L'impact de l'augmentation de données sur la classification audio

Cyril Shalaby **Laetitia Granjon**

Abstract

La compétition BirdCLEF 2024 vise à promouvoir la recherche et le développement de solutions innovantes pour le suivi écologique. En classifiant les chants d'oiseaux, nous pouvons obtenir des données précieuses sur la biodiversité et l'impact des initiatives de restauration de la nature. Cela permet de suivre les populations d'oiseaux et de mesurer le succès des efforts de conservation.

Mots- clés: Réseaux de neurones, CNN, Classification, Kaggle, BirdCLEF2024

1 Introduction

Ce projet se concentre sur la classification des enregistrements audio d'oiseaux en utilisant un réseau de neurones convolutif (CNN). Les enregistrements audio proviennent de la base de données BirdCLEF 2024, et les spectrogrammes de Mel sont utilisés comme caractéristiques d'entrée pour le réseau. Nous utilisons PyTorch pour implémenter le modèle et torchaudio pour la manipulation des fichiers audio.

2 Méthodes

Dans cette section, nous décrivons les étapes de préparation des données, l'architecture du réseau de neurones convolutif et les détails de l'entraînement et de la validation du modèle.

2.1 Préparation des données

Les données sont extraites des dossiers d'enregistrements audio "/kaggle/input/birdclef-2024/train_audio/asbfly" et "/kaggle/input/birdclef-2024/train_audio/ashdro1". Nous utilisons des transformations de spectrogrammes de Mel pour convertir les enregistrements audio en représentations fréquentielles appropriées pour l'entrée dans le CNN.

2.2 Le modèle

2.2.1 Recherches

Avant de sélectionner le modèle final, nous avons mené une recherche sur divers modèles de deep learning pour la classification. Parmi ceux-ci, les Convolutional Neural Networks (CNN) se sont démarqués pour leur capacité à extraire automatiquement des caractéristiques pertinentes des données et leur robustesse aux transformations spatiales. Les travaux de LeCun, Hinton, et Bengio ont particulièrement influencé notre choix en raison de leurs contributions significatives au domaine de l'apprentissage profond.

2.2.2 Test de notre hypothèse sur un autre dataset

Pour évaluer l'efficacité des CNN dans un contexte différent, nous avons testé notre modèle sur un dataset de fichiers audios contenant divers bruits de villes (klaxon, travaux, voitures qui passent, enfants qui crient). Les résultats obtenus étaient prometteurs, démontrant une capacité robuste du modèle à classifier efficacement les différents types de bruits urbains. Ce test a renforcé notre conviction que les CNN sont bien adaptés pour les tâches de classification complexes.

2.2.3 Choix du modèle et adaptation pour le nouveau dataset

Nous avons choisi d'utiliser un modèle CNN pour notre tâche de classification. Pour adapter le modèle au format de données spécifique de la compétition, nous avons modifié la façon de récupérer les données en raison des différences d'architecture du dataset. Nous avons enlevé la couche Softmax, jugée redondante avec la fonction de perte cross entropy utilisée. Nous avons également enelvé les couches de convolution qui nous rendait des erreurs de dimensions matricielles.

2.2.4 Architecture du modèle

Nous avons intercalé des couches de convolution avec d'autres types de couches pour tenter d'optimiser les perfomances du réseau de neurones sur les données audio complexes du dataset. Les types de couches sont les suivantes

- **Convolution et activation** : Les données d'entrée passent par la première couche de convolution, suivie par l'activation ReLU.
- **Pooling** : Une opération de max-pooling réduit la dimension spatiale des cartes de caractéristiques.
- **Deuxième Convolution et activation** : Les données passent par la deuxième couche de convolution, suivie par une nouvelle activation ReLU et une autre opération de max-pooling.
- **Aplatissement** : Les cartes de caractéristiques résultantes sont aplaties en un vecteur.
- **Fully Connected Layers** : Les vecteurs passent par deux couches entièrement connectées, avec une activation ReLU après la première couche.

2.3 Entraînement et Validation

Nous utilisons une validation croisée avec 2 plis pour évaluer les performances du modèle. L'optimiseur Adam est utilisé

avec une fonction de perte de cross-entropie pour entraîner le modèle. Les paramètres d'entraînement incluent un taux d'apprentissage de 0,001 et un nombre d'époques réduits pour accélérer l'entraînement.

3 Points de blocages rencontrés

3.1 Erreurs de dimensions

Lors du développement et de l'entraînement de notre modèle CNN, nous avons rencontré des erreurs de dimensions qui ont provoqué des interruptions dans le processus de prédiction. Les dimensions des tenseurs après chaque couche de convolution étaient cruciales pour le bon fonctionnement du modèle. Les formes suivantes étaient observées :

- Shape after conv1: torch.Size([4, 32, 32, 16])
- Shape after conv2: torch.Size([4, 64, 16, 8])
- Shape after conv3: torch.Size([4, 128, 8, 4])
- Shape after conv4: torch.Size([4, 256, 4, 2])

Ces dimensions représentent respectivement le lot de données (batch size), le nombre de filtres (ou canaux de sortie), ainsi que les dimensions spatiales de la sortie (hauteur et largeur). L'erreur est survenue parce que les dimensions attendues après chaque couche de convolution ne correspondaient pas à celles obtenues, entraînant des incompatibilités dans les opérations subséquentes du réseau. L'erreur est surement liée à une mauvaise configuration de certains paramètres (filtres, stride, padding), et qui causent des discordances entre mes dimensions attendues et résultantes. On a donc changé un petit peu les couches de neurones.

3.2 Problèmes de prédictions du à l'insuffisance des données d'entraînement

Une autre difficulté majeure rencontrée lors du développement de notre modèle CNN concernait la qualité des prédictions. Les prédictions sont incorrectes, car nous avons entraîné sur un seul dossier ("barfly1"). Cela a limité la diversité des données d'entraînement, rendant le modèle incapable de généraliser efficacement à de nouvelles données. le modèle a été limité à un seul dossier en raison du temps de calcul élevé requis pour entraîner sur plusieurs dossiers. Cette limitation a eu pour conséquence directe des prédictions erronées lors de la validation sur de nouveaux échantillons.

3.3 Poblème de surapprentissage

L'une des erreurs critiques rencontrées au cours de notre projet est le surapprentissage (overfitting). Notre modèle a trop bien appris les détails et le bruit de notre dataset, il a donc perdu sa capacité à générer de nouvelles données. On a vu une augmentation continue de la loss et une dégradation des performances de prédictions après un certain point d'entrainement. Cela est sûrement du à la taille de notre dataset. On a sélectionné uniquement un dossier. L'erreur sûrement du au fait que le dataset est sur entrainé sur les fichiers de ce dossier et n'est donc plus capable de reconnaître d'autres fichiers.

4 Etapes suivantes

4.1 Valeur ajoutée de l'entraînement sur un dataset de sons de ville

La zone d'étude pour ce projet est en pleine urbanisation rapide. En enregistrant les données des chants d'oiseaux, il est possible que les enregistrements soient parasités par des bruits urbains inconnus du modèle initial. En entraînant notre modèle CNN sur un dataset comprenant des sons de villes, nous pourrions optimisé sa capacité à distinguer les chants d'oiseaux des bruits urbains. Cette approche permettrait au modèle de capturer une plus grande variété de bruits, augmentant ainsi sa robustesse et sa précision dans des environnements mixtes et bruités.

Ce pré-entraînement sur des sons urbains pourrait garantir que le modèle soit mieux équipé pour faire face à la diversité des bruits environnementaux, améliorant ainsi la fiabilité des classifications dans des contextes d'urbanisation croissante. Cette méthode pourrait permettre de réduire les erreurs de classification causées par les interférences sonores, assurant une surveillance plus précise et fiable des populations d'oiseaux dans des zones en évolution rapide.

Nous pourrions aussi envisager d'entrainer le modèle avec d'autres datasets d'autres sons d'oiseaux ne venant pas du tout de le même région. Ca pourrait permettre au modèle de mieux distinguer les différents types d'oiseaux qui existent dans quelle partie du monde et donc de mieux prédire les oiseaux entendus dans les tests.

4.2 Sous segmentation des données d'entrainement

Pour optimiser la durée d'entraînement du modèle, nous avons envisagé la sous-segmentation des données d'entraînement. Actuellement, certains dossiers contiennent plus de 500 fichiers audio, tandis que d'autres en ont seulement 18.

4.3 Avantages

1. **Réduction de la durée d'entraînement** : En limitant le nombre de fichiers par catégorie, le volume total des données d'entraînement diminue, ce qui réduit le temps nécessaire pour entraîner le modèle.

2. **Équilibrage des classes** : Cela permet d'équilibrer les classes dans le dataset, évitant ainsi un biais vers les classes surreprésentées. Un dataset équilibré peut améliorer les performances de classification et la généralisation du modèle.

3. **Facilitation de la gestion des données** : Un dataset plus petit et équilibré est plus facile à gérer et à manipuler, ce qui peut simplifier les étapes de prétraitement et d'augmentation des données.

4. **Prévention du surapprentissage** : Avec moins de données par classe, le modèle est moins susceptible de mémoriser des détails spécifiques aux échantillons surreprésentés, ce qui peut aider à prévenir le surapprentissage.

5 Résultats

Nous n'avons pas encore de résultats pertinents car comme mentionné précédemment, nous n'avons pas assez entrainé le modèle, il donne donc des résultats "faux".

6 Discussion/Conclusions

6.1 Appendices and acknowledgements

Ce projet démontre l'efficacité des CNN pour la classification des chants d'oiseaux. Les ajustements apportés, tels que l'entraînement sur des sons urbains et la sous-segmentation des données, peuvent permis d'améliorer les performances du modèle. Ces résultats soulignent l'importance de l'adaptation des modèles de deep learning aux spécificités des données et des contextes d'application. Pour des travaux futurs, l'intégration de données supplémentaires et l'optimisation continue des hyperparamètres pourront encore améliorer la robustesse et la précision du modèle.