

Réseaux de Neurones - CIFAR-10

L'objectif de ce TP est de mettre en pratique les notions sur les réseaux de neurones en construisant un modèle de classification d'images pour le dataset CIFAR-10.

Introduction et Réseau de Neurones Denses

Les réseaux de neurones fonctionnent en combinant plusieurs neurones pour obtenir un modèle plus complexe. Chaque neurone est une fonction d'activation. Chaque neurone est connecté à tous les neurones de la couche précédente. Il y a plusieurs couches de neurones.

Dataset CIFAR-10

CIFAR-10 est un dataset d'images de 60 000 images de 32x32 pixels en couleur (3 canaux) réparties en 10 classes. Il est divisé en 50 000 images d'entraînement et 10 000 images de test.

1. Récupérez dataset CIFAR-10:

```
import tensorflow as tf
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()
```

2. Explorez le dataset: Regarder les données, la taille des données, le type des données, le nombre de classes, le nombre d'images par classe, etc.
3. Prétraitez des données: Normalisez les données et transformez les labels en one-hot encoding.

Réseau de Neurones Dense

4. En utilisant Keras, construisez un réseau de neurones dense pour la classification d'images. Vous aurez notamment besoin des fonctions suivantes:
 - `Sequential` pour construire le modèle.
 - `Dense` pour construire les couches de neurones.

Réseaux de Neurones Convolutionnels (CNN)

5. En utilisant Keras, construisez un réseau de neurones convolutionnel pour la classification d'images. Vous aurez notamment besoin des fonctions suivantes:
 - `Conv2D` pour construire les couches de convolution.
 - `MaxPooling2D` pour construire les couches de pooling.
 - `Flatten` pour transformer les images en vecteurs.
 - `Activation` pour définir la fonction d'activation.
 - `Adam` pour l'optimisation.
 - `CategoricalCrossentropy` pour la fonction de coût.
 - `Accuracy` pour la métrique d'évaluation.

6. Entraînez le modèle et évaluez ses performances. Comparez les performances avec le modèle précédent.

Amélioration des performances

7. Améliorez les performances du modèle en utilisant les techniques suivantes:
 - Augmentation de données.
 - Dropout : désactivation aléatoire de neurones.
 - Batch normalization : normalisation des données à chaque couche.

Reprise des TPs précédents

8. Reprendre les TPs précédents, y ajouter des réseaux de neurones et comparer les résultats.