

Machine learning

Ivan KRIVOKUCA , Abdel-malik FOFANA et Theophile TAFFOUREAU

TABLE DES MATIERES:

Base MNIST / DGITS :

Dogs&Cats:

Intel Image Classification:

Base MNIST / DGITS :

Voir le notebook '[MNIST_DGITS notebook.ipynb](#)' pour plus de précision et de détail

Court résumé:

Pour commencer, nous avons choisit d'utiliser le modèle de classification CNN en utilisant la convolution avec keras

Nous avons donc pris l'exemple du code dans l'énoncé du projet en le modifiant à chaque fois pour effectuer plusieurs tests

au départ nous avons utilisé de multiples couches de convolution, mais il s'est avéré que le pourcentage d'accuracy etait plutôt faible (environ 50%)

En multipliant les tests, nous avons simplifié le nombre de convolutions ce qui à permit d'améliorer grandement la précision du modèle jusqu'à 95% de précision

Nous avons donc ici les étapes les plus importantes du modèle choisit :

- défini les images de formes $28 * 28 * 1$
- utiliser la fonction d'activation relu pour les couches cachées du réseau de neurones
- ajouter une couche Dense avec 32 neurones
- comme nous avons 10 classes possibles, nous avons ajouter une autre couche Dense de 10 neurones utilisant la fonction d'activation softmax car cette fonction est plus approprié pour le choix de la couche de sortie pour la classification multiclasse

Dogs&Cats:

Voir le notebook '[chien&chat notebook.ipynb](#)' pour plus de précision et de détail

Court résumé:

Nous avons exploré plusieurs approches pour la classification d'images de chats et de chiens.

KMeans avec ORB : En adoptant une approche de classification non supervisée combinant KMeans pour le clustering et ORB pour l'extraction de caractéristiques, nous avons atteint une précision de 68 %.

Cette méthode, bien que plus simple et plus rapide à mettre en œuvre, montre des imprécision quant tenu des autres approches.

CNN avec Classification Binaire : En passant à une approche supervisée avec un réseau de neurones convolutif (CNN) configuré pour la classification binaire et entraîné sur "30 epochs", nous avons constaté une augmentation de la précision, atteignant en moyenne environ 80 %.

CNN avec Classification Catégorielle et Paramètres Avancés : En optimisant le CNN avec des ajustements de paramètres, tels que l'ajout de couches, l'augmentation du nombre de batchs, et l'extension de la période d'entraînement à 40 epochs, la précision a encore augmenté, pour atteindre environ 85 %. Ici la classification catégorielle n'est pas nécessaire du fait du type de données (chien/chats) et donc deux labels

Intel Image Classification:

Court resumé:

Nous avons essayé SVM, cela nous a donné ~65%, nous avons donc décidé de changer de modèle et avons choisi CNN qui nous a donné ~75%.

Pour améliorer le pourcentage, nous avons également augmenté le nombre d'images sur lesquelles le modèle allait s'entraîner, même si le code prenait plus de temps à se lancer. Nous avons joué avec les fonctions d'activation, le nombre d'epochs (10,25,50), etc...

Pour au final avoir environ ~78% de précision.

Pour plus d'information sur SVM sur intel image classification voir le notebook fournis : "[intel-image-classification SVM.ipynb](#)"

Et pour plus d'information sur CNN voir le notebook : "[intel-image-classification CNN Final.ipynb](#) "

version final de mon intel Image classification sur kaggle également disponible ici :

Intel image classification

Explore and run machine learning code with Kaggle Notebooks | Using data from Intel Image Classification

[k https://www.kaggle.com/code/abdelmalikfofana/intel-image-classification](https://www.kaggle.com/code/abdelmalikfofana/intel-image-classification)

