

4TYE903U: Intelligence Artificielle et Données

TD Data Mining Deep Learning - 1

Avant propos

Pour réaliser l'ensemble des exercices qui seront demandés, vous devez installer Pytorch. Cette librairie en Python est disponible à l'adresse suivante : <http://pytorch.org>.

- *PyTorch* est une bibliothèque Python open source d'apprentissage automatique initialement développée par Meta, et qui fait maintenant partie de la Fondation Linux.
- PyTorch permet d'effectuer les calculs tensoriels nécessaires à l'apprentissage profond (deep learning).
- C'est l'un des frameworks les plus populaires actuellement pour la recherche et le développement (existe aussi: Tensorflow = plus ancien, et JAX = plus récent).
- PyTorch permet, entre autres, de manipuler des tenseurs (tableaux multidimensionnels) et d'effectuer des calculs efficaces sur CPU ou GPU (par exemple, des produits de matrices ou des convolutions), et de calculer des gradients pour appliquer facilement des algorithmes d'optimisation par descente de gradient. <https://en.wikipedia.org/wiki/PyTorch>
- *TorchVision* est une bibliothèque annexe qui fait partie du projet Pytorch, et ajoute de nouvelles fonctionnalités utiles pour les tâches de vision par ordinateur: modèles pré-entraînés, pipeline d'augmentation de données, et méthodes de manipulation d'images.

1 Votre premier Réseau

Pour faciliter la prise en main de Pytorch, on se propose de réaliser l'exercice se trouvant sur la page :

- <https://medium.com/data-science/handwritten-digit-mnist-pytorch-977b5338e627>.
- Reproduire le code d'analyse comme suggéré dans l'analyse de MNIST.
- Donner les valeurs de `loss` et d'`accuracy` pour les jeux de données d'entraînement et de test.
- Faire varier les paramètres comme le `learning rate` ou le nombre d'`epochs` et observer l'influence de ces modifications - entre autre sur l'`overfitting`.
- Les données MNIST sont disponibles ici : <https://dept-info.labri.fr/~beurton/Enseignement/DataMining/DeepLearning/>

Bien sur, vous pouvez aussi les télécharger depuis le site de Yann LeCun comme indiqué dans la page web.

Est-ce que votre modèle est un CNN: Convolutional Neural Network ?

- Implémenter AlexNet: https://docs.pytorch.org/vision/main/_modules/torchvision/models/alexnet.html