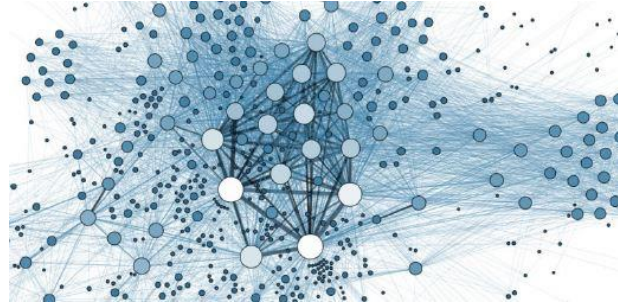
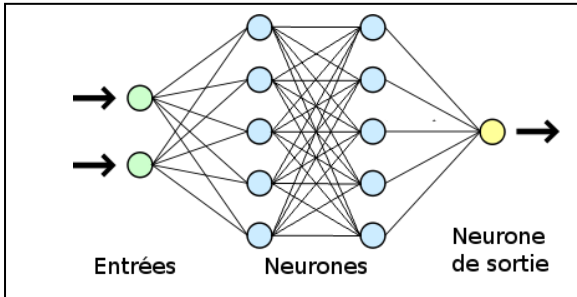


Cycle 8: Mettre en œuvre une démarche de résolution numérique

Chapitre 4 : Apprentissage SUPERVISE ou NON – les réseaux de neurones



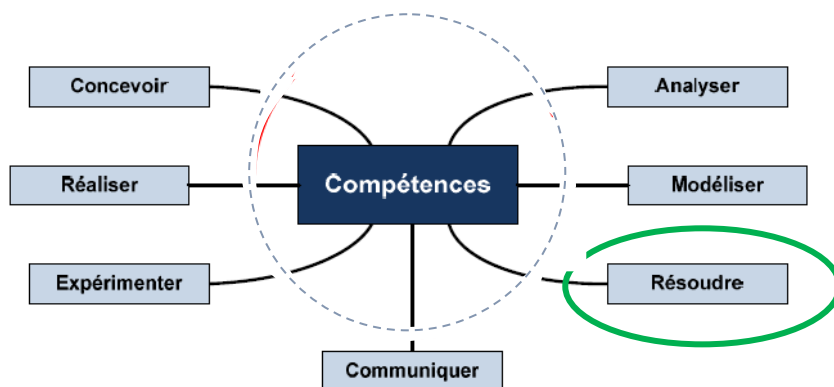
Problématique

Qu'est-ce que l'apprentissage non supervisé ?
Comment fonctionnent les réseaux de neurones ?

Savoir

C. Résoudre:

- C3 : mettre en œuvre une démarche de résolution numérique

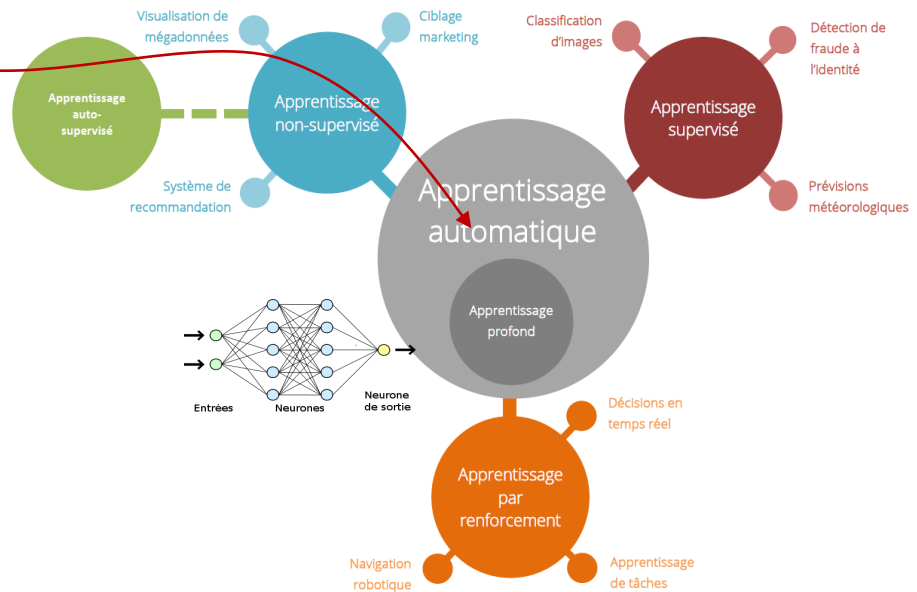




Apprentissage supervisé – la classification

1. Situation des réseaux de neurones en IA

Les réseaux de neurones sont une variété de technologie **Deep Learning (apprentissage profond)**, qui fait elle-même partie de la sous-catégorie d'intelligence artificielle du Machine Learning (apprentissage automatique). On peut utiliser des réseaux de neurones en apprentissage supervisé (avec dataset étiqueté, le plus simple à comprendre) ou **non supervisé (cas le plus utilisé)**.



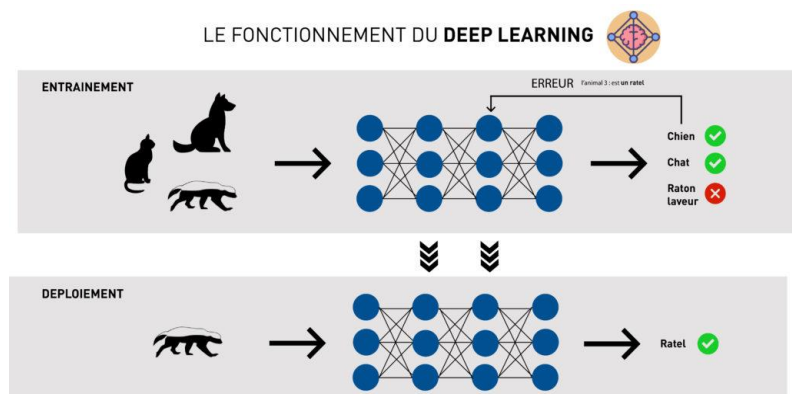
2. Introduction

Un réseau de neurones artificiels ou Neural Network est un système informatique **s'inspirant du fonctionnement du cerveau humain pour apprendre**.

Les Réseaux de Neurones sont des modèles bien plus **complexes** que tous les autres modèles de Machine Learning (descente de gradient....) dans le sens où ils représentent des fonctions mathématiques avec des **millions de paramètres**. Rappelez-vous, pour la régression linéaire nous n'avions que 2 coefficients *a* et *b*...

Avec une telle puissance, il est possible **d'entraîner la machine** sur des tâches bien plus avancées :

- La reconnaissance d'objets et reconnaissance faciale
- L'analyse de sentiments
- L'analyse du langage naturel
- La création artistique
- Etc.





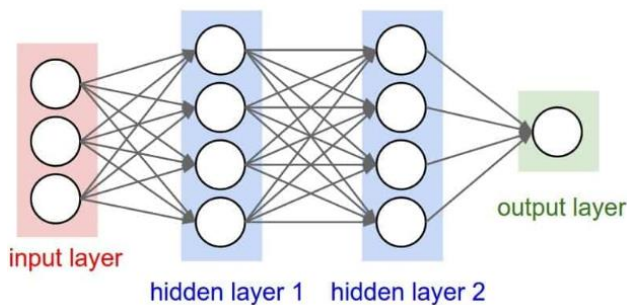
Apprentissage supervisé – la classification

Cependant, développer une fonction aussi complexe à un coût. Pour y parvenir, il faut souvent fournir :

- Un **Dataset** beaucoup **plus grand** (*des millions de données*)
- Un **temps d'apprentissage plus long** (*parfois plusieurs jours*)
- Une plus **grande puissance de calcul**.

3. Anatomie d'un réseau de neurones artificiels

Voilà à quoi ressemble un Réseau de Neurones :

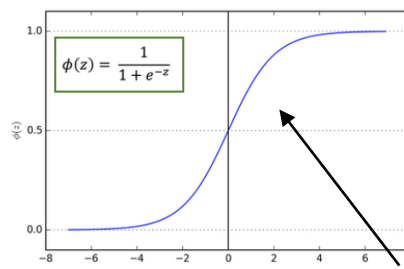
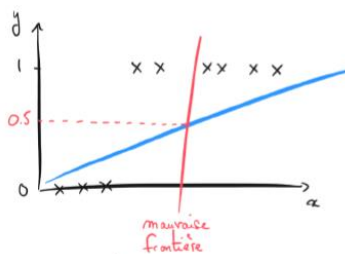


Vous remarquez un niveau d'entrées (*input layer*) à gauche, un niveau de sorties (*output layer*) à droite, et plusieurs niveaux **cachés** entre deux. Les petits ronds sont appelés les **neurones** et représentent des **fonctions d'activation**. Pour un réseau de neurone basique, la **fonction Logistique** est utilisée comme fonction d'activation.

Par la suite, **chaque tiers (neurones) reçoit les sorties d'informations du tiers (neurones) précédent**. On retrouve le même processus chez l'Homme, lorsque les neurones reçoivent des signaux en provenance des neurones proches du nerf optique. Le dernier tiers, quant à lui, produit les résultats du système. **Plus ces réseaux sont profonds, avec beaucoup de couches et neurones, plus la machine sera réaliser des tâches complexes.**

C'est quoi la fonction logistique ?

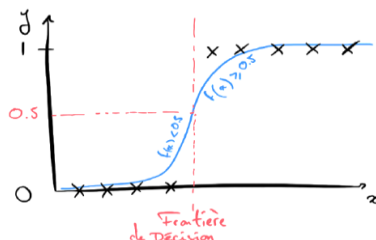
Pour les problèmes de classification binaire, un modèle linéaire $y=ax+b$, comme je l'ai tracé sur la figure ci-dessous, ne convient pas.



On développe alors une nouvelle fonction pour les problèmes de classification binaire, c'est la **fonction logistique** (aussi appelé fonction **sigmoïde** ou tout simplement sigma σ). Cette fonction a la particularité d'être toujours comprise en 0 et 1.

A partir de cette fonction et sur mon exemple, il est possible de définir une **frontière de décision**. Typiquement, on définit un seuil à **0.5** comme ceci :

$$\begin{cases} y = 0 & \text{si } \sigma(X \cdot \theta) < 0.5 \\ y = 1 & \text{si } \sigma(X \cdot \theta) \geq 0.5 \end{cases}$$

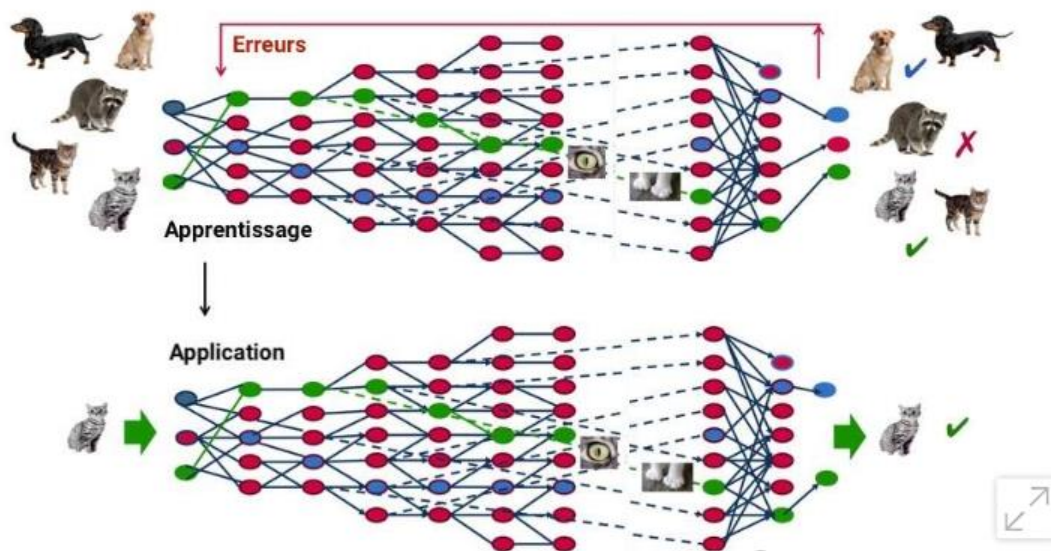


Apprentissage supervisé – la classification

Réseaux de neurones en apprentissage SUPERVISE :

Par le biais d'un **algorithme d'optimisation**, le réseau de neurones artificiels permet à l'ordinateur **d'apprendre à partir de nouvelles données**. L'ordinateur doté du réseau de neurones apprend à effectuer une tâche en **analysant des exemples pour s'entraîner**. Ces exemples ont préalablement été étiquetés afin que le réseau puisse savoir ce dont il s'agit.

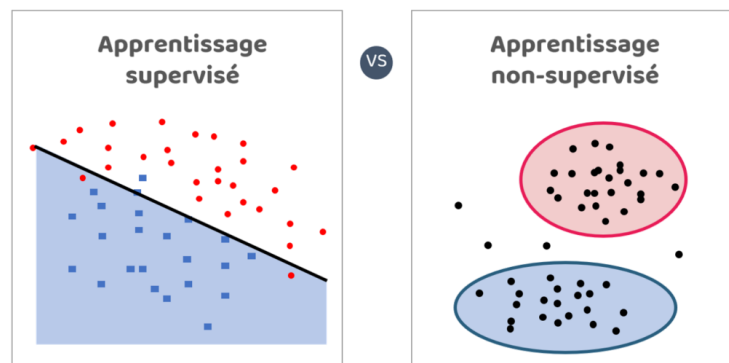
Ex : un réseau de neurones peut être utilisé pour apprendre à l'ordinateur à reconnaître des objets. Un grand nombre d'objets d'une même catégorie est présenté au réseau de neurones, et l'ordinateur apprend à reconnaître cet objet sur de nouvelles images en analysant les patterns récurrentes au sein des images d'exemple. Ainsi, en analysant des milliers de photos de chats, le Neural Network apprendra à reconnaître un chat sur n'importe quelle photo.



À travers un processus d'autoapprentissage, le deep Learning est capable d'identifier un chat sur une photo. À chaque couche du réseau neuronal correspond un aspect particulier de l'image.

Réseaux de neurones en apprentissage NON SUPERVISE :

Dans le cas de l'**apprentissage non-supervisé**, les données ne sont pas étiquetées. Le réseau de neurones analyse l'ensemble de données, et une fonction-coût lui indique dans quelle mesure il est éloigné du résultat souhaité. Le réseau s'adapte alors pour augmenter la précision de l'algorithme.





Apprentissage supervisé – la classification

Parallèle avec les neurones de notre cerveau...

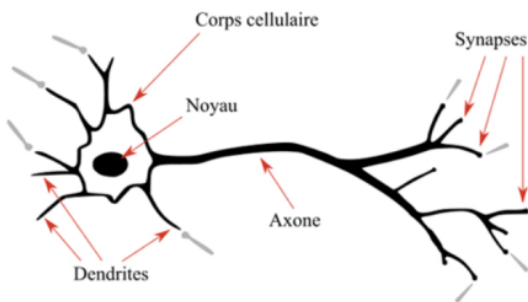
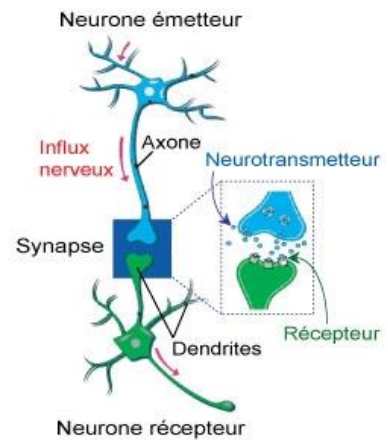
En biologie, les neurones sont des cellules excitables connectées les unes aux autres et ayant pour rôle de transmettre des informations dans notre système nerveux.

Chaque neurone est composé de plusieurs dendrites, d'un corps cellulaire et d'un axone. Les **synapses** accrochés aux dendrites reçoivent les signaux des autres neurones = couche d'entrée.

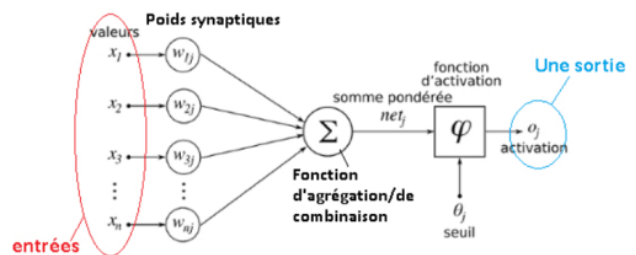
Les **signaux reçus par les synapses peuvent être : excitateurs (+1) ou Inhibiteurs (-1).**

Lorsque la somme de ces signaux dépasse un certain seuil, le **neurone s'active** et véhiculera le long de l'axone vers d'autres synapses et d'autres neurones....

Les neurones artificiels vont s'inspirer fortement des neurones biologiques...



NEURONE BIOLOGIQUE



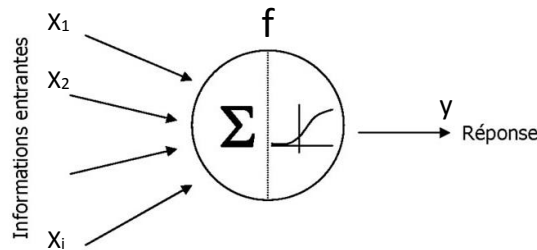
NEURONE ARTIFICIEL



Apprentissage supervisé – la classification

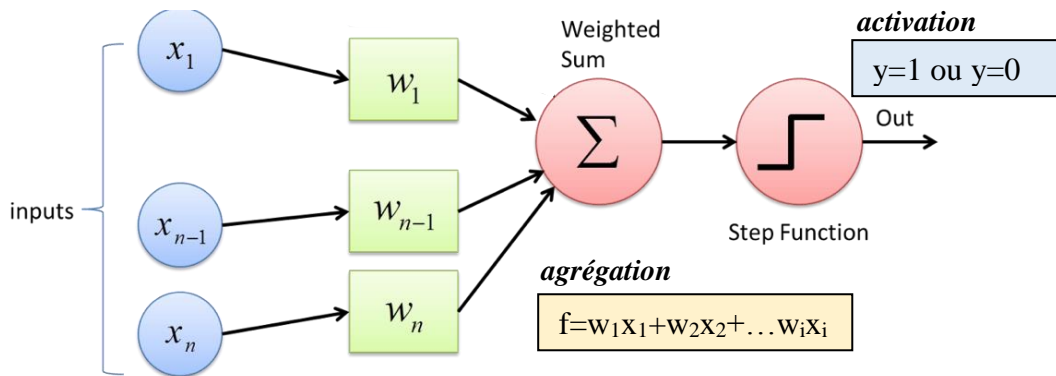
4. Première approche de la modélisation d'un neurone

Un neurone est une **fonction de transfert f**, qui prend en entrée des signaux x_i , et qui retourne une sortie y .



A l'intérieur de cette fonction, on trouve 2 grades étapes :

- **Etape d'agrégation** : on fait la somme de toute les entrées x_i en multipliant par un coefficient w_i chacune d'elle (w est l'activité synaptique : activateur = +1 ou inhibiteur = -1)



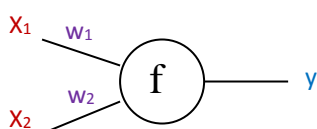
- **Etape d'activation** : on regarde le résultat du calcul précédent de f et si celui-ci **dépasse un certain seuil (en général 0)**, alors le **neurone s'active et retourne une sortie $y=1$** sinon $y=0$.

5. Le modèle du perceptron

Le modèle de neurone artificiel précédent est une première approche mais ne gère que des entrées logiques. Le modèle du perceptron inventé en 1957 est identique avec sa fonction d'activation mais il **dispose en plus d'un algorithme d'apprentissage** lui permettant **de trouver les valeurs des paramètres w_i afin de trouver les sorties y qui nous conviennent...**

L'algorithme d'apprentissage consiste à entraîner un neurone artificiel sur des données de référence (x,y) pour que celui-ci renforce ses paramètres w à chaque fois qu'une entrée x est activée en même temps que la sortie y présente dans ses données.

Pour cela, la formule suivante montre que les paramètres w sont mis à jour en calculant la différence entre la sortie de référence et la sortie produite par le neurone en multipliant par chaque entrée X et un pas d'apprentissage α .



$$w = w + \alpha(y_{true} - y)X$$

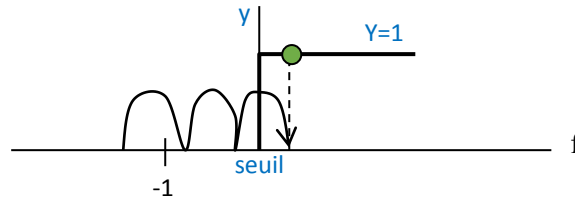
- y_{true} : sortie de référence
- y : sortie produite par neurone
- X : paramètre d'entrée
- α : vitesse apprentissage (<0)



Apprentissage supervisé – la classification

Si notre neurone produit une sortie différente de celle attendue, pour les entrées $x=1$, le **coefficient w se verra augmenté d'un petit pas α** , on parle de renforcement. Pour les entrées $x=0$, w ne changera pas.

Du coup, la fonction f augmentera et cela rapprochera notre neurone du seuil d'activation d'activation.



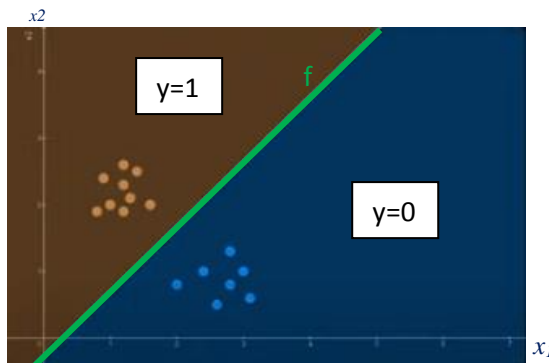
Aussi longtemps qu'on sera en dessous du seuil, c'est-à-dire aussi longtemps que le neurone produira une mauvaise sortie, alors le coef w continuera d'augmenter jusqu'à ce que $y_{true}=y$ et l'activation se produira. Nos paramètres arrêteront d'évoluer.

5. Le perceptron multicouche

Le perceptron vu précédemment est l'unité de base des réseaux de neurones. Il s'agit d'un modèle de classification binaire, capable de séparer linéairement 2 classes de points.

On va rajouter ici un petit paramètre complémentaire, **b = le biais**. Soit $f = w_1x_1 + w_2x_2 + b$

Avec ce modèle on peut séparer par ex 2 classes de points (tout ce qui sera au-dessus de la droite donnera une sortie $y=1$ ou en dessous : $y=0$).



Attention : beaucoup de **problème ne sont pas linéaires**, donc comment faire avec notre perceptron ?

L'idée : **connecter plusieurs neurones ensemble**.

Exemple avec 3 neurones et 2 couches :

$$f_1 = w_{11}x_1 + w_{12}x_2 + b_1$$

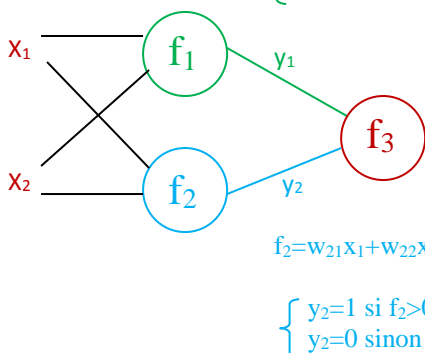
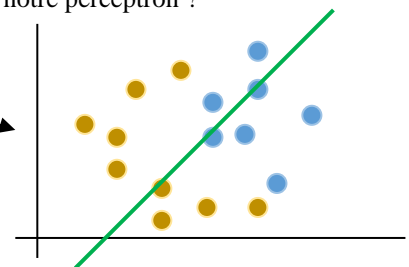
$$\begin{cases} y_1 = 1 & \text{si } f_1 > 0 \\ y_1 = 0 & \text{sinon} \end{cases}$$

$$f_3 = w_{31}y_1 + w_{32}y_2 + b_3$$

$$\begin{cases} y_3 = 1 & \text{si } f_3 > 0 \\ y_3 = 0 & \text{sinon} \end{cases}$$

$$f_2 = w_{21}x_1 + w_{22}x_2 + b_2$$

$$\begin{cases} y_2 = 1 & \text{si } f_2 > 0 \\ y_2 = 0 & \text{sinon} \end{cases}$$

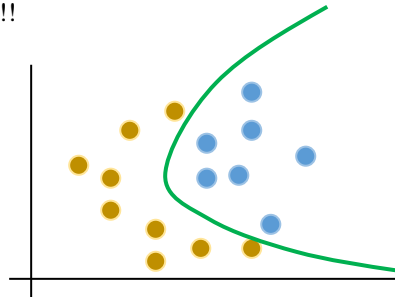


Perceptron à 2 couches (3 neurones)



Apprentissage supervisé – la classification

Si on trace la **représentation graphique de la sortie finale en fonction des entrées x_1 et x_2** , on obtient un modèle non linéaire qui est bien plus intéressant !!

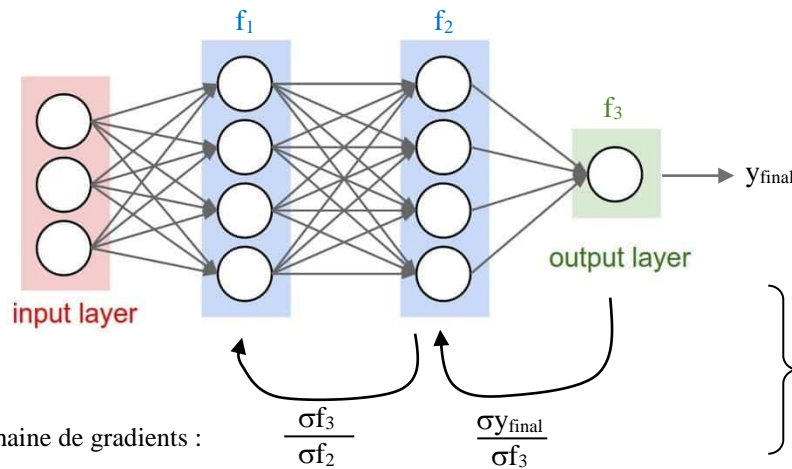


On peut ainsi rajouter autant de couche que l'on souhaite pour résoudre des problèmes de plus en plus complexes...mais attention, il faudra un dataset énorme, être patient pour temps calcul...

Comment entrainer un tel réseau de neurones pour qu'il fasse ce qu'on lui demande de faire ?

Effectivement, comment trouver les valeurs des paramètres w_i et b pour obtenir un bon modèle ?

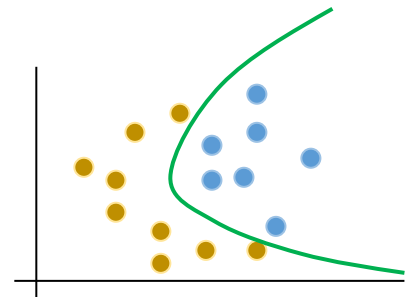
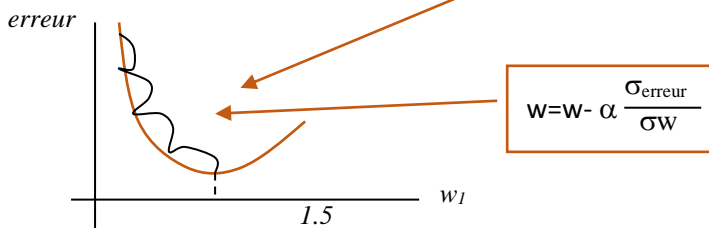
La solution est d'utiliser une technique appelée **Back Propagation** qui consiste à déterminer comment la sortie du réseau varie en fonction des paramètres (w, b) présents dans chaque couche. Pour cela, on **calcule une chaîne de gradients** indiquant comment la sortie varie en fonction de la dernière couche et ainsi de suite jusqu'à arriver à la première couche du réseau.



c'est une **Back propagation**

Grace à cette chaine de gradients, on peut alors **mettre à jour les paramètres (w, b)** de chaque couche de telle sorte à ce qu'ils **minimisent l'erreur entre la sortie du modèle et la réponse attendue.**

Pour cela on utilise ici aussi une descente de gradient que l'on détaillera plus loin.

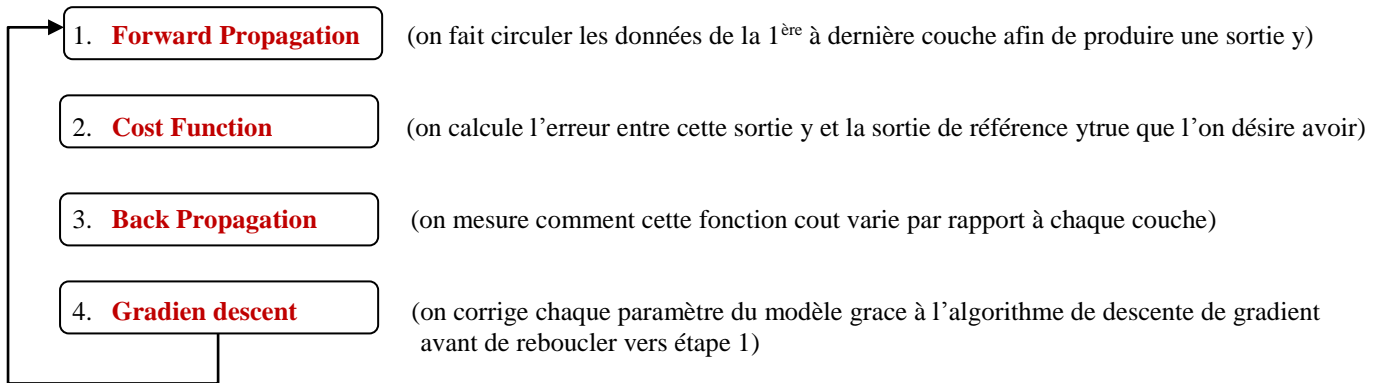




Apprentissage supervisé – la classification

SYNTHESE :

Pour développer et entrainer des réseaux de neurones artificiels, on répète en boucle les **4 étapes suivantes** :



Maintenant, vous pouvez regarder cette vidéo qui résume tout ce que l'on vient de voir :

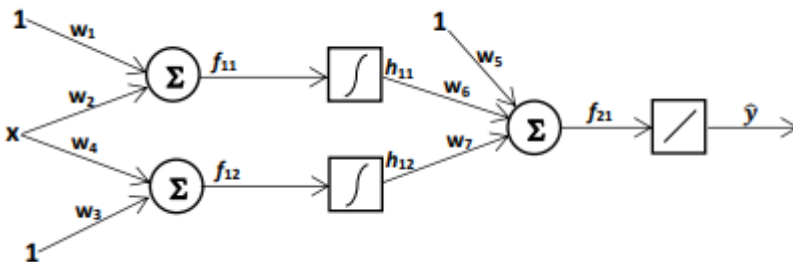
<https://www.youtube.com/watch?v=XUFLq6d>



Exercices corrigés

Exercice n°1 :

Soit le réseau de neurones multicouches décrit par le graphe suivant :



1. Donner les formules mathématiques qui déterminent les sorties intermédiaires f_{11} , f_{12} , h_{11} , h_{12} , f_{21} ainsi que la sortie finale \hat{y} de la propagation avant (forward propagation).

Nota : \int représente la fonction sigmoïde (cf p3)

2. Soit la fonction d'erreur : $\mathbf{E}(\mathbf{w}) = (\mathbf{y} - \hat{\mathbf{y}})^2$. En appliquant l'algorithme de propagation en arrière (back propagation), trouver l'expressions des mises à jour des paramètres Δw_j .



Apprentissage supervisé – la classification

Solutions

1- Propagation en avant (forward propagation)

- $f_{11} = w_2x + w_1$
- $f_{12} = w_4x + w_3$
- $h_{11} = \text{sigm}(f_{11}) = \frac{1}{1+e^{-f_{11}}}$
- $h_{12} = \text{sigm}(f_{12}) = \frac{1}{1+e^{-f_{12}}}$
- $\hat{y} = f_{21} = w_6h_{11} + w_7h_{12} + w_5$

2- Propagation en arrière (backpropagation algorithm)

La fonction d'erreur est donnée par $E(w) = (y - \hat{y})^2$

Donc, on aura $\frac{\partial E(w)}{\partial w_j} = -2(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_j}$

D'après la propagation en avant, on a : $\hat{y} = f_{21} = w_6h_{11} + w_7h_{12} + w_5$

Donc, les dérivées $\frac{\partial \hat{y}}{\partial w_j}$ peuvent être calculées par :

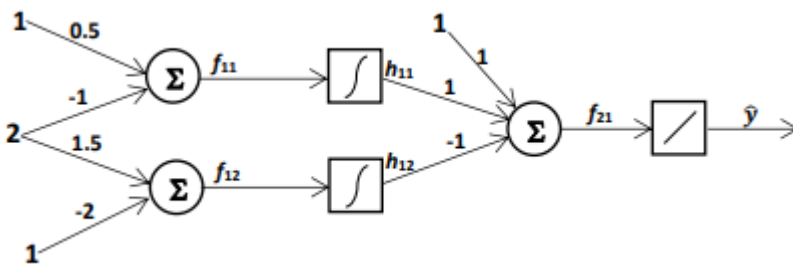
- $\frac{\partial \hat{y}}{\partial w_5} = 1$
- $\frac{\partial \hat{y}}{\partial w_6} = h_{11}$
- $\frac{\partial \hat{y}}{\partial w_7} = h_{12}$
- $\frac{\partial \hat{y}}{\partial w_1} = \frac{\partial \hat{y}}{\partial h_{11}} \frac{\partial h_{11}}{\partial f_{11}} \frac{\partial f_{11}}{\partial w_1} = w_6 h_{11} (1 - h_{11})$
- $\frac{\partial \hat{y}}{\partial w_2} = \frac{\partial \hat{y}}{\partial h_{11}} \frac{\partial h_{11}}{\partial f_{11}} \frac{\partial f_{11}}{\partial w_2} = w_6 h_{11} (1 - h_{11}) x$
- $\frac{\partial \hat{y}}{\partial w_3} = \frac{\partial \hat{y}}{\partial h_{12}} \frac{\partial h_{12}}{\partial f_{12}} \frac{\partial f_{12}}{\partial w_3} = w_7 h_{12} (1 - h_{12})$
- $\frac{\partial \hat{y}}{\partial w_4} = \frac{\partial \hat{y}}{\partial h_{12}} \frac{\partial h_{12}}{\partial f_{12}} \frac{\partial f_{12}}{\partial w_4} = w_7 h_{12} (1 - h_{12}) x$

En fin, la mise à jour de chaque paramètre est donnée par la formule :

$$\Delta w_j = \alpha (y - \hat{y}) \frac{\partial \hat{y}}{\partial w_j}$$

Exercice n°2 :

Soit le réseau de neurones multicouches décrit par le graphe suivant :



Soit la donnée $(x, y) = (2, 1)$

1. Calculer les sorties intermédiaires f_{11} , f_{12} , h_{11} , h_{12} , f_{21} ainsi que la sortie finale \hat{y} .
2. Calculer les paramètres Δw_j et w_j pour $j = 1, \dots, 7$ après une itération de mise à jour (en considérant le paramètre d'apprentissage $\alpha = 0.1$).



Apprentissage supervisé – la classification

Solutions

1- Calcul des sorties intermédiaires f_{11} , f_{12} , h_{11} , h_{12} , f_{21} :

- $f_{11} = 0.5 * 1 + -1 * 2 = -1.5$
- $f_{12} = 1.5 * 2 + -2 * 1 = 1$
- $h_{11} = \text{sigm}(f_{11}) = \frac{1}{1+e^{-f_{11}}} = \frac{1}{1+e^{1.5}} = 0.1824$
- $h_{12} = \text{sigm}(f_{12}) = \frac{1}{1+e^{-f_{12}}} = \frac{1}{1+e^{-1}} = 0.7311$
- $f_{21} = 1 * 1 + 1 * 0.1824 + -1 * 0.7311 = 0.4513$
- $\hat{y} = f_{21} = 0.4513$

2- Calculer les paramètres w_j pour $j = 1, \dots, 7$.

- Soit le paramètre d'apprentissage $\alpha = 0.1$:
- On a $w_j = w_j + \Delta w_j$ pour $j = 1, \dots, 7$:

2-1- Calcul des Δw_j pour $j = 1, \dots, 7$:

- $\Delta w_1 = \alpha(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_1} = 0.1 * (1 - 0.4513) * 1 * 0.1824 * (1 - 0.1824) = 0.0082$
- $\Delta w_2 = \alpha(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_2} = 0.1 * (1 - 0.4513) * 2 * 0.1824 * (1 - 0.1824) = 0.0164$
- $\Delta w_3 = \alpha(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_3} = 0.1 * (1 - 0.4513) * 1 * 0.7311 * (1 - 0.7311) = 0.0108$
- $\Delta w_4 = \alpha(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_4} = 0.1 * (1 - 0.4513) * 2 * 0.7311 * (1 - 0.7311) = 0.0216$
- $\Delta w_5 = \alpha(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_5} = 0.1 * (1 - 0.4513) * 1 = 0.0019 = 0.0549$
- $\Delta w_6 = \alpha(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_6} = 0.1 * (1 - 0.4513) * 0.1824 = 0.0100$
- $\Delta w_7 = \alpha(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_7} = 0.1 * (1 - 0.4513) * 0.7311 = 0.0401$

2-2- Calcul des w_j pour $j = 1, \dots, 7$:

- $w_1 = w_1 + \Delta w_1 = 0.5 + 0.0082 = 0.5082$
- $w_2 = w_2 + \Delta w_2 = -1 + 0.0164 = -0.9836$
- $w_3 = w_3 + \Delta w_3 = -2 + 0.0108 = -1.9892$
- $w_4 = w_4 + \Delta w_4 = 1.5 + 0.0216 = 1.5216$
- $w_5 = w_5 + \Delta w_5 = 1 + 0.0549 = 1.0549$
- $w_6 = w_6 + \Delta w_6 = 1 + 0.0100 = 1.0100$
- $w_7 = w_7 + \Delta w_7 = -1 + 0.0401 = -0.9599$