# Machine learning

## Introduction et utilisation non supervisée

novembre 2018 @ TELECOM Nancy

T. Bagrel

# Définitions

## Machine learning

Apprentissage automatique / statistique (général)

- supervisé
- non supervisé

## Deep learning

Réseau de neurones multicouches

# Deep learning

## Détail

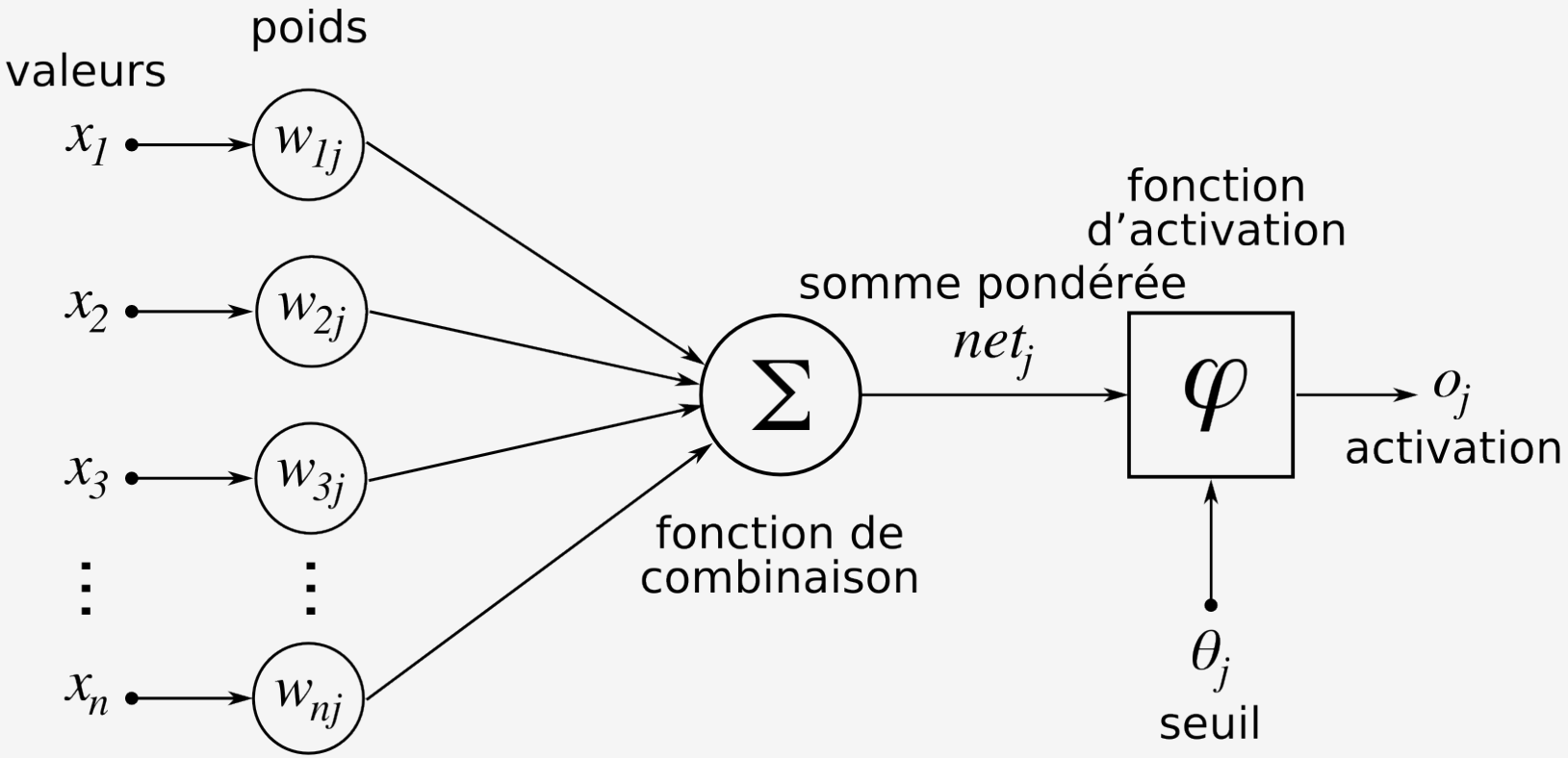
### Plusieurs types de deep learning

- Deep Neural Network (DNN) : beaucoup de couches de neurones
- Convolutional Neural Network (CNN) : découpage du problème en sous-parties, avec un cluster de neurones par sous-partie $\rightarrow$ adapté pour le traitement d'images
- Deep Belief Network (DBN) : première phase non supervisée puis phase supervisée facilitée par le pré-traitement

Source : `https://goo.gl/7rcjph`

# Neurone

## Perceptron



# Machine learning

## Erreur et backpropagation

Important : on est ici dans le cadre de l'apprentissage supervisé

### Erreur

$$\mathrm{E} = f(\mathrm{val}_{\mathrm{attendue}}, \mathrm{val}_{\mathrm{obtenue}})$$

### Descente du gradient

$$\Delta w = -\eta \frac{\partial \mathrm{E}}{\partial w}$$

# Clustering

(Machine learning)

## Idée

Partitionner les données d'entrée sans supervision

## Algorithme simple

K-mean clustering

# K-mean clustering

## Algorithme

### Cas pratique

Points sur une surface délimitée en 2D

### Description

1. On place (aléatoirement) autant de drapeaux (centroids) que l'on veut de groupes (clusters)
2. Chaque point est rattaché au drapeau le plus proche
3. On replace le drapeau au centre des points qui lui sont rattachés
4. Tant que les drapeaux ne sont pas assez stabilisés, on recommence le rattachement (étape 2)

# Cas simple

## étape par étape

Lancement du programme

# Amélioration

## Laisser l'algorithme trouver le nombre optimal de clusters ?

### Contraintes

- Trouver une définition d'une "bonne partition"
- Fixer des limites à la recherche

# Définition d'une bonne partition

(subjectif)

## Ma proposition

- **H** Homogénéité : les groupes doivent avoir des tailles pas trop différentes (pour éviter une partition en $1$ et $10^9$ éléments)
- **D** Distinction : les groupes ne doivent pas être trop proches les uns des autres pour pouvoir les distinguer
- **S** Stabilité : voir `final_eps`
- **P** Proximité : les éléments d'un cluster doivent être le plus proche possible les uns des autres

# Comment équilibrer ces facteurs ?

## Comportement idéal attendu

- H grand quand les longueurs sont proches
- D `1` dès que les groupes sont assez éloignés, 0 sinon
- S inv. de `final_eps`
- P grand quand les points sont proches de leur centroid

# Implémentation de ces fonctions

Code du programme

# Testons !

Lancement du programme

# Application à des profils de patients

## Normalisation et distance

### Normalisation et centrage

Dans le cas où les entrées sont vectorielles, avec des composantes non directement comparables, nécessité d'une normalisation et d'un centrage

### Distance

La question la plus délicate reste celle de la distance, sans introduire de biais !

# Réduire la dimension des entrées

## Machine learning – non supervisé

### Principle Composant Analysis

Ne capturer que les composantes des vecteurs qui impactent le plus la variance des données

### Singular Value Decomposition

Réduire une grande matrice d'entrée en 3 sous-matrices qui contiennent la plus grande partie des informations

# Retour sur la backpropagation ?

Si il reste du temps...

# Sources

## Remerciements

Merci de votre attention !

- https://goo.gl/W3THM6
- https://goo.gl/Wkbkeo
- https://goo.gl/aJLHRo
- https://goo.gl/QACDyV
- https://goo.gl/aGquwV
- https://goo.gl/24oeED
- https://goo.gl/KLymMw
- https://goo.gl/amYAT2