

Intelligence Artificielle : feuille 4

Réseaux de neurones (II) tensorflow et keras

Présentation

Dans ce TD, nous allons essentiellement utiliser `tensorflow`¹ et `keras`².

Le corpus que nous allons utiliser est celui présenté dans le cours. Il s'agit de `mnist`³.

Le but est de tester des architectures différentes (réseaux de neurones classiques, CNN), des hyper-paramètres différents (nombre de couches, taux d'apprentissage, etc) et de distinguer clairement les différentes étapes (training, validation, test).

Rappelons que le corpus `mnist` est composé de 70000 images de chiffres manuscrits. Les images sont de taille 28×28 pixels. Chaque pixel est codé par un nombre entre 0 et 255 correspondant au niveau de gris.

Commencez donc par télécharger le fichier `d1-td_2-sujet.ipynb` disponible à l'adresse du cours. Vous devez également activer l'environnement IA en exécutant l'instruction

```
source /net/ens/python/IA/bin/activate
```

Vous devez ensuite lancer `jupyter notebook` depuis le terminal de l'environnement virtuel que vous venez d'activer.

Quelques rappels.

Rappelons qu'un réseau de neurones est composé d'une suite de couches successives. Chaque couche est un ensemble de neurones formels. Les architectures de base considèrent que les signaux ne circulent que d'une couche vers la suivante.

Chaque neurone dans la $k^{\text{ème}}$ couche est connecté à tous les neurones de la $(k - 1)^{\text{ème}}$ couche et les neurones d'une couche forment un ensemble indépendant.

Un réseau de neurones est composé de :

- une couche d'entrée x
- un (éventuel) nombre arbitraire de couches cachées
- une couche de sortie y
- un ensemble de poids W et de biais b
- un ensemble de fonctions d'activation, une par couche.

1. <https://www.tensorflow.org>

2. <https://keras.io>

3. <http://yann.lecun.com/exdb/mnist/>

Exercice 1. Un réseau pour reconnaître tous les chiffres

Dans cet exercice, nous allons définir un réseau de neurones capable de reconnaître tous les 10 chiffres (corpus `mnist`). Il s'agit d'un réseau à une seule couche.

1. Expliquer, avec une figure, l'architecture du réseau.
2. Définir, dans le programme, l'architecture de votre réseau.
3. Entraîner et tester le réseau. Changer d'algorithme d'optimisation (voir `optimizers` sur <http://keras.io>).
4. Ajouter une couche cachée avec 10 neurones, puis avec 64 neurones.

Exercice 2. Un réseau CNN pour reconnaître tous les chiffres

Même exercice mais en utilisant une architecture en CNN.

1. Définir un réseau de neurones à convolution avec :
 - une couche de convolution avec 32 filters, chacun de taille 3×3
 - une couche de maxpooling (pool de taille 2×2).
 - un réseau "fully connected" à plusieurs couches
2. Entraîner le modèle et le tester. Comparez l'accuracy obtenue avec celles des modèles précédents.
3. Choisir les bons hyper-paramètres pour atteindre une précision d'au moins 99%.