

Réseaux de neurones (Notes de cours)

Romain Tavenard

Chapitre 1

Préambule

Ce document est un ensemble de notes associées au module de classification supervisée pour la deuxième année de master de Statistiques de l'Université de Rennes 2, et plus particulièrement la partie sur les réseaux de neurones. Il est distribué librement (sous licence [CC BY-NC-SA](#) plus précisément) et se veut évolutif, n'hésitez donc pas à faire vos remarques à son auteur dont vous trouverez le contact sur [sa page web](#).

1.1 Contenu du cours

Chapitre 2

Régression logistique et perceptron

2.1 Retour sur la régression logistique

2.2 Le perceptron : notations et représentation

2.2.1 Poids

2.2.2 Fonction d'activation

Chapitre 3

Perceptron multi-couches

<http://cs231n.github.io/neural-networks-1/>

3.1 Cas d'un réseau de neurones à une couche cachée

3.1.1 Calcul de gradient

3.1.2 Visualisation de frontières apprises

Exemple intéressant : <http://cs231n.github.io/neural-networks-case-study/>

3.2 Cas multi-couches

3.2.1 Calcul de gradient

<http://cs231n.github.io/optimization-2/>

3.2.2 Visualisation

Comprendre qu'on apprend une représentation (eg PCA améliorée) suivie d'un classifieur linéaire (logistic regression)

- <https://rajarsheem.wordpress.com/2017/05/04/neural-networks-dynamics/>
- <http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>

3.2.3 Exemples adversaires

3.3 Quelques considérations pratiques

3.3.1 Initialisation des poids du réseau

3.3.2 Régularisation

<http://cs231n.github.io/neural-networks-2/>

3.3.2.1 Régularisations L1/L2

3.3.2.2 Utilisation du *dropout*

Chapitre 4

Réseaux de neurones convolutionnels

4.1 Motivation

4.2 Mise en oeuvre

<http://cs231n.github.io/convolutional-networks/>

4.3 Apprentissage par transfert et *fine-tuning*

<http://cs231n.github.io/transfer-learning/>

Chapitre 5

Mise en oeuvre avec TensorFlow

Éventuellement en les faisant utiliser https://github.com/rtavenar/tf_standardmodels. Mais alors, ce n'est plus vraiment une mise en oeuvre avec TensorFlow car les modèles sont codés pour eux.

Sinon, pour un truc plus générique : <https://keras.io>.