Classification de sons audio à l'aide de l'apprentissage profond : une comparaison entre MLP et CNN

DJIGUINE Mamady[1] DIALLO Mohamed[2]

[1] Université Paul Sabatier
[2] Faculté sciences et ingénierie

Master Intelligence Artificielle Fondements et Applications
118, Route de Narbonne, 31062 TOULOUSE CEDEX 9

Résumé

Cet article explore l'application de modèles d'apprentissage profond pour des tâches de classification audio en utilisant des représentations sous forme de spectrogrammes. Plus précisément, nous étudions les performances des architectures de Perceptron Multicouches (MLP) et de Réseau de Neurones Convolutif (CNN) sur un ensemble de données composé de dix classes audio différentes. Nous présentons une méthodologie détaillée pour l'entraînement et l'évaluation des modèles, incluant le prétraitement des données, les architectures des modèles, les protocoles d'entraînement et les métriques d'évaluation des performances. Nos résultats expérimentaux démontrent l'efficacité des CNN par rapport aux MLP pour les tâches de classification audio, avec le CNN atteignant une précision plus élevée sur l'ensemble de test. De plus, nous fournissons des perspectives sur les améliorations potentielles et les orientations de recherche futures dans le domaine de la classification audio en utilisant des techniques d'apprentissage profond.

1 Introduction

L'apprentissage automatique, et plus spécifiquement l'apprentissage profond, a révolutionné de nombreux domaines en permettant aux machines d'apprendre des modèles à partir de données et de les utiliser pour effectuer des tâches complexes telles que la reconnaissance d'images, la traduction automatique, et bien d'autres. L'un des domaines où l'apprentissage automatique montre un grand potentiel est le traitement des signaux audio, où il peut être utilisé pour la classification des sons, la transcription automatique, et même la génération de musique. Dans cet article, nous nous intéressons à la classification de sons audio en utilisant des spectrogrammes, des représentations visuelles du spectre de fréquence d'un signal audio en fonction du temps. Nous explorons deux architectures de réseaux neuronaux, à savoir le Perceptron Multi-Couches (MLP) et le Réseau de Neurones Convolutif (CNN), pour cette tâche de classification[1]. Nous comparons leurs performances sur un corpus de sons audio contenant dix classes différentes, telles que le bruit de tronçonneuse, le tic-tac d'une horloge, le craquement de feu, etc. L'objectif de cette étude est d'évaluer l'efficacité des modèles de classification audio et d'identifier les meilleures pratiques pour cette tâche spécifique. Nous présentons une méthodologie détaillée pour l'entraînement et l'évaluation des modèles, en mettant l'accent sur les différents paramètres et techniques utilisés. Enfin, nous analysons les résultats obtenus et proposons des pistes pour améliorer les performances des modèles.

2 Modèles utilisés

Pour la réalisation de ce travail nous utilisons deux architecture de réseau neurone à savoir Le **MLP** (Percepton Multi-Couche) et un **CNN** (Réseau de Neurone Convolutif) :

2.1 Perceptron Multi-Couches (MLP)

Le Perceptron Multi-Couches est un modèle de réseau de neurones artificiels largement utilisé dans l'apprentissage automatique. Ils sont constitués d'une ou plusieurs couches de neurones. Les données sont transmises à la couche d'entrée, il peut y avoir une ou plusieurs couches cachées fournissant des niveaux d'abstraction, et des prédictions sont faites sur la couche de sortie, également appelée couche visible[2].

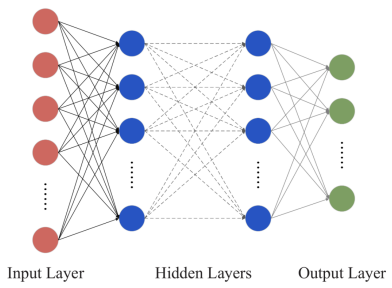

FIGURE 1 – Exemple de réseau MLP. Source :MDPI

2.2 Réseau de Neurones Convolutif (CNN)

Les réseaux de neurones convolutionnels (en anglais Convolutional neural networks), aussi connus sous le nom de CNNs, sont un type spécifique de réseaux de neurones qui sont généralement composés des couches suivantes :

- **La couche de convolution :** applique un ensemble de filtres convolutifs aux images en entrée, chacun d'entre eux activant certaines caractéristiques des images.
- **La couche ReLU :** (Rectified linear unit) favorise un apprentissage plus rapide et plus efficace en remplaçant les valeurs négatives par des zéros et en conservant les valeurs positives. Ce procédé est parfois appelé activation, car seules les caractéristiques activées sont transmises à la couche suivante.
- **La couche de pooling :** simplifie la sortie en réalisant un sous-échantillonnage non linéaire, ce qui permet de réduire le nombre de paramètres que le réseau doit apprendre.

Ces opérations sont répétées sur des dizaines ou des centaines de couches, chaque couche apprenant à identifier différentes caractéristiques[2].

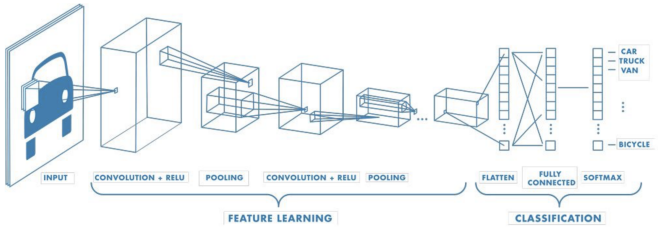

FIGURE 2 – Exemple de réseau comportant de nombreuses couches convolutives. Source : MathWorks

3 Protocol experimental

3.1 Description des données

Les données utilisées dans cette étude sont extraites d'un corpus audio contenant dix classes différentes de sons. Pour prétraiter ces données, nous avons utilisé la bibliothèque librosa en Python, qui offre des fonctionnalités pour charger des fichiers audio, extraire des caractéristiques audio et générer des spectrogrammes. Chaque classe de son est représentée par un ensemble d'échantillons audio au format WAV. Ces échantillons ont été convertis en spectrogrammes, qui sont des représentations visuelles du spectre de fréquence d'un signal audio en fonction du temps. Les spectrogrammes ont été générés en utilisant la fonction **librosa.feature.melspectrogram**, qui calcule le spectrogramme mel d'un signal audio. Chaque spectrogramme est une image de taille 128x216 pixels, où 128 représente le nombre de bandes de fréquence (ou bins) et 216 représente le nombre de trames temporelles. Ces dimensions ont été choisies pour capturer à la fois les informations fréquentielles et temporelles des signaux audio. Le corpus de données est divisé en deux ensembles : un ensemble d'apprentissage et un ensemble de test. L'ensemble d'apprentissage contient 320 exemples, tandis que l'ensemble de test en contient 80. Chaque exemple est associé à une étiquette correspondant à l'une des dix classes de sons. La répartition des données dans chaque ensemble est équilibrée, garantissant ainsi une représentation égale de chaque classe. Cela permet d'éviter tout biais lors de l'entraînement et de l'évaluation des modèles.

	Nombre d'exemples	**Dimension**
Apprentissage	320	128x216
Tests	80	128x216

TABLE 1 – Tableau récapulatif des données

3.2 Architectures des modèles

Perceptron Multi-Couches (MLP). Le modèle MLP comprend une couche cachée avec 50 neurones et une couche de sortie avec 10 neurones correspondant aux classes de sons. La fonction d'activation ReLU est utilisée après la couche cachée pour introduire de la non-linéarité dans le modèle.

Couches	**Nombre de Neurones**	**Activation**
Couche cachée	50	RELU
Couche de sortie	10	-

TABLE 2 – MLP

Réseau de Neurones Convolutif (CNN). Le modèle CNN est composé de quatre (4) couches convolutives suivies de couches de pooling pour réduire la dimensionnalité. Les caractéristiques extraites sont ensuite acheminées vers deux couches entièrement connectées pour la classification.

Couche	**Nb Filtre**	**Taille filtre**	**Activation**
Convolutive 1	8	3x3	LeakyRELU
Convolutive 2	16	3x3	LeakyRELU
Convolutive 3	32	3x3	LeakyRELU
Convolutive 4	64	3x3	LeakyRELU
Maxpooling -	-	2x2	-
Fully connectée(fc1)	50	-	LeakyRELU
Fully connectée(fc2)	10	-	Softmax

TABLE 3 – CNN

Le modèle CNN utilise également une couche de dropout avec une probabilité de 0.5 (pour régulariser le modèle et réduire le surapprentissage) et des couche de normalisation du batch (pour stabiliser et accélérer l'apprentissage en normalisant les activations des couches cachées)

3.3 Méthodologie d'apprentissage

Nous avons divisé nos données en ensembles d'apprentissage et de test, avec **320** exemples dans l'ensemble d'ap-

prentissage et **80** exemples dans l'ensemble de test. Cette division équilibrée nous permet d'évaluer la performance des modèles de manière fiable. Nous avons utilisé un batch size de **32** pour l'apprentissage, ce qui signifie que chaque mise à jour des poids du modèle est effectuée sur un sous-ensemble de **32** exemples à la fois. Cela permet d'accélérer le processus d'apprentissage et de régulariser le modèle en moyennant les gradients calculés sur des mini-lots aléatoires. Nous avons expérimenté deux optimiseurs différents pour l'entraînement des modèles : **SGD** (Stochastic Gradient Descent) et **Adam**(Adaptive Moment Estimation). **SGD** est un optimiseur classique qui met à jour les poids du modèle proportionnellement au gradient de la fonction de perte par rapport à ces poids. **Adam**, quant à lui, est un optimiseur plus sophistiqué qui adapte les taux d'apprentissage pour chaque paramètre du modèle en fonction de l'historique des gradients. Nous avons entraîné chaque modèle pendant un certain nombre d'epochs pour atteindre une précision optimale ce qui signifie que l'ensemble de données a été parcouru au tant de fois (nombre epochs) lors de l'entraînement. À chaque epoch, les poids du modèle sont ajustés pour minimiser la fonction de perte sur l'ensemble d'apprentissage. Après chaque epoch, nous avons évalué les performances des modèles sur l'ensemble de test pour évaluer leur capacité à généraliser à de nouvelles données. Cela nous permet de déterminer si les modèles ont appris des caractéristiques générales des données plutôt que de simplement mémoriser les exemples d'apprentissage.

3.4 Résultats et Interprétation

Après avoir entraîné et évalué nos modèles de classification audio, nous avons obtenu des résultats significatifs qui méritent une analyse détaillée. Voici un résumé des performances de nos modèles et leur interprétation :

Performances des Modèles.

1. **MLP (Perceptron Multi-Couches) :** Avec les Hyper-paramètres : **(Batch_size =32, nb_epoch = 25, learning_rate = 0.0001, optimizer = Adam, weight_dacay = 0.001)**
   — Précision finale sur l'ensemble d'apprentissage : **63.43%**
   — Précision finale sur l'ensemble de test : **41.25%**

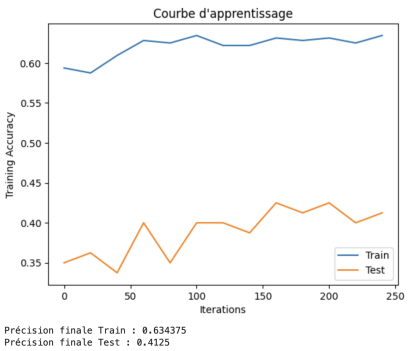

FIGURE 3 – Courbe du train et la validation du MLP

2. **CNN (Réseau de Neurones Convolutif) :** Avec les Hyper-paramètres : **(Batch_size =32, nb_epoch = 20, learning_rate = 0.0001, optimizer = Adam, weight_dacay = 0.0)**
   — Précision finale sur l'ensemble d'apprentissage : **83.43%**
   — Précision finale sur l'ensemble de test : **76.25%**

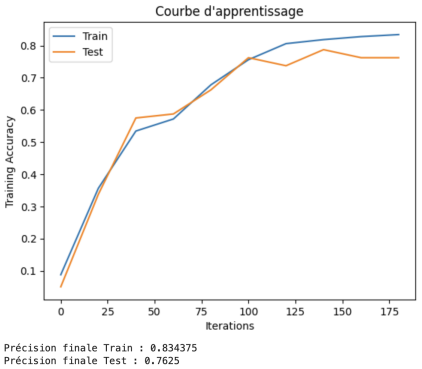

FIGURE 4 – Courbe du train et de la validation du CNN

Analyse des Résultats :.

1. **Performance du MLP :** Le MLP a montré des performances insatisfaisantes, avec une faible précision tant sur l'ensemble d'apprentissage que sur l'ensemble de test. Cela suggère que le MLP n'a pas réussi à capturer les relations complexes présentes dans les données audio, ce qui limite sa capacité à généraliser à de nouvelles données.
2. **Performance du CNN :** Le CNN a affiché des performances nettement meilleures que le MLP, avec une précision plus élevée tant sur l'ensemble d'apprentissage que sur l'ensemble de test. Cela indique que le CNN a pu extraire des caractéristiques discriminantes des spectrogrammes audio grâce à ses couches de convolution et de pooling, ce qui lui a permis d'obtenir de meilleurs résultats de classification.

   Plus loin nous avons fait la matrice de confusion du CNN pour resumer les performances du réseau de neurones convolutifs (CNN) sur la classification des différents sons convertis en spectrogrammes. Cette matrice montre 10 classes différentes qui correspondent à différents sons : tronçonneuse (chainsaw), tic-tac d'horloge (clock_tick), crépitement de feu (crackling_fire), bébé qui pleure (crying_baby), chien (dog), hélicoptère (helicopter), pluie (rain), coq (rooster), vagues de mer (sea_waves), et éternuement (sneezing).

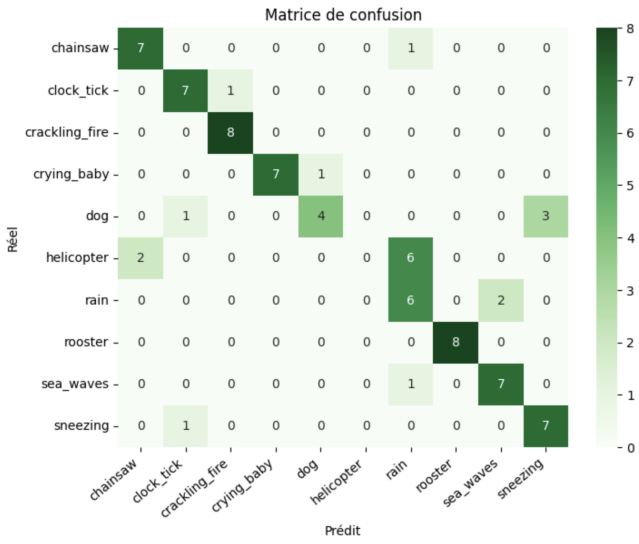

FIGURE 5 – Matrice de confusion du CNN

— **Sur la diagonale, on peut observer le nombre de prédictions correctes pour chaque classe. Par exemple, le CNN a correctement identifié 7 spectrogrammes de tronçonneuse, 7 de tic-tac d'horloge, 8 de crépitement de feu.**

— **Les cases hors diagonale montrent le nombre d'erreurs de classification. Par exemple, un son de tronçonneuse a été confondu avec un éternuement, un son d'hélicoptère a été confondu à deux reprises avec un son de chien, et un son de pluie a été confondu à deux reprises avec un son de coq.**

— **Les couleurs varient du vert au blanc, où le vert foncé représente un nombre plus élevé de prédictions (comme indiqué par l'échelle de couleur à droite), et le blanc représente l'absence de prédictions pour cette combinaison de classe réelle et prédite.**

4 Conclusion et Perspectives

4.1 Comparaison des Modèles

La comparaison des performances entre le MLP et le CNN met en évidence l'importance des architectures de réseaux neuronaux dans la classification audio. Les CNN, en exploitant la structure spatiale des spectrogrammes, ont démontré une meilleure capacité à apprendre des modèles de classification audio que les MLP, qui ne prennent pas en compte cette structure.

4.2 Améliorations Possibles

Pour améliorer davantage les performances des modèles, des techniques telles que l'augmentation des données (que nous avons pas exploré)[3], et l'exploration d'architectures plus complexes peuvent être envisagées[4] [5]. De plus, l'utilisation de techniques telles que le earlier stopping pourrait aider à prévenir le su-rajustement et à améliorer la généralisation des modèles.

En conclusion, notre étude comparative entre le MLP et le CNN pour la classification de sons audio montre que le CNN est nettement plus performant que le MLP. Cela montre que l'utilisation de l'architecture CNN, avec sa capacité à extraire automatiquement les caractéristiques spatiales, est plus adaptée à cette tâche. Cependant, il reste encore des possibilités d'amélioration en ajustant et en explorant d'autres architectures de réseaux neuronaux [4] [5]. Cette étude ouvre la voie à de futures recherches pour améliorer la classification de sons audio en utilisant des techniques d'apprentissage automatique profond.

Références

1. BOTALB, A., MOINUDDIN, M., AL-SAGGAF, U. M. & ALI, S. S. A. *Contrasting Convolutional Neural Network (CNN) with Multi-Layer Perceptron (MLP) for Big Data Analysis* in *2018 International Conference on Intelligent and Advanced System (ICIAS)* 2018 International Conference on Intelligent and Advanced System (ICIAS) (août 2018), 1-5. https://ieeexplore.ieee.org/abstract/document/8540626 (2024).
2. PARDEDE, J. & PUROHITA, A. *Hyperparameter Search for CT-Scan Classification Using Hyperparameter Tuning in Pre-Trained Model CNN With MLP* in *2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM)* 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM) (oct. 2022), 01-08. https://ieeexplore.ieee.org/abstract/document/10034878 (2024).
3. SHORTEN, C. & KHOSHGOFTAAR, T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* **6,** 60. ISSN : 2196-1115. https://doi.org/10.1186/s40537-019-0197-0 (2024) (6 juill. 2019).
4. PELLEGRINI, T. *Deep-Learning-Based Central African Primate Species Classification with MixUp and SpecAugment* in *Interspeech 2021* Interspeech 2021 (ISCA, 30 août 2021), 456-460. https://www.isca-archive.org/interspeech_2021/pellegrini21_interspeech.html (2024).
5. SCHMID, F., KOUTINI, K. & WIDMER, G. Efficient Large-Scale Audio Tagging via Transformer-to-CNN Knowledge Distillation. *Journal of Artificial Intelligence Research* **20,** 245-267 (2024).