

5e Année Majeur IA/DATA Science

OUM Anthony

RAKOTOSON Lauren

Projet Deep Learning Avancé 2023

The logo for esiea, featuring the word "esiea" in white lowercase letters on a blue rectangular background.

esiea

1. Quel type d'architecture avez-vous utilisé, pourquoi ?

Nous avons utilisé une architecture de type CNN (Convolutional Neural Network) avec une architecture en cascade. Cette architecture est bien adaptée aux tâches de classification d'images car elle permet de capturer les caractéristiques locales et globales des images. Elle consiste en une succession de couches de convolutions, chacune suivie d'une couche de pooling. Les couches de convolutions permettent de détecter des caractéristiques locales dans les images, tandis que les couches de pooling permettent de réduire la taille de la représentation des images tout en conservant les informations importantes.

Cette architecture est bien adaptée à notre tâche car elle permet de capturer des informations à différents niveaux d'abstraction. Les couches de convolutions inférieures détectent des caractéristiques simples, telles que les bords et les coins, tandis que les couches de convolutions supérieures détectent des caractéristiques plus complexes, telles que les formes, les objets, les caractéristiques softbio, telles que la forme de la bouche, la présence d'une moustache ou d'un chapeau.

2. Combien de convolutions contient votre architecture, pourquoi ?

Nous avons utilisé 3 couches de convolutions, avec 64, 128 et 256 filtres respectivement. Ce choix est basé sur l'observation que les images de la base de données sont de taille relativement petite (80x80 pixels). Une architecture avec trop de couches de convolutions pourrait entraîner une sur-apprentissage.

3. Comment vous êtes-vous préoccupé de l'over-fitting ?

Nous avons utilisé plusieurs techniques pour prévenir l'over-fitting, notamment :

- Une régularisation L2 sur les poids des couches : c'est une technique qui ajoute une pénalité aux poids du modèle. Cela permet d'empêcher le modèle de trop s'ajuster aux données d'apprentissage.
- Une validation croisée : c'est une technique qui consiste à diviser l'ensemble de données en plusieurs ensembles. Le modèle est entraîné sur un ensemble de données et évalué sur un autre ensemble de données. Cela permet d'évaluer les performances du modèle sur des données qu'il n'a pas vues pendant l'entraînement.

4. Est-ce que l'asymétrie de la base d'apprentissage est un problème ? si oui, comment avez traité ?

L'asymétrie de la base d'apprentissage était un problème car elle pouvait entraîner un biais dans le modèle. Dans notre cas, dans le fichier train.txt, nous avons 80% d'images négatives

et 20% d'images positives. En coupant de fichier, nous avons réussi à augmenter le taux d'image positive de 10% Réduisant ainsi ce déséquilibre. On se retrouve avec 70% d'images négatives et 30% d'images positives. Ce déséquilibre pouvait entraîner le modèle à prédire plus souvent la classe négative.

5. Est-ce que le fait que les labels soient bruités a posé problème ? Comment l'avez-vous géré ?

Le fait que les labels soient bruités a posé un problème, car il a rendu plus difficile l'apprentissage du modèle.

Nous avons tenté de résoudre ce problème en utilisant une régularisation L2 sur les poids des couches. Cette technique ajoute une pénalité aux poids du modèle. Cela permet d'empêcher le modèle de trop s'ajuster aux données d'apprentissage, y compris aux données bruitées.

6. Pour les images de la base de test, à partir de la sortie de votre réseau, comment attribuez-vous le label ?

Pour les images de la base de test, nous attribuons le label à l'image en fonction de la sortie du réseau. Plus précisément, nous considérons que l'image appartient à la classe positive (1) si la sortie du réseau est supérieure à true.

7. Comment avez-vous pris en compte les problèmes d'équité de l'algorithme ?

Nous n'avons pas pris en compte les problèmes d'équité de l'algorithme dans notre modèle. Cependant, ce serait une question importante à considérer dans une application réelle.

8. J'ai précisé que le label correspondait à une combinaison de caractéristiques « softbio » et non image. Qu'est-ce que cela change pour vous ?

Cela change beaucoup de choses pour nous. Dans notre cas, le modèle devrait apprendre à combiner les informations provenant des caractéristiques softbio et des images. Ce serait une tâche plus difficile que la classification d'images uniquement.

C'est plus compliqué de faire une vérification à l'œil nu.

9. Avez-vous tenté d'utiliser un modèle pré-appris ? Que vous l'avez fait ou non, comment procéder, choisir le modèle, qu'apprendre... ?

Nous n'avons pas tenté d'utiliser un modèle pré-appris dans cette étude. Cependant, ce serait une bonne idée pour améliorer les performances du modèle. Un modèle pré-appris sur un ensemble de données d'images similaires pourrait fournir une base solide pour l'apprentissage du modèle.

Pour choisir un modèle pré-appris, il faudrait tenir compte de la taille et de la complexité de l'ensemble de données d'images. Un modèle trop petit pourrait ne pas être capable d'apprendre les caractéristiques spécifiques de l'ensemble de données. Un modèle trop complexe pourrait entraîner un sur-apprentissage.

Le modèle pré-appris pourrait être utilisé pour initialiser les poids du modèle. Cela permettrait au modèle d'apprendre plus rapidement et d'améliorer les performances.

10. Avec un mois supplémentaire pour travailler, que feriez-vous pour améliorer vos résultats ?

Avec un mois supplémentaire pour travailler, nous ferions les choses suivantes pour améliorer nos résultats :

- Nous utiliserions un modèle pré-appris
- Nous collecterions plus de données
- Nous essaierions d'améliorer la régularisation du modèle
- Nous utiliserions la troisième colonne du train.txt afin d'avoir un score de Fair Metric
- Nous utiliserions un appareil plus puissant pour faciliter la vitesse de calcul