Chemins spécifiques pour la classification dans les réseaux de neurones profonds

Contexte :

Avec de plus en plus de donnés disponibles, des améliorations récentes apportées par le 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 librairies 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 prédications.

Souvent, de nombreuses critiques ont été émises pour des algorithmes d'apprentissage comme SVM ou les réseaux de neurones qui justement se comportent comme des boîtes noires [3]. Avec le développement du deep learning, de plus en plus de communautés veulent ouvrir ces "modèles boîtes noires". Les raisons sont multiples. Par exemple, quelle confiance accorder réellement à la prédiction du modèle ? Récemment des auteurs ont montré qu'un système entraîné à prédire le risque de pneumonies arrivait à des conclusions totalement fausses. Le modèle avait appris que les patients asthmatiques souffrant de problèmes cardiaques couraient un risque beaucoup plus faible de mourir de pneumonie que les personnes en bonne santé.

C'est dans ce cadre que ce situe ce TER. L'objectif est ici de mieux comprendre comment s'exécute un modèle. Il s'agit de repérer des signatures d'activation en fonction des données d'entrée pour répondre aux questions du type :

- Si le jeu d'apprentissage ne contient que des 1 et des 3 quels sont les neurones qui sont activés et comment ? que se passe-t-il si le modèle est appliqué sur un 2 ?
- Existe-t-il des signatures caractéristiques de certaines données ?
- A partir de quand (quelle couche ?) le modèle change de comportement pour reconnaître une valeur ?

Il s'adresse à des personnes intéressées par les sciences des données et qui veulent vraiment comprendre ce qu'il y a derrière les réseaux de neurones.

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 sur divers jeux de données d'une application qui soit capable d'extraire les différentes fonctions d'activations d'un réseau
4. Extension du programme précédent pour extraire, pour chaque layer, une signature. Pour cela un algorithme d'apprentissage non supervisé sera utilisé [4].
5. Spécification et réalisation d'une interface de visualisation pour pouvoir faire varier les paramètres, effectuer des expérimentations et visualiser les résultats

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 un programme python.

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] W. Samek, T. Wiegand, K-R.t Müller. "Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models". arXiv:1708.08296, 2017.
[4] Keras. https://keras.io (dernier accès le 15 octobre 2020).
[5] https://www.machinecurve.com/index.php/2020/04/16/how-to-perform-k-means-clustering-with-python-in-scikit/. (dernier accès le 15 octobre 2020).
[6] Colab. https://colab.research.google.com/notebooks/intro.ipynb. (dernier accès le 15 octobre 2020).