

MLP vs KNN

Analyse des résultats obtenus avec KNN

Les résultats montrent une précision globale de 96,31 % pour le modèle KNN appliqué sur la base de données MNIST. Cela signifie que le modèle a correctement classé 96,31 % des 10 000 images de l'ensemble de test. L'analyse par classe met en évidence des variations dans la performance du modèle selon les chiffres :

- ❖ 0 : La précision atteint 99,29 %
- ❖ 1 : La précision atteint 99,65 %
- ❖ 2 : La précision atteint 94,09 %
- ❖ 3 : La précision atteint 96,34 %
- ❖ 4 : La précision atteint 94,91 %
- ❖ 5 : La précision atteint 96,08 %
- ❖ 6 : La précision atteint 98,64 %
- ❖ 7 : La précision atteint 95,33 %
- ❖ 8 : La précision atteint 93,12 %
- ❖ 9 : La précision atteint 95,34 %

En résumé, KNN offre de très bonnes performances globales, mais les variations montrent des points faibles pour certains chiffres, notamment les « 2 » et les « 8 », qui présentent plus de similarités avec d'autres.

MPL

Le programme MLPTester a été conçu pour entraîner et évaluer le perceptron sur les images de la base de données MNIST. L'objectif principal est de tester différentes configurations du réseau ainsi que plusieurs valeurs pour le taux d'apprentissage afin d'analyser leurs performances en termes de précision et d'erreur finale.

Dans un premier temps, les données nécessaires à l'entraînement et aux tests sont chargées. Les fichiers sont divisés en deux ensembles : un ensemble d'entraînement composé de 60 000 images et un ensemble de test contenant 10 000 images.

Ensuite, plusieurs configurations de couches pour le réseau sont définies. Chaque configuration représente une architecture spécifique du réseau, avec un certain nombre de neurones dans chaque couche. Pour chaque configuration, plusieurs valeurs de taux d'apprentissage sont testées, allant de 0,01 à 0,9.

Le programme entraîne le MLP en effectuant une rétropropagation sur les données d'entraînement. À chaque itération, l'erreur moyenne du réseau est calculée et suivie pour

évaluer l'amélioration. Une fois l'entraînement terminé, le réseau est évalué sur les données de test. Pour cela, le programme compare les prédictions du réseau avec les étiquettes réelles des images. La précision est calculée comme le pourcentage d'images correctement classées.

Enfin, les résultats pour chaque combinaison de configuration et de taux d'apprentissage sont enregistrés et affichés. Chaque résultat inclut :

- La configuration des couches du réseau,
- Le taux d'apprentissage utilisé,
- La précision obtenue sur les données de test,
- Et l'erreur finale après entraînement.

Ce processus permet d'identifier les architectures et les paramètres qui offrent les meilleures performances sur le problème donné.

MLP : Analyse des Résultats

- Nombre d'images d'entraînement : 60,000
- Nombre d'images de test : 10,000
- Fonction d'activation utilisée : sigmoid

Configurations testées et résultats :

- **Configuration : [784, 10, 10]**
 - Learning Rate : 0.01
 - Précision : 38.87%
 - Erreur finale : 0.1435
- **Configuration : [784, 64, 10]**
 - Learning Rate : 0.01
 - Précision : 71.29%
 - Erreur finale : 0.0722
- **Configuration : [784, 64, 10]**
 - Learning Rate : 0.1
 - Précision : 33.80%
 - Erreur finale : 0.1425
- **Configuration : [784, 128, 10]**
 - Learning Rate : 0.01
 - Précision : 79.62%
 - Erreur finale : 0.0640
- **Configuration : [784, 128, 10]**
 - Learning Rate : 0.4
 - Précision : 13.09%
 - Erreur finale : 0.1506
- **Configuration : [784, 256, 128, 10]**

- Learning Rate : 0.01
- Précision : 84.96%
- Erreur finale : 0.0422
- **Configuration : [784, 256, 128, 10]**
 - Learning Rate : 0.1
 - Précision : 43.19%
 - Erreur finale : 0.1370
- **Configuration : [784, 512, 256, 128, 10]**
 - Learning Rate : 0.01
 - Précision : 78.68%
 - Erreur finale : 0.0599
- **Configuration : [784, 512, 256, 128, 64, 32, 16, 10]**
 - Learning Rate : 0.01
 - Précision : 10.28%
 - Erreur finale : 0.1799
- **Configuration : [784, 512, 256, 128, 10]**
 - Learning Rate : 0.4
 - Précision : 13.95%
 - Erreur finale : 0.1705

Vous trouverez en annexes tous les graphes avec les différentes configuration et leurs taux d'apprentissages en annexes.

Résultats et observations clés

1. Influence du learning rate

Les performances du modèle varient fortement en fonction du learning rate :

- **Learning rate 0.01** : Ce taux d'apprentissage faible permet d'obtenir les meilleures précisions, jusqu'à **84.96%** pour une architecture optimisée. Cela reflète une convergence plus stable, bien que non parfaite, même sur des modèles plus complexes.
- **Learning rates 0.1, 0.4 et 0.9** : Ces taux plus élevés entraînent une précision beaucoup plus faible (inférieure à 20% dans la majorité des cas), indiquant une incapacité à converger correctement. Ces modèles semblent osciller ou diverger, ce qui explique des erreurs finales relativement élevées.

Conclusion : Un learning rate de **0.01** est le plus adapté dans ce contexte.

2. Influence des couches

L'architecture des couches influence également la précision du modèle :

- **Réseaux avec peu de neurones ([784, 10, 10] et [784, 64, 10])** : Ces configurations atteignent respectivement **38.87%** et **71.29%** de précision avec un learning rate de 0.01. Cela montre qu'un nombre modéré de neurones dans les couches cachées améliore la performance.

- **Réseaux avec plus de couches et plus de neurones ([784, 128, 10] et [784, 256, 128, 10])** : Les performances culminent à **79.62%** et **84.96%** avec un learning rate de 0.01. Ce sont les meilleures performances que l'on a eu pour MLP.
- **Réseaux avec beaucoup de couches ([784, 512, 256, 128, 64, 32, 16, 10])** : Les performances chutent drastiquement (10.28% de précision) ce qui indique que ce n'est pas un bon choix

Conclusion : Une architecture intermédiaire comme **[784, 256, 128, 10]** offre le meilleur compromis entre complexité et précision.

3. Comparaison des erreurs finales

L'erreur finale donne une indication sur la capacité d'un modèle à minimiser la perte :

- Les modèles performants, comme **[784, 256, 128, 10]**, affichent des erreurs finales faibles (0.0422), correspondant à une meilleure optimisation des paramètres.
- Les configurations avec des learning rates élevés ou des architectures profondes ont des erreurs finales élevées.

Comparaison entre KNN et MLP

Pour nos tests, le modèle KNN s'est révélé plus performant que le MLP, atteignant une précision de 96,31 %, contre un maximum de 84,96 % pour le MLP dans la meilleure configuration testée ([784, 256, 128, 10] avec un taux d'apprentissage de 0,01). Cependant on remarque une grande différence de temps de traitement entre les deux modèles. Le modèle KNN a un temps de calcul nettement plus rapide comparé au MLP, malgré son taux de précision supérieur.

En revanche, le MLP, bien que performant dans certaines configurations, nécessite un temps d'entraînement beaucoup plus long, notamment en raison de la recherche des bons paramètres. Cette recherche peut prolonger le processus de manière significative et n'aide pas à réduire le temps total d'exécution.



