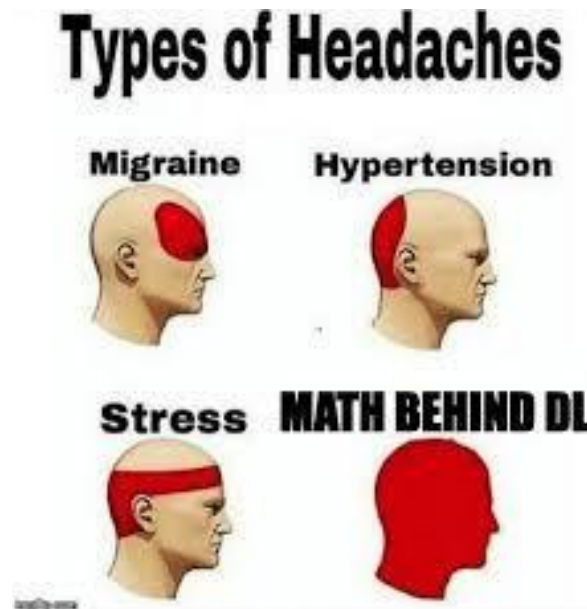


Introduction au Deep Learning

2 : Introduction aux réseaux de neurones convolutifs #1 (Jour 1)




Objectifs

- Introduction au CNN (Convolutional neural network) : architecture, couches, caractéristiques (convolutions), sortie
- Construction d'un modèle CNN simple (2-3 couches)
- Entraînement, évaluation et amélioration du modèle construit

Consignes

1. Nous allons commencer par un peu de lecture sur les réseaux de neurones convolutifs. Lisez attentivement la ressource R2.1.
 - Y a-t-il des concepts que vous n'avez pas compris ? Notez-les au tableau. Les concepts seront vu ensemble avec le groupe.
2. Après lecture de la ressource R2.1, dessinez une architecture type d'un CNN à minimum 2 couches pour une image en entrée d'une taille de 32x32. Sur chaque couche notez les dimensions des tableaux et leur nom.
3. Suivez le tutorial de la ressource R2.2 :
 - Nous allons utiliser la base d'images mnist. Visualisez quelques images de la base mnist
 - Implémentez le NN et le CNN donné dans l'exemple sur la base d'images mnist
 - Comparez le fonctionnement de ces modèles

- Quelle est la précision de tous ces exemples ?
4. Et si on rendait notre réseau de neurones un peu plus “deep” en rajoutant 1 couche convolutive (`Conv2D`) suivie d’une couche de pooling (`MaxPooling2D`), la précision du modèle est-elle la même ? Qu’en est-il si on rajoute une couche entièrement connectée ? Même question si on rajoute du `Dropout` (R2.4). Même question si on essaye d’augmenter de façon artificielle notre dataset (R2.5 et R2.6).
 5. Jouez sur les paramètres de votre choix pour améliorer l’estimation de ce CNN. Affichez les courbes de précision selon les itérations pour vos paramètres. Argumentez vos choix et si la prédiction peut être améliorée ou pas. En deep learning, les itérations sont souvent trop longues pour pouvoir faire des recherches d’hyperparamètres en formation donc je vous propose de regarder la Ressource R2.3 pour suivre une logique de définition d’un CNN. Une autre stratégie est de trouver de l’inspiration en étudiant des modèles connus tels que : LeNet, AlexNet, VGG-16, VGG-19 etc.
 6.
 *Trop facile ?! Essayez d’adapter votre CNN sur d’autres bases de données. Vous pouvez choisir parmi les milliers qui sont disponibles ici :*
<https://www.kaggle.com/datasets>

Ressources

- R2.1 <https://medium.com/@CharlesCrouspeyre/comment-les-r%C3%A9seaux-de-neurones-%C3%A0-convolution-fonctionnent-b288519dbcf8>
- R2.2 <https://becominghuman.ai/step-by-step-neural-network-tutorial-for-beginner-cc71a04eedeb>
- R2.3 <https://www.kaggle.com/cdeotte/how-to-choose-cnn-architecture-mnist>
- R2.4 <https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9>
(+ <https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>)
- R2.5 https://www.tensorflow.org/tutorials/images/data_augmentation?hl=fr
- R2.6 <https://www.tensorflow.org/tutorials/images/classification?hl=fr>
- Extra 1: <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#hyperparameters>
- Extra 2: <https://poloclub.github.io/cnn-explainer/>

Livrables

- ☐ L’implémentation du CNN avec keras