

Manual Hyperparameter Search Report

RALIJAONA Rina
M2 MISA

12 Décembre 2024

1 Introduction

Ce rapport présente une analyse des hyperparamètres d'un neural network appliqué à la classification du dataset CIFAR-10. L'objectif est d'examiner l'impact de trois hyperparamètres sur les performances du modèle, en termes de précision et de perte lors de l'entraînement et à la fin de celui-ci :

- Le taux d'apprentissage, avec une évaluation de la perte durant l'entraînement et de la précision finale.
- Les fonctions d'activation, comparant cinq variantes différentes en termes de précision finale.
- Le nombre de neurones cachés, analysant son effet sur la précision finale.

Ces analyses permettent de déterminer les configurations optimales pour une classification efficace du jeu de données CIFAR-10, tout en soulignant les compromis entre capacité d'apprentissage et généralisation.

2 Impact du Learning Rate sur la Loss et l'Accuracy

2.1 Observation

Les graphiques montrent l'effet du taux d'apprentissage sur la perte et la précision de validation.

Le learning rate 10^{-4} donne les meilleurs résultats avec une perte de validation la plus faible (2.0254) et une accuracy maximale (43.02%). Cela suggère que, pour ce modèle et ces données, un taux d'apprentissage très faible permet une optimisation plus précise des paramètres, bien que les différences avec 10^{-3} soient minimales.

En revanche, à 10^{-3} , la perte (2.0256) et l'accuracy (42.99%) sont très proches des valeurs obtenues avec 10^{-4} , montrant une performance également solide. À 10^{-2} , l'accuracy décroît légèrement (42.76%), ce qui indique une efficacité moindre. Enfin, à 10^{-1} , la perte augmente (2.0457) et l'accuracy

chute significativement (41.01%), révélant une divergence due à des oscillations importantes lors de l'entraînement.

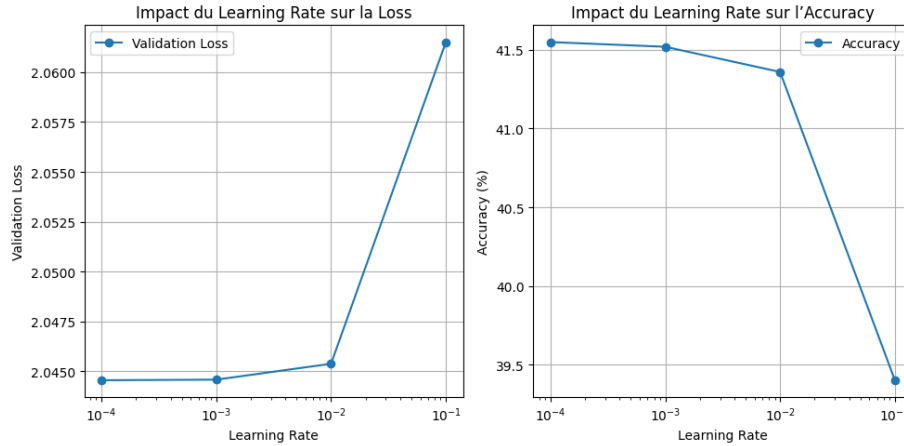


Figure 1: Impact du learning rate sur la loss et l'accuracy.

2.2 Interprétation

Un taux d'apprentissage de 10^{-1} (0.1) ralentit l'apprentissage et entraîne une précision plus faible (41.01%) en raison d'oscillations importantes. Un taux d'apprentissage de 10^{-2} (0.01) améliore légèrement la précision (42.76%), mais la performance n'atteint pas son optimal. Le taux de 10^{-3} (0.001) offre également une amélioration modeste (42.99%), mais c'est avec un taux de 10^{-4} (0.0001) que la meilleure performance est atteinte (43.02%).

3 Performance des Fonctions d'Activation

3.1 Observation

Le graphe en barres illustre les précisions finales pour différentes fonctions d'activation :

- **Tanh** atteint la meilleure précision (14.88%), suivie par **GeLU** et **LeakyReLU** (10.63%), puis par **Sigmoid** (9.94%) et **ReLU** (9.89%).

3.2 Interprétation

La dominance de **Tanh** peut s'expliquer par sa capacité à bien gérer des données normalisées, en particulier si celles-ci sont centrées autour de zéro. Les performances solides de **ReLU** et **LeakyReLU** confirment leur pertinence pour les réseaux profonds modernes, grâce à leur efficacité pour éviter la saturation.

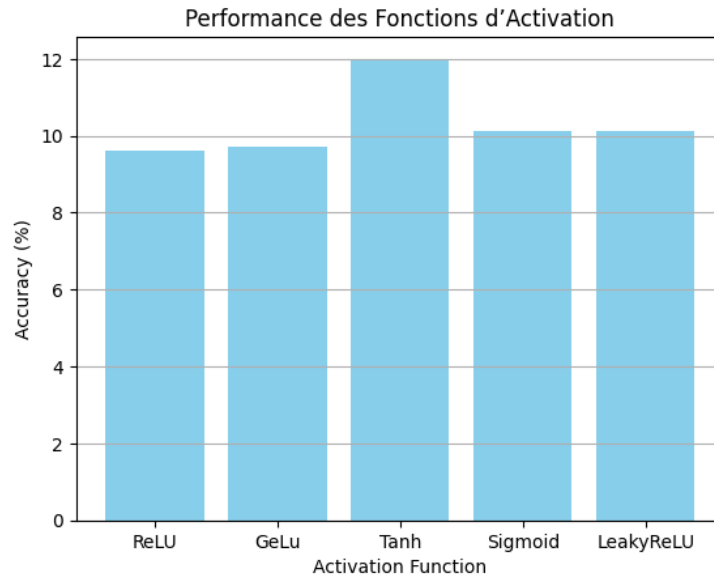


Figure 2: Précision finale des fonctions d'activation.

4 Impact du Nombre de Neurones Cachés

4.1 Observation

Le graphique linéaire révèle l'effet du nombre de neurones cachés sur la précision finale :

- Avec 32 neurones, l'accuracy est d'environ 44.02%, indiquant une capacité d'apprentissage modérée.
- À 64 neurones, la précision atteint un pic (45.42%), représentant un bon compromis entre capacité d'apprentissage et généralisation.
- À 128 neurones, l'accuracy reste élevée (45.62%), montrant une performance stable.
- À 256 neurones, l'accuracy diminue légèrement (43.85%), suggérant une possible instabilité ou un léger surajustement.

4.2 Interprétation

Un nombre insuffisant de neurones (64) limite la complexité des représentations apprises. 128 neurones semble être le choix optimal pour ce modèle et ces données, capturant suffisamment d'information tout en restant généralisable. Trop de neurones (256) peuvent introduire un surajustement, rendant le modèle moins performant sur les données de validation.

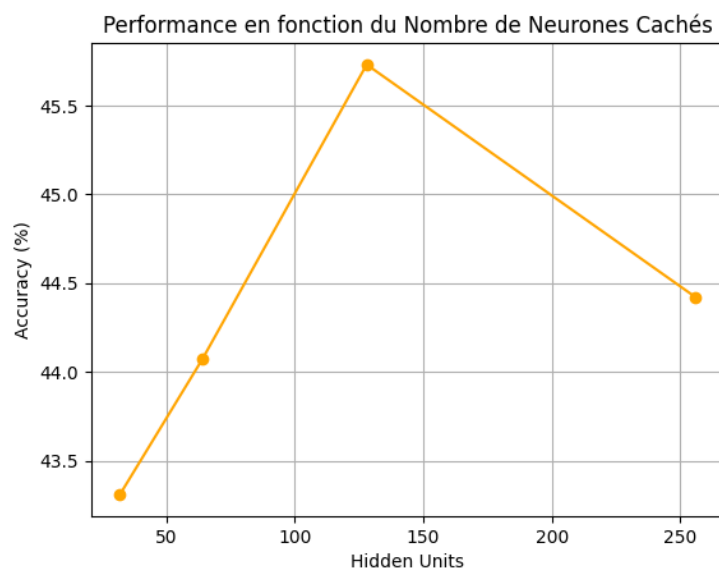


Figure 3: Impact du nombre de neurones cachés sur l'accuracy.

5 Conclusion

Ces résultats montrent l'importance du tuning des hyperparamètres pour améliorer les performances du modèle. En particulier :

- Le learning rate de 10^{-4} est idéal pour un entraînement stable.
- La fonction d'activation **Tanh** et ses alternatives comme **LeakyReLU** et **GeLU** sont robustes.
- Un nombre modéré de 128 neurones cachés est optimal pour ce problème.