

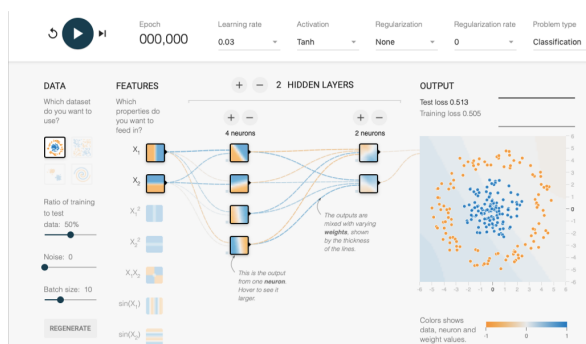


Deep learning : un simulateur pour comprendre

Contexte :

Aujourd'hui le deep learning est partout ! Il permet, par exemple, de reconnaître automatiquement que l'image contient un chat, que le texte exprime une opinion négative et les résultats obtenus sont tels qu'il est capable de battre l'humain dans certaines circonstances.

De nombreuses bibliothèques existent aujourd'hui et créer un réseau de neurones profond pour faire de la classification d'images ne prend maintenant que quelques lignes de codes. C'est dans ce contexte que s'inscrit ce TER : qu'est-ce qu'il y a derrière ces lignes de codes ? Comment est-ce que ce réseau apprend ? quelles sont les différentes étapes ? pour répondre à ces questions l'objectif de ce TER est de réaliser une application de simulation d'un réseau de neurones.



Un exemple de simulateur (<https://playground.tensorflow.org>)

Il s'adresse à des personnes intéressées par les sciences des données et plus particulièrement au fonctionnement des réseaux profonds mais nécessite un intérêt fort pour la visualisation.

Travail à réaliser :

L'objectif est de réaliser une application web permettant de simuler le comportement d'un réseau de neurones. Cette application doit permettre de saisir les différents paramètres (nombres de neurones, de couches, valeur du learning rate, ...), de pouvoir interagir avec une application python qui exécutera le réseau et fournira les différentes valeurs à afficher.

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 chargé d'exécuter le réseau (le code pour le réseau sera fourni).

Prérequis :

- langage de programmation (Python), javascript, HTML, CSS, D3.JS

Nombre d'étudiants : 3 à 5

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

Références :

- [1] "Notebook descente de gradient".
- [2] "Notebook réseaux de neurones", 2019.
- [3] Treecracker pour Python : <https://d3js.org>