Polytech Paris-Sud 5e année - Big Data

T. Sanchez - C. Barras

Projet de reconnaissance d'images - 2ème partie

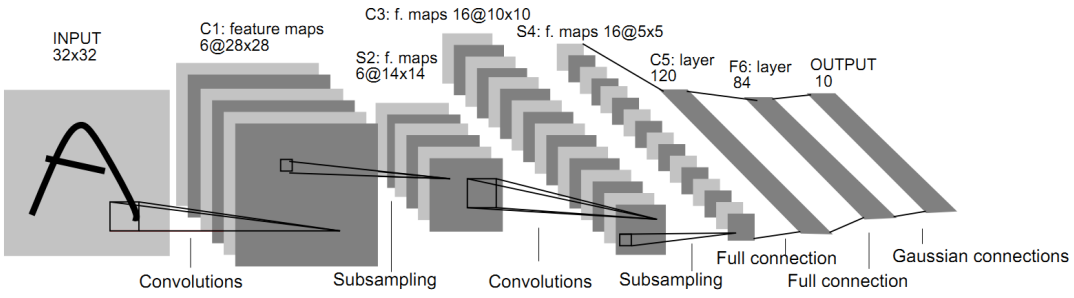
Dans cette deuxième partie, toujours en binôme, l'objectif est de comprendre le fonctionnement d'un programme d'apprentissage statistique profond (*Deep Learning*) et d'implémenter soi-même une architecture de réseau de neurone.

Introduction

Cette partie du projet se base sur le programme `dl_SVNN.py` qui implémente un réseau de neurone de type CNN (*Convolutional Neural Networks*). Les réseaux de neurones convolutionnels ont été inspirés par le fonctionnement du cortex visuel du chat et ils constituent aujourd'hui l'état de l'art pour la plupart des problèmes de reconnaissance d'images. Le code du programme est disponible dans l'espace 'Documents' du cours `Et5BigData` dans l'Environnement Numérique de Travail de l'Université. Pour le faire marcher, vous aurez besoin d'installer la bibliothèque `pytorch` avec `conda` (voir `pytorch.org`). Les données utilisées sont les mêmes que pour la première partie.

Travail à réaliser

1. À l'aide de la documentation de `pytorch` (voir `pytorch.org/docs`) et de vos recherches, répondez de manière détaillée aux questions suivantes :
   — Ligne 52, à quoi correspond une *epoch* ?
   — Qu'est ce qu'un *mini-batch* et pourquoi l'utilise-t-on ?
   — À quoi correspondent les méthodes `__init__` et `forward` de la classe `CNN` ?
   — À partir de la ligne 13, à quoi servent `nn.Conv2d` et `nn.Linear` ?
   — À partir de la ligne 19, à quoi servent `F.relu` et `F.max_pool2d` ?
   — Ligne 50, à quoi sert l'`optimizer optim.SGD` ?
   — Ligne 55, à quoi correspondent les prédictions faites par le réseau ?
   — Ligne 57, qu'est ce que la *loss* ?
   — Que fait la ligne 58, `loss.backward()` ?
2. Modifiez le code afin qu'il affiche l'évolution de la précision (nombre de bonnes prédictions / nombre de prédictions) sur les jeux de données d'entraînement et de validation. Pour cela vous devrez évaluer le réseau sur les deux jeux de données à intervalles réguliers au cours de l'entraînement, puis vous afficherez les courbes correspondantes.
3. Avec les hyperparamètres actuelles le réseau semble avoir des difficultés dans son apprentissage. Modifiez le bon paramètre pour avoir de meilleures prédictions. Quelle était le problème avec la valeur précédente du paramètre (pensez en terme de problème d'optimisation) ?
4. Comparez la précision de l'algorithme sur les données d'entraînement et de validation, que remarquez-vous, comment s'appelle ce phénomène ? Qu'elles peuvent être les causes de ce problème ? Corrigez le code.
5. En vous aidant du CNN déjà implémenté, vous mettrez en place l'un des CNNs historiques, le réseau LeNet. Ce réseau a l'architecture suivante :



— La première couche contient 6 filtres de convolution de taille 5×5 avec une fonction d'activation de type ReLU et un *max pooling* de dimension 2×2.
— La deuxième couche contient 16 filtres de convolution de taille 5×5 avec une fonction d'activation de type ReLU et un *max pooling* de dimension 2×2.
— La troisième couche *fully connected* contient 120 neurones avec une fonction d'activation de type ReLU.
— La quatrième couche *fully connected* contient 84 neurones avec une fonction d'activation de type ReLU.
— La dernière couche *fully connected* contient 10 neurones avec une fonction log-softmax.

6. Implémentez un réseau de type MLP (*MultiLayer Perceptron*). Ce type de réseau est plus simple qu'un réseau convolutionnel car il utilise seulement des couches *fully connected*. Vous choisirez le nombre de couche, de neurone par couche et le type d'activation. Attention, un réseau trop grand peut entraîner des temps de calculs très longs.
7. Comparer les trois architectures (CNN, MLP et LeNet) en donnant leur meilleur score sur les données de validation et le score correspondant sur le jeu d'entraînement. Vous donnerez aussi les temps d'exécution, la valeur des hyperparamètres choisies et le nombre de paramètres (poids et biais du réseau) pour les trois entraînement. Quelle sont les avantages des CNNs par rapport aux MLP ?

Une archive au format .zip d'un répertoire à votre nom contenant le programme `dl_SVNN.py` modifié et votre rapport avec les réponses aux questions est à déposer dans l'espace 'Travaux' du cours Et5BigData.