

## TP 2 : Introduction aux réseaux de neurones

Jérôme Pasquet

### Exercice 1 : Implémentation d'un réseau de neurones

Nous nous proposons d'implémenter une classe `Network` permettant de créer un réseau de neurones. Cette classe possèdera deux classes filles `NetworkMat` et `NetworkInd` qui vont effectuer les mêmes opérations mais en utilisant respectivement que des opérations matricielles et des boucles pour parcourir les vecteurs.

La classe `Network` possède les méthodes suivantes :

- ***add\_layer*** permettant de rajouter une couche. Nous considérerons un ordre séquentiel allant de l'entrée du réseau à sa sortie.
- ***feed\_forward*** activant la totalité des couches.
- ***backpropagation*** activant le réseau puis effectuant une mise à jour des poids.
- ***train*** lançant un apprentissage et effectuant un test à chaque époque.
- ***test*** effectuant un test sur une base de validation transmise.

1. Comparez les temps d'inférence et de rétropropagation de vos deux modèles.
2. Sur les jeux de données du TP effectuer un apprentissage avec une couche cachée contenant 2, 4, 8 et 16 neurones. Le théorème d'approximation universelle est-il respecté ?
3. Rajoutez une couche cachée, comment évolue les résultats. Conclure.

### Exercice 2 : Le théorème d'approximation universelle

Réutilisez l'architecture de l'exercice 1.

Nous allons chercher à vérifier l'exactitude du théorème 1 sur un problème de régression.

- Adaptez votre méthode d'évaluation pour calculer une régression
- Pour chacune des fonctions ci-contre donnez le nombre de neurones nécessaire tel que :  
$$\frac{\sum_i^N ||y_i - \sigma_i||_1}{N} < 0.5$$
 sur la base de validation.

$$g(x) = 0.5x^2 + \frac{1}{5}x^3 - x$$

$$h(x) = 0.5x^2 + \frac{1}{5}x^3 - x + 2\sin(x^2) - 3\cos(x) + 2.\sin\left(\frac{3x}{2}\right)$$

$$f(x) = 0.5x^2 + \frac{1}{5}x^3 - \frac{1}{x}$$

Note: nous considérerons :

- la base d'apprentissage suivante :  $train_x = np.arange(-5.05, 5.05, 0.1)$
- la base de test suivante :  $test_x = np.arange(-5, 5, 0.2) \cup np.arange(-0.05, 0.05, 0.01)$

Que constatez-vous ?

## Devoir maison I

Soit un problème radial. Nous allons étudier les performances d'un réseau à deux couches de 16 neurones chacune.

- Implémenter la fonction de perte de la cross entropy softmax.

Dans les deux scenarios suivants lancer un apprentissage et exposez les résultats de manière visuelle :

- Sans recouvrement entre les deux classes
- Avec recouvrement entre les deux classes

Lisez et comprenez l'article suivant : Heated-Up Softmax Embedding - Xu Zhang et al.

Re-effectuez l'expérience suivante en y incorporant la température  $T=\{0.1, 10.0\}$ .

## Devoir maison II

Lisez avec un regard critique le papier : Activation Adaptation in Neural Networks - Farnoush Farhadi, Vahid Partovi Nia et Andrea Lodi