

# Chemins spécifiques pour la classification dans les réseaux de neurones profonds

Elhouiti Chakib - Kezzoul Massili

Université de Montpellier

1<sup>er</sup> juin 2021

## 1 Introduction

Les réseaux de neurones profonds

Le jeu de données

Problématique

Solution proposée

## 2 Organisation

## 3 Analyse des données

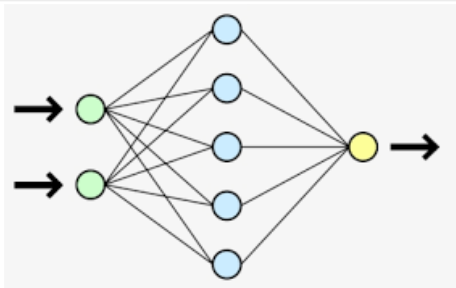
## 4 Développement de l'architecture

## 5 Analyse des résultats

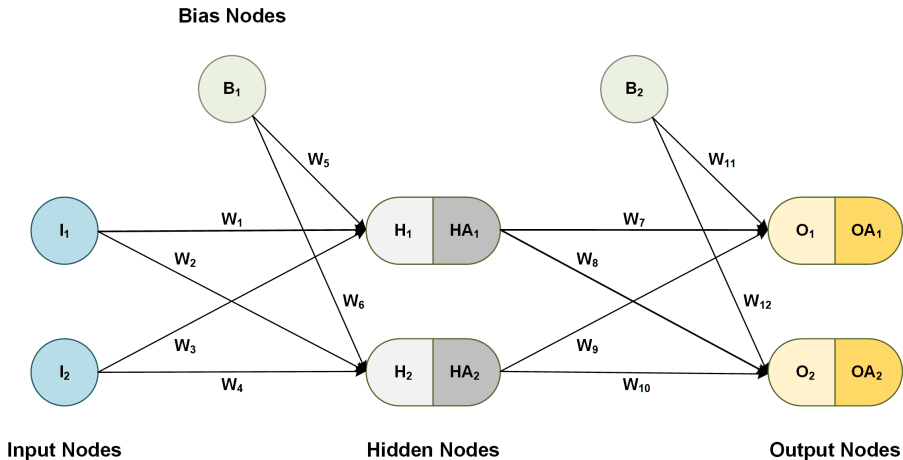
# Présentation des réseaux de neurones

Les réseaux de neurones sont une des méthodes d'apprentissage automatique utilisées en machine learning. Ils sont inspirés du fonctionnement d'un cerveau humain.

Ils sont constitués de plusieurs couches consécutives de neurones. Chaque neurone est relié à un certain nombre de neurones de la couche précédente et de la couche suivante



# Feedforward



# Back propagation

## Comment un réseau *apprend* ?

Pour qu'un réseau puisse « apprendre » :

- ▶ Calculer l'*erreur*  $e = \frac{1}{2}(pred - result)^2$  ;
- ▶ Ajuster les poids afin de minimiser l'erreur  $e$  ;
- ▶ En répétant ces étapes sur une masse importante de données, le réseau pourra « apprendre » à répondre à une question.



# Black-box

Bien que les réseaux de neurones donnent de bons résultats généralement, leur structure leur donne un effet de *boîte noire*. C'est-à-dire qu'ils permettent de donner des prédictions, mais sans pouvoir expliquer le raisonnement qui les a menés à ces dernières.

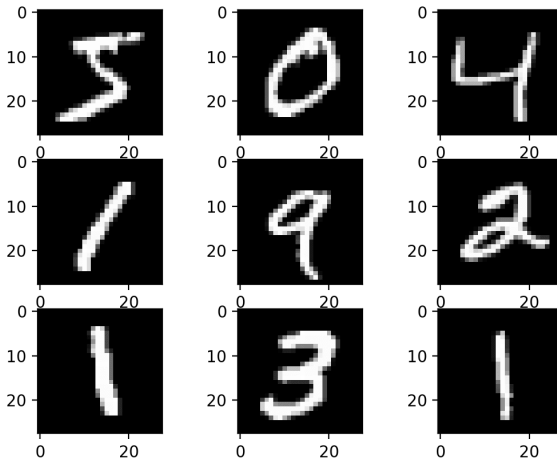


# Le jeu de données

## Mixed National Institute of Standards and Technology

MNIST est une base de données très utilisée par les ingénieurs en *Data Science*. Elle est composée de 70000 images de chiffre manuscrit. Chaque image est composée de  $28 * 28$  pixels de 0 à 255 représentant le niveau de gris.

# MNIST





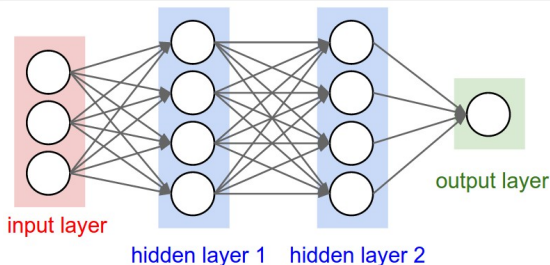
# Problématique

## Boîte noire

Les réseaux de neurones semblent s'appliquer à la manière d'une boîte noire. Aucune information n'est fournie sur ce qui les a conduits à atteindre leurs prédictions.

## Objectifs

L'objectif est de comprendre le fonctionnement interne d'un réseau de neurones et de repérer des signatures d'activation de neurones en variant les données.



# Questions qu'on se pose

Par exemple, si on entraîne un modèle à reconnaître des images de 1 et de 7

- ▶ À partir de quelle couche le modèle change de comportement pour reconnaître une image ?
- ▶ Les signatures des images de 7, sont-elles différentes de ceux des 1 ?
- ▶ Si on passe une image de 3 au modèle, à quoi va ressembler sa signature ?



UNIVERSITÉ  
DE MONTPELLIER

# Solution proposée

- ▶ Construire des réseaux de neurones.
- ▶ Récupérer, pour chaque donnée, la sortie des couches cachées.
- ▶ Extraire les signatures grâce à des algorithmes de *clustering*.
- ▶ Réaliser une interface de visualisation en utilisant différentes techniques.
- ▶ Analyser les résultats et répondre aux questions.



① Introduction

② **Organisation**

③ Analyse des données

④ Développement de l'architecture

⑤ Analyse des résultats

# Organisation du projet

## Temps de travail

travailler un maximum de temps ensemble et de manière très régulière.

## Progression

Réalisation des tâches en retard durant le week-end pour ne pas cumuler de retard

## Réunions

Toutes les deux semaines, nous nous sommes réunis avec notre encadrant, M. Pascal Poncelet.



UNIVERSITÉ  
DE MONTPELLIER

# Découpage du projet

## Analyse des données

La première étape qui consiste à analyser les données MNIST et définition des outils utiles à leur manipulation.

## Développement

Implémentation des modèles d'apprentissage automatique et extraction des informations internes à ces derniers.

## Analyse des Résultats

Réalisation d'une interface web et effectuation d'expérimentations pour pouvoir en tirer des conclusions

## 1 Introduction

## 2 Organisation

## 3 Analyse des données

Découpage des données  
Prétraitement

## 4 Développement de l'architecture

## 5 Analyse des résultats

## Importance de l'analyse

L'objectif de l'analyse des données, c'est de savoir comment sont nos données et comment on peut les utilisées.



UNIVERSITÉ  
DE MONTPELLIER



# Découpage des données

- ▶ Garder un nombre précis d'images pour un ensemble de chiffres définis.
- ▶ Faciliter la phase de développement.
- ▶ Pouvoir mieux visualiser les résultats sur un petit ensemble de données.



## Scaling

Utilisation de la normalisation, qui consiste à mettre les valeurs des images entre 0 et 1 au lieu de 0 et 255.

## Flattening

Aplatissement des images pour avoir un tableau à une seule dimension au lieu d'une matrice à deux dimensions.

Transformation des labels en un vecteur binaire contenant que des 0 et des 1.

- ▶ Taille du vecteur égale au nombre de labels uniques à garder.
- ▶ Tri des labels à garder.
- ▶ Mettre un 1 à la case du label correspondant et des 0 aux autres cases.
- ▶ Ex : transformation en vecteur des images de 1, 3 et 7.
- ▶ Pour un 1  $\Rightarrow [1,0,0]$ .
- ▶ Pour un 3  $\Rightarrow [0,1,0]$ .
- ▶ Pour un 7  $\Rightarrow [0,0,1]$ .

## ① Introduction

## ② Organisation

## ③ Analyse des données

## ④ Développement de l'architecture

Technologies utilisées

Modèle d'apprentissage

Signature et Clustering

Interface de visualisation

## ⑤ Analyse des résultats

# Jupyter notebook

Les notebooks Jupyter offrent un excellent moyen d'écrire et d'itérer sur du code Python.

C'est un outil incroyablement puissant pour développer et présenter de manière interactive des projets de science de données.

Jupyter notebook intègre le code et sa sortie dans un document unique qui combine des visualisations, du texte, des équations mathématiques et d'autres médias riches.



UNIVERSITÉ  
DE MONTPELLIER

# Tensorflow, Keras

## Tensorflow

Initialement créé par l'équipe Google Brain à des fins internes, telles que le filtrage du spam sur Gmail, il est devenu open-source en 2015.

Tensorflow est souvent utilisé pour résoudre des problèmes d'apprentissage profond et pour la formation et l'évaluation des processus jusqu'au déploiement du modèle.

## Keras

Keras est une API de réseau neuronal open source écrite en Python. Il peut fonctionner sur plusieurs frameworks d'apprentissage en profondeur et d'apprentissage automatique.

# Voilà

Voilà permet de transformer un notebook jupyter en une application web autonome (standalone).

Voilà permet aussi de changer l'interface graphique de la page d'accueil grâce à des templates.

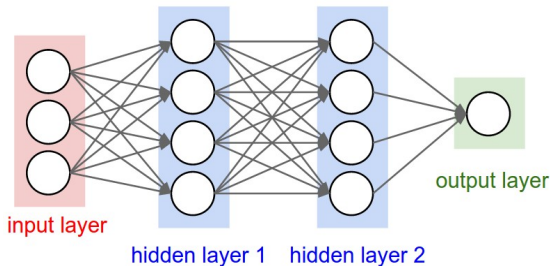


# Modèle d'apprentissage

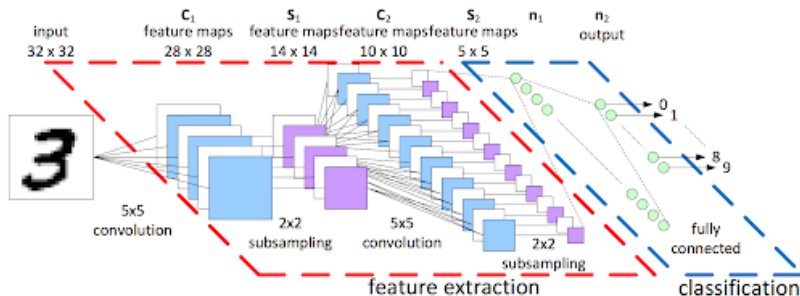
## Types de réseau

Les types de réseau diffèrent par plusieurs paramètres :

- ▶ la topologie des connexions entre les neurones ;
- ▶ la fonction d'agrégation utilisée ;
- ▶ et bien d'autres paramètres.



# Réseau de neurones à convolution



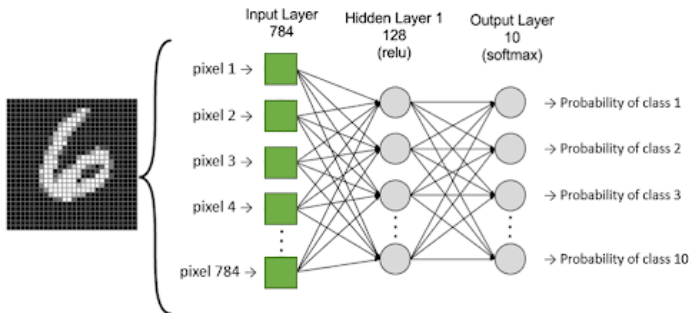
UNIVERSITÉ  
DE MONTPELLIER



# Signature

## Définition

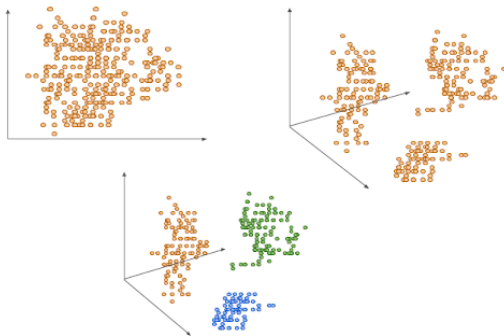
Soit un réseau à  $N$  couches cachées, La *signature*  $S$  d'une image qui traverse ce réseau, se définit ainsi :  $S = (H_1, \dots, H_N)$ , avec  $H_i$  est le vecteur contenant les valeurs de chaque neurone de la couche  $i$ .



# Clustering

## K-means

**K-means** est un algorithme de *clustering*. Il prend en paramètres les données et un certain  $K$  donnée par l'utilisateur, puis construit  $K$  clusters qui regroupent les données qui sont proches (en terme de distance euclidienne).



# Score *Silhouette*

## Choisir un $K$

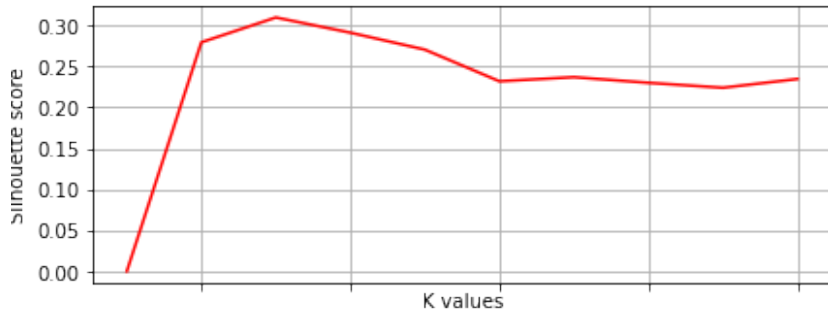
Par contre, l'algorithme ne permet pas de trouver tout seul un  $K$  optimal. Mais il existe une méthode qui calcul la performance d'un *clustering*. La méthode *Silhouette*.

## Score *Silhouette*

Concrètement, cette méthode consiste à calculer pour un clustering, la moyenne du score *Silhouette* de chaque point.  $s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$ .

Avec  $a(i)$  est la mesure de la similarité du point  $i$  avec son cluster et  $b(i)$  le mesure de dissemblance du point  $i$  par rapport aux autres clusters.

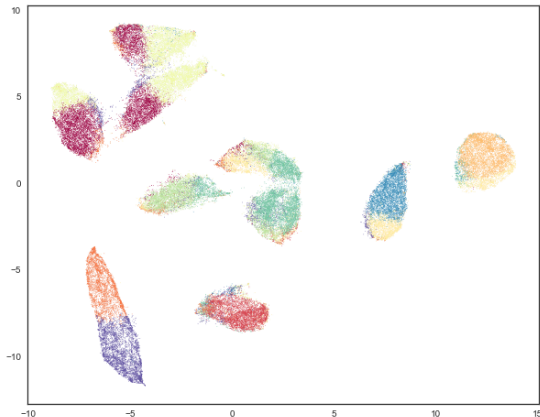




# UMAP

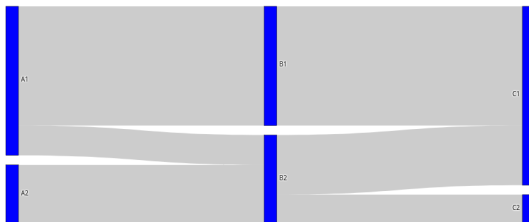
## UMAP

(Uniform Manifold Approximation and Projection) Utilise des algorithmes de mise en page graphique pour organiser les données dans un espace de faible dimension.



# Diagramme de Sankey

Un diagramme de Sankey est un type de diagramme de flux 6 dans lequel la largeur des flèches est proportionnelle au flux représenté.



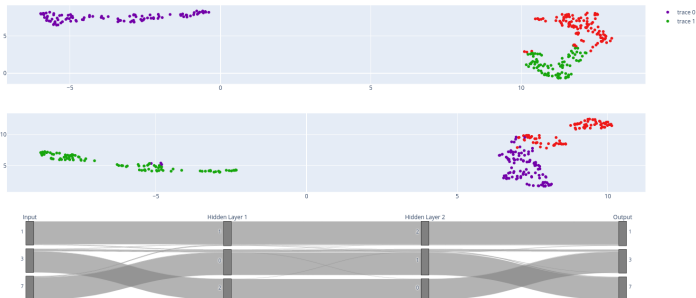
# Application web

## Page web

Transformation d'un Jupyter notebook contenant les différentes visualisations et faisant le lien entre eux.

Color: 

None	Red	Purple	Green	Blue	Pink	Yellow	Orange
Black	Aqua	Brown					



# Application web

## Page d'accueil

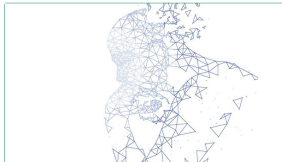
Création d'une page d'accueil personnalisée présentant nos différentes expérimentations.

**MTQ Signature**[Github](#)[À propos](#)

Ce site web présente les différents réseaux de neurones étudiés dans le cadre du projet *mtq-neural-networks*. L'objectif ici est de mieux comprendre comment s'exécute un réseau de neurones profonds. Il s'agit de repérer des signatures d'activation au sein des couches cachées du réseau en fonction des données d'entrées.

Afin de repérer les signatures, nous avons entraîné des modèles sur plusieurs partitions de données. Vous trouverez ci-dessous les liens qui présentent chacune des partitions.

Les liens sont de la forme *signature ABCD N* tel que *A*, *B*, *C* et *D* sont les chiffres sur lesquels s'est entraîné le modèle et *N* le nombre d'exemplaires de chacun de ces chiffres.



Choisir un modèle à ouvrir:

[signature\\_01\\_100.ipynb](#)[signature\\_137\\_100.ipynb](#)[signature\\_14\\_100.ipynb](#)[signature\\_28\\_100.ipynb](#)[signature\\_56\\_100.ipynb](#)[signature\\_69\\_100.ipynb](#)[signature\\_all\\_100.ipynb](#)

Ce site web a été réalisé par l'équipe [MTQ](#). Plus de détails sur le projet sont disponibles sur [Github](#).



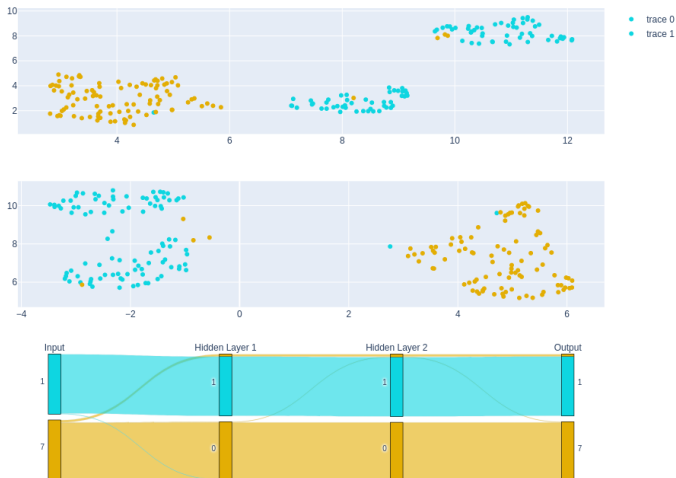
- ① Introduction
- ② Organisation
- ③ Analyse des données
- ④ Développement de l'architecture
- ⑤ Analyse des résultats**
  - Réponses aux questions
  - Conclusion

# Résultats

Afin d'illustrer les résultats obtenu, nous prenons un exemple où on entraîne un modèle à deux couches cachées à reconnaître des images de 1 et de 7.

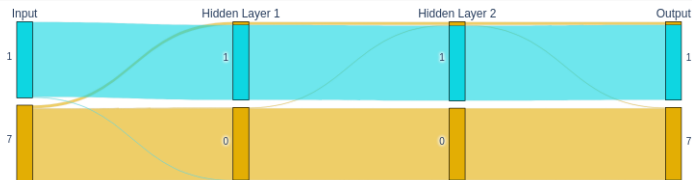


# Résultats



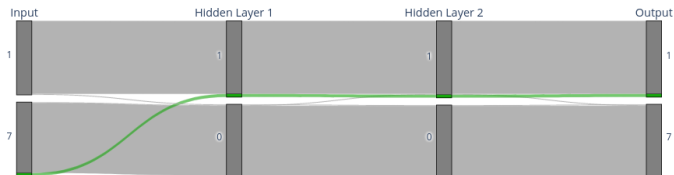
# Changement de comportement

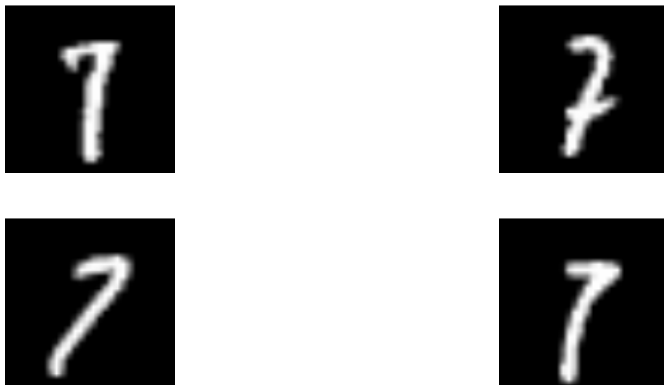
Notre modèle arrive, dès la première couche cachée, à reconnaître une image.



# Différence de signatures

On observe que les signatures des 1 sont majoritairement différentes de celles des 7. Sauf pour quelques rares exceptions.





**Figure** – Les images de 7 ressemblant à des 1



UNIVERSITÉ  
DE MONTPELLIER

# Insertion d'anomalies

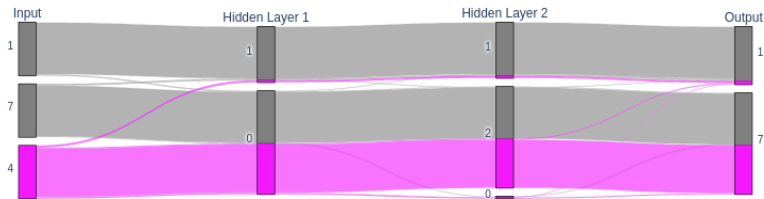


Figure – Insertion d'images de 4



UNIVERSITÉ  
DE MONTPELLIER

# Insertion d'anomalies

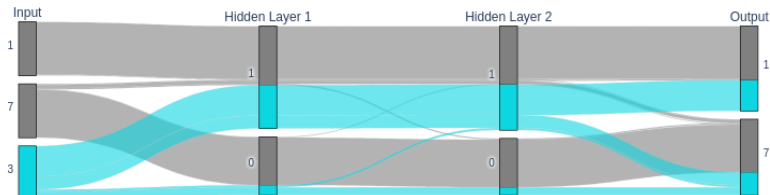


Figure – Insertion d'images de 3





# Conclusion

Soit  $n$  attributs de  $A_1$  à  $A_n$ . On cherche à entraîner un modèle à classifier ces données en deux classes  $C_1$  et  $C_2$ . Le modèle peut détecter des similarités entre les objets sur un attribut  $A_i$  qui indique qu'il existe peut-être *une corrélation*, mais pas forcément *une causalité*, avec les deux classes  $C_1$  et  $C_2$ . Ce qui mènera notre modèle à de possibles mauvaises classifications sur des données qu'il n'a jamais vues.



Merci pour votre attention.



UNIVERSITÉ  
DE MONTPELLIER