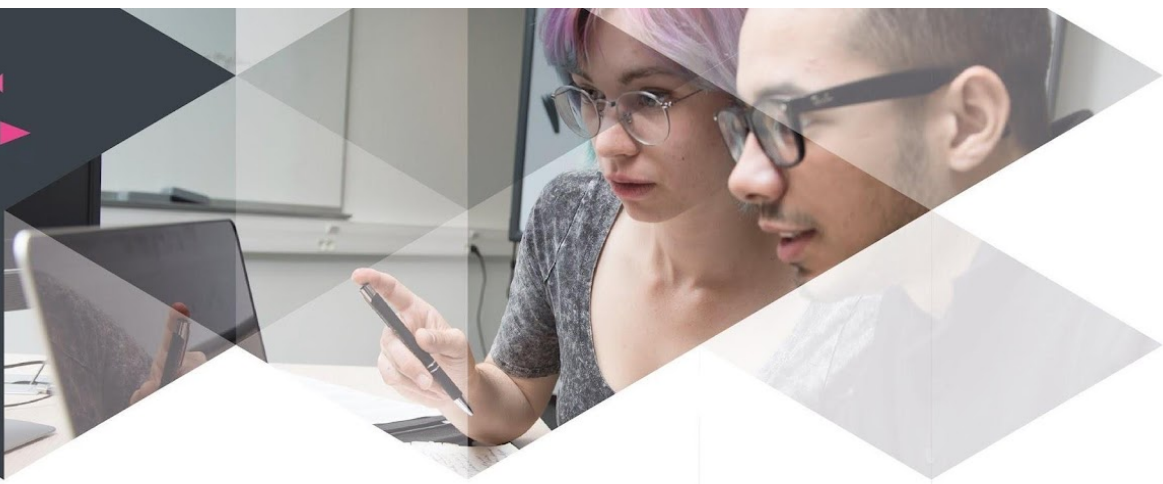




le
campus
numérique
in the ALPS



Deep Learning

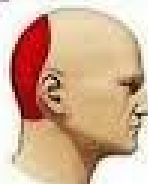
1. Introduction aux réseaux de neurones convolutifs #2 (Jour 1)

Types of Headaches

Migraine



Hypertension



Stress




MATH BEHIND DL



Objectifs:

1. Introduction CNN (Convolutional neural network) : architecture, couches, caractéristiques (convolutions), sortie
2. Construire un modèle CNN simple (2-3 couches)
3. Entraîner, évaluer et améliorer le modèle

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 vus ensemble avec le groupe.
2. Après lecture de la ressource R2.1 et le tour de table, 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). Quelle est la précision du modèle maintenant, est-elle la même ? Et si on rajoute une couche entièrement connectée. Si on rajoute du Dropout ? Et si on essaye d'augmenter de façon artificielle notre dataset avec ImageDataGenerator ? Est-ce que ça augmente la précision ?
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 de 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. <https://www.kaggle.com/cdeotte/how-to-choose-cnn-architecture-mnist>
 Trop easy ?! Essayez d'adapter votre CNN sur d'autres bases de données. Vous pouvez choisir parmi les milliers dispos 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>

Extra :

<https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#hyperparameters>

Livrables:

- ☐ L'implémentation du CNN avec keras