Data Mining

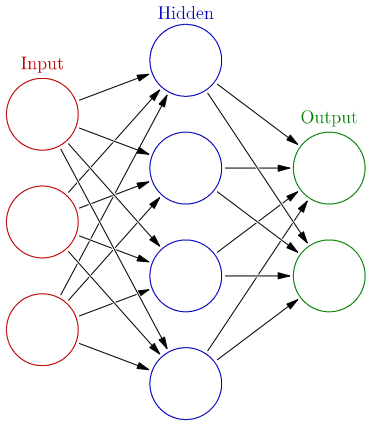
John Samuel
CPE Lyon

Year: 2020-2021
Email: john(dot)samuel(at)cpe(dot)fr

Objectifs

1. Apprentissage machine
2. Apprentissage profond
3. Apprentissage par renforcement
4. Licences de données, éthique et vie privée

Réseaux de neurones artificiels

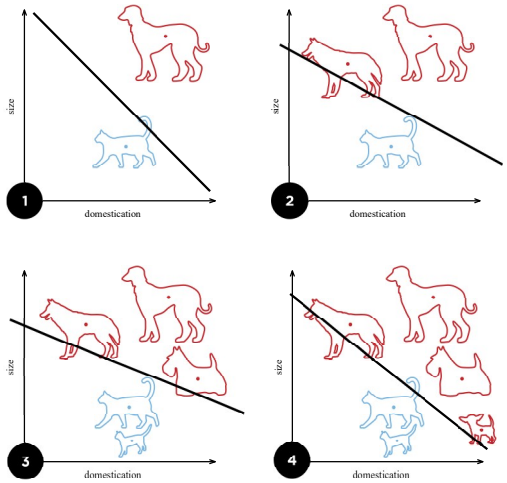


Réseaux de neurones artificiels

Perceptron

- Algorithme pour l'apprentissage supervisé des classificateurs binaires
- Le classificateur binaire est un classificateur qui décide si une entrée donnée appartient ou non à une classe particulière
- Inventé en 1958 par Frank Rosenblatt

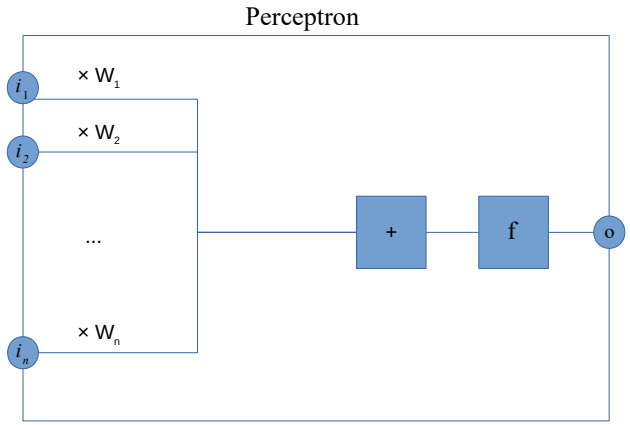
Perceptron



Perceptron en mettant à jour sa limite linéaire à mesure que d'autres exemples de formation sont ajoutés.[1]

1. Source: https://en.wikipedia.org/wiki/File:Perceptron_example.svg

Perceptron



$$o = f\left(\sum_{k=1}^{n} i_k \cdot W_k\right)$$

Perceptron

Perceptron: Définition formelle

- Soit $y = f(z)$ la sortie du perceptron pour un vecteur d'entrée *z*
- Soit N le nombre d'exemples d'entraînement
- Soit ***X*** l'espace de saisie des caractéristiques
- Soit $(x_1, d_1), \ldots, (x_N, d_N)$ be the ***N*** training examples, where
  - x_i est le vecteur caractéristique de *i*^{ème} exemple d'entraînement.
  - d_i est la valeur de sortie souhaitée
  - $x_{j,i}$ est la *i*^{ème} caractéristique de *j*^{ème} exemple d'entraînement.
  - $x_{j,0} = 1$

Perceptron: Définition formelle

- Les poids sont représentés de la manière suivante:
  - w_i est la *i*^{ème} value of weight vector.
  - $w_i(t)$ est la *i*^{ème} valeur du vecteur de poids à un moment donné t.

Perceptron : Étapes

1. Initialiser les poids et les seuils
2. Pour chaque exemple, (x_j, d_j) dans l'ensemble d'entraînement
   - Calculer la sortie actuelle :

$$y_j(t) = f[w(t).x_j]$$

$$= f[w_0(t)x_{j,0} + w_1(t)x_{j,1} + w_2(t)x_{j,2} + \cdots + w_n(t)x_{j,n}]$$

   - Calculer le poids:

$$w_i(t+1) = w_i(t) + r.(d_j - y_j(t))x_{j,i}$$

r est le taux d'apprentissage.

Perceptron : Étapes

3. Répétez l'étape 2 jusqu'à l'erreur d'itération

$$\frac{1}{s}(\Sigma|d_j - y_j(t)|)$$

est inférieur au seuil spécifié par l'utilisateur γ, ou un nombre prédéterminé d'itérations ont été effectuées, où s est à nouveau la taille de l'ensemble de l'échantillon.

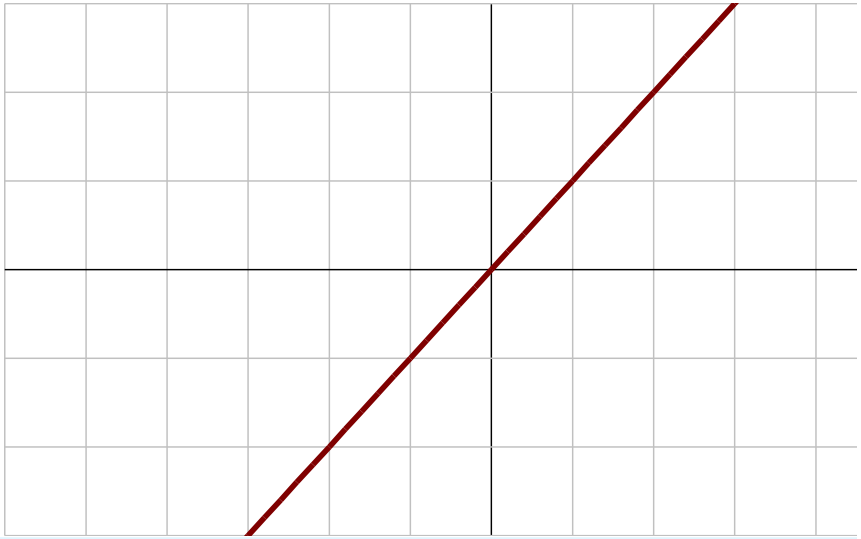
Fonction d'activation: fonction d'identité

Équation

$$f(x) = x$$

Dérivée

$$f'(x) = 1$$



Fonction d'identité

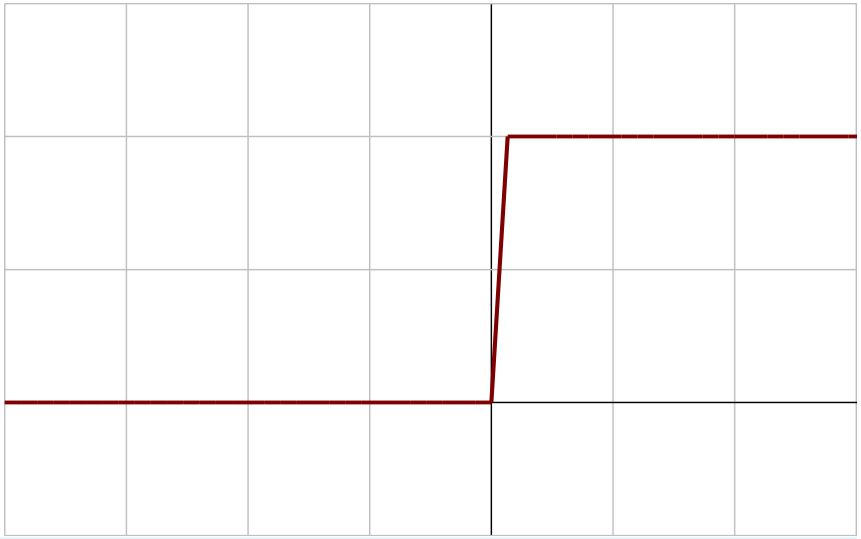
Fonction d'activation: pas binaire

Équation

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

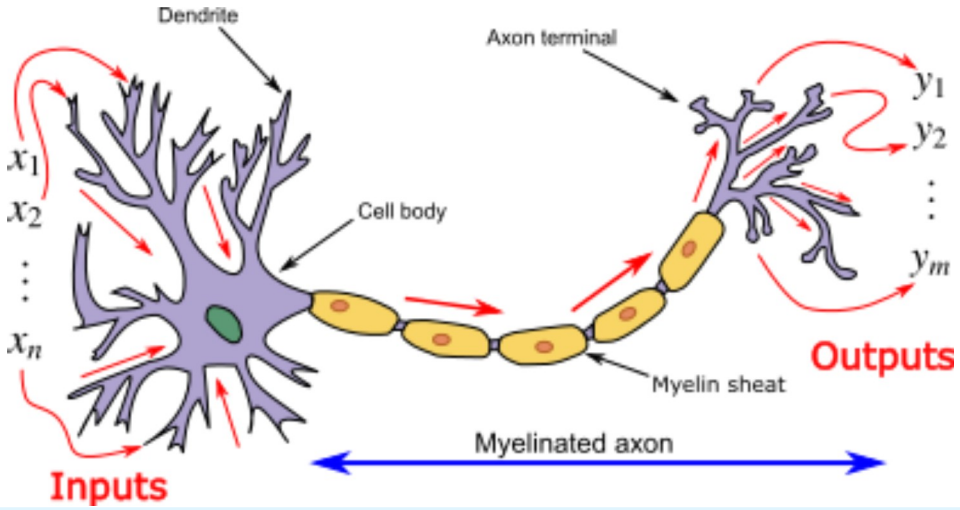
Dérivée

$$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$$



Pas binaire

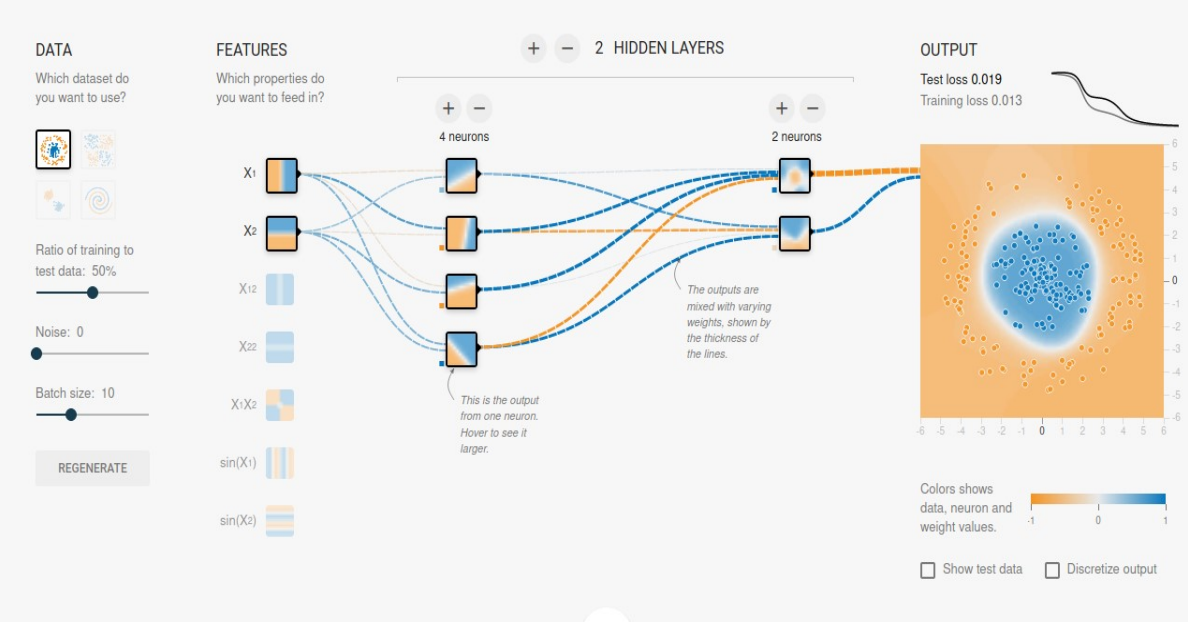
Neurones biologiques



Neurone biologique[1]

1. https://en.wikipedia.org/wiki/File:Neuron3.png

Tensorflow Playground



Source: https://playground.tensorflow.org/

Réseau de neurones artificiels

- collection d'unités ou de nœuds connectés, appelés neurones artificiels, qui modèlent vaguement les neurones d'un cerveau biologique.
- Chaque connexion, comme les synapses dans un cerveau biologique, peut transmettre un signal aux autres neurones.
- Un neurone artificiel qui reçoit un signal le traite ensuite et peut signaler les neurones qui lui sont connectés.
- Le "signal" à une connexion est un nombre réel, et la sortie de chaque neurone est calculée par une fonction non linéaire de la somme de ses entrées.
- Les neurones et les arêtes (connexions) ont généralement un poids qui s'ajuste au fur et à mesure de l'apprentissage.
- Le poids augmente ou diminue la force du signal au niveau d'une connexion.
- Les neurones peuvent avoir un seuil tel qu'un signal n'est envoyé que si le signal global franchit ce seuil.

Réseau de neurones artificiels: les couches

- Les neurones sont agrégés en couches.
- Différentes couches peuvent effectuer des transformations différentes sur leurs entrées.
- Les signaux passent de la première couche (la couche d'entrée) à la dernière couche (la couche de sortie), éventuellement après avoir traversé les couches plusieurs fois.

Réseau de neurones artificiels:

- Les réseaux neuronaux apprennent (ou sont entraînés) en traitant des exemples.
- chaque exemple contient une "entrée" et un "résultat" connus.
- **Erreur**: L'entraînement d'un réseau de neurones à partir d'un exemple donné est généralement effectué en déterminant la différence entre la sortie traitée du réseau (souvent une prédiction) et une sortie cible
- Le réseau ajuste ensuite ses associations pondérées en fonction d'une règle d'apprentissage et en utilisant cette valeur d'erreur.
- Des ajustements successifs amèneront le réseau de neurones à produire un résultat de plus en plus similaire au résultat cible.

Composants des réseaux de neurones

- Neurones
- Connexions et poids
- Fonction de propagation

Composants des réseaux de neurones

Neurones

- Chaque neurone artificiel a des entrées et produit une seule sortie qui peut être envoyée à plusieurs autres neurones.
- Les entrées peuvent être les valeurs caractéristiques d'un échantillon de données externes
- Les sorties des neurones de sortie finale du réseau neuronal accomplissent la tâche
- **Fonction d'activation**
  - Pour trouver la sortie du neurone, nous prenons d'abord la somme pondérée de tous les intrants
  - Nous ajoutons un terme de biais à cette somme. Cette somme pondérée est parfois appelée l'activation.
  - Cette somme est ensuite passée par une fonction d'activation (généralement non linéaire) pour produire le résultat.

Composants des réseaux de neurones

Connexions et poids

- Le réseau est constitué de connexions, chaque connexion fournissant la sortie d'un neurone comme entrée à un autre neurone.
- Chaque connexion se voit attribuer un poids qui représente son importance relative
- Un neurone donné peut avoir plusieurs connexions d'entrée et de sortie.

Composants des réseaux de neurones

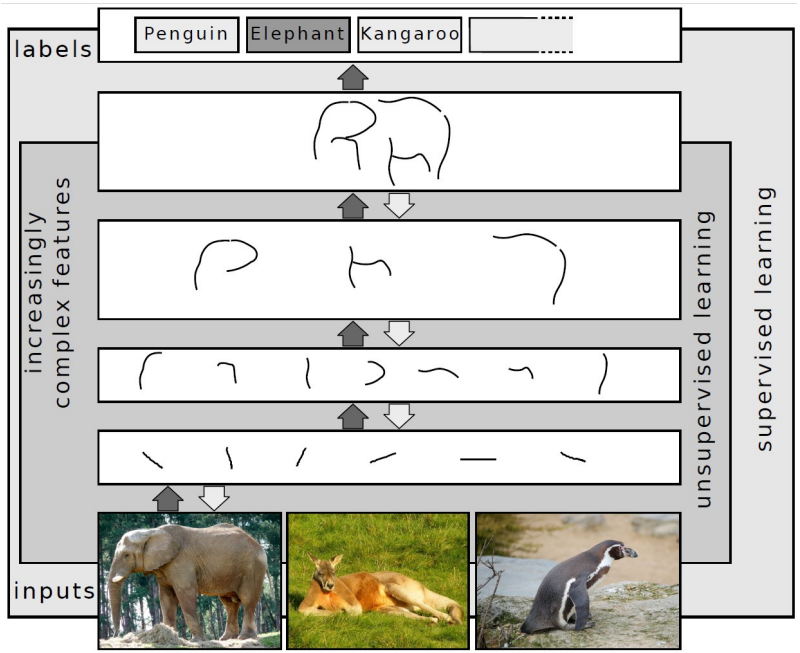
Fonction de propagation

- La fonction de propagation calcule l'entrée d'un neurone à partir des sorties de ses prédécesseurs et de leurs connexions comme une somme pondérée.
- Un terme de biais peut être ajouté au résultat de la propagation

Apprentissage profond

- Le mot "profond" dans l'apprentissage profond vient de l'utilisation de multiples couches dans le réseau neuronal.
- Un perceptron linéaire ne peut pas être un classificateur universel. Un perceptron "monocouche" ne peut pas mettre en œuvre le XOR
- Les réseaux d'apprentissage en profondeur permettent un nombre illimité de couches de taille limitée
- Il utilise plusieurs couches pour extraire progressivement des caractéristiques de l'entrée brute.

2. Apprentissage profond



Source: https://en.wikipedia.org/wiki/File:Deep_Learning.jpg

Composants des réseaux de neurones

Organisation

- Les neurones sont généralement organisés en plusieurs couches
- Les neurones d'une couche se connectent uniquement aux neurones des couches immédiatement précédente et immédiatement suivante.
- La couche qui reçoit les données externes est la **couche d'entrée**.
- La couche qui produit le résultat final est la **couche de sortie**.
- Entre les deux, il y a zéro ou plusieurs couches cachées.

Composants des réseaux de neurones

Organisation et connectivité

- Les couches peuvent être **entièrement connectées**, chaque neurone d'une couche étant connecté à chaque neurone de la couche suivante.
- Les couches peuvent être mis en commun (pooling), c'est-à-dire qu'un groupe de neurones dans une couche se connecte à un seul neurone dans la couche suivante, réduisant ainsi le nombre de neurones dans cette couche

Exemple: Tensorflow

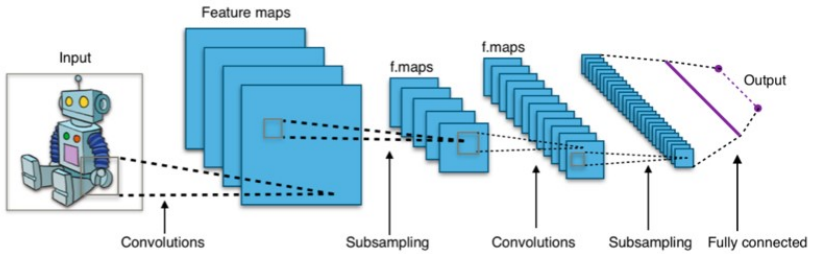
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import SGD

# Créer un modèle séquentiel
model = Sequential()
model.add(Dense(4, activation='relu', input_shape=(3,)))
model.add(Dense(units=2, activation='softmax'))

# Compilation du modèle
sgd = SGD(lr=0.01)
model.compile(loss='mean_squared_error',
     optimizer=sgd,metrics=['accuracy'])
```

Applications

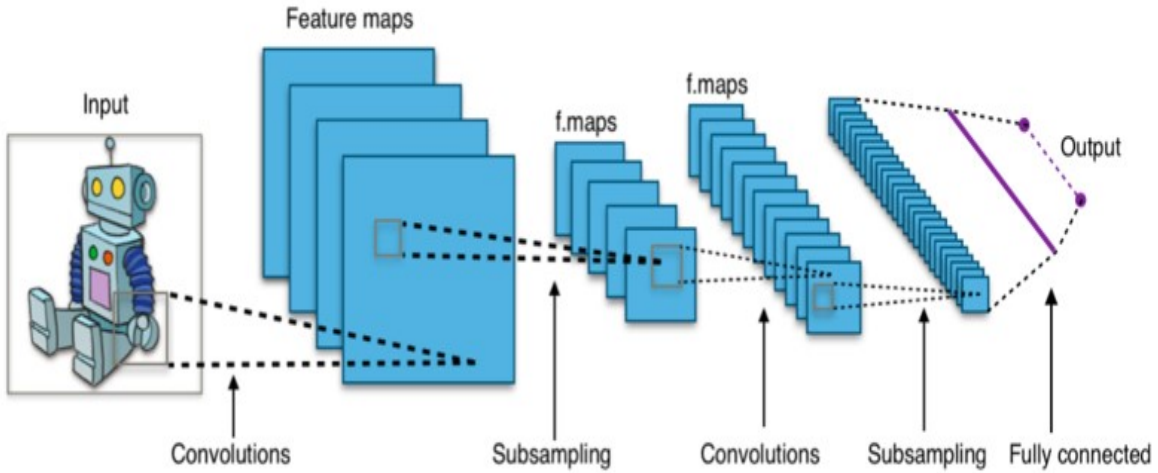
- Vision par ordinateur (reconnaissance de formes)
- Reconnaissance automatique de la parole
- Conception de médicament
- Traitement automatique du langage naturel
- Traduction automatique

Réseaux de neurones convolutionnels

- Convolutional deep neural networks en Anglais
- Inspirés par le cortex visuel des animaux



Réseaux de neurones convolutionnels

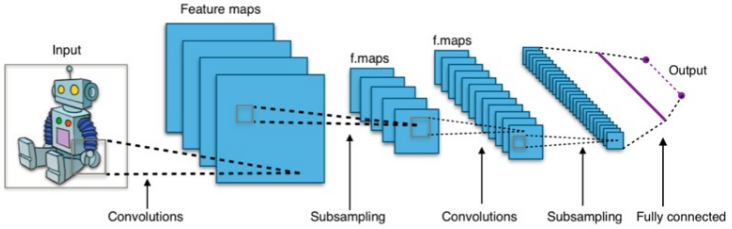


Réseaux de neurones convolutionnels

- Analyse des images
- Utilise la convolution, une opération mathématique linéaire
- Une couche d'entrée et une couche de sortie
- Plusieurs couches cachées, constituées de couches convolutives

Réseaux de neurones convolutionnels

- Ils considèrent le modèle hiérarchique des données et assemblent des modèles plus complexes en utilisant des modèles plus petits et plus simples.
- Un réseau neuronal convolutif est constitué d'une couche d'entrée et d'une couche de sortie, ainsi que de plusieurs couches cachées.
- Les couches cachées d'un CNN consistent généralement en une série de couches convolutionnelles qui se convoluent avec une multiplication
- La fonction d'activation est généralement une couche RELU, et est ensuite suivie par des convolutions supplémentaires telles que des couches de regroupement, des couches entièrement connectées et des couches de normalisation



Noyau (traitement d'image)

Identité

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{matrix}$$

La détection de contours

$$\begin{matrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{matrix}$$

Noyau (traitement d'image)

Box blur

$$\frac{1}{9}\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

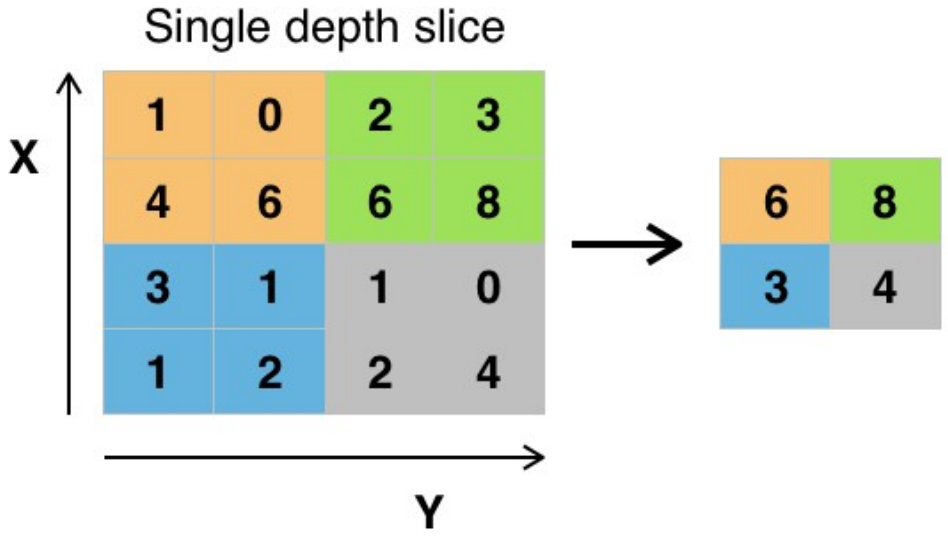
Flou de Gauss 3 × 3

$$\frac{1}{16}\begin{matrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{matrix}$$

Convolution matricielle

$$\begin{matrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{matrix} * \begin{matrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{matrix} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{(m-i)(n-j)} y_{(1+i)(1+j)}$$

Max pooling



Max pooling avec un filtre 2 × 2 et un pas de 2. (Source: https://commons.wikimedia.org/wiki/File:Max_pooling.png)

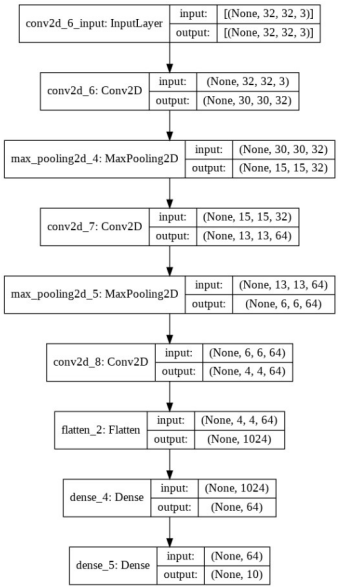
Exemple: Tensorflow (réseaux de neurones

```
import tensorflow as tf

from tensorflow.keras import datasets, layers, models

(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data
train_images, test_images = train_images / 255.0, test_images / 255.0

# Créer un modèle séquentiel (réseaux de neurones convolutionnels)
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

Exemple: Tensorflow (réseaux de neurones

```
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10)

#Compilation du modèle
model.compile(optimizer='adam',
   loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
   metrics=['accuracy'])

history = model.fit(train_images, train_labels, epochs=10,
   validation_data=(test_images, test_labels))
```

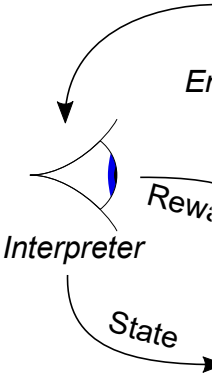
Exemple: Tensorflow (réseaux de neurones



Modèle: https://www.tensorflow.org/tutorials/images/cnn

Apprentissage par renforcement

- Reinforcement learning (en Anglais)
- Inspirée de théories de psychologie animale
- Un agent autonome plongé au sein d'un environnement,
- L'agent doit prendre des décisions en fonction de son état courant.
- L'environnement procure à l'agent une récompense, qui peut être positive ou négative.
- L'objectif est de maximiser la somme des récompenses au cours du temps.

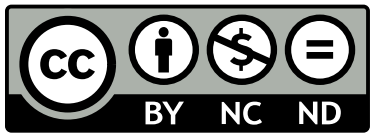


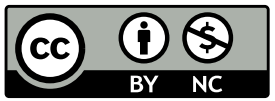
4. Licences, Ethiques et la vie privé

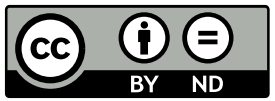
Licences, Ethiques et la vie privé

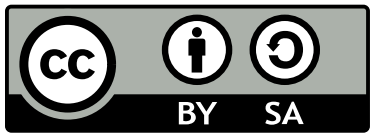
- Droits d'utilisation des données
- Confidentiality and Privacy
- Ethiques

4. Licences, Ethiques et la vie privé

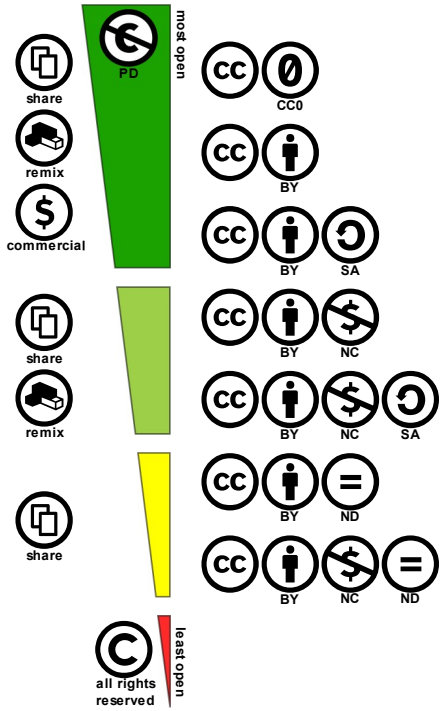








Exemples: Creative Commons (CC)

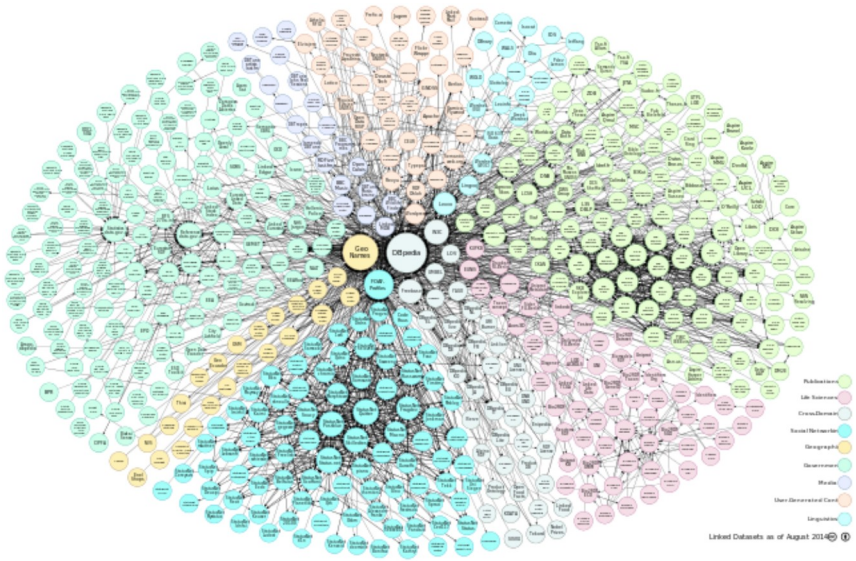


Exemples: Creative Commons (CC)

4. Licences, Ethiques et la vie privé

Données ouvertes

4. Licences, Ethiques et la vie privé



Données ouvertes liées (Linked Open data: LOD)

4. Licences, Ethiques et la vie privé

Données archivées

Ressources en ligne

- Artificial Neural Network
- Noyau (traitement d'image)
- Réseau neuronal convolutif
- Perceptron

Références

Couleurs

- Color Tool - Material Design

Images

- Wikimedia Commons