



Réseaux profonds : ouvrir la boîte noire !

Contexte :

Avec de plus en plus de données disponibles, des améliorations récentes apportées par l'apprentissage profond (*deep learning*), les performances obtenues par les nouveaux systèmes d'apprentissage automatique pour la classification d'image, l'analyse des sentiments, la compréhension de la parole etc. ... sont véritablement impressionnantes. Les bibliothèques très efficaces comme Keras, TensorFlow etc. permettent en seulement quelques lignes de code de créer un réseau complexe composé de structures non linéaire imbriquées. Cependant, à cause de ces structures, ces modèles d'apprentissage automatique s'appliquent à la manière d'une boîte noire : aucune information n'est fournie sur ce qui les a conduits à atteindre leurs prédictions. Il est cependant possible d'ouvrir ces boîtes noires pour voir ce qu'elles font et ainsi essayer de mieux comprendre leur fonctionnement.

Dans le cadre de ce projet de TER, nous nous situons dans de la classification d'images à l'aide d'un réseau profond. L'objectif est dans ce cadre d'offrir une application web qui offrent différentes fonctionnalités pour « ouvrir ce réseau » et visualiser ce qui se passe réellement à l'intérieur et ainsi de pouvoir répondre à des questions du type : comment est effectué le processus d'apprentissage ? quelle est la partie de l'image qui a permis de classer ? à quel moment ou quel niveau la classification se fait-elle réellement ?

Ce projet de TER s'adresse particulièrement à des personnes intéressées par les sciences des données et qui veulent vraiment comprendre ce qu'il y a derrière les réseaux profonds.

Travail à réaliser :

Les différentes étapes pour mener à bien ce TER sont les suivantes :

1. Compréhension des réseaux de neurones [1] [2].
2. Apprentissage de Keras [3] et expérimentation sur plusieurs jeux de données pour savoir mettre en place différentes architectures.
3. Développement en Python et expérimentation d'une application qui soit capable d'afficher dans un premier temps comment se classent les images lors de l'apprentissage [4].
4. Extension du programme précédent pour afficher le contenu des différents niveaux du réseau [5].
5. Spécification et réalisation d'une interface de visualisation pour pouvoir faire varier les paramètres, effectuer des expérimentations et visualiser les résultats. Il est tout à fait possible de s'inspirer de l'application `convnet.playground` [6].

Les parties 2, 3 et 4 pourront être réalisées sur Colaboratory (Colab) de Google [7].

Remarque : une explication détaillée des réseaux de neurones sera faite au début du TER [1,2]. Il n'est donc pas nécessaire d'avoir des connaissances préalables sur ce point. L'important est de bien comprendre le fonctionnement et de pouvoir mettre en place l'architecture pour que l'interface web puisse communiquer avec le programme python réalisé.

Prérequis :

- Langage de programmation (Python)
- Programmation Web (Javascript)
- Curiosité



Nombre d'étudiants : 3 à 5

Encadrant : Pascal Poncelet (contact : Pascal.Poncelet@lirmm.fr)

Références :

- [1] Pascal Poncelet. "Notebook descente de gradient", 2020.
- [2] Pascal Poncelet. "Notebook réseaux de neurones", 2020.
- [3] Keras. <https://keras.io> (dernier accès le 15 octobre 2020).
- [4] Michelle Peters et al. "Visualising the Training Process of Convolutional Neural Networks for Non-Experts". <http://ceur-ws.org/Vol-2491/paper108.pdf> - 2019.
- [5] Arnav Chakravarthy. "Visualizing Intermediate Activations of a CNN trained on the MNIST Dataset". <https://towardsdatascience.com/visualizing-intermediate-activations-of-a-cnn-trained-on-the-mnist-dataset-2c34426416c8> . (dernier accès le 15 octobre 2020).
- [6] Convnet. <https://convnetplayground.fastforwardlabs.com>. (dernier accès le 15 octobre 2020).
- [7] Colab. <https://colab.research.google.com/notebooks/intro.ipynb>. (dernier accès le 15 octobre 2020).