

TRAVAUX PRATIQUE
INTRODUCTION AU DEEP LEARNING

Théo Lopès-Quintas

BPCE Payment Services,
Université Paris Dauphine

Janvier 2024

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
2.1	Comparaison d'hyper paramètres	9
2.2	Une deuxième régularisation	10
2.3	ResNet à la main	11
2.4	Sharpened cosine	12
3	Séance 3 - Compléments	13
3.1	Comparaison d'optimizer	15
3.2	Échéancier pour le <i>learning rate</i>	16
3.3	Batch Normalization et SeLU	17
3.4	Dropout et Data Augmentation	18
4	Séance 4 - Réseaux récurrents	19

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 λ 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

SÉANCE 2 - RÉSEAUX CONVOLUTIONNEL

- 1 Séance 1 - Réseaux denses 1
- 2 **Séance 2 - Réseaux convolutionnel 7**
 - 2.1 Comparaison d'hyper paramètres 9
 - 2.2 Une deuxième régularisation 10
 - 2.3 ResNet à la main 11
 - 2.4 Sharpened cosine 12
- 3 Séance 3 - Compléments 13
- 4 Séance 4 - Réseaux récurrents 19

SÉANCE 2 - RÉSEAUX CONVOLUTIONNEL

CONSIGNES

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

- ▶ Pour l'ensemble des 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 Fashion MNIST pour entraîner les réseaux de neurones denses.

SÉANCE 2 - RÉSEAUX CONVOLUTIONNEL

A - HYPER PARAMÈTRES

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

- ▶ **Taille de convolution** : comment se comporte l'entraînement du réseau de neurones selon différentes tailles de filtre de convolution ?
- ▶ **Fonction d'activation** : comment se comporte l'entraînement du réseau de neurones selon différentes fonction d'activation ?
- ▶ **Pooling** : comment se comporte l'entraînement du réseau de neurones selon différentes type de pooling ?

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 2 - RÉSEAUX CONVOLUTIONNEL

B - BATCHNORMALIZATION

L'article *Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift* écrit par Sergey Ioffe et Christian Szegedy introduit une nouvelle régularisation pour un réseau de neurones. On se propose dans ce sujet de l'étudier pour mieux comprendre son fonctionnement. On pourra faire une liste de l'ensemble des *shift* existant en Machine Learning, présenter des exemples et expliquer comment on peut s'en prémunir, si c'est possible.

Consignes

- ▶ Lire l'article et expliquer le fonctionnement de cette couche
- ▶ Produire des résultats simplement interprétable et lisible
- ▶ Identifier et présenter les limites de la comparaison réalisée

SÉANCE 2 - RÉSEAUX CONVOLUTIONNEL

C - RESTNET À LA MAIN

On souhaiterait pouvoir définir nous-mêmes un réseau de neurones convolutionnel résiduel. Pour cela, on s'appuiera sur l'approche fonctionnel de Keras pour définir soi-même un réseau de neurones résiduels.

Egalement, on utilisera la couche de BatchNormalization après l'avoir étudié d'après le papier présenté au sujet précédent. On expliquera rapidement le fonctionnement.

Consignes

- ▶ Ne pas négliger la qualité de code
- ▶ Essayer de généraliser le plus possible la création d'un ResNet
- ▶ Entraîner avec des performances correctes le réseau défini

SÉANCE 2 - RÉSEAUX CONVOLUTIONNEL

D - SHARPENED COSINE SIMILARITY (SCS)

Suite à thread Twitter en 2020, l'idée de remplacer une couche de convolution par une couche de similarité cosinus mise à la puissance p a été discuté puis implémenté. Voici les points majeur que l'on connaît pour le moment :

1. On ne gagne pas forcément en performance : vitesse de calcul ou précision
2. On gagne énormément en taille de modèle : un modèle avec SCS a beaucoup moins de paramètres qu'un réseau convolutionnel classique
3. Il faut plutôt utiliser une couche de pooling qui travaille en valeur absolue

Il y a bien sûr d'autres points que l'on connaît, on souhaite se focaliser sur ces trois-là.

Consignes

- ▶ Expliquer mathématiquement comment fonctionne SCS et faire le lien avec les points 1 et 3
- ▶ Comprendre comment ajouter ces nouvelles couches qui ne sont pas implémentés pour le moment sur Keras
- ▶ Entraîner avec des performances correctes un réseau avec SCS

SÉANCE 3 - COMPLÉMENTS

- 1 Séance 1 - Réseaux denses 1
- 2 Séance 2 - Réseaux convolutionnel 7
- 3 Séance 3 - Compléments 13
 - 3.1 Comparaison d’optimizer 15
 - 3.2 Échéancier pour le *learning rate* 16
 - 3.3 Batch Normalization et SeLU 17
 - 3.4 Dropout et Data Augmentation 18
- 4 Séance 4 - Réseaux récurrents 19

SÉANCE 3 - COMPLÉMENTS

CONSIGNES

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

- ▶ 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.
- ▶ Noter les différentes remarques que vous pourrez faire de sorte à pouvoir essayer de les traiter théoriquement
- ▶ Vous pouvez choisir votre propre sujet à condition de le faire valider en amont

Le choix du type de réseau de neurones (convolutionnel ou dense) et le choix du dataset est laissé libre sauf mention contraire.

SÉANCE 3 - COMPLÉMENTS

A - COMPARAISON D'OPTIMIZER

Il existe de nombreux schéma d'optimizer, et à chaque présentation l'optimizer est présenté comme le meilleur. A la lumière de l'étude réalisée *Descending through a crowded valley-benchmarking deep learning optimizers* en 2021, on se propose de comparer 3 à 4 schéma de descente de gradient. Les optimizer sélectionnés sont soumis à validation, et s'il ne sont pas décrit dans le cours devront être expliqué mathématiquement.

On peut également comparer uniquement la différence avec le *weigh decay*. Pour cela on entraînera un réseau de neurones avec l'optimiser Adam avec ou sans régularisation \mathcal{L}_2 , et un autre avec AdamW.

Consignes

- ▶ Automatiser le plus possible le processus de comparaison
- ▶ Produire des résultats simplement interprétable et lisible
- ▶ Identifier et présenter les limites de la comparaison réalisée

SÉANCE 3 - COMPLÉMENTS

B - ÉCHÉANCIER POUR LE *learning rate*

On se propose de mesurer à nouveau l'importance du *learning rate* dans l'entraînement d'un réseau de neurone, mais cette fois à la lumière de son échéancier. On peut approcher le sujet de plusieurs manières :

1. Comparer différents paramètres pour un seul échéancier
2. Comparer plusieurs échéancier, en faisant attention que ça soit comparable

On peut également réaliser les deux points précédents en même temps.

Consignes

- ▶ Automatiser le plus possible le processus de comparaison
- ▶ Produire des résultats simplement interprétable et lisible
- ▶ Identifier et présenter les limites de la comparaison réalisée

SÉANCE 3 - COMPLÉMENTS

C - BATCH NORMALIZATION ET SELU

La fonction d'activation SeLU est proposé dans l'article *Self-normalizing neural networks* en 2017. On se propose de vérifier que l'objectif théorique de SeLU est bien rempli. Pour cela, on veut au minimum comparer la performance d'un réseau de neurones entraîné avec ReLU et Batch Normalization par rapport à une réseau de neurones entraîné avec SeLU sans BatchNormalization.

De plus, on souhaiterai savoir quelle est la meilleure position de la couche de BatchNormalization : avant ou après la fonction d'activation ?

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 3 - COMPLÉMENTS

D - DROPOUT ET DATA AUGMENTATION

Nous avons vu en cours la couche de Dropout, et avec le notebook de départ la *data augmentation*. On souhaite mesurer l'impact de ces deux outils pour la généralisation d'un réseau de neurones. On se propose de considérer plusieurs valeurs de Dropout et/ou plusieurs paramètres de la *data augmentation*.

On peut également travailler sur la position et la force du Dropout dans un réseau.

Consignes

- ▶ Automatiser le plus possible le processus de comparaison
- ▶ Produire des résultats simplement interprétable et lisible
- ▶ Identifier et présenter les limites de la comparaison réalisée