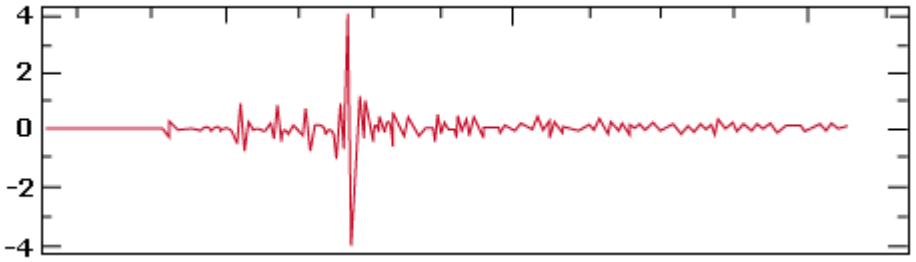
Module OSEC 502 Projet Numérique Transversal
Détermination de polarités d’ondes sismiques par une méthode de deep-learning

Par MOIRROUD Elliot - CERON Franz-Maximilien
Responsable Stéphane Breuils - David Marsan



Sommaire :

Introduction

Dans le domaine de la sismologie, la détermination précise des polarités montantes et descendantes des ondes P est essentielle pour localiser les hypocentres et caractériser les mécanismes focaux des séismes. Cependant, l'analyse manuelle de ces données, bien qu'efficace, s'avère coûteuse en temps et limitée dans sa capacité à traiter des volumes massifs de données générés par les réseaux sismiques modernes. Ce défi a inspiré l'application de méthodes d'apprentissage profond, capables d'exploiter les données à grande échelle tout en offrant une précision comparable à celle des experts humains.

Ce projet utilise Python et TensorFlow pour concevoir une architecture de réseau de neurones convolutifs (CNN) permettant de prédire la polarité des séismes et d'en déterminer la tendance (montante ou descendante) directement à partir des sismogrammes. Notre méthodologie s'inspire des travaux menés par Zachary E. Ross et ses collègues (2018), qui ont démontré que les CNN peuvent surpasser les algorithmes traditionnels et égaler les performances humaines dans la classification des polarités et la détection des arrivées des ondes P.

En nous appuyant sur un ensemble de données massif et bien annoté, nous avons développé un modèle capable de traiter rapidement et précisément les caractéristiques des séismes, ouvrant ainsi la voie à une analyse automatisée à grande échelle. Ce projet vise non seulement à reproduire ces résultats, mais aussi à explorer des améliorations spécifiques pour la détection des polarités montantes et descendantes, en tenant compte des particularités locales et des besoins opérationnels. Pour ce faire, notre objectif est d’entraîner le modèle sur des données de Californie du Sud, identiques à celles utilisées par Zachary E. Ross, avant de valider nos approches en testant le modèle avec des données provenant du Nord du Japon.

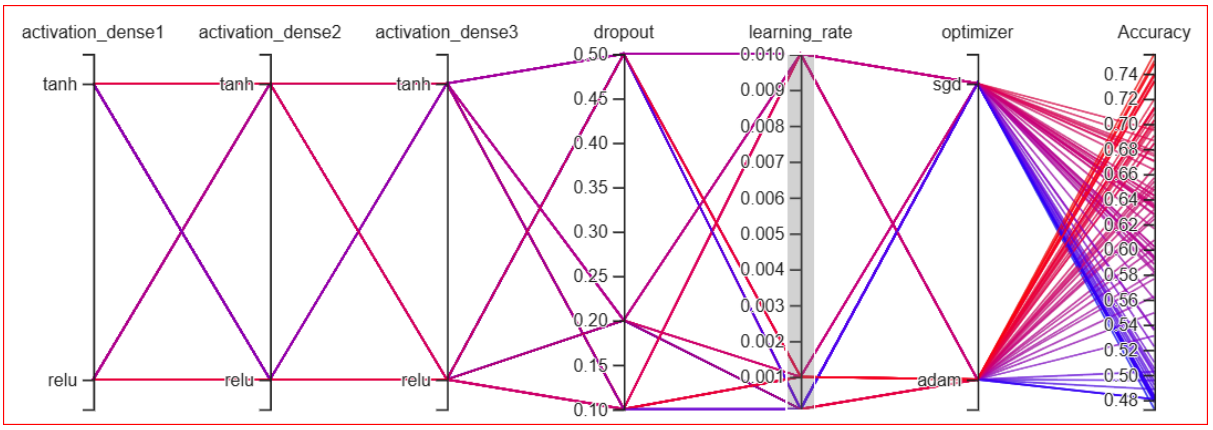
Problématique d'obsolescence des modèles précédents

Au départ, l'objectif de ce projet était sensiblement différent. Nous avions prévu de réutiliser les modèles développés par Zachary E. Ross, disponibles sur son site, et de les tester directement avec des données sismiques du Japon. Cependant, nous avons rapidement rencontré des obstacles liés aux versions obsolètes de TensorFlow utilisées pour ces modèles. Malgré nos efforts pour le contacter afin d'obtenir des versions mises à jour compatibles avec nos outils, ces tentatives n'ont pas abouti.

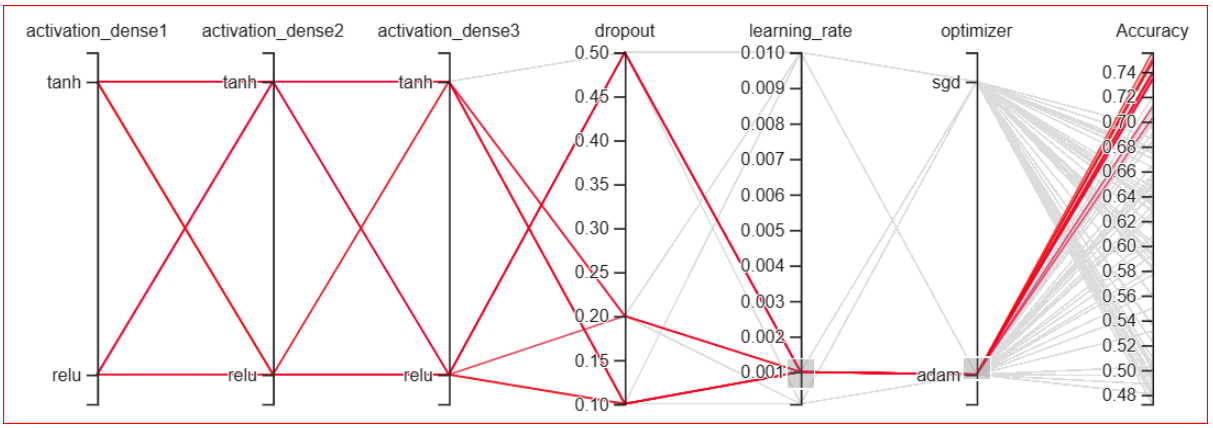
Face à cette impasse, nous avons pris la décision de recommencer le projet à zéro. Cette nouvelle approche nous a conduit à concevoir notre propre architecture de réseau de neurones convolutifs, adaptée aux spécificités des données sismiques et aux exigences technologiques actuelles.

Recherche des hyperparamètres

La recherche des hyperparamètres a constitué une étape importante dans l'optimisation de notre modèle. Pour ce faire, nous avons utilisé TensorBoard afin de suivre et visualiser les performances du réseau au fil des tests. Plusieurs configurations ont été explorées, notamment en jouant sur le nombre de neurones, les types de fonctions d'activation, le taux d'apprentissage (learning_rate), le dropout, ainsi que le choix de l'optimiseur.



Chaque configuration a nécessité l'exécution d'un CNN avec les paramètres modifiés, suivi de l'analyse des graphes générés par TensorBoard. Ces visualisations nous ont permis d'identifier les combinaisons les plus probantes, en observant des métriques comme la perte de validation et la précision au cours des époques. Cette approche méthodique a garanti que les hyper paramètres retenus maximisent les performances du modèle tout en minimisant les risques de surentraînement ou de stagnation.



Développement du modèle CNN

Le développement du modèle CNN a été une étape clé de ce projet. Nous avons utilisé TensorFlow pour concevoir un réseau de neurones convolutifs capable de traiter efficacement les sismogrammes. L'architecture du modèle a été pensée pour extraire les caractéristiques pertinentes des données brutes tout en permettant une classification précise des polarités (montantes, descendantes, ou inconnues).

Pour commencer, les données ont été prétraitées et organisées dans des ensembles d'entraînement et de validation. Cela a permis d'optimiser le processus d'apprentissage et de réduire les risques de surajustement.

Le réseau CNN lui-même est constitué de plusieurs couches successives :

- **Couches de convolution** : Elles détectent des motifs spécifiques dans les données, tels que les variations d'amplitude caractéristiques des ondes P. Des noyaux convolutifs de différentes tailles ont été utilisés pour capturer des détails à différentes échelles.
- **Normalisation par lots (Batch Normalization)** : Cette étape améliore la stabilité et la vitesse de l'apprentissage en normalisant les activations entre les couches.
- **Couches de regroupement (MaxPooling)** : Elles réduisent la taille des données tout en conservant les informations clés, diminuant ainsi la complexité computationnelle.
- **Couches entièrement connectées (Dense)** : Ces couches permettent de combiner les caractéristiques extraites pour réaliser la classification finale en trois classes (up, down, inconnu).

Pour entraîner le modèle, nous avons utilisé l'optimiseur Adam et une fonction de perte basée sur l'entropie croisée. Des mécanismes comme le *Dropout* et le suivi de la perte de validation ont permis d'éviter le surajustement. L'entraînement a été géré sur plusieurs époques, avec une stratégie d'arrêt anticipé pour s'assurer que le modèle atteigne sa meilleure performance sans surajuster.

Enfin, le modèle a été évalué sur un ensemble de validation pour mesurer sa précision et sa capacité de généralisation. Les résultats obtenus ont confirmé l'efficacité de notre approche et la pertinence de notre architecture.

Couche	Type	Paramètres principaux
1	Conv1D	32 filtres, taille noyau=21, activation=ReLU
2	Batch Normalization	-
3	Max Pooling 1D	Taille du pool=2
4	Conv1D	64 filtres, taille noyau=15, activation=ReLU
5	Batch Normalization	-
6	Max Pooling 1D	Taille du pool=2
7	Conv1D	128 filtres, taille noyau=11, activation=ReLU
8	Batch Normalization	-
9	Max Pooling 1D	Taille du pool=2
10	Flatten	-
11	Dense	512 neurones, activation=ReLU
12	Dropout	Taux 0.5
13	Dense	512 neurones, activation=ReLU
14	Dropout	Taux 0.5
15	Dense	3 neurones, activation=Softmax

Générateur de données

Les fichiers de données sismiques utilisés dans ce projet étaient souvent de taille considérable, atteignant plusieurs gigaoctets. Cela posait un défi pour les ressources mémoire et empêchait un chargement complet des données en mémoire vive. Pour surmonter ce problème, nous avons mis en place des générateurs capables de fournir des échantillons de données par lots.

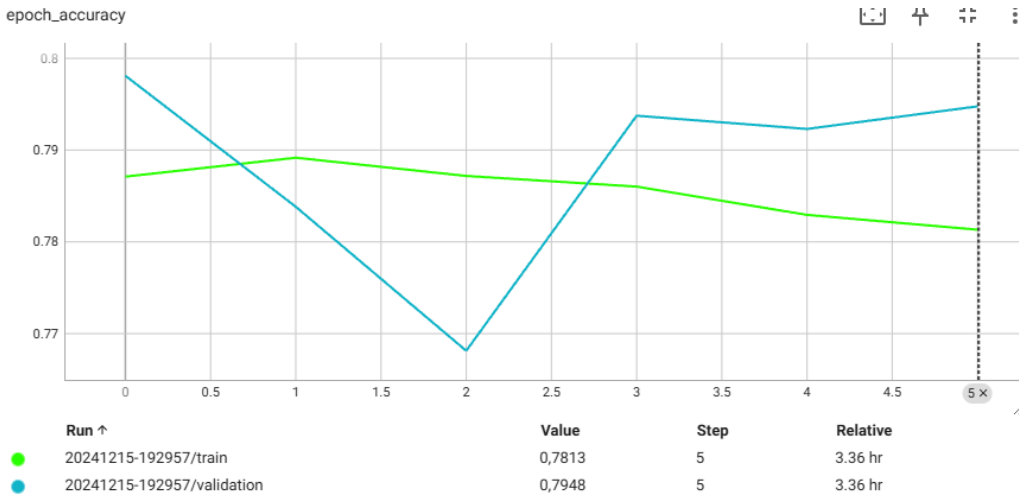
Ces générateurs ont permis de charger uniquement des sous-ensembles de données nécessaires à chaque étape de l'entraînement ou du test du modèle, réduisant ainsi considérablement la pression sur les ressources système. Cette approche a non seulement optimisé l'utilisation de la mémoire, mais elle a également permis une gestion plus fluide des calculs, garantissant que le modèle puisse traiter efficacement les données sans interruption.

Entraînement et test du modèle

L'entraînement de notre modèle CNN a été effectué sur des données sismiques labellisées de Californie. Ces données, précédemment annotées, nous ont permis de tester en continu les performances du modèle à chaque époque grâce à un ensemble de validation distinct. Le processus a été conçu pour éviter les stagnations et maximiser l'efficacité de l'apprentissage.

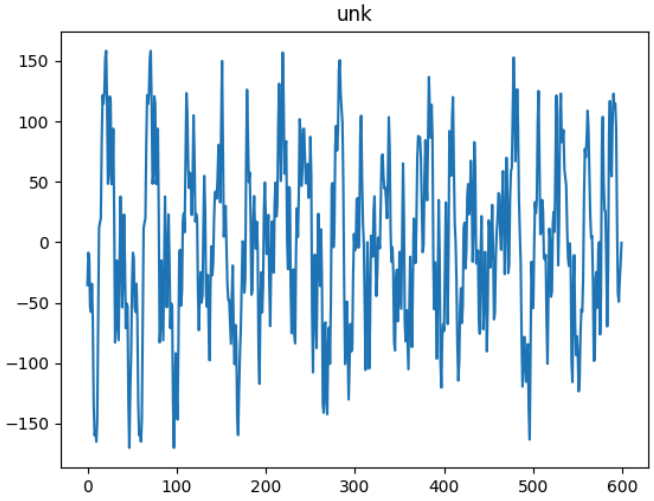
Dès les premières époques, le modèle atteignait rapidement un taux de précision d'environ 80 %. Cependant, après cette phase initiale, les performances commençaient à stagner. Pour remédier à cela, un callback spécifique a été mis en place pour arrêter l'entraînement automatiquement si aucune amélioration significative de la perte de validation n'était observée. Ce mécanisme a évité un gaspillage de ressources et un surentraînement potentiel.

En moyenne, chaque session d'entraînement durait entre 2 et 3 heures, ce qui a permis d'optimiser l'utilisation des ressources tout en assurant une convergence stable. À la fin de l'entraînement, le modèle atteignait une précision stable de 80% sur les données de test, démontrant ainsi sa capacité à généraliser les caractéristiques pertinentes des sismogrammes à partir des données d'entraînement.

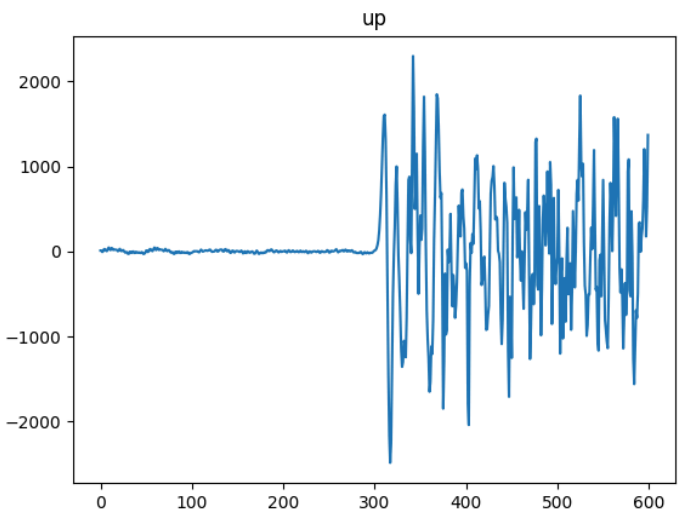


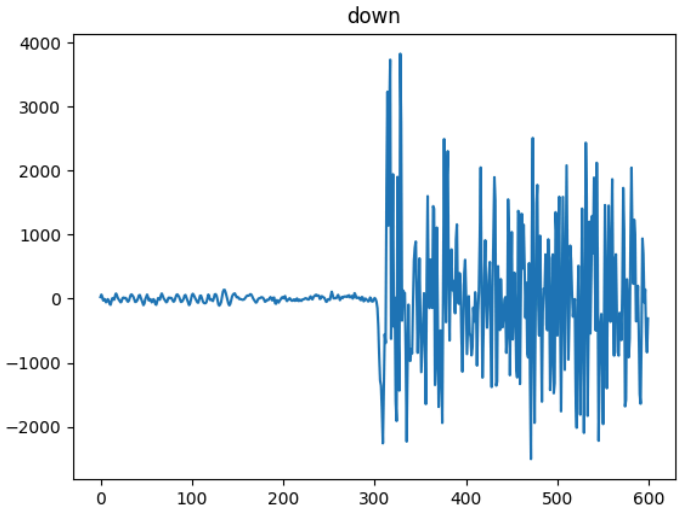
Analyse des données du Japon

Nous avons également reçu un ensemble de données sismiques non labellisées provenant du Japon. Initialement, les résultats obtenus par le modèle sur ces données nous ont préoccupés, car une grande majorité des prédictions étaient classées comme "unknown". Pour comprendre ce phénomène, nous avons visualisé les sismogrammes correspondants et découvert qu'une proportion significative des données était inutilisable en raison d'une qualité médiocre ou de bruits perturbant les signaux d'intérêt.



Cependant, pour les cas où le modèle prédisait une polarité montante ou descendante, les résultats étaient justes et correspondaient aux caractéristiques observées. Nous avons donc entrepris de trier le dataset en supprimant les sismogrammes inutilisables, ce qui nous a permis de créer un sous-ensemble plus propre et exploitable. Avec ce dataset filtré, nous avons pu tester efficacement le modèle, confirmant sa robustesse et sa capacité à généraliser même sur des données provenant d'une région géographique différente.





Conclusion

Ce projet a mis en lumière les défis et opportunités liés à l'utilisation des réseaux de neurones convolutifs pour analyser les données sismiques. Nous avons conçu un CNN performant qui s'est avéré efficace sur des données labellisées de Californie, atteignant une précision de 82 % sur les données de test. L'application à des données non labellisées du Japon a initialement soulevé des préoccupations, mais une analyse approfondie et un tri rigoureux ont permis de valider les capacités du modèle sur des signaux de qualité.

Ces résultats démontrent que les modèles d'apprentissage profond, bien que nécessitant des ajustements importants pour des données brutes ou hétérogènes, constituent une solution puissante pour l'analyse automatisée des sismogrammes. Cette méthodologie ouvre la voie à des applications futures dans des contextes sismiques variés, en optimisant la détection et la classification des polarités pour des régions géographiquement et techniquement diverses.

Annexes

Le but est d’exploiter un programme Python permettant de déterminer la polarité (vers le haut ou vers le bas) des arrivées des ondes P de séismes, ce qui est la 1ere étape pour déterminer la géométrie de la faille ayant rompu. Le logiciel est en accès libre à :

https://scedc.caltech.edu/data/deeplearning.html#picking_polarity

L’application se fera (1) en ré-utilisant les données de l’article (données de Californie du Sud), puis (2) sur des données sismiques au nord Japon.

L’article (Ross et al., 2018) détaillant la méthode est accessible à :

https://www.semanticscholar.org/reader/85df8db729aac6f3a693b0cc4bcc3601fabfdede