

L'IA c'est quoi ?

Chapitre 1

Introduction

Savoirs et compétences :

- Analyser les principes d'intelligence artificielle.
 - Phases d'apprentissage et d'inférence.
 - Réseaux de neurones (couches d'entrée, cachées et de sortie, neurones, biais, poids et fonction d'activation).
- Résoudre un problème en utilisant une solution d'intelligence artificielle :
 - Apprentissage supervisé.
 - Choix des données d'apprentissage.
 - Mise en œuvre des algorithmes (réseaux de neurones).
 - Phases d'apprentissage et d'inférence.

1	Le neurone	2
1.1	Bref historique	2
1.2	Modèle de neurone	2
2	Réseaux de neurones	3
2.1	Modélisation d'un réseau de neurones	3
2.2	Propagation	4
2.3	Rétropropagation	4
3	Algorithmes d'apprentissage	4
4	Définitions	5
4.1	Quelques définitions	5
4.2	Le nerf de la guerre, les données	5
4.3	Méthode de résolution de problèmes d'apprentissage supervisé	5
5	Méthode de résolution d'un	5
6	TP : Synthèse d'un contrôleur piloté par réseau de neurones	6

1 Le neurone

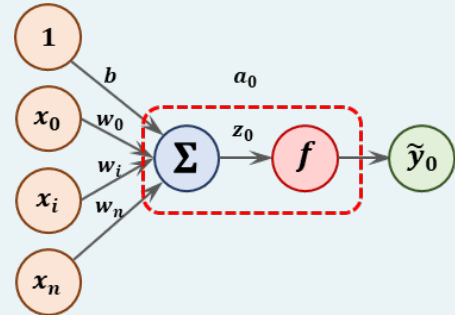
1.1 Bref historique

1.2 Modèle de neurone

Définition — Neurone (ou perceptron).

Prenons la représentation suivante pour un neurone. On note :

- \mathbf{X} le vecteur d'entrée et x_i les données de la couche d'entrée;
- w_i les poids (poids synaptiques);
- b le biais;
- z_0 la somme pondérée des entrées;
- f une fonction d'activation;
- \tilde{y}_0 : la valeur de sortie du neurone.



On a donc, dans un premier temps :

$$z_0 = b + \sum_{i=0}^n w_i x_i.$$


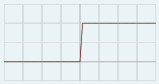


Après la fonction d'activation, on a donc en sortie du neurone :

$$\tilde{y}_0 = f(z_0) = f\left(b + \sum_{i=0}^n w_i x_i\right).$$

R

1. La notation tilde (\tilde{y}_0) vient du fait que la valeur de sortie d'un neurone est une valeur estimée qu'il faudra comparer à y_0 valeur de l'étiquette utilisée pour l'apprentissage supervisé.
2. Par la suite, dans la représentation graphique on ne fera pas apparaître la somme pondérée et la fonction d'activation, mais seulement la valeur de sortie du neurone (notée par exemple a_0).

Définition — Fonction d'activation. Les fonctions d'activation sont des fonctions mathématiques appliquées au signal de sortie (z). Il est alors possible d'ajouter des non linéarités à la somme pondérée. On donne ci-dessous quelques fonctions usuelles :

Identité	Heaviside	Logistique (sigmoïde)	Unité de rectification linéaire (ReLU)
			
$f(x) = x$	$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$	$f(x) = \frac{1}{1 + e^{-x}}$	$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$

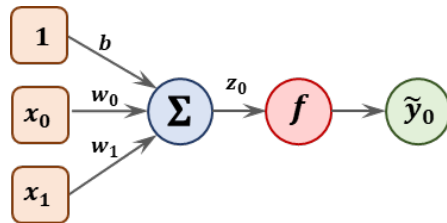
Définition — Biais.

■ Exemple

Prenons un neurone à deux entrées binaires.

Initialisation les poids et le biais avec des valeurs aléatoires : $w_0 = -0,3$, $w_1 = 0,8$ et $b = 0,2$.

On peut donc évaluer l'ensemble des sorties calculable par le neurone.



x_0	x_1	z	Id.	H.	Sig.	ReLu
0	0	0,2	0,2	1	0.549	0,2
0	1	1	1	1	0.731	1
1	0	-0.1	-0.1	0	0.475	0
1	1	0.7	0.7	1	0.668	0.7

2 Réseaux de neurones

<https://playground.tensorflow.org/>

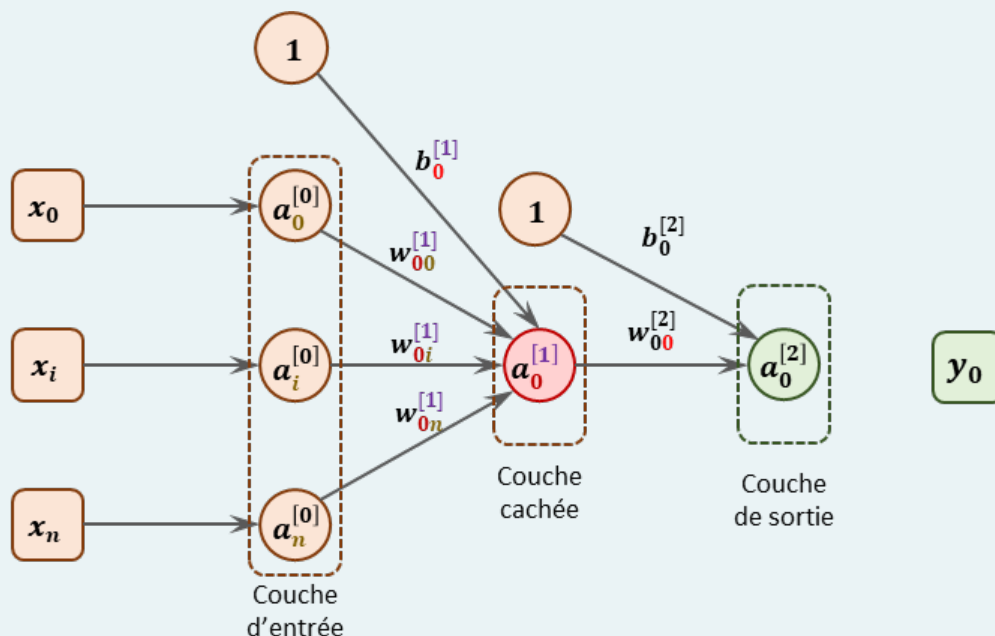
2.1 Modélisation d'un réseau de neurones

Définition — Couches.

Un réseau de neurones va être un ensemble de neurones reliés, par couches, entre eux.

Dans un réseau de neurones **dense** tous les neurones de la couche i seront reliés à tous les neurones de la couche $i + 1$.

- Couche d'entrée : cette couche est une copie de l'ensemble des données d'entrées. Le nombre de neurones de cette couche correspond donc aux nombre de données d'entrées. On note $\mathbf{X} = (x_0, \dots, x_n)$ le vecteur d'entrées.
- Couche cachée (ou couche intermédiaire) : il s'agit d'une couche qui a une utilité intrinsèque au réseau de neurones. Ajouter des neurone dans cette couche (ou ces couches) permet donc d'ajouter de nouveaux paramètres. Pour une couche, la même fonction d'activation est utilisée pour tous les neurones. En revanche la fonction d'activation utilisée peut être différente pour deux couches différentes. Les fonctions d'activations des couches intermédiaires sont souvent non linéaires.
- Couche de sortie : le nombre de neurones de cette couche correspond au nombre de sorties attendues. La fonction d'activation de la couche de sortie est souvent linéaire. On note $\mathbf{Y} = (y_0, \dots, y_y)$ le vecteur des sorties.



En utilisant la loi de comportement du modèle de perceptron, on peut donc exprimer $\mathbf{Y} = \mathcal{F}(\mathbf{X})$ où \mathcal{F} est une fonction dépendant des entrées, des poids et des biais.

Notations :

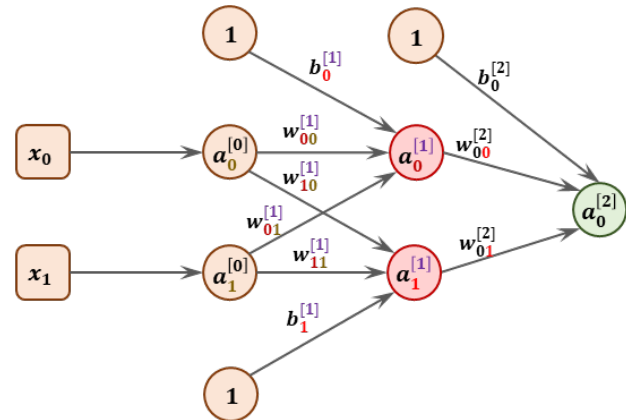
- on note $w_{jk}^{[l]}$ les poids permettant d'aller vers la couche l depuis le neurone k vers le neurone j ;
- $b_j^{[l]}$ le biais permettant d'aller sur le neurone j de la couche l .

■ Exemple

Prenons un réseau de neurones à 3 couches :

- 1 couche d'entrée à 2 neurones ;
- 1 couche cachée à 2 neurones, de fonction d'activation f_1 ;
- 1 couche de sortie à 1 neurone, de fonction d'activation f_2 ;

Initialisation les poids et le biais avec des valeurs aléatoires : $w_0 = -0,3$, $w_1 = 0,8$ et $b = 0,2$.



Il est possible d'écrire que $y_0 = a_0^{[1]} = f_2 \left(b_0^{[2]} + w_{00}^{[2]} a_0^{[1]} + w_{01}^{[2]} a_1^{[1]} \right)$.

Par ailleurs : $a_0^{[1]} = f_1 \left(b_0^{[1]} + w_{00}^{[1]} a_0^{[0]} + w_{01}^{[1]} a_1^{[0]} \right)$ et $a_1^{[1]} = f_1 \left(b_1^{[1]} + w_{10}^{[1]} a_0^{[0]} + w_{11}^{[1]} a_1^{[0]} \right)$.

Au final, on a donc

$$y_0 = a_0^{[1]} = f_2 \left(b_0^{[2]} + w_{00}^{[2]} \left(f_1 \left(b_0^{[1]} + w_{00}^{[1]} a_0^{[0]} + w_{01}^{[1]} a_1^{[0]} \right) \right) + w_{01}^{[2]} \left(f_1 \left(b_1^{[1]} + w_{10}^{[1]} a_0^{[0]} + w_{11}^{[1]} a_1^{[0]} \right) \right) \right)$$

Définition — Paramètres. Les paramètres du réseau de neurones sont les poids et les biais, autant de valeurs que l'entraînement devra déterminer.

Méthode — Calcul du nombre de paramètres – à vérifier.

Soit un jeu de données étiquetées avec n entrées et p sorties.

On construit un réseau possédant ℓ couches et a_ℓ le nombre de neurones de la couche ℓ . Dans ce cas, la première couche est la couche d'entrée ($a_1 = n$) et la dernière couche et la couche de sortie ($a_\ell = p$).

Nombre de poids : $n_w = \sum_{i=1}^{\ell-1} (a_i \times a_{i+1})$.

Nombre de biais : $n_b = \sum_{i=2}^{\ell} (a_i)$.

Au final, le nombre total de paramètre à calculer est donné par $N = n_w + n_b$.

Objectif Soit un jeu de données étiquetées avec n entrées et p sorties. .

L'objectif de l'apprentissage d'un réseau de neurones est de déterminer les valeurs de l'ensemble des poids et des biais de telle sorte que, une fois que le réseau est entraîné,

Définition — Hyper paramètres. Nombre de couches

2.2 Propagation

2.3 Rétropropagation

■ Définition — Quantification de l'erreur.

3 Algorithmes d'apprentissage

4 Définitions

- Data scientist
- machine learning
- deep learning
- réseaux de neurons
- régression
- data mining
- bigdata
- données continues, données discrètes, données nominales, données ordinales, données (semi-)structurées et non structurées
- algorithmes supervisés, non supervisés
- algorithmes de régression et de classification

Définition — Machine Learning – Apprentissage automatique – Wikipedia. Champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune.

La première phase de l'apprentissage consiste à estimer un modèle à partir de données, appelées observations, qui sont disponibles en nombre fini, lors de la phase de conception du système.

La seconde phase correspond à la mise en production : le modèle étant déterminé, de nouvelles données peuvent alors être soumises afin d'obtenir le résultat correspondant à la tâche souhaitée.

Définition — Apprentissage supervisé – Apprentissage non supervisé – Apprentissage par renforcement – Wikipedia. Si les données sont étiquetées (c'est-à-dire que la réponse à la tâche est connue pour ces données), il s'agit d'un apprentissage supervisé. On parle de :

- classification ou de classement si les étiquettes sont discrètes ;
- régression si elles sont continues.

Si le modèle est appris de manière incrémentale en fonction d'une récompense reçue par le programme pour chacune des actions entreprises, on parle d'apprentissage par renforcement.

Dans le cas le plus général, sans étiquette, on cherche à déterminer la structure sous-jacente des données (qui peuvent être une densité de probabilité) et il s'agit alors d'apprentissage non supervisé.

4.1 Quelques définitions

Définition Intelligence Artificielle, première approche

4.2 Le nerf de la guerre, les données

4.3 Méthode de résolution de problèmes d'apprentissage supervisé

1. Choix des données.
2. Normalisation des données.
3. Séparation des données? (entraînement, test, validation)
4. Choix de la méthode d'entraînement (choix d'un modèle, en fonction du type de données, choix des paramètres du modèle)
5. Entraînement du modèle
6. Test du modèle
7. Observation des métriques et visualisation des résultats

5 Méthode de résolution d'un

1 cours Sur les réseau de neurones

6 TP : Synthèse d'un contrôleur piloté par réseau de neurones

TP identification de la boucle ouverte

- Comparaison modèle (a)causal / réel
- Comparaison modèle ANN / réel

TP Synthèse du contrôleur PID à partir du modèle To do XP

TP synthèse du contrôleur et implémentation sur une cible. Déploiement sur control X?

Exemples : <https://makina-corpus.com/blog/metier/2017/initiation-au-machine-learning-avec-python-p>

Références

[1] Éric Biernat et Michel Lutz. *Data science : fondamentaux et études de cas*. Eyrolles.