

TRAVAUX PRATIQUE  
INTRODUCTION AU DEEP LEARNING

**Théo Lopès-Quintas**

BPCE Payment Services,  
Université Paris Dauphine

Janvier 2024

# SÉANCE 1 - RÉSEAUX DENSES

- 1 Séance 1 - Réseaux denses . . . . . 1**
  - 1.1 Comparaison d’hyper paramètres . . . . . 3
  - 1.2 Il y a des hasards meilleurs que d’autres . . . . . 4
  - 1.3 Une première méthode de régularisation . . . . . 5
  - 1.4 Régression . . . . . 6
- 2 Séance 2 - Réseaux convolutionnel . . . . . 7
- 3 Séance 3 - Compléments . . . . . 13
- 4 Séance 4 - Réseaux récurrents . . . . . 19

# SÉANCE 1 - RÉSEAUX DENSES

## CONSIGNES

Quelques consignes et conseils pour l'ensemble des sujets proposés :

- ▶ Pour tous les réseaux de neurones que vous définirez, essayez de calculer à la main le nombre de paramètres de votre réseau de neurones puis contrôler avec la méthode *summary*.
- ▶ Ne négligez pas la qualité de code ! Le travail avec les réseaux de neurones nécessite de rédiger plus de code que les algorithmes de Machine Learning classique, cela peut rendre le notebook plus difficile à lire et à retravailler.
- ▶ Vous pouvez choisir votre propre sujet à condition de le faire valider en amont

Sauf mention contraire, on utilisera le dataset MNIST pour entraîner les réseaux de neurones denses.

# SÉANCE 1 - RÉSEAUX DENSES

## A - HYPER PARAMÈTRES

Nous souhaitons mesurer l'impact du choix des hyper paramètres sur les performances d'un réseau de neurones dense. Pour cela, choisir un ou deux hyper paramètres à tester parmi :

- ▶ **Learning rate** : comment se comporte l'entraînement du réseau de neurones selon différentes valeurs du learning rate ?
- ▶ **Profondeur** : comment se comporte l'entraînement du réseau de neurones selon différentes profondeur du réseau de neurones ?
- ▶ **Époques** : comment se comporte l'entraînement du réseau de neurones sur des temps d'entraînement long ?

On peut trouver d'autres questions, il faut les faire valider pour pouvoir les traiter.

## Consignes

- ▶ Annoncer avant de comparer ce à quoi on s'attend
- ▶ Produire des résultats simplement interprétable et lisible
- ▶ Identifier et présenter les limites de la comparaison réalisée

# SÉANCE 1 - RÉSEAUX DENSES

## B - IL Y A DES HASARDS MEILLEURS QUE D'AUTRES

Nous souhaitons mesurer l'impact de la distribution des valeurs initiales des poids sur l'entraînement du réseau de neurones dense. Pour cela, on se propose de comparer plusieurs distribution :

- ▶ **Glorot** : Normal et uniforme
- ▶ **He** : Normal et uniforme
- ▶ **Random** : Normal et uniforme

Pour être complet, on réalisera l'étude pour la fonction d'activation ReLU et, au choix, sigmoid ou tanh.

### Consignes

- ▶ Annoncer avant de comparer ce à quoi on s'attend
- ▶ Produire des résultats simplement interprétable et lisible
- ▶ Identifier et présenter les limites de la comparaison réalisée

# SÉANCE 1 - RÉSEAUX DENSES

## C - UNE PREMIÈRE MÉTHODE DE RÉGULARISATION

Les réseaux de neurones sont particulièrement sensible au sur-apprentissage à cause de la très forte flexibilité offerte par les nombreux paramètres. Une première manière de régulariser le réseau de neurones est d'exploiter une régularisation  $\mathcal{L}_2$  comme pour une régression linéaire. Cela consiste à modifier la fonction de perte :

$$\mathcal{L}_\lambda(w) = \mathcal{L}(w) + \frac{\lambda}{2} \|w\|_2^2 \quad \text{avec } \lambda \geq 0$$

Pour l'appliquer il faut, couche par couche, choisir la valeur de  $\lambda$  que l'on veut appliquer. On souhaite mesurer l'impact de la régularisation sur l'apprentissage d'un réseau de neurones : proposer une étude répondant au problème.

### Consignes

- ▶ Annoncer avant de comparer ce à quoi on s'attend
- ▶ Produire des résultats simplement interprétable et lisible
- ▶ Identifier et présenter les limites de la comparaison réalisée

# SÉANCE 1 - RÉSEAUX DENSES

## D - RÉGRESSION

Nous souhaitons entraîner un réseau de neurones pour résoudre un problème de régression. Pour cela, nous considérerons le dataset `sklearn.datasets.fetch_california_housing`. Après avoir prit connaissance du problème que l'on cherche à résoudre, et réalisé un rapide modèle Machine Learning classique, on entraînera un réseau de neurones. A noter qu'il faudra modifier :

- ▶ **Couche input** : l'input n'a plus besoin d'être aplati, donc il faudra renseigner la dimension de l'input dans la première couche
- ▶ **Couche output** : l'output n'a pas 10 classes, donc il faudra ne placer qu'un seul neurone en sortie
- ▶ **Loss** : la fonction de perte doit être une fonction de perte de régression, par exemple la MSE

## Consignes

- ▶ Réaliser une étude rapide et claire du dataset utilisé
- ▶ Entraîner avec des performances correctes un algorithme de Machine Learning classique
- ▶ Entraîner avec des performances correctes le réseau défini