
Intelligence artificielle

Cours 1 - Les réseaux de neurones



Table des matières

Neurones	2
Biologique	2
Artificiel	2
Pourquoi les réseaux de neurones sont si populaires ?	4
Les éléments d'un entraînement de réseau de neurones	4
Un réseau de neurones	4
La fonction de coût	5
L'optimiseur	6
Les étapes de l'entraînement	7

Neurones

Biologique

Un neurone biologique est une cellule nerveuse qui reçoit, traite et transmet des informations sous forme de signaux électriques.

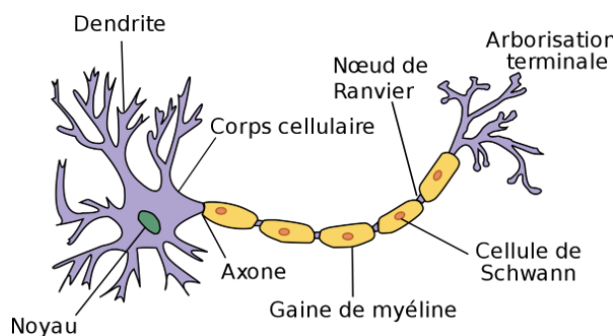


FIGURE 1 – Schéma d'un neurone biologique

Un neurone transmettra un signal électrique à d'autres neurones en fonction des signaux électriques qu'il reçoit. Différents neurones auront des réactions différentes au même signal électrique.

D'une manière simplifiée, un neurone enverra un signal électrique à ses voisins en sortie si la somme des signaux électriques reçus est supérieure à un certain seuil.

Artificiel

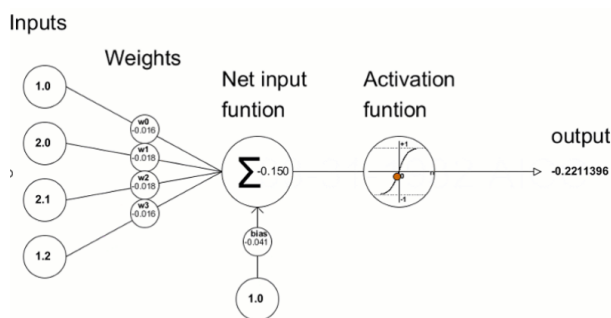


FIGURE 2 – Schéma d'un neurone artificiel "informatique"

En informatique, nous représentons un neurone par une fonction f qui prend en entrée un vecteur \vec{x} et qui renvoie un scalaire y . Cette fonction f dépend de paramètres internes au neurone : un vecteur de poids \vec{w} et un biais b .

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

où σ est une fonction d'activation et w_i le poids de \vec{w} associé à l'entrée x_i .

Les fonctions d'activation utilisées sont généralement des fonctions non linéaires, par exemple :

- La fonction sigmoïde : $\sigma(x) = \frac{1}{1+e^{-x}}$
- La fonction tangente hyperbolique : $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- La fonction ReLU : $\sigma(x) = \max(0, x)$

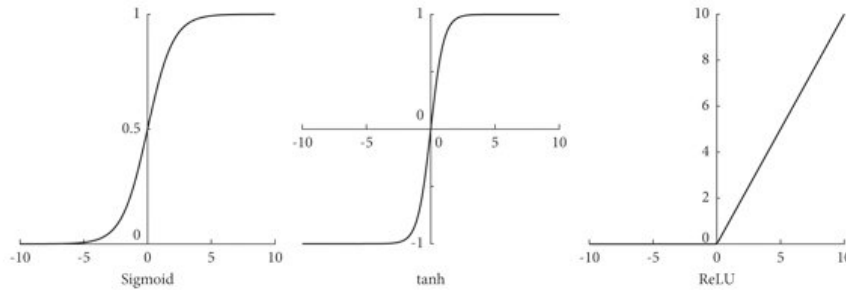


FIGURE 3 – Différentes fonctions d'activation

Exemple numérique

Soit un neurone avec les poids $\vec{w} = [w_1, w_2, w_3, w_4, w_5] = [6, 7, 8, 9, 10]$ et un biais $b = 11$ qui utilise la fonction d'activation ReLU.

Si on lui donne en entrée le vecteur $\vec{x} = [1, 2, 3, 4, 5]$, on obtiendra en sortie :

$$\sigma(\vec{x} \cdot \vec{w} + b) = \sigma(1 \cdot 6 + 2 \cdot 7 + 3 \cdot 8 + 4 \cdot 9 + 5 \cdot 10 + 11) = \sigma(141) = \max(0, 141) = 141$$

Par contre, si on lui donne en entrée le vecteur $\vec{x} = [-1, -1, -1, -1, -1]$, on obtiendra en sortie :

$$\sigma(\vec{x} \cdot \vec{w} + b) = \sigma(-1 \cdot 6 - 1 \cdot 7 - 1 \cdot 8 - 1 \cdot 9 - 1 \cdot 10 + 11) = \sigma(-29) = \max(0, -29) = 0$$

On peut voir que cette fonction réagira différemment en fonction des entrées.

En informatique, un réseau de neurones est simplement un ensemble de neurones interconnectés.

Pourquoi les réseaux de neurones sont si populaires ?

Les réseaux de neurones sont très populaires, car ils sont très efficaces pour approximer des fonctions complexes en utilisant « uniquement » des données.

Les réseaux de neurones doivent être entraînés afin de pouvoir résoudre un problème. C'est-à-dire que les paramètres internes du réseau de neurones vont être optimisés en fonction des données d'entraînement.

Si nous désirons entraîner un réseau de neurones pour qu'il puisse distinguer des images de chats et des images de chiens, nous allons lui donner en entrée des images de chats et des images de chiens et nous allons lui dire si l'image est un chat ou un chien.

Ce processus va être répété de nombreuses fois, car le processus d'entraînement est itératif.

Nous allons désormais voir les éléments permettant d'entraîner un réseau de neurones.

Les éléments d'un entraînement de réseau de neurones

Un réseau de neurones

La structure du réseau est à définir, combien de couches, combien de neurones par couches? Autant d'hyperparamètres à définir.

Des hyperparamètres sont des paramètres qui ne sont pas modifiés par le réseau de neurones mais qui sont définis par l'utilisateur.

De manière générale, plus un réseau de neurones contient de paramètres, plus il sera capable d'approximer des fonctions complexes.

Le réseau de neurones sera vu comme une fonction f qui prend en entrée un vecteur X et qui renvoie un vecteur de sortie \hat{Y} .

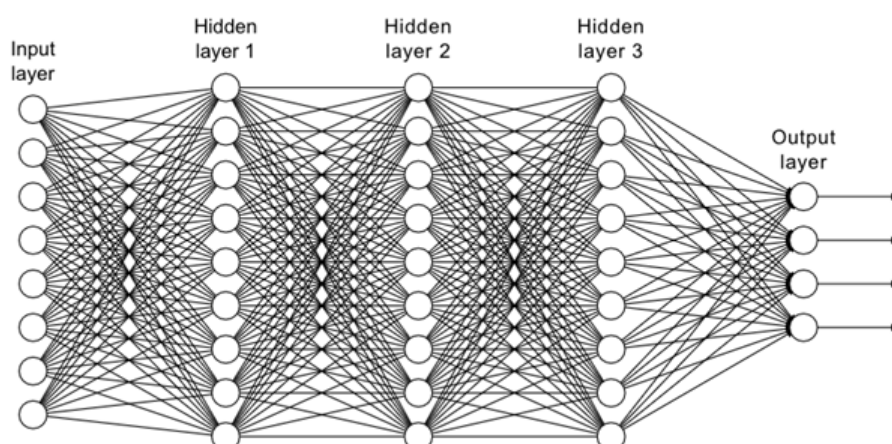


FIGURE 4 – Un réseau de neurones

Le plus important est d'avoir une architecture compatible avec les dimensions des données d'entraînement,

que ce soit les données d'entrée ou les données de sortie.

Les données

Les données sont les exemples que l'on va donner au réseau de neurones.

X est un ensemble de données d'entrée et Y est un ensemble de données de sortie.

Plus le réseau de neurones aura de données d'entraînement, plus il sera capable de résoudre des problèmes complexes.

Plus le réseau de neurones possède de paramètres, plus il aura besoin de données d'entraînement.

Les données d'entraînement sont généralement divisées en 3 ensembles :

- L'ensemble d'entraînement (training set) : Cet ensemble est utilisé pour entraîner le réseau de neurones.
- L'ensemble de validation (validation set) : Cet ensemble est utilisé pour analyser les performances du réseau de neurones pendant l'entraînement.
- L'ensemble de test (test set) : Cet ensemble est utilisé pour analyser les performances du réseau de neurones après l'entraînement.

Pourquoi ne pas utiliser l'ensemble de test pour l'entraînement ?

Car les réseaux ont tendance à mémoriser les données d'entraînement, donc si on utilise l'ensemble de test pour l'entraînement, le réseau de neurones sera très performant sur l'ensemble de test, mais nous ne pourrions plus savoir s'il sera capable de généraliser à de nouvelles données.

La fonction de coût

La fonction de coût permet de mesurer l'erreur entre la sortie \hat{Y} du réseau de neurones et la sortie attendue Y . Il existe plusieurs fonctions de coût, comme l'erreur quadratique moyenne (mean squared error), l'erreur absolue moyenne (mean absolute error), l'entropie croisée (cross entropy), etc.

L'erreur quadratique moyenne est la moyenne des carrés des différences entre les valeurs prédites et les valeurs réelles.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

L'erreur absolue moyenne est la moyenne des valeurs absolues des différences entre les valeurs prédites et les valeurs réelles.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

L'entropie croisée (cross entropy), qui est souvent utilisée pour les problèmes de classification, est une mesure de la différence entre deux distributions de probabilités.

$$cross_entropy = - \sum_{i=1}^n y_i \cdot \log(\hat{y}_i) \quad (4)$$

Au départ, les paramètres du réseau de neurones sont initialisés aléatoirement, la sortie du réseau de neurones sera aléatoire et donc la fonction de coût sera très élevée. L'objectif de l'entraînement est de modifier les paramètres du réseau de neurones afin de minimiser la fonction de coût.

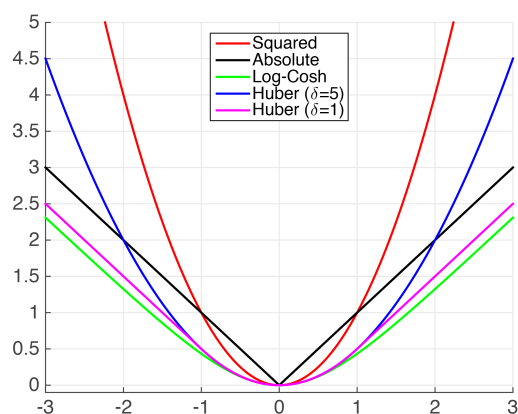


FIGURE 5 – Visualisation de différentes fonctions de coût

La fonction de coût permettra d'analyser si le réseau de neurones est en train de s'améliorer ou non.

L'optimiseur

L'optimiseur permet de modifier les paramètres du réseau de neurones afin de minimiser la fonction de coût.

Il existe plusieurs optimiseurs, le plus connu est la descente de gradient.

Dans l'idée de la descente de gradient, on va modifier les paramètres du réseau de neurones dans le sens opposé du gradient de la fonction de coût.

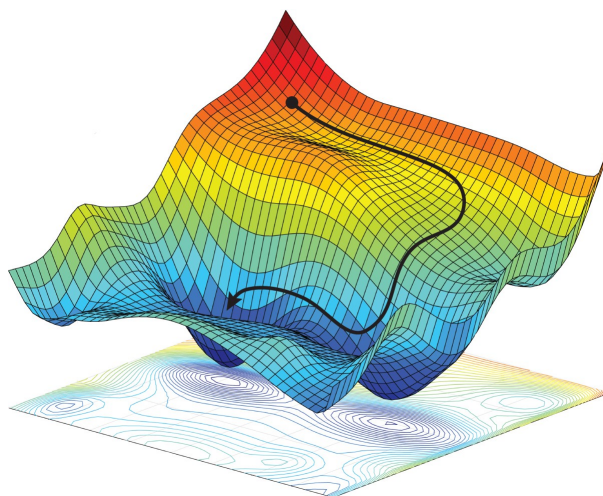


FIGURE 6 – Visualisation d'une descente de gradient

$$\vec{w} = \vec{w} - \alpha \nabla_{\vec{w}} MSE \quad (5)$$

Où α est le taux d'apprentissage.

Les étapes de l'entraînement

Les étapes de l'entraînement sont les suivantes :

1. On donne au réseau de neurones un exemple d'entrée X et on calcule la sortie \hat{Y} .
2. On calcule la fonction de coût $LOSS(Y, \hat{Y})$.
3. On calcule le gradient de la fonction de coût par rapport aux paramètres du réseau de neurones.
4. On modifie les paramètres du réseau de neurones dans le sens opposé du gradient de la fonction de coût.

Une époque (epoch) est une itération sur l'ensemble des données d'entraînement, généralement, on effectue plusieurs époques.

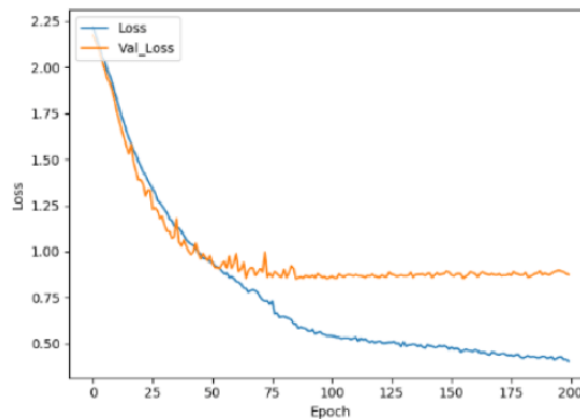


FIGURE 7 – Evolution de la fonction de coût en fonction du nombre d'époques; à partir de l'époque 50, la fonction de coût sur l'ensemble de validation stagne, alors que celle d'entraînement continue de diminuer.