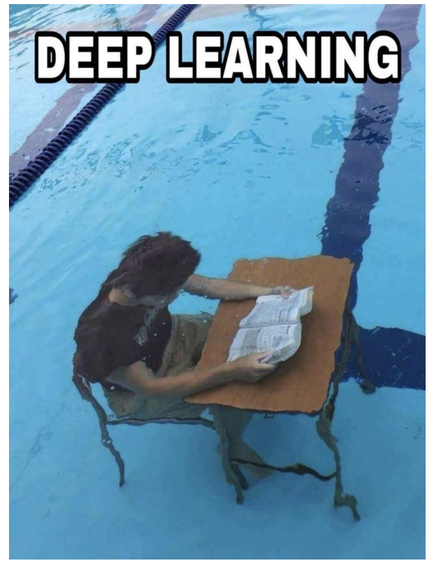
L'apprentissage profond



Julien Cohen-Adad, PhD
Professeur titulaire, Ecole Polytechnique de Montreal
Directeur, Unité de Neuroimagerie Fonctionnelle, Université de Montréal
Membre Académique, Mila - Quebec AI Institute, Montreal, QC, Canada
Chaire de Recherche du Canada en Imagerie par Résonance Magnétique Quantitative

NeuroPoly

Mila

UNF Unité de neuroimagerie fonctionnelle

POLYTECHNIQUE MONTRÉAL

Université de Montréal

Copyrights

Ce cours est inspiré du cours de Jason Dsouza :

👉 **https://www.youtube.com/watch?v=VyWAvY2CF9c&t=1528s**

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

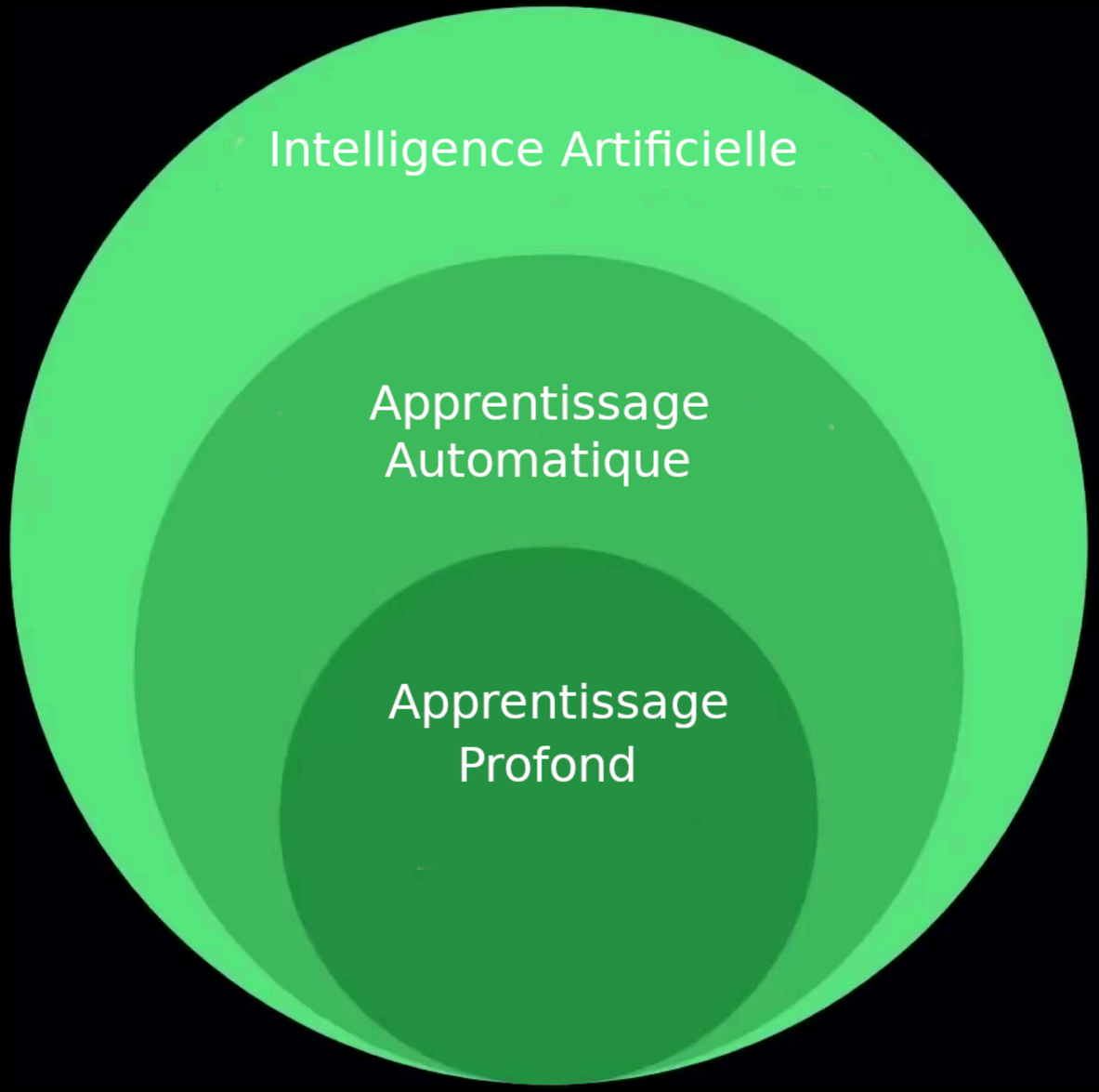
Plan

- Introduction à l'apprentissage profond
  - Qu'est-ce que l'apprentissage profond ?
  - Les Réseaux de Neurones
- Terminologie en apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Introduction

Qu’est-ce que l’Apprentissage Profond ?

Qu’est-ce que l’Apprentissage Profond ?



Qu’est-ce que l’Apprentissage Profond ?

Apprentissage Automatique

Apprendre aux ordinateurs

à reconnaître

des formes dans les données

Qu’est-ce que l’Apprentissage Profond ?

Facile pour

les Humains

Qu’en est-il des Machines ?

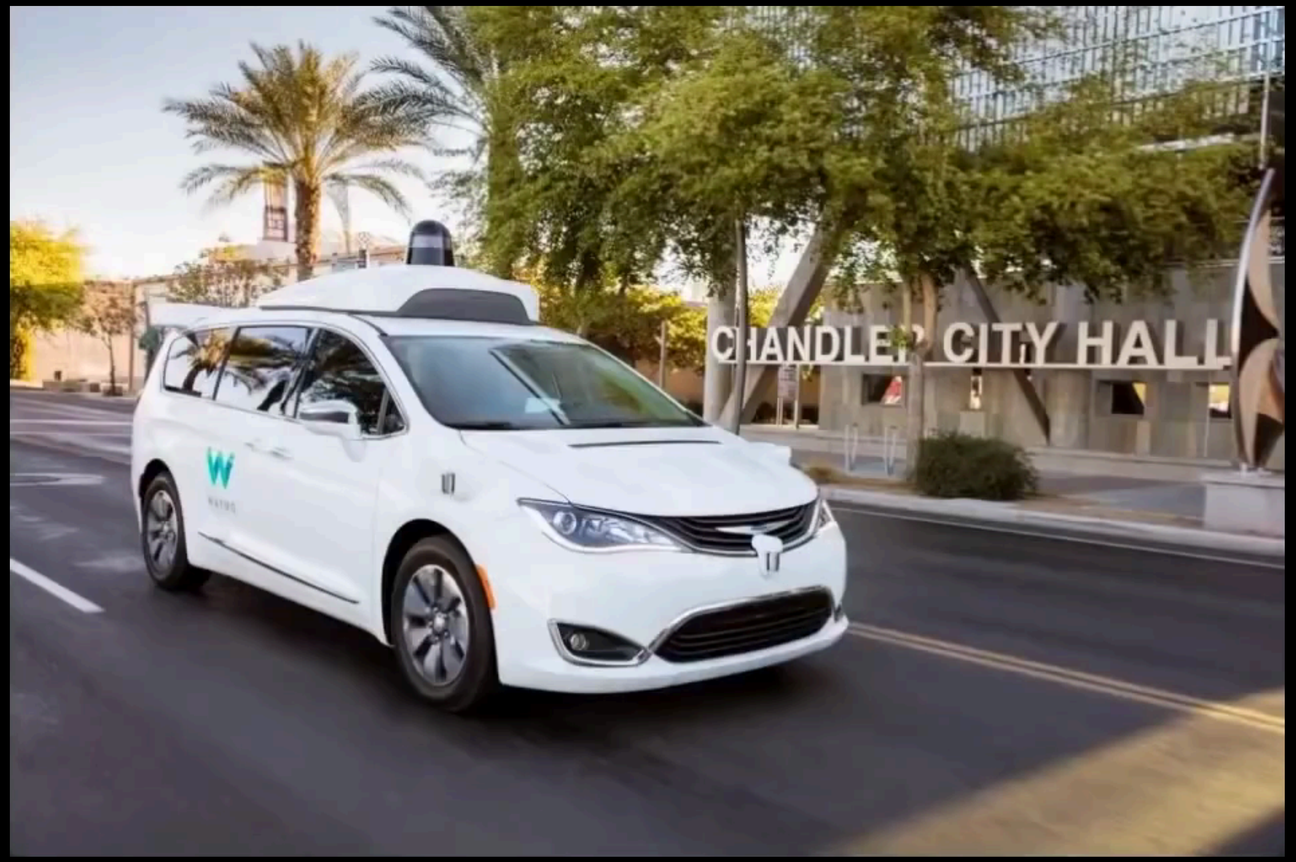
Qu’est-ce que l’Apprentissage Profond ?

Qu’est-ce que l’Apprentissage Profond ?

Qu’est-ce que l’Apprentissage Profond ?

Go

Qu’est-ce que l’Apprentissage Profond ?



Qu’est-ce que l’Apprentissage Profond ?

Qu’est-ce que l’Apprentissage Profond ?

Une technique d’apprentissage automatique qui apprend les fonctionnalités et les tâches directement à partir des données

Les entrées sont exécutées via des réseaux de neurones

Les réseaux de neurones ont des couches cachées "profondes"

Qu’est-ce que l’Apprentissage Profond ?

Intérêt de l’Apprentissage Profond vs. apprentissage "traditionnel" ?

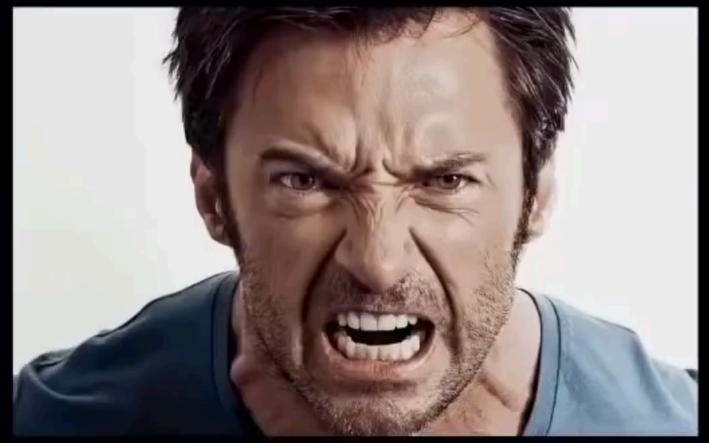
Qu’est-ce que l’Apprentissage Profond ?

Qu’est-ce que l’Apprentissage Profond ?

Qu’est-ce que l’Apprentissage Profond ?

Dans l’Apprentissage Profond, les formes peuvent être apprises uniquement à partir de données brutes !

Qu’est-ce que l’Apprentissage Profond ?



Modèle

Qu’est-ce que l’Apprentissage Profond ?

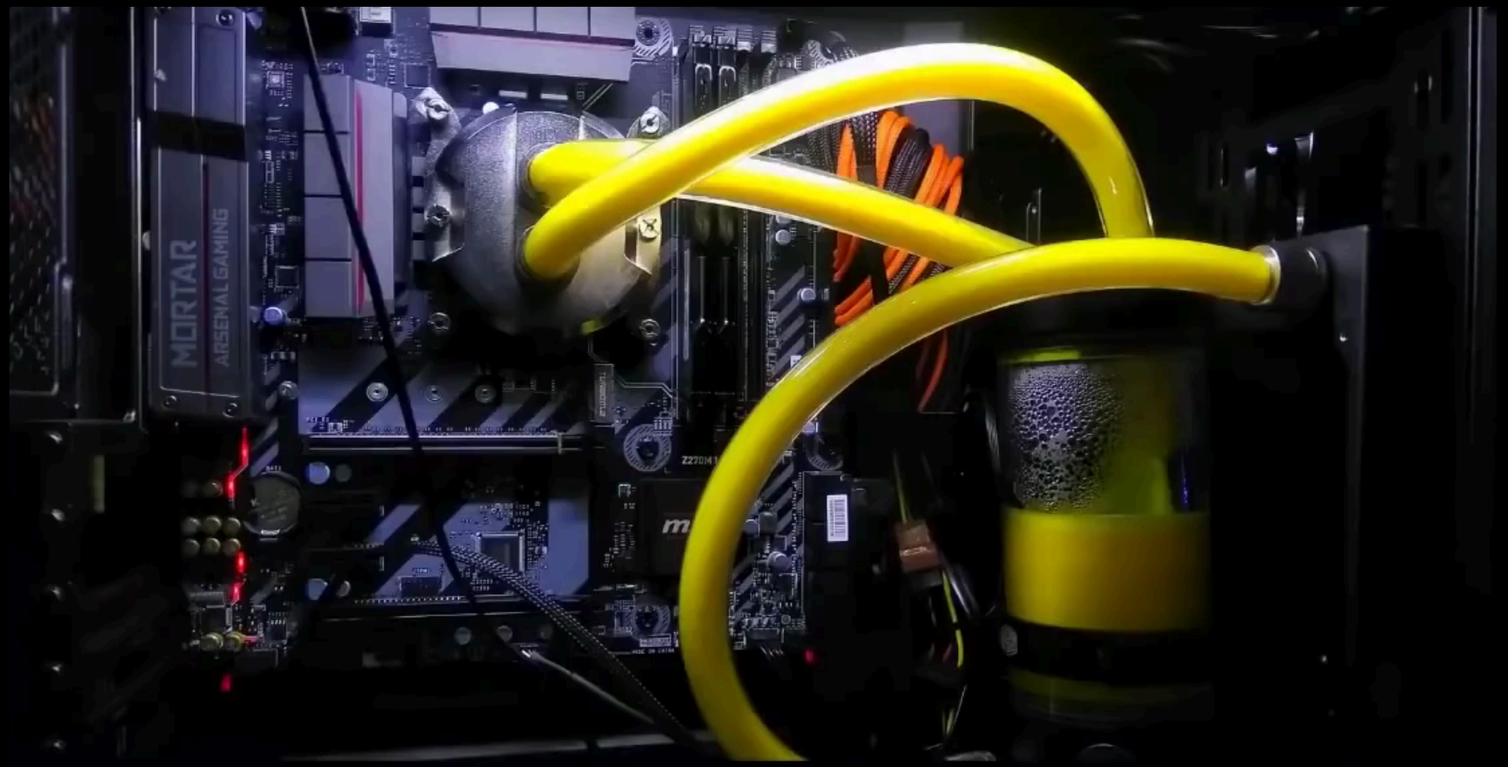
L’Apprentissage Profond - Pourquoi maintenant ?

Qu’est-ce que l’Apprentissage Profond ?

1- Les Données sont répandues

Qu’est-ce que l’Apprentissage Profond ?

2- Du Matériel Amélioré dans son Architecture



Qu’est-ce que l’Apprentissage Profond ?

3- De Nouvelles Architectures Logicielles

Tensorflow

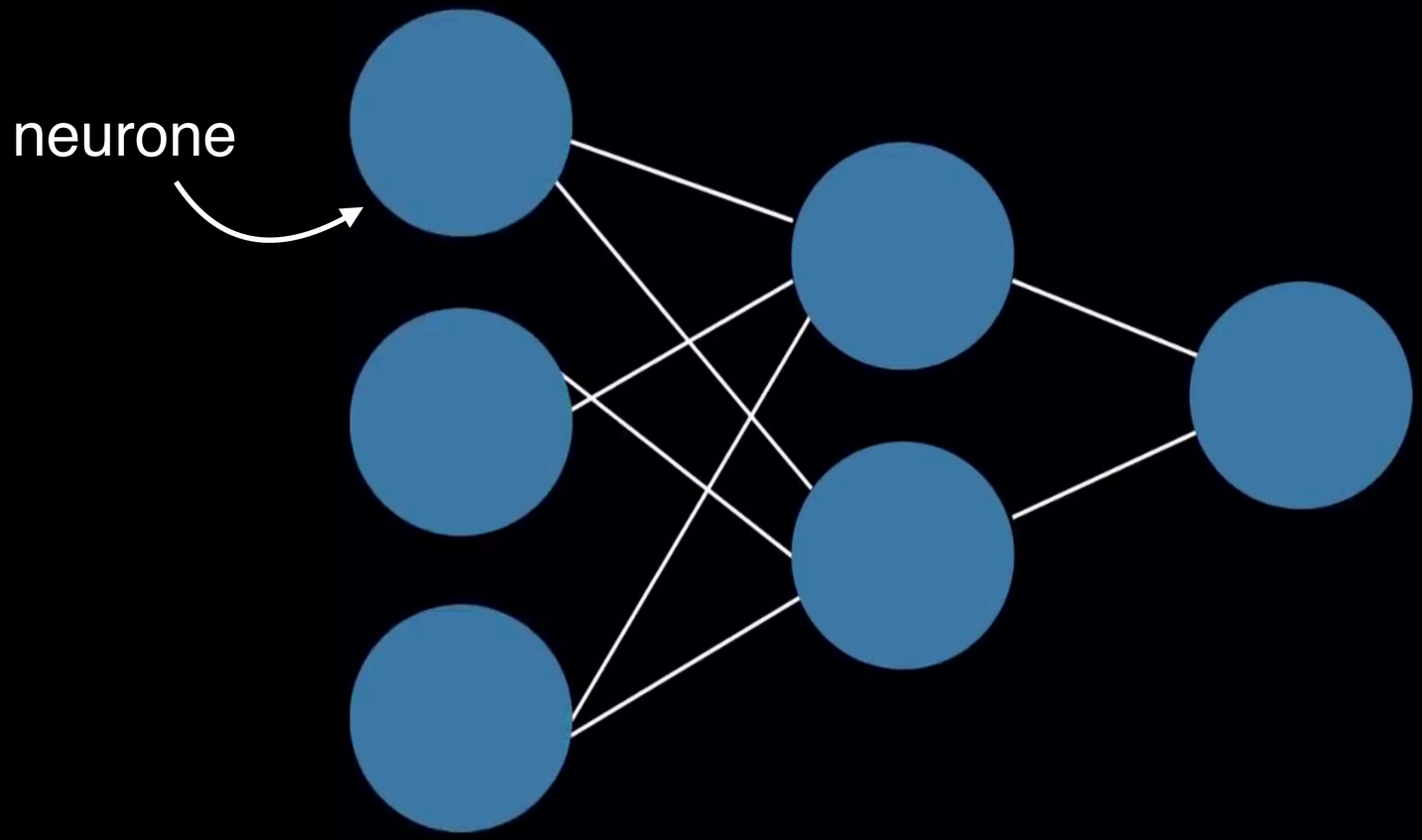
PyTorch

Plan

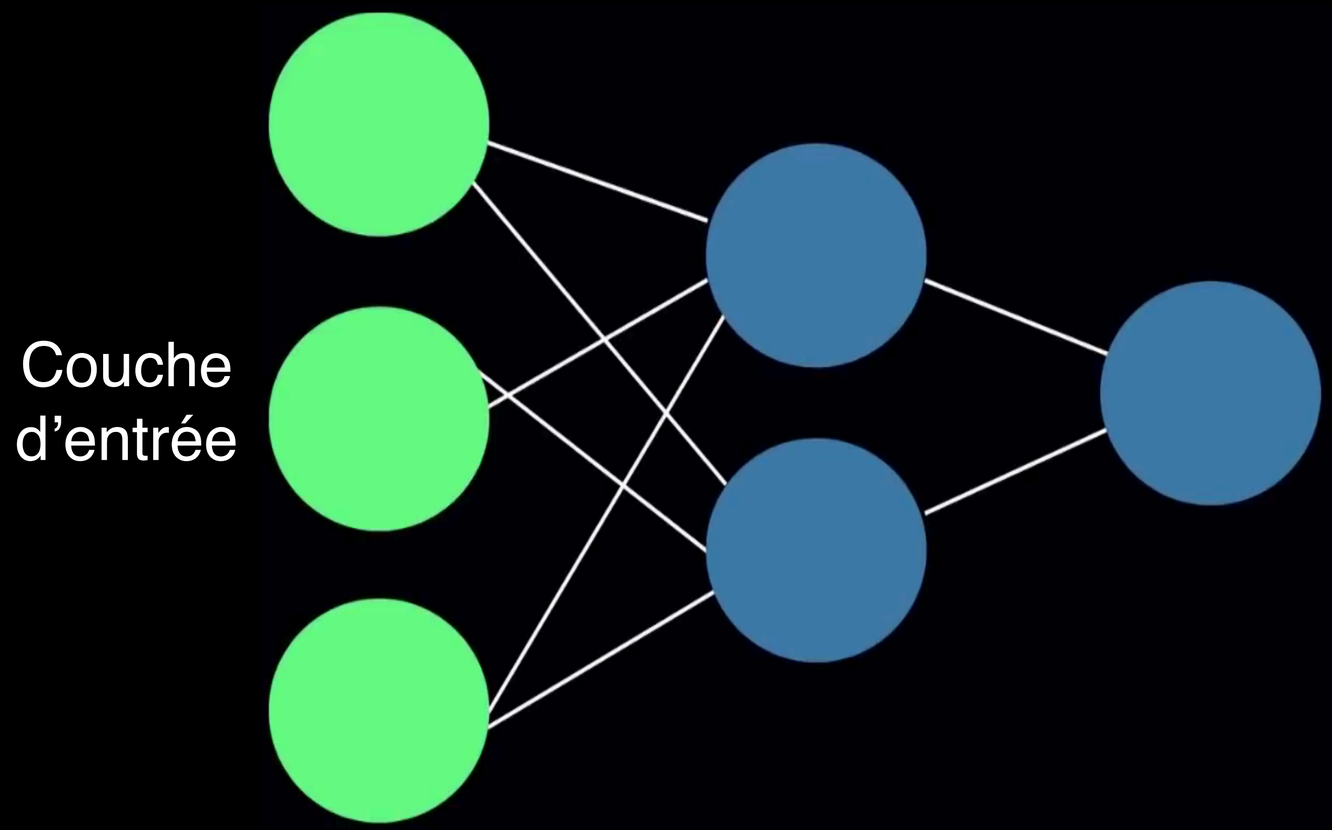
- Introduction à l'apprentissage profond
  - Qu'est-ce que l'apprentissage profond ?
  - Les Réseaux de Neurones
- Terminologie en apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Les Réseaux de Neurones

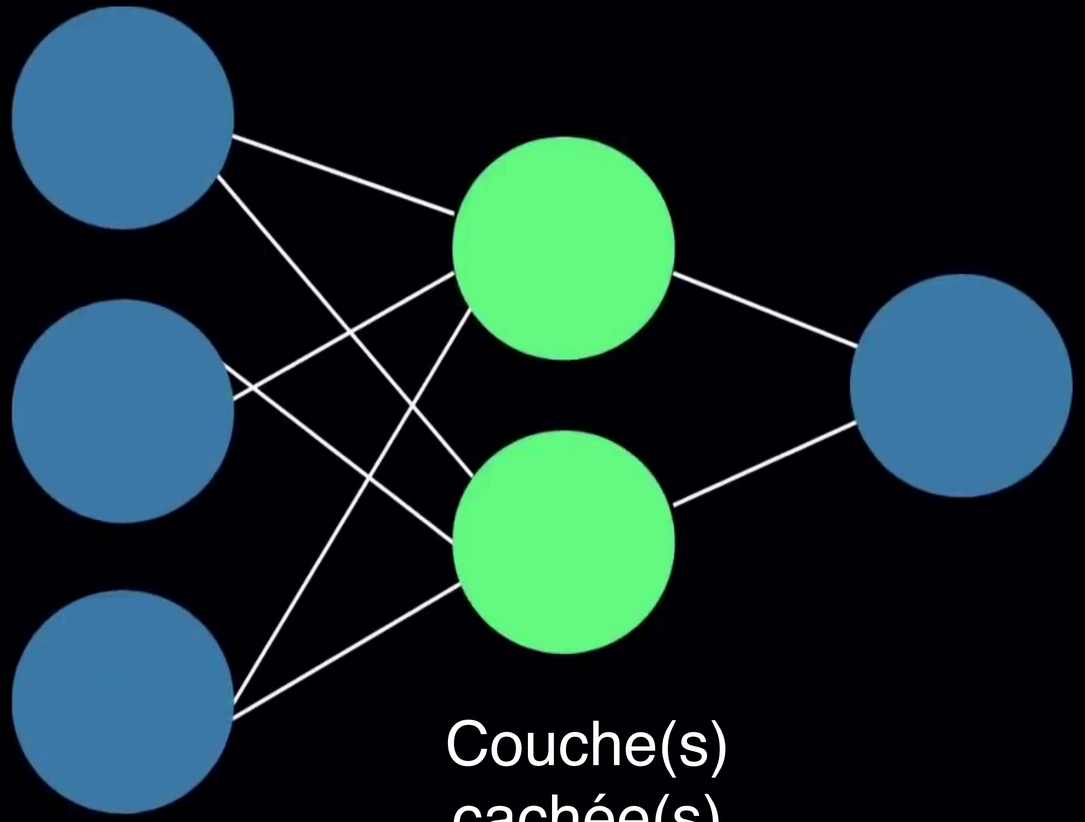
Multi-layer perceptron



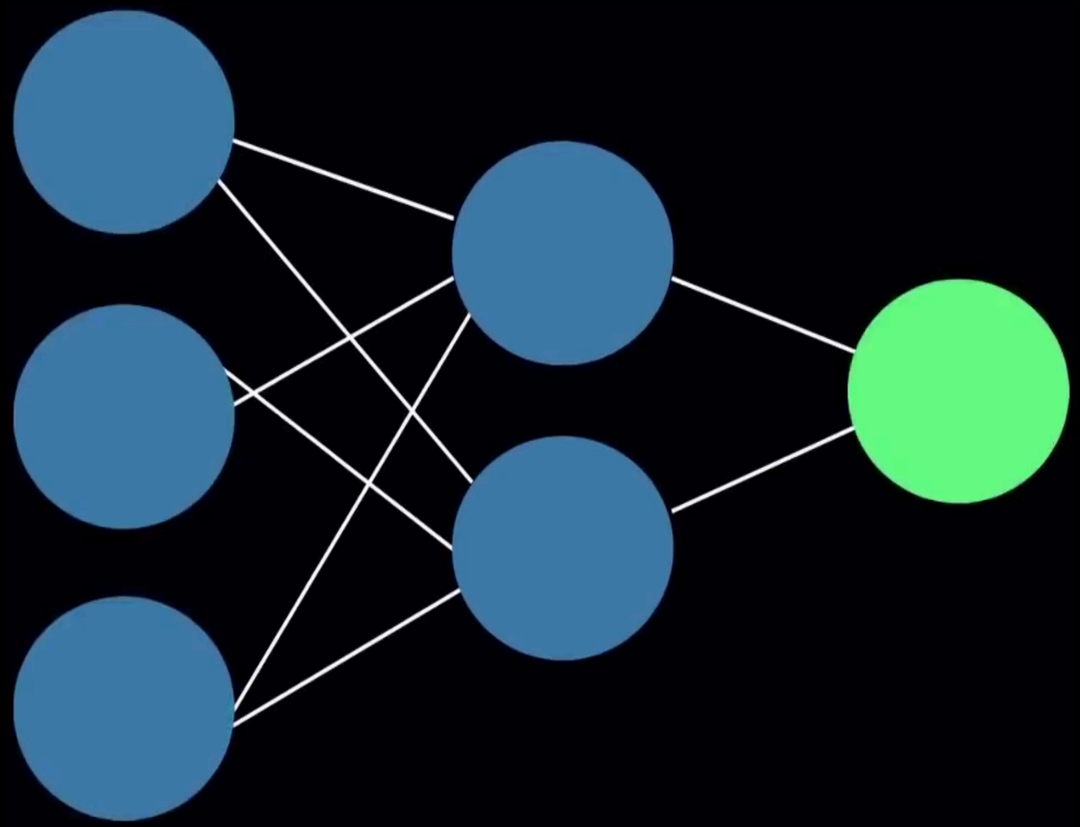
Les Réseaux de Neurones



Les Réseaux de Neurones

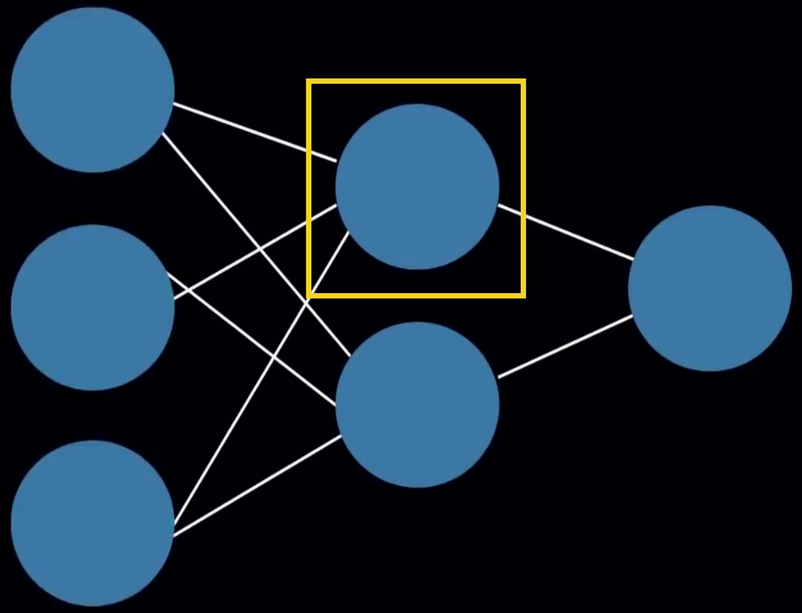


Les Réseaux de Neurones

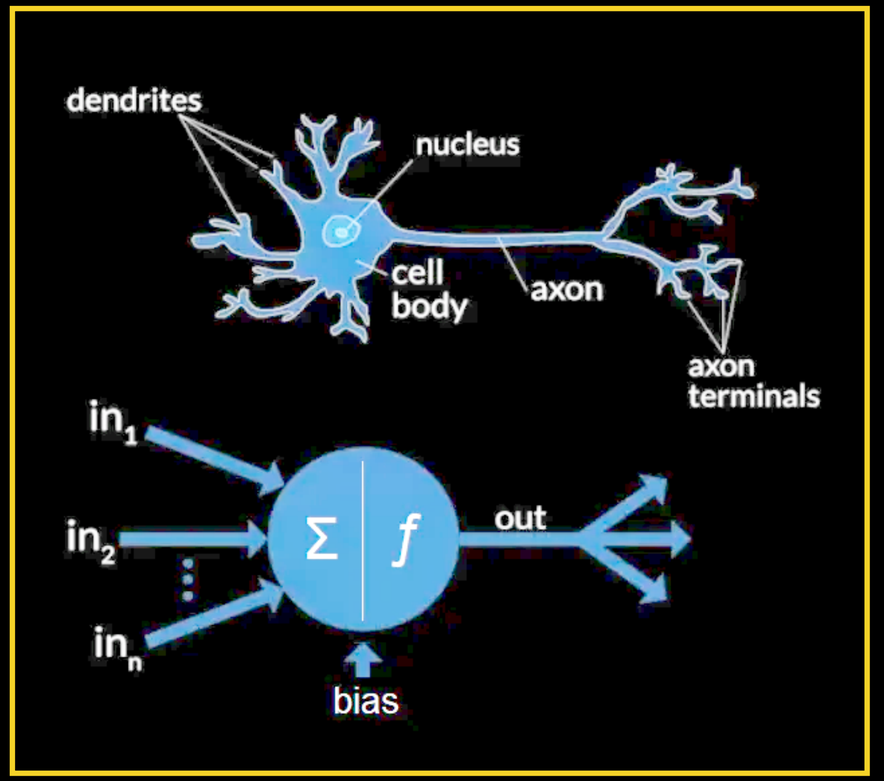


Couche
de sortie

Les Réseaux de Neurones

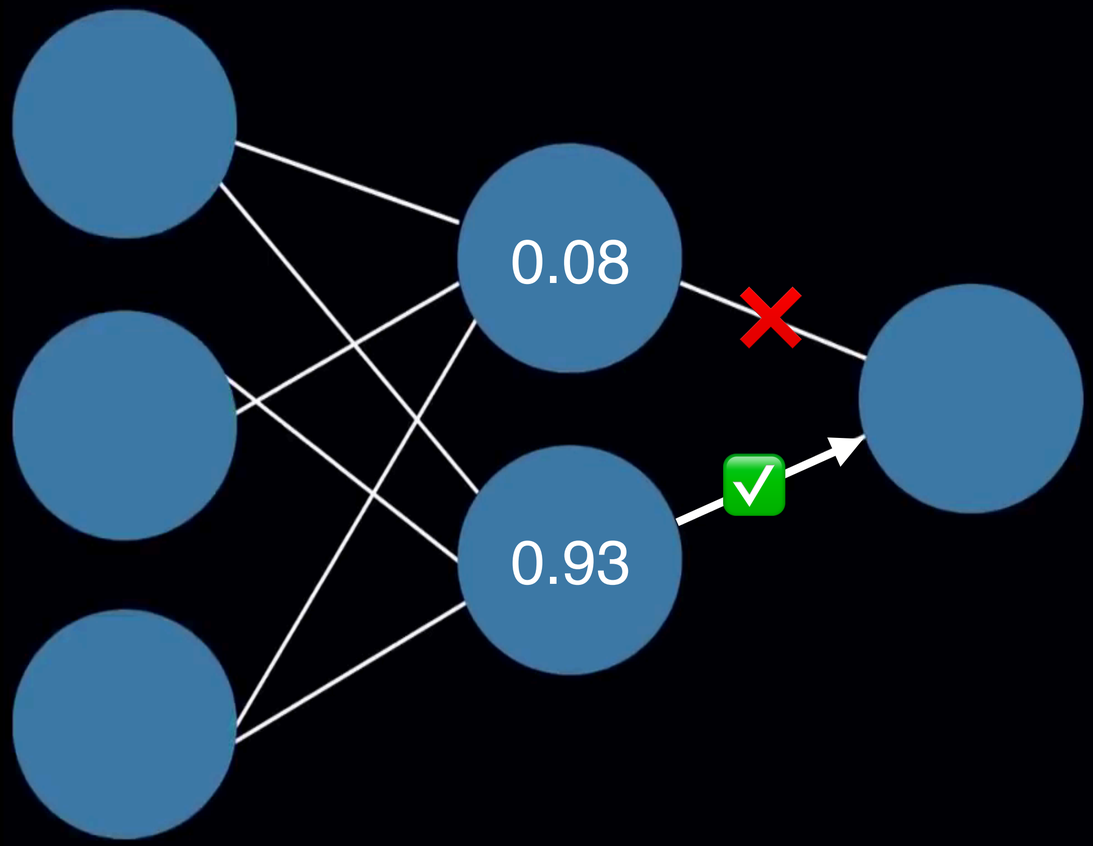


Inspiré par les neurones du cerveau



source: https://idiotdeveloper.com/what-is-deep-learning/

Les Réseaux de Neurones



Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

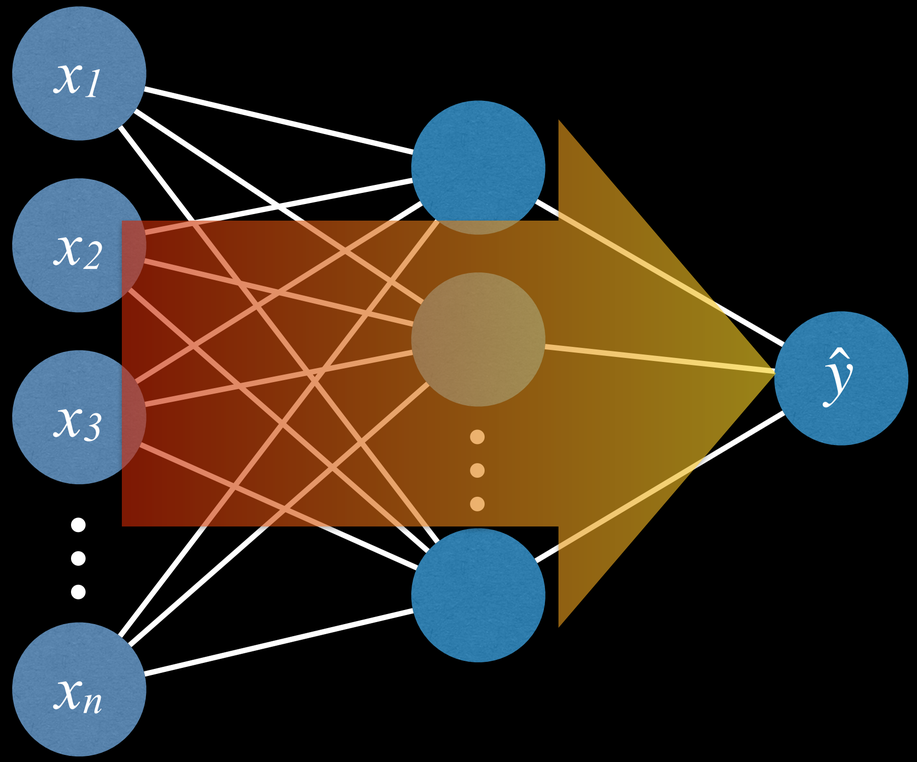
Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
  - Processus d’apprentissage: Propagation avant/arrière
  - Optimiseurs
  - Fonctions d’Activation
  - Fonctions de Perte (*Loss function*)
  - Paramètres et Hyperparamètres
  - *Epochs*, *Batch size* & Itérations
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Processus d'apprentissage

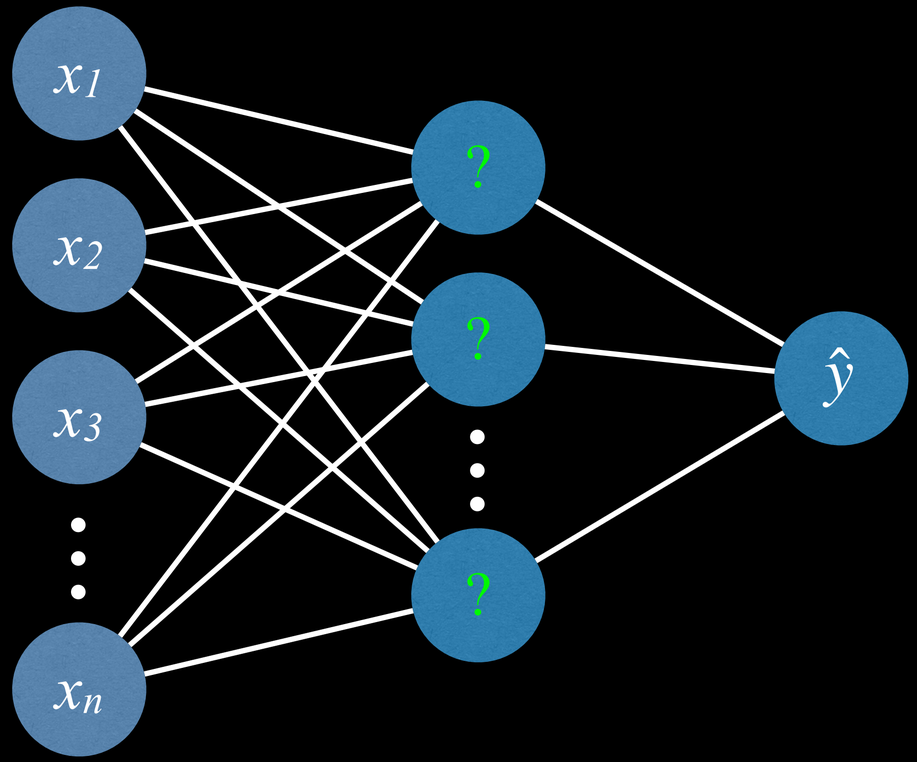
Processus d'apprentissage

1- Propagation vers l'Avant



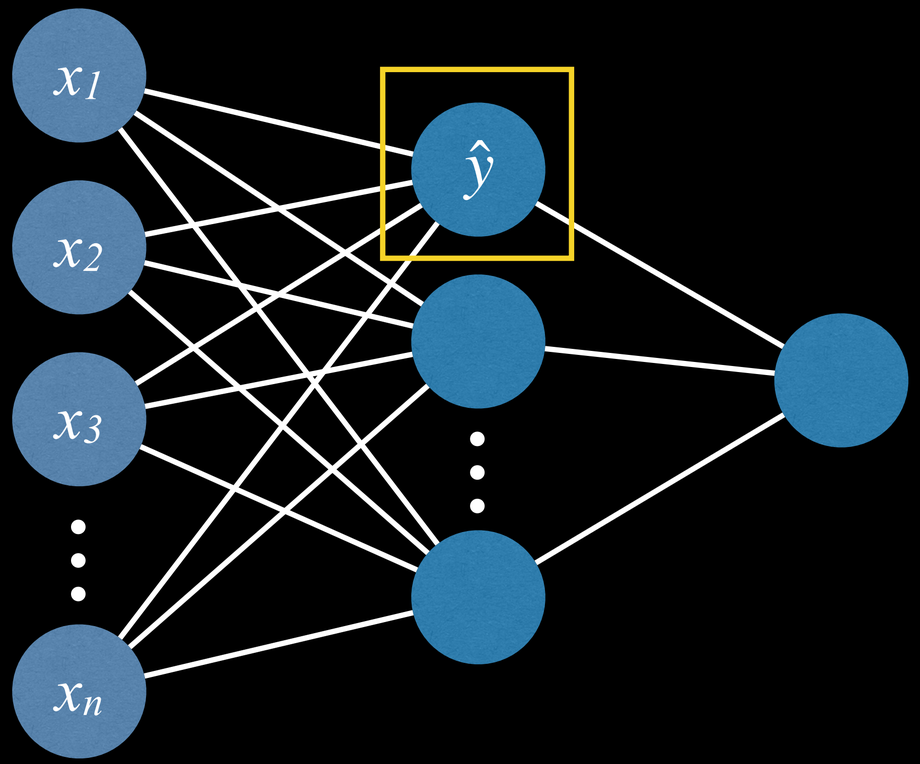
Processus d'apprentissage

1- Propagation vers l'Avant



Processus d'apprentissage

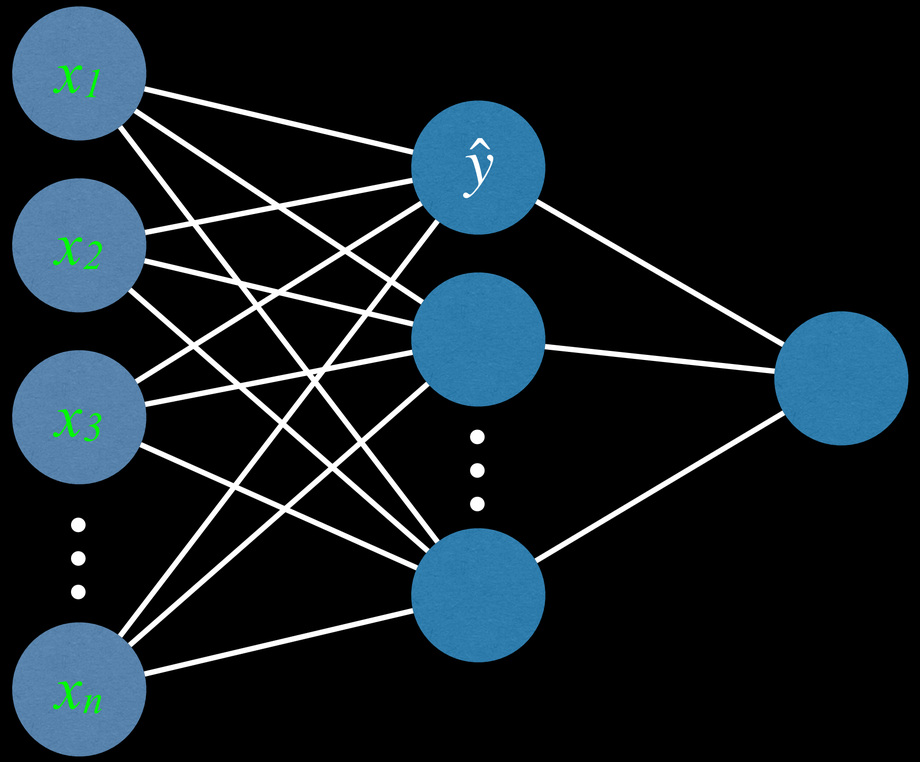
1- Propagation vers l'Avant



$$\hat{y} = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right)$$

Processus d'apprentissage

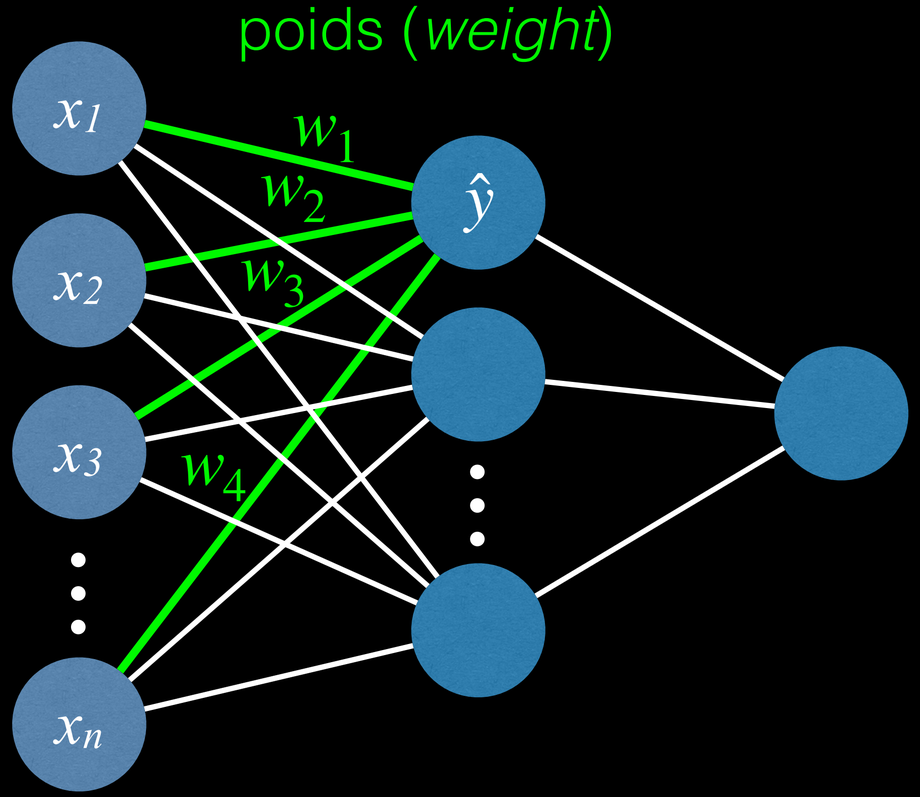
1- Propagation vers l'Avant



$$\hat{y} = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right)$$

Processus d'apprentissage

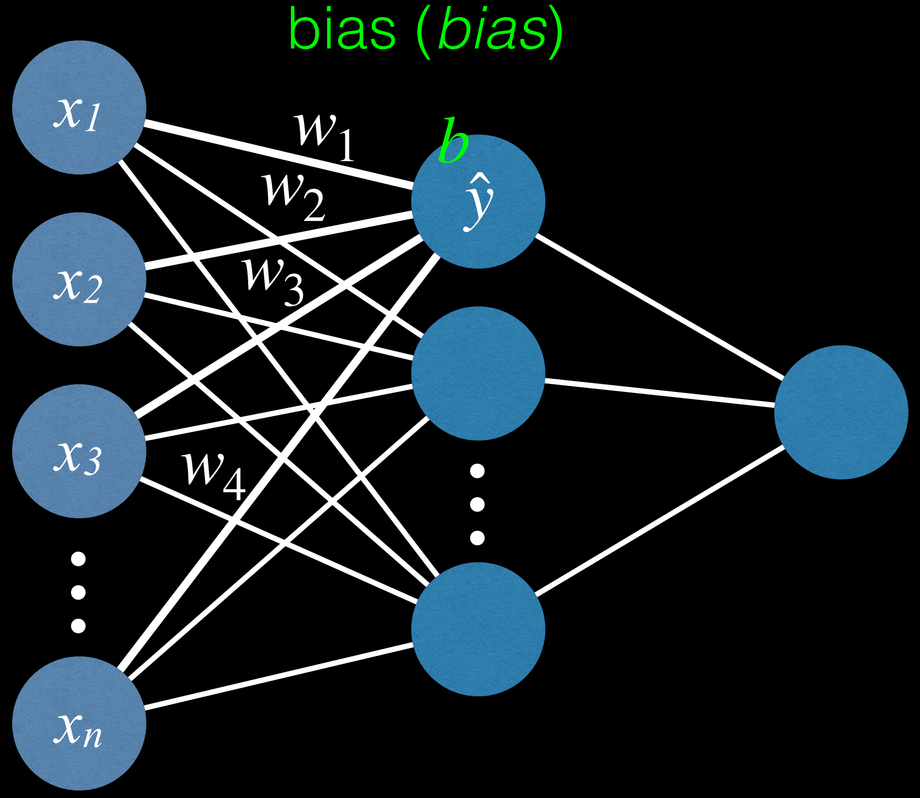
1- Propagation vers l'Avant



$$\hat{y} = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right)$$

Processus d'apprentissage

1- Propagation vers l'Avant

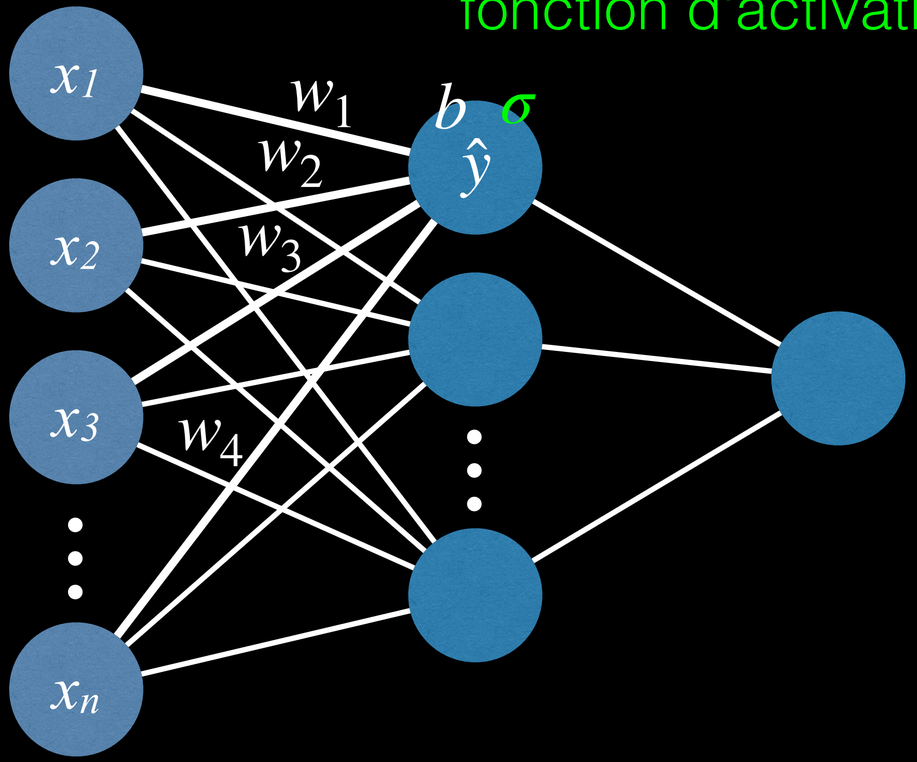


$$\hat{y} = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right)$$

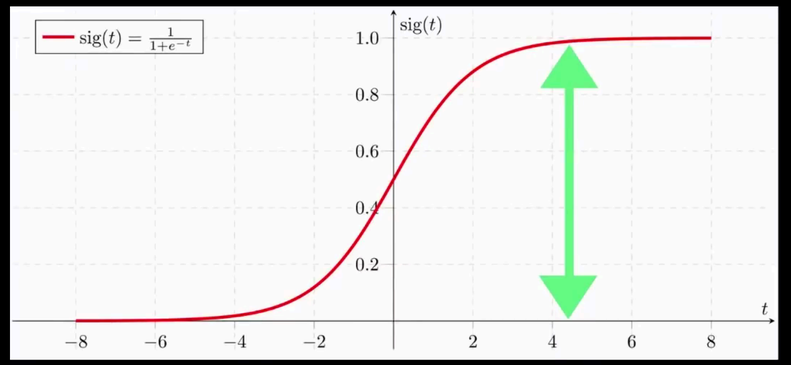
Processus d'apprentissage

1- Propagation vers l'Avant

fonction d'activation



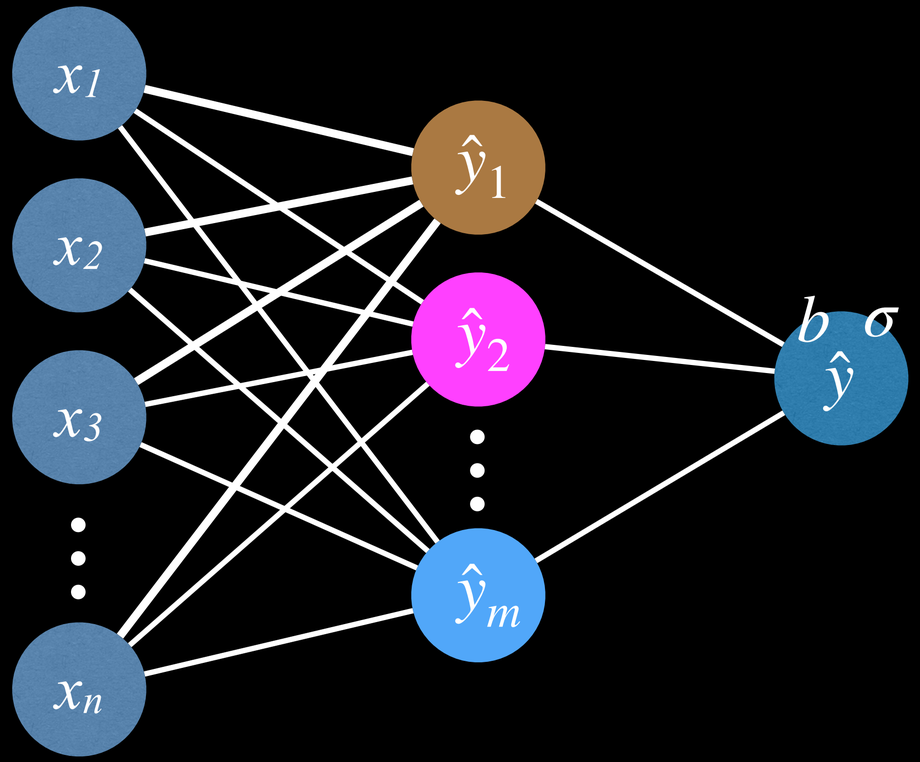
$$\hat{y} = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right)$$



sig(t) ∈ (0,1)

Processus d'apprentissage

1- Propagation vers l'Avant



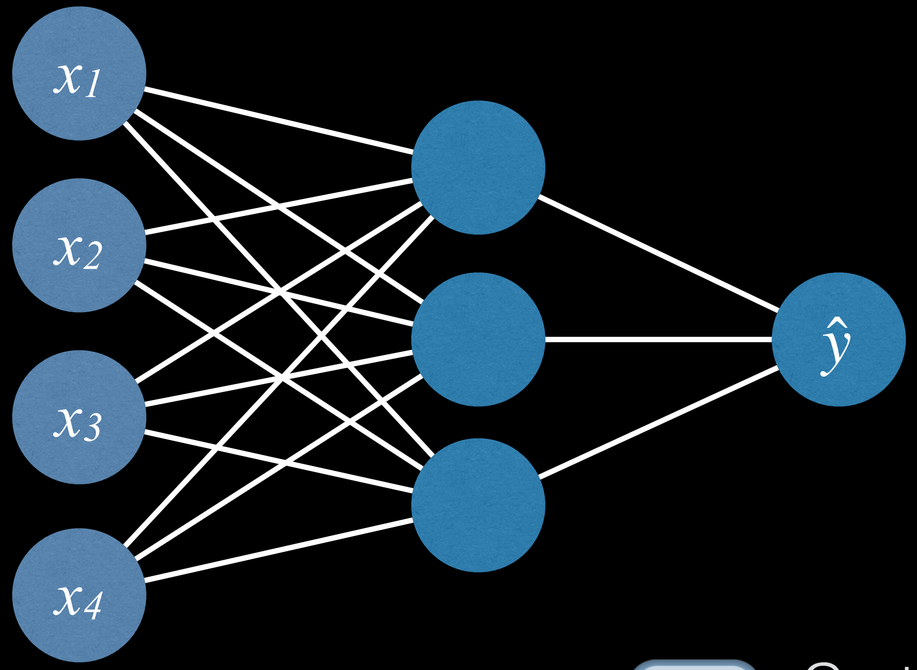
$$\hat{y}_1 = \sigma\left(\sum_{i=1}^{n} x_i w_i + b_1\right)$$

$$\hat{y}_2 = \sigma\left(\sum_{i=1}^{n} x_i w_i + b_2\right)$$

$$\hat{y}_m = \sigma\left(\sum_{i=1}^{n} x_i w_i + b_m\right)$$

Processus d'apprentissage

Nombre de paramètres d'un modèle



Nombre de poids: 4*3 + 3

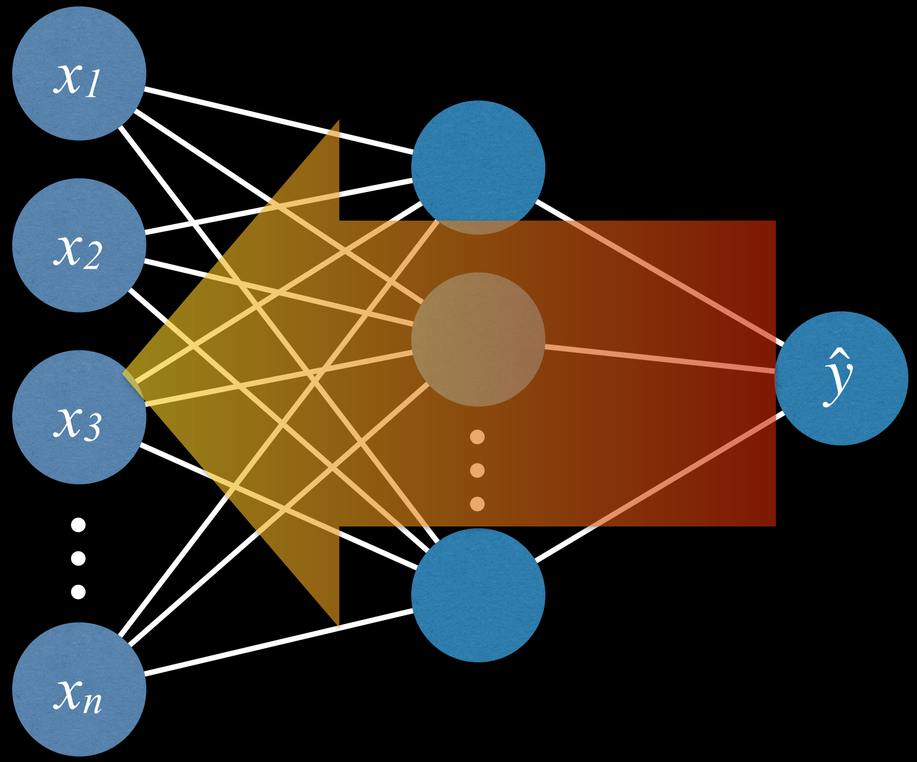
Nombre de bias: 3 + 1

Nombre de paramètres: 19

Certains modèles font plusieurs centaines de milliards de paramètres (eg: GPT4)

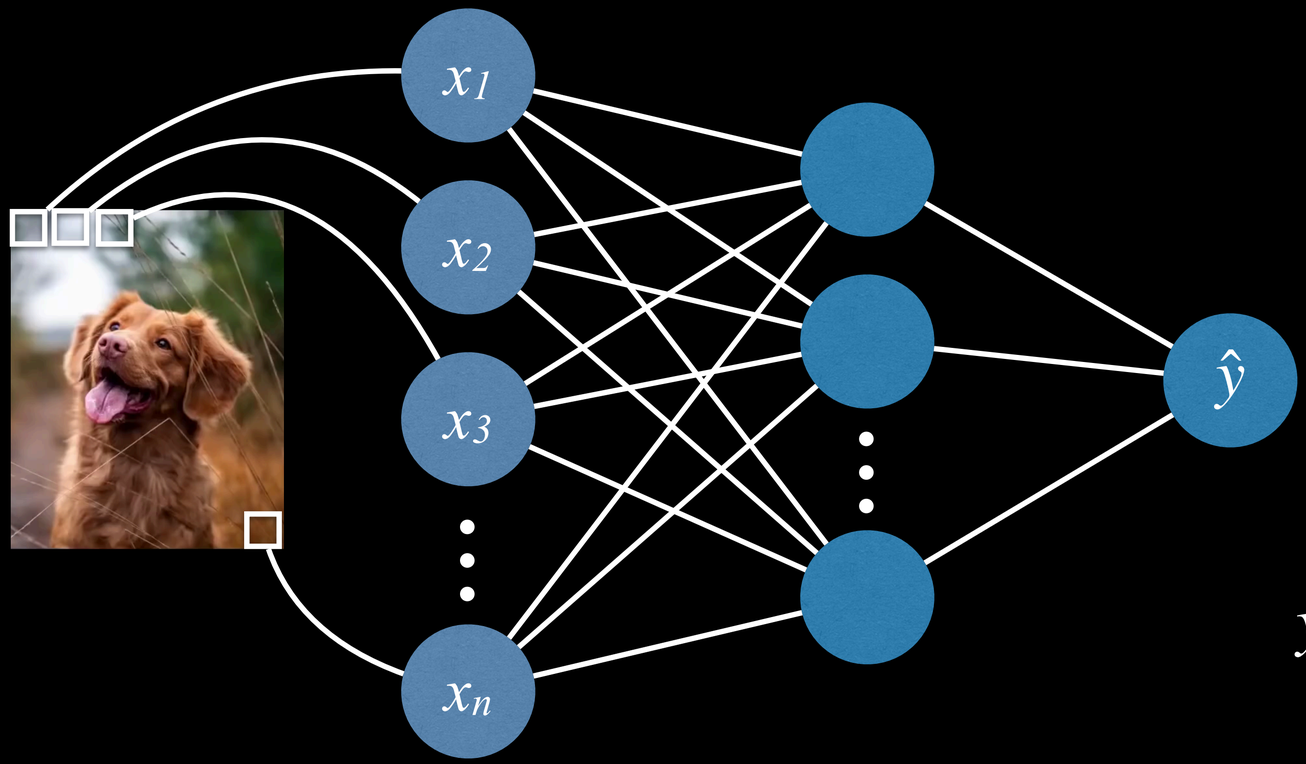
Processus d'apprentissage

2- Propagation vers l'arrière (*back propagation*)



Processus d'apprentissage

2- Propagation vers l'arrière (*back propagation*)



0 1

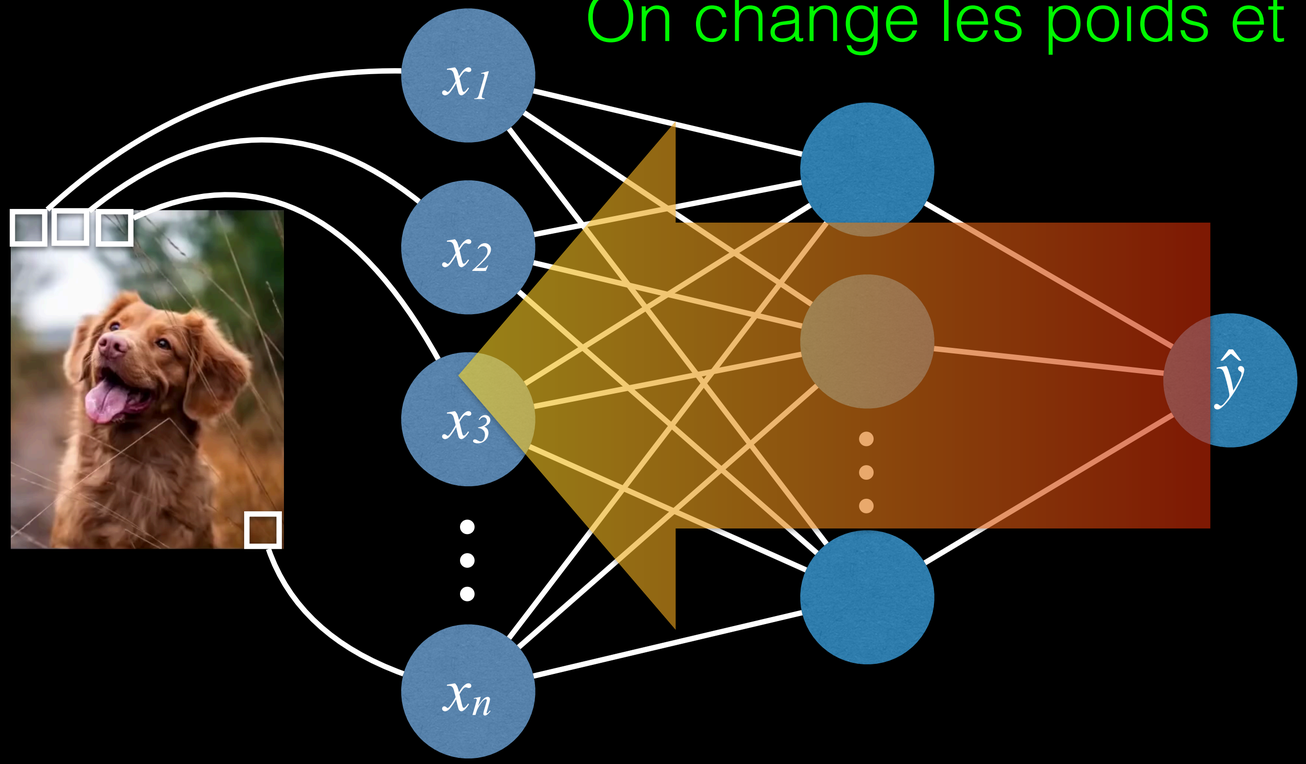
$$\left.\begin{aligned}\hat{y} &= 0.79\\ y_{GT} &= 1.0\end{aligned}\right\} \quad C = (0.79 - 1)^2 = 0.044...$$

Fonction de coût

Processus d'apprentissage

2- Propagation vers l'arrière (*back propagation*)

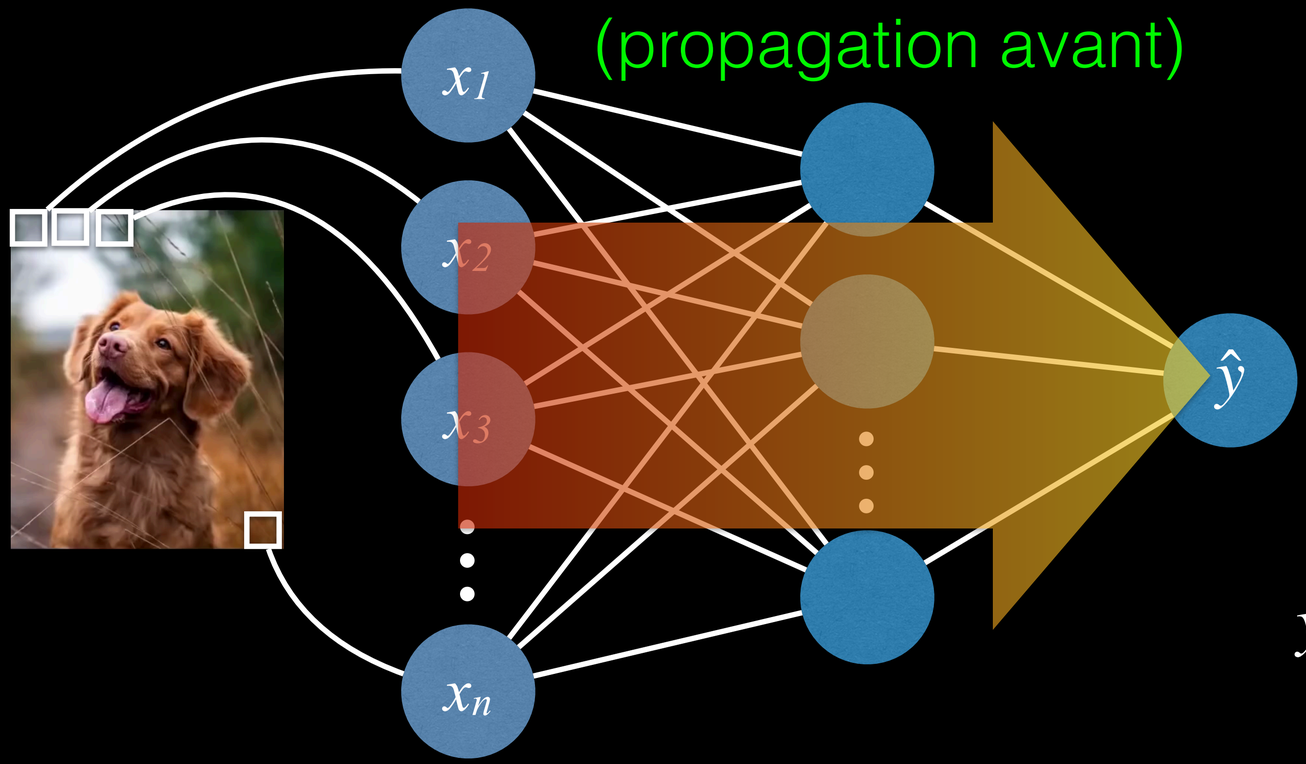
On change les poids et biais



Processus d'apprentissage

2- Propagation vers l'arrière (*back propagation*)

On recalcule la sortie
(propagation avant)

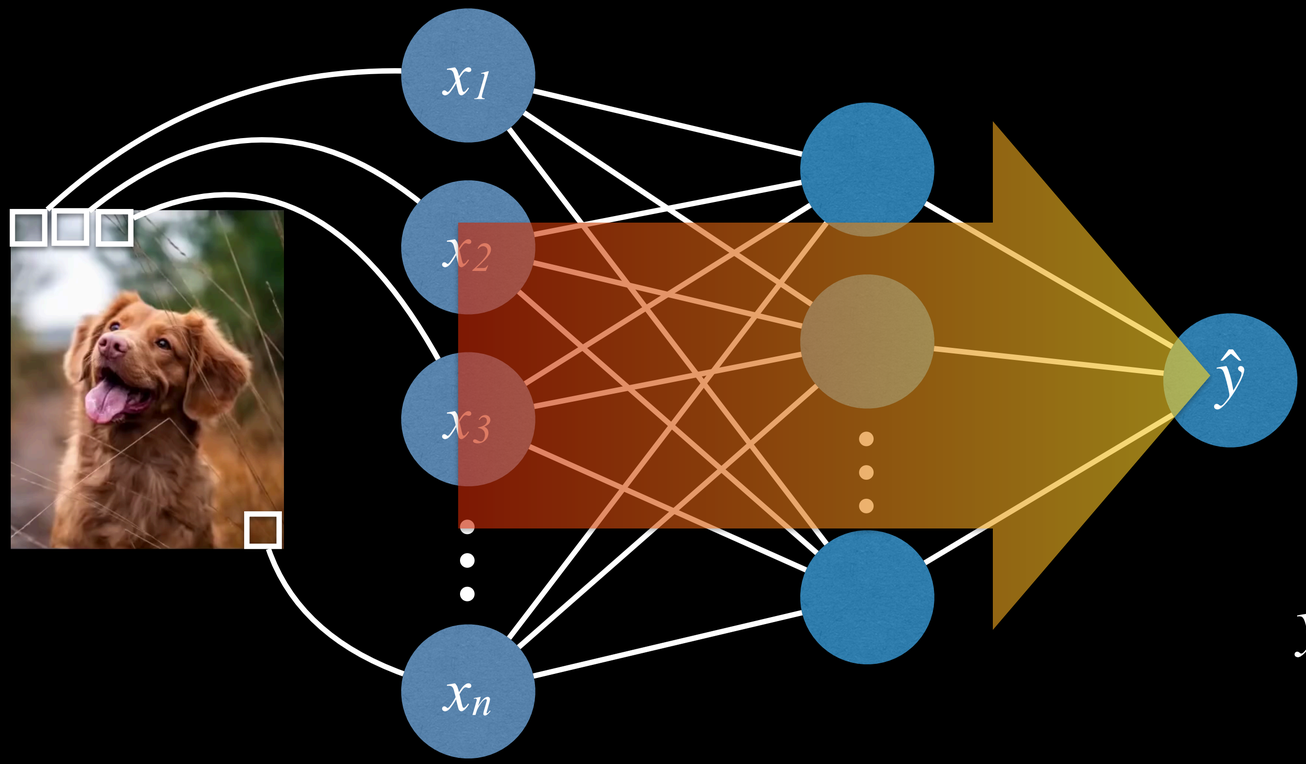


0 1

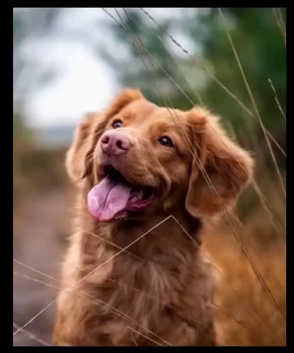
$$\left.\begin{aligned}\hat{y} &= 0.85\\ y_{GT} &= 1.0\end{aligned}\right\} \quad C = (0.85 - 1)^2 = 0.022...$$ 👍

Processus d'apprentissage

2- Propagation vers l'arrière (*back propagation*)



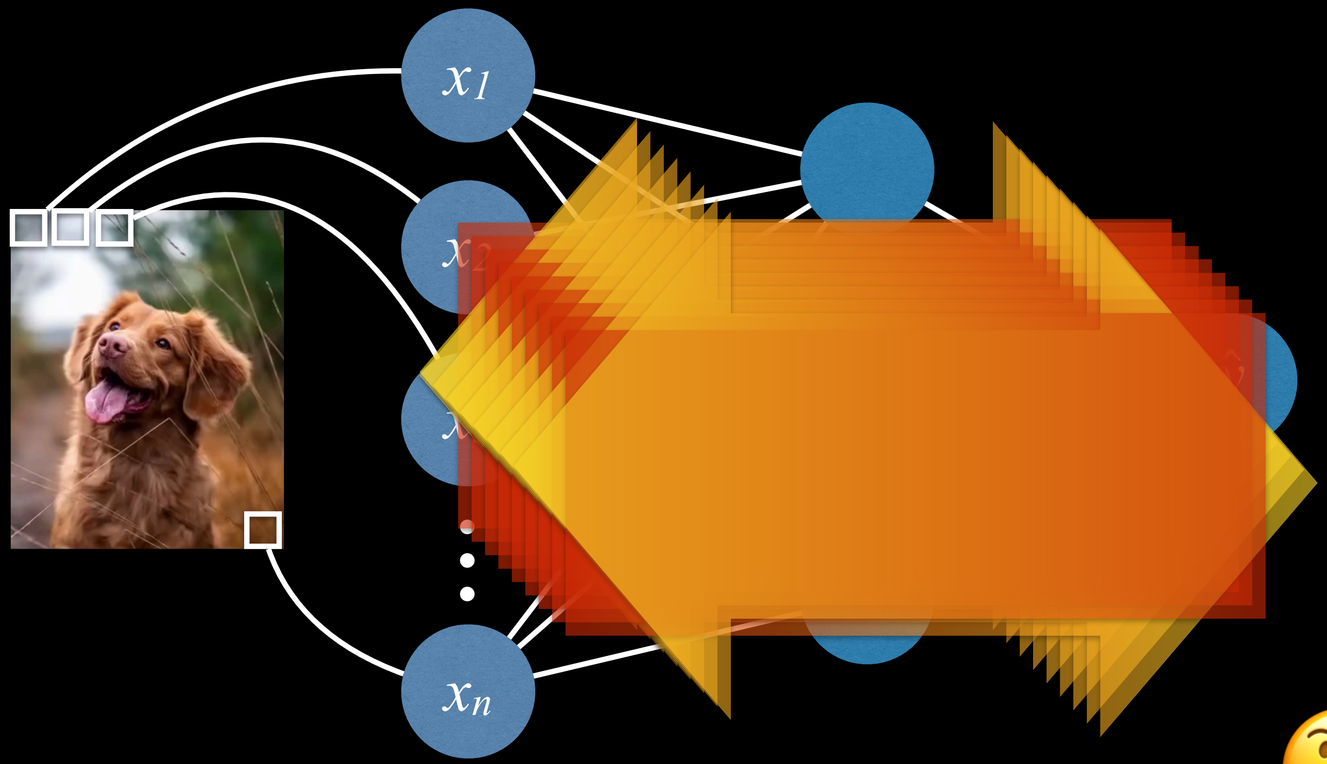
0 1

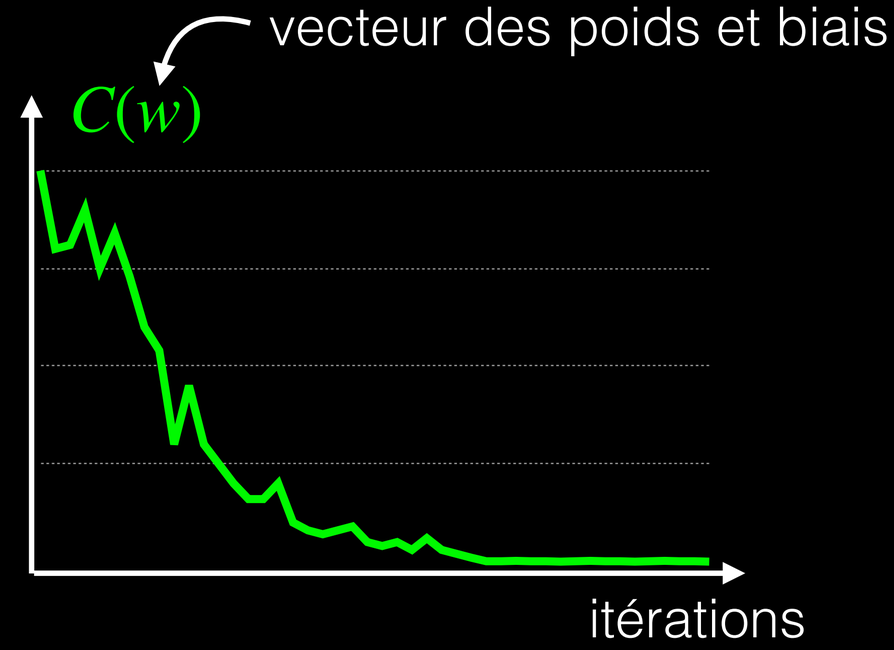


$$\left.\begin{aligned}\hat{y} &= 0.34\\ y_{GT} &= 1.0\end{aligned}\right\} \quad C = (0.34 - 1)^2 = 0.436... \text{ 👎}$$

Processus d'apprentissage

2- Propagation vers l'arrière (*back propagation*)





Comment les poids et biais sont changés ?
-> section "*Optimiseurs*"

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
  - Processus d’apprentissage: Propagation avant/arrière
  - Optimiseurs
  - Fonctions d’Activation
  - Fonctions de Perte (*Loss function*)
  - Paramètres et Hyperparamètres
  - *Epochs*, *Batch size* & Itérations
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Optimiseurs

Pendant l’entraînement, nous ajustons les paramètres afin de minimiser la fonction de perte et optimiser notre modèle le mieux possible

Mais, *COMMENT* fait-on cela ?

Optimiseurs

Mont Everest

Vous = Réseau de Neurones

La Descente = Optimiseur

Les Pieds = Fonctions de Perte

🤔 Comment formaliser ce problème mathématiquement ?

Optimiseurs

Descente de Gradient

Un algorithme itératif, qui démarre à un point aléatoire dans la fonction de perte, et qui diminue sa pente par étapes, jusqu'à ce qu'il atteigne le point le plus bas (minimum) de la fonction

- le Plus Populaire des Optimiseurs
- Rapide, Robuste, Flexible

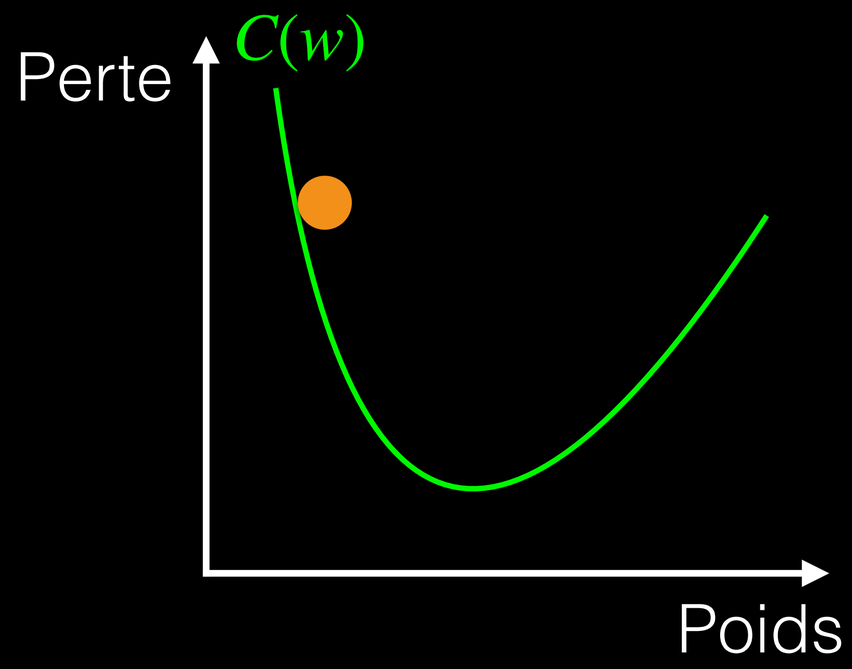
Optimiseurs

Descente de Gradient : Algorithme

- Calculer ce qu'un petit changement, au niveau de chaque poids individuel, ferait sur la fonction de perte
- Ajuster chaque poids individuel en fonction de son gradient (c.à.d, faire un petit pas dans la direction déterminée)
- Répéter les étapes 1 et 2 jusqu'à ce que la fonction de perte soit aussi faible que possible

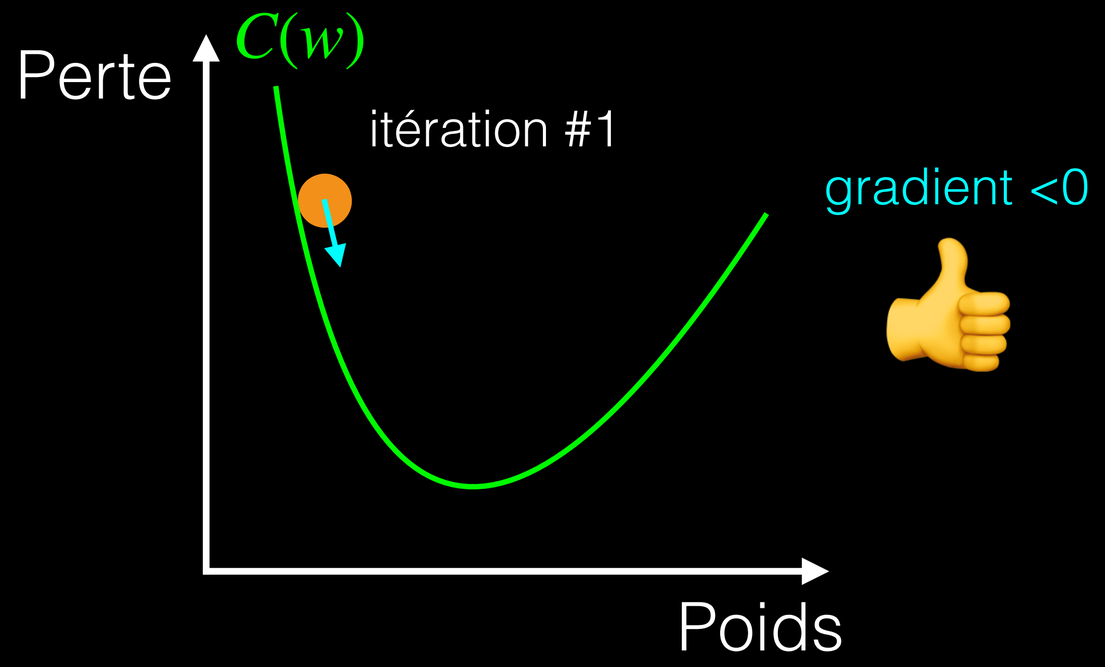
Optimiseurs

Descente de Gradient : Exemple à une dimension



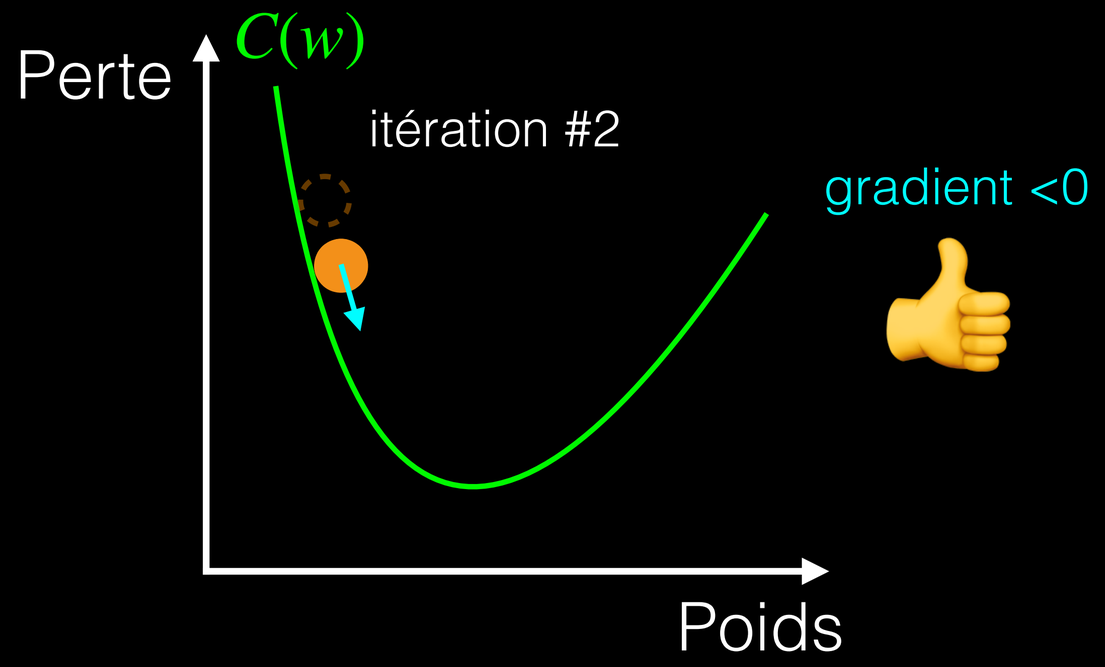
Optimiseurs

Descente de Gradient : Exemple à une dimension



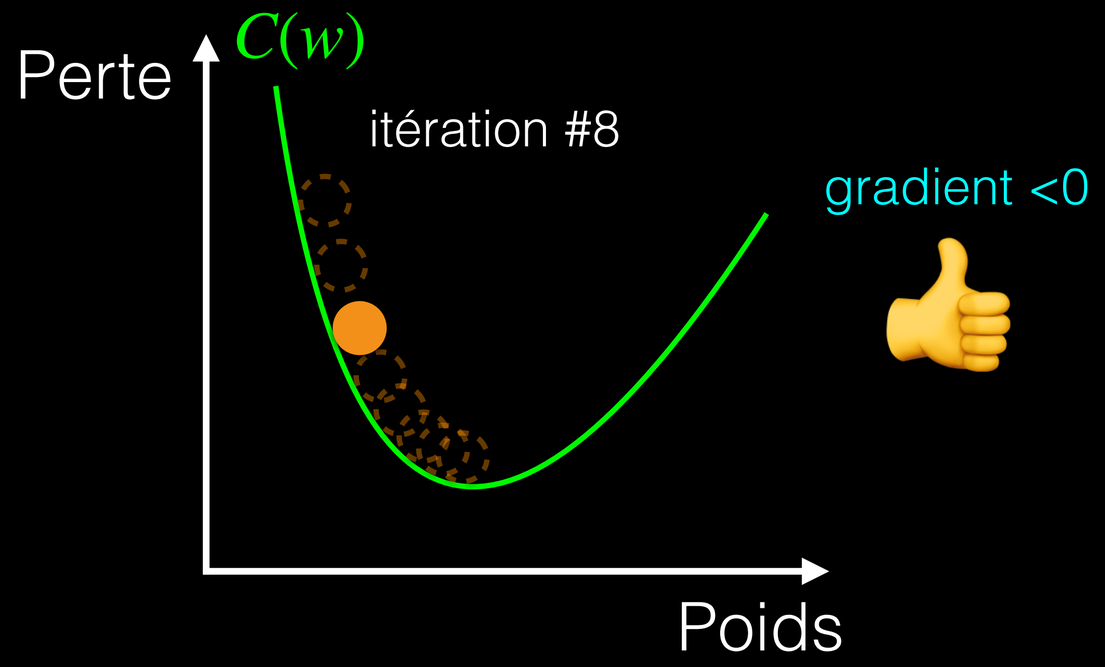
Optimiseurs

Descente de Gradient : Exemple à une dimension



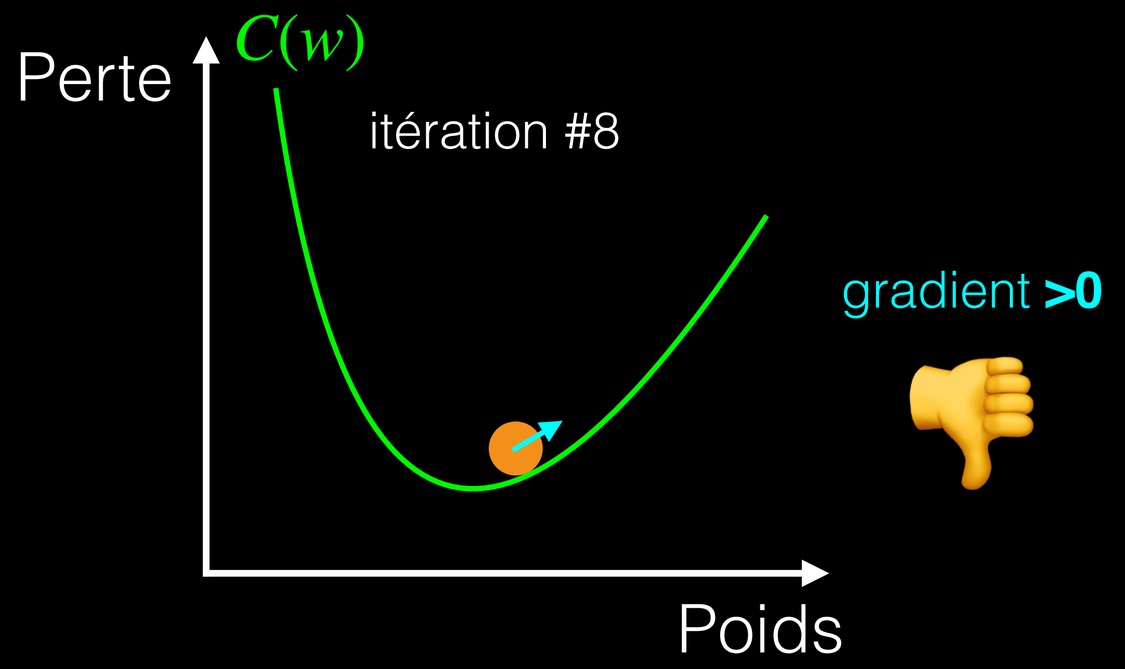
Optimiseurs

Descente de Gradient : Exemple à une dimension



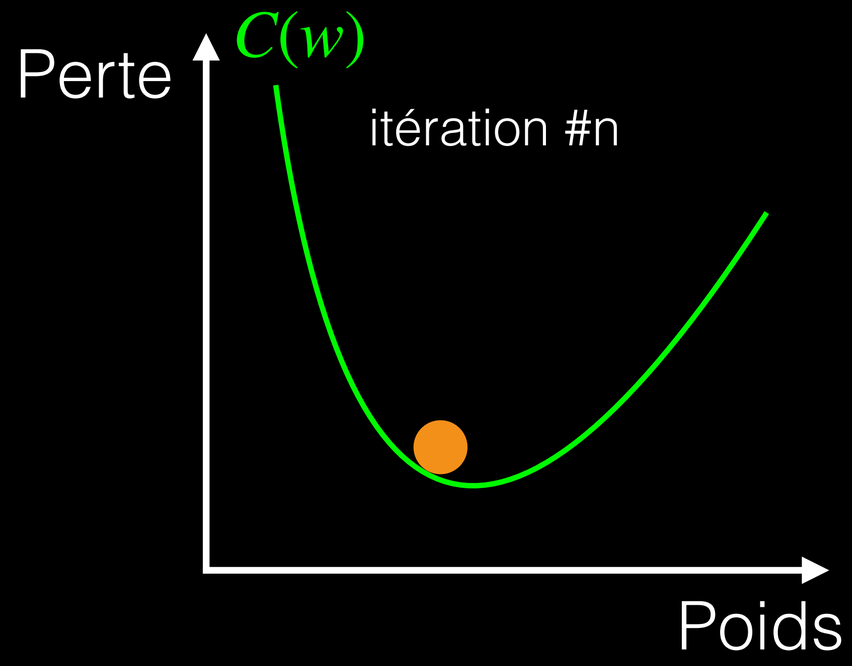
Optimiseurs

Descente de Gradient : Exemple à une dimension



Optimiseurs

Descente de Gradient : Exemple à une dimension



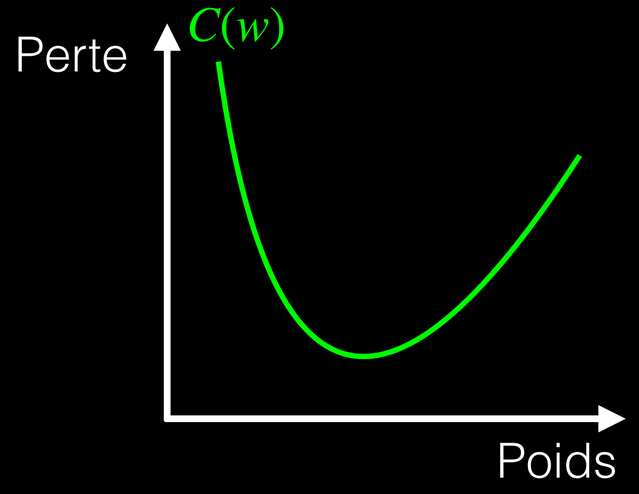
Condition d'arrêt :

$\delta[C(w)] < seuil$

Optimiseurs

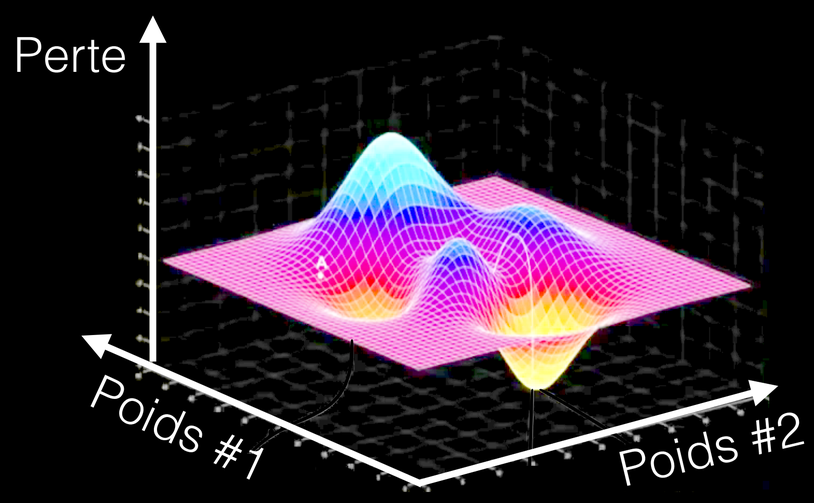
Descente de Gradient : Passage à dimension *n*

1 dimension



$$\nabla C(x) = \left(\frac{dC}{dx} \right)$$

2 dimensions



$$\nabla C(x_1, x_2) = \left(\frac{\partial C}{\partial x_1}, \frac{\partial C}{\partial x_2} \right)$$

n dimensions

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}$$

$$\nabla C(X) = \left(\frac{\partial C}{\partial X} \right)$$

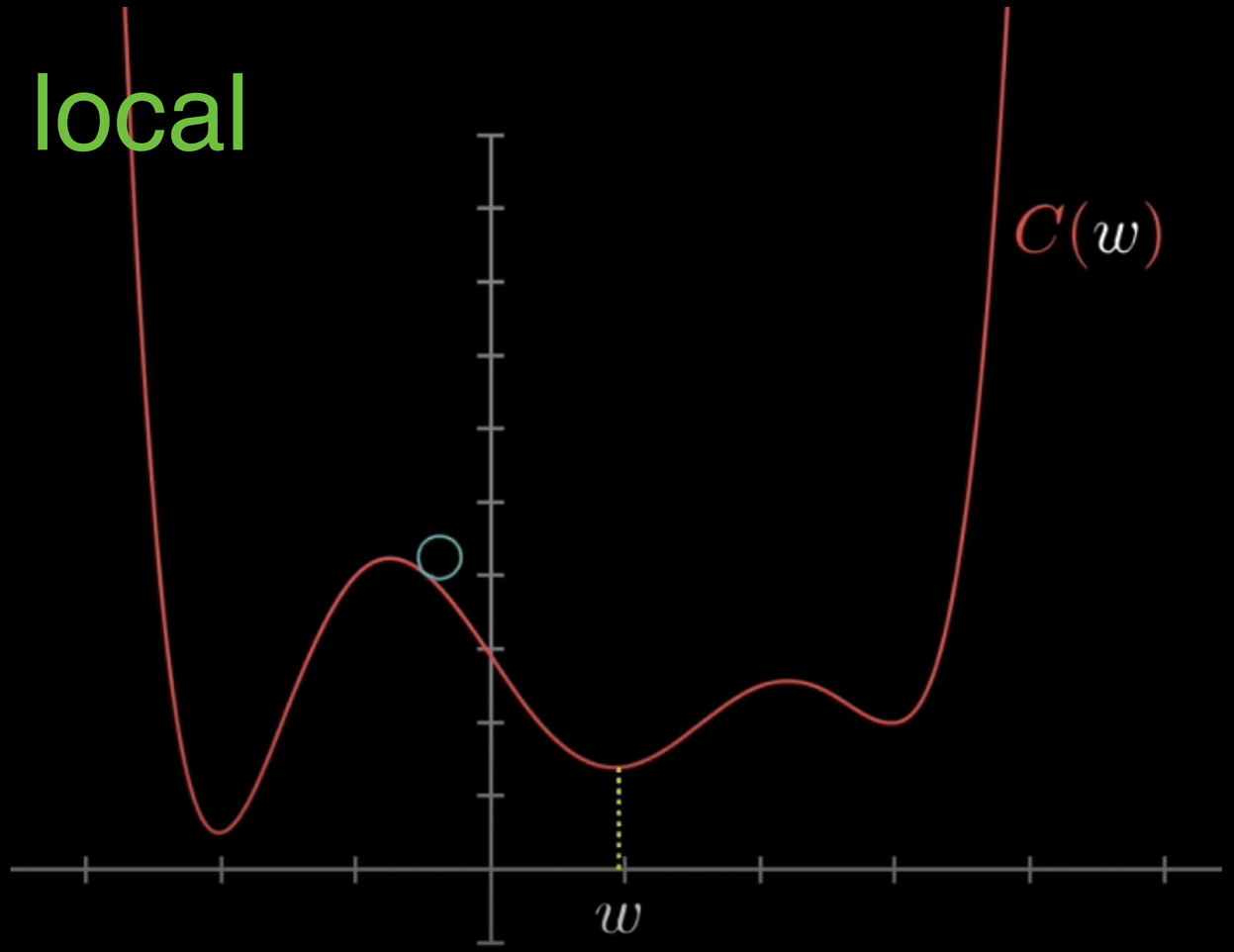
Optimiseurs

Quand calculer le gradient et changer les paramètres ?

- **Descente de gradient par batch (*Batch Gradient Descent*)**
  - Paramètres du modèle mis à jour après avoir passé **toutes** les données d'entraînement (un *epoch*).
  - ✅ Estimation précise des gradients
  - ❌ Coûteuse en calcul pour les grandes données (gradient calculé pour **chaque** échantillon, puis moyenné).
- **Descente de gradient stochastique (*Stochastic Gradient Descent* - SGD)**
  - Paramètres mis à jour après calcul du gradient sur **un seul** échantillon pris aléatoirement.
  - ✅ Peu coûteux en calcul
  - ✅ Converge plus rapidement, aide à échapper aux minima locaux.
  - ❌ Sensible au bruit
- **Descente de gradient par mini-batch (*Mini-batch Gradient Descent*)**
  - Paramètres du modèle sont mis à jour après avoir traité **plusieurs** échantillons pris aléatoirement.
  - Combine les avantages de la descente de gradient par batch et stochastique.

Optimiseurs

Minimum local



source: https://www.youtube.com/watch?v=IHZwWFHWa-w

Optimiseurs

Taux d’Apprentissage

- Généralement un petit nombre, comme .001, qui est multiplié pour mettre à l’chelle les gradients
- Cela garantit que tout changement que l’on apporte à nos poids restent assez petits

Optimiseurs

Taux d’Apprentissage

Nous ne voulons pas d’un taux d’apprentissage élevé, lorsque l’algorithme dépassera le minimum global

De façon similaire, nous n’en voulons pas un trop petit, où l’algorithme mettra une éternité à converger vers le minimum global

Example interactif 👉 https://distill.pub/2017/momentum/

Optimiseurs

Adagrad

- Adapte le Taux d’Apprentissage pour chaque paramètre
- ✅ Pas besoin d'ajuster manuellement le taux d'apprentissage
- ✅ Convergence plus rapide et plus fiable que le SGD simple lorsque l'échelle des poids est inégale
- Il n'est pas très sensible à la taille de la "master step".

- Certains des poids auront des taux d’apprentissage différents

Optimiseurs

RMSprop

- Version spéciale d’Adagrad
- Accumulation de Gradients dans une fenêtre fixe
- Similaire à Adaprop

Optimiseurs

Adam

- Représente l’Estimation du Moment Adaptatif
- Utilise le concept de moment
- Assez répandu aujourd’hui

Optimiseurs

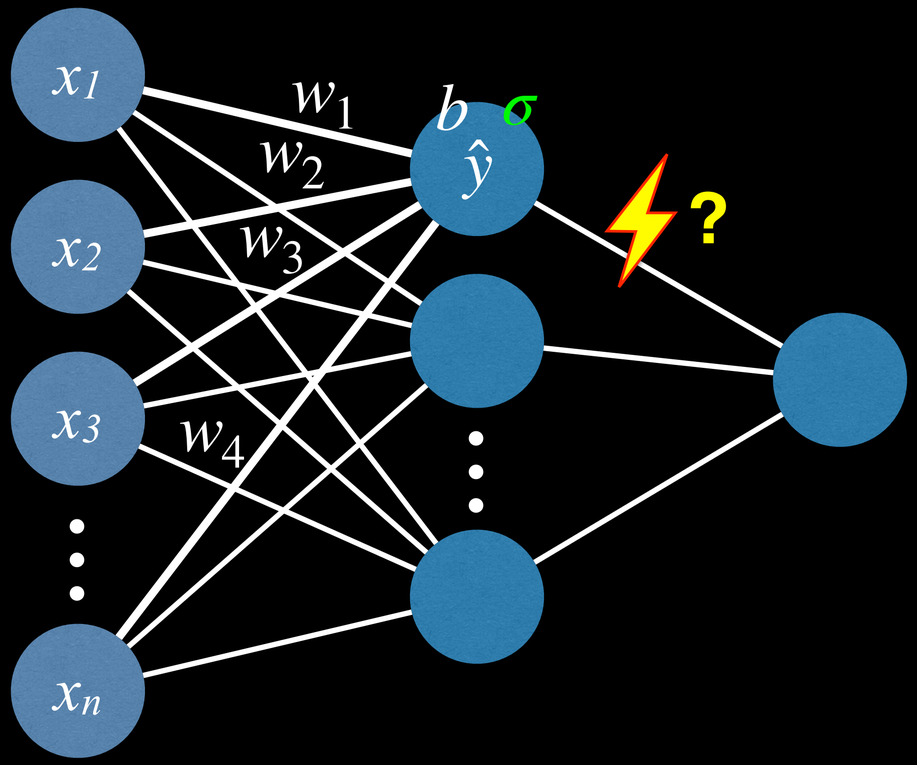
Quelques réflexions pour conclure

- Il est facile de se trouver perdu avec autant d'optimiseurs
- Tous ont le même objectif : la Minimisation de la fonction de perte
- Par essais et erreurs, on y parvient cependant !

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
  - Processus d’apprentissage: Propagation avant/arrière
  - Optimiseurs
  - Fonctions d’Activation
  - Fonctions de Perte (*Loss function*)
  - Paramètres et Hyperparamètres
  - *Epochs*, *Batch size* & Itérations
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Fonctions d'Activation



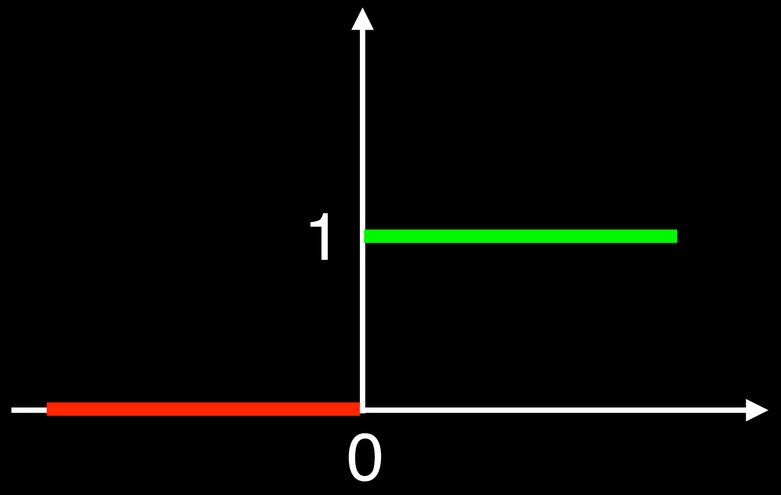
$$\hat{y} = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right)$$

- Introduisent une **non-linéarité** dans le Réseau de Neurones
- Décident de savoir si un neurone peut contribuer à la couche suivante ou non

🤔 *Mais comment décider si un neurone peut se déclencher/activer ou non ?*

Fonctions d'Activation

a. Fonction d'Etape (*step function*)



si valeur > 0 → $\hat{y} = 1$ (activé)

sinon → $\hat{y} = 0$ (pas activé)

Problème : Ne couvre pas **tous** les cas...

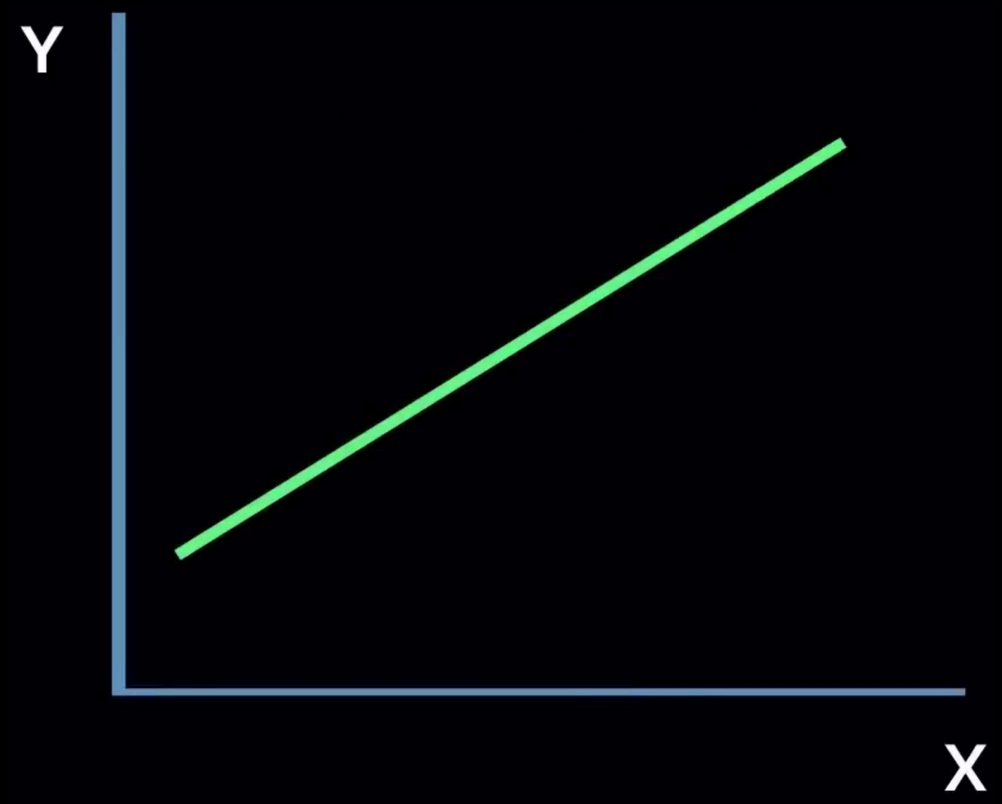
Fonctions d’Activation

a. Fonction d’Etape (*step function*)

...Exemple : Choisir entre 3 classes :

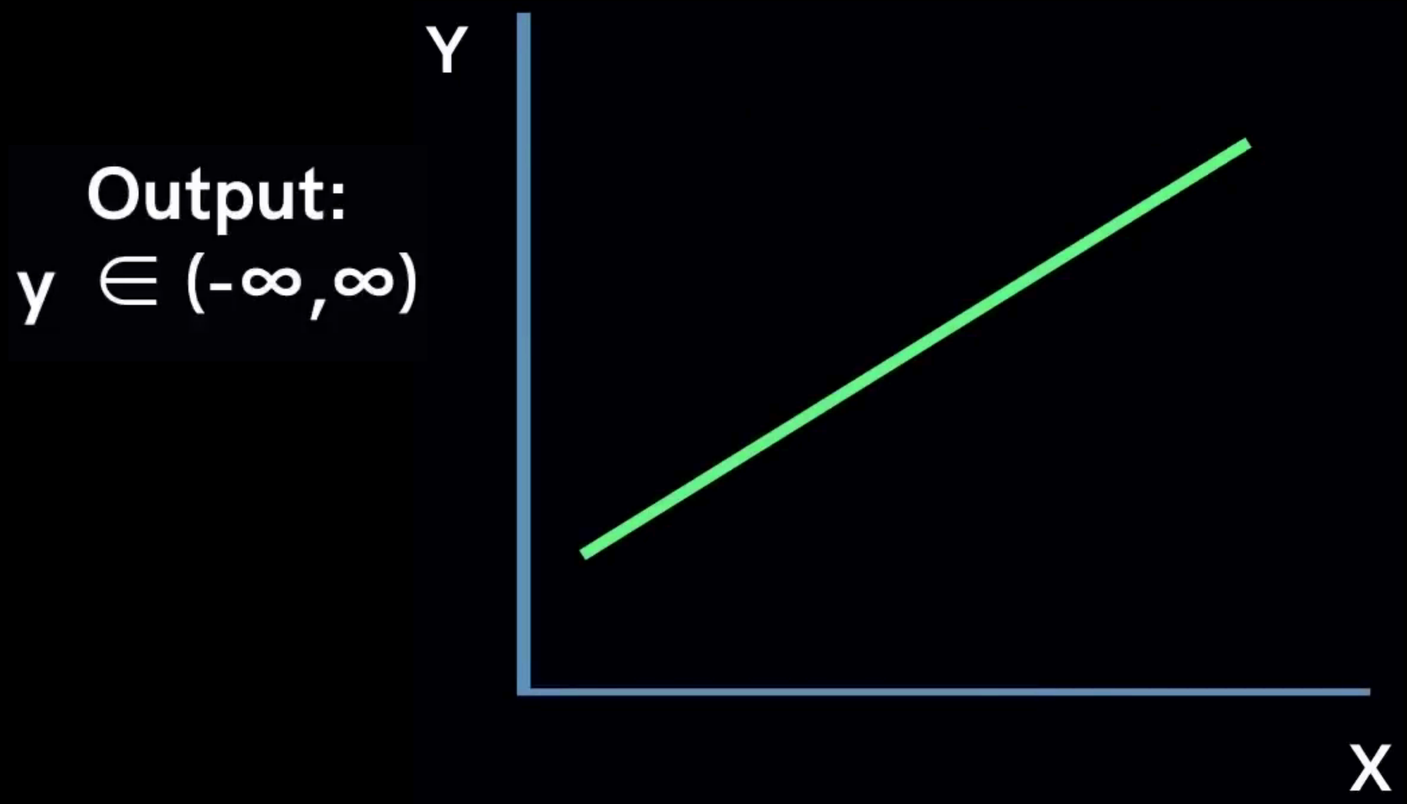
Fonctions d’Activation

b. Fonction Linéaire



Fonctions d’Activation

b. Fonction Linéaire



La Dérivée est une constante

$$y = mx + b$$

Dérivée, y’ = m

Fonctions d’Activation

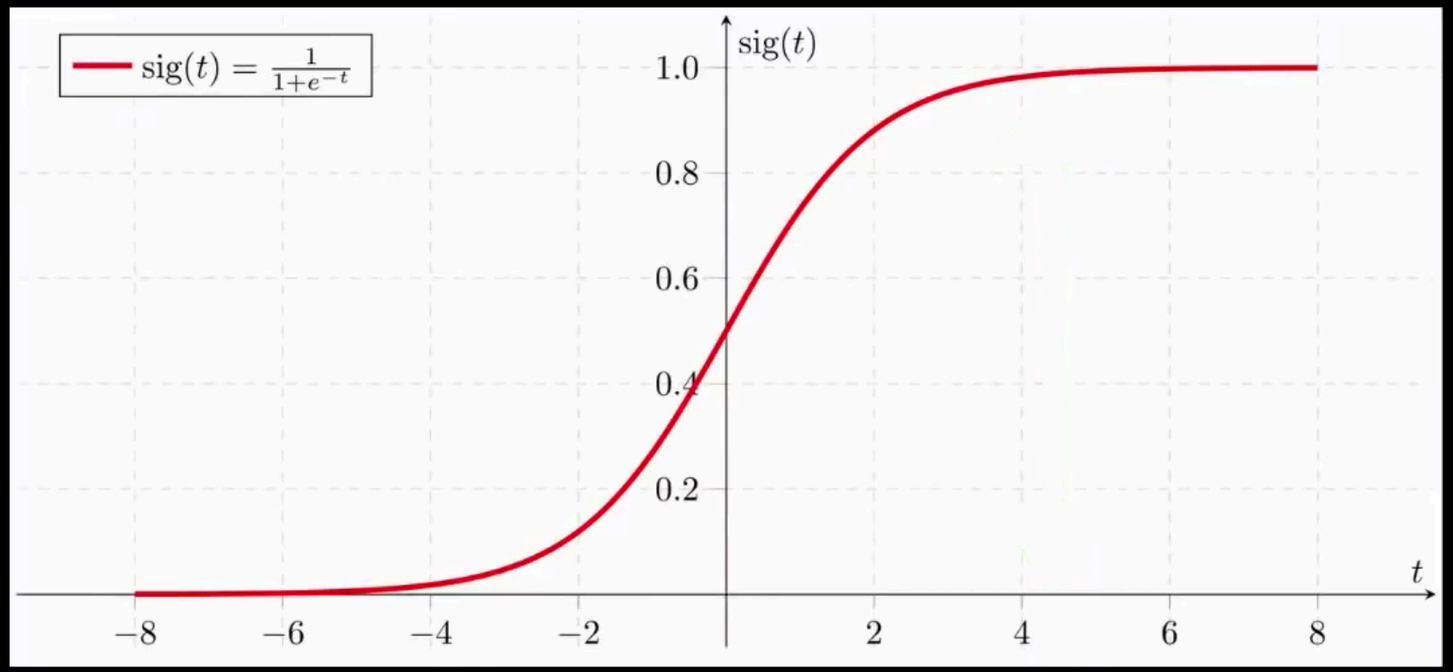
b. Fonction Linéaire

Si f(x) et g(x) sont linéaires

g(f(x)) et f(g(x)) sont linéaires

Fonctions d’Activation

c. Fonction Sigmoïde

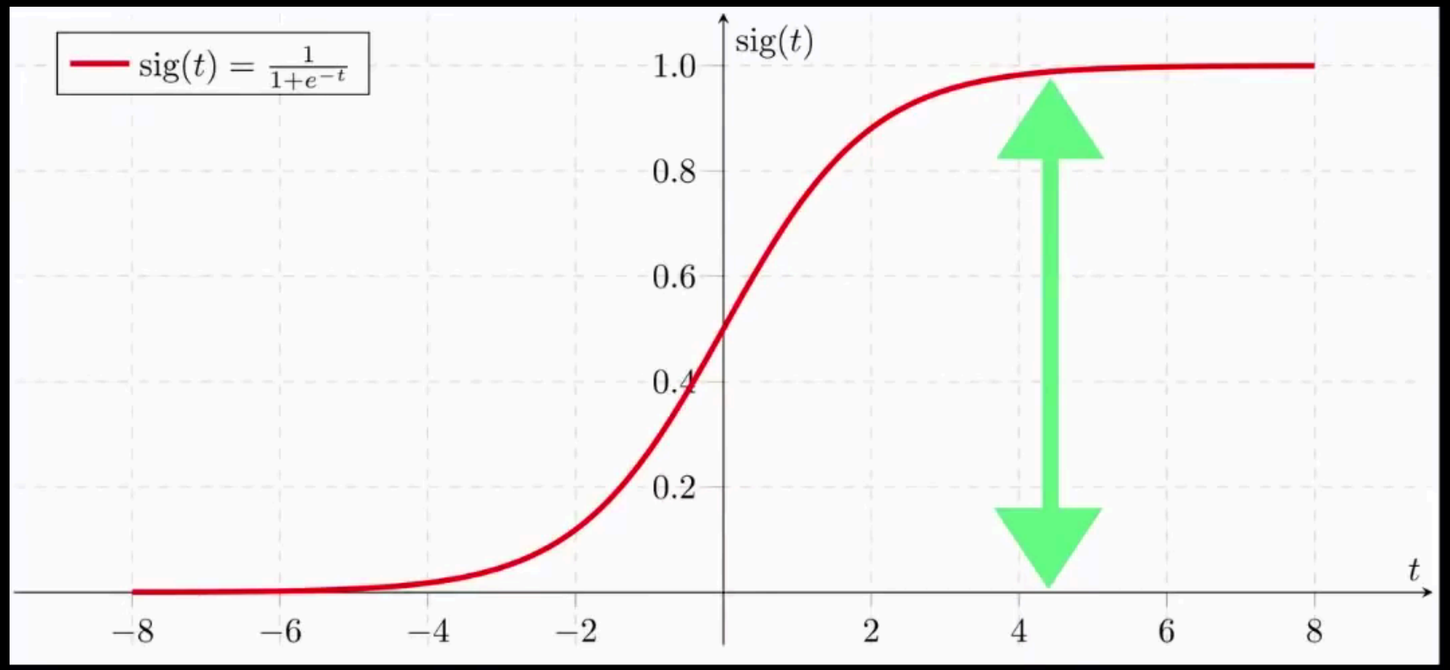


- Non-linéaire
- Sorties Analogiques

$\mathrm{sig}(t) \in (0,1)$

Fonctions d’Activation

c. Fonction Sigmoïde

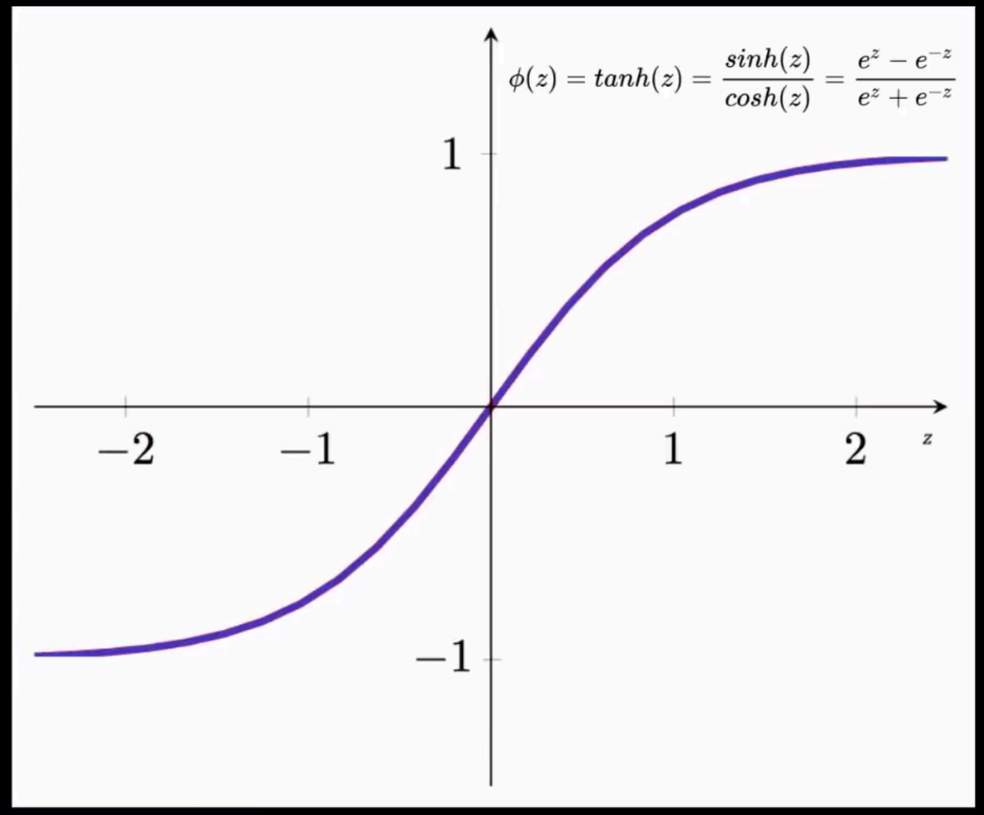


- Non-linéaire
- Sorties Analogiques
- Problème du Gradient de Disparition

sig(t) ∈ (0,1)

Fonctions d’Activation

d. Fonction Tangente Hyperbolique (tanh)



tanh(x) ∈ (-1,1)

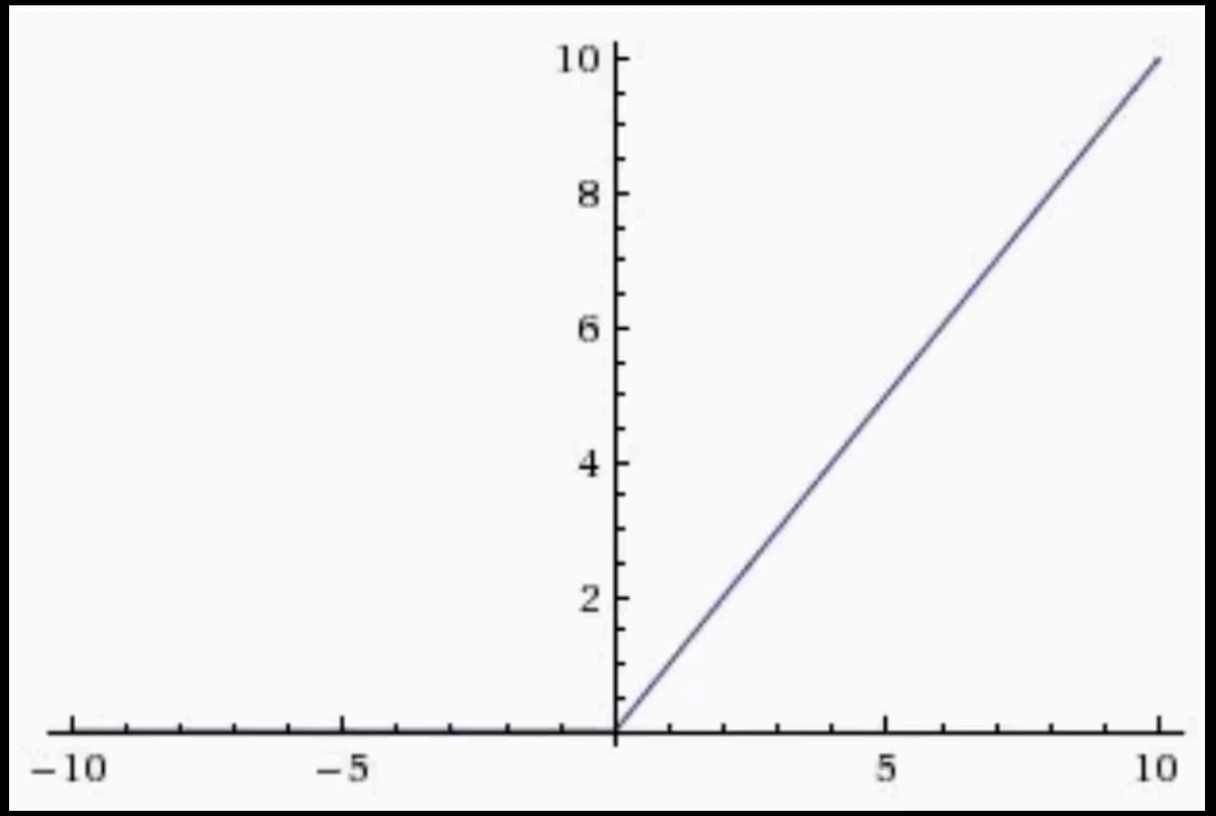
$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

$$\tanh(x) = 2\,\text{sigmoid}(2x) - 1$$

- Non-linéaire
- Dérivée plus raide que la Sigmoïde
- Problème du Gradient de Disparition

Fonctions d’Activation

e. Fonction ReLU (Unité Linéaire Rectifiée)



$R(z) \in [0,\infty)$

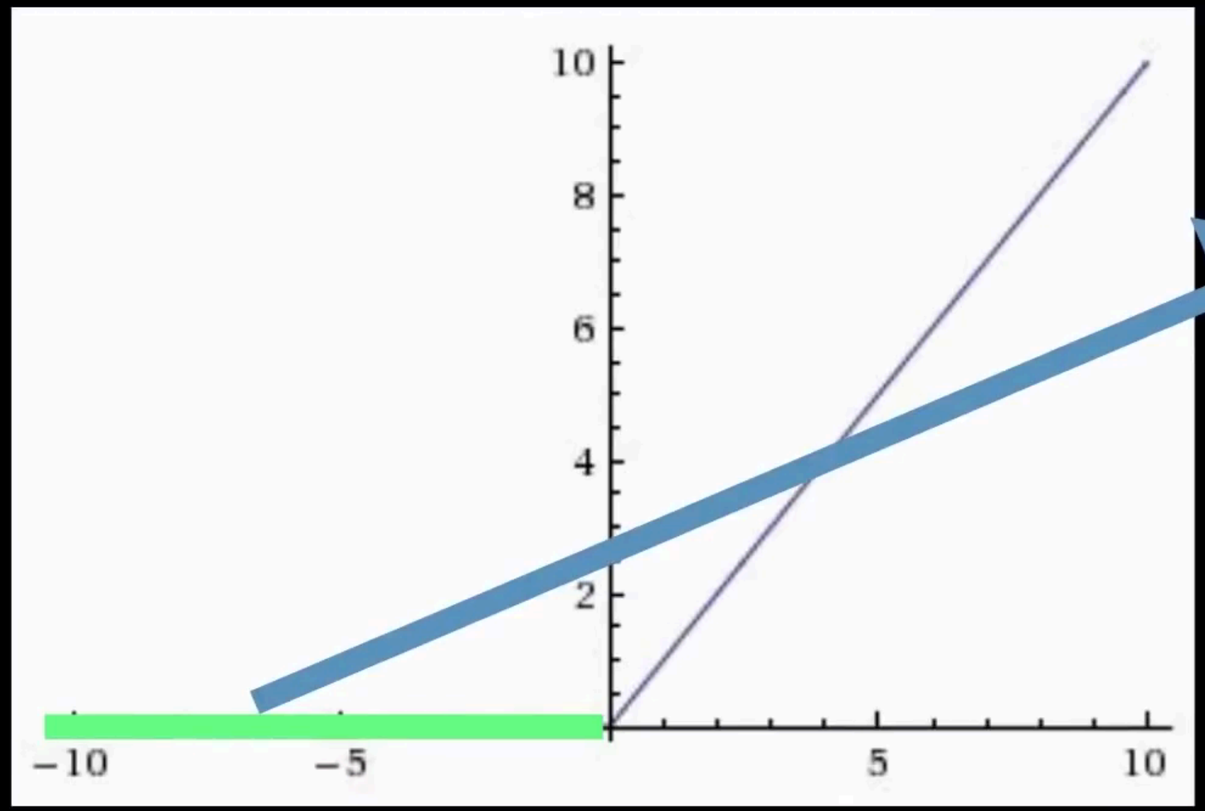
$$R(z) = \max(0,z)$$

- Non-linéaire !
- Activations clairsemées

🤔 *Mais que se passe-t-il pour des valeurs négatives ?*

Fonctions d’Activation

e. Fonction ReLU (Unité Linéaire Rectifiée)

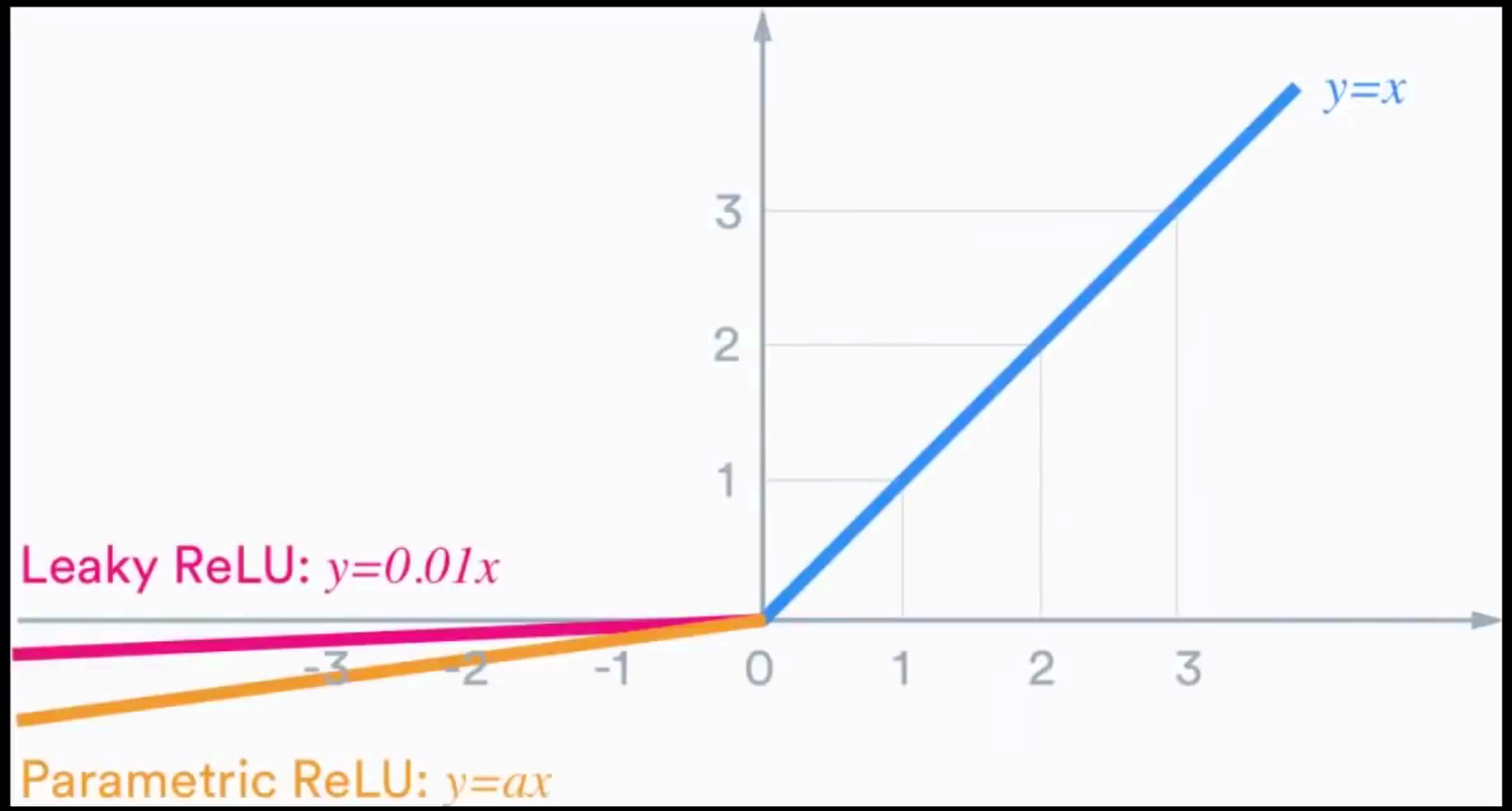


Gradient = 0

Problème de la valeur mourante

Fonctions d’Activation

f. Fonction Leaky ReLU (qui fuit)



Fonctions d’Activation

Quelle Fonction d’Activation utiliser en pratique ?

- Problèmes de Classification Binaire : Sigmoïde
- Si vous n’êtes pas sûr : ReLU (ou ReLU modifiée)

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
  - Processus d’apprentissage: Propagation avant/arrière
  - Optimiseurs
  - Fonctions d’Activation
  - Fonctions de Perte (*Loss function*)
  - Paramètres et Hyperparamètres
  - *Epochs*, *Batch size* & Itérations
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Fonctions de Perte

Quantifier l’écart entre la sortie prédite par le réseau de neurones et la sortie attendue

Différents types de Fonctions de Perte selon le projet qui vous occupe

Fonctions de Perte

Régression : Erreur au carré, perte de Huber

Classification Binaire : Entropie-Croisée Binaire,
Perte de Hinge

Classification Multi-Classe : Entropie-Croisée Multi-Classe,
Divergence de Kullback-Leibler

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
  - Processus d’apprentissage: Propagation avant/arrière
  - Optimiseurs
  - Fonctions d’Activation
  - Fonctions de Perte (*Loss function*)
  - Paramètres et Hyperparamètres
  - *Epochs*, *Batch size* & Itérations
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Paramètres et Hyperparamètres

Qu’est-ce que les Paramètres de Modèle ?

Les variables internes aux réseaux de neurones.
Les valeur peuvent être estimées à partir
des données elles-mêmes

Paramètres et Hyperparamètres

Paramètres de Modèle

- Requis par le modèle pour formuler des prévisions
- Définissent la compétence du modèle
- Estimées directement à partir des données
- Ne sont pas définis manuellement
- Sauvegardés comme composants majeurs du modèle d'apprentissage
- Exemples : Poids, Biais

Paramètres et Hyperparamètres

Qu’est-ce que les Hyper-paramètres de Modèle ?

Configurés de façon externe aux réseaux de neurones.

Les valeur ne peuvent être estimées

à partir des données elles-mêmes

Paramètres et Hyperparamètres

Hyperparamètres de Modèle

- Pas de manière évidente de trouver la meilleure valeur
- Quand un algorithme DL est réglé, on ajuste vraiment les hyperparamètres
- Si l'on doit spécifier manuellement un paramètre, alors c'est un hyperparamètre
- Exemples : Taux d'apprentissage, C & σ dans SVMs

Paramètres et Hyperparamètres

Résumé

- Paramètres de Modèle → Estimés à partir des données
- Hyperparamètres de Modèle → Ne peuvent pas être estimés à partir des données

Les hyperparamètres de modèle sont souvent appelés paramètres
car ce sont les parties de l'apprentissage automatique
qui doivent être définies et réglées manuellement

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
  - Processus d’apprentissage: Propagation avant/arrière
  - Optimiseurs
  - Fonctions d’Activation
  - Fonctions de Perte (*Loss function*)
  - Paramètres et Hyperparamètres
  - *Epochs*, *Batch size* & Itérations
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Epochs, Batch size & Itérations

- Besoin de cette terminologie seulement si les données sont volumineuses
- Diviser l'ensemble de données en de plus petits sous-ensembles et et les fournir, un par un, au réseau de neurones

Epochs, Batch size & Itérations

Epochs

Quand un ensemble de données dans sa totalité est passé au réseau, en avant et en arrière, en une SEULE FOIS

On utilise plus d’une époque, pour aider le modèle à généraliser de meilleure façon

Pas de bon nombre d’époques, malheureusement !

Epochs, Batch size & Itérations

Lot et taille d’un lot

On divise le grand ensemble de données en *batchs* plus petits,
que l’on fournit au réseau de neurones

Batch size : Nombre total d’exemples d’entraînement
contenus dans un *batch*

Epochs, Batch size & Itérations

Itérations

Nombre de lots nécessaires pour compléter une *epoch*

Alternativement,

Nombre de lots = Nombre d'itérations par époque

Epochs, Batch size & Itérations

Supposons que l’on dispose d’un ensemble de données composé de 34000 exemples d’entraînement

et que l’on divise cet ensemble de données en lots de 500

Pour compléter une époque, on aura besoin de 68 itérations

Epochs, Batch size & Itérations

De combien de couches cachées
ai-je besoin ?

Quelle fonction d’Activation dois-je choisir ?

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Plan

- Introduction à l'apprentissage profond
- Terminologie en apprentissage profond
- Types d’apprentissage
  - Apprentissage Supervisé
  - Apprentissage Non Supervisé
  - Apprentissage par Renforcement
- Régularisation
- Architectures
- Comment créer un modèle ?

1- Apprentissage Supervisé

- Algorithmes conçus pour apprendre par l'exemple
- Modèles entrainés sur des données bien labélisées

1- Apprentissage Supervisé

Chaque exemple est une paire composée de :

- Objet en entrée (typiquement un vecteur)
- Une Sortie Souhaitée (Signal de Supervision)

Pendant l’entraînement,

L’algorithme d'apprentissage supervisé recherche des modèles qui corrèlent les sorties souhaitées

Après l’entraînement,

Utiliser des entrées non vues, et détermine le label de classification

1- Apprentissage Supervisé

Objectif d’un Modèle d’Apprentissage Supervisé :

Prédire le label correct
pour les données invisibles

1- Apprentissage Supervisé

$$y = f(x)$$

x = Entrée

y = Sortie Prédite

1- Apprentissage Supervisé

1- Apprentissage Supervisé

1- Classification

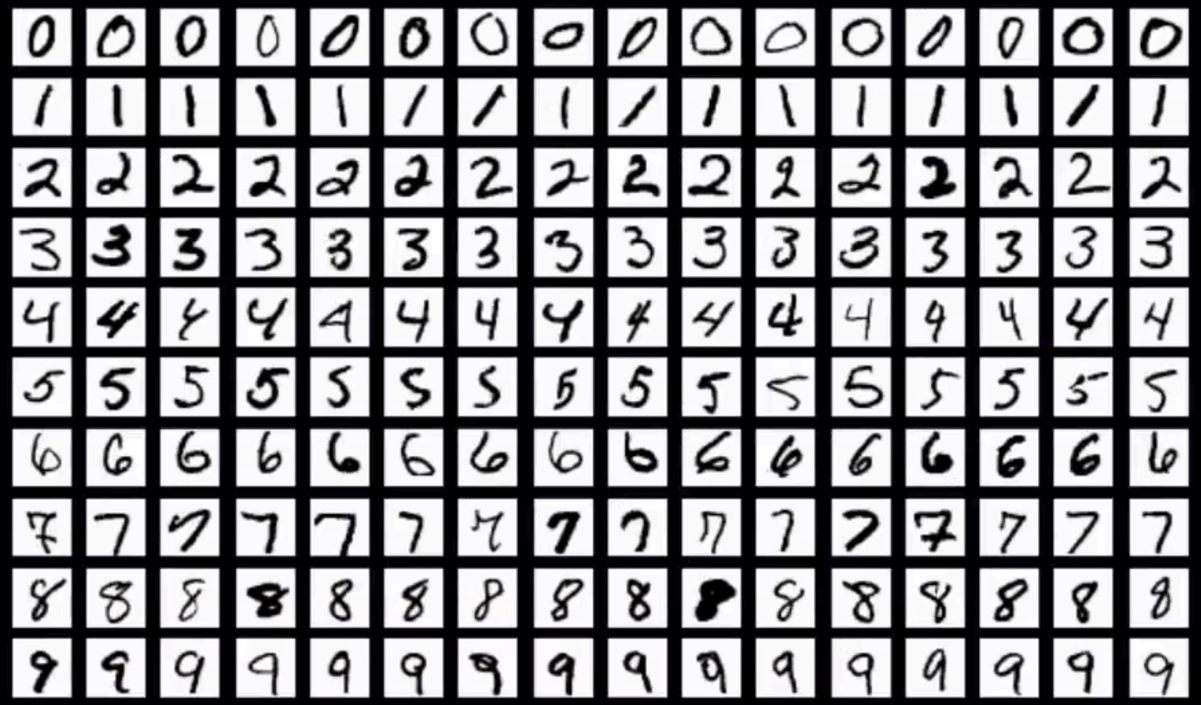
Prendre une valeur en entrée et lui assigner une classe / catégorie

Exemple : est-ce que cet email est un spam ou pas un spam

Les modèles trouvent des fonctionnalités dans les données qui correspondent à une classe, et créent une fonction de mappage

Cette fonction de mappage sera utilisée pour classifier les données invisibles

1- Apprentissage Supervisé



Base de Données MNIST des Chiffres écrits à la main

Introduction aux apprentissages

Algorithmes de Classification Populaires

1. Classificateurs Linéaires
2. *Support Vector Machine* (SVM)
3. K-*nearest neighbour*
4. Forêt Aléatoire (*random forest*)

1- Apprentissage Supervisé

2- Régression

Le modèle essaie de trouver la relation entre variables dépendantes et variables indépendantes

But : Prédire des valeurs continues comme des Résultats de Test

1- Apprentissage Supervisé

Equation pour la Régression Linéaire de Base

$$\hat{y} = w[0]*x[0] + w[1]*x[1] + \ldots\ldots + w[i]*x[i] + b$$

x[i] = i^{th} Caractéristique d'Entrée

w[i],b = Paramètres

De Nombreuses Caractéristiques d'Entrée

1- Apprentissage Supervisé

Equation pour la Régression Linéaire de Base

$$\hat{y} = wx + b$$

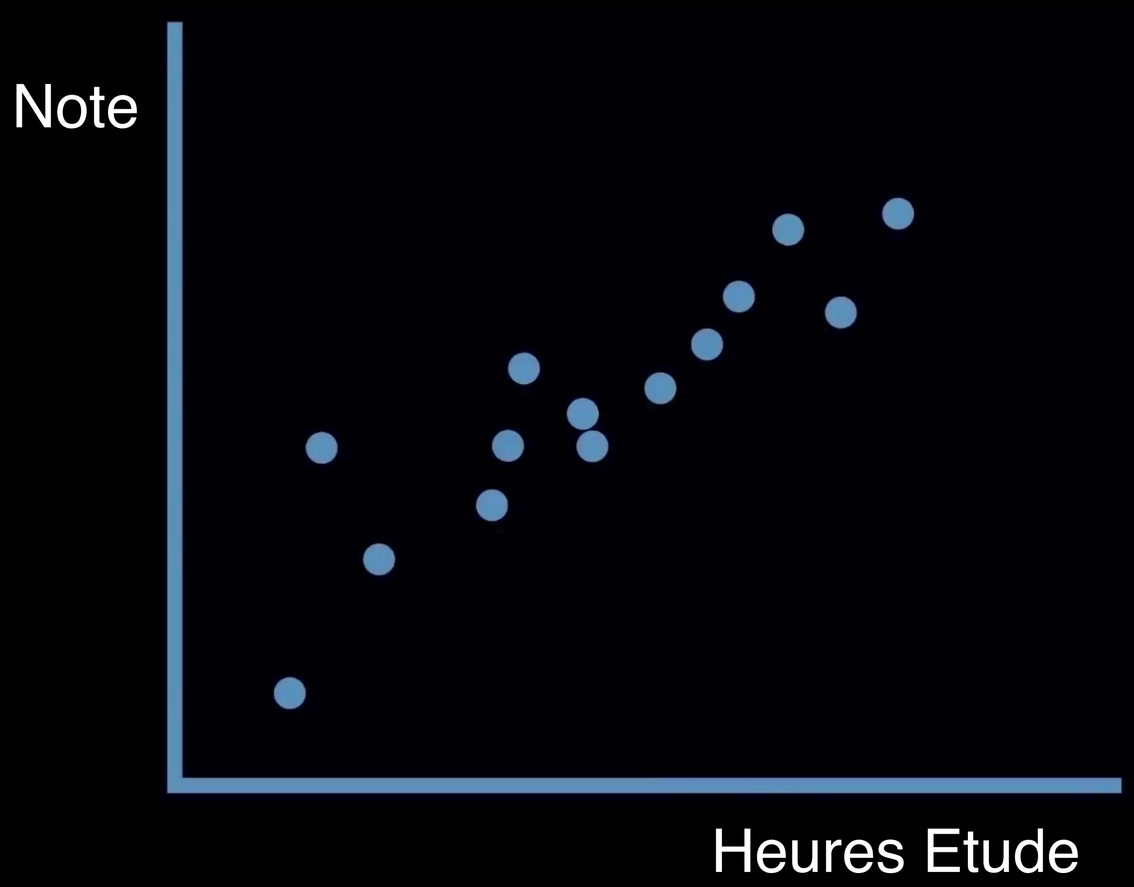
x = Caractéristique Unique d'Entrée

w = Pente

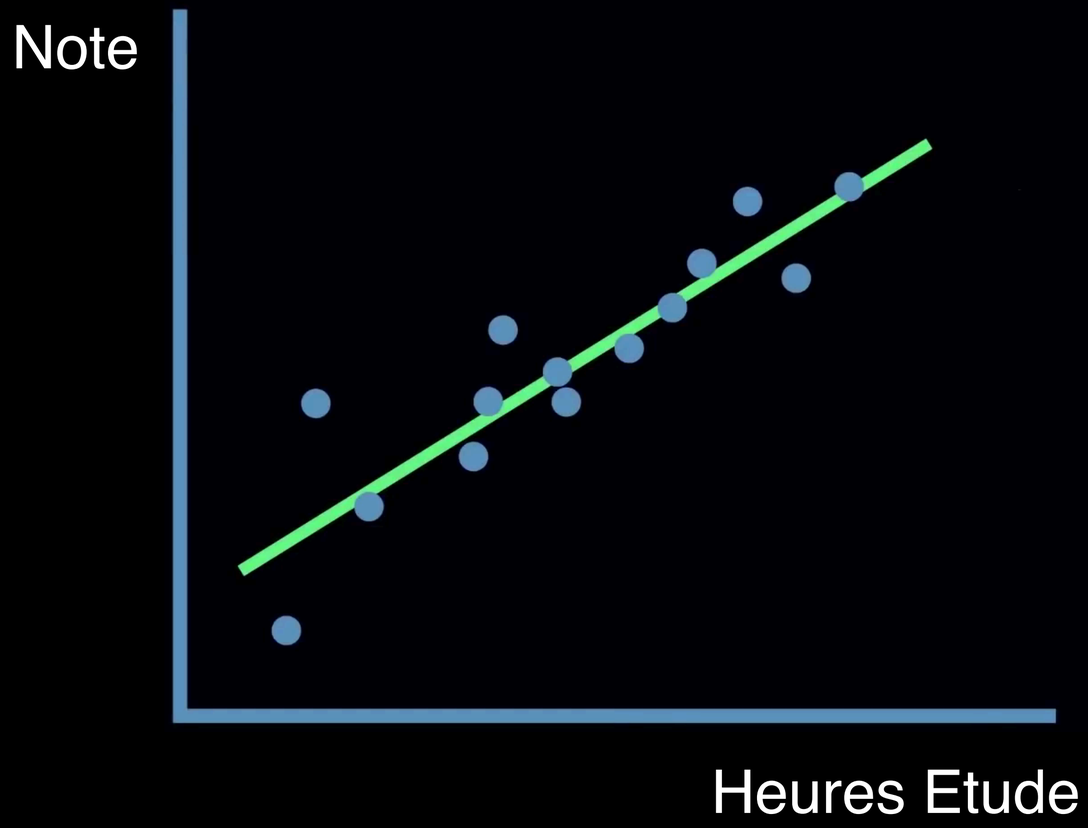
b = Ordonnée à l'Origine

Une Caractéristique d'Entrée

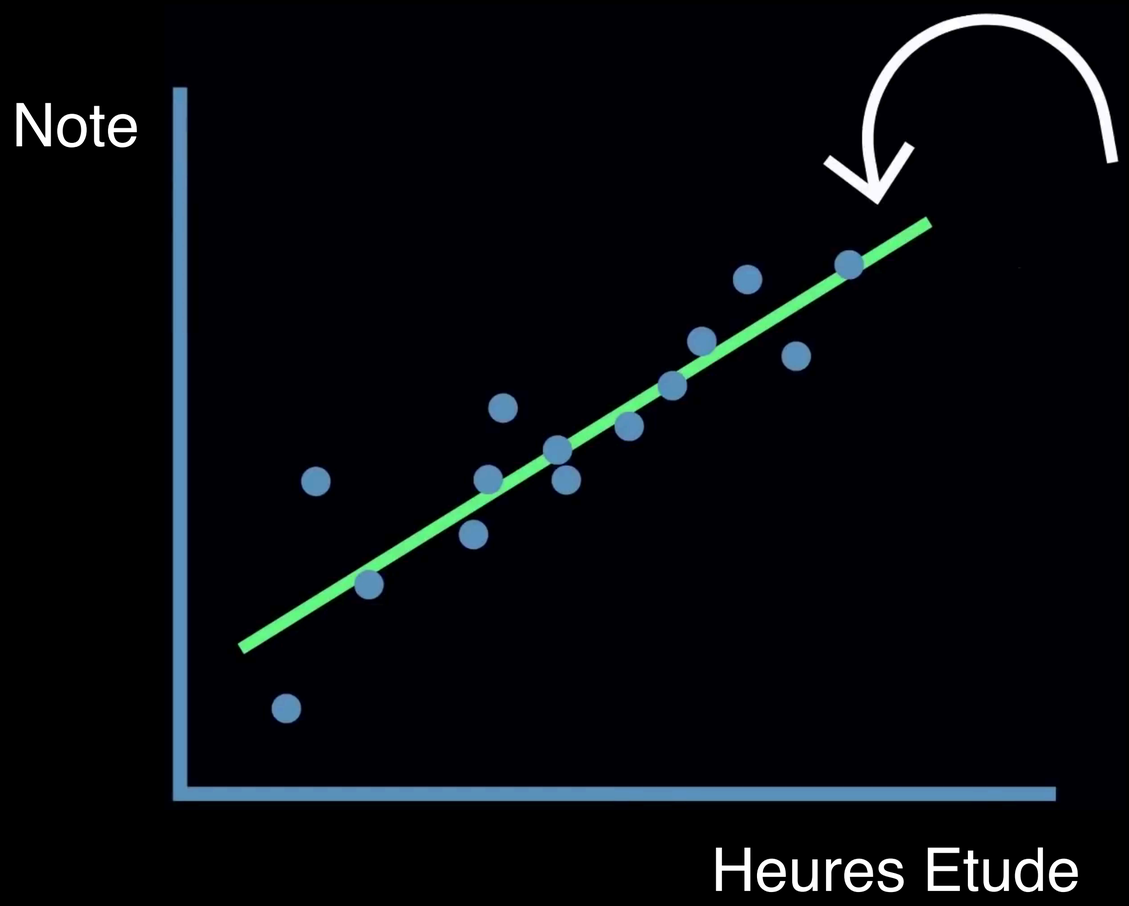
1- Apprentissage Supervisé



1- Apprentissage Supervisé



1- Apprentissage Supervisé



En se fondant sur cette relation, nous pouvons prédire les notes futures

1- Apprentissage Supervisé

La Base de Données sur les Prix des Maisons à Boston
fournit un bon exemple de Problème de Régression

1- Apprentissage Supervisé

Algorithmes de Régression Populaires

1. Régression Linéaire
2. Régression Lasso
3. Régression Multivariée

1- Apprentissage Supervisé

Applications d'Apprentissage Supervisé

1. Bioinformatique
2. Reconnaissance d'Objets
3. Détection de Spam
4. Reconnaissance Vocale

Plan

- Introduction à l'apprentissage profond
- Terminologie en apprentissage profond
- Types d’apprentissage
  - Apprentissage Supervisé
  - Apprentissage Non Supervisé
  - Apprentissage par Renforcement
- Régularisation
- Architectures
- Comment créer un modèle ?

2- Apprentissage Non Supervisé

- Utilisé pour mettre en évidence des modèles sous-jacents dans les données
- Utilisé dans l'analyse exploratoire des données
- N'utilise pas de données étiquetées, s'appuie plutôt sur les données
- Objectif : Analyser les données et trouver des modèles sous-jacents importants

2- Apprentissage Non Supervisé

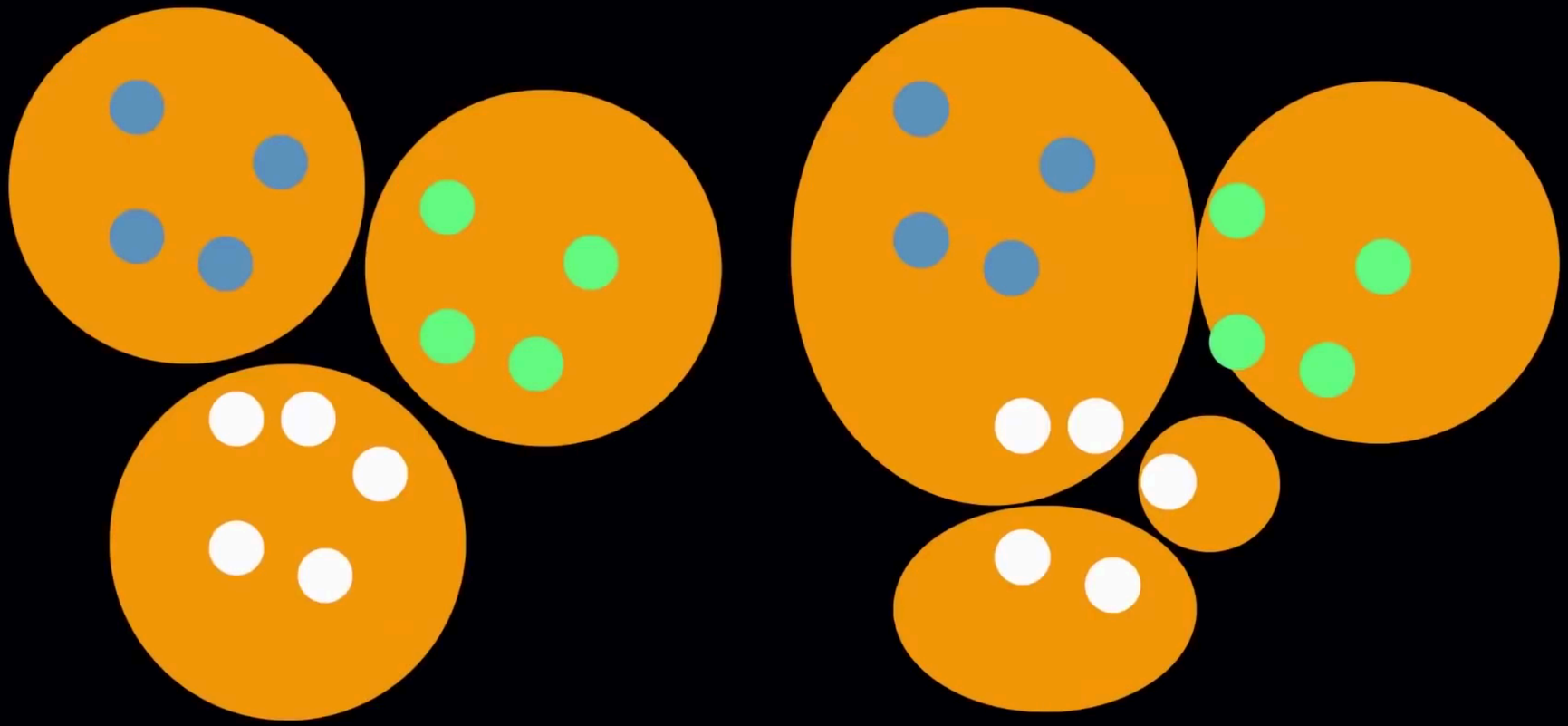
2- Apprentissage Non Supervisé

1- Regroupement

Processus de regroupement des données en différents clusters ou groupes

But : Prédire des valeurs continues telles que des Résultats de Tests

2- Apprentissage Non Supervisé



Bon Clustering

Mauvais Clustering

2- Apprentissage Non Supervisé

Clustering :

1- Clustering Partitionnel

Chaque point de donnée peut appartenir à un seul cluster

2- Clustering Hiérarchique

Des clusters à l’intérieur de clusters

Un point de donnée peut appartenir à plusieurs clusters

2- Apprentissage Non Supervisé

Algorithmes de Clustering Populaires

1. K-Moyennes
2. Maximisation des Attentes
3. Analyse Hiérarchique de Cluster (HCA)

2- Apprentissage Non Supervisé

2- Association

Tente de trouver des relations entre différentes entités

Exemple : Analyse du Panier de Marché

2- Apprentissage Non Supervisé

Applications d'Apprentissage Non Supervisé

1. AirBnb
2. Amazon
3. Détection de la Fraude aux Cartes de Crédit

Plan

- Introduction à l'apprentissage profond
- Terminologie en apprentissage profond
- Types d’apprentissage
  - Apprentissage Supervisé
  - Apprentissage Non Supervisé
  - Apprentissage par Renforcement
- Régularisation
- Architectures
- Comment créer un modèle ?

3- Apprentissage par Renforcement

- Permet à un agent d'apprendre dans un environnement interactif par essais et erreurs, en utilisant les retours de ses propres actions & expériences
- Utilise des récompenses & sanctions comme signaux de comportements positifs ou négatifs
- But : Trouver un modèle adapté qui maximiserait le total cumulé des récompenses

3- Apprentissage par Renforcement

- Maximise les points gagnés dans un jeu sur de nombreux coups

Pénalisent quand ils prennent les mauvaises décisions

Récompensent quand ils en prennent de bonnes

Généralement modélisé comme un Precessus de Décision de Markov

3- Apprentissage par Renforcement

3- Apprentissage par Renforcement

3- Apprentissage par Renforcement

Applications d'Apprentissage par Renforcement

1. Robotique
2. Planification de Stratégie Commerciale
3. Contrôle des Feux de Signalisation
4. Configuration de Systèmes Web

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

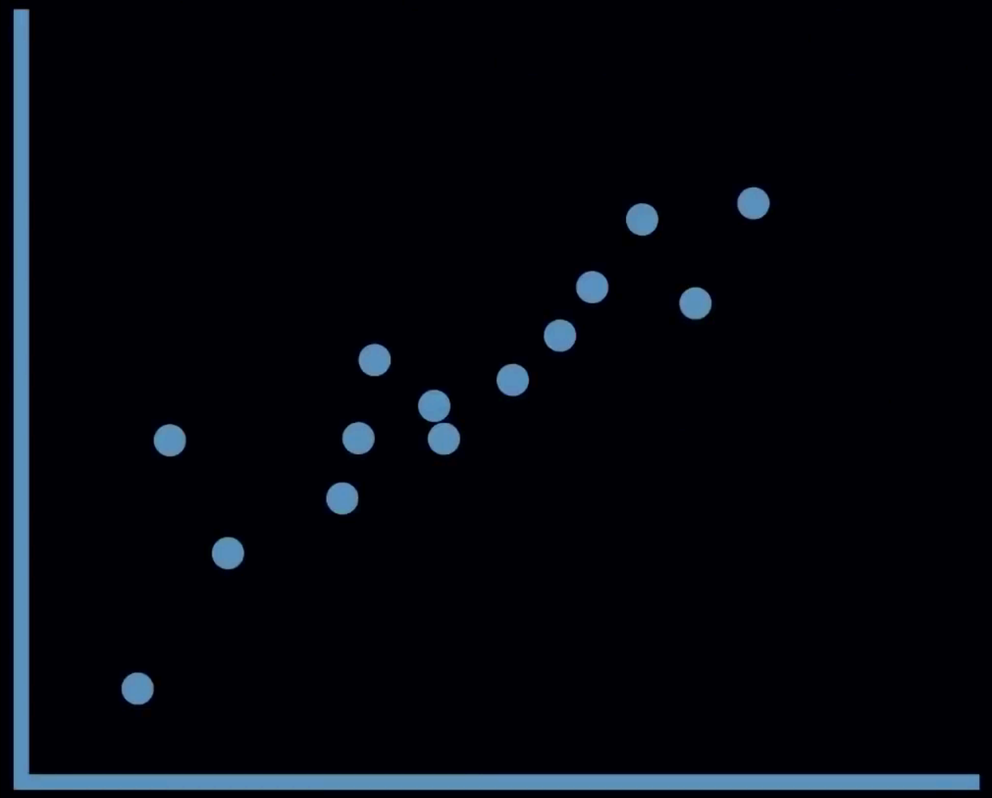
Régularisation

Problème Central dans l'Apprentissage Profond

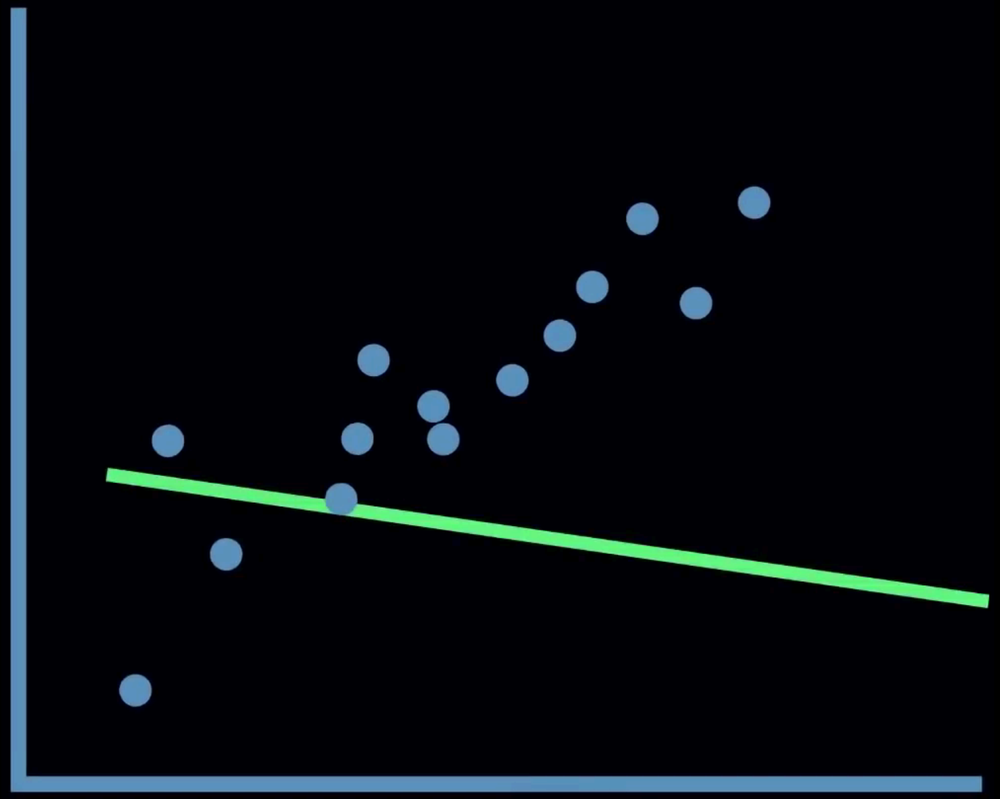
Le modèle doit bien fonctionner sur les données d'entraînement ET sur les nouvelles donnés

Le problème le plus courant est le Surajustement (*overfitting*)

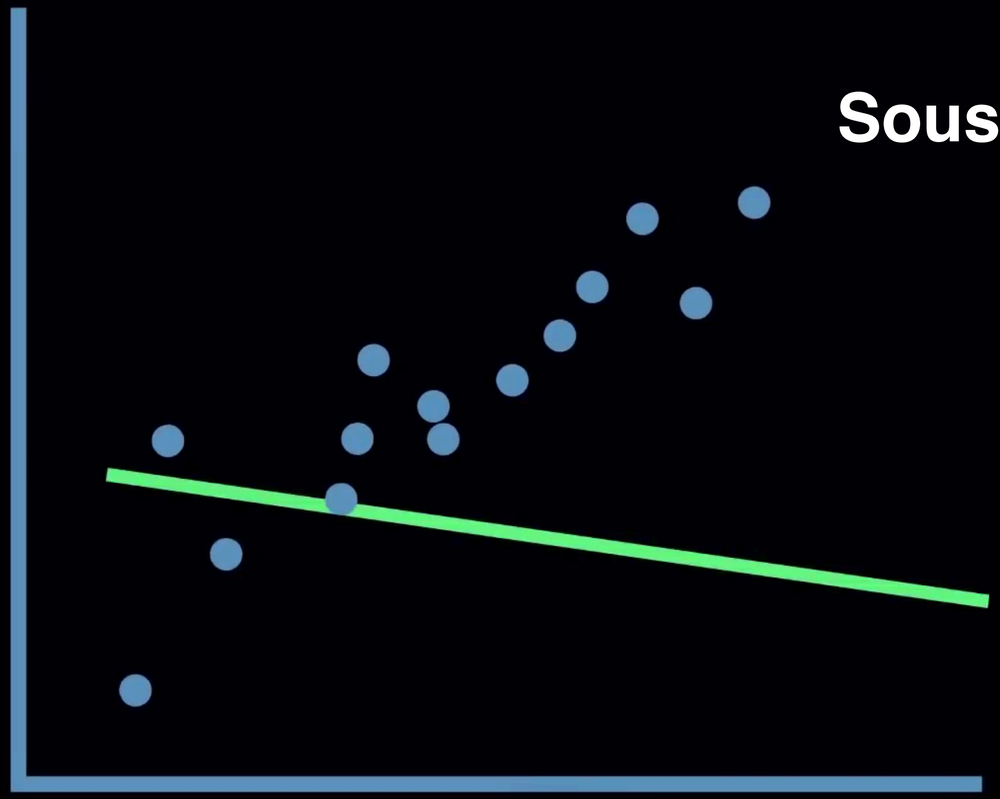
Régularisation



Régularisation

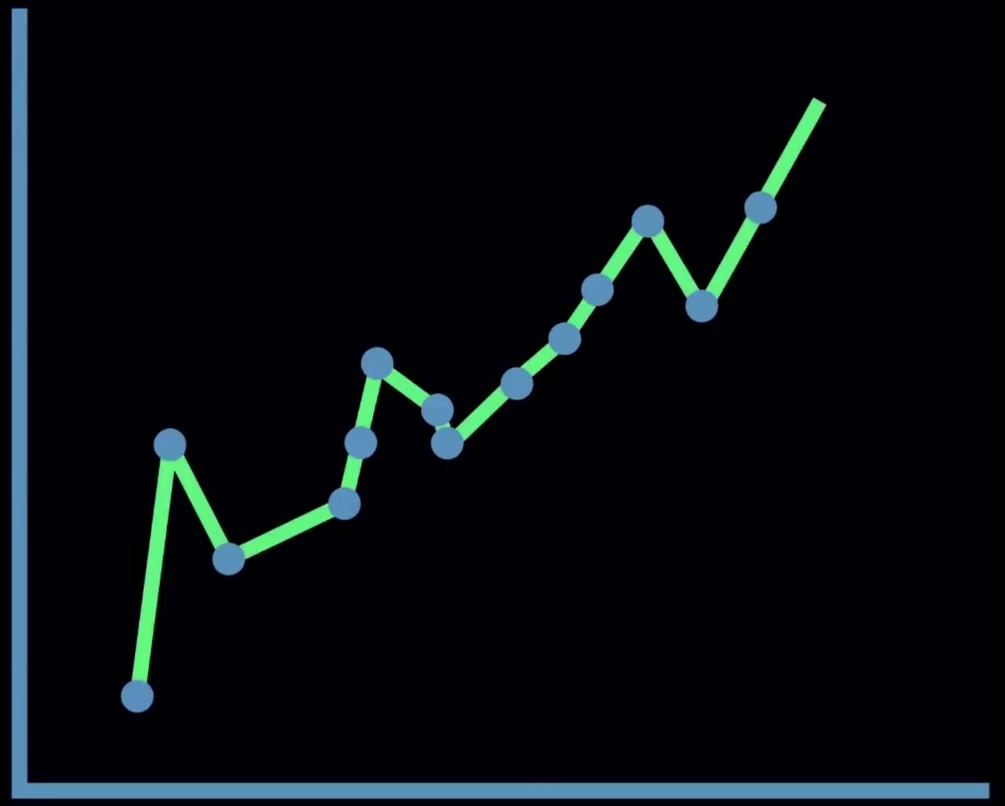


Régularisation



Sous-Ajustement

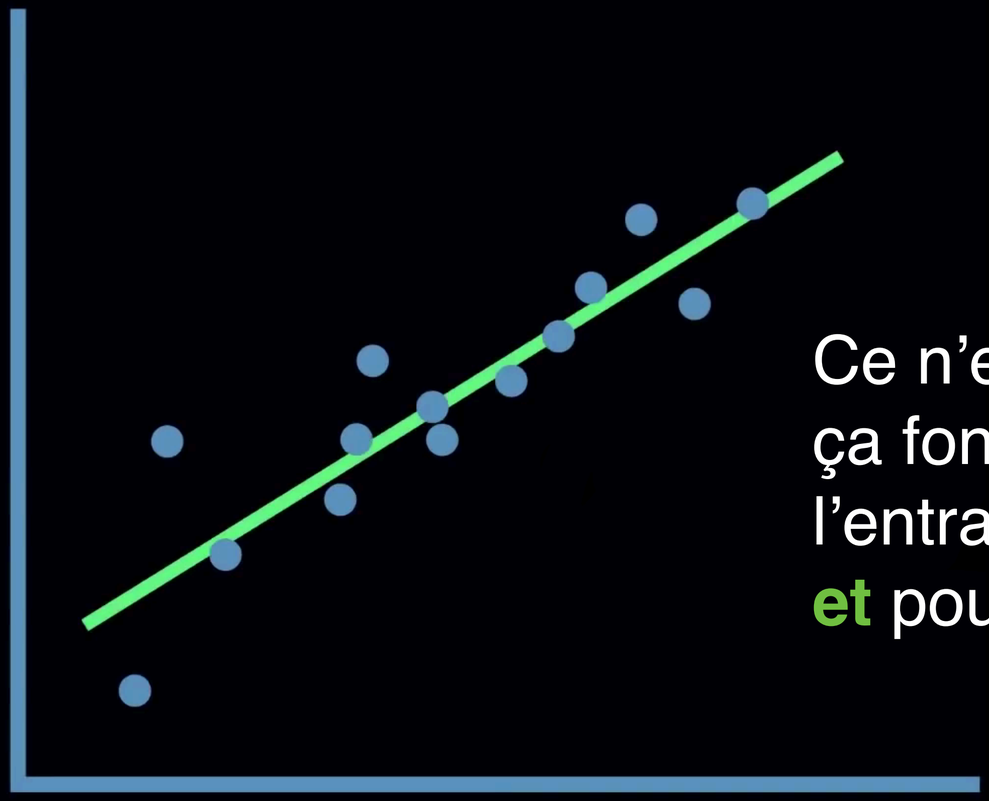
Régularisation



Régularisation

Sur-Ajustement

Régularisation



Ce n’est pas parfait, mais ça fonctionnera mieux pour l’entraînement des données **et** pour le test des données

Régularisation

Lutter contre le sur-Ajustement

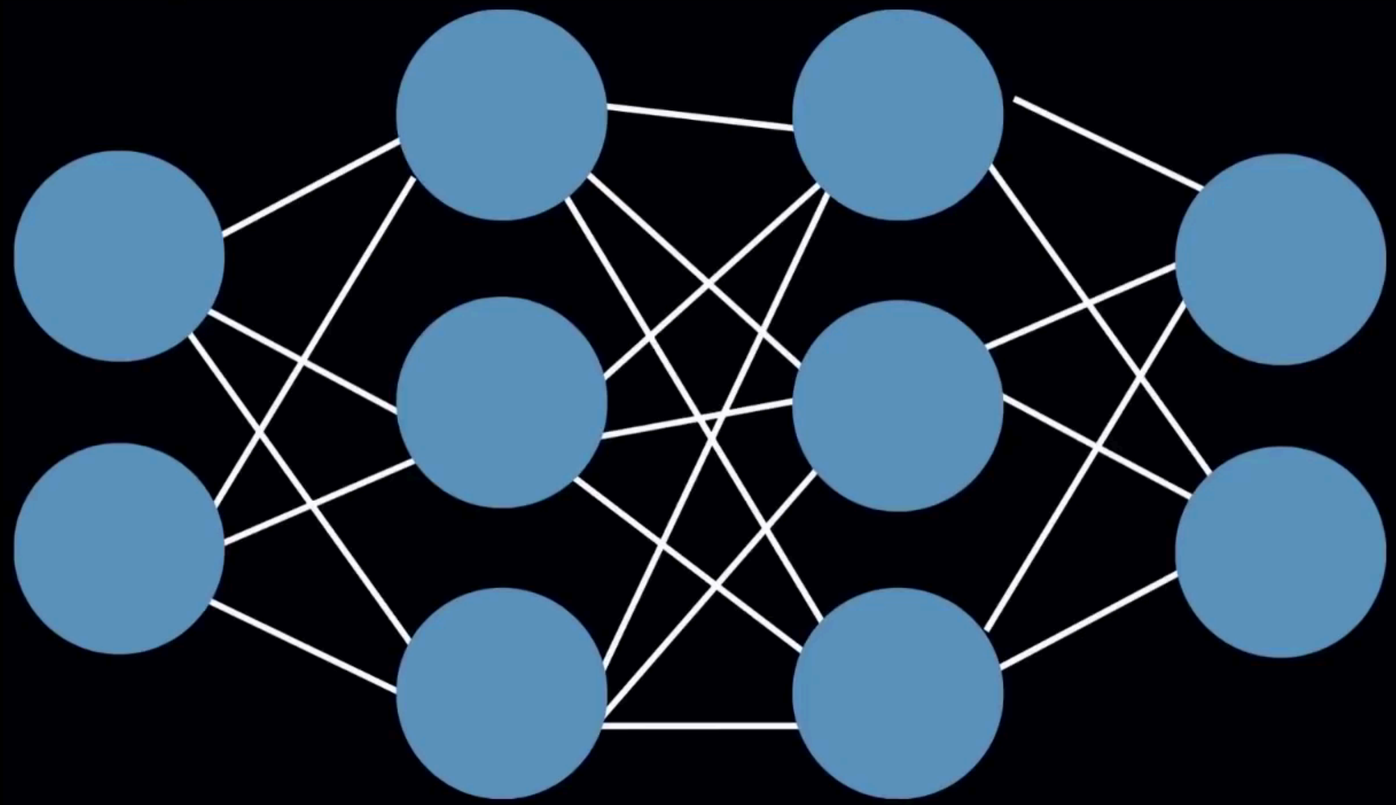
1- Abandon (*Dropout*)

2- Augmentation

3- Arrêt Précoce

Régularisation

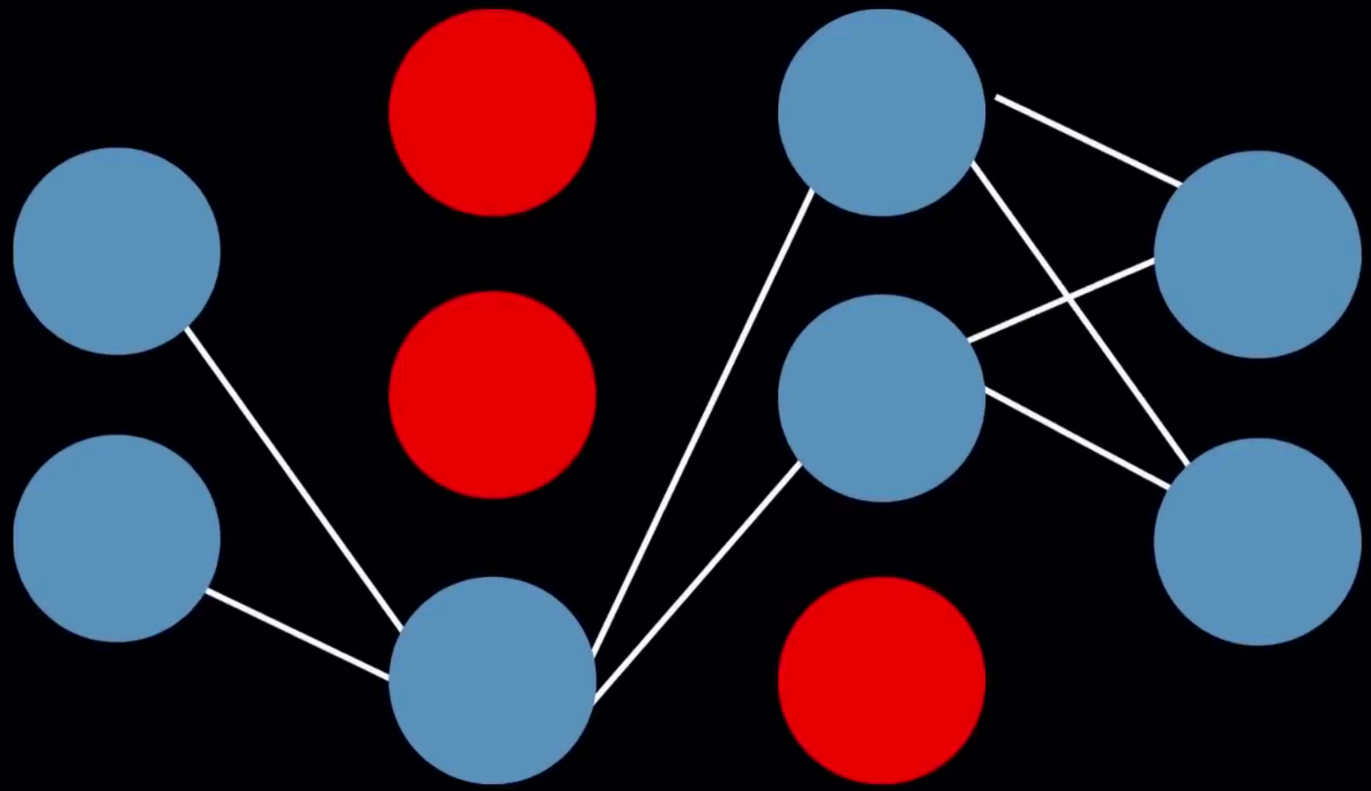
1- Abandon (*Dropout)*



Régularisation

1- Abandon (*Dropout)*

L’abandon au hasard supprime certains noeuds et leurs connexions



Régularisation

2- Augmentation de l’Ensemble de Données

Plus de Données Meilleur Modèle

Fausses Données

Régularisation

2- Augmentation du Jeu de Données

Appliquer des transformations sur l’ensemble de données existant pour synthétiser plus de données

Régularisation

2- Augmentation du Jeu de Données

Appliquer des transformations sur l'ensemble de données existant pour synthétiser plus de données

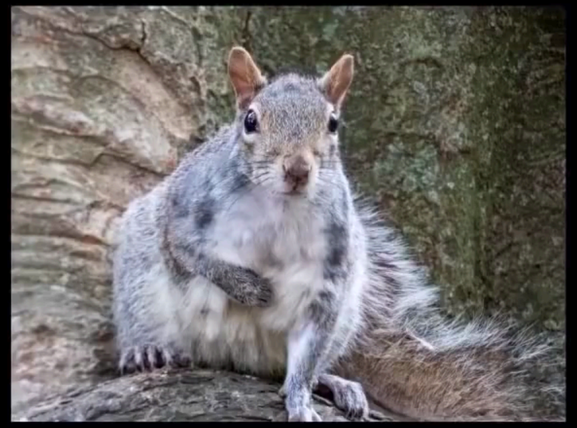


Image Originale

Régularisation

2- Augmentation du Jeu de Données

Appliquer des transformations sur l'ensemble de données existant pour synthétiser plus de données

Image Retournée

Régularisation

2- Augmentation du Jeu de Données

Appliquer des transformations sur l’ensemble de données existant pour synthétiser plus de données

changement d'échelle

Régularisation

2- Augmentation du Jeu de Données

⚠️ Attention au "mauvaises" transformations !

b vs d

6 vs 9

Régularisation

3- Arrêt Précoce (*early stopping*)

L’erreur d’entraînement diminue régulièrement

Mais, l’erreur de validation augmente après un certain point

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Plan

- Introduction à l'apprentissage profond
- Terminologie en apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
  - Réseau de neurones à propagation avant - *feedforward network*
  - Réseau de neurones récurrents - *RNN*
  - Réseau de neurones convolutifs - *CNN*
- Comment créer un modèle ?

Réseaux de Neurones à Propagation Avant

Chaque neurone est connecté
à chaque couche suivante
sans connexion en arrière

Réseaux de Neurones à Propagation Avant

Les neurones ont des fonctions d'activation

- Linéaires
- Sigmoïdes
- Tanh
- ReLU

Non-Linéaires

Réseaux de Neurones à Propagation Avant

Entrées

Sorties

Couches Cachées

Neurones par couche cachée

Fonctions d’Activation

Réseaux de Neurones à Propagation Avant

Plus de Neurones

Ressources de Calcul plus Importantes

Plan

- Introduction à l'apprentissage profond
- Terminologie en apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
  - Réseau de neurones à propagation avant - *feedforward network*
  - Réseau de neurones récurrents - *RNN*
  - Réseau de neurones convolutifs - *CNN*
- Comment créer un modèle ?

Réseaux de Neurones Récurrents

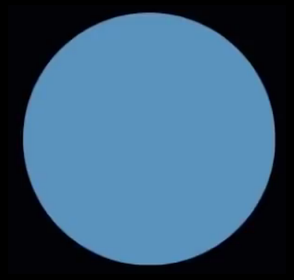
Les Réseaux de Neurones à Propagation Avant

Ils prennent des entrées des valeurs de taille fixe

Ils retournent en sortie des valeurs de taille fixe

Ils ne peuvent, à ce jour, modéliser tous les problèmes

Réseaux de Neurones Récurrents



Réseaux de Neurones Récurrents

"chien"

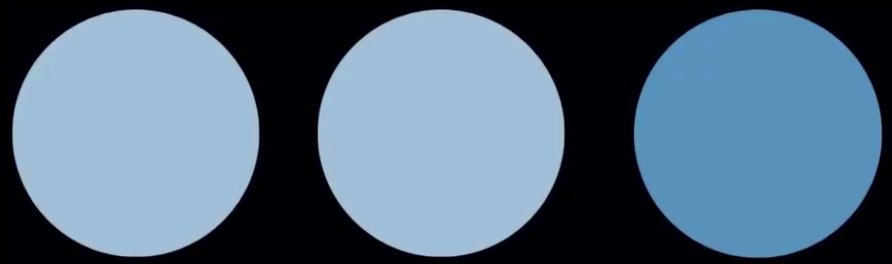
Réseaux de Neurones Récurrents

Cela n’a pas de sens !

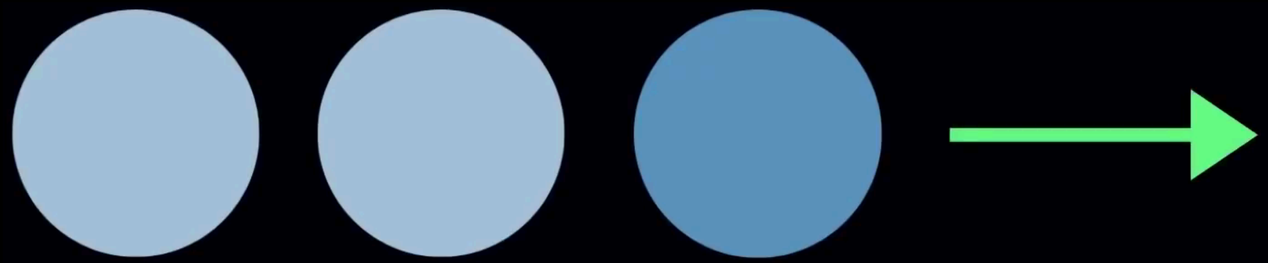
Réseaux de Neurones Récurrents

Peut être compris si
des informations sur le
passé ont été fournies

Réseaux de Neurones Récurrents



Réseaux de Neurones Récurrents



Réseaux de Neurones Récurrents

"J’ai un chien"

Réseaux de Neurones Récurrents

Comment cela s’applique
aux Réseaux de Neurones ?

Réseaux de Neurones Récurrents

L’information relative au passé
doit être fournie

Le Réseau de Neurones Vanille
ne peut pas gérer des données séquentielles

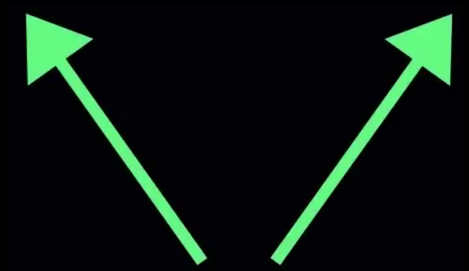
Réseaux de Neurones Récurrents

Les Données séquentielles sont des
données dans une séquence

Vanille NNs ne partagent pas des
paramètres dans le temps

Réseaux de Neurones Récurrents

Il pleuvait
Mardi



Mardi,
il pleuvait

Signifie la même chose

Réseaux de Neurones Récurrents

Les paramètres de partage donnent au réseau
la possibilité de rechercher une fonctionnalité donnée
partout dans une séquence,
plutôt que dans une seule zone

Réseaux de Neurones Récurrents

Gérer des séquences de longueur variable

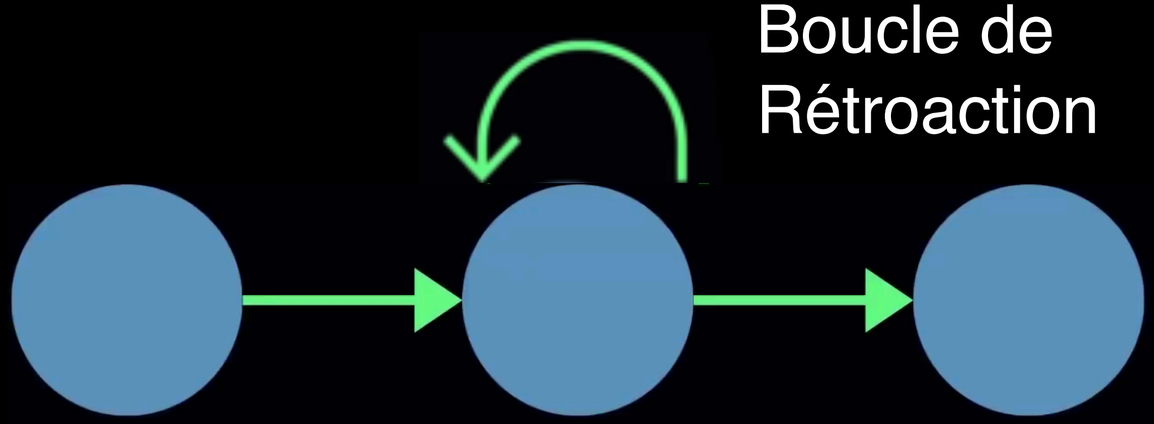
Maintenir l'ordre des séquences

Garder une trace des dépendances à long terme

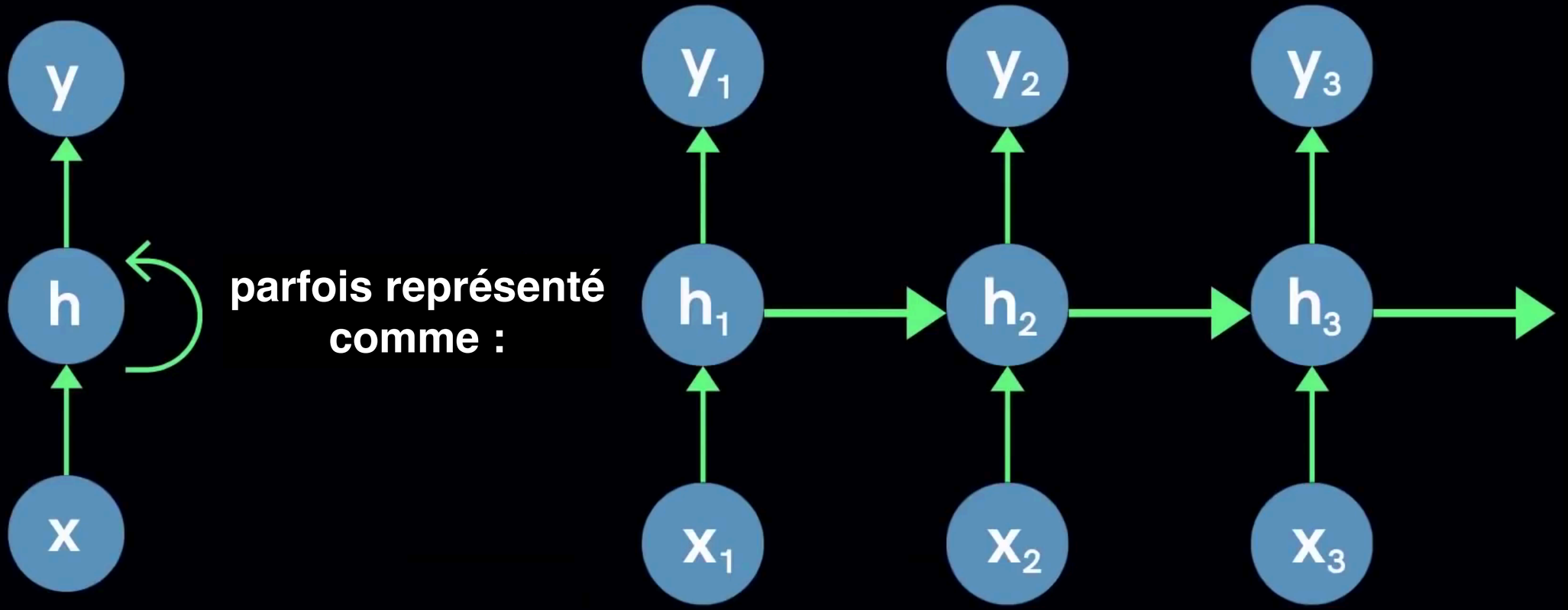
Partager des paramètres dans la séquence

Réseaux de Neurones Récurrents

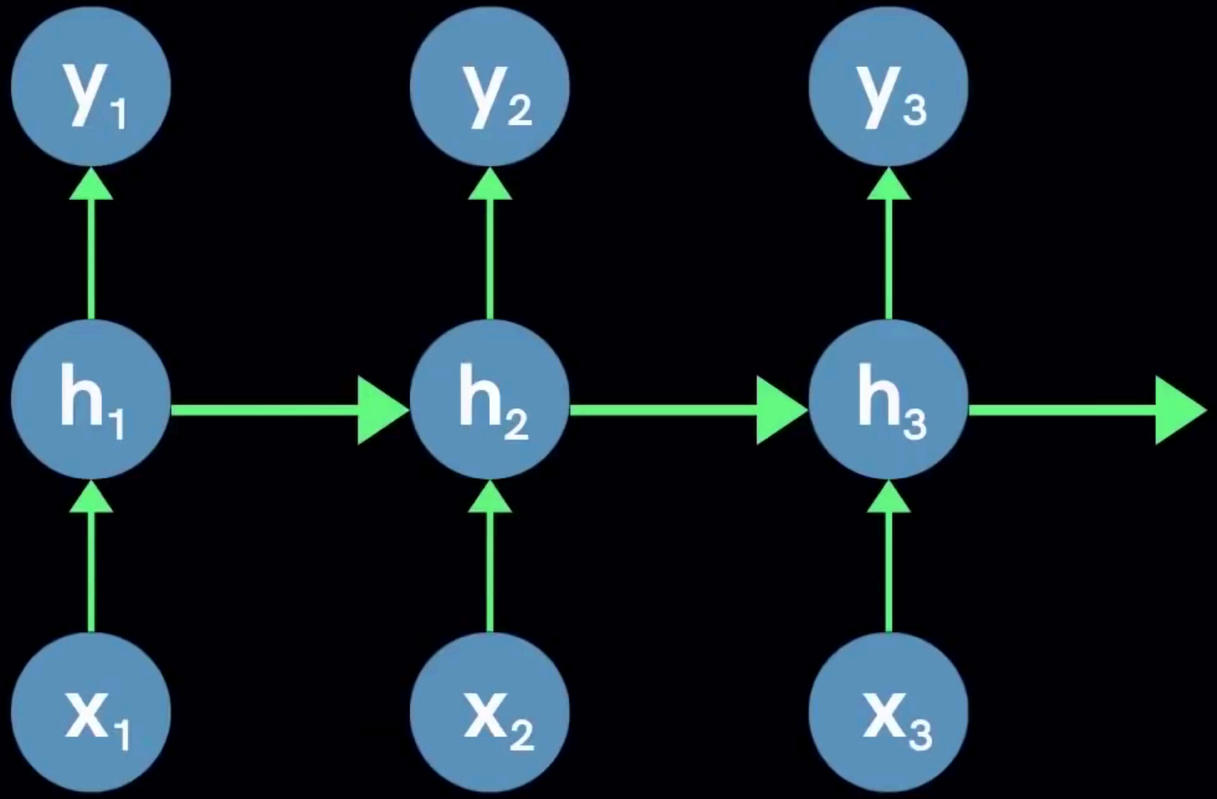
Utilise une boucle de rétroaction dans les couches cachés



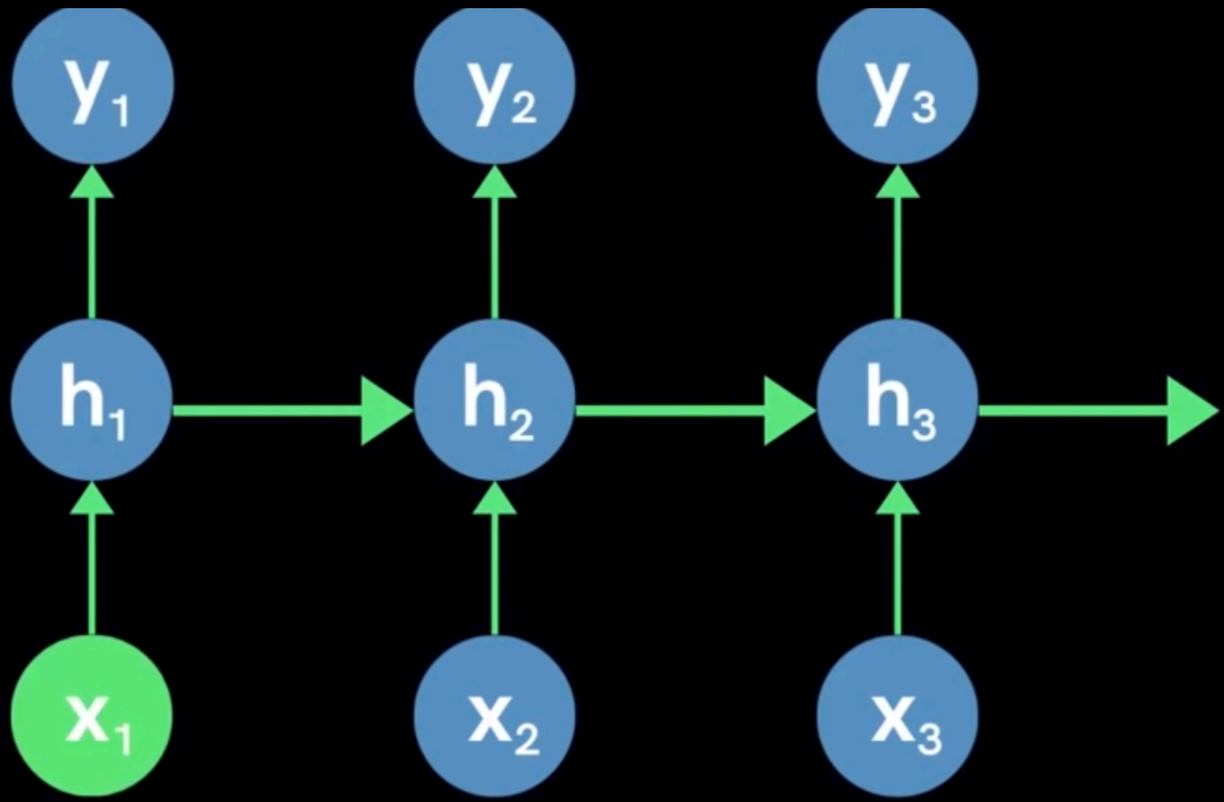
Réseaux de Neurones Récurrents



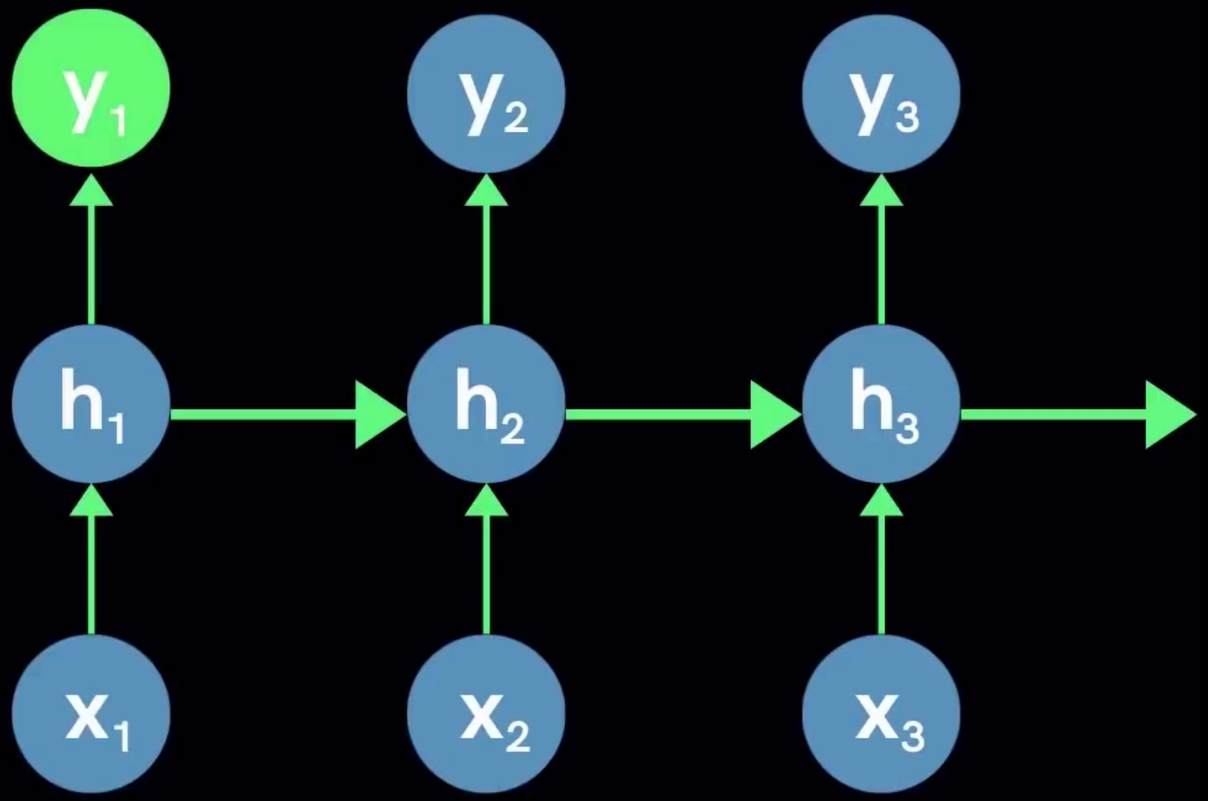
Réseaux de Neurones Récurrents



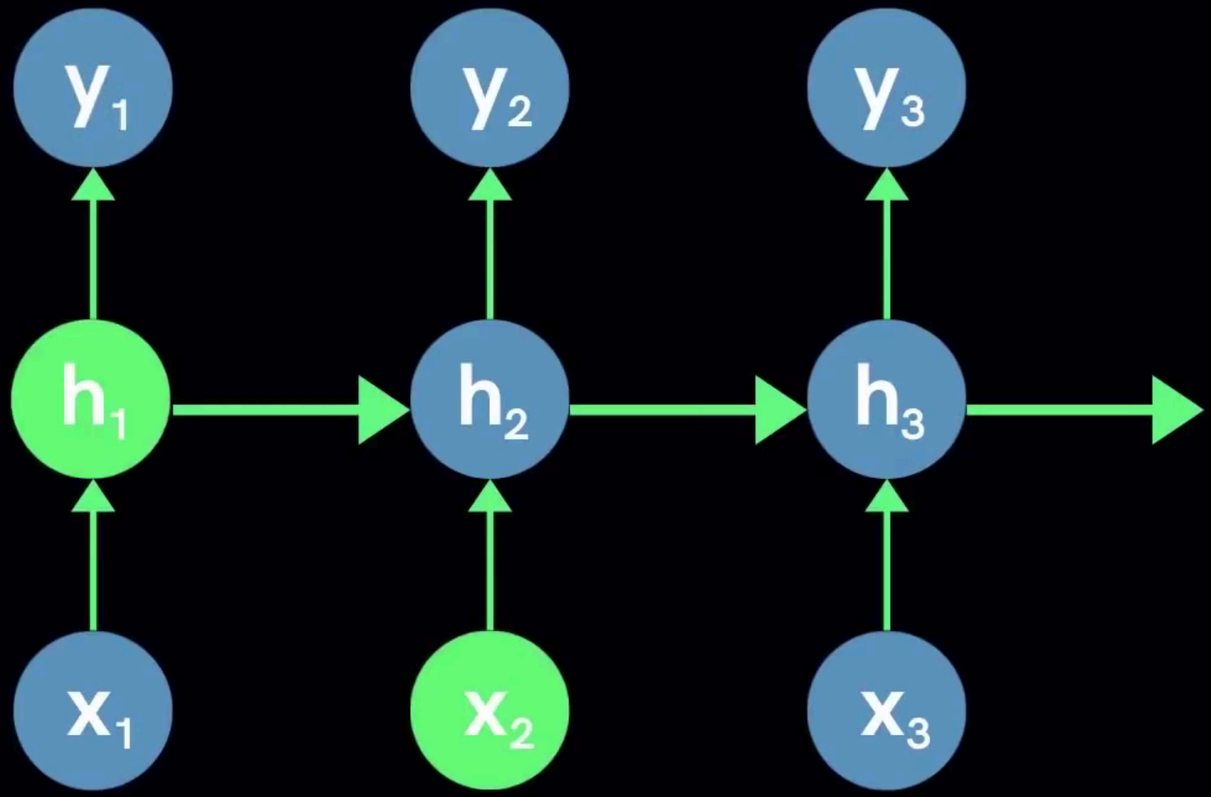
Réseaux de Neurones Récurrents



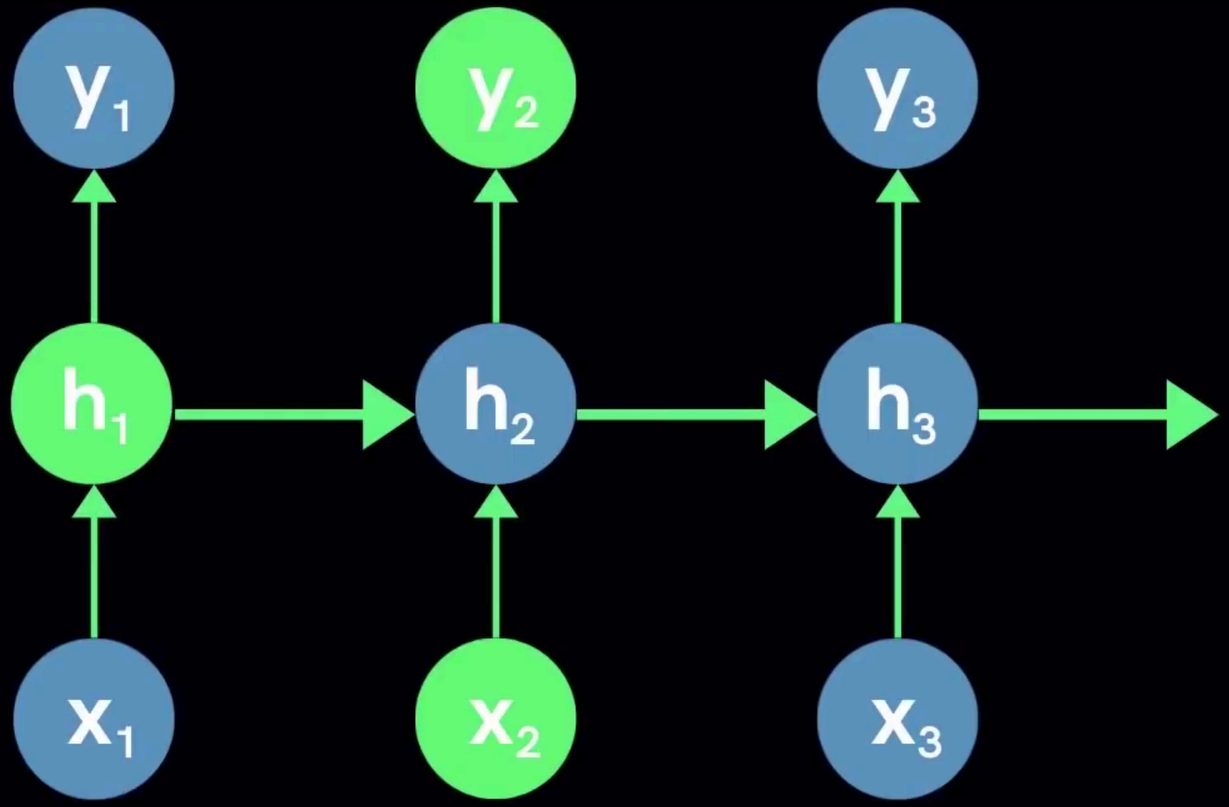
Réseaux de Neurones Récurrents



Réseaux de Neurones Récurrents



Réseaux de Neurones Récurrents



Réseaux de Neurones Récurrents

Former un modèle

Utilise l’algorithme de Propagation Arrière

Backprop appliqué pour chaque point de la séquence

Propagation Arrière à travers le temps (BTT)

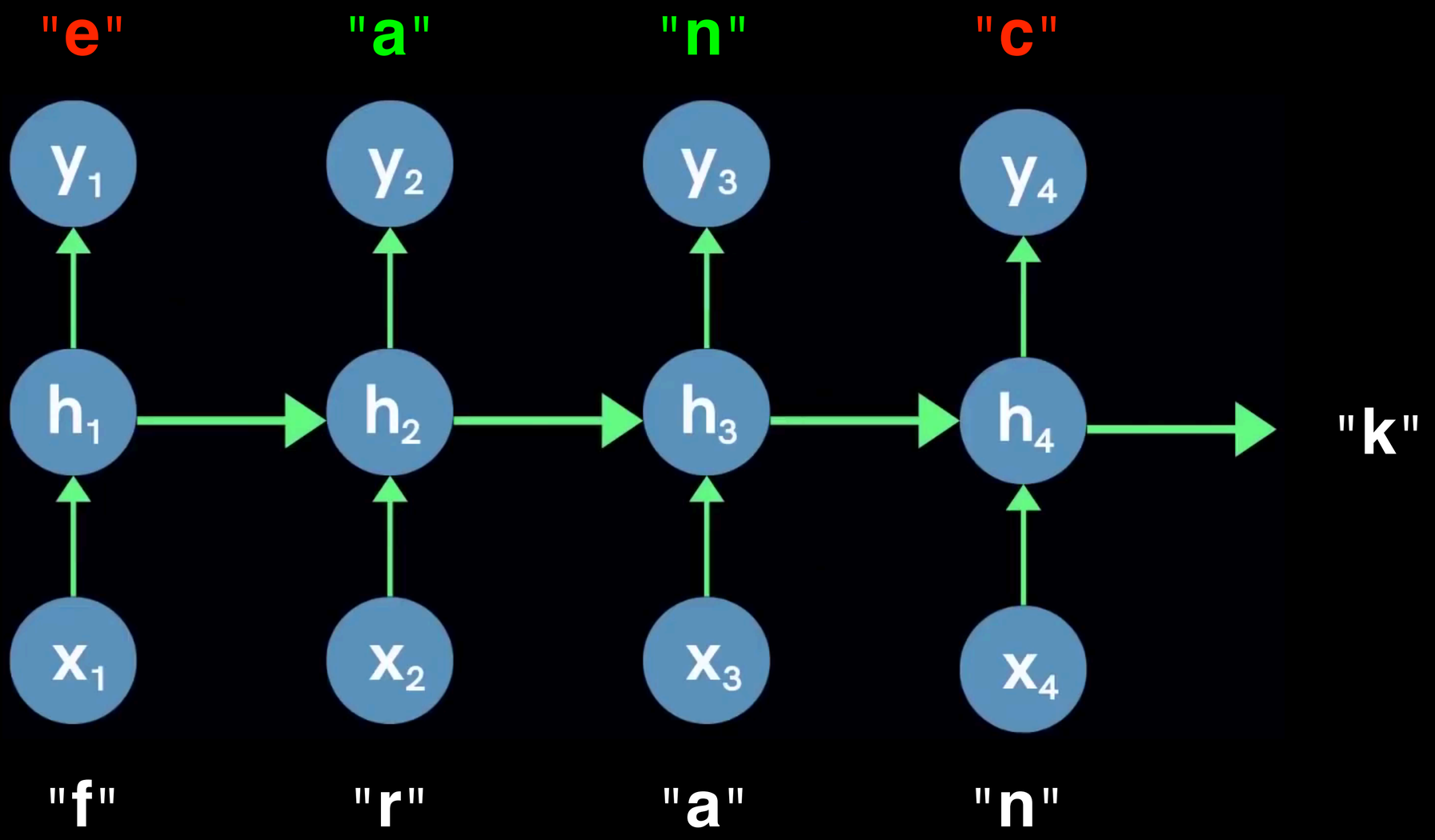
Réseaux de Neurones Récurrents

Prédiction Intelligente

Prédire les lettres en fonction des lettres tapées précédemment

Toutes les lettres tapées ont la même importance dans la prédiction

Réseaux de Neurones Récurrents



Réseaux de Neurones Récurrents

Problème de la Disparition du Gradient

La mémoire à court-terme d’un RNN est due au VGP

Due à la nature de la Propagation Arrière

Réseaux de Neurones Récurrents

En Propagation Arrière :

$$\text{Erreur} = \left(\text{Sortie Prévue} - \text{Sortie Attendue} \right)^2$$

Réseaux de Neurones Récurrents

En Propagation Arrière :

$$\text{Erreur} = \left(\text{Sortie Prévue} - \text{Sortie Attendue} \right)^2$$

$$W = w + n\frac{de}{dw}$$

Réseaux de Neurones Récurrents

Les gradients d'une couche sont calculés en fonction des gradients de la couche précédente

Si le gradient initial est petit, les ajustements aux couches suivantes seront plus petits, donnant lieu au problème de la disparition de gradients (*VGP*)

Réseaux de Neurones Récurrents

A chaque étape dans RNN = Une couche

On utilise la propagation arrière dans le temps (*BTT*)

Les Gradients utilisés font un ajustement sur les poids et les biais

A cause du *VGP*, les RNNs sont incapables d'apprendre les dépendances à longue portée

Réseaux de Neurones Récurrents

"It was raining on Tuesday"

"It" et "was" peuvent ne pas être considérés

Le modèle devra deviner en se basant sur "on Tuesday"

Complètement ambigu !

Réseaux de Neurones Récurrents

LSTM & GRNNs

Réseaux de Neurones Récurrents

LSTM - Long Short Term Memory

GRNN - Gated RNN

Capables d'apprendre des dépendances à long terme
en utilisant des portes

Réseaux de Neurones Récurrents

GRNN

- Porte de Mise à Jour
- Porte de Réinitialisation

LSTM

- Porte de Mise à Jour
- Porte de Réinitialisation
- Porte d’Oubli

Réseaux de Neurones Récurrents

GRNN

1. Traitement du Langage Naturel
2. Analyse des Sentiments
3. Classification des Séquences d’ADN
4. Reconnaissance Vocale
5. Traduction Linguistique

Plan

- Introduction à l'apprentissage profond
- Terminologie en apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
  - Réseau de neurones à propagation avant - *feedforward network*
  - Réseau de neurones récurrents - *RNN*
  - Réseau de neurones convolutifs - *CNN*
- Comment créer un modèle ?

Réseaux de Neurones Convolutif

NNs Convolutifs

Inspiré de l’organisation des neurones dans le cortex visuel du cerveau animal

Efficace dans l’entraînement de données comme : les images, l’audio et la vidéo

Réseaux de Neurones Convolutif

Couches Cachés dans un CNN

- Couches Convolutives
- Mise en commun des Couches - *Pooling*
- Couches entièrement connectés - *Fully-connected*
- Couches de Normalisation

Réseaux de Neurones Convolutif

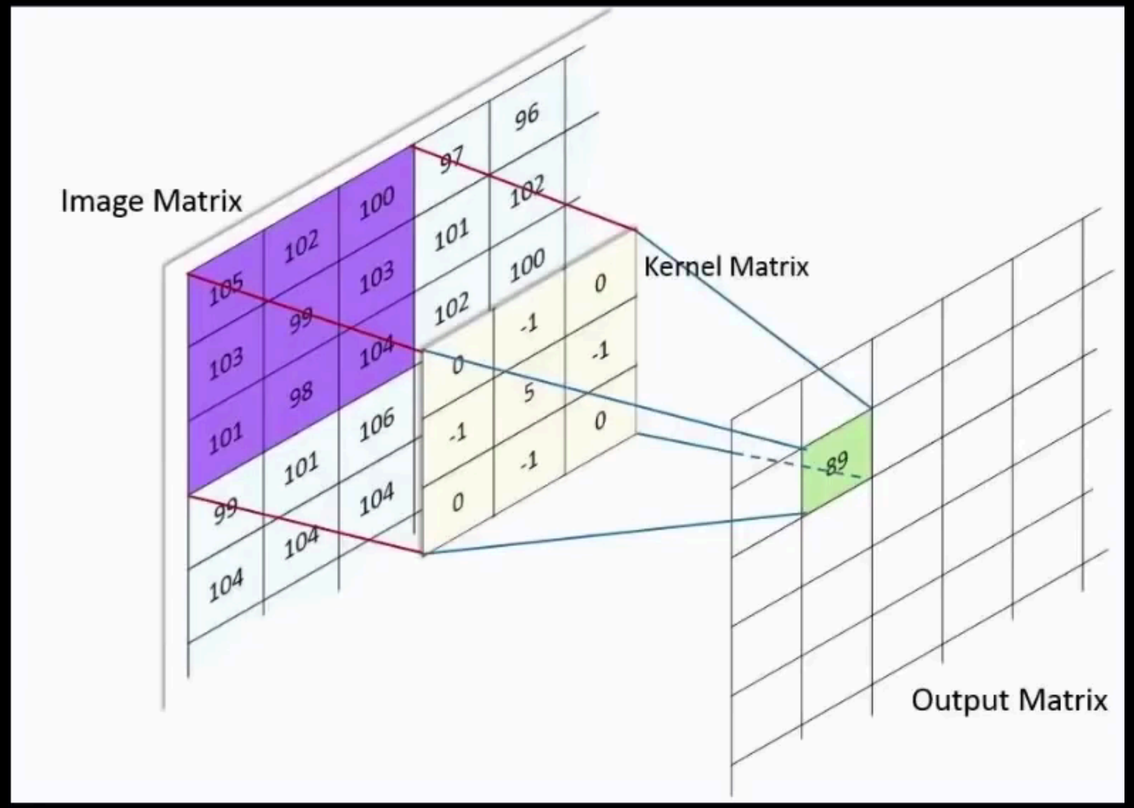
Entrée → 2D

Sortie → 1D

Réseaux de Neurones Convolutif

Convolutif

Nous permet d’extraire des fonctionnalités visuelles en morceaux



Réseaux de Neurones Convolutif

$$(f * g)(i) = \sum_{j=1}^{m} g(j) \cdot f(i - j + m/2)$$

Produit Scalaire !

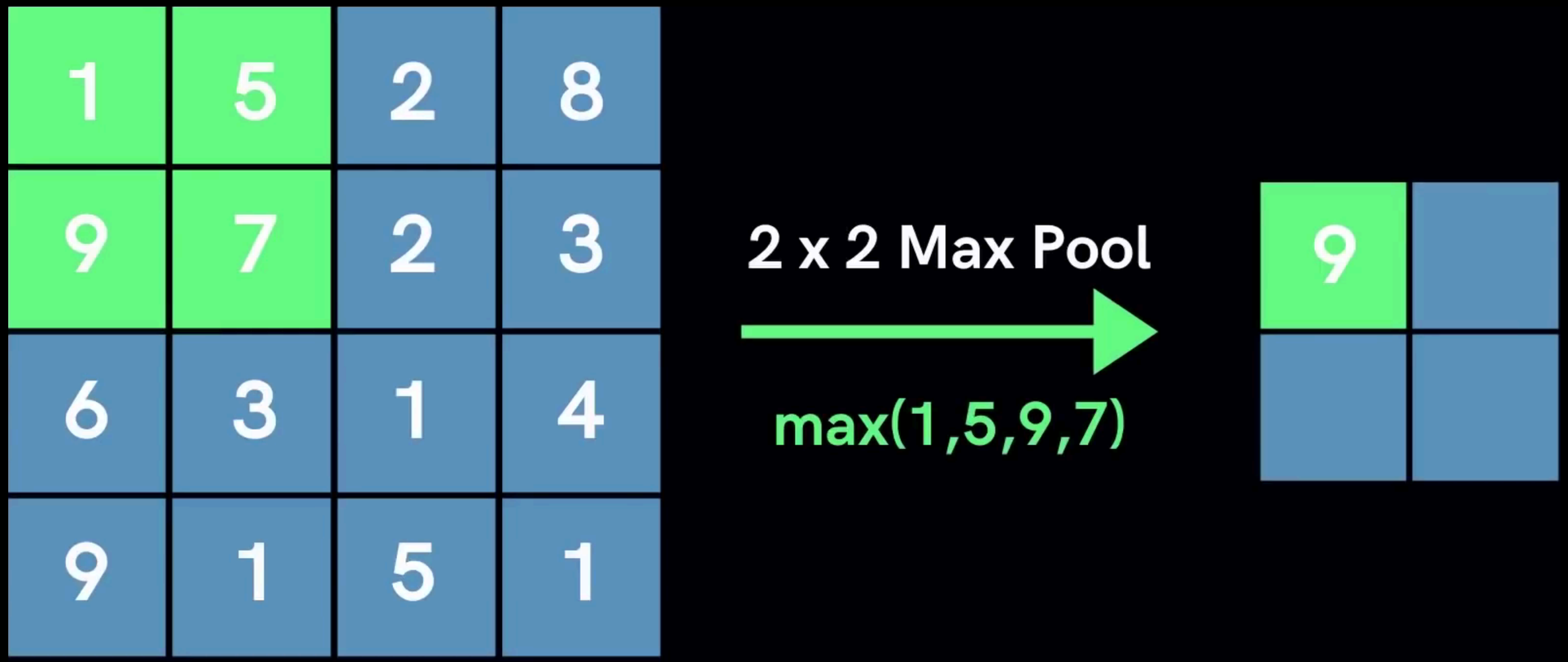
Réseaux de Neurones Convolutif

Regroupement - *pooling*

Réduit le nombre de neurones nécessaires dans les calques suivants

Deux Types : Mise en Commun Max & Min

Réseaux de Neurones Convolutif



Réseaux de Neurones Convolutif

Réseaux de Neurones Convolutif

Convolu l’image

Regroupement du résultat

Répète

Ajoute quelques couches pour aider à classer l’image

Prédiction dans les couches de sortie

Réseaux de Neurones Convolutif

Applications de CNNs

1. Vision par Ordinateur
2. Reconnaissance d’Image
3. Traitement d’Image
4. Segmentation d’Image
5. Analyse Vidéo

Plan

- Introduction à l'apprentissage profond
- Principes de l'apprentissage profond
- Types d’apprentissage
- Régularisation
- Architectures
- Comment créer un modèle ?

Modèle d’Apprentissage Profond - Introduction

1- Rassembler des Données
2- Prétraitement des Données
3- Entraînement du Modèle
4- Evaluation du Modèle
5- Optimisation du Modèle

1- Rassembler des Données

Choisir les bonnes données est essentiel

Mauvaises Données Mauvais Modèle

Faites des hypothèses sur les données dont vous avez besoin

1- Rassembler des Données

Taille

Base de Données IRIS Flowers	=	~ 150 images
Gmail	=	~ 238 millions d’exemples
Google Translate	=	Milliards d’exemples

1- Rassembler des Données

Taille

Pas de taille unique. Dépend beaucoup de l'hétérogénéité des données, du *data augmentation* et du type de tâche.

Montant de données dont vous avez besoin **~ 10 x** Nb de paramètres du modèle

Régression : 10 exemples par variable prédictive

Classification d'Images : 1000 images par classe

1- Rassembler des Données

Qualité

Quelle est la fréquence des erreurs de labels ?

Vos données sont-elles hétérogènes ?

1- Rassembler des Données

archive.ics.uci.edu

1- Rassembler des Données

1- Rassembler des Données

reddit.com

1- Rassembler des Données

openneuro.org

1- Rassembler des Données

2- Prétraitement des Données

a. Fractionnement de l'ensemble de données en sous-ensembles :

- Entrainement : Trouver les paramètres du modèle;
- Validation : Évaluer, optimisation d'hyperparamètres;
- Test : Tester le modèle sur des données non vues

Vous ne pouvez pas diviser l'ensemble de données au hasard !

2- Prétraitement des Données

Fractionnement dépend de :

- Nombre d’échantillons dans les données
- Type de modèle à entraîner

2- Prétraitement des Données

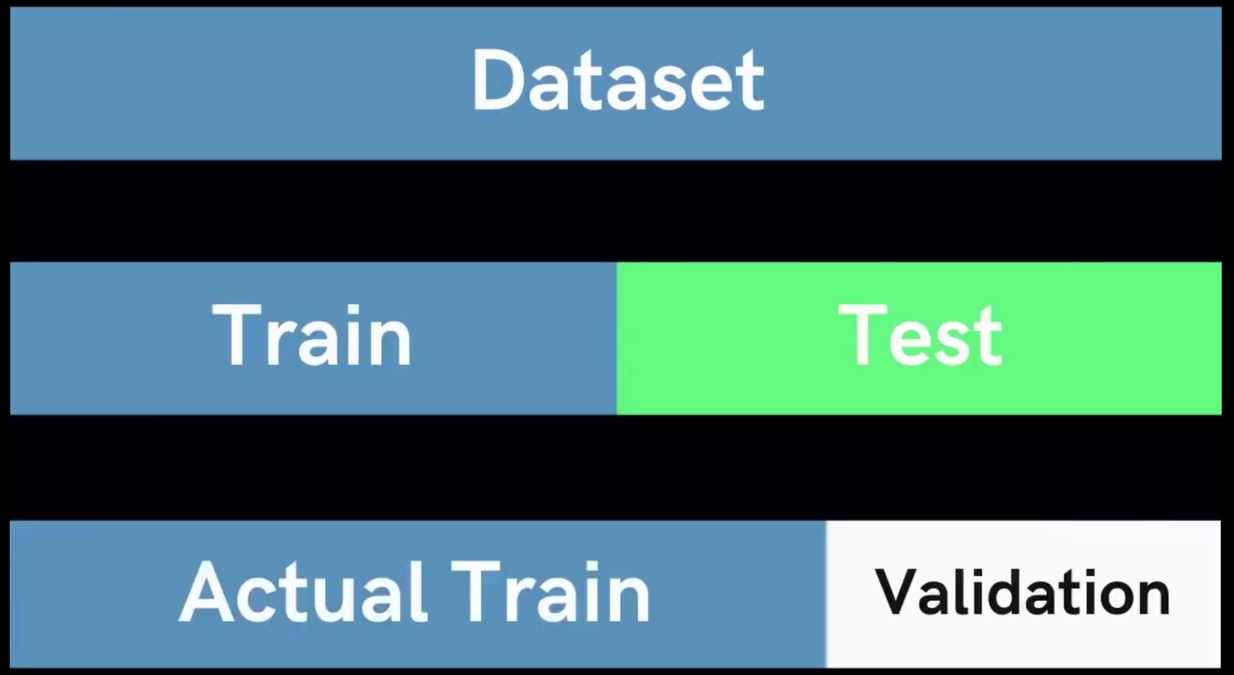
Peu d’hyperparamètres		Petit ensemble de validation
Nombreux hyperparamètres		Grand ensemble de validation
Aucun hyperparamètre		Aucun ensemble de validation requis

2- Prétraitement des Données

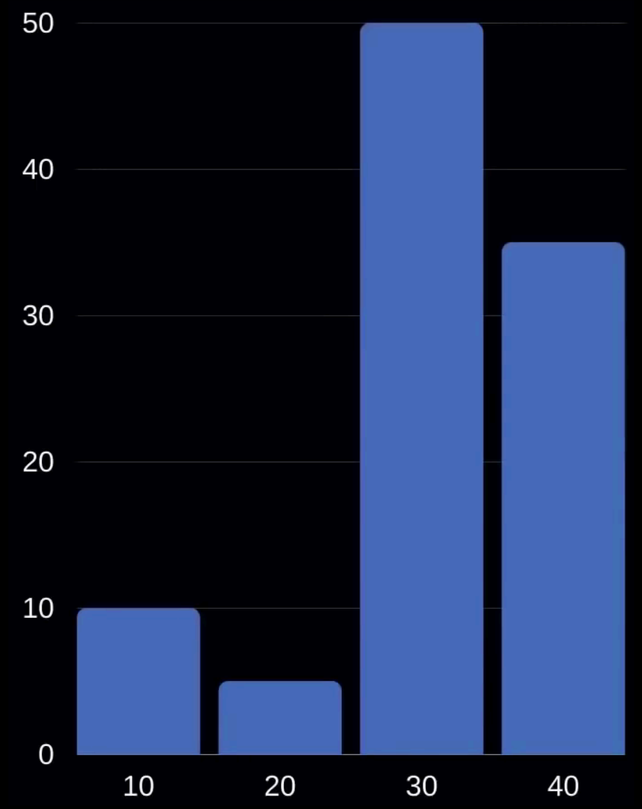
Le rapport de répartition
entraînement-test-validation est
spécifique à votre cas d'utilisation

2- Prétraitement des Données

Validation Croisée



2- Prétraitement des Données



2- Prétraitement des Données

Entraînement pendant les Jours 1 à 39

Test / Validation le Jour 40

2- Prétraitement des Données

Les intervalles de temps sont mieux adaptés
aux grands ensembles de données

2- Prétraitement des Données

b. Formatage

Données dans une base de données

Vous voulez un fichier CSV

Si vous êtes confrontés à une telle situation,
Google est votre meilleur ami !

2- Prétraitement des Données

c. Données Manquantes

Représentées typiquement par ‘NaN’ ou ‘Null’

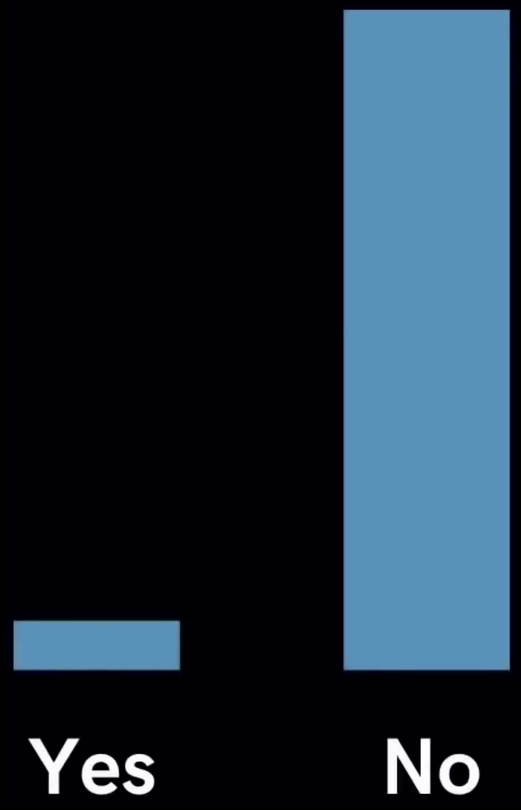
1. Eliminez les caractéristiques qui ont des valeurs manquantes
2. Renseignez les valeurs manquantes

2- Prétraitement des Données

d. Echantillonnage

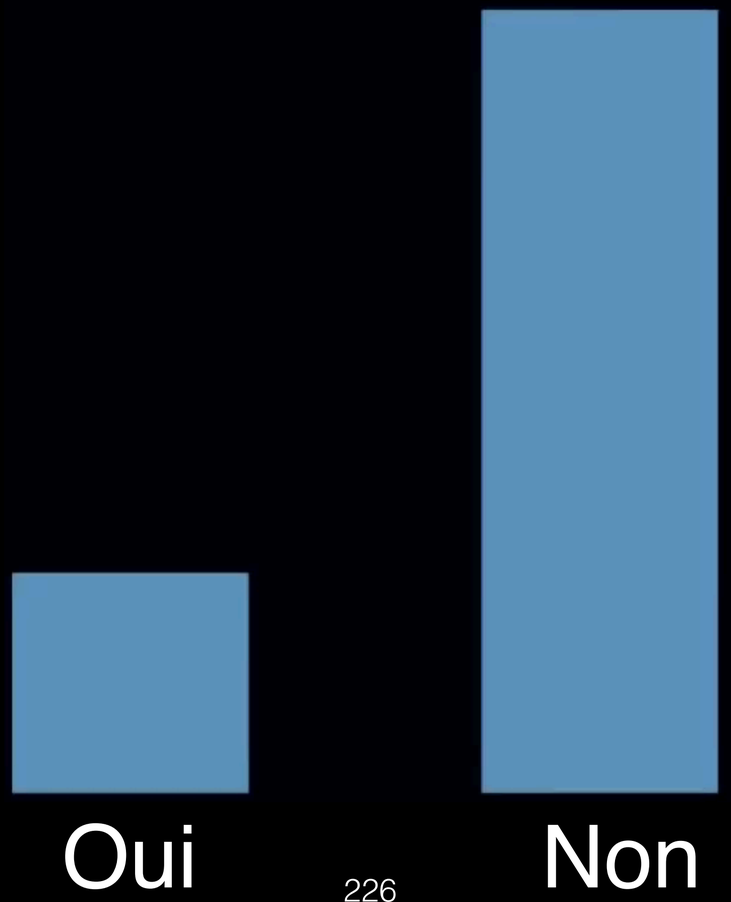
Utilisez un petit échantillon de la Base de Données

2- Prétraitement des Données



2- Prétraitement des Données

Exemple de poids = Poids d'origine x Facteur de sous-Echantillonnage



2- Prétraitement des Données

Convergence plus rapide

Espace disque réduit

La Base de Données est dans un ratio similaire

2- Prétraitement des Données

e. Mise à l'Echelle des Fonctionnalités

Etape cruciale dans l'Apprentissage Profond

1. Normalisation
2. Standardisation

3- Entraînement du Modèle

Flux de données

Propagation en Avant

Fonction de Perte

Propagation Arrière

4- Evaluation du Modèle

Tester un modèle sur un ensemble de validation

Cense être représentatif du modèle dans le monde réel

5- Optimisation du Modèle

a. Réglage des Hyperparamètres

Augmenter le nombre d’époques
Ajuster le Taux d’Apprentissage

Les conditions initiales jouent un rôle important dans la détermination du résultat de votre modèle

5- Optimisation du Modèle

b. Lutter contre le surajustement (*overfitting*)

Obtenir plus de données

Réduire la Taille des Modèles

Régulariser

- Régularisation L1
- Régularisation L2

5- Optimisation du Modèle

c. Augmentation du Nombre de Données

Bon moyen d'augmenter la Base de Données *artificiellement*

Retournement, Flou, Zoom

5- Optimisation du Modèle

d. Abandon (*dropout*)

Laisser tomber certains neurones au hasard

Réduire la codépendance des neurones

Apprentissage Profond

Fin du cours !

Liens 🔗

What's AI

- https://www.youtube.com/channel/UCUzGQrN-lyyc0BWTYoJM_Sg

3Blue1Brown

- https://www.youtube.com/c/3blue1brown

Two Minutes Papers

- https://www.youtube.com/channel/UCbfYPyITQ-7l4upoX8nvctg