Apprentissage statistique
TP5 : Réseaux de neurones

Olivier Schwander <olivier.schwander@lip6.fr>

2018-2019

Préliminaire pour la salle 401

Attention : ça ne fonctionne pas avec des comptes invités.

Dans un terminal, exécuter la commande suivante :

```
echo "export KERAS_BACKEND=tensorflow" >>~/.bashrc
```

Attention : relancer impérativement le terminal !

Ensuite, à chaque séance, il faudra taper la commande :

```
source /users/home/schwander/apprentissage/bin/activate
```

On pourra ensuite lancer normalement `python3` et `spyder3` (spyder*3* uniquement) ou `jupyter` à partir de ce terminal.

Exercice 1 *Prise en main de Keras*

La plateforme Keras est une surcouche de haut niveau pour la plateforme de deep learning Tensorflow, qui permet de construire très facilement des modèles à base de réseaux de neurones et de réaliser l'apprentissage. De nombreux outils supplémentaires facilitent la réalisation d'expériences.

On peut vérifier que tout fonctionne bien en téléchargeant l'exemple suivant :

https://raw.githubusercontent.com/fchollet/keras/master/examples/mnist_cnn.py

et en l'éxécutant avec la commande :

```
/usr/bin/python3 mnist_cnn.py
```

La documentation, à laquelle il faudra se référer régulièrement, est disponible à l'adresse https://keras.io/.

Question 1

Laisser tourner l'exemple quelques minutes. Que représentent les informations affichées ?

Exercice 2 *Réseaux denses*

Question 1 *Perceptron*

Construisez un perceptron à plusieurs sorties (*Dense* dans le vocabulaire Keras).

Quelle fonction calcule ce perceptron ?

Question 2 ***Couches cachées***

Empilez plusieurs couches denses.

Quelle fonction calcule ce réseau ?

Question 3 ***Non-linéarités***

Pour rendre intéressantes les couches cachées, on a besoin d'introduire des non-linéarités dans l'empilement. Les plus courantes sont :

— la tangente hyperbolique,
— la sigmoïde $S(x) = \frac{1}{1+e^{-x}}$,
— le rectifieur $f(x) = \max(0, x)$ (en anglais, *Rectified Linear Unit, ReLU*), c'est cette fonction qui est la plus utilisée en pratique.

Rajoutez des non-linéarités.

Question 4 ***Soft-max***

Pour réaliser un classifier multi-classes à partir d'un réseau à K sorties $z_1, \dots, z_K$, on utilise en général la fonction *soft-max* :

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Quel est l'intérêt de cette fonction par rapport un simple maximum ?

Question 5

Expérimentez et comparez différent modèles à couches cachées pour les données USPS.

On utilisera comme pénalité l'entropie croisée (disponible en utilisant `categorical_crossentropy` comme paramètre `loss`) :

$$L = \sum_{i=1}^{K} \sum_{j=1}^{N} t_j^{(i)} \log z_j^{(i)}$$

où $t_j = (0, \dots, 0, \underbrace{1}_{k}, 0, \dots, 0)$ si l'observation j appartient à la classe k.

On pourra utiliser également la base d'image MNIST (accessible directemennt dans Keras, cf https://keras.io/datasets/#mnist-database-of-handwritten-digits).

Question 6

Utilisez l'historique renvoyé par la méthode `fit` pour tracer les courbes d'apprentissage (fonction de coût, précision en fonction des itérations).

Exercice 3 *Réseaux convolutionnels*

Question 1

Quel est le nombre de paramètres d'un réseau dense à plusieurs couches cachées ?

Question 2

Quel est le nombre de paramètres pour un réseau convolutionnel avec le même nombre de couches ? Commentez.

Question 3

Construisez un réseau convolutionnel pour la classification MNIST.