \neq y} \mathcal{L}(f_w(\mathbf{x}), c). \quad (3.54)$$

Afin de mesurer l'échelle de valeurs obtenue sur un ensemble complet d'apprentissage, une approche simple consiste à considérer la "marge dure", qui est la marge minimale parmi tous les points d'entraînement. Cette définition est cependant très sensible aux points extrêmes ainsi qu'à la taille de l'ensemble d'entraînement. Aussi, [Neyshabur et al. \(2017\)](#) considèrent plutôt une notion plus robuste qui permet à une petite partie des données de violer la marge. Pour un jeu d'apprentissage donné et une valeur faible $\epsilon > 0$, on définit la marge γ_ϵ comme étant la plus petite valeur de γ telle que $\lceil \epsilon N \rceil$ point de données ont une marge inférieure à γ , où N est la taille de l'ensemble d'apprentissage. On peut alors proposer une mesure basée sur la norme L^2 :

$$\frac{1}{\gamma_\epsilon^2} \prod_{l=1}^L 4 \|w^{(l)}\|_{2,2}^2, \quad (3.55)$$

qui permet à [Neyshabur et al. \(2017\)](#) de montrer que, pour une même architecture, un modèle entraîné avec label aléatoire a une mesure plus élevée qu'un modèle entraîné avec de vrais labels.

4.6 Mesure de complexité lié à la netteté des minima

La notion de netteté des minima comme mesure de généralisation, ou "sharpness" comme suggérée par [Keskar et al. \(2016\)](#), correspond à la robustesse aux perturbations adversariales sur l'espace des paramètres. Pour une architecture de réseau de neurones à propagation avant f_w , où $w^{(l)}$ sont les paramètres de la couche l , [Neyshabur et al. \(2017\)](#) proposent la définition suivante :

$$\zeta_\alpha(w) = \frac{\max_{|\nu_i| \leq \alpha(|w^{(l)}|+1)} \mathcal{R}_N(f_{w+\nu}) - \mathcal{R}_N(f_w)}{1 + \mathcal{R}_N(f_w)}, \quad (3.56)$$

où le dénominateur vaut approximativement 1 en pratique lors d'un entraînement et ainsi,

$$\zeta_\alpha(w) \approx \max_{|\nu_l| \leq \alpha(|w^{(l)}|+1)} \mathcal{R}_N(f_{w+\nu}) - \mathcal{R}_N(f_w). \quad (3.57)$$

La notion de netteté définie de cette manière ne capture malheureusement pas toujours le comportement de généralisation, puisqu'elle dépend de l'échelle de valeurs des poids du modèle et peut donc être augmentée ou diminuée artificiellement en multipliant les paramètres par une constante, ce qui, dans certains réseaux (linéaires), ne change pas le comportement de classification. Par conséquent, la netteté seule n'est pas suffisante pour contrôler la capacité du réseau. Pour le cas des prédicteurs linéaires, la netteté ne dépend que de la norme du prédicteur. En revanche, pour les réseaux multi-couches, l'interaction entre les couches joue un rôle majeur et, par conséquent, deux réseaux différents de même normes peuvent avoir des valeurs de netteté radicalement différentes. Par exemple, si on considère un réseau où certains sous-ensembles de couches, malgré une norme non nulle, interagissent faiblement avec leurs voisines, ou sont presque orthogonaux les unes par rapport aux autres. Un tel réseau aura une valeur de netteté très élevée par rapport à un réseau où les couches voisines interagissent fortement.

La notion de netteté peut également être considérée dans le cadre PAC-bayésien où, avec la norme, ils permettent de contrôler la capacité du réseau et peuvent expliquer de nombreux phénomènes observés. Il devient ainsi évident que la notion de netteté ne contrôle qu'un seul des deux termes nécessaires pour contrôler la capacité d'un modèle f_w . [Neyshabur et al. \(2017\)](#) montrent dans leurs travaux que quand l'espérance du risque empirique $\mathbb{E}_\nu[\mathcal{R}_N(f_{w+\nu})]$ est plus petite que

$$\mathcal{K} = \sqrt{\frac{KL(w + \nu | \mathcal{D}) + \ln(2N/\delta)}{N}}, \quad (3.58)$$

ce qui est souvent le cas de réseaux de neurones avec des perturbations assez petites, alors on a avec une probabilité au moins $1 - \delta$ sur le tirage de l'ensemble d'apprentissage \mathcal{S} que :

$$\mathbb{E}_\nu[\mathcal{R}(f_{w+\nu})] \leq \mathcal{R}_N(f_w) + \mathbb{E}_\nu[\mathcal{R}_N(f_{w+\nu})] - \mathcal{R}_N(f_w) + 4\mathcal{K}, \quad (3.59)$$

en majorant l'erreur empirique par 1. Cette borne PAC-Bayésienne dépend ainsi de deux quantités, l'espérance de la netteté et la KL-divergence de la distribution à priori \mathcal{D} et elle est valable pour toute distribution \mathcal{D} , perturbation de distribution ν et méthode de sélection des paramètres w . En particulier attribuer à \mathcal{D} une distribution Gaussienne de moyenne nulle et de variance σ^2 , ainsi que choisir une perturbation ν qui soit Gaussienne sphérique de moyenne nulle et de variance σ^2 dans toutes les directions, donne la garantie suivante :

$$\mathbb{E}_\nu[\mathcal{R}(f_{w+\nu})] \leq \mathcal{R}_N(f_w) + \mathbb{E}_\nu[\mathcal{R}_N(f_{w+\nu})] - \mathcal{R}_N(f_w) + 4 \sqrt{\frac{1}{N} \left(\frac{\|w\|_2^2}{2\sigma^2} + \ln(2N/\delta) \right)}. \quad (3.60)$$

Les garanties de généralisation ci-dessus indiquent que le contrôle de la capacité d'un modèle dépend à la fois de l'espérance de la netteté et de la norme. Dans la borne de généralisation ci-dessus, normes et netteté interagissent de manière directe en fonction de σ , car l'augmentation de la norme avec une diminution de σ provoque une réduction de la netteté et vice versa. Il est donc important de trouver le bon équilibre entre la norme et la netteté, en choisissant σ de manière appropriée afin d'obtenir une limite raisonnable sur la capacité.

5 Optimisation par SGD

5.1 Propriétés de la SGD

Lors d'une descente de gradient stochastique (SGD), seul un petit sous-ensemble d'exemples est utilisé au sein d'une batch (lot de données) pour estimer le gradient. Ceci permet généralement de converger vers des modèles généralisants mieux qu'avec une descente de gradient classique, comme le montrent [Smith & Le \(2017\)](#). Si nous faisons l'hypothèse que cette efficacité est due à une dynamique de la SGD plutôt qu'aux propriétés générales de l'apprentissage profond, alors une régularisation implicite doit être présente lors de l'apprentissage. Une explication donnée par [Keskar et al. \(2016\)](#) pour cette observation empirique concerne la description des paysages de fonction de perte et fait l'hypothèse que la SGD préfère des minima plats aux minima nets dans lesquels la fonction de perte ne change pas trop au voisinage des minima, ce qui peut arriver quand on change les entrées. Le processus stochastique équivaldrait à injecter du bruit dans le processus d'optimisation, permettant de franchir des barrières dans le paysage d'optimisation et ainsi de pouvoir éviter certains minima nets. En particulier, [Zhu et al. \(2018\)](#) soutiennent qu'un bruit anisotropique est nécessaire pour échapper aux minima.

Le choix de la taille des batchs joue également un rôle important dans le comportement d'apprentissage d'une SGD. En effet, prendre des batchs de plus grandes tailles garantit une meilleure estimation du gradient de $\mathcal{R}_N(g_\theta)$, ce qui conduit généralement à une convergence plus rapide sur le jeu d'apprentissage. Il y a cependant un compromis à faire du point de vue de la généralisation. En effet, comme le montrent [Smith & Le \(2017\)](#), le bruit intrinsèque issu des gradients de mini-batchs fournit des effets de régularisation et augmenter leur taille fait perdre cet avantage de généralisation. Dans ce sens, [Wen et al. \(2020\)](#) proposent d'ajouter un terme de bruit aux mises à jour du gradient, dont la structure de covariance serait donnée par la matrice de Fisher diagonale de l'ensemble des mises à jour de gradient.

5.2 Régularisation implicite de la SGD par mise à jour des poids

Pour la descente de gradient ainsi que la SGD, la mise à jour des paramètres θ_u présents dans l'ensemble de paramètres θ , peut être écrite sous la forme :

$$\theta_u(t+1) = \theta_u(t) + \eta \cdot \nabla \mathcal{R}_u^{(\mathcal{B})}(g_{\theta(t)}) \quad (3.61)$$

$$= \theta_u(t) + \eta \cdot \frac{1}{|\mathcal{B}|} \sum_{i=b_0}^{b_0+|\mathcal{B}|} \nabla \mathcal{L}_u(g_{\theta(t)}(\mathbf{x}^{(i)}), y^{(i)}), \quad (3.62)$$

avec un indice u couvrant l'ensemble des paramètres. L'accumulation des mises à jours des paramètres lors de la descente de gradient tend à développer des biais vers le surapprentissage, ce que la SGD mitige en décorrélant les mises à jours proches les unes des autres. Aussi, $\nabla \mathcal{R}_u^{(\mathcal{B})}(g_{\theta(t)})$ étant une somme de plusieurs variables aléatoires indépendantes $\nabla \mathcal{L}_u(g_{\theta(t)}(\mathbf{x}^{(i)}), y^{(i)})$, il en résulte que pour des batchs de tailles assez grandes, le théorème central limite permet de faire l'approximation que l'estimateur du gradient $\nabla \mathcal{R}_u^{(\mathcal{B})}(g_{\theta(t)})$ est Gaussien. [Roberts \(2021\)](#) part de ce constat pour exprimer la différence entre les changements de l'erreur de test comparé à l'erreur d'entraînement $\delta\epsilon$ (estimation du changement de l'erreur

de généralisation) :

$$\delta\epsilon = \eta \cdot \nabla\mathcal{R}^{(\mathcal{B})}(g_\theta)(\nabla\mathcal{R}^{(\mathcal{B})}(g_\theta) - \nabla\mathcal{R}^{(\mathcal{A})}(g_\theta)) + \mathcal{O}(\eta^2), \quad (3.63)$$

$$\mathbb{E}[\delta\epsilon] = \eta \cdot \frac{\text{tr}(\Sigma_{u,v}(g_\theta))}{N} + \mathcal{O}(\eta^2), \quad (3.64)$$

avec \mathcal{A} , \mathcal{B} des batches indépendantes et $\Sigma_{u,v}(g_\theta)$, la matrice de covariance de l'estimateur du gradient supposé anisotropique et dépendant de θ . [Roberts \(2021\)](#) essaye également d'exposer en quoi la SGD est une régularisation implicite par rapport à la descente de gradient. Aussi la différence entre la descente du gradient stochastique et non stochastique peut être interprétée comme la différence de leur mise à jour des poids par le vecteur de gradient $\nabla\mathcal{R}_N(g_\theta)$:

$$\mathbb{E}[\nabla\mathcal{R}_N(g_\theta)] = -\frac{1}{2}\partial_u\mathbb{E}[\delta\epsilon]. \quad (3.65)$$

5.3 Entraînement en deux phases de la SGD et visualisation dans le plan de l'information

L'apprentissage supervisé permet d'obtenir de bonnes représentations $T(X)$ pour la prédiction des labels $y \in \mathcal{Y}$, à partir des motifs d'entrée $\mathbf{x} \in \mathcal{X}$. En considérant les différentes couches d'un réseau à propagation avant comme étant des représentations internes successives du réseau, [Shwartz-Ziv & Tishby \(2017\)](#) mettent en avant que l'apprentissage par SGD fait apparaître 2 phases : minimisation de l'erreur empirique et compression de la représentation. Pour y parvenir, une sortie de couche, T , est considérée comme une variable aléatoire unique, caractérisée par ses distributions d'encodeur, $P(T|X)$, et de décodeur, $P(Y|T)$. La qualité des représentations des couches est ensuite quantifiée en les comparant aux représentations optimales de la théorie de l'information, "Information Bottleneck".

Définition : Information mutuelle

Pour deux variables aléatoires X et Y , suivant une distribution conjointe $p(\mathbf{x}, y)$, l'information mutuelle introduite par Kullback, est définie dans l'article de [Shwartz-Ziv & Tishby \(2017\)](#) comme étant :

$$I(X; Y) = KL(p(\mathbf{x}, y) || p(\mathbf{x})p(y)) \quad (3.66)$$

$$= \sum_{\mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}} p(\mathbf{x}, y) \log \left(\frac{p(\mathbf{x}, y)}{p(\mathbf{x})p(y)} \right) \quad (3.67)$$

$$= \sum_{\mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}} p(\mathbf{x}, y) \log \left(\frac{p(\mathbf{x}|y)}{p(\mathbf{x})} \right) \quad (3.68)$$

$$= H(X) - H(X|Y). \quad (3.69)$$

avec $KL(\cdot || \cdot)$ la divergence de Kullback-Liebler et $H(X)$ ainsi que $H(X|Y)$ sont, respectivement, l'entropie et l'entropie conditionnelle de X et Y . L'information mutuelle quantifie le nombre de bits pertinents que contient en moyenne la variable d'entrée X à propos de l'étiquette Y . Le problème d'apprentissage optimal peut ainsi être défini comme la construction d'un encodeur optimal de cette information pertinente via une représentation efficace, un minimum suffisant statistique de X par rapport à Y .

Afin de calculer l'information mutuelle des couches du réseau avec les variables d'entrée et de sortie, les sorties d'activation arctan des différentes couches, sont discrétisées en 30 intervalles égaux entre -1 et 1. Ces valeurs discrétisées servent à calculer directement les distributions conjointes, sur les motifs d'entrée possibles $\mathbf{x} \in \mathcal{X}$, $P(T_i, X)$ et $P(T_i, Y) = \sum_{\mathbf{x}} P(\mathbf{x}, Y)P(T_i|\mathbf{x})$, en utilisant la chaîne de Markov $Y \rightarrow X \rightarrow T_i$ pour chaque couche cachée. Les informations mutuelles du décodeur et de l'encodeur, $I(X; T_i)$ et $I(T_i; Y)$, peuvent ensuite être déduites.

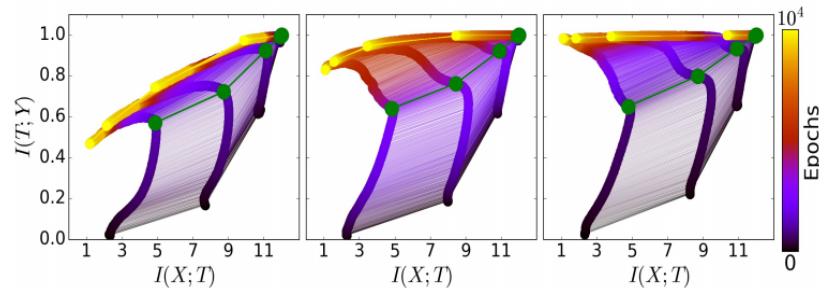


Figure 3.2: Évolution de l'information mutuelle discrète moyenne des parties encodeur $I(X; T_i)$ et décodeur $I(T_i; Y)$ pour chaque représentation T_i et 50 différents tirages de 5%, 45% et 85% du data set. Tiré de l'article de [Shwartz-Ziv & Tishby \(2017\)](#).

Comme on peut le voir sur la figure 3.2, au début de l'optimisation les couches plus profondes du réseau initialisé aléatoirement, ne parviennent pas à préserver les informations pertinentes

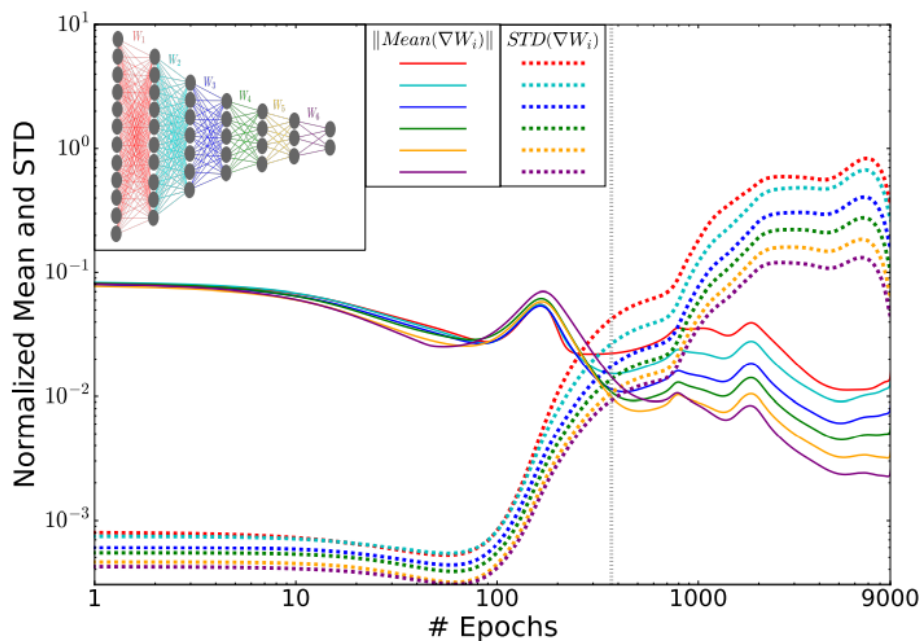


Figure 3.3: Moyenne normalisée et écart type du gradient stochastique des poids de différentes couches. Tiré de l'article de [Shwartz-Ziv & Tishby \(2017\)](#).

et il y a une forte augmentation d' I_Y vers les couches de sortie. Au cours de l'optimisation par SGD, les couches augmentent d'abord I_Y , puis diminuent considérablement I_X , comprimant ainsi la représentation. Sur la figure 3.3, une transition entre deux phases peut être distinguée. Dans la première phase de dérive, les moyennes des gradients sont beaucoup plus grandes que leurs écarts types, indiquant une faible stochasticité du gradient (SNR élevé). Dans la seconde phase de diffusion, les gradients moyens sont très faibles par rapport à leurs fluctuations batch à batch, et le gradient se comporte comme du bruit Gaussien avec des moyennes très faibles, pour chaque couche (faible SNR). Une telle transition est attendue en général, lorsque l'erreur empirique sature et que la SGD est dominée par ses fluctuations. Ces phases distinctes lors de la SGD semblent alors correspondre et même expliquer les phases de minimisation de l'erreur empirique et de compression observées dans le plan d'information.

6 Conclusion

Dans ce chapitre, nous avons posé un cadre mathématique décrivant le problème d'apprentissage supervisé en se basant sur une loi de distribution hypothétique \mathcal{D} reliant les entrées \mathbf{x} aux sorties y . Ce cadre permet de décomposer l'erreur entre le classifieur optimal et l'erreur obtenue avec notre classifieur g_θ en trois types d'erreurs : approximation, optimisation et généralisation. Nous avons ensuite exposé certaines bornes de généralisation et mesures de complexité cherchant à refléter l'impact des paramètres définissant l'architecture g_θ , l'algorithme d'apprentissage ainsi que ses paramètres de configuration, sur le comportement de généralisation de g_θ et pour une distribution \mathcal{D} . Aussi, bien qu'aujourd'hui les causes d'une bonne généralisation ne sont pas totalement identifiées, ces différents travaux sur les mesures

de complexité ont permis de mettre en avant et de partiellement expliquer des comportements d'architectures profondes contradictoires avec la théorie de l'apprentissage classique. Il est alors nécessaire d'exploiter ces comportements pour concevoir des architectures plus efficaces.

Certains points d'intérêt ont cependant été omis durant ce chapitre. C'est notamment le cas du problème d'estimation des performances d'un modèle g_θ sur des données réelles qui suivrait une distribution hypothétique \mathcal{D}_r , différente des données d'entraînement. En effet, il peut y avoir un écart de distribution entre des données d'entraînement simulées et des données réelles de tests, qui est une forme de robustesse ignorée dans le contrôle du risque. Certaines mesures de complexité citées dans ce chapitre sont bien indépendantes de toute distribution des données, mais c'est au prix de bornes trop larges et souvent inutilisables dans la pratique. Nous n'explorons pas non plus le domaine très vaste de l'optimisation non convexe en apprentissage profond, qui est généralement focalisé sur des cas de classification d'images où d'autres problématiques sont présentes. De même, nous n'aborderons pas l'apprentissage non supervisé et le cas des réseaux de neurones bayésiens étant donné que ces méthodes ne sont pas utilisées ou au centre de nos expérimentations.

Chapitre 4

Modèles acoustiques

1 Introduction

Dans le domaine de la classification acoustique sous-marine, nous disposons généralement de données très inégalement réparties, souvent confidentielles et fréquemment labellisées de façon incomplète. Cela peut s'expliquer du fait qu'il soit difficile, même pour un expert, de correctement identifier toutes les sources présentes dans un enregistrement acoustique. Pour ces raisons, il a toujours été difficile de concevoir un algorithme de classification capable de généraliser dans la grande variété d'environnements envisageables en situations réelles.

Nonobstant, un simulateur de signaux acoustiques peut éluder ce problème au prix de quelques pertes de réalismes dues à l'ensemble des simplifications employées à cet effet. L'essentiel pour notre simulation étant de générer des signaux représentant l'ensemble des caractéristiques qui pourrait représenter un enjeu pour la classification. L'utilisation de simulateur pour valider les performances de produits est en soi une pratique plutôt courante dans ce domaine, et Thales a développé de nombreux simulateurs dans ce but. Dans ce chapitre nous commencerons par développer un modèle très restreint dans la section 2, nous permettant de travailler dans un environnement maîtrisé où la distribution des données $\mathcal{D}(\mathbf{x}, y)$ est connue. Un second modèle, plus élaboré mais moins bien maîtrisé, sera ensuite présenté dans la section 3 pour la classification multi-classes multi-labels du chapitre 8.

Remarque

Pour le modèle décrit dans la section 3, certains détails d'implémentation seront omis par mesure de confidentialité. Pour la même raison certaines composantes du signal ne sont pas du tout documentées, mais seront cependant utilisées pour la génération des données. Le modèle de modulation des bruits de cétacés est notamment issu d'une recherche interne financée entièrement par Thales.

2 Modèle de bruit de cavitation d'hélice

2.1 Schéma de cavitation

Soit $x(t)$ un signal reçu après traitement d'antenne, combinaison d'un bruit d'hélice $s_\eta(t)$ et d'un bruit ambiant gaussien et stationnaire $n_a(t)$, comme présenté par Lourens (1997) :

$$x(t) = s_\eta(t) + n_a(t). \quad (4.1)$$

Dans ce modèle, les bruits générés par les perturbations de surface, la vie marine et le trafic lointain sont représentés dans leur ensemble par $n_a(t)$ et sont indépendants du bruit d'hélice $s_\eta(t)$. Le bruit d'hélice $s_\eta(t)$ modélise le bruit de cavitation produit par l'hélice d'un navire de surface grâce à un ensemble de paramètres $\eta = \{p, \omega, \phi_i, c_i\}$. Contrairement à Lourens (1997), nous proposons un modèle dans Mahiout et al. (2020) où chacune des p pales de l'hélice produit son propre bruit de cavitation, tout comme dans Hanson et al. (2008). Par conséquent, le bruit d'hélice $s_\eta(t)$ est modélisé comme un mélange de différents bruits de cavitation $c_{i,\omega}(t)$, modulés en amplitude par leurs signaux périodiques associés $m_{i,\omega}(t)$, plus un bruit de cavitation constant $c_{0,\omega}(t)$:

$$s_\eta(t) = \sum_{i=1}^p m_{i,\omega}(t) \cdot c_{i,\omega}(t) + c_{0,\omega}(t). \quad (4.2)$$

Les signaux modulants périodiques $m_{i,\omega}(t)$, de périodes T_ω , sont dus à la rotation de l'hélice à la vitesse angulaire ω en $rad.s^{-1}$. Dans notre modèle, ce sont les schémas de cavitation de chaque pale, qui en régime stationnaire, déterminent les formes d'onde $m_{i,\omega}(t)$ de façon déterministe. En effet, la rotation de l'hélice a pour conséquence de créer des régions de pression plus faibles au contact de l'hélice qui, une fois que la pression diminue sous un seuil critique, provoquent des ruptures et par conséquent la formation de bulles. Nous savons également par expérience que, tant que l'hélice est en mouvement, l'enveloppe globale du bruit de cavitation induit par l'hélice ne peut pas prendre de valeur nulle au cours de son cycle. Par conséquent, même si des fortes profondeurs d'immersion et des faibles vitesses tendent à atténuer la formation de bulles, nous considérons dans le modèle que ce n'est pas suffisant pour supprimer totalement le bruit de l'hélice. Un bruit de cavitation gaussien $c_{0,\omega}(t)$ est ainsi introduit, afin de modéliser un bruit de cavitation constant dans le sillage de l'hélice.

Revenons sur la modélisation de $m_{i,\omega}(t)$. Le phénomène de cavitation étant lié à la différence de pression dans le sillage des pales comme le montrent Chen & Liu (2016) et Takinaci & Taralp (2021), nous choisissons d'exprimer la cavitation derrière la i -ème pale en fonction de la profondeur de cette dernière, mettant ainsi de côté les autres phénomènes produisant des ondes de pression comme le stipulent Bagheri et al. (2014). Soit $\varphi_i(t)$ l'angle de la i -ème pale à l'instant t , comme décrit dans la figure 4.2. L'extrémité de cette pale est à la profondeur $h + r \sin(\varphi_i(t))$, où h et r sont respectivement la profondeur du centre de l'hélice et la longueur de la i -ème pale. Par souci de simplicité, nous prendrons $r = 1$ mètre et nous ignorerons la profondeur du centre de l'hélice, bien que nous supposons que l'hélice est complètement immergée. Ainsi, lorsque les pales sont séparées par un angle $2\pi/p$, l'angle de la i -ème pale, $\varphi_i(t)$, peut être exprimé tel que suit :

$$\varphi_i(t) = 2\pi\omega t + (i - 1)\frac{2\pi}{p}, \quad (4.3)$$

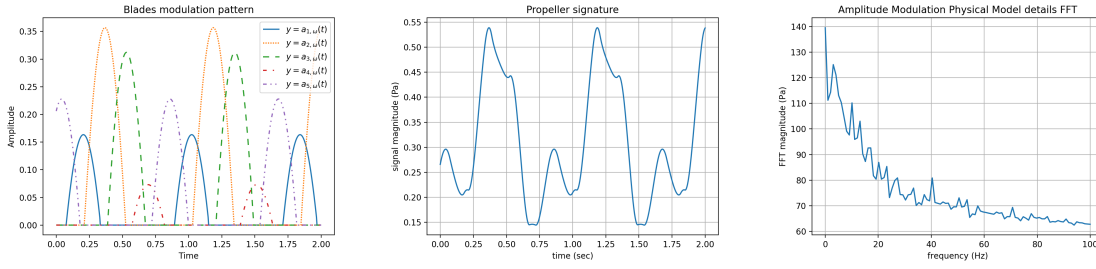


Figure 4.1: Signaux périodiques de modulation discret $m_{i,\omega}(k)$ (gauche) provenant d'une signature d'hélice η disposant de $p = 5$ pales avec son enveloppe globale de modulation $\sigma_\eta(k)^2$ correspondante (milieu) et les détails de 0 à 100 Hz de sa FFT (droite).

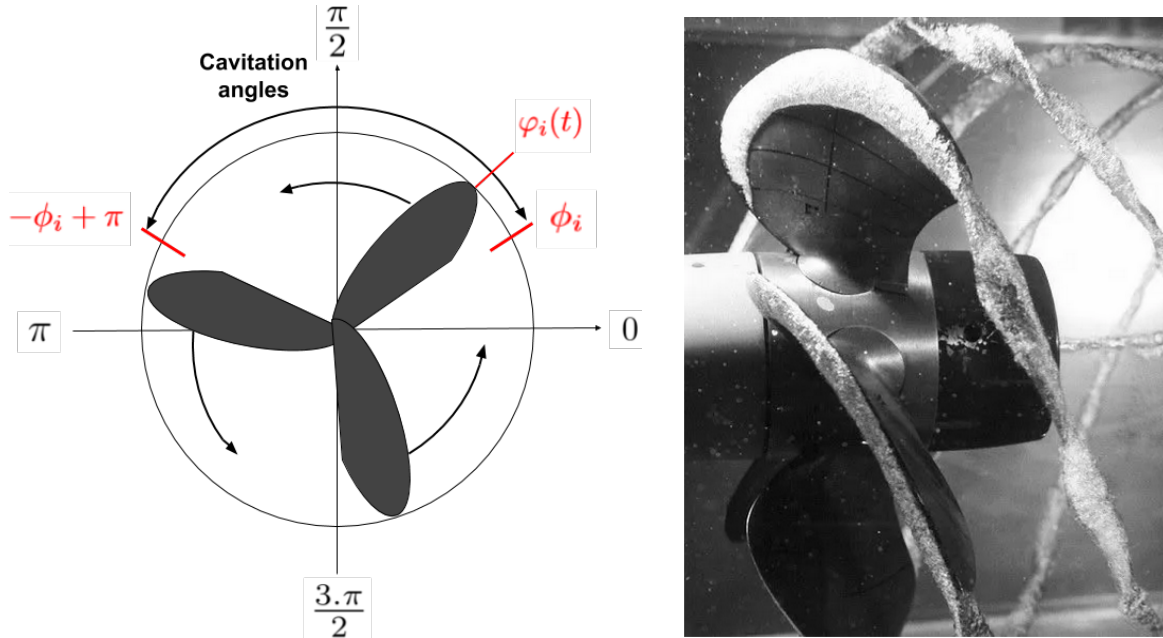


Figure 4.2: Zone de cavitation de la i -ème pale dans le modèle de bruit d'hélice à côté d'une image de cavitation d'hélice, tirée de theshipyardblog.com.

En faisant l'hypothèse simplificatrice que la pression exercée à l'extrémité d'une pale est constante et que des bulles ne sont produites qu'à leur extrémité, dans notre modèle les pales cavitent de l'angle ϕ_i à l'angle $-\phi_i + \pi$. La profondeur de la i -ème pale à ces deux angles est donc la profondeur maximale pour la formation de bulles de cavitation, la pression dans le sillage de la pale correspond alors à la pression de vapeur saturante et tant que la profondeur d'immersion de la pale est inférieure à cette profondeur, il y a formation de bulles de cavitation. Il s'ensuit que le signal modulant $m_{i,\omega}(t)$ est défini tel que

$$m_{i,\omega}(t) = \begin{cases} 0 & \text{pour } \varphi_i(t) \notin [\phi_i, -\phi_i + \pi] \text{ modulo } 2\pi, \\ c_i(\sin(\varphi_i(t)) - \sin(\phi_i)) \sin. & \end{cases} \quad (4.4)$$

où c_i est une constante représentant le facteur de modulation d'amplitude de la i -ème pale. Dans notre modèle, ces deux variables sont considérées comme déterministes mais inconnues pour notre classifieur profond.

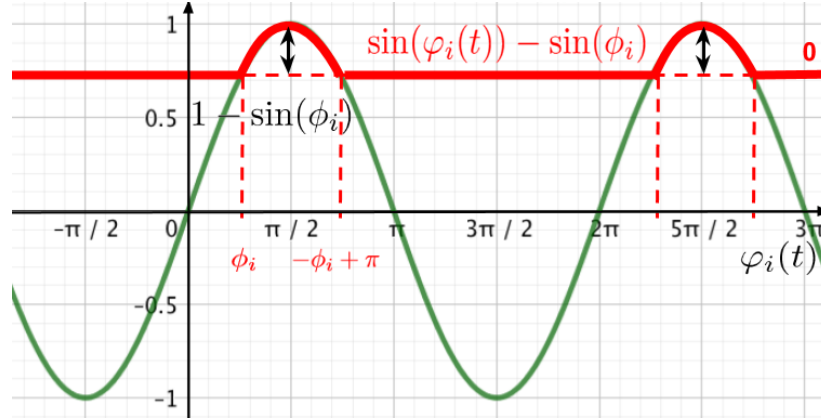


Figure 4.3: Évolution de $\sin(\varphi_i(t)) - \sin(\phi_i)$ en fonction de $\varphi_i(t)$ (en rouge).

2.2 Modèle de bruit de cavitation blanc

Dans cette sous-section nous supposons que les signaux de cavitation $c_{i,\omega}(t) = n_i(t)$ et $c_{0,\omega}(t) = n_0(t)$ sont des bruits blancs gaussiens indépendants. Ainsi, il résulte de (4.2) que $s_\eta(t)$ est également un bruit blanc. Cette hypothèse de bruit blanc pour le bruit de cavitation peut être débattue, mais elle est choisie ici par souci de simplicité puisque nous nous concentrons uniquement sur la périodicité du bruit dans les statistiques du second ordre. Qui plus est, la blancheur est également une hypothèse utilisée dans le traitement DEMON, au moins sur la sous-bande de fréquence d'intérêt. Nous faisons également l'hypothèse que le bruit ambiant $n_a(t)$ est un bruit blanc gaussien. Ainsi, comme le bruit d'hélice $s_\eta(t)$ et le bruit ambiant $n_a(t)$ sont indépendants, les variations de puissance perçues à travers le signal reçu $x(t)$ proviennent uniquement des différentes modulations d'amplitudes effectuées individuellement par chaque pale, comme on peut l'apercevoir sur la figure 4.4.

En supposant que $x(t)$ soit échantillonné à une fréquence f_s en K échantillons $x(k) = x(t_k)$ aux instants t_k tels que l'on obtient le vecteur $\mathbf{x} = [x(1), \dots, x(K)]$, des équations (4.1) et (4.2), il s'ensuit que chaque échantillon $x(k)$ suit la distribution gaussienne :

$$x(k) \sim \mathcal{N}(0, \sigma_\eta^2(k) + \sigma_a^2), \quad (4.5)$$

$$\sigma_\eta^2(k) = \sum_{i=1}^p m_{i,\omega}^2(k) \cdot \sigma_i^2 + \sigma_0^2, \quad (4.6)$$

où σ_a et σ_i sont respectivement les écarts types du bruit ambiant $n_a(k)$ et du bruit de cavitation $n_i(k)$. Par défaut nous avons $n_i(k) \sim \mathcal{N}(0, 1)$, $i = 0, 1, \dots, p$.

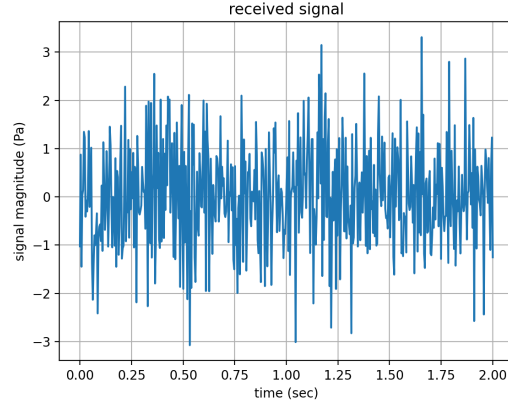


Figure 4.4: Signal discret $\mathbf{x}(k)$ reçu avec des bruits de cavitation et ambiants blancs ainsi que la signature $\sigma_\eta(t)^2$ affichée en Fig. 4.1.

2.3 Modèle de bruit de cavitation coloré

Dans cette sous-section nous supposons que les signaux de cavitation $c_{i,\omega}(t)$ et $c_{0,\omega}(t)$ sont des bruits gaussiens indépendants, mais cette fois-ci colorés, à la manière du modèle de [Bagheri et al. \(2014\)](#). En effet, le mécanisme principal derrière le bruit de cavitation est la résonance acoustique des bulles créées dans le sillage des pales de l'hélice. Une fois formées, ces bulles commencent à grossir puis finissent par éclater rapidement quand elles s'éloignent, ce qui provoque un bruit acoustique étalé sur une très large bande de fréquences. La distribution des fréquences de résonance de ces bulles détermine alors la forme spectrale du bruit de cavitation. Ainsi nous exprimons $c_{i,\omega}(t)$ comme la convolution d'un bruit blanc gaussien $n_i(t) \sim \mathcal{N}(0, 1)$ et d'un filtre de réponse impulsionnelle $h_\omega(t)$ lié à la vitesse de rotation de l'hélice :

$$c_{i,\omega}(t) = (n_i * h_\omega)(t). \quad (4.7)$$

Nous utiliserons une forme simple de réponse impulsionnelle, ne cherchant aucune forme de réalisme concernant la distribution des fréquences de résonance de ces bulles de cavitation :

$$h_\omega(t) = \begin{cases} 0 & \text{pour } t \notin [0, 2t_0] \\ \cos(2\pi f_h(\omega)(t - t_0)) \cdot e^{-\frac{(t-t_0)^2}{b_h}} & \text{sinon,} \end{cases} \quad (4.8)$$

où b_h est une constante caractérisant l'étalement spectral de la cavitation et $f_h(\omega) = 1000 \log(\omega + 1)$. De même, on fera l'hypothèse que le bruit ambiant $n_a(t)$ est composé de la seule composante du bruit de mer, en raison de sa prédominance dans la bande fréquentielle étudiée. Comme le bruit de cavitation $c_{i,\omega}(t)$, le bruit de mer $n_a(t)$ sera calculé en convoluant un bruit blanc gaussien $e_a(t) \sim \mathcal{N}(0, \sigma_a^2)$ avec un filtre de réponse impulsionnelle $h_m(t)$ calqué sur le modèle des courbes de Knudsen prenant uniquement en compte le niveau de mer m :

$$n_a(t) = (e_a * h_m)(t). \quad (4.9)$$

Remarque

Comme le mentionnent [Dahl et al. \(2007\)](#), il est généralement possible d'estimer approximativement la densité spectrale de la pression due aux bruits ambiants sous-marins avec les techniques actuelles de traitement d'antenne. Aussi, le bruit ambiant pouvant être assez bien mesuré, nous pouvons considérer que $|\hat{h}_m(f)|$ est connu en temps réel pour calibrer un détecteur.

2.4 Discussion

Définir la signature de l'hélice $\sigma_\eta(k)^2$ avec des paramètres d'hélice $\eta = \{p, \omega, \phi_i, c_i\}$ tirés de façon aléatoire nous permet de générer une grande variété de signatures réalistes pour notre problème de détection de bruit de navire. D'un autre côté, avec cette approche, comparer notre modèle avec une signature réelle, à des fins de validation, est plus difficile qu'avec un modèle basé sur des fréquences de modulation.

Un autre défaut de ce modèle est l'hypothèse de cyclostationnarité employée pour les bruits d'hélice, qui ignore le fait qu'en réalité le signal produit par l'hélice varie sur des durées d'ordres de grandeur plus grands que sa période et ce, notamment pour des hélices proches de la surface. Ainsi des périodicités dans les statistiques de troisième ordre peuvent être dues à des mouvements de vague ou d'autres défauts du système de l'hélice non modélisés.

3 Modèle de bruits acoustiques sous-marins

Dans cette section nous proposons d'exploiter différents modèles de bruits acoustiques, ainsi que diverses transformations aléatoires des données, afin d'obtenir un simulateur capable de générer une plus grande variété de données. En effet, les séries temporelles générées par ce simulateur serviront pour un problème de classification multi-classes disposant des classes suivantes :

- **Classe 0** : Bruit ambiant uniquement
- **Classe 1** : Navire de transport de marchandises de type 1 (Chalutier)
- **Classe 2** : Navire de transport de marchandises de type 2 (Porte-conteneurs)
- **Classe 3** : Bateau rapide (Zodiac)
- **Classe 4** : Bâtiment d'intérêt
- **Classe 5** : Bruit de cétacés
- **Classe 6** : Bruit de crevettes claqueuses (Banc en migration)

Toutes ces classes représentent des sources de bruits souvent détectés par les antennes passives ou pouvant présenter un intérêt pour la classification. Ce simulateur n'a en revanche pas pour but de simuler un traitement d'antenne. On considère en effet que les signaux 1S simulés sont obtenus après une formation de voie dans la direction de la source principale. Nous ajoutons alors d'autres composantes avec de simples additions, sans tenir compte des déformations induites par un traitement d'antenne spécifique, si ce n'est des pertes d'intensité.

3.1 Chaîne de simulation du signal

Une fois que la classe du signal devant être générée a été sélectionnée (ou plusieurs pour la classification multi-classes multi-labels), on utilise la chaîne de simulation présentée ci-dessous dans la figure 4.5, pour partir du signal du bruiteur jusqu'au signal reçu sur un unique capteur acoustique.

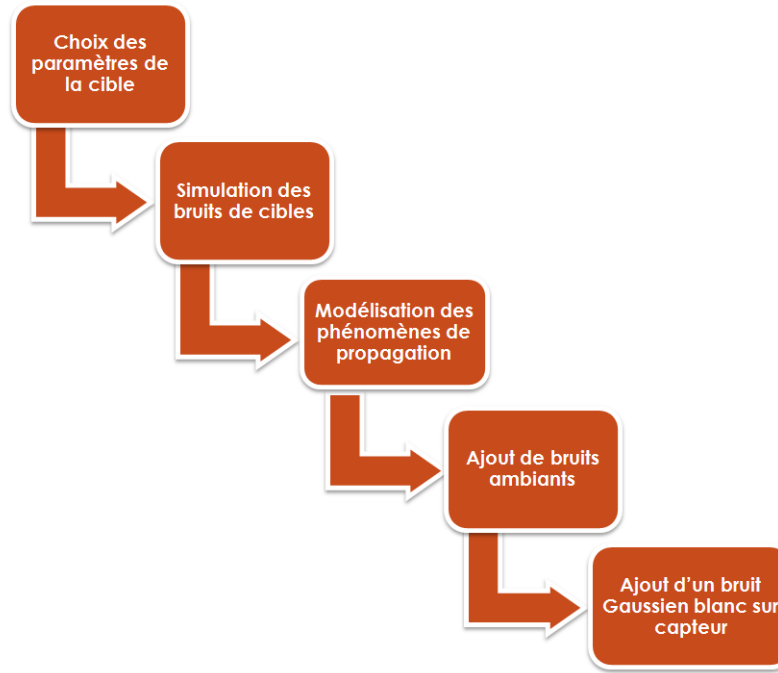


Figure 4.5: Schéma de la chaîne de simulation des signaux pour la classification multi-classes.

Comme indiqué sur la figure 4.5, pour un élément d'une classe donnée, la première étape consiste à définir de façon aléatoire la position, la direction ainsi que la vitesse de la source pendant la durée T de simulation. Ces paramètres sont pris en compte lors de la modélisation des phénomènes de propagation et pour la simulation de bruit de navire en affectant certains paramètres tels que le régime moteur ou la vitesse de rotation des pales. Les autres paramètres impactant le bruit rayonné sont tirés de façon aléatoire entre deux valeurs limites, propres à chaque classe.

La seconde étape consiste à simuler avec les paramètres choisis le bruit de la source à une distance de référence de 1 m, ce qui se fait à partir des modèles présentés en sections 4 et 5. Une fois le bruit source généré, on y applique différents effets de propagation déterminés par le choix des paramètres de la source et la position des capteurs de réception. Du fait de la complexité de la propagation sous-marine, les signatures reçues à moyenne distance peuvent alors être appréciées très différemment en comparaison à une distance de référence de 1 m.

On ajoute ensuite au signal $s(t)$ les différents bruits ambiants $n_a(t)$, caractérisés dans la

section 6, pour finalement obtenir le bruit reçu $x(t)$. Pour chaque exemple d'apprentissage, on emploie un bruit de surface généré à partir de niveaux de vent et de pluies aléatoires. De même, le niveau du bruit de trafic est choisi de manière aléatoire pour chaque exemple d'entraînement.

Enfin un bruit blanc Gaussien peut être ajouté pour modéliser le bruit sur capteur, mais cette étape sera effectuée durant la phase d'augmentation des données présentée dans la sous-section suivante, plutôt que durant la génération des signaux enregistrés.

3.2 Data Augmentation

La Data Augmentation, notamment introduite dans le domaine de la classification d'images par [Krizhevsky et al. \(2012\)](#), est une pratique couramment utilisée du fait que les réseaux profonds ont besoin d'une importante quantité de données pour être capable d'extraire des caractéristiques permettant une forme de généralisation. Cette technique consiste à appliquer plusieurs transformations de façon aléatoire à nos données d'origines afin d'augmenter la quantité de données labellisées de façon artificielle et possiblement d'entraîner notre classifieur à devenir invariant à ces transformations. Son utilisation permet par exemple qu'un réseau destiné à la classification de caractères, puisse être capable de reconnaître une lettre inclinée à 30° si elle est présente dans le jeu d'entraînement, mais pas orientée de la sorte. Ainsi, dans le domaine des images, on retrouve souvent des transformations affines telles que des rotations, translations, zooms, modifications d'intensité ou encore l'ajout de bruit blanc pour rendre un classifieur robuste à ces types de variations.

Pour les signaux acoustiques, la Data Augmentation doit ainsi permettre de faire gagner en diversité notre jeu de données tout en gardant des signaux réalistes. Dans notre cas, les signaux générés par simulation présentent déjà de nombreuses formes de déformations, qui ne pouvaient être appliquées à un signal reçu $x(k)$ du fait que les transformations n'affectent que $s(k)$ et non le bruit ambiant. On peut cependant encore ajouter une forme de bruit blanc Gaussien sur le signal reçu $x(k)$ et effectuer un re-cadrage temporel.

L'ajout de bruit blanc Gaussien au signal reçu doit modéliser les imperfections de la chaîne d'acquisition acoustique, depuis le/les récepteurs jusqu'au signal numérique devant être classifié, $x(k)$. On ajoute ainsi au signal discret simulé $x(k)$, une composante de bruit blanc Gaussien $\alpha \cdot n_c(k)$ où $n_c(k) \sim \mathcal{N}(0, 1)$ et $\alpha \sim \mathcal{U}(\alpha_{min}, \alpha_{max})$, pour obtenir un nouveau signal

$$x'(k) = x(k) + \alpha \cdot n_c(k). \quad (4.10)$$

Remarque

Sans protocole bien défini, nous avons pu constater expérimentalement que l'ajout d'un faible niveau de bruit blanc gaussien sur des données déjà générées, durant la Data Augmentation, peut permettre de légèrement améliorer les performances de généralisations d'algorithmes entraînés sur de petits jeux de données.

Le re-cadrage temporel consiste quant à lui à modifier l'intervalle temporel de définition du signal $x(k) = x(t_k)$. Aussi, pour disposer de signaux disposant de $K = \lfloor fs \cdot T \rfloor$ échantillons temporels après le re-cadrage, le simulateur va générer des signaux pour $t_k \in [0, T']$ avec $T' > T$ et on ne garde le signal que pour $t_k \in [t_0, t_0 + T]$:

$$x(t_k), t_k \in [0, T'] \longrightarrow x(t_k), t_k \in [t_0, t_0 + T],$$

avec $t_0 \sim \mathcal{U}(0, T' - T)$ et correspond à l'instant pour lequel on commence à conserver le signal temporel.

Remarque

Le but du re-cadrage temporel peut également être de rendre notre architecture invariante aux translations temporelles si nous disposons seulement d'une propriété d'équivariance temporelle dans notre architecture profonde. Ainsi, à partir d'un unique signal $x(k)$, on va pouvoir obtenir plusieurs exemples d'entraînement permettant potentiellement d'entraîner notre architecture à être plus robustes aux translations temporelles.

Définition : Équivariance

Une propriété d'équivariance aux translations se caractérise par des réponses à la présence d'un motif toujours comparable, sans tenir compte de la position où ledit motif aura été détecté. Contrairement à une propriété d'invariance, un décalage de la position du motif dans la donnée d'entrée, entraîne cependant un décalage dans sa réponse. Aussi si nous disposons d'un détecteur parfait renvoyant la réponse 1, uniquement quand il passe sur un motif 1,9,6, alors une réponse équivariante de ce détecteur sera de la forme :

$$[0, 0, 1, 9, 6, 0, 0] \implies [0, 0, 1, 0, 0] \quad (4.11)$$

$$[0, 0, 0, 1, 9, 6, 0] \implies [0, 0, 0, 1, 0] \quad (4.12)$$

Une propriété d'invariance aux translations se caractérise par des réponses toujours identiques à la présence d'un motif :

$$[0, 0, 1, 9, 6, 0, 0] \implies [1] \quad (4.13)$$

$$[0, 0, 0, 1, 9, 6, 0] \implies [1] \quad (4.14)$$

4 Modélisation des bruits de navires

4.1 Caractéristiques des bruits de navire

Avec des systèmes de propulsion pouvant délivrer jusqu'à 100 000 chevaux, les navires marchands comme militaires produisent un niveau de bruit significatif dans l'environnement sous-marin. En effet, si on considère les pertes de conversions de l'énergie délivrée par l'arbre

moteur pour la propulsion, une grande partie est rayonnée sous forme d'énergie acoustique dans l'eau environnante.

Les navires à propulsion mécanique produisent ainsi un bruit lié à leur système de propulsion qu'il soit diesel, ou bien un mixte diesel électrique ou diesel gaz. Chaque système comportant des arbres moteurs, des engrenages ainsi que des turbines disposés différemment, le déséquilibre dynamique d'un composant résulte en des oscillations uniques qui vont se transmettre à la coque et émettre un rayonnement acoustique qui se propagera dans l'eau. Le bruit émis par ce type de rayonnement est alors souvent caractérisé par des raies et peignes d'harmoniques liés aux fréquences de rotation du système. L'ordre de grandeur des forces mises en jeu ici, augmente environ de façon proportionnelle à la racine carrée de la vitesse de rotation, comme le stipule [Urick \(1975\)](#). Des forces de friction peuvent également être présentes dans le système de propulsion et les frictions linéaires résultent généralement en une bande de fréquences plus large au niveau de la composante du système de propulsion.

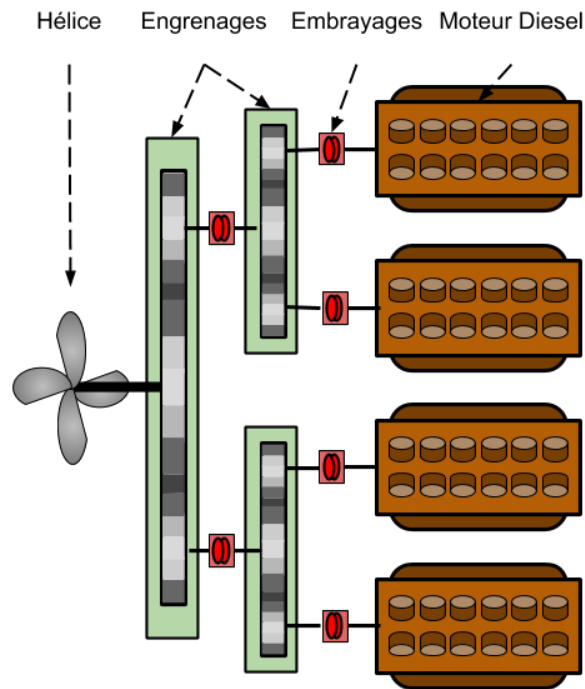


Figure 4.6: Schéma d'un système de propulsion mécanique par combiné de 4 lignes diesels.

Une autre source de bruit est le bruit de cavitation produit par la rotation des lames d'hélices du navire, comme présenté dans la section 2. Ce bruit étant lié à l'effondrement de bulles de cavitation créées par les différences de pression dans le sillage des pales, il diminue avec des niveaux d'immersion croissants et augmente avec la vitesse de rotation de l'hélice. La fréquence du pic du spectre de cavitation, visible sur la figure 4.7, est également affectée par la taille, la profondeur ainsi que de la vitesse de rotation de l'hélice. Le bruit de cavitation, caractérisé par sa modulation au pas de la fréquence de l'arbre moteur, est alors très souvent

la composante principale de la signature d'un bâtiment de surface ou d'un sous-marin proche de la surface et fournit des indices utilisables par un utilisateur pour effectuer une classification du navire à propulsion mécanique. En conséquence, le fait pour un sous-marin d'opérer à une profondeur suffisante lui permet de grandement réduire sa cavitation et donc de très largement réduire sa signature. Enfin, en plus du bruit de cavitation, un champ de propulsion caractérisé par des tonalités de bruits supplémentaires, peut apparaître au niveau de l'hélice à cause des vibrations des lames rotatives dues au flux d'eau.

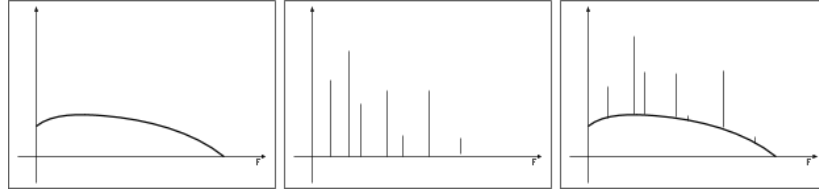


Figure 4.7: *Forme générale d'un spectre associé à un bâtiment de surface à droite, avec la forme d'un spectre de cavitation à gauche, et la forme d'un spectre de bruit de machines au centre.*

Les machines auxiliaires telles que les générateurs électriques, pompes, ou appareils électromagnétiques, sont également des sources de bruit dues aux vibrations qu'ils produisent. Bien qu'ils ne permettent pas de classer un navire, tout comme une transitoire pouvant être produite par l'ouverture d'un sas, ils trahissent la présence d'un navire dans une direction et peuvent permettre de confirmer une classification en cas de doute. Qui plus est, ces machines ne participant pas à la propulsion et elles ont des régimes moteurs qui ne varient que rarement, ce qui fait que leurs composantes tonales sont relativement stables en amplitude et en fréquence.

Enfin une dernière source de bruit, bien que plus faible, est due aux effets hydrodynamiques. L'écoulement de l'eau à la surface de la coque peut notamment devenir turbulent au-dessus d'une vitesse critique, ce qui peut induire des vibrations dans la structure de la coque, et ainsi un rayonnement acoustique comme le précisent [Burdic & Bartram \(1984\)](#). Ces bruits étant cependant négligeables sur de longues distances, tout comme les bruits de machines auxiliaires, ils ne seront pas pris en compte dans notre modèle acoustique.

4.2 Modèle de bruits de navire

Soit $s(t)$ un bruit produit par un navire, combinaison d'une composante machine $s_{ma}(t)$ et d'une composante de bruit d'hélice $s_{ca}(t)$ avec :

$$s(t) = s_{ma}(t) + s_{ca}(t).$$

On définit ici la composante machine $s_{ma}(t)$ comme étant l'ensemble des bruits mécaniques produits par les oscillations d'un système de propulsion diesel durant la durée T du signal. Dans notre modèle ces oscillations sont caractérisées par la formation de raies aux fréquences de rotation du système. En tout, la composante machine comporte $n_l \times n_h$ raies, n_l étant le

nombre de lignes d'arbre moteur et n_h le nombre d'harmoniques de chaque ligne diesel :

$$s_{ma}(t) = \sum_{j=1}^{n_h} \sum_{i=1}^{n_l} c_l^{(i,j)} \cos(2\pi f_l^{(i,j)}(t + g_{ma}(t))). \quad (4.15)$$

Une première raie, à la fréquence f_{ma} , correspond à la première ligne diesel, tandis que les autres correspondent à des répétitions des autres lignes diesel aux fréquences

$$f_l^{(i,j)} = j(f_{ma} + \delta l \cdot (i - 1)). \quad (4.16)$$

Dans notre modèle, $c_l^{(i,j)}$ est un coefficient multiplicateur pour la $j^{\text{ème}}$ harmonique de la $i^{\text{ème}}$ ligne diesel :

$$c_l^{(i,j)} = \frac{1}{j^2} \left(\frac{3}{4} + \frac{1}{2} \cdot (-1)^{i-1} \right). \quad (4.17)$$

Le coefficient $c_l^{(i,j)}$ dépend ainsi de la manière dont sont disposées les lignes diesels. La fonction $g_{ma}(t)$ correspond quant à elle à une déformation temporelle aléatoire pouvant modéliser des défauts de transmission dans les engrenages et est définie comme suit :

$$g_{ma}(t) = \alpha \cdot n_{ma}(t) \sum_{i=1}^{n_\beta} \cos(2\pi \beta_i f_m t), \quad (4.18)$$

où les coefficients α et β_i sont des constantes. Le signal $n_{ma}(t)$ est obtenu en filtrant un bruit blanc gaussien $e_{ma}(t) \sim \mathcal{N}(0, 1)$ avec un filtre $h_{ma}(t)$ servant à définir l'étalement spectral des raies. Sur la figure 4.8, où aucune harmonique n'est présente, on peut visualiser n_l raies ayant chacune une fréquence centrée autour de $f_l^{(i,j)}$. L'écart entre les fréquences de ligne diesel δl étant cependant petit, on peut difficilement différencier toutes les raies à l'oeil nu.

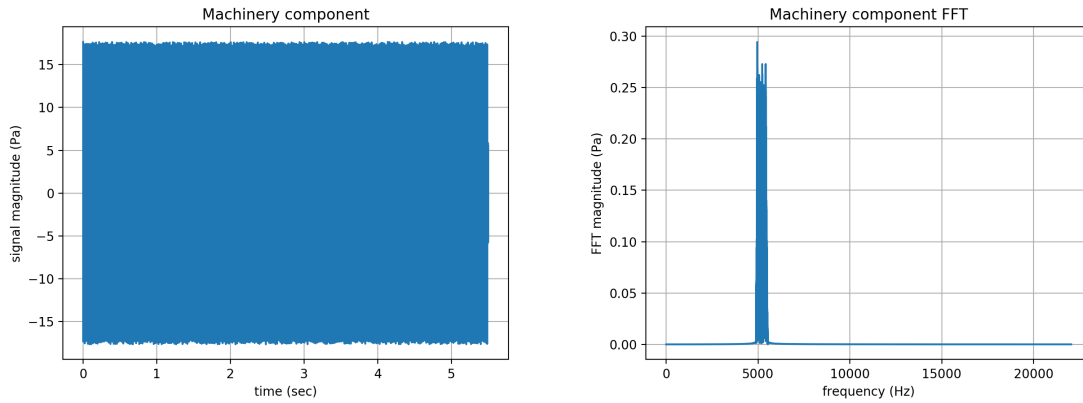


Figure 4.8: Représentation temporelle, à gauche et fréquentielle, à droite, de la composante machine enregistrée pendant 5 secondes à une fréquence d'échantillonnage $f_s = 44100$ Hz.

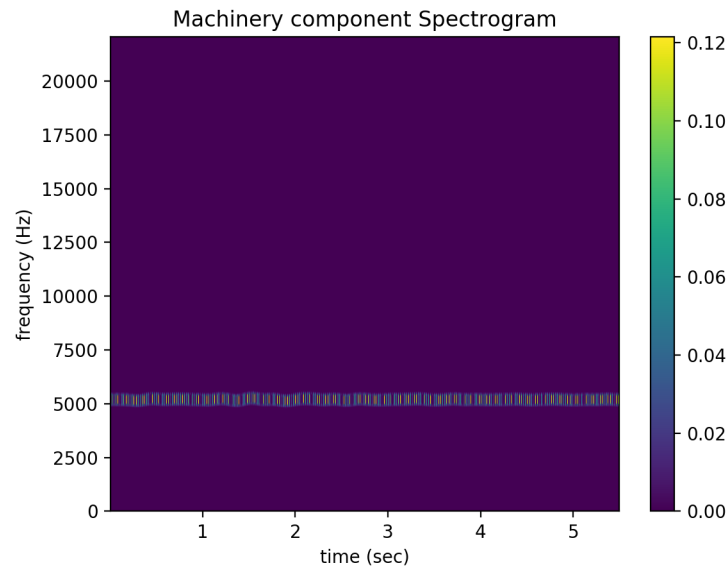


Figure 4.9: Spectrogramme de la composante machine associé à la figure 4.8. Pour tout les spectrogrammes, utilisation d'une fenêtre de hanning de 2048 échantillons avec un recouvrement de 50 %.

Dans un second temps, on définit la composante de cavitation $s_{ca}(t)$ comme étant l'ensemble des bruits de cavitation produits par les hélices du navire durant la durée T du signal. Nous reprendrons ici le modèle de bruit de cavitation d'hélice présenté en section 2, avec une hypothèse de bruit de cavitation coloré, comme décrit dans la sous-section 2.3 et en supposant qu'il n'y a qu'une seule hélice.

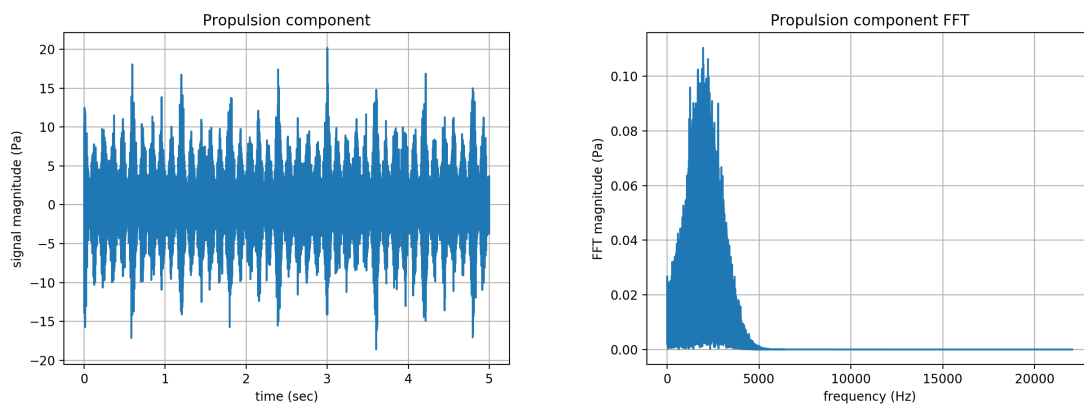


Figure 4.10: Représentation temporelle, à gauche et fréquentielle, à droite, de la composante de cavitation enregistrée pendant 5 secondes à une fréquence d'échantillonnage $f_s = 44100$ Hz.

Sur la figure 4.10, le spectre fréquentiel obtenu à partir de l'équation 4.8 est centré autour de la fréquence moyenne de cavitation hydraulique $f_h(\omega)$, qui vaut ici environ 2000 Hz et s'étale sur une bande de fréquences déterminée par la constante d'étalement spectral b_h . Sur la figure 4.11, on peut ainsi voir que le bruit de cavitation reste localisé à basse fréquence et n'est pas constant dans le temps dû au rythme de l'hélice.

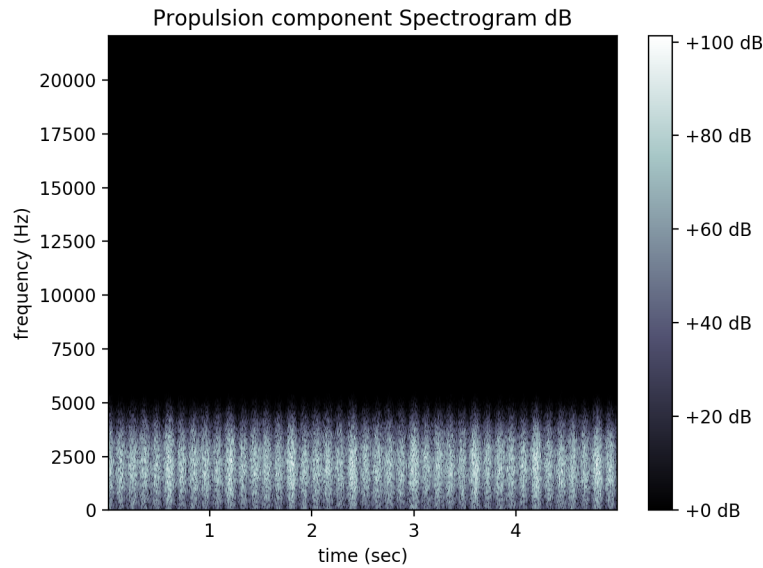


Figure 4.11: Spectrogramme de la composante de cavitation associé à la figure 4.10 en échelle logarithmique.

Sur la figure 4.12 est affichée la forme spectrale d'un bruit de navire issu de notre modèle, que l'on peut comparer à la figure 4.7, où est schématisée la forme spectrale générale d'un bâtiment de surface. Contrairement au cas de la figure 4.8, l'utilisation d'un nombre $n_h > 1$, pour la composante machine, donne des harmoniques pour chaque fréquence de ligne diesel, modélisant ainsi un plus grand nombre de fréquences d'oscillation pour un navire bruyant.

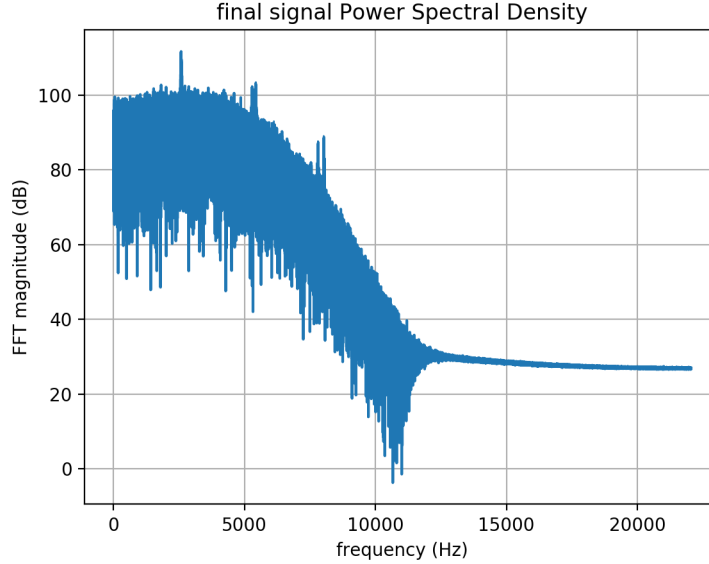


Figure 4.12: Représentation fréquentielle enregistrée pendant 5 secondes à une fréquence d'échantillonnage $f_s = 44100$ Hz du bruit produit par un navire dans notre modèle.

5 Modélisation des bruits biologiques

5.1 Modèle de bruits de cétacés

Soit $s(t)$ un bruit produit par un mammifère marin, combinaison d'une composante de modulation $s_{md}(t)$ et d'une composante de clics $s_{cl}(t)$ avec :

$$s(t) = s_{md}(t) + s_{cl}(t).$$

On définit la composante de modulation $s_{md}(t)$ comme étant l'ensemble des communications produites par les séries de cliquetis d'un ou plusieurs cétacés durant la durée T du signal. En général, la modulation est un processus par lequel le signal est transformé de sa forme originale en une forme adaptée au canal de transmission, en faisant varier l'amplitude, la phase ou bien la fréquence instantanée d'une onde sinusoïdale porteuse. Les cétacés utilisent ainsi les basses fréquences pour transporter sur de longues distances des communications, telles que des appels à l'aide, grâce à une modulation en fréquence. De même les hautes fréquences peuvent être utilisées pour des communications à faible portée comme au sein d'un groupe d'orques. Cette composante, fruit de recherches au sein de Thales, est définie de la manière suivante :

$$s_{md}(t) = \sum_{i=1}^{n_{md}} md_i(t), \quad (4.19)$$

où $md_i(t)$ correspond au $i^{\text{ème}}$ paquet/vocalise produit durant la durée T du signal :

$$md_i(t) = \begin{cases} 0 & \text{pour } t \notin [\tau_{md}^{(i)}, \tau_{md}^{(i)} + dt_{md}] \\ \cos\left(2\pi\phi_{md}^{(i)}(t - \tau_{md}^{(i)})\right) & \text{sinon.} \end{cases} \quad (4.20)$$

Dans ce modèle, $n_{md} = \lfloor 0.5 \cdot T/dt_{md} \rfloor$ est le nombre de paquets présents pendant la durée totale du signal. En outre, la durée entre chaque paquet est approximativement dt_{md} , où dt_{md} la durée de tous les paquets dans ce modèle. L'instant $\tau_{md}^{(i)}$, où le $i^{\text{ème}}$ paquet $md_i(t)$ débute, est déterminée de façon aléatoire grâce à l'équation

$$\tau_{md}^{(i)} = 2(i-1) \cdot dt_{md} + 0.6 \cdot dt_{md} \cdot \alpha_{md}^{(i)}, \quad (4.21)$$

avec $\alpha_{md}^{(i)} \sim \mathcal{N}(0, 1)$. Sur la figure 4.13, $\tau_{md}^{(2)}$ correspond ainsi au début du 2^{ème} paquet.

Dans l'équation 4.20, $\phi_{md}^{(i)}(t - \tau_{md}^{(i)})$, est la phase instantanée du $i^{\text{ème}}$ paquet. Plutôt que de définir directement $\phi_{md}^{(i)}$, on déterminera plutôt l'expression de sa fréquence instantanée associée $\omega_{md}^{(i)}$, la fréquence instantanée étant la dérivée positive de la phase. En pratique, en posant la fonction discrétisée $\phi_{md}^{(i)}(t_k - \tau_{md}^{(i)})$ sous forme de suite, telle que

$$\phi_{md}^{(i)}(k) = \phi_{md}^{(i)}(t_k - \tau_{md}^{(i)}), \quad (4.22)$$

pour tout $k = k_0, \dots, k_0 + K$, avec $k_0 = \lfloor \tau_{md}^{(i)} \cdot fs \rfloor$ et $K = \lfloor fs \cdot dt_{md} \rfloor$, on peut la définir de façon récursive comme suit :

$$\phi_{md}^{(i)}(k) = \phi_{md}^{(i)}(k-1) + \omega_{md}^{(i)}(k), \quad (4.23)$$

avec une initialisation $\phi_{md}^{(i)}(k_0) = 0$. La fréquence instantanée discrète du $i^{\text{ème}}$ paquet

$$\omega_{md}^{(i)}(k) = \omega_{md}^{(i)}(t_k - \tau_{md}^{(i)}), \quad (4.24)$$

est un polynôme de degré 3 issu de l'interpolation cubique de l'équation

$$y_j^{(i)} = \omega_{md}^{(i)}(x_j^{(i)}), \quad (4.25)$$

pour $j = 1, \dots, 4$ et avec

$$x_j^{(i)} = j \cdot \frac{dt_{md}}{4} - \tau_{md}^{(i)}, \quad (4.26)$$

ainsi que

$$y_j^{(i)} = f_{min} + \xi_j^{(i)} \cdot \left(\frac{fs}{2} - f_{min} \right), \quad (4.27)$$

où $\xi_j^{(i)} \sim \mathcal{U}(0, 1)$. La fréquence instantanée $\omega_{md}^{(i)}$ peut ainsi varier entre les fréquences f_{min} et $fs/2$. On va ensuite estimer $\omega_{md}^{(i)}$ aux K valeurs contenues durant l'intervalle de modulation. Chaque paquet $md_i(t)$ a ainsi une fréquence instantanée qui varie dans le temps mais reste bornée par les extremums du vecteur $y_j^{(i)}$. En effet, on peut voir sur la figure 4.13 que chaque paquet correspond à une bande de raie spécifique, bien que ce ne soit pas le cas pour toutes les observations obtenables avec ce modèle. Sur la figure 4.14 on peut voir les formes cubiques des courbes des fréquences instantanées $\omega_{md}^{(i)}(t - \tau_{md}^{(i)})$, ressemblant fortement à des appels pulsés.

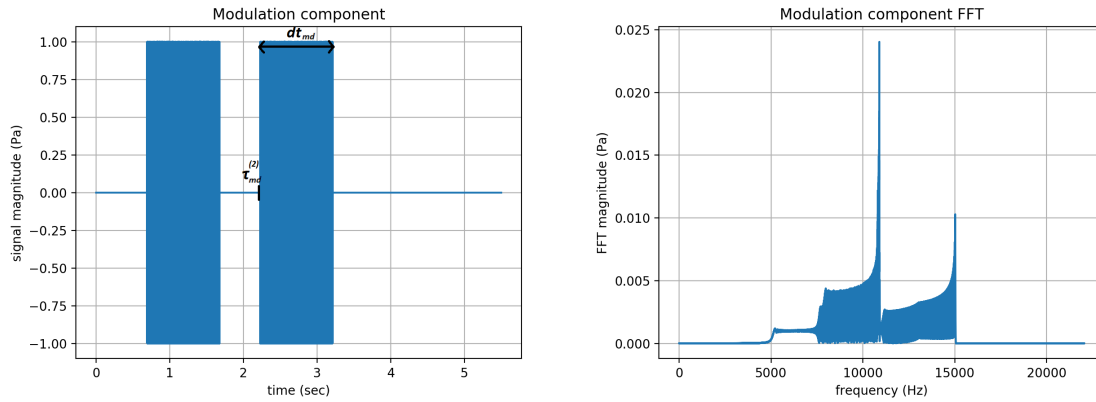


Figure 4.13: Représentation temporelle, à gauche et fréquentielle, à droite, de la composante de modulation enregistrée pendant 5 secondes à une fréquence d'échantillonnage $f_s = 44100$ Hz.

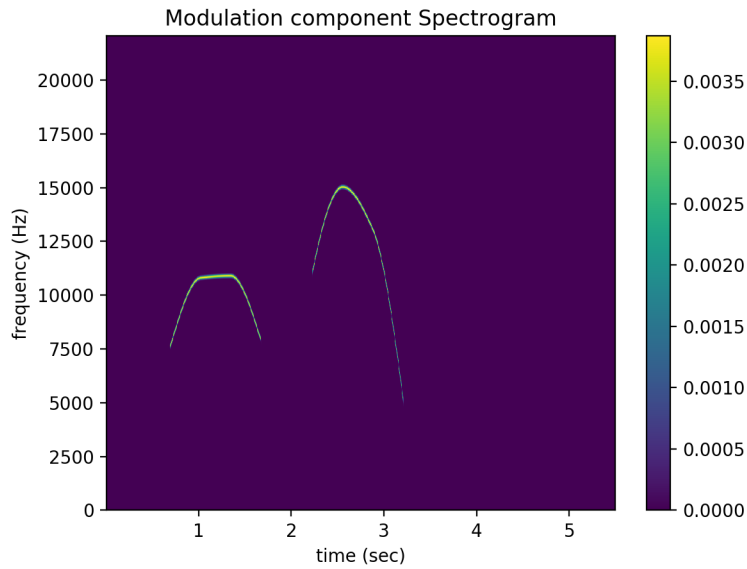


Figure 4.14: Spectrogramme de la composante de modulation associé à la figure 4.13

Dans un second temps, on définit la composante de clics $s_{cl}(t)$ comme étant l'ensemble des clics produits durant la durée T du signal, un clic étant utilisé par certains cétacés pour l'écholocation. Dans notre modèle, cette composante sera définie de la manière suivante :

$$s_{cl}(t) = \sum_{i=1}^{n_{cl}} cl_i(t), \quad (4.28)$$

où $cl_i(t)$ correspond au $i^{\text{ème}}$ clic produit durant la durée T du signal :

$$cl_i(t) = \begin{cases} 0 & \text{pour } t \notin [\tau_{cl}^{(i)}, \tau_{cl}^{(i)} + dt_{cl}] \\ A_{cl}^{(i)} \cdot g_{cl}(t - \tau_{cl}^{(i)}) & \text{sinon} \end{cases} \quad (4.29)$$

Dans ce modèle, $n_{cl} = \lfloor T/\delta_{cl} \rfloor$ est le nombre de clics présents pendant la durée totale du signal et également le nombre de fois que se répète l'intervalle moyen entre les clics δ_{cl} durant la période T . La durée d'un clic biologique dt_{cl} reste fixe dans le modèle. L'instant $\tau_{cl}^{(i)}$, où le $i^{\text{ème}}$ clic $cl_i(t)$ débute, est déterminée de façon aléatoire grâce à l'équation

$$\tau_{cl}^{(i)} = i\delta_{cl} + 0.3 \cdot \delta_{cl} \cdot \alpha_{cl}^{(i)}, \quad (4.30)$$

avec $\alpha_{cl}^{(i)} \sim \mathcal{N}(0, 1)$.

Dans l'équation (4.29), $A_{cl}^{(i)} = \mu_{cl} + \sigma_{cl} \cdot \xi_{cl}^{(i)}$ est l'amplitude du $i^{\text{ème}}$ clic, μ_{cl} et σ_{cl} étant respectivement sa moyenne et son écart type, tandis que $\xi_{cl}^{(i)}$ est une variable aléatoire suivant une loi normale $\mathcal{N}(0, 1)$. Enfin $g_{cl}(t)$ est une sinusoïde de fréquence f_{cl} et d'enveloppe à décroissance exponentielle de durée dt_{cl} tel que

$$g_{cl}(t) = \cos(2\pi f_{cl}t) \cdot e^{-\frac{t}{dt_{cl}}}. \quad (4.31)$$

Sur la figure 4.15 on peut voir un pic à cette valeur du spectre fréquentiel, mais il y a présence d'un étalement spectral du fait que la durée des signaux $cl^{(i)}(t)$ soit trop courte pour être bien représentée après FFT. Sur la figure 4.16, chaque clic est affiché par un trait vertical bien

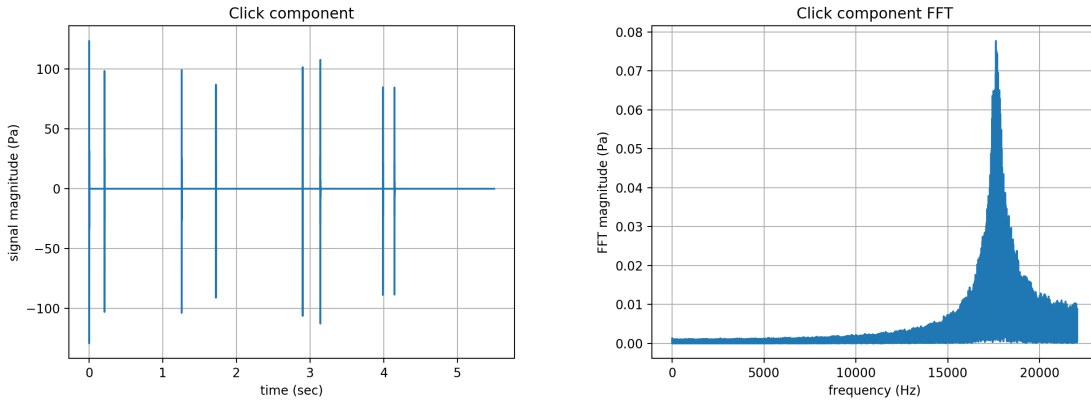


Figure 4.15: Représentation temporelle, à gauche et fréquentielle, à droite, de la composante de clic enregistrée pendant 5 secondes à une fréquence d'échantillonnage $fs = 44100$ Hz.

localisé en temps mais assez étalé en fréquence à l'image de ce qu'on a pu voir sur l'analyse spectrale en figure 4.15.

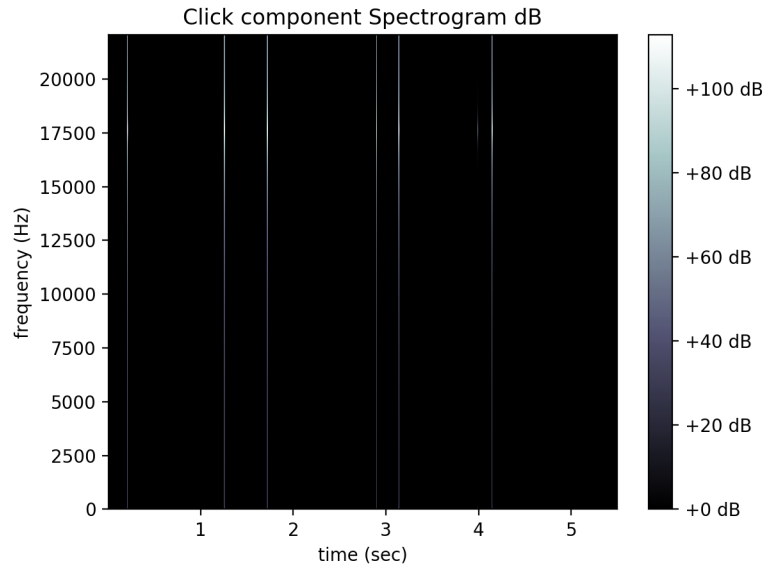


Figure 4.16: Spectrogramme de la composante de modulation associée à la figure 4.15 en échelle logarithmique.

5.2 Modèle de bruits de crevettes claqueuses

Soit $s(t)$ un bruit produit par un banc de crevettes claqueuses, composé uniquement d'une composante de claquements $s_{sn}(t)$:

$$s(t) = s_{sn}(t). \quad (4.32)$$

On définit la composante de claquements $s_{sn}(t)$, comme étant l'ensemble des bruits émis par des claquements de pinces durant la durée T du signal. En effet, les crevettes claqueuses, ou crevettes-pistolet, disposent d'une pince surdimensionnée qui, quand elle est claquée brusquement, produit après quelques instants une forte détonation pouvant atteindre les 200 décibels. Leur présence est donc l'une des principales sources de bruit parasite dans la tâche de classification sous-marine. Nous reprenons ici le modèle acoustique de [Au & Banks \(1998\)](#), similaire à celui de [Kim et al. \(2009\)](#), qui utilise une représentation idéalisée des claquements. Cette dernière est basée sur la synthèse de nombreuses observations de claquements de pinces pouvant être orientées dans différents angles par rapport à la direction du récepteur. Notre modèle ne prendra cependant pas en compte les premières réflexions produites par les claquements. En effet, on considérera que le banc ciblé est suffisamment éloigné de la surface ainsi que du plancher océanique pour pouvoir ignorer les distorsions liées à une réflexion juste après émissions.

Le modèle utilisé consiste en une faible impulsion précurseuse liée à la fermeture de la pince et immédiatement suivie par un intense claquement, créé par l'effondrement de la bulle de cavitation produite par cette action. Ces deux événements sont séparés par une durée liée à la vitesse du jet de vapeur d'eau et à la stabilité de la bulle. Ces deux facteurs sont principalement liés aux conditions de pression environnantes ainsi qu'à l'espèce de crevette. Chaque individu provoque par ailleurs des bruits quasiment uniques, mais le modèle ne prend pas en

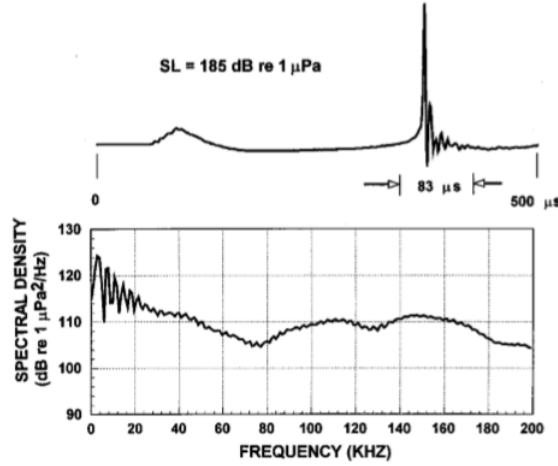


Figure 4.17: Exemple d'une forme d'onde typique du claquement d'une crevette claqueuse, en haut, et sa réponse spectrale en bas. Tiré de l'article de [Au & Banks \(1998\)](#).

compte cet aspect. Dans notre modèle, on a ainsi :

$$s_{sn}(t) = \sum_{i=1}^{n_{sn}} sn_i(t), \quad (4.33)$$

où $sn_i(t)$ correspond au $i^{\text{ème}}$ claquement de pince produit durant la durée T du signal :

$$sn_i(t) = \begin{cases} 0.01 \cdot A_{sn}^{(i)} \cdot h_{sn}^{(i)}(t - \tau_{sn}^{(i)}) & \text{pour } t \in [\tau_{sn}^{(i)}, \tau_{sn}^{(i)} + dt_1^{(i)}], \\ u_{sn}^{(i)}(t - \tau_{sn}^{(i)} - dt_1^{(i)}) & \text{pour } t \in [\tau_{sn}^{(i)} + dt_1^{(i)}, \tau_{sn}^{(i)} + dt_1^{(i)} + dt_2^{(i)}], \\ A_{sn}^{(i)} \cdot g_{sn}^{(i)}(t - \tau_{sn}^{(i)} - dt_1^{(i)} - dt_2^{(i)}) & \text{pour } t \in [\tau_{sn}^{(i)} + dt_1^{(i)} + dt_2^{(i)}, \tau_{sn}^{(i)} + dt_1^{(i)} + dt_2^{(i)} + dt_3^{(i)}], \\ 0 & \text{sinon.} \end{cases} \quad (4.34)$$

Dans ce modèle, comme pour les clics de cétacés $n_{sn} = \lfloor T/\delta_{sn} \rfloor$ est le nombre de claquements présents durant la durée totale du signal et également le nombre de fois que se répète l'intervalle moyen entre les claquements δ_{cl} durant la période T . L'instant $\tau_{sn}^{(i)}$, où le $i^{\text{ème}}$ claquement de pince $sn_i(t)$ débute, est déterminé de façon aléatoire grâce à l'équation

$$\tau_{sn}^{(i)} = i\delta_{sn} + 0.3 \cdot \delta_{sn} \cdot \alpha_{sn}^{(i)}, \quad (4.35)$$

avec $\alpha_{sn}^{(i)} \sim \mathcal{N}(0, 1)$. Comme affiché au centre de la figure 4.18, un claquement est décomposé en 3 périodes de durée $dt_1^{(i)}$, $dt_2^{(i)}$ et $dt_3^{(i)}$, toutes tirées aléatoirement à partir d'une distribution uniforme.

La première période $dt_1^{(i)}$, durant entre 80 et 120 μs , correspond à la durée de l'impulsion précurseuse du $i^{\text{ème}}$ claquement de pince et est initiée à l'instant $\tau_{sn}^{(i)}$. Durant cette période, $sn_i(t)$ a une croissance linéaire :

$$h_{sn}^{(i)}(t) = \frac{1}{dt_1^{(i)}} t. \quad (4.36)$$

$A_{sn}^{(i)} = \mu_{sn} + \sigma_{sn} \cdot \xi_{sn}^{(i)}$ est l'amplitude du $i^{\text{ème}}$ claquement de pince, μ_{sn} et σ_{sn} étant respectivement sa moyenne et son écart type, tandis que $\xi_{sn}^{(i)}$ est une variable aléatoire suivant une loi normale $\mathcal{N}(0, 1)$.

La deuxième période $dt_2^{(i)}$ correspond à la durée entre la fin de l'impulsion précurseuse et le pic de l'impulsion principale. Durant cette période, $u_{sn}^{(i)}(t - \tau_{sn}^{(i)} - dt_1^{(i)})$ décroît d'abord de façon quadratique et l'intégrale de $sn_i(t)$ sur cette période est égale à celle de l'impulsion précurseuse. Dans un second temps, $sn_i(t)$ subit une croissance exponentielle jusqu'à atteindre $A_{sn}^{(i)}$, en une durée $dt_3^{(i)}$ liée à l'effondrement idéal d'une unique bulle de cavitation et égale à la troisième période.

Enfin, durant la troisième période $dt_3^{(i)}$ qui dure entre 80 et 120 μs , $sn_i(t)$ est une série d'oscillations à décroissance exponentielle en réponse à l'impulsion principale :

$$g_{sn}^{(i)}(t) = \cos(2\pi f_{sn}^{(i)} t) \cdot e^{-\frac{t}{dt_3^{(i)}}}, \quad (4.37)$$

avec $f_{sn}^{(i)} = M/dt_3^{(i)}$, M correspondant au nombre moyen d'oscillations notables après l'impulsion.

Les impulsions ayant lieu pendant des durées extrêmement courtes, il faut nécessairement employer des fréquences d'échantillonnages très élevées (≥ 100 kHz) pour correctement distinguer l'intégralité de leur réponse spectrale. Les $sn_i(t)$ ont des réponses spectrales maximales

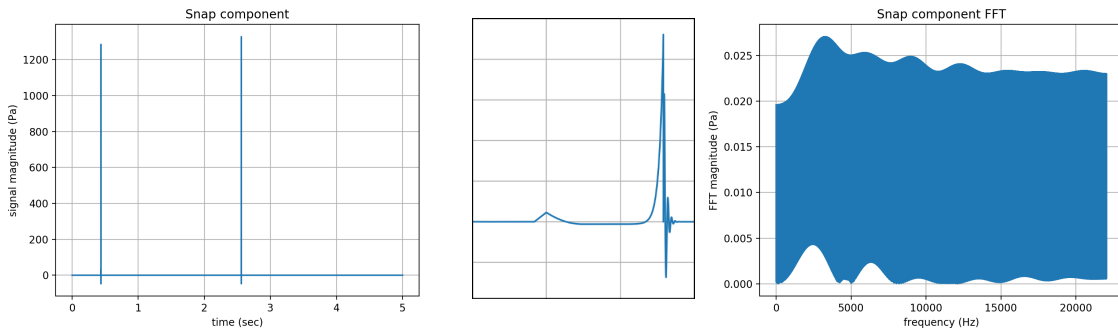


Figure 4.18: Représentation temporelle, à gauche, temporelle avec zoom, au milieu et fréquentielle à droite, de la composante de claquement enregistrée pendant 5 secondes à une fréquence d'échantillonnage $f_s = 44100$ Hz.

entre 2.5 et 4 kHz au niveau de leurs fondamentales. On voit également sur la figure 4.18 que la réponse spectrale du signal généré comporte des harmoniques similaires à celles de la figure 4.17. Sur la figure 4.19, chaque claquement est représenté par un trait vertical bien localisé en temps, mais est bien plus étalé en fréquence que les clics de cétacés visualisables sur la figure 4.16. L'étalement spectral de l'impulsion générée par la crevette claqueuse peut être observé sur des bandes larges de plus de 200 kHz, un record dans le règne animal.

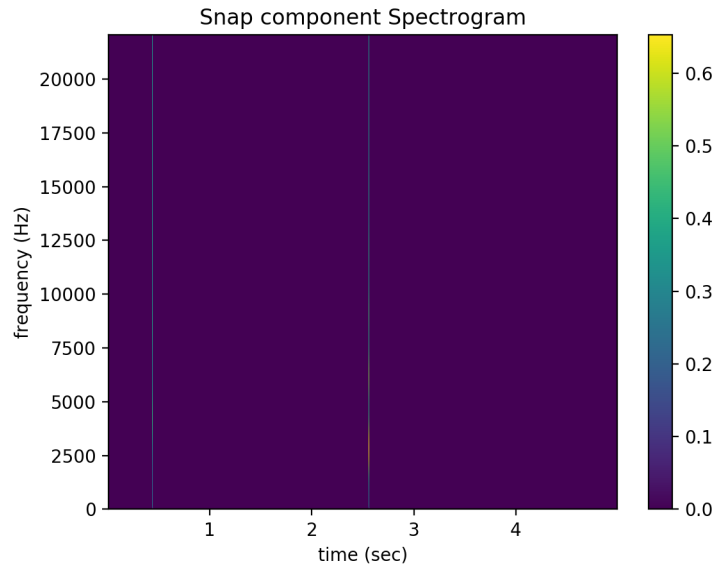


Figure 4.19: Spectrogramme de la composante de claquement associé à la figure 4.18.

5.3 Discussion

D'autres espèces marines, telles que certaines espèces de poissons produisant des bruits "métalliques", peuvent présenter des caractéristiques acoustiques uniques, mais les bruits de cétacés et de crevettes claqueuses sont très répandus et produisent des formes de bruits pouvant s'apparenter à des bruits d'origines humaines. La simulation de bruit biologique se restreindra ainsi qu'à ces deux classes, qui ne représentent qu'une fraction de la diversité de bruits acoustiques rencontrés dans l'océan.

6 Modélisation du bruit ambiant

6.1 Caractéristiques du bruit ambiant

Si on se réfère aux ouvrages de [Burdic & Bartram \(1984\)](#) et [Urlick \(1975\)](#), contrairement aux bruits de rayonnement émis par des navires à propulsion mécanique, ou tout autres sources potentielles et caractérisés par la direction de leur front d'onde, les bruits ambiants sont des bruits de fond, présent n'importe où en mer et ne provenant ni de défauts des capteurs ni de bruits du porteur. Dans notre modélisation du bruit ambiant nous restreindrons ainsi cette grande catégorie à des bruits qui ne varient pas, ou assez peu, selon la direction choisie pour notre formation de voie. Le niveau de bruit ambiant sous-marin dépendant à la fois de l'amplitude et de la densité des sources sonores avant leurs propagations aux récepteurs, il est commun d'observer d'importantes fluctuations de ce niveau lors d'un changement d'heure, de localisation ou de profondeur.

Une variation du niveau de bruit ambiant à une certaine échelle temporelle peut ainsi être

associée à la présence d'un type d'événement acoustique particulier. Pour des variations de l'ordre de la milliseconde, on peut s'attendre à des bruits impulsifs lointains pouvant être biologiques ou naturels, comme par exemple des craquements de glace. À l'ordre de quelques secondes ou de la minute on pourra considérer un bruit de trafic lointain ou des bancs de poissons tels que les poissons-grogneurs. Les variations de l'ordre d'une heure peuvent laisser penser à un changement au niveau météorologique qui va affecter le niveau de mer et ainsi le niveau de bruit produit par les vagues en surfaces. Les variations au cours d'une journée peuvent être dues à l'arrêt d'activités industrielles au niveau des côtes ou à la migration de population de poissons vers la surface durant la nuit pour se nourrir. Enfin, le niveau moyen de bruit ambiant change également entre les saisons.

L'une des principales sources de bruits ambiants est ainsi le bruit de trafic produit par le passage de plusieurs navires dans une voie maritime. Dès les années 60, [Wenz \(1962\)](#) avait soutenu que les navires traversant un bassin océanique pourraient produire un bruit de fond basse fréquence, relativement différent de celui d'un navire isolé qui lui est plutôt distinguable. Ces bruits, qu'on appelle usuellement bruit de trafic, peuvent alors être entendus à des centaines de *km* dans certaines régions du globe et ce, notamment dû au fait que les ondes acoustiques orientées quasiment à l'horizontal subissent moins de pertes de propagations sur de longues distances.

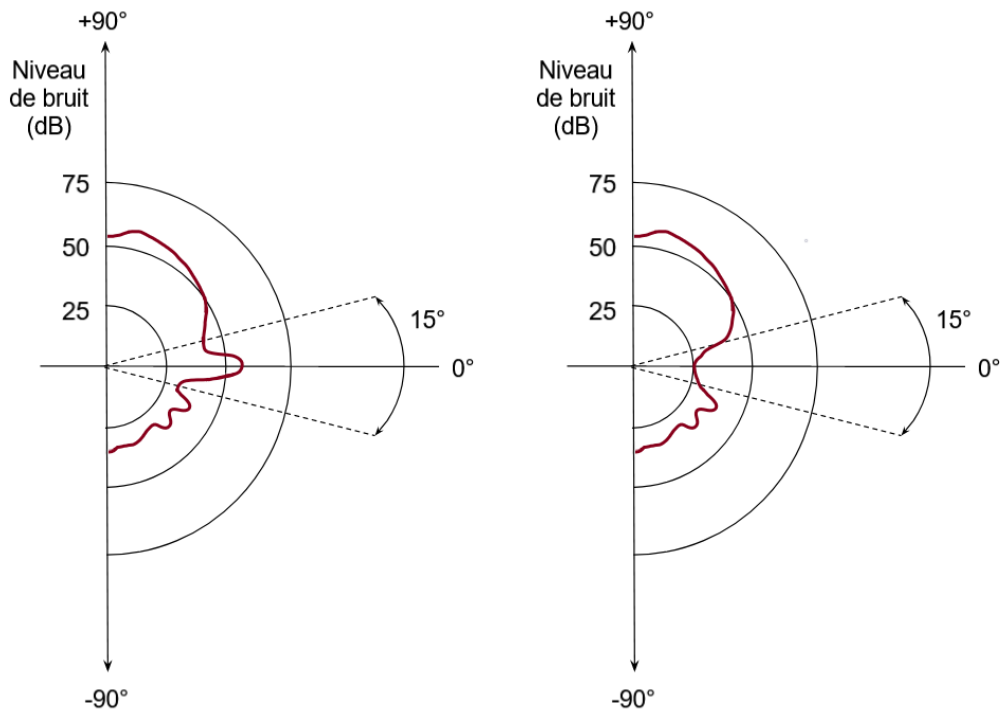


Figure 4.20: Schéma d'une distribution verticale d'un niveau de bruit ambiant typiquement trouvée en ASM, avec (à gauche) et sans prédominance du bruit de trafic (à droite).

Une autre source de bruit ambiant très importante est le bruit de mer ou bruit de surface due à l'agitation en surface. Il est notamment causé par la formation de bulles par des vagues quand le vent est fort, ou par de la pluie ou de la grêle quand la mer n'est pas agitée. Du fait que ce bruit provient de la surface, son niveau d'énergie mesuré est plus important quand on effectue une formation de voie dirigée vers la surface. De même, les réflexions de ces bruits sur les fonds de mers causent un niveau de bruit ambiant plus important vers le fond qu'à l'horizontal en l'absence de bruits de trafic, comme cela est illustré sur la figure 4.20.

Dans la pratique il est important de pouvoir estimer cette distribution du niveau de bruit ambiant selon le site (angle vertical), mais également selon le gisement (direction sur le plan horizontal) d'une formation de voie, du fait que plusieurs mesures de comparaisons peuvent être effectués sur différents couples, gisement et site, proches de la direction d'intérêt.

Hormis le bruit de trafic et le bruit de surface, d'autres sources de bruits ambiants peuvent être considérées comme stationnaires à l'échelle de plusieurs minutes. On peut notamment penser aux bruits sismiques, aux bruits de courants, ou à l'agitation thermique, tous définis sur des bandes de fréquences relativement différentes des deux sources précédemment citées, comme cela est indiqué sur la figure 4.21. Par la suite, du fait que ces sources de bruits ambiants ne soient pas prédominantes dans les bandes de fréquences souvent analysées, nous ne les intégrerons pas dans notre modèle de bruit ambiant.

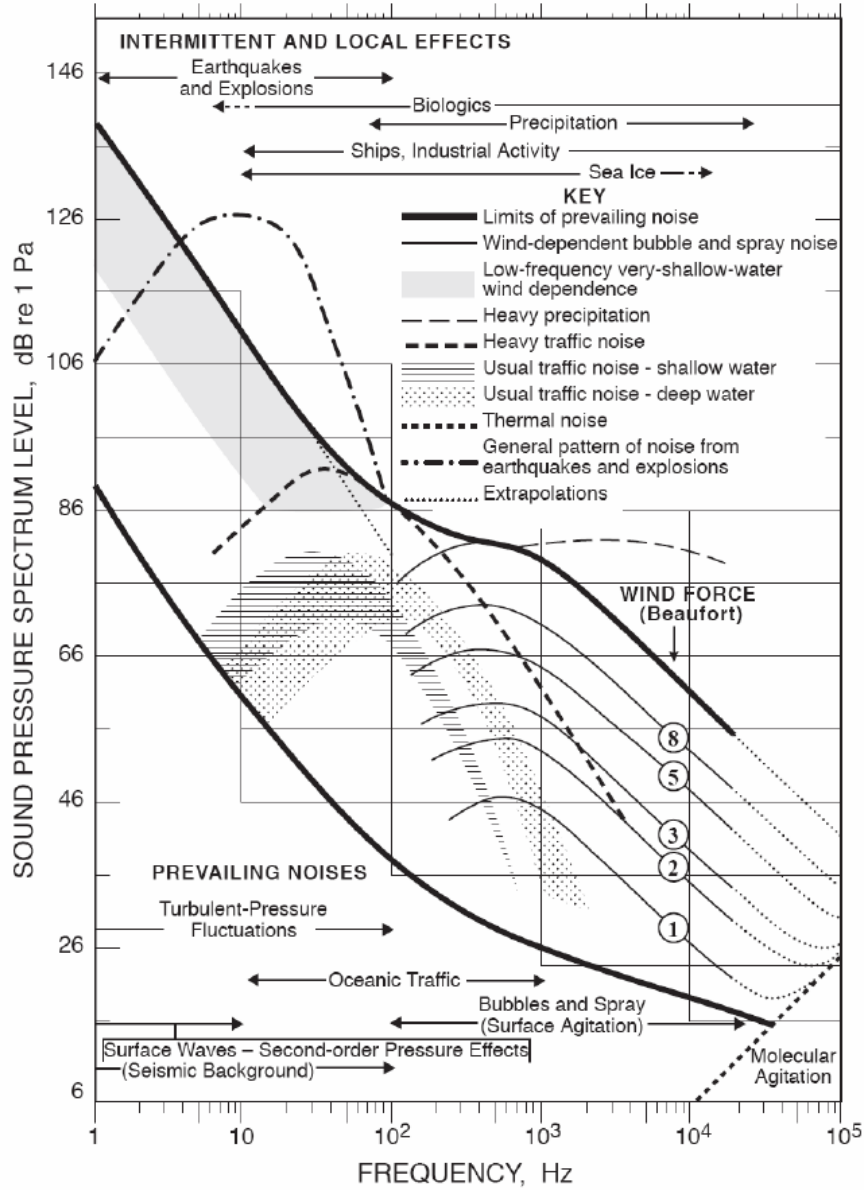


Figure 4.21: Représentation des niveaux limites de bruits ambiants pour des sources continues et intermittentes. Les flèches horizontales en haut de la figure indiquent l'étalement spectral des différentes sources de bruits ambiants : sismique, biologique, précipitations, navires et activités industrielles et bruits de glacier. Différents tracés pour chaque type de bruit sont dessinés pour indiquer qu'il est possible de rencontrer différents niveaux d'intensité dans la nature, en fonction par exemple de la force du vent. Tiré de l'article de Wenz (1962).

6.2 Modèle de bruit ambiant

Soit $n_a(t)$ un bruit ambiant, combinaison d'une composante de bruit de trafic $n_{tr}(t)$ et d'une composante de bruit de mer $n_m(t)$, avec :

$$n_a(t) = n_{tr}(t) + n_m(t).$$

Les sources de ces composantes résultants de processus imprévisibles, elles sont toutes deux considérées comme des réalisations de processus stochastiques.

Pour le bruit de trafic, on générera un bruit blanc gaussien dans le domaine fréquentiel $\widehat{e}_{tr}(f)$, que l'on multipliera à un spectre de puissance associé au bruit de trafic $|\widehat{h}_{tr_m}(f)|$, afin d'obtenir $\widehat{n}_{tr}(f)$:

$$n_{tr}(t) = \mathcal{F}^{-1}\{\widehat{n}_{tr}(f)\} \quad (4.38)$$

$$= \mathcal{F}^{-1}\{\widehat{e}_{tr}(f) \cdot |\widehat{h}_{tr}(f)|\}. \quad (4.39)$$

On utilisera une méthode d'interpolation numérique pour obtenir une courbe discrète dans le domaine spectral définissant $|\widehat{h}_{tr}(f)|$. En particulier on se servira de courbes de Bézier quadratiques de degré 2, qui peuvent être définies par trois points de contrôle (Fréquence (Hz), Puissance (dB)). Dans notre modèle les points de contrôle sont issus d'un tirage aléatoire entre des valeurs de points de contrôle supérieurs $p_{sup}(i)$, $i = 1, \dots, 3$, pris sur la courbe de trafic intense de la figure 4.21 et des valeurs de points de contrôle inférieurs $p_{inf}(i)$, $i = 1, \dots, 3$, pris sur la courbe de trafic régulier de la même figure :

$$p(i) = p_{inf}(i) + \xi_{tr} (p_{sup}(i) - p_{inf}(i)), \quad (4.40)$$

où $\xi_{tr} \sim \mathcal{U}(0, 1)$. L'expression de $|\widehat{h}_{tr}(f)|$ est alors :

$$|\widehat{h}_{tr}(f)| = \sum_{i=1}^3 B_i^3(f) \cdot p(i), \quad (4.41)$$

où les B_i^3 sont les polynômes de Bernstein de degré 2.

Pour le bruit de mer, on utilisera une même méthode de génération de bruits, mais en employant les courbes de Knudsen, décrites par Marsh (1963), pour définir le spectre de puissance du bruit de mer $|\widehat{h}_{m_m}(f)|$. Ce dernier est obtenu en sélectionnant un niveau de mer m tiré aléatoirement, ce qui permet de définir le bruit de mer

$$n_m(t) = \mathcal{F}^{-1}\{\widehat{e}_m(f) \cdot |\widehat{h}_m(f)|\}. \quad (4.42)$$

Alternativement, on peut utiliser une modélisation plus poussée prenant en compte des paramètres caractérisant à la fois l'intensité du vent et de la pluie. En effet, les sources principales de bruit ambiant naturel de 1 à 50 kHz sont les bruits émis par le fracas des vagues ainsi que les éclaboussures dues au contact des gouttes d'eau ou des grêlons avec la surface de la mer. Le mécanisme principal derrière ce type de bruit est la résonance acoustique des bulles

d'air piégées sous l'eau par les vagues ou les éclaboussures comme le font remarquer [Ma et al. \(2005\)](#). Ces bulles produisent alors une onde acoustique à une fréquence de résonance

$$f = \frac{1}{2\pi a} \sqrt{\frac{3\gamma p_0}{\rho_0}}, \quad (4.43)$$

avec a le diamètre de la bulle, p_0 la pression locale, ρ_0 la densité locale et γ l'indice adiabatique de l'air, reflétant le rapport des capacités thermiques à pression constante et à volume constant d'un gaz et dépendant donc aussi de la taille de la bulle.

La forme spectrale du bruit de surface comporte généralement un pic situé entre 500 Hz et 20 kHz et est souvent uniformément décroissante après ce pic représentant la fréquence de résonance des bulles de taille prédominantes en surface. Les variations de forme spectrale peuvent être expliquées par le type de pluie ainsi que la force du vent présent en surface. La très grande quantité de bulle formée en surface permet de conserver une forme spectrale relativement stable sur des échelles de temps de quelques secondes (à part lors de fortes tempêtes) et cela est d'autant plus vrai que le récepteur est immergé profondément.

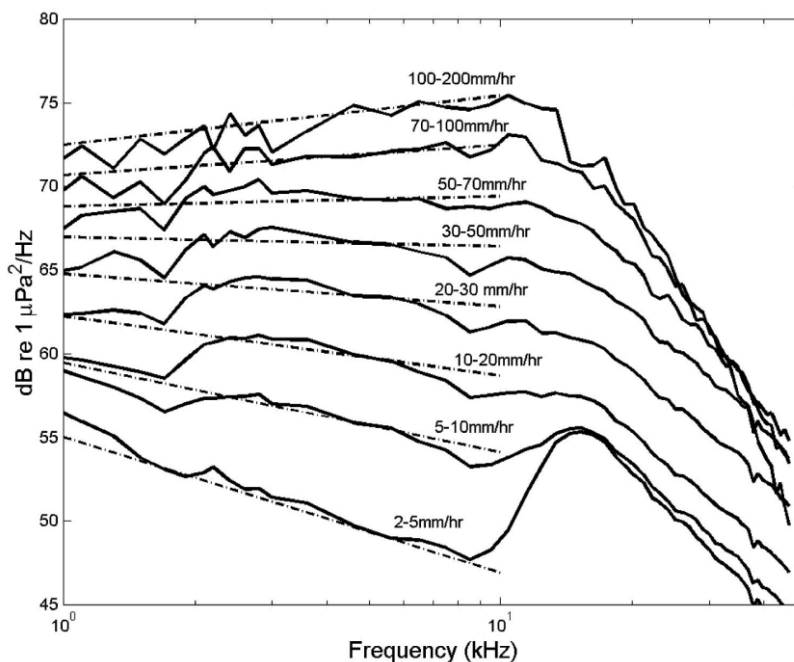


Figure 4.22: Représentation de l'intensité des bruits de surface en fonction de la fréquence, pour différents niveaux de précipitation à faible niveau de vent. Tiré de l'article de [Ma et al. \(2005\)](#).

La principale caractéristique du bruit de surface est l'interaction qui existe entre vent et pluie au niveau de la forme spectrale du bruit de surface. Pour des faibles niveaux de précipitations et peu de vent, il existe un second pic de puissance entre 10 et 20 kHz dans la forme spectrale du bruit de surface comme on peut le voir sur la figure 4.22. Des niveaux de vents plus forts vont venir atténuer ce second pic, avec une houle plus forte, comme on peut le voir sur la

figure 4.23.

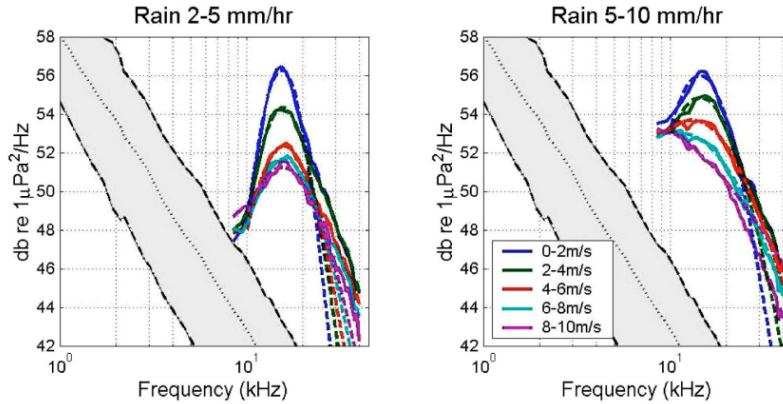


Figure 4.23: Représentation des interactions entre vent et pluie sur les bruits de surface pour différents niveaux de vent et pour deux faibles niveaux de précipitations. Tiré de l'article de *Ma et al.* (2005).

Remarque

De nombreuses études sur la nature du bruit ambiant ayant avancées qu'il n'était pas strictement gaussien, la section 5, dans l'annexe A, présente un moyen de générer un bruit aléatoire $e(t)$ non gaussiens.

7 Modélisation des phénomènes de propagation

7.1 Effet Doppler

L'effet Doppler déforme le signal temporel d'une source du point de vue du récepteur, dans la situation où sources et récepteurs sont en mouvement relatif. Dans le cas où une source et un récepteur se déplacent le long d'un même axe dans le milieu de propagation de l'onde et à des vitesses constantes, l'effet Doppler se caractérise par un décalage linéaire de chaque élément du spectre de raies du signal de la source. Si on considère une source pure (une seule raie) de fréquence f_0 , la fréquence perçue par le récepteur f est obtenue selon la formule ci-dessous :

$$f = f_0 \frac{c + v_r}{c + v_s} \quad (4.44)$$

Avec c la célérité du son dans l'eau (simplification du profil de célérité), v_r la vitesse du récepteur et v_s la vitesse de la source. Pour appliquer cet effet dans le domaine temporel, tout en permettant au récepteur et à la source de se déplacer sur des axes différents, on doit se baser sur le délai de propagation entre source et récepteur. Le délai d'arrivée de l'onde dû à la distance source/récepteur d , est $\delta t = \frac{d}{c}$, avec c la célérité du son sous l'eau que l'on considérera constante dans notre modèle simplifié. Ainsi, si la distance d est amenée à changer à travers le temps, le délai δt sera également modifié.

L'effet Doppler dans le domaine temporel consiste donc à appliquer ce délai de trajet à notre échelle temporelle, soit décaler les échantillons en fonction de ce délai. Néanmoins, puisque le signal temporel est échantillonné, il est rare que le délai soit un multiple de notre fréquence d'échantillonnage. Il est alors souvent nécessaire d'utiliser un schéma d'interpolation à cet effet. Pour appliquer l'effet Doppler sur le trajet direct du signal discret $s(t_k) = s(k)$, nous utilisons ainsi une interpolation Lanczos du signal $s(t_k)$ à de nouvelles valeurs temporelles discrètes t'_k . Chaque échantillon temporel t'_k est décalé de t_k proportionnellement au délai de trajet δt du front d'onde entre la source et le récepteur. La figure 4.24 donne un exemple de délai de trajet δt variant dans le temps à cause des déplacements de la source par rapport au récepteur. Enfin, dans le but de rendre l'impact des artefacts produit par cette interpolation négligeable, nous générerons d'abord un signal $s(t_k)$ à une fréquence $fs_0 \gg fs$ que nous sous-échantillonnerons à la fréquence fs après modélisation des phénomènes de propagation sur le signal.

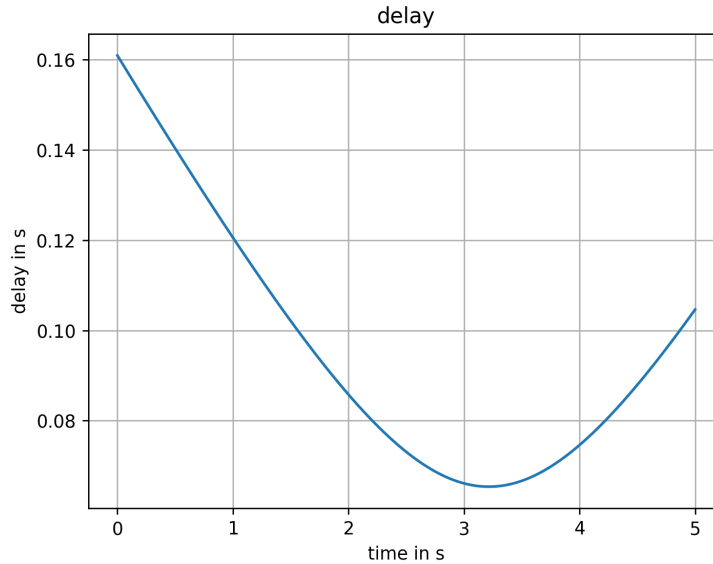


Figure 4.24: Délai d'arrivée du son émis par une source à vitesse constante vers un récepteur placé sous la trajectoire de la source.

7.2 Effet d'absorption du milieu

Si on considère les pertes de puissance d'un front d'onde émis par un rayonnement acoustique, une méthode naïve serait d'utiliser une équation sphérique

$$\mathcal{P} = 4\pi r^2 i, \quad (4.45)$$

où \mathcal{P} est la puissance acoustique en W , r le rayon de propagation de l'onde en m et i est l'intensité en W/m^2 tels que :

$$i = \frac{p^2}{\rho_0 c}, \quad (4.46)$$

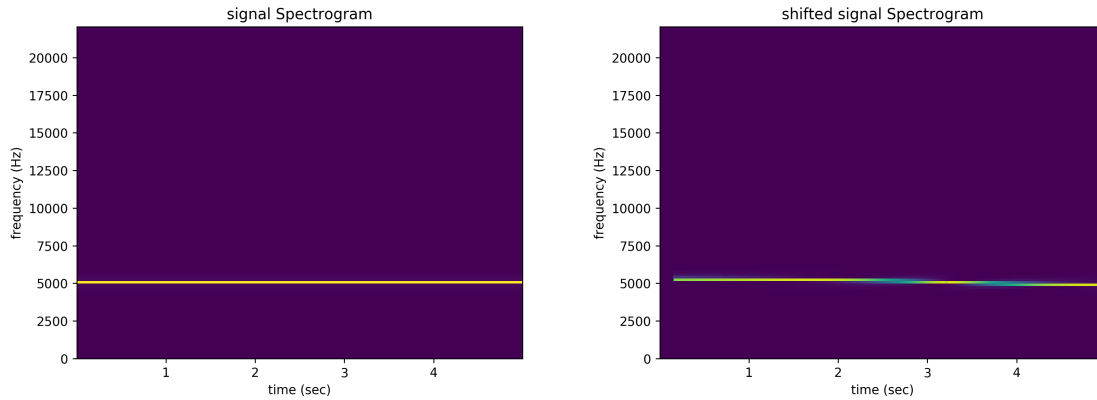


Figure 4.25: Spectrogramme d'un signal pur non déformé $s(t) = \sin(\omega t + \phi)$ de fréquence $f_s = 5000$ Hz (à gauche) et spectrogramme du même signal déformé par l'effet Doppler (à droite).

avec ρ_0 la densité du milieu, c la célérité du son dans l'eau et p la pression émise par la source en Pa . Néanmoins le milieu sous-marin n'atténue pas toutes les bandes de fréquences de façon uniforme. Aussi, l'utilisation d'un profil d'absorption volumique du milieu marin permet de déterminer les pertes d'amplitude subies par le front d'onde durant son trajet en dB par km . En particulier, il est possible d'obtenir le signal temporel atténué par l'absorption volumique de l'eau de mer en calculant la transformée de Fourier inverse des composantes spectrales du signal de référence (relevé à 1 m de la source) en dB, à laquelle on aura soustrait la perte d'amplitude correspondante :

$$s(k) = \mathcal{F}^{-1}\{\hat{s}(k) - a(k)\}, \quad (4.47)$$

avec \mathcal{F}^{-1} la transformée de Fourier inverse discrète, $s(k)$ le vecteur des composantes fréquentielles et $a(k)$ le vecteur de perte d'amplitude, calculé à partir du profil d'absorption et de la longueur parcourue par le front d'onde de la source aux récepteurs.

7.3 Trajets multiples

La propagation des ondes acoustique dans l'environnement sous-marin est intrinsèquement liée aux caractéristiques de ce dernier. On peut notamment penser au profil de célérité du son dans l'eau, à la bathymétrie, aux propriétés acoustiques des fonds marins et de la surface, à la dynamique des océans ainsi que beaucoup d'autres paramètres. Aussi, du fait des caractéristiques de l'environnement de propagation, le bruit rayonné par une unique source directionnelle peut engendrer l'arrivée sur les capteurs de plusieurs fronts d'ondes différents. Il est également fréquent que plusieurs de ces fronts d'ondes, provenant de la même source, atteignent l'antenne réceptrice avec des angles d'incidence très similaires. On considérera ainsi que même après avoir effectué une formation de voie sur l'antenne, le signal reçu $x(t)$ peut contenir à la fois un trajet direct $s_0(t)$, ainsi qu'un nombre fini de trajets réfléchis $s_j(t)$, j

pouvant aller de 1 à n_t :

$$x(t) = s_0(t) + \sum_{j=1}^{n_t} s_j(t) + n_a(t). \quad (4.48)$$

Chacun de ces trajets réfléchis $s_j(t)$ est une copie de $s(t)$ décalée par son propre retard τ_j , atténuée par des pertes dues à l'absorption ainsi que la formation de voie et déformée par l'effet Doppler due à son propre délai.

La modélisation des trajets multiples peut être effectuée grâce à différents modèles de propagation des fronts d'ondes, essayant tous de modéliser le canal de propagation réel le plus finement possible, comme en attestent les travaux de [Socheleau, Passerieux & Laot \(2009\)](#). Ces modélisations sont importantes quand on essaye de détecter ces trajets multiples, comme cela est fait dans les travaux de [THEYS et al. \(1991\)](#). L'objet de notre étude ne concerne cependant que la classification après traitement d'antenne, i.e. des signaux 1D détectés provenant d'une voie énergétique. Il n'est ainsi pas nécessaire de connaître les angles d'arrivées des trajets secondaires sur les différents capteurs ni d'être extrêmement réaliste, mais simplement de modéliser un phénomène qui modifie l'aspect des différents signaux présents dans notre jeu d'apprentissage. Plusieurs modélisations ont ainsi été envisagées pour nos travaux :

- **Modélisation des trajets des fronts d'ondes par rayon (Outil interne Thales ou BELLHOP)** : Les trajets des rayons émis par la source sont modélisés dans un environnement 2D, en dessinant plusieurs trajectoires partant de la source dans diverses directions. Chaque trajectoire arrivant vers le/les récepteurs, est modélisée par un retard et une perte d'amplitude due aux réflexions sur la surface et le fond marin. Chaque modélisation n'est cependant valide que pour une certaine gamme de fréquences.
- **Modélisation par modes pour les basses fréquences (Kraken)** : À basse fréquence, la longueur d'onde de nos signaux commence à poser problème pour la modélisation par rayon. Pour une fréquence de 50 Hz par exemple, la longueur d'onde est d'environ 30 m et on obtient difficilement des rayons moins larges que quelques centaines de mètres. Ainsi, du fait que le champ d'influence du rayon est seulement précis près de l'axe du rayon, la largeur de ces rayons devient une limitation de cette méthode comme stipulé par [Porter \(2016\)](#) dans l'environnement BELLHOP. Une modélisation par modes devrait donc être employée pour modéliser les trajets multiples à basse fréquence où les pertes d'amplitude sont en $o(r)$, et non en $o(r^2)$ comme dans la modélisation par rayons.
- **Modélisation simpliste** : Une dernière solution est de simplement générer de façon aléatoire des pertes d'amplitude et des temps de décalage entre les trajets secondaires et le trajet direct, sans tenir compte de l'environnement de propagation.

8 Conclusion

Dans ce chapitre nous avons fait un tour des différentes sources de bruits dans l'environnement ASM et décrit l'implémentation de modèles de bruits acoustiques souvent rencontrés en ASM tels que les bruits de navires, bruits de cétacés ou encore les bruits de surface. Pour obtenir un jeu de données de bruits ASM assez réaliste, nous avons ensuite proposé une chaîne

de simulation comportant, le choix des caractéristiques des sources, la modélisation des phénomènes de propagation et des opérations de data augmentation. Néanmoins, dans les deux chapitres suivants, nous utiliserons simplement le modèle de bruit de cavitation d'hélice et une modélisation du bruit ambiant comme bruit blanc gaussien, afin d'effectuer une étude plus cadrée de la détection de bruits d'hélice, au prix d'une simplification de la complexité de l'environnement ASM.

Chapitre 5

Détection binaire

1 Introduction

Dans ce chapitre, nous partons du modèle de bruit de cavitation d'hélice décrit dans le chapitre précédent pour poser un problème de détection binaire maîtrisé. À partir du maximum de vraisemblance, nous pouvons déduire l'expression du détecteur de Bayes optimal de ce problème et ce, pour différentes hypothèses de retards, coloration du signal etc Nous proposons par la suite une architecture profonde explicable permettant d'approximer efficacement la structure mathématique de ce détecteur optimal et comparons ses performances à des modèles de comparaison profonds. En particulier, nous proposons d'appliquer un module de réseau de neurones d'approximation de fonction pour permettre à nos modèles explicables d'apprendre les opérations élémentaires du détecteur optimal étape par étape.

2 Problème de détection de bruit d'hélice blanc

2.1 Énoncé du problème

Nous considérons ici le problème de détection de bruit d'hélice de la section 2.2 du chapitre 4, en assumant que les bruits de cavitation $n_i(t)$ et le bruit ambiant $n_a(t)$, sont des bruits blancs gaussiens. Nous supposons également qu'il n'y a qu'une seule signature de bruit d'hélice et que ses motifs $m_{i,\omega}(t)$, ainsi que les écarts types σ_η et σ_a sont connus. Notre problème de détection consiste ainsi à décider si le signal discret \mathbf{x} contient uniquement du bruit ambiant (hypothèse H_0) ou s'il contient un signal provenant d'un bruiteur isolé (4.2) (hypothèse H_1) :

$$\begin{aligned} H_0 &: \{x(k) = n_a(k), k \in \llbracket K \rrbracket\}, \\ H_1 &: \{x(k) = s_\eta(k) + n_a(k), k \in \llbracket K \rrbracket\}, \end{aligned} \tag{5.1}$$

où $\llbracket K \rrbracket = \{1, \dots, K\}$. Dans l'hypothèse sans source H_0 , \mathbf{x} suit une distribution gaussienne multivariée de moyenne nulle et de matrice de covariance $\Sigma_0 = \sigma_a^2 \cdot I_K$, avec I_K la matrice identité de dimension $K \times K$. Dans l'hypothèse H_1 , en se référant à la section (Poor 2013, III.B), \mathbf{x} suit une distribution gaussienne multivariée $\mathcal{N}(0, \Sigma_1 = \Sigma_\eta + \sigma_a^2 \cdot I_K)$ de moyenne

nulle et de matrice de covariance diagonale,

$$\Sigma_\eta = \text{diag}[\sigma_\eta^2(1), \dots, \sigma_\eta^2(K)], \quad (5.2)$$

$$\Sigma_1 = \text{diag}[\sigma_\eta^2(1) + \sigma_a^2, \dots, \sigma_\eta^2(K) + \sigma_a^2], \quad (5.3)$$

$$= \begin{bmatrix} \sigma_\eta^2(1) + \sigma_a^2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_\eta^2(K) + \sigma_a^2 \end{bmatrix}. \quad (5.4)$$

Dans la suite de cette section, nous considérerons un ensemble de données d'entraînement $\mathcal{S} = \{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$ où $(\mathbf{x}^{(i)}, y^{(i)})$ est composé d'un signal reçu $\mathbf{x}^{(i)} \in \mathbb{R}^K$ et de son label $y^{(i)} \in \{0, 1\}$ tel que $y^{(i)} = j$ signifie que $\mathbf{x}^{(i)}$ suit l'hypothèse H_j . Les échantillons d'apprentissage $(\mathbf{x}^{(i)}, y^{(i)})$ sont indépendants et suivent la distribution $\mathcal{D}(\mathbf{x}, y)$:

$$\mathcal{D}(\mathbf{x}, y) = q_0 \text{Pr}_0(\mathbf{x}) + q_1 \text{Pr}_1(\mathbf{x}), \quad (5.5)$$

où $\text{Pr}_0(\cdot)$ et $\text{Pr}_1(\cdot)$ dénotent respectivement les mesures de probabilités quand \mathbf{x} suit H_0 et H_1 . Les probabilités $q_0 = \Pr(y = 0)$ et $q_1 = \Pr(y = 1) = 1 - q_0$ sont respectivement la probabilité d'occurrence de H_0 et H_1 .

Remarque

Afin d'éviter toute confusion quant aux performances des détecteurs optimaux, il est nécessaire que le détecteur optimal de bruit d'hélice, sans délais aléatoires, ait de meilleures performances que le détecteur optimal quand un délai aléatoire est possible. Aussi, nous considérerons que l'instant t_1 , utilisé pour la génération des données tel que $\mathbf{x} = \{x(t_1), \dots, x(t_K)\}$, permet de maximiser l'intégrale de l'énergie de la signature $\sigma_\eta^2(t)$:

$$t_1 = \max_{\tau_0} \int_{\tau_0}^{\tau_0+T} \sigma_\eta^2(t) dt, \quad (5.6)$$

avec $T = K/fs$.

2.2 Apprentissage par réseau de neurones profond

Une règle de décision pour résoudre le problème de détection (5.1) est une fonction $\delta : \mathbb{R}^K \rightarrow \{0, 1\}$ qui décide H_i quand $\delta(\mathbf{x}) = i$. On considère une fonction de décision sigmoïde binaire $\hat{\delta}_\theta : \mathbb{R}^K \rightarrow [0, 1]$ basée sur la sortie d'un réseau de neurones $f_\theta(\mathbf{x})$:

$$\hat{\delta}_\theta(\mathbf{x}) = \mathbb{1}_{\{\rho_s(f_\theta(\mathbf{x})) \geq 0.5\}} = \begin{cases} 0 & \text{si } f_\theta(\mathbf{x}) < 0, \\ 1 & \text{si } f_\theta(\mathbf{x}) \geq 0, \end{cases} \quad (5.7)$$

où $\mathbb{1}_{\mathcal{A}}$ est la fonction indicatrice, $\rho_s : t \mapsto 1/(1 + e^{-t})$ est la fonction d'activation sigmoïde et $\theta \in \mathbb{R}^P$ désigne l'ensemble des paramètres pouvant être entraînés suivant la terminologie employée par [Goodfellow et al. \(2016\)](#). Le réseau de neurones est entraîné de manière conventionnelle sur l'ensemble d'apprentissage \mathcal{S} en minimisant le risque empirique $\mathcal{R}_N(f_\theta)$ comme le définissent [Devroye et al. \(2013\)](#) :

$$\mathcal{R}_N(f_\theta) := \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_\theta(\mathbf{x}^{(i)}), y^{(i)}), \quad (5.8)$$

où \mathcal{L} est ici l'entropie croisée binaire, c'est-à-dire

$$\mathcal{L}(z, y) = y \log(z) + (1 - y) \log(1 - z), \quad (5.9)$$

pour tout $0 \leq z, y \leq 1$. La performance générale de $\widehat{\delta}_\theta(\mathbf{x})$ est mesurée avec le risque réel $\mathcal{R}(f_\theta)$ pour la fonction de perte idéale "0-1", $\mathcal{L}_{0-1}(z, y) = \mathbb{1}_{\{z=y\}}$ et est comparé au risque minimum \mathcal{R}^* , obtenu avec le détecteur de Bayes théorique :

$$\mathcal{R}(f_\theta) := \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}}[\mathcal{L}_{0-1}(\widehat{\delta}_\theta(\mathbf{x}), y)], \quad (5.10)$$

$$\mathcal{R}^* := \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}}[\mathcal{L}_{0-1}(\delta^*(\mathbf{x}), y)]. \quad (5.11)$$

En apprentissage statistique, la qualité de l'apprentissage, c'est-à-dire la convergence d'un modèle $\widehat{\delta}_\theta(\mathbf{x})$ vers $\delta^*(\mathbf{x})$, peut donc être évaluée par la différence $\mathcal{R}(f_\theta) - \mathcal{R}^*$, i.e. l'excès de risque de la section 2 chapitre 3. Notre méthode est cependant basée sur le contrôle théorique de l'erreur d'approximation $\min_\theta \mathcal{R}(f_\theta) - \mathcal{R}^*$. En effet, une petite erreur d'approximation garantit que le détecteur de Bayes optimal peut être approché par une classe de réseaux de neurones donnée, en termes de performances. Cela ne donne cependant aucune garantie sur le fait que le réseau de neurones ait appris une structure explicable. En d'autres termes, même si la convergence réussie, la structure du réseau de neurones reste généralement une boîte noire. Pour cette raison, dans la sous-section suivante, nous calculons le classifieur de Bayes optimal pour montrer qu'il a une structure naturellement explicable que nous voudrions préserver dans notre classe d'architectures profondes bayésiennes.

2.3 Règle de détection bayésienne optimale

Lorsque la distribution $\mathcal{D}(\mathbf{x}, y)$ dans (5.5) est connue, un bref calcul basé sur le rapport de vraisemblance (5.51) permet de déduire la forme du détecteur de Bayes $\delta^* : \mathbb{R}^K \rightarrow [0, 1]$:

$$\delta^*(\mathbf{x}) = \begin{cases} 0 & \text{si } \Lambda^*(\mathbf{x}) = \sum_{k=1}^K b_k x^2(k) \leq \tau, \\ 1 & \text{sinon,} \end{cases} \quad (5.12)$$

où $\Lambda^*(\mathbf{x})$ est la fonction d'estimation et les coefficients

$$b_k = \frac{1}{2} \left(\frac{1}{\sigma_a^2} - \frac{1}{\sigma_\eta^2(k) + \sigma_a^2} \right) \geq 0, \quad (5.13)$$

peuvent être déduits de l'expression du logarithme du rapport de vraisemblance :

$$\log(L(\mathbf{x})) = \frac{1}{2} \log(|\Sigma_0|) - \frac{1}{2} \log(|\Sigma_1|) - \frac{1}{2} \mathbf{x}^T (\Sigma_1^{-1} - \Sigma_0^{-1}) \mathbf{x} \quad (5.14)$$

$$= \frac{1}{2} \log(|\Sigma_0|) - \frac{1}{2} \log(|\Sigma_1|) + \frac{1}{2} \mathbf{x}^T (\sigma_a^{-2} \cdot I_K - (\sigma_a^2 \cdot I_K + \Sigma_\eta)^{-1}) \mathbf{x} \quad (5.15)$$

$$= \frac{1}{2} \log(|\Sigma_0|) - \frac{1}{2} \log(|\Sigma_1|) + \frac{1}{2} \mathbf{x}^T (\sigma_a^{-2} \cdot \Sigma_\eta (\sigma_a^2 \cdot I_K + \Sigma_\eta)^{-1}) \mathbf{x}. \quad (5.16)$$

L'hypothèse de blancheur pour le bruit ambiant et le bruit d'hélice induit ainsi un détecteur basé sur la somme de variables indépendantes, mais pas identiquement distribuées. Le signal $k \mapsto b_k$ joue le rôle d'un filtre adapté qui permet de calculer un score de profil devant être comparé à un seuil, donnant ainsi une structure de détecteur très répandue, comme en attestent [Meillier et al. \(2015\)](#). Cette règle de décision est à la fois le test de Bayes optimal ainsi que le test de Neyman-Pearson optimal. Le choix du seuil détermine l'optimalité du test comme décrit dans l'ouvrage de [Poor \(2013\)](#). On remarquera que dans les deux hypothèses $\Lambda^*(\mathbf{x})$ suit une distribution Khi-carré généralisée. Par conséquent, les probabilités d'erreur de cette règle de décision ne peuvent pas être exprimées sous forme fermée. Dans le but de tracer des courbes ROC, on peut cependant utiliser le théorème de la limite centrale généralisée pour déduire un test de Neyman-Pearson avec des distributions approximées.

Sous l'hypothèse H_0 , $\frac{x(k)}{\sigma_a}$ suit une distribution normale et par conséquent, $\frac{x^2(k)}{\sigma_a^2} \sim \chi^2(1)$. On peut alors déduire la moyenne :

$$\begin{aligned} \mathbb{E}[\Lambda^*(\mathbf{x})|H_0] &= \mathbb{E} \left[\sum_{k=1}^K \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right) \frac{x^2(k)}{\sigma_a^2} \right] \\ &= \sum_{k=1}^K \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right) \mathbb{E} \left[\frac{x^2(k)}{\sigma_a^2} \right] \\ &= \sum_{k=1}^K \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right) = \mu_{\Lambda_0^*} \end{aligned} \quad (5.17)$$

et la variance :

$$\begin{aligned} \mathbb{V}[\Lambda^*(\mathbf{x})|H_0] &= \mathbb{V} \left[\sum_{k=1}^K \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right) \frac{x^2(k)}{\sigma_a^2} \right] \\ &= \sum_{k=1}^K \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right)^2 \mathbb{V} \left[\frac{x^2(k)}{\sigma_a^2} \right] \\ &= 2 \sum_{k=1}^K \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right)^2 = \sigma_{\Lambda_0^*}^2 \end{aligned} \quad (5.18)$$

Le théorème central limite de Lyapunov stipule que pour $\{X_1, X_2, \dots\}$ une séquence de variables aléatoires indépendantes, d'espérance μ_k et de variance σ_k finies, alors si pour un $\delta > 0$, la condition de Lyapunov

$$\lim_{K \rightarrow \infty} \frac{1}{s_K^{2+\delta}} \sum_{k=1}^K \mathbb{E} \left[|X_k - \mu_k|^{2+\delta} \right] = 0, \quad (5.19)$$

est satisfaite, alors la somme des $\frac{X_k - \mu_k}{s_K}$ converge en distribution vers une variable aléatoire normale standard/centrée réduite quand K tend vers l'infini, avec $s_K^2 = \sum_{k=1}^K \sigma_k^2$. Ainsi, nous devons vérifier que dans notre cas, les variables aléatoires $|X_k| = b_k x^2(k)$, ont des moments d'ordre $(2 + \delta)$, et que le taux de croissance de ces moments est limité par la condition de Lyapunov. Pour l'hypothèse H_0 et avec $\delta = 1$ on obtient :

$$\begin{aligned}
 \frac{1}{s_K^3} \sum_{k=1}^K \mathbb{E} [|X_k - \mu_k|^3] &= \frac{1}{(\sigma_{\Lambda_0^*}^2)^{3/2}} \sum_{k=1}^K \mathbb{E} \left[\left| \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right) \left(\frac{x^2(k)}{\sigma_a^2} - 1 \right) \right|^3 \right] \\
 &= \frac{1}{(\sigma_{\Lambda_0^*}^2)^{3/2}} \sum_{k=1}^K \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right)^3 \mathbb{E} \left[\left| \frac{x^2(k)}{\sigma_a^2} - 1 \right|^3 \right] \\
 &= C_1 \cdot \frac{\sum_{k=1}^K \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right)^3}{\left(\sum_{k=1}^K \left(1 - \frac{\sigma_a^2}{\sigma_\eta^2(k) + \sigma_a^2} \right)^2 \right)^{3/2}} \leq C_2 \cdot \frac{1}{\sqrt{K}},
 \end{aligned} \tag{5.20}$$

avec C_1 et C_2 deux constantes. La condition de Lyapunov est ainsi valide pour l'hypothèse H_0 . Esseen a également prouvé que pour tout K , il existe une constante absolue C_0 telle que :

$$\sup_{x \in \mathbb{R}} |F_K(x) - F(x)| \leq C_0 \cdot \psi_0, \tag{5.21}$$

où F_K dénote la fonction de répartition (CDF) pour μ_k et σ_k , F est la CDF de la distribution normale standard et $\psi_0 = (\sum_{k=1}^K \sigma_k^2)^{-3/2} \cdot \sum_{k=1}^K \mathbb{E}[|X_k|^3]$. Esseen (1956) et Shevtsova (2010) ont également établi respectivement une borne inférieure et supérieure pour C_0 :

$$\frac{\sqrt{10} + 3}{6\sqrt{2\pi}} \simeq 0.4097 \leq C_0 \leq 0.55998. \tag{5.22}$$

En conséquence, on obtient de (5.20) et de (5.21) qu'il existe une constante C_0^* telle que :

$$\sup_{x \in \mathbb{R}} |F_K(\Lambda^*(\mathbf{x})) - F(\Lambda^*(\mathbf{x}))| \leq C_0^* \cdot \frac{1}{\sqrt{K}}. \tag{5.23}$$

Sous l'hypothèse H_1 , $\frac{x(k)}{\sqrt{\sigma_\eta^2(k) + \sigma_a^2}}$ suit une distribution normale et ainsi, $\frac{x^2(k)}{\sigma_\eta^2(k) + \sigma_a^2} \sim \chi^2(1)$. On peut alors déduire la moyenne :

$$\begin{aligned}
 \mathbb{E}[\Lambda^*(\mathbf{x})|H_1] &= \mathbb{E} \left[\sum_{k=1}^K \left(\frac{\sigma_\eta^2(k) + \sigma_a^2}{\sigma_a^2} - 1 \right) \frac{x^2(k)}{\sigma_\eta^2(k) + \sigma_a^2} \right] \\
 &= \sum_{k=1}^K \left(\frac{\sigma_\eta^2(k) + \sigma_a^2}{\sigma_a^2} - 1 \right) \mathbb{E} \left[\frac{x^2(k)}{\sigma_\eta^2(k) + \sigma_a^2} \right] \\
 &= \sum_{k=1}^K \left(\frac{\sigma_\eta^2(k) + \sigma_a^2}{\sigma_a^2} - 1 \right) = \mu_{\Lambda_1^*}
 \end{aligned} \tag{5.24}$$

et la variance

$$\begin{aligned}
 \mathbb{V}[\Lambda^*(\mathbf{x})|H_1] &= \mathbb{V} \left[\sum_{k=1}^K \left(\frac{\sigma_\eta^2(k) + \sigma_a^2}{\sigma_a^2} - 1 \right) \frac{x^2(k)}{\sigma_\eta^2(k) + \sigma_a^2} \right] \\
 &= \sum_{k=1}^K \left(\frac{\sigma_\eta^2(k) + \sigma_a^2}{\sigma_a^2} - 1 \right)^2 \mathbb{V} \left[\frac{x^2(k)}{\sigma_\eta^2(k) + \sigma_a^2} \right] \\
 &= 2 \sum_{k=1}^K \left(\frac{\sigma_\eta^2(k) + \sigma_a^2}{\sigma_a^2} - 1 \right)^2 = \sigma_{\Lambda_1^*}^2.
 \end{aligned} \tag{5.25}$$

Tout les $x(k)$ étant indépendants les uns des autres, le théorème central limite généralisé stipule que $\frac{\Lambda^*(\mathbf{x}) - \mu_{\Lambda_0^*}}{\sigma_{\Lambda_0^*}} \sim \mathcal{N}(0, 1)$ si H_0 est vraie et $K \rightarrow +\infty$. Similairement, pour H_1 , $\frac{\Lambda^*(\mathbf{x}) - \mu_{\Lambda_1^*}}{\sigma_{\Lambda_1^*}} \sim \mathcal{N}(0, 1)$. Par conséquent, grâce à cette manipulation, nous pouvons obtenir une approximation du seuil τ' adapté pour obtenir une erreur de type I désirée, α , au sein d'un test de Neyman-Pearson, comme décrit dans la section (Poor 2013, II.D) :

$$\begin{aligned}
 P(\Lambda^*(\mathbf{x}) \leq \tau' | H_0) &= 1 - \alpha \\
 P \left(\frac{\Lambda^*(\mathbf{x}) - \mu_{\Lambda_0^*}}{\sigma_{\Lambda_0^*}} \leq \frac{\tau' - \mu_{\Lambda_0^*}}{\sigma_{\Lambda_0^*}} | H_0 \right) &= 1 - \alpha \\
 F \left(\frac{\tau' - \mu_{\Lambda_0^*}}{\sigma_{\Lambda_0^*}} \right) &= 1 - \alpha \\
 \tau' &= F^{-1}(1 - \alpha) \sigma_{\Lambda_0^*} + \mu_{\Lambda_0^*}.
 \end{aligned} \tag{5.26}$$

On peut alors exprimer la puissance de notre test $1 - \beta$ en fonction de α :

$$\begin{aligned}
 P(\Lambda^*(\mathbf{x}) \leq \tau' | H_1) &= \beta \\
 P \left(\frac{\Lambda^*(\mathbf{x}) - \mu_{\Lambda_1^*}}{\sigma_{\Lambda_1^*}} \leq \frac{\tau' - \mu_{\Lambda_1^*}}{\sigma_{\Lambda_1^*}} | H_1 \right) &= \beta \\
 \beta &= F \left(\frac{\tau' - \mu_{\Lambda_1^*}}{\sigma_{\Lambda_1^*}} \right) \\
 1 - \beta &= 1 - F \left(\frac{F^{-1}(1 - \alpha) \sigma_{\Lambda_0^*} + \mu_{\Lambda_0^*} - \mu_{\Lambda_1^*}}{\sigma_{\Lambda_1^*}} \right).
 \end{aligned} \tag{5.27}$$

Connaissant la structure de la règle de décision de Bayes $\Lambda^*(\mathbf{x})$ et une estimation de ses performances pour n'importe quel seuil τ , la section suivante introduit une classe de réseaux de neurones profonds spécialement conçus pour apprendre la structure de ce détecteur optimal.

3 Apprentissage profond pour la détection de bruit d'hélice blanc

3.1 Architecture profonde bayésienne

Après avoir défini le problème de détection de bruit d'hélice et son détecteur optimal associé, nous souhaitons étudier la convergence d'un algorithme d'apprentissage profond vers

ce détecteur. Les décisions des algorithmes profonds sont cependant généralement difficiles à expliquer en raison de leurs cascades de fonctions internes, trop complexes pour être comprises d'un point de vue "humain". Néanmoins, en remarquant que le détecteur de Bayes $\delta^*(\mathbf{x})$ (5.12) peut être réécrit comme une décision basée sur la sigmoïde $\delta^*(\mathbf{x}) = \mathbb{1}_{\{\rho_s(\Lambda(\mathbf{x})) \geq 0.5\}}$, nous développons une classe d'architectures bayésiennes explicables \mathcal{F}^* qui peut approcher avec précision l'estimateur $\Lambda^*(\mathbf{x})$ pour différents niveaux de bruit/SNRs. Nous proposons ainsi d'approximer $\delta^*(\mathbf{x})$, par le biais d'un modèle d'apprentissage profond $\hat{\delta}_\theta(\mathbf{x}) = \mathbb{1}_{\{\rho_s(f_\theta(\mathbf{x})) \geq 0.5\}} \in \mathcal{F}^*$ où $f_\theta : \mathbb{R}^K \rightarrow [0, 1]$ et $\theta \in \mathbb{R}^{d_\theta}$ dénote l'ensemble des paramètres entraînaibles. Afin d'être explicables, les réseaux profonds dans \mathcal{F}^* devront imiter la structure mathématique de $\Lambda^*(\mathbf{x})$ dans l'équation (5.12), en permettant d'approximer chacune de ses fonctions et opérations internes séparément et avec précision, grâce à des modules de réseau de neurones spécialement conçus pour différents types d'opérations. Cette façon de procéder peut être vue comme une forme de multi-task learning comme le stipulent Zhang & Yang (2021), puisque nous demandons à l'algorithme d'effectuer une tâche de processing en même temps que l'apprentissage du classifieur.

Dès lors que l'on considère (5.12), il est évident qu'un classifieur linéaire, tel qu'un simple perceptron, reproduit la structure de l'estimateur optimal à condition que $x^2(k)$ soit l'entrée du classifieur linéaire. Cependant, si nous donnons uniquement en entrée de ce réseau le signal brut \mathbf{x} , nous ne pouvons pas nous attendre à obtenir un bon classifieur. Par conséquent, pour obtenir un apprentissage réussi avec un réseau profond $f_\theta(\mathbf{x})$ dans \mathcal{F}^* , il est nécessaire dans notre cas de pouvoir apprendre des coefficients $\varphi_1, \dots, \varphi_k, \dots, \varphi_K$, en plus d'un biais φ_0 , ainsi qu'un ensemble de poids $\gamma \in \mathbb{R}^{d_\gamma}$ définissant un réseau neuronal profond de la forme :

$$f_\theta(\mathbf{x}) = \sum_{k=1}^K \varphi_k \cdot \Phi_{p_2, \gamma}(x(k)) + \varphi_0, \quad (5.28)$$

où $\Phi_{p_2, \gamma} : \mathbb{R} \mapsto \mathbb{R}$ et un réseau de neurones de paramètre γ , dédié à l'approximation de la fonction carré $p_2(t) = t^2$ présente dans $\Lambda^*(\mathbf{x})$. Les paramètres φ_k jouent le rôle des paramètres b_k , dans l'équation (5.12). Le biais φ_0 joue un rôle important : modifier sa valeur équivaut à modifier le seuil τ du classifieur Bayes.

La sous-section suivante détaillera le module de réseau de neurones d'approximation de fonction $\Phi_{g, \gamma}$ présent dans $f_\theta(\mathbf{x})$ et destiné à remplacer les couches classiques entièrement connectées et convolutives souvent trouvées dans les modèles d'apprentissage profond et communément utilisées comme des boîtes noires.

3.2 Module de réseau de neurones d'approximation de fonction

Dans cette sous-section nous proposons de décrire un module d'architecture de réseau de neurones dénoté $\Phi_{g, \gamma}$ et conçu pour approximer une fonction $g : \mathbb{R} \mapsto \mathbb{R}$ par le biais d'un jeu de paramètres γ . Le module de réseau $\Phi_{g, \gamma}$ est composé de L couches cachées avec exactement W neurones par couche. Pour une valeur $t \in \mathbb{R}$, $\Phi_{g, \gamma}(t) = \Phi^{(L)}$ est généré par la séquence de

couches suivante :

$$\begin{cases} \Phi^{(0)} &= \rho_r \left(\gamma^{(0)} t + \mathbf{b}^{(0)} \right), \\ \Phi^{(i)} &= \rho_r \left(\gamma^{(i)} \Phi^{(i-1)} + \mathbf{b}^{(i)} \right), \quad 1 \leq i < L, \\ \Phi^{(L)} &= \gamma^{(L)} \Phi^{(L-1)} + \mathbf{b}^{(L)}, \end{cases} \quad (5.29)$$

où $\Phi^{(i)}$ est la $i^{\text{ème}}$ couche de sortie, $\rho_r(t) = \max\{0, t\}$ est la fonction d'activation ReLU, et $\gamma = \left(\gamma^{(0)}, \dots, \gamma^{(L)}, \mathbf{b}^{(0)}, \dots, \mathbf{b}^{(L)} \right)$ est le jeu de paramètres du réseau. Tous les $\gamma^{(i)}$ sont des matrices dans $\mathbb{R}^{W \times W}$, hormis pour la couche d'entrée $\gamma^{(0)} \in \mathbb{R}^{W \times 1}$ et la couche de sortie $\gamma^{(L)} \in \mathbb{R}^{1 \times W}$. Les biais $\mathbf{b}^{(i)}$ appartiennent tous à $\mathbb{R}^{W \times 1}$ excepté $\mathbf{b}^{(L)} \in \mathbb{R}$.

Même si le vecteur d'entrée \mathbf{x} peut théoriquement prendre n'importe quelle valeur dans \mathbb{R}^K , sa masse de probabilité est concentrée dans un sous-espace compact de \mathbb{R}^K noté \mathcal{X} . Ainsi, en acceptant une très faible perte de performance, on peut supposer que la fonction g , qui peut être la fonction carré ou bien exponentielle, est uniquement définie sur un intervalle compact $I = [-D, D]$ où D est connu. Les erreurs commises pour les signaux d'entrée $\mathbf{x} \notin \mathcal{X}$ sont donc supposées négligeables. En effet, comme on peut supposer que les quantités aléatoires et en particulier les échantillons d'entrée $x(k)$ appartiendront à I avec une probabilité élevée, il suffit d'approximer une telle fonction sur l'intervalle I . Nous considérons que la fonction g considérée et restreinte à I appartient à un espace très lisse/régulier \mathcal{S}_D défini par

$$\mathcal{S}_D = \left\{ g \in C^\infty(I, \mathbb{R}) : \left\| g^{(k)} \right\|_{L^\infty(I)} \leq k!, \forall k \in \mathbb{N} \right\}, \quad (5.30)$$

où $C^\infty(I, \mathbb{R})$ est l'espace de fonction infiniment dérivable sur I , $g^{(k)}$ est la $k^{\text{ème}}$ dérivée de g , \mathbb{N} est l'ensemble des entiers non négatifs et $\|f\|_{L^\infty(I)} := \sup_{x \in I} \{|f(x)|\}$.

La théorie sur l'approximation en apprentissage profond, notamment destinée à caractériser une erreur minimale si aucune contrainte n'est présente sur l'algorithme d'apprentissage, a montré depuis les travaux de [Cybenko \(1989\)](#), que les réseaux de neurones profonds peuvent approximer n'importe quelle fonction lisse. En particulier, il est prouvé dans le lemme ([Elbrächter et al. 2019](#), Lemma A.6) que, étant donné une erreur $\epsilon \in (0, \frac{1}{2})$, il existe une constante $C > 0$ telle que, pour toute fonction $g \in \mathcal{S}_D$, il existe un réseau de neurones $\Phi_{g, \gamma} : \mathbb{R} \rightarrow \mathbb{R}$ avec $W \leq 9$ neurones par couche et $L \leq C \log_2^2(1/\epsilon) + \log_2(1 + 1/D)$ couches, satisfaisant :

$$\|\Phi_{g, \gamma} - g\|_{L^\infty(I)} \leq \epsilon. \quad (5.31)$$

Cette borne est notamment obtenue par le biais de constructions en "dents de scie", présentés par [Telgarsky \(2015\)](#), qui exploitent le fait qu'un réseau de neurones de $2n$ couches peut définir une fonction disposant de 2^n formes de dents avec au maximum 2 neurones par couche. Par conséquent, comme le démontrent [Elbrächter et al. \(2019\)](#), en se basant sur le fait que le nombre de régions linéaires dans une construction en dents de scie augmente à un taux exponentiel optimal avec la profondeur du réseau, nous avons la garantie que le module de réseau de neurones $\Phi_{g, \gamma}$ peut approcher avec précision une fonction interne g qui serait présente dans la fonction de décision de Bayes optimale $\Lambda^*(\mathbf{x})$.

Remarque

Cette borne théorique peut cependant ne pas être vérifiée durant l'entraînement d'un réseau de neurones. En effet, dans une publication réalisée pour observer l'évolution du nombre de régions linéaires apprises par des modèles pendant l'entraînement, [Hanin & Rolnick \(2019\)](#) n'ont pas réussi à faire ressortir une croissance exponentielle avec la profondeur du réseau comme pourrait le faire espérer les bornes théoriques. Pour proposer une explication, [Hanin & Rolnick \(2019\)](#) dénotent que de légères perturbations des poids et biais d'un réseau configuré pour approcher un nombre exponentiel de régions, ont généralement pour conséquence de grandement réduire le nombre de régions linéaires obtenu avec ce type de construction.

Nous présentons maintenant comment implémenter la fonction définie par le réseau $\Phi_{g,\gamma}$ afin qu'elle puisse être appliquée à un vecteur plutôt qu'un réel. Étant donné le réseau $x \in \mathbb{R} \mapsto \Phi_{g,\gamma}(x)$ présenté dans l'équation (5.29), nous définissons le vecteur

$$\widehat{\Phi}_{g,\gamma}(\mathbf{x}) = [\Phi_{g,\gamma}(x(1)), \dots, \Phi_{g,\gamma}(x(K))], \quad (5.32)$$

qui applique la fonction $\Phi_{g,\gamma}(\cdot)$ élément par élément aux composantes d'un vecteur \mathbf{x} de longueur K .

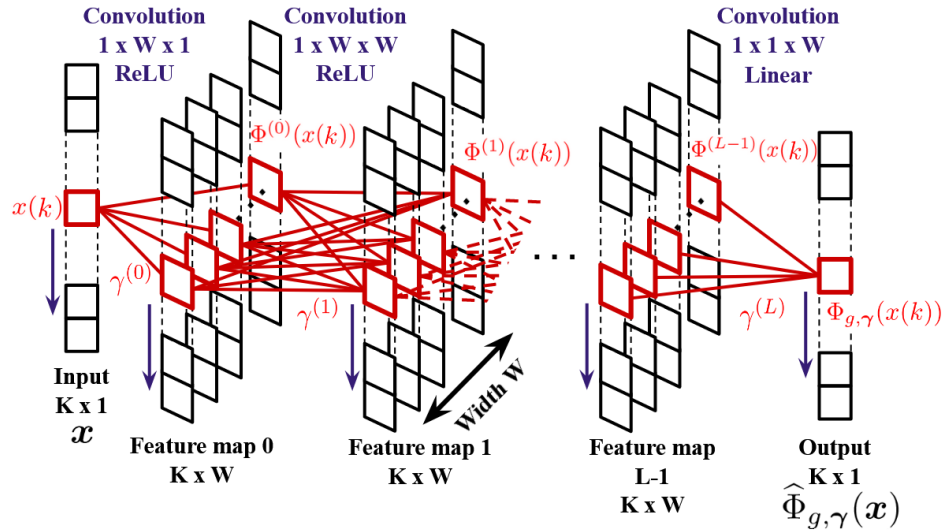


Figure 5.1: Architecture du module CNN d'approximation de fonction $\widehat{\Phi}_{g,\gamma} : \mathbb{R}^K \rightarrow \mathbb{R}^K$ avec un ensemble de noyaux de convolution équivalant à la fonction $\Phi_{g,\gamma} : \mathbb{R} \rightarrow \mathbb{R}$ distincte en rouge.

En reprenant l'équation (5.28), nous devons calculer $\widehat{\Phi}_{p_2,\gamma}(\mathbf{x})$. Dans ce but, nous implémentons le calcul du vecteur $\widehat{\Phi}_{g,\gamma}$ en exploitant l'invariance naturelle aux translations des couches convolutives d'un CNN, afin de limiter le nombre de paramètres de l'architecture profonde f_θ , mais également d'être efficace en termes de calcul. La $i^{\text{ème}}$ couche de $\Phi_{g,\gamma}(x)$ est $\Phi_k^{(i)} =$

$\rho_r \left(\mathbf{z}_k^{(i)} \right)$ où $\mathbf{z}_k^{(i)} = \boldsymbol{\gamma}^{(i)} \Phi_k^{(i-1)} + \mathbf{b}^{(i)}$ et d'indice k , également présent dans $\Phi_k^{(i)}$, souligne que le vecteur dépend d'une entrée spécifique $x(k)$. Pour calculer $\widehat{\Phi}_{g,\boldsymbol{\gamma}}(\mathbf{x})$, nous devons calculer K fois la transformation affine $\mathbf{z}_k^{(i)}$ pour $k \in \llbracket K \rrbracket$. Un moyen efficace de faire tous ces calculs est d'utiliser une convolution. Pour prouver l'équivalence entre la transformation affine et la convolution, on suppose que $\boldsymbol{\gamma}^{(i)} \in \mathbb{R}^{H \times S}$, $\Phi_k^{(i-1)} \in \mathbb{R}^S$ et $\mathbf{b}^{(i)} \in \mathbb{R}^H$ où $H, S \in \{1, W\}$ dans notre cas. Un ensemble de K vecteurs $\Phi_k^{(i-1)} \in \mathbb{R}^S$ est vu comme un tenseur d'ordre 2 $\Phi^{(i-1)}$ dans $\mathbb{R}^{S \times K}$ tel que $\Phi^{(i-1)}(s, k)$ est le $s^{\text{ème}}$ composant de $\Phi_k^{(i-1)}$. De même, l'ensemble des K vecteurs $\mathbf{z}_k^{(i)} \in \mathbb{R}^H$ est noté $Z^{(i)} \in \mathbb{R}^{H \times K}$ tel que $Z^{(i)}(h, k) = \mathbf{z}_k^{(i)}(h)$. La convolution $Z^{(i)} = \Gamma^{(i)} * \Phi^{(i-1)}$ est définie comme le font [Goodfellow et al. \(2016\)](#) :

$$Z^{(i)}(h, k) = \mathbf{b}^{(i)}(h) + \sum_{s=1}^S \sum_{j=1}^K \Gamma^{(i)}(h, s, j) \cdot \Phi^{(i-1)}(s, k+j), \quad (5.33)$$

où $\Gamma^{(i)}(h, s, j)$ est un tenseur d'ordre 3 tel que $\Gamma^{(i)}(h, s, 1) = \boldsymbol{\gamma}^{(i)}(h, s)$ et $\Gamma^{(i)}(h, s, j) = 0$ pour $j > 1$. Pour simplifier les notations, on supposera que $\Phi^{(i-1)}(s, k+j) = 0$ lorsque $k+j > K$. Par conséquent, le calcul de la convolution dans (5.33) permet d'obtenir les K vecteurs $\mathbf{z}_k^{(i)}$. En se basant sur cette équivalence, toutes les couches entièrement connectées apparaissant dans l'équation (5.29) sont implémentées sous la forme d'une séquence de convolutions. L'architecture convolutive complète est représentée sur la figure 5.1. La figure précise la dimension des feature maps et la taille des noyaux de convolution. Le premier ensemble de feature maps est produit par une couche convolutive de noyau $\boldsymbol{\gamma}^{(0)}$, les $L-1$ suivants sont produits par les matrices $\boldsymbol{\gamma}^{(i)}$. Enfin, le dernier ensemble de feature maps, qui ne dispose que d'un canal et donc une seul feature map, est produit par $\boldsymbol{\gamma}^{(L)}$. Un tel CNN permet ainsi de conserver la structure de calcul et l'expressivité du réseau de neurones (5.29) avec une structure convolutive équivalente.

3.3 Architecture CNN

Cette sous-section montre que notre fonction $f_\theta \in \mathcal{F}^*$ peut être facilement implémentée comme un CNN avec des bibliothèques d'apprentissage profond Python telles que Keras ou Pytorch. Notre approximation de $\Lambda^*(\mathbf{x})$ avec $f_\theta(\mathbf{x})$, décrite dans la figure 5.2, est basée sur la composition du module $\Phi_{g,\boldsymbol{\gamma}}$ ainsi qu'une somme pondérée des échantillons temporels $\Phi_{p_2,\boldsymbol{\gamma}}(x(k))$.

Le réseau de neurones $\Phi_{p_2,\boldsymbol{\gamma}}$ est implémenté en tant qu'un CNN $\widehat{\Phi}_{p_2,\boldsymbol{\gamma}}$ grâce au module présenté dans la figure 5.1 et ainsi, les L premières couches convolutives de la figure 5.2 calculent $\widehat{\Phi}_{p_2,\boldsymbol{\gamma}}(\mathbf{x})$. La couche "flatten" suivante produit un vecteur propre de taille $K \times 1$. Enfin, la couche dense finale nommée φ_k , calcule une somme linéaire des coefficients $\Phi_{p_2,\boldsymbol{\gamma}}(x(k))$ et le biais φ_0 pour obtenir la sortie du réseau $f_\theta(\mathbf{x})$ utilisée pour choisir entre les hypothèses H_0 et H_1 . Après un apprentissage approprié, l'extraction des noyaux convolutifs estimés par le module CNN $\Phi_{p_2,\boldsymbol{\gamma}}$, nous donne l'approximation des fonctions carrées apprises, rendant ainsi notre architecture pleinement explicable.

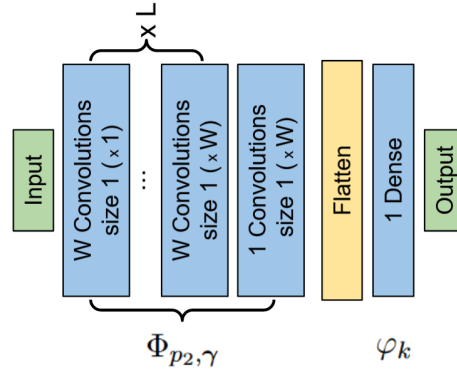


Figure 5.2: Architecture de $f_{\theta}(\mathbf{x})$ avec la terminologie de la librairie Keras.

3.4 Méthodologie pour les expérimentations

Puisque nous nous intéressons à la détection d'événements acoustiques, nous souhaitons comparer les courbes ROC de notre architecture profonde bayésienne sur signaux acoustiques bruts avec celles du détecteur optimal $\delta^*(\mathbf{x})$ ainsi que d'une architecture profonde de type CNN 1D telle que le FCNN M11 disposant de 10 couches de convolution et présenté par Dai et al. (2017). Aussi, après avoir entraîné les modèles sur l'ensemble d'apprentissage \mathcal{S} contenant jusqu'à $N_{max} = 1 \cdot 10^5$ signaux, nous tracerons les courbes ROC avec un deuxième ensemble de test indépendant \mathcal{S}_{test} . Un troisième ensemble de validation également indépendant, \mathcal{S}_{val} sera également utilisé pour déterminer quand arrêter l'entraînement dans la procédure d'optimisation des modèles. \mathcal{S}_{test} et \mathcal{S}_{val} contiendront $2 \cdot 10^4$ échantillons. Les signaux dureront deux secondes et seront échantillonnés à 2200 Hz pour un total de $K = 4400$ échantillons temporels. Pour l'hypothèse H_1 les signaux générés seront un mix de bruit ambiant blanc et de réalisations d'une même hélice de navire à $p = 4$ pales, avec une vitesse de rotation de $\omega = 100$ tours minutes. Pour l'hypothèse H_0 , les signaux générés seront simplement du bruit ambiant blanc.

Le tracé des courbes ROC se fera sur plusieurs SNRs liés à différents ensembles d'entraînement, de test et de validation, tous générés avec des niveaux de bruits ambiants blancs différents. Dans tous ces cas de SNRs, au lieu d'utiliser une méthode de Monte-Carlo, la courbe ROC optimale sera obtenue en utilisant une approximation assez précise (puisque K est grand) de la distribution de $\Lambda^*(\mathbf{x})$ obtenue à partir du théorème central limite dans la sous-section 2.3.

Remarque

Par la suite nous prendrons comme définition du SNR

$$\text{SNR}_{\text{dB}}(\mathbf{x}, \mathbf{n}_a) = 20 \log_{10} \left(\frac{P(\mathbf{x}) - P(\mathbf{n}_a)}{P(\mathbf{n}_a)} \right), \quad (5.34)$$

avec $P(\mathbf{x}) = \sum_{k=1}^K x^2(k)/K$ et $P(\mathbf{n}_a) = \sum_{k=1}^K n_a^2(k)/K$.

3.5 Optimisation des modèles profonds

Pour définir notre architecture d'approximation $\widehat{\delta}_\theta$, nous utilisons l'équivalent de $W = 9$ neurones par couche avec $W = 9$ convolutions de taille 1 pour le module d'approximation de fonction $\Phi_{p_2, \gamma}$. De plus, selon l'équation (5.31), si nous voulons que notre fonction $\Phi_{p_2, \gamma}$ approxime p_2 avec une erreur inférieure à $\epsilon = 10^{-p}$, le nombre de couches L doit être supérieur à $Cp \log_2^2(10) + \log_2(1 + 1/D)$, avec C une constante strictement positive et $D = \max_{\mathbf{x} \in \mathcal{S}} x^2(k)$. Aussi, puisque la constante C est une inconnue, le nombre de couches de $\Phi_{p_2, \gamma}$ devra être déterminé empiriquement, ce que nous ferons grâce à une "GridSearch" sur la profondeur de couche $L = \{1, 3, 6\}$ avec la fonction de perte de validation comme métrique. Notre réseau de neurones profond est comparé à l'architecture FCNN M11 tirée des travaux de Dai et al. (2017) et décrite dans le tableau 8.2.

Notre architecture bayésienne et le FCNN de comparaison suivent la même procédure d'optimisation lors de l'apprentissage, avec l'utilisation de l'optimiseur Adam et de l'entropie croisée binaire comme fonction de perte. Le critère d'arrêt est une absence d'amélioration de la perte de validation pendant 15 époques consécutives, ou au maximum après 100 époques d'entraînement. La contrainte "Maximum kernel" est utilisée pour toutes les couches convolutives afin de majorer les valeurs des poids des filtres convolutifs et nous appliquons une régularisation L^2 (Ridge) aux couches de sortie. De même, nous utilisons également la normalisation par batch pour toutes les couches convolutives afin de réduire l'effet de vanishing gradient et réduire les chances que le gradient explose lors de l'entraînement. Pour chaque courbe ROC considérée dans la figure 5.3, les performances des modèles sont estimées grâce à un test de Neyman-Pearson en calculant la moyenne et l'écart type des erreurs d'entraînement et de test parmi 10 entraînements différents.

Remarque

Pour un modèle f_θ donné, chaque point de sa courbe ROC est obtenu en calculant, pour l'ensemble des entrées \mathbf{x} issu d'un jeu de données \mathcal{S} , les taux de fausse alarme et de vrai positif de la fonction de décision $\widehat{\delta}_{\theta, \tau(m)}(\mathbf{x})$ associé au seuil $\tau(m)$:

$$\widehat{\delta}_{\theta, \tau}(\mathbf{x}) = \mathbb{1}_{\{\rho_s(f_\theta(\mathbf{x})) \geq \tau(m)\}} = \begin{cases} 0 & \text{si } f_\theta(\mathbf{x}) < \tau(m), \\ 1 & \text{si } f_\theta(\mathbf{x}) \geq \tau(m), \end{cases} \quad (5.35)$$

avec $\tau(m)$ un seuil du vecteur $\boldsymbol{\tau} = [\tau(1), \dots, \tau(M)]$ tel que $\tau(1) = \min_{\mathbf{x} \in \mathcal{S}}(f_\theta(\mathbf{x}))$, $\tau(M) = \max_{\mathbf{x} \in \mathcal{S}}(f_\theta(\mathbf{x}))$ et $\boldsymbol{\tau}$ est obtenu avec la fonction Numpy de Python :

$$\boldsymbol{\tau} = \text{logscale}(\tau(1), \tau(M), M). \quad (5.36)$$

3.6 Résultats numériques

Sur la figure 5.3, on constate que nos modèles d'approximation f_θ et les modèles d'architecture M11, tous entraînés sur un jeu de $N = N_{max}$ données, obtiennent des performances quasiment optimales pour les SNRs les plus élevées. Nos modèles conservent cependant de bonnes

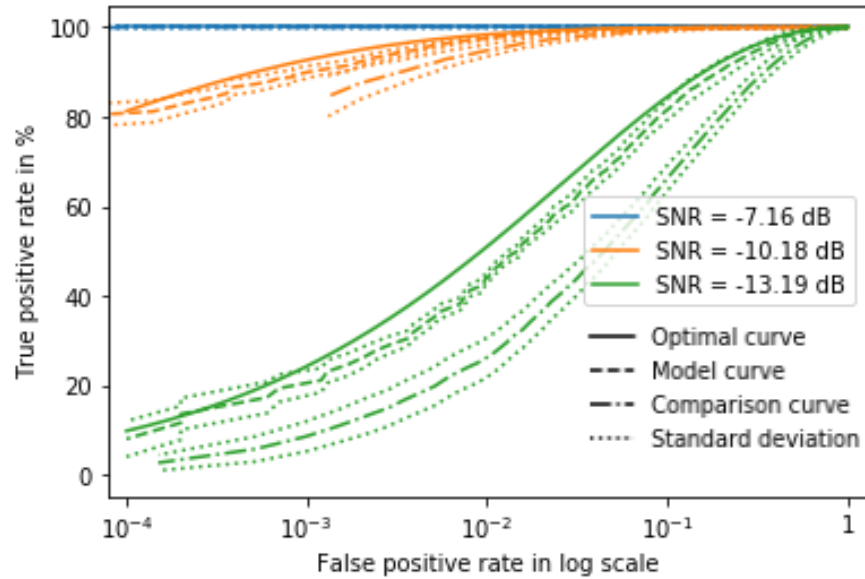


Figure 5.3: Courbes ROCs sur le jeu de test de l'architecture profonde bayésienne proposée (courbes en tirets), du test de Neyman-Pearson du détecteur optimal (courbes continues) et du classifieur de comparaison M11 (courbes en points et tirets).

précisions de test pour les ensembles d'entraînement plus difficiles à SNRs plus faibles, tandis que les modèles de comparaison M11 contenant 300 fois plus de paramètres, s'éloignent de plus en plus nettement des performances du classifieur de Bayes optimal.

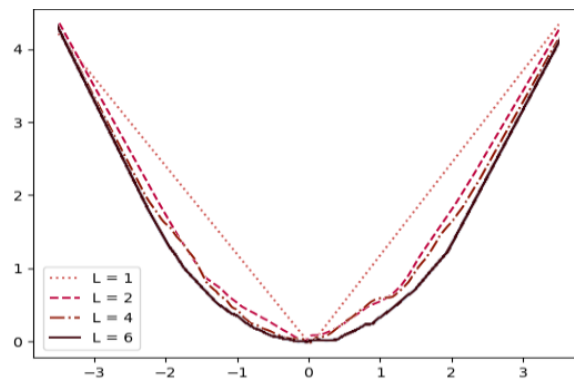


Figure 5.4: Exemples de fonctions $\Phi_{p_2,\gamma}$ apprises avec différentes profondeurs L .

Dans la Fig. 5.4, nous affichons, pour différentes valeurs L , la fonction apprise par le module d'approximation $\Phi_{p_2,\gamma}(t)$, pour $t \in [-3, 3]$. Pour obtenir cette figure, nous retirons la dernière couche de f_θ dans la figure 5.2 et traçons les prédictions de l'architecture résultante pour les valeurs $t \in [-3, 3]$. Nous observons ainsi que pour un nombre suffisant de couche L , notre modèle parvient à apprendre la fonction p_2 , présente dans le détecteur optimal, avec une ap-

proximation relativement lisse et ce, malgré une initialisation aléatoire de ses paramètres.

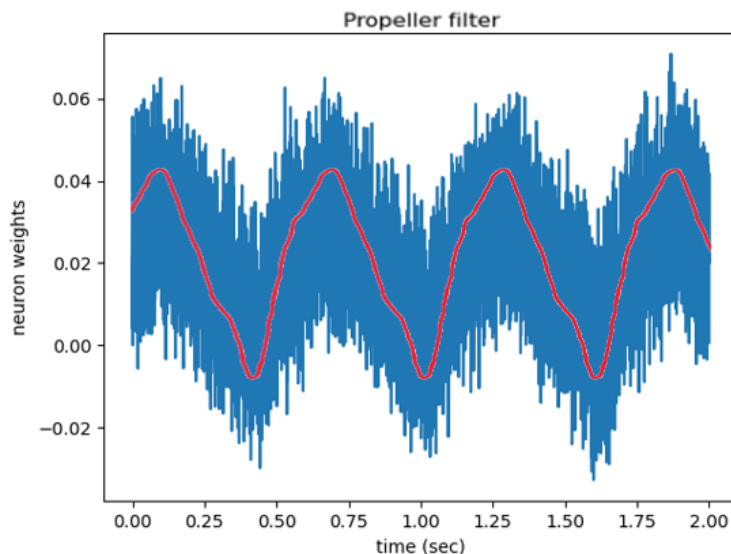


Figure 5.5: Exemple de filtre appris φ_k , en bleu, comparé à la forme du filtre optimal b_k , en rouge.

Dans la figure 5.5, nous affichons le filtre φ_k appris par la dernière couche dense de f_θ pour le SNR -10.18 dB et la comparons à un filtre d'expression $a \cdot b_k + c$, avec b_k le filtre optimal et a ainsi que c deux réels choisis empiriquement pour correspondre à la forme générale de φ_k sur la figure. On constate ainsi que malgré les performances quasiment optimales de nos modèles, les filtres appris sont relativement bruités, ce qui est un résultat assez difficile à comparer étant donné que peu de travaux concernent la classification de signaux bruités, où les signaux eux-mêmes sont du bruit gaussien.

3.7 Résultats sur la généralisation

Dans la sous-section précédente, nous avons évalué la qualité de notre entraînement pour une taille d'ensemble d'entraînement \mathcal{S} fixée, avec l'excès de risque $\mathcal{R}(f_\theta) - \mathcal{R}^*$, i.e. la convergence de notre modèle $f_\theta(\mathbf{x})$ vers $\delta(\mathbf{x})^*$ pour la fonction de perte idéale "0-1". Bien qu'il ne soit pas possible de donner de garanties d'optimisation concernant la convergence de notre modèle à cause de la non-convexité du problème d'optimisation, nous avons cependant pu affecter le comportement de l'erreur d'approximation $\mathcal{R}(f_{\theta^*}) - \mathcal{R}^*$, avec le paramètres L du réseau comme nous l'avons vu dans la sous-section 3.2. Néanmoins, en permettant au nombre de couches convolutives L de croître, nous améliorons la capacité d'approximation de notre modèle au prix d'un plus grand nombre de poids entraînaables dans notre architecture. En particulier, [Zhou & Feng \(2018\)](#), ont montré que l'erreur de généralisation d'un CNN $\mathcal{R}_N(f_\theta) - \mathcal{R}(f_\theta)$ ayant pour fonction de perte l'erreur moyenne quadratique :

$$\mathcal{L}_q(z, y) = \frac{1}{2} \cdot \|z - y\|_2^2, \quad (5.37)$$

croît linéairement avec sa profondeur L et diminue à un taux $\mathcal{O}(\frac{1}{\sqrt{N}})$ (à un facteur logarithmique près), bien que les conditions de validité de cette borne ne soient pas présentes dans notre entraînement.

Le nombre de couches L du module d'approximation de fonction ayant été déterminé par "GridSearch", il reste cependant à analyser le comportement de généralisation de notre architecture f_θ en fonction du nombre d'exemples d'entraînement N présents dans le jeu de données \mathcal{S} . Cette sous-section présente ainsi la différence entre la précision obtenue par notre modèle f_θ sur un jeu de données d'entraînement \mathcal{S} et de test \mathcal{S}_{test} et compare ses performances à un réseau de neurones ne possédant qu'une couche de sortie, équivalant à un perceptron, et à une méthode des moindres carrés tous deux prenant le carré des signaux reçus \mathbf{x}^2 en entrée pour l'apprentissage et la prédiction. Nous effectuerons également différents entraînements de f_θ avec des contraintes ou formes de régularisation différentes pendant l'apprentissage afin de voir leur effet sur l'erreur de généralisation.

Par la suite on considérera qu'un apprentissage avec une régularisation L^2 consistera toujours à minimiser le risque empirique :

$$\mathcal{R}_N(f_\theta) := \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f_\theta(\mathbf{x}^{(i)}), y^{(i)}), \quad (5.38)$$

mais avec une fonction de perte \mathcal{L} modifiée pour faire apparaître un terme lié à la pénalité de régularisation, ce qui donne pour l'entropie croisée binaire :

$$\mathcal{L}(z, y) = y \log(z) + (1 - y) \log(1 - z) + l_2 \cdot \|\mathbf{w}\|_2^2, \quad (5.39)$$

pour tout $0 \leq z, y \leq 1$ et avec l_2 le facteur de régularisation L^2 .

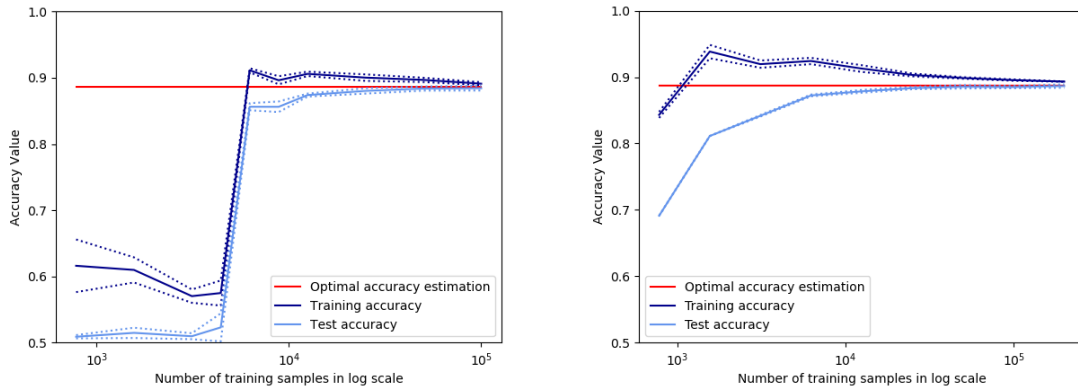


Figure 5.6: Précision d'entraînement et de test de l'architecture f_θ sans aucune forme de régularisation, à gauche, en fonction de la taille du jeu de données et comparaison à un perceptron prenant \mathbf{x}^2 en entrée à droite. La précision optimale correspond à un test de Monte-Carlo avec le détecteur donné dans l'équation (5.12).

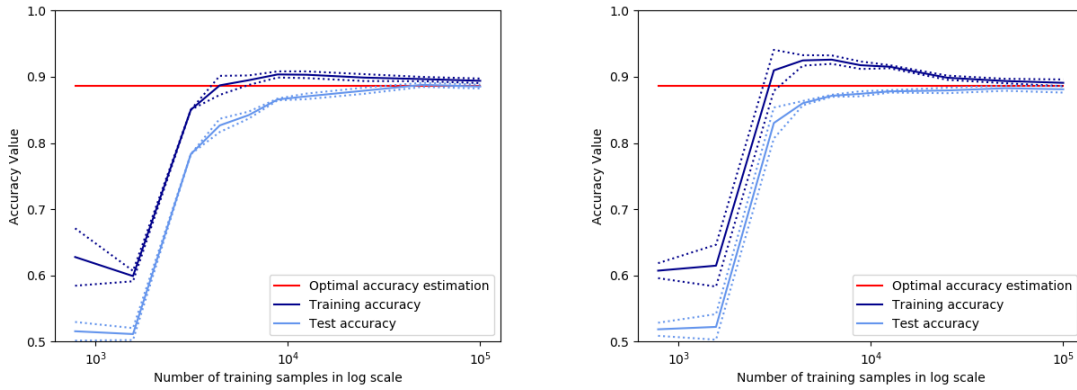


Figure 5.7: Précision d'entraînement et de test de l'architecture f_θ avec uniquement une contrainte "Maximum kernel" à gauche, et avec uniquement une régularisation L^2 à droite, en fonction de la taille du jeu de données.

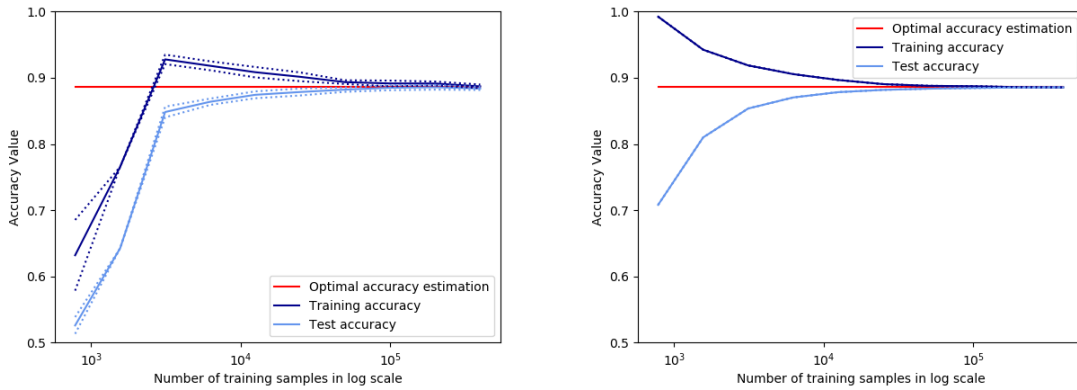


Figure 5.8: Précision d'entraînement et de test de l'architecture f_θ avec les deux formes de régularisation précédentes à gauche, en fonction de la taille du jeu de données et comparaison à une méthode des moindres carrés avec x^2 en entrée, à droite.

La contrainte "Maximum kernel" consistera simplement à contraindre les poids d'une même couche cachée, ou des filtres d'une même couche convolutive, à avoir une norme inférieure ou égale à une valeur c_{max} souhaitée après chaque époque de l'entraînement.

En comparant les performances de notre architecture sans régularisation, à gauche sur la figure 5.6, et avec une forme de régularisation sur la figure 5.7, nous pouvons constater que ces deux formes de régularisation permettent d'obtenir de meilleures précisions avec moins d'exemples d'entraînement, mais sans nettement réduire l'erreur de généralisation à nombre d'exemple d'entraînement égal. Sur la figure 5.8, les performances de notre architecture f_θ ,

avec ces deux formes de régularisation, sont cependant nettement meilleures. On remarquera qu'elles permettent à la fois d'obtenir une meilleure précision plus rapidement et réduisent l'erreur de généralisation de notre modèle pour qu'elle s'approche plus de l'erreur de généralisation obtenue avec la méthode des moindres carrés quand N est grand, plutôt que celle avec un perceptron où, dans les deux cas, l'apprentissage de la fonction p_2 est absent.

Il aurait également été intéressant de comparer un MLP prenant en entrée x, x^2, \dots , jusqu'à x^p , avec l'utilisation de régularisation L^1 ou "elastic Net" pour la sélection des variables adéquates.

4 Détection de bruit d'hélice coloré

4.1 Énoncé du problème

Nous considérons ici le problème de détection de bruit d'hélice de la section 2.3 du chapitre 4, en assumant que les bruits de cavitation $c_{i,\omega}(t)$ et le bruit ambiant $n_a(t)$, sont des bruits gaussiens colorés. Nous supposons également qu'il n'y a qu'une seule signature de bruit d'hélice et que ses motifs $m_{i,\omega}(t)$ sont connus. Notre problème de détection consiste ainsi encore une fois à décider si le signal discret \mathbf{x} contient uniquement du bruit ambiant (hypothèse H_0) ou s'il contient un signal de navire (4.2) (hypothèse H_1) :

$$\begin{aligned} H_0 &: \{x(k) = n_a(k), k \in \llbracket K \rrbracket\}, \\ H_1 &: \{x(k) = s_\eta(k) + n_a(k), k \in \llbracket K \rrbracket\}. \end{aligned} \quad (5.40)$$

Dans l'hypothèse sans sources H_0 , $x(k) = (e_a * h_m)(k)$, avec e_a un bruit gaussien de K échantillons et $h_m(k)$ un filtre discret de taille K mais ne disposant que de p échantillons non nuls : $\mathbf{h}_m = [h_m(1), \dots, h_m(p), 0, \dots, 0]$. Le signal reçu \mathbf{x} suit alors une distribution gaussienne multivariée de moyenne nulle et de matrice de covariance Σ_0 pouvant être exprimée en fonction de la matrice circulante de \mathbf{h}_m notée $C(\mathbf{h}_m)$, ainsi que de la matrice de covariance du bruit gaussien e_a qui est $\sigma_a^2 \cdot I_K$. On obtient ainsi

$$\Sigma_0 = \sigma_a^2 \cdot C(\mathbf{h}_m)^T I_K C(\mathbf{h}_m) = \sigma_a^2 \cdot C(\mathbf{h}_m)^T C(\mathbf{h}_m). \quad (5.41)$$

De façon plus calculatoire, on a :

$$\begin{aligned} \Sigma_0(k, k') &= \mathbb{E}[x(k) \cdot x(k')] \\ &= \mathbb{E}[(e_a * h_m)(k) \cdot (e_a * h_m)(k')] \\ &= \mathbb{E} \left[\sum_{u=1}^p h_m(u) e_a(k-u) \cdot \sum_{l=1}^p h_m(l) e_a(k'-l) \right] \quad \text{mod } K \\ &= \sum_{u=1}^p h_m(u) h_m(u+k'-k) \cdot \mathbb{E} [e_a^2(k-u)] \quad \text{mod } K \\ &= \sigma_a^2 \cdot \sum_{u=1}^p h_m(u) h_m(u+\delta k) \quad \text{mod } K \\ &= \sigma_a^2 \cdot \Gamma_{h_m}(\delta k), \end{aligned} \quad (5.42)$$

avec $\Gamma_{h_m}(0) = |h_m^2|$ et $\Gamma_{h_m}(\delta k) = 0$ pour $\delta k \geq p$ et $\delta k = |k' - k|$. La matrice de covariance de \mathbf{x} est ainsi une matrice de Toeplitz symétrique Σ_0 :

$$\Sigma_0 = \sigma_a^2 \cdot H_m, \quad (5.43)$$

$$H_m = \begin{bmatrix} \Gamma_{h_m}(0) & \Gamma_{h_m}(1) & \cdots & \Gamma_{h_m}(K-1) \\ \Gamma_{h_m}(1) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \Gamma_{h_m}(1) \\ \Gamma_{h_m}(K-1) & \cdots & \Gamma_{h_m}(1) & \Gamma_{h_m}(0) \end{bmatrix}. \quad (5.44)$$

On remarquera que H_m est Toeplitz symétrique de bande passante $p-1$ et ainsi diagonalisable. Dans l'hypothèse H_1 , puisque le bruit d'hélice $s_\eta(k)$ et le bruit ambiant $n_a(k)$ sont indépendants, les changements de puissance perçus à travers le signal discret reçu $x(k)$ proviennent uniquement des différentes modulations d'amplitude effectuées par chacune des pales :

$$x(k) \sim \mathcal{N}(0, |h_\omega^2| \cdot \sigma_\eta^2(k) + |h_m^2| \cdot \sigma_a^2), \quad (5.45)$$

$$\sigma_\eta^2(k) = \sum_{i=1}^p m_{i,\omega}^2(k) + 1, \quad (5.46)$$

où σ_a est l'écart types du bruit ambiant $n_a(k)$. Dans cette hypothèse \mathbf{x} suit également une distribution gaussienne multivariée $\mathcal{N}(0, \Sigma_1)$ toujours de moyenne nulle et, par indépendance, ayant pour matrice de covariance :

$$\Sigma_1 = \Sigma_0 + \sum_{i=1}^p \mathbf{m}_{i,\omega} H_\omega \mathbf{m}_{i,\omega} + H_\omega, \quad (5.47)$$

$$\mathbf{m}_{i,\omega} = \text{diag}[m_{i,\omega}(1), m_{i,\omega}(2), \dots, m_{i,\omega}(K)], \quad (5.48)$$

$$H_\omega = \begin{bmatrix} \Gamma_{h_\omega}(0) & \Gamma_{h_\omega}(1) & \cdots & \Gamma_{h_\omega}(K-1) \\ \Gamma_{h_\omega}(1) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \Gamma_{h_\omega}(1) \\ \Gamma_{h_\omega}(K-1) & \cdots & \Gamma_{h_\omega}(1) & \Gamma_{h_\omega}(0) \end{bmatrix}. \quad (5.49)$$

Cette matrice de covariance est encore une fois symétrique définie positive, et donc inversible, mais cela au coût de $\mathcal{O}(K^2)$ opérations.

4.2 Règle de détection bayésienne optimale

En utilisant l'expression de la fonction de distribution de probabilité conditionnelle sachant y de $\mathcal{D}(\mathbf{x}, y)$, qui est celle d'une distribution gaussienne multivariée

$$\Pr_{\mathbf{x}}(\mathbf{x}) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \cdot e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)}, \quad (5.50)$$

on peut déduire le rapport de vraisemblance du test (5.40) :

$$L(\mathbf{x}) = \frac{\Pr_1(\mathbf{x})}{\Pr_0(\mathbf{x})} = \frac{|\Sigma_0|^{1/2}}{|\Sigma_1|^{1/2}} \cdot e^{-\frac{1}{2}\mathbf{x}^T(\Sigma_1^{-1}-\Sigma_0^{-1})\mathbf{x}} = \tau'. \quad (5.51)$$

En appliquant le logarithme au le rapport de vraisemblance, nous obtenons :

$$\begin{aligned}\log(L(\mathbf{x})) &= \frac{1}{2} \log(|\Sigma_0|) - \frac{1}{2} \log(|\Sigma_1|) - \frac{1}{2} \mathbf{x}^T (\Sigma_1^{-1} - \Sigma_0^{-1}) \mathbf{x} \\ &= \log(\tau').\end{aligned}\quad (5.52)$$

Enfin, selon Poor (2013), nous pouvons déduire du logarithme de ce rapport de vraisemblance la règle de décision optimale $\delta^* : \mathbb{R}^K \rightarrow [0, 1]$ basée sur l'estimateur $\Lambda^*(\mathbf{x})$:

$$\delta^*(\mathbf{x}) = \mathbb{1}_{\{\Lambda^*(\mathbf{x}) \geq 0\}} = \begin{cases} 0 & \text{si } \Lambda^*(\mathbf{x}) < 0, \\ 1 & \text{si } \Lambda^*(\mathbf{x}) \geq 0, \end{cases}\quad (5.53)$$

$$\Lambda^*(\mathbf{x}) = \mathbf{x}^T (\Sigma_0^{-1} - \Sigma_1^{-1}) \mathbf{x},\quad (5.54)$$

avec $\tau = \log(|\Sigma_1|) - \log(|\Sigma_0|) + 2 \log(\tau')$, le seuil utilisé pour un test de Neyman-Pearson.

4.3 Règle de détection sous-optimale avec hypothèse de blancheur dans une sous-bande fréquentielle

Une variante plus facile à calculer pour le cas de signaux colorés consiste simplement à effectuer un filtrage passe-bande dans une bande fréquentielle où le bruit ambiant et le bruit de cavitation sont non corrélés. Cette variante, souvent utilisée dans le domaine du traitement du signal, requiert cependant que les signaux soient blancs sur une bande suffisamment large. Cette méthode permet ainsi d'obtenir un problème où les matrices de covariance du signal filtré \mathbf{x}_f sont diagonales, ce qui permet de prendre une décision sur une somme pondérée des échantillons $x_f(k)$. Aussi, dans l'hypothèse H_0 , \mathbf{x}_f suit une distribution Gaussienne multivariée de moyenne nulle et de matrice de covariance $\Sigma_0 = \sigma_a^2 \cdot I_K$. Pour l'hypothèse H_1 , \mathbf{x}_f suit une distribution Gaussienne multivariée $\mathcal{N}(0, \Sigma_1)$ avec une moyenne nulle et une matrice de covariance diagonale $\Sigma_1 = \Sigma_\eta + \sigma_a^2 \cdot I_K$. La règle de décision $\delta : \mathbb{R}^K \rightarrow [0, 1]$ associée à ce problème est donc la même que celle du cas de bruit d'hélice blanc (5.12) à l'exception que l'on utilise \mathbf{x}_f au lieu de \mathbf{x} :

$$\delta(\mathbf{x}) = \begin{cases} 0 & \text{si } \Lambda(\mathbf{x}) = \sum_{k=1}^K b_k x_f^2(k) \leq \tau, \\ 1 & \text{sinon.} \end{cases}\quad (5.55)$$

4.4 Règle de détection sous-optimale avec déflexion

Dans les cas où l'hypothèse de blancheur des bruits de cavitation et ambiant est irréaliste, ou limitée à une bande trop étroite, nous devons utiliser une méthode qui assume des densités spectrales de puissance non uniforme. Une possibilité est de garder la chaîne de traitement précédente en conservant la structure du détecteur (5.55), mais en modifiant le filtrage utilisé. Ainsi, au lieu d'utiliser un filtre passe-bande, nous utiliserons un filtre maximisant la mesure du rapport signal-sur-bruit (ou déflexion) que l'on peut exprimer dans le domaine spectral tel qu'il suit :

$$D(\hat{h}, T) = T \frac{\int_0^{\frac{fs}{2}} (\gamma_1(f) - \gamma_0(f)) \cdot |\hat{h}(f)|^2 df}{\sqrt{\int_0^{\frac{fs}{2}} \gamma_0^2(f) \cdot |\hat{h}(f)|^4 df}},\quad (5.56)$$

avec $T = kfs$ la durée du signal et $\gamma_i(f)$ l'espérance de la densité spectrale de puissance du signal $x(t)$ sous l'hypothèse H_i : $\gamma_i(f) = \mathbb{E}[|\widehat{x}(f)|^2 | H_i]$. En se référant à l'ouvrage de [Poor \(2013\)](#) le filtre optimal h_{op} maximisant $D(T)$ est alors :

$$|\widehat{h}_{op}(f)|^2 = \frac{\gamma_1(f) - \gamma_0(f)}{\gamma_0^2(f)}. \quad (5.57)$$

4.5 Règle de détection avec blanchiment

Une autre variante, qui requiert cependant dans notre cas que le bruit ambiant soit blanc est la méthode de blanchiment du signal. Supposons que $\mathbf{x} \sim \mathcal{N}(0, \Sigma)$, alors la transformation $y = Wx$, avec W la matrice de blanchiment satisfaisant la condition $W^T W = \Sigma^{-1}$ aboutit au vecteur $y \sim \mathcal{N}(0, I_K)$. Il existe une infinité de matrices de blanchiment W satisfaisant cette condition, mais généralement $W = \Sigma^{-\frac{1}{2}}$ (Mahalanobis ou ZCA whitening), la décomposition de Cholesky de Σ^{-1} , ou encore le système de valeurs propres de Σ (PCA whitening), sont les méthodes les plus couramment utilisées comme le stipule [Friedman \(1987\)](#).

En reprenant l'expression générale de l'estimateur optimal (5.54), pour un cas où $\mathbf{x} \sim \mathcal{N}(0, \Sigma_0 = \sigma_a^2 \cdot I_K)$ sous l'hypothèse H_0 , on obtient :

$$\Lambda(\mathbf{x}) = \mathbf{x}^T (\Sigma_0^{-1} - \Sigma_1^{-1}) \mathbf{x} \quad (5.58)$$

$$= \mathbf{x}^T (\sigma_a^{-2} \cdot I_K - (\sigma_a^2 \cdot I_K + \Sigma_\eta)^{-1}) \mathbf{x} \quad (5.59)$$

$$= \mathbf{x}^T (\sigma_a^{-2} \cdot \Sigma_\eta (\sigma_a^2 \cdot I_K + \Sigma_\eta)^{-1}) \mathbf{x}. \quad (5.60)$$

La matrice Σ_η étant symétrique définie positive, toutes ses valeurs propres λ_k sont ainsi réelles positives et ses vecteurs propres peuvent être choisis afin d'être orthonormés (i.e si v_k est un vecteur propre, $v_k^T v_l = 0$ si $k \neq l$ et $v_k^T v_k = 1$ pour tout $l, k = 1, 2, \dots, K$). Avec ce choix de vecteurs propres on peut ainsi réécrire Σ_η tel que :

$$\Sigma_\eta = \sum_{k=1}^K \lambda_k \cdot v_k v_k^T. \quad (5.61)$$

En définissant $x_k = v_k^T \mathbf{x}$ et en remarquant que $I_K = \sum_{k=1}^K v_k v_k^T$ on obtient alors

$$\delta(\mathbf{x}) = \begin{cases} 0 & \text{si } \Lambda(\mathbf{x}) = \sum_{k=1}^K \frac{\lambda_k}{\sigma_a^2(\sigma_a^2 + \lambda_k)} \cdot x_k^2 \leq \tau, \\ 1 & \text{sinon.} \end{cases} \quad (5.62)$$

4.6 Règle de détection sous-optimale avec hypothèse de blancheur dans les sous-bandes fréquentielles

Le filtrage en sous-bandes fréquentielles est un traitement permettant de décolorer un signal $x(k)$ et pouvant être utile pour estimer la vitesse de rotation d'une hélice dans un bruit coloré, comme le montrent [Clark et al. \(2010\)](#) dans leurs travaux. En effet, les processus aléatoires périodiquement corrélés, tels que les bruits d'hélices, peuvent selon [Hurd \(1988\)](#) être

représentés en termes d'intégrales de Fourier en choisissant des bandes de fréquence tel qu'il n'y a corrélation que dans une bande passante de largeur équivalente à la bande passante de modulation. Les signaux des sous-bandes peuvent ainsi être traités comme étant non corrélés entre eux tant que les domaines fréquentiels des filtres passe-bandes utilisés ne se chevauchent pas et que la bande passante de modulation est beaucoup plus petite que la bande passante du filtre $\frac{f_s}{M}$. Sous ces conditions, l'absence de corrélation entre sous-bandes implique une indépendance statistique qui permet d'exprimer le logarithme du rapport de vraisemblance en fonction de ces sous-bandes :

$$\log(L(\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[M]})) = \log\left(\frac{\text{Pr}_1(\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[M]})}{\text{Pr}_0(\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[M]})}\right) \quad (5.63)$$

$$= \sum_{m=1}^M \log(\text{Pr}_1(\mathbf{x}^{[m]})) - \log(\text{Pr}_0(\mathbf{x}^{[m]})). \quad (5.64)$$

Clark et al. (2010) suggèrent ainsi d'utiliser une somme pondérée des estimateurs sous-bandes comme estimateur du maximum de vraisemblance.

Dans notre cas les bruits $n_a(k)$ et $c_{i,\omega}(k)$ étant tous stationnaires au sens large et mutuellement non corrélés, le signal plus bruit $x(k) = s_\eta(k) + n_a(k)$ peut être décomposé en un ensemble de M signaux de sous-bandes spectralement disjointes tel que :

$$x^{[m]}(k) = (h^{[m]} * x)(k), \quad (5.65)$$

où $h^{[m]}(k)$ est la réponse impulsionnelle du $m^{\text{ème}}$ filtre passe-bande de fréquence de coupure $\frac{(2m\pm 1)}{M}f_s$. L'estimateur de chaque sous-bandes serait alors celui associé à l'hypothèse de bruit d'hélice et bruit ambiant blanc de la section 2.3. Certaines conditions doivent cependant être vérifiées pour appliquer cette méthode. Premièrement, les enveloppes $m_{i,\omega}(k)$ doivent fluctuer doucement comparées à la largeur temporelle des filtres passe-bande $h^{[m]}(k)$ afin que :

$$(h^{[m]} * (m_{i,\omega} \cdot c_{i,\omega}))(k) \approx m_{i,\omega}(k) \cdot (h^{[m]} * c_{i,\omega})(k), \quad (5.66)$$

ce qui donnerait ainsi

$$x^{[m]}(k) = \sum_{i=1}^p m_{i,\omega}(k) \cdot c_{i,\omega}^{[m]}(k) + c_{0,\omega}^{[m]}(k) + n_a^{[m]}(k). \quad (5.67)$$

Deuxièmement, M doit être plus important que la longueur de corrélation discrète de $n_a(k)$ et $c_{i,\omega}(k)$, afin que le signal sous échantillonné par un facteur M reste approximativement blanc. Cela équivaut à supposer que le spectre de $x(k)$ est lisse/régulier dans des sous-bandes de l'ordre de $\frac{2f_s}{M}$ Hz. En outre, les enveloppes modulantes sous-échantillonnées $m_{i,\omega}(Mr)$ ne subiront aucune distorsion de repliement spectral tant que la plus importante fréquence de modulation est inférieure à $\frac{f_s}{M}$.

5 Détection de bruit d'hélice aléatoirement retardé

5.1 Énoncé du problème

Nous considérons ici le problème de détection de bruit d'hélice abordé dans la section 4 mais en assumant que le signal s_η est retardé d'un délai τ_0 supposé uniformément réparti sur

$[0, T]$ ou, de façon équivalente, d'un nombre d'échantillons k_{τ_0} :

$$\begin{aligned} H_0 &: \{x(k) = n_a(k), k \in \llbracket K \rrbracket\}, \\ H_1 &: \{x(k) = s_\eta(k - k_{\tau_0}) + n_a(k), k \in \llbracket K \rrbracket\}. \end{aligned} \quad (5.68)$$

Un tel problème requiert de redéfinir l'estimateur de maximum de vraisemblance $\Lambda^*(\mathbf{x})$ pour obtenir une nouvelle règle de décision remplaçant $\delta'(\mathbf{x})$ qui sera cependant sous optimale :

$$\delta'(\mathbf{x}) = \begin{cases} 0 & \text{si } \Lambda'(\mathbf{x}) \leq \tau, \\ 1 & \text{sinon.} \end{cases} \quad (5.69)$$

5.2 Règles de détection

Un estimateur de vraisemblance $\Lambda'(\mathbf{x})$ couramment utilisé pour (5.68) est la valeur maximale obtenue avec $\Lambda^*(\mathbf{x}, \mathbf{k}_{\tau_0})$ en faisant varier k_{τ_0} :

$$\Lambda_{GLRT}(\mathbf{x}) = \max_{k_{\tau_0}} \Lambda(\mathbf{x}, \mathbf{k}_{\tau_0}) \quad (5.70)$$

$$= \max_{k_{\tau_0}} \mathbf{x}^T (\Sigma_0^{-1} - \Sigma_1(k_{\tau_0})^{-1}) \mathbf{x} \quad (5.71)$$

où

$$\Sigma_1(k_{\tau_0}) = \Sigma_0 + \sum_{i=1}^p \mathbf{m}_{i,\omega}(k_{\tau_0}) H_\omega \mathbf{m}_{i,\omega}(k_{\tau_0}) + H_\omega, \quad (5.72)$$

et

$$\mathbf{m}_{i,\omega}(k_{\tau_0}) = \text{diag} [m_{i,\omega}(1 - k_{\tau_0}), \dots, m_{i,\omega}(K - k_{\tau_0})]. \quad (5.73)$$

Ce test, communément appelé test de vraisemblance généralisé (GLRT en anglais), est facilement implémentable avec un réseau de neurones convolutifs. En effet, pour l'architecture présentée en section 3, il suffirait d'utiliser une taille de filtre inférieure au nombre d'échantillons K du signal d'entrée \mathbf{x} .

Dans un cas particulier où les retards ont pour effet de statistiquement permuter les éléments du signal d'intérêt sur la fenêtre d'observation de façon cyclique et où le bruit ambiant est blanc (généralisable pour les bruits de matrice de covariance circulante), Nicolls & De Jager (2007) ont montré qu'il était possible de proposer un détecteur invariant aux retards uniformément plus puissant (UMPI). De telles conditions appliquées à notre modèle de bruit d'hélice donneraient le problème de détection :

$$\begin{aligned} H_0 &: \{\mathbf{x} = \mathbf{n} \sim \mathcal{N}(0, \sigma_a^2 \cdot I_K)\}, \\ H_1 &: \{\mathbf{x} = \mathbf{n} + P^\theta \mathbf{s}_\eta \sim \mathcal{N}(0, \sigma_a^2 \cdot I_K + P^\theta \Sigma_\eta P^{\theta T})\}, \end{aligned} \quad (5.74)$$

où P est la matrice de permutation d'expression

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & & \ddots & 1 \\ 1 & 0 & \cdots & \cdots & 0 \end{bmatrix} \quad (5.75)$$

et $\theta \in [0, K - 1]$. Les test GLRT et UMPI sont alors :

$$\delta_{GLRT}(\mathbf{x}) = \max_{\theta \in [0, K-1]} (P^\theta \mathbf{x})^T M_{a,\eta} (P^\theta \mathbf{x}) \underset{H_0}{\overset{H_1}{\geq}} \tau, \quad (5.76)$$

$$\delta_{UMPI}(\mathbf{x}) = \ln \left(\sum_{\theta=0}^{K-1} e^{\frac{1}{2} (P^\theta \mathbf{x})^T M_{a,\eta} (P^\theta \mathbf{x})} \right) \underset{H_0}{\overset{H_1}{\geq}} \tau, \quad (5.77)$$

où $M_{a,\eta} = (\sigma_a^{-2} \cdot I_K - (\sigma_a^2 \cdot I_K + \Sigma_\eta)^{-1})$.

6 Conclusion

Dans ce chapitre nous avons étudié le problème de classification binaire d'enregistrements de bruits de navires, en se concentrant sur la tâche d'extraire les périodicités cachées dans les bruits d'hélices. En se basant sur un modèle acoustique assez simple, nous avons pu travailler au sein d'un environnement contrôlé, nous permettant de connaître la distribution des données, ainsi que l'expression du détecteur de Bayes optimal pour différentes hypothèses. Aussi, dans le but de concevoir des architectures profondes explicables, nous avons proposé une classe d'architectures profondes bayésiennes permettant d'approximer efficacement, en terme de nombre de paramètres, la structure mathématique optimale du détecteur Bayésien. De tels réseaux de neurones peuvent approcher tous les composants élémentaires du détecteur optimal en utilisant des modules CNN explicables dans le sens où nous pouvons analyser leur rôle dans la décision. Si nous reprenons la taxonomie proposée par [Linardatos et al. \(2021\)](#), parmi les méthodes d'interprétabilité présentées dans la sous-section 3.2 du chapitre 2, notre classe d'architectures bayésiennes devrait être considérée comme une méthode globale, spécifique à un modèle particulier et intrinsèquement explicable.

Chapitre 6

Détection multi-hypothèses

1 Introduction

Dans ce chapitre, nous repartons du modèle de bruit de cavitation d'hélice décrit dans le chapitre 4, pour poser des problèmes de détection binaire composites, non abordées dans le chapitre précédent. La section 2 aborde le problème de détection dans le cas où le niveau d'énergie de la source est inconnu, tandis que la section 4 aborde le problème de détection dans le cas où la signature de la source est inconnue. Nous proposons par la suite des architectures profondes explicables permettant d'approximer efficacement les structures mathématiques de ces détecteurs optimaux et comparons leurs performances à des architectures décrites dans la littérature, tout en mettant en avant les aspects qui les rendent plus explicables.

2 Problème de détection de bruit d'hélice blanc avec SNR inconnu

2.1 Énoncé du problème

Nous considérons ici le problème de détection de bruit d'hélice en supposant que le signal reçu $x(t)$ comporte un bruit d'hélice avec un niveau α inconnu :

$$x(t) = \alpha \cdot s_\eta(t) + n_a(t). \quad (6.1)$$

Nous supposons également que les bruits de cavitation $n_i(t)$ et le bruit ambiant $n_a(t)$, sont des bruits blancs gaussiens et qu'il n'y a qu'une seule signature de bruit d'hélice dont nous connaissons les motifs $m_{i,\omega}(t)$. En considérant un signal reçu $x(t)$ échantillonné en K échantillons tels que l'on obtient le vecteur $\mathbf{x} = [x(1), \dots, x(K)]$, on déduit des équations (6.1) et (4.2) que chaque échantillon $x(k)$ suit la distribution gaussienne

$$x(k) \sim \mathcal{N}(0, \alpha^2 \cdot \sigma_\eta^2(k) + \sigma_a^2), \quad (6.2)$$

$$\sigma_\eta^2(k) = \sum_{i=1}^{\ell} m_{i,\omega}^2(k) \cdot \sigma_i^2 + \sigma_0^2, \quad (6.3)$$

où σ_i^2 et σ_a^2 sont les variances de $n_i(k)$ et $n_a(k)$ respectivement. L'amplitude α est inconnue mais nous supposons qu'elle appartient à un ensemble fini de valeurs prédéfinies et connues

$\mathcal{A} = \{\alpha_1, \dots, \alpha_M\}$. Cela revient à supposer que le SNR peut prendre un nombre fini de valeurs. Notre problème de détection consiste à décider si le signal \mathbf{x} contient uniquement du bruit ambiant (hypothèse H_0) ou s'il contient un signal cible (4.2) avec un niveau d'amplitude arbitraire $\alpha \in \mathcal{A}$ (hypothèse H_1) :

$$\begin{aligned} H_0 &: \{x(k) = n_a(k), k \in \llbracket K \rrbracket\}, \\ H_1 &: \{x(k) = \alpha \cdot s_\eta(k) + n_a(k), \alpha \in \mathcal{A}, k \in \llbracket K \rrbracket\}. \end{aligned} \quad (6.4)$$

où $\llbracket K \rrbracket = \{1, \dots, K\}$. On remarquera que l'hypothèse H_1 est composite puisque α est inconnue. Par conséquent, contrairement à ce qui a été fait dans la sous-section 2.1 du chapitre précédent, nous devons ici résoudre un problème entre une hypothèse simple et une hypothèse composite (Poor (2013)). Les exemples d'apprentissage $(\mathbf{x}^{(i)}, y^{(i)})$ sont ainsi indépendants et suivent une distribution de mélange dénombrable $\mathcal{D}(\mathbf{x}, y)$:

$$\mathcal{D}(\mathbf{x}, y) = q_0 \Pr_0(\mathbf{x}) + q_1 \sum_{m=1}^M \pi_m \Pr_m(\mathbf{x}), \quad (6.5)$$

où $\Pr_0(\cdot)$ et $\Pr_m(\cdot)$ désignent respectivement les mesures de probabilité quand \mathbf{x} suit H_0 et H_1 avec l'amplitude α_m . Les probabilités $q_0 = \Pr(y = 0)$ et $q_1 = \Pr(y = 1) = 1 - q_0$ sont respectivement les probabilités d'occurrence de H_0 et H_1 , tandis que π_m est la probabilité d'obtenir l'amplitude α_m lorsque H_1 se produit. Dans l'hypothèse H_1 , si le signal de navire a une amplitude α_m , \mathbf{x} suit une distribution gaussienne multivariée $\mathcal{N}(0, \Sigma_1)$ de moyenne nulle et de matrice de covariance diagonale $\Sigma_1 = \alpha_m^2 \cdot \Sigma_\eta + \sigma_a^2 I_K$, avec

$$\Sigma_\eta = \text{diag}[\sigma_\eta^2(1), \sigma_\eta^2(2), \dots, \sigma_\eta^2(K)]. \quad (6.6)$$

2.2 Règle de détection bayésienne optimale

Nous sommes intéressés par le calcul du détecteur de Bayes optimal qui minimise la probabilité d'erreur de détection :

$$\mathcal{R}_B(\delta) = q_0 \Pr_0(\delta(\mathbf{x}) = 1) + q_1 \sum_{m=1}^M \pi_m \Pr_m(\delta(\mathbf{x}) = 0), \quad (6.7)$$

Avec la distribution des données d'entraînement $\mathcal{D}(\mathbf{x}, y)$ connue, un bref calcul basé sur le rapport de vraisemblance (5.51) montre que le détecteur de Bayes optimal qui permet d'obtenir le risque réel minimum \mathcal{R}^* est la règle de décision $\delta^* : \mathbb{R}^K \rightarrow [0, 1]$:

$$\delta^*(\mathbf{x}) = \mathbb{1}_{\{\Lambda^*(\mathbf{x}) \geq 0\}} = \begin{cases} 0 & \text{si } \Lambda^*(\mathbf{x}) < 0, \\ 1 & \text{si } \Lambda^*(\mathbf{x}) \geq 0, \end{cases} \quad (6.8)$$

$$\Lambda^*(\mathbf{x}) = c_0 + \sum_{m=1}^M \left(c_m \exp \left(\sum_{k=1}^K b_{m,k} x^2(k) \right) \right), \quad (6.9)$$

où $\Lambda^*(\mathbf{x})$ est la fonction d'estimation, $c_0 = -q_0/q_1$ est une valeur constante,

$$c_m = \pi_m \prod_{k=1}^K \sqrt{\frac{\sigma_a^2}{\alpha_m^2 \sigma_\eta^2(k) + \sigma_a^2}} \geq 0, \forall m \geq 1, \quad (6.10)$$

et

$$b_{m,k} = -\frac{1}{2} \left(\frac{1}{\alpha_m^2 \cdot \sigma_\eta^2(k) + \sigma_a^2} - \frac{1}{\sigma_a^2} \right) \geq 0. \quad (6.11)$$

La structure de cette règle de décision de Bayes $\Lambda^*(\mathbf{x})$ est encore une fois intrinsèquement explicable. La fonction carré $t \mapsto p_2(t) = t^2$ extrait les informations de variance (statistiques du second ordre) des échantillons $x(k)$. Chaque signal $k \mapsto b_{m,k}$ joue le rôle d'un filtre adapté qui calcule un score de profil, lié à la cohérence de la variance du signal d'entrée avec un profil de SNR donné, apparaissant dans H_1 . La fonction exponentielle $t \mapsto e(t) = \exp(t)$ amplifie le score du profil SNR le plus probable. Enfin, le signal $m \mapsto c_m$ agit comme un filtre effectuant une moyenne pondérée des contributions des scores des différents profils, pour finalement obtenir la valeur $\Lambda^*(\mathbf{x})$ nécessaire pour prendre la décision finale.

2.3 Règle de détection alternative pour le cas indénombrable

Alternativement, on peut considérer le problème (6.4) avec \mathcal{A} un intervalle de la forme $[a, b]$ à valeurs dans \mathbb{R} , avec a et b des réels et $a < b$. En se référant aux travaux de Poor (2013), pour un niveau de bruit d'hélice α fixé on peut déduire du rapport de vraisemblance un estimateur du test de Neyman-Pearson de forme quadratique :

$$\Lambda^*(\mathbf{x}) = \mathbf{x}^T (\gamma \cdot \Sigma_\eta (I_K + \gamma \cdot \Sigma_\eta)^{-1}) \mathbf{x}, \quad (6.12)$$

avec $\gamma = \frac{\alpha^2}{\sigma_a^2}$. Bien que le coefficient γ en facteur puisse être absorbé dans le seuil de détection τ du test de Neyman-Pearson, ce n'est pas le cas pour celui présent dans $(I_K + \gamma \cdot \Sigma_\eta)^{-1}$, qui ne peut pas être découplé des observations \mathbf{x} . Ainsi, il n'existe pas de test uniformément plus puissant (UMP) pour ce problème de détection avec niveaux de bruit d'hélice indénombrable. Cependant, Poor (2013) propose d'utiliser un test statistique localement plus puissant (LMP) en dérivant l'expression du test (6.12) par rapport à γ , puis en fixant cette valeur à 0. Ainsi le test statistique LMP emploierait un estimateur localement optimal

$$\Lambda(\mathbf{x}) = 2 \cdot \mathbf{x}^T \Sigma_\eta \mathbf{x} \quad (6.13)$$

et très souvent utilisé dans la pratique. Sur les figures 6.1 sont affichées les courbes ROC de tests de Monte Carlo comparant les performances de trois différents détecteurs : un détecteur optimal basé sur deux niveaux de source discrets (6.9), en trait continu, un détecteur optimal basé sur unique niveau de source discret (5.12), en trait discontinu et du détecteur localement optimal (6.13), en trait et points. Les courbes en orange représentent un problème de détection comportant uniquement un bruit d'hélice de SNR -4.4 dB sur la figure de gauche et -8.84 dB pour la figure de droite. Similairement, les courbes en vert représentent un problème de détection comportant uniquement un bruit d'hélice de SNR -8.84 dB sur la figure de gauche et -15.29 dB pour la figure de droite. Enfin, les courbes en bleue représentent un problème de détection comportant un mélange équiprobable de deux niveaux de bruit d'hélice de SNR -4.4 dB et -8.84 dB à gauche et -8.84 dB et -15.29 dB à droite. Cet affichage a été réalisé dans le but de vérifier si un détecteur naturel, basé sur un seul filtre, pourrait avoir des performances similaires au détecteur optimal comportant M filtres si M niveaux de bruit d'hélice sont présents dans le jeu de données. Aussi, puisque la courbe qui représente le test LMP disposant d'un seul filtre, en traits et points bleus, a des performances bien inférieures

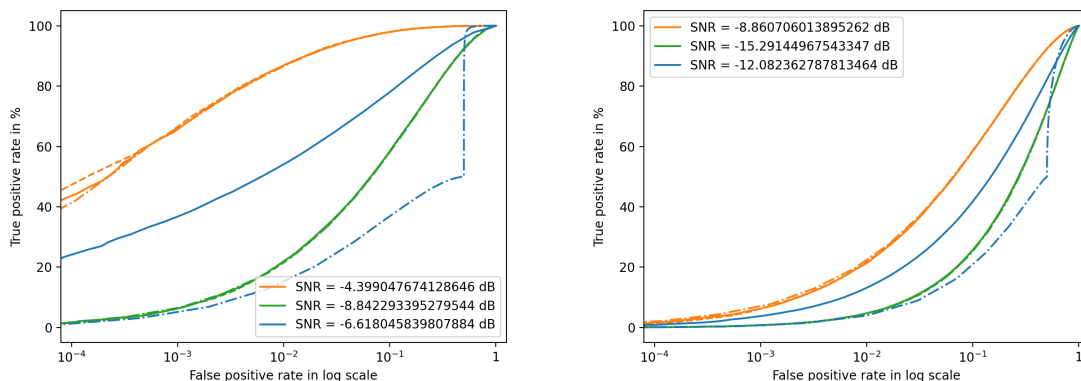


Figure 6.1: Courbes ROC de trois différents détecteurs pour des distributions de données comportant un unique niveau de bruit d'hélice en orange et vert et un mélange de deux niveaux de bruit d'hélice en bleu.

au détecteur optimal, cette expérimentation ne permet pas de répondre à cette question, mais elle renforce l'idée qu'un classifieur profond devrait apprendre plusieurs filtres différents pour obtenir de bonnes performances.

3 Apprentissage profond pour la détection de bruit d'hélice blanc avec SNR inconnu

3.1 Architecture profonde bayésienne

Connaissant la forme du détecteur de Bayes optimal $\delta^*(\mathbf{x})$ pour le problème de détection de bruit d'hélice avec SNR inconnu, tout comme dans le chapitre précédent, nous proposons d'approximer la structure mathématique de l'estimateur optimal $\Lambda^*(\mathbf{x})$ dans l'équation (6.9) par le biais d'un modèle profond appartenant à une classe d'architectures bayésiennes explicables \mathcal{F}^* . L'architecture $f_\theta \in \mathcal{F}^*$ devrait une fois de plus permettre d'approximer séparément chacune des fonctions et opérations internes présentes dans $\Lambda^*(\mathbf{x})$. Ceci doit permettre de différencier le rôle des différentes couches de neurones ou modules dans f_θ , un prérequis indispensable pour mettre en avant une potentielle explicabilité intrinsèque du modèle. Par définition, la classe de fonctions d'apprentissage profond \mathcal{F}^* contient le réseau $f_\theta(\mathbf{x})$ de structure :

$$f_\theta(\mathbf{x}) = \lambda_0 + \sum_{m=1}^M \lambda_m \Phi_{e,\gamma_2} \left(\sum_{k=1}^K \varphi_{m,k} \Phi_{p_2,\gamma_1}(x(k)) \right), \quad (6.14)$$

où $\Phi_{p_2,\gamma_1} : \mathbb{R} \mapsto \mathbb{R}$, et $\Phi_{e,\gamma_2} : \mathbb{R} \mapsto \mathbb{R}$ sont des modules de réseau de neurones d'approximation de fonction de paramètres γ_1 et γ_2 , dédiés à l'approximation des fonctions carré $p_2(t) = t^2$ et exponentielles e^t , présentes dans $\Lambda^*(\mathbf{x})$. Les paramètres $\lambda_0, \dots, \lambda_m$ et $\varphi_{m,k}$ jouent respectivement le rôle des coefficients c_0, \dots, c_m et $b_{m,k}$, dans l'équation (6.9). La structure imposée pour $f_\theta \in \mathcal{F}^*$ nous permet ainsi d'interpréter ultérieurement le rôle de chaque couche ou groupe de couches dans f_θ .

3.2 Approximation de l'estimateur optimal avec une architecture profonde bayésienne

En se référant à la section 3.2 du chapitre 5, il existe deux réseaux de neurones complètement connectés, respectivement Φ_{p_2, γ_1} et Φ_{e, γ_2} , tels que $\|\Phi_{p_2, \gamma_1}(t) - t^2\|_{L^\infty(I)} \leq \epsilon$ et $\|\Phi_{e, \gamma_2}(t) - \exp(t)\|_{L^\infty(I)} \leq \epsilon$. En exploitant des résultats provenant de [Elbrächter et al. \(2019\)](#) et ([Devroye et al. 2013](#), chap 2), nous sommes en mesure d'établir le théorème 10 ci-dessous, la preuve étant détaillée en annexe B section 1.

Borne d'approximation d'une architecture profonde bayésienne

Théorème 10 *Soit $\epsilon \in (0, \frac{1}{2})$. Alors, il existe un réseau complètement connecté f_θ disposant de deux modules d'approximation de fonction $\Phi_{g, \gamma}$ de largeurs $W \leq 9$ et de profondeurs L ainsi que deux constantes Q et C tels que*

$$\inf_{f_\theta \in \mathcal{F}^*} \sup_{\mathbf{x} \in \mathcal{X}} |f_\theta(\mathbf{x}) - \Lambda^*(\mathbf{x})| \leq C \epsilon, \quad (6.15)$$

à condition que $L \leq Q \log_2^2(1/\epsilon) + \log_2(1 + 1/D)$ et avec C dépendant de $e^{\epsilon\sqrt{K}}$, des coefficients $\varphi_{m,k}$, des λ_m , ainsi que de $\|\mathbf{x}\|$. De plus, il s'ensuit que

$$\inf_{f_\theta \in \mathcal{F}^*} \mathcal{R}(f_\theta) - \mathcal{R}^* \leq C' \epsilon, \quad (6.16)$$

où C' est une constante.

En d'autres termes, l'équation (6.16) montre que l'erreur d'approximation de notre classe d'architecture bayésienne de réseaux de neurones \mathcal{F}^* est très petite en raison de la borne (6.15), tout en étant explicite.

3.3 Architecture CNN

Cette sous-section explicite l'implantation avec la terminologie Keras de la fonction $f_\theta \in \mathcal{F}^*$ sous forme de CNN. Notre approximation de $\Lambda^*(\mathbf{x})$ avec l'architecture $f_\theta(\mathbf{x})$ dans (6.14), est décrite dans la figure 6.2. Cette approximation est basée sur la composition de modules d'approximation de fonction avec des modèles profonds distincts Φ_{p_2, γ_1} et Φ_{e, γ_2} . Il découle de (6.14) que nous devons calculer $\widehat{\Phi}_{p_2, \gamma_1}(\mathbf{x})$ et $\widehat{\Phi}_{e, \gamma_2}(\mathbf{s}_\theta)$ où $\mathbf{s}_\theta = [s_\theta(1), \dots, s_\theta(M)]$ et

$$s_\theta(m) = \sum_{k=1}^K \varphi_{m,k} \Phi_{p_2, \gamma_1}(x(k)). \quad (6.17)$$

Les réseaux de neurones Φ_{p_2, γ_1} et Φ_{e, γ_2} sont implémentés comme le CNN dans la figure 5.1. Ainsi, les L premières couches convolutives de la figure 6.2 calculent la fonction CNN $\widehat{\Phi}_{p_2, \gamma_1}(\mathbf{x})$. Ensuite, la couche convolutive suivante, $\varphi_{m,k}$, calcule les M sommes linéaires $s_\theta(m)$ dans (6.17). Une couche "flatten" est ensuite ajoutée pour obtenir un vecteur réarrangé de dimension $M \times 1$. Ensuite, les L couches convolutives suivantes calculent $\widehat{\Phi}_{e, \gamma_2}(\mathbf{s}_\theta)$, comme le

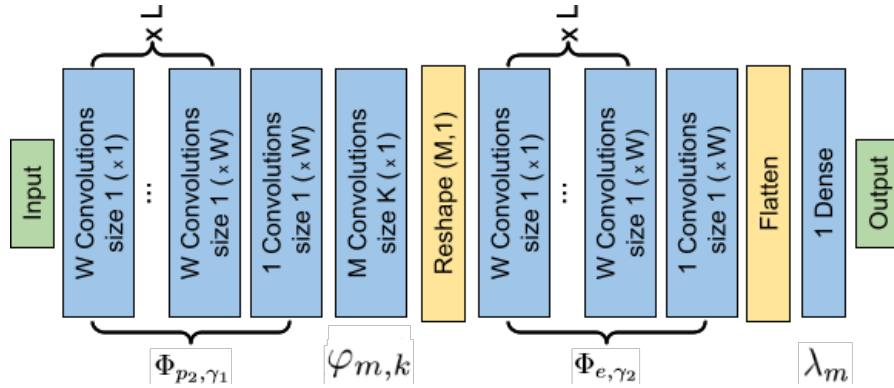


Figure 6.2: Architecture de $f_\theta(\mathbf{x})$ avec la terminologie de la librairie Keras.

montre la figure 5.1. La couche "flatten" suivante produit un vecteur de taille $M \times 1$, tandis que la couche dense finale calcule la sortie linéaire

$$f_\theta(\mathbf{x}) = \lambda_0 + \sum_{m=1}^M \lambda_m \Phi_{e,\gamma_2}(s_\theta(m)). \quad (6.18)$$

L'extraction des noyaux convolutifs appris par les modules CNN Φ_{p2,γ_1} et Φ_{e,γ_2} nous permet ensuite de visualiser l'approximation des fonctions carré et exponentielles apprises, permettant ainsi d'interpréter leurs rôles.

3.4 Méthodologie pour les expérimentations

Afin d'évaluer le nombre d'exemples N requis pour assurer la généralisation de notre architecture bayésienne, comparativement à des architectures classiques, nous confronterons les performances de notre réseau à celle du FCNN présenté par Dai et al. (2017), une autre architecture profonde sur signaux acoustiques bruts présentée dans le chapitre 2 et utilisant bien plus de paramètres pour résoudre des problèmes plus complexes.

Pour estimer les performances de généralisation de notre modèle nous utiliserons une estimation du risque réel $\mathcal{R}(f_\theta)$ avec un ensemble de test indépendant \mathcal{S}_{test} et un ensemble de validation également indépendant \mathcal{S}_{val} . L'ensemble d'apprentissage \mathcal{S} contient jusqu'à $N_{max} = 2 \cdot 10^5$ signaux disposant de $K = 220$ échantillons et $M = 2$ niveaux de bruits α_m différents, tandis que \mathcal{S}_{test} et \mathcal{S}_{val} contiennent $1 \cdot 10^5$ échantillons. La précision de nos modèles sera comparée aux précisions du modèle de comparaison et du test de Bayes optimal $\delta^*(\mathbf{x})$, à la fois sur les ensembles d'apprentissage et de test.

Enfin, comme moyen de voir si la distance entre les distributions H_0 et H_1 impacte l'excès de risque et l'erreur de généralisation, nous utilisons des couples de valeurs α_m : (α_1, α_2) et (α_2, α_3) . Les signaux de bruits d'hélices associés avec les niveaux α_1 , α_2 et α_3 auront respectivement des SNR moyens de -4.4, -8.8 et -13.2 dB dans l'hypothèse H_1 . Dans ces deux cas, étant donné que la structure du détecteur de Bayes et ses paramètres sont connus, la valeur \mathcal{R}^* sera calculée le plus précisément possible par une méthode de Monte-Carlo.

FCNN	$f_{\theta}(\mathbf{x})$
Entrée : 220×1 signal temporel $x(k)$	
Conv [16, 55]	Conv [9, 1] $\times L_1$
MaxPool : 2×2	
Conv [32, 12]	Conv [1, 1]
MaxPool : 2×2	Conv [2, 220]
Conv [64, 12]	Conv [9, 1] $\times L_2$
MaxPool : 2×2	
Conv [128, 12]	Conv [1, 1]
GlobalAveragePool	Flatten layer
Final dense layer	

Tableau 6.1: Tableau des différentes architectures utilisées pour la comparaison, avec leurs titres associés en première ligne. La couche "Conv" représente une couche de convolution avec "same padding". Dans les couches convolutives, les nombres entre crochets indiquent respectivement le nombre de filtres et la taille du noyau de convolution. La stride/le pas est toujours fixée à 1 et le nombre multiplié aux couches Conv est le nombre de fois que la couche est réutilisée. Pour les opérations "Pool", on affiche respectivement la taille et la stride.

3.5 Optimisation des modèles profonds

Le tableau 6.1 décrit notre architecture bayésienne ainsi que le réseau FCNN de comparaison par le biais de couches Keras. Pour définir notre architecture d'approximation $\hat{\delta}_{\theta}$, nous utilisons l'équivalent de $W = 9$ neurones par couche avec $W = 9$ convolutions de taille 1 et déterminons le nombre de couches des modules d'approximation de fonction Φ_{p_2, γ_1} et Φ_{e, γ_2} , respectivement L_1 et L_2 , grâce à une "GridSearch" sur la profondeur de couche $L = \{1, 3, 6\}$ avec la fonction de perte de validation comme métrique. Notre réseau de neurones profond est comparé à une architecture FCNN tirée des travaux de Dai et al. (2017) et adaptée pour correspondre à la taille des données d'entrée utilisées dans notre problème. L'architecture FCNN comprend une succession de 4 couches de convolution et d'opérations de max pooling, suivies d'une opération d'average pooling globale. Les noyaux de convolution utilisés sont d'abord de taille 64 pour la première couche puis de taille 12 pour les couches suivantes. La quatrième couche de convolution dans l'architecture FCNN est appliquée à un ensemble de feature maps de dimension (128, 12), ce qui signifie qu'elle permet de pondérer l'ensemble de l'information temporel et donne une sortie de dimension (256, 1).

Notre architecture bayésienne et le FCNN de comparaison suivent la même procédure d'optimisation lors de l'apprentissage, avec l'utilisation de l'optimiseur Adam et de l'entropie croisée binaire comme fonction de perte. Le critère d'arrêt est une absence d'amélioration de la perte de validation pendant 15 époques consécutives, ou au maximum après 100 époques d'entraînement. La contrainte "Maximum kernel" est utilisée pour toutes les couches convolutives afin de majorer les valeurs des poids des filtres convolutifs et nous appliquons une régularisation L^2 aux couches de sortie, de même qu'une normalisation par batch pour toutes les couches convolutives. Pour chaque valeur N considérée dans les affichages de la figure 6.3, les performances des modèles sont estimées en calculant la moyenne et l'écart type des erreurs

d'entraînement et de test parmi 15 entraînements différents. Un modèle avec une perte de validation supérieure à $1,2\times$ la meilleure perte de validation, pour un N donné, est rejeté pour le calcul des performances.

3.6 Résultats numériques

La figure 6.3 affichées ci-dessous représente la précision d'entraînement $1 - \mathcal{R}_N(f_\theta)$ ainsi que la précision de test, estimation de la précision réelle $1 - \mathcal{R}(f_\theta)$ des réseaux de neurones employés. Chaque courbe est représentée avec son intervalle de confiance basé sur les résultats des différents entraînements d'une même architecture. Sur la figure, la précision de notre

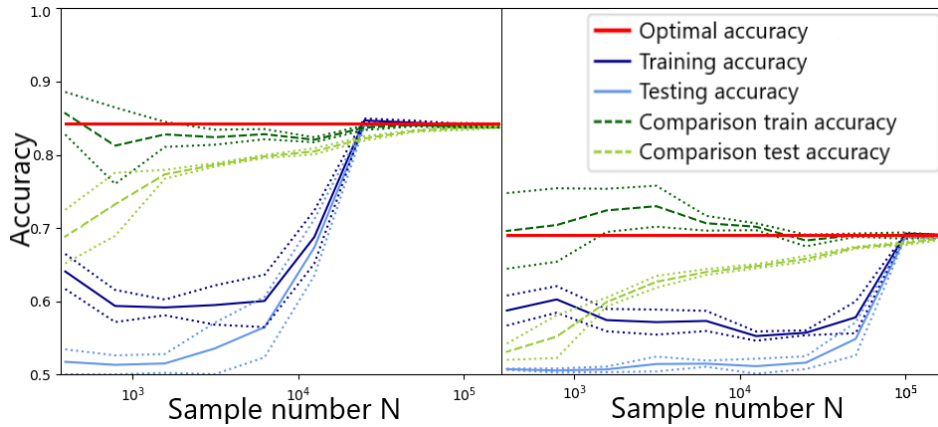


Figure 6.3: Précision d'entraînement et de test en fonction de la taille du jeu de données associé pour deux scénarios de SNR respectivement lié au niveaux $(\alpha_1 = a_1, \alpha_2 = a_2)$ à gauche et $(\alpha_1 = a_2, \alpha_2 = a_3)$ à droite. La précision optimale correspond à un test de Monte-Carlo avec le détecteur (6.8).

modèle, ainsi que la précision du modèle de comparaison convergent vers la précision optimale $1 - \mathcal{R}^*$ pour les deux scénarios de SNR sur un jeu de données disposant de deux niveaux (α_1, α_2) . Il est également évident, en comparant les deux scénarios, que le SNR dans l'ensemble d'apprentissage affecte à la fois le risque et l'erreur de généralisation des deux architectures profondes employées. La baisse de la précision sur le jeu d'entraînement de la part de notre modèle, pour des faibles valeurs N , peut être attribuer au fait qu'il soit plus facile à notre modèle de surapprendre pour de petits jeux de données, tandis qu'à partir d'une certaine taille N , dépendant du problème et du réseau, ajouter des exemples d'entraînement permet d'améliorer la précision sur le jeu d'entraînement. Ceci pourrait indiquer que l'espace défini par la fonction de perte $\mathcal{L}_{0-1}(\hat{\delta}_\theta(\mathbf{x}), y)$ est alors plus favorable à la découverte de bons minima locaux. Le FCNN de comparaison présent dans la figure 6.3 agit comme une boîte noire avec comme seul objectif la précision de classification. Malgré un léger surapprentissage, il est plus efficace que notre modèle pour des ensembles d'apprentissage de petites tailles. C'est assez consistant avec les bonnes performances de généralisation couramment observées dans les réseaux profonds à fortes capacités d'abstraction, dans des situations où il y a peu de données N par rapport au nombre de paramètres à apprendre. Cela semble ainsi être le prix à payer pour l'explicabilité des réseaux de neurones et donc dans notre cas, pour l'estimation des fonctions $\Phi_{p_2, \gamma_1}(t)$ et $\Phi_{e, \gamma_2}(t)$.

3.7 Résultats sur l'explicabilité

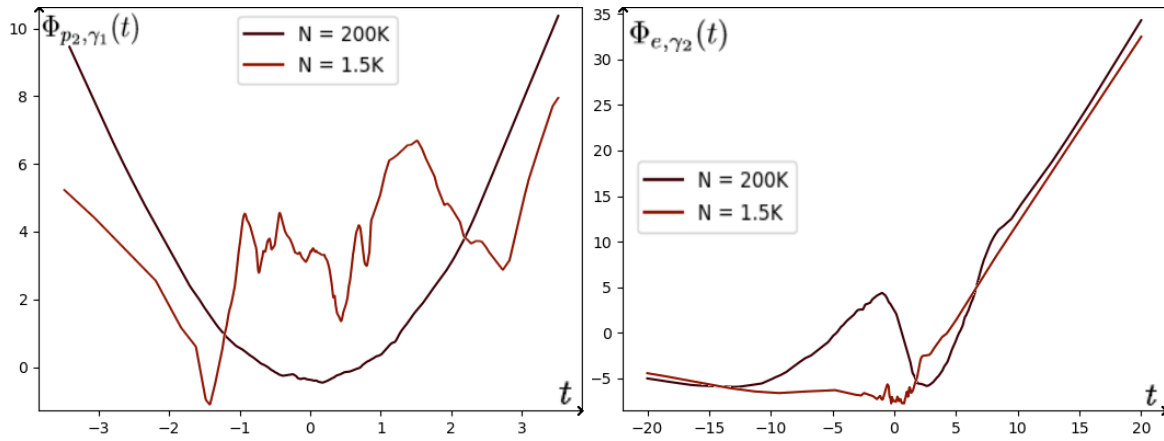


Figure 6.4: Exemple de fonctions apprises par un module CNN Φ_{p_2, γ_1} à gauche et Φ_{e, γ_2} à droite.

La figure 6.4 montre des exemples de fonctions internes $\Phi_{p_2, \gamma_1}(t)$ et $\Phi_{e, \gamma_2}(t)$ apprises pendant l'entraînement d'un modèle $f_\theta(\mathbf{x})$ pour $N = N_{max}$ et $N = 1.5 \cdot 10^3$. Ces fonctions sont reconstruites à partir du modèle de réseau de neurones (5.29) en extrayant les poids correspondants aux modules $\Phi_{g, \gamma}(t)$ de l'ensemble du réseau de neurones $f_\theta(\mathbf{x})$ décrit dans la figure 6.2. On peut voir qu'avec suffisamment d'échantillons d'apprentissage, notre modèle parvient à apprendre une fonction carré $p_2(t)$ relativement lisse. Pour l'exemple choisie, il donne également une approximation de la fonction exponentielle $e(t)$ qui peut être globalement raisonnable, mais pouvant être contrintuitive pour $t \in [0, 7]$ bien que cela n'empêche pas de bonnes performances de classification avec la couche dense suivante dans le cas présent. Cela peut laisser supposer que le nombre de SNRs présents dans le problème, M , ou la différence entre les SNR n'est pas suffisant pour définir $\Phi_{e, \gamma_2}(t)$ comme escompté de façon consistante avec différents entraînements de l'architecture employée pour $f_\theta(\mathbf{x})$.

La figure 6.5 montre que le filtre adapté $k \mapsto b_{1, k}$ est également correctement appris, bien que bruité. C'est également généralement le cas pour les autres filtres adaptés $k \mapsto b_{m, k}$, d'aspects très similaires en raison de la nature du problème (6.4) posé.

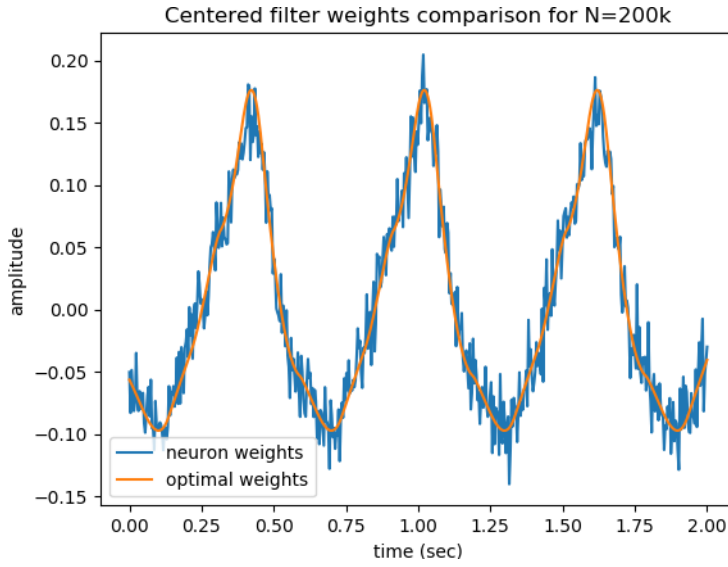


Figure 6.5: *Filtre optimal $k \mapsto b_{1,k}$ et son estimation par le réseau profond Bayésien $k \mapsto \varphi_{1,k}$.*

3.8 Résultats sur le déséquilibre des sources dans les données

Nous évaluons ici comment un déséquilibre de représentation entre deux différentes sources de SNRs (écart entre les probabilités d’apparition du niveau α_1 et α_2 dans H_1) dans les données, peut impacter les performances statistiques de notre architecture, en donnant par exemple un biais trop important à un filtre par rapport à l’autre durant l’entraînement. Nous répétons ici les expérimentations précédentes, avec des ensembles de test et d’apprentissage disposant de $N = N_{max}$ données et caractérisées par des rapports π_1/π_2 différents. Le tableau 6.2 reporte uniquement les performances des modèles ayant obtenus les meilleures pertes de validation parmi 10 différents entraînements.

Bien que nos modèles surapprennent légèrement lorsque $\pi_1 = \pi_2$ et α_1 prédominant, il semble que lorsque le SNR le plus faible prédomine dans le jeu de données d’apprentissage, les modèles ont des performances légèrement inférieures à la précision optimale du détecteur de Bayes.

Probabilité π_1	0.875	0.75	0.875	0.75	0.875
Précision optimal	0.9212	0.8926	0.8416	0.8015	0.7842
Précision d’entraînement	0.9434	0.8958	0.8437	0.7869	0.7709
Précision de test	0.9218	0.8934	0.8408	0.7804	0.7722
Précision de test avec $\pi_1 = \pi_2$	0.7439	0.7933	0.8408	0.8402	0.8399

Tableau 6.2: *Précision d’entraînement de test de modèles entraînés sur des jeux d’entraînements avec différentes probabilités $\pi_m = 1 - \sum_{i \neq m} \pi_i$ d’obtenir un niveau de source α_m dans les données. La précision du détecteur optimal dans pour ces mêmes conditions est donnée en comparaison.*

4 Problème de classification de bruit d'hélice blanc avec signature inconnue

4.1 Énoncé du problème

Nous considérons ici un problème de classification entre bruits d'hélices et bruits parasites qui auraient été initialement détectés sur un critère d'énergie $\Lambda_d(\mathbf{x}_r)$, les distinguant ainsi du bruit ambiant multi-directionnel. Dans un premier temps, nous considérons une détection en sortie de formation de voie, que nous noterons $x_r(t)$. Les bruits parasites $n_p(t)$ représentent un bruit directionnel qui ne serait pas produit par un bruit d'hélice et ne représenterait ainsi pas d'intérêt opérationnel (bruit de cétacés, travaux côtiers, ...), contrairement au bruit d'hélice $s_\eta(t)$. Les paramètres de signatures η sont inconnus, mais nous supposons qu'ils appartiennent à un ensemble fini de signatures prédéfinies $\mathcal{H} = \{\eta_1, \dots, \eta_M\}$ dont nous connaissons les motifs $m_{i,\omega}(t)$. Nous supposons que les bruits de cavitation $n_i(t)$, le bruit ambiant $n_a(t)$ et les bruits parasites $n_p(t)$, sont des bruits blancs gaussiens. Notre problème de classification consiste donc à décider si le signal \mathbf{x}_r contient, en plus du bruit ambiant, uniquement une source parasite ayant abouti à une détection (hypothèse H_0), ou s'il contient un signal de bruit d'hélice (4.2) de paramètre de signature $\eta \in \mathcal{H}$ (hypothèse H_1) :

$$\begin{aligned} H_0 &: \{x_r(k) = n_p(k) + n_a(k), k \in \llbracket K \rrbracket\}, \\ H_1 &: \{x_r(k) = s_\eta(k) + n_a(k), \eta \in \mathcal{H}, k \in \llbracket K \rrbracket\}. \end{aligned} \quad (6.19)$$

La règle de décision optimale $\delta^* : \mathbb{R}^K \rightarrow [0, 1]$ pour ce problème serait alors :

$$\delta^*(\mathbf{x}_r) = \begin{cases} 0 & \text{si } \Lambda^*(\mathbf{x}_r) = \sum_{k=1}^K \left(\frac{1}{a_p^2 + \sigma_a^2} - \frac{1}{\sigma_\eta^2(k) + \sigma_a^2} \right) \cdot x_r^2(k) \leq \tau \\ 1 & \text{sinon.} \end{cases} \quad (6.20)$$

Remarque

La règle de décision employée $\delta_d(\mathbf{x}_r)$, souvent basée sur un estimateur lié au niveau d'énergie

$$\Lambda_d(\mathbf{x}_r) = \int_{t=0}^T x_r(t)^2 dt, \quad (6.21)$$

est choisie pour obtenir un taux de fausse alarme α_d , que l'on considérera négligeable (10^{-6}) et impose une puissance de test $\geq 1 - \beta_d$ pour le problème de détection (5.1). Cela implique que les signaux reçus, devant être classifiés par une règle $\delta(\mathbf{x})$, auront des SNRs plus élevés que ceux induit par le point de fonctionnement choisi pour $\delta_d(\mathbf{x}_r)$. Aussi les performances de tout classifieur ultérieur seront prescrites par celles de δ_d , comme le soulignent Baygun & Hero (1995).

Nous ferons cependant l'hypothèse qu'après détection énergétique, le niveau d'énergie des potentiels signaux des sources n'est plus classifiant. Nous normalisons ainsi le signal détecté

$x_r(k)$ par son niveau d'énergie théorique moyen E_r durant la génération des données, pour obtenir un signal normalisé $x(k)$ tel que :

$$x(k) = \frac{x_r(k)}{\sqrt{E_r}}, \quad (6.22)$$

$$E_r = \mathbb{E} \left[\frac{1}{K} \sum_{k=1}^K x_r^2(k) \right]. \quad (6.23)$$

Notre problème de classification consistera donc à déterminer si une détection énergétique discrète normalisée $x(k)$ contient un signal provenant d'un navire $s_\eta(k)$ ou pas :

$$\begin{aligned} H_0 &: \{x_r(k) = n(k), k \in \llbracket K \rrbracket\}, \\ H_1 &: \{x_r(k) = \frac{1}{\sqrt{E_\eta}} \cdot (s_\eta(k) + n_a(k)), \eta \in \mathcal{H}, k \in \llbracket K \rrbracket\}, \end{aligned} \quad (6.24)$$

avec E_η le niveau d'énergie théorique moyen pour la signature η . Dans H_0 , on considère que les détections énergétiques normalisées $x(k)$ sont des bruits blancs gaussiens modélisant le bruit ambiant et les événements acoustiques parasites. \mathbf{x} suit alors une distribution normale multivariée :

$$\mathbf{x} \sim \mathcal{N}(0, \Sigma_0 = I_K). \quad (6.25)$$

Dans l'hypothèse H_1 , si le signal cible est défini par les paramètres de signature η , \mathbf{x} suit une distribution gaussienne multivariée $\mathcal{N}(0, \Sigma_1)$ de moyenne nulle et de matrice de covariance diagonale

$$\Sigma_1 = \frac{1}{E_\eta} \cdot (\sigma_a^2 \cdot I_K + \Sigma_\eta), \quad (6.26)$$

avec

$$E_\eta = \frac{1}{K} \sum_{k=1}^K \sigma_\eta^2(k) + \sigma_a^2, \quad (6.27)$$

$$\Sigma_\eta = \text{diag}[\sigma_\eta^2(1), \sigma_\eta^2(2), \dots, \sigma_\eta^2(K)]. \quad (6.28)$$

On remarquera que l'hypothèse H_1 est composite puisque les paramètres de signature η sont inconnus. Comme dans la section 2, les exemples d'apprentissage $(\mathbf{x}^{(i)}, y^{(i)})$ sont ainsi indépendants et suivent une distribution de mélange dénombrable $\mathcal{D}(\mathbf{x}, y)$:

$$\mathcal{D}(\mathbf{x}, y) = q_0 \text{Pr}_0(\mathbf{x}) + q_1 \sum_{m=1}^M \pi_m \text{Pr}_m(\mathbf{x}), \quad (6.29)$$

où $\text{Pr}_0(\cdot)$ et $\text{Pr}_m(\cdot)$ désignent respectivement les mesures de probabilité quand \mathbf{x} suit H_0 et H_1 avec les paramètres de signature η_m . Les probabilités $q_0 = \Pr(y = 0)$ et $q_1 = \Pr(y = 1) = 1 - q_0$ sont respectivement les probabilités d'occurrence de H_0 et H_1 , tandis que π_m est la probabilité d'obtenir les paramètres de signature η_m lorsque H_1 se produit.

Remarque

Dans un scénario réel, on normalise généralement le signal détecté $x_r(k)$ par la mesure de son énergie. Or, si on choisit de normaliser les signaux gaussiens $x_r(k)$ du test d'hypothèse (6.19) par leur normes l^2 , on obtient alors des signaux suivant une loi de Student et non une loi Gaussienne. La normalisation que nous employons et qui implique une distribution gaussienne des signaux normalisés, est cependant d'autant plus valide que le nombre d'échantillons temporels discrets K est élevé, puisque toute distribution tend vers une distribution gaussienne avec suffisamment d'échantillons.

4.2 Règle de détection bayésienne optimale

Avec la distribution des données d'entraînement $\mathcal{D}(\mathbf{x}, y)$ du problème (6.24) connue, on peut déduire le détecteur de Bayes optimal permettant d'obtenir le risque réel minimum \mathcal{R}^* , la règle de décision $\delta^* : \mathbb{R}^K \rightarrow [0, 1]$:

$$\delta^*(\mathbf{x}) = \mathbb{1}_{\{\Lambda^*(\mathbf{x}) \geq 0\}} = \begin{cases} 0 & \text{si } \Lambda^*(\mathbf{x}) < 0, \\ 1 & \text{si } \Lambda^*(\mathbf{x}) \geq 0, \end{cases} \quad (6.30)$$

$$\Lambda^*(\mathbf{x}) = c_0 + \sum_{m=1}^M \left(c_m \exp \left(\sum_{k=1}^K b_{m,k} x^2(k) \right) \right), \quad (6.31)$$

où $\Lambda^*(\mathbf{x})$ est la fonction d'estimation, $c_0 = -q_0/q_1$ est une valeur constante,

$$c_m = \pi_m \left(\prod_{k=1}^K \frac{\sigma_a^2 + \sigma_{\eta_m}^2(k)}{E_{\eta_m}} \right)^{-1/2} \geq 0, \forall m \geq 1, \quad (6.32)$$

et

$$b_{m,k} = -\frac{1}{2} \left(\frac{E_{\eta_m}}{\sigma_a^2 + \sigma_{\eta_m}^2(k)} - 1 \right). \quad (6.33)$$

La structure de cette règle de décision de Bayes $\Lambda^*(\mathbf{x})$ est similaire à celle du problème précédent (6.4), si ce n'est que le signal $k \mapsto b_{m,k}$ joue le rôle de filtre adapté lié à une signature d'hélice $\sigma_{\eta_m}^2(k)$.

Les signatures d'hélice $\sigma_{\eta_m}^2(k)$ pouvant être très différentes, comme en attestent les images à gauche et à droite de la figure 6.8, l'image en bas sur la figure 6.8 est affichée dans le but de comparer les performances du détecteur binaire (5.12) du chapitre 5 pour des cas de signatures extrêmes. On peut ainsi constater un écart de probabilité de vrais positifs de plus de 20 % à 0.01% de fausse alarme pour un cas où les signatures générées par notre modèle acoustique sont particulièrement différentes.

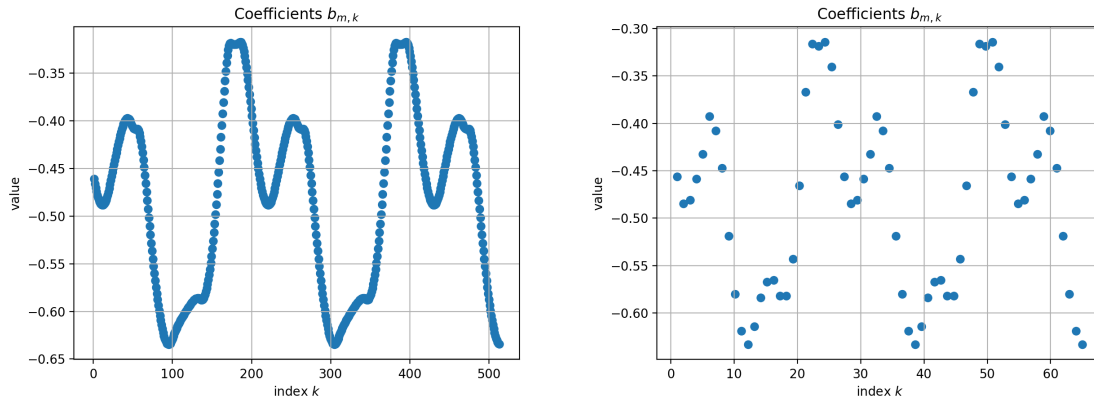


Figure 6.6: Coefficients $b_{m,k}$ pour un signal de 2 secondes avec les paramètres de signature η_m présenté en figure 4.1 et une fréquence d'échantillonnage $f_s = 256$ Hz (gauche) et $f_s = 32$ Hz (droite).

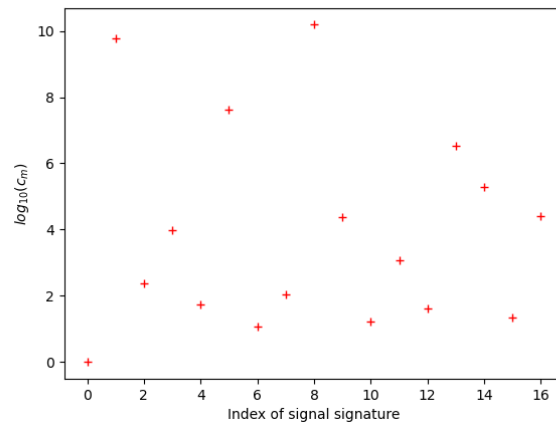


Figure 6.7: Coefficients c_m en échelle logarithmique pour un problème comportant $M = 16$ signatures différentes.

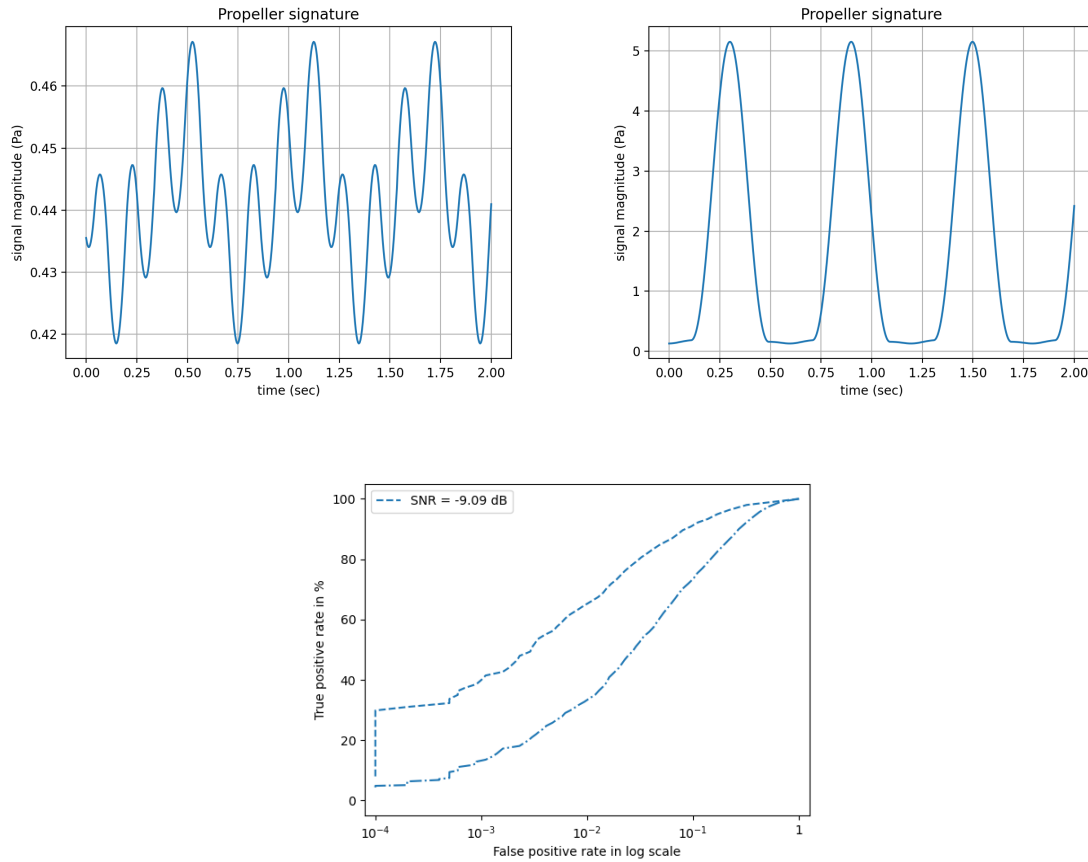


Figure 6.8: Signatures d'hélice $\sigma_{\eta_m}^2(k)$ normalisées liées aux contributions individuelles d'une hélice avec des pales similaires, à gauche et d'une hélice avec une pale relativement distincte à droite. L'image du bas est une comparaison entre les courbes COR issues de tests de détection bayésiens binaires pour ces signatures.

Par la suite nous analysons ainsi les contributions des scores de profils $\sum_{k=1}^K b_{l,k} x^2(k)$ des différents paramètres de signatures η_l pour différentes hypothèses de distribution du signal d'entrée $\mathbf{x} \sim H_m$. Dans l'hypothèse H_0 , $x(k)$ suit une distribution normale et par conséquent, $x^2(k) \sim \chi^2(1)$. On peut alors déduire la moyenne :

$$\begin{aligned} \mathbb{E} \left[\sum_{k=1}^K b_{m,k} x^2(k) | H_0 \right] &= \mathbb{E} \left[\sum_{k=1}^K b_{m,k} x^2(k) \right] \\ &= \sum_{k=1}^K b_{m,k} \mathbb{E}[x^2(k)] \\ &= \sum_{k=1}^K b_{m,k} \end{aligned} \quad (6.34)$$

et la variance

$$\begin{aligned} \mathbb{V} \left[\sum_{k=1}^K b_{m,k} x^2(k) | H_0 \right] &= \mathbb{V} \left[\sum_{k=1}^K b_{m,k} x^2(k) \right] \\ &= \sum_{k=1}^K b_{m,k}^2 \mathbb{V}[x^2(k)] \\ &= 2 \sum_{k=1}^K b_{m,k}^2. \end{aligned} \quad (6.35)$$

Dans l'hypothèse de paramètres de signature η_m que l'on notera H_m , $\sqrt{E_{\eta_m}} \cdot \frac{x(k)}{\sqrt{\sigma_{\eta_m}^2(k) + \sigma_a^2}}$ suit une distribution normale et ainsi, $E_{\eta_m} \cdot \frac{x^2(k)}{\sigma_{\eta_m}^2(k) + \sigma_a^2} \sim \chi^2(1)$. On peut alors déduire la moyenne :

$$\begin{aligned} \mathbb{E} \left[\sum_{k=1}^K b_{l,k} x^2(k) | H_m \right] &= \mathbb{E} \left[\sum_{k=1}^K b_{l,k} \frac{\sigma_{\eta_m}^2(k) + \sigma_a^2}{E_{\eta_m}} \cdot \frac{E_{\eta_m} \cdot x^2(k)}{\sigma_{\eta_m}^2(k) + \sigma_a^2} \right] \\ &= \sum_{k=1}^K b_{l,k} \frac{\sigma_{\eta_m}^2(k) + \sigma_a^2}{E_{\eta_m}} \cdot \mathbb{E} \left[\frac{E_{\eta_m} \cdot x^2(k)}{\sigma_{\eta_m}^2(k) + \sigma_a^2} \right] \\ &= \sum_{k=1}^K b_{l,k} \frac{\sigma_{\eta_m}^2(k) + \sigma_a^2}{E_{\eta_m}} \end{aligned} \quad (6.36)$$

et la variance

$$\begin{aligned} \mathbb{V} \left[\sum_{k=1}^K b_{l,k} x^2(k) | H_m \right] &= \mathbb{V} \left[\sum_{k=1}^K b_{l,k} \frac{\sigma_{\eta_m}^2(k) + \sigma_a^2}{E_{\eta_m}} \cdot \frac{E_{\eta_m} \cdot x^2(k)}{\sigma_{\eta_m}^2(k) + \sigma_a^2} \right] \\ &= \sum_{k=1}^K \left(b_{l,k} \frac{\sigma_{\eta_m}^2(k) + \sigma_a^2}{E_{\eta_m}} \right)^2 \mathbb{V} \left[\frac{E_{\eta_m} \cdot x^2(k)}{\sigma_{\eta_m}^2(k) + \sigma_a^2} \right] \\ &= 2 \sum_{k=1}^K \left(b_{l,k} \frac{\sigma_{\eta_m}^2(k) + \sigma_a^2}{E_{\eta_m}} \right)^2. \end{aligned} \quad (6.37)$$

La figure 6.9 permet de visualiser les moyennes et variances théoriques des différentes signatures, pour différentes hypothèses quant à la signature du signal reçu $x(k)$. Cette visualisation avait été réalisée afin de vérifier s'il y avait une relation entre filtres bien ou mal appris et ces mesures, bien que ça n'ait pas été le cas.

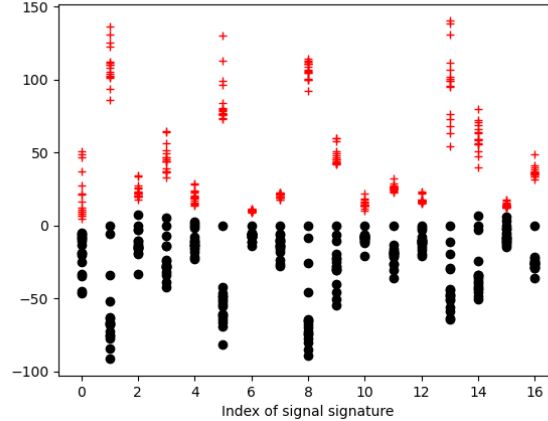


Figure 6.9: Moyennes théoriques (noir) et variances théoriques (rouge) des coefficients $\sum_{k=1}^K b_{l,k} x^2(k) | H_m \forall (l, m) \in \{0, 1, \dots, M\}^2$ pour un problème comportant $M = 16$ signatures différentes.

4.3 Règle de détection alternative multi-hypothèses

Le problème à deux hypothèses peut facilement être étendu au cas d'hypothèses multiples, où H_m , $m = 1, \dots, M$ correspond à l'hypothèse pour laquelle le signal de bruit d'hélice est $s_{\eta_m}(k)$, et H_0 correspond à l'hypothèse de bruit ambiant seul :

$$\begin{aligned} H_0 &: \{x(k) = n_a(k), k \in \llbracket K \rrbracket\}, \\ H_m &: \{x(k) = s_{\eta_m}(k) + n_a(k), k \in \llbracket K \rrbracket\}. \end{aligned} \quad (6.38)$$

Comme le stipule Zhang (2014), la tâche est ainsi de décider en faveur de l'une des hypothèses, compte tenu des mesures discrètes $x(k)$. Le rapport de vraisemblance utilisé pour les problèmes de détection binaire peut être directement utilisé pour écrire la fonction de vraisemblance permettant de décider entre les différentes hypothèses. Ceci peut être réalisé en construisant des paires de probabilités entre chacune des hypothèses de signaux impliquées et une hypothèse "catalystrice" H_0 comportant le bruit seul. Ainsi, en partant de ce postulat, les problèmes comportant plusieurs signaux peuvent être résolus directement en généralisant le cas où un unique signal peut être présent dans le bruit. Typiquement, la complexité de cette solutions croît alors linéairement avec le nombre de signaux à déterminer.

Pour une probabilité d'erreur minimale, et en supposant des probabilités a priori égales entre les différentes hypothèses, il faut alors comparer les quantités :

$$\frac{\Pr_m(\mathbf{x})}{\Pr_0(\mathbf{x})} = \frac{|\Sigma_0|^{1/2}}{|\Sigma_m|^{1/2}} \cdot e^{-\frac{1}{2} \mathbf{x}^T (\Sigma_m^{-1} - \Sigma_0^{-1}) \mathbf{x}} \quad (6.39)$$

Ainsi, pour prendre des décisions sur les maximums de vraisemblance, nous devons résoudre le problème de maximisation :

$$\max_m \Omega_m(\mathbf{x}) \quad (6.40)$$

avec $\Omega_m(\mathbf{x}) = \mathbf{x}^T(\Sigma_0^{-1} - \Sigma_m^{-1})\mathbf{x} - \ln|\Sigma_m| + \ln|\Sigma_0|$. Aussi, par la suite nous comparons les performances du détecteur de Bayes optimal basé sur l'estimateur $\Lambda^*(\mathbf{x})$, tiré du problème de classification binaire de bruit d'hélice (6.4), avec cette règle alternative tirée du problème multi-hypothèses :

$$\Lambda_{\max}(\mathbf{x}) = c_0 + \max_m \left(c_m \exp \left(\sum_{k=1}^K b_{m,k} x^2(k) \right) \right), \quad (6.41)$$

où les coefficients $b_{m,k}$, c_m et c_0 sont les mêmes que dans la sous-section précédente. Des figures 6.10 et 6.11, on peut constater que ces deux détecteurs ont des comportements et des performances assez similaires, ce qui pourrait impliquer que l'usage d'une agrégation par une opération "max" pourrait être envisagée alternativement dans nos architectures profondes.

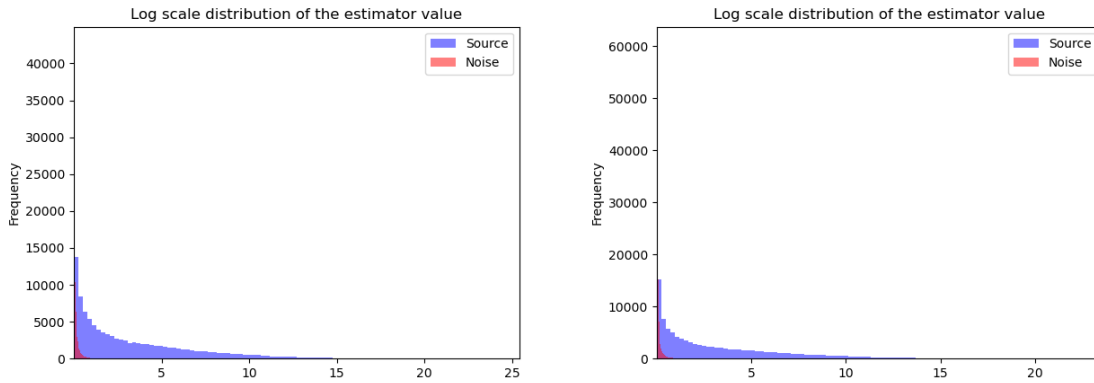


Figure 6.10: Distribution en échelle logarithmique de l'estimateur $\Lambda^*(\mathbf{x})$ (gauche) et $\Lambda_{\max}(\mathbf{x})$ (droite), pour des signaux pouvant suivre l'hypothèse H_0 ou H_1 et appartenant à 16 signatures différentes.

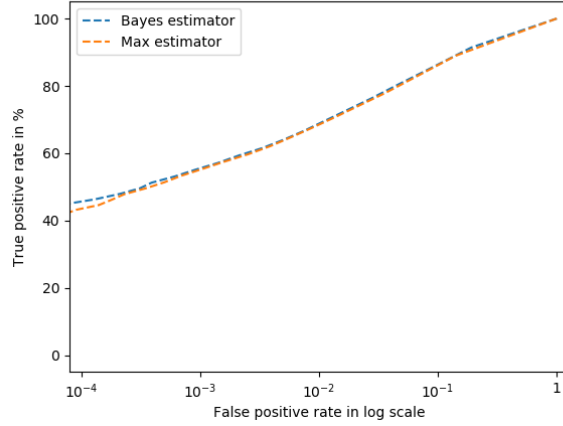


Figure 6.11: Courbes COR basées sur les estimateurs $\Lambda^*(\mathbf{x})$ et $\Lambda_{\max}(\mathbf{x})$ pour des signaux appartenant à 16 signatures différentes.

5 Apprentissage profond pour la classification de bruit d'hélice blanc avec signature inconnue

5.1 Architecture profonde bayésienne

Étant donné que la structure mathématique de l'estimateur optimal $\Lambda^*(\mathbf{x})$ (6.31) pour le problème de classification avec signature inconnue est la même que celle du problème de détection avec SNR inconnu (6.9), nous réemployons la même architecture explicable $f_\theta(\mathbf{x}) \in \mathcal{F}^*$ de structure

$$f_\theta(\mathbf{x}) = \lambda_0 + \sum_{m=1}^M \lambda_m \Phi_{e, \gamma_2} \left(\sum_{k=1}^K \varphi_{m,k} \Phi_{p_2, \gamma_1}(x(k)) \right). \quad (6.42)$$

La seule différence, en terme d'approximation tout du moins, sera que les paramètres $\lambda_0, \dots, \lambda_m$ et $\varphi_{m,k}$ jouent le rôle de coefficients c_0, \dots, c_m et $b_{m,k}$ différents par rapport à ceux du problème de détection avec SNR inconnu, présents dans l'équation (6.9). Similairement, l'implémentation sous forme de CNN de $f_\theta \in \mathcal{F}^*$ sera exactement la même que dans la sous-section 3.3.

5.2 Méthodologie pour les expérimentations

Comme discuté dans la sous-section 2.2 du chapitre précédant, les performances de notre réseau peuvent être évaluées à travers l'excès de risque $\mathcal{R}(f_\theta) - \mathcal{R}^*$. Dans la sous-section 3.2 nous avons démontré que notre réseau avait une erreur d'approximation $\min_\theta \mathcal{R}(f_\theta) - \mathcal{R}^* \leq \epsilon$, avec une profondeur de couche L augmentant polylogarithmiquement par rapport à ϵ^{-1} . En se basant sur une mesure de complexité telle que la dimension VC (Vapnik (1992)), le faible nombre de paramètres de notre modèle, par rapport aux architectures CNN classiques, devrait donc en faire une approximation efficace de $\delta^*(\mathbf{x})$. Les bonnes performances des réseaux profonds sont cependant souvent attribuées à leur capacité à bien généraliser malgré leur

nombre élevé de paramètres. En effet, comme nous l’avons vu dans le chapitre 3, les mesures de complexité récentes telles que celles proposés par Zhou & Feng (2018), Neyshabur et al. (2017), Li, Sun, Su, Suzuki & Huang (2020) tentent de montrer que l’efficacité en terme de nombre de paramètres $\dim(\theta)$, n’est pas le seul facteur à prendre en compte pour obtenir une faible erreur de généralisation et, par extension, un entraînement réussi.

Les performances de notre réseau sont ainsi comparées à des architectures profondes sur signaux acoustiques bruts, issues de l’état de l’art du chapitre 2, telles que les FCNNs présentés par Dai et al. (2017), qui contiennent par ailleurs bien plus de paramètres. Ces réseaux très profonds, pouvant aller jusqu’à 34 couches de poids, ne présentent cependant pas d’amélioration de performances de précision au delà de 18 couches, le surapprentissage semblant alors entraîner des pertes de performances importantes. Sang et al. (2018) suggèrent de résoudre ce problème en utilisant des CNN avec des couches récurrentes prenant en entrée les informations issus des couches convolutives précédentes. Dans l’architecture CRNN qui en résulte, la couche convolutive peut extraire des caractéristiques de haut niveau localement invariantes, tandis qu’une couche récurrente permet au modèle d’agréger les informations temporelles des caractéristiques extraites, à la manière de nos filtres proposés $\varphi_{m,k}$. Enfin, une dernière architecture de comparaison est obtenue en utilisant un réseau avec un spectre DEMON comme caractéristique d’entrée. Cependant, comme le spectre DEMON ne met en évidence que les raies du spectre fréquentiel de la signature pour assister un opérateur, un réseau peu profond ne devrait pas être suffisant pour obtenir de bons résultats avec une telle caractéristique d’entrée. Nous choisissons ainsi de réutiliser une architecture proposée par Lu et al. (2020) pour notre tâche. Leur architecture prédit la fréquence de l’arbre moteur (ω dans notre modèle) présente dans un signal, en utilisant un spectre DEMON d’échelle Mel convolué avec un filtre en peigne adapté. Une différence majeure cependant est que nous utilisons un spectre DEMON d’échelle Mel unidimensionnel, et donc une couche convolutive 1D, au lieu des couches 2D utilisées par Lu et al. (2020), puisque nous utilisons une même modulation pour l’ensemble du spectre du bruit de cavitation (4.5). D’autres architectures, telles qu’un CNN 2D qui prendrait des spectrogrammes en entrée, auraient pu être envisagées, mais cela impliquerait un choix de configuration à la fois pour l’architecture et les caractéristiques d’entrée, ce qui n’est pas l’objet de ce chapitre.

Pour nos expérimentations, afin d’évaluer le nombre d’exemples N requis pour assurer la généralisation, nous utiliserons une estimation de $\mathcal{R}(f_\theta)$ avec un ensemble de test indépendant \mathcal{S}_{test} et un ensemble de validation également indépendant \mathcal{S}_{val} . L’ensemble d’apprentissage \mathcal{S} contient jusqu’à $N_{max} = 8 \cdot 10^5$ signaux tirés de $M = 16$ signatures, tandis que \mathcal{S}_{test} et \mathcal{S}_{val} contiennent chacun $2 \cdot 10^5$ échantillons. La précision de nos modèles sera comparée aux précisions des modèles de comparaison et du test de Bayes optimal $\delta^*(\mathbf{x})$, à la fois sur les ensembles d’apprentissage et de test. Deux scénarios seront considérés pour la fréquence d’échantillonnage donnant des signaux de longueurs $K = 512$ et $K = 64$ échantillons. Dans les deux cas, comme la structure optimale du détecteur de Bayes et ses paramètres sont connus, la valeur \mathcal{R}^* est calculée le plus précisément possible par une méthode de Monte-Carlo. Les deux scénarios ont un SNR similaire pour H_1 ($\approx -0,2$ dB), bien que leur précisions optimales associées diffèrent considérablement.

CRNN (382 529)	FCNN (1 037 281)	$f_{\theta}(x)$ (10 515)
Entrée : 512 × 1 signal temporel $x(k)$		
Conv [32, 64]	Conv [32, 64]	Conv [9, 1] × 6
MaxPool : 2 × 2		
Conv [64, 8]	Conv [64, 8]	
MaxPool : 2 × 2		
Conv [128, 8]	Conv [64, 8]	
MaxPool : 2 × 2		
Conv [256, 8]	Conv [128, 8]	Conv [1, 1]
	MaxPool : 2 × 2	Conv [16, 512]
	Conv [128, 8]	Conv [9, 1] × 3
	MaxPool : 2 × 2	
	Conv [256, 8]	
	MaxPool : 2 × 2	
	Conv [256, 8]	Conv [1, 1]
BLSTM [16]	GlobalAveragePool	Flatten layer
Final dense layer		

Tableau 6.3: Tableau des différentes architectures utilisées pour la comparaison dans le cas $K = 512$, avec leurs titres associés en première ligne. Le nombre en dessous de l'architecture indique son nombre de paramètres pouvant être entraînés. La stride est toujours fixée à 1 et le nombre multiplié aux parenthèses est le nombre de fois que la couche est réutilisée. La couche BLSTM utilise un dropout de 0,1 comme dans la publication de [Sang et al. \(2018\)](#) et le nombre entre crochets indique le nombre de filtres.

5.3 Optimisation des modèles profonds

La figure 6.3 décrit, par le biais de couches Keras, les 3 différentes architectures profondes sur signaux brutes considérées dans le cas $K = 512$. Pour définir notre architecture d'approximation $\hat{\delta}_{\theta}$, nous utilisons l'équivalent de $W = 9$ neurones par couche avec $W = 9$ convolutions de taille 1 et choisissons $L_1 = 6$ et $L_2 = 3$ couches pour les modules d'approximation de fonction Φ_{p_2, γ_1} et Φ_{e, γ_2} respectivement, puisque ce sont les meilleures profondeurs trouvées sur l'ensemble de validation pour une "GridSearch" avec des profondeurs de couche proposées $L = \{1, 3, 6\}$. Notre réseau de neurones profond est comparé à une architecture FCNN tirée des travaux de [Dai et al. \(2017\)](#) et une architecture CRNN publiée par [Sang et al. \(2018\)](#) avec des fonctions d'activation ReLU. Ces architectures sont adaptées à partir de leurs structures initiales pour correspondre à la taille des données d'entrée utilisées dans notre problème.

L'architecture FCNN comprend une succession de 7 couches de convolution et d'opérations de max pooling, suivies d'une opération d'average pooling globale tandis que l'architecture CRNN a 4 couches de convolution et d'opérations de max pooling suivies d'une couche LSTM bidirectionnelle (BLSTM). Les deux architectures utilisent des noyaux de convolution de tailles 64 pour la première couche et des tailles de noyaux de 8 pour les couches suivantes. La septième couche de convolution dans l'architecture FCNN est appliquée à un ensemble de feature maps

CRNN (39841)	FCNN (346 273)	$f_{\theta}(\mathbf{x})$ (3 347)
Entrée : 64×1 signal temporel $x(k)$		
Conv [256, 16]	Conv [32, 16]	Conv [9, 1] $\times 6$
	MaxPool : 2×2	
	Conv [64, 8]	Conv [1, 1]
	MaxPool : 2×2	Conv [16, 64]
	Conv [128, 8]	Conv [9, 1] $\times 3$
	MaxPool : 2×2	
	Conv [256, 8]	Conv [1, 1]
BLSTM [16]	GlobalAveragePool	Flatten layer
Final dense layer		

Tableau 6.4: Tableau des différentes architectures utilisées pour la comparaison dans le cas $K = 64$, avec leurs titres associés en première ligne.

de dimension $(256, 8)$, ce qui signifie qu'elle agit sur l'ensemble de l'information temporelle, de la même manière que le font les couches LSTM de l'architecture CRNN.

Pour le cas $K = 64$, l'architecture FCNN utilisée est similaire mais ne comprend que 4 successions de couches convolutives et d'opérations max pooling, tandis que l'architecture CRNN n'a qu'une seule couche convolutive. Les deux architectures utilisent des noyaux de convolution de taille 16 pour la première couche.

Toutes les architectures suivent la même procédure d'optimisation lors de l'apprentissage, avec l'utilisation de l'optimiseur Adam et de l'entropie croisée binaire comme fonction de perte. Le critère d'arrêt est une absence d'amélioration de la perte de validation pendant 15 époques consécutives, ou au maximum après 100 époques d'entraînement. La contrainte "Maximum kernel" est utilisée pour toutes les couches convolutives afin de majorer les valeurs des poids des filtres convolutifs et nous appliquons une régularisation L^2 aux couches de sortie, de même qu'une normalisation par batch pour toutes les couches convolutives. Pour chaque valeur N considérée dans les affichages des figures 6.12 et 6.13, les performances des architectures sont estimées en calculant la moyenne des erreurs d'entraînement et de test parmi 5 entraînements différents. Un modèle avec une perte de validation supérieure à $1,2 \times$ la meilleure perte de validation, pour un N donné, est rejeté pour le calcul des performances.

Concernant les temps de calcul, en utilisant un GPU GeForce RTX 2080 Ti et un CPU Intel Core i9-9900K, une époque pour les signaux de taille $K = 512$, avec des jeux d'apprentissage et de validation de tailles $2 \cdot 10^5$, dure en moyenne 6,5 minutes pour le FCNN, 12 minutes pour le CRNN et 10 minutes pour $f_{\theta}(\mathbf{x})$.

5.4 Résultats numériques

Les figures 6.12 et 6.13 affichées ci-dessous correspondent respectivement aux scénarios avec une fréquence d'échantillonnage $f_s = 256$ Hz et $f_s = 32$ Hz. Les courbes tracées représentent la précision d'entraînement $1 - \mathcal{R}_N(f_{\theta})$ ainsi que la précision de test, estimation de la

précision réelle $1 - \mathcal{R}(f_\theta)$ de nos réseaux de neurones pour des jeux de données comportant dans les deux scénarios $M = 16$ signatures.

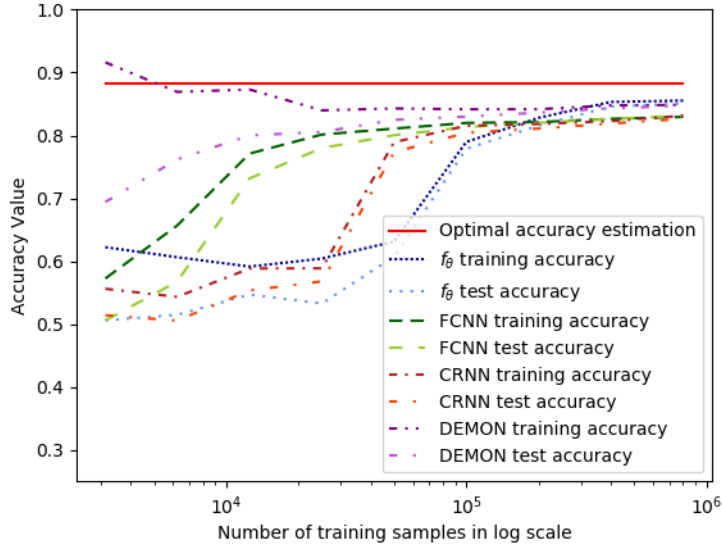


Figure 6.12: Précision d'entraînement et de test en fonction de la taille du jeu de donné associé et des signaux de $K = 512$ échantillons. La précision optimale correspond à un test de Monte-Carlo avec le détecteur (6.30).

En se basant sur la figure 6.12, la précision des différents modèles converge vers la précision optimale $1 - \mathcal{R}^*$ pour le scénario avec une fréquence d'échantillonnage plus élevée. Ce n'est cependant pas le cas pour le scénario présenté en figure 6.13, puisque même avec quasiment 10^6 exemples d'entraînement, la précision des différents modèles est loin d'être optimale, ce qui semble indiquer une fois de plus qu'un problème d'apprentissage plus difficile affecte grandement l'excès de risque et l'erreur de généralisation des modèles profonds employés. Dans les deux figures, du fait de formes de surapprentissage plus importantes sur de petits ensembles d'entraînement, la précision de l'entraînement de nos modèles diminue pour les valeurs N plus faibles. C'est en revanche un comportement moins probable pour des ensembles de données plus grands, car le critère d'arrêt est l'absence d'amélioration de la perte de validation.

Bien que le réseau FCNN surapprenne également sur la figure 6.12, il est toujours plus efficace pour de petits ensembles d'entraînement comparé à notre modèle, compte tenu de sa précision de test. Cette différence peut être due au fait que l'apprentissage des paramètres γ_1 et γ_2 , définissant les fonctions $\Phi_{p_2, \gamma_1}(t)$ et $\Phi_{e, \gamma_2}(t)$, en combinaison avec les coefficients $\varphi_{m, k}$ et λ_m , soit une tâche plus complexe pour de faibles valeurs N , comparé à l'apprentissage d'un modèle CNN $f_w(\mathbf{x})$ qui n'a besoin que de fournir de bonnes estimations $\delta_w(\mathbf{x}^{(i)}) \approx y^{(i)}$. En effet, les réseaux profonds avec de petits filtres convolutifs présentent généralement de bonnes performances de généralisation et ce, même quand le nombre de données est négligeable par rapport au nombre de paramètres. Cette propriété n'est cependant pas vérifiée ici pour la classe d'architectures profondes bayésiennes, bien qu'elle dispose de bien moins de paramètres

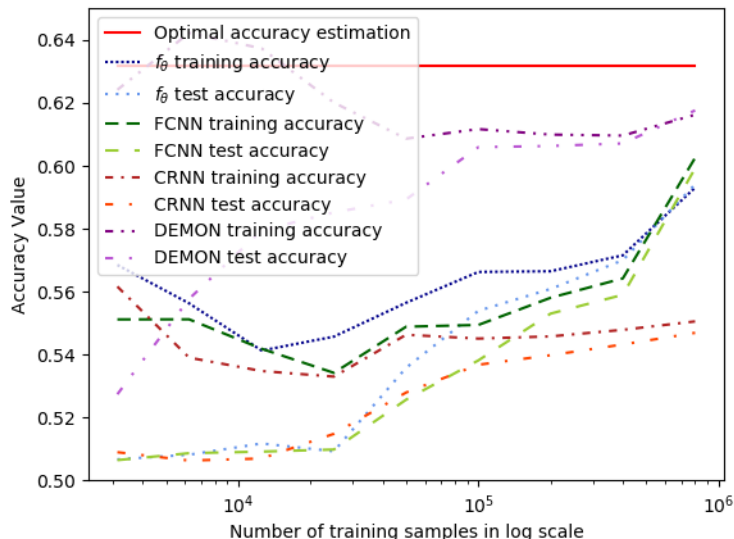


Figure 6.13: Précision d’entraînement et de test en fonction de la taille du jeu de données associé et des signaux de $K = 64$ échantillons. La précision optimale correspond à un test de Monte-Carlo avec le détecteur (6.30).

entraînables. Néanmoins, pour un ensemble de données plus important, notre architecture a de meilleures chances de converger vers un bon minimum local, ce qui peut impliquer un meilleur paysage pour la fonction de perte avec plus d’échantillons de la distribution \mathcal{D} pour notre architecture profonde bayésienne. Dans le scénario présenté par la figure 6.13 l’efficacité de notre méthode interprétable, en terme de nombre de paramètres et d’approximation, semble également se traduire en terme de performances de précision. Ces meilleures performances de notre modèle pour ce scénario, peuvent être liées au fait que la taille des filtres adaptés $\varphi_{m,k}$ devant être appris, est réduite dans ce scénario. En effet, cette taille de filtre correspond plus ou moins à une taille de filtre de convolution communément employée pour une première couche dans un réseau CNN 1D.

5.5 Résultats sur l’explicabilité

La figure 6.14 affiche certaines des plus proches correspondances trouvées entre les filtres appris $\varphi_{m_1,k}$ et les filtres optimaux $b_{m_2,k}$ parmi les M^2 combinaisons possibles, avec $M = 16$. Bien que les filtres semblent principalement apprendre la périodicité à partir des pics des formes d’ondes, aucun filtre appris ne correspond totalement à la signature optimale des filtres quand $M > 1$, bien que des similitudes frappantes puissent régulièrement être trouvées. En effet, les filtres appris $\varphi_{m_1,k}$ semblent plus agir comme une base de forme d’onde plutôt que comme l’ensemble des filtres optimaux théoriques. Certains filtres appris sont par exemple similaires au début d’un filtre optimal, tandis qu’un autre présente des similitudes avec la fin de ce même filtre optimal. On remarquera également que tous les filtres semblent être quelque peu bruités quand $K = 512$ et ce, quelle que soit la taille de l’ensemble de données. Cela peut être dû à la grande taille de nos filtres de convolution, un choix de conception évité dans les

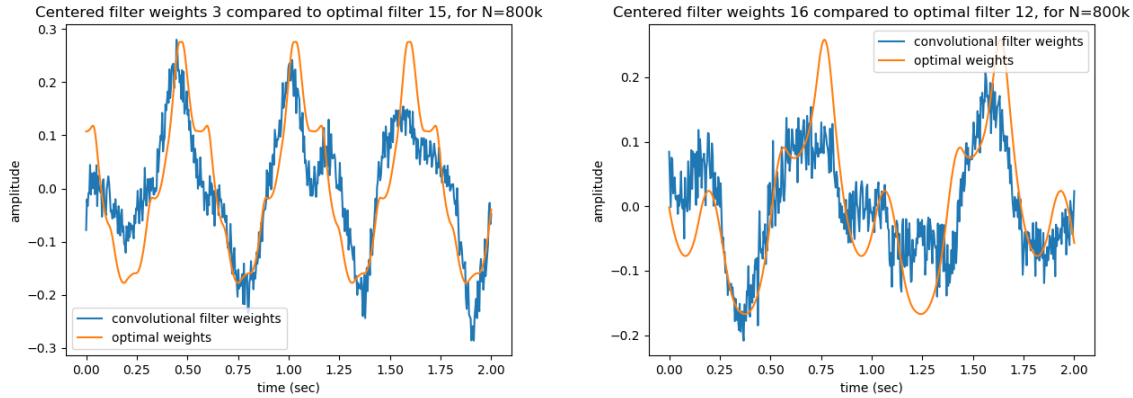


Figure 6.14: Exemples de fortes corrélations trouvées entre des filtres appris $\varphi_{m_1,k}$ et des filtres optimaux $b_{m_2,k}$ après l'apprentissage du modèle $f_\theta(\mathbf{x})$ avec $M = 16$ signatures ($K = 512$).

architectures CNN classiques en utilisant de petits noyaux convolutifs en combinaison avec des couches de pooling. La figure 6.5 montre cependant que le filtre adapté optimal $b_{m,k}$ peut être correctement appris lorsque peu de signatures M sont présentes dans le jeu d'entraînement, bien que ce filtre appris soit toujours bruité pour $K = 512$ échantillons.

On notera également qu'un important pourcentage des filtres appris, $\approx 20\%$ pour $M = 16$, apprennent des formes d'ondes quasiment plates voire nulles si on omet le bruit, comme en atteste la figure 6.15. On pourrait supposer que c'est un moyen pour l'architecture de généraliser plus facilement. En effet en rendant plusieurs filtres nuls, on arrive à une architecture équivalente possédant moins de paramètres entraînaibles et donc moins enclin à surapprendre.

La figure 6.16 montre des exemples de fonctions internes $\Phi_{p_2,\gamma_1}(t)$ et $\Phi_{e,\gamma_2}(t)$ apprises pendant l'entraînement d'un modèle $f_\theta(\mathbf{x})$ avec $N = 8 \cdot 10^5$ et $N = 1.25 \cdot 10^4$ exemples d'entraînement. Ces résultats sont très similaires à ceux de la sous-section 3.7, si ce n'est qu'avec plus de signatures M , il est intéressant de voir la distribution des points $\Phi_{e,\gamma_2}(t)$ réellement calculés par l'architecture $f_\theta(\mathbf{x})$ pour la prédiction. Pour la figure choisie à droite, on voit ainsi que l'architecture aurait plutôt appris une fonction proche de $e^{|t|}$, peut être dû au fait que certains filtres appris soient proches de l'opposé d'un des filtres optimaux. De même le filtre du milieu, comme dans le problème précédent, apprend une forme globalement désirée, mais localement incorrecte bien que cela n'impacte que très peu la performance de l'architecture. Il semble ainsi que l'initialisation des poids aléatoires peut faire converger l'architecture vers un grand nombre de minima locaux auxquels il est difficile d'échapper.

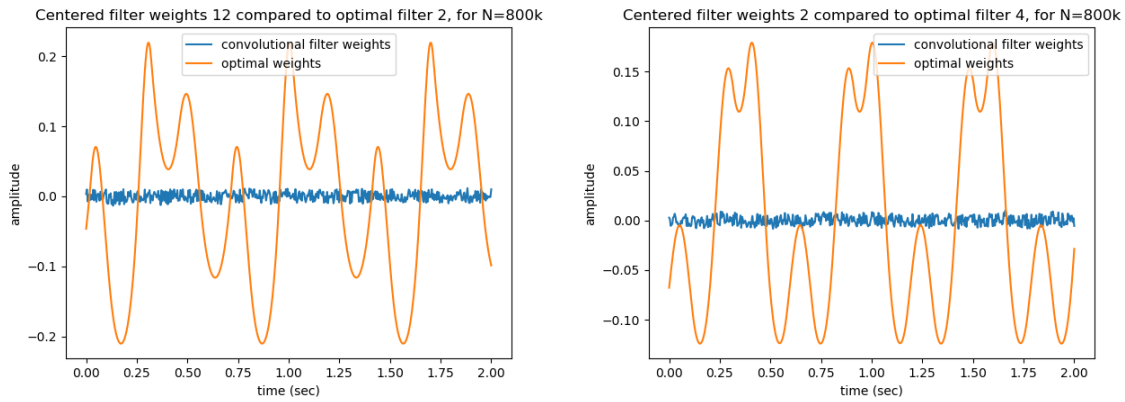


Figure 6.15: Exemples de filtres appris $\varphi_{m_1,k}$ proches de filtre nuls, comparés à des filtres optimaux $b_{m_2,k}$ après l'apprentissage du modèle $f_\theta(\mathbf{x})$ avec $M = 16$ signatures ($K = 512$).

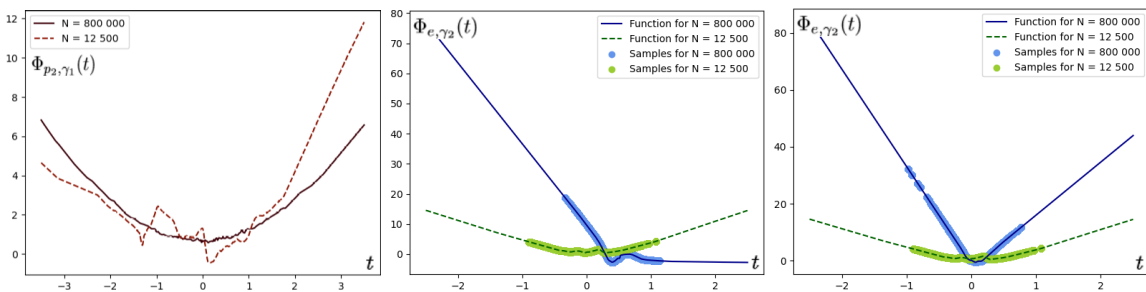


Figure 6.16: Exemples de fonction apprises Φ_{p_2, γ_1} à gauche et Φ_{e, γ_2} au milieu et à droite, pour différentes tailles de jeu d'entraînement. Les points affichés sur les fonctions représentent des valeurs présentes dans le jeu de données.

6 Conclusion

Dans ce chapitre, nous avons continué d'étudier le problème de classification binaire d'enregistrements de bruits de navires, mais cette fois-ci, pour des hypothèses composites. Nous avons une fois de plus proposé des architectures de réseaux de neurones permettant d'approximer efficacement, en termes de nombre de paramètres, la structure mathématique optimale du détecteur Bayésien, dans un cas multi-signatures et un cas avec des niveaux d'énergies inconnues.

Nous avons par la suite comparé nos réseaux Bayésiens à diverses autres architectures de réseaux profonds et, tandis que toutes les classes de réseaux permettaient de converger vers des performances optimales en termes de précision, notre classe bayésienne a également permis une convergence vers la structure du détecteur de Bayes, rendant notre architecture de réseau interprétable. Nous avons néanmoins constaté une plus grande efficacité des architectures profondes, communément employées dans l'état de l'art, quand une très grande quantité d'exemples d'entraînement n'était pas présent. Notre approche d'architectures bayésiennes ne semble ainsi être utile que lorsque énormément d'exemples d'entraînement sont disponibles. Sa capacité d'approximation de la structure mathématique du détecteur de Bayes en fait alors une architecture idéale pour la classification. Dans le chapitre suivant nous tenterons de proposer une alternative à l'utilisation de filtres de convolution larges pour l'apprentissage de longues dépendances temporelles, mais cela, en conservant une forme d'explicabilité, absente des architectures convolutives classiques.

Chapitre 7

Classification par Scattering Transform

1 Introduction

Dans ce chapitre, nous proposons d'utiliser la structure mathématique de la scattering transform, ou transformée de diffusion en français, pour implémenter diverses architecture CNNs explicables, réalisant des analyses multi échelles de signaux cyclostationnaires. Ce type d'architecture doit ainsi permettre d'apprendre des représentations internes avec très peu de paramètres d'entraînement, pour un apprentissage frugal. Sans réeffectuer un état de l'art, la section 2 discute de l'utilisation de représentations fixes pour la tâche de classification et des représentations internes au sein des architectures profondes. La section 3 décrit la structure mathématique ainsi que les propriétés de la scattering transform, tandis que la section 5 présente notre implémentation de modules de scattering transform basée sur l'apprentissage de filtres d'ondelettes.

2 Représentations pour la tâche de classification

2.1 Introduction

Les problèmes de classifications consistent généralement à approximer une fonction idéale $f(\mathbf{x})$ en construisant une distance capable de refléter les similarités entre signaux d'une même classe, tout en séparant des signaux de classes différentes. Dans le cas de signaux \mathbf{x} de grande dimensionnalité, il devient cependant difficile d'obtenir un échantillonnage uniforme d'un volume de dimension d car cela nécessite un nombre d'échantillons qui croît exponentiellement avec d , c'est la malédiction de la dimension. Nonobstant, il est possible d'approximer $f(\mathbf{x})$ en se basant sur peu d'échantillons, mais à condition que $f(\mathbf{x})$ possède des propriétés de régularité permettant de réduire la dimension des signaux dans un espace d'estimation sans pertes de séparabilité (cf. annexe A section 2). Les algorithmes d'apprentissage profond et en particulier les CNNs, reposent sur cette hypothèse sous-jacente de régularité. Ils définissent en effet des structures mathématiques qui imposent certaines régularités aux fonctions pouvant être apprises, comme le montre Yarotsky (2018) en généralisant le théorème d'approximation universelle aux feature maps invariants ou équivariantes par rapport à des représentations

linéaires de groupes. Il y parvient notamment en établissant un modèle de calcul, montrant que ces réseaux profonds sont complets, dans le sens où ils peuvent approximer toute feature map continue invariante ou équivariante. Leur réussite est ainsi souvent attribuée à leur capacité à contracter progressivement l'espace afin de réduire la dimension ou les variations de \mathbf{x} . En effet, comme souligné par Mallat (2016), les algorithmes d'apprentissage profond permettent de linéariser les transformations le long desquelles $f(\mathbf{x})$ reste quasi constante, tout en conservant la capacité de séparation des signaux. Oyallon (2017b) montre d'ailleurs expérimentalement sur le jeu d'image CIFAR, qu'au sein d'un CNN, chaque nouvelle couche, hormis la dernière, permet de progressivement rendre la représentation interne du CNN plus régulière. La spécification de la nature de ces régularités apprises est ainsi un enjeu important de l'apprentissage profond et un facteur de fiabilité supplémentaire.

Les signaux acoustiques ne pouvant cependant que rarement être considérés comme des variétés géométriques de faibles dimensions, leur classification a historiquement impliqué l'utilisation de représentations prédéfinies, permettant d'exploiter certaines de leurs régularités, comme mentionné dans le chapitre 2. Aussi, bien que l'utilisation d'architectures profondes sur signaux acoustiques bruts ait pu récemment montrer son utilité en terme de score de classification, il n'en demeure pas moins qu'en général ils ne retiennent pas l'explicabilité de méthodes de représentations prédéfinies à partir de bases orthogonales, telles que les spectrogrammes et scalogrammes présentés en annexe A section 4.

2.2 Représentation dans une base orthogonale

La transformée de Fourier, présentée dans la sous-section 3.1 du chapitre A, a une place particulière dans le traitement du signal acoustique grâce aux propriétés de la base orthogonale qu'elle définit. On peut cependant définir d'autres représentations à partir d'autres bases orthogonales que certains réseaux profonds pourraient potentiellement apprendre. Considérons un signal discret $s(k)$ et un ensemble de D signaux orthogonaux formant une base

$$\mathcal{B}_D = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_D\}, \quad (7.1)$$

avec $\mathbf{c}_i = [c_i(1), \dots, c_i(K)]$. Un signal $s(k)$ peut être approximé par une combinaison linéaire des signaux orthogonaux $c_i(k)$, tel que :

$$\tilde{s}(k) = \sum_{i=1}^D w_i \cdot c_i(k) = \mathbf{w}^T \bar{\mathbf{c}}(k), \quad (7.2)$$

avec $\bar{\mathbf{c}}(k) = [c_1(k), \dots, c_D(k)]$.

Définition : Signaux orthogonaux

Deux signaux discrets $c_i(k)$ et $c_j(k)$ sont considérés comme étant orthogonaux, noté $c_i(k) \perp c_j(k)$, si et seulement si :

$$\frac{1}{K} \sum_{k=1}^K c_i(k) \cdot c_j^*(k) = \begin{cases} E_{c_i} & \text{si } i = j, \\ 0 & \text{sinon,} \end{cases} \quad (7.3)$$

avec $E_{c_i} = \frac{1}{K} \sum_{k=1}^K c_i(k)^2$ l'énergie du signal $c_i(k)$, qui vaut 1 si les signaux sont ortho-normaux.

La meilleure approximation, dans le sens où les coefficients \mathbf{w} minimisent l'énergie E_ϵ du signal d'erreur $\epsilon(k) = s(k) - \tilde{s}(k)$, est celle qui utilise les coefficients

$$\mathbf{w} = \frac{1}{E_c} \sum_{k=1}^K s(k) \cdot \bar{c}(k)^*, \quad (7.4)$$

où $\bar{E}_c = [E_{c_1}, \dots, E_{c_D}]^T$. L'énergie du signal d'erreur $\epsilon(k)$ est alors minimale quand l'ensemble des signaux $\{\epsilon(k), c_1(k), c_2(k), \dots, c_D(k)\}$ est orthogonal i.e, si $\epsilon(k) \perp c_i(k), \forall i \in [1, D]$:

$$\sum_{k=1}^K \bar{c}(k)^* (s(k) - \mathbf{w}^T \bar{c}(k)) = 0. \quad (7.5)$$

Qui plus est, si $\tilde{s}(k) = s(k)$, alors l'ensemble de signaux orthogonaux \mathcal{B}_D est un ensemble complet de signaux orthogonaux. Ainsi quand $\epsilon(k) = 0$, la connaissance du vecteur \mathbf{w} est équivalente au fait de connaître le signal $s(k)$. Dans ce cas si $\dim \mathbf{w} = D < K$, la base orthogonale peut présenter un avantage en terme de paramètres à apprendre lors de l'apprentissage.

Plusieurs travaux ont néanmoins montré qu'utiliser des bases orthogonales n'était pas optimal pour la classification et ce, souvent en avançant que de telles bases requièrent des dimensions de représentations élevées. Une telle représentation est en effet inadéquate pour des problèmes de classification où les données sont séparables en prenant en compte un nombre réduit de dimensions. C'est peut-être ainsi pour cette raison que les premières couches de réseaux CNN 1D apprennent naturellement des ensembles de filtres différents de peu de bases de formes d'ondes en échelle Mel.

Remarque

L'approximation dans une base orthogonale peut également être généralisée pour le cas de M signaux comme le montre [Manikas \(2009\)](#) pour les "M-ary systems". Un signal $s_m(k)$ peut ainsi être représenté par un point $\mathbf{w}^{(m)}$ dans un espace Euclidien de dimension $D \leq M$. L'ensemble des points spécifiés par la matrice $W = [\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(M)}]$ est alors connu sous le nom de constellation de signaux.

3 Description de la scattering transform

3.1 Introduction

La scattering transform ou transformée de diffusion, est une représentation du spectre de modulation du signal, introduite pour la première fois par [Mallat \(2012\)](#). C'est une représentation localement invariante aux transformations géométriques et permettant de linéariser les difféomorphismes. L'analyse du signal, qui est effectuée à plusieurs ordres et séparément sur différentes échelles temporelles, est obtenue par des cascades de filtres de convolution et d'opérations non linéaires contractantes, similaires à des transformées en ondelettes discrètes.

Pour la tâche de classification, cette représentation peut notamment devenir invariante à certains groupes de transformations, en effaçant les informations de localisations qui y sont présentes. Cela peut ainsi aboutir à une invariance aux translations et aux rotations (pour les signaux 2D ou 3D), aux transpositions fréquentielles (pour les signaux 1D grâce à une analyse cepstrale) comme le font [Andén & Mallat \(2014\)](#) ou encore aux changements d'échelle. Ces transformations étant souvent non impactantes dans les tâches de classification, les retirer de la représentation finale, permet de réduire l'impact de certaines variabilités inutiles au sein d'un jeu de données, tout en conservant les caractéristiques discriminantes des données. La scattering transform génère cependant des représentations de très grandes dimensionnalités afin de trouver un espace dans lequel les données sont séparables. Aussi, pour des domaines d'applications tels que la classification d'images ou d'objets 3d, il est également courant d'employer des techniques de réduction de dimension en sortie de scattering transform, comme le font [Zarka et al. \(2019\)](#) grâce à une technique de codage parcimonieux par entraînement stochastique.

3.2 Transformée d'ondelette analytique

En se référant à l'article de [Andén & Mallat \(2014\)](#), une ondelette analytique est définie comme étant une ondelette complexe dont la transformée de Fourier est nulle aux fréquences négatives. Aussi, on considérera une ondelette analytique dans le domaine spectral $\hat{\psi}$, dont les parties réelles et imaginaires sont en quadrature de phase et ayant une fréquence centrale normalisée à 1 Hz :

$$\begin{cases} \hat{\psi}(0) = 0, \\ \hat{\psi}(\omega) \approx 0, \text{ pour } \omega < 0 \\ \psi_\lambda(t) = \lambda\psi(\lambda t) \iff \hat{\psi}_\lambda(\omega) = \hat{\psi}\left(\frac{\omega}{\lambda}\right) \end{cases} \quad (7.6)$$

et une largeur de bande de l'ordre de $1/Q$ tel que $\hat{\psi}(\omega) \approx 0$ pour $\omega \notin [1 - \frac{1}{Q}, 1 + \frac{1}{Q}]$, Q étant le nombre d'ondelettes par octave tel que $\lambda = 2^{\frac{q}{Q}}$, $q \in \mathbb{Z}$. Avec cette définition, $\hat{\psi}_\lambda(\omega)$ a une fréquence centrée sur λ et une largeur de bande de l'ordre de λ/Q . L'énergie de $\psi_\lambda(t)$ est centrée autour de 0 sur un intervalle de taille $2\pi Q/\lambda$, λ étant supérieur à $2\pi Q/T$, afin que cet intervalle soit plus petit que T .

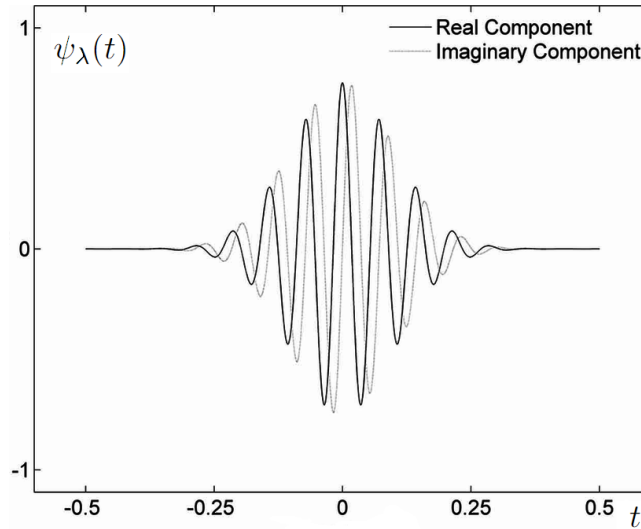


Figure 7.1: Parties réelles et imaginaires d'une ondelette Morlet. Tiré du livre de Mallat (1999).

Définition : Ondelettes analytiques

On peut définir une ondelette analytique avec une résolution d'octave Q tel qu'il suit :

$$\psi(t) = e^{it}\theta(t) \iff \hat{\psi}(\omega) = \hat{\theta}(\omega - 1), \quad (7.7)$$

avec $\hat{\psi}$ la fonction de transfert d'un filtre passe-bas de largeur d'onde de l'ordre de $\frac{1}{Q}$. En particulier, si θ est Gaussien, on définit ainsi une ondelette ψ nommée Morlet.

Si $\hat{\theta}(-1) \neq 0$, on peut définir $\hat{\psi}(\omega) = \hat{\theta}(\omega - 1) - \hat{\theta}(\omega) \frac{\hat{\theta}(-1)}{\hat{\theta}(0)}$, afin d'obtenir $\hat{\theta}(0) = 0$.

L'ensemble des ondelettes devant couvrir l'intégralité de l'axe fréquentiel avec leur bande passante, il est également nécessaire de disposer de fréquences centrales permettant de couvrir la basse fréquence. Ainsi, pour les fréquences $\omega \in [0, \frac{2\pi Q}{T}]$, il est possible d'utiliser $Q - 1$ filtres ϕ de largeur de bande constante $2\pi/T$, que Andén & Mallat (2014) considèrent tout de même comme étant des ondelettes.

La transformée en ondelettes, notée $\mathcal{W}[\mathbf{x}]$, calcule les convolutions de \mathbf{x} avec les passe-bas ϕ de bande passante $2\pi/T$, ainsi que les convolutions avec l'ensemble des ondelettes haute fréquence ψ_λ pour $\lambda \in \Lambda$, l'ensemble des fréquences centrales.

$$\mathcal{W}[\mathbf{x}] = (\mathbf{x} * \phi(t), \mathbf{x} * \psi_\lambda(t))_{t \in \mathbb{R}, \lambda \in \Lambda}. \quad (7.8)$$

Les ondelettes ψ et filtres passe-bas ϕ sont désignés pour construire des filtres couvrant l'ensemble de l'axe fréquentiel, ainsi, l'énergie de $\mathcal{W}[\mathbf{x}]$ à la fréquence ω ,

$$A(\omega) = |\hat{\phi}(\omega)|^2 + \frac{1}{2} \sum_{\lambda \in \Lambda} \left(|\hat{\psi}_\lambda(\omega)|^2 + |\hat{\psi}_\lambda(-\omega)|^2 \right), \quad (7.9)$$

satisfait pour tout $\omega \in \mathbb{R}$,

$$1 - \alpha \leq A(\omega) \leq 1, \quad (7.10)$$

avec α une constante < 1 .

3.3 Implémentation de la scattering transform

La scattering transform est définie comme un réseau de neurones convolutif à valeur complexe, où les filtres de chaque couche sont des ondelettes fixes ne disposant que d'un seul canal complexe. Aussi, du fait que les feature maps produites à chaque couche sont toujours de dimension 1, les filtres de la scattering transform n'effectuent pas de combinaisons d'information sur différents canaux lors de la convolution, ce qui est habituellement le cas dans les CNNs où le nombre de filtres par couche F est souvent supérieur à 1. Chaque couche effectue une partie de la transformation en ondelettes, décomposant ainsi les représentations du signal d'entrée sur différentes échelles temporelles et utilise le module complexe comme non-linéarité pour obtenir des invariances à la phase et aux translations pour ces mêmes échelles. Cette séparation des différentes échelles temporelles par les ondelettes permet alors d'obtenir une stabilité aux déformations du signal d'origine, comme nous le verrons plus tard.

Considérons des familles d'ondelettes $\{\psi_{\lambda_i}\}_{\lambda_i \in \Lambda_i}$, tel qu'il existe une valeur ϵ satisfaisant :

$$1 - \epsilon \leq \sum_{\lambda_i \in \Lambda_i} \left| \widehat{\psi}_{\lambda_i}(\omega) \right|^2 \leq 1. \quad (7.11)$$

On définit $\mathcal{S}[\mathbf{x}] = (\mathcal{S}_m[\mathbf{x}])_{0 \leq m \leq l}$, la scattering transform du signal \mathbf{x} allant jusqu'à l'ordre l , avec les familles d'ondelettes $(\Lambda_1, \Lambda_2, \dots, \Lambda_l)$ tel que :

$$\begin{aligned} |\mathcal{W}[\mathbf{x}] &= (\mathbf{x} * \phi, |\mathbf{x} * \psi_{\lambda_1}|)_{\lambda_1 \in \Lambda_1}, \\ \mathcal{S}_1[\mathbf{x}](t, \lambda_1) &= |\mathbf{x} * \psi_{\lambda_1}| * \phi(t), \\ |\mathcal{W}[|\mathbf{x} * \psi_{\lambda_1}|] &= (|\mathbf{x} * \psi_{\lambda_1}| * \phi, ||\mathbf{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}|)_{\lambda_2 \in \Lambda_2}, \\ \mathcal{S}_2[\mathbf{x}](t, \lambda_1, \lambda_2) &= ||\mathbf{x} * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi(t), \\ \mathcal{S}_m[\mathbf{x}](t, \lambda_1, \dots, \lambda_m) &= |\dots|\mathbf{x} * \psi_{\lambda_1}| \dots * \psi_{\lambda_m}| * \phi(t). \end{aligned} \quad (7.12)$$

Bien qu'une l'implémentation naturelle de l'algorithme serait de calculer les coefficients de diffusion couche par couche, dans le module Python Kymatio proposé par [Andreux et al. \(2020\)](#) avec pyTorch en backend, l'arbre de transformation est plutôt traversé en profondeur. Cela limite entre autres l'utilisation de la mémoire vive qui peut vite devenir critique et rend ainsi la mise en œuvre plus adaptée à l'utilisation de GPUs quand le nombre de coefficients de diffusion devrait exploser, notamment pour les images 2D. Dans Kymatio, le calcul de la transformée s'effectue dans le domaine fréquentiel grâce à la connaissance de la transformée de Fourier des ondelettes et aux fonctions Python `fft` et `ifft`. [Andén & Mallat \(2014\)](#) utilisent également un pas de taille égale à la moitié de la largeur de la fenêtre du filtre ϕ pour calculer $\mathbf{x} * \phi$, ϕ étant toujours un filtre passe bas de fréquence centrale nulle. $|\mathbf{x} * \psi_{\lambda_1}|$ est également sous-échantillonné à un taux égal à 2 fois sa bande passante, ainsi si \mathbf{x} dispose de K échantillons temporels, $2K$ coefficients d'ondelettes du premier ordre seront calculés en tout, ce qui permet de réduire la charge de calcul sans être affecté par un repliement spectral.

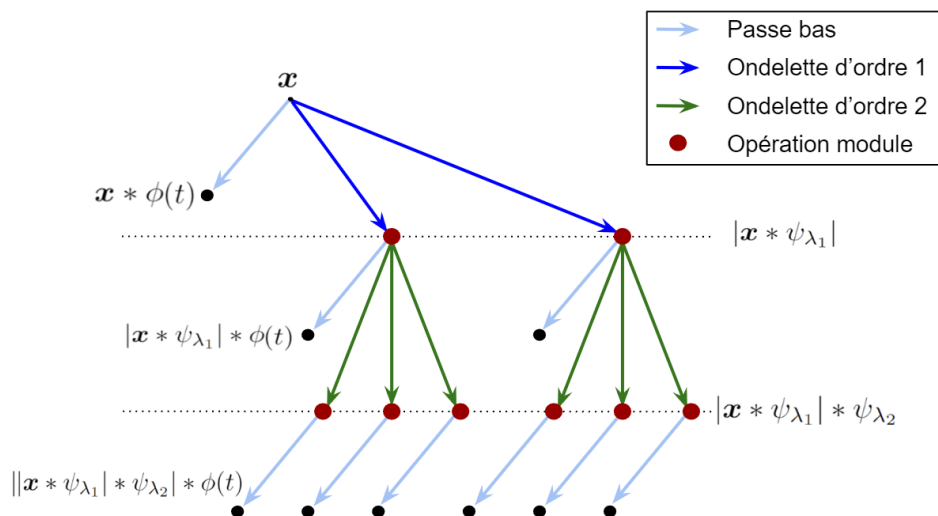


Figure 7.2: Schéma de la scattering transform d'ordre 2. Adapté de l'article de [Andén & Mallat \(2014\)](#).

Dans l'implémentation de Kymatio, si l'entrée \mathbf{x} est un enregistrement de taille $(B, 1, K)$, la sortie de la scattering transform 1D est de taille $(B, P, K/2^J)$, où B est la taille de la batch et P est le nombre de coefficients de diffusion et 2^J est l'échelle maximale des filtres de la transformation. Le plus grand filtre est en effet contenu sur un intervalle temporel discret de taille $T = 2^J$. La valeur de P dépend de l'ordre maximal de la transformée de diffusion et des paramètres Q et J . Pour l'ordre 2, elle est approximativement proportionnelle à $1 + J \times Q + J(J - 1)Q/2$.

3.4 Similarités avec le spectre fréquentiel Mel

Soit $|\hat{x}(t, \omega)|$ un spectrogramme obtenu avec des fenêtres ϕ de durée T , tel que $\int \phi(u) du = 1$ et

$$|\hat{x}(t, \omega)| = \left| \int x(u) \phi(u - t) e^{-i\omega u} du \right|. \quad (7.13)$$

Dans l'article de [Andén & Mallat \(2014\)](#) l'utilisation du spectre fréquentiel Mel consiste à moyenner l'énergie d'un spectrogramme par des filtres passe-bandes $\hat{\psi}_\lambda$ distribués en échelle Mel/logarithmique avec des fréquences centrales λ et des bandes passantes de l'ordre de $\frac{\lambda}{Q}$ pour les hautes fréquences et $\frac{2\pi}{T}$ à basses fréquences. En notant $\hat{x}_t(\omega) = \hat{x}(t, \omega)$, on calcule le spectrogramme Mel tel que :

$$M_x(t, \lambda) = \frac{1}{2\pi} \int |\hat{x}_t(\omega)|^2 \cdot \left| \hat{\psi}_\lambda(\omega) \right|^2 d\omega. \quad (7.14)$$

La formule de Plancherel donne alors :

$$\begin{aligned} M_x(t, \lambda) &= \int |x_t * \psi_\lambda(v)|^2 dv \\ &= \int \left| \int x(u) \phi(u - t) \psi_\lambda(v - u) du \right|^2 dv. \end{aligned} \quad (7.15)$$

De plus si $\lambda \gg Q/T$, $\phi(t)$ est approximativement constant sur le support de $\psi_\lambda(t)$, on a

$$\phi(u-t)\psi_\lambda(v-u) \approx \phi(v-t)\psi_\lambda(v-u) \quad (7.16)$$

et ainsi :

$$\begin{aligned} M_x(t, \lambda) &\approx \int \left| \int x(u)\psi_\lambda(v-u)du \right|^2 \cdot |\phi(v-t)|^2 dv \\ &= |\mathbf{x} * \psi_\lambda|^2 * |\phi|^2. \end{aligned} \quad (7.17)$$

Bien que la transformée de Fourier soit localement invariante aux décalages temporels :

$$|\widehat{x}(t, \omega)| = |\widehat{x}_c(t, \omega)| \text{ avec } x_c(t) = x(t-c), \quad (7.18)$$

si $|c| \ll T$, elle n'est pas pour autant stable aux déformations temporelles, car la dilatation d'une onde sinusoïdale produit une nouvelle onde sinusoïdale de fréquence différente, orthogonale à l'originale. Aussi, pour un signal $x(t - \tau(t))$, les hautes fréquences seront sévèrement déformées par de petites déformations. En prenant $\tau(t) = \epsilon t$ pour $0 < \epsilon \ll 1$ et ainsi $\tau'(t) = \epsilon$, la condition de continuité de Lipschitz pour une norme temps-fréquence $\|\cdot\|$, permettant de vérifier la stabilité aux déformations temporelles, requiert que

$$\|\widehat{\mathbf{x}} - \widehat{\mathbf{x}}_c\| \leq C\epsilon\|\mathbf{x}\|. \quad (7.19)$$

Aussi, puisque la transformée de Fourier de $x_\tau(t) = x((1-\epsilon)t)$ est

$$\widehat{x}_\tau(\omega) = (1-\epsilon)^{-1} \widehat{x}((1-\epsilon)^{-1}\omega), \quad (7.20)$$

la dilatation décale une fréquence ω_0 par $\epsilon|\omega_0|$, ce qui aboutit à l'instabilité aux déformations temporelles de cette représentation.

Cette propriété implique ainsi une absence de stabilité de l'axe fréquentiel du spectrogramme, qui exigerait que de petites déformations du signal produisent de petites différences avec la représentation initiale, par le biais d'une mesure de distance basée sur une norme Euclidienne, ce qui est souvent une propriété importante pour la classification, notamment les convolutions des CNNs 2D. En revanche, le filtre $\widehat{\psi}_\lambda(\omega)$ couvrant la fréquence ω_0 a une largeur de bande de l'ordre de $\lambda/Q \approx |\omega_0|/Q$, ainsi l'erreur relative après moyennage par $|\widehat{\psi}|^2$ est de l'ordre de ϵQ . Le moyennage étale ainsi les zones de définitions des fréquences (réduction de la résolution) proportionnellement à leur valeur. Ainsi, afin que les dilatations soient perçues comme étant linéaires en échelle Mel sur l'axe des fréquences, on s'oblige à perdre de l'information à haute fréquence. Qui plus-est, le moyennage fréquentiel du spectrogramme est équivalent à un moyennage temporel de la banque de filtre de sortie $|\mathbf{x} * \psi_\lambda|^2$, ϕ agissant alors comme un filtre passe-bas afin d'assurer que la représentation en sortie soit localement invariante à des décalages temporels à une échelle $< T$.

4 Propriétés de la scattering transform

4.1 Stabilité aux déformations temporelles

Dans la sous-section 3.4 était présentée la propriété de stabilité aux déformations temporelles des spectrogrammes en échelle Mel. La scattering transform satisfait ainsi la condition

de continuité de Lipschitz, puisque les ondelettes hautes fréquences sont stables aux déformations temporelles. Ainsi, en prenant une déformation restant petite par rapport à la fenêtre ϕ de taille T , $x_\tau(t) = x(t - \tau(t))$ avec $|\tau'(t)| < 1$ ainsi que $\sup_t |\tau(t)| \ll T$ et en notant

$$\|\mathcal{S}[\mathbf{x}]\|^2 = \sum_{m=0}^l \|\mathcal{S}_m[\mathbf{x}]\|^2, \quad (7.21)$$

on obtient que :

$$\|\mathcal{S}[\mathbf{x}_\tau] - \mathcal{S}[\mathbf{x}]\| \leq C \sup_t |\tau'(t)| \cdot \|\mathbf{x}\|. \quad (7.22)$$

Cette propriété permet entre autres d'utiliser des opérateurs linéaires pour calculer des descripteurs invariants à de petites déformations temporelles.

4.2 Conservation d'énergie et contraction

L'analyse par scattering transform a également pour propriété d'être une application contractante. En effet, puisque la scattering transform est calculée grâce à des cascades de modules complexes et des transformations en ondelettes contractantes $|\mathcal{W}|[\mathbf{x}]$, il en résulte que l'ensemble du réseau est contractif :

$$\|\mathcal{S}[\mathbf{x}] - \mathcal{S}[\mathbf{x}']\| \leq \|\mathbf{x} - \mathbf{x}'\|, \quad (7.23)$$

où $\|\cdot\|$ est une norme Euclidienne. Qui plus est, si $\alpha = 0$ dans l'équation (7.10), $\|\mathcal{S}[\mathbf{x}]\|$ préserve la norme du signal. Une conséquence de cette propriété est la réduction de la variance et une stabilité aux bruits additifs de la représentation.

4.3 Mesure d'intervalle de fréquence à partir d'interférence

Une transformation en ondelettes a généralement une résolution fréquentielle inférieure à celle d'une transformée de Fourier pour les hautes fréquences. Il est néanmoins possible de mesurer avec précision les intervalles de fréquence entre les harmoniques par le biais des coefficients de diffusion du second ordre. En effet, si un signal \mathbf{x} est composé de deux fréquences comprises dans le support de $\hat{\psi}_{\lambda_1}$, on a $\mathbf{x} * \psi_{\lambda_1}(t) = \alpha_1 e^{i\xi_1 t} + \alpha_2 e^{i\xi_2 t}$, et ainsi

$$|\mathbf{x} * \psi_{\lambda_1}(t)| \approx \sqrt{|\alpha_1|^2 + |\alpha_2|^2} + \frac{|\alpha_1 \alpha_2|}{\sqrt{|\alpha_1|^2 + |\alpha_2|^2}} \cos(\xi_1 - \xi_2)t. \quad (7.24)$$

Si ϕ a un support $T \gg |\xi_1 - \xi_2|^{-1}$, le moyennage donne alors $\mathcal{S}_1[\mathbf{x}](t, \lambda_1) \approx \sqrt{|\alpha_1|^2 + |\alpha_2|^2}$ et ainsi, les coefficients de diffusion d'ordre 2 normalisés d'expression

$$\bar{\mathcal{S}}_2[\mathbf{x}](t, \lambda_1, \lambda_2) = \frac{\mathcal{S}_2[\mathbf{x}](t, \lambda_1, \lambda_2)}{\mathcal{S}_1[\mathbf{x}](t, \lambda_1) + \epsilon}, \quad (7.25)$$

donnent pour notre cas avec un epsilon petit :

$$\bar{\mathcal{S}}_2[\mathbf{x}](t, \lambda_1, \lambda_2) \approx \frac{|\alpha_1 \alpha_2|}{|\alpha_1|^2 + |\alpha_2|^2} |\hat{\psi}_{\lambda_2}(\xi_1 - \xi_2)| \quad (7.26)$$

Ces coefficients du second ordre normalisés ne sont donc pas négligeables quand λ_2 est de l'ordre de grandeur de l'intervalle $|\xi_1 - \xi_2|$. Ainsi, bien que la première famille d'ondelettes

$\widehat{\psi}_{\lambda_1}$ n'a pas suffisamment de résolution pour distinguer les fréquences ξ_1 et ξ_2 , les coefficients du second ordre permettent de détecter leur présence et de mesurer précisément l'intervalle $|\xi_1 - \xi_2|$.

Comme dans la perception audio, les coefficients de diffusion peuvent mesurer avec précision les intervalles de fréquence mais pas leur localisation. Les coefficients de diffusion du second ordre normalisés (7.26) ne sont importants que si α_1 et α_2 ont le même ordre de grandeur. Comme le précisent [Andén & Mallat \(2014\)](#), ceci est conforme à la perception auditive, où un intervalle de fréquence n'est perçu que lorsque les deux composants de fréquence ont des amplitudes comparables. Qui plus est, si la résolution de $\widehat{\psi}_{\lambda_2}$ est insuffisante pour distinguer deux intervalles $|\xi_1 - \xi_2|$ et $|\xi_3 - \xi_4|$ alors, en suivant ce même principe, les coefficients de dispersion du troisième ordre peuvent possiblement mesurer cet intervalle d'intervalles. Cette propriété est également rependue dans d'autres types d'analyses fréquentielles du second ordre et notamment dans l'analyse DEMON et elle implique ainsi que les coefficients du second d'ordre ne détectent pas uniquement la modulation en amplitude, mais contiennent aussi une information qui peut venir camoufler une information classifiante.

4.4 Analyse du spectre de modulation d'amplitude

Dans leur article, [Andén & Mallat \(2014\)](#) proposent d'analyser les caractéristiques extraites par une scattering transform d'ordre 2, pour un son $x(t)$ résultant d'une excitation assimilée à une parole :

$$e(t) = \frac{2\pi}{\xi} \sum_n \delta\left(t - \frac{2n\pi}{\xi}\right) = \sum_n e^{ik\xi t}, \quad (7.27)$$

filtrée par un filtre de réponse impulsionnelle $h(t)$ et modulée en amplitude par $a(t)$ pour donner

$$x(t) = a(t)(e * h)(t). \quad (7.28)$$

La réponse impulsionnelle $h(t)$ doit être très courte par rapport à l'intervalle de variation minimum du terme de modulation $(\sup_t |a'(t)|)^{-1}$ et sera plus petite que $\frac{2\pi}{\xi}$. Nous considérons une ondelette ψ_{λ_1} dont le support temporel est court par rapport à $(\sup_t |a'(t)|)^{-1}$, i.e. $a(t)$ est presque constante sur l'intervalle temporel de ψ_{λ_1} et également court par rapport à l'intervalle de moyennage T de $\phi(t)$. La bande passante fréquentielle λ_1/Q_1 est plus petite que ξ ainsi que le plus petit intervalle de variation de \widehat{h} , i.e. $\widehat{h}(\omega)$ est presque constante sur l'intervalle de définition de $\widehat{\psi}_{\lambda_1}$. Ces conditions sont alors remplies si

$$\left(\int |t||h(t)|dt\right)^{-1} \gg \frac{\lambda_1}{Q_1} \gg \sup_t |a'(t)|. \quad (7.29)$$

Les coefficients du premier ordre sont alors proportionnels à l'enveloppe spectrale $|\widehat{h}(\lambda_1)|$ à condition que $\widehat{\psi}_{\lambda_1}$ couvre au plus une harmonique dont la fréquence approximative $k\xi$ est proche de λ_1 :

$$\begin{aligned} |\mathbf{x} * \psi_{\lambda_1}| &\approx |\widehat{h}(\lambda_1)| |e * \psi_{\lambda_1}| \cdot a(t), \\ &\approx |\widehat{h}(\lambda_1)| |\widehat{\psi}_{\lambda_1}(k\xi)| \cdot a(t) \end{aligned} \quad (7.30)$$

et ainsi

$$\mathcal{S}_1[\mathbf{x}](t, \lambda_1) \approx |\widehat{h}(\lambda_1)| |\widehat{\psi}_{\lambda_1}(k\xi)| \cdot a * \phi(t). \quad (7.31)$$

Après la normalisation des coefficients de diffusion du premier ordre, tel que

$$\overline{\mathcal{S}}_1[\mathbf{x}](t, \lambda_1) = \frac{\mathcal{S}_1[\mathbf{x}](t, \lambda_1)}{|\mathbf{x}| * \phi(t) + \epsilon}, \quad (7.32)$$

on obtient

$$\overline{\mathcal{S}}_1[\mathbf{x}](t, \lambda_1) \approx |\widehat{\psi}_{\lambda_1}(k\xi)| \frac{|\widehat{h}(\lambda_1)|}{\|h\|_1}, \quad (7.33)$$

où $\|h\|_1 = \int |h(t)| dt$ et k est un entier tel que $|k\xi - \lambda_1| > \xi/2$. En se basant sur l'équation (7.31), il est alors possible de déduire une approximation de $\mathcal{S}_2[\mathbf{x}](t, \lambda_1, \lambda_2)$

$$\mathcal{S}_2[\mathbf{x}](t, \lambda_1, \lambda_2) \approx |\widehat{h}(\lambda_1)| |\widehat{\psi}_{\lambda_1}(k\xi)| |a * \psi_{\lambda_2}| * \phi(t). \quad (7.34)$$

De même, pour les coefficients de diffusion du second ordre, on obtient

$$\overline{\mathcal{S}}_2[\mathbf{x}](t, \lambda_1, \lambda_2) \approx \frac{|a * \widehat{\psi}_{\lambda_2}| * \phi(t)}{a * \phi(t)}. \quad (7.35)$$

Les coefficients du second ordre normalisés ne dépendent ainsi ni de h ni de ξ , mais uniquement de la modulation en amplitude $a(t)$, à condition que $\mathcal{S}_1[\mathbf{x}](t, \lambda_1)$ soit non négligeable.

Sous ces mêmes hypothèses, si $e(t)$ est un bruit blanc Gaussien, on a $e * \psi_{\lambda_1}(t) \sim \mathcal{N}(0, \|\psi_{\lambda_1}\|)$ et $|e * \psi_{\lambda_1}(t)|$ suit une distribution de Rayleigh. Si on décompose alors $|e * \psi_{\lambda_1}(t)|$ tel que : $|e * \psi_{\lambda_1}(t)| = \mathbb{E}(|e * \psi_{\lambda_1}|) + \epsilon(t)$, on obtient :

$$|\mathbf{x} * \psi_{\lambda_1}| \approx |\widehat{h}(\lambda_1)| \left(\frac{\sqrt{\pi}}{2} \|\psi_{\lambda_1}\| \cdot a(t) + a(t) \cdot \epsilon(t) \right) \quad (7.36)$$

où $\epsilon(t)$ suit également une distribution de Rayleigh et l'expression de l'espérance est obtenue grâce à l'hypothèse que ψ est une ondelette complexe en quadrature de phase. Sous l'hypothèse que l'enveloppe $a(t)$ est positive, ce qui est le cas pour la cavitation, [Andén & Mallat \(2014\)](#) ont montré que :

$$\overline{\mathcal{S}}_2[\mathbf{x}](t, \lambda_1, \lambda_2) = \frac{|a * \widehat{\psi}_{\lambda_2}| * \phi(t)}{a * \phi(t)} + \tilde{\epsilon}(t) \quad (7.37)$$

où $\tilde{\epsilon}(t)$ est inférieur à $C \sqrt{\frac{4\lambda_2 Q_1}{(\pi-1)\lambda_1 Q_2}}$, avec $C \approx 1$. Ceci impliquerait que pour l'analyse de la modulation en amplitude, il vaut mieux disposer d'un plus grand nombre d'ondelettes par octaves à l'ordre 2, Q_2 , qu'à l'ordre 1, Q_1 . La scattering transform est ainsi une représentation intéressante pour extraire les périodicités cachées dans le bruit de cavitation, même dans le cas où le bruit de cavitation est coloré.

5 Apprentissage profond de la scattering transform pour la classification de bruit d'hélice blanc avec signature inconnue

5.1 Motivations

Dans le chapitre précédent nous avons développé une classe d'architectures profondes bayésiennes \mathcal{F}^* capable d'approximer avec précision l'estimateur optimal de Bayes $\Lambda^*(\mathbf{x})$ pour le problème de classification binaire de bruit d'hélice avec M signatures (6.24). Les architectures dans \mathcal{F}^* étaient explicables dans le sens où elles permettaient d'imiter la structure mathématique de $\Lambda^*(\mathbf{x})$ en approchant chacune de ses fonctions et opérations internes séparément, par le biais de modules d'approximation de fonction $\Phi_{g,\gamma}$. Leurs performances pour des petits ensembles d'entraînements n'étaient cependant pas comparables à des CNNs de pointe.

Nous avons ainsi supposé que cette perte de performance était due à l'apprentissage de M filtres $k \mapsto b_{m,k}$, de taille K bien supérieure à ce qui se fait généralement dans l'état de l'art. En effet, dans la littérature, les CNNs 1D sont généralement conçus avec des filtres de taille moyennes sur leur première couche, comme le font Dai et al. (2017) et Sailor et al. (2017). Ceci aboutit souvent à ce que la plupart des filtres apprennent collectivement une banque de filtres passe-bande distribuée de manière logarithmique, comme le montrent Golik et al. (2015), où des mêmes motifs de formes d'ondes sont retrouvés à différentes échelles temporelles. Les autres couches convolutives utilisent généralement de plus petites tailles de filtres et sont nécessaires pour obtenir de meilleures performances de classification comme le stipulent Golik et al. (2015). Ces couches de filtres supplémentaires, disposant alors de filtres de dimension 2, sont en revanche plus difficiles à interpréter, de la même manière qu'un réseau de neurones complètement connecté est moins interprétable qu'une régression linéaire. Dans cette section nous proposons ainsi de remplacer ces filtres appartenant à un espace de haute résolution temporelle, par des filtres plus courts dans une représentation de dimension réduite, mais en cherchant tout de même à garder une forme d'explicabilité. Fort heureusement il existe des représentations des signaux pouvant être considérées comme explicables du point de vue du traitement du signal et c'est notamment le cas de la scattering transform de Mallat.

Notre objectif dans cette section sera ainsi de concevoir une architecture de réseau de neurones qui soit explicable par son design et emploie des filtres de convolution courts, ce que nous ferons en s'inspirant de la structure de la scattering transform. La scattering transform de Andén & Mallat (2014) est en effet une représentation du spectre de modulation à plusieurs ordres, où les bandes de fréquences sont également espacées sur une échelle exponentielle. Comme nous l'avons vu dans la section précédente, sa structure mathématique est similaire à une architecture CNN, puisque son calcul s'effectue par le biais d'une cascade de transformées en ondelettes et d'opérations non linéaires, bien qu'elle n'implique aucun apprentissage. La scattering transform définit ainsi une sous classe de CNN permettant d'obtenir des feature maps de sortie structurées, contrairement aux CNNs classiques.

5.2 Architecture profonde bayésienne avec scattering transform apprise

Pour un signal \mathbf{x} donné, la scattering transform du premier ordre $\mathcal{S}_1[\mathbf{x}](\lambda_1)$ est indexée par une fréquence d'analyse λ_1 et définie par l'équation

$$\mathcal{S}_1[\mathbf{x}](\lambda_1) = |\mathbf{x} \star \psi_{\lambda_1}| \star \phi_T. \quad (7.38)$$

En pratique, elle est obtenue en convoluant le signal \mathbf{x} avec une famille de filtre d'ondelettes ψ_{λ_1} de fréquences centrales $\lambda_1 \in \Lambda_1$, où Λ_1 est une grille composée de Q fréquences centrales par octave. Le filtre ϕ_T moyenne localement le signal sur une durée T que nous fixons ici à la taille du signal discret $T \equiv K$, ce qui est similaire à une opération max pooling globale et fait que les coefficients $\mathcal{S}_1[\mathbf{x}](\lambda_1)$ sont des valeurs réelles invariantes dans le temps et non des vecteurs. Cette opération pourrait être justifiée quand le signal \mathbf{x} ne présente pas de périodicité d'ordre supérieur à la décomposition effectuée par la scattering transform, ce qui est le cas ici, puisque les signaux $s(t)$ ne présentent pas de périodicités d'ordre 3. Qui plus est, une scattering transform d'ordre 1 est suffisante dans notre cas, du fait que la fonction carré extrait déjà la variance d'un signal blanc.

L'analyse du signal \mathbf{x} , avec des filtres ψ_{λ_1} distribués en échelle Mel, permet alors de réduire sa dimension initiale. Qui plus est, en se basant sur des travaux concernant la théorie de l'approximation dans le domaine des ondelettes, comme ceux de [Peyré \(2020\)](#), il doit exister une séquence de coefficients $\lambda_1 \in \Lambda_1 \mapsto \beta_{m,\lambda_1}$, satisfaisant l'équation :

$$\left| \sum_{\lambda_1 \in \Lambda_1} \beta_{m,\lambda_1} \mathcal{S}_1[\mathbf{x}^2](\lambda_1) - \sum_{k=1}^K b_{m,k} x^2(k) \right| \leq \epsilon, \forall m \in \llbracket M \rrbracket, \quad (7.39)$$

où ϵ est une constante qui dépend des M signatures $s_\eta(t)$ et de la famille d'ondelettes Λ_1 utilisée. Il est ainsi, possible d'éviter l'apprentissage de longs filtres $k \mapsto b_{m,k}$ avec des pertes d'approximations mineures. En pratique, le nombre de paramètres nécessaires pour apprendre les filtres optimaux est alors considérablement réduit puisque $\dim(\Lambda_1) \ll K$, bien qu'il faille également apprendre les filtres ψ_{λ_1} .

Dans cette section nous utiliserons un module de scattering transform apprise d'ordre 1, dénoté LST (Learning-based Scattering Transform), afin d'obtenir une représentation plus flexible du signal à l'intérieur de notre architecture profonde bayésienne. L'architecture proposée doit permettre d'apprendre les coefficients β_{m,λ_1} , le module LST, ainsi que les modules d'approximation de fonction simultanément, avec un algorithme de rétro-propagation du gradient classique. Dans la sous-section 5.3, nous décrirons comment nous construisons une famille de filtres appris $\psi_{\lambda_1} \in \mathbb{R}^p$ au sein du module LST d'ordre 1, où $\lambda_1 = (s, f) \in \Lambda_1 = \llbracket S \rrbracket \times \llbracket F \rrbracket$ peut s'interpréter tel que s correspond à une échelle (la taille du signal d'entrée est réduite récursivement) et f à un index de filtre (plusieurs filtres sont utilisés dans notre LST). En procédant de la sorte, nous obtenons un module LST, $\tilde{\mathcal{S}}_1[\mathbf{x}] : \mathbb{R}^K \mapsto \mathbb{R}^{S \times F}$, qui prend en entrée le signal \mathbf{x} et retourne l'ensemble des valeurs $\tilde{\mathcal{S}}_1[\mathbf{x}](\lambda_1)$ pour $\lambda_1 = (s, f)$. Il est également possible de montrer que la borne (7.39) est valable pour notre LST d'ordre 1. En effet, on peut remplacer $\mathcal{S}_1[\mathbf{x}](\lambda_1)$ par $\tilde{\mathcal{S}}_1[\mathbf{x}](\lambda_1)$ en utilisant F filtres similaires aux Q ondelettes par octave présentes dans la scattering transform de [Mallat \(2012\)](#), à la différence près que les ondelettes sont définies dans \mathbb{C}^p et nous utiliserons dans notre implémentation des filtres

$\psi_{\lambda_1} \in \mathbb{R}^p$.

Remarque

Il est possible d'utiliser des CNNs disposant de couches de convolution complexes grâce à l'implémentation proposée par [Brooks et al. \(2020\)](#) dans leurs travaux de thèse. Néanmoins, bien qu'il soit envisageable d'implémenter un module LST complexe, dans notre cas d'application où nous cherchons une représentation ne faisant pas ressortir d'information temporelle, nous utiliserions plutôt deux fois plus de filtres appris que des filtres complexes, afin d'obtenir de meilleures performances de classification, sous des conditions de mémoire GPU limitées.

Notre architecture profonde bayésienne avec scattering transform apprise reprend l'architecture présentée en section 5 du chapitre précédent en y ajoutant le module LST d'ordre 1. Nous proposons ainsi d'approximer $\Lambda^*(\mathbf{x})$ dans (6.31) par

$$f_\nu(\mathbf{x}) = \alpha_0 + \sum_{m=1}^M \alpha_m \Phi_{e,\gamma_2} \left(\sum_{\lambda_1 \in \Lambda_1} \beta_{m,\lambda_1} \tilde{\mathcal{S}}_1[\Phi_{p_2,\gamma_1}(\mathbf{x})](\lambda_1) \right). \quad (7.40)$$

Pour expliquer cette équation, les approximations de la fonction exponentielle $e(t)$ et de la fonction carré $p_2(\mathbf{x})$ dans $f_\nu(\mathbf{x})$ sont réalisées par les réseaux de neurones d'approximation $\Phi_{e,\gamma_2} : \mathbb{R} \mapsto \mathbb{R}$ et $\Phi_{p_2,\gamma_1} : \mathbb{R} \mapsto \mathbb{R}$ respectivement. Le module LST proposé dans cette section est utilisé pour approximer les sommes $\sum_{k=1}^K b_{m,k} x^2(k)$ comme détaillé dans l'équation (7.39). Le vecteur de paramètres entraînaibles ν contient γ_1 , γ_2 , α_0 , les coefficients α_m et β_{m,λ_1} ainsi que les F filtres du module LST. Dans la sous-section suivante, nous décrivons notre implémentation du module LST d'ordre 1 $\tilde{\mathcal{S}}_1[\mathbf{x}](\lambda_1)$ utilisée dans $f_\nu(\mathbf{x})$.

5.3 Module de réseau convolutif de scattering transform d'ordre 1

Dans la figure 7.3 nous décrivons notre implémentation en tant que CNN du module LST d'ordre 1 $\tilde{\mathcal{S}}_1[\mathbf{x}](\lambda_1)$. Tout d'abord, introduisons la décomposition à plusieurs échelles temporelles du signal. La décomposition correspond aux branches en vert partant du haut vers le bas de la figure 7.3. Pour obtenir chaque nouvelle échelle de décomposition, un filtre de convolution 1D ψ_{LF} de taille $p \times 1$ est appliqué au signal d'entrée \mathbf{x} , de façon similaire au filtre passe-bas de la décomposition en ondelette discrète. La convolution est effectuée avec un padding identique, afin de ne pas modifier la taille du signal. Une opération average pooling ϕ_2 de taille et de pas 2 est ensuite appliquée pour sous-échantillonner le signal, réduisant ainsi sa taille effective par deux. Le filtre ψ_{LF} n'est pas appris. Il s'agit d'une ondelette de Morlet passe-bas, ce qui contraint les couches suivantes à analyser les octaves de fréquences inférieures. Par conséquent, nous définissons la décomposition sur différentes échelles temporelles de façon récursive telle que

$$\mathbf{x}_{\dagger s+1} = (\mathbf{x}_{\dagger s} \star \psi_{LF}) \star \phi_2 \quad \forall s \in [1, S-1], \quad (7.41)$$

avec l'initialisation $\mathbf{x}_{\dagger 1} = \mathbf{x}$. Chaque échelle de décomposition correspond à une couche spécifique du réseau convolutif. La longueur du signal $\mathbf{x}_{\dagger s}$ doit être plus large que la taille du filtre

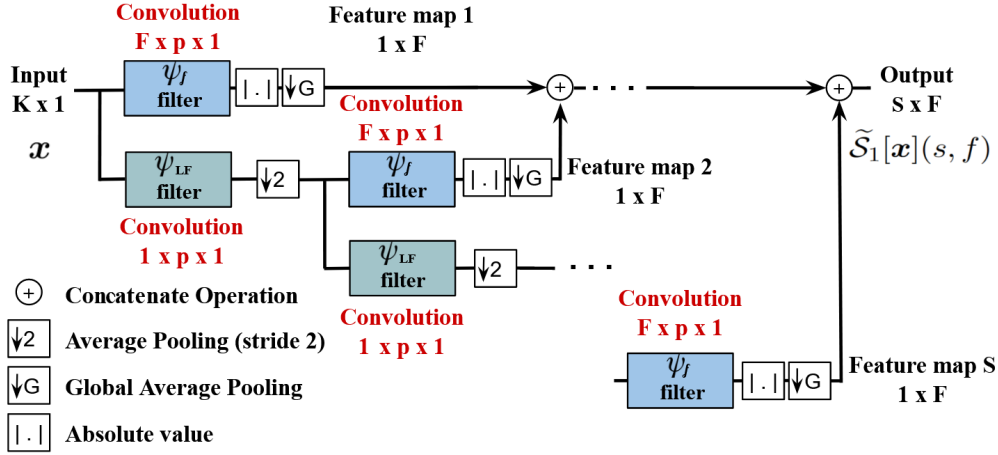


Figure 7.3: Architecture du module LST d'ordre 1 $\tilde{\mathcal{S}}_1[\mathbf{x}]$.

ψ_{LF} , p . Ainsi, le nombre de couche de décomposition S doit satisfaire la condition

$$2p > \frac{K}{2^S} \geq p. \quad (7.42)$$

Comme le font [Andén & Mallat \(2014\)](#), on pourra également envisager une condition d'arrêt sur une durée discrète maximale à analyser ΔK , telle que :

$$\Delta K \geq p \cdot 2^S. \quad (7.43)$$

Nous introduisons ensuite les filtres appris ψ_{λ_1} avec $\lambda_1 = (s, f)$. Le jeu de filtres $\psi_{(s,f)}$ ne dépend en revanche pas de l'échelle de décomposition s , puisque nous utilisons le même jeu de filtres pour toutes les couches du module LST d'ordre 1. Le LST applique ainsi F filtres ψ_f au niveau de la couche s :

$$\tilde{\mathcal{S}}_1[\mathbf{x}](s, f) = |\mathbf{x}_{\uparrow s} \star \psi_f| \star \phi_G, \quad (7.44)$$

où ϕ_G est une opération average pooling globale, nécessairement de taille $K/2^s$ et $\lambda = (s, f)$. Chaque filtre ψ_f , de taille $p \times 1$, est appris et comme indiqué dans l'équation (7.44), la fonction d'activation est la valeur absolue. De cette façon, chaque couche s analyse une octave différente de \mathbf{x} avec F différents filtres ψ_f au lieu des Q ondelettes par octave habituellement utilisées dans la scattering transform [Andén & Mallat \(2014\)](#). Le sous-échantillonnage nous permet alors d'apprendre des filtres ψ_{λ_1} couvrant un grand champ récepteur, malgré leurs tailles limitées.

Enfin, au niveau de chaque nœud de l'architecture, deux ensembles de feature maps sont produites, donnant naissance à deux branches équivalentes aux décompositions hautes fréquences (HF) et basses fréquences (LF) trouvées dans la transformée en ondelettes discrète et présentes à chaque ordre de la scattering transform. Dans la première branche, positionnée en haut de chaque nœud sur la figure 7.3, nous appliquons au signal $\mathbf{x}_{\uparrow s}$ les F filtres ψ_f pour obtenir les valeurs $\tilde{\mathcal{S}}_1[\mathbf{x}](s, f)$ pour chaque indice f . La sortie $\tilde{\mathcal{S}}_1[\mathbf{x}](s, f)$ à l'échelle s a ainsi une taille $1 \times F$ en raison de l'opération de pooling globale. Cette sortie est concaténée avec

les sorties des premières couches, créant ainsi des connexions directes entre chaque couche et la sortie finale du réseau $\tilde{\mathcal{S}}_1[\mathbf{x}]$, de taille $S \times F$. Cette structure peut ainsi s'apparenter à un réseau résiduel comme celui présenté par He et al. (2016), ou d'autres types de "shortcut/skip connections". Cette architecture s'inspire ainsi de la structure mathématique de la scattering transform d'ordre 1, en adaptant certaines opérations, classiquement utilisées en traitement du signal, par des opérations ou couches convolutives présentes dans les CNNs habituels. Ceci permet en effet de procéder à un entraînement du module LST d'ordre 1 par rétro-propagation du gradient.

Remarque

Une variante possible, plus proche de l'implémentation de Mallat (2012), consiste à utiliser un filtre d'average pooling de durée effective T_1 , afin d'obtenir une représentation liée au temps. On obtient alors une représentation équivalente à un scalogramme avec F canaux, où chaque canal est lié à une ondelette mère différente.

$$\tilde{\mathcal{S}}_1[\mathbf{x}](t, s, f) = |\mathbf{x}_{\dagger s} \star \psi_f| \star \phi_{T_1}(t). \quad (7.45)$$

5.4 Méthodologie et optimisation des modèles pour les expérimentations

Dans cette sous-section, nous voulons analyser les performances de notre architecture $f_\nu(\mathbf{x})$, comparée à l'architecture $f_\theta(\mathbf{x})$ et le réseau FCNN, détaillé dans le tableau 6.3. Pour nos expérimentations, nous utiliserons un ensemble d'apprentissage \mathcal{S} contenant jusqu'à $N_{max} = 8 \cdot 10^5$ signaux tirés de $M = 16$ signatures, tandis que les jeux de test et de validation, respectivement \mathcal{S}_{test} et \mathcal{S}_{val} , contiennent $2 \cdot 10^5$ échantillons. La précision de nos modèles sera comparée aux précisions de l'architecture FCNN et du test de Bayes optimal $\delta^*(\mathbf{x})$ pour des signaux comprenant $K = 512$ échantillons et pouvant appartenir à $M = 16$ signatures de navires marchands différents.

Le tableau 6.3 décrit par le biais de couches Keras les 3 différentes architectures profondes sur signaux acoustiques bruts utilisées dans cette comparaison. Pour définir nos architectures d'approximations f_θ et f_ν , nous utiliserons des modules d'approximation de fonction Φ_{p_2, γ_1} et Φ_{e, γ_2} disposant respectivement de $L_1 = 6$ et $L_2 = 3$ couches, comportant toutes $W = 9$ filtres de convolution de taille 1. L'architecture f_ν est la même que f_θ si ce n'est qu'elle contient le module LST d'ordre 1 présenté précédemment. Ce module contient $F = 4$ filtres de taille $p = 32$ initialisés aléatoirement. Cette méthode requiert d'apprendre $(p + 1)F = 132$ paramètres pour les filtres ψ_f , plus $M(S \times F + 1) = 272$, avec $S = 4 \approx \ln_2(K) - \ln_2(p)$, pour les coefficients β_{m, λ_1} , comparés aux $M(K + 1) = 8208$ paramètres requis pour les coefficients $b_{m, k}$.

Les différentes architectures suivent la même procédure d'optimisation lors de l'apprentissage, avec l'utilisation de l'optimiseur Adam et de l'entropie croisée binaire comme fonction de perte. Le critère d'arrêt est une absence d'amélioration de la perte de validation pendant 15 époques consécutives, ou au maximum après 100 époques d'entraînement. La contrainte "Maximum kernel" est utilisée pour toutes les couches convolutives afin de majorer les valeurs

FCNN (1 037 281)	$f_\theta(\mathbf{x})$ (10 515)	$f_\nu(\mathbf{x})$ (2 711)
Entrée : 512×1 signal temporel $x(k)$		
Conv [32, 64]	Conv [9, 1] $\times 6$	
MaxPool : 2×2		
Conv [64, 8]		
MaxPool : 2×2		
Conv [64, 8]		
MaxPool : 2×2	Conv [1, 1]	
Conv [128, 8]		$LST_1(S, F)$
MaxPool : 2×2	Conv [16, 512]	[16, 16]
Conv [128, 8]	Conv [9, 1] $\times 3$	
MaxPool : 2×2		
Conv [256, 8]		
MaxPool : 2×2		
Conv [256, 8]	Conv [1, 1]	
GlobalAveragePool	Flatten layer	
Final dense layer		

Tableau 7.1: Tableau des différentes architectures utilisées pour une comparaison avec des signaux de $K = 512$ échantillons. $LST_1(S, F)$ représente un module LST d'ordre 1 disposant de respectivement S couches et F filtres.

des poids des filtres convolutifs et nous appliquons une régularisation L^2 aux couches de sortie, de même qu'une normalisation par batch pour toutes les couches convolutives. Pour chaque valeur N considérée dans la figure 7.4, les performances des architectures sont estimées en calculant la moyenne des erreurs d'entraînement et de test parmi 10 entraînements différents. Un modèle avec une perte de validation supérieure à $1,2 \times$ la meilleure perte de validation, pour un N donné, est rejeté pour le calcul des performances.

5.5 Résultats numériques

En se basant sur la figure 7.4, comparée à f_θ , f_ν donne des performances similaires à des CNNs classiques pour des jeux de données de faibles tailles, mais sans être une boîte noire. Ceci semble confirmer qu'apprendre de longues dépendances temporelles avec des filtres de convolution tout aussi larges, est préjudiciable à l'apprentissage d'un classifieur performant. Notre méthode, consistant à apprendre ces longues dépendances temporelles au sein d'une représentation apprise de dimension réduite par le biais de notre module LST, semble en revanche être une solution à ce problème garantissant une forme d'explicabilité. Le classifieur f_θ , disposant de plus de paramètres, est cependant plus proche du test optimal pour des ensembles de données plus volumineux. Ceci peut être dû aux pertes d'approximation de notre LST d'ordre 1.

Les filtres ψ_f , appris par f_ν , sont bien moins bruités que les longs filtres $b_{m,k}$ et peuvent apprendre différentes réponses fréquentielles au sein d'une même octave avec des périodicités divergeantes entre les pics. Les formes d'ondes apprises par les filtres $\psi_f \in \mathbb{R}^p$ peuvent

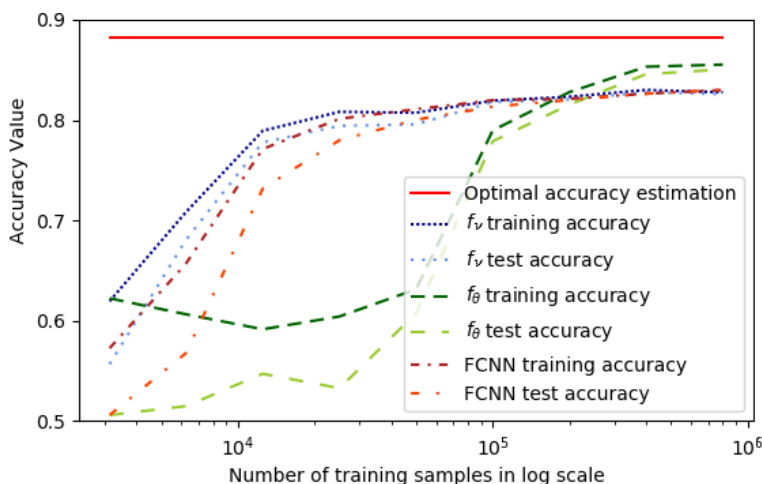


Figure 7.4: Précision d'entraînement et de test en fonction de la taille du jeu de données associé et des signaux de $K = 512$ échantillons. La précision optimale correspond à un test de Monte-Carlo avec le détecteur (6.30).

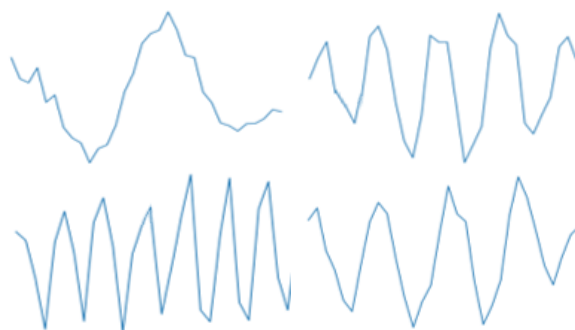


Figure 7.5: Exemples de filtres appris ψ_f au sein du module LST d'ordre 1 présent dans l'architecture f_ν .

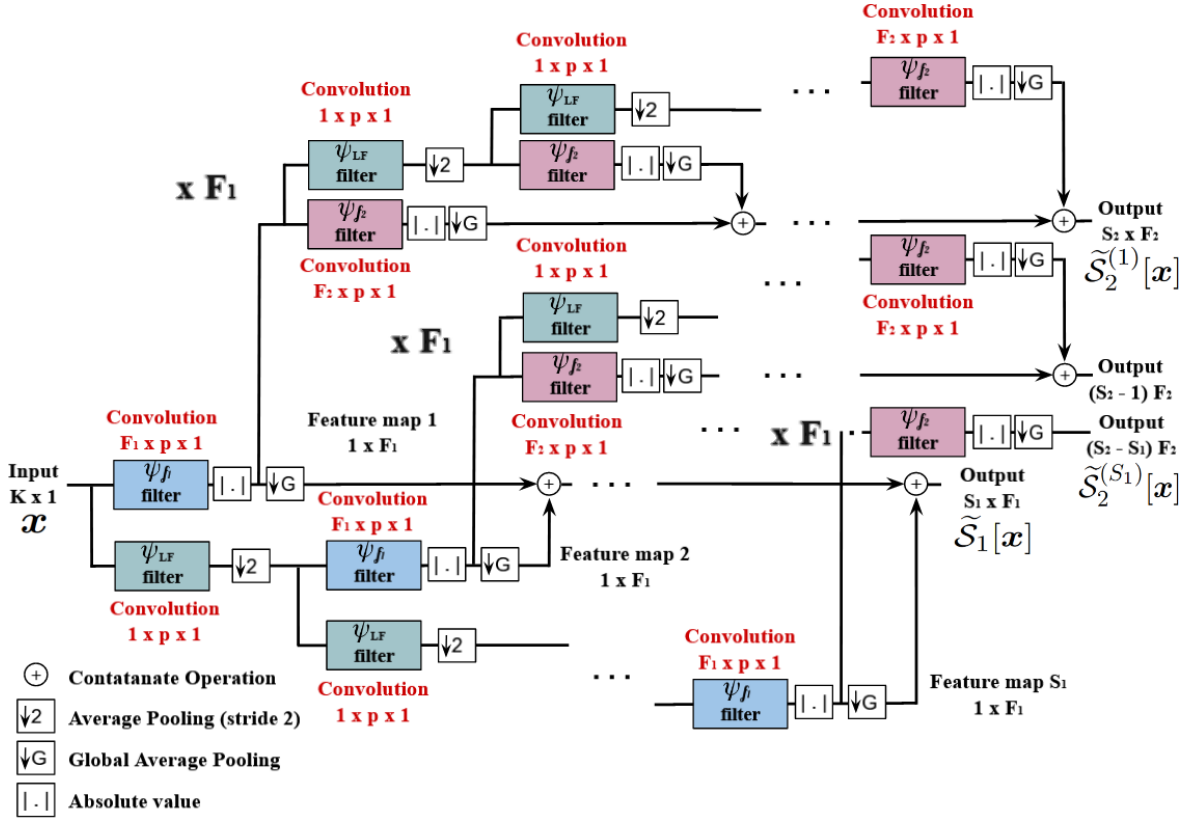
également ressembler aux parties réelles ou imaginaires d'ondelettes classiques telles que des ondelettes de daubechies d'ordre 2.

5.6 Module de réseau convolutif scattering transform d'ordre 2

Dans la figure 7.6 nous décrivons un module LST d'ordre 2 noté $\tilde{\mathcal{S}}_2[\mathbf{x}](\lambda_1, \lambda_2)$. Le module LST d'ordre 2 reprend la structure du module LST d'ordre 1, présentée dans la sous-section 5.3, en ajoutant des embranchements après chaque analyse du signal $\mathbf{x}_{\dagger s}$ par les F_1 filtres ψ_{f_1} . Aussi, à l'embranchement de chaque couche s , en haut après chaque branche bleue sur la figure 7.6, de nouveaux modules LST sont formés avec pour entrée le signal

$$\mathbf{x}'_{\dagger s'} = |\mathbf{x}_{\dagger s} \star \psi_{f_1}| \quad (7.46)$$

et chaque embranchement continue la décomposition du signal sur plusieurs échelles temporelles.


 Figure 7.6: Architecture du module LST d'ordre 2 $\tilde{\mathcal{S}}_2[\mathbf{x}]$.

Les modules LST à l'embranchement de la couche s , que l'on notera $\tilde{\mathcal{S}}_2^{(s)}[\mathbf{x}]$, utilisent les mêmes filtres ψ_{LF} , de taille $p \times 1$, que le module LST d'ordre 1 $\tilde{\mathcal{S}}_1[\mathbf{x}](s, f_1)$ et toujours avec un padding identique. Par la suite nous différencierons l'index s des couches du LST d'ordre 1 et l'index s' pour les couches du LST à l'embranchement de la couche s . La décomposition sur différentes échelles temporelles se fait encore de façon récursive telle que

$$\mathbf{x}'_{\dagger s'+1} = (\mathbf{x}'_{\dagger s'} \star \psi_{LF}) \star \phi_2 \quad \forall s' \in [s, S_2 - 1]. \quad (7.47)$$

Comme précédemment, la longueur du signal $\mathbf{x}'_{\dagger s'}$ doit être plus large que la taille du filtre ψ_{LF} , p . Ainsi, le nombre de couches de décomposition de $\tilde{\mathcal{S}}_2^{(s)}[\mathbf{x}]$, S_2 , doit satisfaire la condition

$$2p > \frac{K}{2^{S_2-s-1}} \geq p. \quad (7.48)$$

Plus généralement on utilisera une condition d'arrêt sur une durée discrète maximale à analyser $\Delta K_2 > \Delta K_1$, pour limiter le nombre total de coefficients de la scattering transform d'ordre 2, tel que :

$$\Delta K_2 \geq p \cdot 2^{S_2}. \quad (7.49)$$

Les filtres appris ψ_{λ_2} avec $\lambda_2 = (s', f_2)$ sont différents des filtres ψ_{λ_1} utilisés dans le module LST d'ordre 1. Les modules LST de la couche s appliquent ainsi chacun F_2 filtres ψ_{f_2} au

niveau de la couche s' :

$$\tilde{\mathcal{S}}_2^{(s)}[\mathbf{x}](s', f_1, f_2) = |\mathbf{x}'_{\dagger s'}(f_1) \star \psi_{f_2}| \star \phi_G, \quad (7.50)$$

où ϕ_G est une opération average pooling globale, nécessairement de taille $K/2^{s'}$. Les filtres d'ordre 1 ψ_{f_1} , peuvent permettre d'effectuer des analyses dans différentes bandes de fréquences, où même à des fréquences très précises. Les filtres ψ_{f_2} , de taille $p \times 1$, peuvent quant à eux apprendre avec plus de flexibilités des formes d'ondes qui peuvent être utiles à l'analyse des moments d'ordre 2 d'un signal. En effet, si les filtres d'ordre 1 ψ_{f_1} effectuent une analyse fréquentielle en échelle Mel, les filtres ψ_{f_2} peuvent réaliser collectivement une analyse des fréquences de modulation du signal ou apprendre directement des filtres adaptés. Il est cependant possible d'apprendre directement des filtres pouvant s'apparenter à des modulations en amplitude dans le module LST du premier ordre quand les filtres sont initialisés aléatoirement, comme en attestent les filtres affichés sur la figure 7.7. Des filtres similaires peuvent également être trouvés pour les filtres d'ordre 2 sur d'autres types de jeux de données, ce qui permet d'analyser des moments d'ordre 3. Pour nos signaux ce serait équivalent à extraire des variations de l'ordre de plusieurs secondes au niveau de l'intensité d'une signature d'hélice.

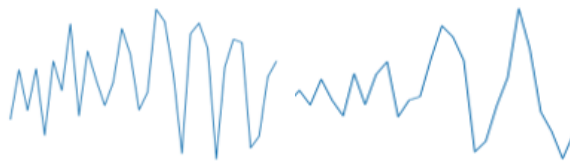


Figure 7.7: Exemples de filtres appris ψ_{f_1} au sein d'un module LST d'ordre 2.

Comme c'est le cas pour le module LST d'ordre 1, les sorties à l'échelle s' , $\tilde{\mathcal{S}}_2^{(s)}[\mathbf{x}](s', f_1, f_2)$ ont une taille $1 \times F_2 \times F_1$, en raison des opérations d'average pooling globales effectuées. Une fois calculées, elles sont concaténées avec les sorties précédentes, pour former la sortie finale du module à l'embranchement de la couche s , $\tilde{\mathcal{S}}_2^{(s)}[\mathbf{x}]$, de taille $(S_2 - s)F_2 \times F_1$. Le module de scattering transform d'ordre 2, $\tilde{\mathcal{S}}_2[\mathbf{x}]$, est ainsi la concaténation du module LST d'ordre 1 $\tilde{\mathcal{S}}_1[\mathbf{x}]$ et des S_1 modules LST $\tilde{\mathcal{S}}_2^{(s)}[\mathbf{x}]$. Au total $S_1 \times F_1 + (S_2 - S_1 - 1)(S_2 + S_1) \times F_2 \times F_1/2$ coefficients de diffusion sont ainsi calculés. Encore une fois, l'architecture du module LST d'ordre 2 s'inspire de la structure mathématique de la scattering transform d'ordre 2, mais permet d'être entraînée comme un réseau CNN classique.

Remarque

Pour former moins de branches dans l'architecture CNN, il est possible d'utiliser l'ensemble des feature maps

$$\mathbf{x}'_{\dagger s'} = (|\mathbf{x}_{\dagger s} \star \psi_{f_1}|)_{f_1 \in [1, F_1]}, \quad (7.51)$$

comme entrée, afin d'ajouter un unique embranchement après l'analyse du signal $\mathbf{x}_{\dagger s}$ par les F_1 filtres du LST d'ordre 1. Ceci permet ainsi de générer des sorties à l'échelle s' , $\tilde{\mathcal{S}}_2^{(s)}[\mathbf{x}](s', f_2)$, et d'obtenir au total $S_1 \times F_1 + (S_2 - S_1 - 1)(S_2 + S_1) \times F_2/2$ coefficients de diffusion. En effet, si les filtres ψ_{f_2} ne sont pas appliqués aux signaux $\mathbf{x}'_{\dagger s'}$, mais à l'ensemble de feature maps $\mathbf{x}'_{\dagger s'}$, ils ont donc une dimension effective $p \times F_1$. Cette variante permet d'obtenir des architectures disposant de plus de filtres F_2 pour une même mémoire GPU et, en conséquence, souvent de meilleures performances. Cette variante devient cependant moins explicable en impliquant le calcul de combinaisons non linéaires de plusieurs feature maps, comme c'est le cas dans les CNNs classiques.

5.7 Algorithme d'apprentissage pour Q filtres par octave

Dans cette sous-section nous proposons de modifier le module LST afin de permettre d'inclure Q versions d'une même forme d'onde apprise dans une même octave. Pour y parvenir, nous générons $Q \times F$ filtres lors de l'initialisation, afin d'obtenir Q versions de chaque filtre ψ_f avec des dimensions différentes. Ces filtres seront notés $\psi_{f,q}$ et leurs tailles, que nous noterons p_q , sont telles que :

$$p_q = \left\lfloor 2^{\ln_2(\frac{p}{2}) + \frac{q}{Q}(\ln_2(p) - \ln_2(\frac{p}{2}))} \right\rfloor. \quad (7.52)$$

Les filtres $\psi_{f,Q}$ ont ainsi une taille maximale, $p_Q = p$, avec p le paramètre de taille de filtre de notre algorithme.

Nous partons ensuite du filtre $\psi_{f,1}$, ayant une taille minimale p_1 pour obtenir ses $Q - 1$ copies redimensionnées avec un algorithme ré-échantillonnage. Ces différentes versions sont obtenues avec la fonction "resample" de la librairie Signal de Python, qui utilise une interpolation par série de Fourier. Pendant l'apprentissage, chacune des Q versions du filtre ψ_f est entraînée par descente de gradient classique. À la fin de chaque batch, toutes les versions différemment échantillonnées du filtre sont mises à contribution pour modifier le filtre $\psi_{f,1}$, du fait que ce soit celui qui dispose du moins d'échantillons et donc qui peut stocker le moins d'information. Une moyenne de ces copies est effectuée donnant ainsi une mise à jour pour le filtre $\psi_{f,1}$. $\psi_{f,1}$ est ensuite ré-échantillonnée $Q - 1$ fois pour donner les mises à jours des autres versions du filtre.

Remarque

Bien que nous ayons présenté ici un module d'architecture profonde de scattering transform apprise suivant des algorithmes de CNNs conventionnels, d'autres méthodes d'apprentissage de la scattering transform sont possibles. Dans le cas 2D, [Gauthier et al. \(2021\)](#) proposent en effet d'apprendre des ondelettes $\psi_{\sigma,\theta,\epsilon,\gamma}$ en effectuant une optimisation sur leurs paramètres d'échelle de fenêtre gaussienne σ , d'orientation globale θ , d'échelle fréquentielle ϵ et d'inclinaison γ .

5.8 Conclusion

Dans ce chapitre nous avons mis en avant les propriétés la scattering transform de [Mallat \(2012\)](#), une représentation capable d'extraire les statistiques d'un signal à plusieurs ordres et exploitable pour une implémentation sous forme de CNN, de par sa structure mathématique. Cette représentation présente un intérêt important pour la classification acoustique sous-marine puisqu'elle permet d'extraire les caractéristiques de bruits cyclostationnaires tels que nos bruits d'hélices et bruits de moteurs, tout en présentant une stabilité aux déformations dues à l'effet Doppler.

Nous avons par la suite proposé notre propre implémentation d'un module de scattering transform apprise (LST). Notre implémentation permet de remplacer les ondelettes présentes dans la scattering transform, par un apprentissage de filtres de convolution. Dans ce sens, elle diffère du module Kymatio, proposé par [Andreux et al. \(2020\)](#), en utilisant des filtres de convolutions réels et non des ondelettes complexes prédéterminées. Son calcul s'effectue alors couche par couche, comme dans un CNN classique avec Tensorflow en backend. Ceci impose cependant de limiter le nombre de filtres appris dans le module LST, afin de ne pas faire exploser le nombre de coefficients de diffusion et ainsi dépasser la capacité mémoire des GPUs. Bien que nous ayons utilisé des opérations d'average pooling globales pour notre problème de classification de bruit d'hélice, il est également possible de garder une information temporelle dans notre implémentation du module LST. Notre module LST, modifié de cette manière, apprendrait ainsi une autre forme de représentation dépendant encore du temps et pouvant servir d'entrée à d'autres couches de convolutions. De même, la possibilité de pouvoir apprendre plusieurs filtres d'ondelette pourrait permettre à la fois d'analyser des signaux cyclostationnaires, mais également des impulsions ou transitoires avec d'autres filtres spécialisés. Les transformées en ondelettes ont en effet montré leur capacité à extraire des informations non-stationnaires dans le domaine des images, comme le montrent [Atto & Berthoumieu \(2011\)](#).

Au final, de par son architecture comportant plusieurs filtres et générant ainsi plusieurs feature maps, notre implémentation diffère de la scattering transform fixe, en offrant plus de flexibilité pour l'extraction de caractéristiques, tout en conservant certaines propriétés de la scattering transform pouvant être intéressantes pour notre problème acoustique. L'utilisation de très peu de paramètres, avec des petits filtres de convolution appris, permet également d'éviter le surapprentissage et se caractérise par de petites erreurs de généralisation malgré une importante expressivité du réseau. Du point de vue de l'explicabilité, cette architecture garde une structure compréhensible en conservant une sortie indexée par les indices d'échelles

temporelles s et s' ainsi que les indices de feature maps liées aux filtres f_1 et f_2 , ce qui en fait une architecture intrinsèquement explicable par design. Avec la variante plus efficace proposée en remarque de la sous-section 5.6, on obtient cependant une architecture un peu moins explicable, puisque les indices des feature maps de sortie ne sont plus liés qu'aux indices f_2 des filtres de convolution multi-canal.

Chapitre 8

Expérimentations sur jeux de données

1 Introduction

Dans la section 2 de ce chapitre nous effectuons une expérimentation sur un jeu de données réel avec un module LST présenté dans le chapitre précédent. Nous discutons ensuite d'autres possibilités d'utilisation pour les modules LST. La section 3 compare les performances de différentes architectures profondes citées durant la thèse sur notre jeu de données ASM simulé et analyse l'impact de différents paramètres d'architectures FCNNs sur leurs performances de généralisation.

2 Expérimentation sur l'architecture convolutive de scattering transform apprise

2.1 Expérimentation sur jeux de données réelles

Afin de valider l'utilité de notre implémentation CNN de la scattering transform, nous testons le module LST au sein d'une architecture décrite dans le tableau 8.1, sur le jeu de données ShipEar proposé par Santos-Domínguez et al. (2016). Ce jeu de données contient des enregistrements acoustiques sous-marins de longueurs variables provenant de $M = 11$ types de navires distincts (navire à passagers, dragueur, chalutier, zodiac, ferry, voilier, ...), effectuant différentes manoeuvres et recueillis par des capteurs dans différents ports en Espagne. Pour faciliter le problème, nous rassemblons les enregistrements en deux classes. La classe H_0 est composée d'enregistrements ambiants ainsi que de certains enregistrements parasites présents dans le jeu de données comme des bruits de dauphins. La classe H_1 est composée de tous les signaux contenant des bruits de navires à propulsion mécanique.

Tous les enregistrements initiaux sont divisés en segments de 10 secondes avec une même fréquence d'échantillonnage $fs = 22050$ Hz. Nous analysons à l'écoute tous les segments résultants pour ne conserver que les signaux de navires audibles dans la classe H_1 . Nous dupliquons également des segments de la classe H_0 pour équilibrer le nombre d'échantillons de chaque classe et obtenir un total de 1412 échantillons d'entraînement et 352 échantillons de validation. La figure 8.1 permet de visualiser des spectrogrammes d'échantillons représentatifs du jeu de données.

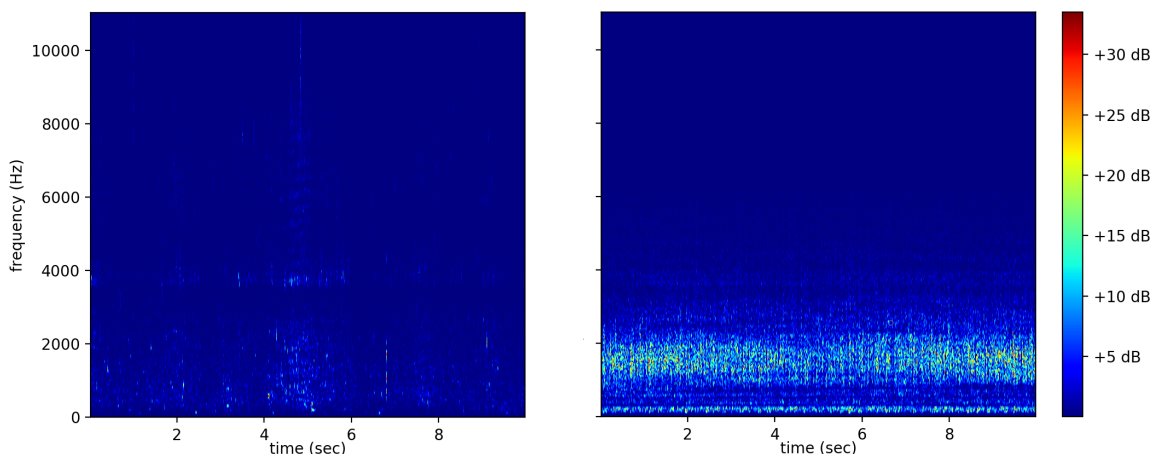


Figure 8.1: Spectrogrammes de signaux du jeu *ASM ShipsEar* appartenant à la classe H_0 , avec seulement du bruit ambiant (gauche) et H_1 , avec un bruit de navire à passagers (droite).

En ajustant l’architecture de base pour l’adapter aux données d’entrée, nous obtenons un CNN incluant d’abord un module LST d’ordre 1 dépendant du temps. En effet, en utilisant une opération d’average pooling ϕ_{T_1} , de durée $T_1 = 0,25$ seconde, à la place d’une opération d’average pooling globale ϕ_K , comme dans l’équation (7.44), on obtient un module LST :

$$\tilde{\mathcal{S}}_1[\mathbf{x}](t, s, f) = |\mathbf{x}_{\uparrow s} \star \psi_f| \star \phi_{T_1}(t). \quad (8.1)$$

Le critère de décomposition multi-échelles pour S est tel que $p \cdot 2^S \leq \Delta k_1 = 0.2 fs$, avec p la taille des filtres ψ_f , qui vaut ici 32 et fs la fréquence d’échantillonnage. Par conséquent, $\tilde{\mathcal{S}}_1[\mathbf{x}](t, s, f)$ donne une caractéristique de sortie 2D multicanal de dimension $(K/2^S, S, F)$, que nous redimensionnons pour obtenir une représentation 1D multicanal de dimension $(K/2^S, S \times F) = (81, 112)$, en concaténant les différentes échelles d’analyse s avec les indices f , pour $S = 7$ et $F = 16$. L’architecture CNN est ensuite suivie d’une succession de 4 couches de convolution et d’opérations d’average pooling de dimensions $[(32, 16), (9, 24), (9, 32), (9, 48)]$, ainsi que d’une opération d’average pooling globale et une dernière couche dense.

Sur l’ensemble de données de validation, notre modèle entraîné avec l’optimiseur Adam et une entropie croisée binaire, obtient une précision de 81,25%. C’est un score inférieur aux 94,3% obtenus par [Hong et al. \(2021\)](#) avec un ResNet sur le jeu de données ShipEar original avec 11 classes différentes et un pré-traitement des signaux différents. Ces meilleures performances du ResNet ne sont cependant pas surprenantes. En effet, [Hong et al. \(2021\)](#) utilisent une agrégation de représentations expertes, notamment des coefficients cepstraux de fréquence Mel (MFCC) et un spectrogramme Mel, tandis que notre réseau doit apprendre des filtres de convolution ainsi que la représentation du module LST en s’entraînant uniquement sur un petit ensemble d’échantillons (1412). Cette architecture est cependant meilleure comparée à l’utilisation d’un module LST $\tilde{\mathcal{S}}_1[\mathbf{x}](s, f)$ invariant dans le temps, suivi d’une dernière couche dense de décision. De même, l’utilisation de modules LST d’ordre 2 invariant dans le temps,

LST CNN (159 778)
Entrée : signal temporel $x(k)$ 220500×1
LST ₁ (7, 16)
Conv [16, 32]
AveragePool : 2×2
Conv [24, 9]
AveragePool : 2×2
Conv [32, 9]
AveragePool : 2×2
Conv [48, 9]
GlobalAveragePool
Final dense layer

Tableau 8.1: Architecture LST CNN utilisée pour la classification de signaux du jeu de données ShipEar.

n'a pas permis d'obtenir de bons résultats. En effet, ce dernier type d'architecture est bien plus difficile à entraîner sur ce problème, et sans une initialisation des filtres ψ_{f1} avec par exemple des ondelettes Morlet, on obtient des scores de précision de 50 %, équivalents à une décision prise au hasard.

2.2 Discussions

D'autres essais des modules LST d'ordre 1 et d'ordre 2 ont été réalisés sur les jeux de données ESC-50 et TUT cités dans la section 2 du chapitre 2, mais sans pouvoir aboutir à des conclusions quant au potentiel de notre méthode. En effet, quelques expérimentations ont été réalisées sans "GridSearch" par faute de temps et nous n'avons par exemple pas pu analyser l'impact de la taille des filtres ψ_{f_1} et ψ_{f_2} sur l'apprentissage. L'utilisation d'une initialisation des filtres ψ_{f_1} avec des ondelettes Morlet ne semble pas non plus trop impacter les performances de modules LST d'ordre 2 suivis d'une couche dense de décision. En effet, avec ou sans, nous obtenons environ 71 % de classification sur le jeu de données ESC-50, mais l'initialisation des filtres à d'ordre 1 permet d'obtenir de bonnes performances en beaucoup moins d'époques. Pour des modules LST dépendant du temps, ces scores pourraient cependant être améliorés avec l'utilisation de couches de convolution 1D ou 2D, ainsi que des couches d'attention. Les modules LST d'ordre 2 produisent en effet un ensemble de feature maps de très grande dimension qui contient les $S_1 \times F_1 + (S_2 - S_1 - 1)(S_2 + S_1) \times F_2 \times F_1/2$ coefficients de diffusion d'ordre 1 et d'ordre 2, et sans étapes de réduction de dimension, cela nous force à utiliser une couche de sortie avec des neurones disposant d'autant de paramètres.

Qui plus est, les modules LST pouvant être entraînés de façon non supervisée, il serait également intéressant d'observer l'impact d'un pré-entraînement de ce type sur les performances de notre module LST. En l'occurrence, nous avons pu réaliser une implémentation du LST d'ordre 1 en tant qu'auto encodeur et auto encodeur variationnel (VAE) avec un module LST

d'ordre 1 inverse, comme décodeur, qui utilise des couches de "Upsampling" en lieu et place de couches d'average pooling locales. Il reste également nécessaire de prendre en compte les limitations de notre implémentation apprise de la scattering transform qui est plus limitée, en termes de mémoire GPU, que l'implémentation de la scattering transform fixe dans la librairie Kymatio, où l'arbre de transformation est traversé en profondeur et non couche par couche. Aussi, il est probable qu'une approche avec de multiples ondelettes fixes, comme dans l'article de [Oyallon et al. \(2017\)](#), où on permet tout de même d'apprendre des couches de convolutions supplémentaires, puisse être plus efficace en termes de performances que notre méthode.

3 Expérimentations sur simulateur de bruits acoustiques sous-marins

3.1 Classification multi classes mono label

Dans cette sous-section nous comparons divers types d'architectures profondes, sans couche de neurones à propagation avant, avec un jeu de données \mathcal{S} obtenu en générant des couples de données (\mathbf{x}, y) avec le simulateur de bruits acoustiques sous-marins présenté dans la section 3 du chapitre 4. Pour nos expérimentations, \mathcal{S} contient en tout $N = 3.5 \cdot 10^5$ signaux tirés de $M = 7$ classes différentes, tandis que les jeux de test et de validation, respectivement \mathcal{S}_{test} et \mathcal{S}_{val} , contiennent $7 \cdot 10^4$ échantillons. Les signaux durent 5 secondes et sont échantillonnés à 22050 Hz, pour un total de $K = 110250$ échantillons temporels.

Quatre architectures sont comparées durant l'expérimentation : le FCNN M11 de [Dai et al. \(2017\)](#), disposant de 10 couches de convolutions, EnvNet-v2, une architecture CNN 1D et 2D adaptée de la publication [Tokozume et al. \(2017\)](#), un LST-CNN, ainsi qu'un CNN 2D prenant un spectrogramme d'échelle Mel en entrée. Les architectures sont décrites dans le tableau 8.2 et suivent la même procédure d'optimisation lors de l'apprentissage, avec l'utilisation de l'optimiseur Adam et de l'entropie croisée catégorique comme fonction de perte. Le critère d'arrêt est une absence d'amélioration de la perte de validation pendant 20 époques consécutives, ou au maximum après 100 époques d'entraînement. Les performances des modèles pour chaque valeur affichée sont estimées en calculant la moyenne et l'écart type des erreurs d'entraînement et de test parmi 5 entraînements différents.

On constate ainsi que, hormis le CNN 1D et 2D, tous ces types de CNNs ont des performances relativement comparables sur notre jeu de données de bruits ASM, peut être dues au fait que les architectures ont convergé vers des performances quasi optimales. Il est également nécessaire de noter que la majorité des paramètres entraînaibles dans l'architecture LST-CNN, se trouvent dans la couche dense de sortie avec ses 17 311 paramètres, et non dans le module LST d'ordre 2, $LST_2(\Delta K_1, \Delta K_2, F_1, F_2)$. Chacun des 7 neurones de sortie doit en effet apprendre un hyperplan de dimension $D = 2472$, ce qui commence à s'éloigner de ce qu'il se fait dans la littérature et ce, malgré l'utilisation de seulement 4 filtres d'ordre 1 et 12 filtres d'ordre 2.

FCNN M11 (2 781 767)	EnvNet-v2 (1 694 281)	LST-CNN (22 816)	CNN 2D (1 479 177)
Signal temporel 110250 × 1			Spectrogramme Mel 192 × 256
Conv [256, 64]	Conv [256, 64]	LST ₂ (0.05, 2.0, 4, 12)	Conv2d [128, 3, 3]
MaxPool : 2 × 2	MaxPool : 2 × 2		MaxPool : [2, 2] × [2, 2]
Conv [64, 8]	Conv [64, 8]		Conv2d [256, 3, 3]
MaxPool : 2 × 2	MaxPool : 2 × 2		MaxPool : [2, 2] × [2, 2]
Conv [64, 8]	Conv [64, 8]		Conv2d [512, 3, 3]
MaxPool : 2 × 2	MaxPool : 2 × 2		
Conv [128, 8]	Conv [128, 8]		
MaxPool : 2 × 2	MaxPool : 2 × 2		
Conv [128, 8]	Conv [128, 8]		
MaxPool : 2 × 2	Reshape(. , 128 , 1)		
Conv [128, 8]	Conv2d [128, 3, 3]		
MaxPool : 2 × 2	MaxPool : [6, 2] × [6, 2]		
Conv [256, 8]	Conv2d [256, 3, 3]		
MaxPool : 2 × 2	MaxPool : [6, 2] × [6, 2]		
Conv [256, 8]	Conv2d [512, 3, 3]		
MaxPool : 2 × 2			
Conv [256, 8]			
MaxPool : 2 × 2			
Conv [512, 8]			
GlobalAveragePool	GlobalMaxPool		GlobalMaxPool
Final dense layer			

Tableau 8.2: Tableau des différentes architectures complètement convolutives utilisées pour une comparaison avec des signaux de bruits acoustiques sous-marins. $LST_2(\Delta K_1, \Delta K_2, F_1, F_2)$ représente un module LST d'ordre 2 disposant respectivement d'une condition d'arrêt à l'ordre 1 ΔK_1 et à l'ordre 2 ΔK_2 , F_1 filtres à l'ordre 1 et F_2 filtres à l'ordre 2.

Architecture	Précision de classification Entraînement (%) $\pm \sigma$	Précision de classification Test (%) $\pm \sigma$
FCNN M11	94,75 \pm 0.09	94,29 \pm 0.36
EnvNet-v2	88,79 \pm 0.14	85,92 \pm 0.63
LST-CNN	94,14 \pm 0.08	93,68 \pm 0.16
CNN 2D	94,17 \pm 0.02	94,02 \pm 0.28

Tableau 8.3: Précision moyenne de classification de différentes architectures profondes sur le problème de classification de bruits acoustiques sous-marins simulés. Le symbole $\pm \sigma$ dénote l'écart type de performances des différents modèles d'une même architecture.

3.2 Influence du SNR sur l'apprentissage

Dans cette sous-section, nous reprenons une étude précédemment réalisée en interne concernant l'effet du niveau de bruit sur l'apprentissage d'architectures CNNs 1D. Pour ce faire, nous appliquons un coefficient multiplicateur à la composante de bruit lors de la génération des signaux de l'ensemble des classes, tel que :

$$x(k) = s(k) + \alpha \cdot n(k), \quad (8.2)$$

avec $s(k)$ le signal source ainsi que $n(k)$ la composante de bruit comportant du bruit ambiant ainsi qu'un bruit blanc Gaussien. Les signaux sont normalisés en leur imposant une échelle d'intensité logarithmique, tout en conservant leur signe :

$$x'(k) = \text{sign}(x(k)) \cdot 20 \log \left(\frac{|x(k)| + 1 \cdot 10^6}{1 \cdot 10^6} \right) / 120. \quad (8.3)$$

Les valeurs de SNR données par la suite désigneront le SNR moyen du jeu de données de signaux acoustiques sous-marins concerné, qui est lui-même calculé à partir du SNR moyen mesuré pour chacune des classes. Les valeurs de précision données seront celles obtenues sur un jeu de test après un entraînement similaire à celui présenté dans la section précédente et en prenant le meilleur modèle parmi 4 entraînements.

La figure 8.2 permet de visualiser la précision de test obtenue quand un modèle entraîné avec un certain niveau de bruit est testé sur un différent niveau de bruit. Comme on pourrait s'y attendre, un modèle entraîné dans des conditions de bruit plus élevées fonctionne globalement mieux dans des conditions de bruit plus faibles et les performances baissent avec le niveau de bruit. De plus, les apprentissages sur le jeu de données avec un bruit plus élevé (SNR de -12 dB) ne fonctionnent pas très bien lorsqu'ils sont testés sur le jeu de données avec le plus fort SNR. Il semble ainsi que le bruit sur cet ensemble de données soit alors trop élevé pour obtenir un apprentissage de filtres efficaces avec l'architecture CNN 1D. Qui plus est, les modèles entraînés et testés sur des jeux de données avec de faibles SNRs, comme ceux à -6 et -12 dB, fonctionnent moins bien que les modèles entraînés sur 0 dB, quand testés sur les jeux de données à -6 et -12 dB. Néanmoins, il semblerait bien qu'un minimum de bruit soit nécessaire pour permettre la généralisation, puisque les modèles entraînés à 12 dB de SNR fonctionnent moins bien sur le jeu à -12 dB que ceux à 0 et -6 dB.

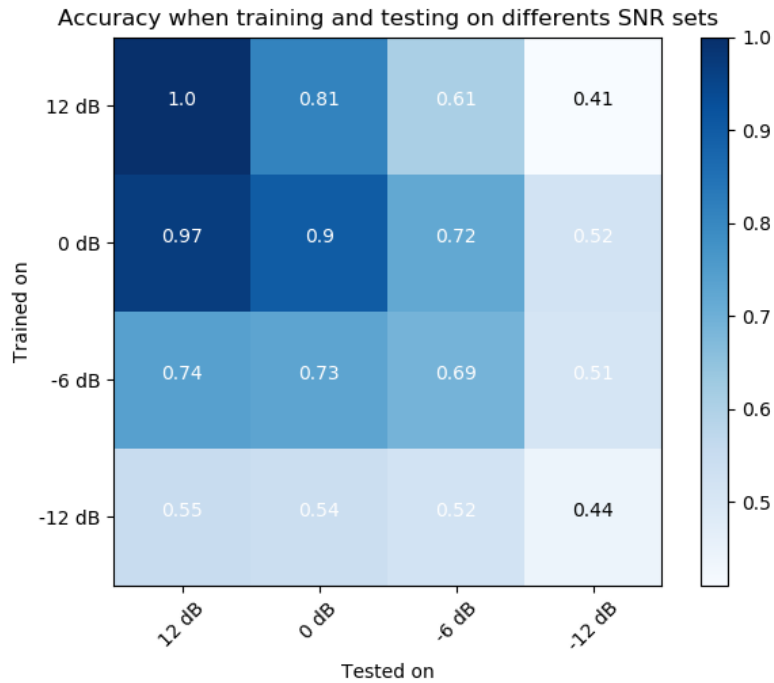


Figure 8.2: Matrice de précision de modèles CNN 1D entraînés sur des jeux disposants de différents SNR moyens, puis testés sur ces mêmes différents jeux de données.

En revanche, en pré-entraînant d'abord les modèles CNN 1D, sur un jeu de données avec de faibles niveaux de bruit, puis en les re-entraînant de bout en bout sur le même niveau de bruit que celui sur lequel ils sont testés, on obtient alors de meilleurs résultats qui sont affichés sur la figure 8.3. Là encore on peut noter que l'apprentissage ou le pré-apprentissage des filtres sur de faibles niveaux de bruit, semble préférable pour nos données ASM.

3.3 Influence des paramètres d'apprentissage sur la généralisation

Comme cela a été discuté dans le chapitre 3, le comportement de généralisation des architectures profondes ne peut pas être prédit avec des mesures de complexité classiques comme la théorie de Vapnik et Chervonenkis. En effet le nombre de couches peut avoir pour effet de réduire l'écart de généralisation g , malgré le fait qu'il augmente le nombre de paramètres de l'architecture. La taille de la batch est également un paramètre influençant la généralisation des modèles rarement pris en compte dans les mesures de complexité classiques. Ces comportements sont cependant souvent observés sur des CNNs 2D dans le domaine des images et il restait nécessaire de voir s'ils s'appliquent également aux architectures CNN 1D en ASM. Aussi, afin d'analyser l'impact qu'ont les paramètres d'apprentissage sur les performances d'architectures CNNs 1D, nous effectuons dans cette sous-section une "GridSearch" sur deux architectures FCNNs, notées f_θ et f_ν , disposant de nombres de couches L différentes, mais disposant d'un dernier ensemble de feature maps de même dimension. La figure 8.4 décrit, par le biais de couches Keras, ces deux différentes architectures profondes considérées pour la

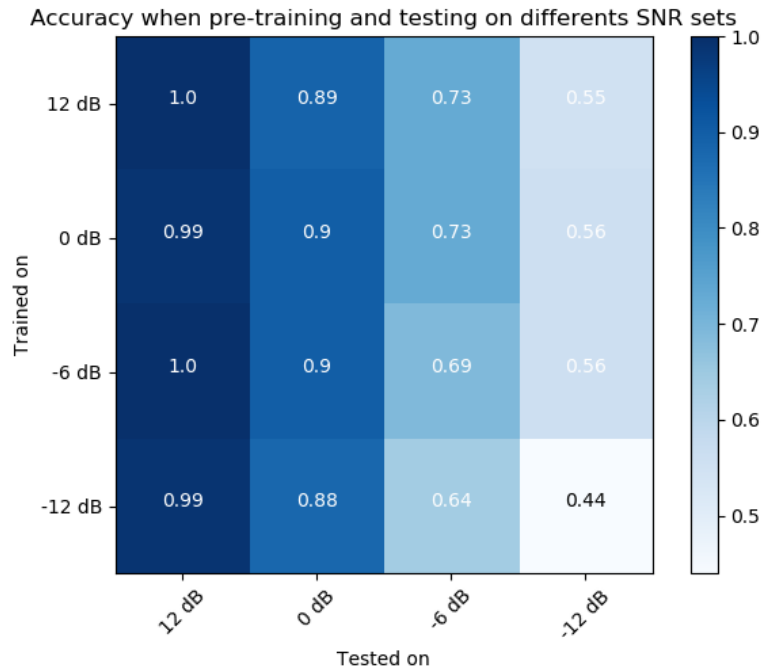


Figure 8.3: Matrice de précision de modèles CNN 1D pré-entraînés sur des jeux disposants de différents SNR moyens, puis re-entraînés et testés sur ces mêmes différents jeux de données.

"GridSearch".

L'architecture FCNN f_θ comprend une succession de 9 couches de convolution et d'opérations de pooling de taille et pas 2, suivies d'une concaténation d'opérations d'average pooling et de max pooling globales. L'architecture FCNN f_ν comprend elle une succession de 5 couches de convolution et d'opérations de pooling de taille et pas 4, suivies d'une concaténation d'opérations d'average pooling et de max pooling globales. Les deux architectures utilisent des noyaux de convolution de taille p_1 avec un pas de 4 pour la première couche et des tailles de noyaux p pour les couches suivantes. Les dernières couches de convolution des deux architectures sont appliquées à des représentations de mêmes dimensions (108, 256), avant l'opération de pooling globale. Pour sélectionner les paramètres variables pour la "GridSearch", nous reprenons des paramètres connus pour affecter la généralisation g dans le domaine des images, ainsi que le choix de taille de filtre et type d'opération de pooling locale :

- $L = \{5, 9\}$, le nombre de couches de convolution.
- $p_1 = \{256, 64\}$, la dimension des filtres de convolution sur la première couche de convolution.
- $p = \{36, 9\}$, la dimension des filtres de convolution sur les couches de convolution suivantes.
- $B = \{16, 4\}$, la taille de batch employée.
- Pool = { "Max", "Average" }, le type d'opération de pooling locale.

$f_\theta(\mathbf{x})$ ($\approx 1\,185\,655$)	$f_\nu(\mathbf{x})$ ($\approx 1\,128\,391$)	
Entrée : 110250×1 signal temporel $x(k)$		
Conv [16, p_1 , $s=4$]	Conv [64, p_1 , $s=4$]	
Pool : 2×2	Pool : 4×4	
Conv [32, p]	Conv [128, p]	
Pool : 2×2	Pool : 4×4	
Conv [32, p]	Conv [128, p]	
Pool : 2×2	Pool : 4×4	
Conv [64, p]	Conv [256, p]	
Pool : 2×2	Pool : 4×4	
Conv [64, p]	Conv [256, p]	
Pool : 2×2		
Conv [128, p]		
Pool : 2×2		
Conv [128, p]		
Pool : 2×2		
Conv [256, p]		
Pool : 2×2		
Conv [256, p]		
GlobalAveragePool + GlobalMaxPool		
Final dense layer : 7×512		

Tableau 8.4: Tableau des différentes architectures utilisées pour la "GridSearch" sur les paramètres de FCNNs. Le nombre de paramètres affiché est celui d'une des combinaisons possibles. Le symbole +, sur une même couche, se réfère à une concaténation. La valeur s , dans la couche Conv, est le pas ou la stride de convolution employée, qui vaut 1 par défaut.

Les architectures FCNNs suivent la même procédure d'optimisation sur le jeu d'apprentissage de la sous-section 3.1, avec l'utilisation de l'optimiseur Adam et de l'entropie croisée catégorique comme fonction de perte. Le critère d'arrêt est une absence d'amélioration de la perte de validation pendant 10 époques consécutives, ou au maximum après 50 époques d'entraînement. La contrainte "Maximum kernel" est utilisée pour toutes les couches convolutives afin de majorer les valeurs des poids des filtres convolutifs et nous appliquons une régularisation L^2 aux couches de sortie, de même qu'une normalisation par batch pour toutes les couches convolutives. Les performances des modèles pour chaque combinaison de la "GridSearch" sont estimés en calculant la moyenne et l'écart type des erreurs d'entraînement et de test parmi 5 entraînements différents, les modèles ayant des performances trop mauvaises étant tout de même écartés.

En se référant aux résultats sur le tableau 8.5, L'utilisation de plus de couches de convolutions semble permettre d'obtenir de meilleures performances de classification sur les ensembles d'entraînement et de test, il est cependant difficile de statuer quant à l'écart de généralisation g avec aussi peu d'entraînements.

Nombre de couches L	Précision de classification Entraînement (%) $\pm \sigma$	Précision de classification Test (%) $\pm \sigma$	Erreur de généralisation g (%)
5	92,54 \pm 0,8	92,57 \pm 2,3	0,03
9	94,19 \pm 0,9	93,87 \pm 1,8	0,32

Tableau 8.5: Précision de classification des architectures f_θ et f_ν pour un nombre de couches différent.

En prenant les résultats visibles sur le tableau 8.6, l'utilisation de taille de filtre 256 ou 64 n'impacte que peu les performances. De même il est difficile de statuer quant à la généralisation avec de tels écarts types. Néanmoins, avec quelques essais, l'utilisation d'un premier filtre de dimension 2048 entraîne cependant de plus importantes erreurs de généralisation.

Dimension de filtre p_1	Précision de classification Entraînement (%) $\pm \sigma$	Précision de classification Test (%) $\pm \sigma$	Erreur de généralisation g (%)
256	93,02 \pm 1,9	92,98 \pm 2,2	0,04
64	92,67 \pm 0,9	91,31 \pm 2,6	1,36

Tableau 8.6: Précision de classification des architectures f_θ et f_ν pour différentes dimensions de filtres de convolution sur la première couche.

Des résultats présentés sur le tableau 8.7, à première vue, l'utilisation de taille de filtres 36 ou 9 n'impacte que peu les performances. Dans le détail cependant, f_θ , avec ses 9 couches, obtient de nettement meilleures performances avec des filtres de taille 9 et f_ν , avec ses 5 couches, obtient de nettement meilleures performances avec des filtres de taille 36. Il semble

ainsi que le choix de la taille des filtres de convolution doit être intrinsèquement lié au nombre de couches.

Dimension de filtre p	Précision de classification Entraînement (%) $\pm \sigma$	Précision de classification Test (%) $\pm \sigma$	Erreur de généralisation g (%)
36	91,62 \pm 3,7 (f_θ : 88,49 et f_ν : 94,75)	89,68 \pm 3,1 (f_θ : 87,41 et f_ν : 91,95)	1,94
9	92,11 \pm 4,1 (f_θ : 96,23 et f_ν : 87,99)	91,07 \pm 2,5 (f_θ : 94,01 et f_ν : 88,13)	0,14

Tableau 8.7: Précision de classification des architectures f_θ et f_ν pour différentes dimensions de filtres de convolution avec précision des valeurs obtenues pour chacune des deux architectures séparément.

En se référant aux résultats sur le tableau 8.8, l'utilisation de batchs de taille 16 permet d'obtenir de meilleures performances de classification qu'avec une taille de batch 4. Il est cependant difficile de se prononcer quant aux performances de généralisation avec un tel écart type. Il aurait également été préférable d'utiliser des batchs plus grands pour constater un écart de généralisation, mais nous sommes limités en termes de mémoire GPU avec une GeForce RTX 2080 Ti de 8 Go de RAM pour cette expérimentation.

Taille de batch B	Précision de classification Entraînement (%) $\pm \sigma$	Précision de classification Test (%) $\pm \sigma$	Erreur de généralisation g (%)
16	93,29 \pm 1,2	93,15 \pm 2,1	0,14
4	88,40 \pm 0,09	88,35 \pm 1,6	0,05

Tableau 8.8: Précision de classification des architectures f_θ et f_ν pour des tailles de batchs différentes.

En analysant les résultats sur le tableau 8.9, l'utilisation de l'opération de max pooling induit une plus grande erreur de généralisation g de façon assez nette.

Type de pooling $Pool$	Précision de classification Entraînement (%) $\pm \sigma$	Précision de classification Test (%) $\pm \sigma$	Erreur de généralisation g (%)
"Max",	93,13 \pm 1,1	87,52 \pm 2,3	2,61
"Average"	92,23 \pm 0,8	92,02 \pm 0,4	0,21

Tableau 8.9: Précision de classification des architectures f_θ et f_ν pour un type de pooling différent.

3.4 Mesure de complexité sur modèles appris

Après avoir analysé l'influence des paramètres sur l'apprentissage de réseaux FCNNs, nous avons essayé d'utiliser des mesures de complexité, vues dans la section 4 du chapitre 3, afin de voir si certaines d'entre elles permettaient de déterminer quels modèles déjà appris généraliseraient moins bien que d'autres. En se référant à l'article de [Vakanski & Xian \(2021\)](#) et certaines de ses références, les mesures PAC bayésiennes de "flatness", fortement liées à la notion de netteté vue dans la sous-section 4.6, semblent être des mesures assez consistantes. Les mesures liées à la norme des poids ne semblent pas être utilisables ici du fait que nous appliquons déjà des contraintes et régularisations sur les poids de nos réseaux. De façon similaire, les mesures basées sur la dimension VC donneraient toujours une même valeur pour des modèles d'une même architecture. Enfin, les mesures d'optimisation liées au nombre d'époques nécessaires pour parvenir à une certaine erreur ont été très inconsistantes sur notre cas d'application.

Aussi, sans pouvoir trouver trop d'implémentations "Open Source" de ces mesures à l'heure actuelle, nous nous sommes limités à vérifier sur nos modèles le comportement d'une mesure PAC-bayésienne de "flatness" μ décrite dans l'article de [Vakanski & Xian \(2021\)](#). Pour la calculer, on ajoute des perturbations gaussiennes $\nu \sim \mathcal{N}(\theta, \sigma^2 \cdot I_D)$ sur les poids θ d'un modèle entraîné f_θ disposant de D paramètres. La mesure est ensuite calculée telle que :

$$\mu = \frac{1}{\sigma_\theta^2}, \quad (8.4)$$

avec σ_θ , la plus grande valeur de variance de la perturbation ν , telle qu'on ne dégrade pas les performances de f_θ de plus de Δe % :

$$\sigma_\theta = \arg \max_{\sigma} \mathbb{E}_{\nu \sim \mathcal{N}(\theta, \sigma^2 \cdot I_D)} [\mathcal{R}_N(f_\theta) - \mathcal{R}_N(f_{\theta+\nu})] \leq \Delta e. \quad (8.5)$$

Dans la pratique nous avons effectué une dichotomie sur des valeurs $\sigma = 2^m$, avec m un entier. Pour les expérimentations, nous avons pris en compte la valeur d'écart de performances d'un modèle f_θ sur seulement 2 tirages de bruits de perturbation ν et avec un écart de performance $\Delta e = 10\%$. Des quelques essais effectués, nos modèles ayant une erreur de généralisation $g > 5\%$ ont bien des mesures μ supérieures à la mesure maximale, $\mu_{g < 2\%}$, trouvée sur les modèles ayant des erreurs de généralisation $g < 2\%$.

4 Conclusion

Dans ce chapitre nous avons détaillé les protocoles expérimentaux ainsi que les résultats obtenus sur certaines expérimentations d'intérêt réalisées durant la thèse. La sous-section 2.1 a montré que notre module LST était capable d'apprendre sur un jeu de données réel, sans initialisations des filtres. La sous-section 3.2 a permis de constater qu'un modèle entraîné au sein d'un environnement trop bruité n'était pas exploitable, tandis qu'un entraînement sur trop peu de bruit ne fonctionnait pas très bien sur des niveaux de SNR moyens. Les expérimentations sur "GridSearch", de la sous-section 3.3, ont fourni des résultats sur l'impact du choix de paramètres sur les performances de réseaux FCCNs. Enfin, bien que les résultats de la section 3.4 doivent être pris avec des pincettes du fait que trop peu de modèles aient été utilisés, elles ont cependant permis de mettre en avant l'existence de mesures de complexité d'expressions simples et calculables en des temps "raisonnables".

Chapitre 9

Conclusions et perspectives suite à la thèse

1 Conclusions

Les architectures profondes telles que ResNet, ont récemment montré qu'elles permettaient d'obtenir des performances de classification bien supérieures aux algorithmes d'apprentissage machine basés sur des représentations expertes. En effet, en tirant avantage de l'expressivité de réseaux plus profonds ainsi que des structures de décisions plus récentes comme les mécanismes d'attention, ces architectures sont susceptibles d'extraire des caractéristiques décrivant un type de données de façon bien plus efficace que ce que pourrait proposer un être humain. Dans le domaine acoustique, nous avons ainsi pu voir que les performances d'architectures très profondes surpassaient les algorithmes d'apprentissage automatique classiques. Les méthodes employant des CNNs 2D, sur des représentations temps fréquences, permettent d'obtenir d'excellentes performances quand peu de données sont disponibles, tandis que les architectures CNNs 1D peuvent potentiellement apprendre de meilleures représentations internes de bout en bout, avec des architectures plus complexes. Ces algorithmes de pointe sont malheureusement souvent conçus pour un objectif de performances, sans trop tenir compte de facteurs de robustesse, de fiabilité ou encore d'explicabilité. Aussi, du fait de leurs cascades de convolutions multicanal et d'opérations internes non linéaires, ces algorithmes sont perçus comme des boîtes noires dont on ne peut comprendre le fonctionnement.

La théorie de l'apprentissage, vue dans le chapitre 3, permet aujourd'hui de s'assurer que les performances d'un modèle restent bornées autour d'un comportement cible pour des données suivant une distribution prédéfinie. Dans ce sens, les mesures de complexités récentes tentent de toujours plus affiner les bornes de généralisation des architectures profondes, en se basant sur divers critères, soupçonnés de caractériser et de coller au comportement de leurs erreurs d'estimation. Aussi, bien que même dans un cadre maîtrisé nous soyons encore loin de pouvoir complètement caractériser le processus d'apprentissage d'une architecture profonde, ou encore de déterminer la procédure d'apprentissage optimale, ces travaux permettent de déduire de bonnes intuitions pour concevoir des architectures plus efficaces.

Du côté de l'explicabilité, nous avons pu voir dans le chapitre 2, différentes méthodes per-

mettant d'interpréter la décision d'un algorithme de classification profond, à travers l'ajout de visualisations ou la prédiction forcée d'informations supplémentaires. Ces approches ne changent cependant pas le fait que les architectures profondes fonctionnent comme des boîtes noires. Durant la thèse nous avons ainsi essayé de nous baser sur des structures mathématiques, déjà connues dans le traitement du signal pour leurs propriétés, afin de concevoir des architectures profondes qui puissent être intrinsèquement explicables. Dans le cadre maîtrisé de la classification de bruit d'hélice, basée sur un modèle acoustique simple et présentée dans les chapitres 5 et 6, nous avons ainsi pu proposer des structures d'architectures profondes permettant d'approximer efficacement la structure mathématique de détecteurs de Bayes optimaux. L'explicabilité de ces architectures émanait alors de leur capacité à approximer chaque opération effectuée par les détecteurs bayésiens de façon indépendante, grâce à des modules d'approximation de fonction efficaces en termes de nombre de paramètres et visualisables. Une de leur limitation était cependant la taille excessive des filtres adaptées devant être appris, afin d'intégrer les signaux à analyser sur de longues durées.

Dans le chapitre 7 nous proposons une implémentation en tant que CNN de l'algorithme de la scattering transform proposé par Mallat (2012) et ce, afin d'obtenir une structure de représentation apprise explicable. En utilisant cette approche nous pouvons en effet conserver certaines propriétés mathématiques imposées par la structure de la scattering transform afin d'obtenir une certaine forme de robustesse. Une architecture profonde classique pourrait également apprendre une structure mathématiques robuste, si on se base sur son expressivité, mais ce n'est jamais une garantie. L'optimisation non-convexe d'architectures profondes tend généralement à obtenir des minima locaux qui ne sont satisfaisants que pour la précision de classification, les objectifs de robustesse n'étant pas intégrés dans la fonction de perte. Aussi, bien que cette démarche ait des performances réduites comparée à des architectures très profondes plus complexes, et soit également moins bien maîtrisée qu'une représentation fixe, c'est un bon compromis entre boîtes noires et algorithmes transparents, capable de s'adapter à un problème particulier, mais moins enclin à complètement le surapprendre.

Au final, les architectures explicables que nous avons proposées sont étroitement liées à la théorie de l'approximation, puisque nous connaissons assez bien leur expressivité. Elles sont en effet capables de profiter de l'un des principaux avantages de l'apprentissage profond : la puissance d'expressivité due à la profondeur du réseau. Cela nous a ainsi permis de proposer des architectures très efficaces en termes de nombre de paramètres. Ceci s'est reflété par des erreurs de généralisation relativement faibles sur nos jeux de données acoustiques simulés, avec une forme d'apprentissage frugal pour le module LST. Le module LST CNN, avec dépendances temporelles et apprentissage de bout en bout, a également permis d'obtenir des performances encourageantes sur des petits jeux de données réels, là où les CNNs 1D ont généralement des performances inférieures aux CNNs 2D sur représentations temps fréquence.

2 Perspectives pour la suite des travaux

2.1 Perspectives pour le module LST

En l'absence de larges jeux de données "Open Source" issus de l'ASM, nous n'avons pas pu évaluer en profondeur l'efficacité de notre architecture de scattering transform apprise dans

ce domaine et il reste ainsi difficile de se prononcer sur son utilité réelle. Sa structure imposée laisserait en effet penser qu'elle pourrait être plus robuste qu'une architecture profonde conventionnelle, sur des données réelles jamais observées. Beaucoup d'autres aspects d'intérêts ont également dus être mis de côté par manque de temps ou de capacités de calcul. Il serait ainsi intéressant de voir si des solutions d'ingénieries comme l'ajout de couches d'attention, un pré-entraînement des poids ou certaines formes de normalisations, permettraient d'améliorer les performances de notre module LST.

Une comparaison entre la scattering transform classique, comme représentation fixe et notre module LST serait aussi intéressante. En effet, l'implémentation de la scattering transform dans Kymatio permet d'utiliser beaucoup de filtres par octave et d'obtenir plusieurs représentations avec des ondelettes mères différentes, alors que notre module LST a des limitations de mémoire. Il est alors envisageable d'apprendre différents ensembles de filtres d'ondelettes au cours de différents apprentissages, puis de les intégrer dans une représentation fixe avec Kymatio. Dans notre implémentation du LST CNN, il serait également nécessaire de vérifier que toutes les versions d'un même filtre, reçoivent des mises à jour des poids provenant de chaque échelle temporelle de façon équitable, une problématique importante dans les couches de "skip connection".

Dans un aspect plus exploratoire, il serait intéressant de voir si une scattering transform apprise 2D peut être utile pour la classification d'images. De plus, après avoir implémenté le module LST d'ordre 1 en tant qu'encodeur au sein d'un auto encodeur et d'un VAE, on pourrait se demander si son efficacité en termes de paramètres permettrait en faire une structure d'encodeur ou de décodeur efficace au sein d'un GAN ou dans d'autres types de générateurs ainsi que dans des applications de débruitage.

2.2 Perspectives pour la classification acoustique sous-marine

Durant la thèse, nous nous sommes limités à un cas d'usage simple où nous devons classifier des signaux détectés sur un critère énergétique, après formation de voie. De cette manière nous pouvons aider l'utilisateur final en l'assistant sur des tâches simples. Dans ce cadre, nous nous sommes surtout focalisés sur les signaux cyclostationnaires, mais il peut également y avoir un intérêt à classifier des transitoires, notamment concernant les communications acoustiques. Il serait alors important d'étudier l'impact de l'utilisation d'opérations d'average pooling locales par rapport aux max pooling locaux, comme le font [Dinkel & Yu \(2020\)](#) et [Kao et al. \(2020\)](#). En effet, l'utilisation d'une agrégation par maximum local est peut-être plus cohérente pour des transitoires que pour des signaux stationnaires. Il existe alors des possibilités pour apprendre une opération max-average pooling, où un facteur est appris pour pondérer les deux opérations. On pourrait aussi envisager de concaténer ces opérations, au prix d'obtenir deux fois plus de feature maps après l'opération. De même, les différents mécanismes d'attention n'ont pas été comparés durant la thèse et comme le montrent [Gong et al. \(2021\)](#), il est possible d'obtenir d'excellents résultats pour la détection d'évènements acoustiques avec des architectures profondes de pure auto-attention, sans couches de convolutions, mais avec des opérations équivalentes non apprises.

Concernant la théorie de l'optimisation, elle est souvent étudiée du point de vue du domaine

des images où d'autres problématiques sont mises en jeu, et il est souvent possible de complètement séparer les données. Plus de recherches sont ainsi nécessaires pour voir si notre cas d'apprentissage, où le bruit peut être prédominant, pourrait nécessiter des algorithmes d'apprentissage plus adaptés qu'une simple SGD ou Adam. De même, la classification multi-classes multi-labels aurait peut-être nécessité une autre comparaison d'architectures profondes, avec des métriques plus adaptées ainsi qu'une "GridSearch" sur les algorithmes d'apprentissage.

Enfin, il restera toujours nécessaire d'intégrer le contexte opérationnel afin d'obtenir de meilleures performances avec l'apport d'informations complémentaires. En dehors du cadre de classification simple prévue dans cette thèse, l'implémentation d'algorithmes profonds pour la classification de signaux ASM, ouvre la voie à de nombreuses thématiques de recherches. On peut par exemple imaginer de ne plus se limiter aux signaux 1D après formation de voie, mais profiter de plusieurs canaux, comme cela est fait par [van den Ende et al. \(2021\)](#) où les informations de différents capteurs sont utilisées pour l'analyse du trafic routier. Dans le domaine ASM, [Ferguson et al. \(2018\)](#) utilisent comme entrée d'un CNN 2D, les corrélations croisées généralisées (GCC) ainsi que les MFFCs calculés sur chaque capteur d'une antenne. L'architecture qu'ils emploient est ainsi conçue pour obtenir à la fois les informations de retards, liées à la directivité du front d'onde, et des informations pouvant caractériser une source. Une solution moins complexe envisagée durant la thèse, serait peut-être d'analyser simplement le signal en sortie de formation de voie, ainsi que le signal provenant d'une voie adjacente. Cette solution est en effet souvent effectuée dans la pratique par des utilisateurs de sonar et leur permet d'inférer la nature du bruit ambiant capturée dans la direction de la source principale.

Les thématiques explorées durant la thèse pourraient ainsi constituer d'importantes briques pour la réalisation d'un système expert plus complexe. En effet, comme le soulignent [LeCun et al. \(2015\)](#), la plus grande force de l'apprentissage profond est sa capacité à apprendre des représentations capturant la structure des données dans de larges jeux de données, et pas nécessairement l'application qu'on en fait. Dans le domaine acoustique, la classification de sources potentielles via des distances au sein de bases de données déjà labialisées, est par exemple une solution assez répandue pouvant exploiter ces représentations apprises. Une étude de différentes distances serait alors à envisager, certaines d'entre elles étant d'ailleurs adaptées aux représentations temps-fréquence, comme une dérivée de l'entropie de Rényi, présentée dans les travaux de [Baraniuk et al. \(2001\)](#).

Bibliographie

- Abeßer, J. (2020), ‘A review of deep learning based methods for acoustic scene classification’, *Applied Sciences* **10**(6).
- Adadi, A. & Berrada, M. (2018), ‘Peeking inside the black-box : a survey on explainable artificial intelligence (xai)’, *IEEE access* **6**, 52138–52160.
- Andén, J. & Mallat, S. (2014), ‘Deep scattering spectrum’, *IEEE Transactions on Signal Processing* **62**(16), 4114–4128.
- Andreux, M., Angles, T., Exarchakis, G., Leonarduzzi, R., Rochette, G., Thiry, L., Zarka, J., Mallat, S., Andén, J., Belilovsky, E. et al. (2020), ‘Kymatio : Scattering transforms in python.’, *J. Mach. Learn. Res.* **21**(60), 1–6.
- Antoni, J. (2009), ‘Cyclostationarity by examples’, *Mechanical Systems and Signal Processing* **23**(4), 987–1036.
- Asadi, B. & Jiang, H. (2020), ‘On approximation capabilities of relu activation and softmax output layer in neural networks’, *arXiv preprint arXiv :2002.04060* .
- Atto, A. M. & Berthoumieu, Y. (2011), ‘Wavelet transforms of non-stationary random processes : Contributing factors for stationarity and decorrelation’, *Preprint,[Available Online]* .
- Au, W. W. & Banks, K. (1998), ‘The acoustics of the snapping shrimp *synalpheus parneomeris* in kaneohe bay’, *The Journal of the Acoustical Society of America* **103**(1), 41–47.
- Azimi-Sadjadi, M. R., Yao, D., Huang, Q. & Dobeck, G. J. (2000), ‘Underwater target classification using wavelet packets and neural networks’, *IEEE Transactions on Neural Networks* **11**(3), 784–794.
- Bagheri, M., Seif, M. & Mehdigholi, H. (2014), ‘Numerical simulation of underwater propeller noise’, *Journal of Ocean, Mechanical and Aerospace, Science and Engineering* **4**.
- Baraniuk, R. G., Flandrin, P., Janssen, A. J. & Michel, O. J. (2001), ‘Measuring time-frequency information content using the rényi entropies’, *IEEE Transactions on Information theory* **47**(4), 1391–1409.
- Bartlett, P. L., Maiorov, V. & Meir, R. (1998), ‘Almost linear vc-dimension bounds for piecewise polynomial networks’, *Neural computation* **10**(8), 2159–2173.

- Baygun, B. & Hero, A. O. (1995), ‘Optimal simultaneous detection and estimation under a false alarm constraint’, *IEEE Transactions on Information Theory* **41**(3), 688–703.
- Berg, H. & Hjelmervik, K. T. (2018), Classification of anti-submarine warfare sonar targets using a deep neural network, *in* ‘OCEANS 2018 MTS/IEEE Charleston’, IEEE, pp. 1–5.
- Bordes, J.-P. (1996), ‘À l’écoute du grand bleu’, *Cols Bleus* *2353*.
- Bousquet, O., Boucheron, S. & Lugosi, G. (2003), Introduction to statistical learning theory, *in* ‘Summer school on machine learning’, Springer, pp. 169–207.
- Bouvet, M. & Schwartz, S. C. (1988), ‘Underwater noises : Statistical modeling, detection, and normalization’, *The Journal of the Acoustical Society of America* **83**(3), 1023–1033.
- Brooks, D., Cord, M. & Schwander, O. (2020), Deep Learning and Information Geometry for Time-Series Classification, PhD thesis, Sorbonne Universite.
- Burdic, W. S. & Bartram, J. F. (1984), ‘Underwater acoustic system analysis by william s. burdic’.
- Byun, S.-H., Kim, S.-M., Park, C., Kim, K. & Lee, C.-M. (2016), Cyclostationary analysis of underwater noise for vehicle propeller monitoring, *in* ‘OCEANS 2016 MTS/IEEE Monterey’, IEEE, pp. 1–4.
- Cai, W., Cai, Z., Zhang, X., Wang, X. & Li, M. (2018), A novel learnable dictionary encoding layer for end-to-end language identification, *in* ‘2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)’, IEEE, pp. 5189–5193.
- Cakır, E., Parascandolo, G., Heittola, T., Huttunen, H. & Virtanen, T. (2017), ‘Convolutional recurrent neural networks for polyphonic sound event detection’, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **25**(6), 1291–1303.
- Cao, X., Zhang, X., Togneri, R. & Yu, Y. (2019), ‘Underwater target classification at greater depths using deep neural network with joint multiple-domain feature’, *IET Radar, Sonar & Navigation* **13**(3), 484–491.
- Cao, X., Zhang, X., Yu, Y. & Niu, L. (2016), Deep learning-based recognition of underwater target, *in* ‘2016 IEEE International Conference on Digital Signal Processing (DSP)’, IEEE, pp. 89–93.
- Chaudhari, P. & Soatto, S. (2018), Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks, *in* ‘2018 Information Theory and Applications Workshop (ITA)’, IEEE, pp. 1–10.
- Chen, Y.-h. & Liu, Z.-q. (2016), Ship propeller broadband noise prediction using the les and caa, *in* ‘2016 IEEE/OES China Ocean Acoustics (COA)’, IEEE, pp. 1–4.
- Chung, K. W., Sutin, A., Sedunov, A. & Bruno, M. (2011), ‘Demon acoustic ship signature measurements in an urban harbor’, *Advances in Acoustics and Vibration* **2011**.

- Clark, P., Kirsteins, I. P. & Atlas, L. E. (2010), ‘Multiband analysis for colored amplitude-modulated ship noise’, *2010 IEEE International Conference on Acoustics, Speech and Signal Processing* pp. 3970–3973.
- Cybenko, G. (1989), ‘Approximation by superpositions of a sigmoidal function’, *Mathematics of control, signals and systems* **2**(4), 303–314.
- Dahl, P. H., Miller, J. H., Cato, D. H. & Andrew, R. K. (2007), ‘Underwater ambient noise’, *Acoustics Today* **3**(1), 23–33.
- Dai, W., Dai, C., Qu, S., Li, J. & Das, S. (2017), Very deep convolutional neural networks for raw waveforms, in ‘2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)’, IEEE, pp. 421–425.
- Das, A., Kumar, A. & Bahl, R. (2013), ‘Marine vessel classification based on passive sonar data : the cepstrum-based approach’, *IET Radar, Sonar & Navigation* **7**(1), 87–93.
- Deng, L., Hinton, G. & Kingsbury, B. (2013), New types of deep neural network learning for speech recognition and related applications : An overview, in ‘2013 IEEE international conference on acoustics, speech and signal processing’, IEEE, pp. 8599–8603.
- Devroye, L. (1982), ‘Any discrimination rule can have an arbitrarily bad probability of error for finite sample size’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2**, 154–157.
- Devroye, L., Györfi, L. & Lugosi, G. (2013), *A probabilistic theory of pattern recognition*, Vol. 31, Springer Science & Business Media.
- Devroye, L., Krzy, A. et al. (1989), ‘An equivalence theorem for l_1 convergence of the kernel regression estimate’, *Journal of statistical planning and Inference* **23**(1), 71–82.
- Dinkel, H. & Yu, K. (2020), Duration robust weakly supervised sound event detection, in ‘ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)’, IEEE, pp. 311–315.
- Doshi-Velez, F. & Kim, B. (2017), ‘Towards a rigorous science of interpretable machine learning’, *arXiv preprint arXiv :1702.08608* .
- Du, S., Lee, J., Li, H., Wang, L. & Zhai, X. (2019), Gradient descent finds global minima of deep neural networks, in ‘International Conference on Machine Learning’, PMLR, pp. 1675–1685.
- Dyer, I. (1973), ‘Statistics of distant shipping noise’, *The Journal of the Acoustical Society of America* **53**(2), 564–570.
- Elbrächter, D., Perekrestenko, D., Grohs, P. & Bölcskei, H. (2019), ‘Deep neural network approximation theory’, *arXiv preprint arXiv :1901.02220* .
- Espi, M., Fujimoto, M., Kinoshita, K. & Nakatani, T. (2015), ‘Exploiting spectro-temporal locality in deep learning based acoustic event detection’, *EURASIP Journal on Audio, Speech, and Music Processing* **2015**(1), 1–12.

- Esseen, C.-G. (1956), ‘A moment inequality with an application to the central limit theorem’, *Scandinavian Actuarial Journal* **1956**(2), 160–170.
- Faragó, A. & Lugosi, G. (1993), ‘Strong universal consistency of neural network classifiers’, *IEEE Transactions on Information Theory* **39**(4), 1146–1151.
- Fel, T. & Vigouroux, D. (2020), ‘Representativity and consistency measures for deep neural network explanations’, *arXiv preprint arXiv :2009.04521* .
- Ferguson, E. L., Williams, S. B. & Jin, C. T. (2018), Sound source localization in a multipath environment using convolutional neural networks, *in* ‘2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)’, IEEE, pp. 2386–2390.
- Friedman, J. H. (1987), ‘Exploratory projection pursuit’, *Journal of the American statistical association* **82**(397), 249–266.
- Fukushima, K. & Miyake, S. (1982), Neocognitron : A self-organizing neural network model for a mechanism of visual pattern recognition, *in* ‘Competition and cooperation in neural nets’, Springer, pp. 267–285.
- Furui, S. (1986), Speaker-independent isolated word recognition based on emphasized spectral dynamics, *in* ‘ICASSP’86. IEEE International Conference on Acoustics, Speech, and Signal Processing’, Vol. 11, IEEE, pp. 1991–1994.
- Gauthier, S., Th’erien, B., Alsene-Racicot, L., Rish, I., Belilovsky, E., Eickenberg, M. & Wolf, G. (2021), ‘Parametric scattering networks’, *ArXiv* **abs/2107.09539**.
- Gemmeke, J. F., Ellis, D. P., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M. & Ritter, M. (2017), Audio set : An ontology and human-labeled dataset for audio events, *in* ‘2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)’, IEEE, pp. 776–780.
- Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M. & Kagal, L. (2018), Explaining explanations : An overview of interpretability of machine learning, *in* ‘2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)’, IEEE, pp. 80–89.
- Girard, T. & Roi, P. (2013), ‘Analogie entre la harpe urukéenne et le système auditif (analogy between the urukean harp and the auditory system)’, *Roi, Philippe et Girard, Tristan, «La harpe urukéenne et le système auditif», La Théorie Sensorielle. I-Les Analogies Sensorielles, First Edition Design Publishing (2013) pp pp. 83–100.*
- Golik, P., Tüske, Z., Schlüter, R. & Ney, H. (2015), Convolutional neural networks for acoustic modeling of raw time signal in lvcsr, *in* ‘Sixteenth annual conference of the international speech communication association’.
- Gong, Y., Chung, Y.-A. & Glass, J. (2021), ‘Ast : Audio spectrogram transformer’, *arXiv preprint arXiv :2104.01778* .
- Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep learning*, MIT press.
- Guedj, B. (2019), ‘A primer on pac-bayesian learning’, *arXiv preprint arXiv :1901.05353* .

- Hanin, B. & Rolnick, D. (2019), ‘Deep relu networks have surprisingly few activation patterns’, *arXiv preprint arXiv :1906.00904* .
- Hanson, D., Antoni, J., Brown, G. & Emslie, R. (2008), ‘Cyclostationarity for passive underwater detection of propeller craft : A development of demon processing’, *Proceedings of Acoustics 2008* pp. 24–26.
- Hardt, M., Recht, B. & Singer, Y. (2016), Train faster, generalize better : Stability of stochastic gradient descent, *in* ‘International Conference on Machine Learning’, PMLR, pp. 1225–1234.
- Harvey, N., Liaw, C. & Mehrabian, A. (2017), Nearly-tight vc-dimension bounds for piecewise linear neural networks, *in* ‘Conference on learning theory’, PMLR, pp. 1064–1068.
- He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, *in* ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 770–778.
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. et al. (2012), ‘Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups’, *IEEE Signal processing magazine* **29**(6), 82–97.
- Hong, F., Liu, C., Guo, L., Chen, F. & Feng, H. (2021), Underwater acoustic target recognition with resnet18 on shipsear dataset, *in* ‘2021 IEEE 4th International Conference on Electronics Technology (ICET)’, IEEE, pp. 1240–1244.
- Hu, G., Wang, K., Peng, Y., Qiu, M., Shi, J. & Liu, L. (2018), ‘Deep learning methods for underwater target feature extraction and recognition’, *Computational intelligence and neuroscience* **2018**.
- Hurd, H. L. (1988), ‘Spectral coherence of nonstationary and transient stochastic processes’, *Fourth Annual ASSP Workshop on Spectrum Estimation and Modeling* pp. 387–390.
- Izacard, G., Mohan, S. & Fernandez-Granda, C. (2019), ‘Data-driven estimation of sinusoid frequencies’, *arXiv preprint arXiv :1906.00823* .
- Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D. & Bengio, S. (2019), ‘Fantastic generalization measures and where to find them’, *arXiv preprint arXiv :1912.02178* .
- Kamal, S., Mohammed, S. K., Pillai, P. S. & Supriya, M. (2013), Deep learning architectures for underwater target recognition, *in* ‘2013 Ocean Electronics (SYMPOL)’, IEEE, pp. 48–54.
- Kao, C.-C., Sun, M., Wang, W. & Wang, C. (2020), A comparison of pooling methods on lstm models for rare acoustic event classification, *in* ‘ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)’, IEEE, pp. 316–320.
- Kendall, A., Gal, Y. & Cipolla, R. (2018), Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, *in* ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 7482–7491.

- Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M. & Tang, P. T. P. (2016), ‘On large-batch training for deep learning : Generalization gap and sharp minima’, *arXiv preprint arXiv :1609.04836* .
- Kim, B.-N., Hahn, J., Choi, B. K. & Kim, B.-C. (2009), 1p6-8 acoustic characteristics of pure snapping shrimp noise measured under laboratory conditions, *in* ‘Proceedings of symposium on ultrasonic electronics’, Vol. 30, Institute for Ultrasonic Electronics, pp. 167–168.
- Kirsteins, I., Clark, P. & Atlas, L. (2011), ‘Maximumlikelihood estimation of propeller noise modulation characteristics’, *Underwater Acoustic Measurements : Technologies and Results* .
- Kong, Q., Xu, Y., Wang, W. & Plumbley, M. D. (2020), ‘Sound event detection of weakly labelled data with cnn-transformer and automatic threshold optimization’, *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **28**, 2450–2460.
- Kozlenko, M., Lazarovych, I., Tkachuk, V. & Kuz, M. (2020), Deep learning demodulation of amplitude noise shift keying spread spectrum signals, *in* ‘2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)’, IEEE, pp. 717–720.
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), ‘Imagenet classification with deep convolutional neural networks’, *Advances in neural information processing systems* **25**, 1097–1105.
- Kulkarni, S. R. & Harman, G. (2011), ‘Statistical learning theory : a tutorial’, *Wiley Interdisciplinary Reviews : Computational Statistics* **3**(6), 543–556.
- Kumar, A. & Ithapu, V. (2020), A sequential self teaching approach for improving generalization in sound event recognition, *in* ‘International Conference on Machine Learning’, PMLR, pp. 5447–5457.
- Kumar, A., Khadkevich, M. & Fügen, C. (2018), Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes, *in* ‘2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)’, IEEE, pp. 326–330.
- Larochelle, H. & Hinton, G. E. (2010), ‘Learning to combine foveal glimpses with a third-order boltzmann machine’, *Advances in neural information processing systems* **23**, 1243–1251.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015), ‘Deep learning’, *nature* **521**(7553), 436–444.
- Li, J., Sun, Y., Su, J., Suzuki, T. & Huang, F. (2020), Understanding generalization in deep learning via tensor methods, *in* ‘International Conference on Artificial Intelligence and Statistics’, PMLR, pp. 504–515.
- Li, Y. & Liang, Y. (2018), ‘Learning overparameterized neural networks via stochastic gradient descent on structured data’, *arXiv preprint arXiv :1808.01204* .
- Li, Y., Liu, M., Drossos, K. & Virtanen, T. (2020), Sound event detection via dilated convolutional recurrent neural networks, *in* ‘ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)’, IEEE, pp. 286–290.

- Liang, T., Poggio, T., Rakhlin, A. & Stokes, J. (2019), Fisher-rao metric, geometry, and complexity of neural networks, *in* 'The 22nd International Conference on Artificial Intelligence and Statistics', PMLR, pp. 888–896.
- Linardatos, P., Papastefanopoulos, V. & Kotsiantis, S. (2021), 'Explainable ai : A review of machine learning interpretability methods', *Entropy* **23**(1), 18.
- Lopez-Meyer, P., del Hoyo Ontiveros, J. A., Lu, H. & Stemmer, G. (2021), Efficient end-to-end audio embeddings generation for audio classification on target applications, *in* 'ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 601–605.
- Lourens, J. G. (1997), Passive sonar ml estimator for ship propeller speed, *in* 'Proceedings of the 1997 South African Symposium on Communications and Signal Processing. COM-SIG'97', IEEE, pp. 13–18.
- Lu, J., Song, S., Hu, Z. & Li, S. (2020), Fundamental frequency detection of underwater acoustic target using demon spectrum and cnn network, *in* '2020 3rd International Conference on Unmanned Systems (ICUS)', IEEE, pp. 778–784.
- Ma, B. B., Nystuen, J. A. & Lien, R.-C. (2005), 'Prediction of underwater sound levels from rain and wind', *The journal of the acoustical society of America* **117**(6), 3555–3565.
- Mahiout, T., Fillatre, L. & Deruaz-Pepin, L. (2020), Propeller noise detection with deep learning, *in* 'ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 306–310.
- Mallat, S. (1999), *A wavelet tour of signal processing*, Elsevier.
- Mallat, S. (2012), 'Group invariant scattering', *Communications on Pure and Applied Mathematics* **65**(10), 1331–1398.
- Mallat, S. (2016), 'Understanding deep convolutional networks', *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences* **374**(2065), 20150203.
- Manikas, A. (2009), *Principles of M-ary Detection Theory*, Imperial College London.
- Marsh, H. (1963), 'Origin of the knudsen spectra', *The Journal of the Acoustical Society of America* **35**(3), 409–410.
- McAllester, D. A. (1999), Pac-bayesian model averaging, *in* 'Proceedings of the twelfth annual conference on Computational learning theory', pp. 164–170.
- McLoughlin, I., Zhang, H., Xie, Z., Song, Y. & Xiao, W. (2015), 'Robust sound event classification using deep neural networks', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **23**(3), 540–552.
- Meillier, C., Chatelain, F., Michel, O. & Ayasso, H. (2015), Error control for the detection of rare and weak signatures in massive data, *in* '2015 23rd European Signal Processing Conference (EUSIPCO)', IEEE, pp. 1974–1978.

- Mesaros, A., Heittola, T. & Virtanen, T. (2016), Tut database for acoustic scene classification and sound event detection, *in* ‘2016 24th European Signal Processing Conference (EUSIPCO)’, IEEE, pp. 1128–1132.
- Nagarajan, V. & Kolter, J. Z. (2019a), ‘Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience’, *arXiv preprint arXiv :1905.13344* .
- Nagarajan, V. & Kolter, J. Z. (2019b), ‘Generalization in deep networks : The role of distance from initialization’, *arXiv preprint arXiv :1901.01672* .
- Nanni, L., Ghidoni, S. & Brahnam, S. (2020), ‘Ensemble of convolutional neural networks for biimage classification’, *Applied Computing and Informatics* .
- Neyshabur, B., Bhojanapalli, S., McAllester, D. & Srebro, N. (2017), ‘Exploring generalization in deep learning’, *arXiv preprint arXiv :1706.08947* .
- Neyshabur, B., Tomioka, R. & Srebro, N. (2014), ‘In search of the real inductive bias : On the role of implicit regularization in deep learning’, *arXiv preprint arXiv :1412.6614* .
- Nicolls, F. & De Jager, G. (2007), ‘Optimality in detecting targets with unknown location’, *Signal processing* **87**(5), 841–852.
- Nielsen, R. O. (1991), *Sonar signal processing*, Artech House, Inc.
- Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. & Kavukcuoglu, K. (2016), ‘Wavenet : A generative model for raw audio’, *arXiv preprint arXiv :1609.03499* .
- Oyallon, E. (2017a), Analyzing and introducing structures in deep convolutional neural networks, PhD thesis, Paris Sciences et Lettres.
- Oyallon, E. (2017b), Building a regular decision boundary with deep networks, *in* ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition’, pp. 5106–5114.
- Oyallon, E., Belilovsky, E. & Zagoruyko, S. (2017), Scaling the scattering transform : Deep hybrid networks, *in* ‘Proceedings of the IEEE international conference on computer vision’, pp. 5618–5627.
- O’Shea, T. J., Corgan, J. & Clancy, T. C. (2016), Convolutional radio modulation recognition networks, *in* ‘International conference on engineering applications of neural networks’, Springer, pp. 213–226.
- Peyré, G. (2020), ‘Mathematical foundations of data sciences’, *Rn* **1**, 2.
- Piczak, K. J. (2015a), Environmental sound classification with convolutional neural networks, *in* ‘2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)’, IEEE, pp. 1–6.
- Piczak, K. J. (2015b), Esc : Dataset for environmental sound classification, *in* ‘Proceedings of the 23rd ACM international conference on Multimedia’, pp. 1015–1018.

- Poggio, T. & Liao, Q. (2018), 'Theory ii : Deep learning and optimization', *Bulletin of the Polish Academy of Sciences : Technical Sciences* pp. 775–787.
- Poor, H. V. (2013), *An introduction to signal detection and estimation*, Springer Science & Business Media.
- Porter, M. B. (2016), *BELLHOP3D User Guide*, Heat, Light, and Sound Research. <https://usermanual.wiki/Document/Bellhop3D20User20Guide202016725.1524880335>.
- Ribeiro, M. T., Singh, S. & Guestrin, C. (2016), " why should i trust you?" explaining the predictions of any classifier, *in* 'Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining', pp. 1135–1144.
- Roberts, D. A. (2021), 'Sgd implicitly regularizes generalization error', *arXiv preprint arXiv :2104.04874* .
- Sahidullah, M. & Saha, G. (2012), 'Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition', *Speech communication* **54**(4), 543–565.
- Sailor, H. B. (2018), Auditory representation learning, PhD thesis, Dhirubhai Ambani Institute of Information and Communication Technology.
- Sailor, H. B., Agrawal, D. M. & Patil, H. A. (2017), Unsupervised filterbank learning using convolutional restricted boltzmann machine for environmental sound classification., *in* 'Interspeech', Vol. 8, p. 9.
- Sang, J., Park, S. & Lee, J. (2018), Convolutional recurrent neural networks for urban sound classification using raw waveforms, *in* '2018 26th European Signal Processing Conference (EUSIPCO)', IEEE, pp. 2444–2448.
- Santos-Domínguez, D., Torres-Guijarro, S., Cardenal-López, A. & Pena-Gimenez, A. (2016), 'Shipsear : An underwater vessel noise database', *Applied Acoustics* **113**, 64–69.
- Schnupp, J., Nelken, I. & King, A. (2011), *Auditory neuroscience : Making sense of sound*, MIT press.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. & Batra, D. (2017), Grad-cam : Visual explanations from deep networks via gradient-based localization, *in* 'Proceedings of the IEEE international conference on computer vision', pp. 618–626.
- Shalev-Shwartz, S. & Ben-David, S. (2014), *Understanding machine learning : From theory to algorithms*, Cambridge university press.
- Shapley, L. (1953), A value for n-person games, *in* 'Contributions to the Theory of Games', pp. 307–317.
- Sherin, B. & Supriya, M. (2015a), Ga based selection and parameter optimization for an svm based underwater target classifier, *in* '2015 International Symposium on Ocean Electronics (SYMPOL)', IEEE, pp. 1–7.

- Sherin, B. & Supriya, M. (2015*b*), Selection and parameter optimization of svm kernel function for underwater target classification, in ‘2015 IEEE Underwater Technology (UT)’, IEEE, pp. 1–5.
- Shevtsova, I. (2010), An improvement of convergence rate estimates in the lyapunov theorem, in ‘Doklady Mathematics’, Vol. 82, Springer, pp. 862–864.
- Shi, M. & Xu, X. (2013), Underwater target recognition based on wavelet packet entropy and probabilistic neural network, in ‘2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013)’, IEEE, pp. 1–3.
- Shwartz-Ziv, R. & Tishby, N. (2017), ‘Opening the black box of deep neural networks via information’, *arXiv preprint arXiv :1703.00810* .
- Smith, S. L. & Le, Q. V. (2017), ‘Understanding generalization and stochastic gradient descent’, *arXiv preprint arXiv :1710.06451* .
- Soares-Filho, W., de Seixas, J. M. & Caloba, L. P. (2000), Averaging spectra to improve the classification of the noise radiated by ships using neural networks, in ‘Proceedings. Vol. 1. Sixth Brazilian Symposium on Neural Networks’, IEEE, pp. 156–161.
- Socheleau, F.-X., Ciblat, P. & Houcke, S. (2009), Ofdm system identification for cognitive radio based on pilot-induced cyclostationarity, in ‘2009 IEEE Wireless Communications and Networking Conference’, IEEE, pp. 1–6.
- Socheleau, F.-X., Passerieux, J.-M. & Laot, C. (2009), Characterisation of time-varying underwater acoustic communication channel with application to channel capacity, in ‘Underwater acoustic measurements’.
- Steinwart, I. & Christmann, A. (2008), *Support vector machines*, Springer Science & Business Media.
- Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. A. (2017), Inception-v4, inception-resnet and the impact of residual connections on learning, in ‘Thirty-first AAAI conference on artificial intelligence’.
- Takinacı, A. C. & Taralp, T. (2021), ‘Prediction and simulation of broadband propeller noise’, *Journal of Marine Science and Technology* **21**(5), 6.
- Telgarsky, M. (2015), ‘Representation benefits of deep feedforward networks’, *arXiv preprint arXiv :1509.08101* .
- THEYS, C., Ferrari, A., Le Dard, M. & Pitarque, T. (1991), Detection de trajets multiples de signaux sonar par modelisation evolutive, in ‘13ème Colloque sur le traitement du signal et des images, FRANCE, 1991’, GRETSI, Groupe d’Etudes du Traitement du Signal et des Images.
- Tian, S., Chen, D., Wang, H. & Liu, J. (2021), ‘Deep convolution stack for waveform in underwater acoustic target recognition’, *Scientific reports* **11**(1), 1–14.

- Tokozume, Y., Ushiku, Y. & Harada, T. (2017), ‘Learning from between-class examples for deep sound recognition’, *arXiv preprint arXiv :1711.10282* .
- Traverso, F., Vernazza, G. & Trucco, A. (2012), Simulation of non-white and non-gaussian underwater ambient noise, *in* ‘2012 Oceans-Yeosu’, IEEE, pp. 1–10.
- Urick, R. J. (1975), ‘Principles of underwater sound-2’.
- Vakanski, A. & Xian, M. (2021), Evaluation of complexity measures for deep learning generalization in medical image analysis, *in* ‘2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)’, IEEE, pp. 1–6.
- Valenti, M., Squartini, S., Diment, A., Parascandolo, G. & Virtanen, T. (2017), A convolutional neural network approach for acoustic scene classification, *in* ‘2017 International Joint Conference on Neural Networks (IJCNN)’, IEEE, pp. 1547–1554.
- Valle-Perez, G., Camargo, C. Q. & Louis, A. A. (2018), ‘Deep learning generalizes because the parameter-function map is biased towards simple functions’, *arXiv preprint arXiv :1805.08522* .
- Valle-Pérez, G. & Louis, A. A. (2020), ‘Generalization bounds for deep learning’, *arXiv preprint arXiv :2012.04115* .
- van den Ende, M., Ferrari, A., Sladen, A. & Richard, C. (2021), ‘Deep deconvolution for traffic analysis with distributed acoustic sensing data’.
- Vapnik, V. (1992), Principles of risk minimization for learning theory, *in* ‘Advances in neural information processing systems’, pp. 831–838.
- Vapnik, V. & Chervonenkis, A. (2015), *On the uniform convergence of relative frequencies of events to their probabilities*, Springer.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017), ‘Attention is all you need’, *Advances in neural information processing systems* **30**.
- Wang, W., Zhao, X. & Liu, D. (2021), ‘Design and optimization of 1d-cnn for spectrum recognition of underwater targets’, *Integrated Ferroelectrics* **218**(1), 164–179.
- Webster, R. J. (1994), ‘A random number generator for ocean noise statistics’, *IEEE journal of oceanic engineering* **19**(1), 134–137.
- Wen, Y., Luk, K., Gazeau, M., Zhang, G., Chan, H. & Ba, J. (2020), An empirical study of stochastic gradient descent with structured covariance noise, *in* ‘International Conference on Artificial Intelligence and Statistics’, PMLR, pp. 3621–3631.
- Wenz, G. M. (1962), ‘Acoustic ambient noise in the ocean : spectra and sources’, *The Journal of the Acoustical Society of America* **34**(12), 1936–1956.
- Wilson, A. C., Roelofs, R., Stern, M., Srebro, N. & Recht, B. (2017), ‘The marginal value of adaptive gradient methods in machine learning’, *arXiv preprint arXiv :1705.08292* .

- Wu, H., Song, Q. & Jin, G. (2018), Deep learning based framework for underwater acoustic signal recognition and classification, in 'Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence', pp. 385–388.
- Xie, N., Ras, G., van Gerven, M. & Doran, D. (2020), 'Explainable deep learning : A field guide for the uninitiated', *arXiv preprint arXiv :2004.14545* .
- Yang, S., Yang, C., Feng, D., Hao, X. & Wang, M. (2019), 'One-dimensional deep attention convolution network (odacn) for signals classification', *IEEE Access* **8**, 2804–2812.
- Yarotsky, D. (2018), 'Universal approximations of invariant maps by neural networks', *arXiv preprint arXiv :1804.10306* .
- Yue, H., Zhang, L., Wang, D., Wang, Y. & Lu, Z. (2017), 'The classification of underwater acoustic targets based on deep learning methods', *Adv. Intell. Syst. Res* **134**, 526–529.
- Zak, A. (2008), 'Ships classification basing on acoustic signatures', *WSEAS Transactions on Signal Processing* **4**(4), 137–149.
- Zarka, J., Thiry, L., Angles, T. & Mallat, S. (2019), 'Deep network classification by scattering and homotopy dictionary learning', *arXiv preprint arXiv :1910.03561* .
- Zellers, R., Bisk, Y., Farhadi, A. & Choi, Y. (2019), From recognition to cognition : Visual commonsense reasoning, in 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition', pp. 6720–6731.
- Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. (2017), 'Understanding deep learning requires rethinking generalization (2016)', *arXiv preprint arXiv :1611.03530* .
- Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. (2021), 'Understanding deep learning (still) requires rethinking generalization', *Communications of the ACM* **64**(3), 107–115.
- Zhang, H., McLoughlin, I. & Song, Y. (2015), Robust sound event recognition using convolutional neural networks, in '2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)', IEEE, pp. 559–563.
- Zhang, J. (2014), Bayesian Multiple Hypotheses Testing with Quadratic Criterion, PhD thesis, Beijing Academy of Agriculture and Forestry Sciences.
- Zhang, W., Li, C., Peng, G., Chen, Y. & Zhang, Z. (2018), 'A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load', *Mechanical Systems and Signal Processing* **100**, 439–453.
- Zhang, Y. & Yang, Q. (2021), 'A survey on multi-task learning', *IEEE Transactions on Knowledge and Data Engineering* .
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. & Torralba, A. (2016), Learning deep features for discriminative localization, in 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 2921–2929.

- Zhou, P. & Feng, J. (2018), Understanding generalization and optimization performance of deep cnns, *in* 'International Conference on Machine Learning', PMLR, pp. 5960–5969.
- Zhu, B., Wang, C., Liu, F., Lei, J., Huang, Z., Peng, Y. & Li, F. (2018), Learning environmental sounds with multi-scale convolutional neural network, *in* '2018 International Joint Conference on Neural Networks (IJCNN)', IEEE, pp. 1–8.

Appendices

Annexe A

Appendice A

1 Introduction

Cette annexe décrit des représentations des signaux abordées durant la thèse. La section 2 donne la définition mathématique de termes associés aux représentations des signaux. La section 3 décrit des méthodes de traitement du signal classiques comme la transformée de Fourier, le DEMON et le cepstre. Le fonctionnement du système auditif humain est également résumé pour faire un lien avec les méthodes de traitement du signal. La section 4 concerne des notions de bases des représentations temps-fréquence. La section 5 est un peu à part et présente une façon de simuler des bruits non-gaussiens.

2 Définitions de termes associés aux représentations

Une alternative à l'apprentissage d'opérateurs contractifs est la stratégie de linéarisation proposée par Mallat (2016). Cette méthode consiste à linéariser les variations de la fonction idéale f avec un premier changement de variable pour obtenir une représentation $\Phi(\mathbf{x})$ de dimension K' , potentiellement beaucoup plus grande que la dimension K du signal \mathbf{x} . On peut alors optimiser une projection linéaire de faible dimension dans les directions où f est constante.

Définition : Linéarisation

Soit $\Phi(\mathbf{x}) = \{\phi_k(\mathbf{x})\}_{k \leq K'}$ une représentation du signal $\mathbf{x} \in \mathbb{R}^K$. On dit alors que Φ sépare f linéairement si $f(x)$ est bien approximée par la projection unidimensionnelle :

$$\tilde{f}(\mathbf{x}) = \langle \Phi(\mathbf{x}), \omega \rangle = \sum_{k=1}^{K'} \omega_k \phi_k(\mathbf{x}). \quad (\text{A.1})$$

Le vecteur de régression ω est optimisé en minimisant une fonction de perte sur les données d'entraînement, qui doit entre autre être régularisée si $K' > N$ (avec N le nombre d'exemples d'entraînement).

Une autre propriété souhaitable pour une représentation est la séparabilité. Selon la définition donnée par Mallat (2016), c'est une réduction de la dimension de \mathbf{x} en calculant un vecteur de basse dimension $\Phi(\mathbf{x})$ tel que si $f(\mathbf{x}) \neq f(\mathbf{x}')$, alors $\Phi(\mathbf{x}) \neq \Phi(\mathbf{x}')$. On dit alors que Φ sépare f . Dans un problème de classification, $f(\mathbf{x}) \neq f(\mathbf{x}')$ implique que \mathbf{x} et \mathbf{x}' ne sont pas dans la même classe. La condition de séparation de Lipschitz devient ainsi une condition de marge spécifiant une distance minimale entre les classes :

$$\exists \epsilon > 0, \forall (\mathbf{x}, \mathbf{x}') \in \Omega^2, f(\mathbf{x}) \neq f(\mathbf{x}') \implies \|\Phi(\mathbf{x}) - \Phi(\mathbf{x}')\| \geq \epsilon. \quad (\text{A.2})$$

On peut alors essayer de trouver une projection linéaire de \mathbf{x} dans un espace V de dimension moins élevée K' , qui séparerait f . Cela requiert que $f(\mathbf{x}) = f(\mathbf{x} + \mathbf{z})$ pour tout $\mathbf{z} \in V^\perp$, où V^\perp est le complément orthogonal à V dans \mathbb{R}^K , de dimension $K - K'$, bien que dans la majorité des cas, la dimension finale K' ne peut pas être beaucoup plus petite que K .

3 Traitement du signal

3.1 Transformée de Fourier

La transformée de Fourier d'une fonction $f(t)$ réelle ou complexe est définie dans L^2 par :

$$\hat{f}(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt. \quad (\text{A.3})$$

La transformée de Fourier inverse va représenter f comme une somme de sinusoidales :

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega)e^{i\omega t} d\omega. \quad (\text{A.4})$$

La transformée de Fourier discrète d'un signal discrétisé $f(k)$, de K échantillons réels ou complexes, sera représentée par ses périodes tel que :

$$\hat{f}(n) = \mathcal{F}\{f(k)\} = \frac{1}{K} \sum_{k=0}^{K-1} f(k) \cdot e^{-ik \frac{n}{K}}. \quad (\text{A.5})$$

Le signal analytique $f_a(t)$ d'un signal $f(t)$ est défini tel que :

$$f_a(t) = f(t) + i \cdot \mathcal{H}\{f\}(t), \quad (\text{A.6})$$

où $f(t) = \text{Re}\{f_a(t)\}$ et $\mathcal{H}\{\cdot\}$ est la transformation de Hilbert :

$$\mathcal{H}\{f\}(t) = \mathcal{V}_c \left\{ \int_{-\infty}^{\infty} s(\tau)h(t-\tau)d\tau \right\} = \frac{1}{\pi} \mathcal{V}_c \left\{ \int_{-\infty}^{\infty} \frac{s(\tau)}{t-\tau} d\tau \right\}, \quad (\text{A.7})$$

avec $h(t) = \frac{1}{\pi t}$ et $\mathcal{V}_c\{\cdot\}$ la valeur principale de Cauchy tel que :

$$VP \left\{ \int_{-\infty}^{\infty} s(\tau)h(t-\tau)d\tau \right\} = \lim_{\epsilon \rightarrow 0} \left\{ \int_{-\infty}^{t-\epsilon} s(\tau)h(t-\tau) d\tau + \int_{t+\epsilon}^{+\infty} s(\tau)h(t-\tau) d\tau \right\}. \quad (\text{A.8})$$

En se référant au livre de Mallat (1999), on dit qu'un signal L^2 est analytique si sa transformée de Fourier est nulle pour les fréquences négatives. La partie analytique $f_a(t)$ d'un signal réel f peut alors être exprimée par sa transformée de Fourier :

$$\widehat{f}_a(\omega) = \begin{cases} 2\widehat{f}(\omega) & \text{pour } \omega \geq 0 \\ 0 & \text{sinon.} \end{cases} \quad (\text{A.9})$$

Ainsi, le signal analytique f_a peut être obtenu avec la transformée de Fourier inverse de \widehat{f} , à laquelle on aurait fixé toutes les composantes fréquentielles négatives à zéro. Cette opération n'engendre aucune perte d'information du fait que la transformée de Fourier d'un signal réel respecte la propriété de symétrie hermitienne $\widehat{f}^*(\omega) = \widehat{f}(-\omega)$. En temps discret cette opération revient à annuler toutes les valeurs $\widehat{f}(n)$ pour $n > \frac{K}{2}$ et à prendre $\widehat{f}_a(n) = \widehat{f}(n)$ pour $n = 0$ ainsi que $n = \frac{K}{2}$. Pour un signal analytique $f_a(t)$ on peut définir une amplitude instantanée $a(t)$ ainsi qu'une phase instantanée complexe $\phi(t)$ tel que :

$$f_a(t) = a(t) \cdot e^{i\phi(t)}. \quad (\text{A.10})$$

La fréquence instantanée d'un signal analytique est alors la dérivée positive de sa phase :

$$\omega(t) = \phi'(t) \geq 0. \quad (\text{A.11})$$

3.2 Représentation DEMON

L'analyse DEMON est une analyse bande étroite utilisée principalement dans l'analyse de bruits de cavitation, où elle permet d'obtenir des informations nécessaires aux analystes pour aboutir à la classification et l'identification d'un navire. Son traitement nécessite une quantité conséquente d'échantillons provenant d'une même source, soit environ entre quelques secondes et quelques dizaines de secondes. En pratique, l'analyse par DEMON doit permettre de séparer la composante du bruit de cavitation du reste du bruit produit par un système de propulsion. En conséquence, il permet alors d'extraire les informations de la signature du navire source, soit une estimation du nombre d'hélices, leur vitesse de rotation ainsi que leur nombre de pales. L'analyse par DEMON est généralement effectuée selon les étapes suivantes :

- **Pointage de la source** : à travers un traitement d'antenne en gisement et en site, la source cible, que l'on souhaite analyser, est isolée des sources provenant d'autres directions, avec un gain lié au traitement d'antenne utilisé.
- **Filtrage passe bande** : permet de restreindre la bande de fréquences analysée à une bande quasiment dépourvue de bruit ambiant coloré, où le bruit de cavitation reste important. L'analyse DEMON étant déduite d'un modèle basé sur des hypothèses de bruit ambiant et de cavitation blanc, il est préférable, pour justifier son utilisation, que la bande de fréquences conservée présente ces caractéristiques. La fréquence centrale du bruit de cavitation dépend directement de la vitesse de rotation des pales et de la profondeur, ce qui rend cette tâche plus difficile du fait de la forte corrélation des échantillons dans cette zone du spectre. En général, une analyse DEMON ignore souvent la fréquence centrale de cavitation pour filtrer directement entre 1 et 5.5 kHz. Cela permet de supprimer le bruit ambiant basse fréquence, tout en gardant un bruit de cavitation relativement blanc. Des bruits parasites peuvent cependant exister dans

cette bande de fréquence, c'est pourquoi il est nécessaire pour un utilisateur d'ajuster les paramètres de filtrage en situation réelle pour tenter de distinguer un NPM au sein d'un environnement bruyant.

- **Élever au carré** : l'enveloppe réelle moyenne du signal filtré peut être calculée au moyen de la valeur efficace ou racine de la moyenne du carré de la transformée de Hilbert du signal, bien que plusieurs autres méthodes soient valides. La taille de la fenêtre employée à cet effet doit comporter un nombre d'échantillons suffisamment conséquents pour les besoins de l'analyse.
- **Transformée de Fourier à court terme** : Les propriétés de l'enveloppe du signal peuvent ensuite être présentées sous la forme d'une analyse spectrale ou d'un spectrogramme qui sont respectivement le DEMON et le DEMONgram affichés en figure A.1.
- **Traitement du signal par normalisation** : Une dernière étape optionnelle est de soustraire l'analyse spectrale DEMON obtenue, par une analyse spectrale classique (simple FFT), afin de réduire le bruit ambiant et de mieux distinguer les pics sur la représentation du DEMON.

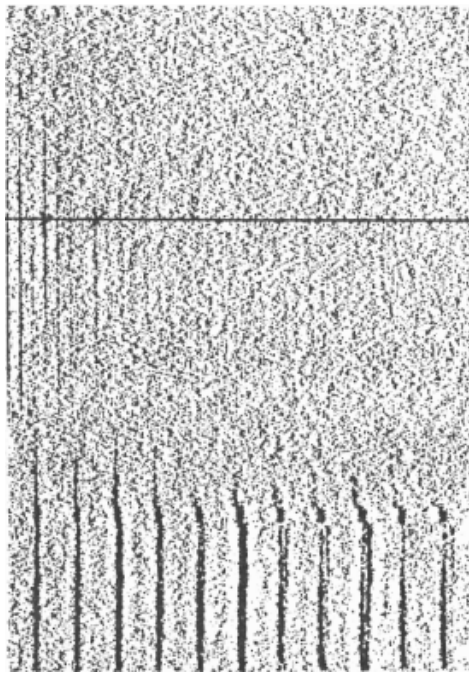


Figure A.1: Exemple de DEMONgram.

L'origine de la représentation DEMON est souvent attribuée à [Lourens \(1997\)](#) ou [Nielsen \(1991\)](#) dans le cadre de la détection de bruits modulés en amplitude, du fait que leurs preuves, liées à des modèles de signaux différents, aboutissent à des expressions similaires. [Lourens \(1997\)](#) proposent ainsi la représentation DEMON comme une approche graphique en l'absence d'expression d'estimateur optimal de la fréquence de rotation ω_k d'une hélice. Dans leur approche, la modulation d'un bruit blanc Gaussien par une signature d'hélice $m_{\omega_k}(t)$ est

alors modélisée par une série de Fourier de paramètres inconnus :

$$m_{\omega_k}(t) = \sum_{n=0}^{\infty} F_n \cos(n\omega_k(t)). \quad (\text{A.12})$$

3.3 Système auditif humain

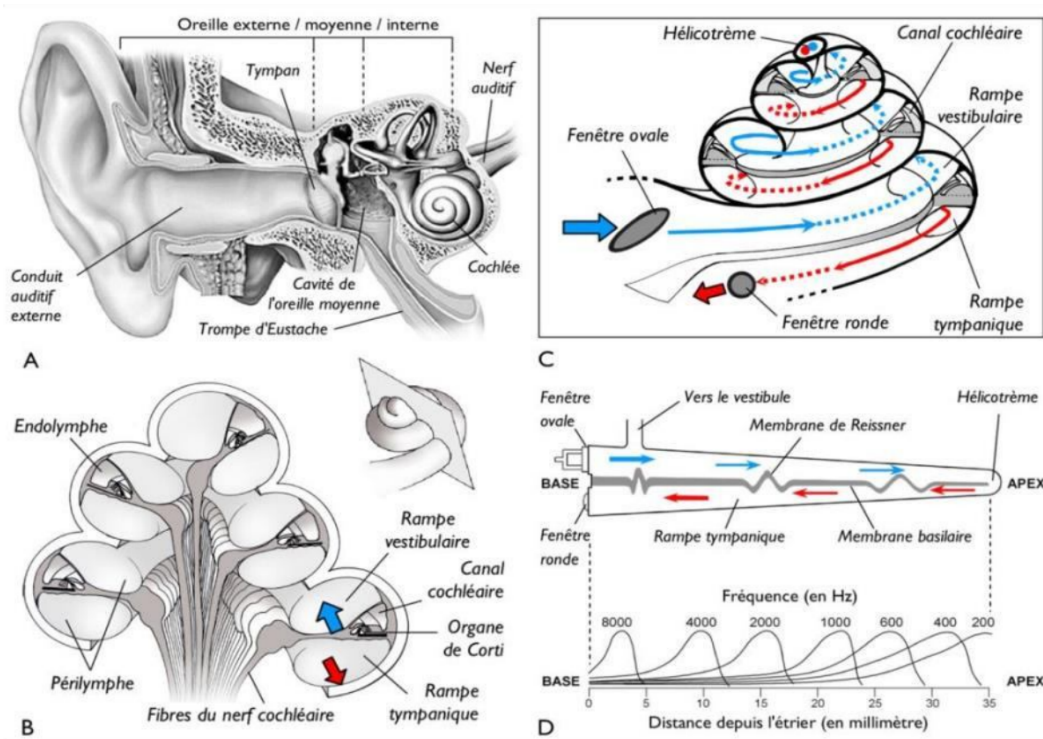


Figure A.2: *Partie A : Coupe longitudinale de l'appareil auditif humain. Partie B : Coupe de la cochlée. Partie C : Propagation des vibrations dans la cochlée. Partie D : Schéma de la cochlée déroulée. Tiré de l'article de Girard & Roi (2013).*

Dans leur article, Girard & Roi (2013) décrivent dans la partie C de la figure A.2, le chemin des vibrations au niveau de l'oreille interne et la transmission de l'information portée par ces vibrations dans la partie D de la figure A.2. Les vibrations engendrées au niveau de la fenêtre ovale par le marteau, l'enclume et l'étrier de l'oreille moyenne, suivent ainsi la rampe vestibulaire du canal spiral de la cochlée jusqu'à son apex, en faisant onduler la membrane basilaire à travers le trajet en bleu. Elles redescendent ensuite en suivant le trajet en rouge et ressortent par la fenêtre ronde. Les différentes fréquences des ondes de pression dans la rampe tympanique, font ensuite onduler une certaine partie de la membrane basilaire plus que les autres, stimulant ainsi l'organe de Corti qui repose sur elle et transmettant alors au cortex auditif, via le nerf auditif, la présence d'une fréquence précise dans les vibrations reçues. La cochlée agit ainsi comme un analyseur de fréquence et on remarquera dans la partie D de la figure A.2, que la distribution des fréquences d'analyse de la cochlée approche une échelle logarithmique. D'un point de vue mathématique, si on se réfère au livre de Schnupp et al.

(2011), elle permet de décomposer tout son complexe en différentes sous-bandes fréquentielles. Comme dans (Sailor 2018, Section 2.6), un ensemble de filtres linéaires peut ainsi fournir une approximation au premier ordre de la réponse mécanique de la membrane basilaire, notamment une banque de filtres gammatone.

3.4 Cepstre

Le cepstre d'un signal temporel $s(t)$ est la transformée de Fourier inverse du logarithme du module de son spectre (phases ignorées).

$$\mathcal{C}\{s(t)\} = \mathcal{F}^{-1}\{\log(|\mathcal{F}\{s(t)\}|)\}. \quad (\text{A.13})$$

Il existe un cepstre complexe et un cepstre réel. Le réel, défini ci-dessus, n'utilise que l'amplitude du spectre du signal et ne permet donc pas de reconstruire parfaitement le signal de départ. Le cepstre réel est parfois défini avec la puissance spectrale en élevant le module au carré, les deux définitions étant équivalentes à un facteur près, étant donné l'action du logarithme. Élevé au carré, le cepstre peut être appelé cepstre de puissance :

$$\mathcal{C}_P\{s(t)\} = |\mathcal{F}^{-1}\{\log(|\mathcal{F}\{s(t)\}|^2)\}|^2. \quad (\text{A.14})$$

On peut également définir un cepstre complexe avec le logarithme complexe, permettant ainsi de conserver l'information de la phase et de reconstruire le signal d'origine. Une propriété importante du domaine cepstrale est que la convolution de deux signaux devient une addition :

$$\mathcal{C}\{s_1(t) * s_2(t)\} = \mathcal{C}\{s_1(t)\} + \mathcal{C}\{s_2(t)\} \quad (\text{A.15})$$

C'est une représentation utilisée afin de convertir des signaux combinés par convolution, comme par exemple une source et un filtre, en sommes de leurs cepstres. En particulier, le cepstre de puissance est souvent utilisé dans le domaine du traitement de la parole comme un vecteur de variables caractéristiques de la voix, un signal caractérisé par ses peignes de raies. Pour cette application, le cepstre est généralement d'abord transformé à l'aide d'une échelle Mel (logarithmique). Le résultat est appelé cepstre de fréquence Mel ou MFC et les coefficients sont les MFCCs. Le cepstre est utile dans l'identification de voix car l'excitation périodique des cordes vocales à basse fréquence, ainsi que le filtrage par formants du canal vocal, sont additifs et se situent dans différentes régions du domaine cepstrale. Le domaine cepstrale est cependant une représentation peu robuste aux bruits à faible SNR et son utilisation dans le domaine de l'ASM est en conséquence assez limitée.

Le cepstre de fréquence Mel est une forme de représentation du spectre de puissance à court terme d'un signal acoustique. Il est basé sur une transformation en cosinus linéaire du logarithme d'un spectre de puissance sur une échelle de fréquence Mel, non linéaire. Les coefficients cepstraux de fréquence de Mel, les MFCCs, sont les coefficients qui constituent collectivement un MFC sous forme de matrice. La différence entre le cepstre et le cepstre de fréquence Mel réside dans le fait que, dans le MFC, les bandes de fréquences sont équidistantes sur l'échelle Mel, ce qui se rapproche davantage de la réponse du système auditif humain que des bandes de fréquences espacées linéairement comme dans le cepstre classique. En se référant à l'article de Sahidullah & Saha (2012), les MFCCs peuvent être obtenus en effectuant tout d'abord une transformée de Fourier du signal, puis en représentant le spectre de puissance en échelle Mel,

à l'aide de fenêtres triangulaires superposées. Il est ensuite nécessaire d'appliquer la fonction logarithme aux puissances de chaque fréquence Mel et d'effectuer une transformation en cosinus discrète sur chaque vecteur de logarithme de puissance comme s'il s'agissait de signaux. Les MFCC sont alors les amplitudes du spectre résultant. À noter que l'ajout d'éléments dynamiques; tels que les coefficients delta et delta delta utilisés par [Furui \(1986\)](#); est une pratique assez courante pour l'extraction de caractéristiques d'un signal acoustique en vue d'une classification.

4 Représentation temps-fréquence

4.1 Propriétés des représentations dans le domaine temps-fréquence

Une transformation linéaire dans le domaine temps-fréquence met en corrélation un signal avec un dictionnaire de formes d'ondes localisées en temps et en fréquence. Ces formes d'ondes sont communément dénommées atomes temps-fréquence. Dans un tel dictionnaire d'atomes

$$\mathcal{B} = \{\phi_\gamma\}_{\gamma \in \Gamma}, \quad (\text{A.16})$$

où γ peut être un paramètre multi-index, on suppose que les fonctions ϕ_γ sont définies dans $L^2(\mathbb{R})$ et $\|\phi_\gamma\| = 1$. La transformée temps-fréquence linéaire d'une fonction $f \in L^2(\mathbb{R})$ associée à un tel atome est alors définie par :

$$\Phi f(\gamma) = \langle f, \phi_\gamma \rangle = \int_{-\infty}^{\infty} f(t) \cdot \phi_\gamma^*(t) dt \quad (\text{A.17})$$

et d'après la formule de Parseval,

$$\Phi f(\gamma) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{f}(\omega) \widehat{\phi}_\gamma^*(\omega) d\omega. \quad (\text{A.18})$$

En particulier, si ϕ_γ est proche de zéro quand t est en dehors d'un voisinage d'abscisse u , $\langle f, \phi_\gamma \rangle$ dépend uniquement des valeurs de f dans ce voisinage. De façon similaire, dans le domaine fréquentiel, si $\widehat{\phi}_\gamma(\omega)$ est négligeable pour ω en dehors du voisinage de l'ordonnée ξ , alors $\langle f, \phi_\gamma \rangle$ permet de caractériser les propriétés de \widehat{f} dans le voisinage de ξ . La transformée de Fourier, permet en particulier de représenter un signal à partir d'une base de sinusoides. Au sein de cette base, les sinusoides sont très bien localisées en fréquence, mais pas aussi bien dans le domaine temporel, du fait que leur support soit infini, en conséquence de leur périodicité, comme le fait remarquer [Mallat \(1999\)](#). Si on veut représenter les propriétés fréquentielles d'un signal localement en temps, il convient alors de les analyser par des signaux localisés en temps et en fréquence. Dans l'idéal, on utilisera une base de fonctions à support compact en temps et fréquence. Malheureusement les représentations dans le domaine temps-fréquence sont limitées par le théorème d'incertitude d'Heisenberg, ainsi qu'un résultat concernant la compacité des supports dans le domaine temps-fréquence. En effet, les informations apportées par le produit $\langle f, \phi_\gamma \rangle$ peuvent être représentées dans un plan du domaine temps-fréquence par un rectangle appelé boîte de Heisenberg qui dépend de ϕ_γ pour sa position et ses dimensions.

Théorème d'incertitude de Heisenberg

Théorème 11 Soit f une fonction de L^2 , alors on peut définir l'écart-type de sa transformée $\Phi f(\gamma)$ en temps σ_t ainsi que l'écart-type en fréquence σ_ω en se basant sur l'atome $\phi(\gamma)$. On a alors $\sigma_t \cdot \sigma_\omega \leq 1/2$, ou bien $\sigma_t^2 \cdot \sigma_\omega^2 \leq 1/4$, pour sa forme de boîte de Heisenberg. Puisque $\|\phi_\gamma\|^2 = 1$, on peut considérer que $|\phi_\gamma(t)|^2$ est une distribution de probabilité. On peut alors inférer son centre :

$$\mu_t(\gamma) = \int_{-\infty}^{\infty} t \cdot |\phi_\gamma(t)|^2 dt, \quad (\text{A.19})$$

ainsi que sa dispersion autour de $\mu_t(\gamma)$:

$$\sigma_t^2(\gamma) = \int_{-\infty}^{\infty} (t - \mu_t(\gamma))^2 |\phi_\gamma(t)|^2 dt. \quad (\text{A.20})$$

Il est alors possible de faire de même avec les fréquences puisque $\|\widehat{\phi}_\gamma\|^2 = 2\pi\|\phi_\gamma\|^2$ d'après la formule de Plancherel :

$$\mu_\omega(\gamma) = \int_{-\infty}^{\infty} \omega \cdot |\widehat{\phi}_\gamma(\omega)|^2 d\omega \quad (\text{A.21})$$

$$\sigma_\omega^2(\gamma) = \int_{-\infty}^{\infty} (\omega - \mu_\omega(\gamma))^2 |\widehat{\phi}_\gamma(\omega)|^2 d\omega. \quad (\text{A.22})$$

L'inégalité du théorème d'incertitude est atteinte pour une fonction f qui serait un chirp de Gabor. Pour une sinusoïdale pure, σ_ω est nul et σ_t est infini. Dans une représentation temps-fréquence, des rectangles d'aires inférieures à $\frac{1}{4}$ ne peuvent nullement correspondre à un atome temps-fréquence. Ce théorème implique donc qu'une précision en fréquence impose des contraintes sur notre précision en temps, et inversement. La localisation en temps-fréquence ne peut donc être atteinte qu'en écart-type et sa représentation sous la forme de boîte de Heisenberg est censée montrer l'incertitude de localisation d'un point (t_0, ω_0) . Si f est non nul

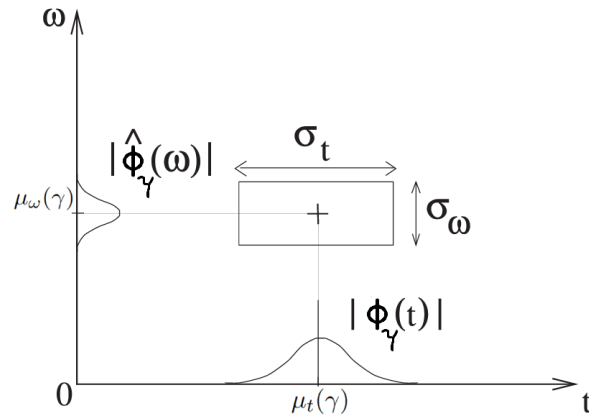


Figure A.3: Boîte de Heisenberg d'un atome ϕ_γ . Tiré du livre de Mallat (1999).

et à support compact, alors sa transformée de Fourier ne peut s'annuler sur tout un intervalle. De même, si sa transformée de Fourier est à support compact, $f(t)$ ne peut s'annuler sur tout un intervalle de temps. d'après la propriété de compacité des supports, il est alors impossible d'avoir une fonction de L^2 qui soit à la fois à support compact en temps et en fréquence.

4.2 Transformée de Fourier fenêtrée/à court terme

Une transformée de Fourier fenêtrée consiste à remplacer les fonctions provenant d'une base de sinusoïdales de support infini par des produits de sinusoïdales et de fenêtres localisées en temps.

$$\mathcal{S}_{\mathcal{F}}[f(t)](u, \xi) = \langle f, g_{u, \xi} \rangle = \int_{-\infty}^{\infty} f(t)g_{u, \xi}(t)dt, \quad (\text{A.23})$$

avec $g_{u, \xi}$ un atome constitué d'une fenêtre temporelle symétrique g , de centre u et d'énergie finie, ainsi que d'une sinusoïde tel que :

$$g_{u, \xi}(t) = g(t - u)e^{-i\xi t}, \quad (\text{A.24})$$

avec ξ la localisation en fréquence. Les écarts-types en temps σ_t et en fréquence σ_ω sont constants pour tout couple (u, ξ) . On peut ainsi obtenir une famille de vecteurs d'analyse en translatant la fenêtre de taille unique g par u ou en modulant la fenêtre par la fréquence ξ . La transformée de Fourier fenêtrée a donc une résolution temps-fréquence fixe, cependant, elle peut être modifiée par un changement d'échelle sur la fenêtre g . C'est une représentation complète, stable et redondante du signal et donc inversible. La redondance se traduit par l'existence d'un noyau reproduisant comme le précise [Mallat \(1999\)](#).

On définit le spectrogramme d'un signal $f(t)$ comme étant le carré du module de sa transformée de Fourier fenêtrée, tel que :

$$\mathcal{P}_{\mathcal{F}}[f(t)](u, \xi) = |\mathcal{S}_{\mathcal{F}}[f(t)](u, \xi)|^2 = \left| \int_{-\infty}^{\infty} f(t)g(t - u)e^{-i\xi t} dt \right|^2. \quad (\text{A.25})$$

L'implémentation de la transformée de Fourier fenêtrée discrète d'un signal $f(k)$, disposant de K échantillons, se fait au moyen de K transformées de Fourier à court terme, en $\mathcal{O}(K^2 \cdot \log_2(K))$ opérations :

$$\mathcal{S}_{\mathcal{F}}[f(k)](m, l) = \langle f, g_{m, l} \rangle = \sum_{k=1}^{K-1} f(k)g(k - m)e^{-i2\pi l \frac{k}{K}}. \quad (\text{A.26})$$

La transformée de Fourier fenêtrée est caractérisée par l'atome la définissant, ainsi que sa fenêtre g , dont on cherche au maximum à concentrer l'énergie autour de 0. g doit avoir un support compact, tandis que \hat{g} est à support infini. g est par définition symétrique et diminue vers zéro en oscillant. On peut généralement caractériser la répartition d'énergie d'une fenêtre grâce à trois paramètres :

- La largeur quadratique moyenne de base $\Delta\omega$ est définie telle que :

$$\frac{|\hat{g}(\Delta\omega/2)|^2}{|\hat{g}(0)|^2} = \frac{1}{2}. \quad (\text{A.27})$$

Un $\Delta\omega$ petit signifiera que l'énergie est fortement concentrée autour du centre de la fenêtre.

- L'amplitude maximale A du premier lobe latéral, mesurée en décibels :

$$A = 10 \log_{10} \left(\frac{|\hat{g}(\omega_0)|^2}{|\hat{g}(0)|^2} \right). \quad (\text{A.28})$$

- Le taux de décroissance polynomial de \hat{g} à l'infini, p , permet d'obtenir des informations sur le comportement de \hat{g} après le premier lobe secondaire.

$$|\hat{g}(\omega)| = \mathcal{O}(\omega^{-p-1}). \quad (\text{A.29})$$

Fonction	$g(t)$	$\Delta\omega$	A	p
Rectangle	1	0.89	-13 dB	0
Hamming	$0.54 + 0.46 \cos(2\pi t)$	1.36	-43 dB	0
Gaussienne	e^{-18t^2}	1.55	-55 dB	0
Hanning	$\cos^2(\pi t)$	1.44	-32 dB	2
Blackman	$0.42 + 0.5 \cos(2\pi t) + 0.08 \cos(4\pi t)$	1.68	-58 dB	2

Tableau A.1: Paramètres caractéristiques de fenêtres de Fourier usuelles.

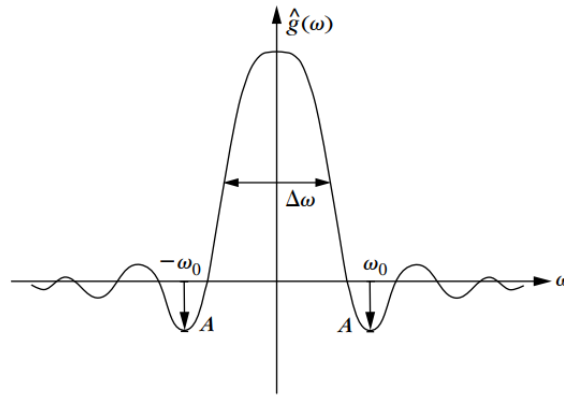


Figure A.4: Dispersion de l'énergie d'une fenêtre \hat{g} . Tiré du livre de [Mallat \(1999\)](#).

4.3 Transformée en ondelettes rapide

Si on se réfère au livre de [Mallat \(1999\)](#), une transformée en ondelettes revient à remplacer l'atome de la transformée de Fourier par une famille de translations et dilatations d'une même forme d'ondelette. Les atomes créés par translations et dilatations de l'ondelette de base permettent d'analyser des structures de signaux de différentes tailles grâce à la variabilité de leur support temporel :

$$\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right). \quad (\text{A.30})$$

Une ondelette ψ est une fonction de L^2 de moyenne nulle et normalisée telle que $\|\psi\| = 1$. L'ondelette de base $\psi(t)$ est centrée au voisinage de $t = 0$. Tous les atomes $\psi_{u,s}$ du dictionnaire des translations et dilatations de ψ sont également normalisés tels que $\|\psi_{u,s}\| = 1$ et centré au voisinage de u . Qui plus est, si la fréquence centrale de ψ est ξ , alors celui de $\psi_{u,s}$ est ξ/s . La transformée en ondelettes d'une fonction f de L^2 est définie par :

$$\mathcal{W}[f(t)](u, s) = \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{\infty} \frac{1}{\sqrt{s}} f(t) \psi^* \left(\frac{t-u}{s} \right) dt. \quad (\text{A.31})$$

L'écart-type en temps sera proportionnel à s tandis que l'écart-type en fréquence est inversement proportionnel à s . On peut par ailleurs réécrire la transformée en ondelette sous forme de produit de convolution :

$$\mathcal{W}[f(t)](u, s) = f * \bar{\psi}_s(u), \quad (\text{A.32})$$

avec

$$\bar{\psi}_s(t) = \frac{1}{\sqrt{s}} \psi^* \left(\frac{-t}{s} \right). \quad (\text{A.33})$$

La transformée de Fourier de $\bar{\psi}_s(t)$ est alors $\widehat{\bar{\psi}}_s(\omega) = \sqrt{s} \widehat{\psi}^*(s\omega)$. Aussi, puisque

$$\widehat{\psi}(0) = \int_{-\infty}^{\infty} \psi(t) dt = 0, \quad (\text{A.34})$$

$\widehat{\psi}$ est une fonction de transfert d'un filtre passe bande. Comme pour la transformée de Fourier fenêtrée, on a une représentation complète, stable et redondante du signal à la condition que :

$$C_\psi = \int_0^\infty \frac{|\widehat{\phi}(\omega)|^2}{\omega} d\omega < \infty. \quad (\text{A.35})$$

On définit le Scalogramme d'un signal $f(t)$ comme étant le carré du module de sa transformée en ondelettes, tel que :

$$\mathcal{P}_\mathcal{W}[f(t)](u, \xi) = |\mathcal{W}[f(t)](u, s)|^2 = |\mathcal{W}[f(t)](u, \frac{\eta}{\xi})|^2, \quad (\text{A.36})$$

où, en notant η la fréquence centrale de l'ondelette élémentaire, le centre fréquentiel d'une ondelette dilatée est $\xi = \eta/s$ et le scalogramme normalisé est défini par $(\xi/\eta) \cdot \mathcal{P}_\mathcal{W}[f(t)](u, \xi)$. La transformée en ondelettes peut se calculer par une transformée en ondelettes rapide. On utilise alors une suite d'échelles discrètes a^j , avec $a = 2^{\frac{1}{q}}$ et q un entier représentant le nombre d'ondelette dans l'octave. On suppose que le support de l'ondelette ψ est $[-\frac{P}{2}, \frac{P}{2}]$, et pour un signal de taille K et $1 \leq a^j \leq \frac{K}{P}$, on définira une ondelette discrète en échantillonnant les fréquences à l'échelle a^j telle que :

$$\psi_j[k] = \frac{1}{\sqrt{a^j}} \psi \left(\frac{k}{a^j} \right). \quad (\text{A.37})$$

La transformée en ondelettes discrètes d'un signal $f(k)$, sur une période K sera alors :

$$\mathcal{W}[f(k)](m, a^j) = \sum_{k=0}^{K-1} f(k) \psi_j^*(k-m). \quad (\text{A.38})$$

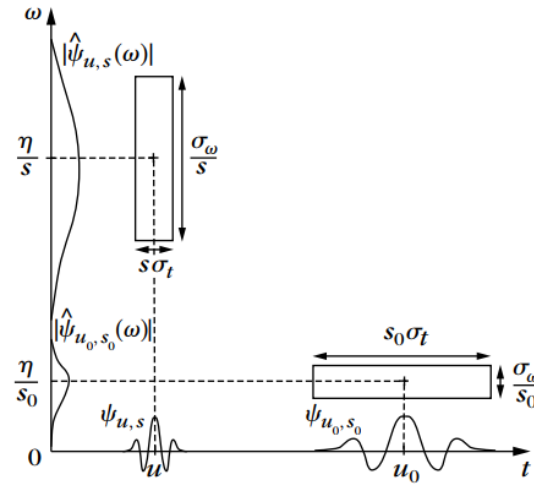


Figure A.5: Boîte de Heisenberg de deux ondelettes. Des échelles plus petites diminuent l'étalement temporel mais augmentent le support fréquentiel, qui est alors décalé vers des fréquences plus élevées. Tiré du livre de [Mallat \(1999\)](#).

Celle-ci effectue une transformée discrète par des convolutions circulaires, elles-mêmes calculées par FFT (accélération des calculs avec des signaux 2M-périodiques), pour un total de $\mathcal{O}(K \cdot \log_2(K))$ opérations. Ces calculs peuvent notamment être accélérés en utilisant des ondelettes dyadiques qui s'implémentent par bancs de filtres.

Remarque

[Mallat \(1999\)](#) précise également que, en plus de sa moyenne nulle et de son énergie finie, on peut caractériser une ondelette par son nombre de moments nuls :

$$\int_{-\infty}^{\infty} t^m \psi(t) dt = 0, \quad (\text{A.39})$$

pour $0 \leq m < M$. La nullité des moments d'une ondelette permet entre autre d'analyser la régularité locale d'un signal.

5 Modélisation d'un bruit non gaussien

D'anciennes études sur des mesures du bruit océanique, effectuées par [Dyer \(1973\)](#) et [Bouvet & Schwartz \(1988\)](#), ont montré que les composantes de surface et de trafic pouvaient être modélisées par des bruits non gaussiens. Aussi, nous avons utilisé la technique employée par [Traverso et al. \(2012\)](#), s'inspirant des travaux de [Webster \(1994\)](#) pour générer des signaux de bruit ambiants $e(t)$ non gaussiens. L'algorithme employé génère des séquences de nombres aléatoires auxquelles on peut attribuer une enveloppe spectrale souhaitée ainsi qu'une valeur à priori de kurtosis permettant la simulation du processus stochastique représentatif d'un bruit sous-marin réel.

La valeur de kurtosis d'une variable aléatoire est un indicateur permettant de déterminer dans quelle mesure sa fonction de densité de probabilité a une forme gaussienne. Un processus gaussien aura un kurtosis égal à 3, et tout décalage par rapport à cette valeur correspondra à une fonction de densité de probabilité de forme non gaussienne. Pour une variable aléatoire réelle X d'espérance μ et d'écart-type σ , on définit son kurtosis non normalisé comme le moment d'ordre quatre de la variable centrée réduite :

$$\beta_2 = \mathbb{E} \left[\left(\frac{X - \mu}{\sigma} \right)^4 \right] \text{ ou} \quad (\text{A.40})$$

$$\beta_2 = \frac{\mu_4}{\sigma^4} \text{ si on connaît } \mu, \quad (\text{A.41})$$

avec μ_4 le moment centré d'ordre 4. Le kurtosis non normalisé étant défini en fonction de moments centrés, il est difficile à manipuler lorsqu'on tente de calculer celui de la somme de variables indépendantes. On définit ainsi le kurtosis normalisé en fonction de cumulants κ_i :

$$\gamma_2 = \frac{\kappa_4}{\kappa_2^2} \quad (\text{A.42})$$

Aussi, puisque $\mu_2 = \kappa_2$ et $\mu_4 = \kappa_4 + 3.\kappa_2^2$, on a :

$$\gamma_2 = \frac{\mu_4 - 3\mu_2^2}{\mu_2^2} = \beta_2 - 3 \quad (\text{A.43})$$

Pour une composante donnée, la génération d'un signal caractéristique d'un bruit ambiant, commence par le biais de l'échantillonnage aléatoire avec loi uniforme d'une forme d'onde d'enveloppe logarithmique spécifique à la valeur de kurtosis voulue.

$$x_m = \left[\log\left(\frac{1}{t_m}\right) \right]^n \cdot \sin(\omega t_m + \phi), \quad (\text{A.44})$$

avec $t_m \sim \mathcal{U}(0, 1)$ et pour $m = 1, 2, 3, \dots, fs$. La génération du bruit vise ainsi à produire une séquence de valeurs aléatoires possédant une valeur de kurtosis supérieure ou inférieure à 3, selon la source de bruit que l'on souhaite simuler.

À ce stade, les nombres aléatoires générés par la technique de Webster (1994) sont statistiquement indépendants (bruit blanc). Dans le but d'imposer une forme spectrale souhaitée à notre séquence de nombres aléatoires, un filtre FIR approprié peut être conçu et utilisé pour filtrer la séquence générée. Cependant, puisque le filtrage est un processus additif qui tend à rendre notre séquence plus gaussienne, comme décrit par le théorème central limite, la méthode de Webster impose une valeur de kurtosis spécifique à la séquence filtrée liée à la relation :

$$\beta_2(y_m) = 3 + (\beta_2(x_m) - 3) \cdot \frac{\sum_{i=0}^K W_i^4}{\left(\sum_{i=0}^K W_i^4 \right)^2} \quad (\text{A.45})$$

Cette relation permet d'exprimer le kurtosis $\beta_2(y)$ des échantillons de bruits filtrés y_m , en fonction des $K + 1$ coefficients W_i du filtre FIR ainsi que du kurtosis $\beta_2(x)$ de la séquence de nombres aléatoires x_m . À noter qu'on utilise l'expression du kurtosis normalisé pour obtenir cette relation :

$$\beta_2(x) = \frac{\Gamma(4.n + 1)}{[\Gamma(2.n + 1)]^2}, \quad (\text{A.46})$$

avec Γ la fonction gamma. Il est donc possible de spécifier le kurtosis $\beta_2(y)$ de la séquence de sortie, indépendamment des coefficients de filtrage W_i , en cherchant le kurtosis $\beta_2(x)$ de la séquence d'entrée correspondante. Cela revient à imposer un comportement non gaussien souhaité aux statistiques de bruit généré, tout en concevant un filtre FIR approprié, dont la réponse fréquentielle reproduit le niveau spectral de bruit voulu. De faibles valeurs de kurtosis vont mieux convenir pour des bruits de trafic, tandis que des valeurs plus élevées conviennent bien aux bruits d'agitation en surface.

Annexe B

Appendice B

1 Preuve du théorème 10

Supposons que

$$\sum_{k=1}^K \varphi_{m,k}^2 < \bar{\varphi}^2, \forall m, \quad (\text{B.1})$$

avec $\bar{\varphi}$ une constante positive. Nous supposons également que

$$\sum_{m=1}^M \lambda_m^2 < \bar{\lambda}^2, \quad (\text{B.2})$$

avec $\bar{\lambda}$ une autre constante positive. Considérons que $\|\mathbf{x}\| < \bar{\mathbf{x}}$ pour tout $\mathbf{x} \in \mathcal{X}$. Soit $D_1 = \bar{\mathbf{x}}$ une borne valide pour tout $x(k)$. En se basant sur le résultat de (Elbrächter et al. 2019, Lemma A.6) et la définition de \mathcal{S}_D dans l'équation (5.30), il existe un réseau de neurones complètement connecté Φ_{p_2, γ_1} avec L_1 couches, tel que nous pouvons avoir

$$\|\Phi_{p_2, \gamma_1}(t) - t^2\|_{L^\infty(I_1)} \leq \varepsilon, \quad (\text{B.3})$$

sur l'intervalle $I_1 = [-D_1, D_1]$. Le nombre de couches L_1 est borné par $L_1 \leq A(D_1)(\log \varepsilon)^2 + B(D_1)$ où $A(D_1)$ et $B(D_1)$ sont des constantes dépendantes de D_1 . Le nombre maximum de neurones par couche est limité à 16. Par souci de simplicité, nous introduisons les notations

$$s_\theta(m) = \sum_{k=1}^K \varphi_{m,k} \Phi_{p_2, \gamma_1}(x(k)), \quad (\text{B.4})$$

$$s(m) = \sum_{k=1}^K \varphi_{m,k} x^2(k), \quad (\text{B.5})$$

et considérons que les paramètres $\lambda_1, \dots, \lambda_m$ et $\varphi_{m,k}$ prennent respectivement les valeurs finies optimales c_1, \dots, c_m et $b_{m,k}$.

Puisque $\|\mathbf{x}\| < \bar{\mathbf{x}}$ est borné, nous déduisons de l'équation (B.1) que $|s(m)| < \bar{\varphi}\bar{\mathbf{x}}$ est borné pour tout m . De plus, en se basant sur l'équation (B.3), il s'ensuit que

$$|s_\theta(m)| < \bar{\varphi}(\bar{\mathbf{x}} + \varepsilon\sqrt{K}) \quad (\text{B.6})$$

est borné pour tout m . Ainsi, la valeur réelle D_2 définie par

$$\sup_{\mathbf{x} \in \mathcal{X}, m \in \llbracket M \rrbracket} \{|s_\theta(m)|, |s(m)|\} \leq \bar{\varphi}(\bar{\mathbf{x}} + \varepsilon\sqrt{K}) = D_2 \quad (\text{B.7})$$

est finie. De (Elbrächter et al. 2019, Lemma A.6), il est possible de trouver un réseau de neurones complètement connecté Φ_{e,γ_2} défini sur l'intervalle fini $I_2 = [-D_2, D_2]$ et de taille L_2 , tel que :

$$\|\Phi_{e,\gamma_2}(t) - e^t\|_{L^\infty(I_2)} \leq \varepsilon. \quad (\text{B.8})$$

Le nombre de couches L_2 est borné tel que

$$L_2 \leq A(D_2)(\log \varepsilon)^2 + B(D_2) \quad (\text{B.9})$$

où $A(D_2)$ et $B(D_2)$ sont des constantes dépendantes de D_2 et le nombre de neurones par couche est également limité à 16. Il s'ensuit que

$$\|\Phi_{e,\gamma_2}(s_\theta(m)) - e^{s_\theta(m)}\|_{L^\infty(I_1)} \leq \varepsilon. \quad (\text{B.10})$$

Aussi puisque $\exp(t)$ est une fonction croissante et dérivable $\forall t \in I_2$, et D_2 est telle que $\forall t \in I_2, |\exp'(t)| \leq \exp(D_2)$, il résulte du théorème des accroissements finis que

$$|e^{s_\theta(m)} - e^{s(m)}| \leq e^{D_2} |s_\theta(m) - s(m)|. \quad (\text{B.11})$$

De l'équation (B.3) et l'inégalité de Cauchy-Schwarz, on peut vérifier que

$$\begin{aligned} |s_\theta(m) - s(m)| &= \left| \sum_{k=1}^K \varphi_{m,k} (\Phi_{p_2,\gamma_1}(x(k)) - x^2(k)) \right| \\ &\leq \bar{\varphi}\varepsilon\sqrt{K} \end{aligned} \quad (\text{B.12})$$

Il en résulte que

$$e^{s_\theta(m)} \leq \bar{\varphi}e^{D_2}\varepsilon\sqrt{K} + e^{s(m)} \quad (\text{B.13})$$

$$|e^{s_\theta(m)} - e^{s(m)}| \leq \bar{\varphi}e^{D_2}\varepsilon\sqrt{K}. \quad (\text{B.14})$$

De là nous obtenons

$$\|\Phi_{e,\gamma_2}(s_\theta(m)) - e^{s(m)}\|_{L^\infty(I_1)} \leq \quad (\text{B.15})$$

$$\|\Phi_{e,\gamma_2}(s_\theta(m)) - e^{s_\theta(m)}\|_{L^\infty(I_1)} + \|\Phi_{e,\gamma_2}(s_\theta(m)) - e^{s(m)}\|_{L^\infty(I_1)}, \quad (\text{B.16})$$

et il s'en suit que

$$\|\Phi_{e,\gamma_2}(s_\theta(m)) - e^{s(m)}\|_{L^\infty(I_1)} \leq (1 + \bar{\varphi}e^{D_2}\varepsilon\sqrt{K})\varepsilon, \quad (\text{B.17})$$

ce qui nous donne

$$\begin{aligned} \sup_{\mathbf{x} \in \mathcal{X}} |f_\theta(\mathbf{x}) - \Lambda^*(\mathbf{x})| &\leq \sum_{m=1}^M \lambda_m \left(\Phi_{e, \gamma_2}(s_\theta(m)) - e^{s(m)} \right) \\ &\leq \bar{\lambda} (1 + \bar{\varphi} e^{D^2 \sqrt{K}}) \sqrt{M} \varepsilon. \end{aligned} \quad (\text{B.18})$$

Enfin, nous pouvons obtenir

$$\inf_{f_\theta \in \mathcal{F}^*} \sup_{\mathbf{x} \in \mathcal{X}} |f_\theta(\mathbf{x}) - \Lambda^*(\mathbf{x})| \leq C \varepsilon, \quad (\text{B.19})$$

avec $C = \bar{\lambda} (1 + \bar{\varphi} e^{\bar{\varphi}(\bar{\mathbf{x}} + \varepsilon \sqrt{K})} \sqrt{K}) \sqrt{M}$.

2 Figures

Les figures suivantes sont un complément à la section 4.2 du chapitre 6.

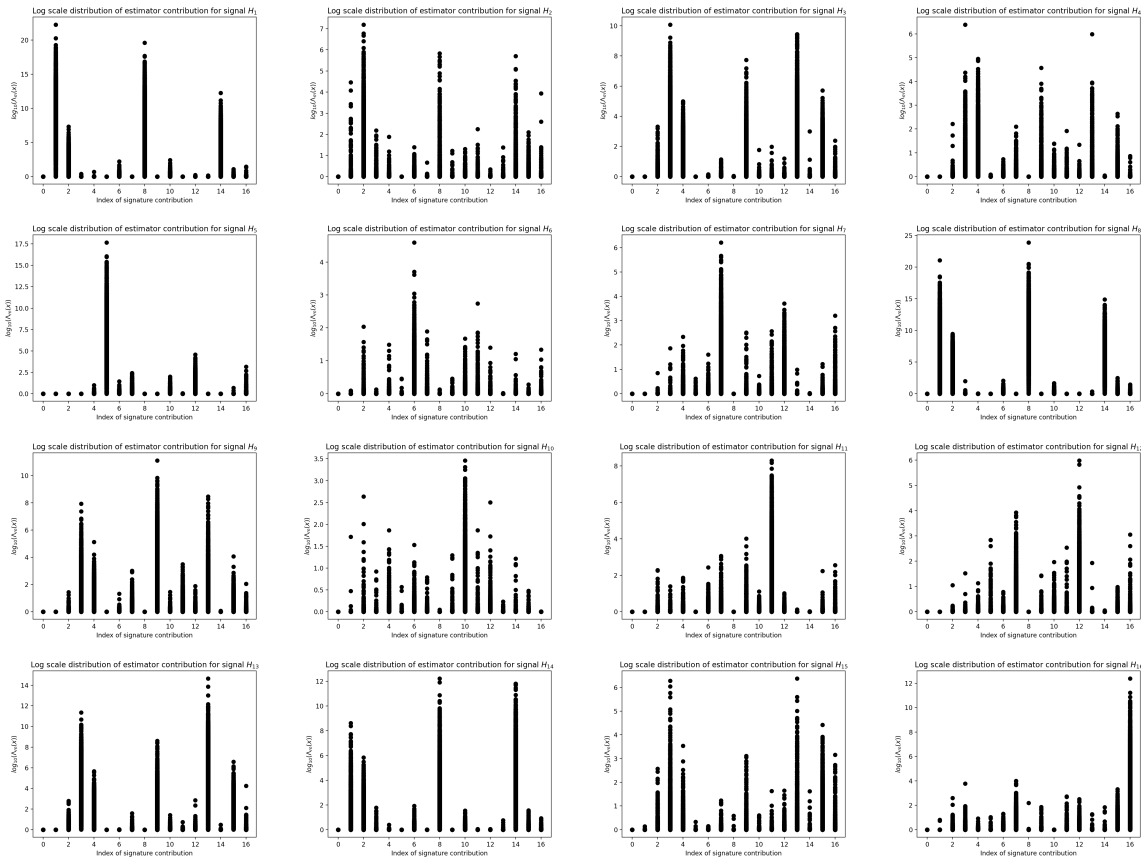


Figure B.1: Distributions des coefficients $\sum_{k=1}^K b_{l,k} x^2(k) |H_m \forall l \in \{0, 1, \dots, M\}$ où chaque image correspond à une distribution $\mathbf{x} \sim H_m$ différente.