

# ESTIMATION DE MODÈLES DE VITESSE SISMIQUE PAR APPRENTISSAGE PROFOND

Par

Jérôme Simon

Thèse présentée pour l'obtention du grade de  
*Philosophiæ Doctor, Ph.D.*  
en Sciences de la Terre

## Jury d'évaluation

Examineur interne et Président du jury	Bernard Giroux Centre Eau Terre Environnement Institut national de la recherche scientifique
Examineur externe	Jean-François Lalonde Département de génie électrique et de génie informatique Faculté de sciences et de génie Université Laval
Examineur externe	Gilles Bellefleur Commission géologique du Canada Ressources naturelles Canada
Codirecteur de recherche	Gabriel Fabien-Ouellet Département des génies civil, géologique et des mines Polytechnique Montréal
Directeur de recherche	Erwan Gloaguen Centre Eau Terre Environnement Institut national de la recherche scientifique



## REMERCIEMENTS

Cette thèse est une aventure que je n'aurais su amorcer sans la bienveillance de mon directeur de thèse, Erwan. Merci beaucoup pour la liberté que tu m'as donnée, elle était nécessaire à mon émancipation en tant que chercheur. Travailler sous ta supervision a été un privilège!

J'aimerais aussi remercier mon codirecteur de recherche, Gabriel. Tes commentaires ont toujours été extrêmement pertinents, car attentionnés, et je l'apprécie beaucoup! Merci de m'avoir donné des bases solides sur lesquelles bâtir mon projet de recherche.

Merci à Bernard, avec qui j'ai eu l'occasion de discuter et collaborer à de nombreuses reprises. Ton implication dans mon cheminement m'a marqué et je garde un beau souvenir de nos travaux de terrain ensemble.

Merci aux examinateurs externes, Jean-François et Gilles, d'avoir accepté de se joindre au jury. Je suis heureux que votre expertise puisse profiter à ma thèse.

Un merci particulier à mes anciens associés de chez Geolearn, Martin, Lorenzo et Jean-Sébastien. L'aventure que nous avons partagée m'a amené très loin et j'ai beaucoup appris de vous trois.

Les chaudes journées d'été et les vendredis après-midi n'auraient pas été les mêmes sans mes collègues et amis de l'INRS et de Polytechnique Montréal. Merci à vous tous!

Merci à mes amis proches pour les heures passées à travailler en parallèle et pour les moments de célébration partagés ensemble.

Un énorme merci à Camille. Tu m'as aidé à traverser ces longues années de doctorat à coup de rêves. Ta complicité m'a fait du bien et je n'aurais pas pu y arriver sans toi.

Merci du fond du cœur à Myriam. J'apprécie infiniment comment tu as su prendre soin de moi tout au long de cette dernière ligne droite. Merci de prendre autant de place dans ma vie, tu es bienveillante et douce.

Enfin, merci à ma famille, Manon, Christophe, Olivier, Philippe, Pénélope et Arlette. Votre support inconditionnel m'a permis d'en arriver où je suis aujourd'hui!



## AVANT-PROPOS

En parallèle avec cette thèse, j'ai eu le plaisir de participer à plusieurs projets connexes d'envergure. Je crois que ces engagements sont dignes de mention, puisqu'ils témoignent de l'impact positif que ma recherche et moi avons eu sur la vie scientifique et communautaire de l'INRS.

### Articles publiés

Les articles suivants figurent dans cette thèse :

- Simon J, Fabien-Ouellet G, Gloaguen E, Khurjekar I (2023). Hierarchical transfer learning for deep learning velocity model building. *GEOPHYSICS*, pages R79–R93. DOI :10.1190/geo2021-0470.1.
- Simon J, Fabien-Ouellet G, Gloaguen E (soumis le 16 décembre 2022). Quality control in deep learning and confidence quantification : seismic velocity regression through classification. *Computers & Geosciences*.
- Simon J, Fabien-Ouellet G, Gloaguen E (prêt à soumettre). Domain gaps in velocity model estimation : characterizing the transferability between domains. *Computers & Geosciences*.

J'ai aussi pu collaborer avec d'autres groupes de recherche au sein de l'INRS, ce qui a mené aux publications suivantes :

- Simon J, Giroux B, Bérubé R, Bérubé F, Bergeron N (soumis le 22 mars 2023). Effect of railroad traffic on embryonic survival of Atlantic salmon of Nipissis River. *Near-Surface Geophysics*.
- Dufour-Beauséjour S, Bernier M, Simon J, Hodayouni S, Gilbert V, Gauthier Y, Tuniq J, Wendleder A, Roth A (2021). Tenuous Correlation Between Snow Depth or Sea Ice Thickness and C- or X-band Backscattering in Nunavik Fjords of the Hudson Strait. *Remote Sensing*, 13(4) :768-787. DOI : 10.3390/rs13040768.

### Résumés publiés

Le matériel de cette thèse a été présenté dans diverses conférences, dont l'une internationale :

- Simon J, Fabien-Ouellet G, Gloaguen E, Khurjekar I, Araya-Polo M (2021). Velocity model building using transfer learning. *First International Meeting for Applied Geoscience & Energy Expanded Abstracts*, Society of Exploration Geophysicists, SEG Technical Program Expanded Abstracts, pages 3365–3369.
- Simon J, Fabien-Ouellet G, Gloaguen E (2020). Estimer des modèles de vitesse sismique acoustique par apprentissage profond. *Journée des Sciences de la Terre et de l'Environnement*.
- Simon J (2019). Modéliser la causalité avec des réseaux neuronaux. *Congrès Eau Terre Environnement*.

## Engagements

Mon implication a bénéficié à la recherche et à la vie étudiante de mes collègues à l'INRS et à Polytechnique Montréal, notamment à travers les engagements suivants :

- Instigateur et instructeur du cours *ETE-532/ETE-432 Géoscience des données* (2020 – auj.)
- Mainteneur de *ModelGenerator* (2021 – auj.)
- Mainteneur principal de *GeoFlow* (2020 – auj.)
- Mainteneur principal du gabarit LaTeX de l'INRS (2019 – auj.)
- Coadministrateur des serveurs de calculs du Laboratoire d'Imagerie et d'Assimilation de Mesures Géophysiques (2018 – 2021)
- Ambassadeur étudiant pour l'INRS (2020 – 2021)

## Organismes subventionnaires

J'aimerais remercier les organismes subventionnaires suivants pour leur support financier :

- Programme de Bourses d'études supérieures du Canada Vanier
- Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG)
- Fonds de recherche du Québec – Nature et technologies (FRQNT)
- Programme de bourses de la *Society of Exploration Geophysicists* (SEG)

# RÉSUMÉ

La sécurité énergétique, économique et environnementale de la société dépend grandement de l'exactitude de la connaissance de l'intérieur du sol. Une stratégie efficace pour y accéder réside dans les méthodes sismiques, qui peuvent estimer de façon non destructive la géométrie des structures souterraines à grande échelle. Or, les méthodes sismiques sont soumises à des limitations méthodologiques qui réduisent leur efficacité et leur résolution : l'imagerie et l'inversion sismiques dépendent d'un modèle de vitesse d'essai très lisse et coûteux à obtenir, en plus de nécessiter beaucoup d'intervention humaine et, par leur caractère itératif, de calculs. L'apprentissage profond permettrait de contourner ces difficultés, par sa capacité à résoudre des problèmes non linéaires. Par contre, la réussite de l'apprentissage profond dépend d'un entraînement, très coûteux en calculs, et de la conception de données massives spécifiques à chaque problème.

Dans cette thèse, j'ai pour objectif d'accélérer et d'améliorer les techniques de DL qui existent actuellement pour l'estimation de modèles de vitesse 2D, dans l'optique que les praticiens de la sismique puissent bénéficier d'un outil mature, plus rapide et versatile. Les contributions présentées dans cette thèse permettent d'améliorer nettement les méthodes existantes en diminuant par au plus 80 fois les temps de calcul et d'au plus 45% l'erreur. Pour ce faire, j'ai développé une approche d'apprentissage par transfert permettant de préentraîner des réseaux de neurones sur des modèles 1D, plus simples, et de conserver toute l'information apprise une fois en 2D. Cette stratégie réduit les besoins en calculs et permet de meilleures performances. Ensuite, j'ai reformulé le problème de régression de vitesse en un problème de classification pour contourner l'utilisation d'ensembles, lesquels étaient coûteux, mais nécessaires au réalisme. Les classificateurs fournissent des résultats plus exacts que les régresseurs et permettent d'estimer la confiance à un coût moindre que celui d'ensembles. Enfin, j'ai quantifié le surajustement de réseaux de neurones sur plusieurs domaines d'entraînement et j'ai confirmé que les écarts de domaine diminuent la performance. La performance des réseaux de neurones dépend au moins en partie des *a priori* portés par les données d'entraînement, notamment en ce qui a trait aux plages de vitesses attendues.

Quoique la création d'un outil d'estimation de modèles de vitesse parfaitement versatile à partir du réseau de neurones testé est toujours improbable, cette thèse suggère que l'adaptation de domaine, l'apprentissage par transfert et l'apprentissage actif feront partie intégrante des futures méthodologies.

**Mots-clés** : Estimation de modèles de vitesse; Inversion sismique; Réseaux de neurones profonds; Apprentissage profond; Apprentissage automatique; Intelligence artificielle; Sismique d'exploration; Géophysique appliquée.





## ABSTRACT

The energetic, economic, and environmental security of society depends greatly on the accurate knowledge of the interior of the earth. An effective strategy for accessing this knowledge lies in seismic methods, which can non-destructively estimate the geometry of the ground at a large scale. However, seismic methods are subject to methodological limitations that reduce their efficiency and resolution : seismic imaging and inversion depend on a very smooth and expensive to obtain test velocity model, in addition to requiring a great deal of human intervention and, by their iterative nature, computation. Deep learning could circumvent these difficulties, by its ability to solve non-linear problems. On the other hand, the success of deep learning depends on a training step, which is very computationally expensive, and on the design of big data specific to each problem.

In this thesis, I aim to improve the computation time and accuracy of the current DL techniques for 2D velocity model estimation, with the goal of providing seismic practitioners with a mature, faster, and versatile tool. The contributions described in this thesis significantly improve existing methods by reducing computation times by up to 80 times and errors by up to 45%. To accomplish this, I have developed a transfer learning approach that allows neural networks to be pre-trained on 1D models, which are simpler to estimate, and to retain all the information learned once in 2D. This strategy reduces computational requirements and allows reaching better performances. Next, I reformulated the velocity regression problem into a classification problem to bypass the use of ensembles, which were expensive but necessary to obtain realistic results. Classifiers provide more accurate results than regressors and allow confidence to be estimated at a cost lower than that of ensembles. Last, I quantified the overfitting of neural networks over several training domains and confirmed that domain gaps decrease performance. The performance of neural networks depends at least partly on the assumptions made through the training dataset, especially with respect to the expected velocity ranges.

Although the creation of a perfectly versatile velocity model estimation tool from the tested neural network is still unlikely, this thesis suggests that domain adaptation, transfer learning and active learning will be integral parts of future methodologies.

**Keywords** : Velocity model estimation; Seismic inversion; Deep neural networks; Deep learning; Machine learning; Artificial intelligence; Exploration seismics; Applied geophysics.



# TABLE DES MATIÈRES

<b>REMERCIEMENTS</b> .....	<b>iii</b>
<b>AVANT-PROPOS</b> .....	<b>v</b>
<b>RÉSUMÉ</b> .....	<b>vii</b>
<b>ABSTRACT</b> .....	<b>ix</b>
<b>TABLE DES MATIÈRES</b> .....	<b>xi</b>
<b>LISTE DES FIGURES</b> .....	<b>xiii</b>
<b>LISTE DES TABLEAUX</b> .....	<b>xv</b>
<b>LISTE DES ALGORITHMES</b> .....	<b>xvii</b>
<b>LISTE DES ABRÉVIATIONS</b> .....	<b>xix</b>
<b>1 INTRODUCTION</b> .....	<b>1</b>
1.1 MÉTHODES SISMIQUES .....	2
1.2 APPRENTISSAGE PROFOND .....	4
1.3 ESTIMATION DE MODÈLES DE VITESSE PAR APPRENTISSAGE PROFOND.....	5
1.4 OBJECTIFS ET CONTRIBUTIONS.....	8
1.4.1 <i>Objectif 1 : tirer parti du faible coût des données 1D pour les estimés 2D</i> .....	9
1.4.2 <i>Objectif 2 : estimer à coût réduit l'étendue possible des prédictions</i> .....	9
1.4.3 <i>Objectif 3 : caractériser la transférabilité d'un NN à d'autres données</i> .....	10
<b>2 ÉLÉMENTS THÉORIQUES</b> .....	<b>13</b>
2.1 APPRENTISSAGE PROFOND .....	13
2.1.1 <i>Entraînement</i> .....	14
2.1.2 <i>Réseaux de neurones convolutifs</i> .....	17
2.1.3 <i>Réseaux de neurones récurrents</i> .....	18
2.1.4 <i>Réseaux antagonistes génératifs</i> .....	21
2.1.5 <i>Apprentissage par transfert et adaptation de domaine</i> .....	23
2.1.6 <i>Apprentissage actif</i> .....	24
2.2 MÉTHODES SISMIQUES .....	25
2.2.1 <i>Équation d'onde</i> .....	26
2.2.2 <i>Données sismiques</i> .....	29
2.2.3 <i>Imagerie sismique</i> .....	30
2.2.4 <i>Inversion sismique</i> .....	38

<b>3</b>	<b>ESTIMER LA VITESSE PAR APPRENTISSAGE PROFOND .....</b>	<b>43</b>
3.1	GÉNÉRATION DE MODÈLES PAR OBJETS .....	44
3.1.1	<i>ModelGenerator.....</i>	46
3.2	INTERFAÇAGE À KERAS AVEC GEOFLOW .....	48
3.2.1	<i>Choix de réseau de neurones profond .....</i>	48
<b>4</b>	<b>APPRENTISSAGE PAR TRANSFERT HIÉRARCHIQUE .....</b>	<b>55</b>
4.1	MÉTHODOLOGIE .....	56
4.1.1	<i>Apprentissage par transfert.....</i>	56
4.1.2	<i>Jeu de données synthétique .....</i>	60
4.2	RÉSULTATS .....	61
4.2.1	<i>Données synthétiques.....</i>	62
4.2.2	<i>Données de terrain .....</i>	69
4.3	DISCUSSION .....	73
4.4	CONCLUSION .....	78
<b>5</b>	<b>RÉGRESSION PAR CLASSIFICATION .....</b>	<b>81</b>
5.1	MÉTHODOLOGIE .....	83
5.1.1	<i>Estimer la vitesse par classification .....</i>	83
5.1.2	<i>Jeu de données.....</i>	86
5.2	RÉSULTATS ET DISCUSSION.....	86
5.2.1	<i>Quantité de classes .....</i>	86
5.2.2	<i>Comparaison des régresseurs et des classificateurs.....</i>	86
5.2.3	<i>Analyse de sensibilité.....</i>	90
5.3	CONCLUSION .....	95
<b>6</b>	<b>CARACTÉRISER LES ÉCARTS DE DOMAINES .....</b>	<b>99</b>
6.1	MÉTHODOLOGIE .....	100
6.1.1	<i>Jeux de données synthétiques .....</i>	101
6.2	RÉSULTATS ET DISCUSSION.....	102
6.2.1	<i>Impact du contraste maximal de vitesse <math>\Delta V</math> entre couches .....</i>	103
6.2.2	<i>Impact de la fréquence de crête <math>f</math> de l'ondelette .....</i>	106
6.2.3	<i>Impact de discontinuités verticales de décrochement maximal <math>d</math>.....</i>	109
6.3	CONCLUSION .....	111
<b>7</b>	<b>CONCLUSION.....</b>	<b>115</b>
	<b>BIBLIOGRAPHIE.....</b>	<b>119</b>

## LISTE DES FIGURES

FIGURE 2.1	NEURONE ARTIFICIEL .....	14
FIGURE 2.2	RÉSEAU DE NEURONES PROFOND .....	15
FIGURE 2.3	COUCHE CONVOLUTIVE D'UN RÉSEAU DE NEURONES CONVOLUTIF .....	18
FIGURE 2.4	RÉSEAU DE NEURONES RÉCURRENT .....	19
FIGURE 2.5	ÉLARGISSEMENT DU CHAMP RÉCEPTIF AVEC WAVENET .....	21
FIGURE 2.6	RÉSEAU ANTAGONISTE GÉNÉRATIF .....	22
FIGURE 2.7	APPRENTISSAGE PAR TRANSFERT .....	24
FIGURE 2.8	RÉGIMES DE PROPAGATION DES ONDES .....	28
FIGURE 2.9	CONVERSION DES ONDES .....	29
FIGURE 2.10	GÉOMÉTRIE D'ACQUISITION .....	30
FIGURE 2.11	DONNÉES SISMIQUES REGROUPÉES EN TIRS .....	30
FIGURE 2.12	TRAITEMENT DE DONNÉES TYPIQUE EN IMAGERIE SISMIQUE .....	32
FIGURE 2.13	DIFFÉRENTES REPRÉSENTATIONS DE LA VITESSE .....	34
FIGURE 2.14	ANALYSE DE VITESSE .....	34
FIGURE 2.15	SAUT DE CYCLE .....	40
FIGURE 3.1	ESTIMER DES MODÈLES DE VITESSE PAR APPRENTISSAGE PROFOND .....	44
FIGURE 3.2	EXEMPLES DE MODÈLES DE VITESSE 2D PRODUITS PAR MODELGENERATOR .....	47
FIGURE 3.3	RÉSEAU DE NEURONES UTILISÉ PAR ARAYA-POLO ET AL. (2018).....	50
FIGURE 3.4	RÉSEAU DE NEURONES UTILISÉ PAR KAZEI ET AL. (2021) .....	50
FIGURE 3.5	RÉSEAU DE NEURONES INSPIRÉ DE FABIEN-OUELLET ET AL. (2020) .....	51
FIGURE 4.1	RÉSEAU DE NEURONES MULTITÂCHE .....	60
FIGURE 4.2	FONCTION COÛT LORS DE L'ENTRAÎNEMENT .....	63
FIGURE 4.3	ESTIMATIONS MOYENNES POUR LE 50 <sup>E</sup> PERCENTILE .....	65
FIGURE 4.4	ESTIMATIONS MOYENNES POUR LE 90 <sup>E</sup> PERCENTILE .....	66
FIGURE 4.5	ESTIMATIONS MOYENNES POUR LE 10 <sup>E</sup> PERCENTILE .....	67
FIGURE 4.6	ESTIMATIONS INDIVIDUELLES DE $V(t, x)$ POUR LE 50 <sup>E</sup> PERCENTILE .....	68
FIGURE 4.7	ESTIMATIONS INDIVIDUELLES DE $V(z, x)$ POUR LE 50 <sup>E</sup> PERCENTILE .....	69
FIGURE 4.8	HISTOGRAMMES NORMALISÉS DE LA RMSE SUR $V(t, x)$ PAR COUCHE .....	70
FIGURE 4.9	MODÈLES DE VITESSE ESTIMÉS POUR LA LIGNE 32 DE USGS (1978) .....	72

FIGURE 4.10	ESTIMATIONS INDIVIDUELLES DE $V(t, x)$ POUR LES DONNÉES DE TERRAIN .....	73
FIGURE 4.11	ESTIMATIONS INDIVIDUELLES DE $V(z, x)$ POUR LES DONNÉES DE TERRAIN.....	74
FIGURE 4.12	CORRECTION NMO ET SEMBLANCE POUR LES DONNÉES DE TERRAIN .....	75
FIGURE 4.13	IMAGES SISMIQUES TRAITÉES DE LA LIGNE 32 DE USGS (1978).....	76
FIGURE 5.1	MODIFIER UN RÉGRESSEUR EN UN CLASSIFICATEUR .....	85
FIGURE 5.2	FONCTION COÛT ASSOCIÉE À $V(t)$ PENDANT L'ENTRAÎNEMENT .....	87
FIGURE 5.3	SOUS-ENSEMBLE DES PRÉDICTIONS DU JEU DE DONNÉES TEST .....	89
FIGURE 5.4	COMPARAISON DÉTAILLÉE DES PRÉDICTIONS INDIVIDUELLES .....	90
FIGURE 5.5	HISTOGRAMME DES PAIRES ERREUR-MÉFIANCE .....	91
FIGURE 5.6	IMPACT DU BRUIT BLANC SUR LES PRÉDICTIONS ET LA MÉFIANCE .....	92
FIGURE 5.7	IMPACT DU PENDAGE $\theta$ SUR LES PRÉDICTIONS ET LA MÉFIANCE .....	94
FIGURE 5.8	IMPACT DES DISCONTINUITÉS SUR LES PRÉDICTIONS ET LA MÉFIANCE.....	96
FIGURE 6.1	MÉTHODOLOGIE POUR ÉVALUER LA TRANSFÉRABILITÉ INTERDOMAINES.....	102
FIGURE 6.2	EXEMPLES AVEC CONTRASTE MAXIMAL DE VITESSE $\Delta V$ VARIABLE .....	104
FIGURE 6.3	IMPACT DU CONTRASTE MAXIMAL DE VITESSE $\Delta V$ .....	105
FIGURE 6.4	EXEMPLES AVEC FRÉQUENCE DE CRÊTE MOYENNE $f$ VARIABLE .....	107
FIGURE 6.5	IMPACT DE LA FRÉQUENCE DE CRÊTE MOYENNE $f$ .....	108
FIGURE 6.6	EXEMPLES AVEC DISCONTINUITÉ DE DÉCROCHEMENT MAXIMAL $d$ VARIABLE...	110
FIGURE 6.7	IMPACT DE DISCONTINUITÉS DE DÉCROCHEMENT MAXIMAL $d$ .....	111

## LISTE DES TABLEAUX

TABLEAU 3.1	DÉTAILS DES COUCHES CONVOLUTIVES .....	51
TABLEAU 4.1	PONDÉRATIONS $\alpha_i$ PENDANT LES TROIS ÉTAPES D'ENTRAÎNEMENT .....	60
TABLEAU 4.2	JUSTESSE DE $V(t, x)$ SUR LE JEU D'ENTRAÎNEMENT OU CELUI DE TEST .....	63
TABLEAU 5.1	OPTIMISATION DE LA QUANTITÉ DE CLASSES DU CLASSIFICATEUR.....	87
TABLEAU 6.1	IMPACT DU CONTRASTE MAXIMAL DE VITESSE $\Delta V$ .....	103
TABLEAU 6.2	IMPACT DE LA FRÉQUENCE DE CRÊTE MOYENNE $f$ .....	106
TABLEAU 6.3	IMPACT DE DISCONTINUITÉS DE DÉCROCHEMENT MAXIMAL $d$ .....	109





## LISTE DES ALGORITHMES

ALGORITHME 3.1	GÉNÉRER DES MODÈLES DE PROPRIÉTÉS AVEC MODELGENERATOR.....	46
ALGORITHME 3.2	ESTIMER DES MODÈLES DE PROPRIÉTÉS AVEC GEOFLOW .....	49



## LISTE DES ABRÉVIATIONS

$V_P$ ou $V$	Vitesse des ondes de compression
$V_S$	Vitesse des ondes de cisaillement
Adam	<i>Adaptive Moment Estimation</i>
AI	Intelligence artificielle ( <i>artificial intelligence</i> )
AL	Apprentissage actif ( <i>active learning</i> )
CMP	Point milieu commun ( <i>common mid-point</i> )
CNN	Réseau de neurones convolutif ( <i>convolutional neural network</i> )
DL	Apprentissage profond ( <i>deep learning</i> )
DMO	<i>Dip moveout</i>
FCL	Couche entièrement connectée ( <i>fully-connected layer</i> )
FWI	Inversion en onde complète ( <i>full waveform inversion</i> )
GPU	Processeur graphique ( <i>graphics processing unit</i> )
LSRTM	<i>Least-Squares Reverse Time Migration</i>
LSTM	Mémoire à court et à long terme ( <i>long-short term memory</i> )
ML	Apprentissage automatique ( <i>machine learning</i> )
NMO	<i>Normal moveout</i>
NN	Réseau de neurones profond ( <i>neural network</i> )
RMSE	Racine carré de l'erreur quadratique moyenne ( <i>root-mean-square error</i> )
RNN	Réseau de neurones récurrent ( <i>recurrent neural network</i> )
RTM	<i>Reverse Time Migration</i>
SGD	Algorithme du gradient stochastique ( <i>stochastic gradient descent</i> )
SSIM	Indice de similarité structurelle ( <i>structural similarity index measure</i> )
TL	Apprentissage par transfert ( <i>transfer learning</i> )



# 1 INTRODUCTION

---

La garantie de l'approvisionnement énergétique de la société et la gestion responsable des émissions de dioxyde de carbone et des ressources souterraines (géothermiques, hydrocarburées, minérales) sont grandement liées à la connaissance détaillée de la structure du sol (Société Géologique de Londres et al., 2015; Adler et al., 2021). Il est donc nécessaire de savoir imager le plus correctement et efficacement possible les structures géologiques. Par exemple, le taux de succès en exploration pétrolière, où les moyens sont pourtant énormes, est environ de 30% (Rudolph et al., 2017), ce qui signifie qu'accélérer et améliorer le processus décisionnel menant aux forages pourrait sauver des centaines de millions de dollars (Adler et al., 2021) et participer à mitiger les risques environnementaux associés à l'exploitation des réservoirs profonds. Or, identifier précisément, objectivement et de façon continue de grands volumes de roches est un défi majeur. Sonder le sol à grande échelle et à grande profondeur à l'aide de méthodes d'investigations invasives, par exemple les forages, est irréalisable et coûteux, d'autant plus qu'elles représentent un risque pour la sécurité et l'environnement (Adler et al., 2021). Il existe donc un besoin pour des méthodes d'estimation non intrusives et de grande échelle de la distribution spatiale de propriétés physiques : c'est le besoin auquel répondent les méthodes géophysiques, dont les méthodes sismiques.

Les méthodes sismiques actives permettent d'effectuer une échographie du sol : des ondes sismiques sont générées artificiellement à la surface, une partie du signal est récupérée par des capteurs et le parcours des ondes est déduit à l'aide des connaissances liées à la physique de la propagation des ondes. Les données recueillies sont traitées par imagerie sismique ou par inversion. L'imagerie sismique permet de décrire le sol en termes de ses interfaces, à travers un modèle de réflectivité des ondes sismiques. Les algorithmes d'imagerie sismique dépendent fortement d'un modèle de vitesse décrivant le plus justement possible la dynamique des ondes dans le sol. Obtenir ce modèle représente une grande quantité des efforts déployés en imagerie sismique (Araya-Polo et al., 2018). L'inversion sismique, au contraire, cherche à estimer directement le modèle de vitesse représentant un milieu. Or, l'inversion sismique est sensible aux basses fréquences, plutôt lente par son caractère itératif, et demande néanmoins un modèle de vitesse de départ réaliste (Adler et al., 2021). Une méthode permettant d'obtenir un modèle de vitesse de

haute qualité à partir des données sismiques brutes sans passer par un modèle d'essai représenterait un progrès méthodologique considérable.

L'intelligence artificielle (AI, de *artificial intelligence*), par sa facilité à solutionner des problèmes hautement non linéaires et mal posés, constitue un candidat idéal pour l'estimation de modèles de vitesse. L'avènement récent de l'AI a amené l'apparition d'algorithmes basés sur les données pouvant surpasser tous les autres lorsqu'une quantité massive de données est disponible, ce qui est le cas en sismique (Adler et al., 2021). Tirer parti de ces méthodes pourrait améliorer la justesse et la rapidité du traitement des données sismiques, puisqu'elles sont très performantes sur des problèmes non linéaires et qu'elles peuvent générer des résultats en aussi peu que quelques secondes. Or, les problèmes de sismique sont difficiles à résoudre même à l'aide de ces nouveaux outils : il n'existe souvent pas de valeurs de référence pour les cibles à prédire et il n'existe pas de façon directe et unique d'appliquer l'AI à la sismique, à moins de concevoir des hypothèses très contraignantes et/ou spécifiques à une géologie visée. Actuellement, avec l'AI, il faut créer des méthodologies personnalisées pour venir à bout de chaque nouveau problème de sismique. Pour permettre aux praticiens de la sismique de tirer profit des capacités de l'AI, il faut que les outils d'AI puissent être utilisables plus rapidement et dans une plus grande variété de contextes.

## **1.1 Méthodes sismiques**

Plus spécifiquement, l'objectif principal de la sismique active (simplement appelée « sismique » dans le reste de cette thèse) est d'imager le sol en actionnant des sources de vibrations artificielles à la surface, vibrations qui vont ensuite se réfléchir et se réfracter sur les hétérogénéités rencontrées. Les méthodes sismiques sont couramment utilisées en exploration minière et pétrolière (Sheriff et al., 1995; Yilmaz, 2001), en caractérisation de réservoirs pour la géothermie ou la séquestration du CO<sub>2</sub> (Sauvageau et al., 2014), en génie civil (Malhotra et al., 2003) et en archéologie (Sheriff et al., 1995). L'utilité de la sismique relève du fait qu'elle permet de compléter la couverture spatiale des investigations par puits en offrant une couverture bidimensionnelle ou tridimensionnelle à travers le volume étudié à un coût relativement faible, d'où il s'agit d'un outil de choix en exploration minière et pétrolière (Sheriff et al., 1995). Dans un contexte d'explora-

tion, la sismique est nettement avantageuse par rapport aux forages, puisqu'elle ne nécessite pas d'infrastructure et peu de matériel encombrant, l'étude s'effectuant depuis la surface.

Un levé sismique, c'est-à-dire une étude sismique, est un procédé dynamique : des géophones sont disposés au sol ou en forage et une source de vibrations est activée, puis ce processus est répété en déplaçant la source et/ou les géophones. Les géométries d'acquisition sont variées, mais elles consistent généralement en une série de géophones, alignés ou non avec la source. Les vibrations sont habituellement engendrées par une masse heurtant le sol, par des sources vibrantes, ou par des explosifs (Yilmaz, 2001). Les données acquises sont composées d'un signal, appelé trace sismique, par combinaison de géophone et de source.

La séquence de traitement typique en imagerie sismique est organisée autour d'algorithmes de « migration ». À terme, l'objectif de l'imagerie sismique (et de la migration, en particulier) est de déformer les trains d'ondes de façon à ce que les maxima d'amplitude coïncident avec les structures géologiques. Le résultat est un modèle de réflectivité, semi-quantitatif car hautement manipulé, décrivant la géométrie du sol en profondeur. La nécessité d'une intervention humaine presque constante biaise l'interprétation et contraint les possibilités méthodologiques (Araya-Polo et al., 2019). De plus, pour déformer correctement les trains d'ondes, la migration dépend d'un modèle de vitesse, obtenu en amont. L'analyse de vitesse en imagerie sismique, quoique nécessaire, est trop imprécise et de trop faible résolution pour être utilisée *a posteriori* (Fabien-Ouellet, 2017). Pendant la migration, les structures détectées sont considérées comme des perturbations et le modèle de vitesse, quoiqu'il gagnerait à être modifié, reste généralement inchangé, ce qui limite la résolution pouvant être atteinte.

C'est pour pallier certains de ces problèmes qu'existe l'inversion sismique. L'inversion est le processus par lequel la physique des ondes est inversée pour retrouver le modèle de propriétés physiques d'après les traces sismiques acquises. Les méthodes inverses permettent d'inférer de grands cubes de propriétés physiques (par exemple, la vitesse, la porosité) directement à partir des traces sismiques et elles permettent de reconstituer des milieux anisotropes et viscoélastiques (Virieux et al., 2009; Operto et al., 2013; Fabien-Ouellet, 2017; Yang et al., 2017, 2019). Avec l'augmentation des capacités de traitement numérique, l'entière des trains d'ondes peut maintenant être modélisée dans l'inversion, ce qui a donné naissance à l'inversion en onde complète (FWI,

de *full waveform inversion*). La FWI est un procédé itératif par lequel les erreurs entre le signal sismique modélisé et celui observé sont retropropagées à travers le modèle de vitesse pour y apporter des corrections. À la différence de l'imagerie sismique, la FWI peut donc modifier dynamiquement le modèle de vitesse, quoiqu'elle nécessite tout de même un modèle de vitesse initial suffisamment réaliste pour converger vers une solution acceptable (Bunks et al., 1995; Li et al., 2016; Ovcharenko et al., 2019). Bien qu'elle soit l'état de l'art actuel en sismique (Araya-Polo et al., 2019), la FWI est un processus itératif très long, puisqu'à chaque correction du modèle de vitesse, l'algorithme doit estimer numériquement les traces sismiques à travers une approximation de l'équation d'onde (Mosser et al., 2018; Araya-Polo et al., 2019).

## 1.2 Apprentissage profond

L'apprentissage profond (DL, de *deep learning*) est un sous-domaine de l'apprentissage automatique (ML, de *machine learning*), qui, lui-même, est un sous-domaine de l'AI. L'AI est décrite comme « l'effort d'automatiser des tâches intellectuelles normalement effectuées par des humains » (Chollet, 2018, traduction libre). Par souci de pertinence, AI réfèrera spécifiquement au DL à travers cette thèse.

Le DL regroupe des algorithmes appelés les réseaux de neurones profonds (NN, de *neural network*). Les NNs sont des algorithmes entraînaibles où des cellules de calcul, les neurones, sont agencées en couches. L'utilité du DL (supervisé) est de permettre de modéliser des relations hautement non linéaires et abstraites, et ce, en mettant à profit une quantité immense de paires de données et de réponses (Chollet, 2018). Avec une quantité suffisante de ces paires, c'est-à-dire en échantillonnant pratiquement tous les cas de figure possibles, un NN peut cartographier le lien entre les données et les réponses.

Le DL existe depuis plusieurs décennies, c'est-à-dire depuis l'invention du perceptron par Rosenblatt (1958), et les paradigmes du ML depuis plus longtemps encore. Pourtant, ce n'est que récemment que le DL est parvenu à se tailler une place dans pratiquement tous les domaines de la science (LeCun et al., 2015; Jin et al., 2017; Lucas et al., 2018). Ce retard est dû à la quantité immense de calculs et à la tout aussi immense quantité de données nécessaires au bon



fonctionnement d'un algorithme de DL (Chollet, 2018). Seule l'apparition récente de processeurs graphiques (GPU, de *graphics processing unit*) ultraperformants pour le calcul parallèle et la disponibilité de bibliothèques *open-source*, dont *TensorFlow* (Abadi et al., 2015), a pu rendre l'utilisation du DL accessible. Avec le temps, les besoins en données massives pour des applications générales se sont comblés et il existe des jeux de données massifs disponibles publiquement pour plusieurs des problèmes de vision par ordinateur. Le jeu de données COCO de Microsoft (Lin et al., 2015), par exemple, contient des centaines de milliers d'images destinées à la détection d'objets et la segmentation de scènes. Or, la création de jeux de données aussi vastes pour chaque tâche donnée s'avère impossible et le DL ne peut être appliqué à plusieurs domaines dans lesquels il y a un manque en données massives, malgré leur existence dans des domaines plus ou moins connexes (Wang et al., 2018a).

En d'autres mots, le DL performe de façon excellente dans des tâches très précises, mais utiliser efficacement les NNs et les données massives existantes dans des tâches qui s'écartent des applications courantes demeure un enjeu. Ainsi, plus récemment, les freins à l'application du DL se sont avérés méthodologiques : on cherche comment mettre en rapport tous les jeux de données existants et étendre facilement les apprentissages d'un NN. Les approches actuellement en développement font appel à l'apprentissage par transfert (TL, de *transfer learning*; Bengio et al. 2011), à l'adaptation de domaine (Wang et al., 2018a) et à l'apprentissage actif (AL, de *active learning*; Bengio et al. 2021; Ren et al. 2021; Bengio et al. 2022).

### **1.3 Estimation de modèles de vitesse par apprentissage profond**

Ainsi, la disponibilité d'un jeu de données massif demeure un prérequis à l'utilisation du DL. Les centaines de tébioctets de données sismiques existantes à l'échelle mondiale (Adler et al., 2021) font de la sismique une candidate particulièrement bien adaptée à des applications de DL. Réciproquement, le DL est un outil intéressant pour la sismique, puisque l'estimation de modèles de propriétés physiques du sol est un problème non linéaire et très mal posé.

En raison de la quantité de données sismiques acquises et du coût en temps de leur traitement, il existe un désir d'automatisation (Araya-Polo et al., 2018). Pour cette raison, de nombreuses

applications de DL ont été développées pour contribuer à des sous-étapes du traitement sismique : l'identification des premières arrivées (Ross et al., 2018a), le suivi d'horizon (Wu et al., 2018), l'identification de la phase (Ross et al., 2018a,b), la déconvolution (Gao et al., 2022b; Liu et al., 2022), la correction NMO (Biswas et al., 2018), la migration (Vamaraju et al., 2019; Ding et al., 2022), l'analyse de vitesse (Park et al., 2020; Wang et al., 2021b), le calage de données aux puits (Tschannen et al., 2022), la décomposition et la classification de signaux sismiques (Lowney et al., 2021; Wang et al., 2021a; Kim et al., 2022a; Roncoroni et al., 2022), le filtrage des multiples (Siahkoochi et al., 2019b; Wang et al., 2022a), le filtrage du bruit (Pham et al., 2022; Saad et al., 2022; Yang et al., 2022, 2023), l'interpolation de données manquantes (Wang et al., 2020; Chai et al., 2021; Fang et al., 2021; Liu et al., 2021b), l'identification de faciès (Li et al., 2021a; Zhang et al., 2021; Di et al., 2022b; Pradhan et al., 2022), la détection de chenaux (Pham et al., 2020; Gao et al., 2021), la détection de dômes de sel (Waldeland et al., 2018) et la détection de failles (Araya-Polo et al., 2017; Wu et al., 2019a; Feng et al., 2021b; Gao et al., 2022a).

Parallèlement, le DL a été utilisé pour estimer directement la vitesse des milieux, en espérant passer entièrement outre le long processus de traitement sismique.

Comme la FWI apporte un degré de certitude supplémentaire par rapport au DL, plusieurs s'en remettent encore à la FWI et utilisent le DL pour régulariser le processus (Lewis et al., 2017; Huang et al., 2018; Richardson, 2018a; Campos et al., 2019; Wu et al., 2019c; Zhang et al., 2019; Li et al., 2021c; Yao et al., 2022). D'autres remplacent la FWI par un NN duquel les paramètres entraînés sont la vitesse sismique en chaque point du modèle (Richardson, 2018b; Sun et al., 2019; Ren et al., 2020; Zhu et al., 2021). En effet, puisque les NNs peuvent approximer n'importe quelle fonction (Hornik et al., 1989), un NN peut être composé de cellules implémentant les lois de propagation des ondes. Ces approches de DL sont « basées uniquement sur la physique » et elles sont soumises aux mêmes limitations physiques que la FWI, c'est-à-dire qu'elles sont très coûteuses en calculs, qu'elles requièrent un bon estimé de départ pour ne pas converger vers un minimum local (Araya-Polo et al., 2018, 2019; Ovcharenko et al., 2019) et qu'elles souffrent de non-unicité sévère causée par des « sauts de cycle » (Bunks et al., 1995; Li et al., 2016; Ovcharenko et al., 2019). Par contre, l'apparition des approches basées uniquement sur la physique a permis la création de méthodes hybrides de DL « informées sur la physique », lesquelles peuvent com-

porter des composantes arbitraires (Sun et al., 2021a). Les méthodes informées par la physique permettent de balancer les contributions des hypothèses physiques et des observations pendant l'entraînement, ce qui régularise le problème. Elles permettent aussi de contourner l'aspect itératif de la FWI et de produire des prédictions rapidement sur les données cibles, puisqu'un NN produit les modèles de vitesse. Néanmoins, les méthodes informées sur la physique imposent des restrictions physiques sur les NNs lors de leur entraînement, ce qui pourrait limiter leur expressivité.

À l'opposé, plusieurs tentent de réduire au minimum les hypothèses nécessaires à l'estimation de modèles de vitesse par DL en utilisant des méthodes « basées uniquement sur les données ». Les méthodes basées uniquement sur les données permettent en théorie d'utiliser à plein potentiel la capacité remarquable des NNs à approximer des fonctions non linéaires (Li et al., 2020). Le DL permet de tirer parti de toute l'information disponible dans un signal, de filtrer les hypothèses incorrectes sur la propagation des ondes, en plus d'être beaucoup plus rapide que la FWI (Araya-Polo et al., 2019). Or, la plupart des méthodes basées sur les données ne parviennent à résoudre que des problèmes simples ou très bien contraints. Par exemple, bien que plusieurs soient en mesure de prédire des modèles comportant des couches à pendage non nul, des failles et des dômes de sel (Araya-Polo et al., 2018, 2019; Yang et al., 2019; Li et al., 2020; Wu et al., 2020; Zhang et al., 2020; Sun et al., 2021b; Liu et al., 2021a), leurs données d'entraînement présentent généralement de fortes ressemblances géométriques et statistiques aux données de test. Notamment, Kazei et al. (2021) sont en mesure d'appliquer le DL à des modèles aussi complexes que celui de Marmousi2 (Martin et al., 2006), mais leur NN nécessite un entraînement sur des modèles très ressemblants obtenus à partir de celui de Marmousi (Brougois et al., 1990). D'autres approches demandent de contraindre l'entraînement avec des *a priori* géologiques qui sont généralement indisponibles au moment de l'inversion (Aleari et al., 2021; Di et al., 2022b) ou demandent de l'intervention humaine (Aleari et al., 2021), ou encore demandent de compresser le signal sismique (Di et al., 2022a,b). Les approches présentant les plus forts potentiels de généralisation traitent des modèles 1D, simples ou sans réalisme géologique (Das et al., 2019; Zheng et al., 2019; Fabien-Ouellet et al., 2020; Mosser et al., 2020a; Wang et al., 2022b; Yan et al., 2022; Alzahrani et al., 2022). Enfin, peu de méthodes sont applicables à des données réelles (Fabien-Ouellet et al., 2020; Di et al., 2022a,b; Wang et al., 2022b; Yan et al., 2022). Dans les méthodes basées uniquement

sur les données, les hypothèses physiques sont essentiellement portées par le jeu de données d'entraînement. La plupart des approches basées uniquement sur les données sont peu versatiles, car les NNs ont de la difficulté à extrapoler leurs apprentissages à des données qui diffèrent de celles d'entraînement.

En général, des contraintes fortes, qu'elles soient physiques, géométriques ou statistiques, sont nécessaires pour effectuer de l'estimation de modèles de vitesse par DL. Il existe encore plusieurs freins à la création d'un NN pouvant estimer, à partir de données brutes, les propriétés physiques de milieux dont la géologie sous-jacente est arbitraire.

## 1.4 Objectifs et contributions

En d'autres termes, pour imaginer correctement le sol avec les techniques actuelles de DL basé uniquement sur les données, il apparaît nécessaire que les algorithmes soient fortement contraints. Cela demande aux praticiens de conditionner les NNs avec des *a priori* statistiques se matérialisant dans les jeux de données d'entraînement. Or, concevoir un jeu de données d'entraînement bien contraint demande du temps et de l'expertise et plusieurs prototypes peuvent être requis avant de pouvoir mener un NN à convergence. Ces difficultés, combinées au caractère de « boîte noire » des NNs, détournent les professionnels du DL basé sur les données, qui vont souvent préférer utiliser les modèles obtenus par DL comme point de départ à la FWI, plus robuste et versatile. **L'objectif général de cette thèse est d'accélérer et d'améliorer les techniques de DL qui existent actuellement pour l'estimation de modèles de vitesse.** Les stratégies développées dans cette thèse permettent d'améliorer nettement les méthodes actuelles d'estimation de modèles de vitesse 2D par DL en diminuant par au plus 80 fois les temps de calcul et d'au plus 45% l'erreur. Mon approche utilise le NN proposé par Fabien-Ouellet et al. (2020) et certaines des conclusions resteront spécifiques à leur NN en particulier.

Plus spécifiquement, après un survol théorique (chapitre 2) et méthodologique (chapitre 3), cette thèse couvre trois volets qui ont pour objectifs de :

1. Tirer parti du faible coût des données 1D pour l'estimation de la vitesse en 2D;
2. Estimer à coût réduit l'étendue possible des prédictions de vitesse;

3. Caractériser la transférabilité d'un NN à d'autres données sismiques.

#### 1.4.1 Objectif 1 : tirer parti du faible coût des données 1D pour les estimés 2D

Fabien-Ouellet et al. (2020) ont montré qu'un NN inspiré vaguement de la séquence de traitement en analyse de semblance peut prédire des modèles de vitesse 1D en couches minces et qu'il peut être appliqué à des données de terrain. **L'objectif au chapitre 4 est d'entraîner ce NN sur des données 2D en préservant toute l'information apprise en 1D, de façon à accélérer l'entraînement.** J'y montre qu'il existe un équivalent géométrique entre le cas 1D et 2D, duquel on peut tirer profit par TL.

Ce matériel est publié chez *GEOPHYSICS* :

Simon J, Fabien-Ouellet G, Gloaguen E, Khurjekar I (2023). Hierarchical transfer learning for deep learning velocity model building. *GEOPHYSICS*, pages R79–R93. DOI :10.1190/geo2021-0470.1.

Les contributions à l'avancement des connaissances en DL appliqué à la sismique présentées dans cet article sont les suivantes :

- Développer une méthode qui, par des arguments physiques et géométriques, permet d'effectuer du TL entre les cas 1D et 2D sans perte d'information;
- Étendre le NN de Fabien-Ouellet et al. (2020) au cas 2D;
- Étudier la capacité de ce NN à prédire dans le domaine de la profondeur;
- Accélérer l'entraînement par 10 fois (au plus) l'entraînement;
- Diminuer l'erreur d'un NN de 35% (au plus) par rapport à un entraînement en 2D seulement.

#### 1.4.2 Objectif 2 : estimer à coût réduit l'étendue possible des prédictions

Pour parvenir à des résultats satisfaisants lorsque les prédictions sont incertaines, plusieurs auteurs moyennent les résultats d'ensembles de NNs (Fabien-Ouellet et al., 2020; Kazei et al., 2021). Cela survient lorsque le domaine d'application diffère du domaine d'entraînement, par exemple pour l'application à des données de terrain. **L'objectif au chapitre 5 est d'obtenir des distributions de vitesses probables en chaque point de l'espace à un coût réduit par rapport aux**

**méthodes d'ensemble.** J'y montre qu'il est possible de le faire en reformulant le problème de régression de vitesses en un problème de classification. Par le fait même, cela permet l'accès à de l'information sur la fiabilité des NNs dans le domaine de test, en plus de régulariser l'entraînement.

Ce matériel est soumis chez *Computers & Geosciences* :

Simon J, Fabien-Ouellet G, Gloaguen E (soumis le 16 décembre 2022). Quality control in deep learning and confidence quantification : seismic velocity regression through classification. *Computers & Geosciences*.

Les contributions présentées dans cet article sont les suivantes :

- Reformuler l'estimation de modèles de vitesse en faisant prédire à des NNs leur degré de confiance envers des vitesses données;
- Accélérer par 8 fois l'entraînement en remplaçant les ensembles par un seul NN (lorsque l'ensemble comporte 16 NNs);
- Diminuer l'erreur de 15% (au plus) par rapport à l'approche par régression;
- Créer et mettre à l'essai une métrique de méfiance permettant de vérifier l'utilisabilité des résultats de NNs.

### **1.4.3 Objectif 3 : caractériser la transférabilité d'un NN à d'autres données**

On observe des baisses de performance dès qu'il existe un écart entre les données d'entraînement et de test, et ce, malgré l'utilisation de procédés de régularisation. Ce frein méthodologique apparaît dans la littérature, mais n'a jamais été nommé explicitement. **L'objectif au chapitre 6 est de montrer l'importance d'un jeu d'entraînement représentatif en DL pour l'estimation de modèles de vitesse et des pistes de solution pour éviter les réentraînements.** J'y montre que la transférabilité d'un NN entre les domaines synthétiques d'entraînement et de test est affectée par du surentraînement dès qu'un écart statistique ou géométrique est introduit entre les jeux de données.

Ce matériel fait l'objet d'un article prêt à soumettre chez *Computers & Geosciences* :

Simon J, Fabien-Ouellet G, Gloaguen E (prêt à soumettre). Domain gaps in velocity model estimation : characterizing the transferability between domains. *Computers & Geosciences*.

Les contributions présentées dans cet article sont les suivantes :

- Montrer quantitativement que les *a priori* statistiques du domaine d'entraînement sont internalisés par un NN;
- Analyser qualitativement la performance des NNs lorsqu'ils sont présentés à des structures ou à des phénomènes qui sont absents des modèles de vitesse d'entraînement;
- Formaliser la génération de modèles de vitesse par objets et justifier son utilisation.





## 2 ÉLÉMENTS THÉORIQUES

---

### 2.1 Apprentissage profond

Tout algorithme qui n'est pas explicitement programmé, dont le comportement découle de l'observation de données, constitue du ML (Goodfellow et al., 2016; Chollet, 2018). Le ML comporte des algorithmes qui permettent de relier des données d'entrée à des étiquettes (*labels*) en optimisant des modèles en amont à l'aide de données de référence, appelées données d'« entraînement ». La régression linéaire en est un exemple, mais le ML inclut des outils beaucoup plus sophistiqués, dont les forêts aléatoires (Ho, 1995), les modèles de mélange gaussien (Reynolds, 2015) et la panoplie d'algorithmes existants en DL. Le DL est une branche du ML qui regroupe une série de modèles basés sur les NNs.

Les NNs sont des modèles où des cellules de calcul, les neurones, sont agencées en couches. Les connexions complexes entre les couches permettent au DL de se démarquer des algorithmes classiques de ML : ces connexions permettent au NNs de modéliser des relations hautement non linéaires, par leur quantité et la présence de fonctions d'activation, non linéaires elles-mêmes. Lesdites cellules consistent en des imitations mathématiques de neurones vivants. En effet, chaque cellule comporte des entrées  $x_i$ , lesquelles sont assemblées linéairement avec des poids  $h_{ji}$  propres à chaque connexion. C'est l'analogie biologique de la réception d'un influx nerveux par les dendrites d'un neurone. Puis, ces combinaisons subissent une transformation non linéaire  $f$  appelée fonction d'activation, imitant le potentiel d'activation biologique, pour, enfin, produire la sortie  $y_j$ . Ce processus est illustré à la figure 2.1. Mathématiquement, la sortie se résume donc à

$$y_j = f\left(\sum_i h_{ji}x_i\right). \quad (2.1)$$

Le choix de la fonction d'activation est dicté par le domaine attendu de la sortie. Des choix courants pour la fonction d'activation  $f(x'_j)$  sont la fonction sigmoïde  $\sigma(x'_j) = 1/(1 + e^{-x'_j})$ , la fonction  $\tanh(x'_j) = (e^{x'_j} - e^{-x'_j})/(e^{x'_j} + e^{-x'_j})$  et la fonction *softmax*  $e^{x'_j}/\sum_k e^{x'_k}$ . Cette dernière est couramment uti-

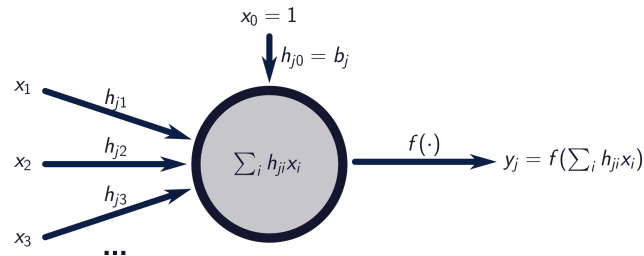


FIGURE 2.1 : Neurone artificiel

**Le neurone artificiel  $j$  combine linéairement ses entrées  $x_i$  avec des poids  $h_{ji}$ . Ce résultat est passé à une fonction d'activation non linéaire  $f$  pour produire la sortie  $y_j$ . Le biais  $h_{j0} = b_j$  est un facteur de translation indépendant des entrées.**

lisée pour les sorties probabilistes, puisque la somme des composantes du vecteur de sortie est 1 et que chacune des sorties a pour domaine  $[0, 1]$ . Les sorties présentent alors toutes les propriétés d'une distribution de probabilité valide. Ce sont les fonctions d'activation qui introduisent la non-linéarité dans les NNs : sans elles, les connexions ne forment que des réseaux de combinaisons linéaires. Typiquement, un biais est introduit aux sorties du NN par l'entremise de  $x_0 = 1$  et  $h_{j0} = b_j$ , le biais. Celui-ci permet d'obtenir des solutions non triviales en introduisant un décalage par rapport à l'origine, ce qui est nécessaire à la généralité. Les poids  $h_{ji}$  sont les éléments entraînés d'un neurone et, par extension, des NNs : ce sont les paramètres du modèle.

Lorsque plusieurs neurones sont connectés de façon à effectuer un calcul avec les mêmes entrées et les mêmes sorties, cet ensemble de neurones est appelé une couche. Un NN est une combinaison de ces couches. Il est qualifié de profond lorsqu'au moins deux couches séparent les entrées de la couche de sorties, par convention. Ces couches séparant la couche de sorties et les entrées sont appelées couches cachées. Un exemple de NN est illustré à la figure 2.2.

### 2.1.1 Entraînement

Pour obtenir un outil prédictif à base de DL, l'utilisateur doit entraîner un NN à l'aide de paires de données et de réponses — préférablement une quantité massive de ces paires — qui illustrent bien la relation à établir. Lorsque les modèles sont simples ou lorsque les jeux de données sont petits, l'entièreté d'un jeu de données peut être présentée d'un coup à l'algorithme. La plupart du temps, ce n'est pas possible et les données doivent être présentées itérativement par paquets (*batches*). Les NNs peuvent bénéficier du fait d'observer plusieurs fois le même jeu de données.

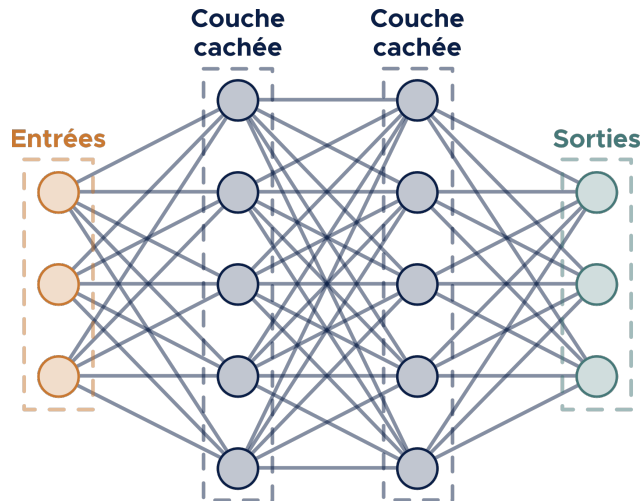


FIGURE 2.2 : Réseau de neurones profond

Chaque neurone des couches cachées et de la couche de sorties consiste en fait en un neurone comme celui illustré à la figure 2.1.

Montrer à un NN la totalité d'un jeu de données constitue une époque (*epoch*) et, alors, utiliser une grande quantité d'époques avec un taux d'apprentissage (*learning rate*) bas aide un algorithme à avoir un aperçu plus global du problème pendant son entraînement. Les jeux de données sont généralement découpés en trois sous-ensembles mutuellement exclusifs : les jeux de données d'entraînement sont utilisés pour l'optimisation du NN, alors que les jeux de données de validation et de test servent à évaluer sa performance.

En DL, la performance d'un réseau est quantifiée par une fonction coût  $L$  (*loss*) et l'entraînement vise à minimiser celle-ci, et ce, par l'algorithme du gradient (*gradient descent*). Des fonctions coût courantes sont

$$\text{L'erreur moyenne quadratique :} \quad \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (2.2)$$

$$\text{L'erreur moyenne absolue :} \quad \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (2.3)$$

$$\text{L'entropie croisée :} \quad - \sum_{i=1}^n \sum_{j=1}^m y_{ij} \log(\hat{y}_{ij}), \quad (2.4)$$

où  $y$  dénote les étiquettes,  
 $\hat{y}$  dénote les estimations,  
 $n$  est la quantité d'échantillons,  
 $m$  est la quantité de classes.

Les corrections individuelles des poids  $h_{ji}$  sont calculées par rétropropagation du gradient

(*backpropagation*). Le développement subséquent est inspiré de Goodfellow et al. (2016). Soit l'erreur  $L$  observée entre la sortie prédite et celle attendue pour un paquet de données. La direction dans laquelle apporter une correction  $\Delta h_{ji}^{(k)}$  aux poids  $h_{ji}^{(k)}$  d'une couche  $k$  pour diminuer  $L$  est la dérivée  $-\frac{\partial L}{\partial h_{ji}^{(k)}}$ , puisque  $\frac{\partial L}{\partial h_{ji}^{(k)}} > 0$  ( $h_{ji}^{(k)}$  augmente aussi  $L$ ) diminuera le poids par le biais du signe négatif, et vice-versa. La rétropropagation du gradient consiste à effectuer le calcul de la dérivée précédente pour chacun des poids, ce qui est non trivial à calculer, puisque  $L$  est relié aux poids des couches cachées par le biais des couches qui les séparent. Pour y parvenir, il faut tirer parti de la règle de la dérivée en chaîne, laquelle stipule que

$$\frac{\partial f(\mathbf{g}(\mathbf{x}))}{\partial x_i} = \sum_j \frac{\partial f}{\partial g_j} \frac{\partial g_j}{\partial x_i}. \quad (2.5)$$

Appliquer cette règle récursivement permet d'obtenir la correction à apporter à chacun des poids, quelle que soit la couche. Une fois le gradient rétropropagé, il existe de nombreuses façons d'appliquer les corrections en utilisant le gradient. La plus simple constitue l'algorithme du gradient, mais notons aussi l'algorithme du gradient stochastique (SGD, de *stochastic gradient descent*; Kiefer et al. 1952) et l'*Adaptive Moment Estimation* (Adam; Kingma et al. 2017), lesquelles sont toutes deux des variantes de l'algorithme du gradient utilisées massivement en DL. L'algorithme du gradient consiste simplement à calculer les poids corrigés  $h'$  en ajoutant la dérivée calculée plus tôt pondérée par un facteur  $\lambda > 0$  appelé taux d'apprentissage, c'est-à-dire

$$h' = h - \lambda \frac{\partial L}{\partial h}, \quad (2.6)$$

où les indices de  $h_{ji}^{(k)}$  ont été omis par souci de clarté. Le principe de fonctionnement du SGD est de n'utiliser qu'un échantillon par paquet pour le calcul de  $\frac{\partial L}{\partial h}$  plutôt que d'utiliser l'entièreté du paquet. En effet, la sommation de la dérivée à travers tous les échantillons d'un paquet est coûteuse et une seule valeur de gradient est suffisante pour permettre la convergence (Kiefer et al., 1952). Adam met à profit la moyenne mobile de  $\frac{\partial L}{\partial h}$  et de son deuxième moment pour conférer une inertie

aux corrections. Sous l'optimiseur Adam, la correction devient

$$h' = h - \lambda \frac{m'}{\sqrt{v'} + \epsilon} \quad (2.7)$$

$$m' = \frac{\beta_1 m + (1 - \beta_1) \frac{\partial L}{\partial h}}{1 - \beta_1^t} \quad (2.8)$$

$$v' = \frac{\beta_2 v + (1 - \beta_2) \left(\frac{\partial L}{\partial h}\right)^2}{1 - \beta_2^t} \quad (2.9)$$

où  $m$  est une moyenne mobile du gradient,  
 $v$  est une moyenne mobile du deuxième moment du gradient,  
 $\beta_1$  et  $\beta_2$  sont des constantes d'oubli,  
 $\epsilon \ll 1$  prévient la division par 0.  
(Kingma et al., 2017)

L'art du DL réside dans le choix des composantes d'un NN et leur agencement. Par exemple, pour analyser une image, il sera préférable d'utiliser un CNN, un sujet couvert à la sous-section suivante, puisque l'organisation des couches en filtres bidimensionnels est une géométrie idéale pour extraire des patrons (*features*) dans une image.

### 2.1.2 Réseaux de neurones convolutifs

Les NNs convolutifs (CNN, de *convolutional neural network*) sont des NNs dans lesquels les connexions sont organisées de façon à engendrer des convolutions de l'entrée. Cela a pour effet de forcer l'hypothèse que les entrées sont spatialement organisées. Les poids sont alors interprétés comme les valeurs de filtres qui seraient glissés sur l'entrée. La figure 2.3 illustre la géométrie d'une couche convolutive utilisée dans un CNN.

Au terme de l'entraînement, les filtres consistent en des patrons présents dans les images convoluées. L'image initiale est généralement une combinaison de traits, de couleurs, d'ombres, et une couche convolutive apprendra à reconnaître ces caractéristiques. La convolution d'une image avec un des filtres donne une carte d'activation, c'est-à-dire une carte de l'intensité à laquelle un filtre correspond à la zone étudiée. Typiquement, les couches convolutives n'apparaissent pas seules, puisqu'agencer des couches convolutives en série permet de composer ces patrons primaires en des caractéristiques de haut niveau.

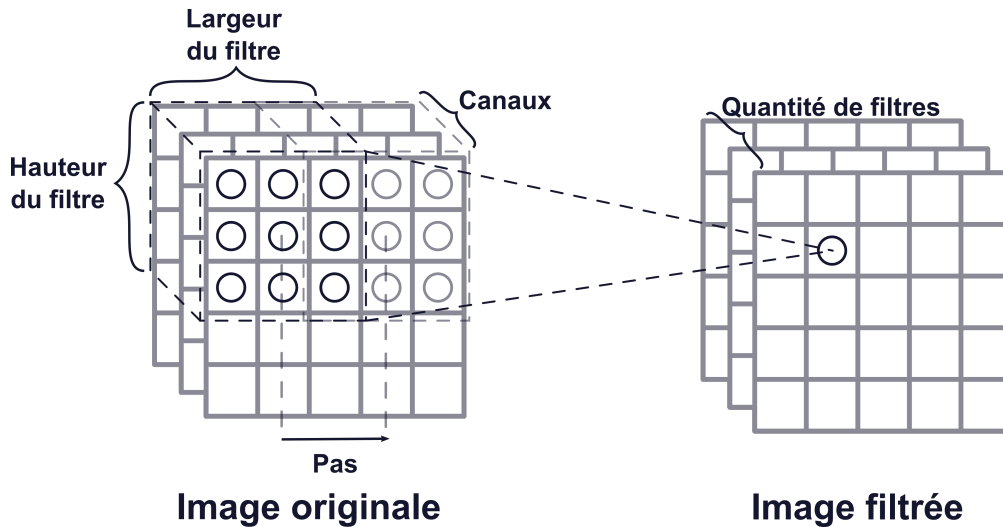


FIGURE 2.3 : Couche convolutive d'un réseau de neurones convolutif

Une couche convolutive consiste en une série de filtres qui sont tradés sur l'image à analyser. Les valeurs de chacun des pixels sont multipliées par celles des poids du filtre, puis sommées, pour former l'une des sorties à l'emplacement correspondant de l'image filtrée. Le tout consiste une convolution. On assume par défaut que les filtres combinent l'information dans tous les canaux à la fois et, donc, que les filtres ont une dimension supplémentaire (par exemple trois dimensions, dans cet exemple). À chaque filtre différent est associé une image filtrée différente, ce qui crée effectivement une dimension de canaux (annotée « quantité de filtres ») dans l'image de sortie, remplaçant celle présente initialement. Un pas optionnel peut être ajouté à la convolution, réduisant la taille de l'image de sortie. L'image d'entrée est typiquement élargie (*padded*) pour préserver les dimensions initiales.

Ajouter un pas à la convolution permet de réduire les dimensions de l'image de sortie, puisque la quantité de pas est alors moins grande que la dimension d'entrée. Toutefois, une autre façon plus courante de sous-échantillonner une image est d'ajouter des couches de mise en commun (*pooling*) au CNN. Ces couches consistent en des filtres qui ne peuvent être entraînés. Des choix courants de couches de mise en commun comportent des filtres qui sélectionnent le maximum sur la fenêtre ou qui calculent la moyenne. Le sous-échantillonnage introduit graduellement de l'invariance spatiale, permettant au NN de transformer l'information spatiale en caractéristiques haut niveau pertinentes (Waldeland et al., 2018).

### 2.1.3 Réseaux de neurones récurrents

Les NNs récurrents (RNN, de *recurrent neural network*) ont un potentiel immense en sismique (Alfarraj et al., 2019). En effet, par construction, les RNNs modélisent des relations de causalité (Rumelhart et al., 1986; Lipton et al., 2015), alors que les réseaux à propagation avant ont pour hypothèse intrinsèque que les points sont indépendants (Alfarraj et al., 2019). Les RNNs sont conçus

de telle façon que la sortie de chaque neurone sert d'entrée au neurone de l'itération suivante. Le processus de calcul récurrent est schématisé à la figure 2.4. Cela fait des RNNs une solution appropriée pour les problèmes à caractère temporel, notamment la propagation d'une onde sismique, puisque la notion de cause à effet est naturellement prise en compte par cette dépendance en chaîne. En effet, les RNNs sont des solutions reconnues pour les problèmes temporels (van den Oord et al., 2016a), mais leur utilité peut être étendue à tout problème comportant de la causalité, qu'elle soit temporelle ou spatiale. Le terme RNN décrit à la fois le RNN classique décrit à la figure 2.4 et la classe d'architectures récurrentes présentées ci-après, par abus de langage.

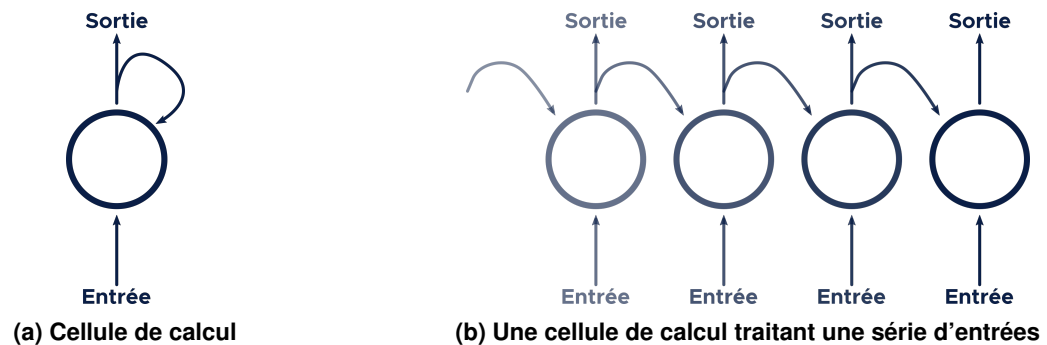


FIGURE 2.4 : Réseau de neurones récurrent

Dans un RNN classique, les cellules de calcul composent l'entrée actuelle et la sortie précédente en une sortie obtenue de la même façon que pour le neurone artificiel de la figure 2.1. En réalité, l'application d'un RNN consiste à appliquer de façon récurrente la cellule de calcul — il n'y en a qu'une, utilisée de façon récurrente — sur la série temporelle d'entrée.

Le RNN classique n'est en mesure de modéliser que des processus de Markov, puisque la seule dépendance qu'il peut modéliser est celle entre deux itérations successives. Ce RNN ne peut modéliser des relations à long terme que par un encodage implicite dans le vecteur de sorties, mais cela implique que l'information est fortement atténuée sur le long terme en faveur des dépendances à court terme.

Les réseaux à mémoire à court et à long terme (LSTM, de *long-short term memory*, Hochreiter et al. 1997) sont une façon de vaincre l'atténuation du flux d'information intrinsèque au RNN classique. Ce problème et ses implications sont décrits en détail par Bengio et al. (1994). Le LSTM comporte un vecteur d'état  $c$  qui lui confère une mémoire, laquelle est modifiée au fil des entrées.

Une cellule LSTM s'écrit

$$f_t = \sigma(H_f[x_t, y_{t-1}] + b_f) \quad (2.10)$$

$$i_t = \sigma(H_i[x_t, y_{t-1}] + b_i) \quad (2.11)$$

$$o_t = \sigma(H_o[x_t, y_{t-1}] + b_o) \quad (2.12)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(H_c[x_t, y_{t-1}] + b_c) \quad (2.13)$$

$$y_t = o_t \odot \tanh(c_t) \quad (2.14)$$

où  $t$  est l'indice de l'itération courante,  
 $x$  est un vecteur d'entrées,  
 $y$  est un vecteur de sorties,  
 $H$  est une matrice de poids,  
 $b$  est un biais,  
 $f$  est un vecteur d'activation de la porte d'oubli (*forget gate*),  
 $i$  est un vecteur d'activation de la porte d'entrées (*input gate*),  
 $o$  est un vecteur d'activation de la porte de sorties (*output gate*),  
 $c$  est un vecteur d'état de la cellule,  
 $\odot$  est l'opérateur de multiplication élément par élément.  
(Hochreiter et al., 1997; Gers et al., 2000; Olah, 2015)

Une description détaillée et illustrée d'une cellule LSTM et de ses composantes est donnée par Olah (2015).

WaveNet (van den Oord et al., 2016a) utilise des convolutions dilatées pour augmenter le champ réceptif associé à une sortie. Le principe des convolutions dilatées est illustré à la figure 2.5. Les contributions associées à chacune des entrées sont *a priori* les mêmes qu'importe l'échelle. En effet, si les convolutions dilatées sont bien agencées, comme dans la figure 2.5b, la quantité de liens qui associent une entrée à une sortie est la même peu importe l'entrée considérée, au contraire de la figure 2.5a. Cela a pour effet de stabiliser l'apprentissage de relations à long terme, en les mettant sur le même pied que les relations à court terme.

PixelRNN et PixelCNN (van den Oord et al., 2016b) sont des améliorations à WaveNet. PixelCNN, plus particulièrement, innove en utilisant des couches convolutives pour implémenter la récurrence. Une des difficultés des RNNs est le fait que l'entraînement peut difficilement être parallélisé, puisque les sorties dépendent successivement de toutes les sorties précédentes. PixelCNN contourne ce problème en utilisant la convolution pour obtenir un comportement de dépendance.



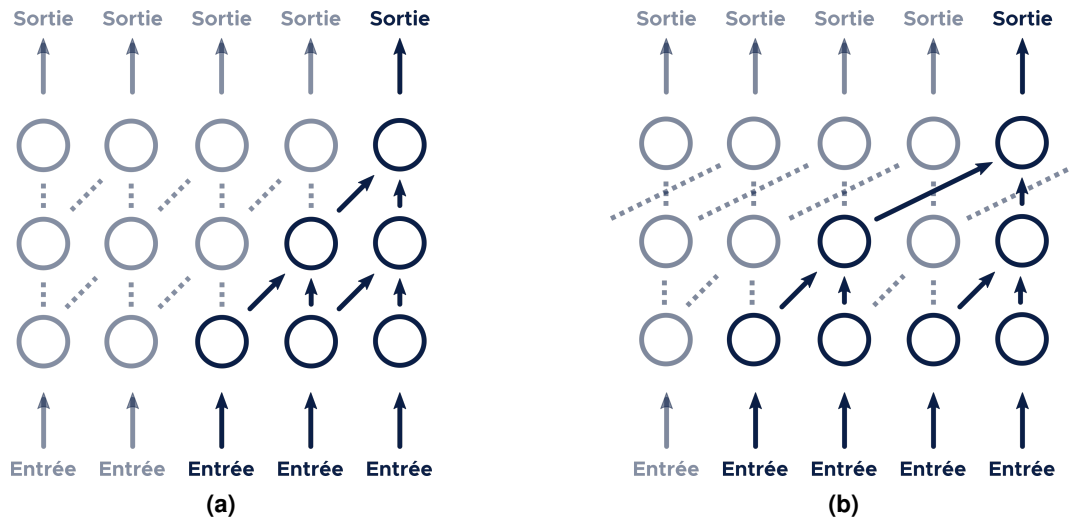


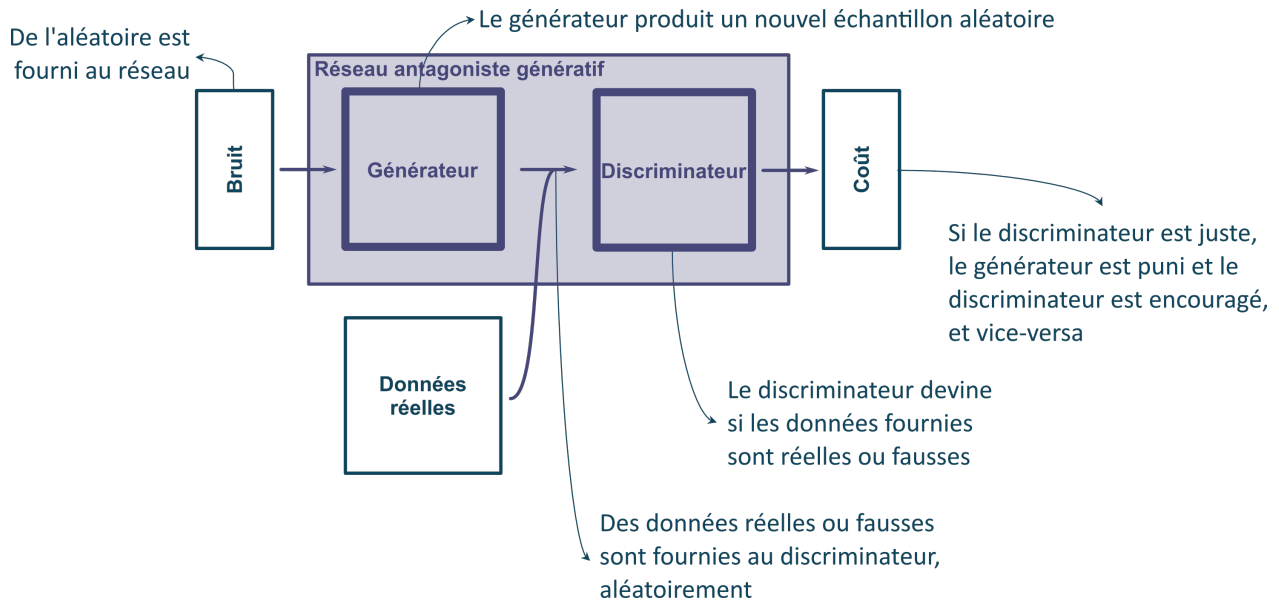
FIGURE 2.5 : Élargissement du champ réceptif avec WaveNet

L'utilisation de convolutions dilatées, c'est-à-dire des convolutions dont les filtres comportent des trous, permet d'élargir la quantité d'entrées considérées dans une prédiction.

#### 2.1.4 Réseaux antagonistes génératifs

Les GANs sont des NNs modernes permettant de générer des échantillons ressemblant à ceux des jeux de données d'entraînement. Pour l'estimation de modèles de vitesse, les GANs peuvent constituer une fonction coût généralisée (Lin et al., 2019; Zhang et al., 2020) ou un générateur de modèles synthétiques (Araya-Polo et al., 2019; Richardson, 2018a; Mosser et al., 2020a). Les GANs sont composés en deux parties : l'une appelée le « générateur » et, la seconde, le « discriminateur ». Leur entraînement repose sur le fait que le discriminateur peut apprendre à classifier des échantillons réels et faux pour quantifier le réalisme. Une fois sur deux, un échantillon réel est fourni au discriminateur; l'autre fois, c'est le produit du générateur. Le générateur, quant à lui, est en mesure de générer des échantillons de plus en plus réalistes en trompant le discriminateur : effectivement, si le discriminateur n'est pas en mesure de distinguer les échantillons réels des faux, c'est que le générateur est parvenu à générer des contrefaçons convaincantes. La dynamique résultante est analogue à un jeu dans lequel le discriminateur est un policier et le générateur est un faussaire. L'architecture d'un GAN est résumée à la figure 2.6.

La fonction coût associée à ce jeu est entièrement dictée par la capacité du discriminateur à distinguer les contrefaçons : le gradient propagé au discriminateur minimise sa propre erreur, tandis que



**FIGURE 2.6 : Réseau antagoniste génératif**

celui propagé au générateur tente de la maximiser. Mathématiquement, cette dynamique s'écrit

$$\min_G \max_D \{L(D, G)\} \quad (2.15)$$

$$L(D, G) = \mathbb{E}_{\mathbf{x}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}}[\log (1 - D(G(\mathbf{z})))] \quad (2.16)$$

où

- $L$  est la fonction coût,
- $D$  est le discriminateur,
- $G$  est le générateur,
- $\mathbf{x}$  est un échantillon réel,
- $\mathbf{z}$  consiste en du bruit,
- $D(\cdot)$  est une probabilité que l'échantillon soit réel, telle que prédite par  $D$ ,
- $G(\mathbf{z})$  est un échantillon contrefait aléatoire.

(Goodfellow et al., 2014)

Cette dynamique est intrinsèquement instable. Obtenir un équilibre entre la performance du générateur et celle du discriminateur sous la dynamique de l'équation 2.15 est difficile et, en pratique, de nombreux problèmes surviennent pendant l'entraînement d'un GAN. L'objectif de chacun des réseaux est contraire et la solution recherchée est en fait un équilibre où chacun des sous-réseaux est au maximum de sa performance, un point selle (Goodfellow et al., 2014, 2016; Gui et al., 2020).

### 2.1.5 Apprentissage par transfert et adaptation de domaine

Supposons un problème de DL plutôt bien contraint, car relativement simple et comportant des données massives et redondantes, notamment. Il peut s'agir, par exemple, de vision par ordinateur sur des jeux de données massifs *open source* populaires, comme ImageNet (Deng et al., 2009) ou COCO (Lin et al., 2015). Il pourrait aussi s'agir d'un problème basé sur des données synthétiques plus simples à analyser. Ce jeu de données « source » est composé de données  $X_s \in \mathcal{X}_s$  et d'étiquettes  $Y_s \in \mathcal{Y}_s$  constituant un domaine  $\mathcal{D}_s = (\mathcal{X}_s, p_s(X_s))$  et une tâche  $\mathcal{T}_s = (\mathcal{Y}_s, p_s(Y_s|X_s))$ , laquelle consiste à relier les données aux bonnes étiquettes. Puisque le domaine source est bien échantillonné, un NN peut être entraîné sur ce domaine pour obtenir un outil prédictif  $\text{NN}_s : \mathcal{X}_s \rightarrow \mathcal{Y}_s$  performant.

Le TL est une stratégie pour entraîner plus facilement un NN à effectuer une tâche  $\mathcal{T}_t$  dans un domaine « cible »  $\mathcal{D}_t$ . Le TL consiste à exploiter l'information apprise dans le domaine source pour améliorer la généralisation dans le domaine cible (Goodfellow et al., 2016), généralement en initialisant les poids de certaines ou de toutes les couches du NN utilisé dans le domaine cible avec les poids appris dans le domaine source. C'est ce qui est illustré à la figure 2.7. Cela permet de paramétriser le NN près de la solution recherchée, si le minimum global était atteint dans le problème analogue (Bengio et al., 2011). L'effet est de réduire les temps de convergence, de réduire la quantité nécessaire de données et d'éviter efficacement les minima locaux de la fonction coût (Bengio et al., 2011; Tan et al., 2018). Le TL est applicable lorsqu'il y a un écart de domaine (*domain gap*), tel que  $\mathcal{D}^s \neq \mathcal{D}^t$ , mais aussi lorsque la tâche diffère ( $\mathcal{T}^s \neq \mathcal{T}^t$ ). Un cas particulier de TL est l'adaptation de domaine, qui vise à combler l'écart de domaine  $\mathcal{D}^s \neq \mathcal{D}^t$ , mais pour laquelle  $\mathcal{T}^s = \mathcal{T}^t$ . Par exemple, en sismique, il existe un écart de domaine entre les données synthétiques couramment utilisées pour entraîner les NNs et les données de terrain auxquelles on applique ces NNs : même si la tâche demeure celle d'estimation de modèles de vitesse, les données de terrain comportent une signature propre. Un exemple d'adaptation de domaine illustré à l'aide de cartes d'activation est donné par Zhao et al. (2021).

La transférabilité est normalement meilleure pour des domaines sources bien contraints et dont la ressemblance au domaine cible est meilleure, mais on observe tout de même des améliora-

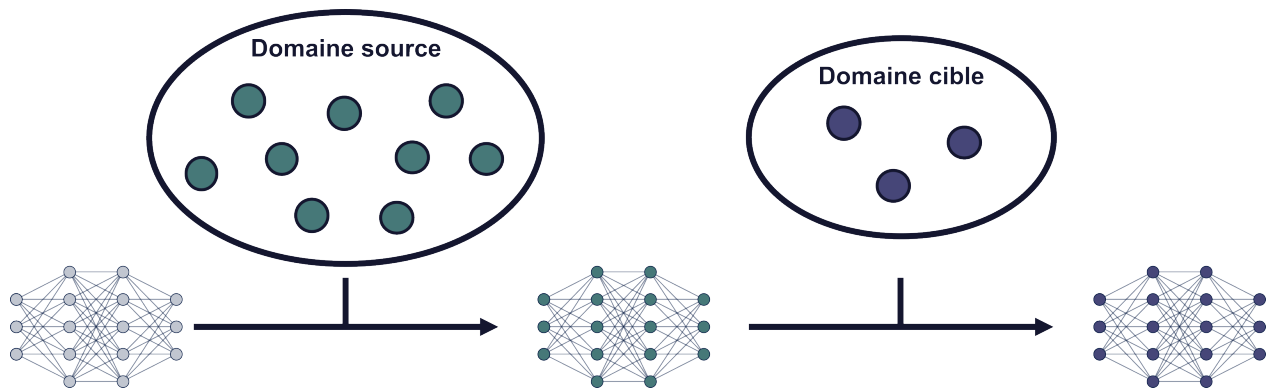


FIGURE 2.7 : Apprentissage par transfert

tions pour des domaines source et cible dissemblables. Cela est dû au fait que presque tous les problèmes auxquels s'attaque l'AI ont généralement de vagues similarités (Caruana, 1995). Par exemple, à petite échelle, le problème d'analyse d'images est toujours le même : produire des filtres qui permettent de détecter des interfaces ou des traits. Ces filtres se ressembleront, peu importe le problème. Autrement dit, les représentations intermédiaires de problèmes qui ne se ressemblent pas *a priori* bénéficient pourtant de redondance, statistiquement parlant (Bengio et al., 2011).

### 2.1.6 Apprentissage actif

La performance du DL est essentiellement garantie par la disponibilité de jeux de données massifs et publics (Ren et al., 2021). Or, dans des sphères de la société où les données massives sont inexistantes, le DL tarde à être adopté, puisqu'étiqueter avec précision plusieurs milliers, voire millions d'exemples requiert une quantité de travail disproportionnée (Ren et al., 2021). Même s'il est possible de mettre à profit les jeux de données massifs préexistants par TL, les mettre en rapport avec le problème courant et étendre efficacement les apprentissages d'un NN à un nouveau domaine est encore un défi (Wang et al., 2018a). Il faut souvent tout de même recourir à un étiquetage dans le domaine cible. En d'autres termes, l'étiquetage est un frein à l'adoption du DL.

L'AL est une branche de ML qui permet d'entraîner un algorithme à l'aide d'échantillons étiquetés en temps réel et choisis dynamiquement par l'algorithme lui-même. L'AL permet de démarrer des projets de ML rapidement à partir de prototypes fonctionnels qui sont améliorés dynamique-

ment à mesure qu'ils sont utilisés. Pour accélérer l'étiquetage, l'algorithme doit maximiser l'apport de chacun des exemples à l'entraînement en sélectionnant les exemples pour lesquels les prédictions sont les plus incertaines ou ceux représentant des cas limites, c'est-à-dire en échantillonnant le domaine d'entraînement de façon optimale. Effectuer du DL par AL permet donc de réduire les besoins en données massives du DL tout en préservant sa performance (Ren et al., 2021). Les exemples à étiqueter peuvent être choisis depuis un jeu de données sans étiquettes ou même être générés de zéro. Les plus récents développements en DL actif concernent la génération d'exemples d'entraînement. Avec des NNs génératifs, il est possible de créer les exemples d'entraînement synthétiquement : les GFlowNets (Bengio et al., 2021, 2022), notamment, ont été conçus pour générer des exemples diversifiés et informatifs en modélisant le processus de génération comme un flux de probabilités.

L'AL n'a jamais encore été appliqué aux méthodes sismiques, mais, considérant que la création de jeux de données suffisamment réalistes et diversifiés est encore un défi en sismique, l'AL permettrait de contourner certains problèmes méthodologiques. Les méthodes de DL en sismique sont généralement applicables dans des contextes très contraints et les apprentissages peuvent difficilement être extrapolés à des géologies ou des géométries d'acquisition différentes, comme je le montre au chapitre 6. En sismique, l'essentiel de l'information nécessaire pour résoudre l'équation d'onde est contenue dans l'équation elle-même. Un NN pourrait donc être entraîné de façon autosupervisée en permettant au NN de générer lui-même les exemples d'entraînement par AL.

## **2.2 Méthodes sismiques**

Les méthodes sismiques sont des approches géophysiques qui permettent d'estimer la distribution des propriétés physiques du sol à l'aide de mesures de vibration depuis la surface. Les méthodes sismiques sont donc analogues à une échographie : le volume à étudier est mis en oscillation depuis la surface et ces ondes se propagent dans le milieu, pour enfin se réfléchir sur les hétérogénéités rencontrées. Enfin, les données acquises sont analysées pour reconstruire les hétérogénéités et trouver les dépôts recherchés par des arguments géologiques. On peut reconstruire par une variété de méthodes la composition du milieu en termes de ses propriétés physiques ou de

ses interfaces. Un des processus de traitement des données sismiques se nomme l'« inversion ». En inversion sismique, les mesures captées à la surface sont utilisées pour reconstruire un modèle des propriétés physiques du sol étudié. Le processus de traitement des données sismiques n'est souvent pas effectué jusqu'à l'inversion : l'interprétation des données sismiques en un modèle qualitatif ou semi-quantitatif peut suffire, puisqu'il permet d'aiguiller le travail de géologues (Giroux et al., 2019). Le processus de traitement est alors appelé « imagerie ».

### 2.2.1 Équation d'onde

La dynamique de la propagation des ondes dans le sol est dictée par les propriétés mécaniques du milieu. Les développements de cette sous-section sont inspirés de Sheriff et al. (1995). La mécanique des solides peut être décrite en termes de contraintes  $\sigma_{ij}$ , c'est-à-dire les forces par unité de surface agissant sur des éléments de volume infinitésimaux. Les 9 éléments du tenseur de contraintes décrivent chacune des forces agissant dans une direction  $i \in \{x, y, z\}$  sur les faces  $j \in \{x, y, z\}$  des éléments de volume. Les contraintes ont pour effet d'entraîner des déplacements  $u_i$ , des rotations  $\theta_i$  et des déformations  $\epsilon_{ij}$  des éléments de volume. Dans la limite des petites déformations, les déformations sont données par

$$\epsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.17)$$

où  $(x_1, x_2, x_3) = (x, y, z)$  sont les dimensions spatiales usuelles.

De même, la rotation  $\theta_i$  du solide autour de l'axe  $x_i$  est donnée par la soustraction des changements d'angles  $\delta_i = \frac{\partial u_j}{\partial x_i}$ , c'est-à-dire

$$\theta_i = \frac{1}{2} \sum_{j,k} \epsilon_{ijk} \frac{\partial u_k}{\partial x_j}, \quad (2.18)$$

où  $\epsilon_{ijk}$  est le tenseur antisymétrique de Lévi-Civita.

En combinant la deuxième loi de Newton et la loi de Hooke, qui relie les contraintes et les déformations, on déduit l'équation décrivant la propagation des ondes dans un milieu solide, l'équation

d'onde élastique. La loi de Hooke généralisée consiste en une approximation linéaire de la relation reliant  $\sigma_{ij}$  et  $\epsilon_{ij}$ , valide pour de petites contraintes. Pour un milieu isotrope, elle s'écrit

$$\sigma_{ij} = \begin{cases} \lambda\Delta + 2\mu\epsilon_{ij} & \text{si } i = j \\ \mu\epsilon_{ij} & \text{sinon} \end{cases}, \quad (2.19)$$

où  $\Delta = \sum_i \epsilon_{ii}$  est le coefficient de dilatation,  
 $\lambda$  et  $\mu$  sont les constantes de Lamé.

Les constantes de Lamé sont des propriétés physiques qui peuvent être reliées aux vitesses sismiques. Ce sont des paramètres propres à chaque point de l'espace et ce sont elles — ou, de façon équivalente, les vitesses — qui sont estimées par inversion et qu'on cherchera à estimer par DL à travers cette thèse. La deuxième loi de Newton s'écrit

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \sum_j \frac{\partial \sigma_{ij}}{\partial x_j} \quad (2.20)$$

où  $\rho$  est la densité du matériau.

En utilisant la loi de Hooke, on obtient l'équation d'onde, c'est-à-dire

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = (\lambda + \mu) \nabla \Delta + \mu \nabla^2 \mathbf{u}, \quad (2.21)$$

où  $\mathbf{u} = \sum_i u_i \hat{\mathbf{e}}_i$  est le vecteur des déplacements.

Généralement, une équation d'onde peut s'écrire

$$\frac{1}{V^2} \frac{\partial^2 \phi}{\partial t^2} = \nabla^2 \phi, \quad (2.22)$$

où  $V$  est la vitesse de l'onde, aussi appelée vitesse d'intervalle,  
 $\phi$  est la fonction décrivant l'onde.

Or, en calculant la divergence et le rotationnel de l'équation 2.21, on obtient deux équations d'onde décrivant, respectivement, les ondes de compression et les ondes de cisaillement,

c'est-à-dire

$$\frac{1}{V_P^2} \frac{\partial^2 \Delta}{\partial t^2} = \nabla^2 \Delta, \quad (2.23)$$

$$\frac{1}{V_S^2} \frac{\partial^2 \Theta}{\partial t^2} = \nabla^2 \Theta, \quad (2.24)$$

où  $\Theta = \sum_i \theta_i \hat{e}_i$  est le vecteur des angles de rotation,  
 $V_P = \sqrt{(\lambda + 2\mu)/\rho}$  est la vitesse des ondes de compression dans le milieu,  
 $V_S = \sqrt{\mu/\rho}$  est la vitesse des ondes de cisaillement dans le milieu.

L'existence de deux équations d'ondes décrivant la mécanique des solides implique que les ondes peuvent se propager dans deux régimes, comme l'illustre la figure 2.8 : en compression, c'est-à-dire qu'elles oscillent dans le sens de leur propagation, et en cisaillement, c'est-à-dire qu'elles oscillent dans l'une ou l'autre des directions perpendiculaires à celle de leur propagation. Les ondes de cisaillement sont absentes dans un fluide, qu'il s'agisse d'air ou d'eau. Les ondes de compression peuvent être converties en ondes de cisaillement, et vice-versa, lorsqu'elles rencontrent une interface à angle d'incidence non nul, tel qu'illustré à la figure 2.9. Dans cette thèse, j'utilise des milieux marins et l'approximation acoustique, et ce, pour faciliter l'interprétation des résultats. Ainsi,  $V$  réfère systématiquement à  $V_P$ .

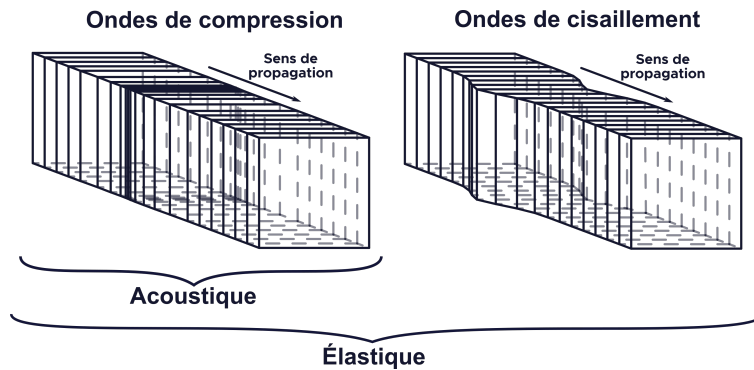
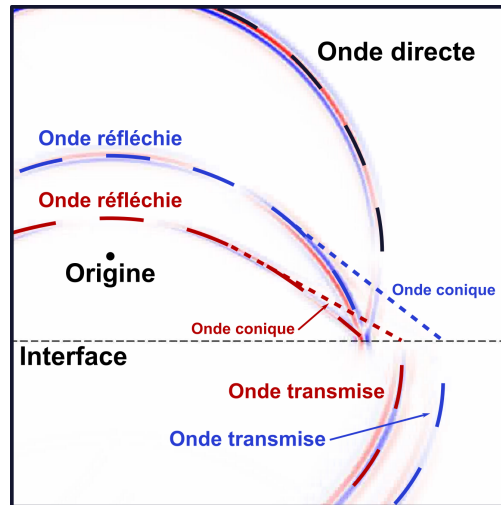


FIGURE 2.8 : Régimes de propagation des ondes

Dans un solide, les ondes peuvent se propager sous forme d'ondes de compression ou d'ondes de cisaillement. Lorsque seules les ondes de compression sont considérées, le problème est qualifié d'acoustique, sinon, il est qualifié d'élastique. Les ondes de cisaillement sont inexistantes dans un fluide.





Légende : **Onde de compression**  
**Onde de cisaillement**

FIGURE 2.9 : Conversion des ondes (modifié de Giroux, 2019)

Les ondes de compression créées par une source explosive sont converties à la rencontre d'une interface dans un milieu isotrope élastique. Les vitesses sismiques sont plus grandes dans le milieu inférieur. Les ondes coniques sont causées par des réfractions à angle critique.

## 2.2.2 Données sismiques

En sismique réflexion, les données expérimentales consistent en des vibrations mesurées par des géophones ou des hydrophones. Les vibrations mesurées sont de nature artificielle : elles consistent généralement en des impulsions forcées à la surface grâce à des masses heurtant le sol, des explosifs ou un fusil à air comprimé en milieu marin. Ces vibrations se propagent dans le sol, rencontrent des interfaces entre des milieux et se réfléchissent ou se réfractent. Ces ondes reviennent enfin à la surface pour être captées par les géophones. Elles peuvent aussi continuer à se propager en profondeur ou latéralement pour ne jamais être captées.

Les méthodologies d'acquisition sismique sont variées. Généralement, un levé sismique en surface consiste à disposer quelques dizaines de géophones au sol et à activer la source à une distance raisonnable des géophones, de façon à éviter que le signal ne sature. La source et les géophones sont ensuite redistribués à nouveau, de façon à améliorer la couverture du levé. Dans un levé sismique 2D, les géophones sont typiquement alignés avec la source, comme l'illustre la figure 2.10. En somme, les données acquises consistent en un tenseur de traces sismiques  $n_t \times n_{\text{géophones}} \times n_{\text{tirs}}$ , lequel est typiquement enregistré sous le format SEG-Y (Committee, 2017). Pendant le traitement de données sismiques, les données sont généralement réarrangées en point

milieu commun (CMP, de *common mid-point*). Un regroupement CMP est une section sismique  $n_t \times n_h$  contenant toutes les traces pour lesquelles les points milieux entre la source et le receveur concordent, comme l'illustre la figure 2.10. Les déports  $h$  — les demi-distances entre les sources et les receveurs — de la dimension  $n_h$  sont les mêmes pour tous les  $n_{\text{CMP}}$  regroupements CMP. Des exemples de sections sismiques  $n_t \times n_{\text{géophones}}$  organisées en tirs sont montrés à la figure 2.11.

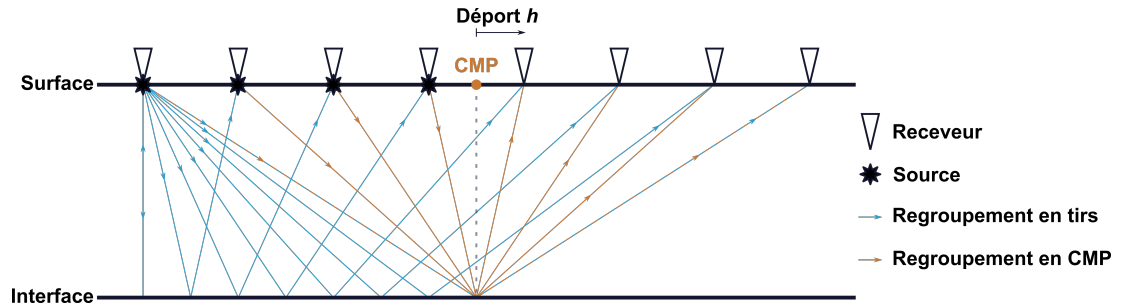


FIGURE 2.10 : Géométrie d'acquisition (inspiré de Laakso, 2019)

L'équipement d'acquisition sismique comprend des sources et des récepteurs, idéalement disposés à intervalle constant. La position des sources n'a pas à concorder avec la position des récepteurs et l'intervalle entre les sources diffère généralement de celui entre les récepteurs. Les données sismiques sont généralement triées par tirs ou par CMP. Le déport  $h$  est la demi-distance entre la source et le récepteur.

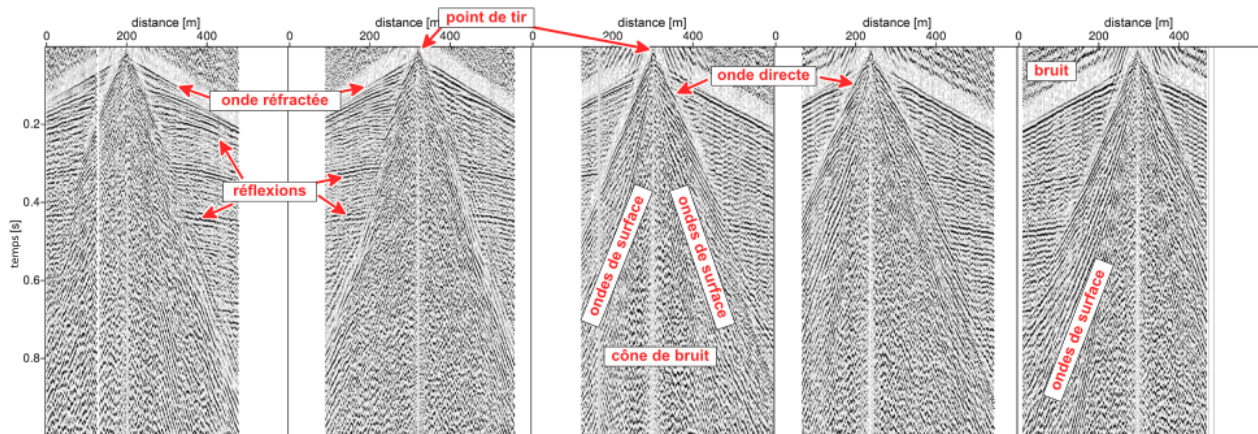


FIGURE 2.11 : Données sismiques regroupées en tirs (Burschil et al., 2015, distribué sous CC BY 3.0)

Les données sismiques montrées sont des enregistrements sismiques réels effectués en milieu terrestre. L'amplitude est ajustée avec un gain automatique sur une fenêtre de 150 ms. Des ondes de surface (aussi appelées ondes de Rayleigh) dominent le signal sur une portion de la section. Les ondes de surface sont un problème récurrent en sismique terrestre.

### 2.2.3 Imagerie sismique

En sismique réflexion, la séquence de traitement utilisée dépend des contraintes budgétaires, des échéanciers et des hypothèses de travail (Hill et al., 2019). Notamment, certains traitements de

données peuvent durer plusieurs jours ou plusieurs semaines : accélérer ceux-ci vient au prix de plateformes de calcul plus coûteuses. En ce qui concerne les hypothèses de travail, en présence de pendages ou en présence de forts contrastes de vitesse latéraux, par exemple, les algorithmes à privilégier varient. Ce sont les raisons pour lesquelles il existe une grande panoplie d'algorithmes de traitement en sismique. L'imagerie sismique regroupe la famille d'algorithmes organisés autour de la prédiction du modèle de réflectivité, plutôt que du modèle de propriétés physiques (telles que les vitesses) elles-mêmes. Ces algorithmes sont généralement moins coûteux et plus rapides, mais ils nécessitent l'intervention presque constante d'un spécialiste en interprétation sismique.

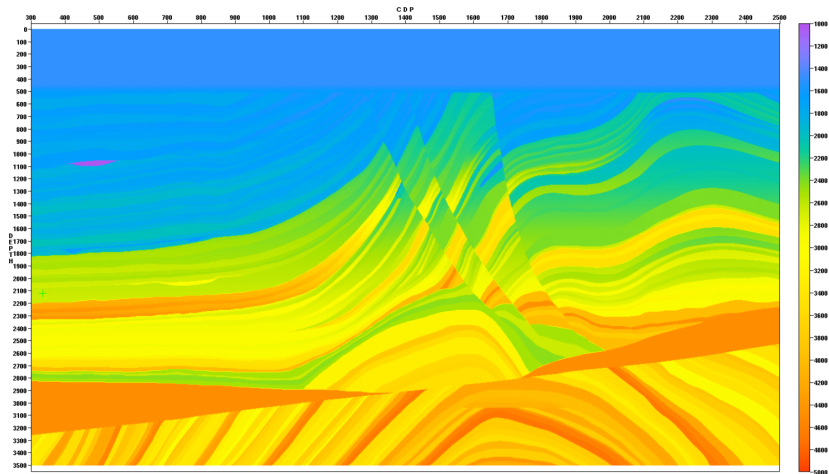
La séquence de traitement classique en imagerie est constituée de la déconvolution, de la sommation CMP et de la migration. Les prochains paragraphes de cette sous-section sont inspirés de Yilmaz (2001) et de Sheriff et al. (1995). Des exemples de résultats pouvant être obtenus par imagerie sismique sont montrés à la figure 2.12.

La déconvolution a pour objectif de remplacer les trains d'ondes créés par les réflexions de l'ondelette d'origine par des pics représentant les interfaces rencontrées. L'ondelette d'origine  $w(t)$  générée par la source, lorsqu'elle se propage dans le sol, rencontre des hétérogénéités représentées par un modèle de réflectivité  $e(t)$ . Les pics de ce modèle de réflectivité indiquent là où se produisent les réflexions. Ainsi, le train d'ondes  $x(t)$  enregistré à la surface contient la convolution de  $w(t)$  et  $e(t)$ , c'est-à-dire

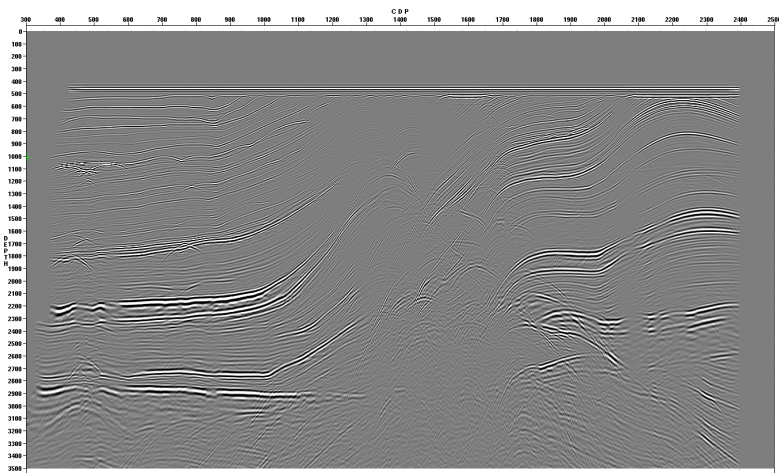
$$x(t) = w(t) * e(t). \quad (2.25)$$

Si l'on connaît la forme de  $w(t)$ , le problème se simplifie grandement : en procédant par déconvolution déterministe, on peut extraire directement le modèle de réflectivité. Lorsque  $w(t)$  est inconnue, on procède plutôt par déconvolution statistique. Le filtre de déconvolution est alors déduit par moindres carrés. Enfin, la déconvolution prédictive permet d'éliminer les multiples d'un signal, permettant d'extraire les réflexions primaires du signal.

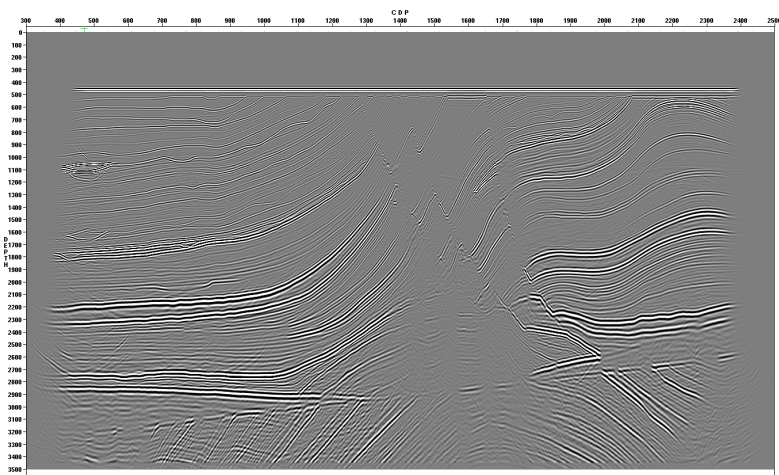
La sommation CMP inclut la correction *normal moveout* (NMO), l'analyse de vitesse et la correction *dip moveout* (DMO). La sommation CMP a pour objectif d'améliorer le rapport signal sur bruit en combinant l'information de toutes les traces de chaque regroupement CMP. Pour ce faire, les



(a) Modèle de Marmousi2



(b) Correction NMO, sommation et conversion en profondeur



(c) Migration Kirchhoff avant sommation, avec traçage de rais par énergie maximale

FIGURE 2.12 : Traitement de données typique en imagerie sismique (Martin, 2004)

réflexions doivent être ramenées au même point dans le temps, de façon à ce que la sommation soit cohérente. La correction NMO vise à corriger les traces pour l'obliquité des trajets des ondes. En effet, les événements des traces d'un même regroupement CMP subissent un retard en forme d'hyperbole à mesure que le déport augmente. Cette déformation est dictée par

$$t^2 = t_0^2 + \frac{x^2}{V_{\text{NMO}}^2} \quad (2.26)$$

où  $V_{\text{NMO}} = V_{\text{RMS}} / \cos \theta$  est la vitesse de correction,  
 $\theta$  est le pendage.

La vitesse  $V$ , couramment appelée vitesse d'intervalle, est représentée en sismique de plusieurs autres façons, notamment par la vitesse RMS  $V_{\text{RMS}}$  qui est définie par

$$V_{\text{RMS}}(t_N) = \frac{\sum_{i=0}^N V(t_i)^2 \Delta t_i}{\sum_{i=0}^N \Delta t_i}. \quad (2.27)$$

Un exemple de ces représentations de la vitesse est montré à la figure 2.13. La vitesse de correction  $V_{\text{NMO}}$  est déterminée itérativement par un processus nommé l'analyse de vitesse. Une des façons de procéder à une analyse de vitesse est de mettre en graphique la cohérence des traces — mesurée, par exemple, avec l'intercorrélacion ou la semblance — en fonction de la vitesse de correction et du temps. Les maxima du spectre donnent la vitesse de correction en fonction du temps. Un exemple d'une telle analyse de vitesse est montré à la figure 2.14. Un exemple de données corrigées par NMO est montré à la figure 2.14c. Enfin, la correction DMO permet d'imager correctement des réflecteurs dont les pendages diffèrent à même profondeur. Pour ce faire, une correction NMO est effectuée en ignorant les pendages, puis les données sont regroupées par déports, pour finalement effectuer une correction DMO. La migration avant sommation permet de contourner les corrections NMO et DMO (Hill et al., 2019).

Une fois corrigées, les données peuvent être sommées, puis migrées. Pour réaliser l'étape de la migration, il existe une très grande variété d'algorithmes, dépendamment des hypothèses de travail et de la qualité souhaitée des résultats (Hill et al., 2019). La migration remplace les positions des réflexions aux réflecteurs et focalise les hyperboles de diffraction, et ce, en combinant l'information de tous les regroupements sommés. Cela ne signifie pas pour autant que, à terme, l'axe vertical

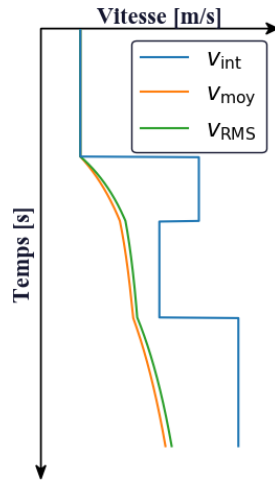


FIGURE 2.13 : Différentes représentations de la vitesse

La vitesse sismique  $V$  peut aussi être représentée avec  $V_{RMS}$ . À des fins de comparaison, la vitesse moyenne  $V_{moy}$  est aussi affichée.

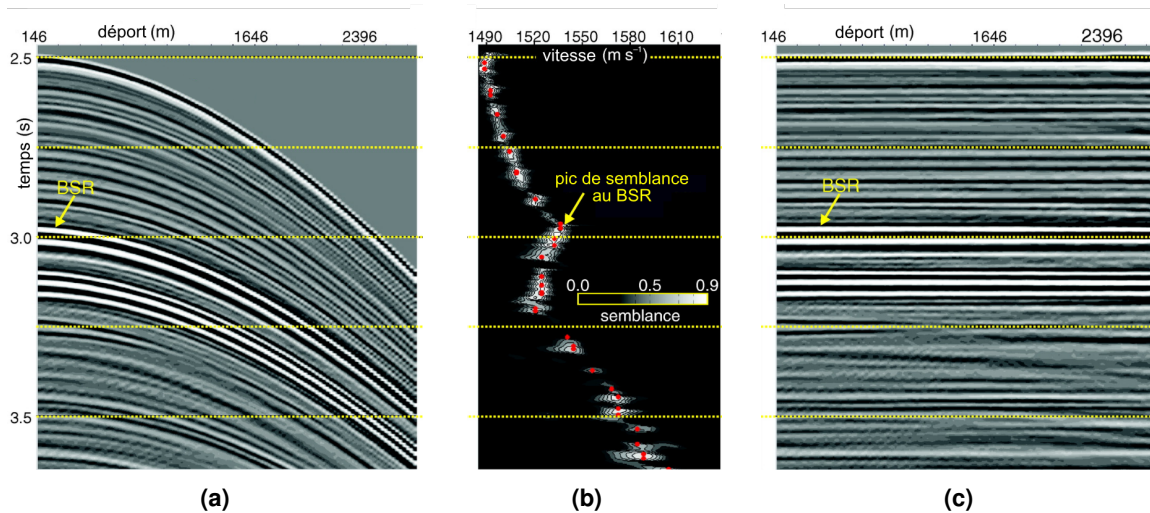


FIGURE 2.14 : Analyse de vitesse (Crutchley et al., 2015)

(a) Un regroupement CMP de données marines migrées en temps avant sommation. La *bottom-simulating reflection* (BSR) est une réflexion parallèle au fond marin, souvent associée à un contraste entre des hydrates de gaz et des sédiments chargés en gaz (Sheriff et al., 1995). (b) Le spectre de semblance. La vitesse de sommation  $V_{NMO}$  sélectionnée est une interpolation linéaire des points sélectionnés en rouge. (c) Données après correction NMO.

a des dimensions de profondeur : la migration peut être qualifiée de migration en temps ou de migration en profondeur.

Cette séquence de traitement (déconvolution, sommation, migration) comporte plusieurs défauts du point de vue de l'estimation quantitative et non biaisée des paramètres du sol. Elle nécessite une intervention humaine presque constante, ce qui ajoute un biais d'interprétation et limite les

possibilités méthodologiques (Araya-Polo et al., 2018). C'est un processus long, puisqu'il est itératif et lourd en manipulations. Couplé à l'interprétation géologique, cela a pour effet de rendre l'interprétation non unique et hautement subjective, d'où l'interprétation sismique est en quelque sorte un art (Giroux et al., 2019). De plus, une grande partie de l'information quantitative sur l'intensité des contrastes de propriétés est broyée par l'application de gain, la déconvolution ou la sommation. Pour ces raisons, la plupart des professionnels se tournent dorénavant vers des techniques plus flexibles, notamment les algorithmes de migration avant sommation.

### 2.2.3.1 Méthodes d'imagerie modernes

La *reverse time migration* (RTM, Baysal et al. 1983) se situe parmi les méthodes de migration modernes les plus populaires. La RTM permet de reconstituer un modèle de réflectivité en profondeur en extrapolant à rebours les sections sismiques. La RTM peut être effectuée avant et après sommation.

L'explication suivante de la RTM après sommation est donnée par Yilmaz (2001). Les données d'entrée représentent un plan  $x-t$  pour des capteurs situés à  $z = 0$ , c'est-à-dire une section  $P(x, z = 0, t)$ . En considérant les interfaces comme des « réflecteurs explosifs » (Claerbout, 1985), c'est-à-dire en considérant que les réflecteurs agissent comme des sources, par le principe de Huygens,  $P(x, z, t = 0)$  correspond à la section migrée. Autrement dit, à  $t = 0$ , les ondelettes coïncident avec les interfaces, puisqu'on les considère comme étant les sources. L'objectif de la RTM est donc de propager la section sismique de  $t = t_{\max}$  à  $t = 0$  pour obtenir la section migrée  $x-z$ .

La RTM avant sommation est intéressante en raison de ses liens avec le DL et certaines méthodes d'inversion sismique. Les explications suivantes sont inspirées de Feng et al. (2017) et Chen et al. (2017). La RTM avant sommation procède par méthode adjointe et son point de départ est l'équation d'onde isotrope élastique, formulée avec les vitesses et les contraintes. En 2D, elle s'écrit

$$\begin{cases} \rho \frac{\partial v_i}{\partial t} - \left( \frac{\partial \sigma_{ij}}{\partial x_j} + \frac{\partial \sigma_{xz}}{\partial z} \right) = 0 \\ \frac{\partial \sigma_{ij}}{\partial t} - \lambda \left( \frac{\partial v_x}{\partial x} + \frac{\partial v_z}{\partial z} \right) - 2\mu \frac{\partial v_i}{\partial x_j} = S_{ij} \\ \frac{\partial \sigma_{xz}}{\partial t} - \mu \left( \frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) = 0 \end{cases} \quad (2.28)$$

où  $i \in \{x, z\}$ ,  
 $v_i = \frac{\partial u_i}{\partial t}$  est la vitesse des particules,  
 $S_{ij}$  décrit la source de vibrations.

L'hypothèse de travail de la RTM consiste à supposer que des perturbations  $\delta \mathbf{m}$  (des diffracteurs) d'un modèle de propriétés physiques d'essai  $\mathbf{m}$  sont les responsables des différences  $\delta \mathbf{d}$  entre les données acquises  $\mathbf{d}$  et les données modélisées  $\mathcal{F}(\mathbf{m})$ . Autrement dit, la RTM suppose que le modèle d'essai est globalement proche de la solution et applique une correction de premier ordre au modèle. En effectuant un développement en série de Taylor de  $\mathbf{d}(\mathbf{m} + \delta \mathbf{m})$  et en ne conservant que le premier ordre, on retrouve l'approximation de Born, c'est-à-dire

$$\mathbf{d}(\mathbf{m} + \delta \mathbf{m}) \approx \mathbf{d}(\mathbf{m}) + \frac{\partial \mathbf{d}}{\partial \mathbf{m}} \delta \mathbf{m} = \mathbf{d}(\mathbf{m}) + \delta \mathbf{d}. \quad (2.29)$$

La jacobienne  $\frac{\partial \mathbf{d}}{\partial \mathbf{m}} = \mathcal{L}$  est appelée « opérateur de modélisation de Born » et permet de relier les perturbations du modèle à celles des données. En ce sens,  $\mathcal{L}$  est le modèle direct  $\mathcal{F}$ . L'opérateur de migration  $\mathcal{L}^\dagger$  est l'opérateur adjoint de  $\mathcal{L}$ , lequel approxime l'inverse de  $\mathcal{L}$ . L'opérateur de migration  $\mathcal{L}^\dagger$  n'est pas obtenu directement en raison de sa taille. La correction apportée au modèle est plutôt calculée en propageant les erreurs  $\delta \mathbf{d}$  à rebours dans le domaine associé aux champs de perturbations de vitesses et de contraintes, et ce, en déduisant l'équation adjointe à l'équation d'onde. Cette stratégie est appelée méthode adjointe en optimisation linéaire. Pour la modélisation adjointe, on utilise une approximation linéaire de l'équation d'onde, puisque les perturbations sont associées à des diffracteurs. La correction  $\delta \mathbf{m}$  correspond alors à la corrélation croisée des champs directs et adjoints. Autrement dit, les emplacements des diffracteurs correspondent aux endroits qui peuvent être associés à la fois à la présence d'une onde incidente, telle que calculée par la modélisation directe, et à la focalisation d'une diffraction, telle que calculée par la modélisation adjointe.

Plus récemment, la RTM a été reformulée dans un contexte de méthodes des moindres carrés, ce qui a donné naissance à la *least-squares RTM* (LSRTM; Guitton 2004; Dai et al. 2012; Schuster 2017). Puisque l'opérateur adjoint de la jacobienne n'est qu'une approximation de l'opérateur inverse, une image obtenue par RTM souffre de basse résolution, d'amplitudes non équilibrées dues à la divergence géométrique et d'*acquisition footprint*, une modulation de l'amplitude corrélée à la position des sources et des receveurs. Formuler le problème de migration comme un problème



de moindres de carrés permet d'améliorer la solution itérativement. La LSRTM vise à résoudre le problème d'optimisation

$$\min_{\delta \mathbf{m}} L \quad (2.30)$$

$$L = \frac{1}{2}(\mathcal{L}\delta \mathbf{m} - \delta \mathbf{d})^2. \quad (2.31)$$

La solution à ce problème est  $\frac{\partial L}{\partial \delta \mathbf{m}} = 0$ , ce qui mène aux équations normales du problème de moindres carrés, c'est-à-dire

$$\mathcal{L}^\dagger \mathcal{L} \delta \mathbf{m} = \mathcal{L}^\dagger \delta \mathbf{d} \quad (2.32)$$

$$\Rightarrow \delta \mathbf{m} = (\mathcal{L}^\dagger \mathcal{L})^{-1} \mathcal{L}^\dagger \delta \mathbf{d}. \quad (2.33)$$

La RTM peut être vue comme la première étape de la LSRTM, puisque  $\mathcal{L}^\dagger \delta \mathbf{d}$  est l'image migrée par RTM. L'équation précédente montre aussi que l'image migrée par RTM n'est qu'une approximation de l'image attendue. Appliquer l'inverse de la hessienne  $(\mathcal{L}^\dagger \mathcal{L})^{-1}$  permet de rendre l'image obtenue par RTM plus claire. L'inverse de la hessienne ne peut être appliquée directement elle non plus en raison de sa taille, ce qui fait qu'on utilise la méthode du gradient conjugué pour obtenir l'image migrée. La méthode du gradient conjugué demande d'utiliser à nouveau la méthode adjointe pour calculer l'application de  $\mathcal{L}$  et  $\mathcal{L}^\dagger$  à des résultats intermédiaires. On peut effectuer la modélisation adjointe facilement avec un code de modélisation sismique directe, puisque l'équation d'onde adjointe ne diffère de l'équation d'onde que par leurs signes. Je réfère à Fabien-Ouellet (2017) pour l'équation d'onde anisotrope viscoélastique et son adjointe, par souci de brièveté.

La RTM et la LSRTM, contrairement au traitement déconvolution-sommation-migration, tirent parti de toute l'information portée par les traces sismiques. Ces techniques réduisent la quantité de manipulations nécessaires à l'obtention du modèle de réflectivité, mais elles dépendent encore grandement d'un modèle de vitesse pour la modélisation. Ce modèle doit être obtenu en amont et doit être suffisamment adéquat pour permettre de propager les ondes approximativement au bon endroit et de justifier l'hypothèse des perturbations. Le modèle grossier initial reste inchangé à travers les itérations, ce qui limite la justesse pouvant être atteinte. Enfin, puisque la LSRTM est

une approche de moindres carrés, les résultats produits sont lisses (Wu et al., 2016; Hong-Qiao et al., 2021).

En résumé, l'imagerie sismique permet de décrire la composition du sol en termes de ses interfaces, lesquelles se matérialisent dans les modèles de réflectivité obtenus. Cela permet effectivement d'obtenir une idée assez précise de la distribution des structures géologiques, que le résultat final soit donné en temps ou en profondeur. La migration repose sur l'obtention d'un modèle de vitesse fiable, ce qui est encore un défi. Or, l'estimation de vitesse utilisée en imagerie sismique est de trop faible fidélité et de trop faible résolution pour qu'on y observe les structures de petite échelle (Fabien-Ouellet, 2017). Une grande partie des efforts déployés en imagerie visent à améliorer la qualité du modèle de vitesse en amont.

#### **2.2.4 Inversion sismique**

L'imagerie sismique, en divisant le problème de reconstitution de modèles de réflectivité en de nombreuses étapes et en introduisant des approximations, fait abstraction d'une grande partie de la non-linéarité intrinsèque du problème (Fabien-Ouellet, 2017). Pour surmonter cela, un processus reliant le plus directement possible les traces sismiques et le modèle de propriétés les ayant générées est souhaitable : c'est ce que propose l'inversion sismique. L'inversion sismique est le processus par lequel la physique des ondes est inversée pour retrouver le modèle de propriétés physiques, dont les vitesses, d'après les traces sismiques acquises. On distingue le terme « inversion » d'« estimation de modèles de propriétés physiques », dans le sens où l'inversion, quoiqu'elle ait précisément le but d'estimer les propriétés physiques du milieu étudié, réfère généralement aux algorithmes qui inversent les données par des arguments physiques.

Actuellement, la FWI constitue la fine pointe de la technologie en termes d'inversion sismique et, même, d'estimation de modèles de vitesses sismiques (Araya-Polo et al., 2019). Avec la FWI, il est possible de reconstituer des milieux anisotropes et viscoélastiques, ce qui en fait un outil versatile. La FWI permet de contourner les simplifications de l'imagerie sismique en comparant directement les traces expérimentales à des traces obtenues en propageant des ondes dans un modèle numérique. Elle tire son nom du fait que le train d'ondes synthétique au complet est modélisé et comparé

aux traces expérimentales. La FWI fonctionne itérativement : le modèle optimal est retrouvé en minimisant l'écart quadratique moyen ou, de façon équivalente, la norme  $\ell^2$  des traces mesurées par rapport à celles obtenues de la modélisation directe (Tarantola, 1984; Fabien-Ouellet, 2017). Le modèle obtenu est donc celui pour lequel les trains d'ondes simulés numériquement correspondent le mieux à ceux expérimentaux. À chaque itération, un modèle direct est calculé, ce qui peut s'avérer long et hautement coûteux en calculs (Mosser et al., 2018; Araya-Polo et al., 2019). Les paramètres du modèle de propriétés physiques sont obtenus par méthode adjointe, laquelle est, tel qu'expliqué à la sous-section précédente, l'équivalent de la descente de gradient et de la rétropropagation utilisées en DL.

Mathématiquement, le problème à résoudre s'écrit

$$\min_{\mathbf{m}} L \quad (2.34)$$

$$L = \|\mathbf{d} - \mathcal{F}(\mathbf{m})\|_2 \quad (2.35)$$

où  $\mathbf{d}$  représente les données acquises,  
 $\mathbf{m}$  représente le modèle de propriétés physiques,  
 $\mathcal{F}$  représente le modèle direct.

Autrement dit, le modèle de propriétés  $\mathbf{m}$  optimal est celui qui minimise la différence  $L$  entre les traces sismiques mesurées  $\mathbf{d}$  et celles générées synthétiquement à l'aide de  $\mathcal{F}$ . Il existe plusieurs choix de modélisateurs directs pour la FWI. L'un d'eux est *SeisCL* (Fabien-Ouellet et al., 2017; Fabien-Ouellet, 2017), lequel résout l'équation d'onde anisotrope viscoélastique par différences finies. *SeisCL* est utilisé comme modélisateur direct pour les données synthétiques de cette thèse.

La FWI souffre d'un problème appelé le « saut de cycle ». Combinés à la non-linéarité inhérente au problème d'inversion, les sauts de cycle entraînent une multiplication des minima locaux de la fonction coût  $L$  (Bunks et al., 1995; Li et al., 2016; Fabien-Ouellet, 2017; Li et al., 2017; Ovcharenko et al., 2019; Li et al., 2020). Ce problème est causé par le fait que la correspondance entre deux trains d'ondes est maximale lorsque leurs crêtes sont alignées. Cela signifie que, si l'ondelette est composée de plusieurs crêtes successives, il existe plusieurs maxima locaux autour d'un maximum global et que ces maxima locaux sont séparés du maxima global par de plutôt forts minima correspondant à des chevauchements de polarité inverse. Les forts minima locaux empêchent ty-

piquement la descente de gradient de converger vers le maximum global de correspondance en agissant comme un obstacle. Le problème du saut de cycle est illustré à la figure 2.15. Pour contrevenir à ce problème, il faut que le modèle de départ dans le processus itératif de la FWI soit plutôt proche de la solution optimale. Une autre stratégie consiste à inverser les traces par fréquences croissantes, puisque les basses fréquences sont moins sensibles au problème de saut de cycle : leurs cycles (périodes) sont beaucoup plus longs. Une approche basée sur le DL permettrait de passer entièrement outre le problème des sauts de cycles. En effet, les NNs sont en mesure de fournir un modèle de vitesse d'après les traces d'entrée en une itération, sans tenter de faire correspondre des traces générées synthétiquement aux traces initiales. C'est l'une des principales raisons qui suggèrent que les algorithmes modernes d'estimation de modèles de vitesse seront à base de DL.

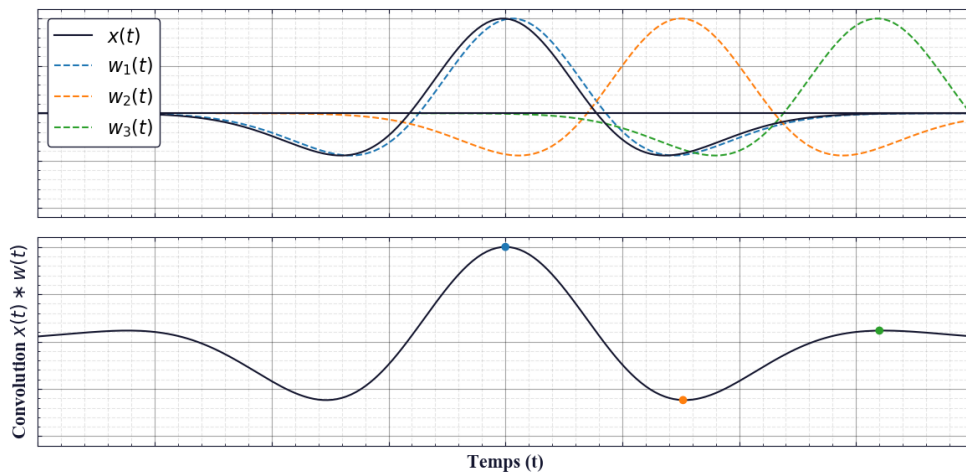


FIGURE 2.15 : Saut de cycle (inspiré de Fabien-Ouellet, 2017)

**Dans un problème impliquant des phénomènes ondulatoires, autour d'un minimum global, il y a successivement des minima causés par des polarités inverses ( $w_2$ ) et des maxima causés par des polarités cohérentes ( $w_3$ ). Le modèle inversé par FWI converge vers des maxima locaux et y demeure.**

Enfin, il est à noter que la RTM, la LSRTM et la FWI consistent toutes en des algorithmes d'optimisation par descente de gradient. Les algorithmes utilisant la méthode adjointe peuvent être vus comme des analogues du DL du point de vue de la rétropropagation : une passe directe est réalisée et des corrections sont apportées au modèle en retropropageant l'erreur. Dans tous les cas, la correction est obtenue en appliquant l'opérateur adjoint de la jacobienne à l'erreur. La LSRTM et la FWI, en particulier, sont itératives, comme l'est l'entraînement en DL.





### 3 ESTIMER LA VITESSE PAR APPRENTISSAGE PROFOND

---

Dans ce chapitre, je vais détailler la méthodologie générale qui servira au reste de cette thèse, alors que les chapitres 4 à 6 porteront sur l'accélération de certaines étapes en particulier.

Le problème que je cherche à résoudre est celui de l'estimation de modèles de vitesse d'intervalle sismique dans le domaine temporel  $V(t, x)$  à partir des formes d'ondes sismiques brutes  $d(t, h, x)$  obtenues par

$$d(t, h, x) = \mathcal{F}(V(t, x)), \quad (3.1)$$

où  $\mathcal{F}$  est le modèle physique direct permettant de générer une réponse acoustique.

Les formes d'onde d'entrée sont triées par CMPs en tenseurs  $n_t \times n_h \times n_x$ , où  $t$  est le temps de parcours double,  $h$  est le déport et  $x$  est l'emplacement du CMP.

Le DL est habituellement une approche portant uniquement sur les données. Au lieu d'utiliser explicitement les relations physiques incarnées par les équations d'onde, les NNs apprennent généralement la relation entre les données sismiques  $d(t, h, x)$  et les modèles de vitesse cibles  $V(t, x)$  de manière supervisée. Ce processus est résumé à la figure 3.1. Les NNs peuvent être visualisés comme une fonction

$$\hat{V}(t, x) = \text{NN}(d(t, h, x); \theta), \quad (3.2)$$

où  $\hat{\cdot}$  dénote les estimations,  $\theta$  sont les paramètres du NN qui sont entraînés en minimisant une fonction coût  $L$ .

L'estimation de modèles de vitesses peut donc être vue comme un problème de régression dans lequel on cherche à prédire  $V(t, x)$ , une matrice qui contient une seule valeur de vitesse scalaire pour chaque point dans le temps et l'espace. En résumé, les méthodologies de DL supervisé procèdent généralement en trois étapes :

1. Acquérir des données d'entrée et les étiquettes correspondantes;
2. Entraîner un NN avec ces données synthétiques;
3. Prédire les étiquettes des données du domaine cible.

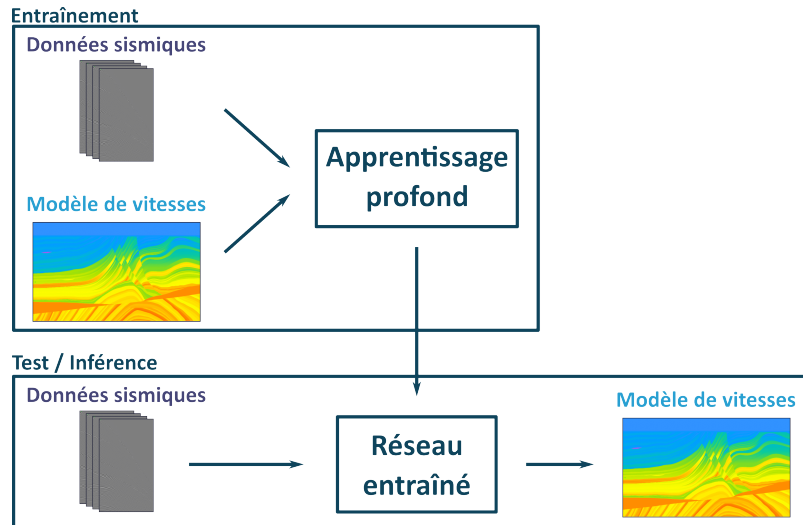


FIGURE 3.1 : Estimer des modèles de vitesse par apprentissage profond

En pratique, l'estimation de modèles de vitesses sismiques par DL est effectuée en présentant à un NN une grande quantité de modèles de vitesses connus et leurs réponses sismiques. Une fois le NN entraîné, il pourra être utilisé sur de nouvelles données sismiques pour estimer des modèles de vitesses inconnus. Le modèle de vitesse et les données affichés sont issus de Martin (2004).

### 3.1 Génération de modèles par objets

Pour réaliser la solution présentée à l'équation 3.2, des quantités massives de modèles de vitesse et de réponses sismiques doivent être utilisées. En sismique, par manque de données étiquetées, les étiquettes utilisées dans l'entraînement sont des modèles de vitesse synthétiques (Adler et al., 2021).

Certains auteurs obtiennent ces quantités massives par augmentation de données (Richardson, 2018a; Araya-Polo et al., 2019; Kazei et al., 2021; Li et al., 2021c). L'augmentation de données permet d'étendre la distribution statistique d'un jeu de données de petite taille en créant des données y ressemblant et, ainsi, de vaincre le surentraînement (Shorten et al., 2019). Si le jeu de données d'origine est constitué de modèles de propriétés physiques ayant une complexité représentative des données réelles, par exemple Marmousi2 (Brougois et al., 1990; Martin et al., 2006), SEAM (Fehler et al., 2011) ou Overthrust (Aminzadeh et al., 1997), l'augmentation favorise une meilleure applicabilité aux cas réels. En vision par ordinateur, l'augmentation de données est un standard (Shorten et al., 2019). L'augmentation consiste typiquement à déformer, à recolorer, à filtrer ou à combiner les entrées et les étiquettes associées. Les rotations d'images et l'ajout de bruit sont parmi les techniques les plus accessibles d'augmenta-



tion qu'utilisent entre autres Kazei et al. (2021). Bien qu'elles permettent de préserver certaines propriétés statistiques du jeu de données d'origine, ces transformations créent généralement des modèles de propriétés qui ne sont pas réalistes physiquement (Araya-Polo et al., 2019). C'est la raison pour laquelle plusieurs auteurs préfèrent les GANs comme technique d'augmentation (Richardson, 2018a; Araya-Polo et al., 2019). En effet, les GANs permettent de conserver l'aspect d'ensemble des données d'entraînement et, ainsi, de préserver le réalisme géologique dans les données augmentées. La géostatistique est une autre avenue d'augmentation (Li et al., 2021c). Néanmoins, l'augmentation de petits jeux de données peut conduire à un surentraînement. En choisissant sélectivement des modèles représentatifs de la géologie à imager, un fort *a priori* géologique est introduit, ce qui peut limiter l'applicabilité dans des cas réels où il y a peu d'information sur le modèle de propriétés à estimer.

Pour cette raison, les approches de génération « par objets », aussi appelées « procédurales » ou « par règles » (Deutsch et al., 1996, cités dans Mosser et al. 2020a), sont celles qui dominent en génération de modèles de propriétés physiques pour le DL (Araya-Polo et al., 2018; Mosser et al., 2018; Wang et al., 2018b; Li et al., 2020; Lin et al., 2019; Yang et al., 2019; Fabien-Ouellet et al., 2020; Mosser et al., 2020a; Wu et al., 2020; Zhang et al., 2020; Simon et al., 2022, 2023). La génération par objets consiste à ajouter pseudo-aléatoirement à un canevas des objets de façon à créer une multitude de scénarios qui diffèrent les uns des autres par le positionnement des objets, leur quantité, leurs dimensions, etc. En vision par ordinateur, la génération par objets est une approche reconnue et elle fait généralement appel à la modélisation 3D (Nevatia et al., 1977; Little et al., 1989; Gan et al., 2021; Kim et al., 2022b). Il existe une multitude de méthodes de DL utilisant uniquement les données synthétiques dans l'entraînement tout en assurant une bonne performance dans le domaine cible (de Souza et al., 2017; Hwang et al., 2021; Varol et al., 2021), quoique ce n'est que récemment qu'il est devenu possible de franchir l'écart interdomaines sans technique spécialisée (Kim et al., 2022b). Dans le cas de la géophysique, les objets générés sont géologiques : par exemple, il peut s'agir de couches, de failles, de diapirs ou de chenaux. Leurs propriétés statistiques et géométriques (par exemple les pendages et les vitesses) sont choisies par le concepteur et une variété de modèles aléatoires peut être générée en échantillonnant pseudo-aléatoirement les structures choisies. Les structures générées dépendent typiquement de

l'application envisagée, c'est-à-dire du domaine cible, et chaque auteur conçoit généralement un générateur synthétique spécifique à sa recherche.

### 3.1.1 *ModelGenerator*

Dans cette thèse, je génère synthétiquement les étiquettes en procédant par objets. Les réponses sismiques correspondantes sont générées à l'aide de *SeisCL* (Fabien-Ouellet et al., 2017).

L'étape de génération par objet est réalisée à l'aide de *ModelGenerator* (Fabien-Ouellet et al., 2022b), une librairie dédiée à la génération de modèles de propriétés physiques pour la géophysique. *ModelGenerator* fonctionne principalement par couches géologiques : les modèles sont générés en échantillonnant aléatoirement l'épaisseur, le pendage et les propriétés physiques des couches dans les limites prescrites par l'utilisateur. Certaines autres structures géologiques, comme des failles et des diapirs, sont implémentées. *ModelGenerator* a été développée pour l'estimation de modèles de vitesses sismiques, mais cette librairie peut être facilement étendue à la génération de modèles de propriétés quelconques. L'algorithme 3.1 présente le fonctionnement détaillé de *ModelGenerator*. *ModelGenerator* présente des ressemblances avec les méthodes de modélisation synthétique utilisées dans la communauté de la vision par ordinateur en ce sens que *ModelGenerator* permet de placer des objets dans un environnement et de produire et gérer les rendus aisément. Des exemples de modèles de vitesse 2D en couches pouvant être produits à l'aide de *ModelGenerator* sont affichés à la figure 3.2.

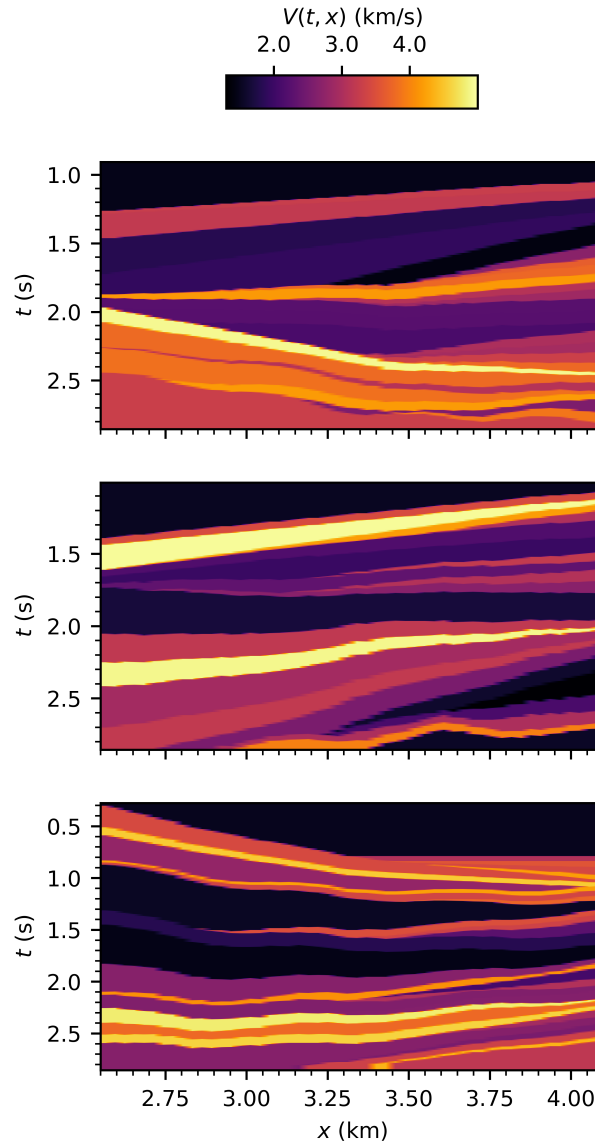
---

**Algorithme 3.1** : Générer des modèles de propriétés avec *ModelGenerator*

---

1 <b>for</b> $i \leftarrow 1$ to $N$ <b>do</b>	
2 <i>Couches</i> $\leftarrow$ INITIALISER LISTE	} Initialiser
3 <b>for</b> <i>Séquence lithologique</i> $\in$ <i>Stratigraphie</i> <b>do</b>	} environnement
4 <b>for</b> <i>Couche</i> $\in$ <i>Séquence lithologique</i> <b>do</b>	
5 <i>Couche</i> $\leftarrow$ CHOISIR PROPRIÉTÉS DE <i>Couche</i>	} Créer et placer
6 <i>Couches</i> $\leftarrow$ AJOUTER <i>Couche</i> À <i>Couches</i>	} nouvel objet
7 <b>end</b>	
8 <b>end</b>	
9 <i>Modèle</i> $\leftarrow$ PRODUIRE GRILLE 2D DE <i>Couches</i>	} Produire rendu
10     SAUVEGARDER <i>Modèle</i>	} Sauvegarder rendu
11 <b>end</b>	

---



**FIGURE 3.2 : Exemples de modèles de vitesse 2D produits par *ModelGenerator***

En général, en plus de prévenir le surajustement (*overfitting*), les données synthétiques et, en particulier, la génération par objet permettent de contrôler finement les modèles produits (Araya-Polo et al., 2019). Cela s'avère pertinent pour montrer les limites d'une méthodologie, puisque les modèles peuvent être rendus arbitrairement simples ou complexes. Concevoir un jeu de données synthétique adéquat est un défi, puisque le phénomène à caractériser doit être bien visible et que les exemples générés doivent répondre aux limites physiques du problème, tout en cherchant à explorer les cas extrêmes. La génération par objets garantit que les modèles sont créés sous des contraintes fermes, pour autant que les modèles soient conçus et inspectés par un spécialiste. En

comparaison, les GANs sont plus difficiles à contraindre, quoique les méthodes basées sur le style (Karras et al., 2018) ont le potentiel d’être plus faciles à contrôler.

## 3.2 Interfaçage à *Keras* avec *GeoFlow*

Une fois les données synthétiques générées avec *ModelGenerator*, l’entraînement d’un NN à l’estimation de propriétés physiques peut être réalisé avec *GeoFlow* (Fabien-Ouellet et al., 2022a). *GeoFlow* facilite l’interfaçage à *ModelGenerator* et à *Keras* (Abadi et al., 2015; Chollet, 2015), notamment en permettant automatiquement de :

- Gérer les jeux de données d’entraînement, de validation et de test;
- Distribuer les calculs de génération de modèles et d’entraînement;
- Gérer des NNs à plusieurs entrées et/ou plusieurs sorties;
- Sauvegarder et restaurer les poids des NNs.

En pratique, *GeoFlow* est une librairie Python implémentant l’algorithme 3.2 et où le NN est conçu avec *Keras*. L’entraînement d’un NN est réalisé en groupant les données d’entraînement et les modèles de vitesse cible par paquets et en les fournissant au NN itérativement. La mise à jour des poids du NN est effectuée en minimisant une fonction coût  $L$  programmée par l’utilisateur. L’optimisation est effectuée par descente de gradient ou des alternatives, comme Adam, et par rétropropagation (Goodfellow et al., 2016).

### 3.2.1 Choix de réseau de neurones profond

La composition d’un NN est critique pour obtenir une justesse et une robustesse accrues dans des applications ayant trait à la physique, selon le théorème *No Free Lunch*, car les choix de conception introduisent une spécificité (Wolpert et al., 1997). Or, les NNs utilisés actuellement dans la littérature sur l’estimation de modèles de vitesse par DL ne s’adaptent pas nécessairement à des levés volumineux et à des levés de terrain.

Par exemple, Araya-Polo et al. (2018), qui sont les pionniers du DL moderne pour l’estimation de modèles de vitesses (Adler et al., 2021), utilisent des couches convolutionnelles et des couches

---

**Algorithme 3.2** : Estimer des modèles de propriétés avec *GeoFlow*

---

```
1 for  $i \leftarrow 1$  to  $N_{\text{Ent.}} + N_{\text{Test}}$  do
2   | Modèle  $\leftarrow$  GÉNÉRER ALÉATOIREMENT MODÈLE
3   | Réponse  $\leftarrow$  SIMULER RÉPONSE DE Modèle
4   | SAUVEGARDER Modèle ET Réponse
5 end
6 INITIALISER NN
7 if  $\exists$  préentraînement then
8   | RESTAURER NN
9 end
10 if exécuter entraînement then
11   | INITIALISER ENTRAÎNEMENT
12   | for Époque  $\leftarrow 1$  to  $N_{\text{Époques}}$  do
13     | MÉLANGER Jeu de données d'entraînement
14     | for Paquet  $\in$  Jeu de données d'entraînement do
15       | Paquet  $\leftarrow$  PRÉTRAITER Paquet
16       | NN, Métriques  $\leftarrow$  METTRE À JOUR NN AVEC Paquet
17       | RAPPORTER Métriques
18     | end
19     | SAUVEGARDER NN
20   | end
21 end
22 if exécuter inférence then
23   | for Paquet  $\in$  Jeu de données de test do
24     | Paquet  $\leftarrow$  PRÉTRAITER Paquet
25     | Prédictions  $\leftarrow$  PRÉDIRE PROPRIÉTÉS DE Paquet AVEC NN
26     | SAUVEGARDER Prédictions
27   | end
28 end
```

} Générer étiquettes  
et données

} Entraîner NN  
avec étiquettes  
et données

} Générer prédictions  
avec nouvelles  
données

---

entièrement connectées (FCL, de *fully-connected layer*) pour relier la semblance à la vitesse d'intervalle dans des modèles composés de couches à pendages et de dômes de sel. Leur NN est illustré à la figure 3.3. Les FCLs nuisent à l'aspect spatial du problème en forçant la dimensionnalité de sortie, contrairement aux CNNs qui sont invariants spatialement. Ces auteurs sont en mesure d'atteindre des résultats d'une très bonne justesse avec aussi peu que 4 couches convolutionnelles, probablement parce que le signal sismique est compressé en amont par le biais de la semblance et que le domaine d'entraînement et le domaine de test ont de fortes ressemblances.

Plus récemment, Kazei et al. (2021) ont pu relier les traces sismiques brutes à des modèles aussi complexes que celui de Marmousi2 (Martin et al., 2006). Leur NN, entièrement convolutionnel, est illustré à la figure 3.4. Les CNNs sont facilement applicables à des levés de toutes tailles et per-

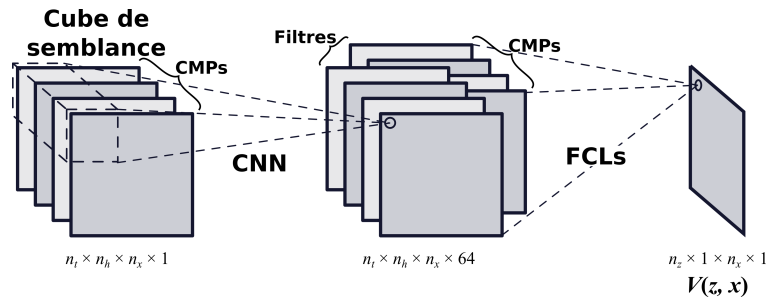


FIGURE 3.3 : Réseau de neurones utilisé par Araya-Polo et al. (2018)

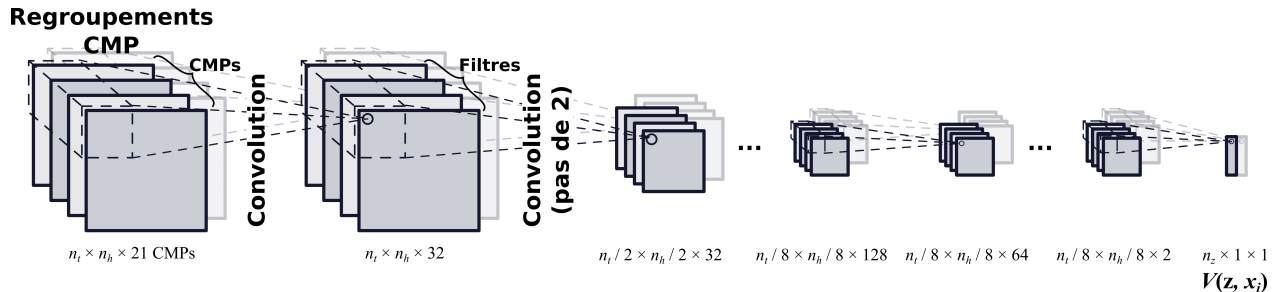


FIGURE 3.4 : Réseau de neurones utilisé par Kazei et al. (2021)

Ce NN est purement convolutionnel. Pour chaque point  $x_i$  de l'espace, Kazei et al. (2021) agrègent les 21 regroupements CMP les plus proches en un un profil de vitesse. La concaténation de tous les profils donne le modèle de vitesse 2D. Chaque couche paire dans la première partie du NN diminue la dimensionnalité de l'entrée de moitié à l'aide d'un pas de 2. La résolution de sortie est beaucoup plus basse que celle d'entrée.

mettent d'apprendre des représentations spatiales des données d'entrée de façon plus efficace, en réduisant la quantité de paramètres entraînaables. Ils sont couramment utilisés dans l'analyse d'images (He et al., 2018) et sont communs en géophysique. Quoique le NN de Kazei et al. (2021) est actuellement le plus performant sur Marmousi2, l'entraînement qu'ils utilisent requiert des modèles très ressemblants obtenus à partir de celui de Marmousi (Brougois et al., 1990). De plus, l'applicabilité de ce NN à des données réelles n'a pas été démontrée.

En fait, le NN de Fabien-Ouellet et al. (2020) est parmi les seuls ayant été testés sur des données de terrain. Fabien-Ouellet et al. (2020) montrent que le DL peut être appliqué à une série de modèles en couches horizontales ou à faible pendage. Cependant, les modèles 2D estimés par leur approche manquent de continuité latérale, puisque leur NN prédit des profils 1D en traitant les regroupements CMP un par un. En effet, pour les vitesses variant latéralement, le *moveout* des réflexions conduit à des estimations erronées et les CMPs adjacents doivent être utilisés pour corriger cet effet, comme le font la correction DMO et la migration. De plus, leurs prédictions sont effectuées dans le domaine du temps, alors que les approches précédentes fonctionnent directement dans le domaine spatial.

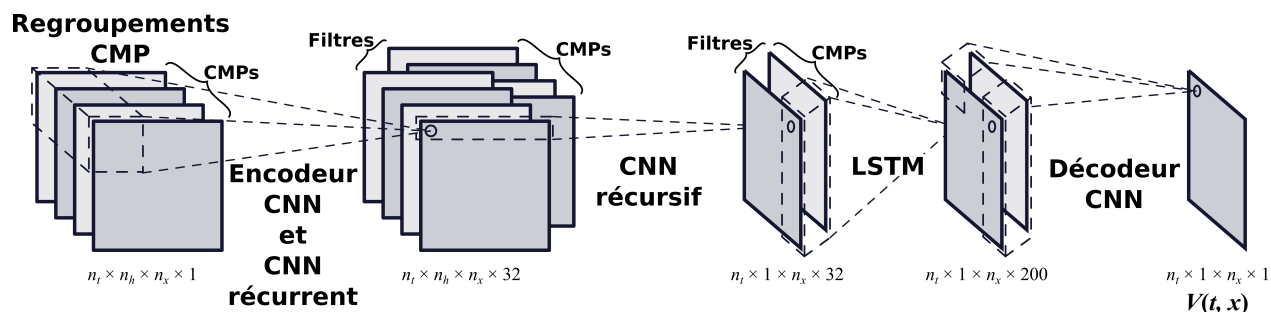


FIGURE 3.5 : Réseau de neurones inspiré de Fabien-Ouellet et al. (2020)

Une fonction d'activation linéaire est utilisée pour le décodage des vitesses. Des détails supplémentaires sur l'implémentation sont fournis au tableau 3.1.

Sous-réseau	Quantité de filtres	Taille du filtre
CNN encodeur	16	$15 \times 1 \times 1$
	16	$1 \times 9 \times 9$
	32	$15 \times 1 \times 1$
	32	$1 \times 9 \times 9$
CNN récurrent	32	$15 \times 3 \times 3$
CNN récurrent	32	$1 \times 2 \times 1$
Décodeur	1	$1 \times 1 \times 1$

TABLEAU 3.1 : Détails des couches convolutives de la figure 3.5

Les dimensions des filtres sont  $c_t \times c_h \times c_x$ . En estimation de profils 1D,  $c_x = 1$ . Chaque couche de l'encodeur CNN est suivie d'un *dropout*. Le CNN récurrent utilise des pas de 2 le long de la dimension  $h$ . Le CNN récurrent est appliqué 7 fois et le CNN récurrent est appliqué jusqu'à ce que la taille de la dimension  $h$  atteigne 1.

À travers cette thèse, j'ai choisi d'utiliser un NN qui est une extension 2D de celui proposé par Fabien-Ouellet et al. (2020) (voir la figure 3.5 et le tableau 3.1). En plus des CNNs, ce NN utilise des RNNs, lesquels construisent des représentations de signaux causaux (Rumelhart et al., 1986; Lipton et al., 2015) et sont bien adaptés aux trains d'ondes sismiques. Ce NN construit une représentation abstraite en utilisant 4 couches convolutives d'encodeur et un CNN récurrent, puis réduit la dimension des dépôts  $h$  à 1 à l'aide d'un CNN récurrent (Socher et al., 2013) avec un pas de 2 et, enfin, produit les vitesses cibles à l'aide de cellules LSTMs et de couches convolutives décodeuses. Chaque couche convulsive de l'encodeur est suivie d'une couche de *dropout* (Srivastava et al., 2014). Les couches de *dropout* éliminent aléatoirement les entrées pendant l'entraînement afin d'éviter le surajustement.

Ce NN présente deux limitations majeures : il n'existe pas de façon triviale de relier les données d'entrée à la vitesse dans le domaine spatial  $V(z, x)$  et les forts pendages apparaissent impossibles à résoudre. En effet, j'ai exploré la possibilité d'estimer  $V(z, x)$  à l'aide de RNNs, mais les

résultats convergeaient systématiquement vers la vitesse moyenne, non informative. Cette difficulté a pu être causée par l'effet d'étirement et de contraction imposé par la conversion temps-profondeur, lequel prévient un LSTM, unidirectionnel, de relier systématiquement un point dans le temps dans les données d'entrée à un point qui se trouve plus tard dans le domaine spatial de sortie, mais cela reste à démontrer. Au chapitre 4, je propose de passer par des relations analytiques pour contourner ce problème, mais utiliser des RNNs aurait pu donner une meilleure justesse, puisqu'il s'agit d'une approche basée uniquement sur les données. De plus, à ce jour, ce NN n'a pas pu être appliqué à des modèles de vitesse dans lesquels se trouvaient des pentages supérieurs à  $35^\circ$ , comme ceux dans la partie supérieure du modèle Marmousi2 (Simon et al., 2021). L'état de l'art en DL a passablement changé depuis Fabien-Ouellet et al. (2020) et les nombreuses évolutions des *Transformers* (Vaswani et al., 2017), entre autres, ont fait leur place comme remplacement aux RNNs. Il est primordial de se rappeler que certaines des conclusions observées dans cette thèse ne sont valides que pour mon choix particulier de NN et que plusieurs avenues d'amélioration sont possibles.

Néanmoins, j'ai choisi de n'utiliser que ce NN, pour limiter la quantité de degrés de liberté dans les expérimentations et car l'entraînement de NNs purement convolutionnels dans le domaine spatial avec *ModelGenerator* et *GeoFlow* s'est avéré impossible. Il semble que le fort *a priori* imposé par un domaine d'entraînement bien contraint et très ressemblant à celui de test est nécessaire au bon fonctionnement du DL pour l'estimation de modèles de vitesse dans le domaine spatial. Les résultats du chapitre 6 appuient cette hypothèse. De plus, les autres approches existantes sont d'une moins grande résolution ou organisent les données d'entrée inefficacement.

En bref, malgré l'absence de comparaisons à d'autres NNs, j'observe généralement dans cette thèse que plusieurs limitations sont d'ordre méthodologique plutôt qu'algorithmique. Néanmoins, je montrerai dans les chapitres suivants des stratégies permettant d'alléger la limitation algorithmique liée au temps d'entraînement.







## 4 APPRENTISSAGE PAR TRANSFERT HIÉRARCHIQUE

---

Pour estimer avec justesse des modèles de vitesse réalistes, variant latéralement, un NN doit être entraîné sur un jeu de données avec une configuration d'acquisition réaliste. La création d'un tel jeu de données et l'entraînement d'un NN sur ce jeu deviennent rapidement difficiles : le coût de calcul de la modélisation et de l'entraînement évolue de façon exponentielle avec la quantité de dimensions, c'est-à-dire des levés 1D à 2D à 3D. Les coûts en calculs du DL, quoique compétitifs par rapport à ceux de la FWI, sont encore très élevés. Des gains en rapidité doivent être atteints avant que les praticiens veuillent adopter le DL.

Le TL constitue une avancée majeure pour accélérer et régulariser l'entraînement, qui est l'étape la plus longue des méthodologies à base de DL, outre la création des jeux de données synthétiques. Dans le domaine de l'exploration sismique, le TL a été utilisé pour inclure des informations haute fidélité de levés voisins dans le traitement d'un levé donné (Siahkoobi et al., 2019a), pour combler l'écart entre les données synthétiques et les données de terrain en pointé des vitesses (Park et al., 2020), ainsi que pour combler l'écart entre les modèles de vitesse synthétiques et ceux générés par un GAN en estimation de modèles de vitesse (Sun et al., 2021b). Bien qu'il existe de multiples façons d'effectuer du TL, certaines peuvent être plus appropriées lorsqu'il existe des équivalences fortes entre deux problèmes donnés.

Dans ce chapitre, je montre que le TL est une bonne stratégie pour réduire les coûts de calcul de l'application du DL à l'estimation de modèles de vitesse pour les levés sismiques 2D et, par extrapolation, 3D. J'y développe un moyen efficace de réduire les coûts de calcul et d'améliorer la justesse de l'estimation de modèles de vitesse 2D en utilisant toute l'information acquise dans des contextes plus simples. Plus spécifiquement, je préentraîne d'abord un NN sur le problème d'estimation de vitesse 1D tel que formulé par Fabien-Ouellet et al. (2020). Leur méthodologie peut être appliquée à des données de terrain, mais peu ou pas à des modèles 2D, parce que leur NN traite les regroupements CMP un par un et que, pour des vitesses variant latéralement, la *moveout* des réflexions conduit à des estimations erronées. Je permets ensuite au NN préentraîné de traiter l'information 2D à travers des regroupements CMPs adjacents en étendant les filtres convolutifs

le long de l'axe des CMPs. Cette dernière formulation est similaire à celle de Kazei et al. (2021). Enfin, je vais montrer le potentiel de cette stratégie sur des données marines réelles.

Le matériel de ce chapitre est publié chez *GEOPHYSICS* :

Simon J, Fabien-Ouellet G, Gloaguen E, Khurjekar I (2023). Hierarchical transfer learning for deep learning velocity model building. *GEOPHYSICS*, pages R79–R93. DOI :10.1190/geo2021-0470.1.

Le code source est disponible en ligne (Simon, 2022a).

## 4.1 Méthodologie

### 4.1.1 Apprentissage par transfert

Le TL consiste à exploiter l'information d'un contexte précédent pour améliorer la généralisation dans un autre contexte. Le raisonnement est le suivant : en résolvant d'abord des tâches faciles, on rapproche les poids du NN de l'optimum, ce qui évite les minima locaux dans les tâches difficiles. Le TL est utile lorsque la distribution des données pour le problème source est suffisamment bien échantillonnée (c'est-à-dire en présence de données massives), mais lorsque le jeu de données pour le problème actuel est mal annoté ou manque en quantité. La plupart des exemples de la littérature sur la vision par ordinateur s'appuient sur des jeux de données tels que COCO de Microsoft (Lin et al., 2015) ou ImageNet de Stanford (Deng et al., 2009), qui contiennent des images manuellement étiquetées de personnes, d'animaux, de véhicules, de meubles, etc. Le problème cible peut contenir des objets qui étaient absents du problème initial, mais l'entraînement présenterait néanmoins un temps de convergence et une généralisation améliorés et nécessiterait moins de données. Essentiellement, le TL relâche l'hypothèse selon laquelle les données d'entraînement suivent la même distribution statistique que les données de test (Tan et al., 2018). Dans le cas d'images, cela est rendu possible par le fait que la plupart des objets partagent des caractéristiques à basse échelle, telles que les traits, l'éclairage, et ainsi de suite (Goodfellow et al., 2016). Par conséquent, les représentations intermédiaires dans des problèmes qui ne sont pas reliés à première vue peuvent bénéficier de redondance statistique (Bengio et al., 2011). En d'autres

termes, les distributions statistiques des représentations de basse échelle se chevauchent et le TL aide un NN à identifier les caractéristiques qui sont pertinentes pour des jeux de données autres que celui en question.

Le TL basé sur le NN (Tan et al., 2018) est l'approche la plus populaire de TL (Goodfellow et al., 2016) et je l'appellerai simplement TL. Cette stratégie est réalisée en préentraînant un NN sur un problème plus simple et en utilisant le NN préentraîné en totalité ou en partie dans le nouveau contexte. L'entraînement peut être relancé avec un taux d'apprentissage, une taille de jeux de données et une taille de paquets réduits. Si le nouveau contexte partage une vague ressemblance avec le précédent, les poids du NN seront plus proches de l'optimum qu'ils ne le seraient s'ils étaient initialisés de manière aléatoire; ceci est particulièrement vrai lorsque les tâches sont étroitement liées. Le TL peut bénéficier à l'estimation de modèles de vitesse 2D de deux façons : en étendant le NN le long de l'axe des CMP lors du passage du traitement des données sismiques 1D au 2D et en résolvant des sous-tâches de difficulté incrémentale.

#### 4.1.1.1 Transfert de 1D à 2D

La stratégie d'entraînement utilisée met à profit le TL pour entraîner un NN afin d'estimer des modèles de vitesse synthétiques 1D avant d'estimer des modèles 2D. Essentiellement, cela permet d'exploiter le faible coût de calcul et la faible complexité des données 1D. Une fois la convergence atteinte sur le cas 1D, les poids sont étendus le long de la nouvelle dimension et l'entraînement est redémarré avec un taux d'apprentissage plus faible. En d'autres termes, le NN a deux modes de fonctionnement : le traitement 1D et le traitement 2D. En mode 1D, les CNNs encodeurs n'ont aucune composante le long de la dimension de l'axe  $x$  des CMPs, c'est-à-dire que chaque CMP est traité indépendamment, comme le font Fabien-Ouellet et al. (2020). En mode 2D, une dimension est ajoutée le long de l'axe  $x$ . L'entraînement se fait en deux étapes : d'abord, le NN est entraîné sur des modèles à couches horizontales en mode 1D. Ensuite, le NN entraîné est augmenté en mode 2D et l'entraînement se poursuit sur un jeu de données avec des modèles contenant des pentages non nuls. Pour convertir le NN 1D en NN 2D, les poids 1D  $\mathbf{H}_{thij}^{1D}$  sont insérés le long de

la nouvelle dimension, à l'indice central des filtres convolutifs  $\mathbf{H}_{thxij}^{2D}$ , c'est-à-dire,

$$\mathbf{H}_{thxij}^{2D} = \begin{cases} \mathbf{H}_{thij}^{1D} & \text{si } x = \lceil c_x/2 \rceil \\ 0.1 \hat{\mathbf{H}}_{thxij}^{2D} & \text{sinon} \end{cases}, \quad (4.1)$$

où  $i$  et  $j$  sont les indices des filtres source et cible,  
 $\hat{\mathbf{H}}_{thxij}^{2D}$  est échantillonné par initialisation normale de Glorot (Glorot et al., 2010).

Le raisonnement est que les NNs 1D et 2D sont équivalents lorsque  $\hat{\mathbf{H}}_{thxij}^{2D} = 0$ . En d'autres termes, j'aborde l'estimation de modèles de vitesse 2D comme une perturbation du cas 1D.

#### 4.1.1.2 Apprentissage multitâche

La deuxième façon dont je tire parti du TL est celle utilisée par Fabien-Ouellet et al. (2020). J'utilise l'apprentissage multitâche (Caruana, 1995; Wu et al., 2019b) pour régulariser l'entraînement en faisant résoudre à un NN quatre tâches interreliées et progressivement plus difficiles : la détection des réflexions primaires et l'estimation des vitesses  $V_{RMS}(t, x)$ ,  $V(t, x)$  et  $V(z, x)$ . Pour produire ces nouvelles sorties, le NN présenté à la figure 3.5 du chapitre précédent est modifié de la façon présentée à la figure 4.1. L'entraînement se concentre d'abord sur les tâches les plus faciles (par exemple, la vitesse RMS, plus lisse), puis progresse à travers les objectifs en mettant à jour leurs pondérations individuelles. La vitesse d'intervalle en profondeur  $V(z, x)$  est déduite de  $V(t, x)$  analytiquement à l'aide de

$$V(z_i, x) = V(t(z_i), x), \quad t(z_i) = \sum_{j=1}^{i-1} \frac{2\Delta z}{V(z_j, x)}, \quad (4.2)$$

où  $z_i$  est la discrétisation en cellules du domaine spatial,  
 $\Delta z$  est la longueur de ces cellules.

Puisque cette équation est entièrement différentiable, elle peut être utilisée pour la rétropropagation. Cette formule n'est valide que pour des modèles à couches planes. Pour de petits pendages,  $V_{RMS}(t, x)$  est équivalent à la vitesse de sommation et  $V(t, x)$  peut être utilisée pour la migration en temps avant sommation. Pour des modèles plus complexes, la migration en profondeur doit être privilégiée et  $V(z, x)$  est la seule vitesse pertinente. Dans ce cas, les vitesses  $V_{RMS}(t, x)$  et  $V(t, x)$  ne sont pas valides physiquement et elles ne peuvent être utilisées pour la prédiction des

arrivées. Par contre, elles peuvent être traitées comme des versions filtrées de  $V(z, x)$  ( $V_{\text{RMS}}(t, x)$  met l'emphase sur les basses fréquences et  $V(t, x)$  balance la résolution verticale des couches). En ce sens, elles peuvent servir de régularisation pour la tâche plus complexe consistant à prédire  $V(z, x)$ . Enfin, l'ajout de pendages peut être vu comme un incrément en difficulté par rapport à l'estimation de modèles de vitesses en couches planes. Ces considérations suggèrent que le TL peut faciliter l'entraînement.

La fonction coût totale a quatre composantes, chacune d'entre elles correspondant à l'une des sorties du NN. De plus, deux termes de régularisation sont ajoutés : l'écart avec la dérivée temporelle (ou spatiale) et la variation totale. La fonction coût  $L$  est

$$L = \alpha_{\text{ref}} L_{\text{ref}} + \sum_{\mathbf{V}} \sum_{i=1}^3 \alpha_{\mathbf{V}i} L_{\mathbf{V}i} \quad (4.3)$$

$$L_{\text{ref}} = \begin{cases} -\log \hat{r}(t, x) & \text{si } r(t, x) = 1 \\ -\log(1 - \hat{r}(t, x)) & \text{si } r(t, x) = 0 \end{cases} \quad (4.4)$$

$$L_{\mathbf{V}1} = \frac{\|\hat{\mathbf{V}} - \mathbf{V}\|_2^2}{\|\mathbf{V}\|_2^2} \quad (4.5)$$

$$L_{\mathbf{V}2} = \frac{\|D_z \hat{\mathbf{V}} - D_z \mathbf{V}\|_2^2}{\|D_z \mathbf{V}\|_2^2} \quad (4.6)$$

$$L_{\mathbf{V}3} = 0.02 \cdot \frac{\|D_z \hat{\mathbf{V}}\|_1 + \|D_x \hat{\mathbf{V}}\|_1}{\|\hat{\mathbf{V}}\|_1}, \quad (4.7)$$

où  $\alpha_{\text{ref}}$  et  $\alpha_{\mathbf{V}i}$  sont des poids,  
 $\mathbf{V}$  représente l'une des trois représentations de la vitesse,  
 $D_z$  est la dérivée discrète en temps ou en profondeur,  
 $L_{\text{ref}}$  est l'entropie croisée binaire,  
 $r(t, x)$  et  $\hat{r}(t, x)$  sont les primaires attendues et estimées.

La différence L2  $L_{\mathbf{V}1}$  sur les vitesses et les termes de régularisation  $L_{\mathbf{V}2}$  et  $L_{\mathbf{V}3}$  sont pondérés dans des rapports 7 : 2 : 1, c'est-à-dire,  $\alpha_{\mathbf{V}1} = 0.7\alpha_{\mathbf{V}}$ ,  $\alpha_{\mathbf{V}2} = 0.2\alpha_{\mathbf{V}}$  et  $\alpha_{\mathbf{V}3} = 0.1\alpha_{\mathbf{V}}$ , avec un poids global de  $\alpha_{\mathbf{V}}$ , sauf pour  $V_{\text{RMS}}(t, x)$ , pour lequel  $\alpha_{\mathbf{V}1} = \alpha_{\mathbf{V}}$ . Ces rapports ont été obtenus par essai-erreur.

Pour tirer parti du TL, les entraînements 1D et 2D sont divisés en trois étapes de 2 000 itérations chacune. La difficulté de l'entraînement est incrémentale : les pondérations entre les objectifs sont énumérées dans le tableau 4.1 et sont les mêmes en 1D et 2D.

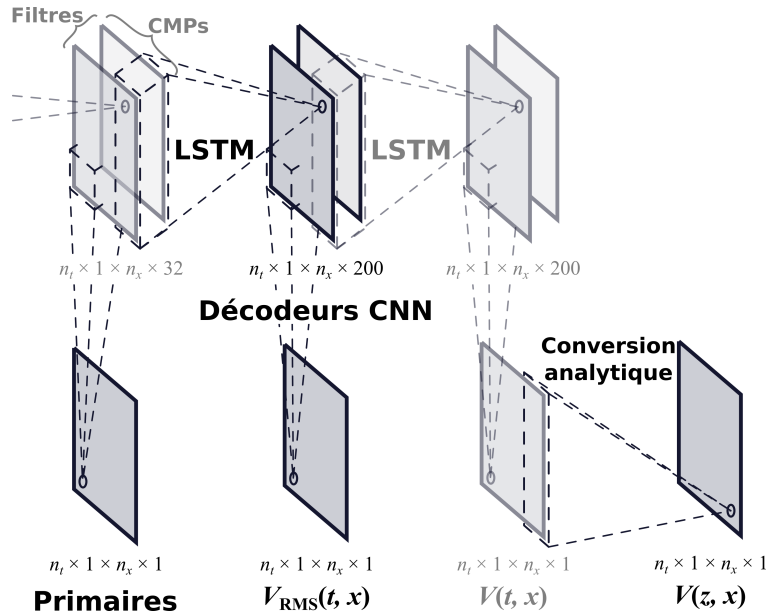


FIGURE 4.1 : Réseau de neurones multitâche

Le NN présenté à la figure 3.5 du chapitre précédent peut être modifié pour lui permettre de réaliser quatre sous-tâches. Les couches et les tenseurs préexistants sont en gris. Les sorties sont décodées à partir des représentations intermédiaires à l'aide de couches convolutives avec des filtres de taille  $1 \times 1$ , la seule exception étant que  $V(z, x)$  est déduite de  $V(t, x)$  en convertissant  $V(t, x)$  de façon analytique. Une fonction d'activation sigmoïde est utilisée pour le décodage des réflexions primaires et des fonctions d'activation linéaires sont utilisées pour le décodage des vitesses.

$\alpha_{ref}$	$\alpha_{V_{RMS}(t,x)}$	$\alpha_{V(t,x)}$	$\alpha_{V(z,x)}$
0.6	0.3	0.1	0.0
0.1	0.7	0.2	0.0
0.1	0.3	0.5	0.1

TABEAU 4.1 : Pondérations  $\alpha_i$  pendant les trois étapes d'entraînement

#### 4.1.2 Jeu de données synthétique

J'ai généré de manière aléatoire 20 000 modèles de vitesse latéralement homogènes (modèles 1D) et 2 000 modèles de vitesse en couches avec des pendages allant jusqu'à  $10^\circ$  et des interfaces ondulées (modèles 2D). Des exemples de modèles de vitesse 2D sont présentés aux figures 4.3, 4.4 et 4.5. Les deux jeux de données sont échantillonnés de la même manière et tous deux imitent des environnements sédimentaires. L'unique différence est que des interfaces ondulées de pendage non nul sont ajoutées aux modèles 2D. Les modèles ont une largeur de 8 650 m et une profondeur de 9 400 m et sont discrétisés en cellules de  $6.25 \text{ m} \times 6.25 \text{ m}$ . L'estimation de modèles de vitesse est effectuée sur les regroupements CMP bruts; peu de traitement est nécessaire. Les tirs sont modélisés à l'aide d'ondelettes gaussiennes de première et de deuxième dérivées dont



les fréquences de crête  $f$  sont choisies aléatoirement entre 21 Hz et 31 Hz et dont les phases sont aléatoires. Les données 1D comportent 1 tir, alors que les données 2D en comportent 32 tirs espacés de 50 m. Les 72 récepteurs sont répartis uniformément dans une géométrie en ligne avec un déport minimum de 470 m et un déport maximum de 4 075 m. Le temps d'écoute est de 8 s, le taux d'échantillonnage est de 0.4 ms et les données sont sous-échantillonnées par 10. Les frontières extérieures sont absorbantes sur 125 m. Les tirs et les récepteurs sont situés juste sous la frontière absorbante. Un gain aléatoire entre  $t^{-2}$  et  $t^2$  est appliqué pour réduire l'impact de la divergence sphérique. Un bruit uniformément distribué de 2% l'amplitude maximale est ajouté pour améliorer la robustesse. La génération de modèles synthétiques prend environ une demi-semaine que ce soit en 1D ou 2D, et ce, avec 8 GPUs Tesla V100 SXM2.

Hormis les modèles de vitesse d'intervalle, les étiquettes utilisées pour l'entraînement sont le modèle de vitesse RMS et les emplacements des réflexions primaires. Les deux sont générés de la même manière que Fabien-Ouellet et al. (2020). L'étiquette pour les emplacements des primaires  $R_t$  est binaire : chaque cellule de la grille est associée à une réflexion primaire (1) ou non (0). La couverture temporelle des emplacements est étendue à l'aide d'une fenêtre de largeur  $\ell = 0.4f^{-1}$ . Mathématiquement,

$$R_t = \begin{cases} 1 & \text{si } t_0^i - \frac{\ell}{2} < t < t_0^i + \frac{\ell}{2}, \text{ pour } i \in \mathcal{R} \\ 0 & \text{sinon} \end{cases}, \quad (4.8)$$

où  $\mathcal{R}$  est l'ensemble des réflexions primaires.

## 4.2 Résultats

J'ai entraîné le NN avec des paquets de 50 de modèles 1D ou de 2 de modèles 2D et j'ai utilisé l'optimiseur Adam (Kingma et al., 2017) avec  $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$  et des taux d'apprentissage de  $\lambda = 8 \cdot 10^{-4}$  en 1D et  $\lambda = 8 \cdot 10^{-5}$  en 2D. J'ai entraîné indépendamment 16 NNs en ensemble (Opitz et al., 1999) pour assurer une bonne performance dans les régions où les prédictions sont incertaines. Les NNs diffèrent par leur initialisation et l'ordre dans lequel les exemples d'entraînement sont fournis. J'ai supprimé 2 NNs qui fournissaient des résultats aberrants, ce qui porte le total à 14. Tous les résultats présentés ci-après sont en fait la moyenne sur l'ensemble de 14 prédictions.

Les hyperparamètres ont été choisis par essais et erreurs. Le préentraînement d'un NN sur des modèles 1D prend en moyenne 2 heures et 34 minutes alors que l'entraînement sur des modèles 2D prend en moyenne 20 heures et 19 minutes. Toutes les statistiques de la section des résultats sont données pour 2 GPUs.

#### 4.2.1 Données synthétiques

J'utilise la mesure de l'indice de similarité structurelle (SSIM, de *structural similarity index measure*; Wang et al. 2004) et la racine carrée de l'erreur quadratique moyenne (RMSE, de *root-mean-square error*) sur les modèles de vitesse d'intervalle  $V(t, x)$  pour quantifier la performance. Le SSIM est couramment utilisé en vision par ordinateur pour mesurer la similarité des images tout en tenant compte des distorsions et des déplacements, notamment. Le SSIM varie entre 0 (aucune similarité) et 1 (image exactement identique). L'estimation de modèles de vitesse par DL sur un jeu de test de 100 nouveaux modèles 2D a pris environ 3 minutes par NN 1D et 4 minutes par NN 2D. Le SSIM sur  $V(t, x)$  pour le jeu de test est indiqué à la ligne (a) du tableau 4.2. Les mesures pour le jeu d'entraînement 2D (ligne (b)) sont comparables, ce qui indique un surajustement faible ou nul. À titre de comparaison, la justesse de ces mêmes 100 modèles avec les NNs entraînés uniquement sur des données 1D est indiquée à la ligne (c) : l'entraînement 2D a effectivement amélioré la justesse des estimations. La convergence de la fonction coût pendant chaque étape de l'entraînement est montrée à la figure 4.2. Au début de la phase d'entraînement 2D, les composantes de la fonction coût sont plus faibles qu'au début de la phase d'entraînement 1D : la solution fournie par l'entraînement pour l'estimation de modèles de vitesse 1D est une bonne solution de départ pour l'estimation de modèles de vitesse 2D. Bien que Fabien-Ouellet et al. (2020) aient signalé que la fonction coût diminuait de façon drastique au cours des premières itérations, la convergence est plus lente, très probablement à cause du *dropout*. La courbe pour  $V(z, x)$  ne diminue pas aussi régulièrement et aussi intensément que les autres courbes, ce qui est particulièrement vrai pour la phase d'entraînement 1D. Rappelons que l'entraînement explicite de  $V(z, x)$  n'est effectué qu'à la dernière étape de l'entraînement 1D et 2D et reste minimal ( $\alpha_{V(z,x)} = 0.1$ ), la raison étant que l'utilisation de  $\alpha_{V(t,x)} \gg 0.1$  fait diverger l'entraînement.

	Jeu de données	Données d'entraînement		$\lambda$	Justesse		
		1D	2D		SSIM	RMSE (m/s)	
(a)	Avec TL	Test	✓	✓		$0.893 \pm 0.052$	$198 \pm 91$
(b)		Entr.	✓	✓		$0.890 \pm 0.053$	$193 \pm 81$
(c)	Préentraînement	Test	✓			$0.788 \pm 0.081$	$292 \pm 112$
(d)	Sans TL	Test		✓	$8 \cdot 10^{-4}$	$0.851 \pm 0.056$	$304 \pm 120$
(e)					$8 \cdot 10^{-5}$	$0.863 \pm 0.056$	$364 \pm 154$

TABLEAU 4.2 : Justesse de  $V(t, x)$  sur le jeu d'entraînement ou celui de test

Les incertitudes sur la justesse correspondent à un seul écart-type sur la totalité des exemples. N.B. : La justesse est fournie pour des comparaisons une à une entre la prédiction moyenne de l'ensemble et la référence; l'écart-type n'est pas l'écart-type sur l'ensemble des NNs, mais plutôt sur la totalité des exemples de test. Sauf indication contraire,  $\lambda = 8 \cdot 10^{-4}$  en 1D et  $\lambda = 8 \cdot 10^{-5}$  en 2D. (a) Justesse finale. (b) Justesse sur le jeu de données d'entraînement. (c) Justesse avec préentraînement seul. (d, e) Justesse sans TL entre le cas 1D et le cas 2D.

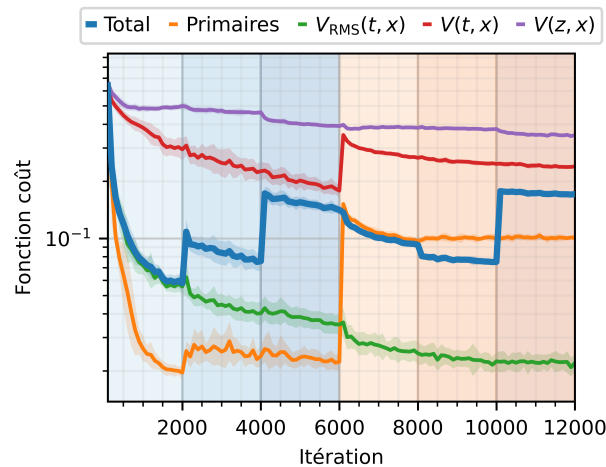


FIGURE 4.2 : Fonction coût lors de l'entraînement

( , ■, ■) Fonction coût pendant le préentraînement sur les modèles 1D. ( , ■, ■) Fonction coût lors de l'entraînement sur des modèles 2D. ( , ■)  $\alpha_{ref} = 0.6$ ,  $\alpha_{V_{RMS}(t,x)} = 0.3$ ,  $\alpha_{V(t,x)} = 0.1$ ,  $\alpha_{V(z,x)} = 0.0$ . (■, ■)  $\alpha_{ref} = 0.1$ ,  $\alpha_{V_{RMS}(t,x)} = 0.7$ ,  $\alpha_{V(t,x)} = 0.2$ ,  $\alpha_{V(z,x)} = 0.0$ . (■, ■)  $\alpha_{ref} = 0.1$ ,  $\alpha_{V_{RMS}(t,x)} = 0.3$ ,  $\alpha_{V(t,x)} = 0.5$ ,  $\alpha_{V(z,x)} = 0.1$ . L'écart-type sur les 14 NNs est indiqué par l'ombrage.

La figure 4.3 comporte le 50<sup>e</sup> percentile en termes du SSIM sur les modèles de test. Des résultats supplémentaires pour les 90<sup>e</sup> et 10<sup>e</sup> percentiles sont présentés aux figures 4.4 et 4.5. La variabilité des estimés de  $V(t, x)$  et de  $V(z, x)$  est illustrée aux figures 4.6 et 4.7. Comme affiché à la première colonne de la figure 4.3, le préentraînement sur les données 1D a convergé de manière appropriée, bien que les résultats manquent de continuité latérale. Ce résultat était attendu, la raison étant que, pour un seul regroupement CMP, il y a peu d'informations concernant l'étendue latérale des structures. Après l'entraînement des NNs sur les données 2D, les vitesses estimées ont une meilleure continuité en raison du fait que les informations sont amassées sur plusieurs regroupements CMP. Les modèles de vitesse estimés sont généralement moins extrêmes que ceux

attendus, mais les interfaces sont nettes et précises. Les couches plus profondes sont estimées avec moins de justesse et de continuité, mais les NNs entraînés uniquement sur des données 1D ne peuvent pas du tout reconstruire ces couches. En comparant les regroupements à déport constant et les arrivées des réflexions primaires estimées, on observe que les réflexions les plus fortes sont les plus faciles à détecter. En retour, les arrivées claires se traduisent par des interfaces nettes, comme le suggère l'estimation de  $V(t, x)$ . Les estimés de  $V(z, x)$  sont sensibles à la sous-estimation des vitesses dans les couches moins profondes, produisant des artefacts ondulatoires. Les interfaces supplémentaires sont des artefacts causés par le calcul de la moyenne de toutes les 14 estimés de  $V(z, x)$ ; le calcul de la moyenne dans le domaine de la profondeur ne se comporte pas aussi bien, car les vitesses erronées dans les couches supérieures entraînent une diminution de la justesse des emplacements des couches. Néanmoins, les interfaces des couches en profondeur présentent une meilleure continuité par rapport aux résultats des NNs entraînés sur des données 1D.

L'erreur pour chaque couche individuelle des exemples de test triée par vitesse cible, par épaisseur de couche et par profondeur est reportée à la figure 4.8. La figure 4.8a permet d'observer que les couches avec des vitesses extrêmes sont plus sujettes à une estimation erronée. La figure 4.8b permet d'observer que les couches très fines, dont l'épaisseur est inférieure à 10 m, ne sont généralement pas bien estimées. La figure 4.8c permet d'observer que l'erreur augmente légèrement pour les couches très profondes. Dans chacune des trois sous-figures, il existe un groupe de faible erreur correspondant aux couches d'eau, qui ont une vitesse d'environ 1 500 m/s, une épaisseur de plus de 1 000 m et une profondeur de 0 m. Le DL est toujours limité par les contraintes physiques de l'imagerie et retrouver des couches plus fines que  $\lambda/4$  n'est pas envisageable. Pour des fréquences comprises entre 26 Hz et 31 Hz et des vitesses comprises entre 1500 m/s et 4000 m/s, la résolution devrait être comprise entre 39 m et 12 m. Le jeu de données d'entraînement contient de telles couches, qui présentent effectivement de grandes erreurs.

Afin de confirmer que le TL facilite la convergence dans le cas 2D, j'ai également entraîné des NNs uniquement sur des modèles 2D, avec les poids du tableau 4.1. Qu'un taux d'apprentissage de  $\lambda = 8 \cdot 10^{-4}$  ou de  $\lambda = 8 \cdot 10^{-5}$  soit utilisé, des justesses aussi élevées que celles obtenues avec TL ne peuvent être atteintes. Les justesses pour les deux cas sont indiquées aux lignes

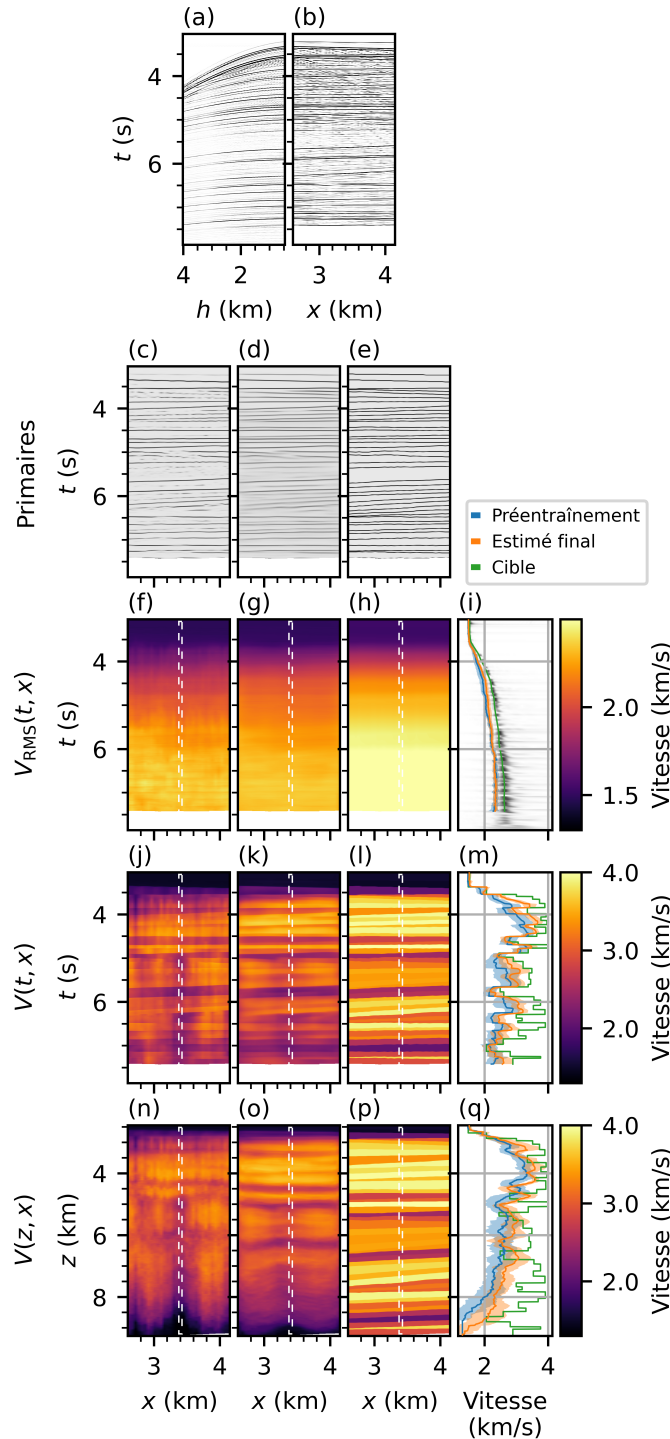


FIGURE 4.3 : Estimations moyennes pour le 50<sup>e</sup> percentile

La moyenne est effectuée sur 14 NNs. Le SSIM sur  $V(t, x)$ , tel que mesuré après l'entraînement sur les données 2D, est de 0.906. La rangée supérieure comporte des tranches de l'entrée : (a) premier regroupement CMP et (b) regroupement à départ constant. Les rangées suivantes comportent (c-e) l'identification des réflexions primaires, (f-i)  $V_{RMS}(t, x)$ , (j-m)  $V(t, x)$  et (n-q)  $V(z, x)$ . Les colonnes comportent (c-n) les estimations après le préentraînement sur des données 1D, (d-o) les estimations finales après l'entraînement sur les données 2D, (e-p) les valeurs cibles et (i-q) les tranches à  $x \approx 3.4$  km, comme indiqué par les tirets, ainsi que l'écart-type sur 14 NNs. L'arrière-plan de (i) est la semblance. Les entrées et les sorties sont rognées au-delà de la dernière réflexion. La variabilité des estimations de  $V(t, x)$  et de  $V(z, x)$  est illustrée aux figures 4.6 et 4.7.

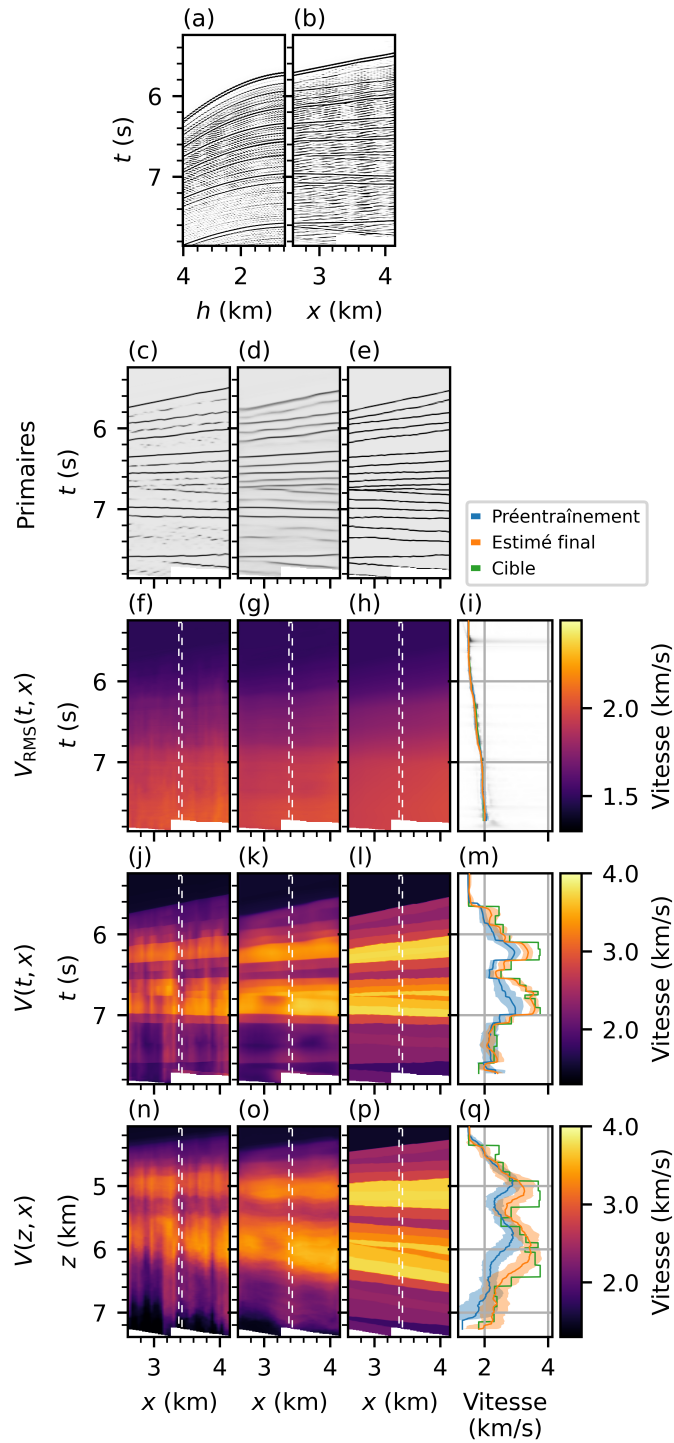


FIGURE 4.4 : Estimations moyennes pour le 90<sup>e</sup> percentile

La mise en forme est la même que celle de la figure 4.3. Le SSIM sur  $V(t, x)$  est de 0.951

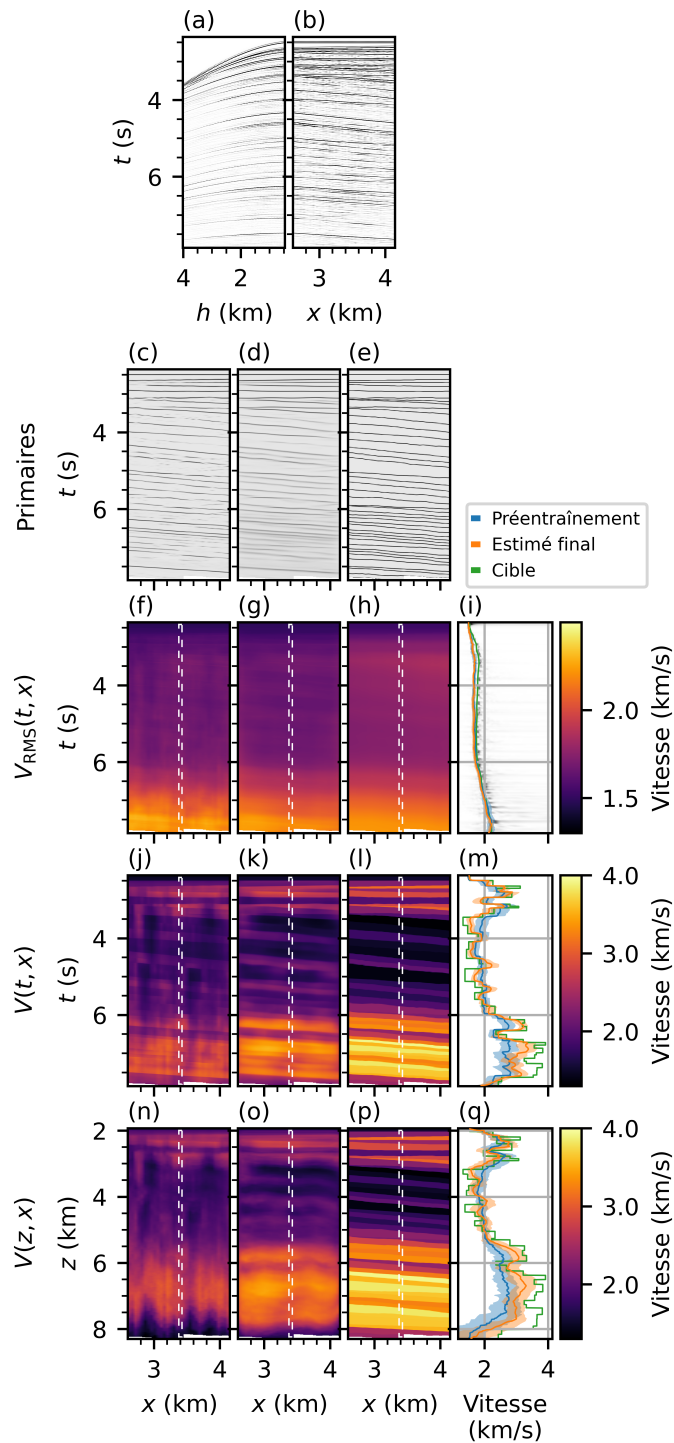


FIGURE 4.5 : Estimations moyennes pour le 10<sup>e</sup> percentile

La mise en forme est la même que celle de la figure 4.3. Le SSIM sur  $V(t, x)$  est de 0.831

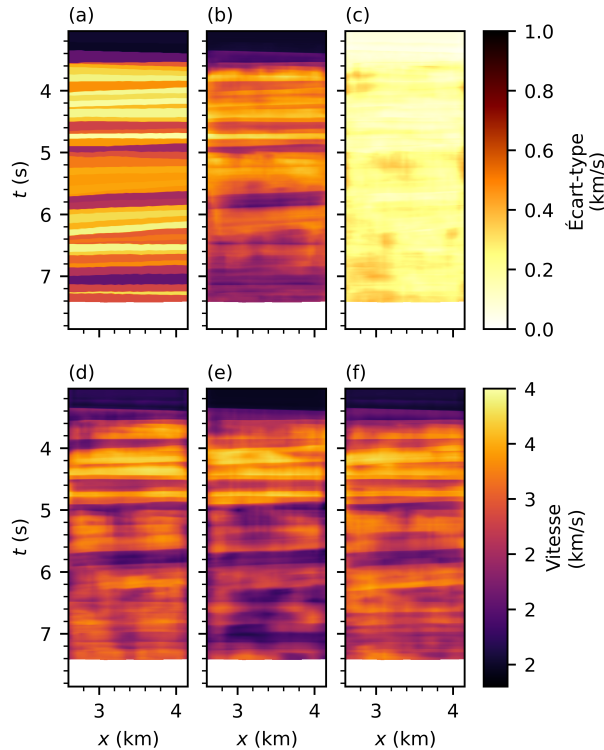


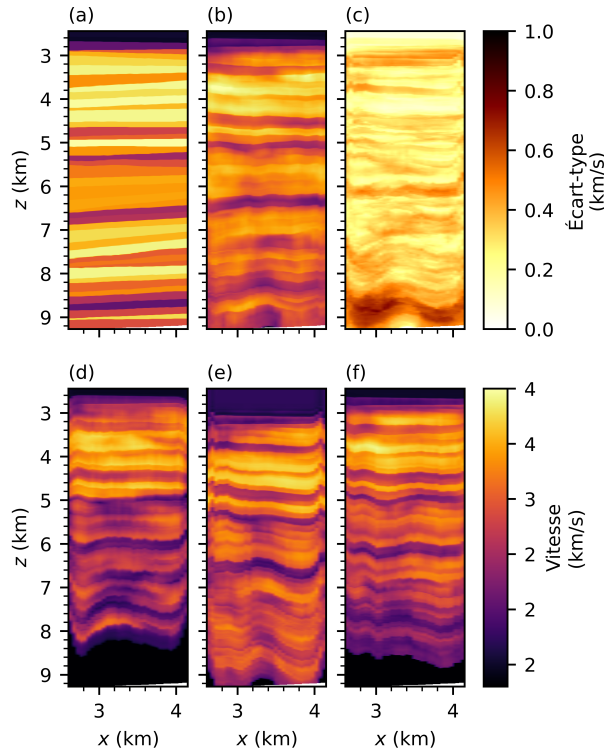
FIGURE 4.6 : Estimations individuelles de  $V(t, x)$  pour le 50<sup>e</sup> percentile

Les estimations affichées sont celles utilisées pour calculer la moyenne à la figure 4.3k. (a) Modèle de vitesse cible. (b) Estimation qui ressemble le plus à la cible, telle que calculée par SSIM (0.877). (c) Écart-type des estimations. (d-f) Les trois estimations qui diffèrent le plus de la cible, par ordre croissant de SSIM (0.758, 0.807, 0.857).

(d) et (e) du tableau 4.2. Pour essayer d’atteindre des justesses plus élevées sans TL, il faudrait disposer d’un plus grand jeu de données 2D et/ou optimiser les hyperparamètres de manière plus approfondie. Le prototypage s’avère prohibitif dans les deux cas, en raison des dizaines de jours de calcul nécessaires, et je n’ai donc pas poursuivi dans cette direction.

Je corrobore les résultats de Fabien-Ouellet et al. (2020) qui indiquent que le TL par le biais d’objectifs hiérarchiques est essentiel pour obtenir de faibles erreurs de prédiction en un temps d’entraînement raisonnable. Par exemple, l’optimisation de la vitesse en profondeur est particulièrement instable. Les entraînements qui se concentrent sur la vitesse d’intervalle en profondeur ( $\alpha_{V(z,x)} \gg 0.1$ ) divergent, ce qui indique que l’apprentissage multitâche est utile pour régulariser l’estimation de modèles de vitesse. Par le passé, j’ai tenté d’estimer  $V(z, x)$  par le biais de RNNs, sans succès. À ce jour, l’aptitude du NN utilisé dans cette thèse à fournir des prédictions dans le domaine de la profondeur demeure limitée.





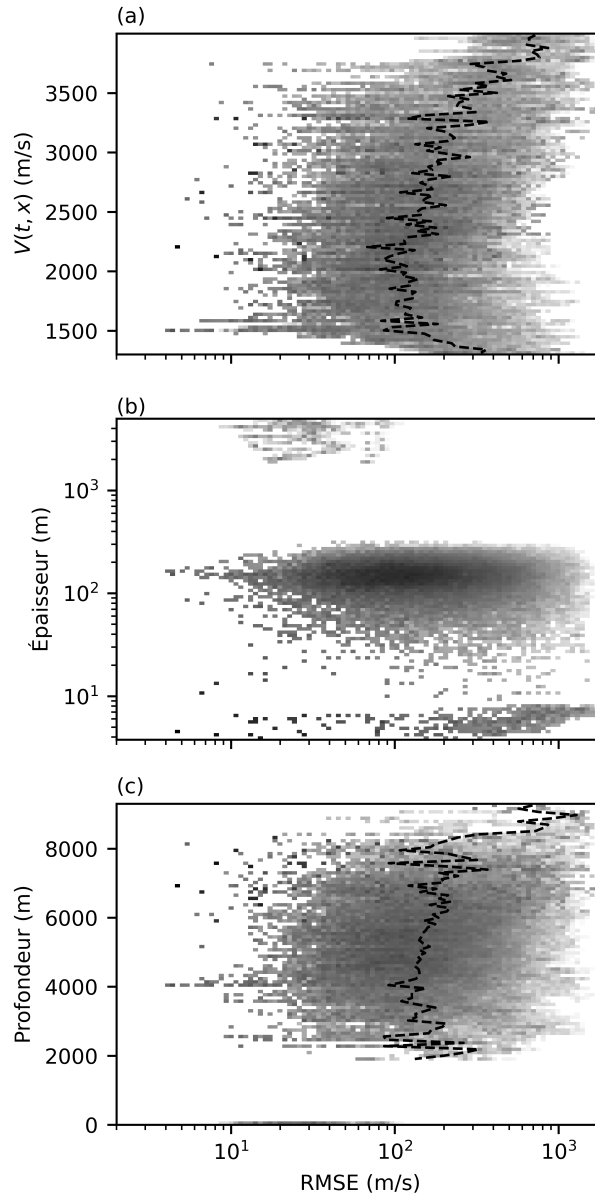
**FIGURE 4.7 : Estimations individuelles de  $V(z, x)$  pour le 50<sup>e</sup> percentile**

Les estimations affichées sont celles utilisées pour calculer la moyenne à la figure 4.3o. (a) Modèle de vitesse cible. (b) Estimation qui ressemble le plus à la cible, telle que calculée par SSIM (0.718). (c) Écart-type des estimations. (d-f) Les trois estimations qui diffèrent le plus de la cible, par ordre croissant de SSIM (0.572, 0.587, 0.629).

#### 4.2.2 Données de terrain

J'ai testé cette méthode sur la section la plus profonde de la ligne 32 des lignes publiques acquises par l'USGS le long de la marge continentale atlantique des États-Unis, qui sont disponibles en ligne (USGS, 1978; Hutchinson et al., 1997).

Comme les paramètres d'acquisition du jeu de données synthétique reproduisent ceux des données de terrain, peu de prétraitement est nécessaire. Les traces sont interpolées à des intervalles de 50 m et un contrôle de gain automatique de l'amplitude RMS est appliqué sur des fenêtres de 21 pas de temps  $\times$  5 CMPs. Un lissage gaussien est appliqué aux modèles de vitesse de sortie sur des fenêtres de 5 pas de temps  $\times$  15 CMPs à des fins de visualisation. Le prétraitement des données réelles et la prédiction ont pris environ 4 minutes pour un seul NN 1D, alors qu'il a fallu environ 8 minutes pour le NN 2D. Les modèles de vitesse résultants sont présentés à la figure 4.9



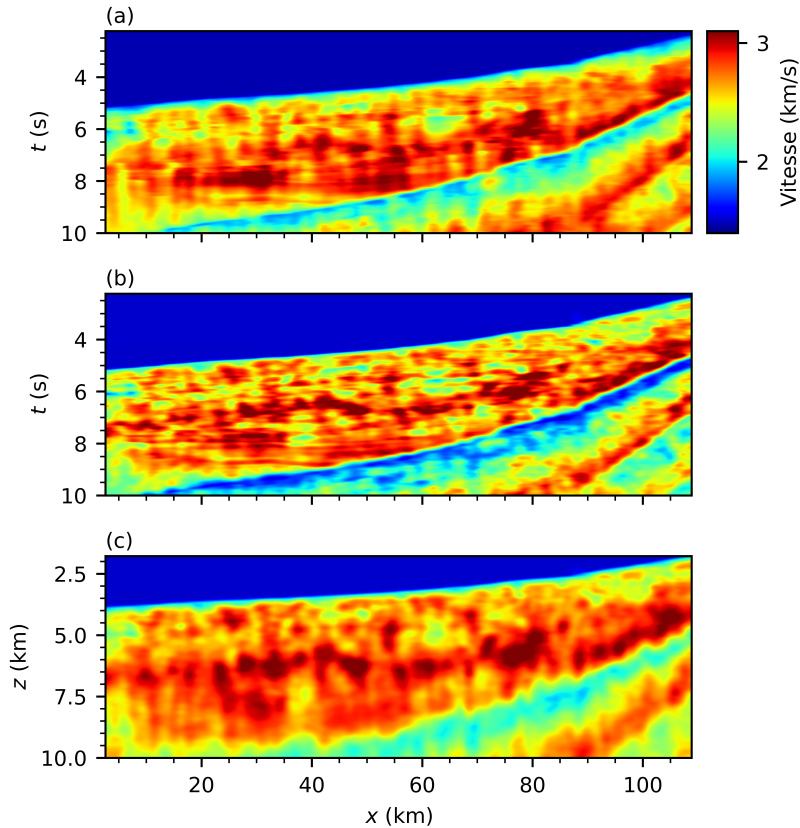
**FIGURE 4.8 : Histogrammes normalisés de la RMSE sur  $V(t, x)$  par couche**

**Le tri est effectué par (a) valeur cible, (b) épaisseur de couche et (c) profondeur de couche. La ligne superposée représente la moyenne.**

et les estimations individuelles de l'ensemble sont présentées aux figures 4.10 et 4.11. La gamme de vitesses estimées est cohérente avec celle de Fabien-Ouellet et al. (2020). Une amélioration apportée par l'estimation de modèles de vitesse 2D est que les interfaces sont généralement plus nettes, alors que la figure 4.9a est plus lisse. Les gammes de vitesses comprises entre 2 000 m/s et 3 000 m/s sont compatibles avec la géologie du plateau de Blake et de la crête externe (Hutchinson et al., 1997). Les couches continues entre 8 s et 9 s avec des vitesses supérieures à

3 000 m/s correspondent à des réflecteurs identifiés comme étant du Jurassique supérieur au Jurassique moyen supérieur (Hutchinson et al., 1997). La vitesse prédite du diapir inférieur gauche est en réalité la vitesse moyenne, la raison apparente étant que les NNs individuels prédisent une valeur aléatoire là où elle n'est pas assez bien contrainte et que la moyenne des prédictions aléatoires sur 14 NNs donne la vitesse moyenne globale. Ce comportement est dû à l'absence de structures complexes dans le jeu de données synthétiques. Un autre défaut de cette approche est l'impact du lien analytique entre  $V(t, x)$  et  $V(z, x)$  sur les données réelles. En comparaison aux prédictions de  $V(z, x)$  sur les données synthétiques, les prédictions sur les données de terrain sont beaucoup plus variables. Cela diminue grandement la qualité du modèle moyenné et laisse croire que la conversion analytique a peu de chances d'être suffisamment généralisable pour être utile en pratique. Enfin, les multiples de surface libre ne sont pas automatiquement ignorés dans l'estimation de la vitesse, comme en témoigne la diminution soudaine de vitesse dans les trois modèles de vitesse. Le jeu de données d'entraînement synthétique ne contenait pas de multiples de surface, ce qui explique pourquoi les NNs n'ont pas été en mesure de les traiter de manière appropriée. Comme l'a souligné Fabien-Ouellet et al. (2020), les solutions seraient d'appliquer des techniques conventionnelles de suppression des multiples de surface ou de concevoir un NN qui soit robuste aux multiples de surface. Bien que j'aurais pu utiliser des données synthétiques qui incorporent des multiples de surface, l'entraînement à la reconnaissance de telles dépendances ponctuelles à longue portée s'est avéré impossible avec le NN actuel (non démontré ici). Je crois que la conception d'une stratégie d'entraînement qui incorpore des données réelles en plus des données synthétiques pourrait permettre d'ignorer les multiples de surface ainsi que d'autres artefacts inhérents aux données réelles.

La figure 4.12 comporte trois regroupements CMP d'entrée provenant des données de terrain, ainsi que les profils de vitesse estimés superposés à la semblance calculée à partir des données d'entrée et des regroupements CMP après correction NMO pour le contrôle qualité. L'estimé moyen suit la tendance des maxima de la semblance, bien que la vitesse soit généralement légèrement surestimée, et l'incertitude englobe la plupart des maxima. La surestimation est due à une valeur particulièrement élevée de la vitesse dans l'eau, à des différences dans les paramètres d'acquisition ou à la divergence sphérique, entre autres. La diminution due aux multiples de surface est



**FIGURE 4.9 : Modèles de vitesse estimés pour la ligne 32 de USGS (1978)**

**(a)  $V(t, x)$  telle que reconstruite par les NNs 1D. (b)  $V(t, x)$  telle que reconstruite par les NNs 2D. (c)  $V(z, x)$  telle que reconstruite par les NNs 2D. La variabilité des estimations est illustrée aux figures 4.10 et 4.11.**

également visible dans tous les profils de vitesse de la figure 4.12. L'estimé de  $V_{\text{RMS}}(t, x)$  aplatit correctement les hyperboles dans les regroupements CMP. Le contrôle qualité sur  $V_{\text{RMS}}(t, x)$  indique que les résultats sont dignes de confiance.

La figure 4.13 comporte la section sommée produite à l'aide du modèle de vitesse RMS obtenu par DL et une section sommée de référence traitée par Hutchinson et al. (1997). Les traces des données d'entrée et de la section sommée de sortie sont normalisées par rapport à leur RMS. La plupart des caractéristiques saillantes de la section sommée de référence avant les arrivées multiples en surface ont pu être reproduites. À  $x = 82$  km, on peut voir le sommet d'un diapir de sel représentatif du talus continental de la Caroline, bien que la majeure partie du dôme soit cachée par les multiples de surface. Les diapirs du coin inférieur gauche et les réflecteurs cohérents après  $t = 7$  s sont clairement visibles. Les réflecteurs cohérents sont généralement associés à des couches avec une plus grande continuité, en comparant avec la figure 4.9b. Des hyperboles de

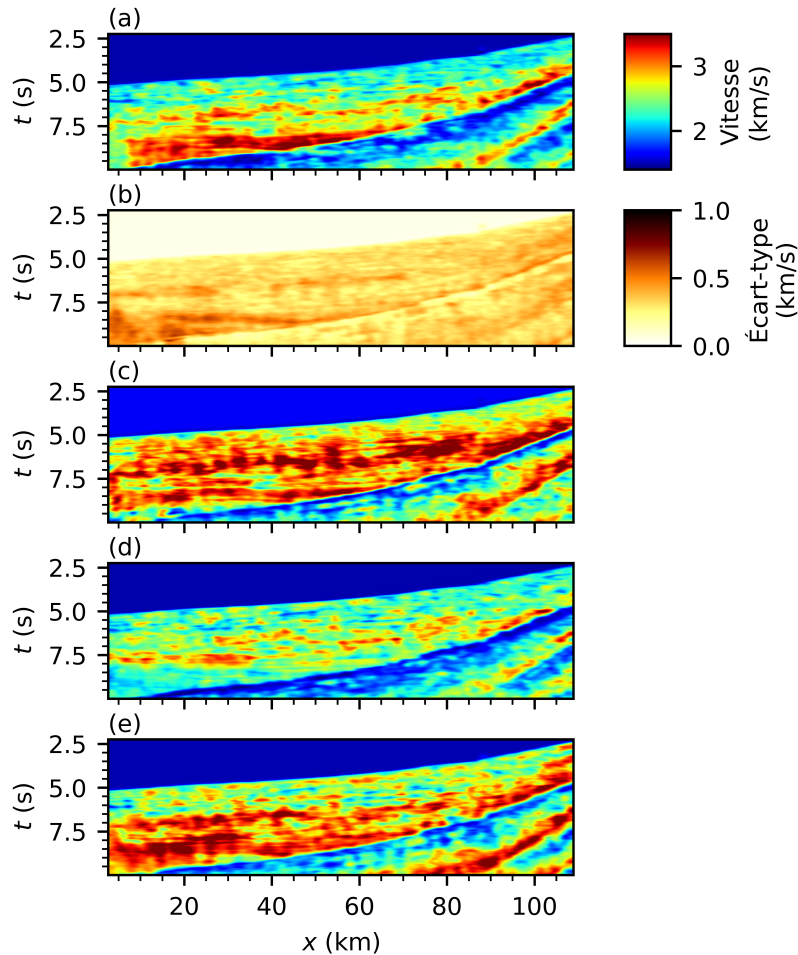


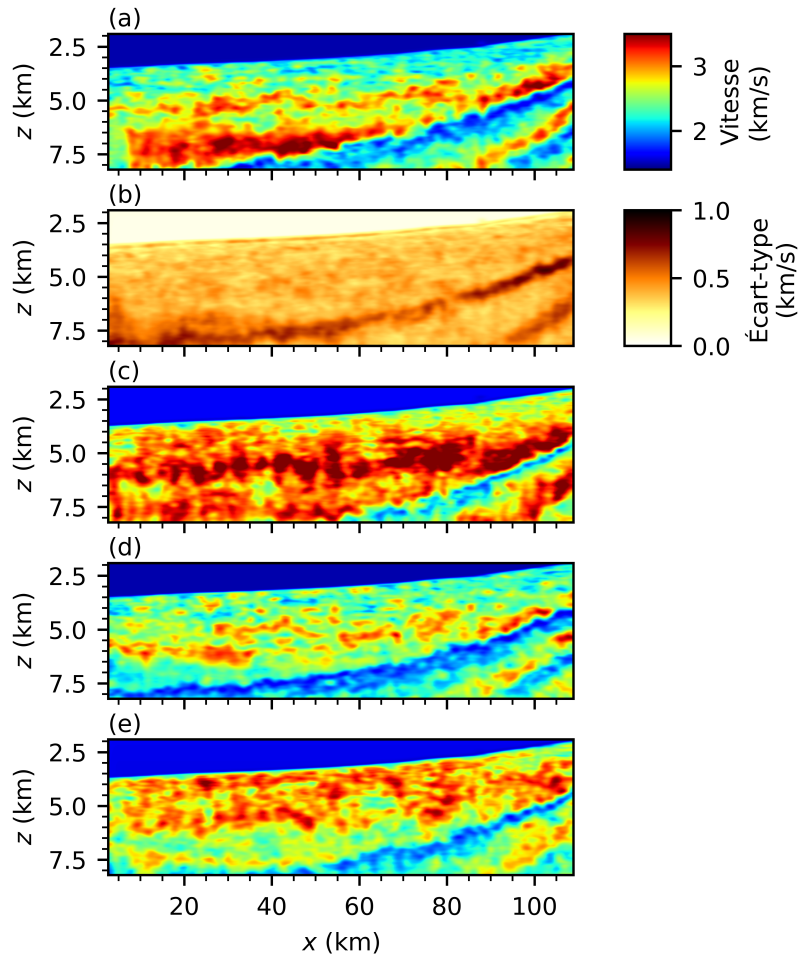
FIGURE 4.10 : Estimations individuelles de  $V(t, x)$  pour les données de terrain

Les estimations affichées sont celles utilisées pour calculer la moyenne à la figure 4.9b. (a) Estimation qui ressemble le plus à la moyenne, telle que calculée par SSIM (0.836). (b) Écart-type des estimations. (c-e) Les trois estimations qui diffèrent le plus de la moyenne, par ordre croissant de SSIM (0.762, 0.769, 0.781).

réflexion non focalisées sont visibles dans la section supérieure, où la vitesse est généralement surestimée. La sommation se dégrade dans la section peu profonde à  $x > 100$  km en raison de la présence de réflexions non cohérentes et d'hyperboles de diffractions fortes (Fabien-Ouellet et al., 2020).

### 4.3 Discussion

Les résultats indiquent que l'estimation de modèles de vitesse pour des milieux où la vitesse varie latéralement peut être considérée comme un problème 1D perturbé, ce qui permet de reconstruire des modèles de vitesse de dimensions réalistes dans un délai d'au plus une semaine à partir de



**FIGURE 4.11 : Estimations individuelles de  $V(z, x)$  pour les données de terrain**

Les estimations affichées sont celles utilisées pour calculer la moyenne à la figure 4.9c. (a) Estimation qui ressemble le plus à la moyenne, telle que calculée par SSIM (0.661). (b) Écart-type des estimations. (c-e) Les trois estimations qui diffèrent le plus de la moyenne, par ordre croissant de SSIM (0.562, 0.611, 0.612).

regroupements CMP bruts avant sommation. Le TL s'est avéré essentiel pour y parvenir. Alors que les travaux précédents se concentrent principalement sur des modèles de vitesse synthétiques qui peuvent ne pas être représentatifs de levés réels (Li et al., 2020; Lin et al., 2019; Yang et al., 2019; Mosser et al., 2020a; Wu et al., 2020; Zhang et al., 2020; Adler et al., 2021), j'ai pu montrer que l'estimation de modèles de vitesse par DL peut être appliquée à un jeu de données de terrain. Un défaut de l'approche de Fabien-Ouellet et al. (2020), dont celle présentée ici s'inspire, est que les regroupements CMP sont traités un par un et qu'ils ne peuvent estimer que des modèles de vitesse à faible pendage qui répondent approximativement à l'hypothèse 1D.

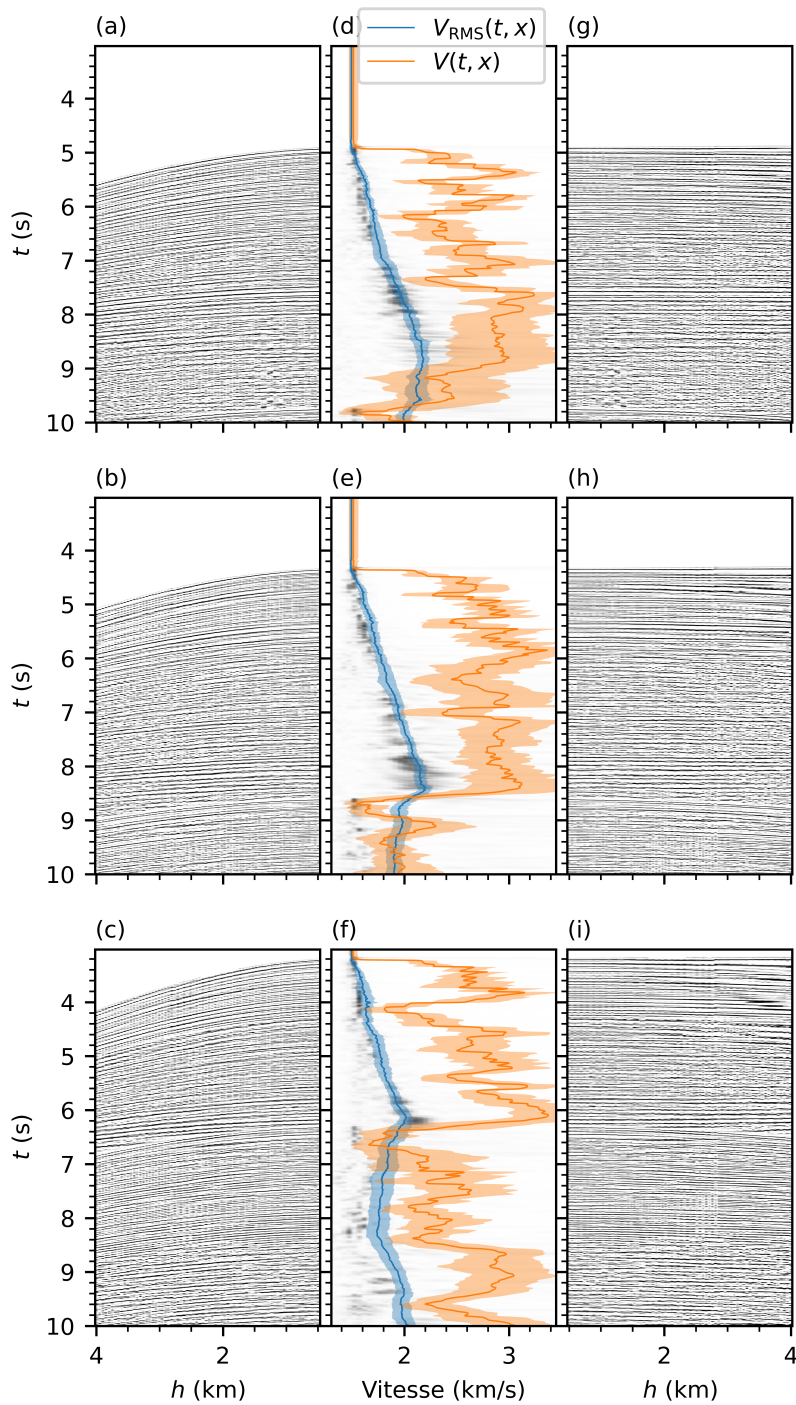


FIGURE 4.12 : Correction NMO et semblance pour les données de terrain

(a-c) Regroupements CMP d'entrée à  $x = 14.6$  km,  $x = 52.1$  km et  $x = 89.6$  km. (d-f) Panneaux de semblance et vitesses estimées. Les régions ombragées représentent l'écart-type sur l'ensemble des prédictions de tous les 14 NNs. (e-g) Regroupements CMP après correction NMO.

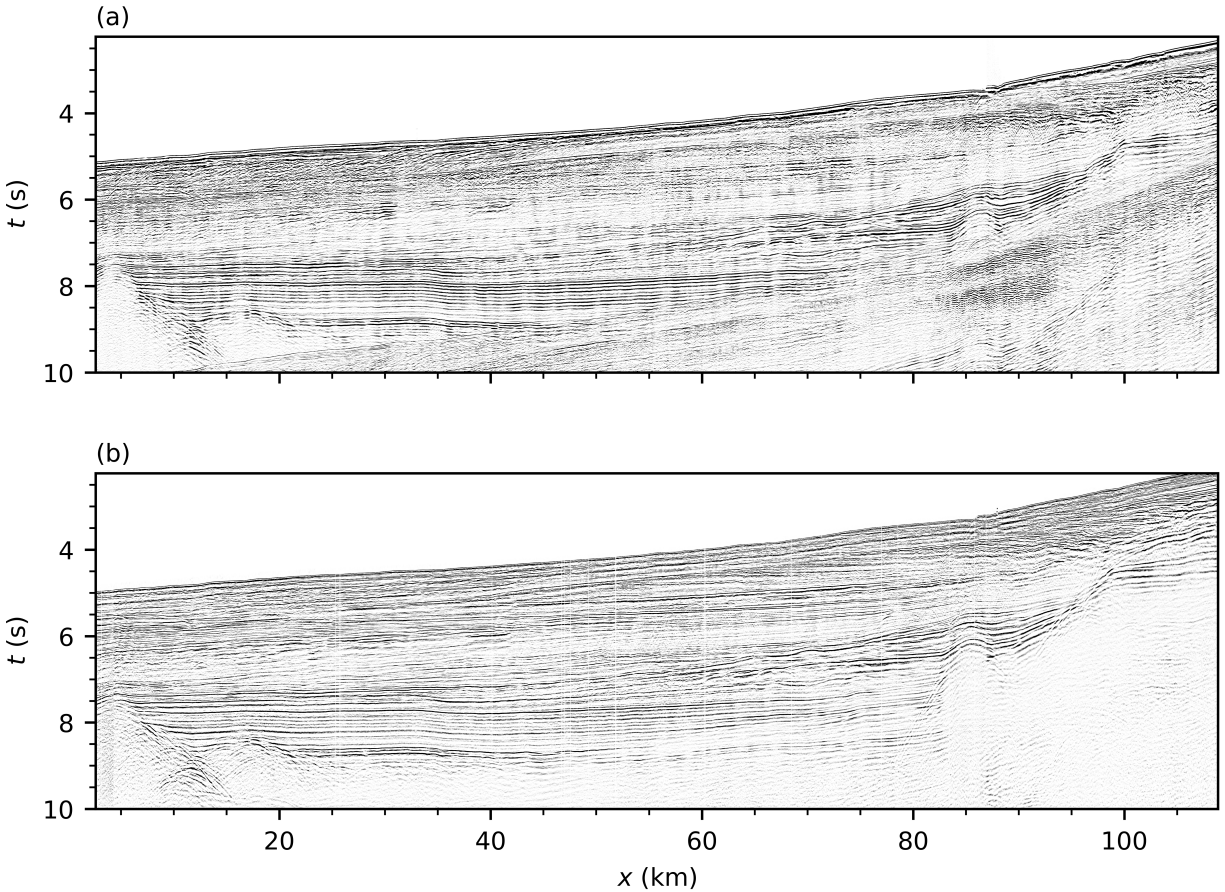


FIGURE 4.13 : Images sismiques traitées de la ligne 32 de USGS (1978).

(a) Coupe sommée associée à l'estimation de modèles de vitesse 2D par DL. (b) Coupe sommée traitée par Hutchinson et al. (1997).

Je tire parti du TL de deux façons : en ayant des objectifs incrémentaux et dans le passage du 1D au 2D. Ces deux stratégies ont amélioré le taux de convergence. Le TL vers le 2D, en particulier, a permis d'utiliser moins de données 2D et de réduire considérablement le temps d'entraînement. Par exemple, utiliser 20 000 modèles 2D comme en 1D nécessiterait 10 fois plus de temps de calcul (ou de GPUs). Entraîner un NN entièrement en 2D pendant 12 000 itérations plutôt que d'en allouer 6 000 à l'entraînement 1D coûterait plus de 40 heures de calculs au lieu d'en coûter 23. En DL, le TL est généralement effectué sur des données d'entraînement ayant la même quantité de dimensions, par exemple entre un domaine source comportant des images, qui sont des tenseurs d'ordre 3 (hauteur, largeur, canaux), et un domaine cible comportant lui aussi des images. Lorsque la taille des dimensions diffère, une stratégie courante consiste à remplacer complètement les couches de tête du NN. Par exemple, c'est le cas lorsque des NNs préentraînés



sur des jeux de données de détection d'objets multiclassés sont réutilisés sur des problèmes avec des quantités différentes de classes. Cependant, le fait d'ajouter carrément de nouvelles dimensions est une caractéristique unique aux problèmes sismiques. La stratégie de TL personnalisée développée dans ce chapitre incorpore toutes les informations acquises lors du préentraînement grâce à l'équivalence géométrique existante entre le traitement 1D et 2D. Bien que cette stratégie se soit avérée efficace pour faire le pont entre le traitement 1D et 2D, l'ajout de dimensions aux filtres existants entraîne une escalade exponentielle des coûts en mémoire, ce qui pourrait s'avérer problématique pour un passage au 3D. Une solution serait de traiter les nouvelles dimensions en intercalant plutôt des filtres 1D traitant la nouvelle dimension indépendamment. Néanmoins, nos résultats indiquent que le TL rend l'entraînement pour les données de plus haute dimension moins prohibitif en termes de temps d'entraînement et performance.

Une fois entraîné, le NN a pu reconstruire trois représentations de modèles de vitesse synthétiques de  $8650 \text{ m} \times 9400 \text{ m}$  en moins de 10 secondes. Utiliser du DL sur des données de terrain ne nécessite peu ou pas de prétraitement et est peu coûteux par rapport aux approches basées sur la physique. Cependant, si l'on considère le temps d'entraînement, un coût de traitement quasi constant ne peut être garanti que pour des levés ayant la même géométrie et la même géologie : l'approche utilisée étant basée sur les données, le réentraînement, coûteux, est nécessaire si l'on souhaite appliquer cette méthode à des données 3D ou à des levés provenant d'endroits différents. Le temps d'entraînement (y compris la génération du modèle synthétique) totalise 5 jours et le NN résultant ne peut être utilisé que lorsque les patrons observés sur les données de terrain (par exemple la forme des ondelettes ou des hyperboles, ou encore le bruit) recourent statistiquement ceux des données d'entraînement, c'est-à-dire que le jeu de données d'entraînement doit être adapté pour avoir les mêmes paramètres d'acquisition et inclure une gamme de structures géologiques réalistes. La perspective de concevoir un NN ou un jeu de données d'entraînement qui conduise à des NNs capables de traiter toutes sortes de géométries et de géologies (c'est-à-dire pour qu'un seul entraînement soit nécessaire) est actuellement improbable, mais rendrait le coût de calcul de l'estimation de modèles de vitesse négligeable.

J'ai montré que cette méthode pouvait être facilement appliquée à des données réelles, à condition que les paramètres d'acquisition correspondent. Peu d'études ont montré le potentiel du DL

pour l'estimation de modèles de vitesse dans des cas d'utilisation réels (Adler et al., 2021), les exceptions étant Zheng et al. (2019) et Fabien-Ouellet et al. (2020). Cependant, Zheng et al. (2019) traitent un petit levé terrestre réel avec des variations latérales minimales de la vitesse et aucun pendage. Néanmoins, leur utilisation de données sismiques en onde complète est une preuve encourageante des capacités de DL. Bien que nos résultats confirment ceux de Fabien-Ouellet et al. (2020), l'applicabilité de DL aux données sismiques réelles reste à vérifier. La justesse du modèle de vitesse estimé à partir des données de terrain a été vérifiée principalement par des moyens qualitatifs. Il s'agit d'une limitation des méthodes basées sur les données dans les applications aux problèmes d'inversion sismique : il n'y a pas de modèles de vitesse réels de référence pour la validation, moins encore pour l'entraînement. La solution généralement privilégiée consiste à utiliser des méthodes basées sur le DL pour améliorer les méthodes de FWI (Lewis et al., 2017; Huang et al., 2018; Campos et al., 2019; Wu et al., 2019c; Zhang et al., 2019). Je soutiens que l'inclusion de données de terrain directement dans la procédure d'entraînement par le biais de GANs ou d'AEs pourrait fournir une garantie supplémentaire d'applicabilité à des cas d'utilisation réels, en contournant complètement l'utilisation de la FWI.

#### **4.4 Conclusion**

En considérant l'effet des structures 2D comme une perturbation des hyperboles sur les regroupements CMP, l'estimation de modèles de vitesse 2D peut être traitée comme un problème de TL et on peut utiliser un NN entraîné à l'estimation de modèles de vitesse 1D comme solution de départ. Cette stratégie permet d'entraîner un NN avec une quantité moindre de modèles 2D, de réduire le temps de calcul par au plus 10 fois et de diminuer l'erreur d'au plus 35% par rapport à un entraînement sans TL. Le TL peut être considéré comme un moyen supplémentaire d'informer un NN sur la physique : le passage de 1D à 2D marque les structures 2D comme perturbateurs. De même, le fait de disposer de plusieurs sorties permet au NN de se concentrer sur les relations physiques entre les représentations. En résumé, sans TL, les exigences en matière de données massives deviendraient prohibitives pour des levés 2D et 3D réalistes. Par extension, le traitement des levés 3D pourrait être considéré comme un problème de TL vers un domaine à plus haute dimension. Par rapport au NN de Fabien-Ouellet et al. (2020), le traitement 2D améliore la continuité latérale

et reconstruit les couches ayant un pendage non nul et estime la vitesse d'intervalle en profondeur, quoique les prédictions dans le domaine de la profondeur manquent encore de stabilité.

La capacité d'un NN à fournir des estimations pour des modèles de vitesse plus diversifiés, incluant par exemple des discontinuités, des pendages plus forts ou des dômes de sel, est étudiée dans les chapitres suivants. Pour l'instant, la méthode proposée dans ce chapitre est encore limitée par le choix des jeux de données d'entraînement et le lien analytique entre le temps et la profondeur. Bien que cette approche de DL ait été conçue pour être appliquée aux données de terrain avec peu de traitement, les NNs ne sont pas entraînés explicitement à ignorer le bruit contenu dans des données sismiques réelles. Par exemple, il peut s'avérer difficile d'ignorer automatiquement les multiples et la variabilité de la signature de la source. Il serait envisageable d'inclure des données de terrain en utilisant des GANs ou des AEs. Néanmoins, l'impact du bruit et des écarts de domaines pour des données synthétiques est étudié dans les prochains chapitres.



## 5 RÉGRESSION PAR CLASSIFICATION

---

L'interprétabilité insuffisante du ML et son incapacité à fournir systématiquement une incertitude ralentissent son adoption (Kim, 2015; Caruana et al., 2015; Stiglic et al., 2020; Kolyshkina et al., 2021). Les exemples les plus évidents d'adoption prudente du DL résident dans les soins de santé, notamment la planification des interventions d'urgence et la prise de décision médicale, où la sécurité est primordiale (Kim, 2015; Caruana et al., 2015; Stiglic et al., 2020; Kolyshkina et al., 2021). Les NNs n'indiquent généralement pas quand et où leurs prédictions sont incertaines, laissant à l'utilisateur le soin de décider de faire confiance ou non aux résultats (Molnar, 2018; Li et al., 2021b). Par conséquent, la communauté du ML montre un intérêt croissant pour les mesures d'interprétabilité et de confiance en DL.

En régression de vitesses par DL, certains auteurs s'appuient sur les méthodes d'ensemble (Opitz et al., 1999) afin d'améliorer les performances dans les zones où les prédictions sont incertaines (Fabien-Ouellet et al., 2020; Kazei et al., 2021; Simon et al., 2023). L'entraînement simultané de plusieurs NNs et l'obtention d'une distribution de prédictions pour chaque position donnée dans l'espace et le temps leur ont permis d'estimer l'incertitude de leurs NNs et de lisser leurs prédictions. Cependant, nul ne sait pourquoi et quand un ensemble de NNs peut fournir des incertitudes véritables (Abdar et al., 2021). De plus, les méthodes d'ensemble sont très coûteuses et ne permettent pas aux NNs d'apprendre explicitement ce qui est incertain dans les résultats. Par exemple, au chapitre précédent, entraîner 14 NNs a requis 13.5 jours GPU de calculs. L'unique but d'utiliser un ensemble était d'obtenir des résultats plus lisses et réalistes sur les données de terrain.

En sismique, les NNs bayésiens convolutifs sont de plus en plus utilisés pour la quantification de l'incertitude, principalement dans les tâches de classification telles que la détection de faciès (Feng et al., 2021a), la détection de canaux (Pham et al., 2020) et la détection de failles (Mosser et al., 2020b; Feng et al., 2021b). Dans l'estimation de modèles de vitesse par DL, les études qui incluent le contrôle de qualité se concentrent principalement sur la caractérisation des erreurs. Lin et al. (2019), Zhang et al. (2020) et Wu et al. (2020) caractérisent qualitativement les erreurs en

utilisant des exemples hors distribution, c'est-à-dire en ajoutant des patrons absents des données d'entraînement et en observant leur impact sur les prédictions. Aucune étude de quantification de l'incertitude n'a encore été réalisée dans le domaine de l'estimation du modèle de vitesse par DL, probablement parce que moins d'attention a été accordée à la quantification de l'incertitude pour les problèmes de régression (Abdar et al., 2021; Li et al., 2021b), les techniques bayésiennes étant plus simples à appliquer aux problèmes de classification. De plus, en détection des failles, les probabilités associées aux classes positives et négatives sont déjà couramment interprétées comme un degré de confiance (Feng et al., 2021b). Dans les problèmes de classification, en général, la réponse *softmax* brute est parfois utilisée pour l'estimation de l'incertitude, puisqu'elle peut être interprétée comme un degré de confiance en chacune des classes (Cordella et al., 1995; Geifman et al., 2019).

En considérant cela, je propose la classification sur les vitesses discrétisées en intervalles comme alternative à la régression de vitesses. Plus spécifiquement, dans ce chapitre, je vise à montrer que reformuler la régression en tant que problème de classification facilite l'entraînement (van den Oord et al., 2016a,b), mais fournit également un moyen d'avoir des distributions de confiance sur des vitesses à un coût inférieur à celui des ensembles. Formuler un problème de régression en classification n'est pas nouveau en ML (Weiss et al., 1995; Torgo et al., 1997), mais cette stratégie mériterait d'être démocratisée à l'estimation de modèles de vitesse par DL, où les ensembles sont couramment utilisés. De plus, je vise à montrer que la confiance telle qu'interprétée par les distributions de confiance sur les vitesses discrétisées peut être utilisée pour créer une métrique de méfiance pertinente et peut servir de remplacement aux ensembles en régression de vitesse par DL. Je montrerai comment la métrique de méfiance conçue s'applique à des modèles de vitesse synthétiques 1D de la même distribution que le jeu d'entraînement, mais aussi à des exemples hors distribution auquel est ajouté un bruit cohérent. Bien que les techniques bayésiennes puissent être utilisées pour fournir des incertitudes appropriées, l'alternative proposée est simple à mettre en œuvre (seule l'extrémité du NN doit être modifiée) et permet des distributions de confiance arbitrairement multimodales.

Le matériel de ce chapitre est soumis chez *Computers & Geosciences* :

Simon J, Fabien-Ouellet G, Gloaguen E (soumis le 16 décembre 2022). Quality control in deep learning and confidence quantification : seismic velocity regression through classification. *Computers & Geosciences*.

Le code source est disponible en ligne (Simon, 2022b).

## 5.1 Méthodologie

### 5.1.1 Estimer la vitesse par classification

L'objectif est de modifier l'approche de DL exposée au chapitre 3 de manière à permettre au NN d'exprimer sa confiance en les prédictions. L'approche choisie consiste à réécrire le problème de régression en un problème de classification, où les classes correspondent à une discrétisation de la variable de sortie d'origine. En d'autres termes, le domaine continu des vitesses  $\mathcal{V} = [V_{\min}, V_{\max}]$  est divisé en intervalles uniformes  $n_v$  de longueur  $\Delta V$ . Les vitesses admissibles deviennent alors les valeurs au centre de chaque intervalle, c'est-à-dire  $(V_{\min} + \frac{\Delta V}{2}, V_{\min} + \frac{3\Delta V}{2}, \dots, V_{\max} - \frac{\Delta V}{2}) = \mathcal{V}'_{n_v}$ . Le processus de conversion des vitesses du domaine continu aux valeurs discrètes admissibles est appelé « groupement en classes » (*binning*) et les intervalles sont appelés « classes » (*bins*). L'opérateur de groupement  $\mathcal{B}_{n_v} : \mathcal{V} \rightarrow \mathcal{V}'_{n_v}$  discrétise la variable continue  $V(t)$  en remplaçant chaque valeur des données d'entrée par la vitesse admissible la plus proche dans le domaine discret  $\mathcal{V}'_{n_v}$ . Au lieu de tenter de trouver la vitesse correcte par régression, l'objectif est maintenant de prédire quelles sont les chances que chaque classe soit la bonne, ce qui est un problème de classification, et la sortie du NN devient la confiance en chacune des classes. Dans ce chapitre, j'appelle le NN d'origine un « régresseur » et le NN refactorisé un « classificateur ».

La classification comme alternative à la régression en DL est inspirée de WaveNet (van den Oord et al., 2016a) et de PixelRNN (van den Oord et al., 2016b) et elle a été initialement proposée par Weiss et al. (1995) et Torgo et al. (1997). Cette stratégie s'est avérée efficace pour les séquences temporelles (van den Oord et al., 2016a). Les distributions *softmax* discrètes se sont avérées faciles à apprendre (van den Oord et al., 2016a,b), la raison étant que le chevauchement entre les distributions estimées et attendues est plus informatif pour le NN que la RMSE conventionnelle

en ce sens que le chevauchement donne des informations d'erreur sur l'entièreté du spectre des vitesses, tandis que la RMSE donne seulement l'amplitude et la direction de l'erreur. En d'autres termes, le résultat est plus riche, ce qui permet de mieux cibler les lacunes pendant l'entraînement. La classification a l'avantage sur les méthodes bayésiennes et variationnelles de permettre des distributions arbitraires (van den Oord et al., 2016a,b), lesquelles peuvent être utiles lorsque plus d'une valeur de vitesse est acceptable, par exemple exactement à la position d'une faille ou d'une interface.

En pratique, pour effectuer de la régression par classification, le problème de régression doit être modifié de quatre manières :

- Le tenseur  $n_t \times 1 \times 1$  de sortie (la vitesse estimée  $\hat{V}(t)$ ) doit être remplacé par un tenseur  $n_t \times 1 \times n_v$  (la distribution de confiance discrétisée  $\hat{p}(v, t)$ ), ce qui signifie que le tenseur de sortie de la dernière couche doit être étendu le long d'une nouvelle dimension  $v$  et que chaque valeur de vitesse scalaire dans le temps est maintenant un vecteur de confiance;
- La sortie non normalisée  $\hat{q}$  de la dernière couche du NN doit être normalisée à l'aide de la fonction d'activation *softmax*

$$\hat{p}(v, t) = \frac{\exp\{\hat{q}(v, t)\}}{\sum_{w \in \mathcal{V}'} \exp\{\hat{q}(w, t)\}}; \quad (5.1)$$

- Les étiquettes doivent être encodées *one-hot* en classes, c'est-à-dire que la cible  $V(t)$  doit être remplacée par

$$p(v, t) = \begin{cases} 1 & \text{si } v = \mathcal{B}_{n_v}\{V(t)\} \\ 0 & \text{sinon} \end{cases}; \quad (5.2)$$

- La fonction coût doit être remplacée par une fonction permettant de comparer des distributions, par exemple l'entropie croisée catégorielle

$$L(\hat{p}, p) = - \sum_t \sum_{v \in \mathcal{V}'} p(v, t) \log(\hat{p}(v, t)). \quad (5.3)$$

Les modifications précédentes sont résumées à la figure 5.1. Elles sont simples à mettre en œuvre en ce sens qu'elles ne nécessitent que la modification de l'extrémité du NN et peuvent être appliquées à n'importe quelle architecture et à tout problème de reconstruction de modèle ou de



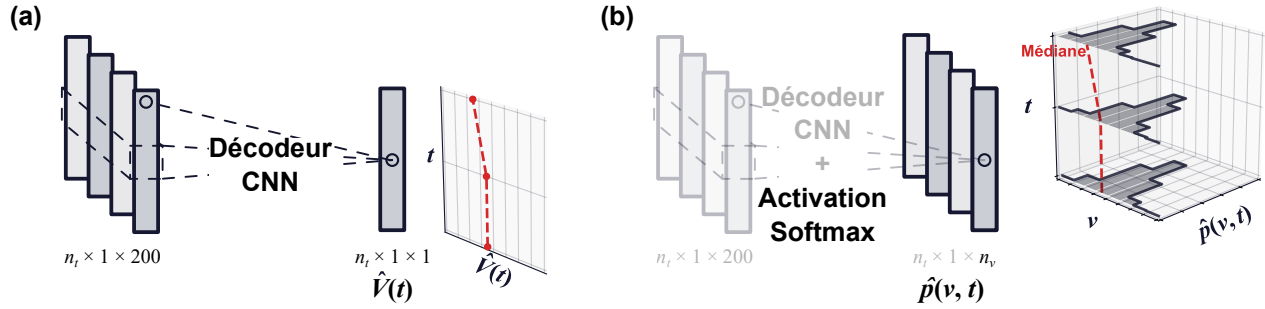


FIGURE 5.1 : Modifications à appliquer à l'extrémité du régresseur pour en faire un classificateur

(a) L'extrémité d'un régresseur 1D, inspiré du NN présenté initialement à la figure 3.5 du chapitre 3. (b) L'extrémité du classificateur correspondant. Les tenseurs et les couches qui restent les mêmes sont en gris. Le tenseur final  $V(t)$  est remplacé par  $\hat{p}(v, t)$ , qui est une distribution de la confiance en chacune des vitesses données (un vecteur) pour chaque pas de temps. La médiane de la distribution est choisie comme vitesse prédite.

régression à  $N$  dimensions (c'est-à-dire que la méthode ne se limite pas à la régression de vitesses sismique). Pour plus de simplicité, seul le cas 1D est traité, mais cette méthode peut être étendue aux modèles de vitesse 2D ou 3D et au domaine de profondeur de la même manière qu'au chapitre 4.

Les résultats finaux (c'est-à-dire les modèles de vitesse prédits) sont obtenus en réduisant à un scalaire représentatif la distribution *softmax* à chaque point donné dans le temps et l'espace. J'ai comparé le mode, la moyenne et la médiane de la distribution et la médiane donnait la meilleure justesse (non illustré ici), la médiane permettant un bon compromis entre lissage et prédiction de valeurs extrêmes. Mathématiquement, la médiane de la distribution est calculée comme suit :

$$\hat{V}(t) \text{ telle que } \sum_{v=V_{\min}+\frac{\Delta V}{2}}^{\hat{V}(t)} \hat{p}(v, t) \geq \frac{1}{2} \text{ et } \sum_{v=V_{\min}+\frac{\Delta V}{2}}^{\hat{V}(t)+\Delta V} \hat{p}(v, t) < \frac{1}{2}. \quad (5.4)$$

Enfin, pour estimer quantitativement l'étendue des distributions *softmax*, celles-ci doivent être réduites à un scalaire (la métrique de méfiance). J'ai choisi l'écart-type comme mesure de méfiance. L'écart-type est

$$\hat{\sigma}(t) = \sqrt{n_v^{-1} \sum_{v \in \mathcal{V}} \hat{p}(v, t) (v - \hat{\mu}(t))^2}, \quad (5.5)$$

où  $\hat{\mu}(t) = \sqrt{n_v^{-1} \sum_{v \in \mathcal{V}} \hat{p}(v, t) v}$ .

D'autres mesures de méfiance pourraient être l'entropie (Shannon, 1948) ou l'écart interquartile,

par exemple, mais celles-ci n'ont pas été comparées, l'objectif étant de montrer que les classificateurs sont plus informatifs que les régresseurs sous une même métrique.

### 5.1.2 Jeu de données

Le NN est entraîné à prédire la vitesse en fonction de regroupements de traces sismiques d'entrée à l'aide de 20 000 modèles de vitesse 1D, lesquels sont conçus de la même façon qu'au chapitre 4. La génération du jeu de données prend 9 heures avec 16 GPUs (9 jours GPU). J'utilise dans ce chapitre des GPUs NVIDIA A100 (modèle 40 GB).

## 5.2 Résultats et discussion

### 5.2.1 Quantité de classes

Afin de déterminer la quantité optimale de classes, j'ai entraîné un ensemble de 4 NNs pour chaque quantité de classes  $n_v \in (8, 16, 24, 32, 48, 64)$ . Les ensembles ont été utilisés pour réduire l'impact du hasard. La RMSE obtenue et l'entropie croisée calculée à l'aide de l'équation 5.3 sont indiquées dans le tableau 5.1. Les résultats sont obtenus en faisant d'abord la moyenne des prédictions de tous les NNs pour un  $n_v$  donné (les estimés d'ensemble) pour chaque exemple d'un jeu de données test composé de 200 exemples, puis en calculant les métriques des estimés d'ensemble. La RMSE minimale des estimés d'ensemble est atteinte à  $n_v \in [24, 48]$ , et  $n_v = 48$  (ou  $\Delta V = 56.25$  m/s) a été choisi pour permettre des sorties plus riches. En effet, la discrétisation du domaine de sortie entraîne une perte d'information (Fedorov et al., 2009) et l'augmentation de l'entropie, qui est causée par une distribution cible plus étroite, indique qu'il y a plus d'information à utiliser lorsque  $n_v = 48$ . Par conséquent,  $n_v = 48$  est gardé constant tout au long du reste de ce chapitre.

### 5.2.2 Comparaison des régresseurs et des classificateurs

J'ai entraîné un ensemble de 16 régresseurs et un ensemble de 16 classificateurs. La fonction coût pour les régresseurs et les classificateurs est illustrée à la figure 5.2. Le coût plus élevé pour les

$n_v$	RMSE (m/s)	Entropie croisée
8	192	0.210
16	218	0.502
24	<b>183</b>	0.546
32	<b>184</b>	0.638
48	<b>185</b>	0.836
64	199	0.886

TABLEAU 5.1 : Optimisation de la quantité de classes du classificateur

Les métriques sont calculées entre la vitesse cible et la distribution de confiance des estimés d'ensemble et sont moyennées sur des exemples de 200. La RMSE minimale est atteinte à  $n_v \in [24, 48]$ , comme indiqué.

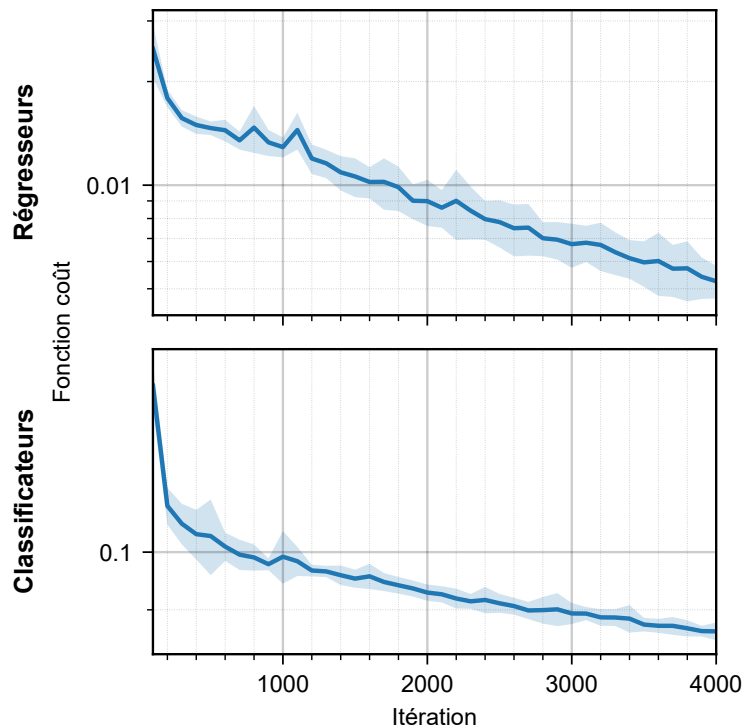


FIGURE 5.2 : Fonction coût associée à  $V(t)$  pendant l'entraînement

L'écart-type sur 16 NNs est indiqué par l'ombrage. Les régresseurs utilisent la RMSE comme fonction coût, alors que les classificateurs utilisent l'entropie croisée.

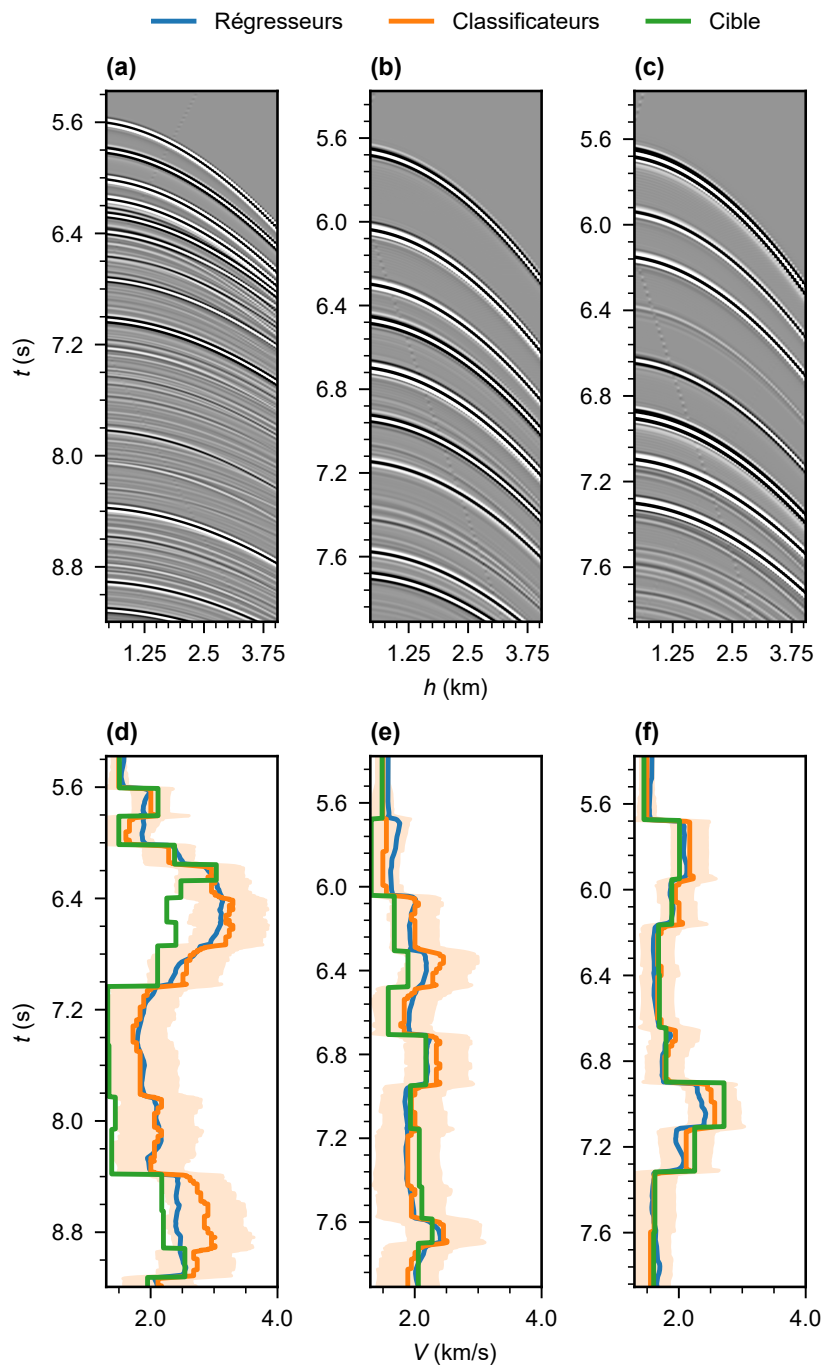
classificateurs est dû à l'utilisation de l'entropie croisée plutôt que de la RMSE. Il faut 30 minutes pour entraîner un seul régresseur sur 2 GPU, alors qu'il faut 65 minutes pour un classificateur. Par conséquent, l'entraînement d'un ensemble de 16 régresseurs prend 16 heures GPU, alors qu'il faut environ 2 heures GPU pour un seul classificateur et environ 34.5 heures GPU pour 16 d'entre eux.

J'ai calculé la performance de l'ensemble de régresseurs et de l'ensemble de classificateurs sur le jeu de données test de 200 modèles. Trois exemples représentatifs et les prédictions associées

sont présentés à la figure 5.3. La RMSE moyenne sur le jeu de données test est de 213 m/s pour les régresseurs et de 181 m/s pour les classificateurs, ce qui indique que les régresseurs sont moins justes.

La figure 5.4 est une comparaison détaillée des prédictions présentées à la figure 5.3f. La différence RMS entre les résultats de l'ensemble de régresseurs et ceux de l'ensemble de classificateurs est de 86 m/s, elle est de 105 m/s entre les classificateurs individuels, et de 72 m/s entre les classificateurs et leur moyenne d'ensemble. La dernière différence est faible : par conséquent, les classificateurs communiquent sensiblement la même information seuls ou en ensemble.

L'écart-type moyen des prédictions des classificateurs individuels sur le jeu de données test est de 387 m/s, tandis que celui de l'ensemble est de 460 m/s : l'utilisation d'ensembles augmente la métrique de méfiance de 73 m/s. En utilisant des régresseurs, l'écart-type diminue à 223 m/s, ce qui indique que les résultats sont moins étalés que ceux des classificateurs. En d'autres termes, les régresseurs sont moins justes, mais plus fidèles, ce qui conduit à un biais indétectable. Lorsque la vitesse est plus extrême, il y a sous-estimation autant pour les régresseurs que pour les classificateurs, mais la vitesse cible est plus souvent à l'intérieur d'un seul écart-type de la prédiction lorsqu'il s'agit de classificateurs, qu'ils soient utilisés seuls ou dans un ensemble. Ce constat est quantifié à travers la figure 5.5 : pour les régresseurs, les prédictions de vitesse sont à moins de 1 écart-type de la vitesse cible 49.9% du temps, alors qu'il s'agit de 77.4% pour un seul classificateur et de 89.7% pour un ensemble de classificateurs. Cette augmentation de la couverture pour les classificateurs est due à l'augmentation de la justesse et de l'écart-type moyen. Cependant, l'écart-type sur l'ensemble de régresseurs n'est presque jamais assez grand pour englober les vitesses extrêmes. Ainsi, les régresseurs sont trop confiants par rapport aux classificateurs. Enfin, les vitesses extrêmes sont corrélées à une augmentation de l'écart-type dans les distributions *softmax*, ce qui peut s'expliquer physiquement par le fait que la résolution diminue avec l'augmentation des vitesses sismiques.



**FIGURE 5.3 : Sous-ensemble des prédictions du jeu de données test**

Les exemples sont triés par percentiles, ceux-ci étant calculés à partir de la RMSE sur  $V(t)$  pour les classificateurs sur le jeu de données test de 200 modèles. (a-d) 90<sup>e</sup> percentile (319 m/s), (b-e) 50<sup>e</sup> percentile (152 m/s), et (c-f) 10<sup>e</sup> percentile (86 m/s), (a-c) Regroupement CMP d'entrée. (d-f) Vitesse cible et prédictions. L'aire orangée représente l'intervalle de confiance égale à 1 écart-type autour de la médiane.

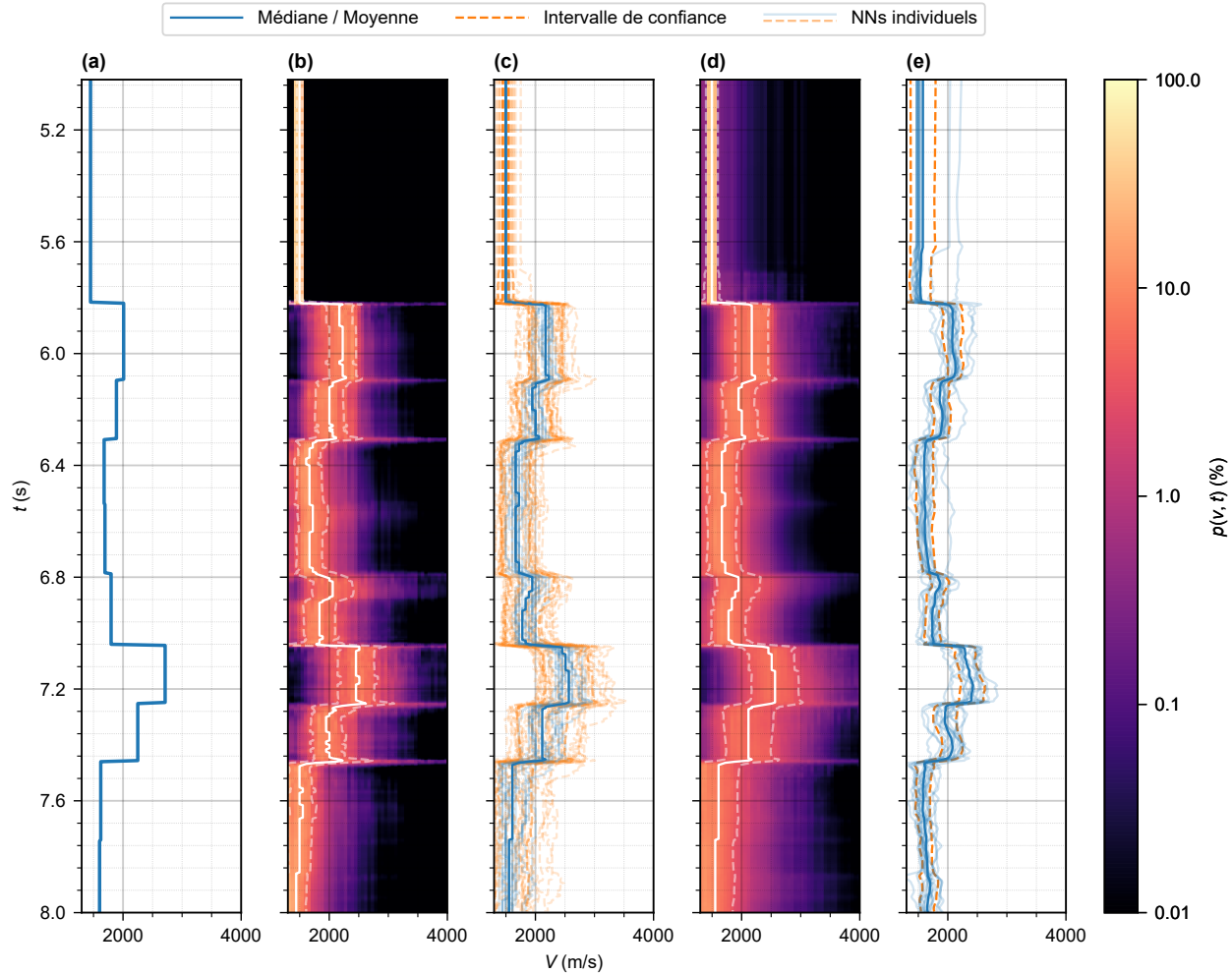


FIGURE 5.4 : Comparaison détaillée des prédictions individuelles de la figure 5.3f

(a) Vitesse cible. (b) La prédiction d'un seul classificateur, superposé à sa distribution *softmax*. (c) L'estimé d'ensemble superposé à la prédiction de chacun des classificateurs. (d) L'estimé d'ensemble pour les classificateurs. (e) L'estimé d'ensemble superposé à la prédiction de chacun des régresseurs. Pour les classificateurs, la ligne pleine est la médiane de la distribution et les lignes pointillées sont des intervalles de 1 écart-type autour de la médiane. Pour les régresseurs, la ligne pleine est la moyenne et les lignes pointillées sont des intervalles de 1 écart-type autour de la moyenne.

### 5.2.3 Analyse de sensibilité

Afin d'évaluer l'utilisabilité de l'écart-type sur la distribution *softmax* en tant que mesure de méfiance, je souhaite voir comment le bruit dans les données d'entrée affecte la confiance et, plus précisément, s'il se traduit par une augmentation de la méfiance, comme on peut s'y attendre. J'ai utilisé un seul classificateur tout au long de cette section pour montrer comment les classificateurs peuvent transmettre de l'information utile sur le contrôle de qualité même s'ils ne sont pas utilisés en ensemble.

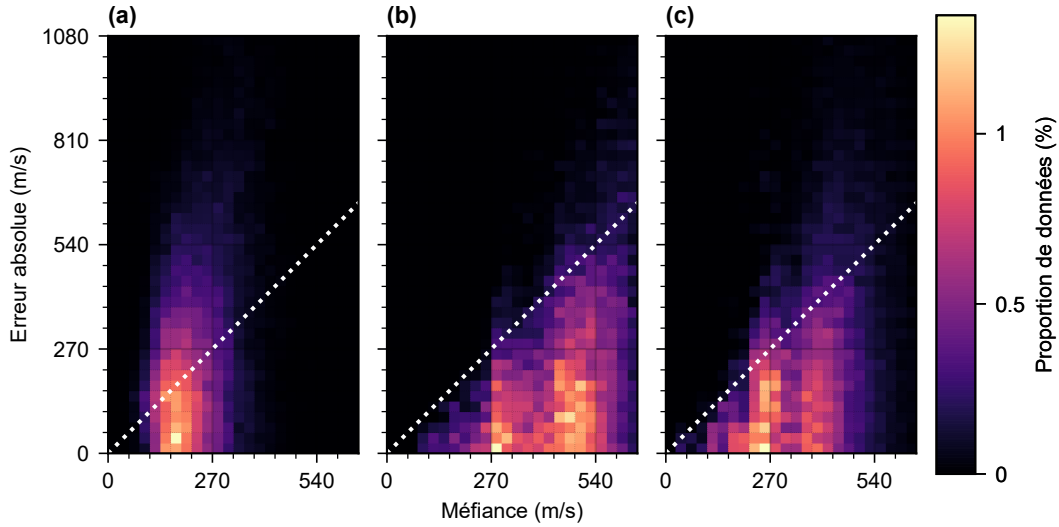


FIGURE 5.5 : Histogramme des paires erreur-médiance

La ligne blanche indique où la médiance est égale à l'erreur. (a) Ensemble de régresseurs. (b) Ensemble de classificateurs. (c) Un seul classificateur.

### 5.2.3.1 Sensibilité au bruit blanc

J'ai ajouté du bruit blanc de variance  $N^2$  au signal d'entrée  $d(t)$  d'un exemple synthétique du jeu de données test de telle façon que le rapport signal sur bruit est  $S/N = \frac{n_t^{-1} \sum_{t=1}^{n_t} d^2(t)}{N^2} \in \left(1, \frac{1}{4}, \frac{1}{16}, \frac{1}{64}\right)$ . Les prédictions sont montrées à la figure 5.6. Dans le fond marin, les prédictions et les intervalles de confiance sont comparables lorsque  $S/N = 1$  ou  $S/N = \frac{1}{4}$ . La médiane et l'écart-type ne semblent pas affectés par la présence de quantités raisonnables de bruit blanc. Cependant, à mesure que  $S/N$  diminue, la distribution de confiance devient de plus en plus instable et la médiane moyenne  $\bar{\sigma}$  augmente, de 299 m/s pour  $S/N = 1$  à 533 m/s pour  $S/N = \frac{1}{64}$ . Dans la première couche, en particulier, l'absence de signal provoque une médiane élevée même pour  $S/N = 1$ . Pour un bruit très élevé, la vitesse prédite dans la première couche tend à être très élevée, ce qui pourrait être associé à la fréquence élevée du bruit blanc. Il semble que, en l'absence de signal, le classificateur puisse associer le bruit à des réflexions très incertaines et de grande vitesse. La médiane semble être accumulée par le LSTM et propagée dans les couches inférieures par la mémoire à long terme. Néanmoins, les classificateurs sont généralement robustes au bruit blanc jusqu'à  $S/N = 1$  et la métrique de médiance est utile pour estimer la confiance en présence d'un bruit blanc fort.

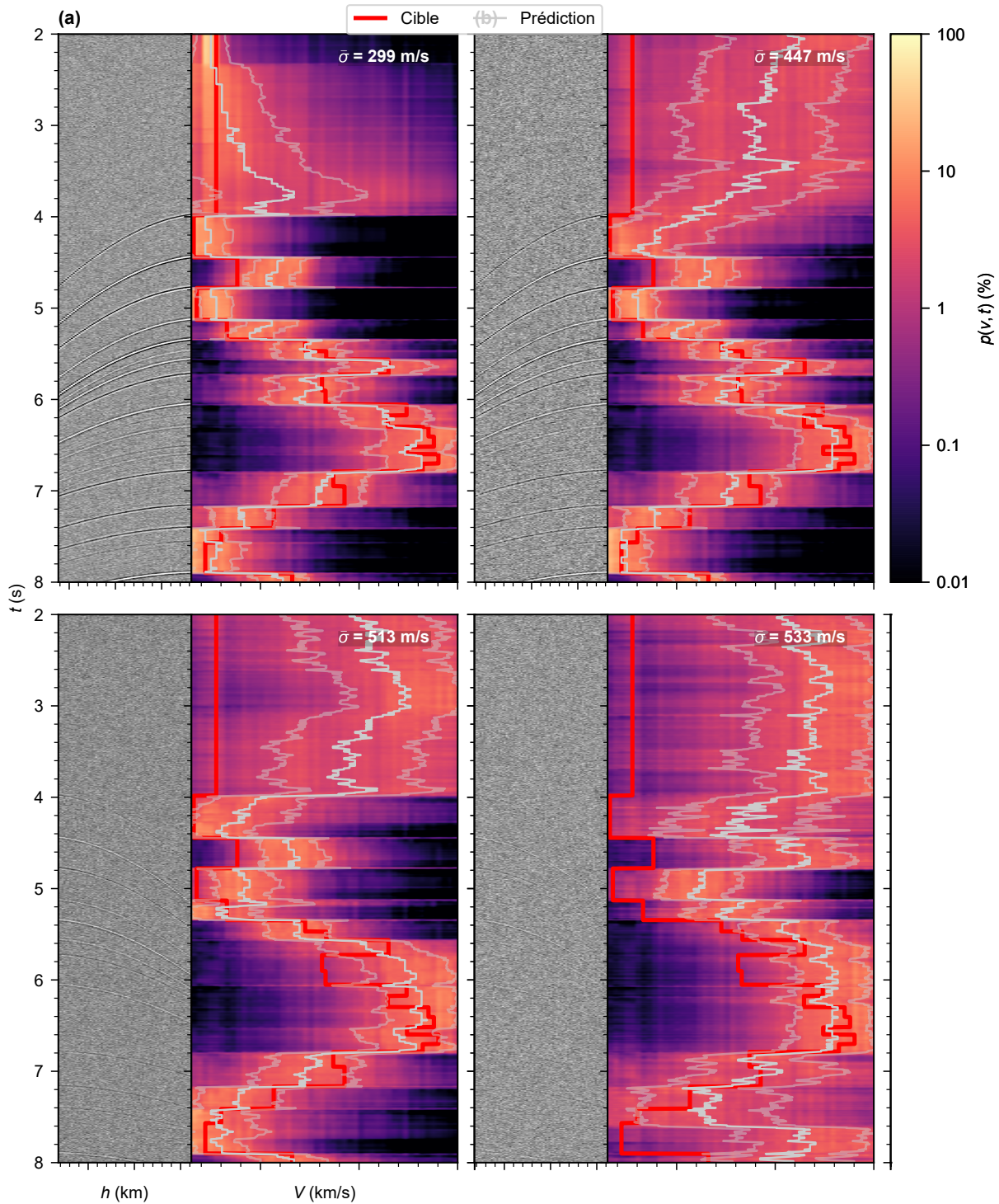


FIGURE 5.6 : Impact du bruit blanc sur les prédictions et la métrique de méfiance

(Panneaux de gauche) Regroupement CMP d'entrée. (Panneaux de droite) Prédiction d'un seul classificateur et valeur moyenne de l'écart-type  $\bar{\sigma}$ . (a)  $S/N = 1$ . (b)  $S/N = \frac{1}{4}$ . (c)  $S/N = \frac{1}{16}$  (d)  $S/N = \frac{1}{64}$ .



### 5.2.3.2 Sensibilité au bruit cohérent : exemples hors distribution

Étant donné que les classificateurs ne sont affectés que par un fort bruit blanc, l'analyse de la façon dont les prédictions et la mesure de méfiance sont affectées par le bruit cohérent semble plus pertinente pour les applications de terrain. J'ai généré des exemples 2D à deux couches avec des changements dans les pendages et avec la présence de discontinuités verticales, puis la réponse sismique associée au CMP au centre de l'exemple a été calculée et j'ai fourni le regroupement CMP au classificateur. Parce qu'ils sont 2D, ces exemples sont différents de ceux vus par le classificateur pendant l'entraînement. J'ai choisi d'observer l'effet de ces structures géologiques en particulier, parce que les pendages et les failles (sans pour autant qu'elles soient verticales) sont courants et parce que le régresseur est affecté négativement par la présence de forts pendages et de discontinuités, comme rapporté au chapitre 4.

L'effet des pendages  $\theta \in (0^\circ, 5^\circ, 10^\circ, 20^\circ)$  sur la distribution de confiance est illustré à la figure 5.7. Au fur et à mesure que le pendage augmente, la vitesse prédite reste correcte jusqu'à ce que  $\theta$  atteigne  $20^\circ$ . À ce stade, la vitesse prédite est inférieure à celle attendue, ce qui est cohérent avec le fait que la vitesse apparente diminue à mesure que l'angle de pendage augmente (à  $20^\circ$ ,  $\cos(20^\circ) = 94\%$  de la vitesse réelle). La méfiance augmente également de  $\bar{\sigma} = 160$  m/s pour  $\theta = 10^\circ$  à  $\bar{\sigma} = 167$  m/s pour  $\theta = 20^\circ$ , ce qui pourrait indiquer que la signature ou l'amplitude de la réflexion semble erronée pour le NN. Cependant, l'augmentation de la méfiance n'est pas aussi drastique que celle du biais introduit (environ 350 m/s), ce qui indique que la méfiance ne peut être directement et systématiquement reliée à l'erreur.

L'effet de discontinuités extrêmes dans le modèle de vitesse sur les prédictions est montré à la figure 5.8. J'ai ajouté une discontinuité verticale avec un décrochement de 500 m au modèle à deux couches de façon à observer l'impact de la discontinuité sur les prédictions aux CMPs situés autour de son emplacement. Ce test est basé sur le modèle de Marmousi, lequel comporte des discontinuités ayant des pendages atteignant jusqu'à  $70^\circ$  et des décrochements atteignant jusqu'à 600 m (Brougois et al., 1990; Martin, 2004). Les prédictions à des endroits éloignés des discontinuités verticales sont légèrement surestimées, mais dans l'intervalle de confiance. La distribution de confiance s'élargit aux arrivées tardives, par exemple à 4.2 s sur la figure 5.8b. Ces arrivées

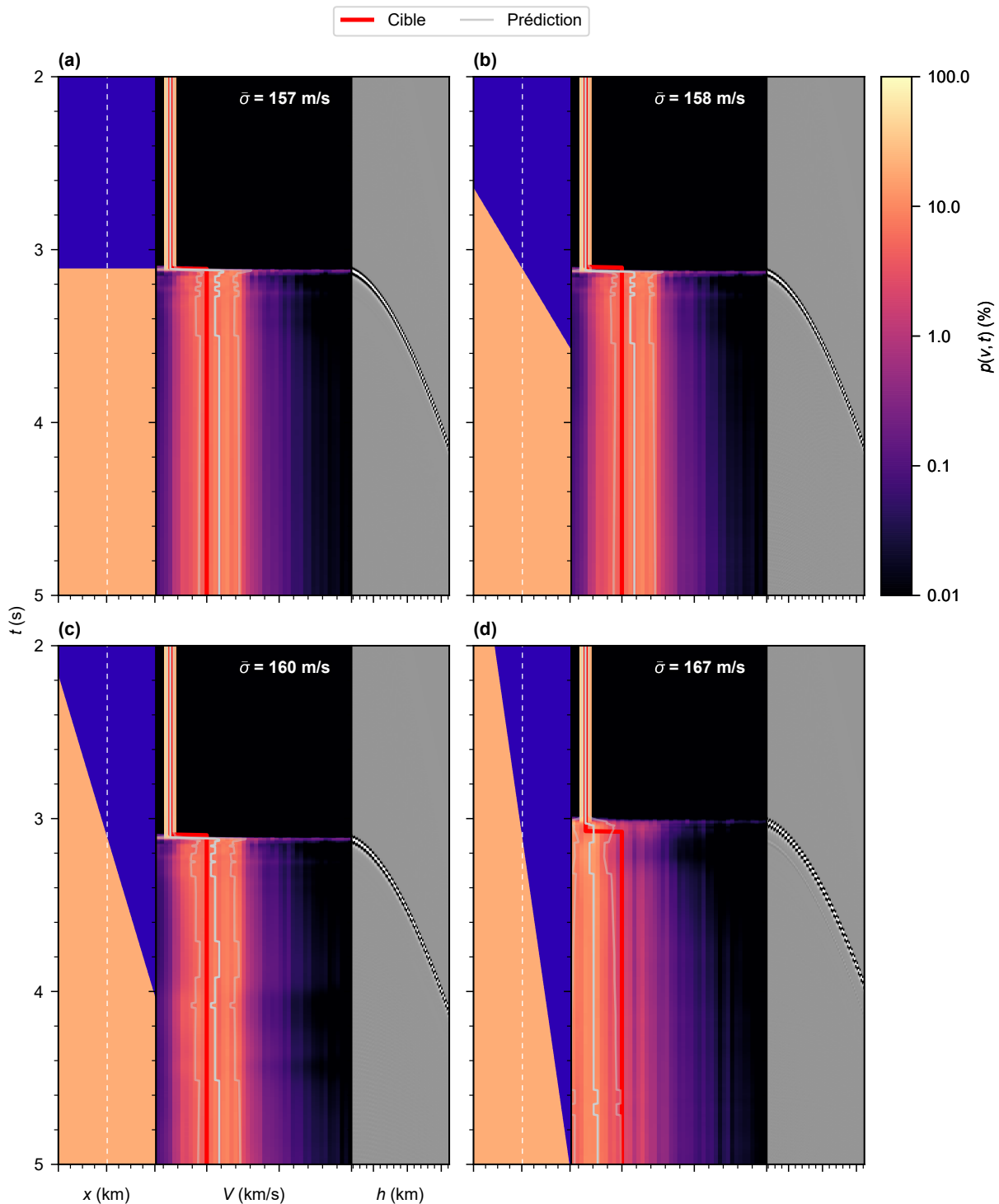


FIGURE 5.7 : Impact du pendage  $\theta$  sur les prédictions et la métrique de méfiance

(Panneaux de gauche) Modèle de vitesse 2D. Le CMP est indiqué par une ligne blanche pointillée. (Panneaux centraux) Prédications d'un seul classificateur et valeur moyenne de l'écart-type  $\bar{\sigma}$ . (Panneaux de droite) Regroupement CMP d'entrée. (a)  $\theta = 0^\circ$ . (b)  $\theta = 5^\circ$ . (c)  $\theta = 10^\circ$ . (d)  $\theta = 20^\circ$ .

sont tardives parce que la couche est plus éloignée du CMP. Ils ne perturbent pas fortement la sortie, peut-être parce que l'amplitude est plus faible ou parce que la signature est différente de celle des exemples d'entraînement. Lorsque le CMP coïncide avec l'emplacement de la discontinuité, comme à la figure 5.8c, deux couches sont détectées et la méfiance augmente d'au moins 20%, c'est-à-dire de  $\bar{\sigma} = 148$  m/s à  $\bar{\sigma} = 180$  m/s. Il n'est pas clair quelle devrait être la vitesse cible entre 3.2 s et 3.8 s, mais le classificateur indique une vitesse qui est à peu près la moyenne de la précédente et de la suivante. La vitesse est surestimée d'au moins 20% en profondeur.

Ces observations indiquent que les structures qui n'ont pas été observées pendant l'entraînement, à savoir les couches avec un pendage non nul et les discontinuités, se traduisent par une augmentation de la méfiance. En d'autres termes, si un bruit cohérent est introduit par des structures géologiques hors distribution, les classificateurs sont en mesure d'indiquer, quoique faiblement, où les résultats peuvent être incertains. Ces observations pourraient s'étendre à l'effet que pourraient avoir des structures 3D sur l'estimation du modèle de vitesse 2D. Cependant, on ne sait toujours pas comment on pourrait décider d'éliminer un résultat. Il semble qu'une méfiance de  $\bar{\sigma} > 160$  m/s indiquerait qu'un profil devrait être inspecté plus en détail, par exemple lorsqu'une faille coïncide avec un CMP ou lorsqu'il y a des couches très inclinées. Cependant, l'incertitude épistémique due à l'hypothèse d'une homogénéité latérale pendant l'entraînement ne se traduira pas nécessairement par une augmentation de la méfiance. En comparant les prédictions pour la figure 5.7c et la figure 5.7d, il n'y a pas une augmentation suffisamment forte de la méfiance qui justifierait l'abandon du résultat de la figure 5.7d. En général, les réflexions plus faibles peuvent ne pas être détectées par les NNs, l'amplitude n'étant pas grande pour que le classificateur enregistre une augmentation de la méfiance. L'étude de l'incertitude épistémique reste une question ouverte.

### 5.3 Conclusion

J'ai traduit le problème de régression de la vitesse sismique par DL en problème de classification. Au lieu de prédire une valeur de vitesse par point dans l'espace et/ou le temps, un classificateur prédit un vecteur de valeurs de confiance associé à chaque vitesse admissible. En comparant des ensembles de régresseurs et de classificateurs, on observe que la classification sur les vitesses

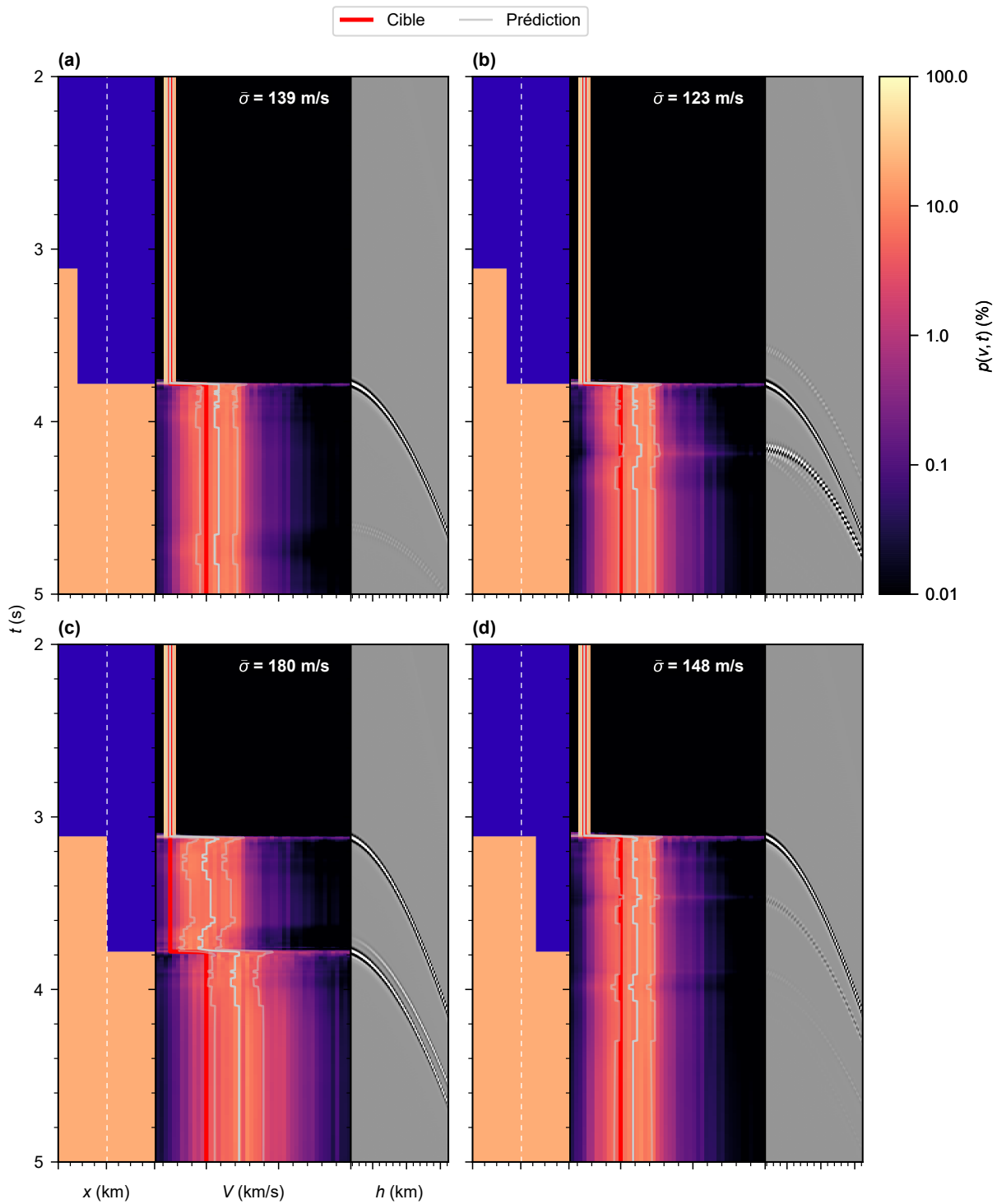


FIGURE 5.8 : Impact des discontinuités sur les prédictions et la méfiance

(Panneaux de gauche) Modèle de vitesse 2D. Le CMP est indiqué par une ligne blanche en pointillés. (Panneaux centraux) Prédiction d'un seul classificateur et valeur moyenne de l'écart-type  $\bar{\sigma}$ . (Panneaux de droite) Regroupement CMP d'entrée.

discrétisées est une meilleure alternative à la régression en ce sens qu'elle permet d'estimer les degrés de confiance en les résultats tout en facilitant l'entraînement. Remplacer 16 régresseurs par 1 classificateur permet d'accélérer par 8 fois l'entraînement et les remplacer par 16 classificateurs permet de réduire l'erreur de 15%. On observe que la classification est plus appropriée pour estimer l'erreur que les ensembles de régresseurs, ces derniers conduisant à des écarts-types plus conservateurs. Les ensembles, néanmoins, sont utiles pour réduire l'erreur de toute stratégie de DL. Le fait de considérer l'écart-type de la distribution de confiance comme une mesure de méfiance a permis d'identifier quantitativement certaines sections dans les prédictions où les résultats étaient plus incertains. En utilisant des exemples hors distribution, j'ai pu montrer qu'en présence de bruit cohérent, les classificateurs réduisent parfois, mais pas systématiquement, la confiance. L'analyse de l'impact des écarts de domaines est poussée plus en profondeur au chapitre suivant. Enfin, bien que les approches bayésiennes puissent fournir des estimations d'incertitude réelles, la régression par classification est simple et fournit néanmoins plus d'information utile à la prise de décision que les ensembles.



## 6 CARACTÉRISER LES ÉCARTS DE DOMAINES

---

En plus de conduire à des NNs utilisables dans des contextes réels (Kim et al., 2022b), les données synthétiques permettent un contrôle précis et riche du domaine d'entraînement. Par exemple, en vision par ordinateur, les jeux de données sont souvent porteurs de biais ethnique ou de biais de genre. Les données synthétiques peuvent mitiger ces biais à l'aide de modèles 3D d'êtres humains de toutes les ethnies et de tous les genres (Kim et al., 2022b). En général, les données synthétiques permettent de contrer facilement les biais, puisque les hypothèses ne sont plus portées par des jeux de données, mais plutôt par les connaissances des concepteurs. En sismique, il a été montré qu'un entraînement sur des données 3D purement synthétiques permet de détecter des failles beaucoup plus efficacement et avec une justesse accrue par rapport aux méthodes conventionnelles (Wu et al., 2019a). Les méthodologies basées purement sur le synthétique permettent d'élargir arbitrairement le domaine d'entraînement, pourvu que les capacités de modélisation (le coût de calcul, le réalisme, la versatilité) soient suffisantes.

Néanmoins, la création de données synthétiques ayant toutes les caractéristiques et toute la richesse des données réelles est actuellement impossible (Araya-Polo et al., 2019). Quelles que soient les capacités de modélisation, l'utilisation de modèles de vitesse synthétiques en sismique implique un écart de domaine par rapport au domaine réel, c'est-à-dire qu'il existe nécessairement une différence de caractéristiques entre les données synthétiques et les données réelles. Pour appliquer le DL à des données réelles, il est donc nécessaire d'adopter une méthodologie de génération de modèles impliquant certaines hypothèses simplificatrices et, donc, un biais (Wolpert et al., 1997). Il existe une panoplie de façons de générer des modèles synthétiques et il est primordial de s'assurer que les modèles générés sont adéquats, puisque l'écart de domaine peut nuire à la généralisation et la transférabilité au domaine cible (Wang et al., 2018a). Par exemple, les méthodes de DL peuvent être appliquées à un levé marin de la marge continentale atlantique des États-Unis, comme il a été montré au chapitre 4, mais l'écart de domaine doit être minimisé en amont en concevant un jeu de données d'entraînement représentatif des structures attendues.

Entre les domaines synthétiques eux-mêmes, on peut aussi observer des performances variables dues à des écarts de domaines. Par exemple, l'absence d'un objet géologique (par exemple des discontinuités) dans les données d'entraînement peut entraîner des erreurs lorsque celui-ci est présent dans le domaine cible. Au chapitre 4, il a été noté que les prédictions des NNs aux localisations de tels objets pourraient être aléatoires. Au chapitre 5, il a été noté que certaines sources de bruit peuvent être prises en compte lorsqu'elles ont une signature bien définie, mais que la présence de bruit épistémique pouvant être due à l'absence d'objets dans le jeu d'entraînement n'est pas bien caractérisée. Ces effets restent à vérifier.

Les conditions pour une bonne transférabilité ne sont toujours pas expliquées et la transférabilité entre domaines n'est pas garantie. Actuellement, il est incertain si cet effet est dû au processus d'entraînement, à des limitations physiques ou à la composition du jeu de données d'entraînement. Dans ce chapitre, je cherche à éclaircir la nature de cet effet, puisque la question de transférabilité interdomaines est applicable à tout algorithme de DL pour l'estimation de modèles de vitesse et qu'y répondre permettrait d'expliquer d'éventuels effets observés dans les données réelles.

Le matériel de ce chapitre est prêt à soumettre chez *Computers & Geosciences* :

Simon J, Fabien-Ouellet G, Gloaguen E (prêt à soumettre). Domain gaps in velocity model estimation : characterizing the transferability between domains. *Computers & Geosciences*.

Le code source est disponible en ligne (Simon, 2023).

## 6.1 Méthodologie

Pour la suite de ce chapitre, le NN utilisé est le même que celui du chapitre 5. Pour l'entraînement en 2D, j'utilise les mêmes hyperparamètres qu'au chapitre 4. Par contre, contrairement à ce qui est effectué au chapitre 4, le NN est entraîné directement sur des modèles 2D, sans TL, pour limiter les degrés de liberté de l'analyse.



### 6.1.1 Jeux de données synthétiques

Au chapitre 4, il a été observé que la justesse diminue lorsque les couches sont plus minces, lorsqu'elles sont plus profondes et lorsque la vitesse cible augmente. Au chapitre 5, il a été constaté que la transférabilité diminue lorsque le domaine source est composé de couches uniformes latéralement et que le domaine cible contient des discontinuités. Les limites physiques des méthodes sismiques demeurent valides malgré l'utilisation de DL. Il se peut qu'il existe des biais envers certaines caractéristiques ou échelles des jeux de données choisis et il est primordial de caractériser les cas pour lesquels on approche ou on atteint ces limites. Les NNs étant des boîtes noires, la façon la plus facile de les caractériser est de quantifier l'impact de la variation continue de certains paramètres des modèles d'entrée sur l'erreur.

Dans ce chapitre, la transférabilité interdomaines est caractérisée en générant par objets des modèles dont les propriétés sont de plus en plus extrêmes et en comparant la performance atteinte pour chaque paire de domaines d'entraînement et de test. Le processus est illustré schématiquement à la figure 6.1. En concordance avec certaines des observations des chapitres 4 et 5, je génère des jeux de données pour lesquels varient le contraste de vitesse maximal  $\Delta V$  entre deux couches consécutives, la fréquence de crête  $f$  de l'ondelette et les décrochements  $d$  de discontinuités verticales. Le paramètre  $\Delta V$  contrôle le choix de la vitesse d'une couche  $V_i$ , celle-ci étant échantillonnée uniformément sur  $[V_{i-1} - \Delta V, V_{i-1} + \Delta V]$ . La variation de chacun de ces paramètres est ensuite reliée à la justesse.

Pour l'étude de  $\Delta V$  et de  $f$ , j'utilise des modèles de vitesse homogènes latéralement (1D), tandis que, pour l'étude de  $d$ , j'utilise des modèles 2D. Chacun des paramètres varie indépendamment, les autres demeurant constants ( $\Delta V = 1000$  m/s,  $f = 25$  Hz,  $d = 0$  m). Pour chaque jeu de données 1D, je génère 2 000 modèles d'entraînement et 100 modèles de test, tandis que pour le jeu de données 2D, 200 modèles d'entraînement et 10 modèles de test sont utilisés. La génération d'un jeu de données 1D prend 15 minutes avec 64 GPU (16 heures GPU), tandis que celle d'un jeu 2D prend 90 minutes (4 jours GPU). J'utilise des GPUs NVIDIA A100 (modèle 40 GB). Ces quantités de modèles ne constituent pas des données massives (10 fois plus d'exemples sont utilisés au chapitre 4), mais en générer davantage s'avère irréaliste, puisqu'il y a au total 8 jeux

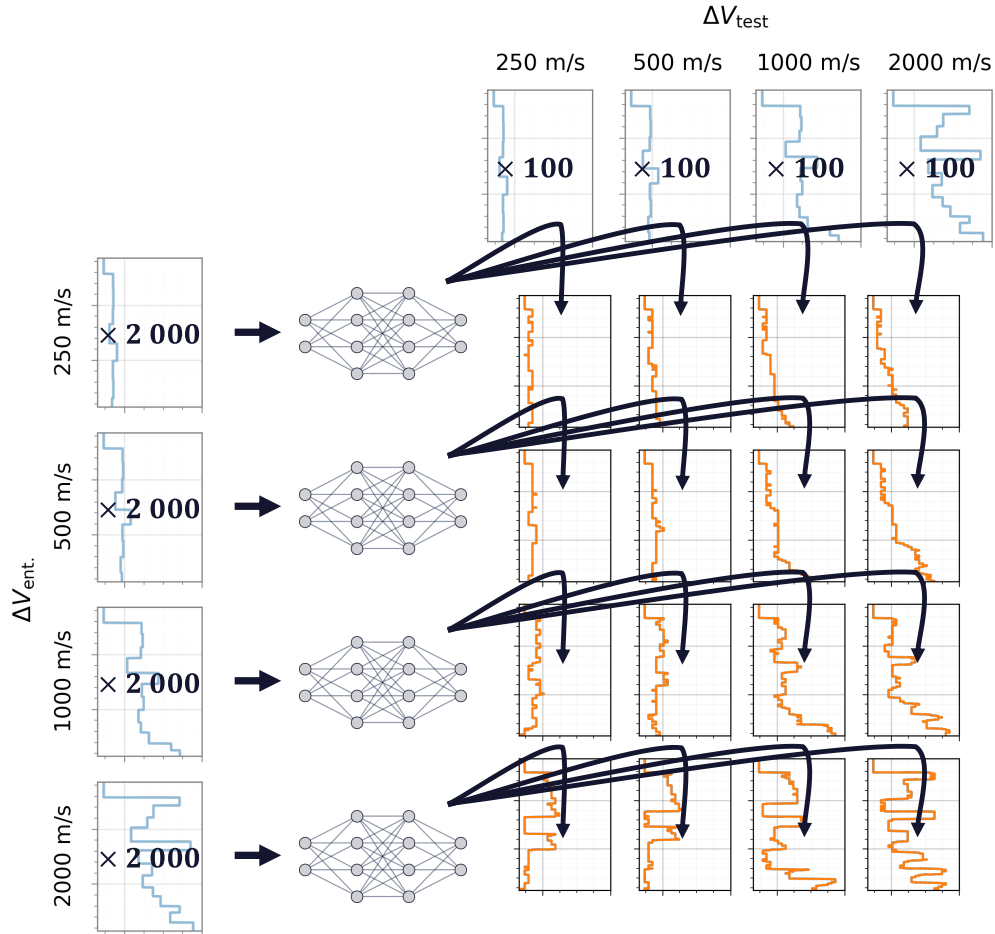


FIGURE 6.1 : Méthodologie pour évaluer la transférabilité interdomaines

Plusieurs jeux de données sont générés en faisant varier un des paramètres du générateur de modèles, dans ce cas  $\Delta V$ , le contraste de vitesse maximal entre deux couches consécutives. Chaque jeu de données d'entraînement a un jeu de données test correspondant. Les jeux de données d'entraînement servent chacun à entraîner un NN différent. La performance de chaque NN est ensuite évaluée sur chacun des jeux de données test.

de données 1D et 3 jeux de données 2D à générer, pour un total de plus de 17 jours GPU. En d'autres termes, il y a de forts risques qu'il soit difficile d'ajuster les NNs aux données. Outre ces détails, les modèles et les réponses sismiques sont générés de la même façon qu'au chapitre 4.

## 6.2 Résultats et discussion

L'entraînement d'un NN sur des données 1D prend 30 minutes avec 2 GPUs (1 heure GPU), tandis que l'entraînement sur des données 2D prend 4.5 heures (9 heures GPU).

		$\Delta V_{\text{test}}$ (m/s)			
		250	500	1 000	2 000
$\Delta V_{\text{ent.}}$ (m/s)	250	69 ± 43	171 ± 117	397 ± 163	530 ± 131
	500	93 ± 50	100 ± 55	282 ± 119	465 ± 140
	1 000	158 ± 89	140 ± 61	138 ± 66	329 ± 152
	2 000	233 ± 71	246 ± 98	231 ± 89	221 ± 98
Tous		77 ± 27	108 ± 49	178 ± 78	272 ± 141

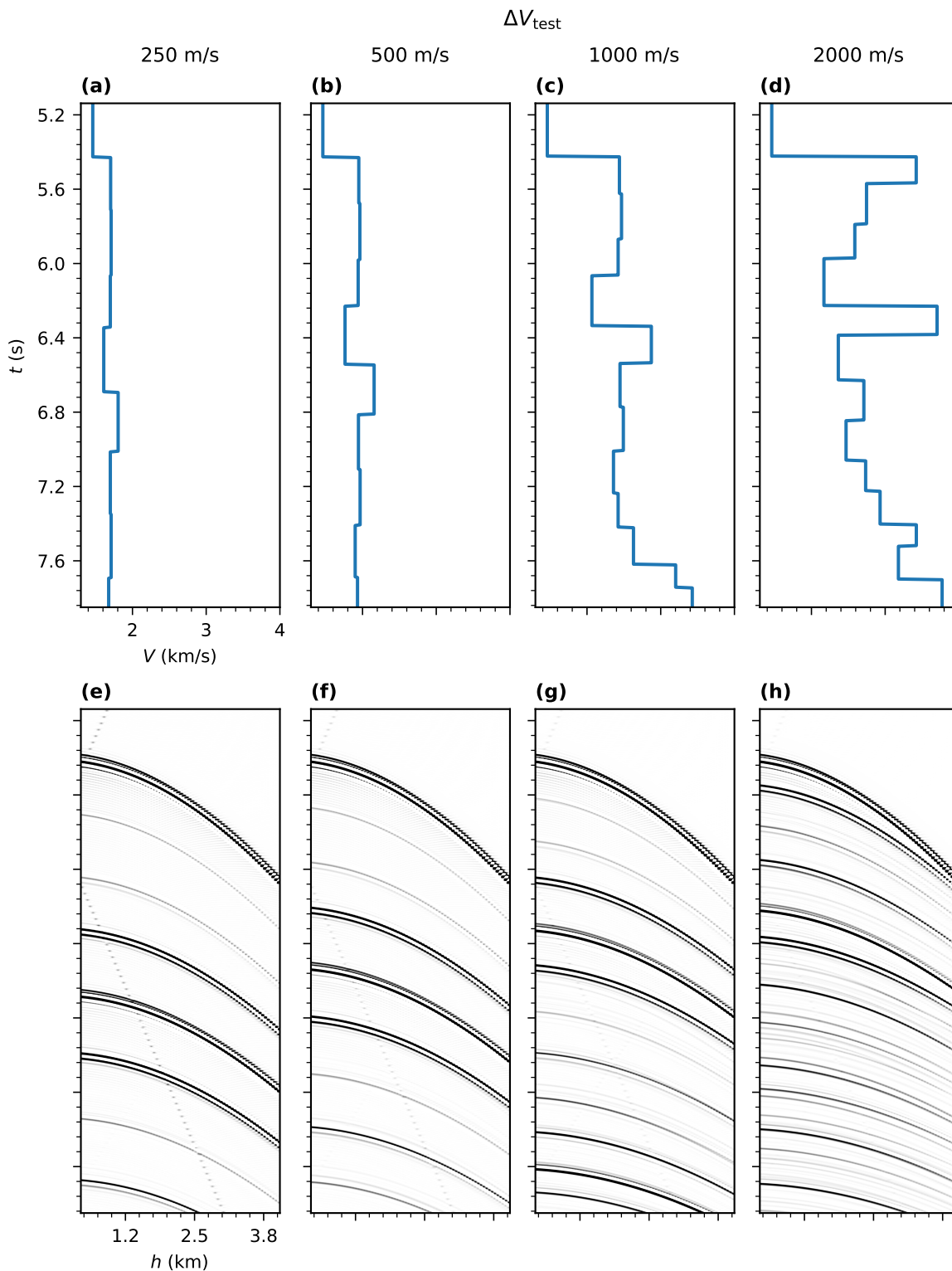
TABLEAU 6.1 : Impact du contraste maximal de vitesse  $\Delta V$

RMSE (en m/s) selon la paire de jeux de données d'entraînement et de test. La disposition est la même qu'à la figure 6.3.

Dans tous les résultats suivants, on observe que l'erreur pour un paramètre d'entraînement donné est généralement plus basse lorsque ce paramètre est le même en phase de test, ce qui est dû à la bonne adaptation entre le domaine source et le domaine cible.

### 6.2.1 Impact du contraste maximal de vitesse $\Delta V$ entre couches

L'impact de la variation de la différence maximale de vitesse intercouches  $\Delta V$  sur les données d'entrée est montré à la figure 6.2. Les prédictions correspondantes peuvent être analysées qualitativement à la figure 6.3 et quantitativement au tableau 6.1. Les résultats des figures 6.3k et 6.3p semblent être les plus justes qualitativement. Pour ces cas, l'*a priori* sur la vitesse est le moins fort : les plages de vitesse possibles sont très vastes et les NNs semblent être poussés à se baser sur les données d'entrées pour produire les prédictions, plutôt que de se fier à la distribution statistique des vitesses du jeu d'entraînement. Néanmoins, les NNs entraînés sur de forts  $\Delta V$  sont plus réactifs : les figures 6.3m à 6.3p permettent d'observer que le NN a tendance à surestimer les variations de vitesse dès qu'il rencontre un signal indicatif d'un contraste de vitesse. Au contraire, les NNs entraînés sur des faibles  $\Delta V$  n'atteignent des vitesses extrêmes que progressivement, comme à la figure 6.3d, par exemple. Le tableau 6.1 permet d'observer que, pour  $\Delta V_{\text{ent.}} \in (500 \text{ m/s}, 1000 \text{ m/s})$ , les justesses obtenues sur des jeux de données dont les propriétés sont moins extrêmes que celles du domaine d'entraînement sont plus grandes ou comparables à celles pour lesquelles  $\Delta V_{\text{ent.}} = \Delta V_{\text{test}}$ . Cela concorde avec les résultats du chapitre 4, auquel il a été observé que les propriétés du domaine cible doivent être moins extrêmes que celles du domaine d'entraînement pour assurer une bonne performance. Cela concorde aussi avec la mé-



**FIGURE 6.2 : Exemples avec contraste maximal de vitesse  $\Delta V$  variable**

L'exemple affiché est un exemple aléatoire du jeu de données de test. La suite aléatoire utilisée pour produire chacun des exemples est la même. Seul le paramètre  $\Delta V_{\text{test}}$  du jeu de données test change de colonne en colonne. (a-d) Modèle de vitesse. (e-h) Regroupement CMP correspondant.

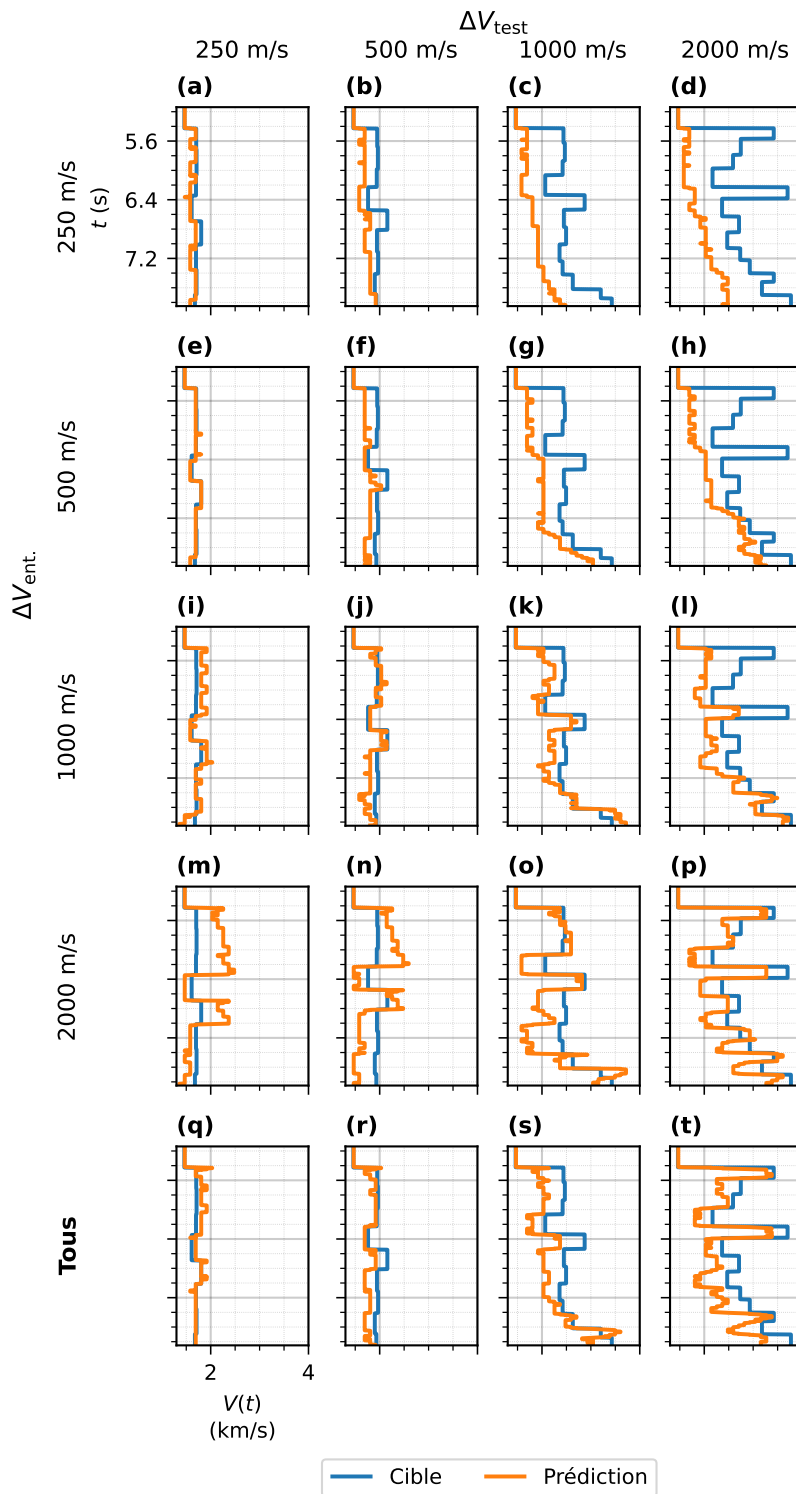


FIGURE 6.3 : Impact du contraste maximal de vitesse  $\Delta V$

Les exemples test de chacune des colonnes sont les mêmes qu'à la figure 6.2. Les prédictions de chaque ligne sont produites par des NNs différents, lesquels sont entraînés sur des jeux de données pour lesquels  $\Delta V_{\text{ent}}$  varie. Le NN utilisé pour la dernière ligne (q-t) a été entraîné sur tous les exemples utilisés pour les autres lignes.

		$f_{\text{test}}$ (Hz)		
		10	15	25
$f_{\text{ent.}}$ (Hz)	10	204 ± 125	186 ± 110	243 ± 132
	15	247 ± 198	164 ± 101	190 ± 111
	25	392 ± 264	234 ± 158	161 ± 79

TABLEAU 6.2 : Impact de la fréquence de crête moyenne  $f$

**RMSE (en m/s) selon la paire de jeux de données d'entraînement et de test. La disposition est la même que celle du tableau 6.1 (sauf la dernière ligne).**

thodologie de Kazei et al. (2021), lesquels entraînent leur NN sur le modèle de vitesse Marmousi pour être en mesure de l'appliquer aux données de Marmousi2, les vitesses de Marmousi2 étant moins extrêmes que celles de Marmousi.

J'ai tenté de combler l'écart entre les domaines des différents jeux d'entraînement en entraînant un dernier NN sur l'ensemble des jeux de données à disposition, c'est-à-dire  $\Delta V_{\text{ent.}} \in (250 \text{ m/s}, 500 \text{ m/s}, 1000 \text{ m/s}, 2000 \text{ m/s})$ . Les résultats sont affichés aux dernières lignes de la figure 6.3 et du tableau 6.1. En comparant la dernière ligne du tableau 6.1 avec la diagonale de la partie supérieure ( $\Delta V_{\text{ent.}} = \Delta V_{\text{test}}$ ), on constate qu'une partie de la performance du NN est effectivement due à un *a priori* statistique, puisque la performance diminue légèrement pour chaque cas. Par contre, l'erreur de la dernière ligne est plus basse que celle de tous les éléments hors diagonale ( $\Delta V_{\text{ent.}} \neq \Delta V_{\text{test}}$ ). Mettre à disposition tous les exemples, bien que cela permette à un NN d'atteindre une erreur plus uniforme à travers toutes les échelles de vitesses, ne permet pas de lui conférer les capacités suffisantes pour minimiser au maximum l'erreur, cela demandant de connaître en amont les plages de vitesses attendues.

### 6.2.2 Impact de la fréquence de crête $f$ de l'ondelette

Au chapitre 4, la fréquence de crête des ondelettes utilisées dans les modélisations sismiques est choisie aléatoirement dans une plage de 10 Hz. Ce choix assurerait une meilleure transférabilité aux données de terrain utilisées, pour lesquelles il s'avère impossible de déterminer exactement les propriétés de l'ondelette source. De façon à vérifier la transférabilité entre plages de fréquences, j'utilise une fréquence de crête choisie aléatoirement entre  $f - 5$  Hz et  $f + 5$  Hz (ci-après, le symbole  $f$  seul est utilisé). L'impact du choix de fréquence de crête moyenne  $f$  de l'ondelette

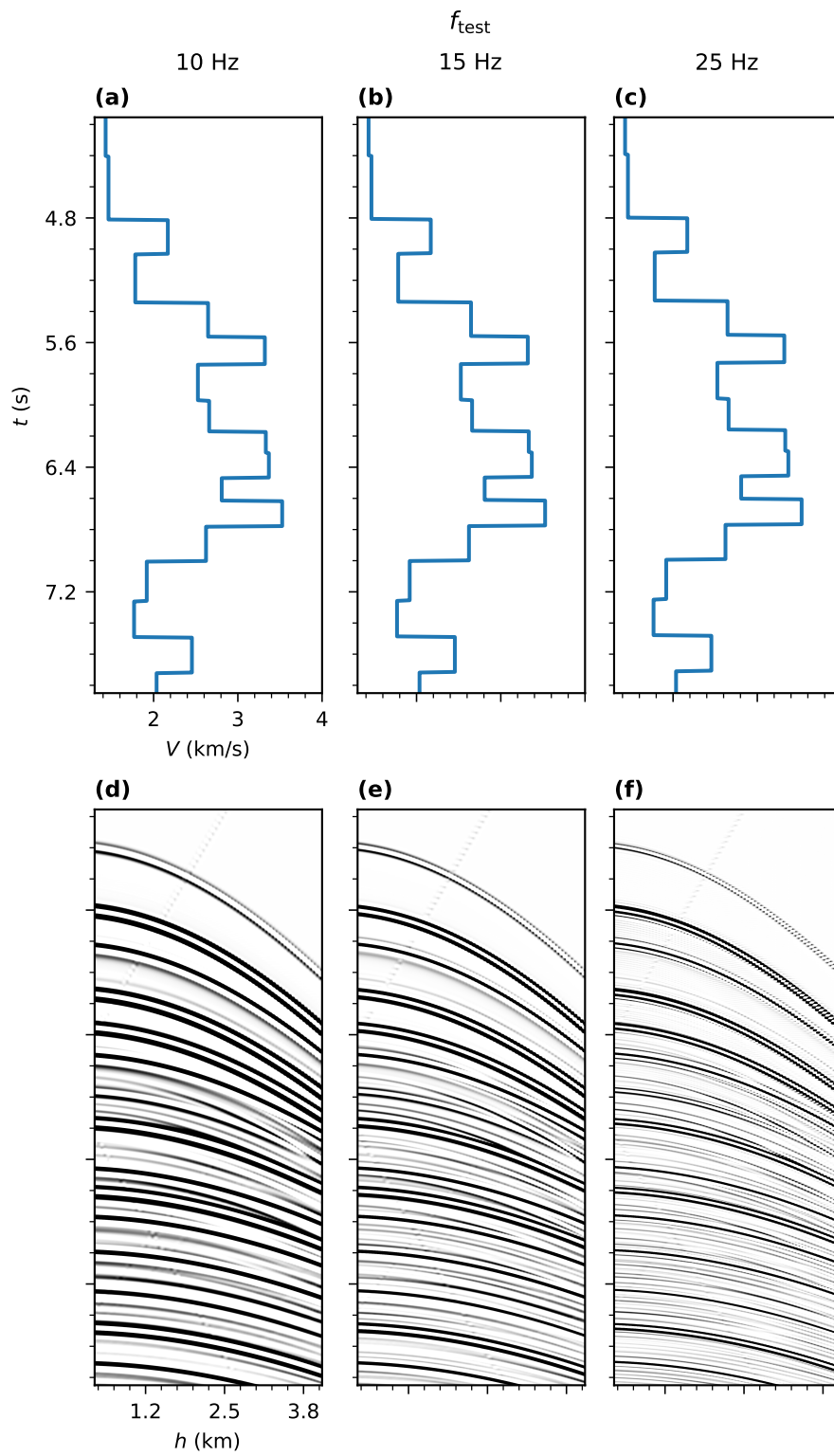


FIGURE 6.4 : Exemples avec fréquence de crête moyenne  $f$  variable

La disposition est la même que celle de la figure 6.2.

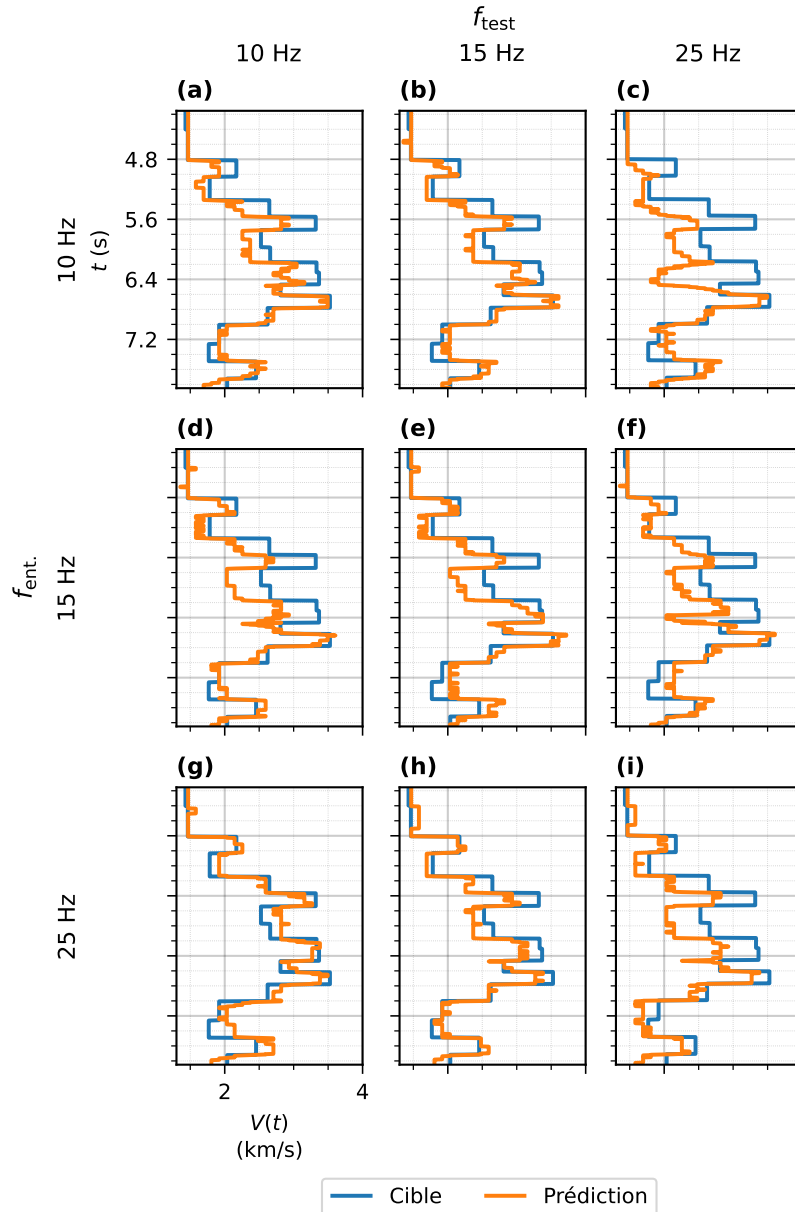


FIGURE 6.5 : Impact de la fréquence de crête moyenne  $f$

La disposition est la même que celle de la figure 6.3 (sauf la dernière ligne).

utilisée dans la modélisation sismique sur les données d'entrée est montré à la figure 6.4. Les prédictions correspondantes peuvent être analysées qualitativement à la figure 6.5 et quantitativement au tableau 6.2. On observe à la figure 6.5c que les NNs entraînés sur de basses fréquences ont tendance à ne pas détecter les hautes fréquences ou à sous-estimer les vitesses. Qualitativement, il semble y avoir une bonne transférabilité entre les autres combinaisons de jeux d'entraînement et de test. Utiliser  $f \pm 5$  Hz dans l'entraînement semble permettre d'étendre le domaine d'applicabilité au-delà de 5 Hz : par exemple, l'exemple de la figure 6.5g est très bien résolu, même si les do-



		$d_{\text{test}}$ (m)		
		200	400	800
$d_{\text{ent.}}$ (m)	0	95 ± 33	242 ± 53	221 ± 48
	200	183 ± 80	171 ± 58	173 ± 60
	400	204 ± 111	179 ± 73	176 ± 67

**TABLEAU 6.3 : Impact de discontinuités de décrochement maximal  $d$**

**RMSE (en m/s) selon la paire de jeux de données d'entraînement et de test. La disposition est la même que celle du tableau 6.2.**

maines ne se superposent pas. Il est possible que le NN apprenne à analyser des caractéristiques de plus haut niveau lorsqu'il est soumis à des ondelettes de fréquence variable. Physiquement, l'épaisseur de couche la plus petite pouvant être résolue est  $\Delta x = \frac{V}{4f}$  (Yilmaz, 2001) ou, dans le domaine du temps,  $\Delta t = \frac{1}{4f}$ . La résolution croît lorsque  $f$  augmente et les épaisseurs pouvant être résolues pour les fréquences utilisées sont de 25 ms, 17 ms et 10 ms. Aucune couche dans les exemples utilisés n'est plus mince que 25 ms, mais on observe néanmoins que l'erreur décroît le long de la diagonale du tableau 6.2.

### 6.2.3 Impact de discontinuités verticales de décrochement maximal $d$

L'impact de la présence de discontinuités verticales sur les données d'entrée est montré à la figure 6.6. Les prédictions correspondantes peuvent être analysées qualitativement à la figure 6.7 et quantitativement au tableau 6.3. Je fais varier le décrochement vertical maximal  $d$  des discontinuités. À l'emplacement de la discontinuité, on observe que les prédictions ont tendance à se diffuser latéralement à travers la discontinuité et que la division n'est pas nette. Cela peut s'expliquer par la présence d'un CNN récurrent dans le NN, lequel peut moyenniser l'encodage sur l'axe horizontal. Si ces conjectures sont vraies, le NN utilisé dans cette thèse est légèrement inadéquat en présence de discontinuités. En présence de discontinuités, le NN entraîné sur  $d_{\text{ent.}} = 0$  m surestime fortement les vitesses, comme on peut l'observer aux figures 6.7b et 6.7c. Au tableau 6.3, on peut voir que les NNs entraînés sur  $d_{\text{ent.}} = 200$  m produit de meilleurs résultats sur  $d_{\text{test}} = 400$  m que le NN pour lequel  $d_{\text{ent.}} = 400$  m. On observe que ces NNs sont capables d'extrapoler la distribution d'entraînement dans le cas d'un décrochement vertical, puisque les RNNs et les CNNs dont ils sont constitués sont invariants à la translation.

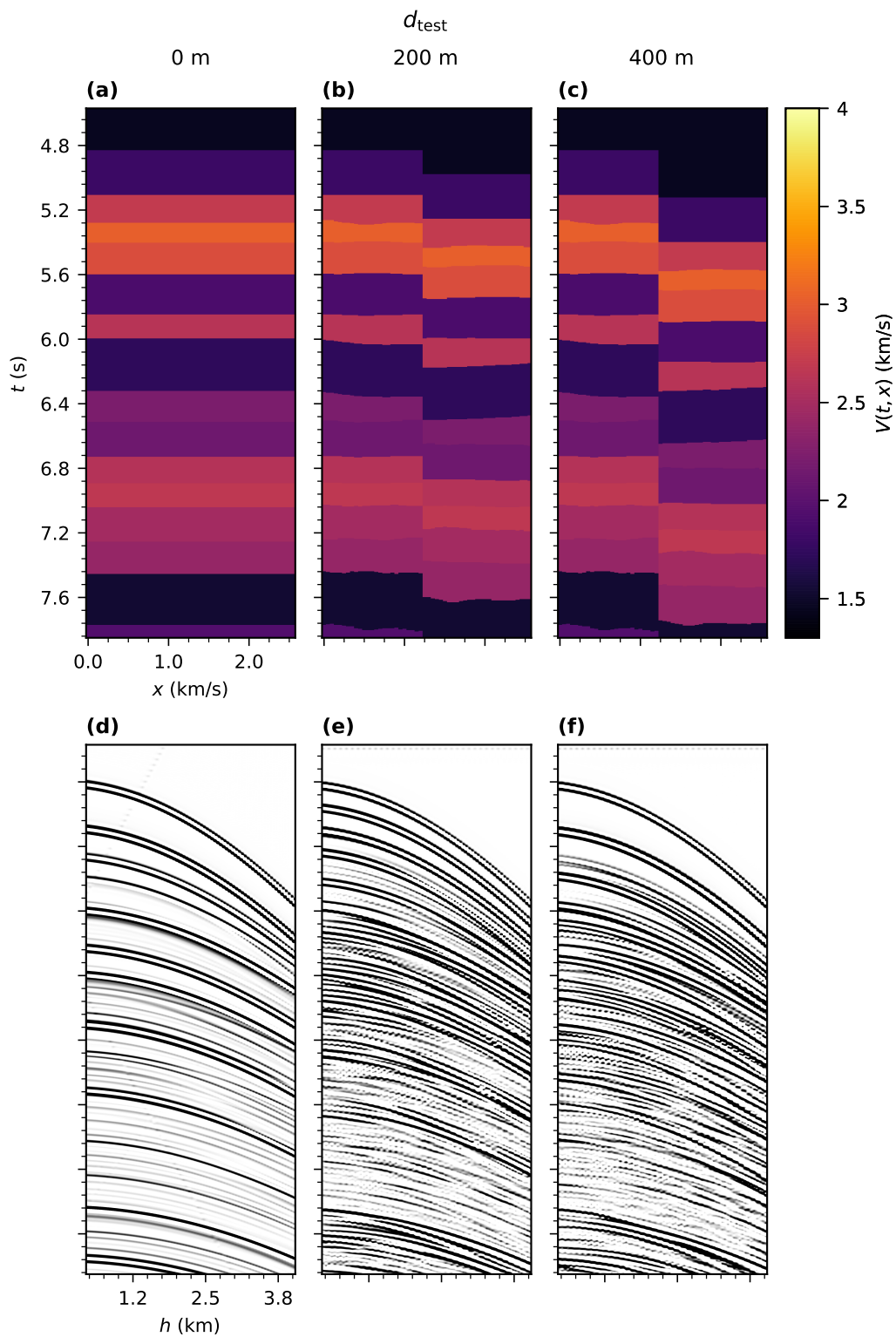


FIGURE 6.6 : Exemples avec discontinuité de décrochement maximal  $d$  variable

La disposition est la même que celle de la figure 6.2, mais les modèles affichés sont 2D. Les données affichées correspondent au CMP situé pile à la position de la discontinuité.

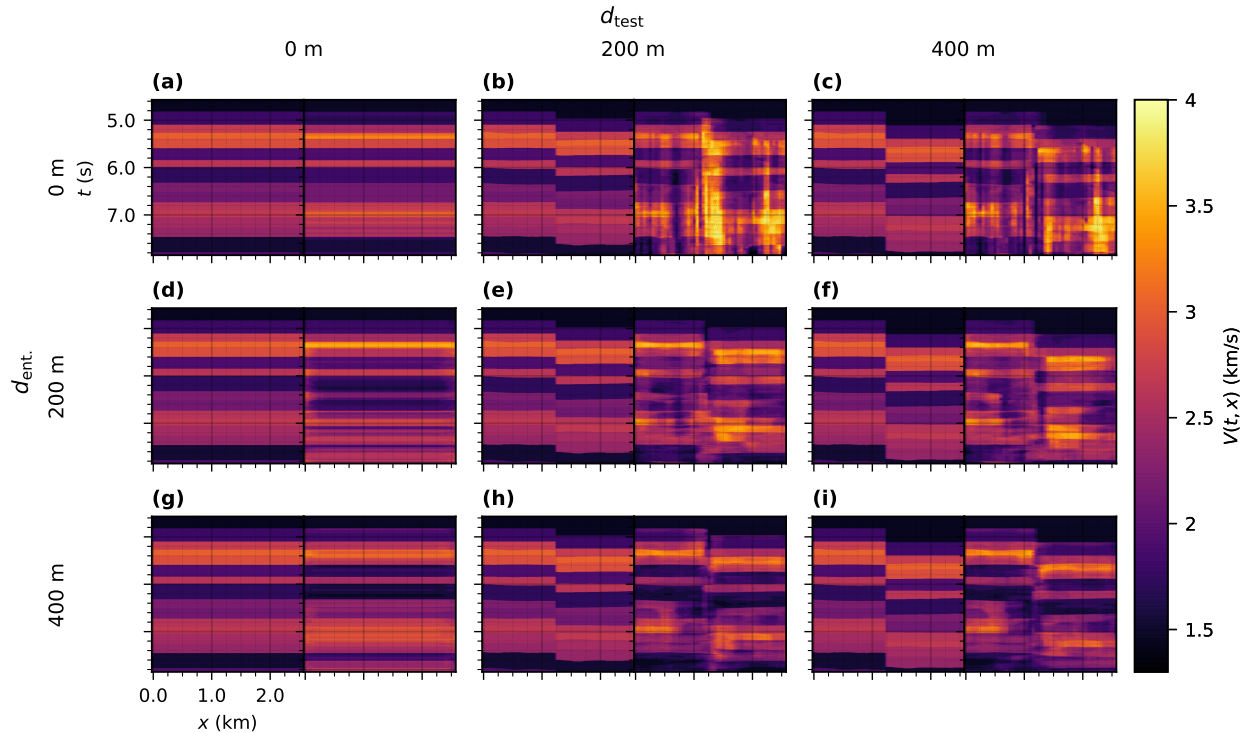


FIGURE 6.7 : Impact de discontinuités de décrochement maximal  $d$

La disposition est la même que celle de la figure 6.5, mais les modèles affichés sont 2D. (Panneaux de gauche) Modèle de vitesse cible. (Panneaux de droite) Modèle de vitesse prédit.

### 6.3 Conclusion

Dans ce chapitre, j'ai généré par objets des modèles de vitesse en faisant varier les paramètres géométriques ou statistiques des structures générées. J'ai caractérisé la transférabilité entre chacun des domaines en mesurant la RMSE sur le domaine cible en fonction du jeu de données d'entraînement choisi. En présence de discontinuités, les NNs sont capables de traiter des données pour lesquelles le décrochement est plus élevé. Cela confirme que les NNs sont parfois aptes à extrapoler hors de leur domaine d'entraînement, lorsque l'invariance temporelle est garantie, par exemple. Néanmoins, il faut une forte connaissance des distributions de vitesse du domaine cible pour assurer une bonne performance, les prédictions des NNs étant soit trop progressives, soit trop contrastantes lorsque les différences de vitesse intercouches sont inadéquates. En incluant toutes les plages de vitesse à l'entraînement, le NN devient plus versatile, mais une partie de la performance est sacrifiée, ce qui confirme que la justesse d'un NN en sismique dépend au moins en partie des *a priori* statistiques. J'ai confirmé que le domaine d'entraînement doit parfois être

plus vaste (extrême) que celui cible pour permettre une bonne performance, tel qu'observé au chapitre 4.

À l'heure actuelle, les bases théoriques pour maximiser la transférabilité interdomaines est incomplète. Avec les techniques actuelles utilisées en DL pour l'estimation de modèles de vitesse, le réentraînement peut rarement être entièrement évité. Une façon de contourner ce défi théorique serait de laisser à un NN la tâche de déterminer dynamiquement les sous-ensembles d'entraînement par AL. L'AL gagne du terrain actuellement (Bengio et al., 2021, 2022) et je prévois que des débouchés en adaptation de domaine pour les données sismiques pourraient voir le jour grâce à l'AL.





## 7 CONCLUSION

---

L'estimation de modèles de vitesse sismique par DL nécessite de fortes contraintes pour performer correctement. Cela implique qu'appliquer une méthode de DL existante à chaque nouveau levé sismique requiert une quantité d'efforts qui n'est pas avantageuse. Cette thèse avait pour objectif d'accélérer et d'améliorer les techniques actuelles de DL appliquées à l'estimation de modèles de vitesse, de façon à favoriser son développement et son utilisation. Les contributions de cette thèse pourraient permettre de diminuer par au plus 80 fois les temps de calcul des méthodologies actuelles d'estimation de modèles de vitesse 2D par DL et de diminuer d'au plus 45% leur erreur.

Au chapitre 4, j'ai accéléré et régularisé l'entraînement en 2D à l'aide d'une stratégie de TL permettant de conserver par des arguments géométriques toute l'information apprise dans le cas 1D. Cette stratégie permet d'accélérer par au plus 10 fois l'entraînement et de diminuer l'erreur d'au plus 35% par rapport à un entraînement sans TL. En entraînant d'abord un NN à l'estimation des modèles de vitesse en couches horizontales, puis en l'augmentant et en l'entraînant à l'estimation des modèles 2D en couches à pendage non nul, un NN est plus à même de prédire des modèles 2D que s'il avait seulement été entraîné sur des modèles 2D. De plus, pourvu que les modèles synthétiques utilisés dans l'entraînement soient représentatifs de la géologie étudiée, les NNs peuvent être appliqués à des données réelles marines en produisant des sommations cohérentes et un modèle de vitesse en temps fidèle aux caractéristiques saillantes de la section sommée. Par contre, l'applicabilité du NN utilisé dans cette thèse au domaine de la profondeur reste à démontrer.

Au chapitre 5, j'ai pu diminuer le coût d'obtention de plages de prédictions acceptables pour un NN en reformulant la régression de la vitesse sismique en un problème de classification. Par rapport à des ensembles de 16 régresseurs, les classificateurs permettent d'accélérer par 8 fois les temps de calcul et de diminuer l'erreur d'au plus 15%. Le problème de classification est obtenu en discrétisant les vitesses admises et en permettant aux NNs d'estimer une confiance en chacune des vitesses plutôt que de prédire une seule vitesse scalaire. Pour ce faire, je n'ai modifié que la fin des NNs, transformant les scalaires de sortie en distributions *softmax*. Les résultats finaux sont les médianes des distributions et leur écart-type est interprété comme une mesure de méfiance. En

comparant la performance de la régression et de la classification, j'ai observé que la classification est plus juste et qu'elle fournit des résultats moins surconfiants. De plus, la métrique de méfiance conçue a permis de détecter certains bruits cohérents de structures 2D, mais elle ne peut pas systématiquement servir d'indicateur de l'incertitude, notamment lorsque celle-ci est épistémique. En bref, un seul NN procédant par classification peut être utilisé au lieu d'un ensemble de NNs procédant par régression, tout en utilisant moins de ressources et en étant plus facile à interpréter.

Au chapitre 6, j'ai caractérisé la capacité des NNs à extrapoler leurs apprentissages à des domaines synthétiques non couverts par le jeu d'entraînement. Pour ce faire, j'ai étudié la performance de NNs pour des paires de jeux d'entraînement et de test pour lesquels certaines propriétés varient indépendamment. En générant synthétiquement par objets les modèles utilisés, j'ai été en mesure de faire varier continuellement et indépendamment leurs propriétés géométriques et statistiques. J'ai notamment observé que les échelles de vitesse doivent être sensiblement les mêmes que celles du jeu d'entraînement pour garantir une bonne transférabilité au domaine cible. Les NNs sont parfois très sensibles au contenu statistique du jeu de données d'entraînement, ce qui explique pourquoi, en sismique, les jeux de données et les méthodes doivent généralement être adaptés aux applications visées. Actuellement, le réentraînement d'un NN ne peut être évité que lorsque les données cibles sont très similaires aux données d'entraînement.

Les conclusions de cette thèse ont été vérifiées pour un seul NN, à défaut d'en avoir testé d'autres. Par exemple, le TL et la classification permettent d'atteindre des temps de calcul réduits, et ce, en modifiant le NN, mais il n'est pas exclu que d'autres NNs puissent réagir différemment. Plusieurs résultats recourent tout de même ce qui était sous-entendu par la littérature actuelle. Par exemple, conditionner des NNs à des données sismiques n'est pas trivial : il faut contraindre fortement les problèmes pour que le DL performe bien. Conformément à ce qui peut être observé dans la majorité des méthodologies de DL appliquées à l'estimation de modèles de vitesse sismique, la transférabilité interdomaine n'est pas garantie et les stratégies d'entraînement doivent souvent être personnalisées. Le DL est la source d'avancements techniques importants pour le traitement de données massives, mais, quoique le potentiel du DL pour le traitement de données sismiques n'est plus à démontrer, il n'existe pas de façon directe et unique d'utiliser des NNs pour l'estimation de modèles de vitesse sismique. Les données synthétiques demeurent nécessaires à



l'entraînement des NNs. Cela implique un écart de domaine aux données réelles, lequel n'a pas été encore comblé explicitement. Ainsi, la majorité des études sur le DL appliqué aux données sismiques concordent : pour entraîner un NN sur des données synthétiques, il faut utiliser des modèles dont la complexité est représentative de la réalité. L'adaptation de domaine, le TL et la régression par classification, entre autres, sont des façons d'accéder à des coûts de calculs réduits, à une généralité accrue et, conséquemment, à des levés de plus grande envergure et de plus grande complexité géologique.

Néanmoins, les besoins en régularisation pour les problèmes d'estimation de vitesse sismique par DL demeurent. Il n'est toujours pas envisageable de résoudre l'entièreté des problèmes auxquels peut être appliquée la FWI, par exemple l'estimation de modèles viscoélastiques. Heureusement, les développements constants en DL motivent toujours la création de nouveaux NNs et de nouvelles stratégies d'entraînement. Les *Transformers* (Vaswani et al., 2017), par exemple, restent inexploités dans cette thèse. Pour régulariser les entraînements, on pourrait aussi présager le développement d'un entraînement multiéchelles, dans lequel les modèles estimés sont de moins en moins grossiers, à la façon de StyleGAN et de l'inversion par FWI en fréquences croissantes. De plus, l'AL, pendant lequel les exemples d'entraînement sont générés dynamiquement, prend de l'essor. Les méthodes sismiques pourraient en bénéficier, car l'AL mitigerait le besoin de concevoir des jeux de données d'entraînement sur mesure. À terme, un NN polyvalent pouvant être appliqué facilement à toute la variété des problèmes sismiques permettrait de réduire drastiquement les coûts du traitement sismique et de mitiger les hypothèses actuelles en imagerie et en inversion sismiques.



## BIBLIOGRAPHIE

- Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mane D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viegas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow : Large-Scale Machine Learning on Heterogeneous Distributed Systems. 19 pages.
- Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, Ghavamzadeh M, Fieguth P, Cao X, Khosravi A, Acharya UR, Makarekovic V, Nahavandi S (2021) A review of uncertainty quantification in deep learning : Techniques, applications and challenges. *Information Fusion*, 76:243–297. DOI :10.1016/j.inffus.2021.05.008.
- Adler A, Araya-Polo M, Poggio T (2021) Deep learning for seismic inverse problems : toward the acceleration of geophysical analysis workflows. *IEEE Signal Processing Magazine*, 38(2):89–119. DOI :10.1109/MSP.2020.3037429.
- Aleardi M, Salusti A (2021) Elastic prestack seismic inversion through discrete cosine transform reparameterization and convolutional neural networks. *GEOPHYSICS*, 86(1):R129–R146. DOI :10.1190/geo2020-0313.1.
- Alfarraj M, AlRegib G (2019) Semisupervised sequence modeling for elastic impedance inversion. *Interpretation*, 7(3):SE237–SE249. DOI :10.1190/INT-2018-0250.1.
- Alzahrani H, Shragge J (2022) Seismic velocity model building using neural networks : Training data design and learning generalization. *GEOPHYSICS*, 87(2):R193–R211. DOI :10.1190/geo2020-0547.1.
- Aminzadeh F, Brac J, Kunz T (1997) *SEG/EAGE 3-D Salt and Overthrust Models*. Numéro No. 1 : Distribution CD of Salt and Overthrust models de SEG/EAGE 3-D Modeling Series. SEG book series.
- Araya-Polo M, Dahlke T, Frogner C, Zhang C, Poggio T, Hohl D (2017) Automated fault detection without seismic processing. *The Leading Edge*, 36(3):208–214. DOI :10.1190/tle36030208.1.
- Araya-Polo M, Farris S, Florez M (2019) Deep learning-driven velocity model building workflow. *The Leading Edge*, 38(11):872a1–872a9. DOI :10.1190/tle38110872a1.1.
- Araya-Polo M, Jennings J, Adler A, Dahlke T (2018) Deep-learning tomography. *The Leading Edge*, 37(1):58–66. DOI :10.1190/tle37010058.1.
- Baysal E, Kosloff DD, Sherwood JWC (1983) Reverse time migration. *GEOPHYSICS*, 48(11):1514–1524. DOI :10.1190/1.1441434.
- Bengio E, Jain M, Korablyov M, Precup D, Bengio Y (2021) Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. DOI :10.48550/arXiv.2106.04399.
- Bengio Y, Bastien F, Bergeron A, Boulanger-Lewandowski N, Breuel T, Chherawala Y, Cisse M, Cote M, Erhan D, Eustache J, Glorot X, Muller X, Lebeuf SP, Pascanu R, Rifai S, Savard F, Sicard G (2011) Deep Learners Benefit More from Out-of-Distribution Examples. 9 pages.

- Bengio Y, Lahlou S, Deleu T, Hu EJ, Tiwari M, Bengio E (2022) GFlowNet Foundations. DOI :10.48550/arXiv.2111.09266.
- Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166. DOI :10.1109/72.279181.
- Biswas R, Vassiliou A, Stromberg R, Sen MK (2018) Stacking velocity estimation using recurrent neural network. *SEG Technical Program Expanded Abstracts 2018*, Society of Exploration Geophysicists, pages 2241–2245, Anaheim, California.
- Brougois A, Bourget M, Lailly P, Poulet M, Ricarte P, Versteeg R (1990) Marmousi, model and data. *EAGE Workshop - Practical Aspects of Seismic Data Inversion*, European Association of Geoscientists & Engineers, cp pages.
- Bunks C, Saleck FM, Zaleski S, Chavent G (1995) Multiscale seismic waveform inversion. *GEOPHYSICS*, 60(5):1457–1473. DOI :10.1190/1.1443880.
- Burschil T, Beilecke T, Krawczyk CM (2015) Finite-difference modelling to evaluate seismic P-wave and shear-wave field data. *Solid Earth*, 6(1):33–47. DOI :10.5194/se-6-33-2015.
- Campos L, Nogueira P, Nascimento E (2019) Estimating Initial Velocity Models for the FWI Using Deep Learning. *Proceedings of the 16th International Congress of the Brazilian Geophysical Society&Expogef*, Brazilian Geophysical Society, pages 1–4.
- Caruana R (1995) Learning Many Related Tasks at the Same Time with Backpropagation. 8 pages.
- Caruana R, Lou Y, Gehrke J, Koch P, Sturm M, Elhadad N (2015) Intelligible Models for Health-Care : Predicting Pneumonia Risk and Hospital 30-day Readmission. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, KDD '15, pages 1721–1730, New York, NY, USA.
- Chai X, Tang G, Wang S, Lin K, Peng R (2021) Deep Learning for Irregularly and Regularly Missing 3-D Data Reconstruction. *IEEE Transactions on Geoscience and Remote Sensing*, 59(7):6244–6265. DOI :10.1109/TGRS.2020.3016343.
- Chen K, Sacchi MD (2017) Elastic least-squares reverse time migration via linearized elastic full-waveform inversion with pseudo-Hessian preconditioning. *GEOPHYSICS*, 82(5):S341–S358. DOI :10.1190/geo2016-0613.1.
- Chollet F (2015) *Keras*. Keras.
- Chollet F (2018) *Deep learning with Python*. Manning, Shelter Island, NY.
- Claerbout JF (1985) *Imaging the Earth's Interior*. Blackwell Science Inc, Oxford England ; Boston.
- Committee STS (2017) *SEG-Y\_r2.0 : SEG-Y revision 2.0 Data Exchange format*. Hagelund R, Levin SA (éditeurs). Society of Exploration Geophysicists.
- Cordella L, De Stefano C, Tortorella F, Vento M (1995) A method for improving classification reliability of multilayer perceptrons. *IEEE Transactions on Neural Networks*, 6(5):1140–1147. DOI :10.1109/72.410358.

- Crutchley GJ, Maslen G, Pecher IA, Mountjoy JJ (2015) High-resolution seismic velocity analysis as a tool for exploring gas hydrate systems : An example from New Zealand's southern Hikurangi margin. *Interpretation*, 4(1):SA1–SA12. DOI :10.1190/INT-2015-0042.1.
- Dai W, Fowler P, Schuster GT (2012) Multi-source least-squares reverse time migration. *Geophysical Prospecting*, 60(4):681–695. DOI :10.1111/j.1365-2478.2012.01092.x.
- Das V, Pollack A, Wollner U, Mukerji T (2019) Convolutional neural network for seismic impedance inversion. *GEOPHYSICS*, 84(6):R869–R880. DOI :10.1190/geo2018-0838.1.
- de Souza CR, Gaidon A, Cabon Y, Peña AML (2017) Procedural Generation of Videos to Train Deep Action Recognition Networks. DOI :10.48550/arXiv.1612.00881.
- Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet : A Large-Scale Hierarchical Image Database. 8 pages.
- Deutsch CV, Wang L (1996) Hierarchical object-based stochastic modeling of fluvial reservoirs. *Mathematical Geology*, 28(7):857–880. DOI :10.1007/BF02066005.
- Di H, Abubakar A (2022a) Estimating subsurface properties using a semisupervised neural network approach. *GEOPHYSICS*, 87(1):IM1–IM10. DOI :10.1190/geo2021-0192.1.
- Di H, Li Z, Abubakar A (2022b) Using relative geologic time to constrain convolutional neural network-based seismic interpretation and property estimation. *GEOPHYSICS*, 87(2):IM25–IM35. DOI :10.1190/geo2021-0257.1.
- Ding C, Ma J (2022) Automatic migration velocity analysis via deep learning. *GEOPHYSICS*, 87(4):U135–U153. DOI :10.1190/geo2020-0947.1.
- Fabien-Ouellet G (2017) Inversion des formes d'ondes complètes viscoélastique. 172 pages.
- Fabien-Ouellet G, Gloaguen E, Giroux B (2017) Time-domain seismic modeling in viscoelastic media for full waveform inversion on heterogeneous computing platforms with OpenCL. *Computers & Geosciences*, 100:142–155. DOI :10.1016/j.cageo.2016.12.004.
- Fabien-Ouellet G, Sarkar R (2020) Seismic velocity estimation : A deep recurrent neural-network approach. *GEOPHYSICS*, 85(1):U21–U29. DOI :10.1190/geo2018-0786.1.
- Fabien-Ouellet G, Simon J (2022a) *GeoFlow*.
- Fabien-Ouellet G, Simon J (2022b) *ModelGenerator*.
- Fang W, Fu L, Zhang M, Li Z (2021) Seismic data interpolation based on U-net with texture loss. *GEOPHYSICS*, 86(1):V41–V54. DOI :10.1190/geo2019-0615.1.
- Fedorov V, Mannino F, Zhang R (2009) Consequences of dichotomization. *Pharmaceutical Statistics*, 8(1):50–61. DOI :10.1002/pst.331.
- Fehler M, Keliher PJ (2011) *SEAM Phase 1 : Challenges of Subsalt Imaging in Tertiary Basins, with Emphasis on Deepwater Gulf of Mexico*. General Series. Society of Exploration Geophysicists.
- Feng R, Balling N, Grana D, Dramsch JS, Hansen TM (2021a) Bayesian Convolutional Neural Networks for Seismic Facies Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(10):8933–8940. DOI :10.1109/TGRS.2020.3049012.

- Feng R, Grana D, Balling N (2021b) Uncertainty quantification in fault detection using convolutional neural networks. *GEOPHYSICS*, 86(3):M41–M48. DOI :10.1190/geo2020-0424.1.
- Feng Z, Schuster GT (2017) Elastic least-squares reverse time migration. *GEOPHYSICS*, 82(2): S143–S157. DOI :10.1190/geo2016-0254.1.
- Gan C, Schwartz J, Alter S, Mrowca D, Schrimpf M, Traer J, De Freitas J, Kubilius J, Bhandwaldar A, Haber N, Sano M, Kim K, Wang E, Lingelbach M, Curtis A, Feigelis K, Bear DM, Gutfreund D, Cox D, Torralba A, DiCarlo JJ, Tenenbaum JB, McDermott JH, Yamins DLK (2021) ThreeDWorld : A Platform for Interactive Multi-Modal Physical Simulation. DOI :10.48550/arXiv.2007.04954.
- Gao H, Wu X, Liu G (2021) ChannelSeg3D : Channel simulation and deep learning for channel interpretation in 3D seismic images. *GEOPHYSICS*, 86(4):IM73–IM83. DOI :10.1190/geo2020-0572.1.
- Gao K, Huang L, Zheng Y, Lin R, Hu H, Cladohous T (2022a) Automatic fault detection on seismic images using a multiscale attention convolutional neural network. *GEOPHYSICS*, 87(1):N13–N29. DOI :10.1190/geo2020-0945.1.
- Gao Z, Hu S, Li C, Chen H, Jiang X, Pan Z, Gao J, Xu Z (2022b) A Deep-Learning-Based Generalized Convolutional Model For Seismic Data and Its Application in Seismic Deconvolution. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–17. DOI :10.1109/TGRS.2021.3076991.
- Geifman Y, El-Yaniv R, Uziel G (2019) BIAS-REDUCED UNCERTAINTY ESTIMATION FOR DEEP NEURAL CLASSIFIERS. 14 pages.
- Gers F, Schmidhuber J, Cummins F (2000) Learning to Forget : Continual Prediction with LSTM. *Neural computation*, 12:2451–71. DOI :10.1162/089976600300015015.
- Giroux B (2019) 1 - Les ondes sismiques. *GEO1303 – Méthodes sismiques*.
- Giroux B, Duchesne MJ (2019) Interprétation. *GEO1303 – Méthodes sismiques*.
- Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, pages 249–256.
- Goodfellow I, Bengio Y, Courville A (2016) *Deep learning*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts.
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, Curran Associates, Inc., volume 27.
- Gui J, Sun Z, Wen Y, Tao D, Ye J (2020) A Review on Generative Adversarial Networks : Algorithms, Theory, and Applications.
- Guitton A (2004) Amplitude and kinematic corrections of migrated images for nonunitary imaging operators. *GEOPHYSICS*, 69(4):1017–1024. DOI :10.1190/1.1778244.
- He K, Gkioxari G, Dollár P, Girshick R (2018) Mask R-CNN. *arXiv :1703.06870 [cs]*.

- Hill SJ, Rüger A (2019) *Illustrated Seismic Processing : Volume 1 : Imaging*. Course Notes Series. Society of Exploration Geophysicists.
- Ho TK (1995) Random decision forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1.
- Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780. DOI :10.1162/neco.1997.9.8.1735.
- Hong-Qiao X, Xiao-Yi W, Chen-Yuan W, Jiang-Jie Z (2021) Sparse Constrained Least-Squares Reverse Time Migration Based on Kirchhoff Approximation. *Frontiers in Earth Science*, 9.
- Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366. DOI :10.1016/0893-6080(89)90020-8.
- Huang L, Polanco M, Clee TE (2018) Initial Experiments on Improving Seismic Data Inversion with Deep Learning. *2018 New York Scientific Data Summit (NYSDDS)*, pages 1–3.
- Hutchinson DR, Poag C, Johnson AH, Popenoe P, Wright C (1997) Geophysical database of the east coast of the United States; southern Atlantic margin, stratigraphy and velocity in map grids. U.S. Geological Survey.
- Hwang H, Jang C, Park G, Cho J, Kim IJ (2021) ElderSim : A Synthetic Data Generation Platform for Human Action Recognition in Eldercare Applications. *IEEE Access*, pages 1–1. DOI :10.1109/ACCESS.2021.3051842.
- Jin KH, McCann MT, Froustey E, Unser M (2017) Deep Convolutional Neural Network for Inverse Problems in Imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522. DOI :10.1109/TIP.2017.2713099.
- Karras T, Laine S, Aila T (2018) A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv :1812.04948 [cs, stat]*.
- Kazei V, Ovcharenko O, Plotnitskii P, Peter D, Zhang X, Alkhalifah T (2021) Mapping full seismic waveforms to vertical velocity profiles by deep learning. *GEOPHYSICS*, 86(5):R711–R721. DOI :10.1190/geo2019-0473.1.
- Kiefer J, Wolfowitz J (1952) Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, 23(3):462–466. DOI :10.1214/aoms/1177729392.
- Kim B (2015) *Interactive and interpretable machine learning models for human machine collaboration*. Thesis, Massachusetts Institute of Technology.
- Kim S, Jee Seol S, Byun J, Oh S (2022a) Extraction of diffractions from seismic data using convolutional U-net and transfer learning. *GEOPHYSICS*, 87(2):V117–V129. DOI :10.1190/geo2020-0847.1.
- Kim Yw, Mishra S, Jin S, Panda R, Kuehne H, Karlinsky L, Saligrama V, Saenko K, Oliva A, Feris R (2022b) How Transferable are Video Representations Based on Synthetic Data? *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Kingma DP, Ba J (2017) Adam : A Method for Stochastic Optimization. *arXiv :1412.6980 [cs]*.

- Kolyshkina I, Simoff S (2021) Interpretability of Machine Learning Solutions in Public Healthcare : The CRISP-ML Approach. *Frontiers in Big Data*, 4.
- Laakso V (2019) *Testing of reflection seismic, GPR and magnetic methods for mineral exploration and mine planning at the Siilinjärvi phosphate mine site in Finland*. Thèse de doctorat, University of Helsinki, Helsinki, Finland.
- LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature*, 521(7553):436–444. DOI :10.1038/nature14539.
- Lewis W, Vigh D (2017) Deep learning prior models from seismic images for full-waveform inversion. *SEG Technical Program Expanded Abstracts 2017*, Society of Exploration Geophysicists, SEG Technical Program Expanded Abstracts, pages 1512–1517.
- Li F, Zhou H, Wang Z, Wu X (2021a) ADDCNN : An Attention-Based Deep Dilated Convolutional Neural Network for Seismic Facies Analysis With Interpretable Spatial–Spectral Maps. *IEEE Transactions on Geoscience and Remote Sensing*, 59(2):1733–1744. DOI :10.1109/TGRS.2020.2999365.
- Li S, Liu B, Ren Y, Chen Y, Yang S, Wang Y, Jiang P (2020) Deep-Learning Inversion of Seismic Data. *IEEE Transactions on Geoscience and Remote Sensing*, 58(3):2135–2149. DOI :10.1109/TGRS.2019.2953473.
- Li X, Xiong H, Li X, Wu X, Zhang X, Liu J, Bian J, Dou D (2021b) Interpretable Deep Learning : Interpretation, Interpretability, Trustworthiness, and Beyond. *arXiv :2103.10689 [cs]*.
- Li Y, Alkhalifah T, Zhang Z (2021c) Deep-learning assisted regularized elastic full waveform inversion using the velocity distribution information from wells. *Geophysical Journal International*, 226(2):1322–1335. DOI :10.1093/gji/ggab162.
- Li Y, Demanet L (2017) Extrapolated full-waveform inversion : An image-space approach. *SEG Technical Program Expanded Abstracts 2017*, Society of Exploration Geophysicists, SEG Technical Program Expanded Abstracts, pages 1682–1686.
- Li YE, Demanet L (2016) Full-waveform inversion with extrapolated low-frequency data. *GEOPHYSICS*, 81(6):R339–R348. DOI :10.1190/geo2016-0038.1.
- Lin TY, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollár P (2015) Microsoft COCO : Common Objects in Context. *arXiv :1405.0312 [cs]*.
- Lin Y, Zhang Z (2019) Learning or memorizing : A study of generalization issues on data-driven seismic full-waveform inversion. *SEG Technical Program Expanded Abstracts 2019*, Society of Exploration Geophysicists, pages 2644–2648, San Antonio, Texas.
- Lipton ZC, Berkowitz J, Elkan C (2015) A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv :1506.00019 [cs]*.
- Little J, Verri A (1989) Analysis of differential and matching methods for optical flow. *Workshop on Visual Motion [1989] Proceedings*, pages 173–180.
- Liu B, Yang S, Ren Y, Xu X, Jiang P, Chen Y (2021a) Deep-learning seismic full-waveform inversion for realistic structural models. *GEOPHYSICS*, 86(1):R31–R44. DOI :10.1190/geo2019-0435.1.



- Liu C, Sun M, Dai N, Wu W, Wei Y, Guo M, Fu H (2022) Deep learning-based point-spread function deconvolution for migration image deblurring. *GEOPHYSICS*, 87(4):S249–S265. DOI :10.1190/geo2020-0904.1.
- Liu Q, Fu L, Zhang M (2021b) Deep-seismic-prior-based reconstruction of seismic data using convolutional neural networks. *GEOPHYSICS*, 86(2):V131–V142. DOI :10.1190/geo2019-0570.1.
- Lowney B, Lokmer I, O'Brien GS (2021) Multi-domain diffraction identification : A supervised deep learning technique for seismic diffraction classification. *Computers & Geosciences*, 155:104845. DOI :10.1016/j.cageo.2021.104845.
- Lucas A, Iliadis M, Molina R, Katsaggelos AK (2018) Using Deep Neural Networks for Inverse Problems in Imaging : Beyond Analytical Methods. *IEEE Signal Processing Magazine*, 35(1):20–36. DOI :10.1109/MSP.2017.2760358.
- Malhotra VM, Carino NJ (2003) *Handbook on Nondestructive Testing of Concrete*. CRC Press, 2e edition.
- Martin G (2004) *The Marmousi2 model, elastic synthetic data, and an analysis of imaging and AVO in a structurally complex environment*. Thèse de doctorat, University of Houston, Houston.
- Martin GS, Wiley R, Marfurt KJ (2006) Marmousi2 : An elastic upgrade for Marmousi. *The Leading Edge*, 25(2):156–166. DOI :10.1190/1.2172306.
- Molnar C (2018) *Interpretable Machine Learning (Second Edition)*. Leanpub.
- Mosser L, Dubrule O, Blunt MJ (2020a) Stochastic Seismic Waveform Inversion Using Generative Adversarial Networks as a Geological Prior. *Mathematical Geosciences*, 52(1):53–79. DOI :10.1007/s11004-019-09832-6.
- Mosser L, Kimman W, Dramsch J, Purves S, Briceño ADIF, Ganssle G (2018) Rapid Seismic Domain Transfer : Seismic Velocity Inversion and Modeling Using Deep Generative Neural Networks. *80th EAGE Conference and Exhibition 2018*, European Association of Geoscientists & Engineers, volume 2018, pages 1–5.
- Mosser L, Purves S, Naeini EZ (2020b) Deep Bayesian Neural Networks for Fault Identification and Uncertainty Quantification. *First EAGE Digitalization Conference and Exhibition*, European Association of Geoscientists & Engineers, volume 2020, pages 1–5.
- Nevatia R, Binford TO (1977) Description and recognition of curved objects. *Artificial Intelligence*, 8(1):77–98. DOI :10.1016/0004-3702(77)90006-6.
- Olah C (2015) *Understanding LSTM Networks – colah's blog*. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- Operto S, Gholami Y, Prioux V, Ribodetti A, Brossier R, Metivier L, Virieux J (2013) A guided tour of multiparameter full-waveform inversion with multicomponent data : From theory to practice. *The Leading Edge*, 32(9):1040–1054. DOI :10.1190/tle32091040.1.
- Opitz D, Maclin R (1999) Popular Ensemble Methods : An Empirical Study. *Journal of Artificial Intelligence Research*, 11:169–198. DOI :10.1613/jair.614.

- Ovcharenko O, Kazei V, Kalita M, Peter D, Alkhalifah T (2019) Deep learning for low-frequency extrapolation from multioffset seismic data. *GEOPHYSICS*, 84(6):R1001–R1013. DOI :10.1190/geo2018-0884.1.
- Park MJ, Sacchi MD (2020) Automatic velocity analysis using convolutional neural network and transfer learning. *GEOPHYSICS*, 85(1):V33–V43. DOI :10.1190/geo2018-0870.1.
- Pham N, Fomel S (2020) Uncertainty estimation using Bayesian convolutional neural network for automatic channel detection. *SEG Technical Program Expanded Abstracts 2020*, Society of Exploration Geophysicists, SEG Technical Program Expanded Abstracts, pages 3462–3466.
- Pham N, Li W (2022) Physics-constrained deep learning for ground roll attenuation. *GEOPHYSICS*, 87(1):V15–V27. DOI :10.1190/geo2020-0691.1.
- Pradhan A, Mukerji T (2022) Consistency and prior falsification of training data in seismic deep learning : Application to offshore deltaic reservoir characterization. *GEOPHYSICS*, 87(3):N45–N61. DOI :10.1190/geo2021-0568.1.
- Ren P, Xiao Y, Chang X, Huang PY, Li Z, Gupta BB, Chen X, Wang X (2021) *A Survey of Deep Active Learning*.
- Ren Y, Xu X, Yang S, Nie L, Chen Y (2020) A Physics-Based Neural-Network Way to Perform Seismic Full Waveform Inversion. *IEEE Access*, 8:112266–112277. DOI :10.1109/ACCESS.2020.2997921.
- Reynolds D (2015) Gaussian Mixture Models. *Encyclopedia of Biometrics*. Li SZ, Jain AK (éditeurs), Springer US, pages 827–832. Boston, MA.
- Richardson A (2018a). Generative Adversarial Networks for Model Order Reduction in Seismic Full-Waveform Inversion. *arXiv :1806.00828 [physics]*.
- Richardson A (2018b). Seismic Full-Waveform Inversion Using Deep Learning Tools and Techniques. *arXiv :1801.07232 [physics]*.
- Roncoroni G, Forte E, Bortolussi L, Pipan M (2022). Efficient extraction of seismic reflection with Deep Learning. *Computers & Geosciences*, 166:105190. DOI :10.1016/j.cageo.2022.105190.
- Rosenblatt F (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408. DOI :10.1037/h0042519.
- Ross ZE, Meier MA, Hauksson E (2018a). P Wave Arrival Picking and First-Motion Polarity Determination With Deep Learning. *Journal of Geophysical Research : Solid Earth*, 123(6):5120–5129. DOI :10.1029/2017JB015251.
- Ross ZE, Meier MA, Hauksson E, Heaton TH (2018b). Generalized Seismic Phase Detection with Deep Learning. *Bulletin of the Seismological Society of America*, 108(5A):2894–2901. DOI :10.1785/0120180080.
- Rudolph KW, Goulding FJ (2017). Benchmarking exploration predictions and performance using 20+ yr of drilling results : One company's experience. *AAPG Bulletin*, 101(2):161–176. DOI :10.1306/06281616060.
- Rumelhart DE, Hinton GE, Williams RJ (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536. DOI :10.1038/323533a0.

- Saad OM, Oboué YASI, Bai M, Samy L, Yang L, Chen Y (2022). Self-Attention Deep Image Prior Network for Unsupervised 3-D Seismic Data Enhancement. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14. DOI :10.1109/TGRS.2021.3108515.
- Sauvageau M, Gloaguen E, Claproud M, Lefebvre R, Bêche M (2014). Multimodal reservoir porosity simulation : An application to a tight oil reservoir. *Journal of Applied Geophysics*, 107:71–79. DOI :10.1016/j.jappgeo.2014.05.007.
- Schuster GT (2017). *Seismic Inversion*. Investigations in Geophysics. Society of Exploration Geophysicists.
- Shannon CE (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423. DOI :10.1002/j.1538-7305.1948.tb01338.x.
- Sheriff RE, Geldart LP (1995). *Exploration Seismology*. Cambridge University Press, Cambridge ; New York, 2nd edition.
- Shorten C, Khoshgoftaar TM (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60. DOI :10.1186/s40537-019-0197-0.
- Siahkoobi A, Louboutin M, Herrmann FJ (2019a). The importance of transfer learning in seismic modeling and imaging. *GEOPHYSICS*, 84(6):A47–A52. DOI :10.1190/geo2019-0056.1.
- Siahkoobi A, Verschuur DJ, Herrmann FJ (2019b). Surface-related multiple elimination with deep learning. *SEG Technical Program Expanded Abstracts 2019*, Society of Exploration Geophysicists, pages 4629–4634, San Antonio, Texas.
- Simon J (2022a). *Hierarchical transfer learning for deep learning velocity model building*. <https://github.com/CloudyOverhead/velocity-model-building-using-transfer-learning>.
- Simon J (2022b). *Quality control in deep learning and confidence quantification : seismic velocity regression through classification*. <https://github.com/CloudyOverhead/velocity-regression-via-classification>.
- Simon J (2023). *Domain gaps in velocity model estimation : characterizing the transferability between domains*. <https://github.com/CloudyOverhead/training-domain-in-velocity-model-building>.
- Simon J, Fabien-Ouellet G, Gloaguen E (prêt à soumettre). Domain gaps in velocity model estimation : characterizing the transferability between domains. *Computers & Geosciences*.
- Simon J, Fabien-Ouellet G, Gloaguen E (soumis le 16 décembre 2022). Quality control in deep learning and confidence quantification : seismic velocity regression through classification. *Computers & Geosciences*.
- Simon J, Fabien-Ouellet G, Gloaguen E, Khurjekar I (2023). Hierarchical transfer learning for deep learning velocity model building. *GEOPHYSICS*, pages R79–R93. DOI :10.1190/geo2021-0470.1.
- Simon J, Fabien-Ouellet G, Gloaguen E, Khurjekar I, Araya-Polo M (2021). Velocity model building using transfer learning. *First International Meeting for Applied Geoscience & Energy Expanded Abstracts*, Society of Exploration Geophysicists, SEG Technical Program Expanded Abstracts, pages 3365–3369.

- Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng A, Potts C (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pages 1631–1642, Seattle, Washington, USA.
- Société Géologique de Londres, Fédération Européenne des Géologues, Société Géologique de France, Union Belgo-Luxembourgeoise des Géologues (2015). *La Géologie au service de l'homme*.
- Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014). Dropout : a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Stiglic G, Kocbek P, Fijacko N, Zitnik M, Verbert K, Cilar L (2020). Interpretability of machine learning-based prediction models in healthcare. *WIREs Data Mining and Knowledge Discovery*, 10(5):e1379. DOI :10.1002/widm.1379.
- Sun J, Innanen KA, Huang C (2021a). Physics-guided deep learning for seismic inversion with hybrid training and uncertainty analysis. *GEOPHYSICS*, 86(3):R303–R317. DOI :10.1190/geo2020-0312.1.
- Sun J, Niu Z, Innanen KA, Li J, Trad DO (2019). A theory-guided deep learning formulation of seismic waveform inversion. *SEG Technical Program Expanded Abstracts 2019*, Society of Exploration Geophysicists, SEG Technical Program Expanded Abstracts, pages 2343–2347.
- Sun Y, Araya-Polo M, Williamson P (2021b). Data characterization and transfer learning for DL-driven velocity model building. *First International Meeting for Applied Geoscience & Energy Expanded Abstracts*, Society of Exploration Geophysicists, SEG Technical Program Expanded Abstracts, pages 1475–1479.
- Tan C, Sun F, Kong T, Zhang W, Yang C, Liu C (2018). A Survey on Deep Transfer Learning. *arXiv :1808.01974 [cs, stat]*.
- Tarantola A (1984). Inversion of seismic reflection data in the acoustic approximation. *GEOPHYSICS*, 49(8):1259–1266. DOI :10.1190/1.1441754.
- Torgo L, Gama J (1997). Regression using classification algorithms. *Intelligent Data Analysis*, 1(1):275–292. DOI :10.1016/S1088-467X(97)00013-9.
- Tschannen V, Ghanim A, Ettrich N (2022). Partial automation of the seismic to well tie with deep learning and Bayesian optimization. *Computers & Geosciences*, 164:105120. DOI :10.1016/j.cageo.2022.105120.
- USGS (1978). *Field activity details for field activity 1978-015-fa*.
- Vamaraju J, Sen MK (2019). Unsupervised physics-based neural networks for seismic migration. *Interpretation*, 7(3):SE189–SE200. DOI :10.1190/INT-2018-0230.1.
- van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016a). WaveNet : A Generative Model for Raw Audio. *arXiv :1609.03499 [cs]*.
- van den Oord A, Kalchbrenner N, Kavukcuoglu K (2016b). Pixel Recurrent Neural Networks. *arXiv :1601.06759 [cs]*.

- Varol G, Laptev I, Schmid C, Zisserman A (2021). Synthetic Humans for Action Recognition from Unseen Viewpoints. *International Journal of Computer Vision*, 129(7):2264–2287. DOI :10.1007/s11263-021-01467-7.
- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017). *Attention Is All You Need*.
- Virieux J, Operto S (2009). An overview of full-waveform inversion in exploration geophysics. *GEOPHYSICS*, 74(6):WCC1–WCC26. DOI :10.1190/1.3238367.
- Waldeland AU, Jensen AC, Gelius LJ, Solberg AHS (2018). Convolutional neural networks for automated seismic interpretation. *The Leading Edge*, 37(7):529–537. DOI :10.1190/tle37070529.1.
- Wang B, Li J, Luo J, Wang Y, Geng J (2021a). Intelligent Deblending of Seismic Data Based on U-Net and Transfer Learning. *IEEE Transactions on Geoscience and Remote Sensing*, 59(10):8885–8894. DOI :10.1109/TGRS.2020.3048746.
- Wang B, Zhang N, Lu W, Geng J, Huang X (2020). Intelligent Missing Shots' Reconstruction Using the Spatial Reciprocity of Green's Function Based on Deep Learning. *IEEE Transactions on Geoscience and Remote Sensing*, 58(3):1587–1597. DOI :10.1109/TGRS.2019.2947085.
- Wang K, Hu T, Wang S, Wei J (2022a). Seismic multiple suppression based on a deep neural network method for marine data. *GEOPHYSICS*, 87(4):V341–V365. DOI :10.1190/geo2021-0206.1.
- Wang M, Deng W (2018a). Deep visual domain adaptation : A survey. *Neurocomputing*, 312:135–153. DOI :10.1016/j.neucom.2018.05.083.
- Wang W, McMechan GA, Ma J, Xie F (2021b). Automatic velocity picking from semblances with a new deep-learning regression strategy : Comparison with a classification approach. *GEOPHYSICS*, 86(2):U1–U13. DOI :10.1190/geo2020-0423.1.
- Wang W, Yang F, Ma J (2018b). Velocity model building with a modified fully convolutional network. *SEG Technical Program Expanded Abstracts 2018*, Society of Exploration Geophysicists, pages 2086–2090, Anaheim, California.
- Wang Y, Niu L, Zhao L, Wang B, He Z, Zhang H, Chen D, Geng J (2022b). Gaussian mixture model deep neural network and its application in porosity prediction of deep carbonate reservoir. *GEOPHYSICS*, 87(2):M59–M72. DOI :10.1190/geo2020-0740.1.
- Wang Z, Bovik A, Sheikh H, Simoncelli E (2004). Image quality assessment : from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612. DOI :10.1109/TIP.2003.819861.
- Weiss SM, Indurkha N (1995). Rule-based Machine Learning Methods for Functional Prediction. *Journal of Artificial Intelligence Research*, 3:383–403. DOI :10.1613/jair.199.
- Wolpert D, Macready W (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82. DOI :10.1109/4235.585893.
- Wu D, Yao G, Cao J, Wang Y (2016). Least-squares RTM with L1 norm regularisation. *Journal of Geophysics and Engineering*, 13(5):666–673. DOI :10.1088/1742-2132/13/5/666.

- Wu H, Zhang B (2018). A deep convolutional encoder-decoder neural network in assisting seismic horizon tracking. *arXiv :1804.06814 [physics]*.
- Wu X, Liang L, Shi Y, Fomel S (2019a). FaultSeg3D : Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation. *GEOPHYSICS*, 84(3):IM35–IM45. DOI :10.1190/geo2018-0646.1.
- Wu X, Liang L, Shi Y, Geng Z, Fomel S (2019b). Multitask learning for local seismic image processing : fault detection, structure-oriented smoothing with edge-preserving, and seismic normal estimation by using a single convolutional neural network. *Geophysical Journal International*, 219(3):2097–2109. DOI :10.1093/gji/ggz418.
- Wu Y, Lin Y (2020). InversionNet : An Efficient and Accurate Data-Driven Full Waveform Inversion. *IEEE Transactions on Computational Imaging*, 6:419–433. DOI :10.1109/TCI.2019.2956866.
- Wu Y, McMechan GA (2019c). Parametric convolutional neural network-domain full-waveform inversion. *GEOPHYSICS*, 84(6):R881–R896. DOI :10.1190/geo2018-0224.1.
- Yan S, Sun X, Wu X, Zhang S, Si H (2022). Building subsurface models with horizon-guided interpolation and deep learning : Application to the Volve field data. *GEOPHYSICS*, 87(4):B233–B245. DOI :10.1190/geo2021-0505.1.
- Yang D, Oldenburg DW (2017). 3D inversion of total magnetic intensity data for time-domain EM at the Lalor massive sulphide deposit. *Exploration Geophysics*, 48(2):110–123. DOI :10.1071/EG15070.
- Yang F, Ma J (2019). Deep-learning inversion : A next-generation seismic velocity model building method. *GEOPHYSICS*, 84(4):R583–R599. DOI :10.1190/geo2018-0249.1.
- Yang L, Fomel S, Wang S, Chen X, Chen W, Saad OM, Chen Y (2023). Denoising of distributed acoustic sensing data using supervised deep learning. *GEOPHYSICS*, pages WA91–WA104. DOI :10.1190/geo2022-0138.1.
- Yang L, Wang S, Chen X, Saad OM, Chen W, Oboué YASI, Chen Y (2022). Unsupervised 3-D Random Noise Attenuation Using Deep Skip Autoencoder. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16. DOI :10.1109/TGRS.2021.3100455.
- Yao J, Wang Y (2022). Building a full-waveform inversion starting model from wells with dynamic time warping and convolutional neural networks. *GEOPHYSICS*, 87(2):R223–R230. DOI :10.1190/geo2021-0168.1.
- Yilmaz Ö (2001). *Seismic Data Analysis*. Investigations in Geophysics. Society of Exploration Geophysicists.
- Zhang H, Chen T, Liu Y, Zhang Y, Liu J (2021). Automatic seismic facies interpretation using supervised deep learning. *GEOPHYSICS*, 86(1):IM15–IM33. DOI :10.1190/geo2019-0425.1.
- Zhang Z, Lin Y (2020). Data-Driven Seismic Waveform Inversion : A Study on the Robustness and Generalization. *IEEE Transactions on Geoscience and Remote Sensing*, 58(10):6900–6913. DOI :10.1109/TGRS.2020.2977635.
- Zhang ZD, Alkhalifah T (2019). Regularized elastic full-waveform inversion using deep learning. *GEOPHYSICS*, 84(5):R741–R751. DOI :10.1190/geo2018-0685.1.

Zhao S, Li B, Xu P, Yue X, Ding G, Keutzer K (2021). MADAN : Multi-source Adversarial Domain Aggregation Network for Domain Adaptation. *International Journal of Computer Vision*, 129(8): 2399–2424. DOI :10.1007/s11263-021-01479-3.

Zheng Y, Zhang Q, Yusifov A, Shi Y (2019). Applications of supervised deep learning for seismic interpretation and inversion. *The Leading Edge*, 38(7):526–533. DOI :10.1190/tle38070526.1.

Zhu W, Xu K, Darve E, Beroza GC (2021). A general approach to seismic inversion with automatic differentiation. *Computers & Geosciences*, 151:104751. DOI :10.1016/j.cageo.2021.104751.