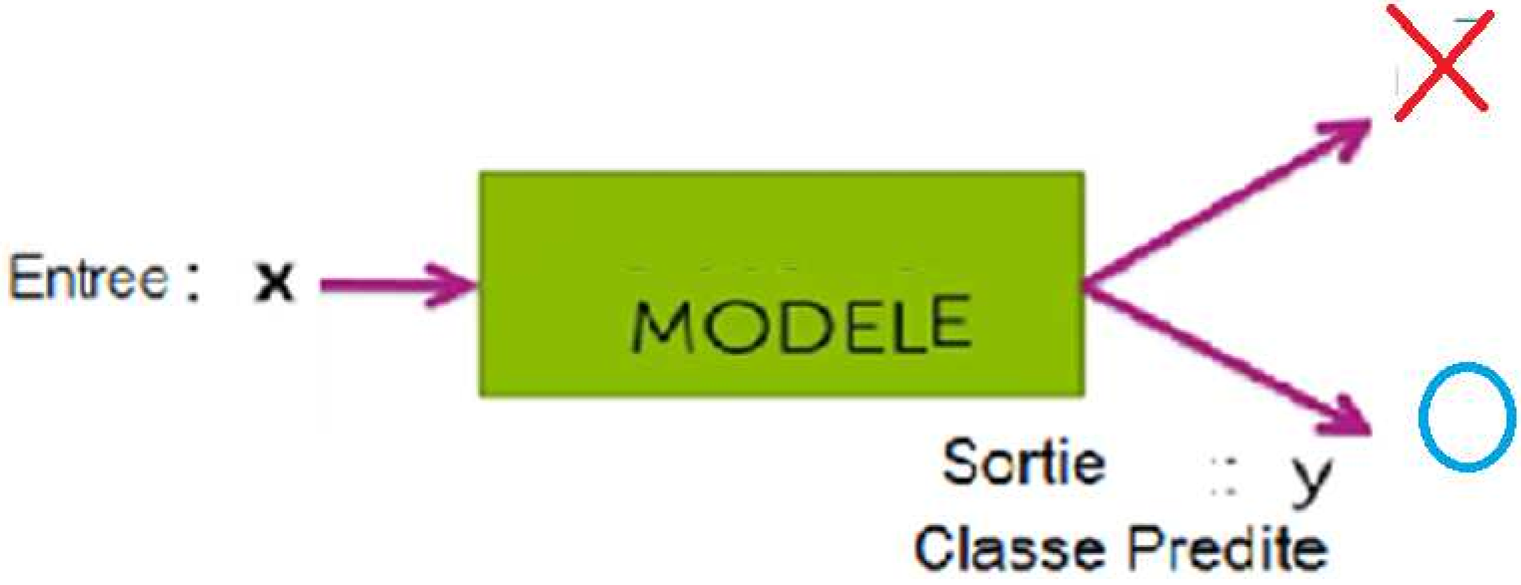
Chapitre III : Analyse Prédictive :

(Prédire une Catégorie : Classification)

III- 3- Réseau de Neurone

Introduction :

Classifieur



Introduction:

<u>Exemple 1:</u>

Si on considère le problème de classification de logement: avec les variables d'entrées suivantes:

x1=Superficie; x2=nbre SDB;

x3=nbre Chambres;

x4= age maison ;

x5=nbre niveaux….=>**n=100 vars**

- On veut créer un modèle de régression logistique pour prédire la classe d'un logement.
- Si features avec 2 vars ➔ le nombre des features va augmenter en $O(n^2)$ et c en fait plus proche de $n^2/2$.

➔DONC : nb features=5000.

Puisque on doit trouver $g(x_1x_2+\ldots.x_1x_{100}+x_2x_3+\ldots x_2x_{100}+\ldots..)$

Introduction:

Exemple 1 suite:

➢On peut considérer : $x_1^2, x_2^2 \ldots x_{100}^2$

✓mais ceci ne va pas bien séparer les clases,.

➢On peut aussi considérer les features avec 3 vars :$x_1x_2x_3$, $x_9x_{10}x_{17}$… donc n^3 features.

✓ceci va donner encore plus de features.

➔Donc ce n'est pas une bonne idée d'augmenter le nombre de features.

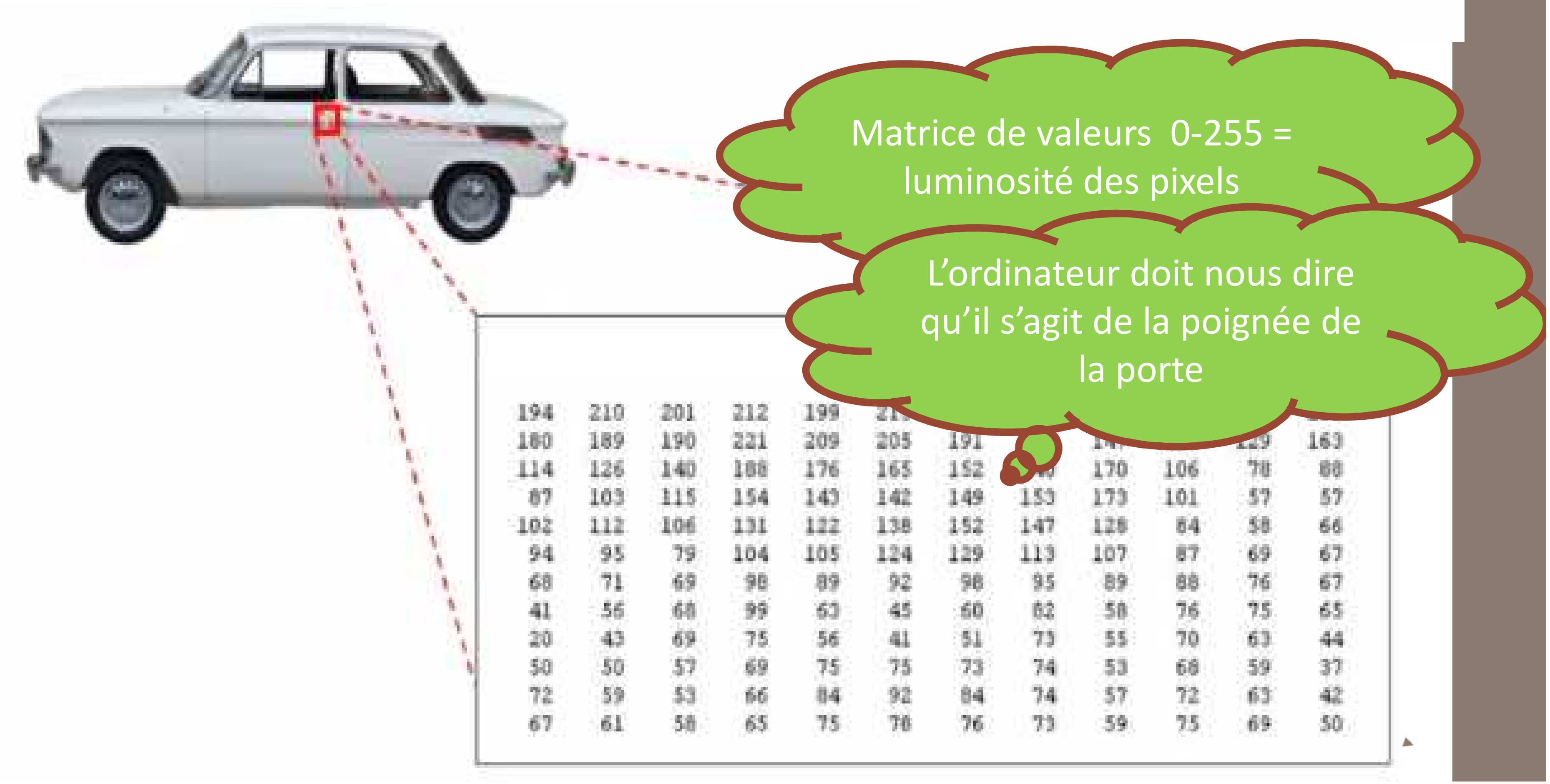
➔Ceci peut amener à un surapprentissage en plus que les calculs deviennent **très complexes**.

Introduction:

Exemple 2:

Problème plus complexe :

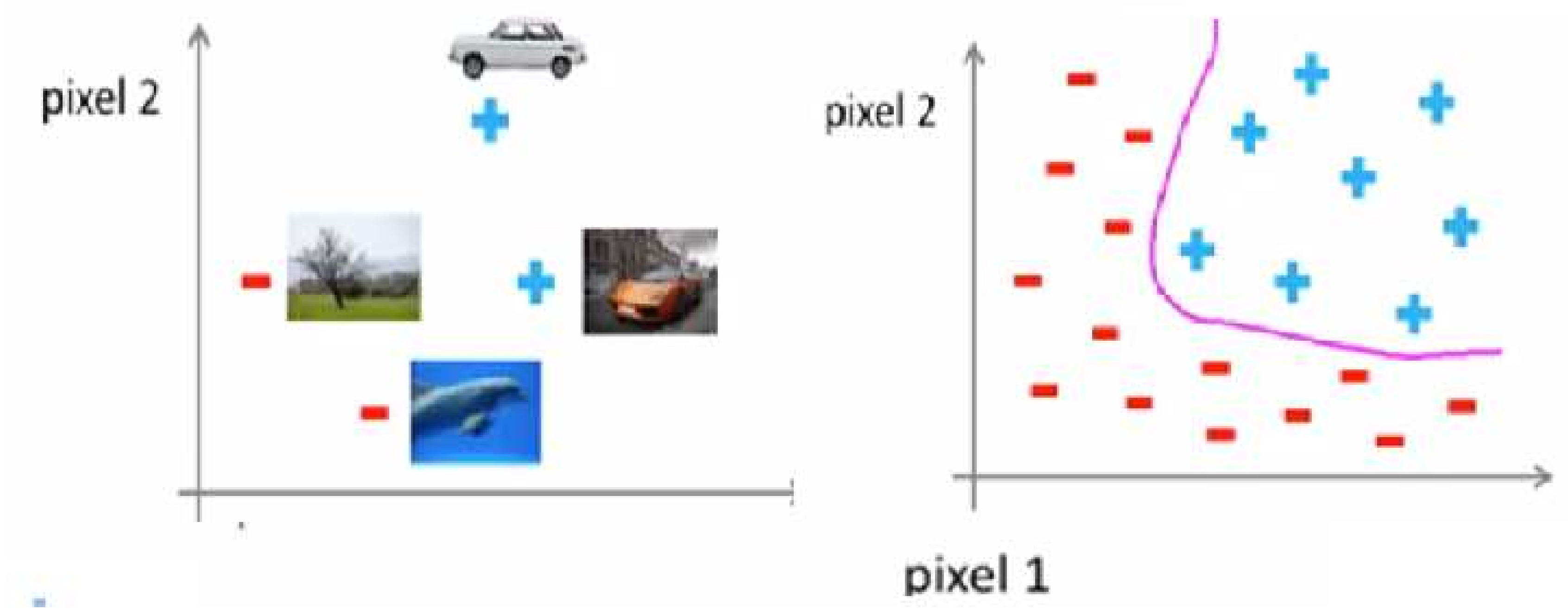
La vision par ordinateur



194	210	201	212	199	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
180	189	190	221	209	205	191	[illegible]	[illegible]	[illegible]	[illegible]	163
114	126	140	188	176	165	152	[illegible]	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	62	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Test :

➢on projette les images selon deux pixels on obtient le graphe suivant:



➢on aura besoin d'une hypothèse **non linéaire** pour séparer entre les deux classes.

Explication:

Image =ensemble pixels chaque pixel a une intensité :

- si Niveau de gris =0..255
- si Couleur =chaque pixel RVB.

Image Carrée NG de 50pixels→ n=2500 vals

- Si on considère des features ($x_i x_j$) quadratiques on aura 3millions de features($n^2/2=2500^2/2$)

→trop grand pour être raisonnable.

→le calcul serait très coûteux.

→Donc, la régression logistique avec ajout de features quadratiques ou cubiques n'est pas une bonne façon d'apprendre des hypothèses non linéaires complexes quand n est grand .

Déduction:

➔**les Réseaux Neurones** sont un bien meilleur moyen d'apprendre des hypothèses complexes, non linéaires, ET lorsque le nombre de features est grand.

Définition d’Un Réseau de Neurones :

C’Est un classifieur non linéaire utilisé pour apprendre des hypothèses complexes quand le nombre de features est très grand.

-Les réseaux de neurones sont un algorithme assez ancien qui était à l'origine motivé par le but d'avoir des machines qui peuvent imiter le cerveau.

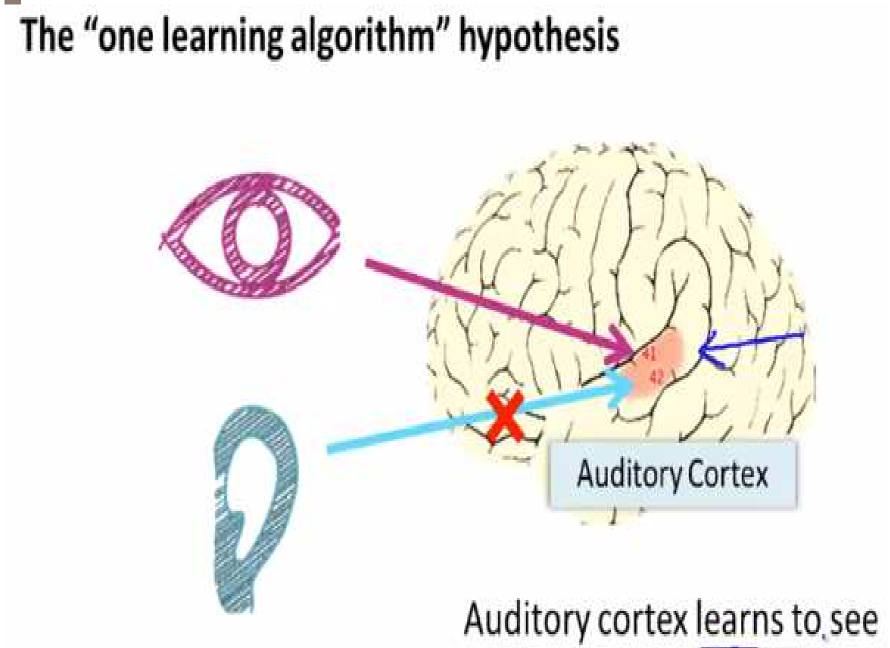
Historique:

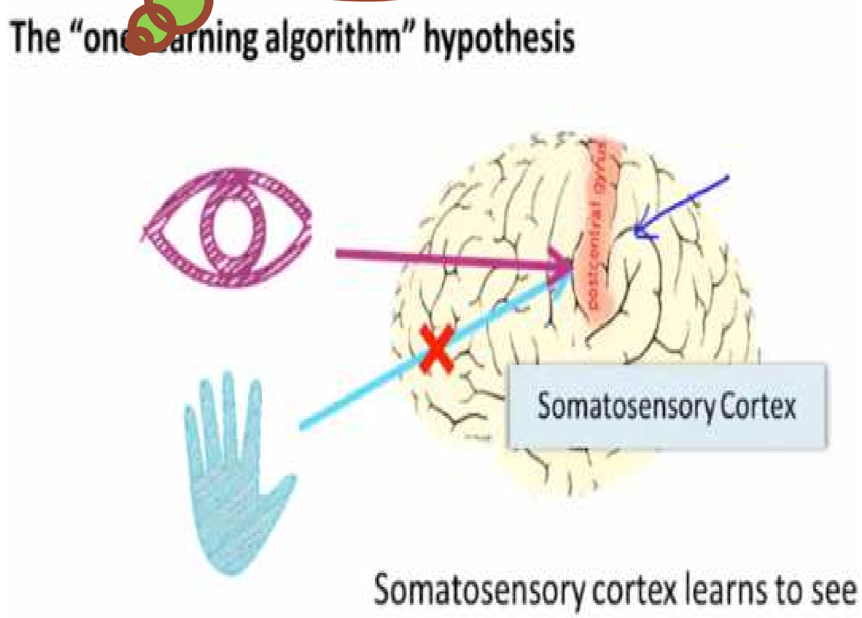
Les réseaux neuronaux ont été très largement utilisés tout au long des années 1980 et 1990 mais leur popularité diminuait à la fin des années 90.
-Récemment les RN ont resurgi car les ordinateurs sont devenus assez rapides pour fonctionner vraiment à grande échelle

ORIGINE:

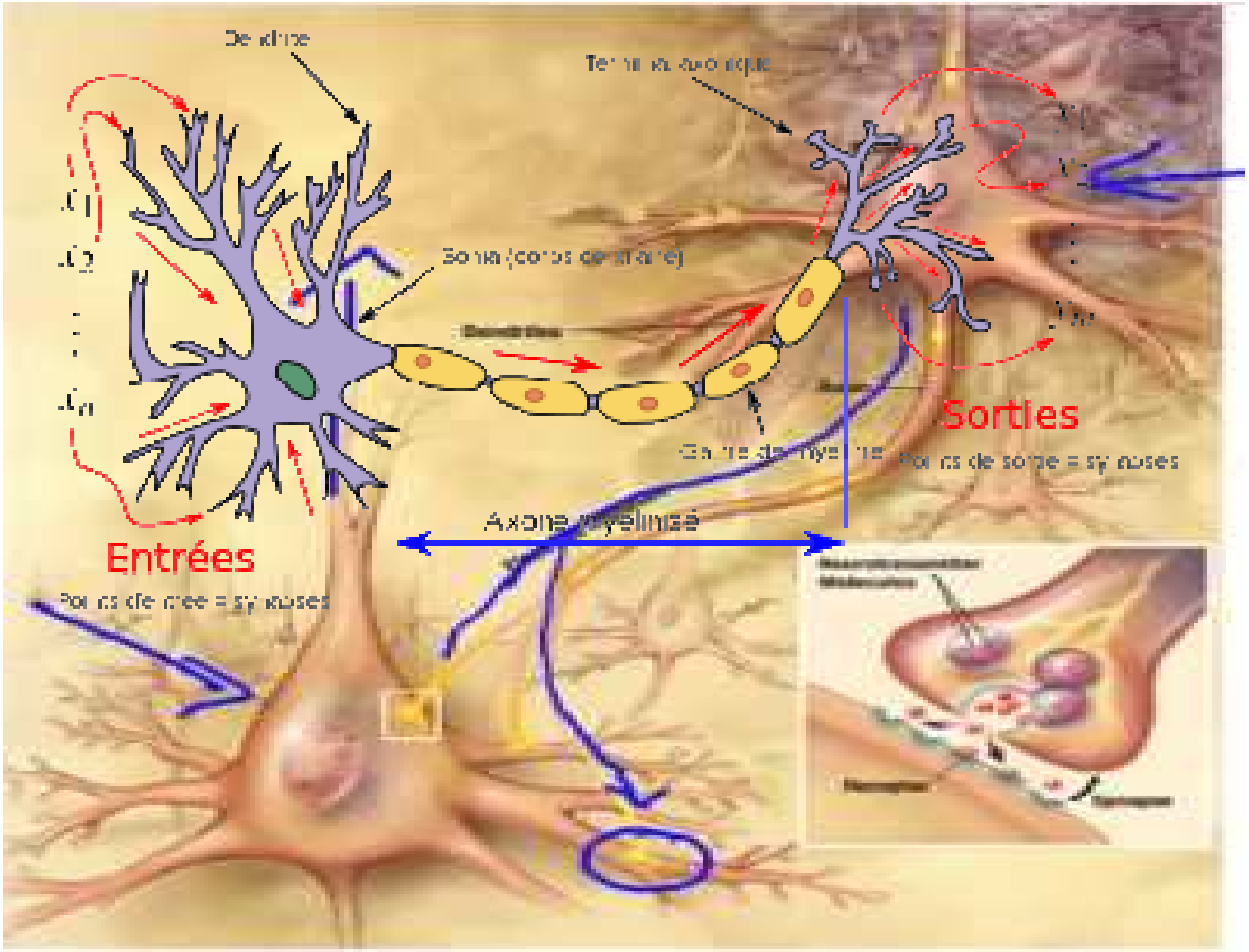
Un Réseau neuronal Artificiel est inspirée du fonctionnement des neurones bi... qui par la suite s'est r... statistiques.

De la même façon on voudrait imiter le cerveau et écrire un seul algorithme d'apprentissage et non plusieurs

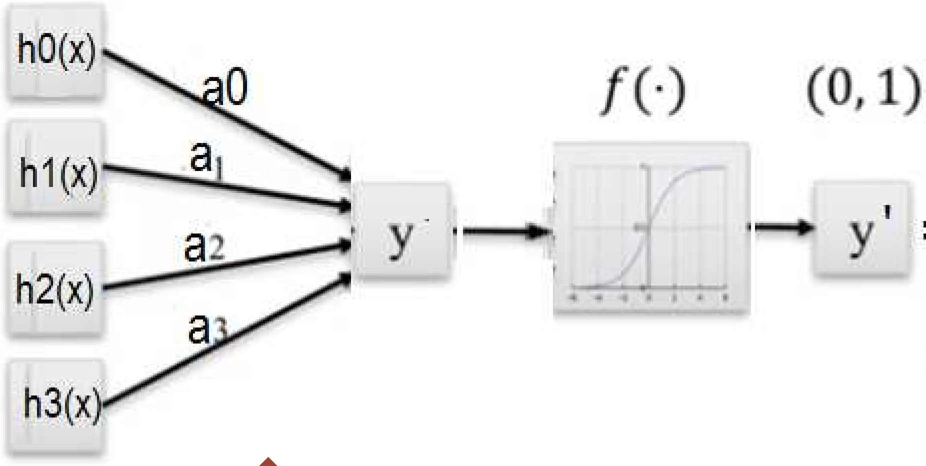




Neurone biologique:

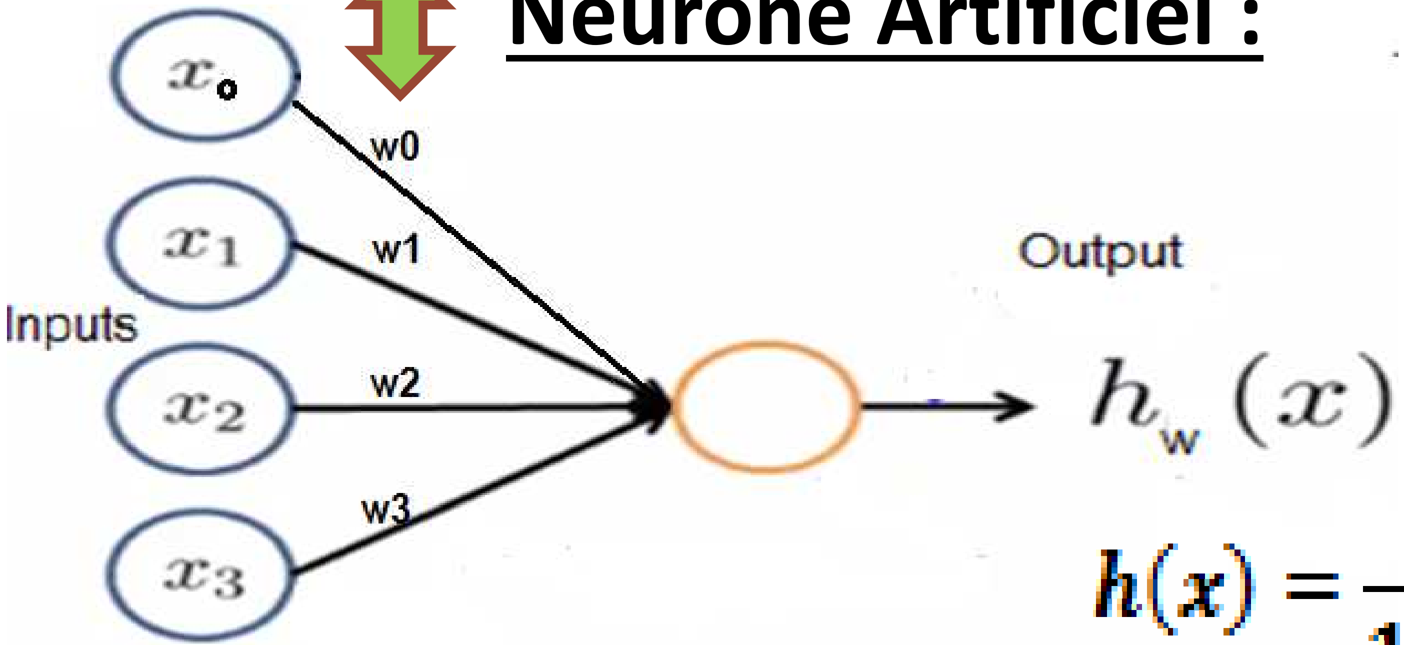


Régression Logistique :



$$=g(a^t H(x_i)) = \frac{1}{1+\ e^{-atH(x)}}$$

Neurone Artificiel :



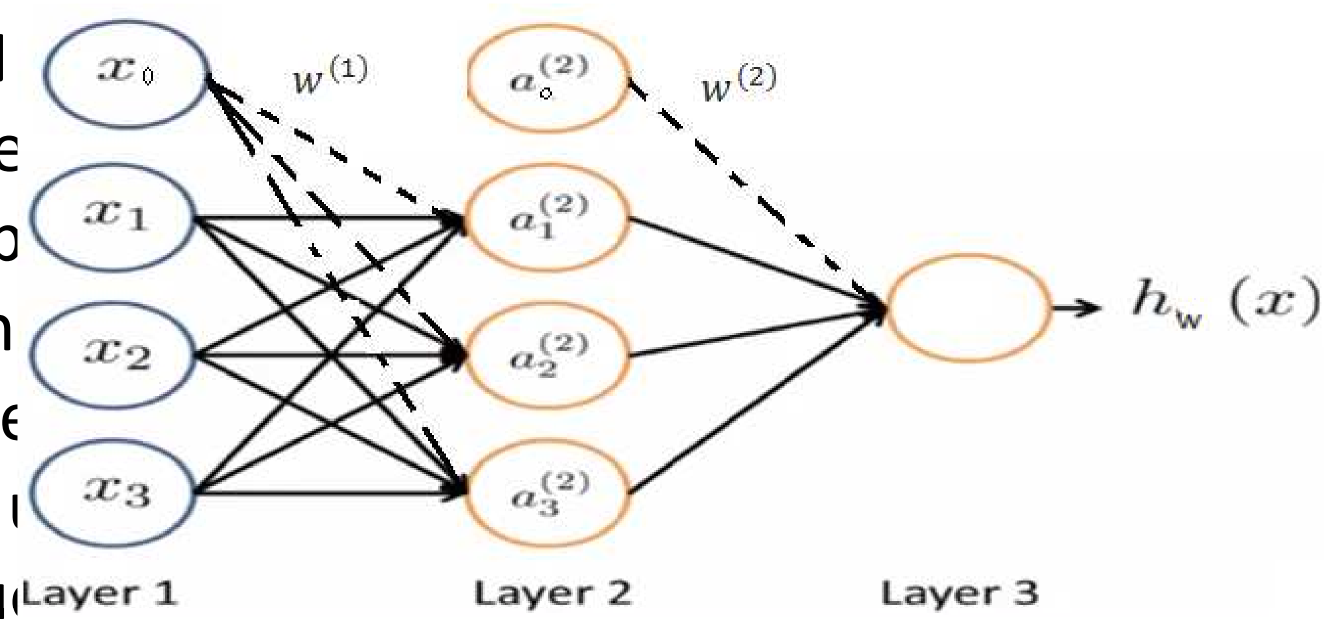
$$h(x) = \frac{1}{1+\ e^{-wtx}}$$

Structure de RN Multicouche :

➢Un RN … s;

➢ Ses ne … ouche;

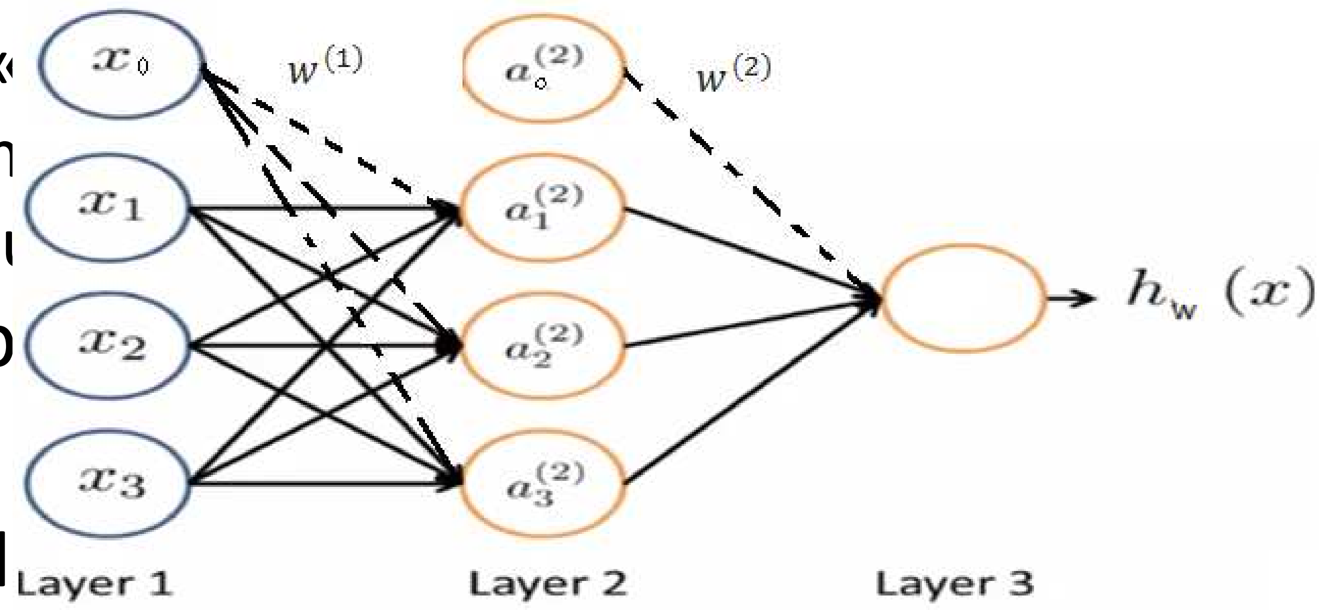
➢ Ce typ … s :



- ✓ Un
- ✓Une
- ✓ et

➢Chaqu … ____ neurones

de la couche précédente (si existe):



➢$a_i^{(j)}$ = «

➢w(j)=n … mappage

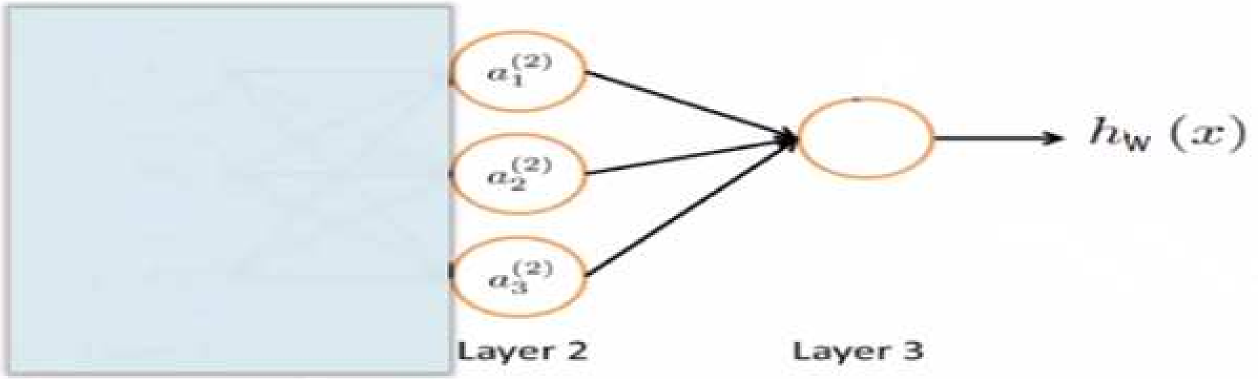
de la cou

➢W_{fg}: p … neurone

'g'

➢Chaq … é biais=1

Remarque: Le RN apprend ses propres features.



✓Cette vue de propagation vers l'avant également nous aide à comprendre ce que les réseaux neuronaux pourraient faire et pourquoi ils pourraient nous aider à apprendre des hypothèses non-linéaires intéressantes.

✓Considérez le réseau de neurones suivant et disons que je couvre le chemin de gauche de cette image pour l'instant. Si vous regardez ce qui reste dans cette image:

✓Notez que dans cette dernière étape, entre la couche j et la couche j+1, nous faisons exactement la même chose que dans la régression logistique.

✓L'ajout de toutes ces couches intermédiaires dans les réseaux de neurones nous permet de produire avec plus d'élégance des hypothèses non linéaires intéressantes et plus complexes.

<u>Apprentissage d'un RN:</u>

➢Pour apprendre un réseau neurone il faut <u>**deux passes** </u>une passe Forward et une passe Backward;

I. Dans la ***passe Forward*** on calcule les « a » l'activation de chaque unité de chaque couche;
II. Dans la ***passe Backward*** on calcule les erreurs de chaque unité entre les couches;

➢Le modèle RN doit minimiser l'erreur entre la classe prédite et la classe réelle pour tous les exemples d'apprentissage➔<u>on doit calculer cette fonction du cout à minimiser</u>.

➢Principe de Descente de Gradient

Apprentissage d'un RN:

I- Forward Propagation: Propagation vers l'avant :
Dans cette passe on doit calculer les a(j) «Activation» des neurones.
Implémentation Vectorisée:

$$a_1^{(2)}=g(w_{10}^{(1)}x_0 + w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2+ w_{13}^{(1)}x_3)=g(z_1^{(2)})$$
$$a_2^{(2)}=g(w_{20}^{(1)}x_0 + w_{21}^{(1)}x_1 + w_{22}^{(1)}x_2+ w_{23}^{(1)}x_3) =g(z_2^{(2)})$$
$$a_3^{(2)}=g(w_{30}^{(1)}x_0 + w_{31}^{(1)}x_1 + w_{32}^{(1)}x_2+ w_{33}^{(1)}x_3) =g(z_3^{(2)})$$

$$a_1^{(3)} = \underline{\mathbf{h_w(x)}}= g(w_{10}^{(2)}a_0^{(2)} + w_{11}^{(2)}a_1^{(2)} + w_{12}^{(2)}a_2^{(2)}+ w_{13}^{(2)}a_3^{(2)})$$

X0=1 EST LE BIA

$a_0^{(2)}$=1 =1 EST LE BIAIS

TOUS RESEAU NEURONE ENGENDRE UN BIAIS

$$w^{(1)} = \begin{bmatrix} w_{10}^{(1)} & w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{20}^{(1)} & w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \\ w_{30}^{(1)} & w_{31}^{(1)} & w_{32}^{(1)} & w_{33}^{(1)} \end{bmatrix} \quad w^{(2)} = \begin{bmatrix} w_{10}^{(2)} & w_{11}^{(2)} & w_{12}^{(1)} & w_{13}^{(1)} \end{bmatrix}$$

$$\begin{bmatrix} a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix} = g\left(\begin{bmatrix} w_{10}^{(1)} & w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ w_{20}^{(1)} & w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \\ w_{30}^{(1)} & w_{31}^{(1)} & w_{32}^{(1)} & w_{33}^{(1)} \end{bmatrix} X \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \right) = g\left(\begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix} \right)$$

$$a^{(2)} = g\left(w^{(1)} X\, a^{(1)}\right) = g\left(z^{(2)}\right)$$

$$a^{(3)} = g\left(w^{(2)} X\, a^{(2)}\right) = g\left(z^{(3)}\right) = \mathrm{hw(x)}$$

✓X0: est l'unité ou neurone biais elle est égale =1

$$a^{(2)} = g\left(w^{(1)} X\, a^{(1)}\right) = g\left(z^{(2)}\right)$$

$$a^{(3)} = g\left(w^{(2)} X\, a^{(2)}\right) = g\left(z^{(3)}\right) = \mathrm{hw}(\mathrm{x})$$

$a0^{(2)}$=1 =1 EST LE BIAIS

✓ Pour calculer w(2)Xa(2), on ajou … iais :
$a0^{(2)}=1 \rightarrow a^{(2)} \in R^4$

Si nous implémentons ses équations on trouve la valeur h(x).

contoso

Fonction du cout a mini

Puisque la dernière couche est

si

L'Entropie croisée sert à mesurer la quantité d'erreur dans la prédiction par rapport à la vérité.

$(h_w(x))_k$ est la kième sortie prédite par l'unité k

Yk : est la classe réelle pour l'exemple i

$$J(w) = -\frac{1}{m}\left[\underbrace{\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log\left(\left(h_w\left(x^{(i)}\right)\right)_k\right) + \left(1-y_k^{(i)}\right)\log\left(1-\left(h_w\left(x^{(i)}\right)\right)_k\right)}_{\text{Partie Estimation}}\right]$$

$$\underbrace{+\frac{\lambda}{2m}\sum_{l}^{L-1}\sum_{i}^{s_l}\sum_{j}^{s_{l+1}}\left(w_{ji}^{l}\right)^2}_{\text{Partie Régularisation}}$$

La triple somme additionne simplement le carré de tous les ws dans tout le réseau.

Pour minimiser cette fonction de cout on utilise **<u>l'Algorithme de Backpropagation</u>** :

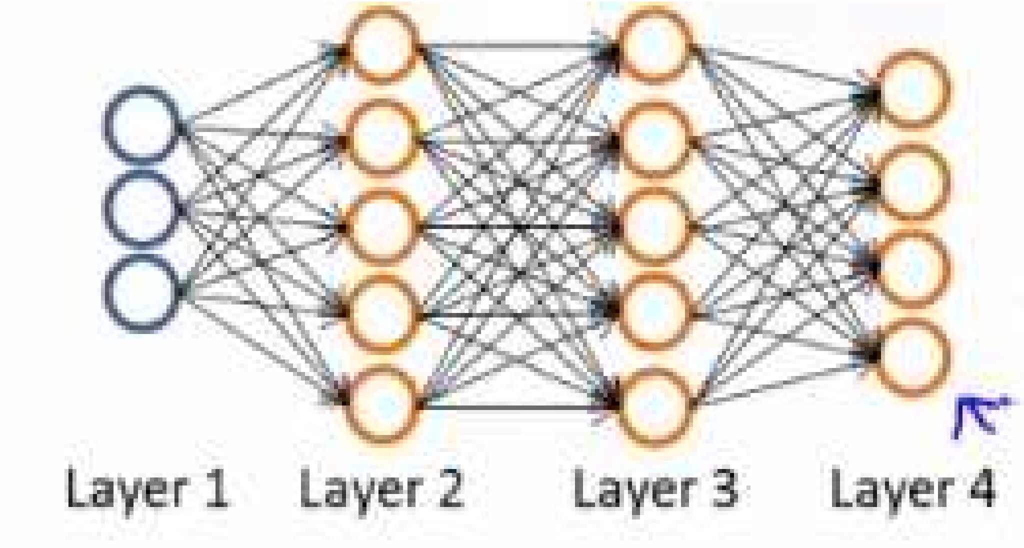
DICTIVE

II- Backpropagation :

Dans cette deuxième passe d’apprentissage du réseau de neurone, on utilise l’algorithme Backpropagation suivant. Cet algorithme est utilisé pour estimer la fonction du coût à minimiser.

Pour dérouler cet algorithme, nous allons utiliser un réseau avec plus de couches et qui produit plus de classes.

Backpropagation:



-{(x1,y1),(x2,y2),...(xm,ym)} un ensemble de points

-On note L le nombre de couche du RN L=4 ;

-S_l=nb Unité de la couche l : $S_1=3$;$S_2=5$;$S_3=5$;$S_4=4$

-K nombre de classes **ou** nb Unités de la couche sortie

Si Classification binaire➔ y=0,1

k=nombre unité (neurones)de couche de sortie=1

-Si multi class Classification (K Classes)➔ $S_l=K$

Algorithme Backpropagation :

Début :

$\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),\ldots$ … l'apprentissage

Initialiser : $\Delta_{ij}^{(l)} = 0$ … éros)

For i=1 to m / (P…

1- Définir $a^{(}$…

2- Appliquer … ,…L

3- Utilisant $y^{(i)}$ …

4- Calculer : δ^{l-1}, …

5- $\Delta_{ij}^{(l)} := \Delta_{ij}^{(l)} + a_j^{(l)}\delta_i^{l}$ …

$D_{ij}^{(l)} := \frac{1}{m}\Delta_{ij}^{(l)} + \lambda w_{ij}^{(l)}$ If j<>0;

$D_{ij}^{(l)} := \frac{1}{m}\Delta_{ij}^{(l)}$ If j=0; / $D_{ij}^{(l)} = \frac{\partial J(w)}{\partial w_{ij}^{l}}$

Fin.

Les v…

On multiplie ensuite par élément avec une fonction appelée g ', ou g ‘, qui est la dérivée de la f…

ou par … $\Delta^{(l)} := \Delta^{(l)} +$ …

D est utilisé comme un accumulateur pour additionner nos valeurs au fur et à mesure

$a^{(1)}=x$

$z^{(2)}=w_1a^{(1)}$

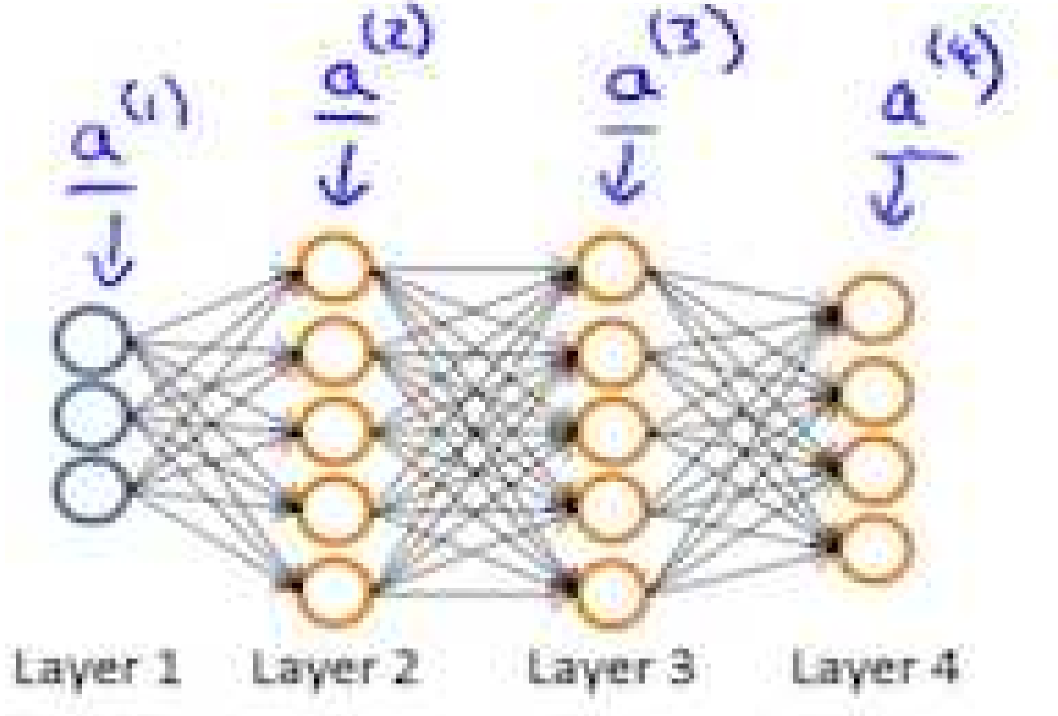
$a^{(2)}=g(z^{(2)})$ ajouter $a_0^{(2)}$ biais

$z^{(3)}=w_2a^{(2)}$

$a^{(3)}=g(z^{(3)})$ ajouter $a_0^{(3)}$ biais

$z^{(4)}=w_3a^{(3)}$

$a^{(4)}=g(z^{(4)})$



Réseau Neurone: Multi_classe Classification :

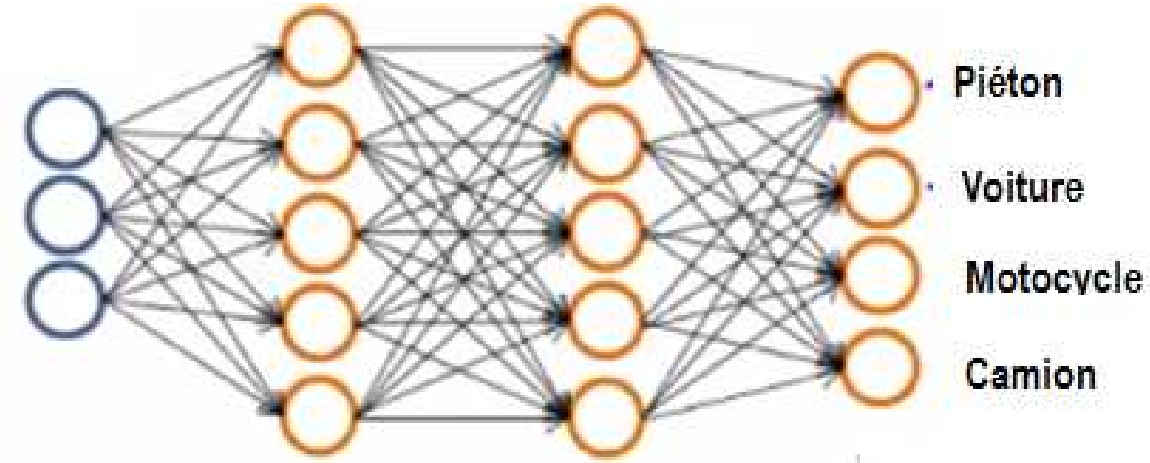
- Pour classer les données en plusieurs classes, nous laissons notre fonction d'hypothèse retourner un vecteur de valeurs.
- Disons que nous voulions classer nos données dans l'une des quatre catégories, Nous utiliserons l'exemple suivant pour voir comment cette classification est faite.
- Cet algorithme prend en entrée une image et la classe en conséquence:

Piéton

voiture

motocycle

camion



$$h_w(x) \in R^4$$

$$\text{On veut : } h_w(x) \approx \begin{bmatrix}1\\0\\0\\0\end{bmatrix};\quad h_w(x) \approx \begin{bmatrix}0\\1\\0\\0\end{bmatrix};\quad h_w(x) \approx \begin{bmatrix}0\\0\\1\\0\end{bmatrix};\quad \text{etc}$$

Quand Piéton　　Qd Voiture　　Qd motorcycle

Base d'apprentissage : $(x^{(1)},y^{(1)})$; $(x^{(2)},y^{(2)})$; ….. $(x^{(m)},y^{(m)})$

$y^{(i)}$ est un de : $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

- Chaque y (i) représente une image différente correspondante à une voiture, un piéton, un camion ou une motocycle.
- Les couches internes nous fournissent chacune de nouvelles informations qui conduisent à notre fonction d'hypothèse finale:

$$\begin{bmatrix}x_0\\x_1\\x_2\\..\\x_n\end{bmatrix} = \begin{bmatrix}a_0^{(1)}\\a_1^{(1)}\\a_2^{(1)}\\\dots\end{bmatrix} \rightarrow \begin{bmatrix}a_0^{(2)}\\a_1^{(2)}\\a_2^{(2)}\\\dots\end{bmatrix} \rightarrow \begin{bmatrix}a_0^{(3)}\\a_1^{(3)}\\a_2^{(3)}\\\dots\end{bmatrix} \rightarrow \cdots \rightarrow \begin{bmatrix}h_w(x)_1\\h_w(x)_2\\h_w(x)_3\\h_w(x)_4\end{bmatrix}$$

<u>Résumé</u> :

- Nous avons montré l'intérêt et la nécessité d'utiliser un autre classifieur que la régression logistique dans le cas ou nous avons des limites non linéaires et n est grand

-Nous avons définis le lien entre neurone biologique et neurone artificiel

-Un réseau de neurones est une connexion entre plusieurs neurones

-Pour apprendre un réseau neurone il faut deux passes une passe Forward et une autre passe backward

Résumé :

- Dans la passe forward on calcule les « a » l'activation de chaque unité » de chaque couche

-Dans la passe backword on calcule les erreurs de chaque unité entre les couches

-Nous avons définis la multi classification pour un RN.