

Modèles mathématiques et
computationnels en neurosciences

41702 CM II

Réseaux de neurones feed-forward et
l'apprentissage supervisé

Classification

non-supervisée

Deux classes





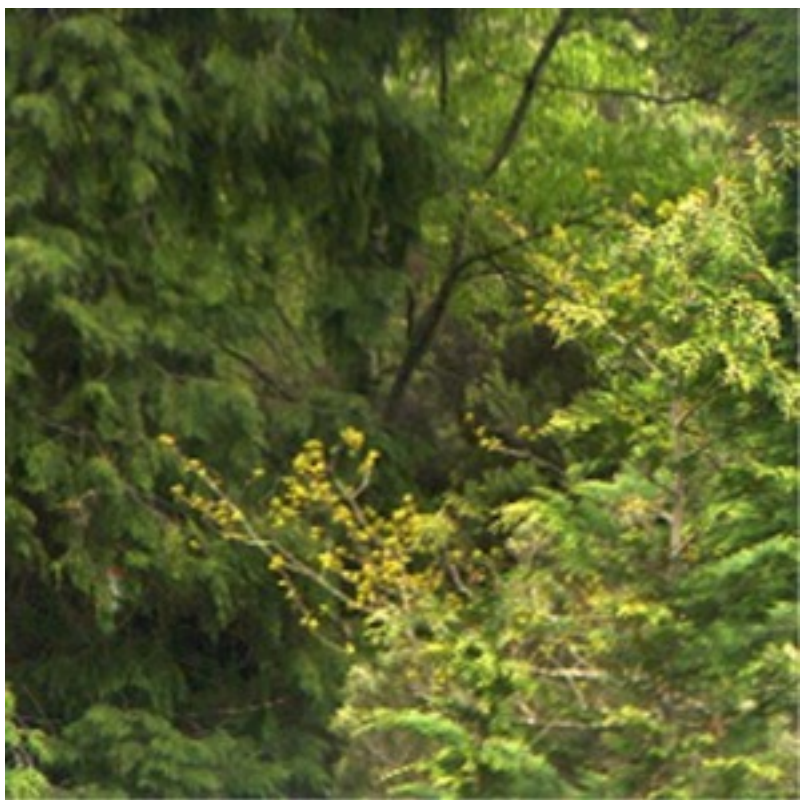












Quelles sont les
deux classes ?

Classification
supervisée

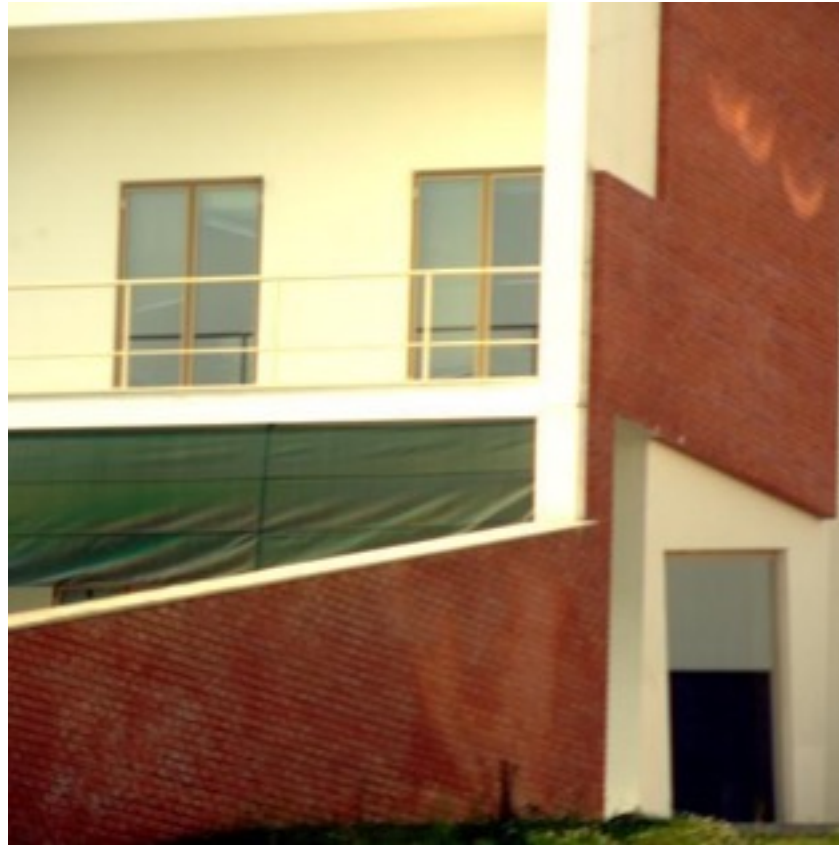
Deux classes



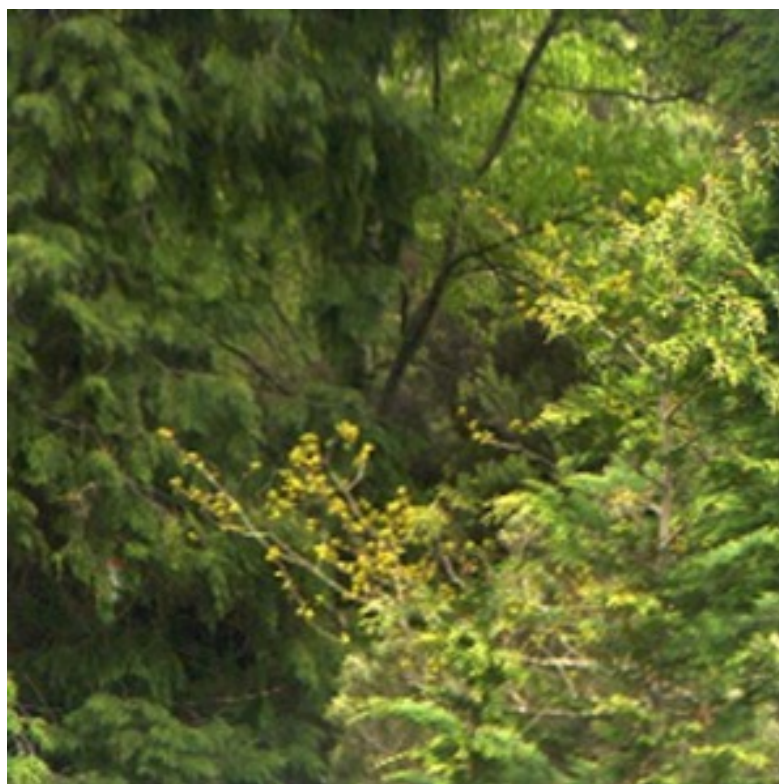
0



1



1



0



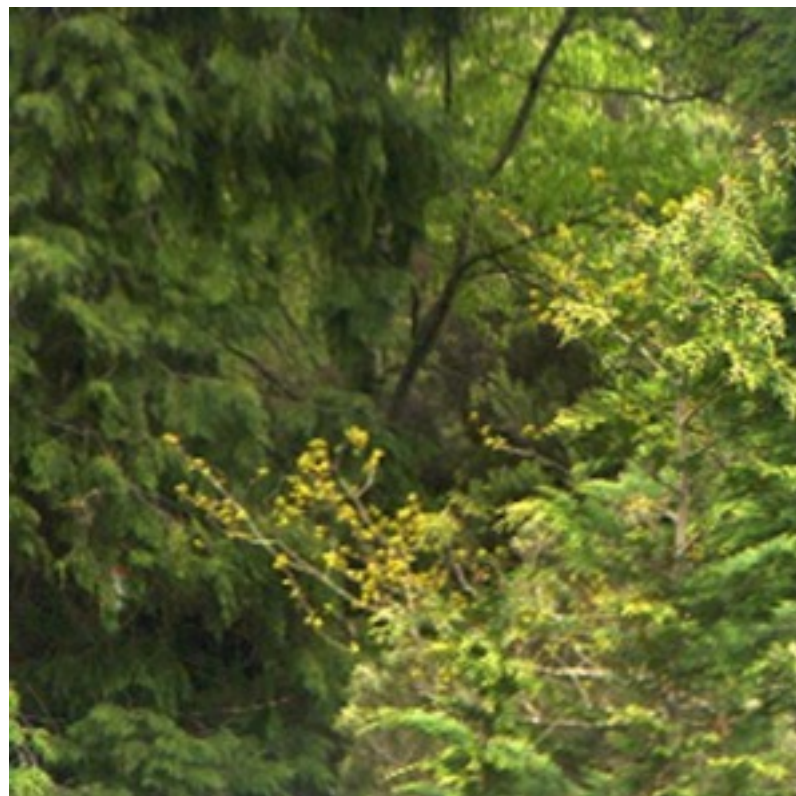
1



0

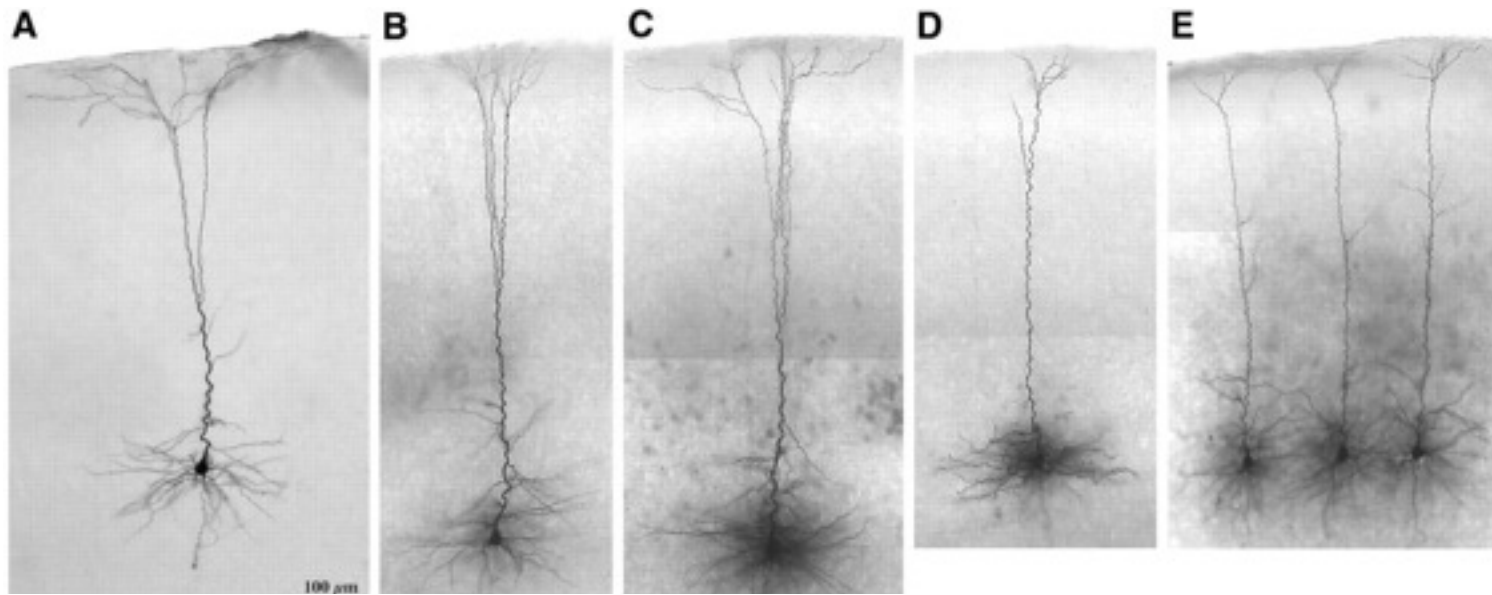


0

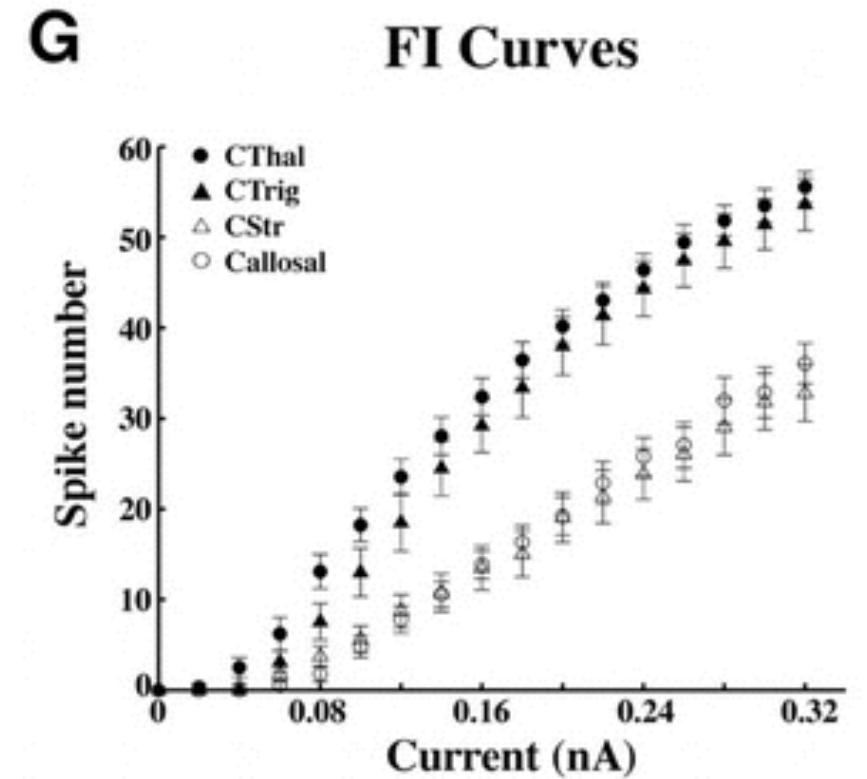


?

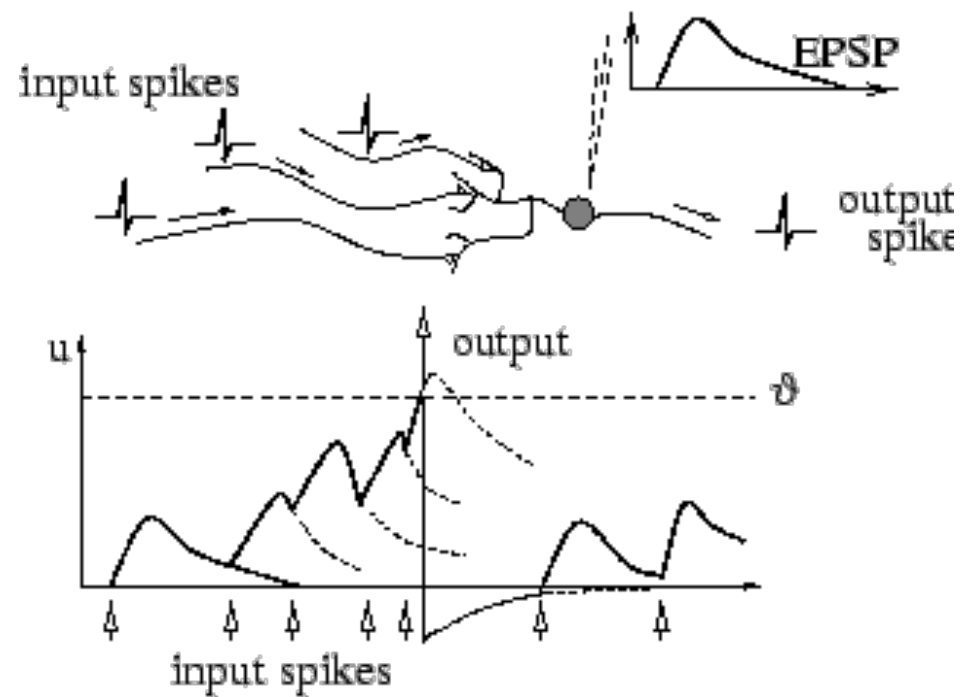
Fonction d'activation dans les neurones biologiques



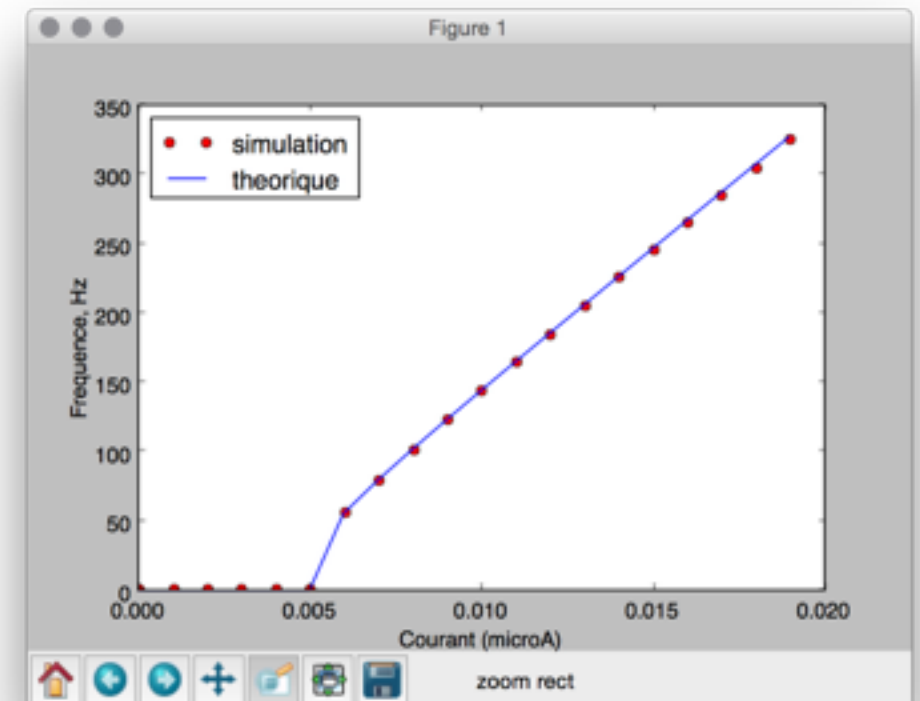
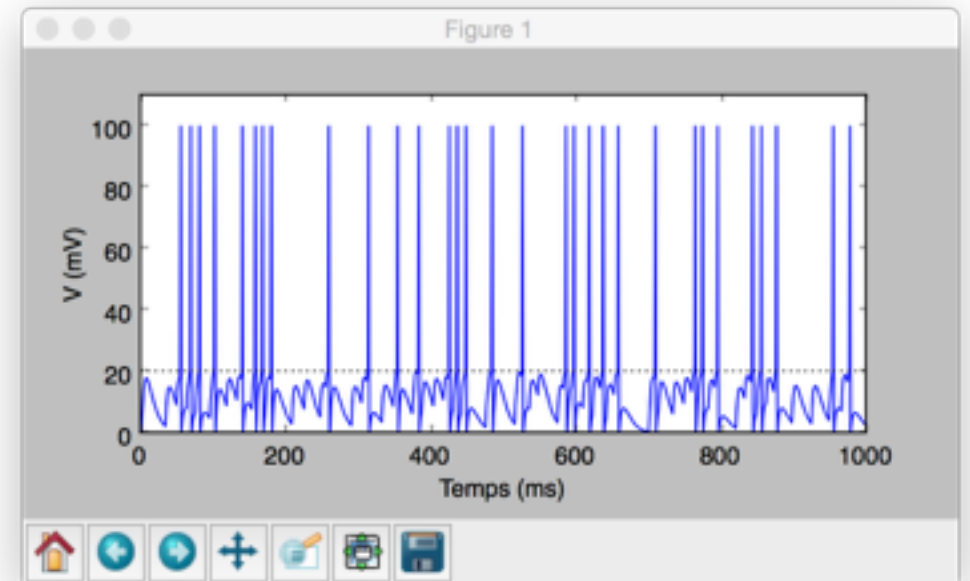
Layer V Neurons in Mouse Cortex Projecting to Different Targets Have Distinct Physiological Properties. *Journal of Neurophysiology*, 2007, 98:3330-3340



Fonction d'activation dans le modèle intègre et tire

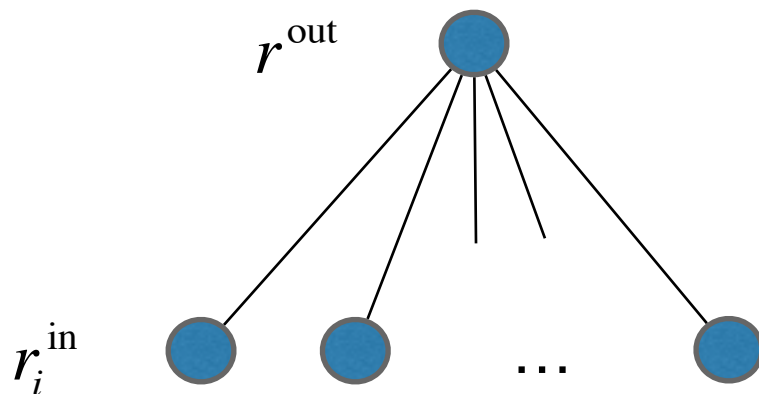


$$\tau_m \frac{du}{dt} = -u + R_m I$$



Fonction d'activation dans le modèle à taux de décharge

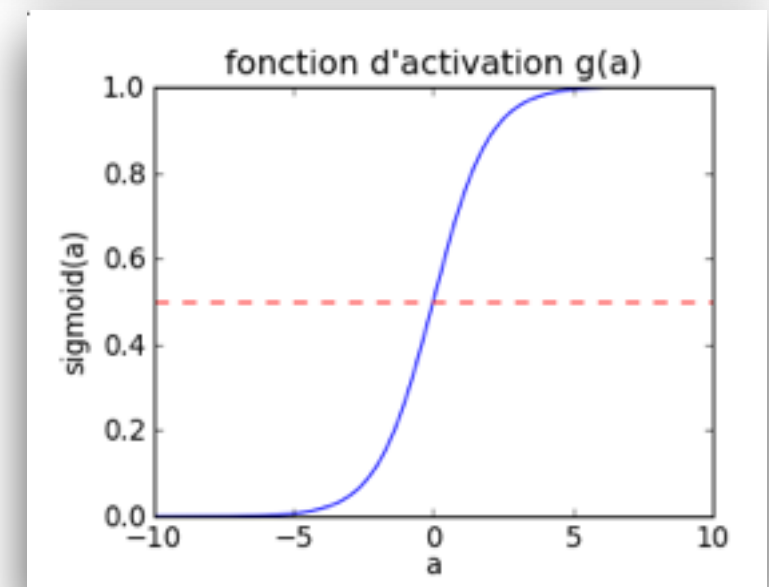
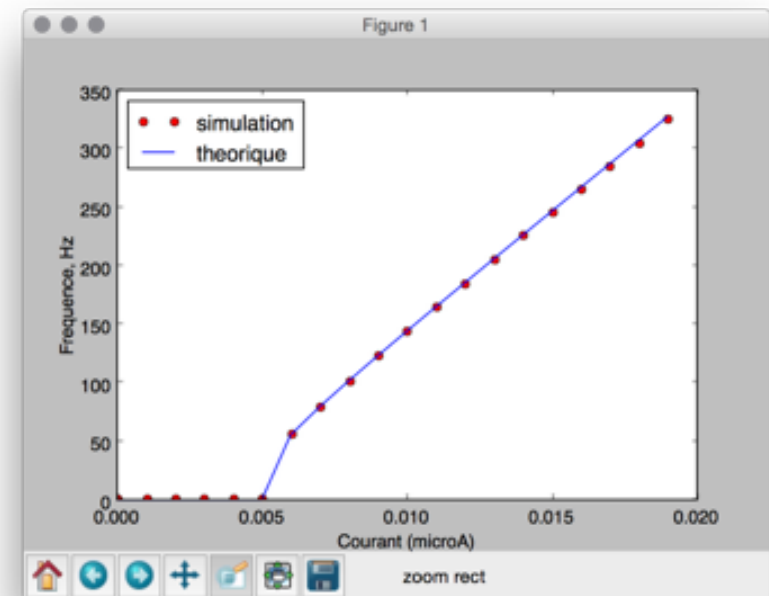
On suppose que le taux de décharge contient toute l'information transmise par neurones



r_i^{in} le taux de décharge du i-ème neurone présynaptique (en Hz ou en spikes/s)

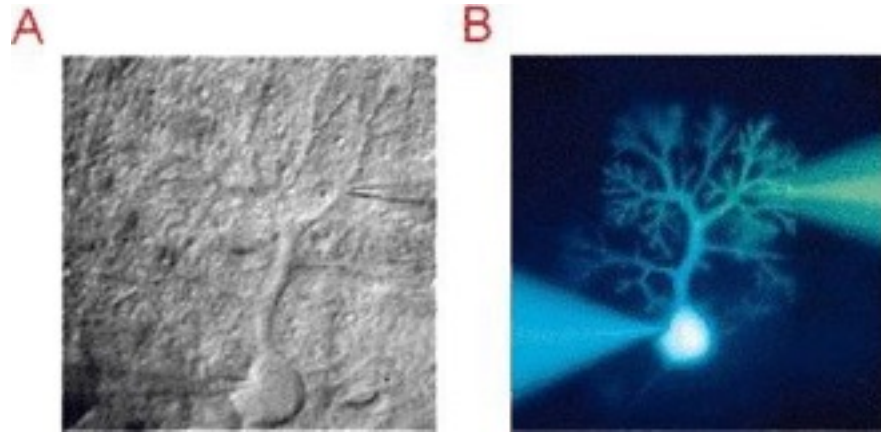
$a = \sum_i r_i^{in}$ l'**activation** du neurone postsynaptique

$r^{out} = g(a)$ le taux de décharge du neurone postsynaptique avec la fonction d'activation $g()$



$g(a) = \frac{1}{1 + e^{-a}}$ Fonction "sigmoïde"

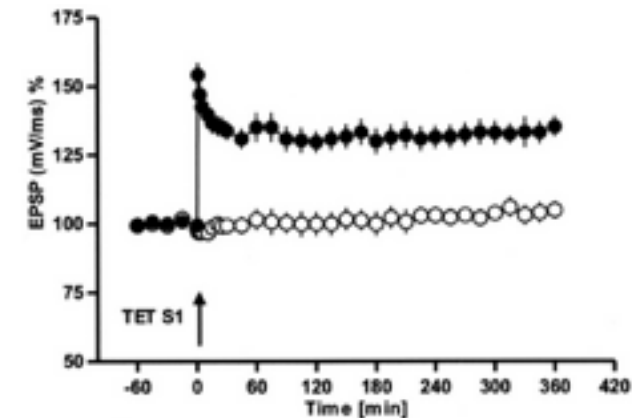
Plasticité synaptique dans les neurones biologiques



1. On observe expérimentalement la réponse d'un neurone à une impulsion électrique et on note l'amplitude de la réponse (condition de base)
2. On stimule le neurone avec un train de PA d'une haute (Fig.1) ou basse (Fig. 2) fréquence (condition test)
3. On observe que, après la stimulation, l'amplitude de réponse au même impulsion de base a augmenté (LTP) ou diminué (LTD).

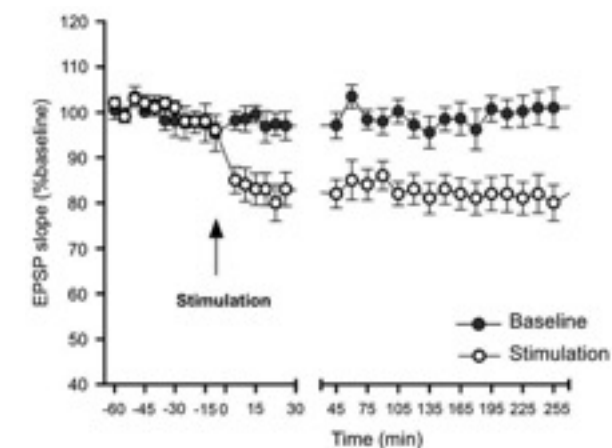
1. L'augmentation d'efficacité d'une synapse induite par une stimulation : **potentialisation à long terme (LTP)**

Stimulation d'une haute fréquence (100-300 Hz)



2. La diminution d'efficacité induite par une stimulation : **dépression à long terme (LTD)**

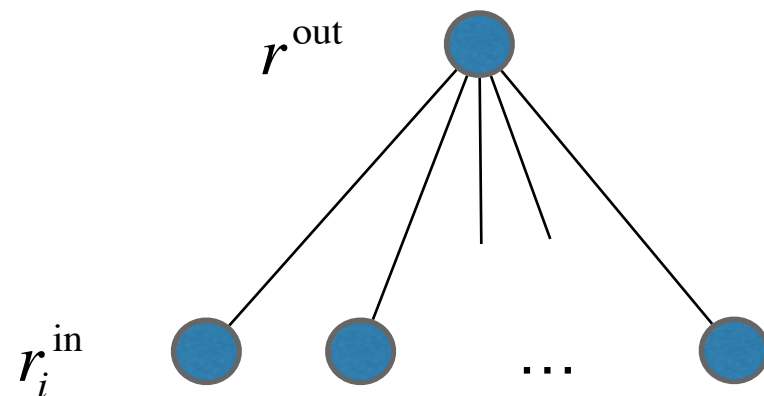
Stimulation d'une basse fréquence (1-3 Hz)



L'efficacité d'un synapse n'est pas constante.
Elle change en fonction de l'activité précédente

Fonction d'activation dans le modèle à taux de décharge **avec plasticité synaptique**

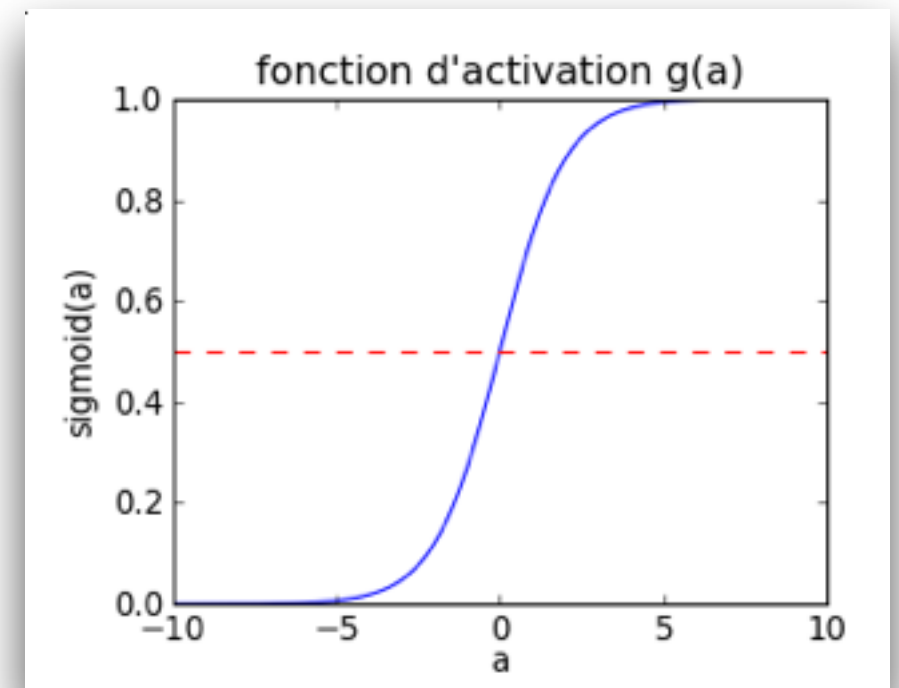
On suppose que le taux de décharge contient toute l'information transmise par neurones



r_i^{in} le taux de décharge du i-ème neurone présynaptique

$a = \sum_i r_i^{\text{in}}$ l'activation du neurone postsynaptique

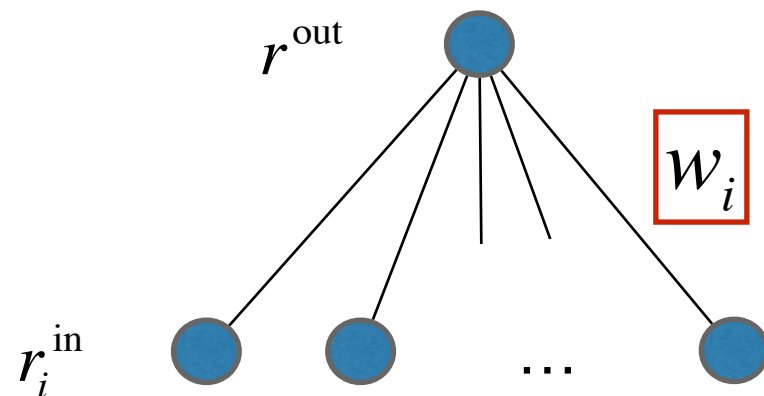
$r^{\text{out}} = g(a)$ le taux de décharge du neurone postsynaptique avec la fonction d'activation $g()$



$g(a) = \frac{1}{1 + e^{-a}}$ Fonction "sigmoïde"

Fonction d'activation dans le modèle à taux de décharge **avec plasticité synaptique**

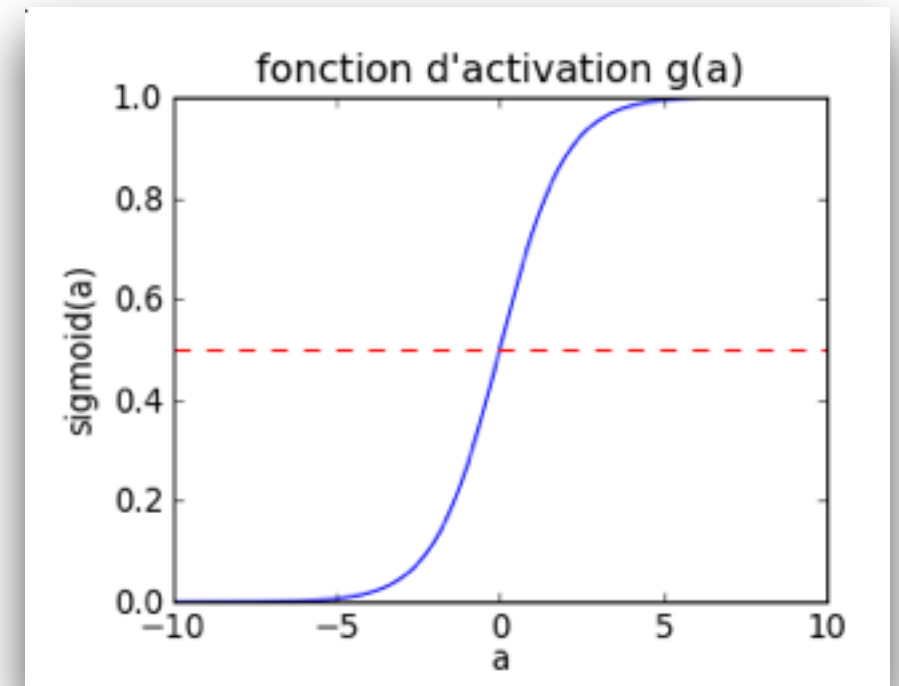
On suppose que le taux de décharge contient toute l'information transmise par neurones



r_i^{in} le taux de décharge du i-ème neurone présynaptique

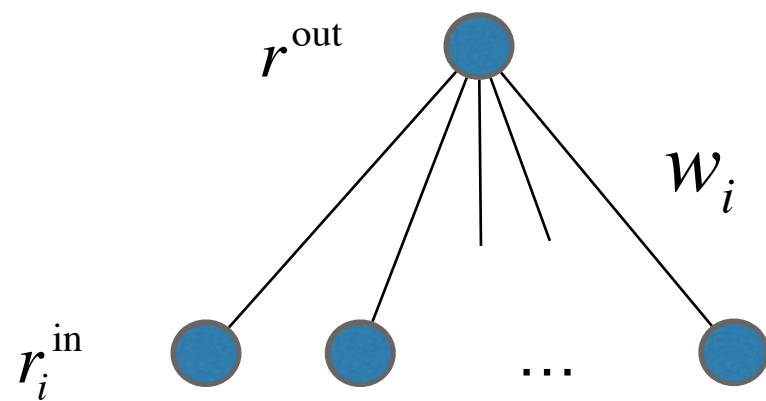
$a = \sum_i w_i r_i^{\text{in}}$ l'activation du neurone postsynaptique

$r^{\text{out}} = g(a)$ le taux de décharge du neurone postsynaptique avec la fonction d'activation $g()$



$g(a) = \frac{1}{1 + e^{-a}}$ Fonction "sigmoïde"

Questions principales



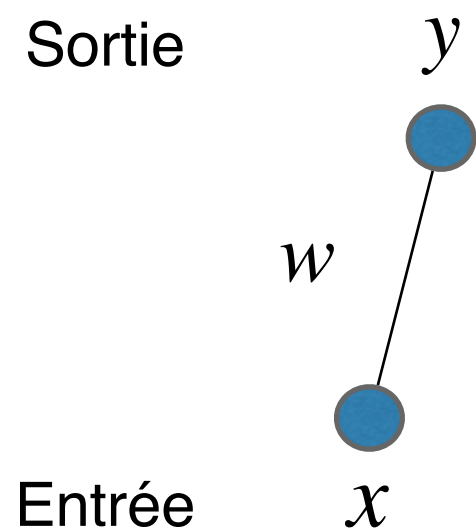
$$a = \sum_i w_i r_i^{\text{in}}$$

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

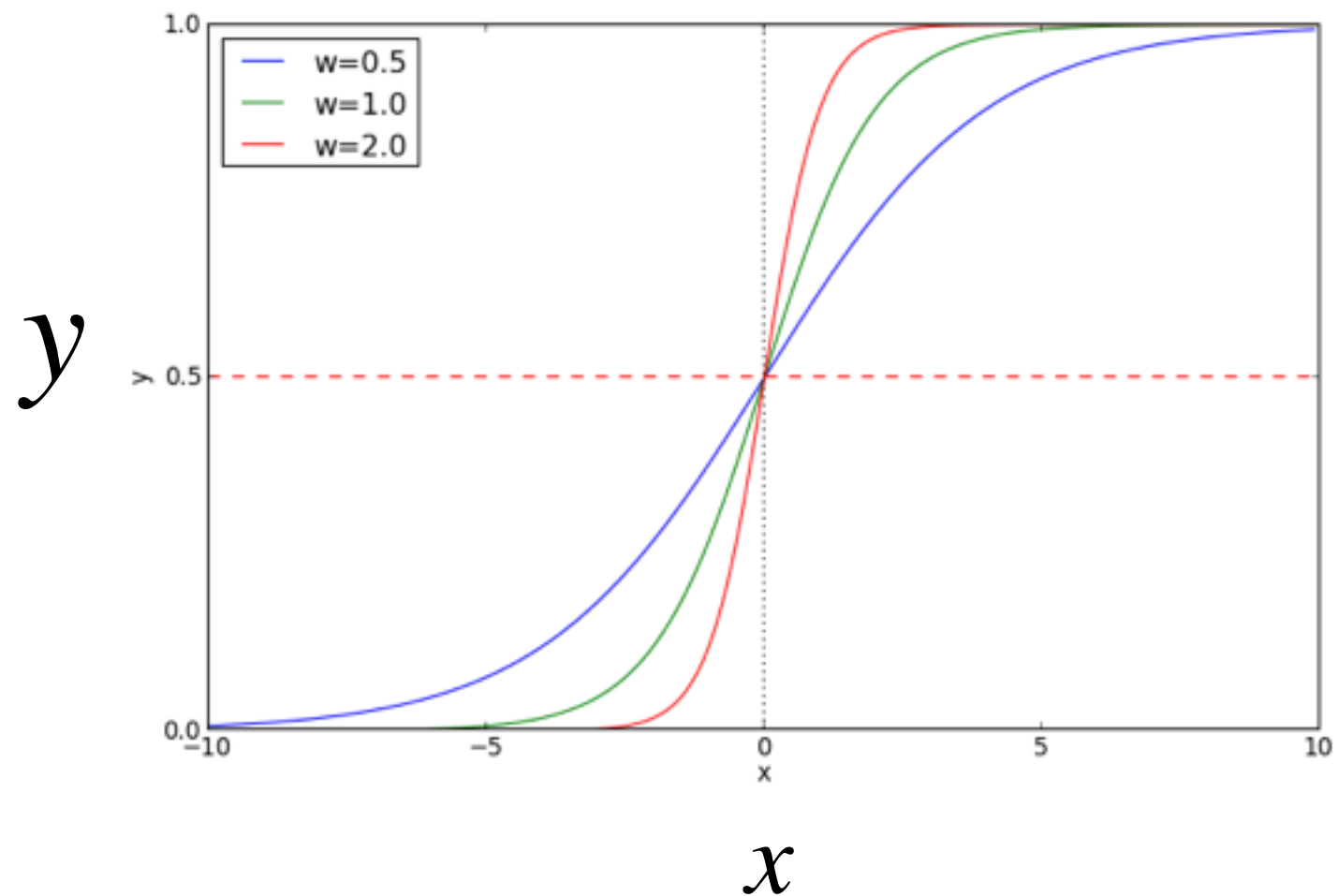
$$r^{\text{out}} = g(a)$$

1. Que peut faire un neurone avec la fonction d'activation sigmoïde et les poids malléables ?
2. Que peut faire un réseau de tels neurones ?

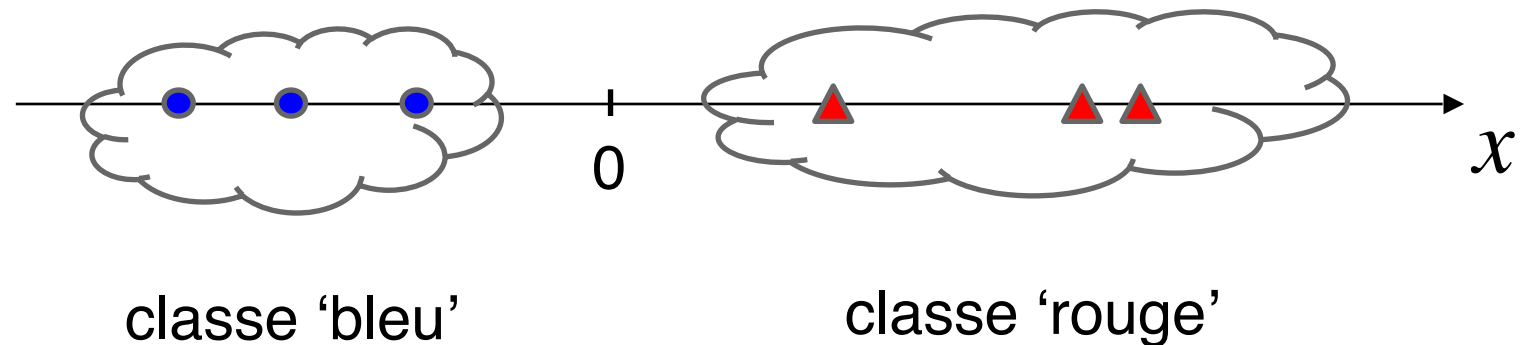
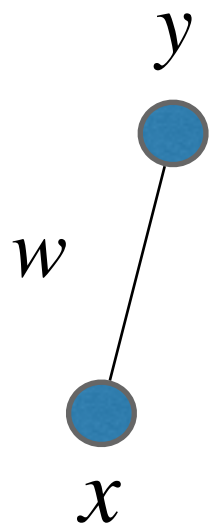
Un neurone avec une entrée



$$y = \frac{1}{1 + e^{-wx}}$$

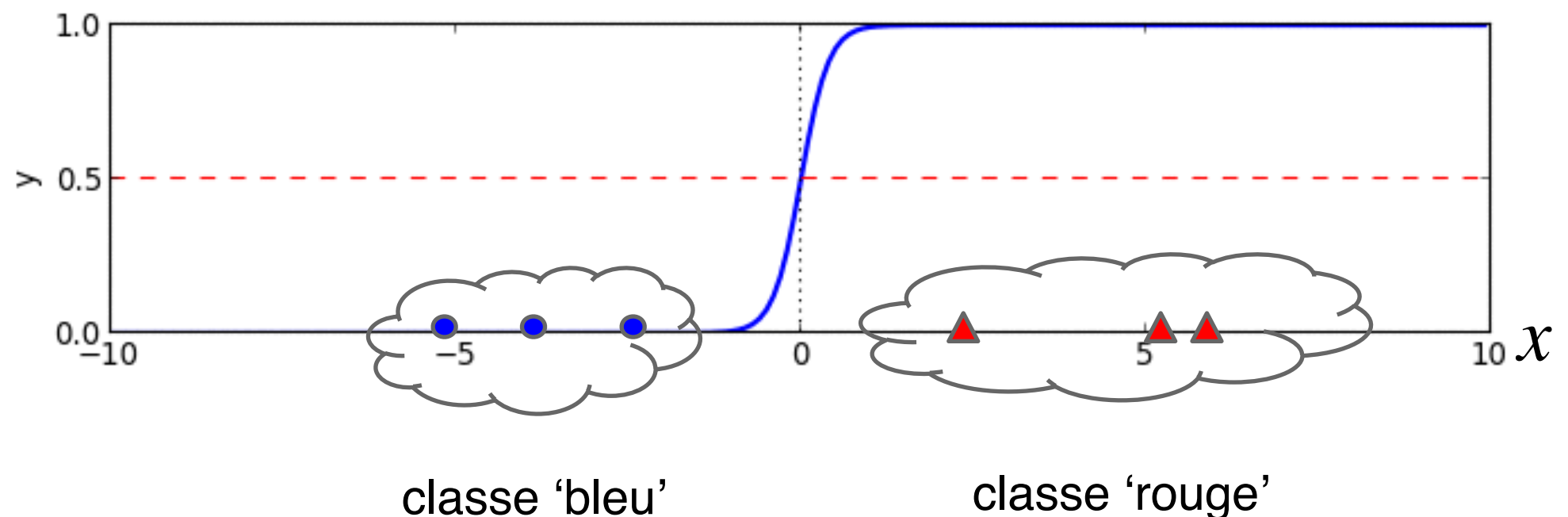
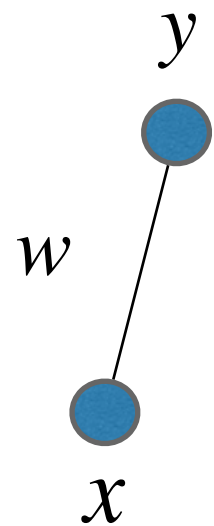


Un neurone avec une entrée



$$y = \frac{1}{1 + e^{-wx}}$$

Un neurone avec une entrée

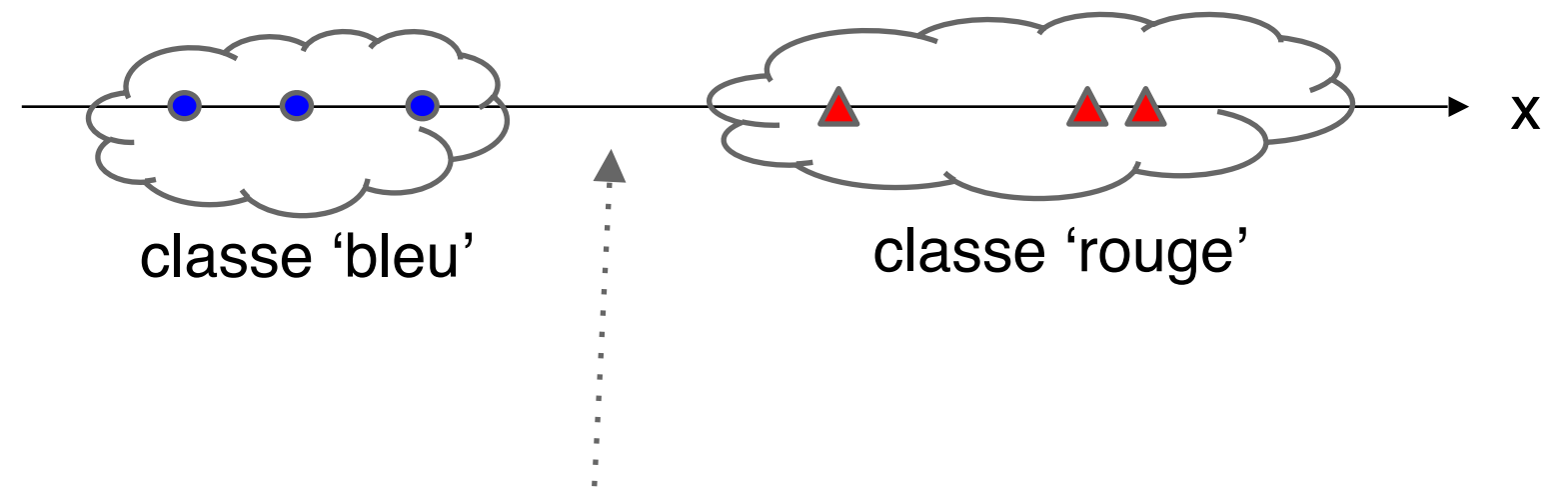


$$y = \frac{1}{1 + e^{-wx}}$$

Classification :

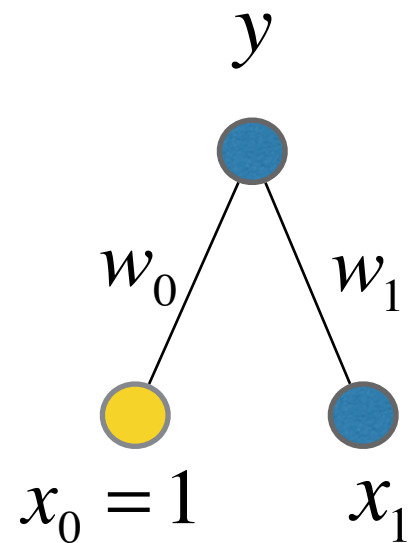
1. On présente au réseau une valeur d'entrée (par ex. $x = 5$)
2. On calcule la sortie y
3. Si $y = 1$, l'entrée appartient à la classe 'rouge'
4. Sinon, l'entrée appartient à la classe 'bleu'

Un neurone avec une entrée

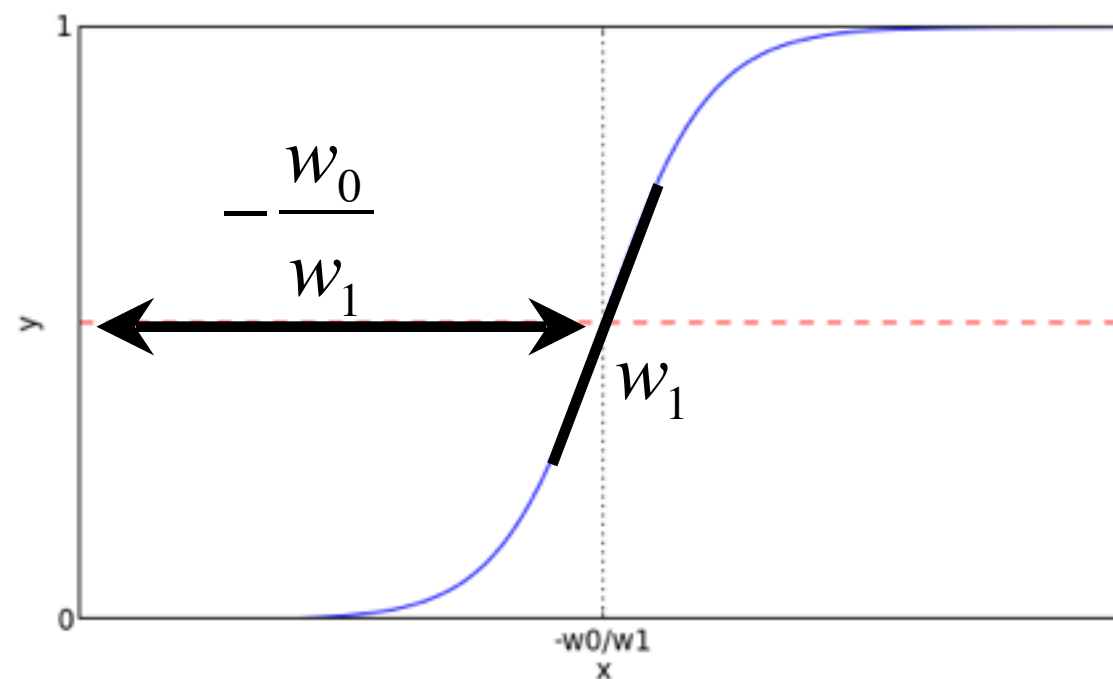
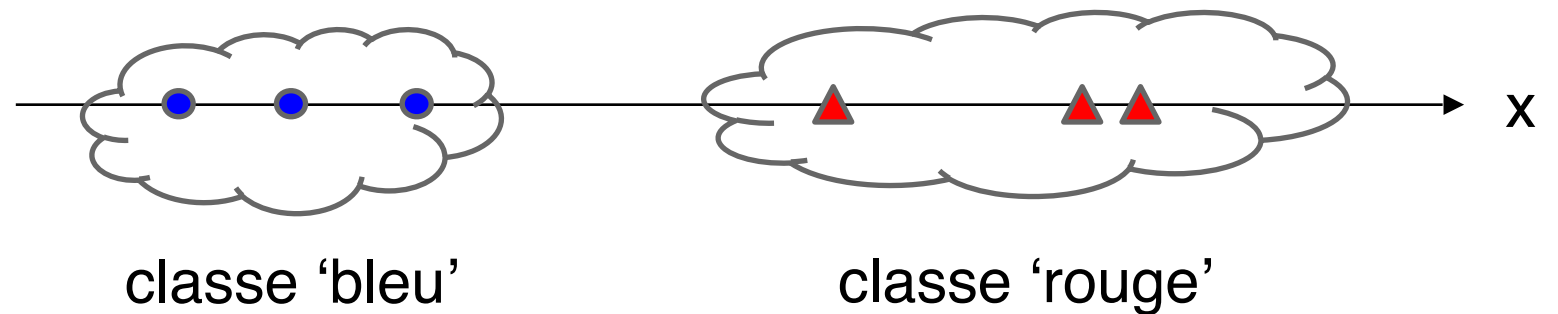


Il serait mieux si le réseau pourrait ajuster automatiquement le point de séparation entre les deux classes

Un neurone avec une entrée

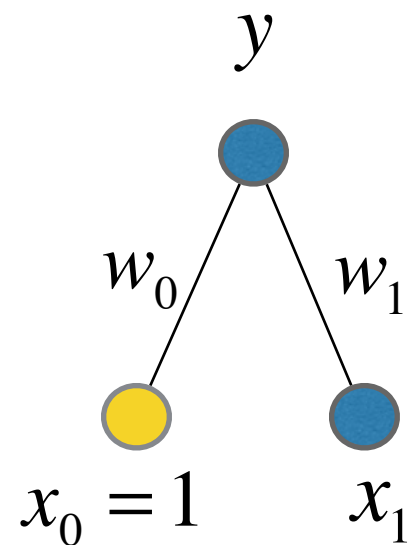


↑
'biais' où
'seuil'



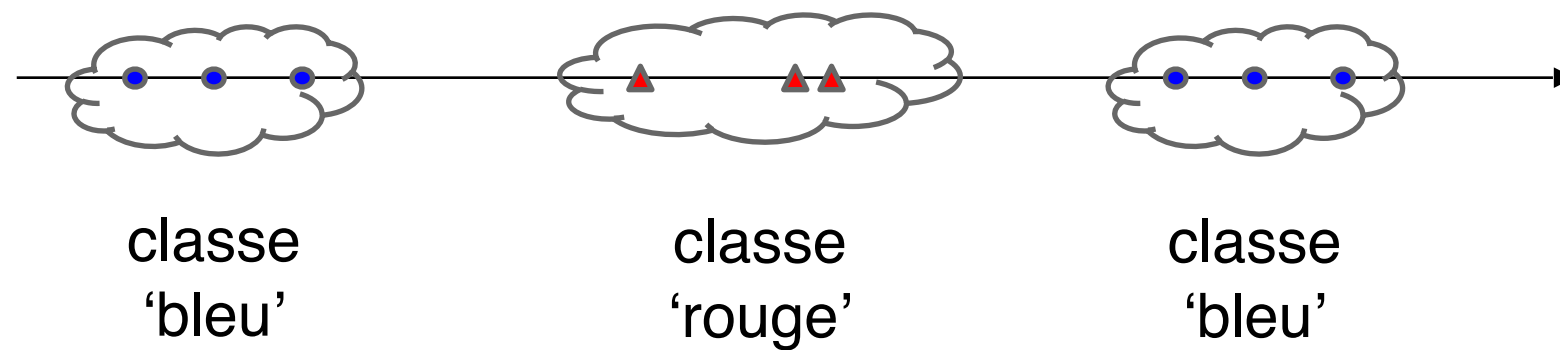
$$y = \frac{1}{1 + e^{-w_0 - w_1 x}}$$

Un neurone avec une entrée



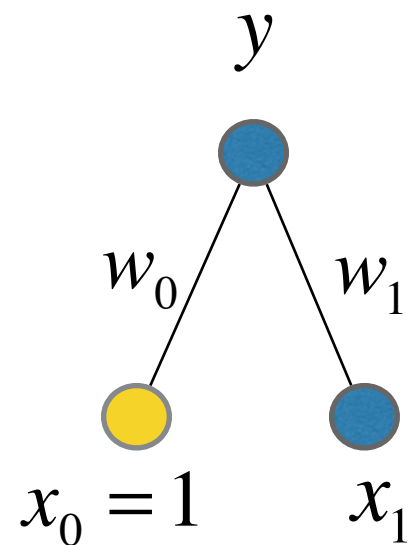
Un neurone avec une entrée (+ biais) peut séparer les données dans une dimension en deux classes

Un neurone avec une entrée



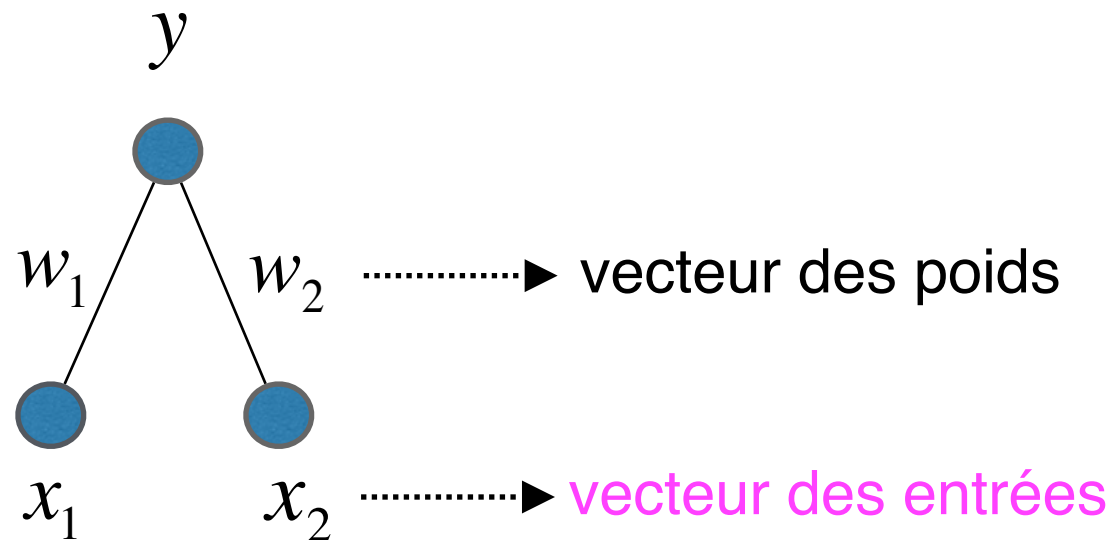
Impossible de classifier par un neurone !

Un neurone avec une entrée

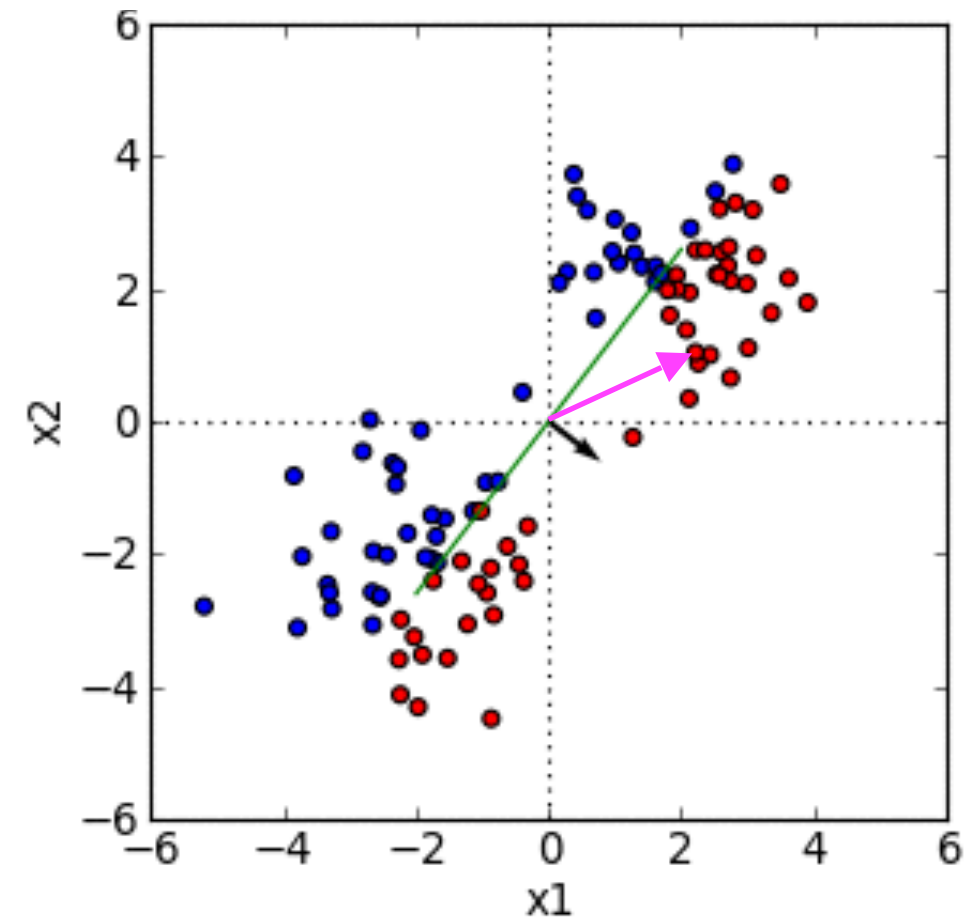
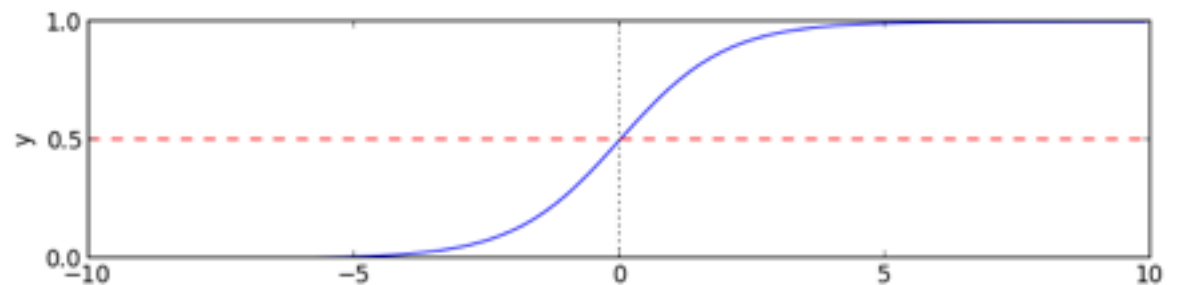


Un neurone avec une entrée (+ biais) peut séparer les données **linéairement séparables** dans une dimension en deux classes

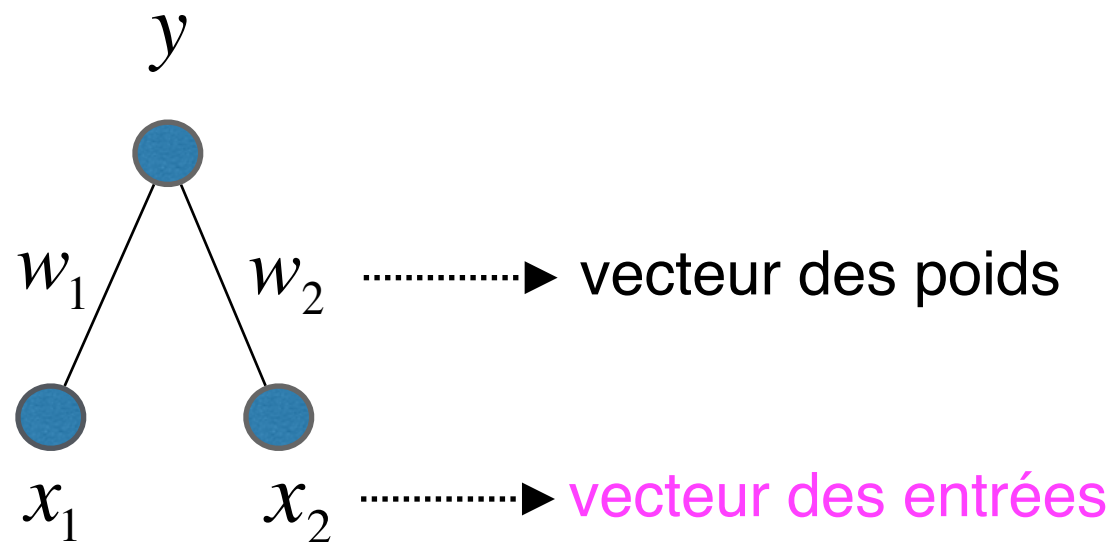
Un neurone / deux entrées - sans biais



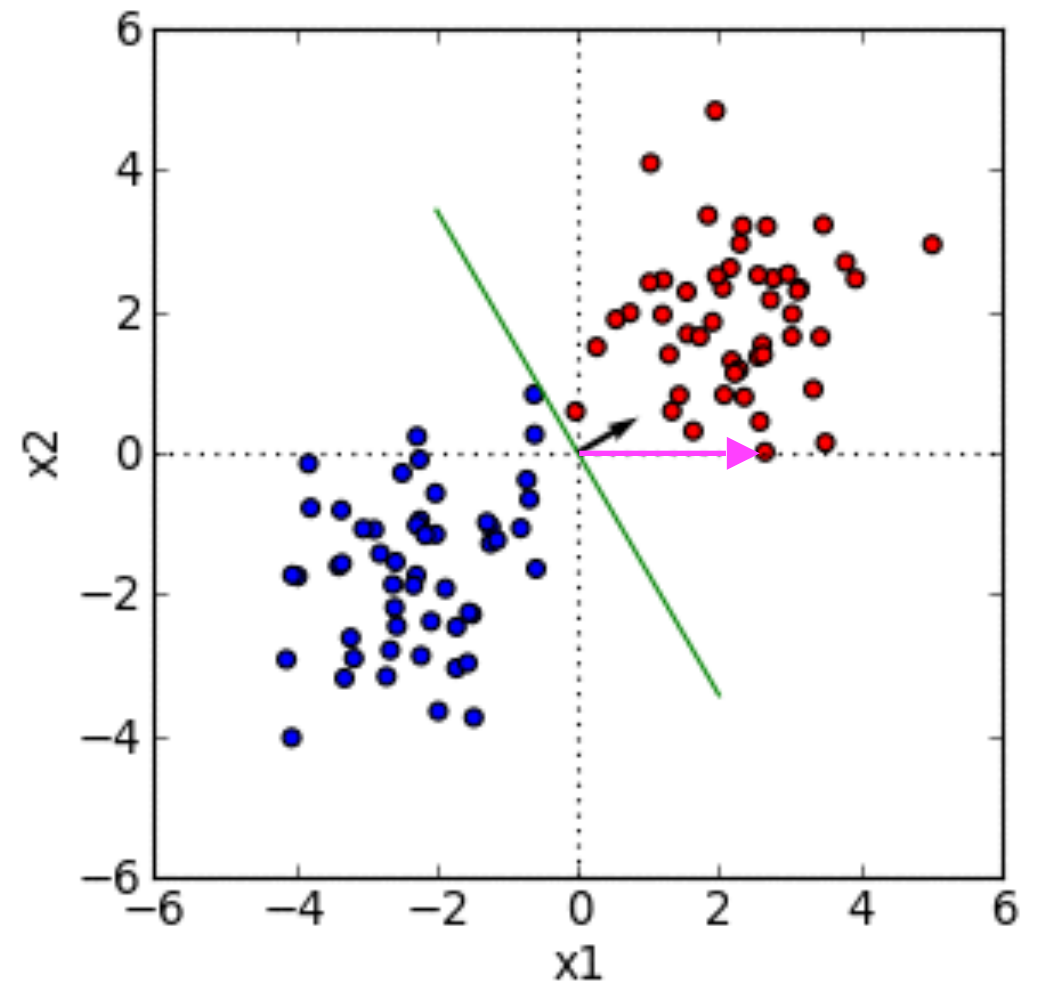
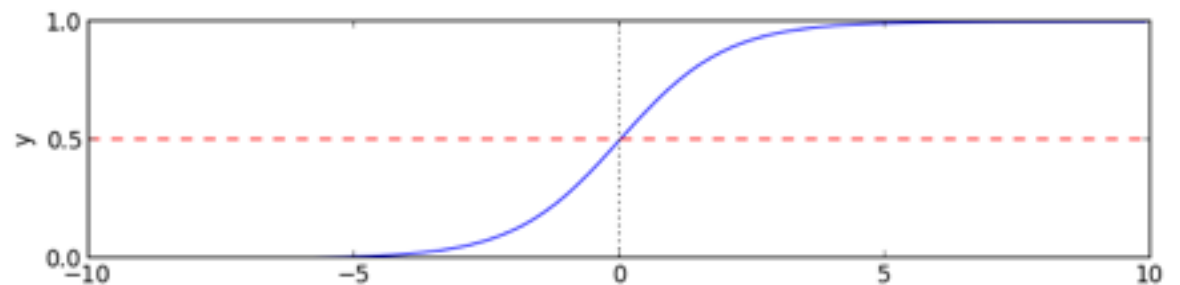
$$y = \frac{1}{1 + e^{-w_1 x_1 - w_2 x_2}} = \frac{1}{1 + e^{-\vec{w} \cdot \vec{x}}}$$



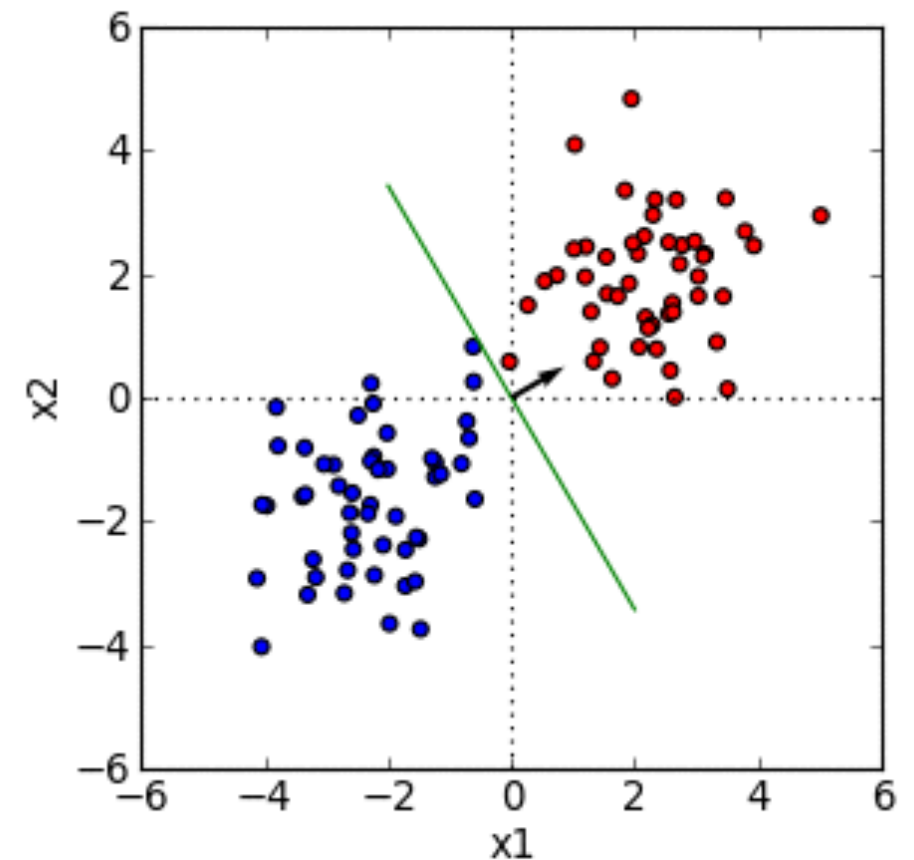
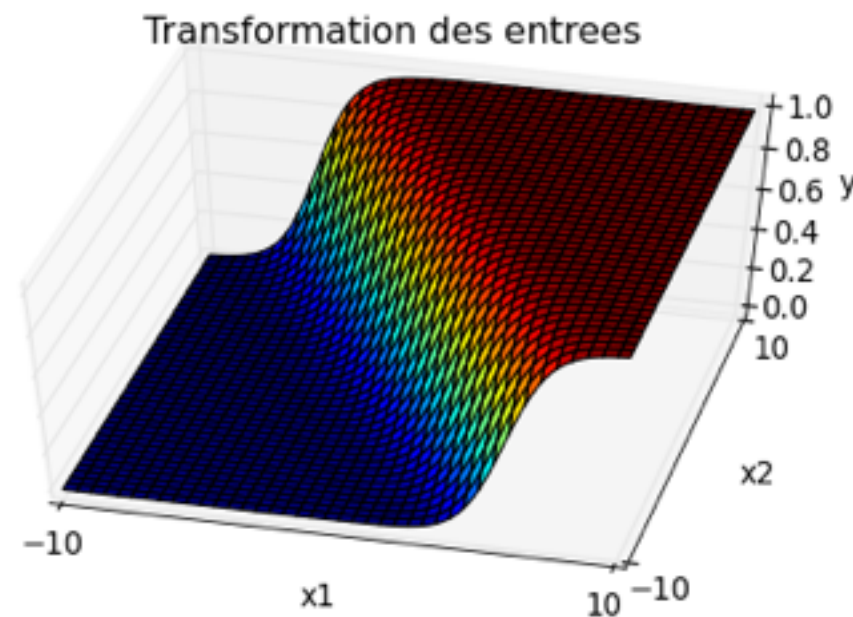
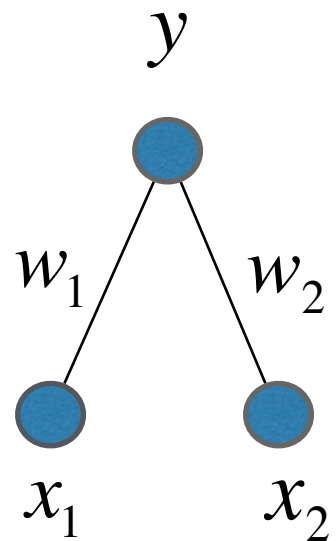
Un neurone / deux entrées - sans biais



$$y = \frac{1}{1 + e^{-w_1 x_1 - w_2 x_2}} = \frac{1}{1 + e^{-\vec{w} \cdot \vec{x}}}$$

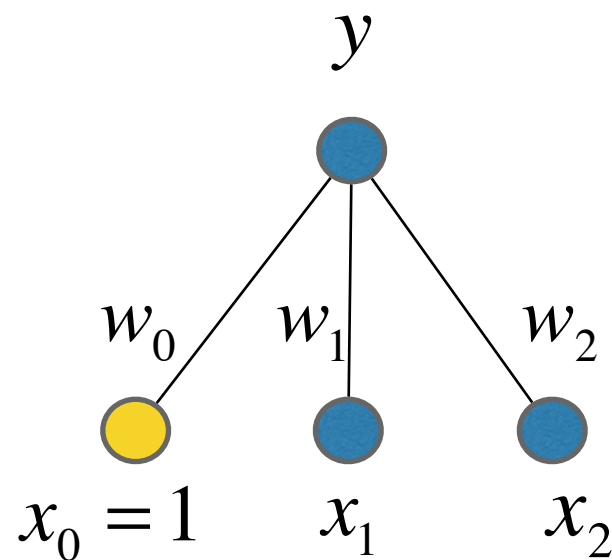


Un neurone / deux entrées - sans biais

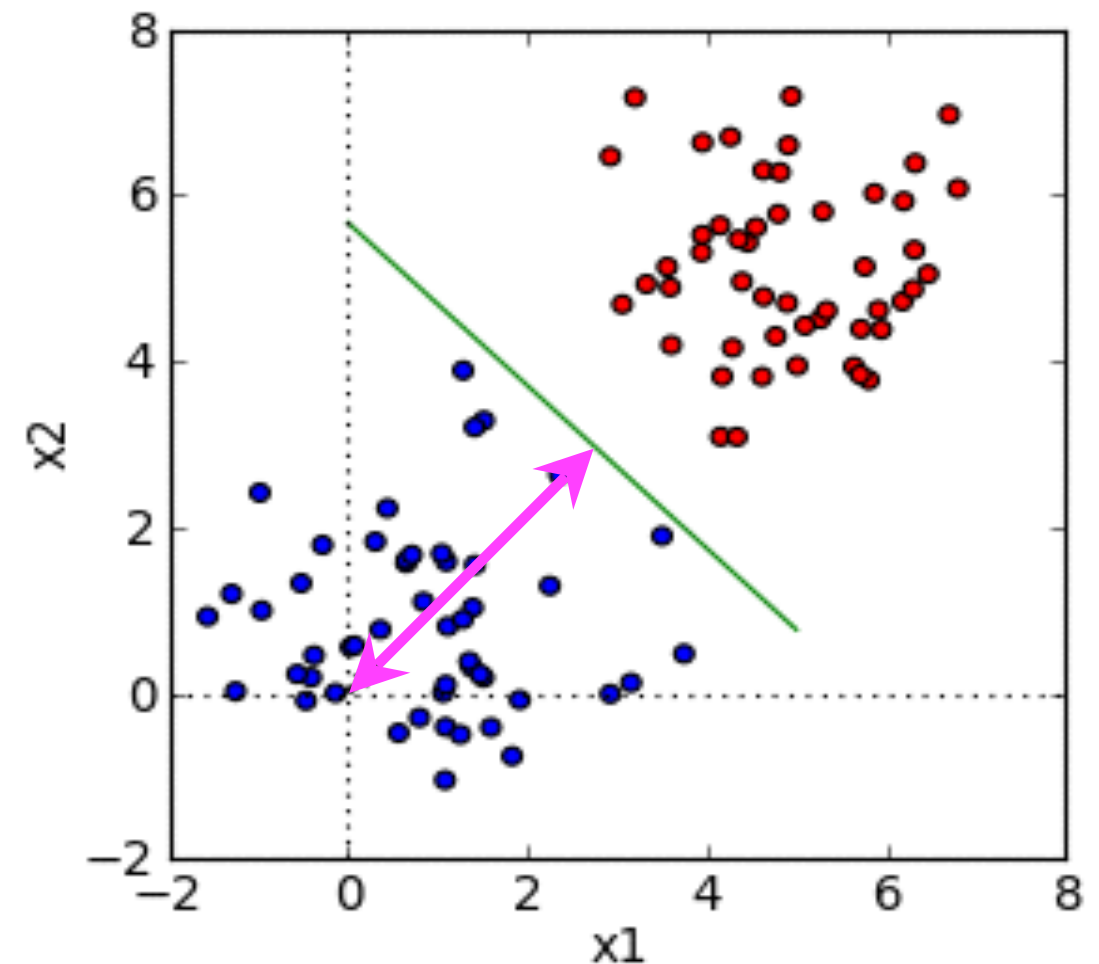


$$y = \frac{1}{1 + e^{-w_1 x_1 - w_2 x_2}} = \frac{1}{1 + e^{-\vec{w} \cdot \vec{x}}}$$

Un neurone / deux entrées - avec biais

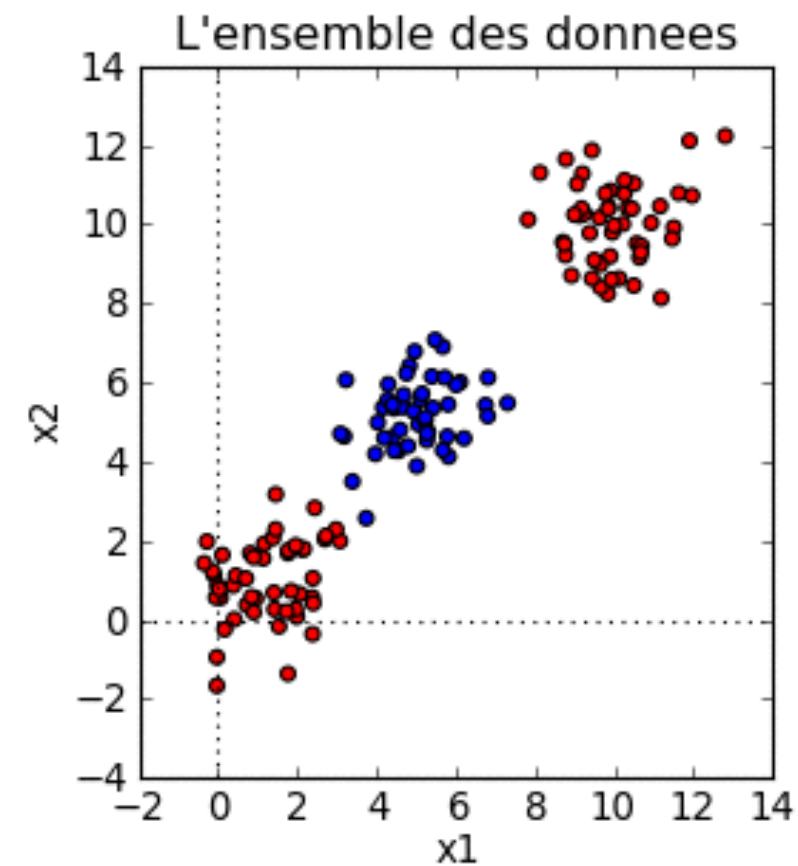
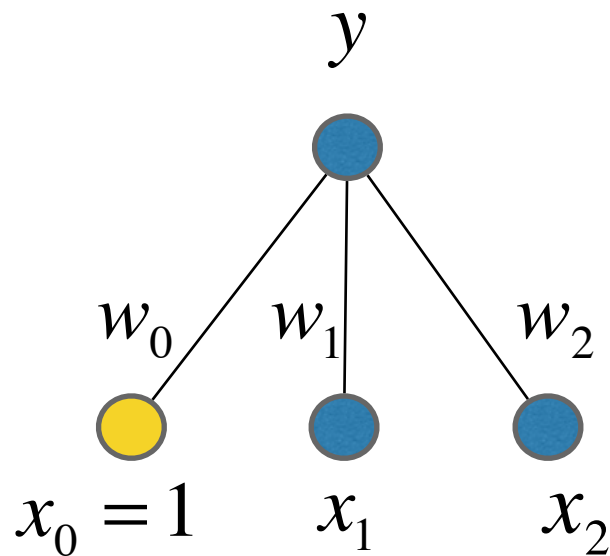


$$y = \frac{1}{1 + e^{-w_0 - w_1 x_1 - w_2 x_2}} = \frac{1}{1 + e^{-w_0 - \vec{w} \cdot \vec{x}}}$$



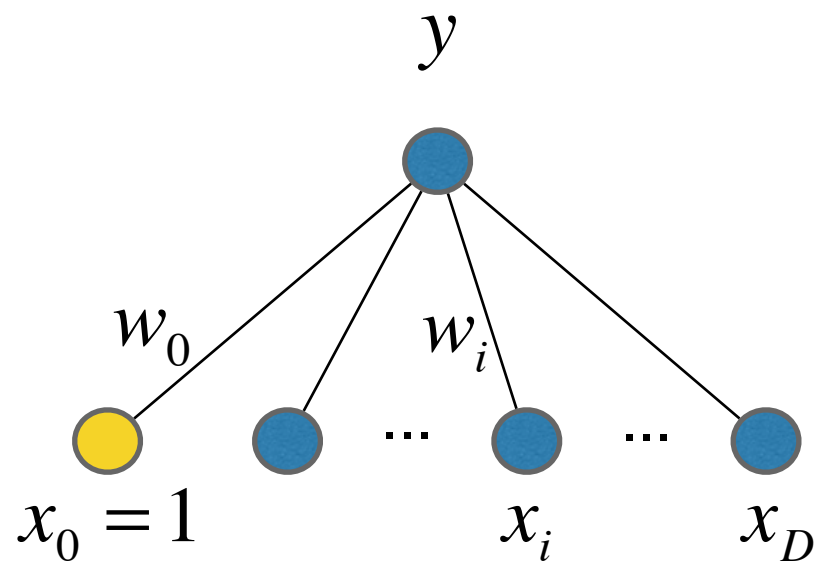
- distance de l'origine à la ligne de séparation : $-\frac{w_0}{\|\vec{w}\|}$
- orientation de la ligne de séparation: \vec{w}
- la pente de la fonction sigmoïde : $\|\vec{w}\|$

Un neurone / deux entrées - avec biais



Impossible de classifier par un neurone !

Un neurone / D entrées

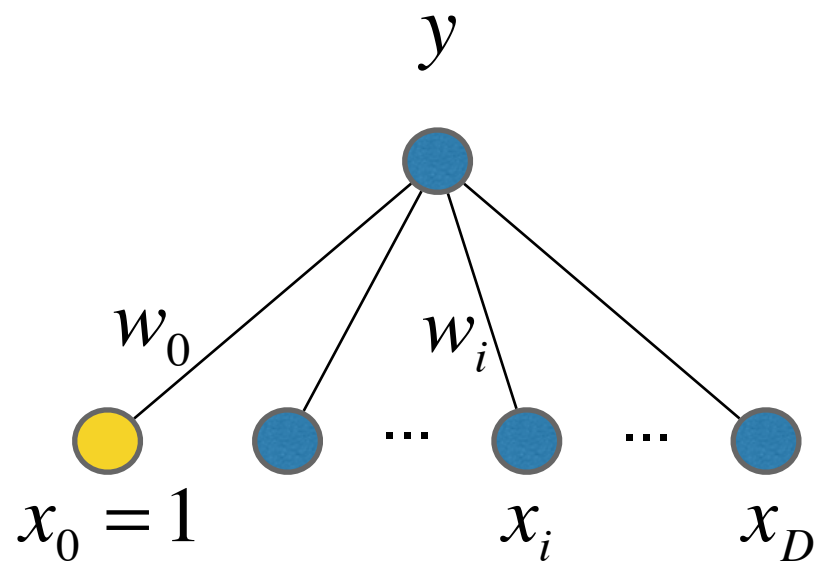


1. Que peut faire un neurone avec la fonction d'activation sigmoïde et les poids malléables ?

Réseau de neurones feed-forward
à une couche de poids :

“perceptron”

Un neurone / D entrées



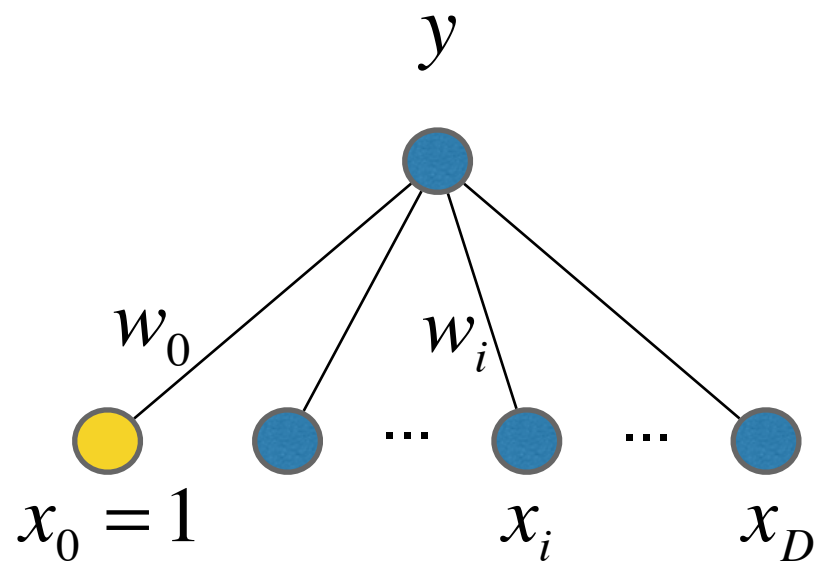
1. Que peut faire un neurone avec la fonction d'activation sigmoïde et les poids malléables ?

Réseau de neurones feed-forward
à une couche de poids :
“perceptron”

Un neurone avec D entrées (+ biais) peut séparer des données **linéairement séparables** dans D dimensions en deux classes

La surface de séparation est un **hyperplan**

Un neurone / D entrées



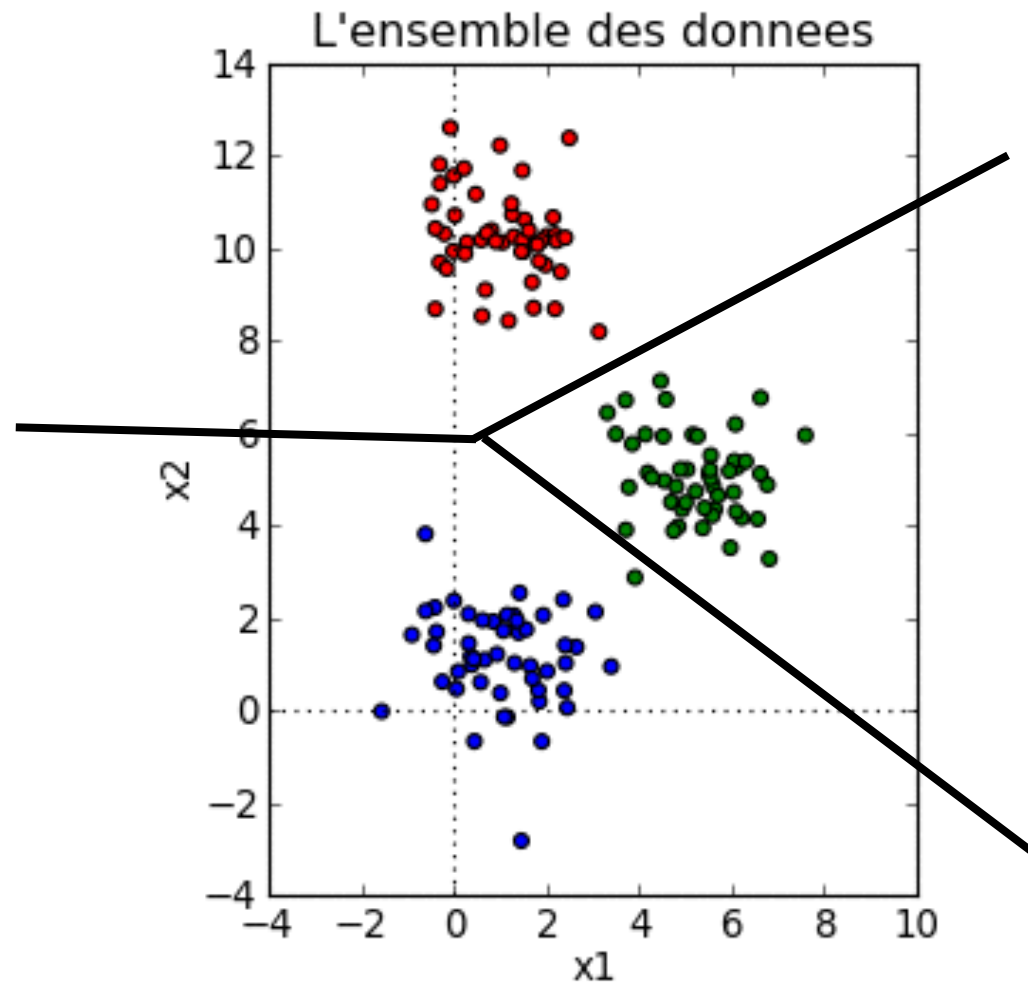
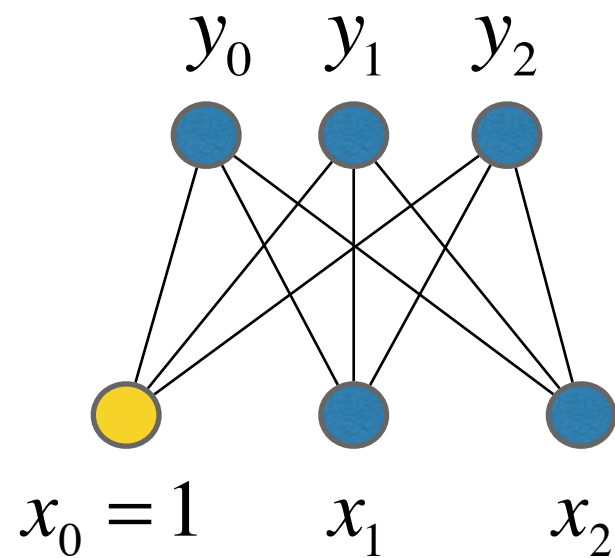
Relation avec
classification
d'images ?

1. Que peut faire un neurone avec la fonction d'activation sigmoïde et les poids malléables ?

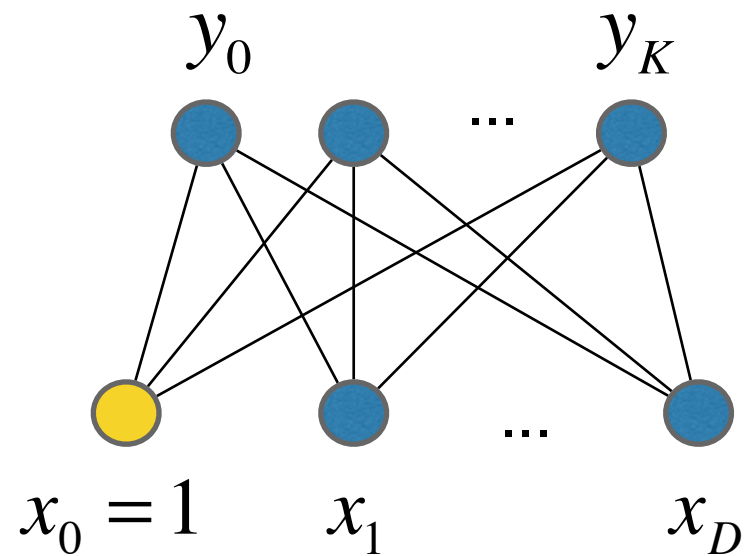
Un neurone avec D entrées (+ biais) peut séparer des données **linéairement séparables** dans D dimensions en deux classes

La surface de séparation est un **hyperplan**

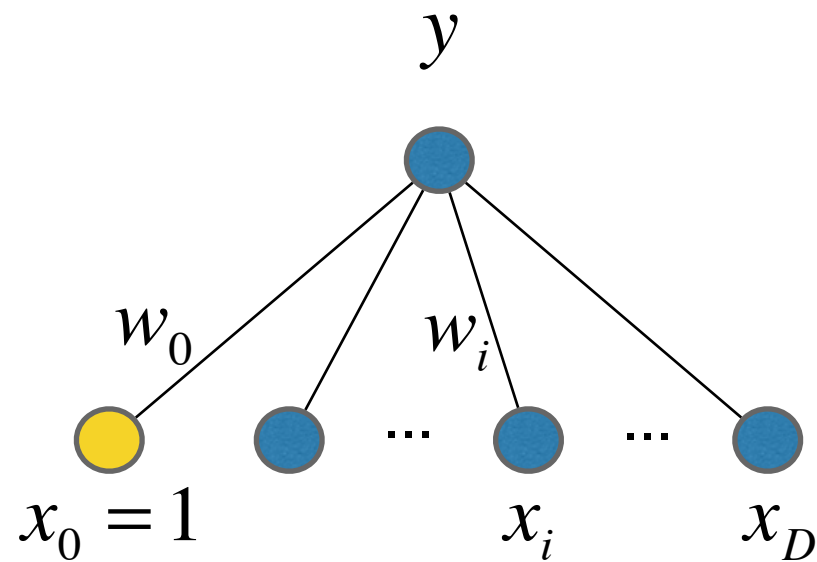
3 neurones / 2 entrées



K neurones / D entrées



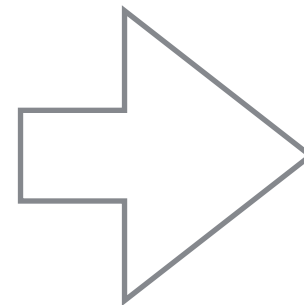
K neurones avec D entrées (+biais)
partage l'espace des données en K
volumes convexes séparées par des
hyperplans



On a compris que peut faire un neurone avec poids synaptiques malléables



Mais comment trouver les poids correctes ?



Apprentissage !

Neurones avec **apprentissage**

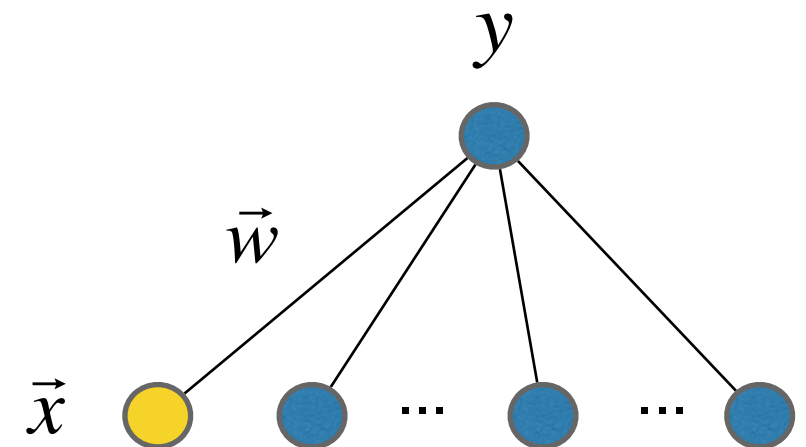
Apprentissage **supervisé**

- On a un ensemble des N vecteurs de données avec les étiquettes correspondantes (rouge/bleu) :

$$\left\{ \vec{x}^{(1)}, \vec{x}^{(2)}, \dots, \vec{x}^{(p)}, \dots, \vec{x}^{(N)} \right\}$$
$$\left\{ t^{(1)}, t^{(2)}, \dots, t^{(p)}, \dots, t^{(N)} \right\}$$

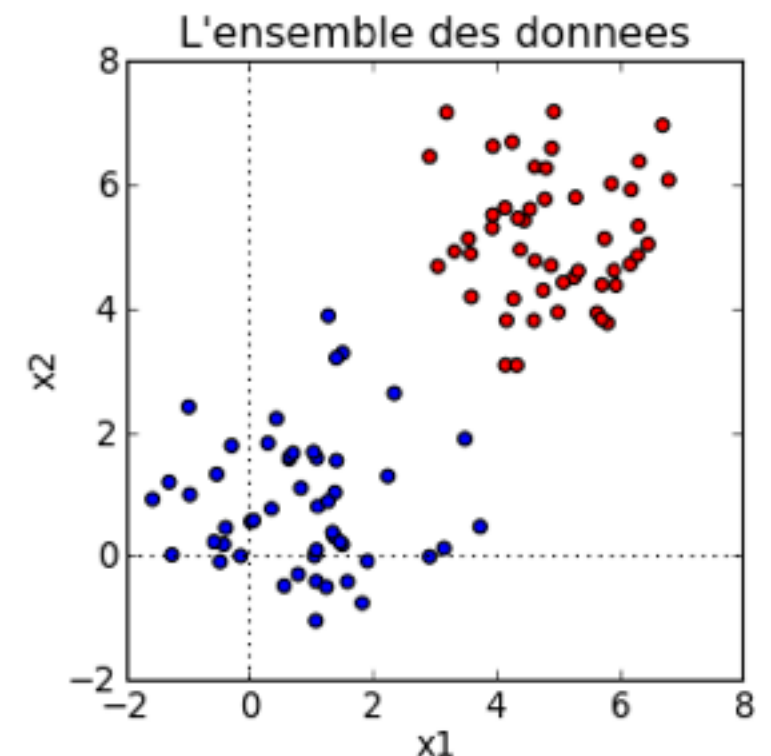
- On applique le vecteur des données $\vec{x}^{(p)}$ au réseau
- On calcule la sortie $y^{(p)}$
- On vérifie si la valeur $y^{(p)}$ correspond à l'étiquette de la classe, en calculant l'erreur

$$E^{(p)} = (y^{(p)} - t^{(p)})^2$$



$t = 1$
rouge

$t = 0$
bleu



Apprentissage **supervisé**

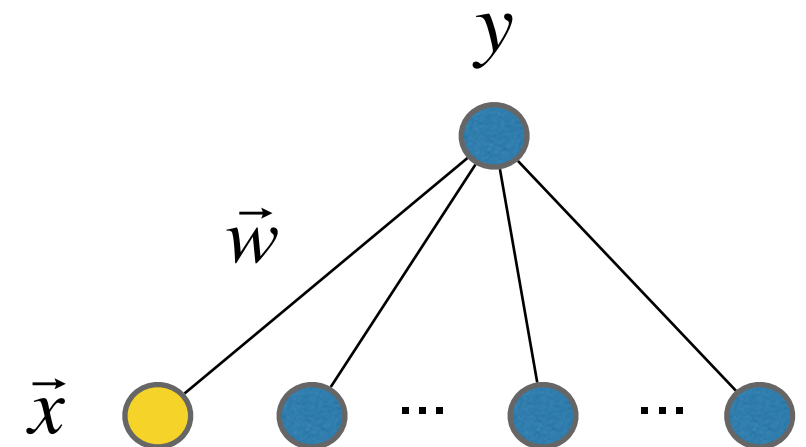
Après la présentation de toutes les données, on calcule l'erreur totale, où la **fonction-erreur** :

$$E(\vec{w}) = \frac{1}{2} \sum_{p=1}^N E^{(p)} = \frac{1}{2} \sum_{p=1}^N (y^{(p)} - t^{(p)})^2$$

où

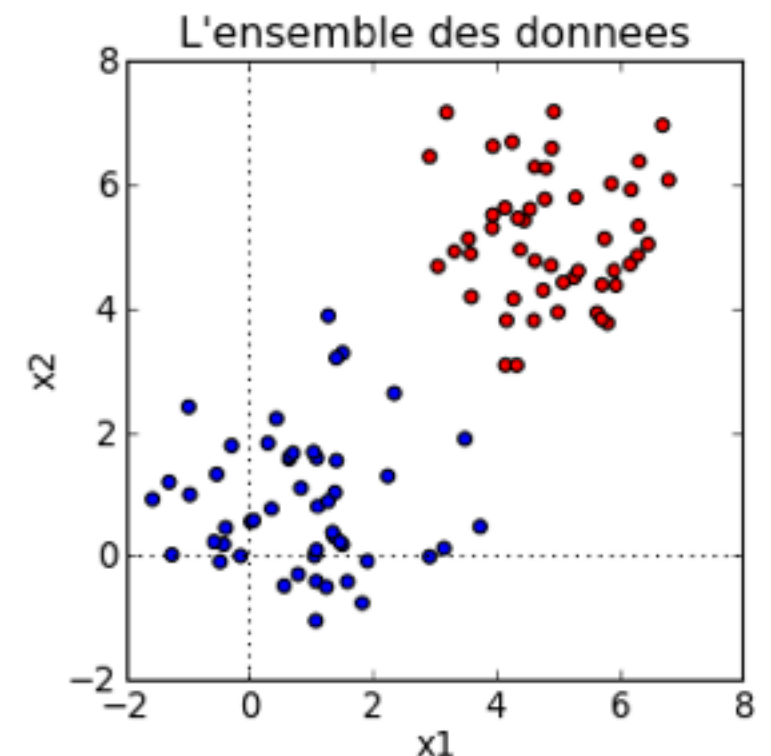
$$y^{(p)} = g(a) = g\left(\sum_{i=0}^D w_i x_i^{(p)}\right) = g(\vec{w} \cdot \vec{x}^{(p)})$$

La fonction-erreur est une fonction de poids. Le minimum de la fonction d'erreur correspond aux poids optimales pour classification

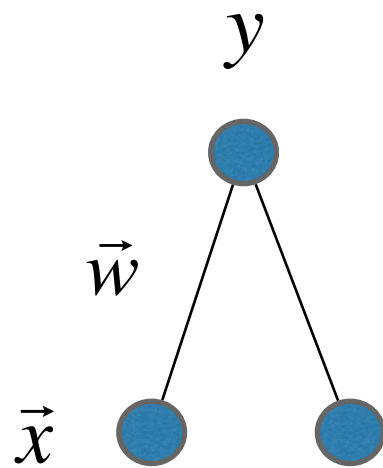


$t = 1$
rouge

$t = 0$
bleu

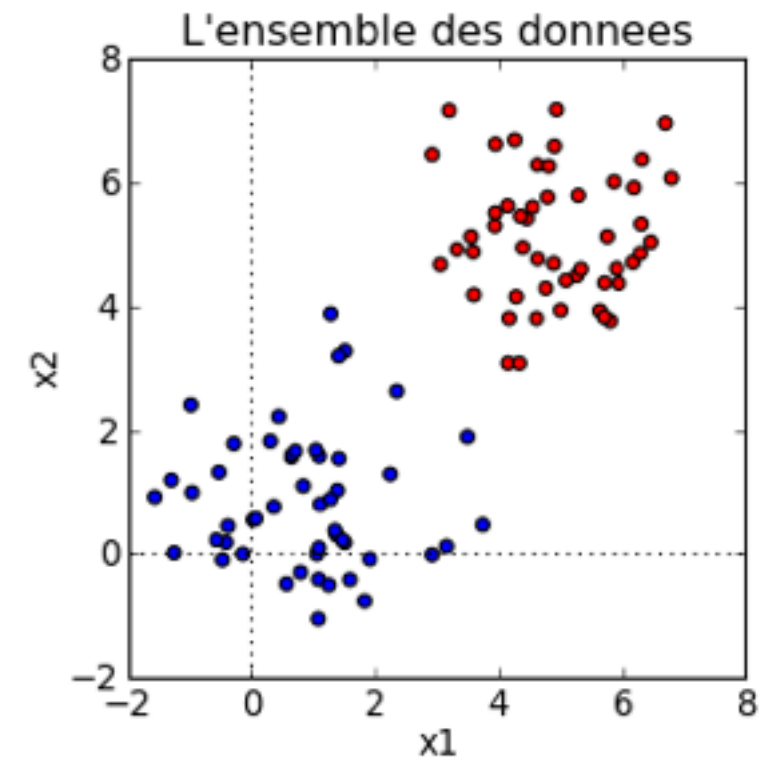


Apprentissage supervisé : fonction-erreur



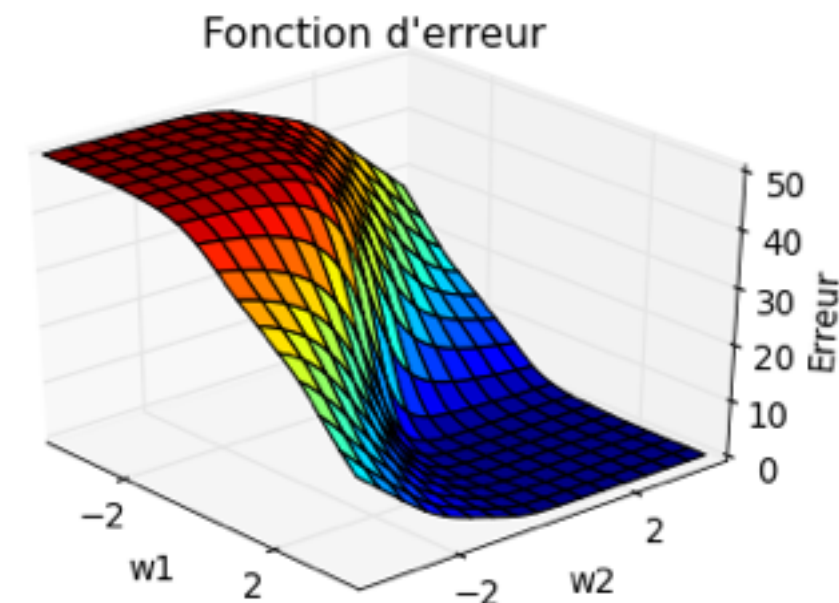
$t = 1$
rouge

$t = 0$
bleu



$$E(w_1, w_2) = \frac{1}{2} \sum_{p=1}^N (y^{(p)} - t^{(p)})^2$$

$$y^{(p)} = g(w_1 \cdot x_1^{(p)} + w_2 \cdot x_2^{(p)})$$



Apprentissage supervisé : **descente de gradient**

Minimisation de la fonction-erreur :

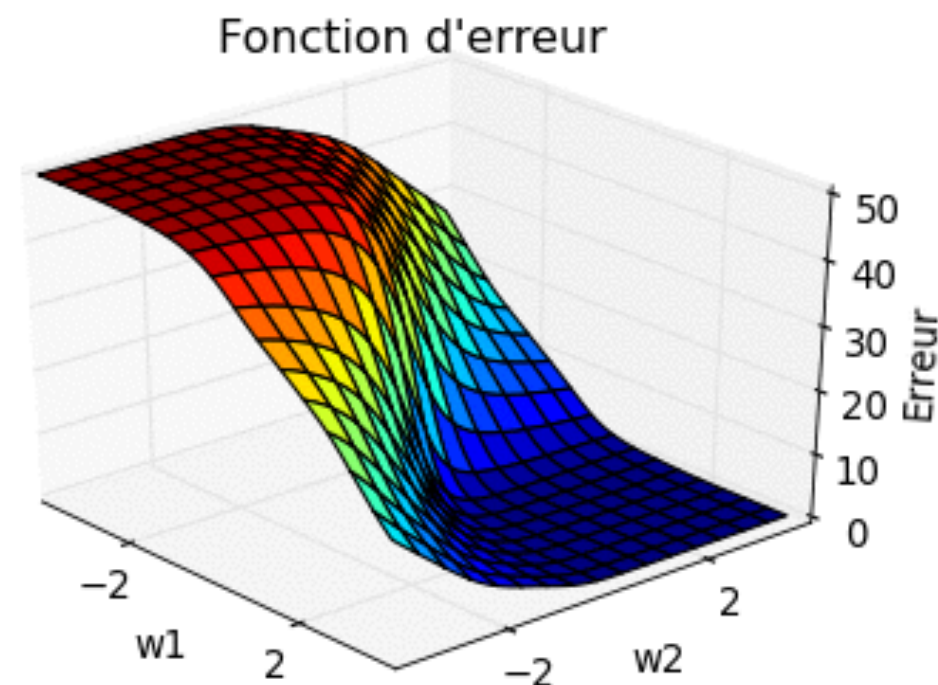
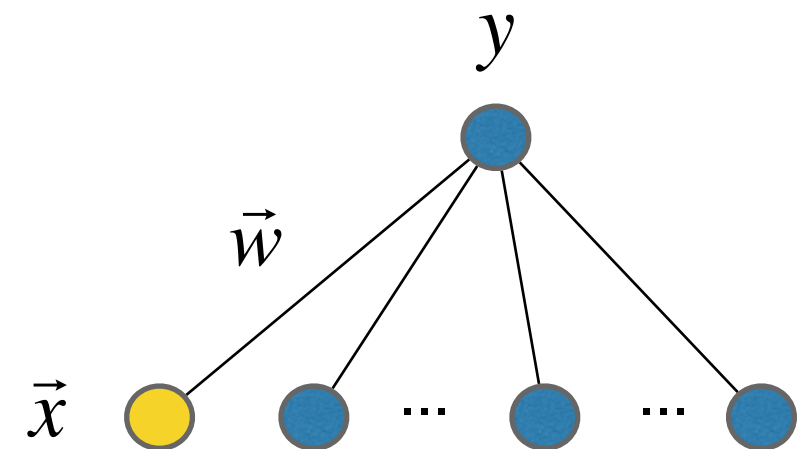
- On commence par les poids aléatoires
- On suit le gradient négative de la fonction-erreur pour descendre au minimum
- En calculant le gradient on arrive (voir TD) à la règle suivante pour la mise à jour des poids après la présentation du vecteur $\vec{x}^{(p)}$:

$$\Delta w_i = -\eta \delta x_i^{(p)}$$

où

$$\delta = g'(a)(y^{(p)} - t^{(p)})$$

s'appelle erreur delta (**delta-error**) et $\eta \ll 1$ est le taux d'apprentissage



La règle d'apprentissage si-dessus pour des réseaux à une couche s'appelle "delta rule" (anglais)

Apprentissage supervisé : procédure

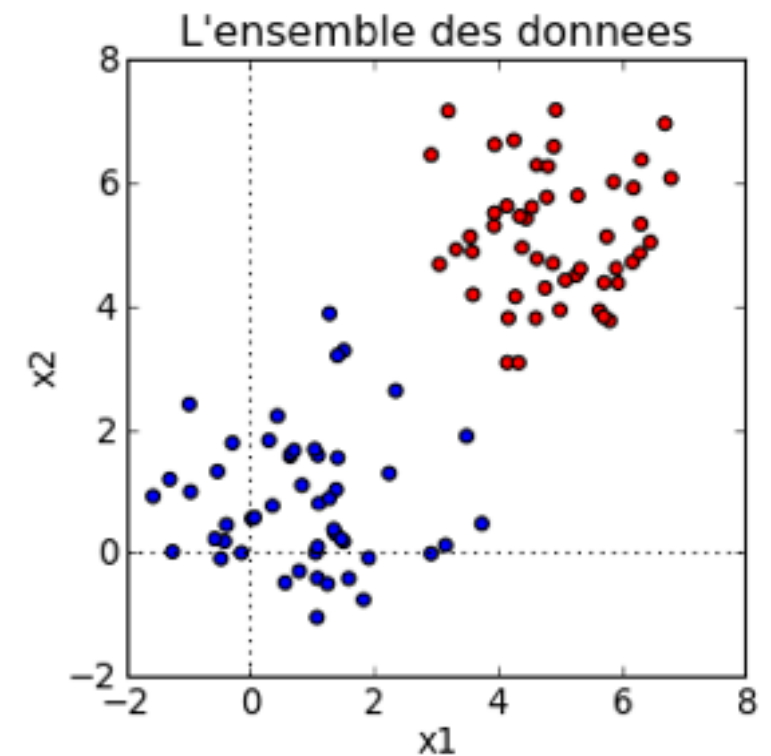
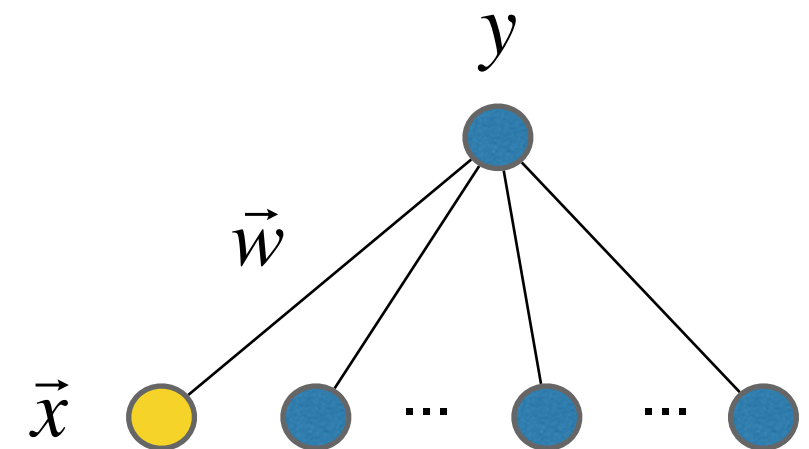
1. Choisir aléatoirement vecteur $\vec{x}^{(p)}$
2. Calculer $y^{(p)}$
3. Calculer l'erreur delta

$$\delta = g'(a)(y^{(p)} - t^{(p)})$$

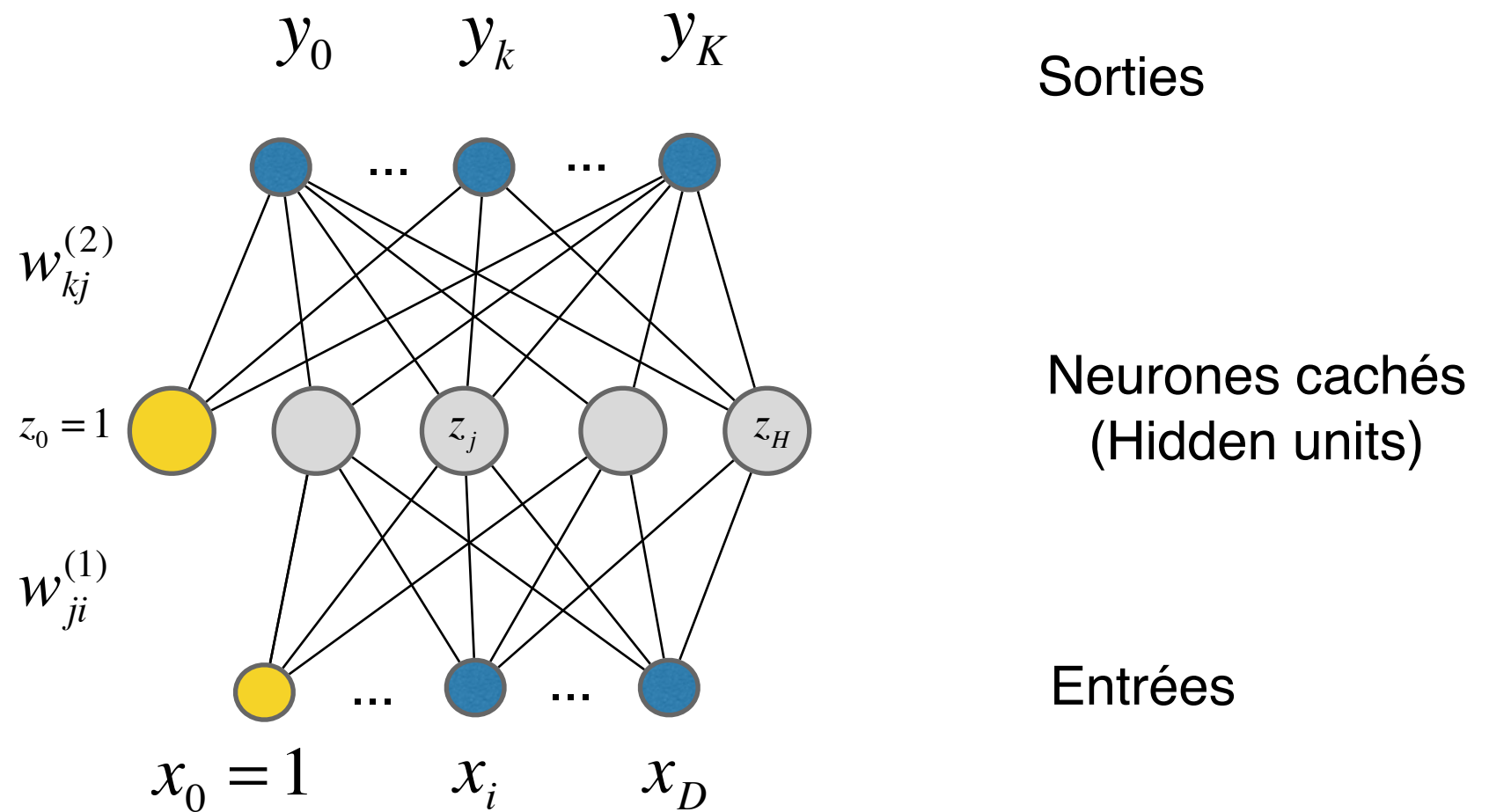
4. Mettre à jour les poids synaptiques en utilisant "delta rule"

$$\Delta w_i = -\eta \delta x_i^{(p)}$$

5. Répéter les étapes 1-4. On arrête l'apprentissage quand tous les vecteurs $\vec{x}^{(p)}$ sont correctement classifiés.

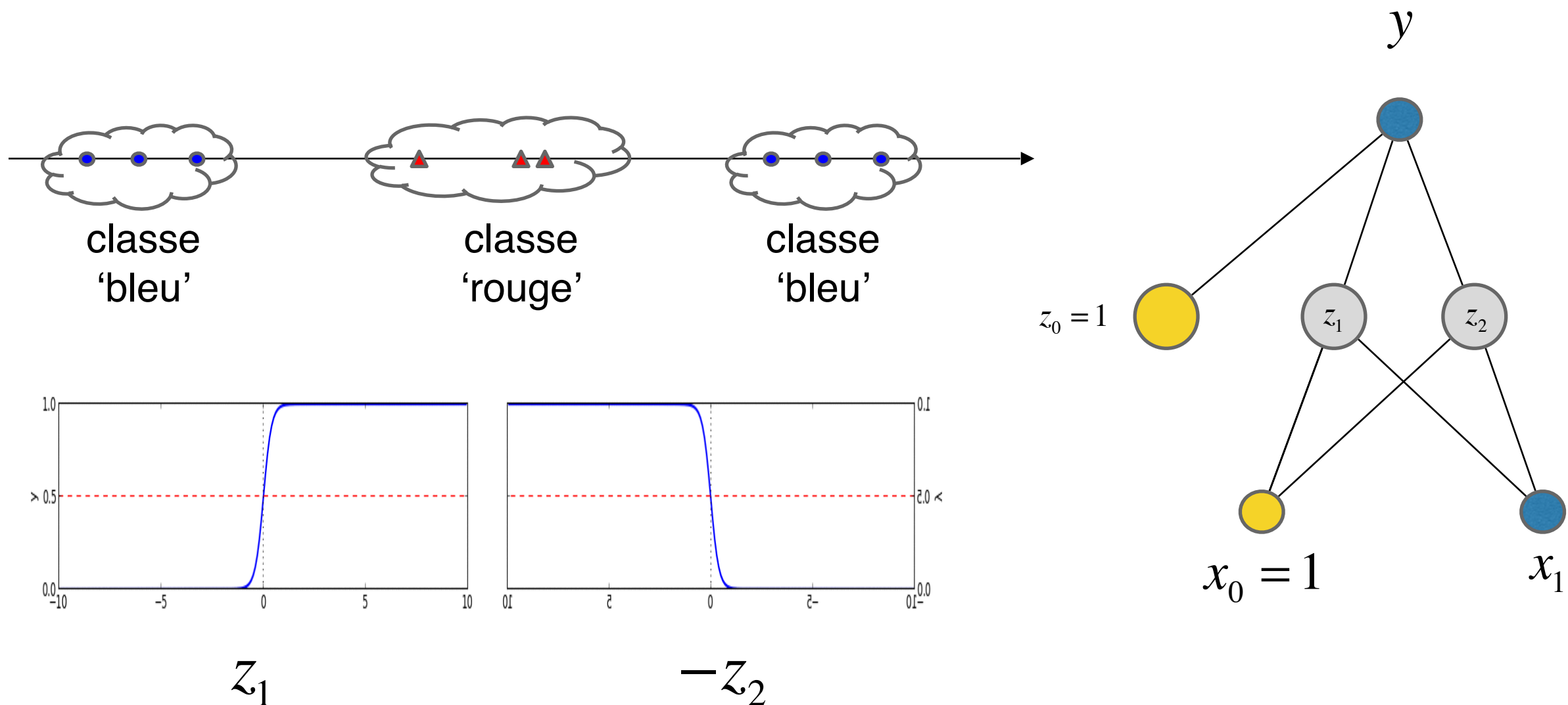


Réseau multicouche



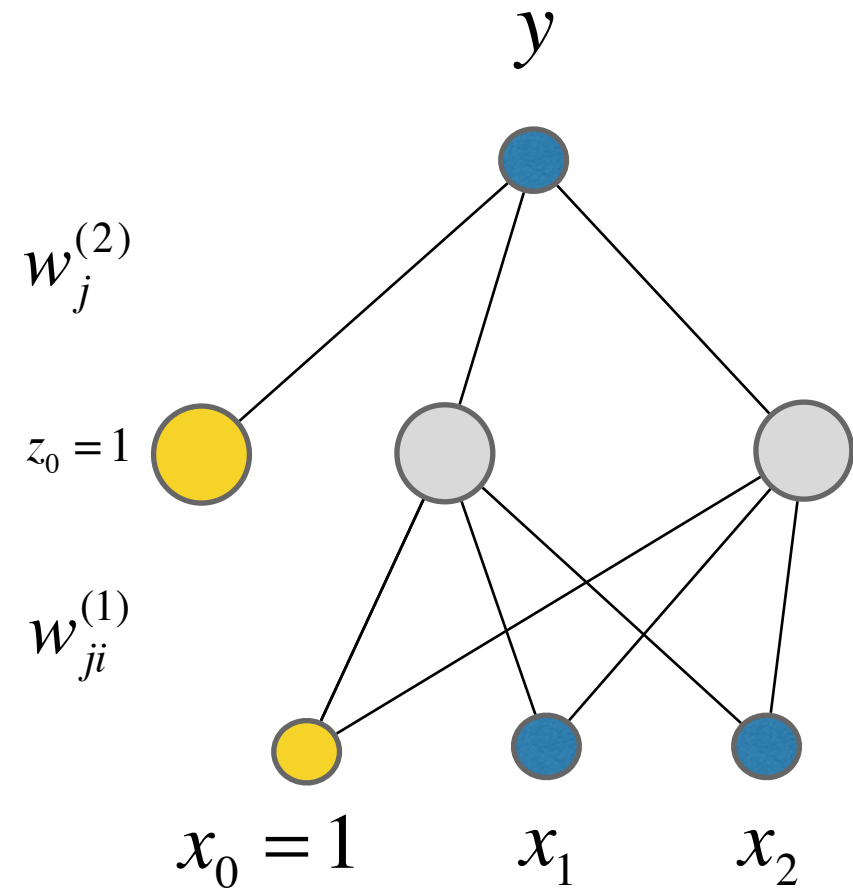
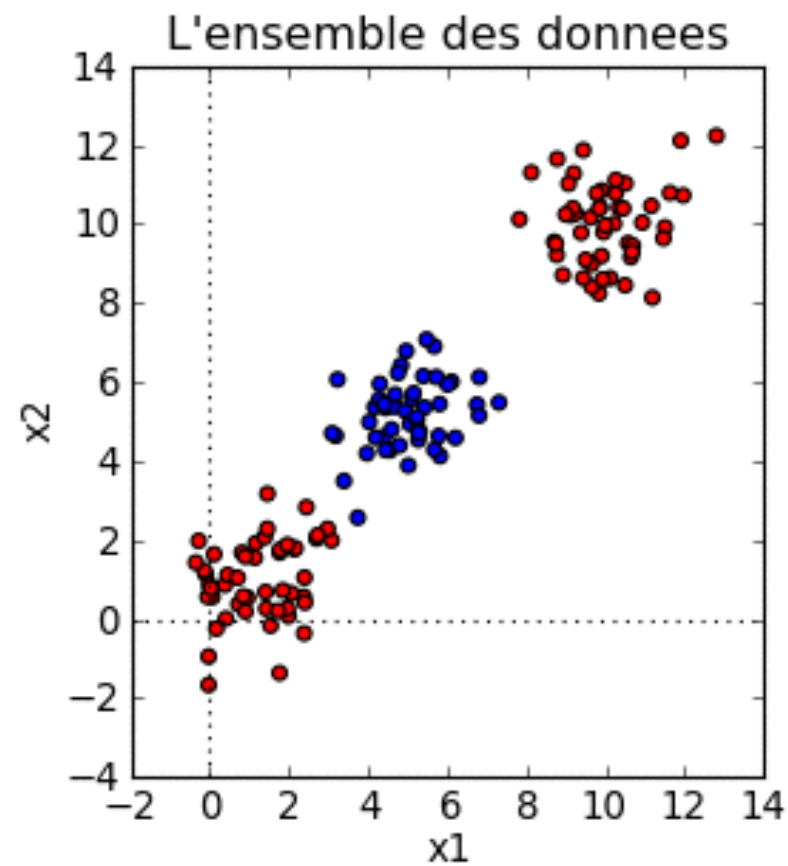
$$y_k = \sum_{j=0}^H w_{kj}^{(2)} z_j = w_{k0} + \sum_{j=1}^H w_{kj}^{(2)} g(a_j) = w_{k0} + \sum_{j=1}^H w_{kj}^{(2)} g\left(\sum_{i=0}^D w_{ji}^{(1)} x_i\right)$$

Réseau multicouche



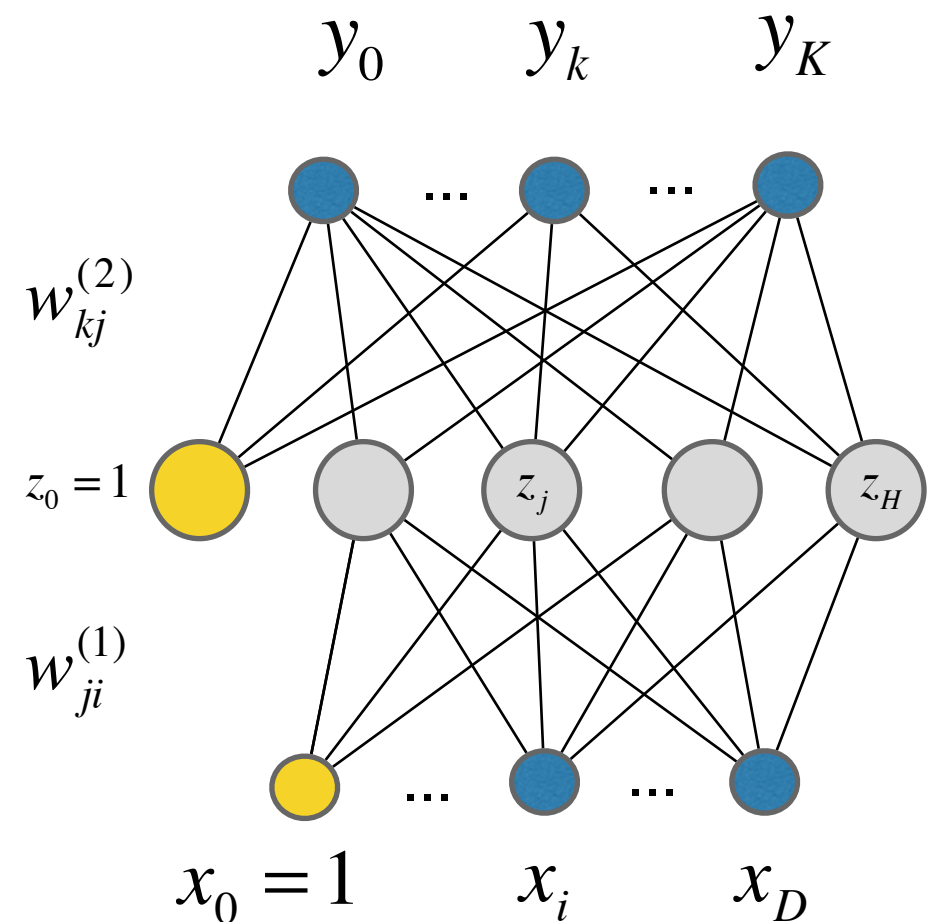
Le neurone de sortie calcule une somme pondérée de sigmoïdes.
Une combinaison linéaire des sigmoïdes peut approximer
n'importe quelle fonction continue (la précision dépend du nombre des neurones cachés).

Réseau multicouche



Réseau multicouche - classification

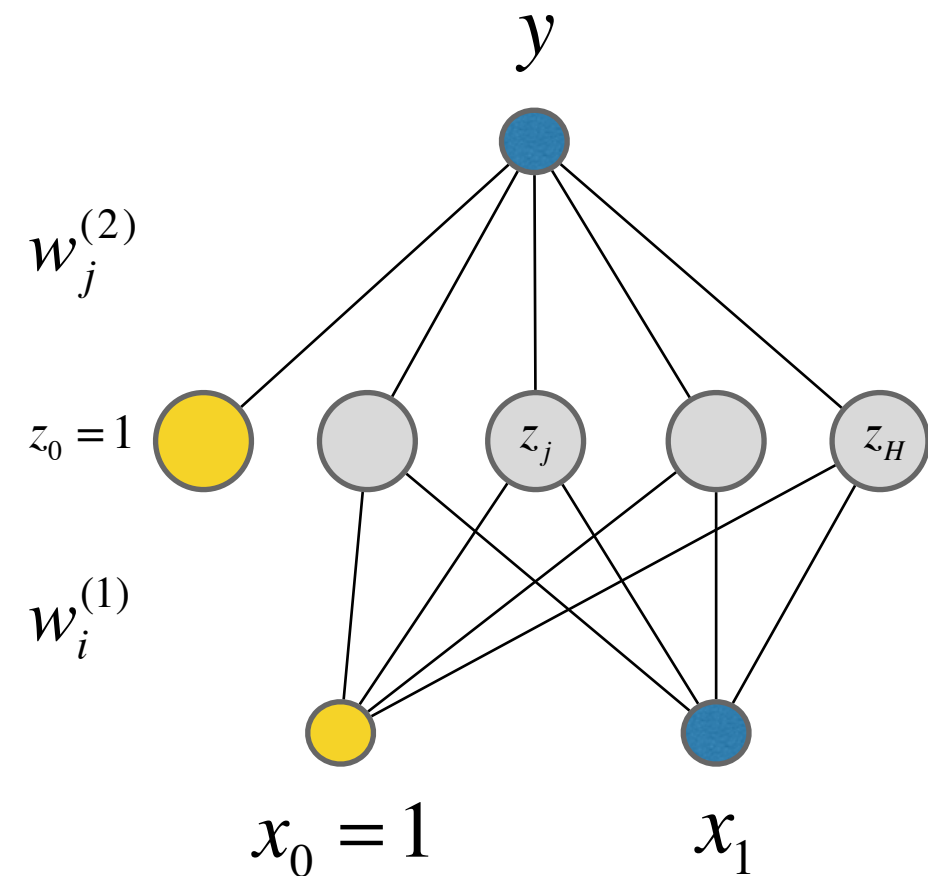
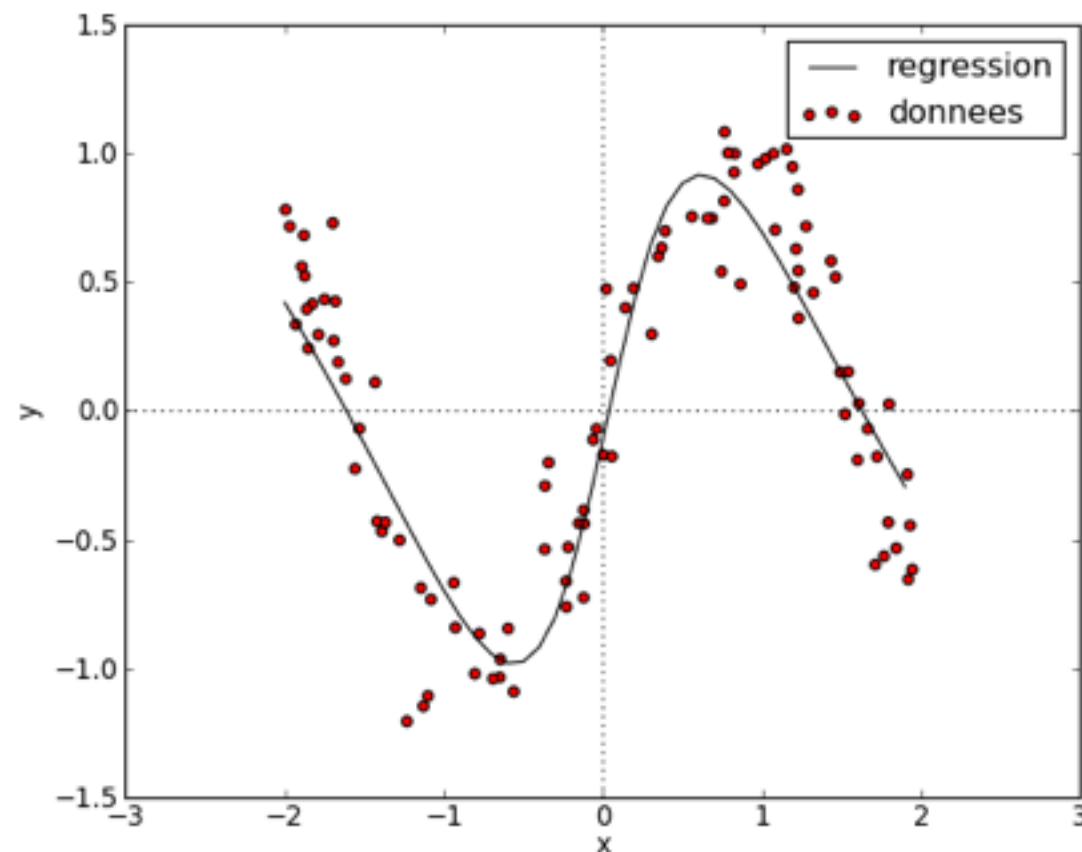
Un réseau multicouche avec D entrées et K sorties peut séparer des données dans D dimensions en K classes avec les surfaces de séparation **arbitraires**



Réseau de neurones feed-forward
à deux (ou plus) couches de poids :

“perceptron multi-couche”

Réseau multicouche : régression non-linéaire



Une combinaison linéaire des sigmoïdes peut approximer n'importe quelle fonction continue ->

ce réseau peut effectuer une régression linéaire
(la précision dépend du nombre des neurones cachés).

Réseau multicouche : apprentissage

La méthode de descente du gradient peut être utilisé pour des réseaux à plusieurs couches (voir TD).

Le nom de cet algorithme est
“rétro-propagation du gradient de l’erreur”

Réseau multicouche : apprentissage

1. Choisir aléatoirement vecteur $\vec{x}^{(p)}$
2. Calculer $y^{(p)}$
3. Calculer l'erreur delta pour la 2^{ème} couche
$$\delta = y^{(p)} - t^{(p)}$$
4. Mettre à jour les poids synaptiques pour la 2^{ème} couche

$$\Delta w_j^{(2)} = -\eta \delta z_j^{(p)}$$

où

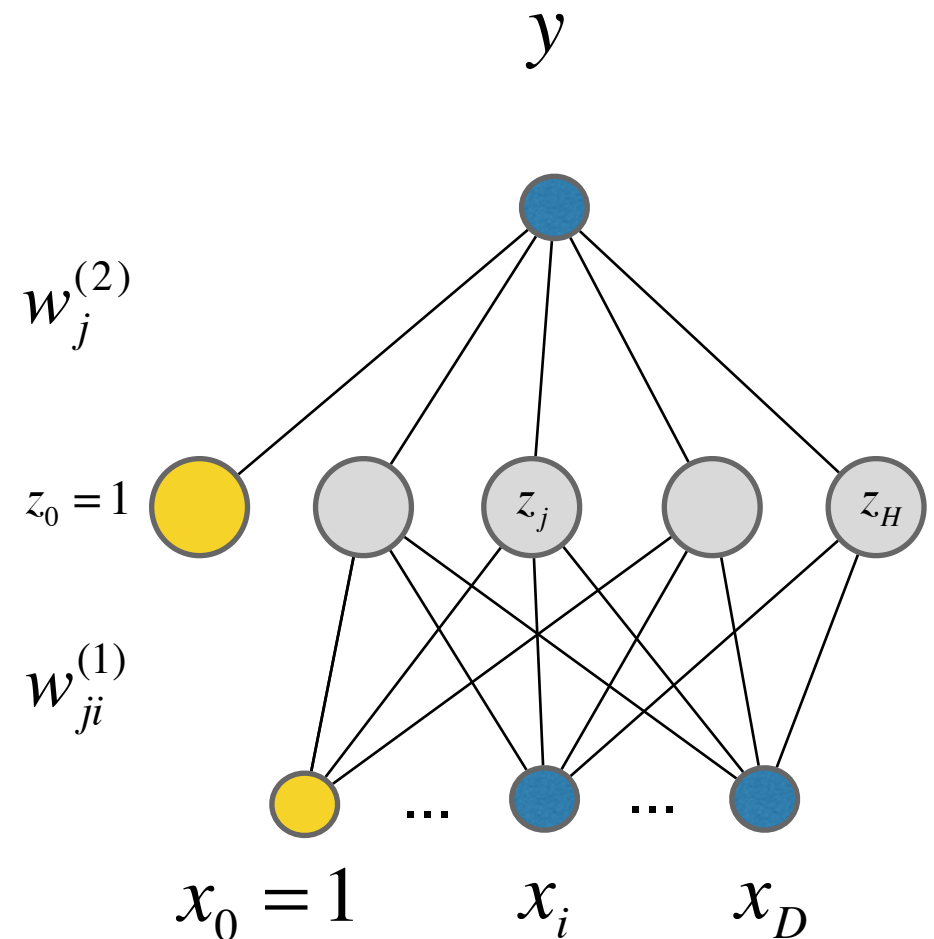
$$z_j = g \left(\sum_{i=0}^D w_{ji}^{(1)} x_i \right)$$

5. Calculer l'erreur delta pour la 1^{ère} couche

$$\delta_j = g'(a_j) w_j^{(2)} \delta$$

6. Mettre à jour les poids pour la 1^{ère} couche

$$\Delta w_{ji}^{(1)} = -\eta \delta_j x_i^{(p)}$$

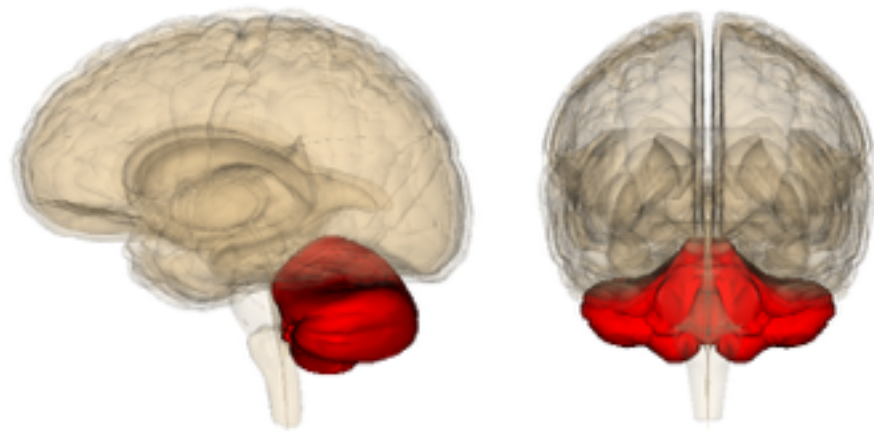


Questions principales

1. Que peut faire un neurone avec la fonction d'activation sigmoïde et les poids malléables ?
2. Que peut faire un réseau de tels neurones ?

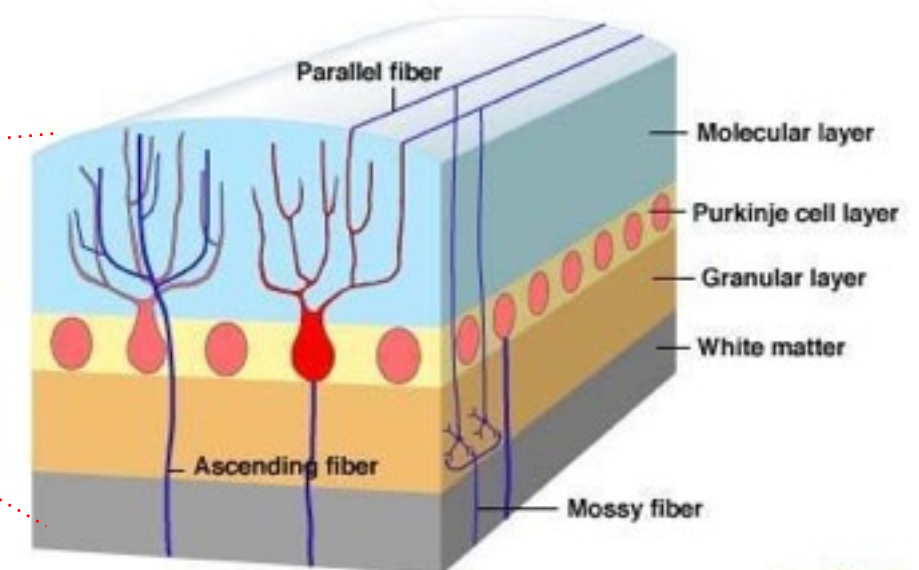
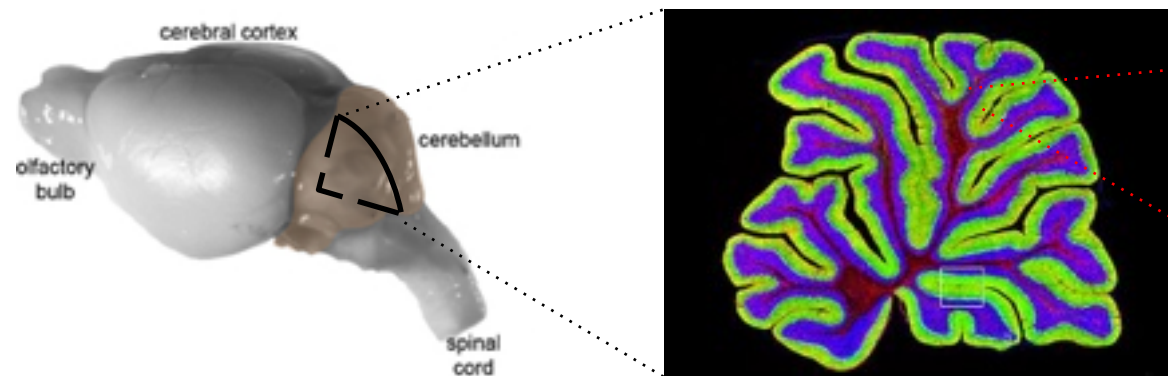
Exemple I : Apprentissage supervisé dans le cervelet

Homme

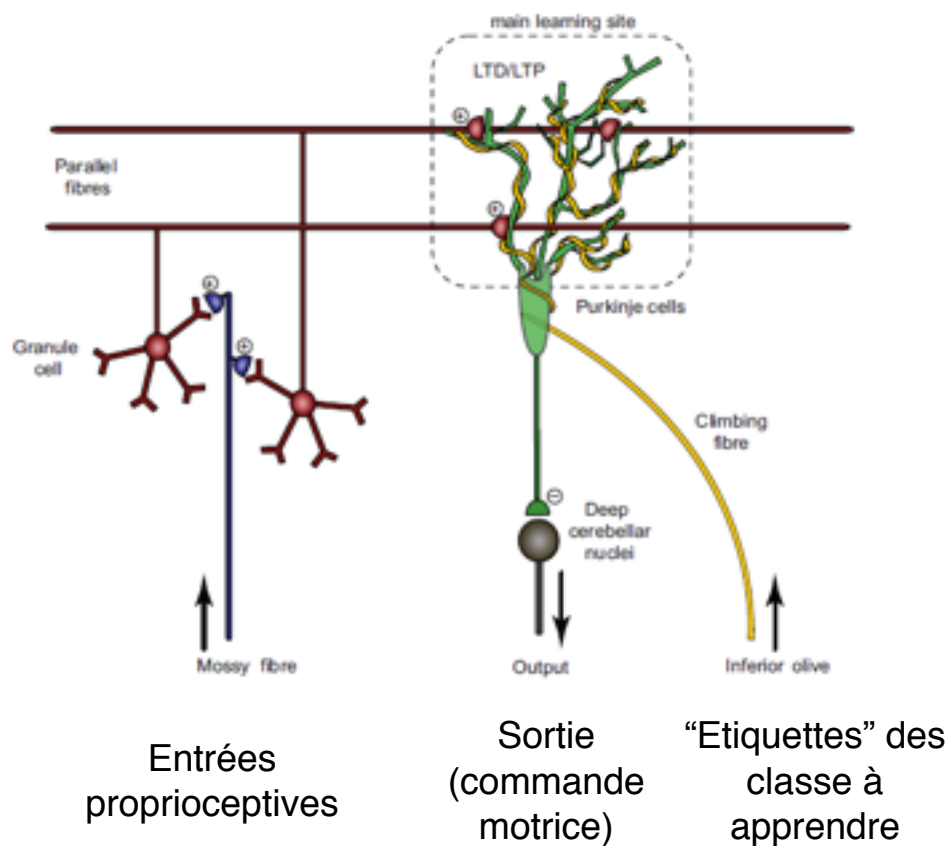


- Cervelet joue le rôle principal dans le contrôle moteur (ajustement et contrôle du mouvement)
- Exemple : réflexe vestibulaire - oculaire, ajustement d'une erreur de préhension, ajustement du mouvement orienté vers le but
- Cervelet est constitué d'une couche mince et continue de tissu nerveux, comprenant deux couches de neurones

Rat



Exemple I : Apprentissage supervisé dans le cervelet



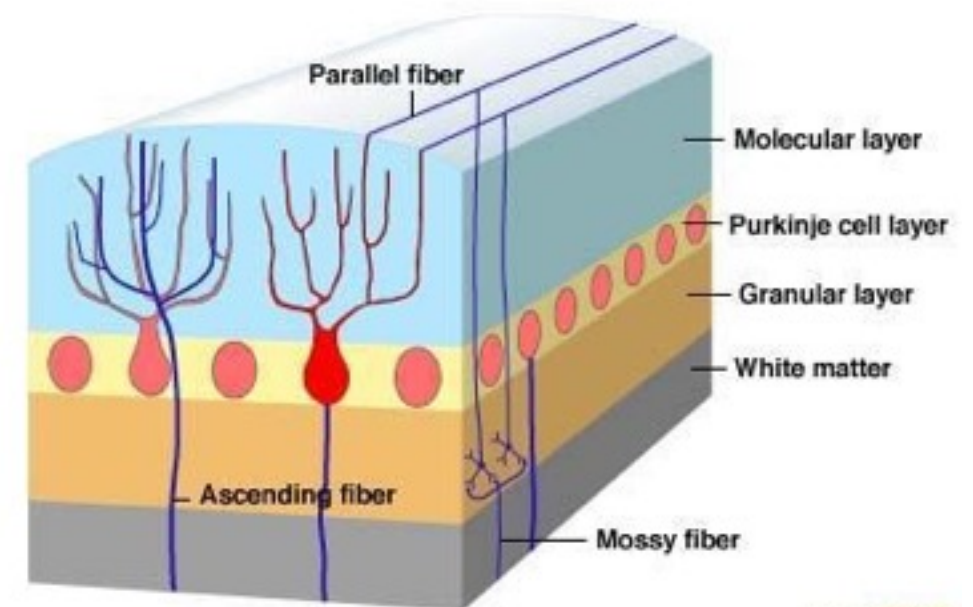
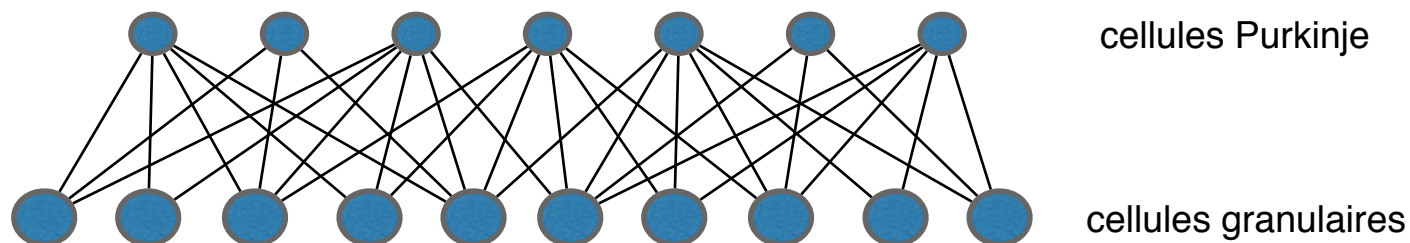
A Theory of Cerebellar Function

JAMES S. ALBUS

ABSTRACT

A comprehensive theory of cerebellar function is presented, which ties together the known anatomy and physiology of the cerebellum into a pattern-recognition data processing system. The cerebellum is postulated to be functionally and structurally equivalent to a modification of the classical Perceptron pattern-classification device. It is suggested that the mossy fiber → granule cell → Golgi cell input network performs an expansion recoding that enhances the pattern-discrimination capacity and learning speed of the cerebellar Purkinje response cells.

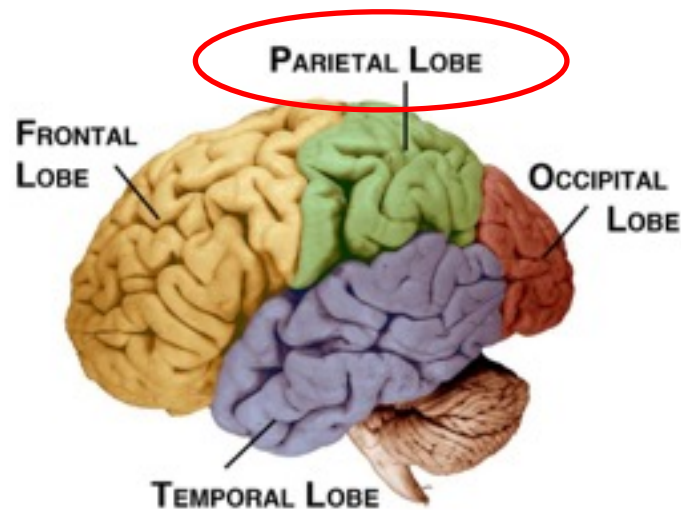
Albus, J. S. (1971). 10, 25–61.



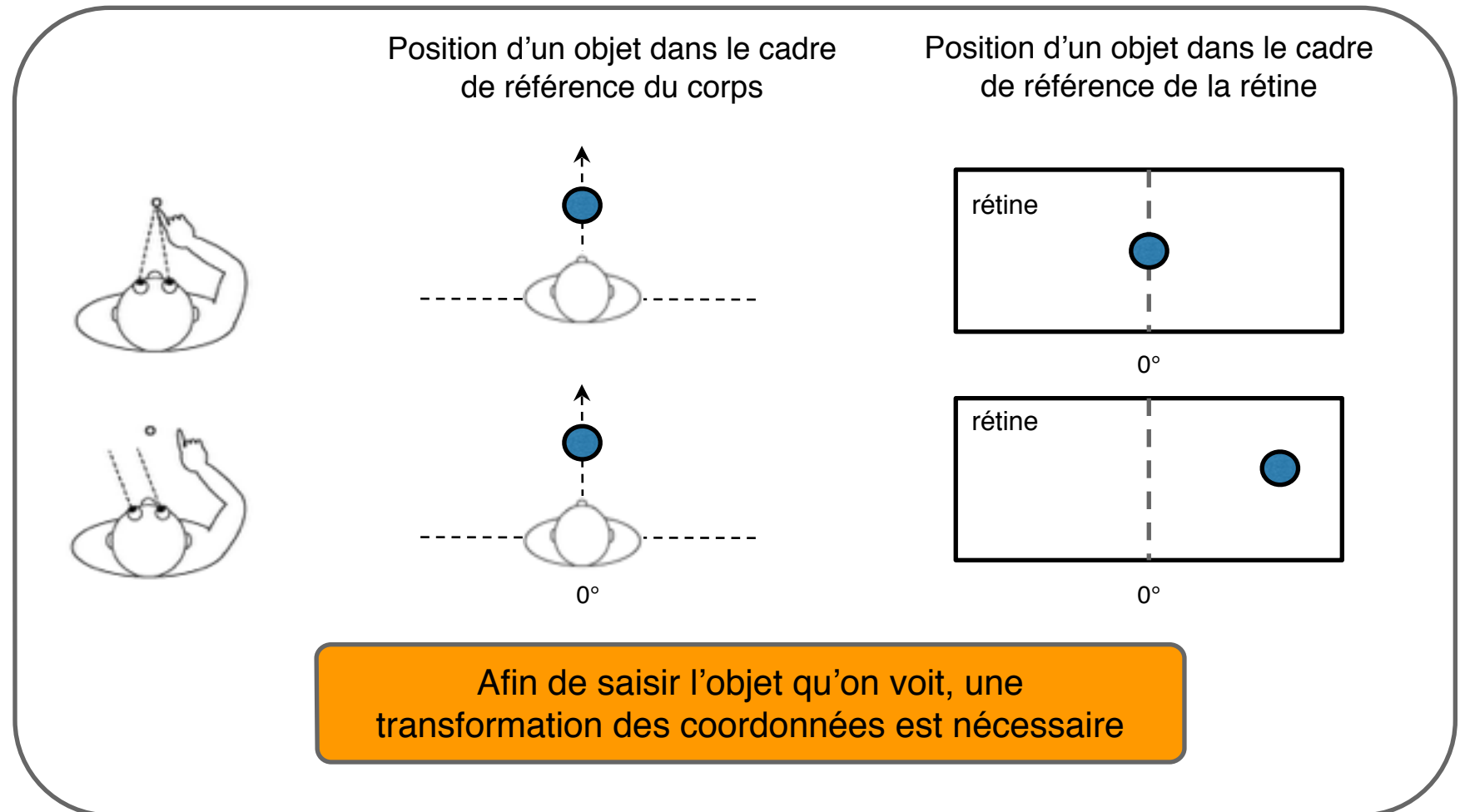
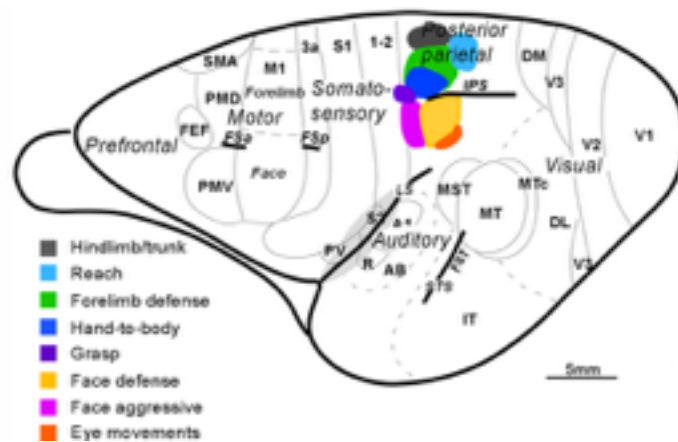
Exemple II: Cortex pariétal

Homme

- planification du mouvement
- coordination sensorielle - motrice
- **transformation des coordonnées**
- attention

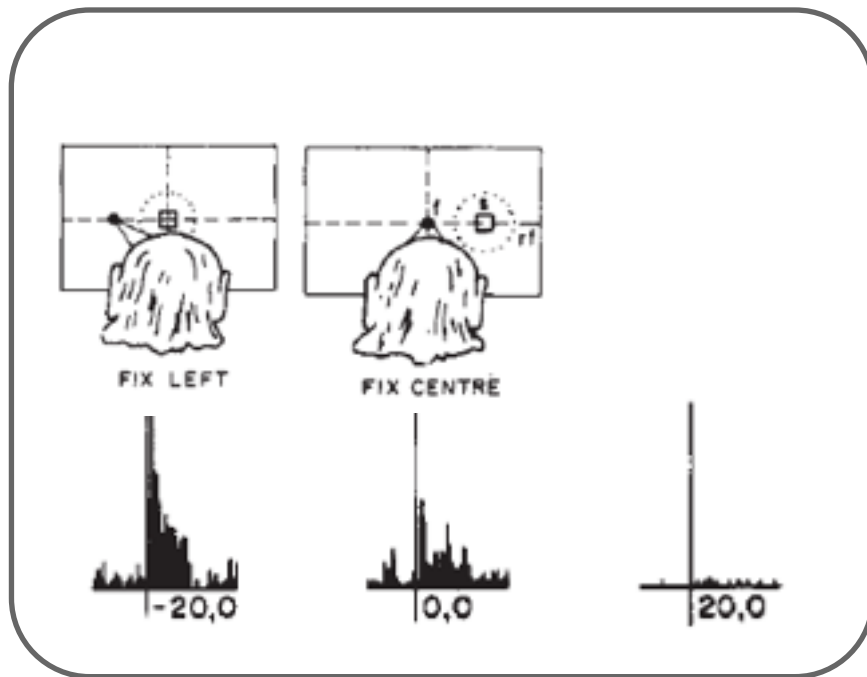


Rat



Transformation des coordonnées dans le cortex pariétal

Expérience chez les singes

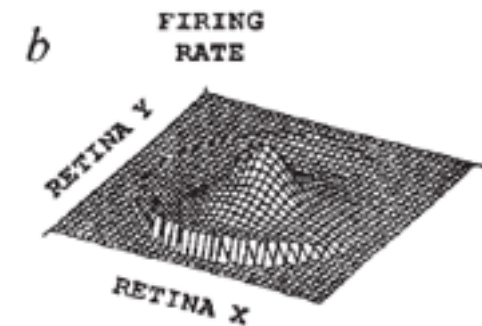
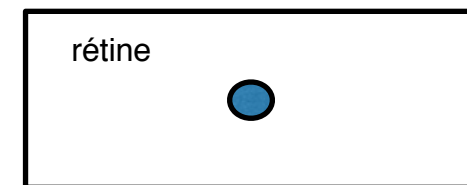


Zipser & Andersen (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, 331, 679–684.

Quel est le rôle des cellules les plus nombreuses (de type 3) ?

Trois types de neurones ont été observés :

1. Neurones qui encodent la position rétinienne du stimulus (21%)



2. Neurones qui encodent la position des yeux (15%)

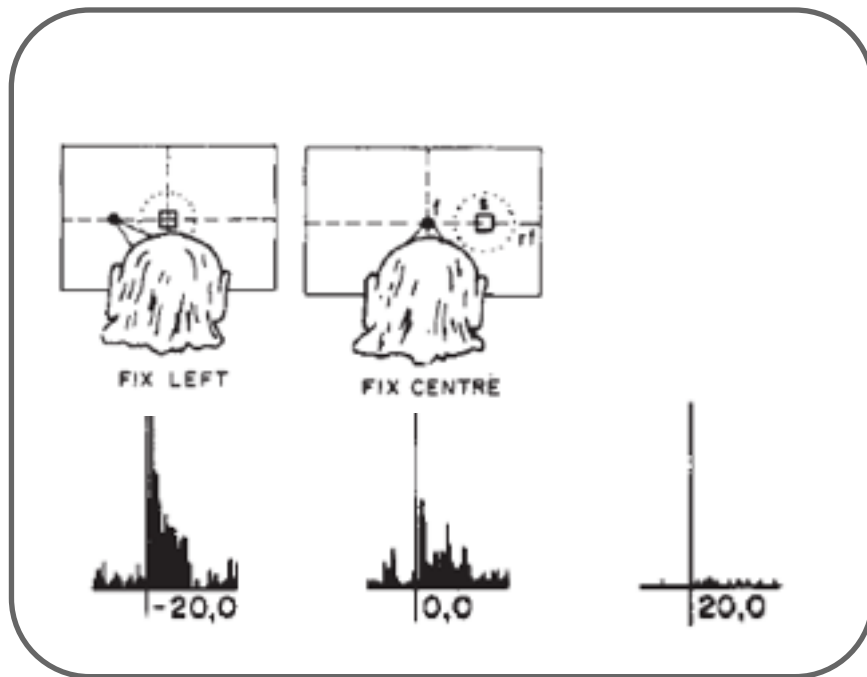


3. Neurones qui encodent les deux au même temps (57%)



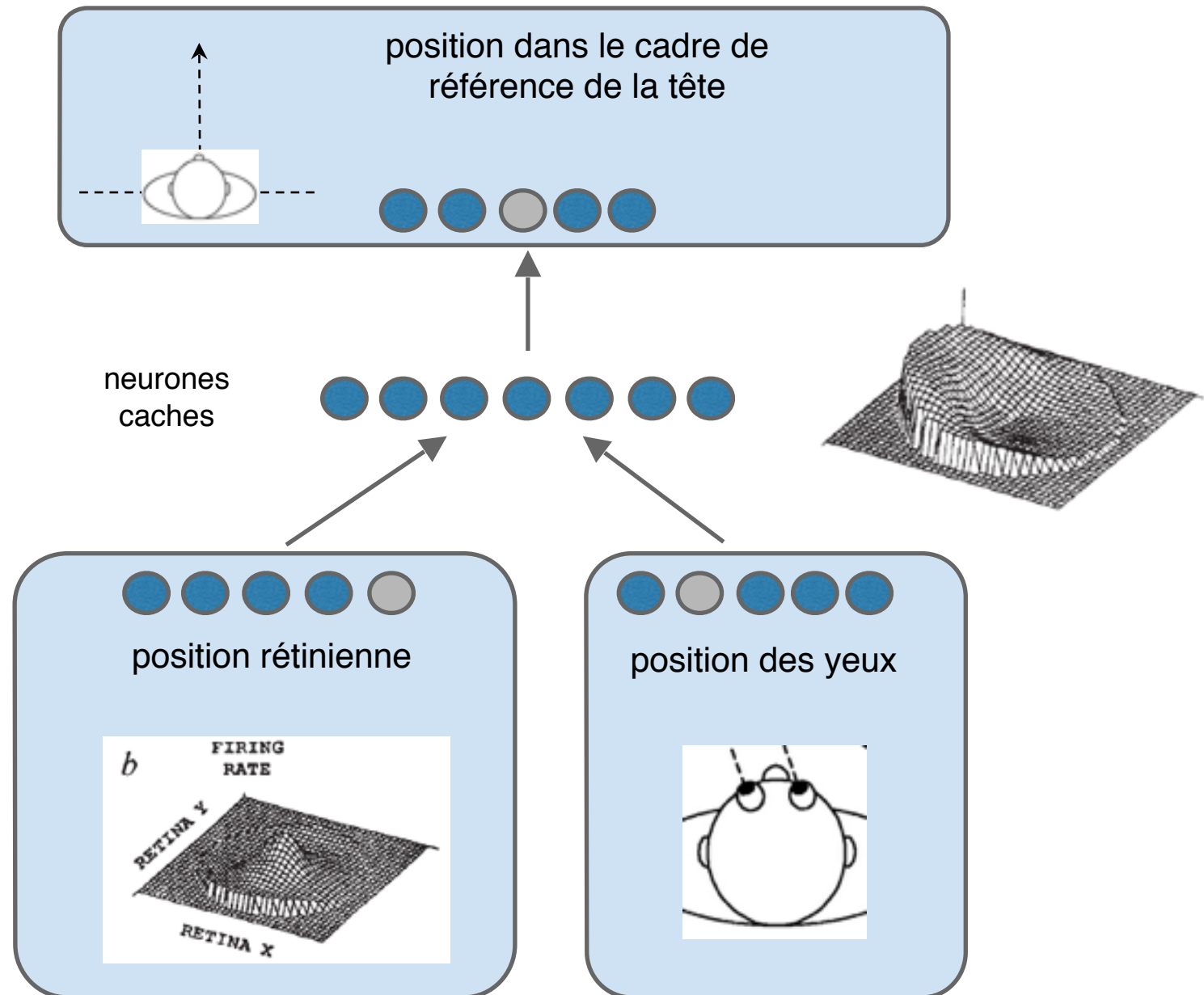
Transformation des coordonnées dans le cortex pariétal

Expérience chez les singes



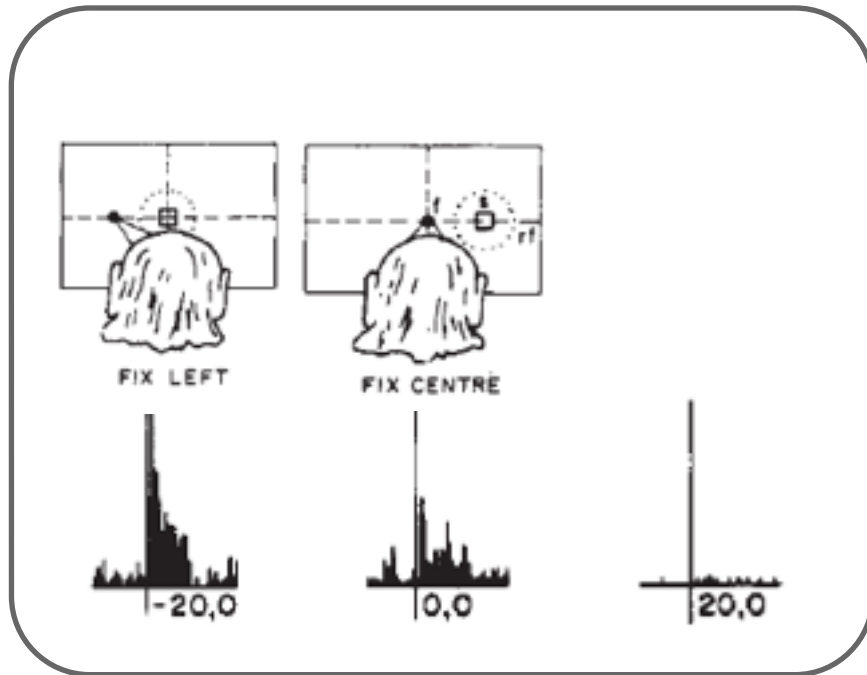
Zipser & Andersen (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, 331, 679–684.

Quel est le rôle des cellules les plus nombreuses (de type 3) ?



Transformation des coordonnées dans le cortex pariétal

Expérience chez les singes



Zipser & Andersen (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature*, 331, 679–684.

Quel est le rôle des cellules les plus nombreuses (de type 3) ?

A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons

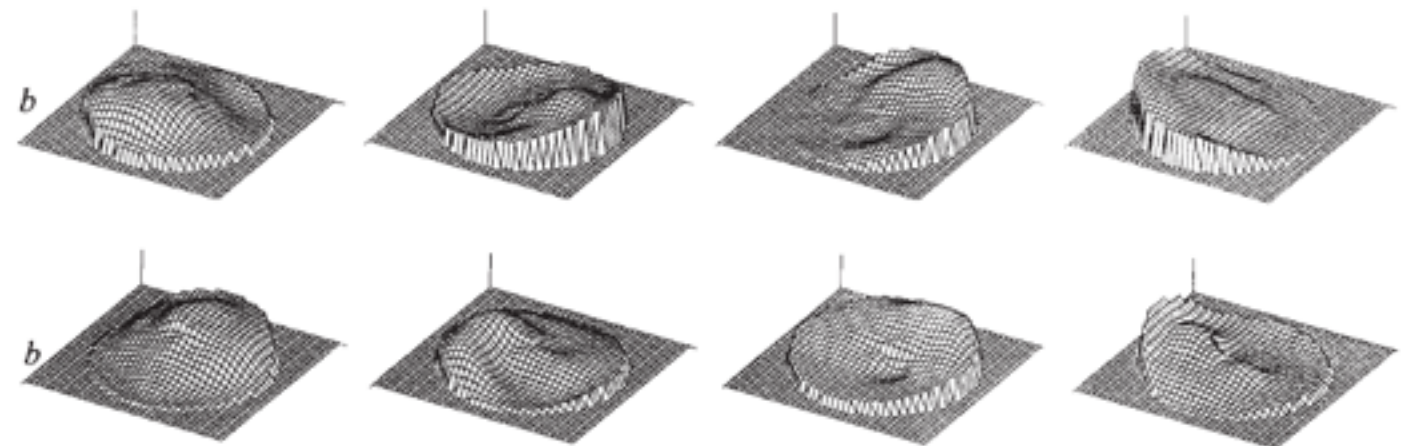
David Zipser* & Richard A. Andersen*

* Institute for Cognitive Science, University of California, San Diego, La Jolla, California 92093, USA

† Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

Neurons in area 7a of the posterior parietal cortex of monkeys respond to both the retinal location of a visual stimulus and the position of the eyes and by combining these signals represent the spatial location of external objects. A neural network model, programmed using back-propagation learning, can decode this spatial information from area 7a neurons and accounts for their observed response properties.

Nature, 1988, 331, 679–684.



Les activités des neurones cachés dans le modèle reproduisent bien les activités des neurones enregistrés expérimentalement

Bibliographie

- Hertz, Krogh, Palmer, 1991. Introduction to the Theory of Neural Computation
- Bishop, 1995. Neural networks for pattern recognition
- MacKay, 2003. Information Theory, Inference, and Learning Algorithms