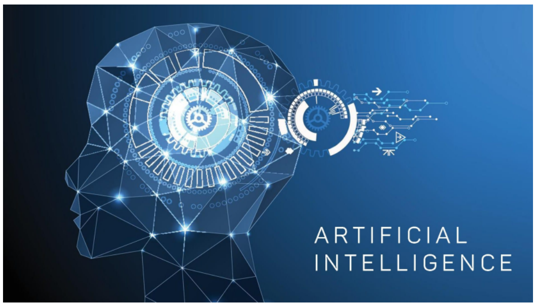
Chapitre 2 : Le neurone formel



Pr. Mustapha BOURAHLA, Département d'Informatique, Université de M'Sila, Contact : mustapha.bourahla@univ-msila.dz

Mars 2021
v 1.0

Table des matières

Introduction

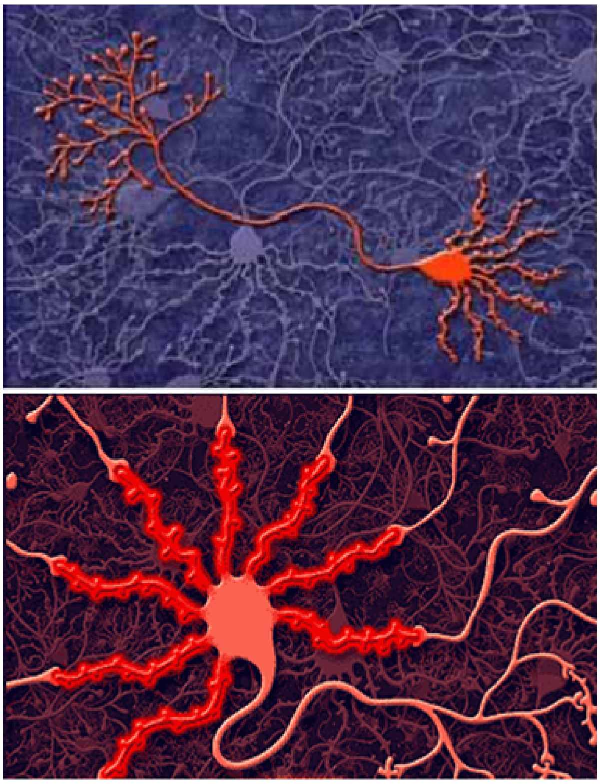
Présentation du neurone formel

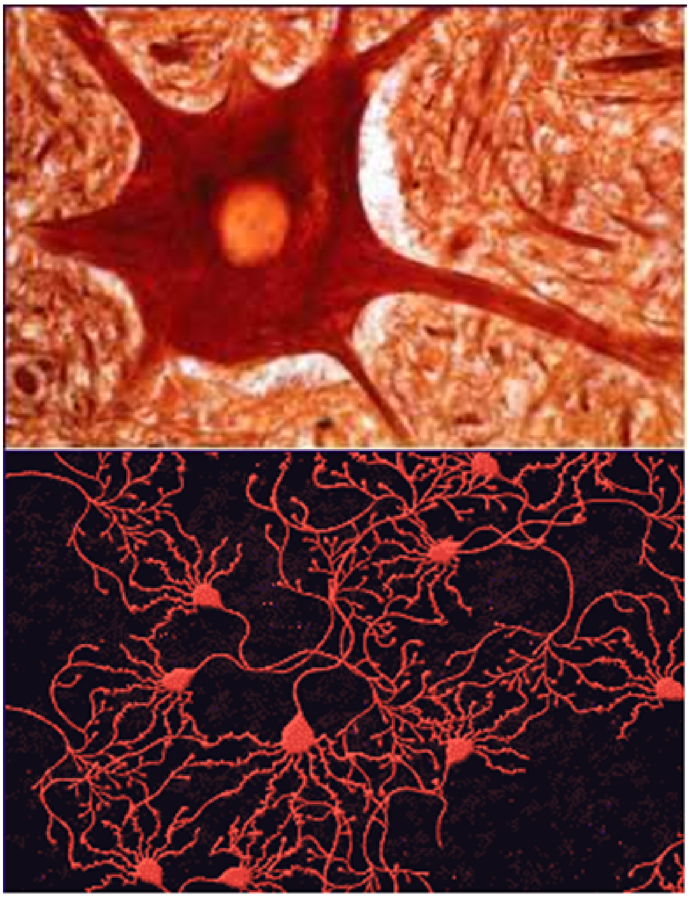
Définition du neurone formel

I

Le neurone biologique

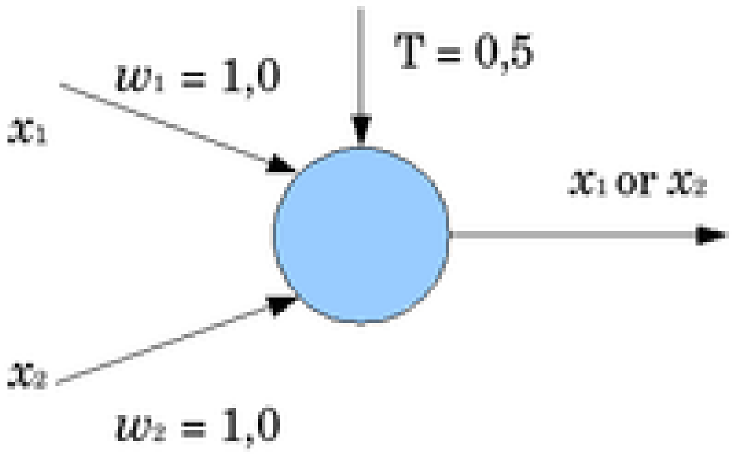
La notion du neurone formel est inspirée des neurones et leurs liaisons (les synapses) du cerveau humain





Neurone formel

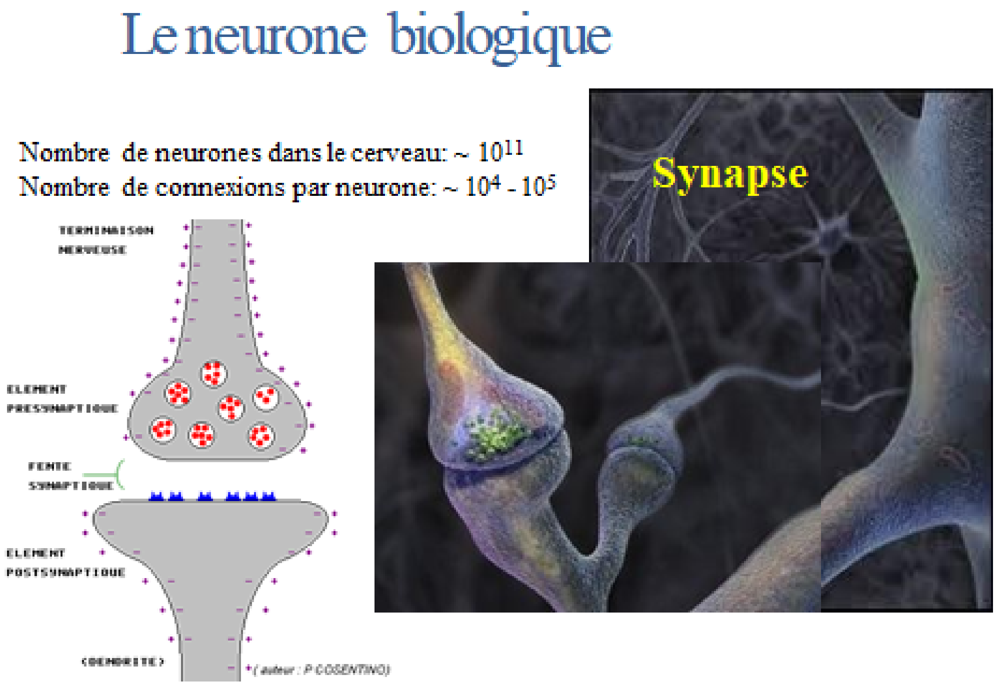
- Un neurone formel est une représentation mathématique et informatique d'un neurone biologique.
- Le neurone formel possède généralement plusieurs entrées et une sortie qui correspondent respectivemen aux dendrites et au cône d'émergence du neurone biologique (point de départ de l'axone).
- Les actions excitatrices et inhibitrices des synapses sont représentées, la plupart du temps, par des coefficients numériques (les poids synaptiques) associés aux entrées.
- Les valeurs numériques de ces coefficients sont ajustées dans une phase d'apprentissage.
- Dans sa version la plus simple, un neurone formel calcule la somme pondérée des entrées reçues, puis applique à cette valeur une fonction d'activation, généralement non linéaire. La valeur finale obtenue est la sortie du neurone.
- Le neurone formel est l'unité élémentaire des réseaux de neurones artificiels dans lesquels il est associé à ses semblables pour calculer des fonctions arbitrairement complexes, utilisées pour diverses applications en intelligence artificielle.
- Mathématiquement, le neurone formel est une fonction à plusieurs variables et à valeurs réelles.


T = 0,5
w_1 = 1,0
x_1
x_1 or x_2
x_2
w_2 = 1,0

II

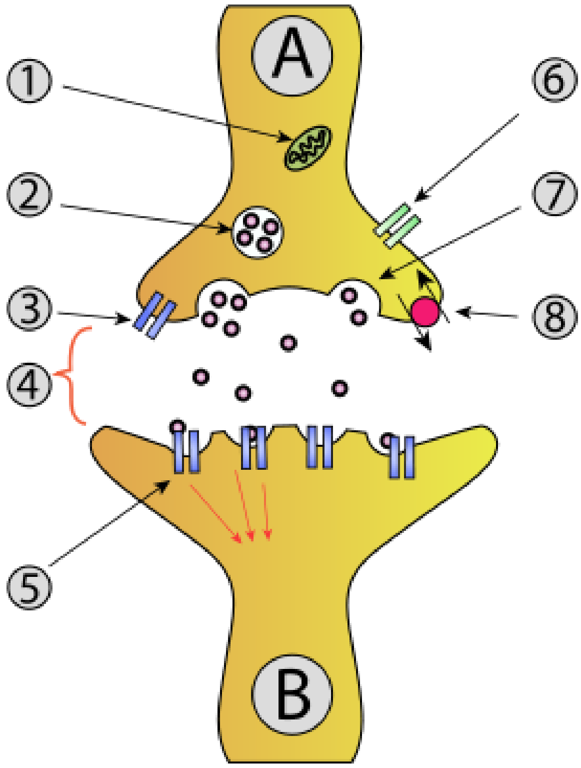
Formulation mathématique du neurone formel

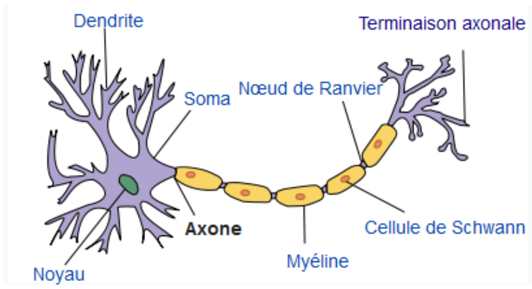
La synapse chimique est la plus fréquente des synapses du système nerveux. Ce type de synapse transmet le signal nerveux d'un neurone à un autre en utilisant un neurotransmetteur



Transmission chimique du neurone A (émetteur) au neurone B (récepteur)

1. Mitochondrie
2. Vésicule synaptique avec des neurotransmetteurs
3. Autorécepteur
4. Fente synaptique avec neurotransmetteur libéré (ex : sérotonine ou dopamine)
5. Récepteurs postsynaptiques activés par neurotransmetteur (induction d'un potentiel postsynaptique)
6. Canal calcium
7. Exocytose d'une vésicule
8. Neurotransmetteur recapturé





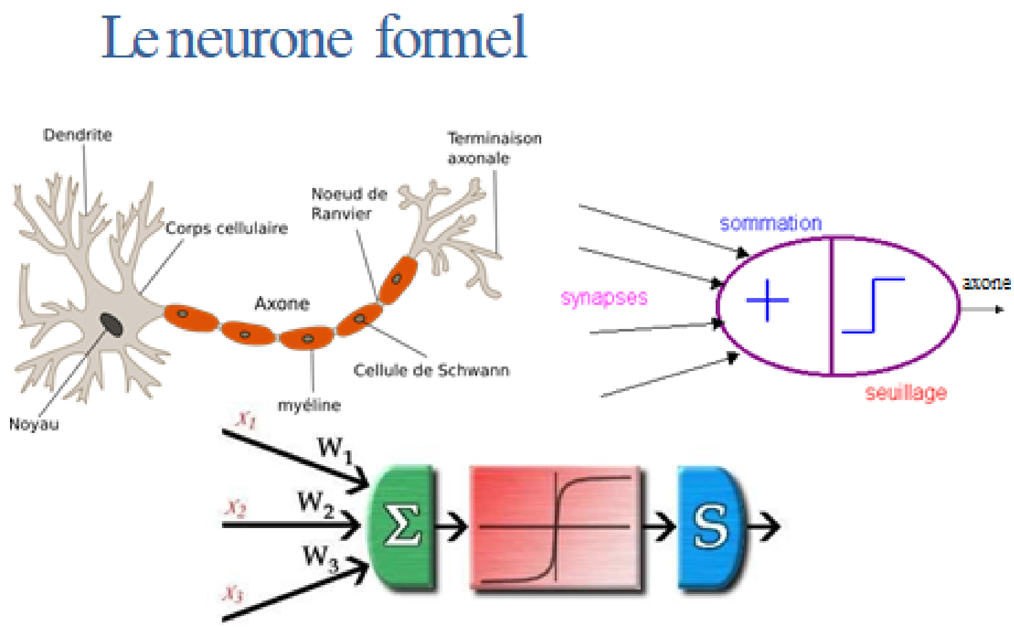
L'axone, ou fibre nerveuse, est le prolongement du neurone qui conduit le signal électrique du corps cellulaire vers les zones synaptiques.

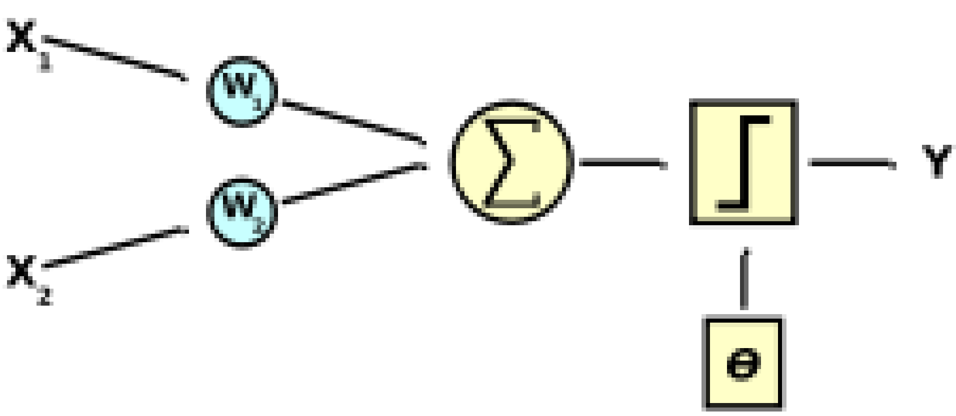
Formulation mathématique

- On considère le cas général d'un neurone formel à m entrées, auquel on doit donc soumettre les m grandeurs numériques (ou signaux, ou encore stimuli) notées x_1 à x_m.
- Un modèle de neurone formel est une règle de calcul qui permet d'associer aux m entrées une sortie : c'es donc une fonction à m variables et à valeurs réelles.
- Neurone formel avec 2 entrées et une fonction d'activation à seuil
- Dans le modèle à chaque entrée est associé un poids synaptique, c'est-à-dire une valeur numérique notée de w_1 pour l'entrée 1 jusqu'à w_m pour l'entrée m.
- La première opération réalisée par le neurone formel consiste en une somme des grandeurs reçues en entrées, pondérées par les coefficients synaptiques, c'est-à-dire la somme $w_1 x_1 + \ldots + w_m x_m = \sum_{j=1}^{m} w_j x_j$.
  À cette grandeur s'ajoute un seuil w_0. Le résultat est alors transformé par une fonction d'activation non linéaire (parfois appelée fonction de sortie), φ . La sortie associée aux entrées x_1 à x_m est ainsi donnée par
  $\varphi\left(w_0 + \sum_{j=1}^{m} w_j x_j\right)$,

qu'on peut écrire plus simplement :
$\varphi\left(\sum_{j=0}^{m} w_j x_j\right)$,
en ajoutant au neurone une entrée fictive x_0 fixée à la valeur 1.

- Dans la formulation, la fonction d'activation est la fonction de Heaviside (fonction en marche d'escalier), dont la valeur est 0 ou 1.
- Dans ce cas, on préfère parfois définir la sortie par la formule suivante $\varphi\left(\sum_{j=1}^{m} w_j x_j - w_0\right)$, qui justifie le nom de seuil donné à la valeur w_0.
- En effet, si la somme $\sum_{j=1}^{m} w_j x_j$ dépasse w_0 la sortie du neurone est 1, alors qu'elle vaut 0 dans le cas contraire : w_0 est donc le seuil d'activation du neurone, si on considère que la sortie 0 correspond à un neurone « éteint ».



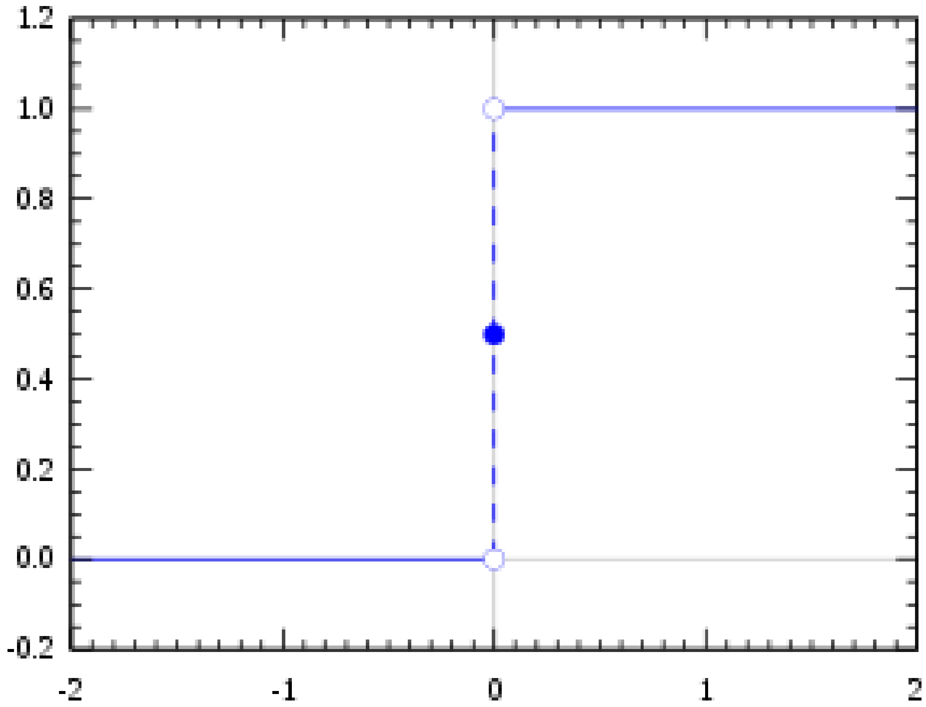


Fonction de Heaviside

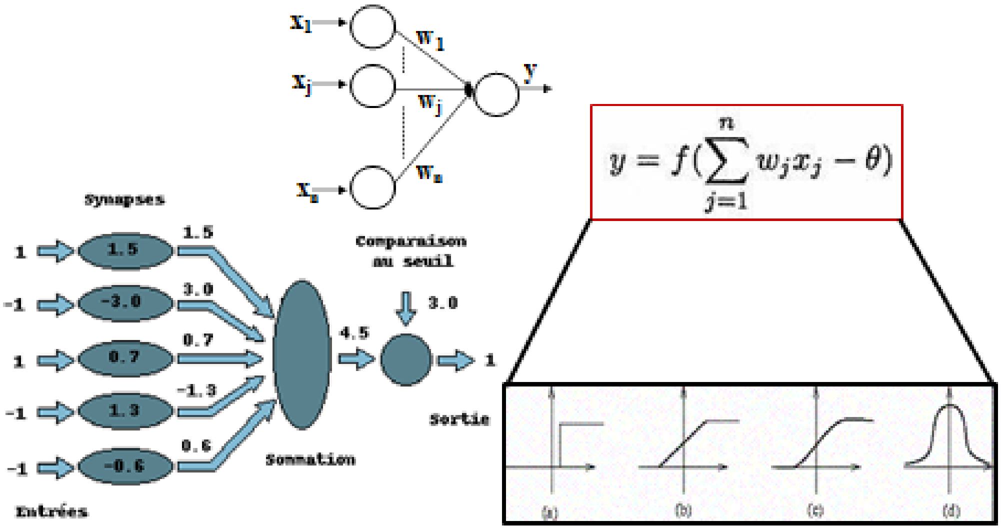
En mathématiques, la fonction de Heaviside (également fonction échelon unité, fonction marche d'escalier), du nom d'Oliver Heaviside, est la fonction indicatrice de $\mathbb{R}^+$.

C'est donc la fonction H (discontinue en 0) prenant la valeur 1 pour tous les réels positifs et la valeur 0 pour les réels strictement négatifs :

$$\forall x \in \mathbb{R},\ H(x) = \begin{cases} 0 & \text{si} \quad x < 0 \\ Undef & \text{si} \quad x = 0 \\ 1 & \text{si} \quad x > 0. \end{cases}$$



Le neurone formel



$$y = f(\sum_{j=1}^{n} w_j x_j - \theta)$$

21

Variantes du neurone formel

III

Variantes du neurone

- La plupart des neurones formels utilisés actuellement sont des variantes du neurone dans lesquelles la fonction de Heaviside est remplacée par une autre fonction d'activation.
- Les fonctions les plus utilisées sont :

  1. la fonction sigmoïde ;
  2. la fonction tangente hyperbolique ;
  3. la fonction identité ;
  4. la fonction rectifieur : $f(x) = \max(0, x)$;La lissage de la moyenne ou $f(x, y) = \frac{1}{2}\left(\frac{x+y}{2} + x\right), x =$ poids du synapse actuel, $y = x$ attendu
  5. dans une moindre mesure, certaines fonctions linéaires par morceaux.

- Ces choix sont motivés par des considérations théoriques et pratiques issues de la combinaison des neurones formels en un réseau de neurones formels.

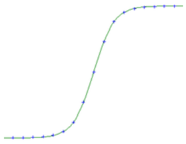
Propriétés importantes de la fonction d'activation

- Les propriétés de la fonction d'activation influent en effet sur celle du neurone formel et il est donc important de bien choisir celle-ci pour obtenir un modèle utile en pratique.
- Quand les neurones sont combinés en un réseau de neurones formels, il est important par exemple que la fonction d'activation de certains d'entre eux ne soit pas un polynôme sous réserve de limiter la puissance de calcul du réseau obtenu.
- Un cas caricatural de puissance limitée correspond à l'utilisation d'une fonction d'activation linéaire, comme la fonction identité : dans une telle situation le calcul global réalisé par le réseau est lui aussi linéaire et il est donc parfaitement inutile d'utiliser plusieurs neurones, un seul donnant des résultats strictement équivalents.
- Cependant, les fonctions de type sigmoïde sont généralement bornées. Dans certaines applications, il est important que les sorties du réseau de neurones ne soient pas limitées a priori : certains neurones du réseau doivent alors utiliser une fonction d'activation non bornée. On choisit généralement la fonction identité.
- Il est aussi utile en pratique que la fonction d'activation présente une certaine forme de régularité.
- Pour calculer le gradient de l'erreur commise par un réseau de neurones, lors de son apprentissage, il faut que la fonction d'activation soit dérivable.
- Pour calculer la matrice hessienne de l'erreur, ce qui est utile pour certaines analyses d'erreur, il faut que la fonction d'activation soit dérivable deux fois.

- Comme elles comportent généralement des points singuliers, les fonctions linéaires par morceaux sont relativement peu utilisées en pratique.

La fonction sigmoïde

- La fonction sigmoïde (aussi appelée fonction logistique ou fonction en S) est définie par : $f_{sig}(x) = \dfrac{1}{1 + e^{-x}}$

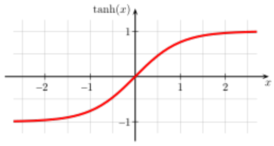


- Elle possède plusieurs propriétés qui la rendent intéressante comme fonction d'activation.
- Elle n'est pas polynomiale
- Elle est indéfiniment continûment dérivable et une propriété simple permet d'accélérer le calcul de sa dérivée. Ceci réduit le temps calcul nécessaire à l'apprentissage d'un réseau de neurones. On a en effet $\dfrac{d}{dx} f_{sig}(x) = f_{sig}(x)\left(1 - f_{sig}(x)\right)$.
- On peut donc calculer la dérivée de cette fonction en un point de façon très efficace à partir de sa valeur en ce point.
- De plus, la fonction sigmoïde renvoie des valeurs dans l'intervalle $[0; 1]$, ce qui permet d'interpréter la sortie du neurone comme une probabilité.
- Elle est aussi liée au modèle de régression logistique et apparaît naturellement quand on considère le problème de la séparation optimale de deux classes de distributions gaussiennes avec la même matrice de covariance.
- Cependant, les nombres avec lesquels travaillent les ordinateurs rendent cette fonction difficile à programmer.
- En effet, $e^{-60} \simeq 10^{-26}$, et avec la précision des nombres à virgule des ordinateurs, $1 - 10^{-26} = 1$, e donc $f_{sig}(60) = 1$.
- Lorsque vous essayez de coder un réseau de neurones en commettant cette erreur, après apprentissage, quelles que soient les valeurs que vous mettrez en entrée de votre réseau, vous obtiendrez toujours le même résultat.
- Il existe heureusement de nombreuses solutions pour tout de même utiliser $f_{sig}(x) = \dfrac{1}{1 + e^{-x}}$ dans votre programme, et certaines d'entre elles sont dépendantes d'un langage.
  Elles reviennent toutes à utiliser une représentation des nombres flottants, plus précise que l'IEEE754.
  Parfois, normaliser les entrées entre 0 et 1 peut suffire à régler le problème.

La fonction tangente hyperbolique

- La fonction tangente hyperbolique, définie par $\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ est aussi très utilisée en pratique, car elle partage avec la fonction sigmoïde certaines caractéristiques pratiques :
  - non polynomiale
  - indéfiniment continûment dérivable
  - calcul rapide de la dérivée par la formule $\dfrac{\mathrm{d}}{\mathrm{d}x}\tanh(x) = 1 - (\tanh(x))^2$

- C'est, à une transformation linéaire près, la fonction sigmoïde : $\tanh(x) = 1 - 2\frac{1}{1 + e^{-2x}}$



Autres neurones formels

IV

Autres neurones formels

- Le développement des réseaux de neurones artificiels a conduit à l'introduction de modèles différents.
- La motivation n'était pas de mieux représenter les neurones réels, mais plutôt d'utiliser les propriétés de certaines constructions mathématiques pour obtenir des réseaux plus efficaces (par exemple avec un apprentissage plus simple, ou encore utilisant moins de neurones).

Neurone à base radiale

- Un neurone à base radiale est construit à partir d'une fonction du même nom. Au lieu de réaliser une somme pondérée de ses entrées, un tel neurone compare chaque entrée à une valeur de référence et produit une sortie d'autant plus grande (proche de 1) que les entrées sont proches des valeurs de références.
- Chaque entrée x_i est donc associée à une valeur c_i. La comparaison entre les deux jeux de valeurs se fait généralement au sens de la norme euclidienne. Plus précisément, le neurone commence par calculer la grandeur suivante $||\mathbf{x} - \mathbf{c}|| = \sqrt{\sum_{i=1}^{m} (x_i - c_i)^2}$
- Il est parfaitement possible d'utiliser une autre norme que la norme euclidienne, et plus généralement une distance quelconque, pour comparer les vecteurs $\mathbf{x} = (x_1, \ldots, x_m)$ et $\mathbf{c} = (c_1, \ldots, c_m)$.
- Le neurone transforme ensuite la valeur obtenue grâce à une fonction d'activation. Sa sortie est donc finalement donnée par $\varphi\left(||\mathbf{x} - \mathbf{c}||\right)$.
- La fonction calculée par le neurone est dite à base radiale car elle possède une symétrie radiale autour du point de référence $\mathbf{c}$: si on réalise une rotation quelconque autour de ce point, la sortie du neurone reste inchangée.
- En pratique, il est très courant d'utiliser une fonction d'activation gaussienne définie par $\varphi(u) = \exp\left(-\beta u^2\right)$.
- Le paramètre β peut être interprété comme l'inverse de la sensibilité du neurone : plus il est grand, plus il faut que les entrées soient proches des valeurs de références pour que la sortie du neurone soit proche de 1.

Neurone Sigma-Pi

- Un neurone Sigma-Pi est obtenu en remplaçant la somme pondérée des entrées du modèle par une somme pondérée de produits de certaines entrées.
- Considérons par exemple le cas de deux entrées. La sortie d'un neurone s'écrit sous la forme $\varphi\left(w_0 + w_1 x_1 + w_2 x_2\right)$, alors que celle d'un neurone Sigma-Pi est donnée par $\varphi\left(w_0 + w_1 x_1 + w_2 x_2 + w_{12} x_1 x_2\right)$, la différence résidant donc dans la présence du terme produit $x_1 x_2$.

- Dans le cas général à m entrées, on obtient une sortie de la forme suivante $\varphi\left(w_0 + \sum_{j=1}^{p} w_j \prod_{k \in I_j} x_k\right)$. Dans cette formule, p est un entier quelconque qui désigne le nombre de produits d'entrées utilisés. L'ensemble I_j est un sous-ensembles de $\{1,\ldots,m\}$ qui précise les entrées à multiplier entre elles pour obtenir le terme numéro j.
- Selon cette notation, l'exemple donné plus haut pour 2 entrées correspond à $p = 3$ (3 termes dans la somme), $I_1 = \{1\}$ (entrée 1 utilisée seule), $I_2 = \{2\}$ (entrée 2 utilisée seule) et $I_3 = \{1, 2\}$ (produit des entrées 1 et 2).
- La formulation utilisée montre qu'il existe de nombreuses possibilités pour construire un neurone Sigma-Pi pour un nombre donné d'entrées.
- Ceci est lié à la croissance exponentielle avec le nombre d'entrées du nombre de sous-ensembles d'entrées utilisables pour construire un neurone Sigma-Pi : il existe en effet 2^m combinaisons possibles (en considérant la combinaison vide comme celle correspondant au seuil w_0).
- En pratique, quand m devient grand (par exemple à partir de 20), il devient quasi impossible d'utiliser tous les termes possibles et il faut donc choisir les produits à privilégier.
- Une solution classique consiste à se restreindre à des sous-ensembles de k entrées pour une valeur faible de k.
- La formulation générale de ces neurones est aussi à l'origine du nom Sigma-Pi qui fait référence aux lettres grecques capitales Sigma (Σ) et Pi (Π) utilisées en mathématiques pour représenter respectivement la somme et le produit.

Exercice :

V

Question

[*solution* n°1 p.21]

Nous avons un neurone formel avec deux entrées $x_1 = 1, x_2 = 0.5$, les poids sont $w_1 = 0.2, w_2 = -0.1$ et un seuillage $w_0 = 1$ et la fonction d'activation la fonction sigmoïde, calculer sa sortie

Exercice : VI

Question

[*solution* n°2 p.21]

Donner les types de neurones qui peuvent être construits à partir d'une fonction autre que la somme pondérée de ses entrées

Exercice :

VII

Question 1

[solution n°3 p.21]

La plupart des neurones formels utilisés actuellement sont des variantes du neurone dans lesquelles la fonction d'activation est différente.

Donner les fonctions les plus utilisées.

Question 2

[solution n°4 p.21]

Donner leurs propriétés

Question 3

[solution n°5 p.21]

Donner les propriétés de la fonction sigmoïde

Question 4

[solution n°6 p.22]

Donner les propriétés de la fonction tangente hyperbolique

Conclusion

Ce deuxième chapitre a présenté un aperçu sur le neurone formel

Solutions des exercices

> **Solution** n°1

Exercice p. 17

$\varphi\left(\sum_{j=1}^{m} w_j x_j - w_0\right)$, la fonction d'activation est $\varphi(x) = \dfrac{1}{1+e^{-x}}$

$\varphi(w_1 \times x_1 + w_2 \times x_2 - w_0) = \dfrac{1}{1+e^{-(w_1 \times x_1 + w_2 \times x_2 - w_0)}}$

$w_1 \times x_1 + w_2 \times x_2 - w_0 = 0.2 \times 1 - 0.1 \times 0.5 - 1 = 0.2 - 0.05 - 1 = -0.85$

$\varphi(-0.85) = \dfrac{1}{1+e^{0.85}} = 0.7$

> **Solution** n°2

Exercice p. 18

- Un neurone à base radiale
- Un neurone Sigma-Pi

> **Solution** n°3

Exercice p. 19

- la fonction sigmoïde ;
- la fonction tangente hyperbolique ;
- la fonction identité ;
- la fonction rectifieur : $f(x) = \max(0, x)$;La lissage de la moyenne ou $f(x, y) = \dfrac{1}{2}\left(\dfrac{x+y}{2} + x\right)$, $x =$ poids du synapse actuel, $y = x$ attendu

> **Solution** n°4

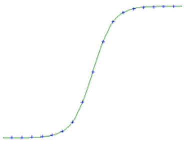
Exercice p. 19

- la fonction d'activation ne doit pas être un polynôme.
- Il est utile en pratique que la fonction d'activation présente une certaine forme de régularité.
- Pour calculer le gradient de l'erreur commise par un réseau de neurones, lors de son apprentissage, il faut que la fonction d'activation soit dérivable.
- Pour calculer la matrice hessienne de l'erreur, ce qui est utile pour certaines analyses d'erreur, il faut que la fonction d'activation soit dérivable deux fois.

> **Solution** n°5

Exercice p. 19

- La fonction sigmoïde (aussi appelée fonction logistique ou fonction en S) est définie par : $f_{sig}(x) = \dfrac{1}{1 + e^{-x}}$



- Elle n'est pas polynomiale
- Elle est indéfiniment continûment dérivable $\dfrac{d}{dx} f_{sig}(x) = f_{sig}(x)\left(1 - f_{sig}(x)\right)$.
- De plus, la fonction sigmoïde renvoie des valeurs dans l'intervalle $[0; 1]$, ce qui permet d'interpréter la sortie du neurone comme une probabilité.

> **Solution** n°6

Exercice p. 19

- La fonction tangente hyperbolique, définie par $\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$ est aussi très utilisée en pratique, car elle partage avec la fonction sigmoïde certaines caractéristiques pratiques :
  - non polynomiale
  - indéfiniment continûment dérivable
  - calcul rapide de la dérivée par la formule $\dfrac{\mathrm{d}}{\mathrm{d}x} \tanh(x) = 1 - (\tanh(x))^2$
- C'est, à une transformation linéaire près, la fonction sigmoïde : $\tanh(x) = 1 - 2\dfrac{1}{1 + e^{-2x}}$