Traitement et Analyse d'Images

Réseaux de neurones

olivier.bernard@insa-lyon.fr

Contexte

Apprentissage profond

Acquisition / pré-traitement

$$x^{(1)} = \left(x_1^{(1)}, \cdots, x_n^{(1)}\right)^T$$
$$x^{(2)} = \left(x_1^{(2)}, \cdots, x_n^{(2)}\right)^T$$

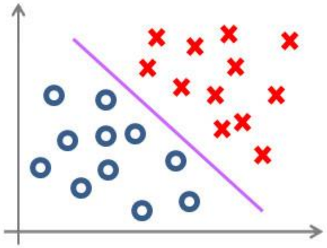
↓

Extraction de caractéristiques

$$a^{(1)} = \left(a_1^{(1)}, \cdots, a_d^{(1)}\right)^T$$
$$a^{(2)} = \left(a_1^{(2)}, \cdots, a_d^{(2)}\right)^T$$

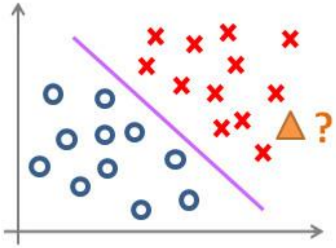
↓

Classification / Régression



↓

Décision



Contexte

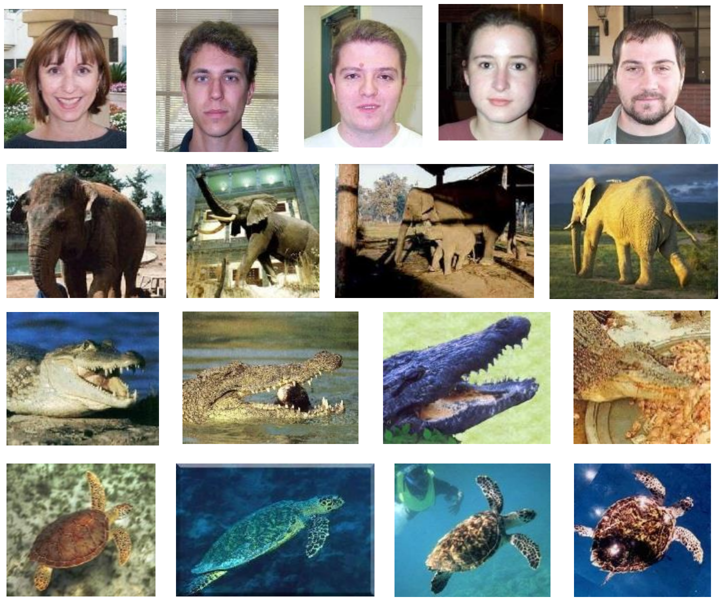
- Base de données d'entrainement
  - Apprendre à reconnaitre des objets, des tendances, des groupes, …

- Base de données de test
  - Appliquer les modèles appris sur de nouvelles données (en dehors de la base d'entrainement)

Démarche générale

- Illustration

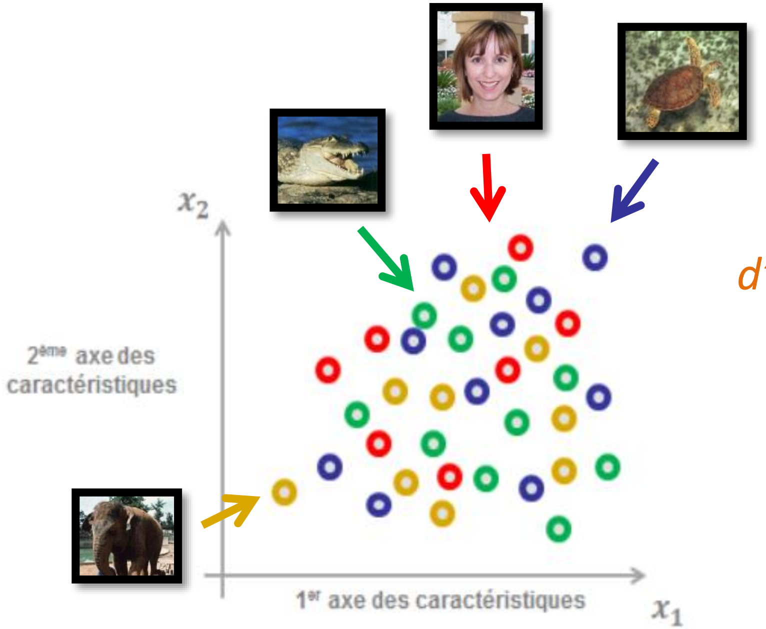
Reconnaissance de visages parmi un ensemble d'imagettes



Données d'entrainement

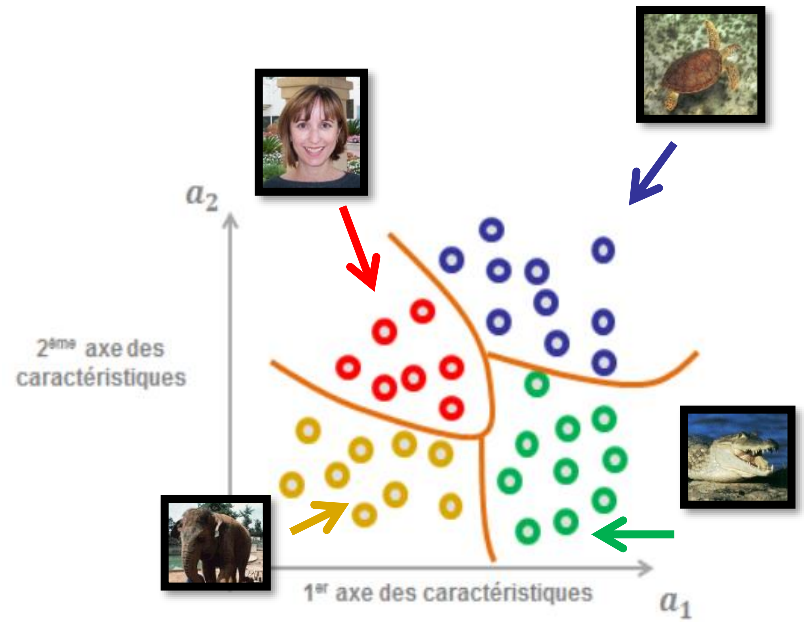
Apprentissage automatique et simultané des caractéristiques discriminantes et de la fonction de décision associée

Démarche générale



Espace image initial

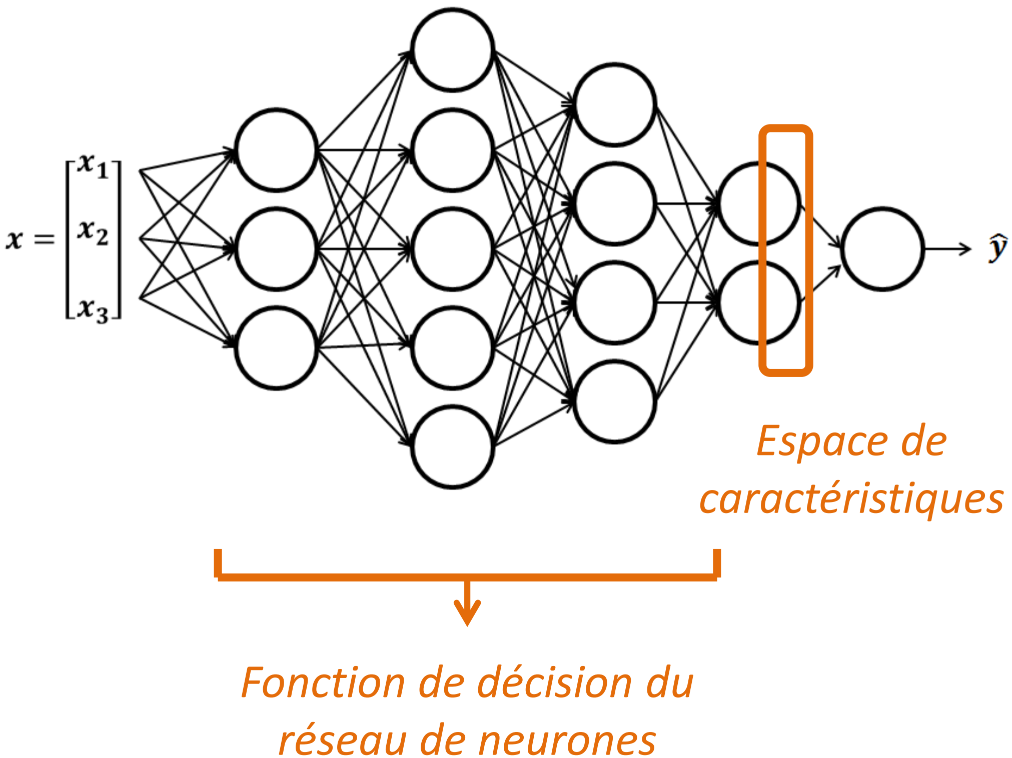
Algorithme d'apprentissage profond

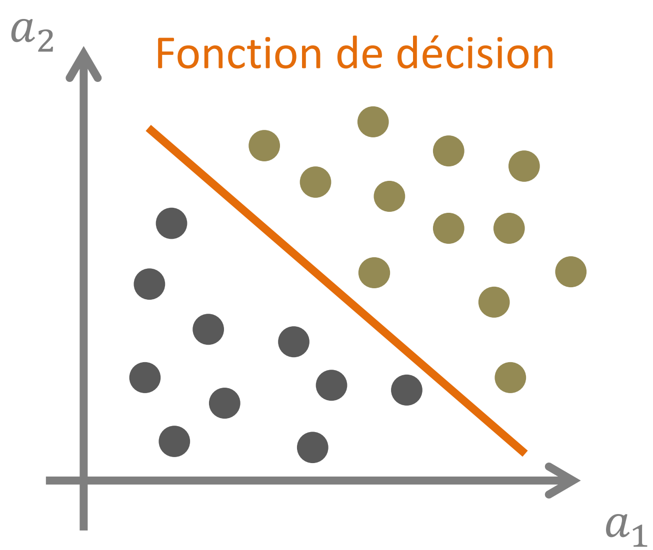


Espace de caractéristiques appris avec la fonction de décision associée

MLP: Réseaux de neurones multicouches

- Famille de méthodes la plus simple en apprentissage profond





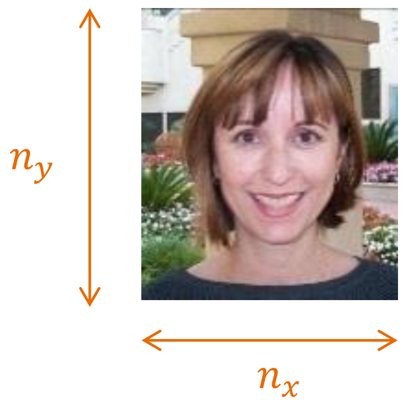
Sorties (espace de caractéristiques + fonction de décision) d'un réseau de neurones

Les réseaux de neurones

Modélisation d’un neurone

Concepts de base

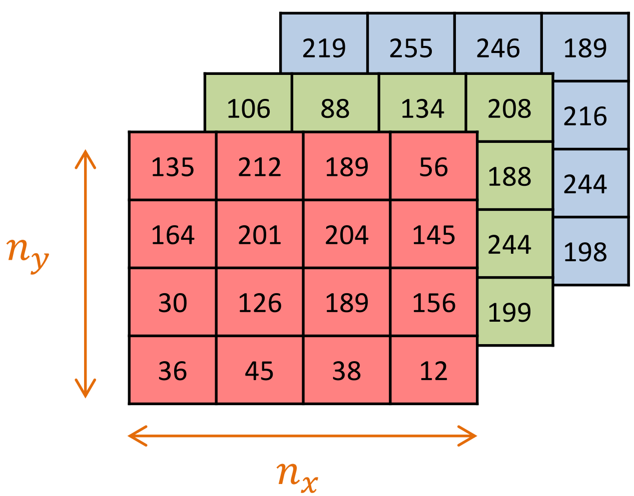
▶ Classification binaire



Présence d'un visage ?

1 (visage)
0 (pas de visage)

y

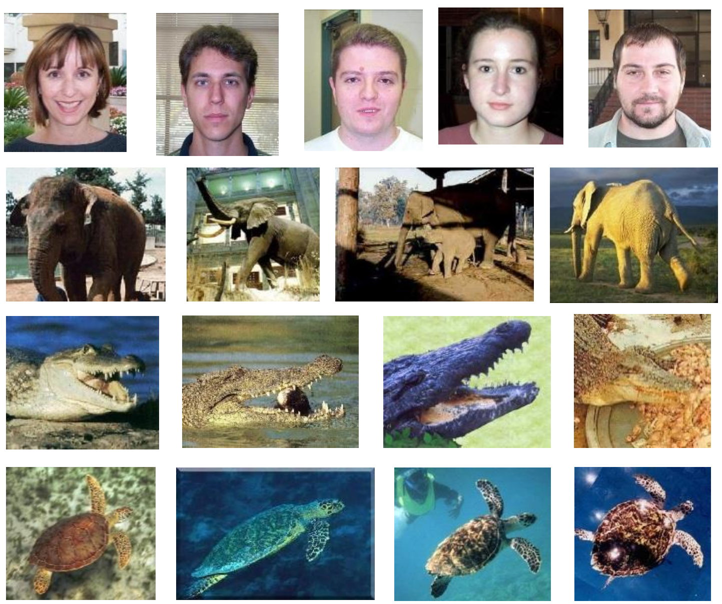


$$x = \begin{bmatrix} 135 \\ 212 \\ 189 \\ \vdots \\ 198 \end{bmatrix} \in \mathbb{R}^{[n_X \times 1]}$$

avec $n_X = n_x \times n_y \times 3$

Concepts de base

▶ Base de données



Base de données composées de m échantillons

- Chaque échantillon (image) est labélisé

$$(x^{(i)}, y^{(i)})$$

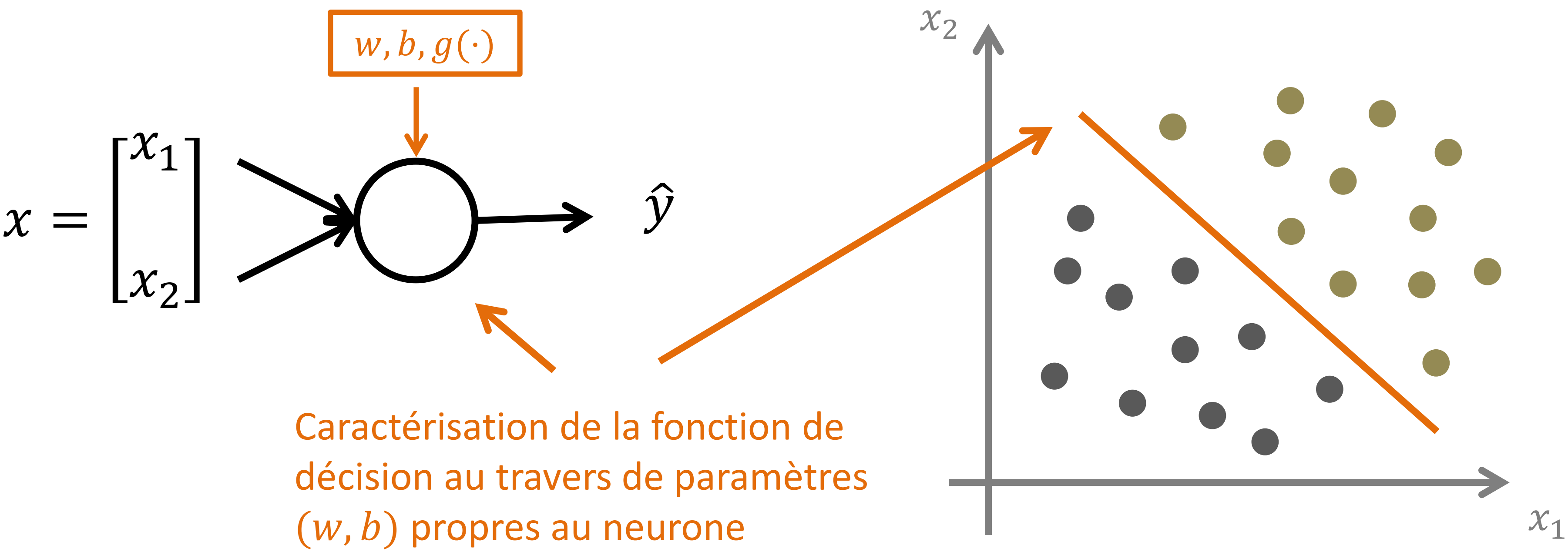
$$x^{(i)} \textit{ image et } y^{(i)} = \begin{cases} 1 \rightarrow \text{visage} \\ 2 \rightarrow \text{éléphant} \\ 3 \rightarrow \text{croco.} \\ 4 \rightarrow \text{tortue} \end{cases}$$

- Base de données composée de m échantillons

$$\{(x^{(1)}, y^{(1)}), \cdots, (x^{(i)}, y^{(i)}), \cdots, (x^{(m)}, y^{(m)})\}$$

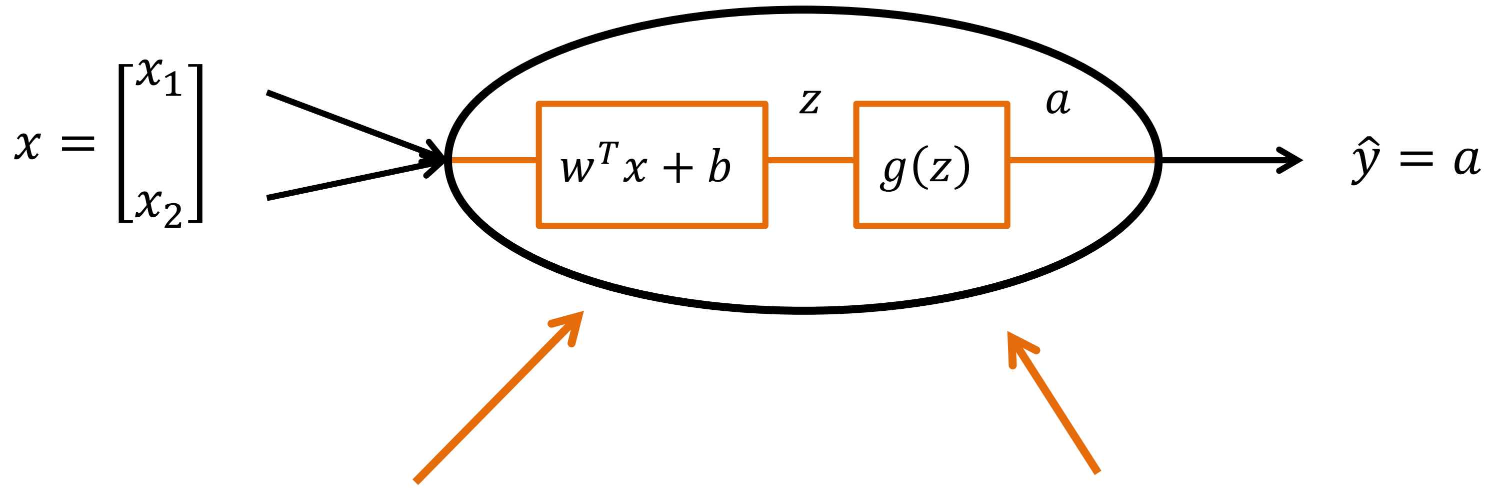
Fonction de décision

- Comment définir au niveau d'un neurone une fonction de décision qui permet de classifier les échantillons ?



Fonction de décision

▶ Deux éléments clés



1) Projection du vecteur de données x sur un vecteur de paramètres w avec un décalage de b

2) Application d'une transformation non-linéaire (fonction d'activation) afin de prendre une décision

Paramétrage d'un neurone

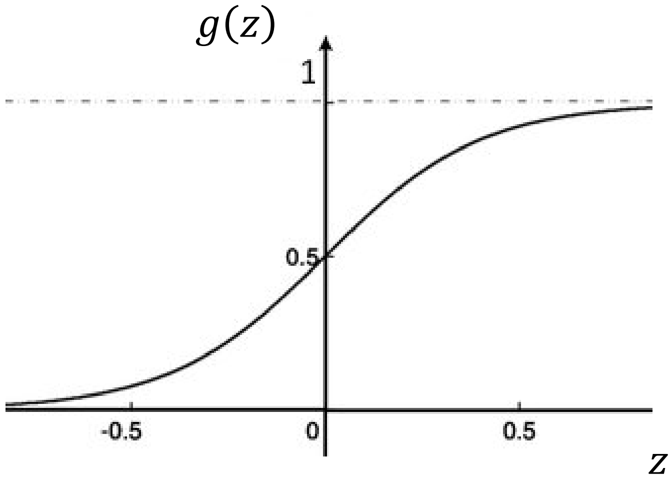
- Paramètres intervenant dans le 1er élément

$$w \in \mathbb{R}^{[n_x \times 1]} \text{ et } b \in \mathbb{R}$$

- Fonction d'activation intervenant dans le 2ème élément

Fonction sigmoïde

$$g(z) = \frac{1}{1+e^{-z}}$$

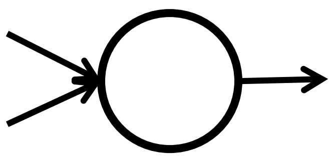


$$g'(z) = \frac{e^{-z}}{1+e^{-z}} = g(z) \cdot (1 - g(z))$$

Utilisation d’un neurone

▶ Définition

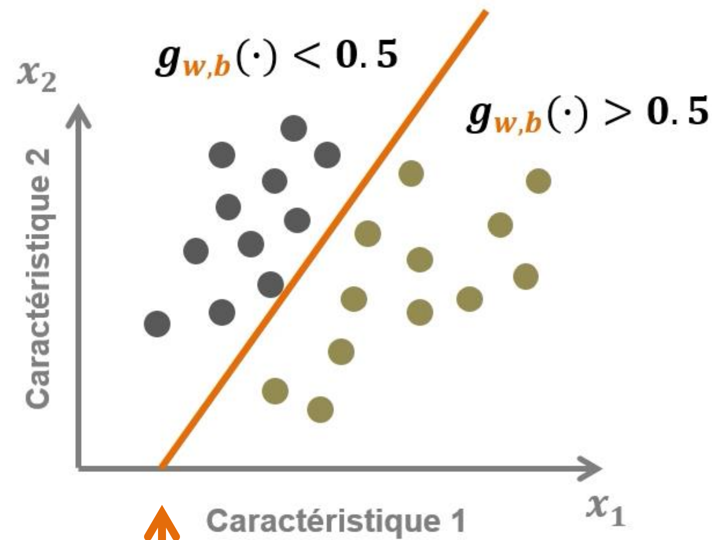
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



$$\begin{cases} \hat{y} = g(w^T x + b) \\ \hat{y} = g_{w,b}(x) \end{cases}$$

$$\hat{y} = 0.5 \implies \begin{aligned} & g(w^T x + b) = 0.5 \\ & w^T x + b = 0 \\ & \underbrace{w_1 x_1 + w_2 x_2 + b = 0}_{\textit{Équation d'une droite}} \end{aligned}$$

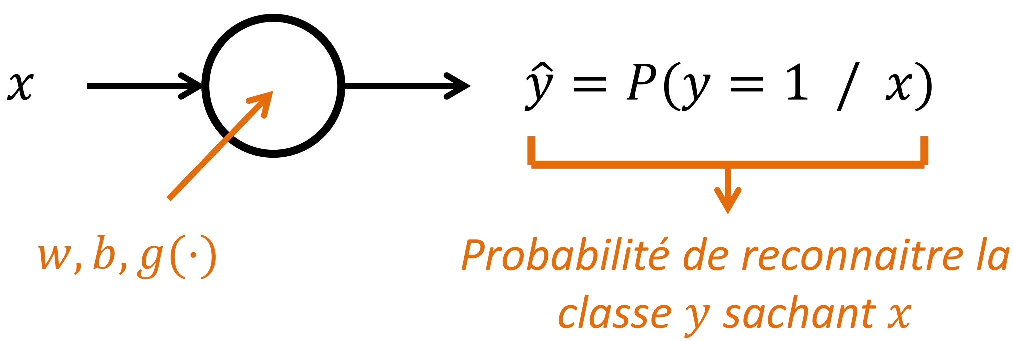
▶ Frontière de décision



Frontière de décision dépendant de *w* et *b*

Apprentissage de la fonction de décision

- A partir d'une base de données, comment apprendre les paramètres d'un neurone ?
  - Les paramètres d'un neurone doivent permettre de reconnaitre la classe y associée à l'échantillon entrant x
  - Approche probabiliste



Apprentissage de la fonction de décision

- Définition d'une fonction de perte à minimiser

  - Lorsque l'on a $\{x, y\}$ en entrée d'un neurone, on souhaite que sa sortie $\hat{y}$ soit la plus proche de y

  - Fonction de perte classiquement utilisée

$$\underbrace{\mathcal{L}(\hat{y}, y) = -[y\, log(\hat{y}) + (1 - y)\, log(1 - \hat{y})]}_{\textit{Entropie croisée}}$$

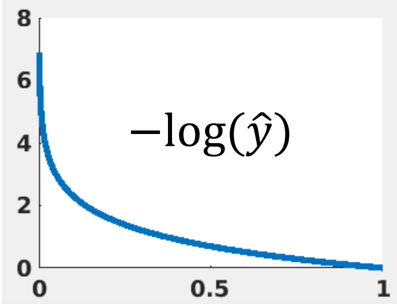
Apprentissage de la fonction de décision

- Fonction de perte à minimiser

$$\mathcal{L}(\hat{y}, y) = -[y\, log(\hat{y}) + (1 - y)\, log(1 - \hat{y})]$$

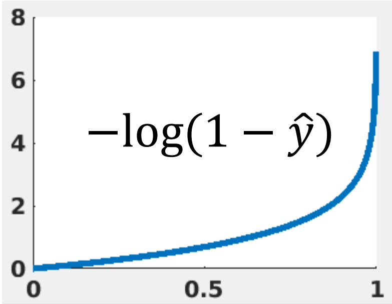
- Quand $y = 1$

  $\mathcal{L}(\hat{y}, 1)$ est minimum lorsque $\hat{y} \rightarrow 1$



- Quand $y = 0$

  $\mathcal{L}(\hat{y}, 0)$ est minimum lorsque $\hat{y} \rightarrow 0$



Apprentissage de la fonction de décision

- Fonction de perte à minimiser

  - Application à l'ensemble de la base de données

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

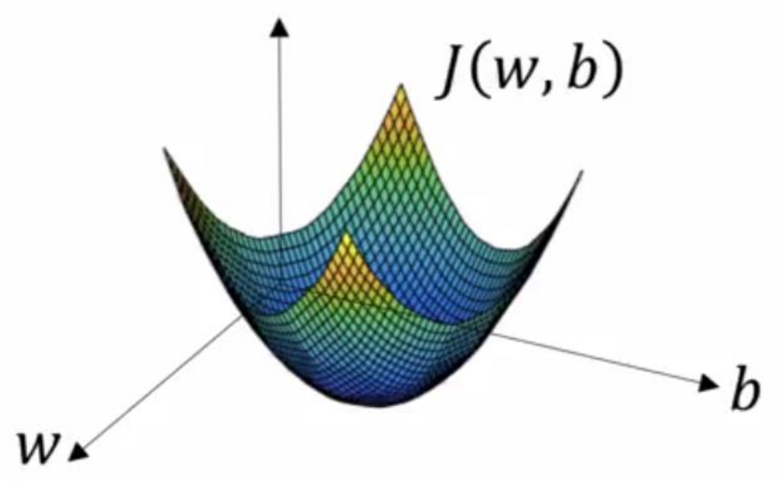
  - La minimisation de $J(w,b)$ a pour but de trouver les paramètres du neurone $\{w, b\}$ qui permettent de reconnaitre pour chaque échantillon $x^{(i)}$ de la base de données la classe $y^{(i)}$ associée

Apprentissage de la fonction de décision

- Fonction de perte à minimiser

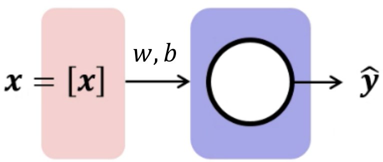
$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

- Les paramètres optimaux $\{\tilde{w}, \tilde{b}\}$ peuvent être estimés par une méthode de descente de gradient



Elément de modélisation

- Calculer les dérivées partielles $\frac{\partial J}{\partial b}$ et $\frac{\partial J}{\partial w}$ dans le cas où la fonction de perte est l'entropie croisée



$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

avec $\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) = -\left[y^{(i)}\,log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{y}^{(i)}\right)\right]$

$$\left\{\begin{array}{l} \dfrac{\partial J}{\partial b} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\dfrac{\partial \mathcal{L}}{\partial b} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\left(\dfrac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}}\cdot\dfrac{\partial \hat{y}^{(i)}}{\partial z^{(i)}}\cdot\dfrac{\partial z^{(i)}}{\partial b}\right) \\[2em] \dfrac{\partial J}{\partial w} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\dfrac{\partial \mathcal{L}}{\partial w} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\left(\dfrac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}}\cdot\dfrac{\partial \hat{y}^{(i)}}{\partial z^{(i)}}\cdot\dfrac{\partial z^{(i)}}{\partial w}\right) \end{array}\right.$$

Apprentissage de la fonction de décision

▶ Approche empirique - Descente de gradient

Initialisation de w et b

Répéter jusqu'à convergence

$\{$

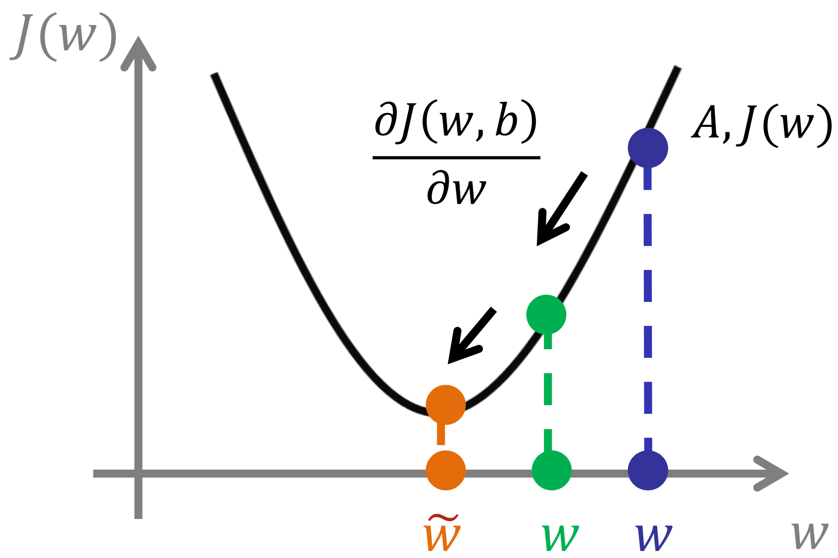
calcul de $A = [\hat{y}^{(1)}, \dots, \hat{y}^{(m)}]$

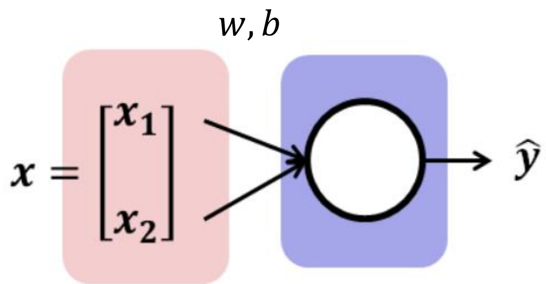
calcul de $J(w, b)$

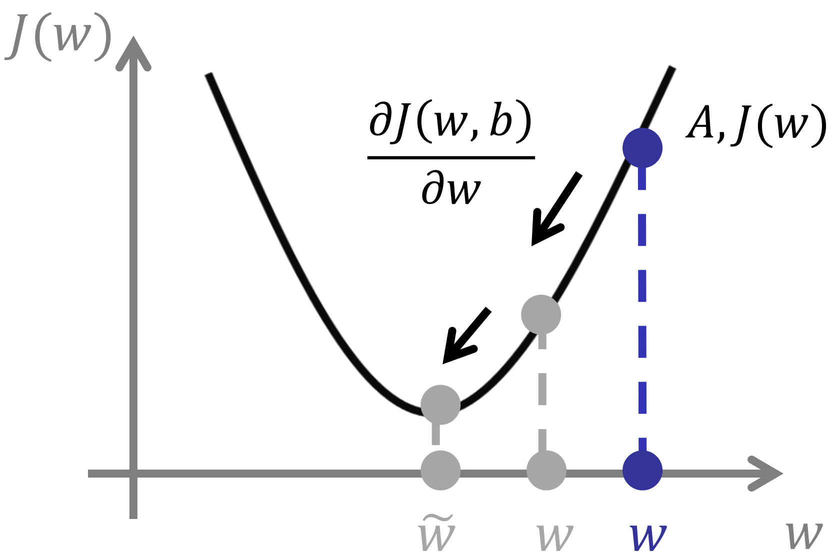
$$w := w - \alpha \frac{\partial J(w,b)}{\partial w}$$

$$b := b - \alpha \frac{\partial J(w,b)}{\partial b}$$

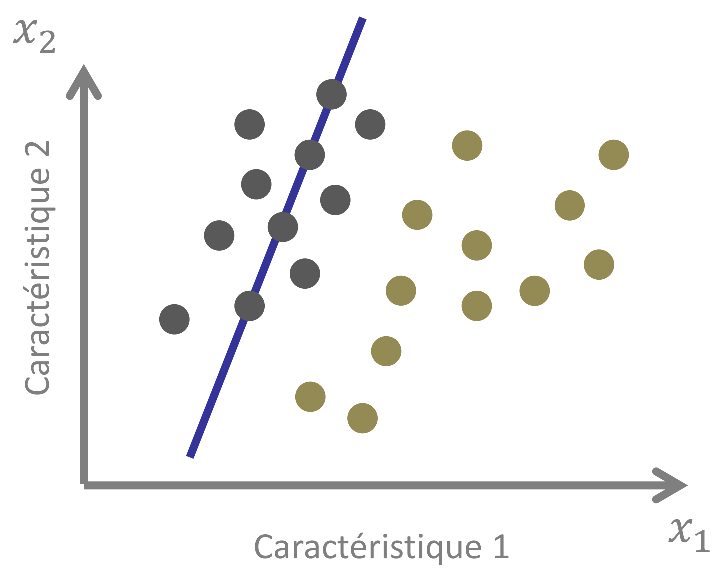
$\}$

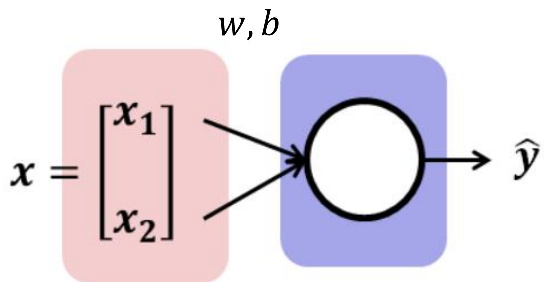


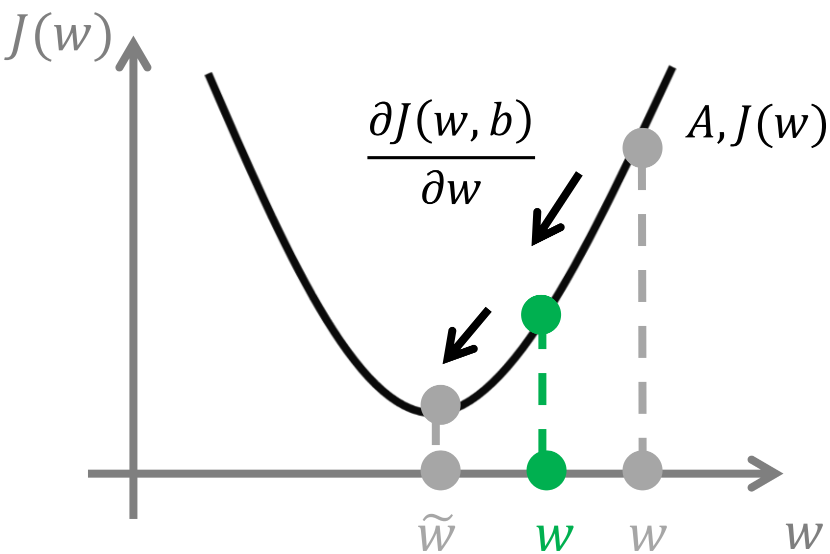

w, b
$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
$\hat{y}$


$J(w)$
$\frac{\partial J(w,b)}{\partial w}$
$A, J(w)$
$\tilde{w}$
w
w
w

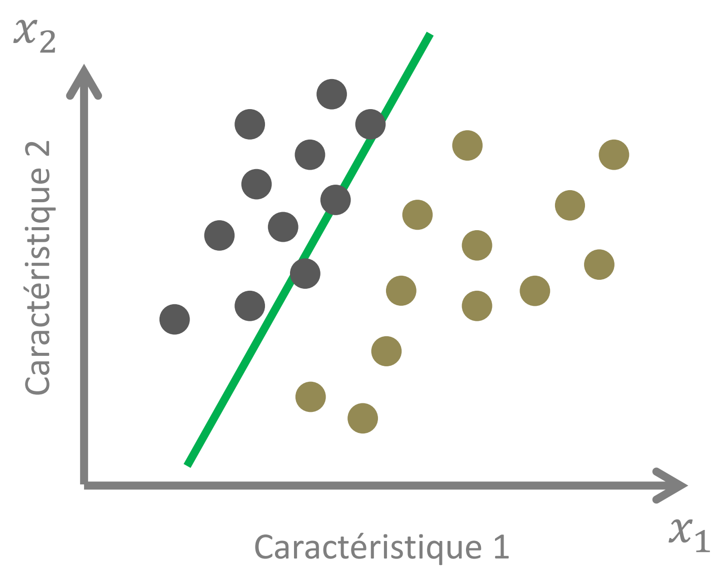
$$w^T x + b = 0$$

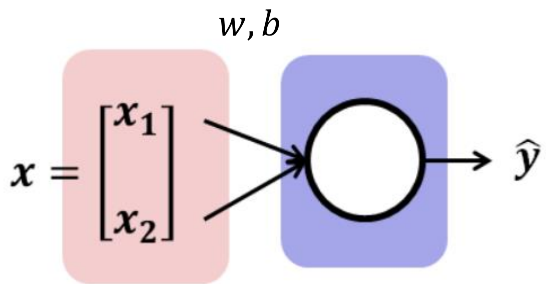

x_2
Caractéristique 2
Caractéristique 1
x_1

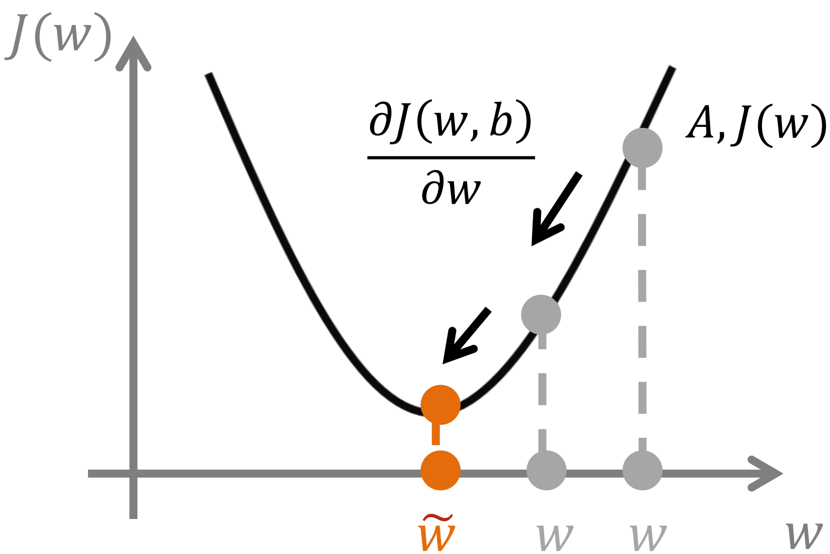

w, b
x = [x1 x2]
ŷ

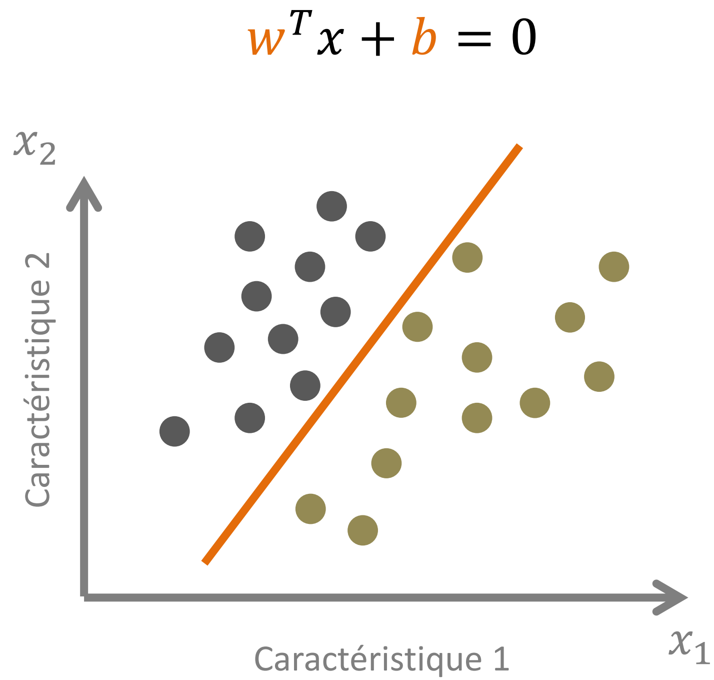

J(w)
∂J(w, b) / ∂w
A, J(w)
w̃
w
w
w

$$w^T x + b = 0$$


x2
Caractéristique 2
Caractéristique 1
x1

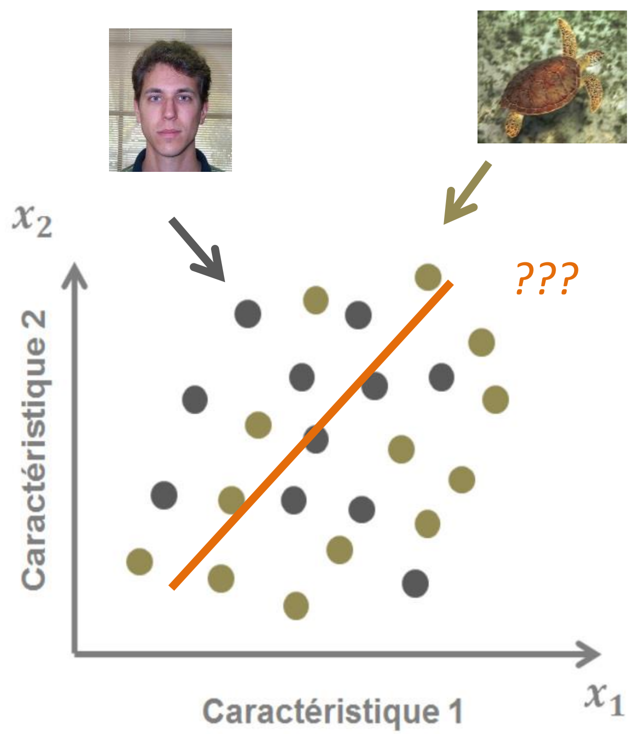

w, b
$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
$\hat{y}$


$J(w)$
$\frac{\partial J(w,b)}{\partial w}$
$A, J(w)$
$\tilde{w}$
w
w
w


$w^T x + b = 0$
x_2
Caractéristique 2
Caractéristique 1
x_1

Problématique

- Comment faire lorsque les données d'entrée sont trop complexes ?



Espace image initial

Les réseaux de neurones

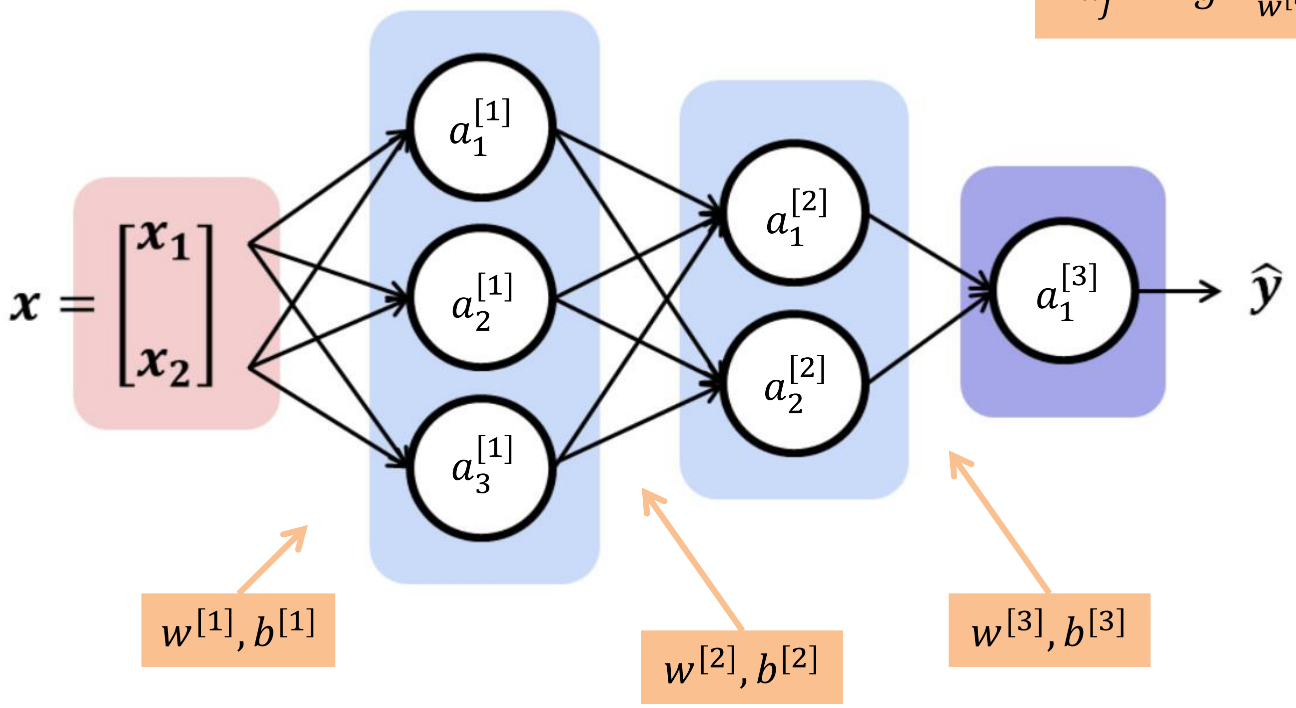
Modélisation par couches de neurones

Concept de base

- Passage d'un neurone à un ensemble de neurones organisés suivant un réseau

$a_j^{[i]}$: activation neurones

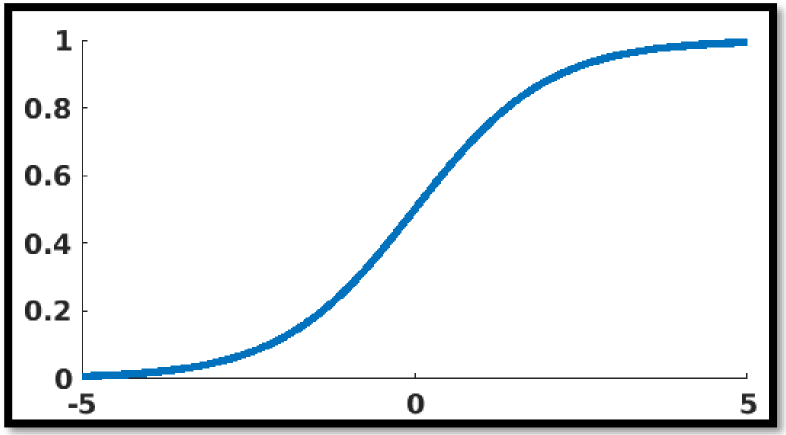
$a_j^{[i]} = g^{[i]}{}_{w^{[i]},b^{[i]}}(a^{[i-1]})$



Fonctions d'activation

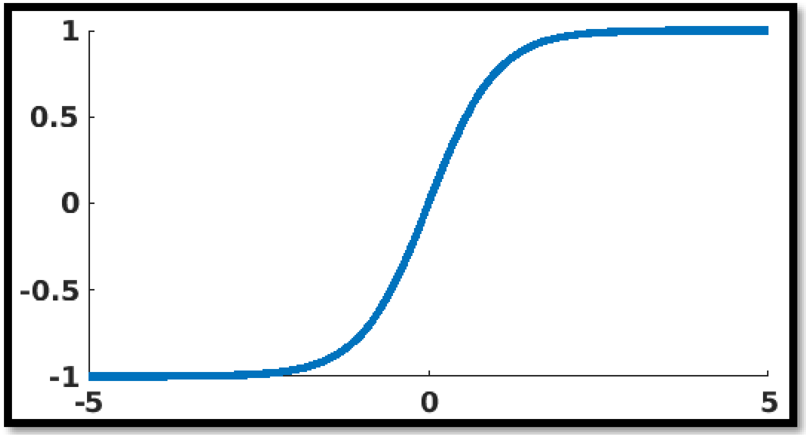
- Différentes fonctions d'activation peuvent être utilisées au travers des couches

Sigmoïde



$$g^{[l]}(z) = \frac{1}{1 + e^{-z}}$$

Tangente hyperbolique

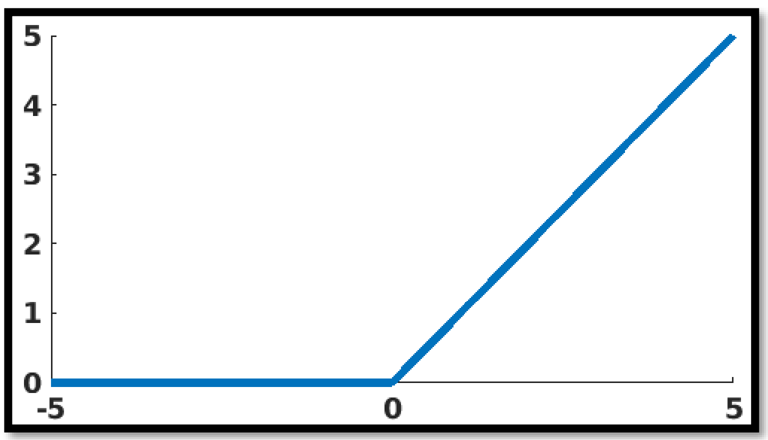


$$g^{[l]}(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

Fonctions d'activation

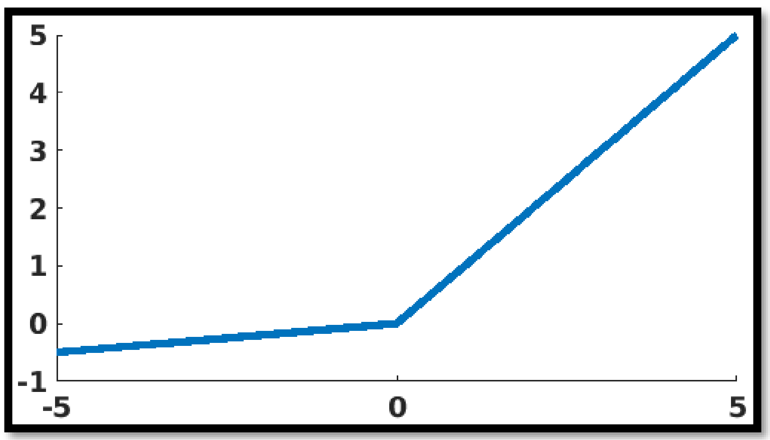
- Différentes fonctions d'activation peuvent être utilisées au travers des couches

Rectified Linear Unit (ReLU)



$$g^{[l]}(z) = max(0, z)$$

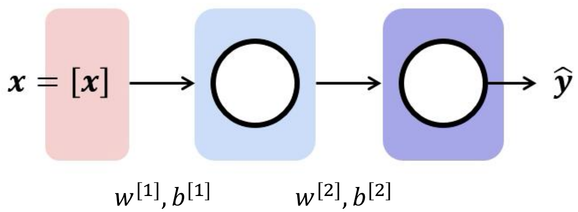
leaky ReLU



$$g^{[l]}(z) = max(0.01 \cdot z, z)$$

Elément de modélisation

- Calculer les dérivées partielles $\frac{\partial J}{\partial b^{[1]}}, \frac{\partial J}{\partial b^{[2]}}$ et $\frac{\partial J}{\partial w^{[1]}}, \frac{\partial J}{\partial w^{[2]}}$ dans le cas du réseau de neurones à deux couches suivant



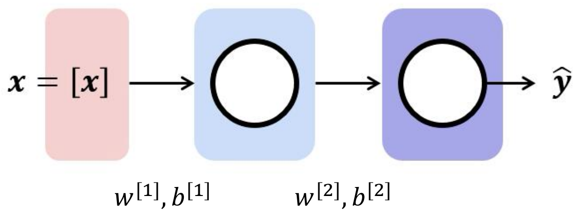
$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

avec $\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) = -\left[y^{(i)}\, log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{y}^{(i)}\right)\right]$

$$\begin{cases} \dfrac{\partial J}{\partial b^{[2]}} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} \dfrac{\partial \mathcal{L}}{\partial b^{[2]}} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\left(\dfrac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} \cdot \dfrac{\partial \hat{y}^{(i)}}{\partial {z^{[2]}}^{(i)}} \cdot \dfrac{\partial {z^{[2]}}^{(i)}}{\partial b^{[2]}}\right) \\[2ex] \dfrac{\partial J}{\partial b^{[1]}} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} \dfrac{\partial \mathcal{L}}{\partial b^{[1]}} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\left(\dfrac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} \cdot \dfrac{\partial \hat{y}^{(i)}}{\partial {z^{[2]}}^{(i)}} \cdot \dfrac{\partial {z^{[2]}}^{(i)}}{\partial {a^{[1]}}^{(i)}} \cdot \dfrac{\partial {a^{[1]}}^{(i)}}{\partial z^{[1]}} \cdot \dfrac{\partial {z^{[1]}}^{(i)}}{\partial b^{[1]}}\right) \end{cases}$$

Elément de modélisation

- Calculer les dérivées partielles $\frac{\partial J}{\partial b^{[1]}}, \frac{\partial J}{\partial b^{[2]}}$ et $\frac{\partial J}{\partial w^{[1]}}, \frac{\partial J}{\partial w^{[2]}}$ dans le cas du réseau de neurones à deux couches suivant



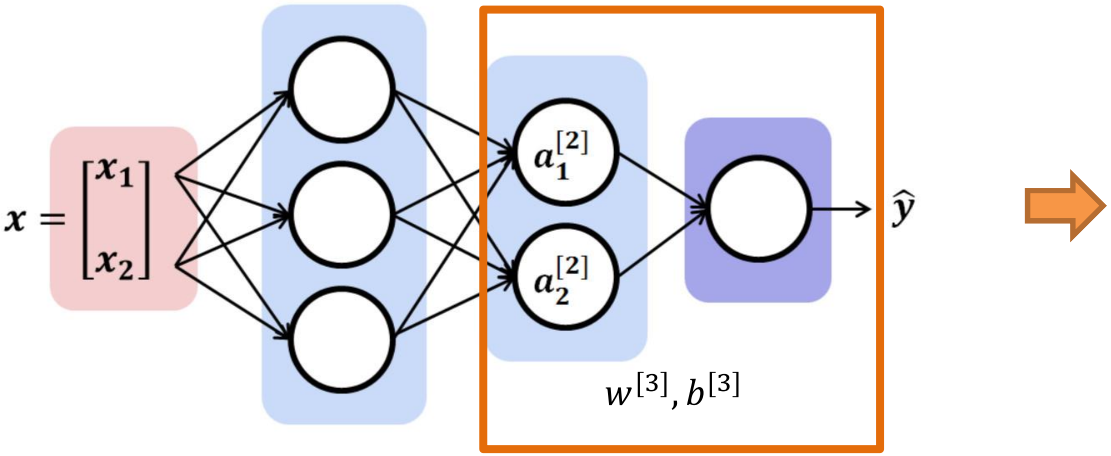
$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

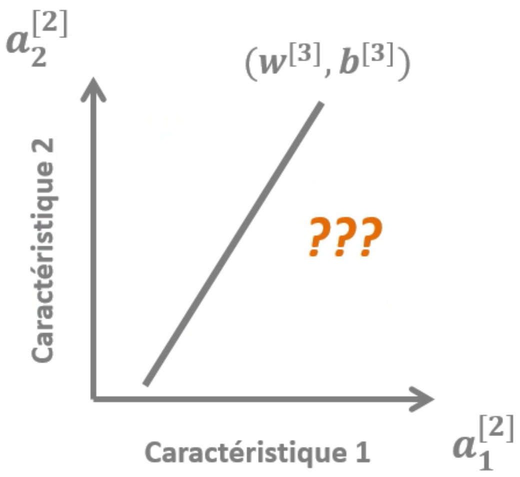
avec $\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) = -\left[y^{(i)}\, log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{y}^{(i)}\right)\right]$

$$\begin{cases} \dfrac{\partial J}{\partial w^{[2]}} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} \dfrac{\partial \mathcal{L}}{\partial w^{[2]}} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\left(\dfrac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} \cdot \dfrac{\partial \hat{y}^{(i)}}{\partial z^{[2]^{(i)}}} \cdot \dfrac{\partial z^{[2]^{(i)}}}{\partial w}\right) \\ \dfrac{\partial J}{\partial w^{[1]}} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} \dfrac{\partial \mathcal{L}}{\partial w^{[1]}} = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m}\left(\dfrac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} \cdot \dfrac{\partial \hat{y}^{(i)}}{\partial z^{[2]^{(i)}}} \cdot \dfrac{\partial z^{[2]^{(i)}}}{\partial a^{[1]^{(i)}}} \cdot \dfrac{\partial a^{[1]^{(i)}}}{\partial z^{[1]}} \cdot \dfrac{\partial z^{[1]^{(i)}}}{\partial w^{[1]}}\right) \end{cases}$$

Concept de base

- Apprentissage simultané d'un espace d'informations discriminantes et d'une fonction de décision associée





Apprentissage de la fonction de décision

- Phase d'apprentissage

Initialisation des $w^{[l]}$ et $b^{[l]}$

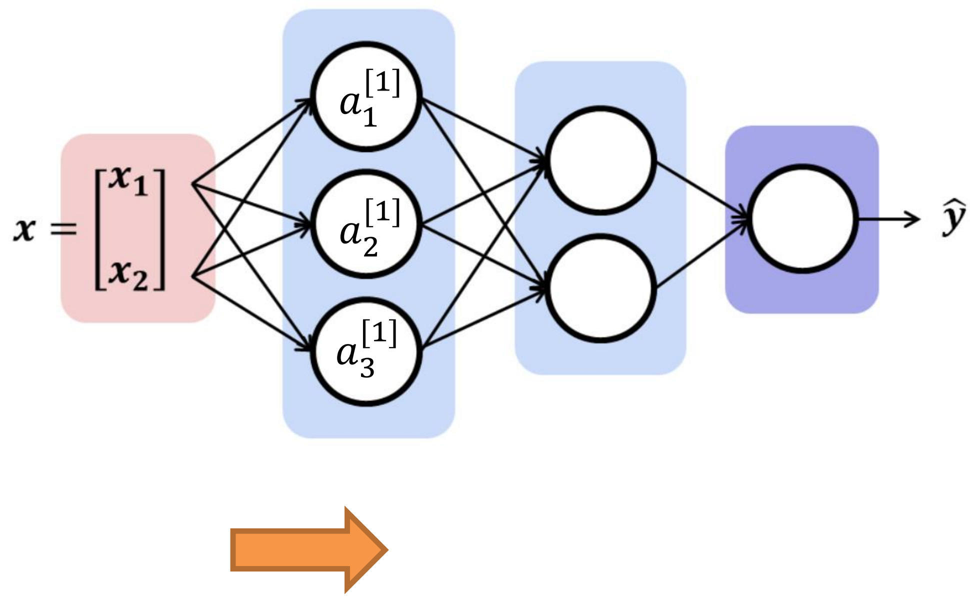
Répéter jusqu'à convergence

{

propagation avant

propagation arrière

}



Apprentissage de la fonction de décision

- Phase d'apprentissage

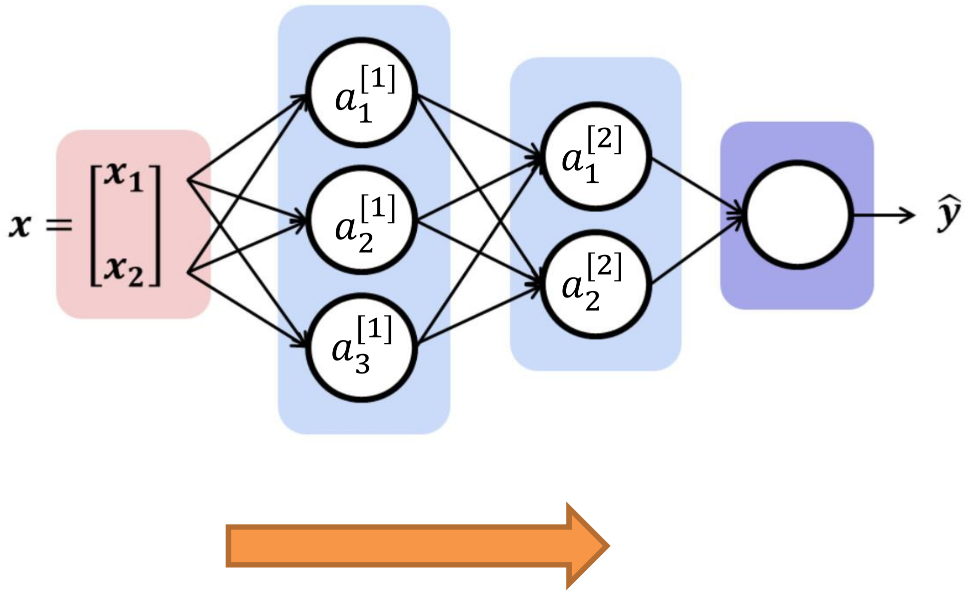
Initialisation des $w^{[l]}$ et $b^{[l]}$

Répéter jusqu'à convergence
{

propagation avant

propagation arrière

}



Apprentissage de la fonction de décision

- Phase d'apprentissage

Initialisation des $w^{[l]}$ et $b^{[l]}$

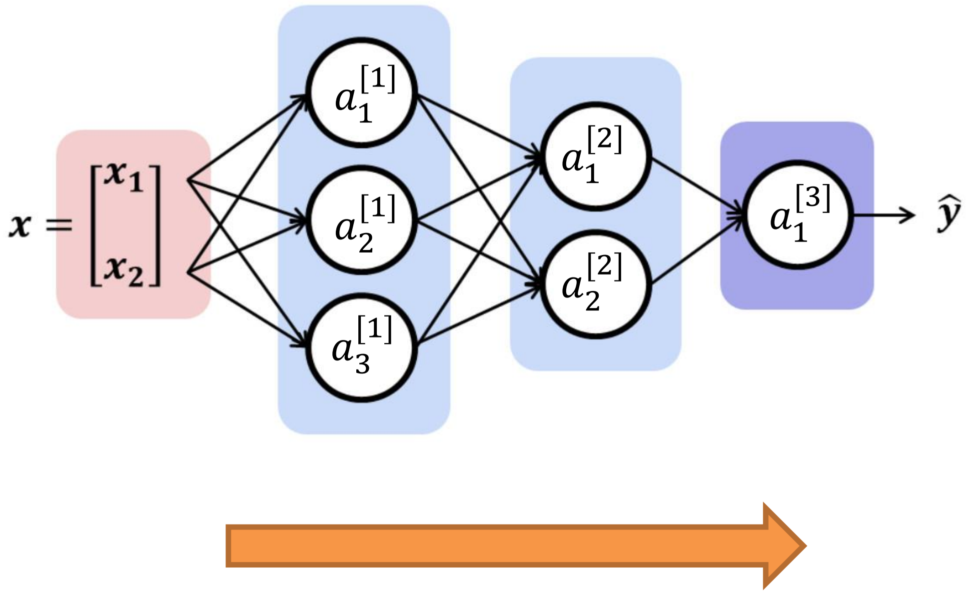
Répéter jusqu'à convergence

{

propagation avant

propagation arrière

}



Apprentissage de la fonction de décision

- Phase d'apprentissage

Initialisation des $w^{[l]}$ et $b^{[l]}$

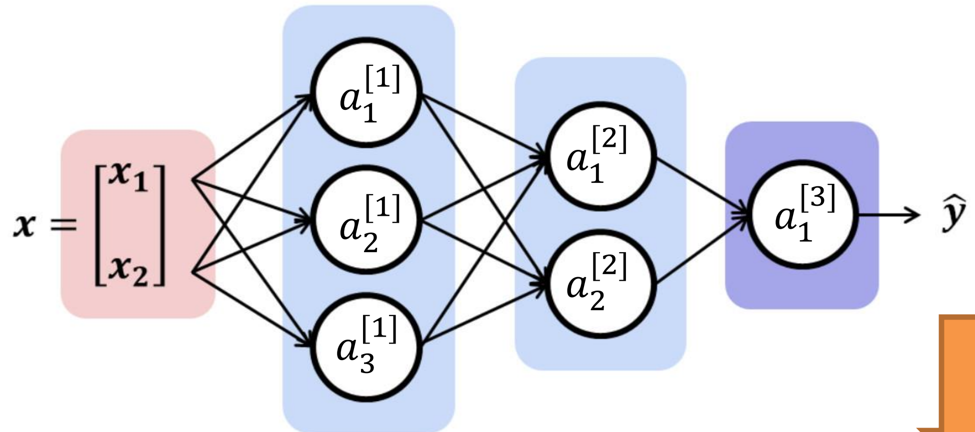
Répéter jusqu'à convergence

{

propagation avant

propagation arrière

}



Calcul de $J(w, b) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$

Apprentissage de la fonction de décision

- Approche empirique - Descente de gradient

Initialisation des $w^{[l]}$ et $b^{[l]}$

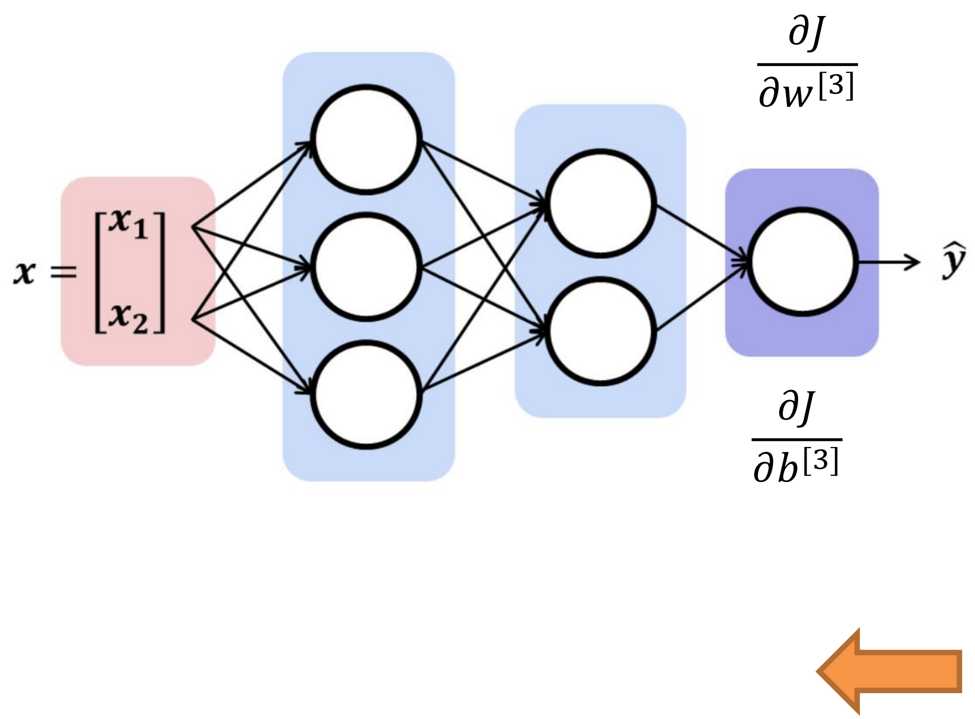
Répéter jusqu'à convergence

{

propagation avant

propagation arrière

}



Apprentissage de la fonction de décision

- Approche empirique - Descente de gradient

Initialisation des $w^{[l]}$ et $b^{[l]}$

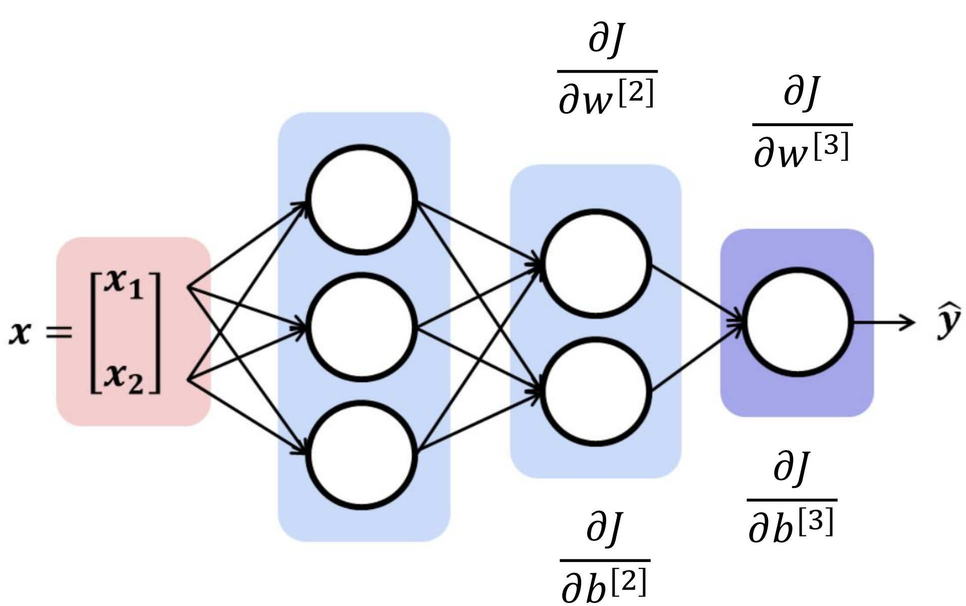
Répéter jusqu'à convergence

{

propagation avant

propagation arrière

}



Apprentissage de la fonction de décision

▶ Approche empirique - Descente de gradient

Initialisation des $w^{[l]}$ et $b^{[l]}$

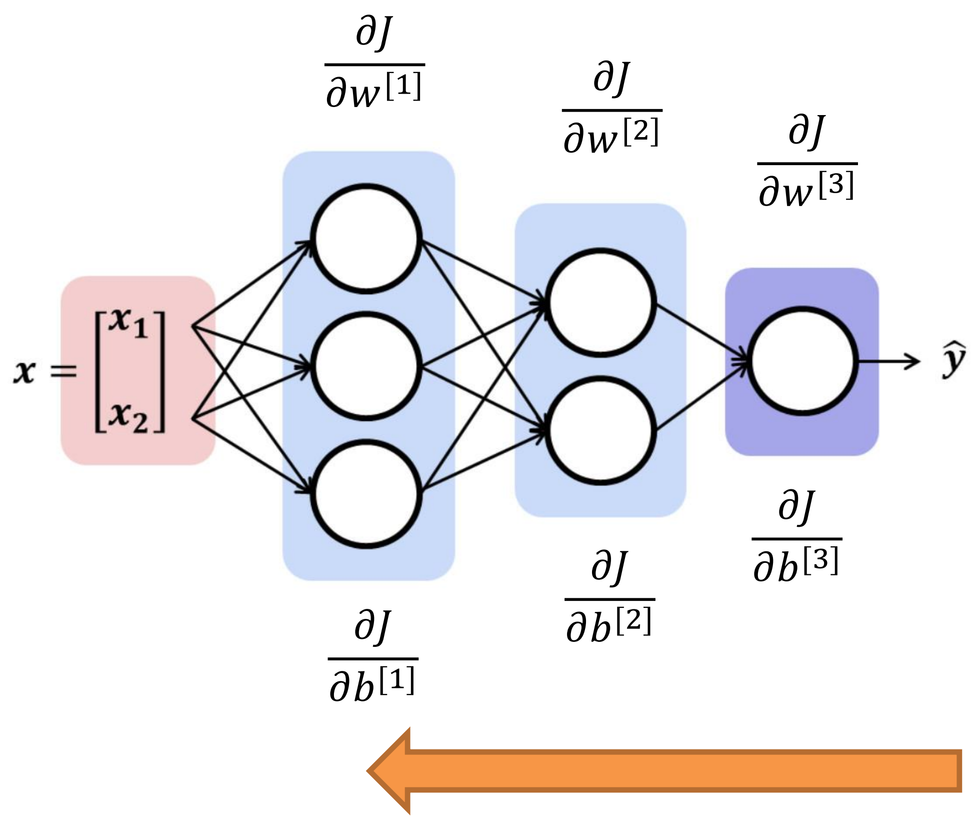
Répéter jusqu'à convergence

{

propagation avant

propagation arrière

}



Apprentissage de la fonction de décision

- Approche empirique - Descente de gradient

Initialisation des $w^{[l]}$ et $b^{[l]}$

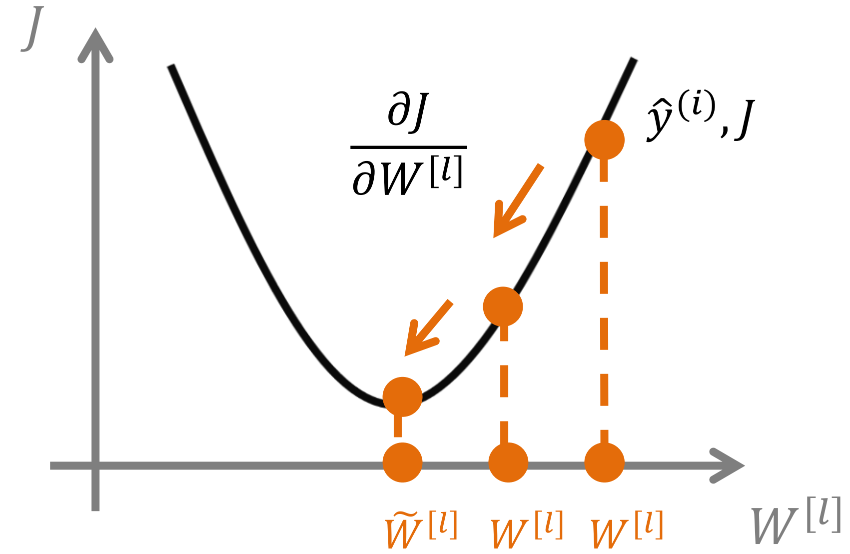
Répéter jusqu'à convergence

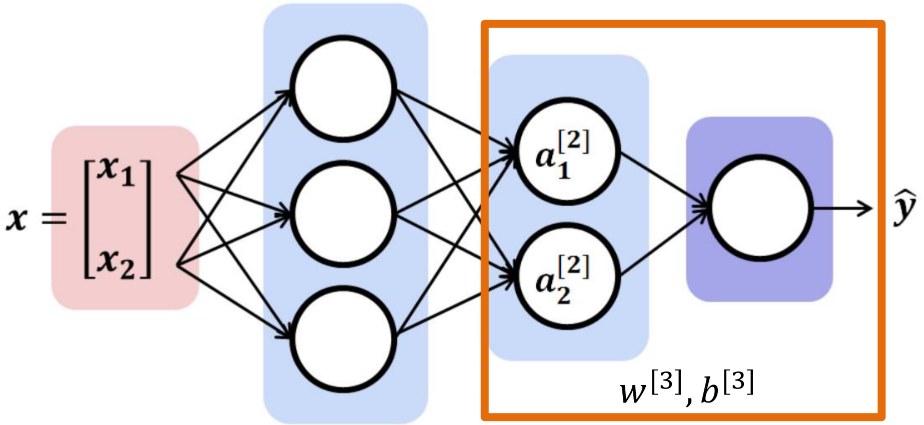
{

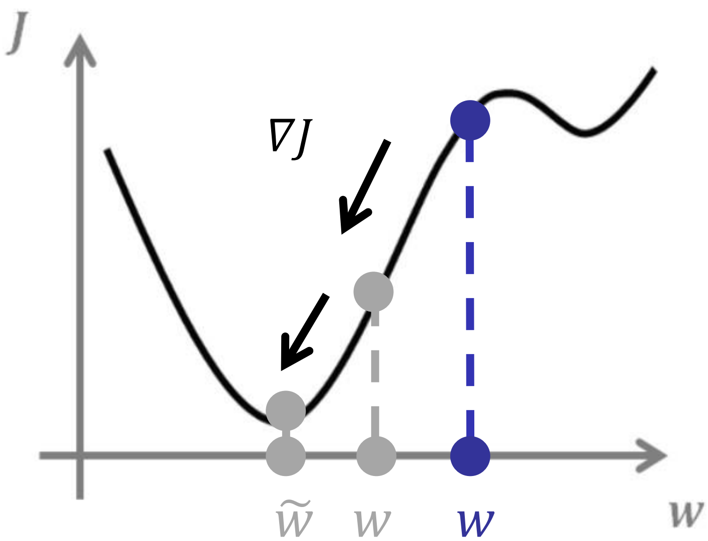
propagation avant

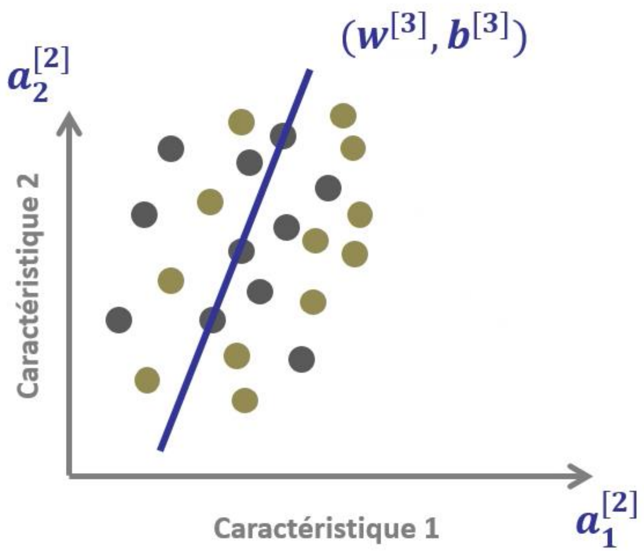
propagation arrière

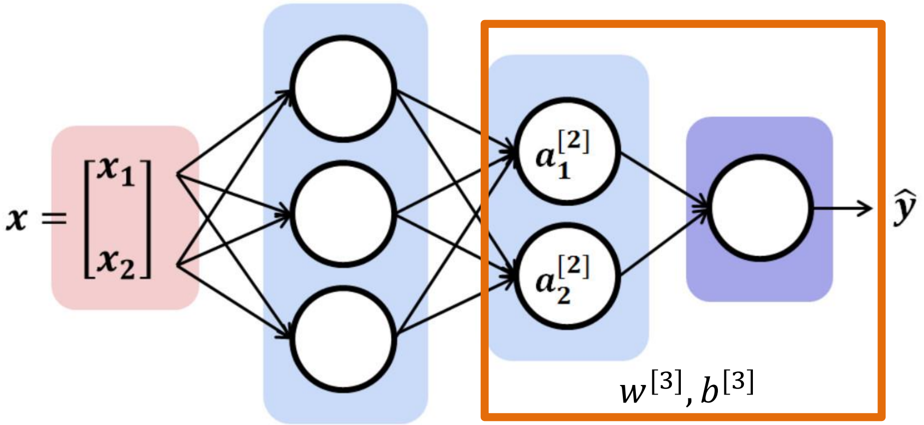
}

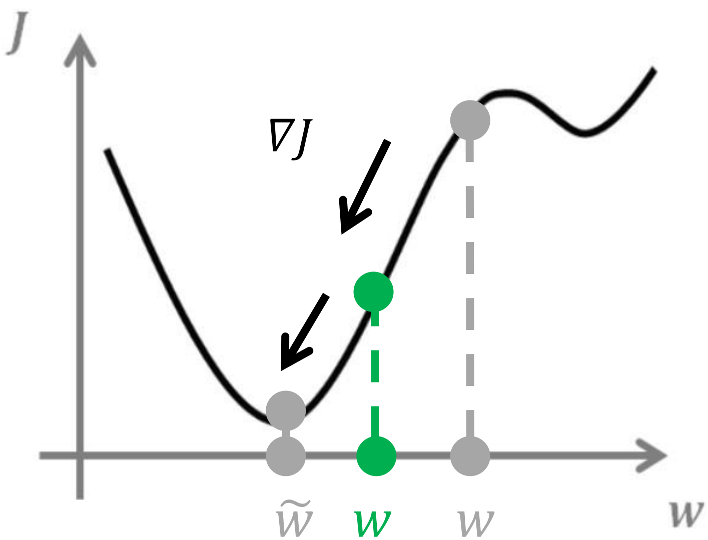


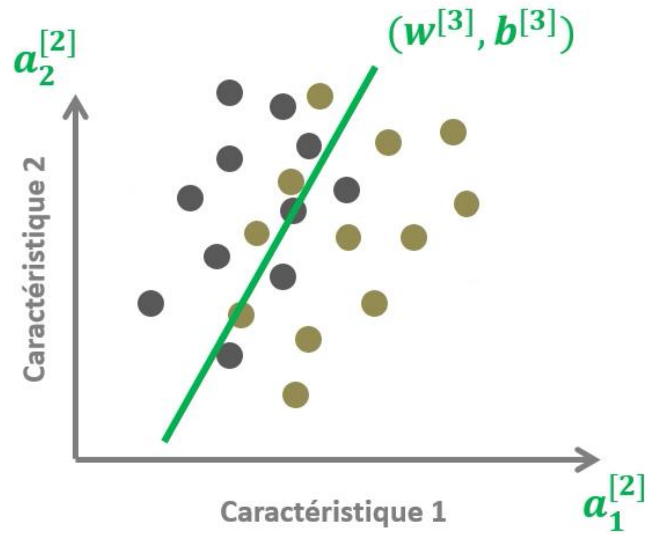

$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
$a_1^{[2]}$
$a_2^{[2]}$
$\hat{y}$
$w^{[3]}, b^{[3]}$

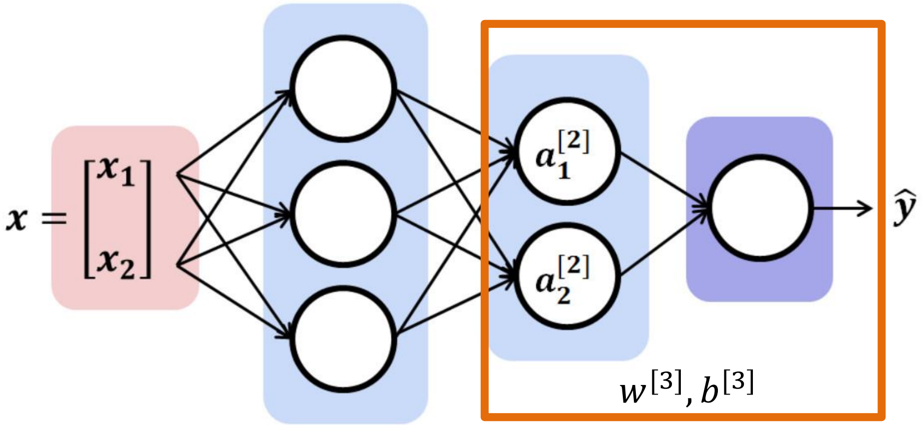

J
∇J
$\tilde{w}$
w
w
w

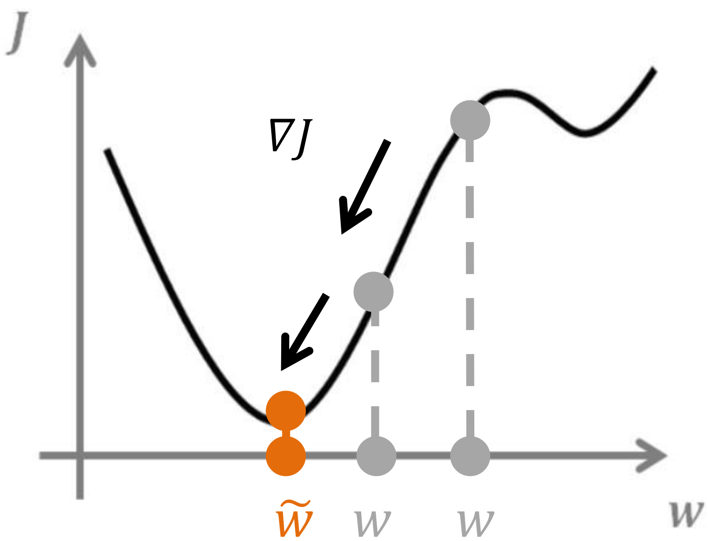

$(w^{[3]}, b^{[3]})$
$a_2^{[2]}$
Caractéristique 2
Caractéristique 1
$a_1^{[2]}$

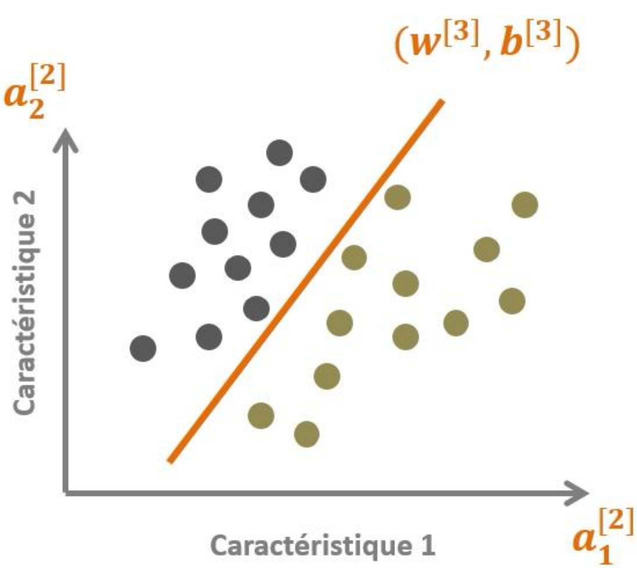

$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
$a_1^{[2]}$
$a_2^{[2]}$
$\hat{y}$
$w^{[3]}, b^{[3]}$


J
∇J
$\tilde{w}$
w
w
w

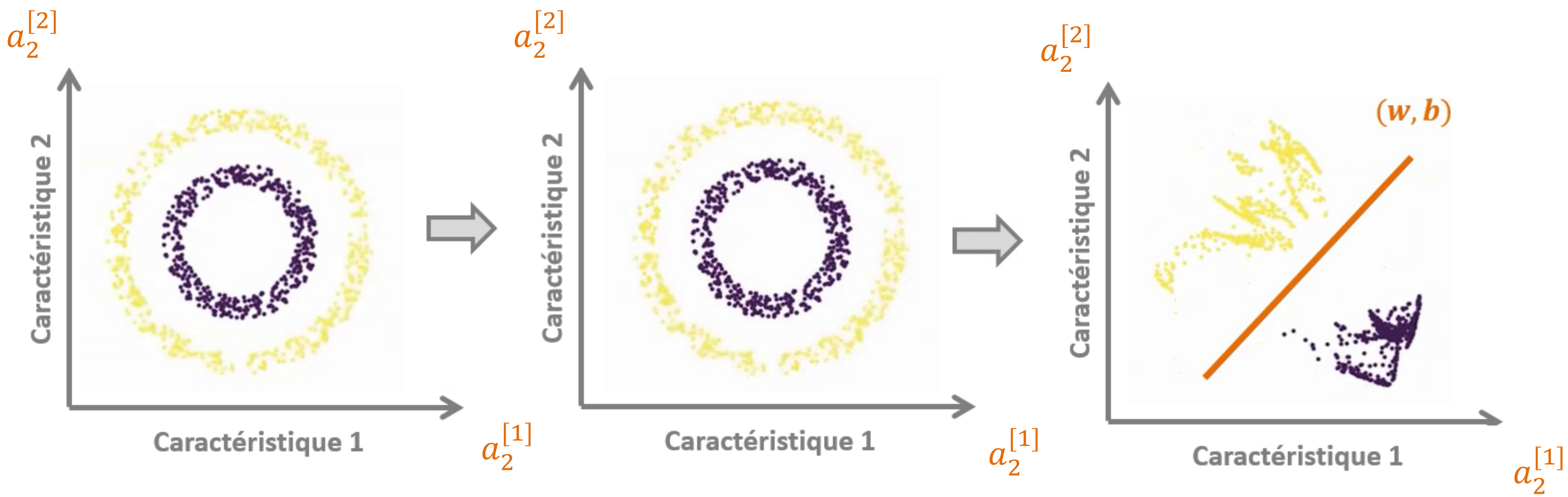

$(w^{[3]}, b^{[3]})$
$a_2^{[2]}$
Caractéristique 2
Caractéristique 1
$a_1^{[2]}$


$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
$a_1^{[2]}$
$a_2^{[2]}$
$\hat{y}$
$w^{[3]}, b^{[3]}$


J
∇J
$\widetilde{w}$
w
w
w


$(w^{[3]}, b^{[3]})$
$a_2^{[2]}$
Caractéristique 2
Caractéristique 1
$a_1^{[2]}$

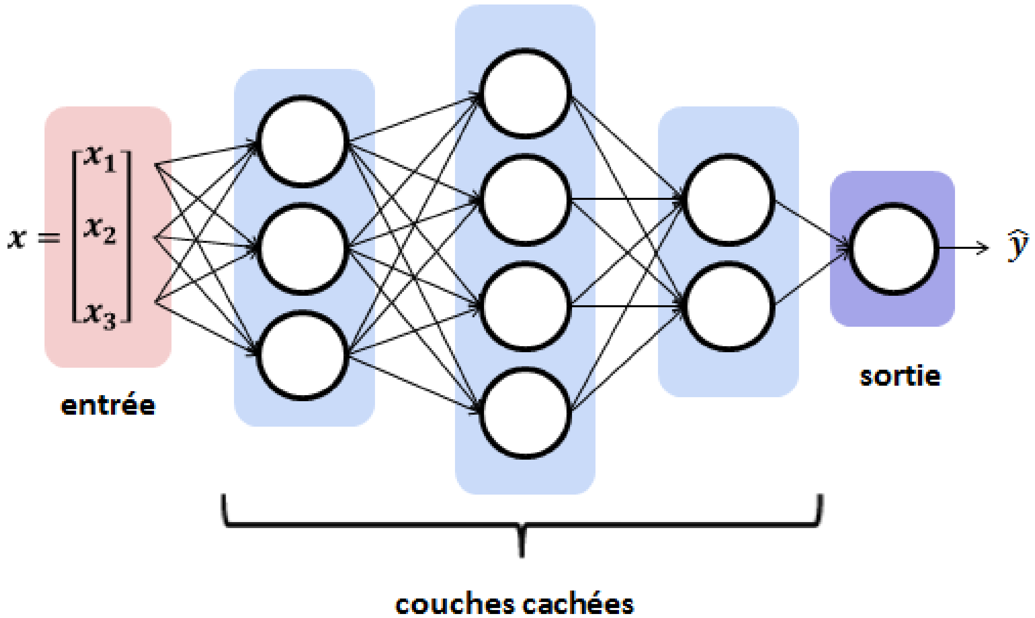
Evolution de l’espace des caractéristiques de la dernière couche au cours des itérations



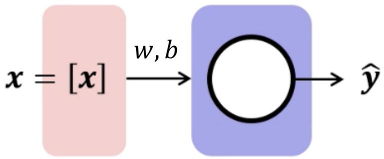
https://slideslive.com/38959153/momentum-residual-neural-networks?ref=speaker-83006

Et maintenant jouons !

https://playground.tensorflow.org



That's all folks



$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

$$\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$
$$= -\left[y^{(i)}\, log\left(\hat{y}^{(i)}\right)\right.$$
$$\left.+ \left(1 - y^{(i)}\right) log\left(1 - \hat{y}^{(i)}\right)\right]$$

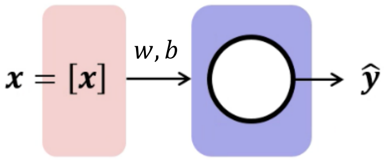
$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m} \frac{\partial \mathcal{L}}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} \cdot \frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}} \cdot \frac{\partial z^{(i)}}{\partial b}\right)$$

$$\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) = -\left[y^{(i)}\, log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{y}^{(i)}\right)\right]$$

$$\Rightarrow \quad \frac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} = -\frac{y^{(i)}}{\hat{y}^{(i)}} + \frac{1 - y^{(i)}}{1 - \hat{y}^{(i)}}$$

$$\hat{y}^{(i)} = g\left(z^{(i)}\right) = \frac{1}{1 + e^{-z^{(i)}}} \quad \Rightarrow \quad \frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}} = \hat{y}\left(1 - \hat{y}^{(i)}\right)$$

$$z^{(i)} = wx^{(i)} + b \quad \Rightarrow \quad \frac{\partial z^{(i)}}{\partial b} = 1$$



$$\boxed{J(w,b) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)}$$

$$\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) = -\left[y^{(i)}\, log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{y}^{(i)}\right)\right]$$

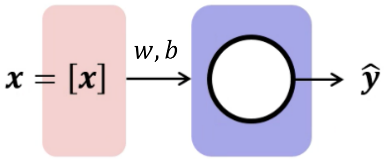
$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} \cdot \frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}} \cdot \frac{\partial z^{(i)}}{\partial b}\right)$$

$$\frac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} = -\frac{y^{(i)}}{\hat{y}^{(i)}} + \frac{1 - y^{(i)}}{1 - \hat{y}^{(i)}}$$

$$\frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}} = \hat{y}\left(1 - \hat{y}^{(i)}\right)$$

$$\frac{\partial z^{(i)}}{\partial b} = 1$$

$$\Rightarrow \quad \boxed{\frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}^{(i)} - y^{(i)}\right)}$$



$$\boxed{J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)}$$

$$\begin{aligned}&\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)\\ &= -\left[y^{(i)}\, log\left(\hat{y}^{(i)}\right)\right.\\ &\left.+ \left(1 - y^{(i)}\right) log\left(1 - \hat{y}^{(i)}\right)\right]\end{aligned}$$

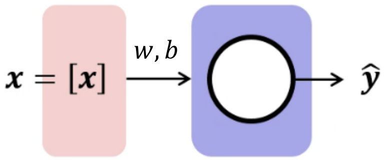
$$\frac{\partial J}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} \cdot \frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}} \cdot \frac{\partial z^{(i)}}{\partial w}\right)$$

$$\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) = -\left[y^{(i)}\, log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{y}^{(i)}\right)\right]$$

$$\Rightarrow \quad \frac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} = -\frac{y^{(i)}}{\hat{y}^{(i)}} + \frac{1 - y^{(i)}}{1 - \hat{y}^{(i)}}$$

$$\hat{y}^{(i)} = g\left(z^{(i)}\right) = \frac{1}{1 + e^{-z^{(i)}}} \quad \Rightarrow \quad \frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}} = \hat{y}\left(1 - \hat{y}^{(i)}\right)$$

$$z^{(i)} = wx^{(i)} + b \quad \Rightarrow \quad \frac{\partial z^{(i)}}{\partial w} = x^{(i)}$$



$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right)$$

$$\mathcal{L}\left(\hat{y}^{(i)}, y^{(i)}\right) = -\left[y^{(i)}\, log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) log\left(1 - \hat{y}^{(i)}\right)\right]$$

$$\frac{\partial J}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial \mathcal{L}}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} \cdot \frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}} \cdot \frac{\partial z^{(i)}}{\partial w}\right)$$

$$\frac{\partial \mathcal{L}}{\partial \hat{y}^{(i)}} = -\frac{y^{(i)}}{\hat{y}^{(i)}} + \frac{1 - y^{(i)}}{1 - \hat{y}^{(i)}}$$

$$\frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}} = \hat{y}\left(1 - \hat{y}^{(i)}\right)$$

$$\frac{\partial z^{(i)}}{\partial w} = x^{(i)}$$

$$\Rightarrow \quad \boxed{\frac{\partial J}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}^{(i)} - y^{(i)}\right) \cdot x^{(i)}}$$