Artificial Intelligence for robotics with Python

Nennouche Mohamed

Séance 5

Sommaire

- Classification des images
  - Principe de base
  - Méthodes
  - Applications possibles
- Réseaux de neurones
- CNN
- Librairies qu'on va utiliser durant les projets
- Comment implémenter un CNN en Python
- Projet 1 classification des images
- Projet 2 classification des images

Qu'est ce que la classification des images ?

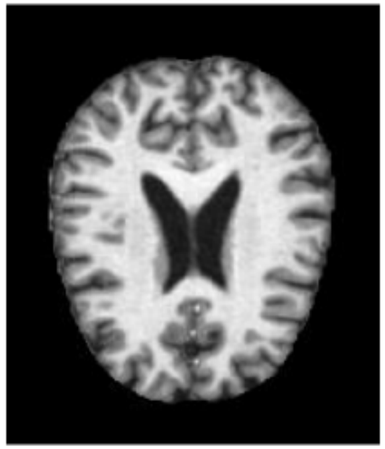
1. Suivant des méthodes classiques de computer vision en utilisant différents descripteurs statistiques.
2. En utilisant des méthodes de Deep Learning
    a. CNN
    b. ViT

Les deux méthodes peuvent fonctionner sur un bon nombre de sujets et de problèmes mais avec le temps les méthodes basées sur de l'apprentissage se sont avérées bien plus efficaces (à partir de 2015)

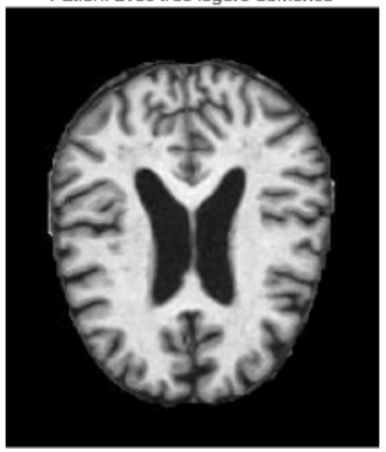
Quelques applications

Diagnostiques médicaux

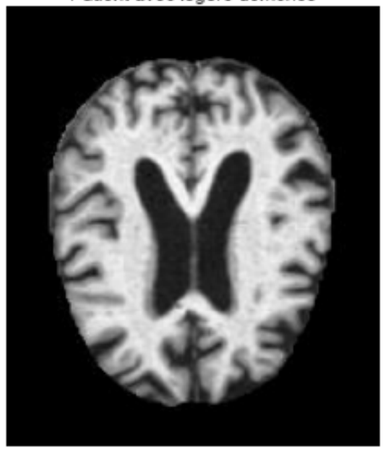
Patient sans démence



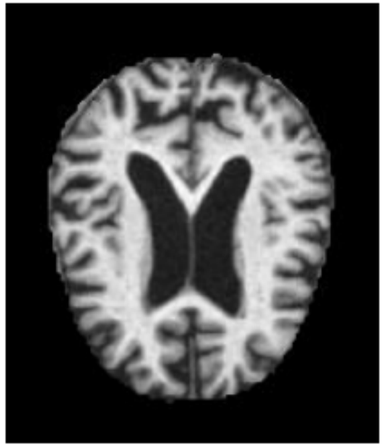
Patient avec très légère démence



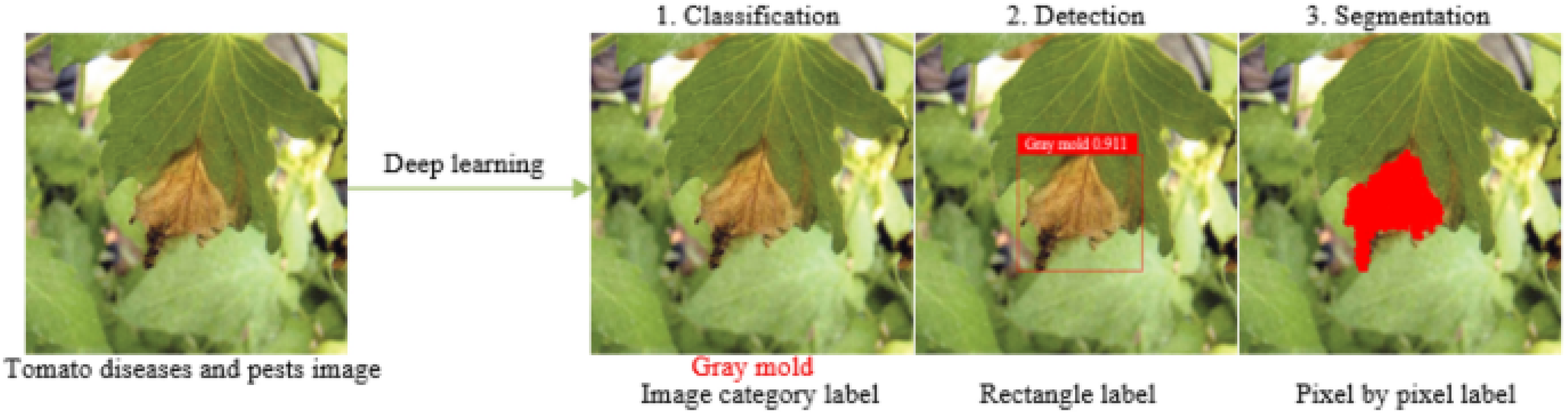
Patient avec légère démence



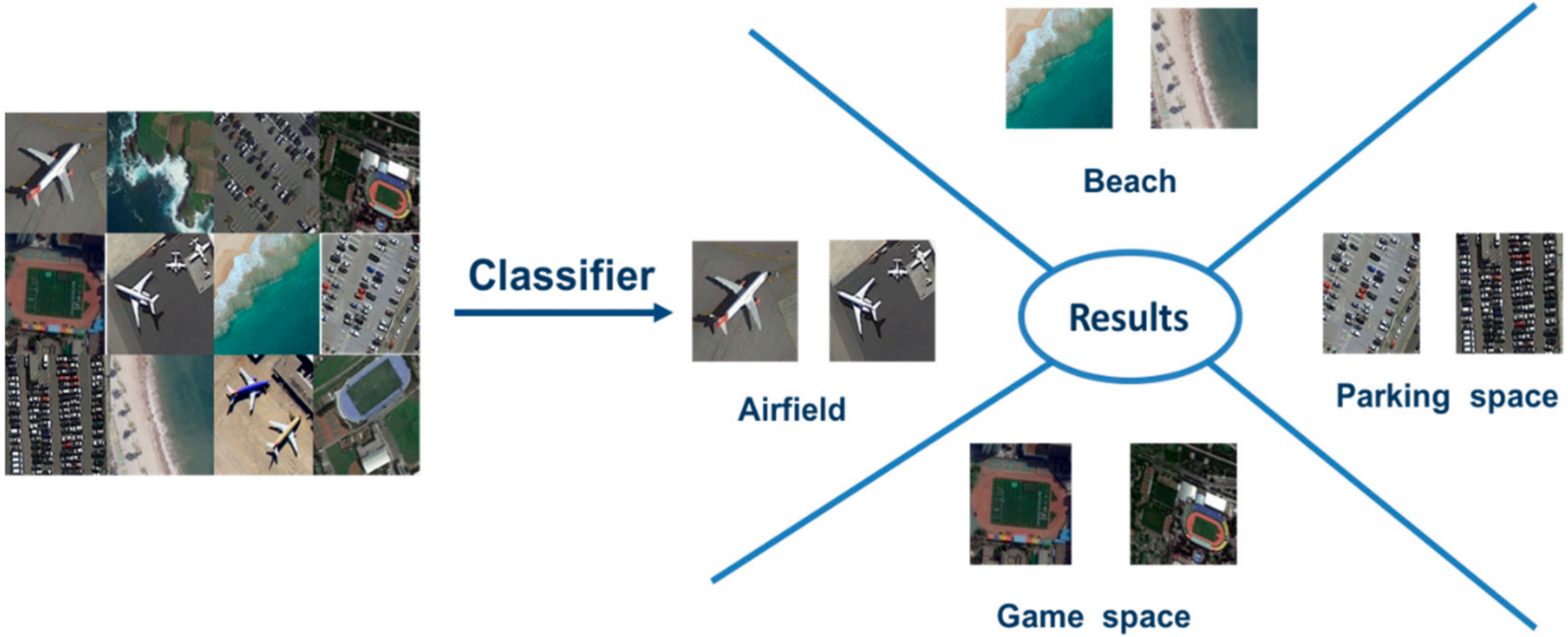
Patient avec une démence avancée



Détection de la maladie des plantes



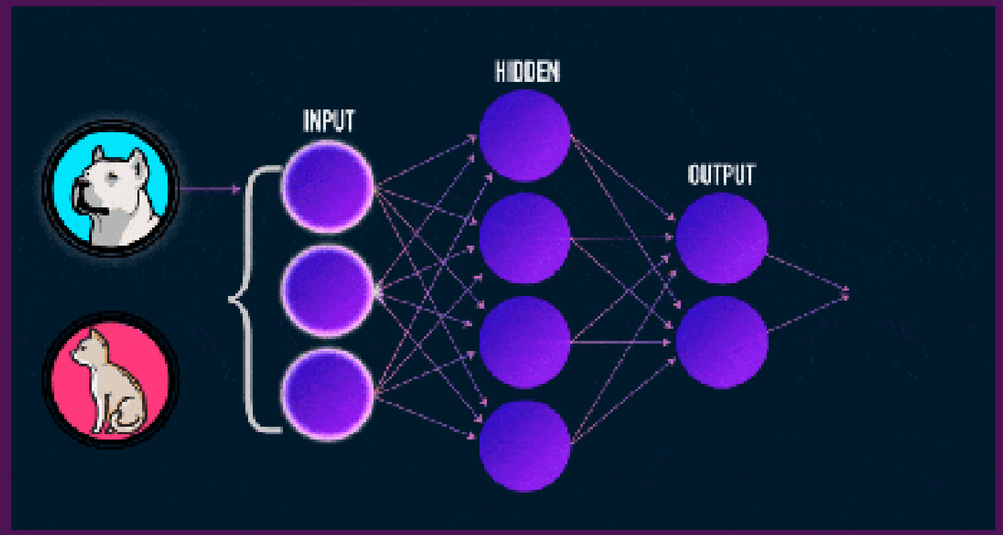
Classification des images satellitaires



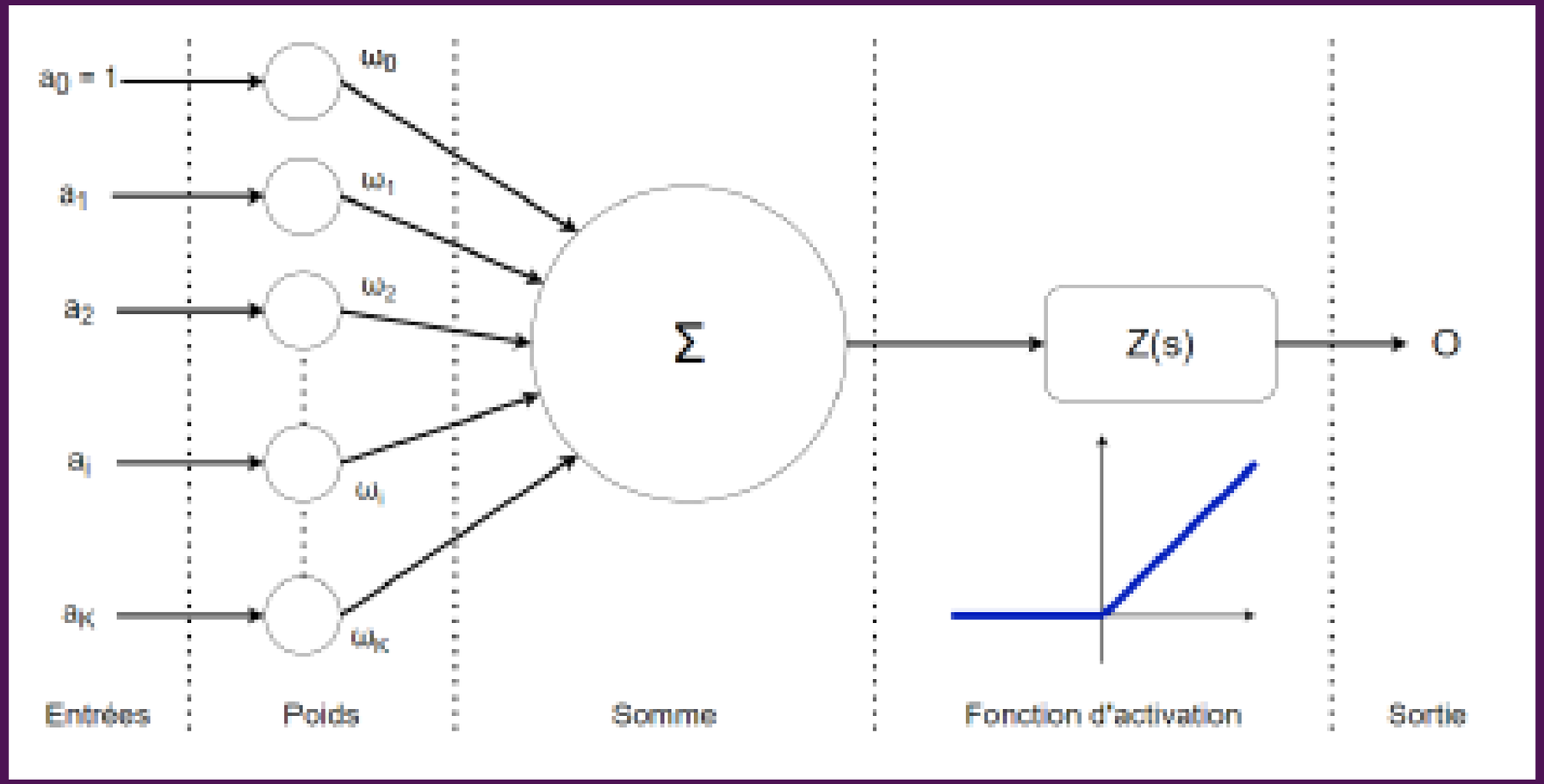
Les réseaux de neurones

Définition d'un réseau de neurone

Les réseaux de neurones (ou réseaux de perceptrons multicouches) est à la base du deep learning. Il est constitué comme son nom l'indique de plusieurs neurones (perceptrons) arrangés en couches, à travers une phase d'apprentissage (l'entraînement du modèle) le réseau de neurone peut faire un nombre incalculable de choses : classification, prédiction, segmentation, restauration...



Le perceptron



Principe d'un réseau multicouche

On empile les différents perceptrons et ensuite pour ajuster les poids synaptiques, on utilise la retropropagation du gradient :

- On calcule l'erreur quadratique entre la sortie et la prédiction (au moindre carrés)

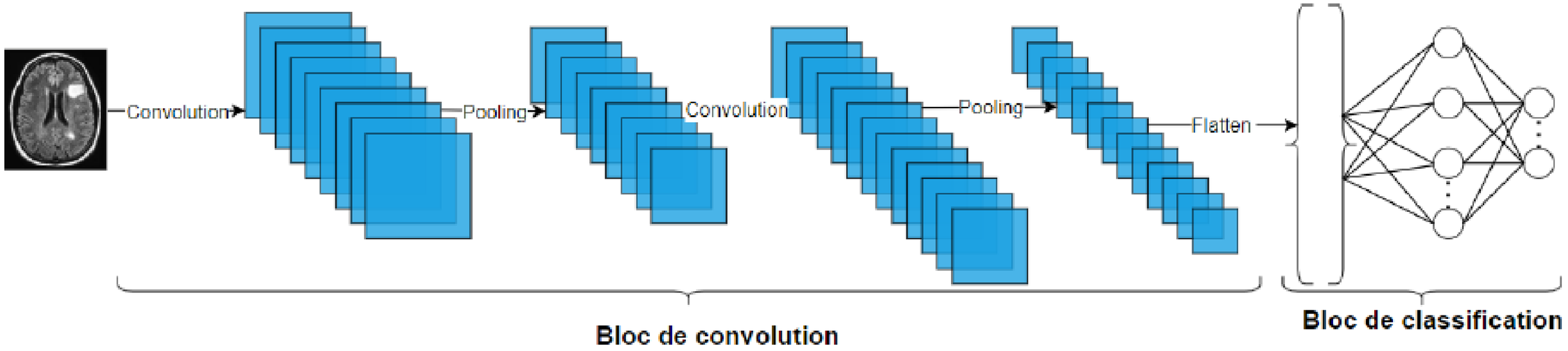
$$e_{\hat{\mu},\mu} = \frac{1}{2}\sum_{i=1}^{m}(\hat{\mu} - \mu)^2$$

- On ajuste le poids synaptique en fonction de l'erreur, du poids à l'itération précédente et en fonction des hyper-paramètres du réseau (taux d'apprentissage et momentum)

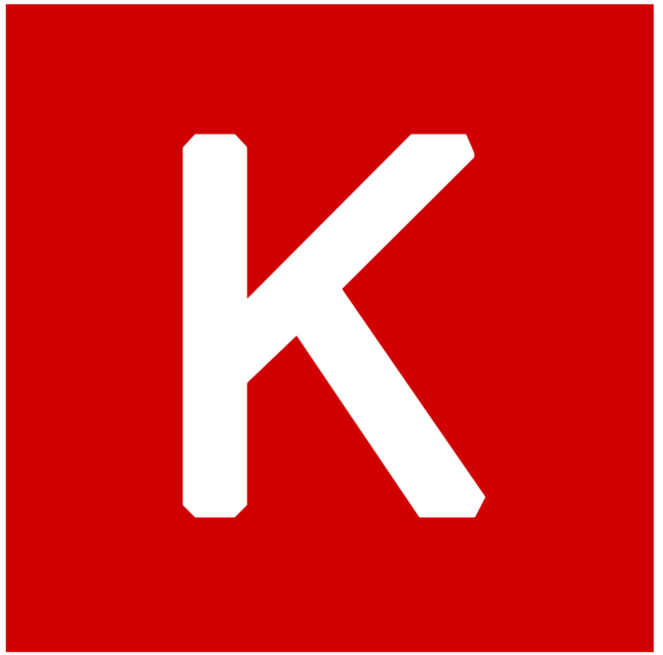
$$\omega(l+1) = \omega(l) + \Delta\omega(l+1) \qquad \text{Avec} \qquad \Delta\omega(l+1) = -\tau\frac{e}{\omega} + \rho\Delta\omega(l)$$

Les CNN

Principe d'un CNN



Deep Learning avec Python



Exemple d'utilisation de Tensorflow : MNIST

Projet 1 : GTSRB classification

Et maintenant ?

Merci