

Plan

1. Pourquoi toute cette excitation ?
2. Grands types d'apprentissage
3. Apprentissage prédictif par réseaux de neurones
4. Quelles garanties ?
5. Recette pour créer des algorithmes d'apprentissage
6. Les réseaux de neurones profonds
7. Ce que l'on sait faire et les défis à relever

Les grandes familles d'apprentissage.

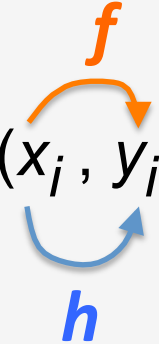
Enfin l'univers connu

Parce qu'il y a la matière noire et l'énergie sombre

Que l'on ne voit pas

Apprentissage prédictif (*supervisé*)

- A *learning set*

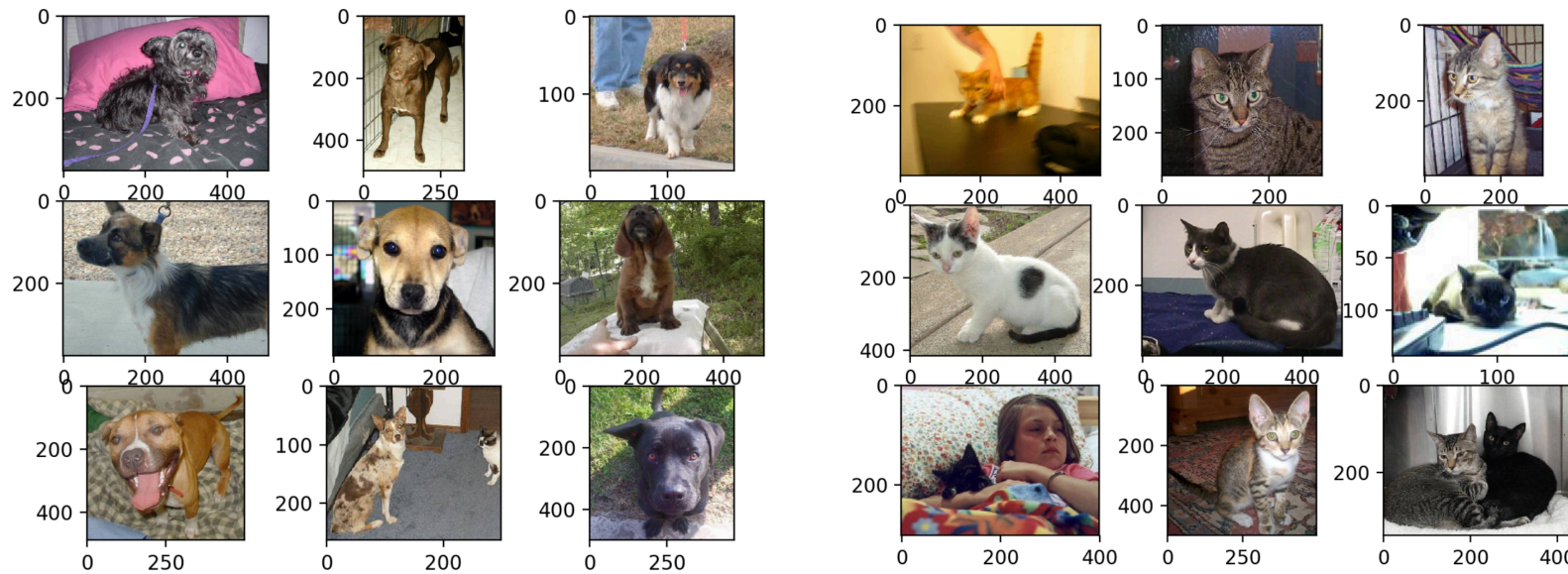
$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_j, y_j), \dots, (x_m, y_m)\}$$


Prediction for new examples $x \xrightarrow{h} y ?$

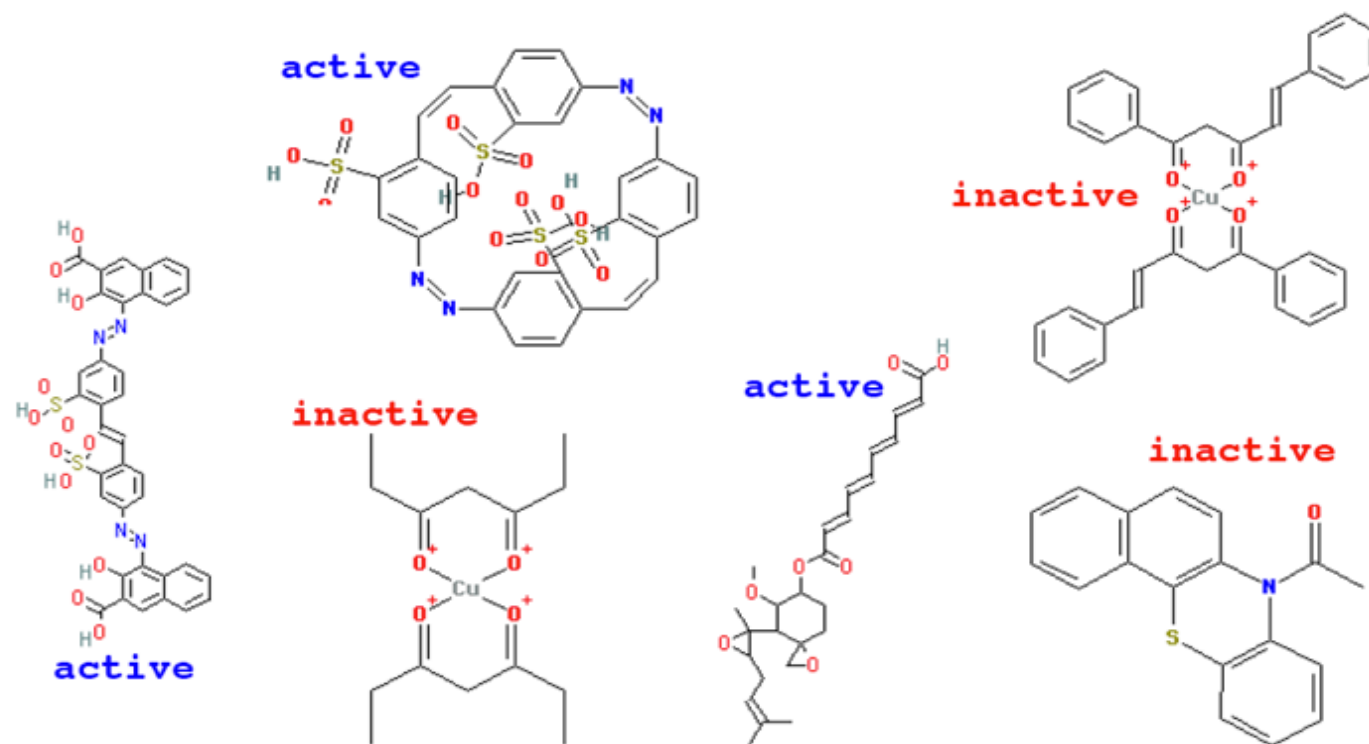
Machine Learning as ...

... Learning a **function** from an **input** space X to an **output** space Y

Cats vs. dogs



Apprentissage prédictif



NCI AIDS screen results (from <http://cactus.nci.nih.gov>).

Apprentissage descriptif

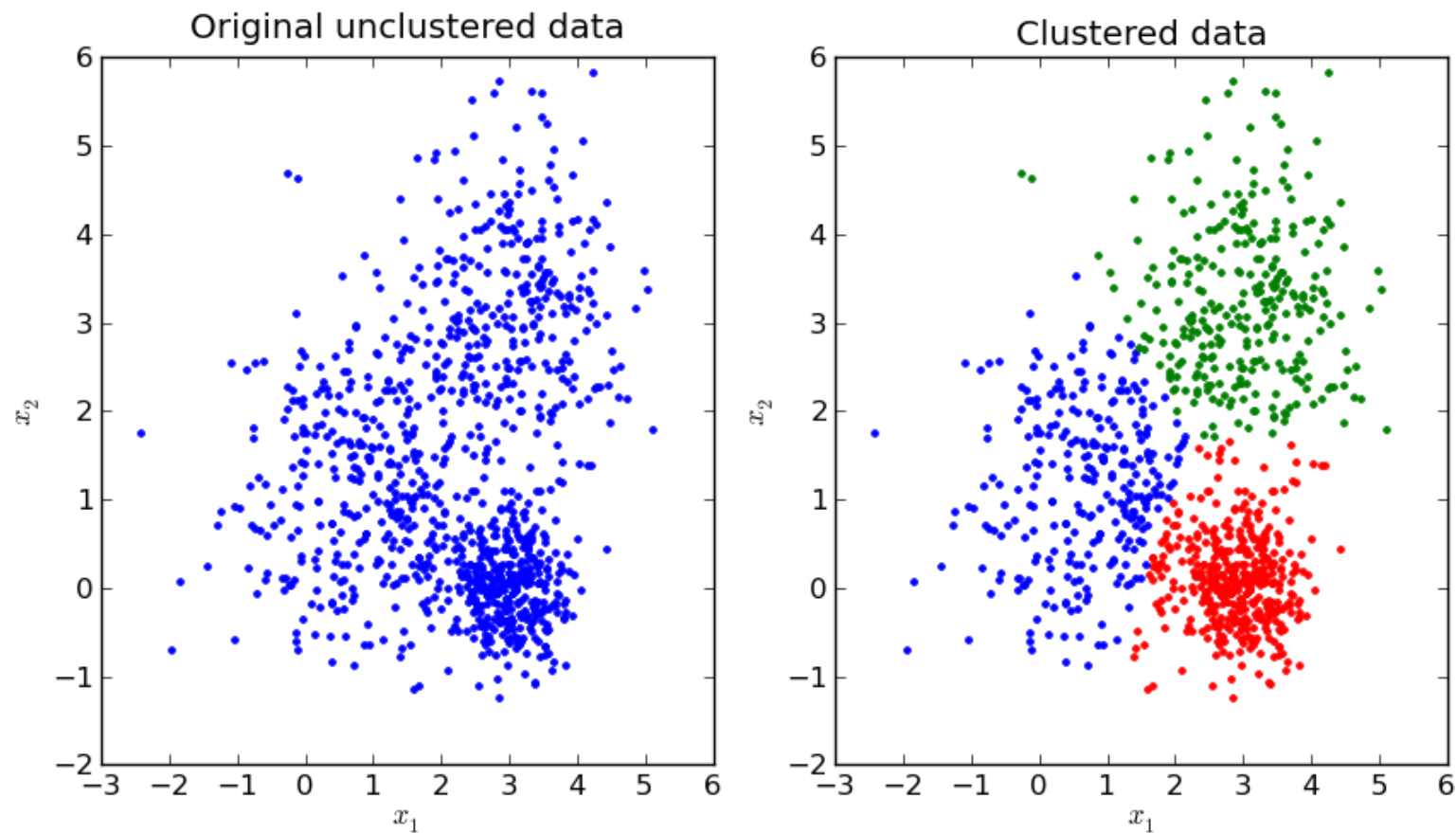
À propos d'un *échantillon d'apprentissage* $s = \{(x_i)\}_{1,m}$
identifier des régularités rendant compte de S

- E.g. under the form of clusters (e.g. *mixture of Gaussians*)
 - CLUSTERING
- E.g. sous la forme de motifs fréquents (fouille de données)

pour résumer, suggérer des régularités, comprendre ...

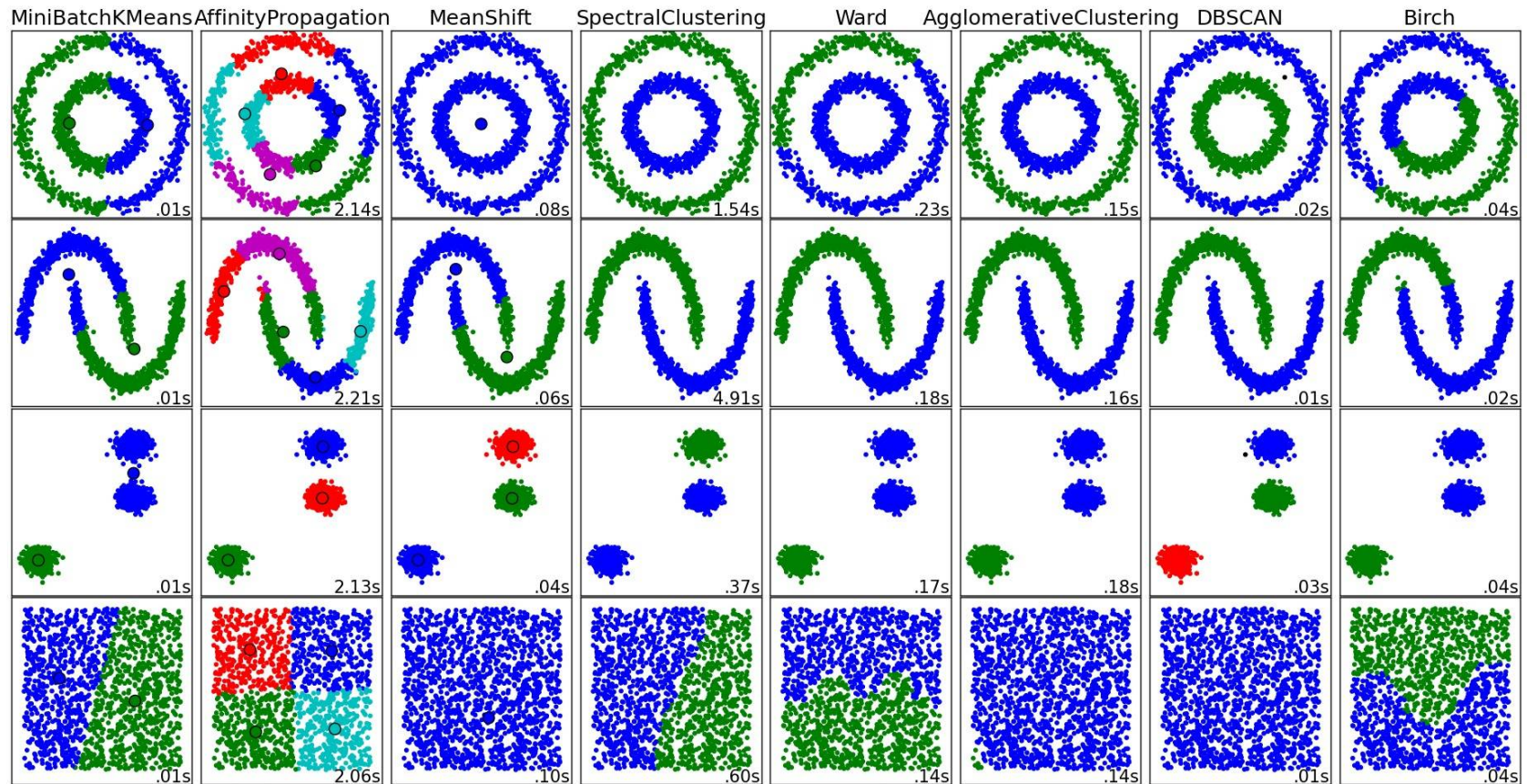
Les grands types d'apprentissage

- Apprentissage « **descriptif** » (non supervisé)



Clustering

Dépend beaucoup des **biais a priori**



Apprentissage prescriptif

- Apprentissage « **prescriptif** » (recherche de *causalités*)

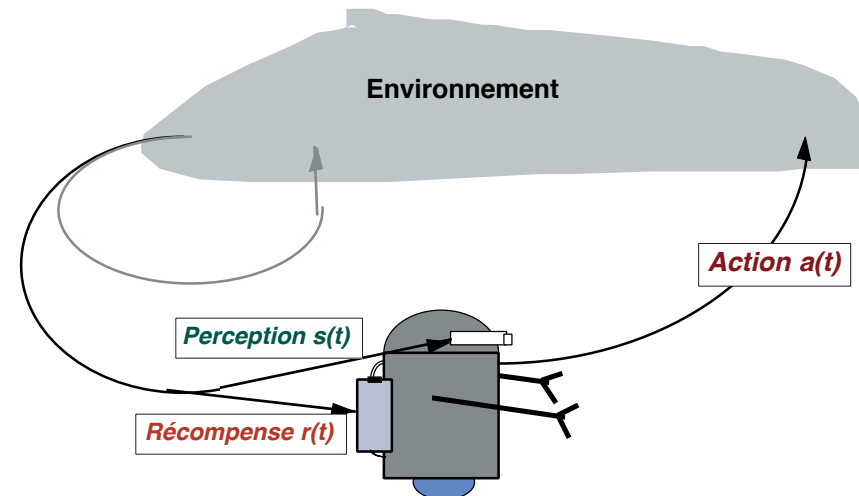
1. J'observe que les gens qui mangent des glaces
sont souvent en maillot de bain

2. Je voudrais vendre davantage de glaces

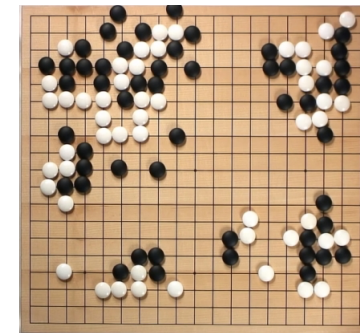
→ Je demande aux gens de se mettre en maillot de bain

Apprentissage « par renforcement »

- comment (ré)agir



1. Piloter un hélicoptère
2. Apprendre à jouer au tennis de table
3. Battre le champion de back-gammon (1992), de Go (2016)
4. Gérer un porte-feuille d'investissements
5. Contrôler une ferme



Plan

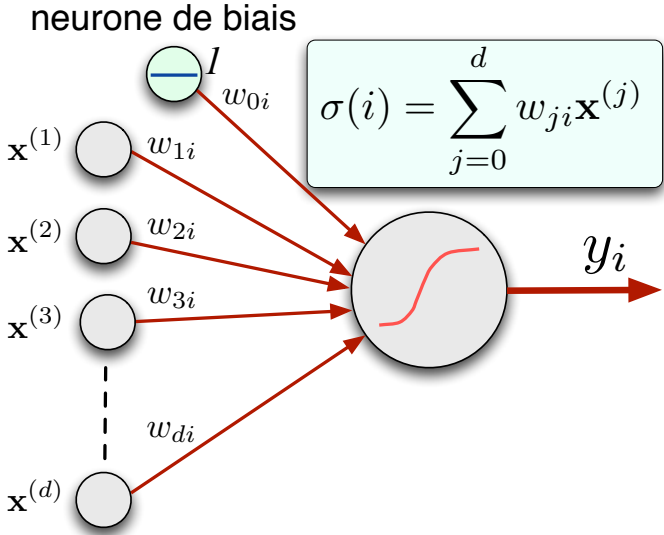
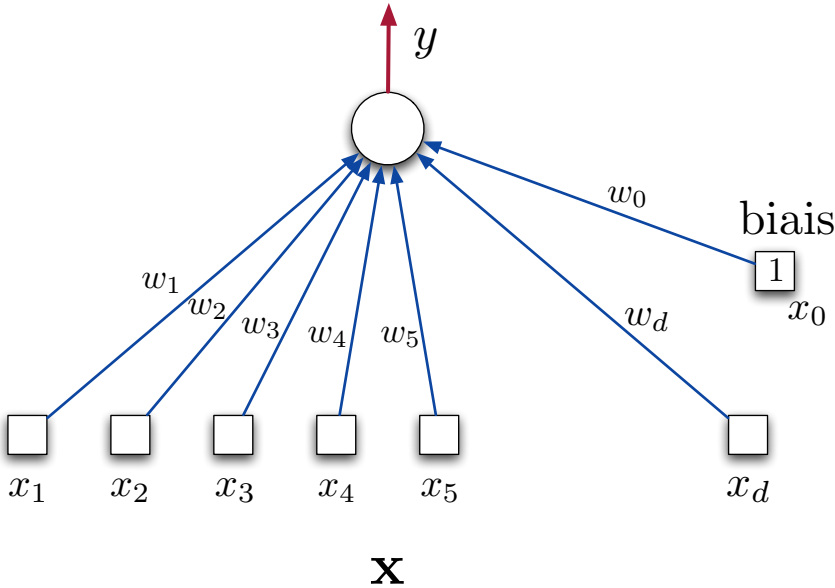
1. Pourquoi toute cette excitation ?
2. Grands types d'apprentissage
3. Apprentissage prédictif par réseaux de neurones
4. Quelles garanties ?
5. Recette pour créer des algorithmes d'apprentissage
6. Les réseaux de neurones profonds
7. Ce que l'on sait faire et les défis à relever

Où l'on va illustrer comment ça marche

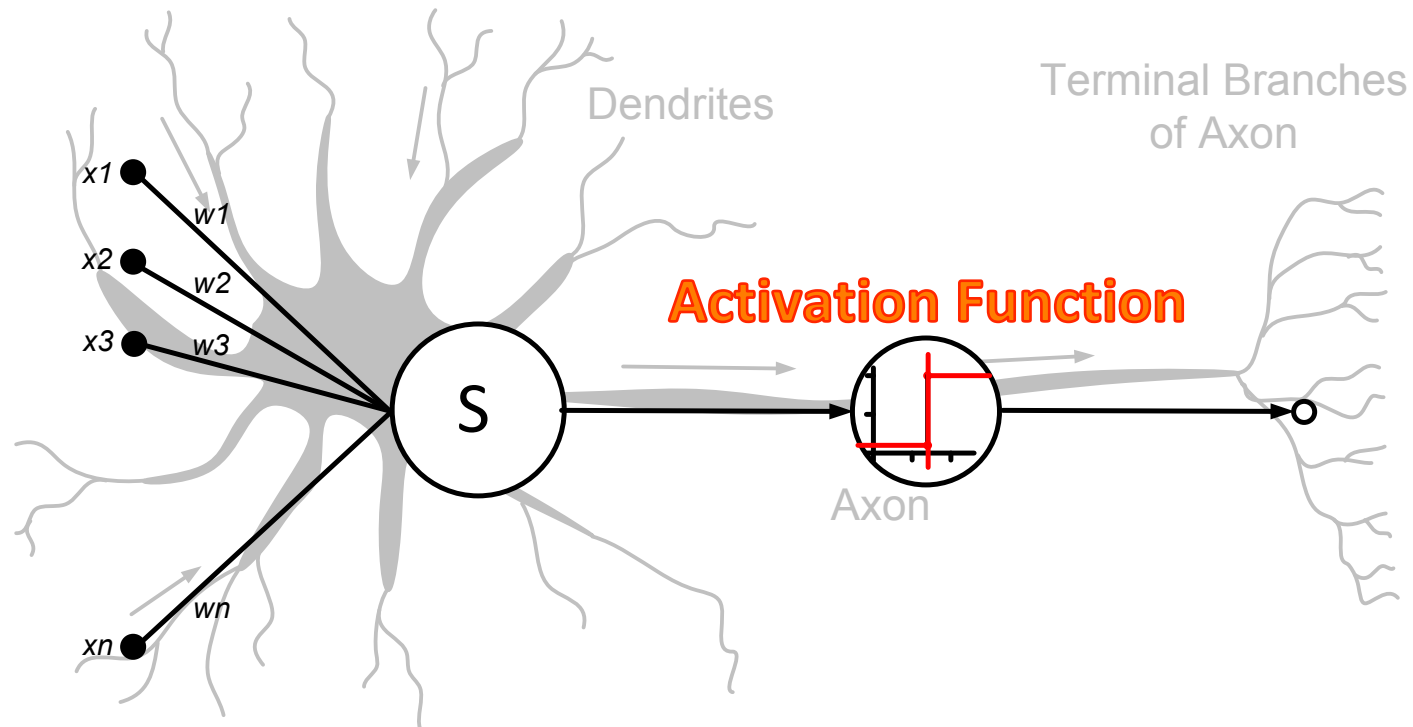
Le cas du perceptron

Le perceptron

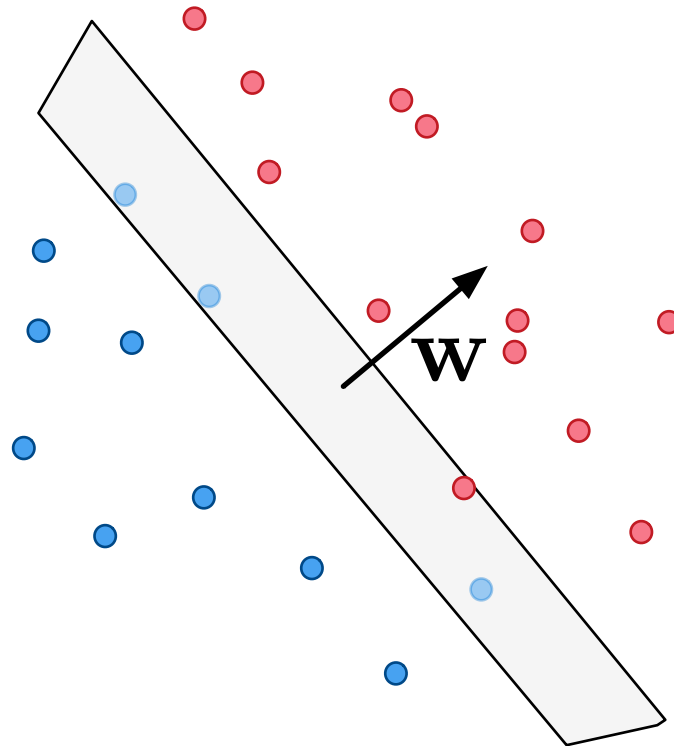
– Rosenblatt (1958-1962)



Artificial Neural Networks (ANN)



Le perceptron : un discriminant linéaire



Le perceptron

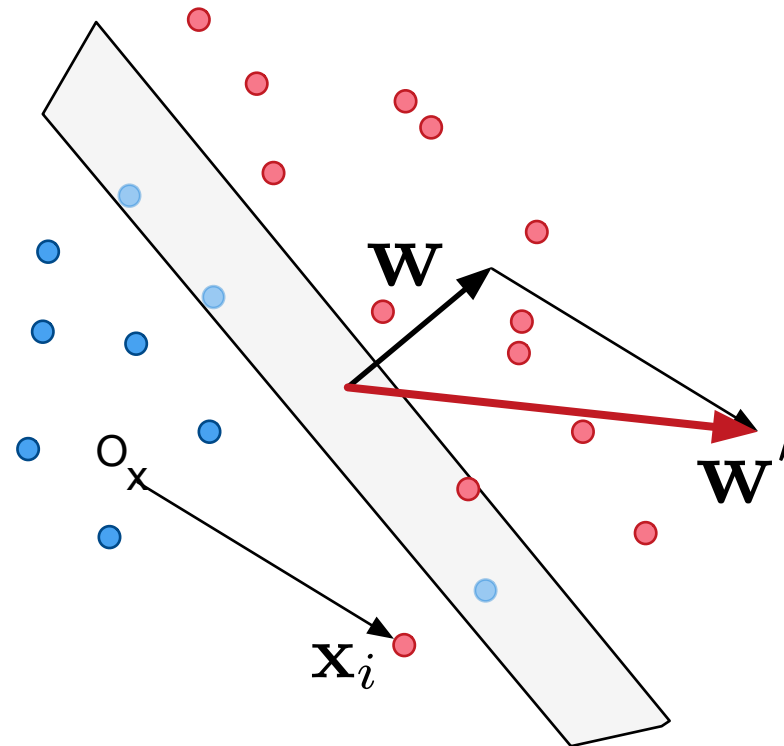
- **Apprentissage des poids w_i**
 - Principe (*règle de Hebb*) : en cas de succès, ajouter à chaque connexion quelque chose de proportionnel à l'entrée et à la sortie

Règle du perceptron : **apprendre seulement en cas d'échec**

Algorithme 1 : Algorithme d'apprentissage du perceptron

```
tant que non convergence faire  
  | si la forme d'entrée est correctement classée alors  
  |   | ne rien faire  
  | sinon  
  |   |  $w(t + 1) = w(t) \pm \eta x_i y_i$   
  | fin  
  | Passer à la forme d'apprentissage suivante  
fin
```

The perceptron learning algorithm: intuition



$$w' = w + \eta y_i x_i$$

Des propriétés remarquables !!

- **Convergence** en un **nombre fini d'étapes**
 - Indépendamment du **nombre** d'exemples
 - Indépendamment de la **distribution** des exemples
 - (quasi)indépendamment de la **dimension** de l'espace d'entrée



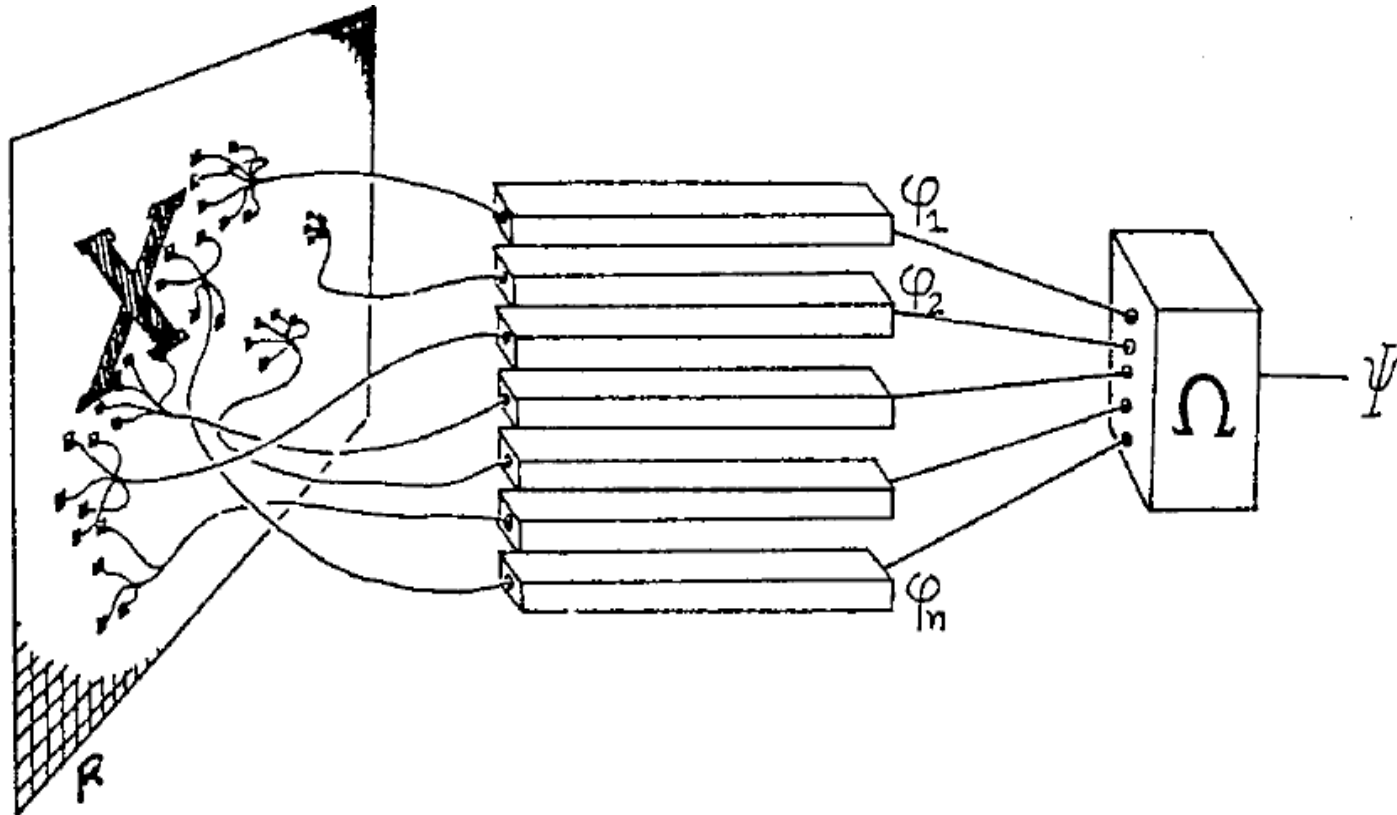
Si il existe au moins *une séparatrice linéaire des exemples*

Garantie de généralisation ??

- Théorèmes sur la performance
par rapport à l'échantillon d'apprentissage
- Mais qu'en est-il pour des **exemples à venir** ?

Le Perceptron

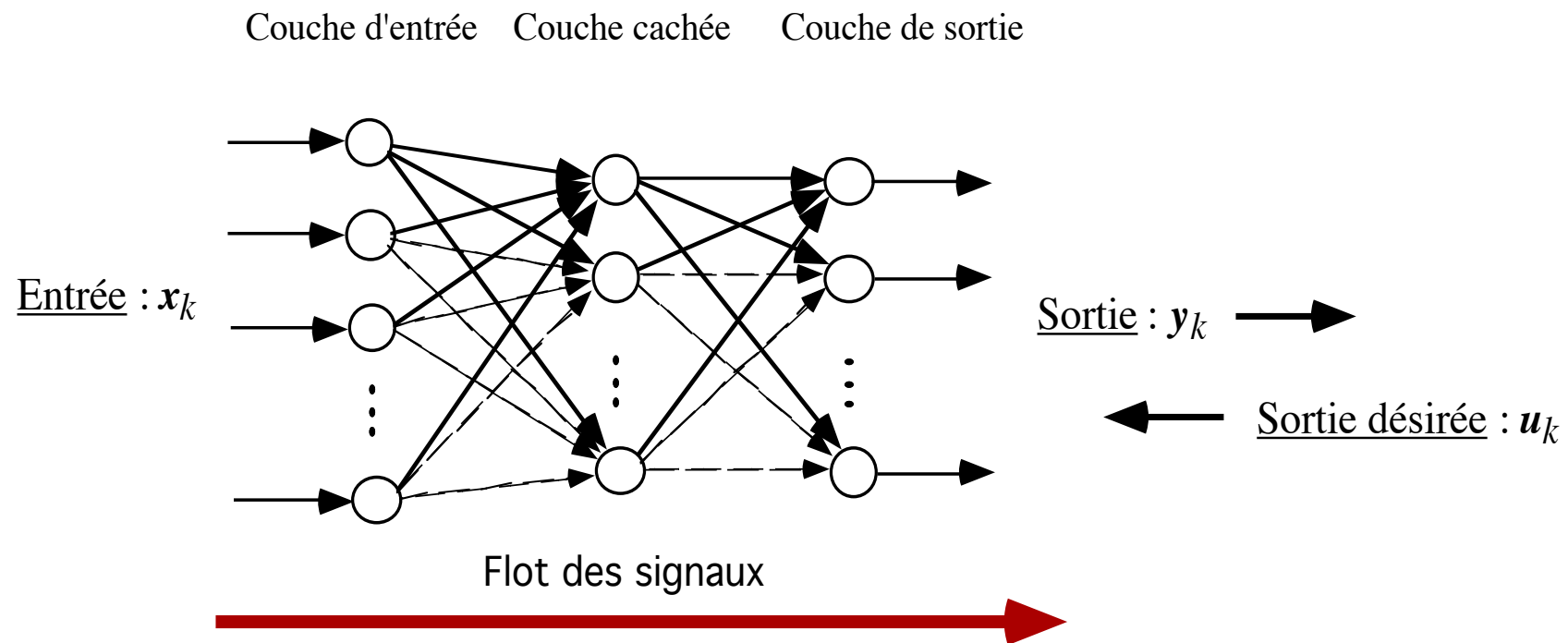
- Rosenblatt (1958-1962)



Les perceptrons multi-couches

Le perceptron multi-couche

- Topologie typique



Le Perceptron Multi-Couches : propagation

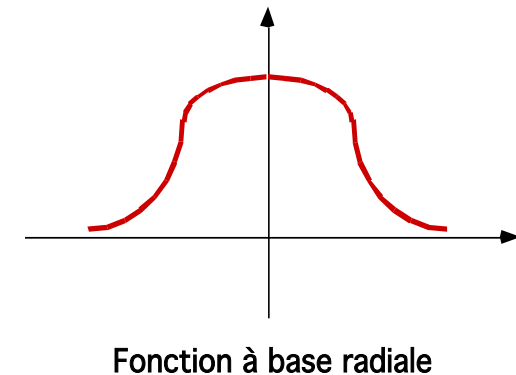
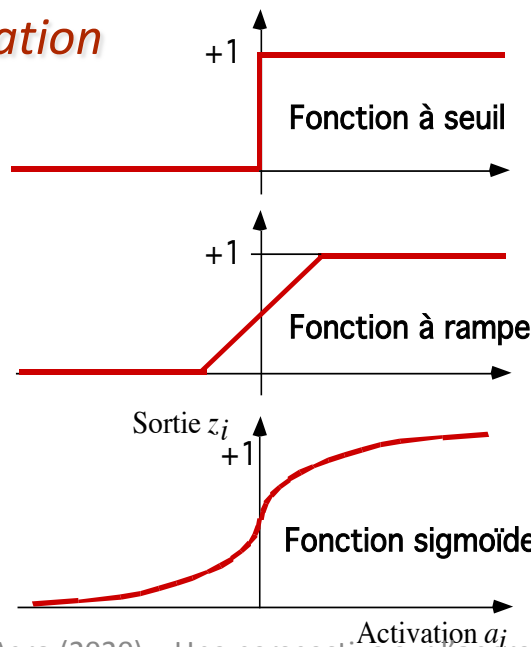
- Pour chaque neurone :


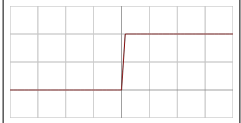
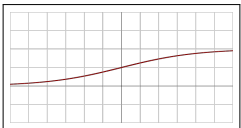
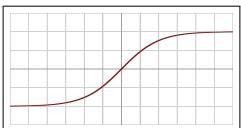
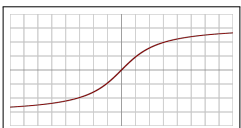
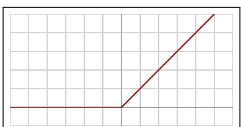
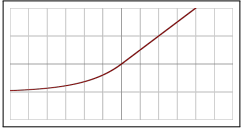
$$y_l = g\left(\sum_{j=0,d} w_{jk} \phi_j\right) = g(a_k)$$

- w_{jk} : *poids* de la connexion de la cellule j à la cellule k
- a_k : *activation* de la cellule k
- g : *fonction d'activation*

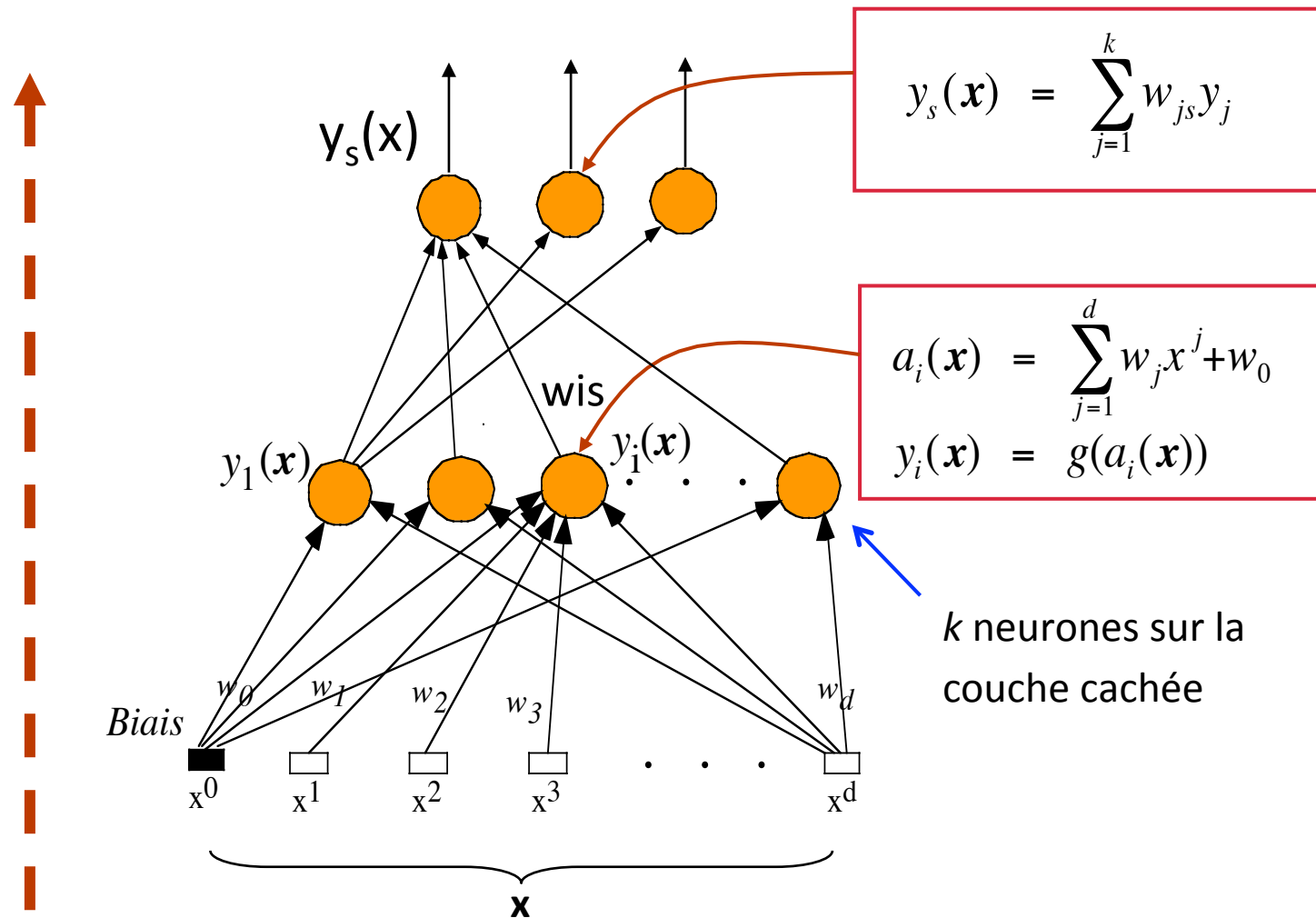
$$g(a) = \frac{1}{1 + e^{-a}}$$

$$g'(a) = g(a)(1-g(a))$$



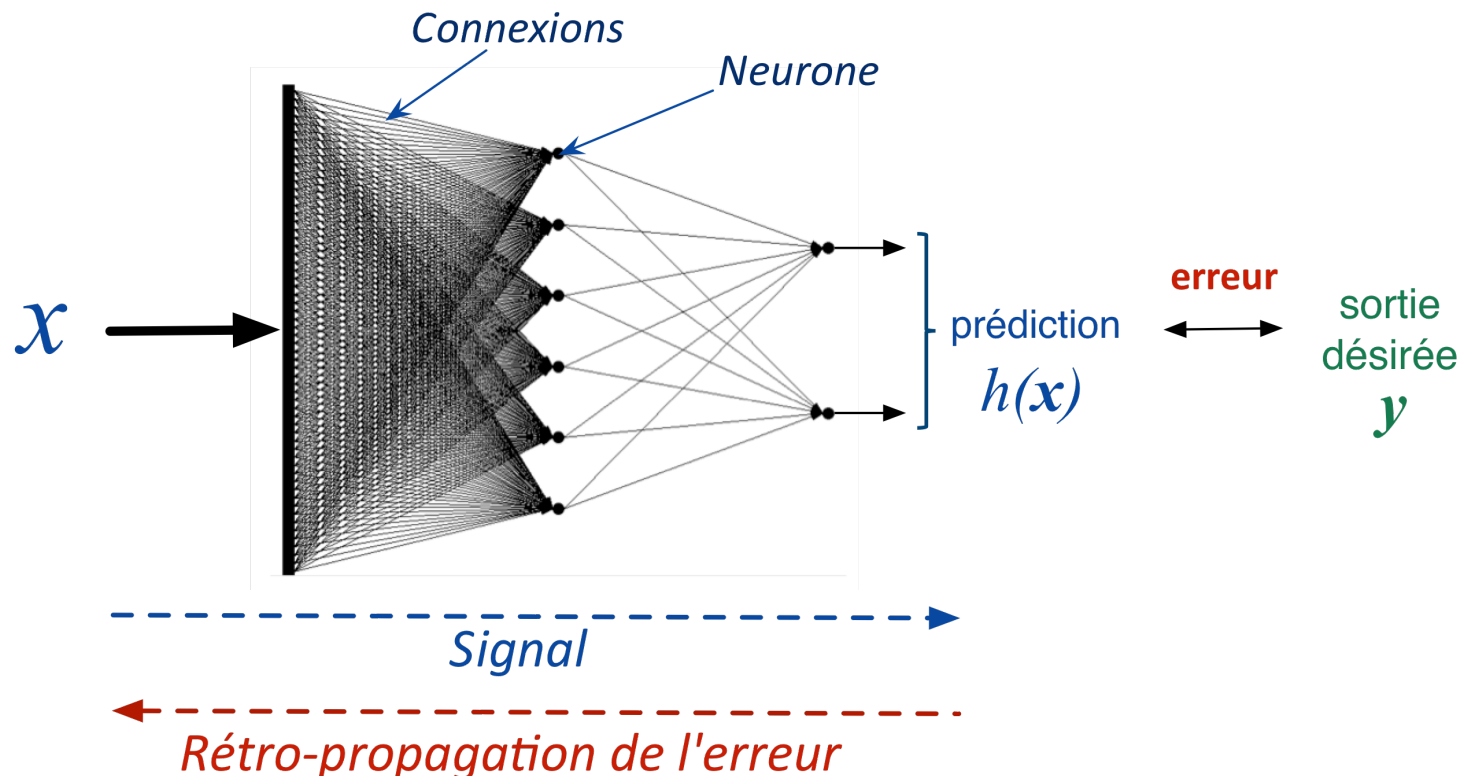
Nom	Graphe	Équation	Dérivée
Rampe		$f(x) = x$	$f'(x) = 1$
Heaviside		$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{si } x \neq 0 \\ ? & \text{si } x = 0 \end{cases}$
Logistique ou sigmoïde		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tangente hyperbolique		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f^2(x)$
Arc Tangente		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Unité ReLU		$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$
Unité Exponentielle Linéaire		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$

Le PMC : passes avant et arrière (résumé)



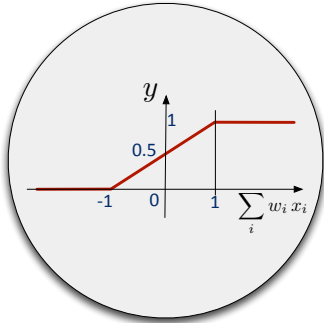
Learning with Multi-Layer Perceptrons

- Questions:
 - How to learn the **parameters** (weights of the connections) ?
 - How to set the **architecture** of the network?



Compute the weights such that ...

Calcul de la fonction XOR



Entrée x_1	Entrée x_2	Sortie y
0	0	0
0	1	1
1	0	1
1	1	0

$W_1 =$

$W_2 =$

$W_3 =$

$W_4 =$

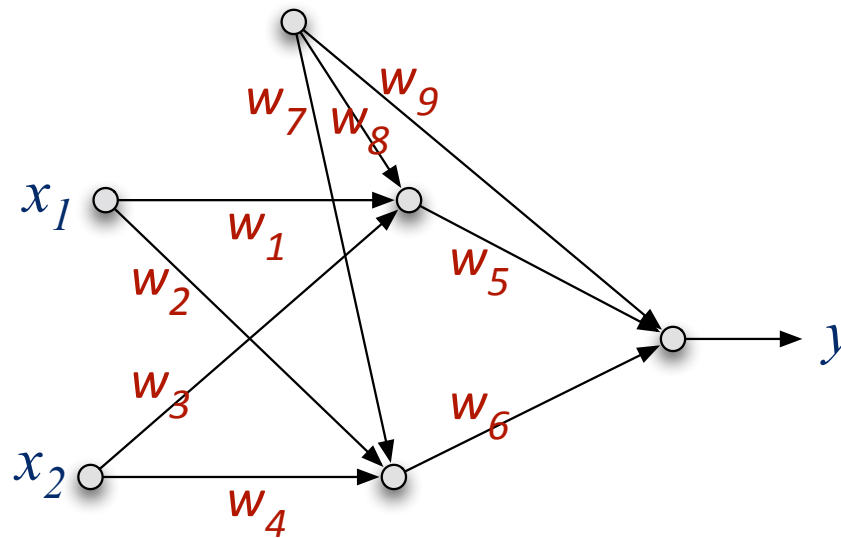
$W_5 =$

$W_6 =$

$W_7 =$

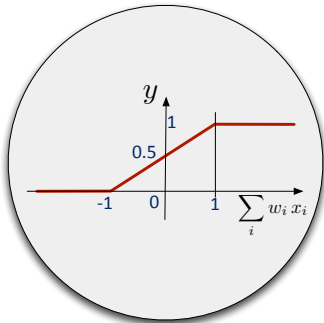
$W_8 =$

$W_9 =$



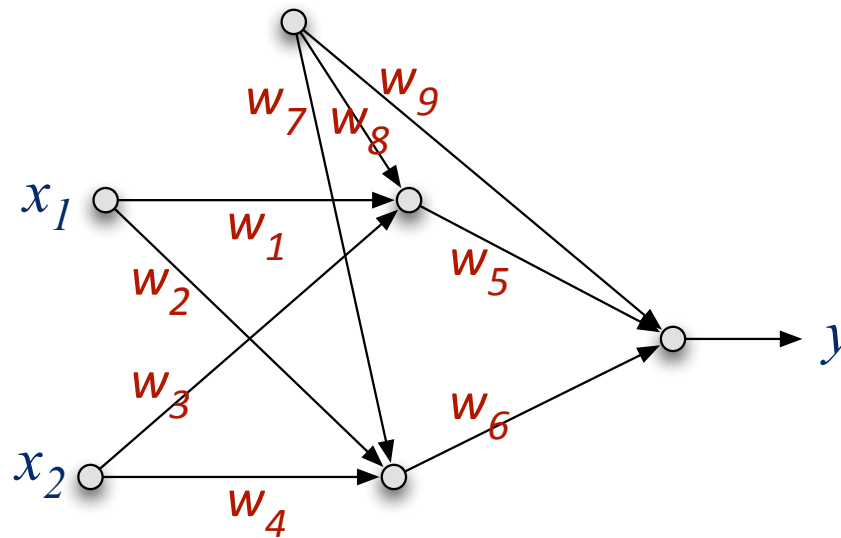
Compute the weights such that ...

Calcul de la fonction XOR

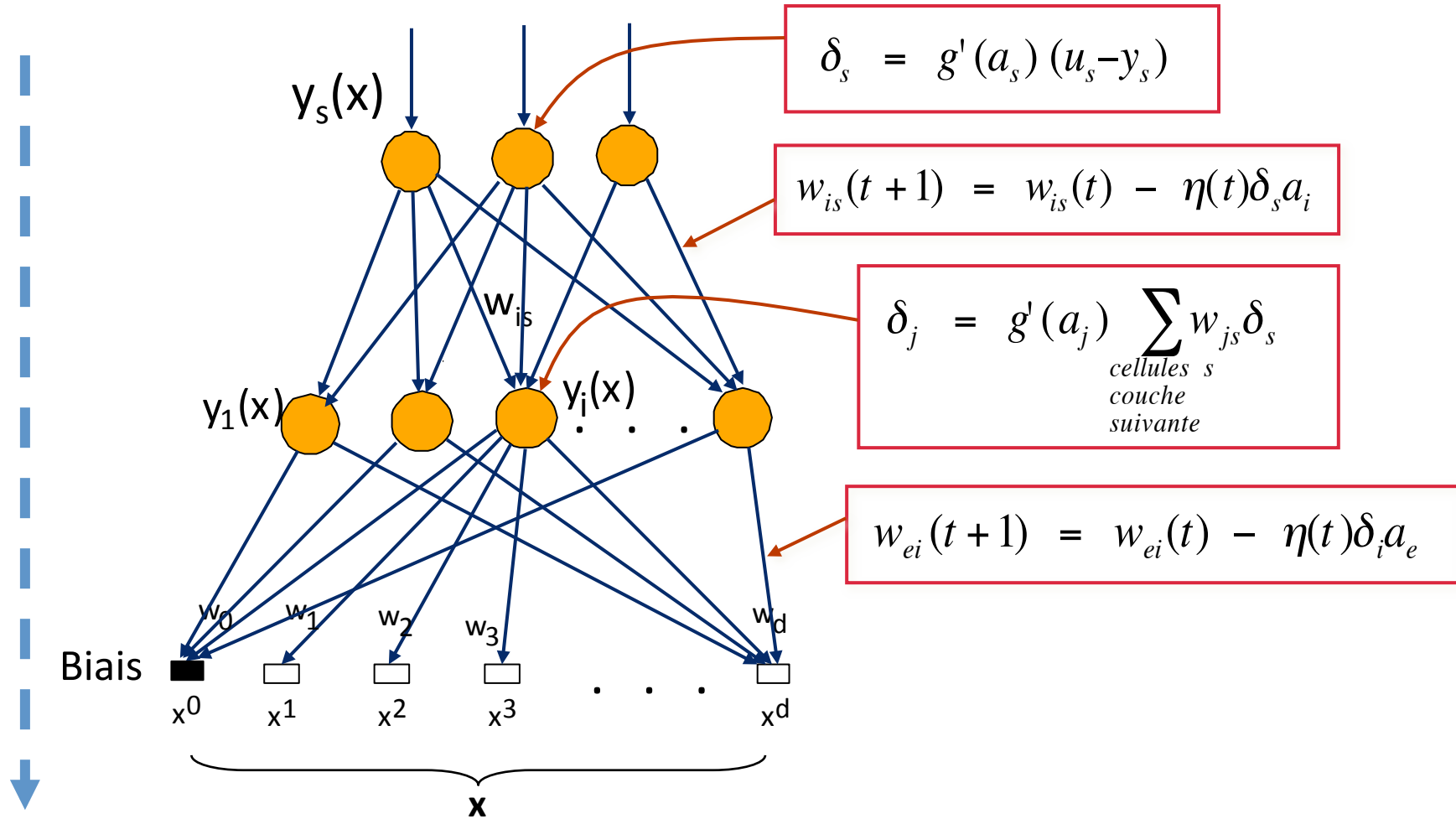


Entrée x_1	Entrée x_2	Sortie y
0	0	0
0	1	1
1	0	1
1	1	0

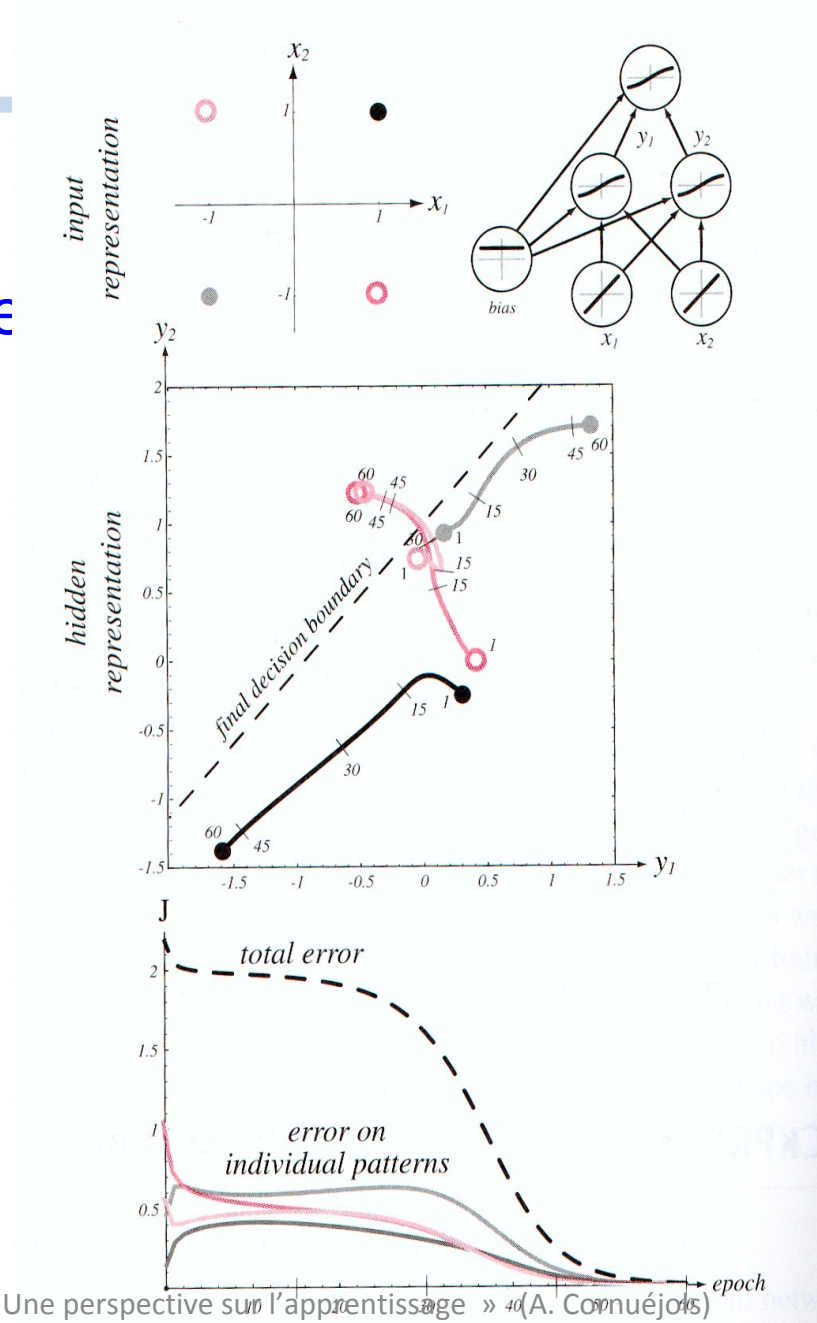
- $W_1 = 1$
- $W_2 = -1$
- $W_3 = -1$
- $W_4 = 1$
- $W_5 = 4$
- $W_6 = 4$
- $W_7 = -1$
- $W_8 = -1$
- $W_9 = -1$



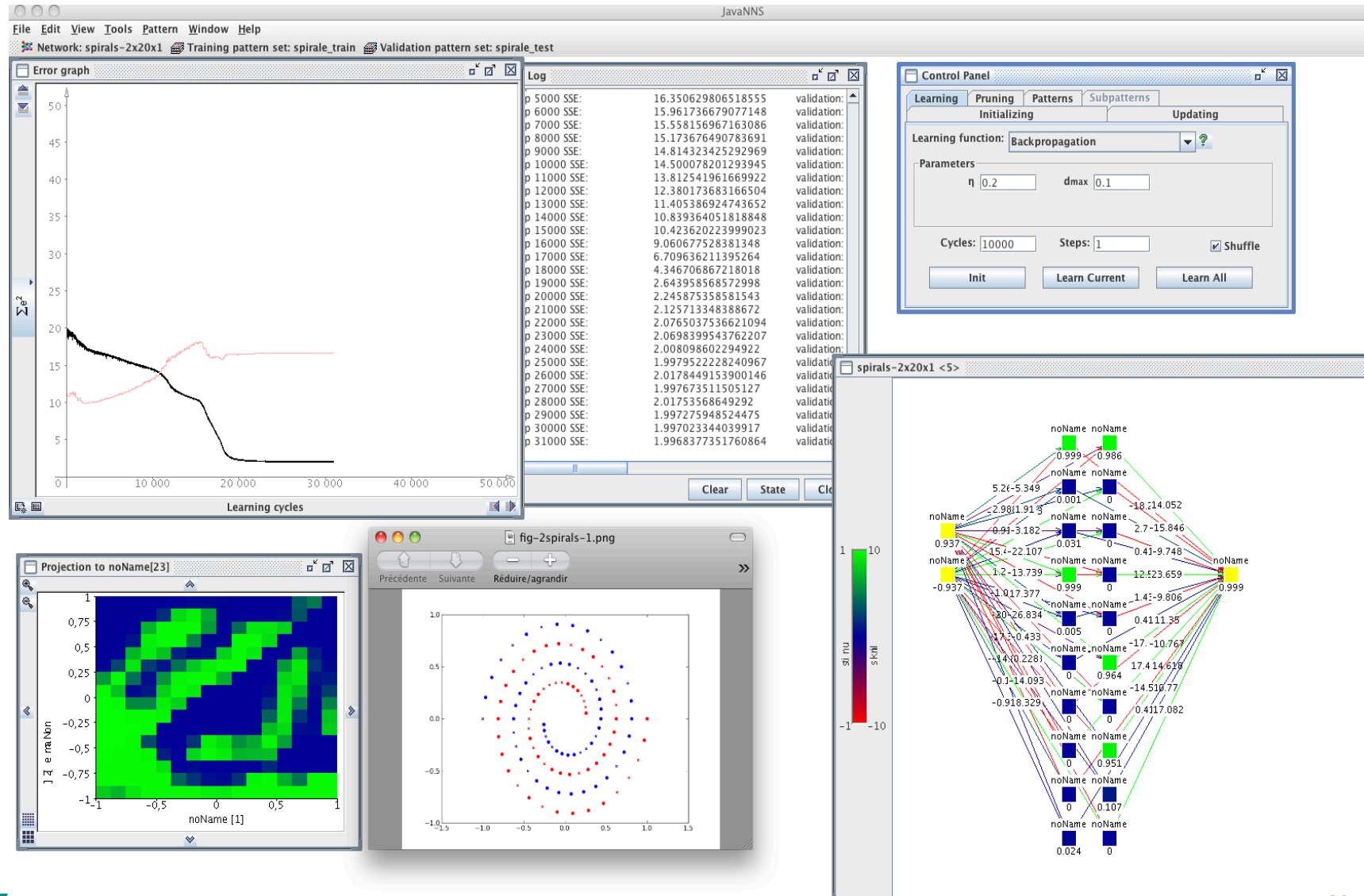
Le PMC : passes avant et arrière (résumé)



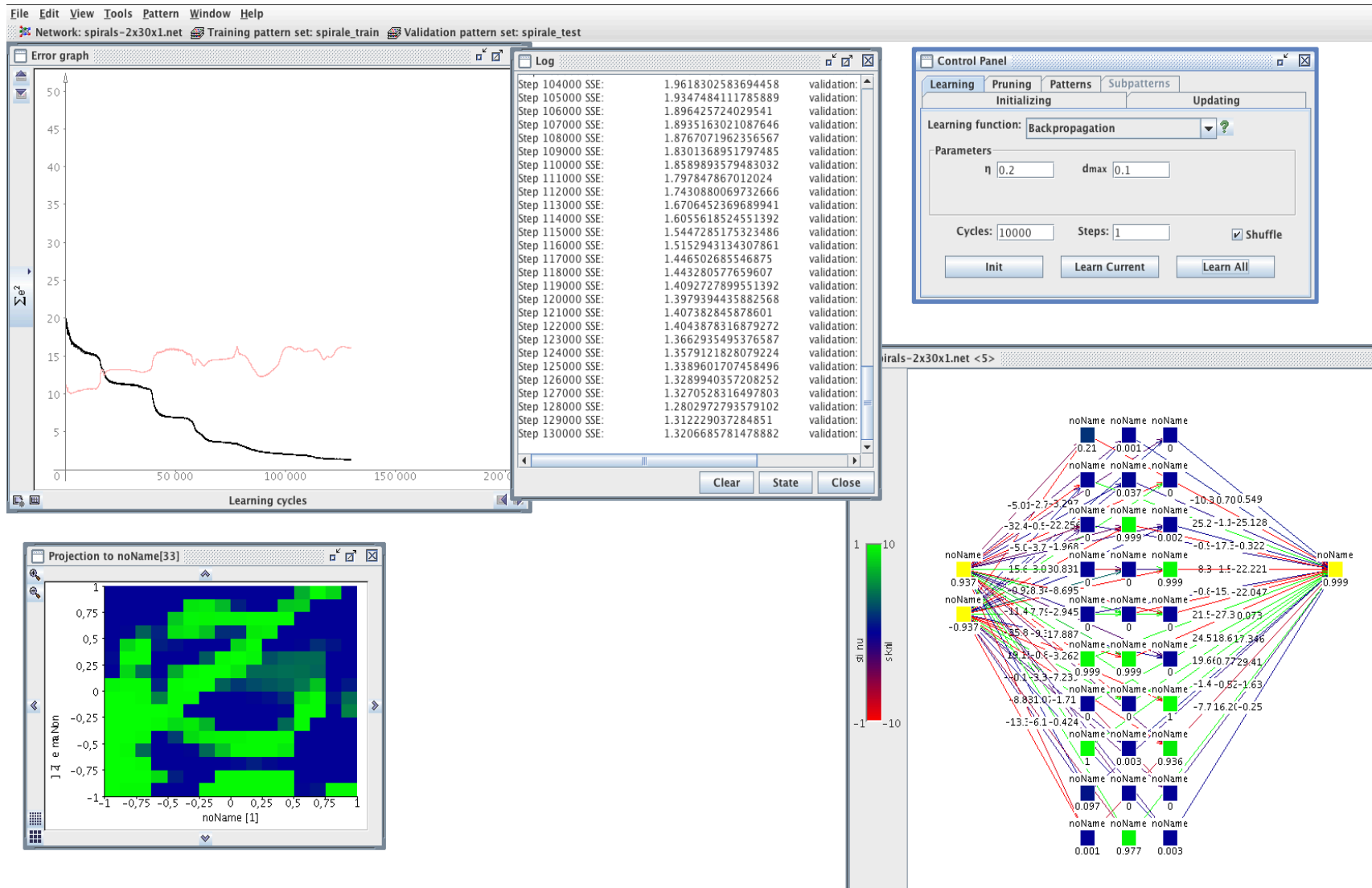
Rôle de la couche cachée



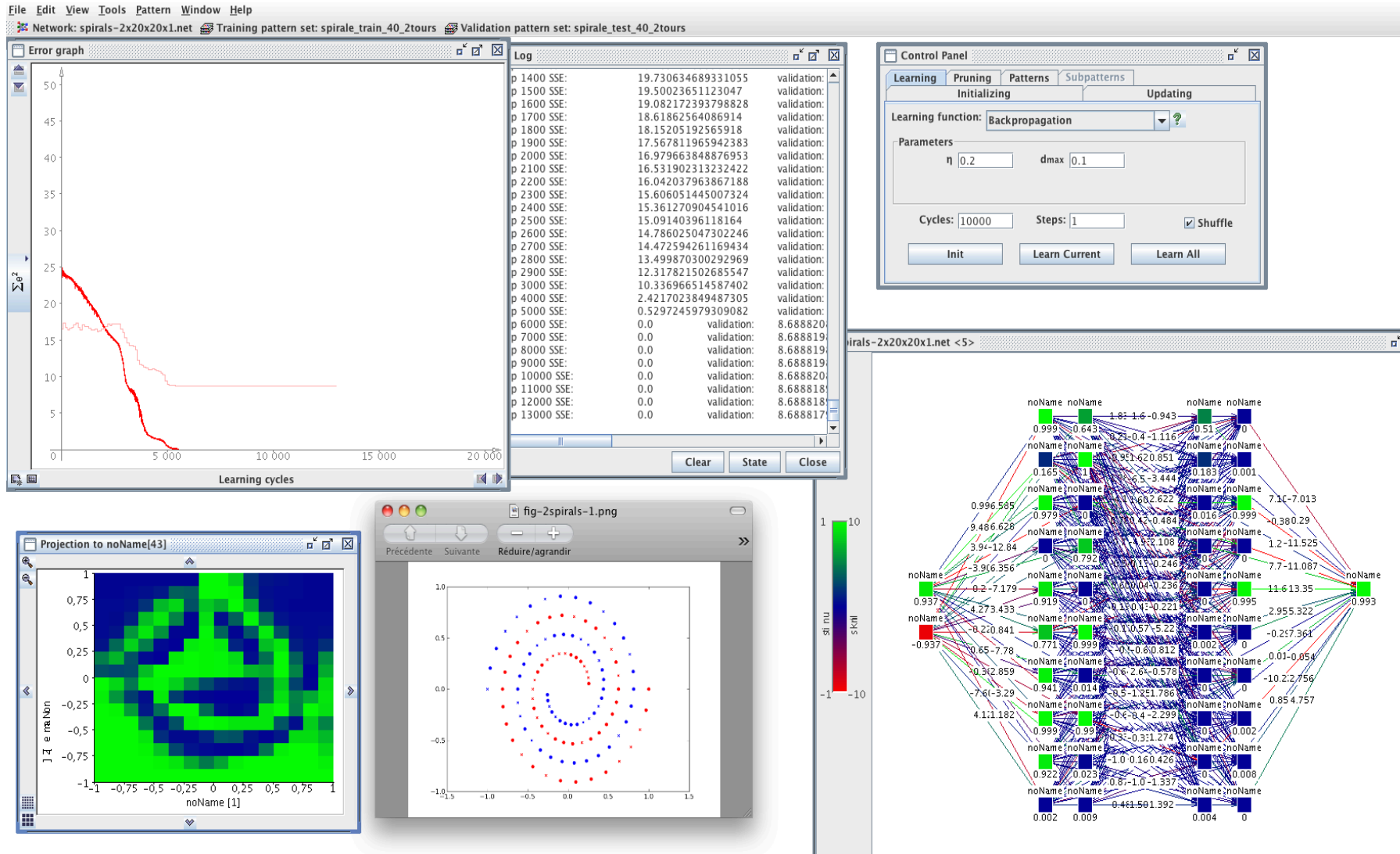
Illustration



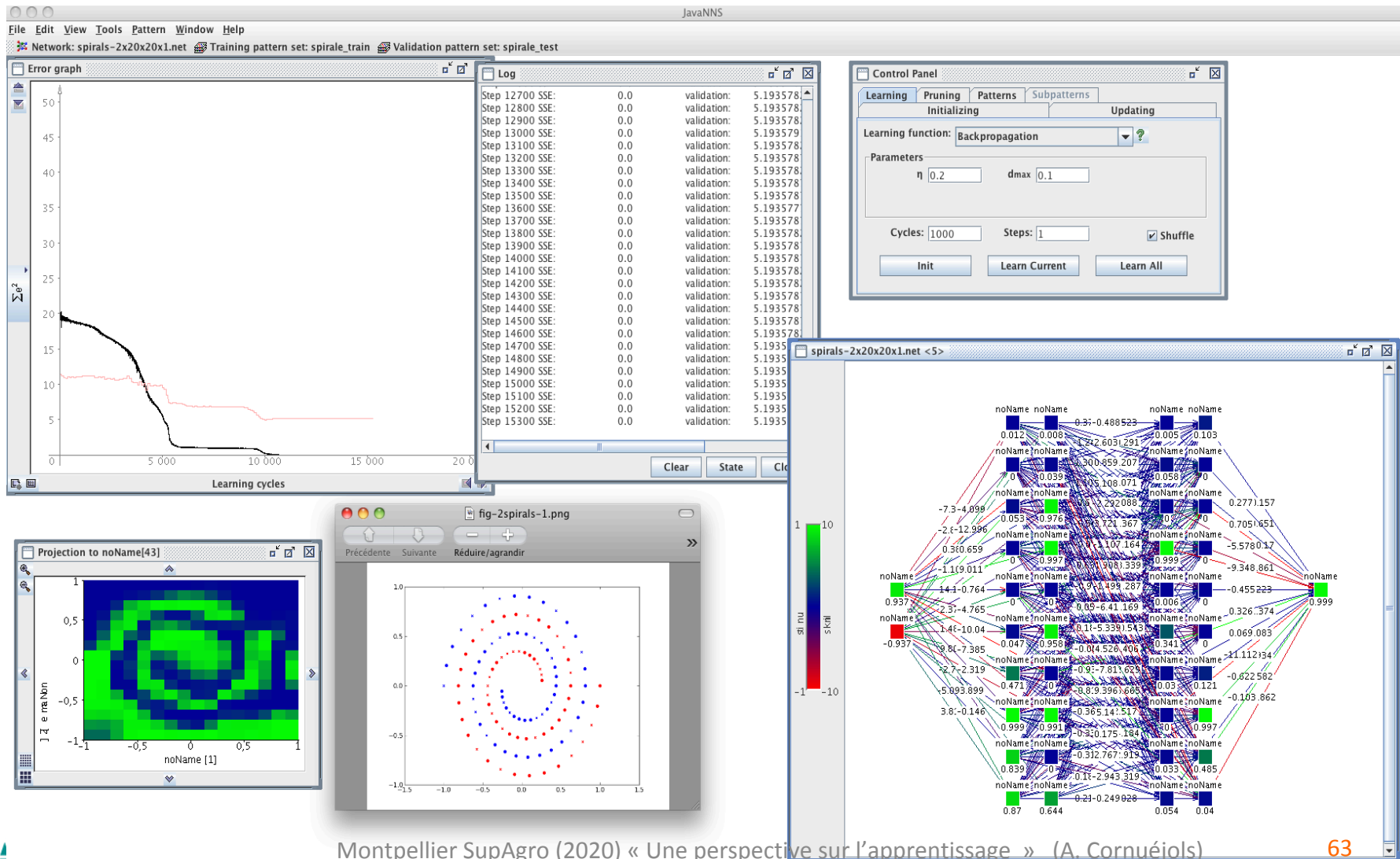
Illustration



Illustration

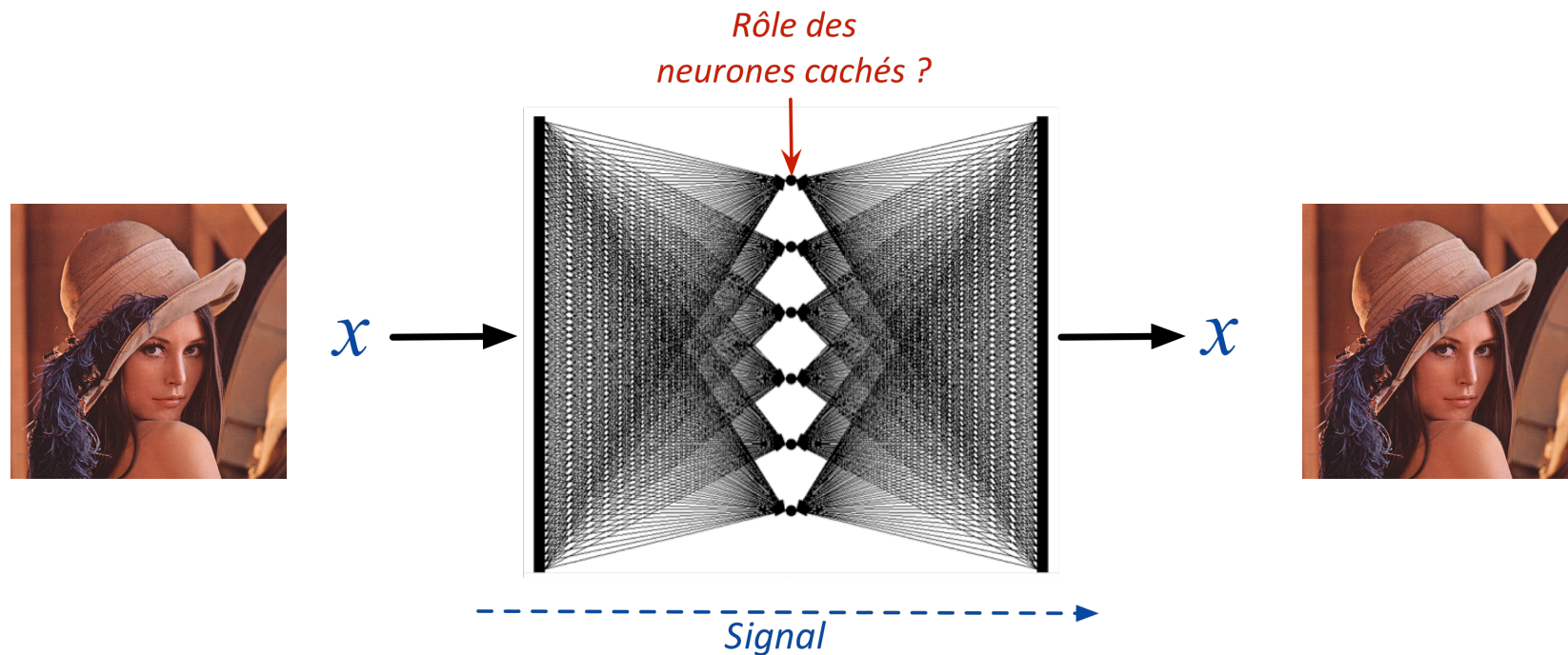


Illustration



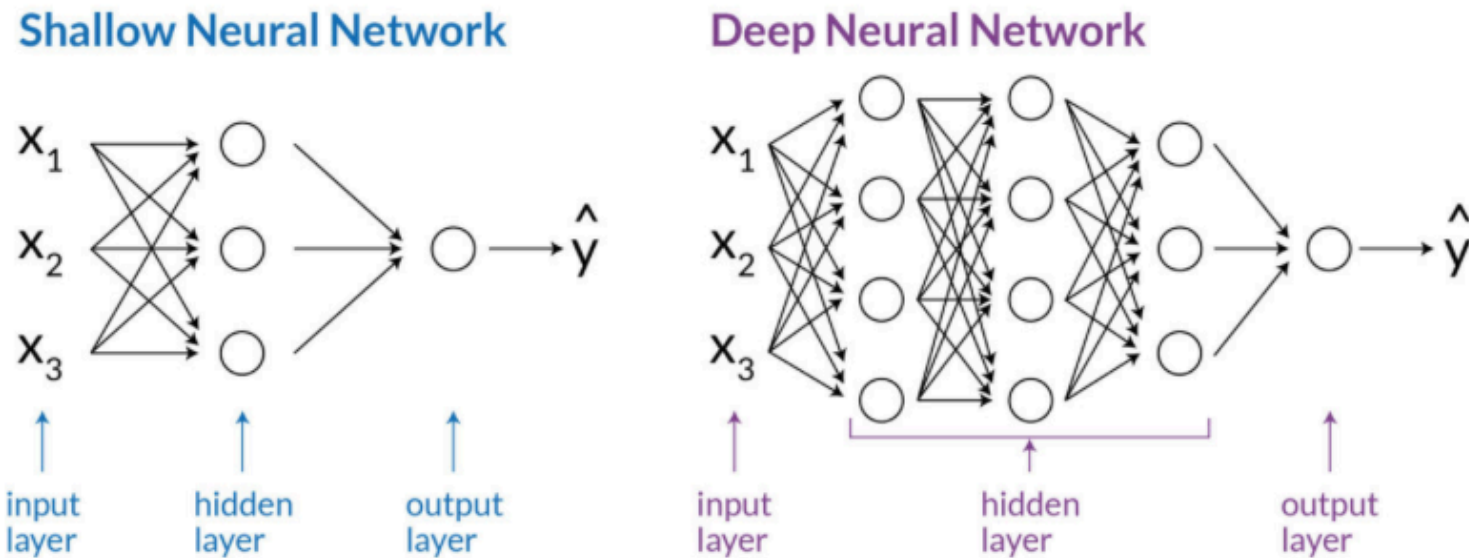
Change of representation: the role of hidden layer(s)

- Which new representation (latent variables)?
- How to choose the architecture?



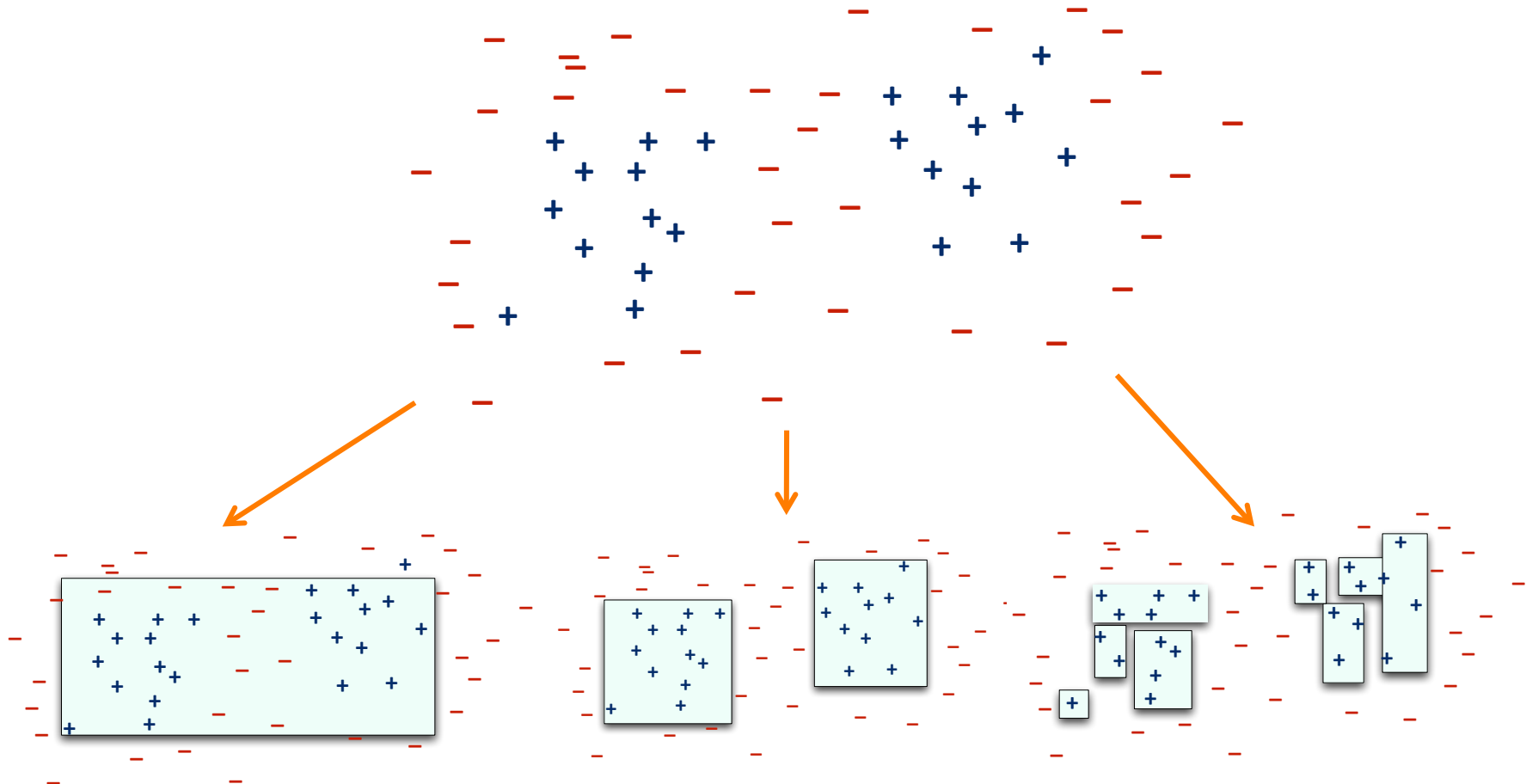
Comment choisir l'architecture du RN ?

- Contrôle la performance en généralisation



Shallow and Deep Neural Networks.

Illustration

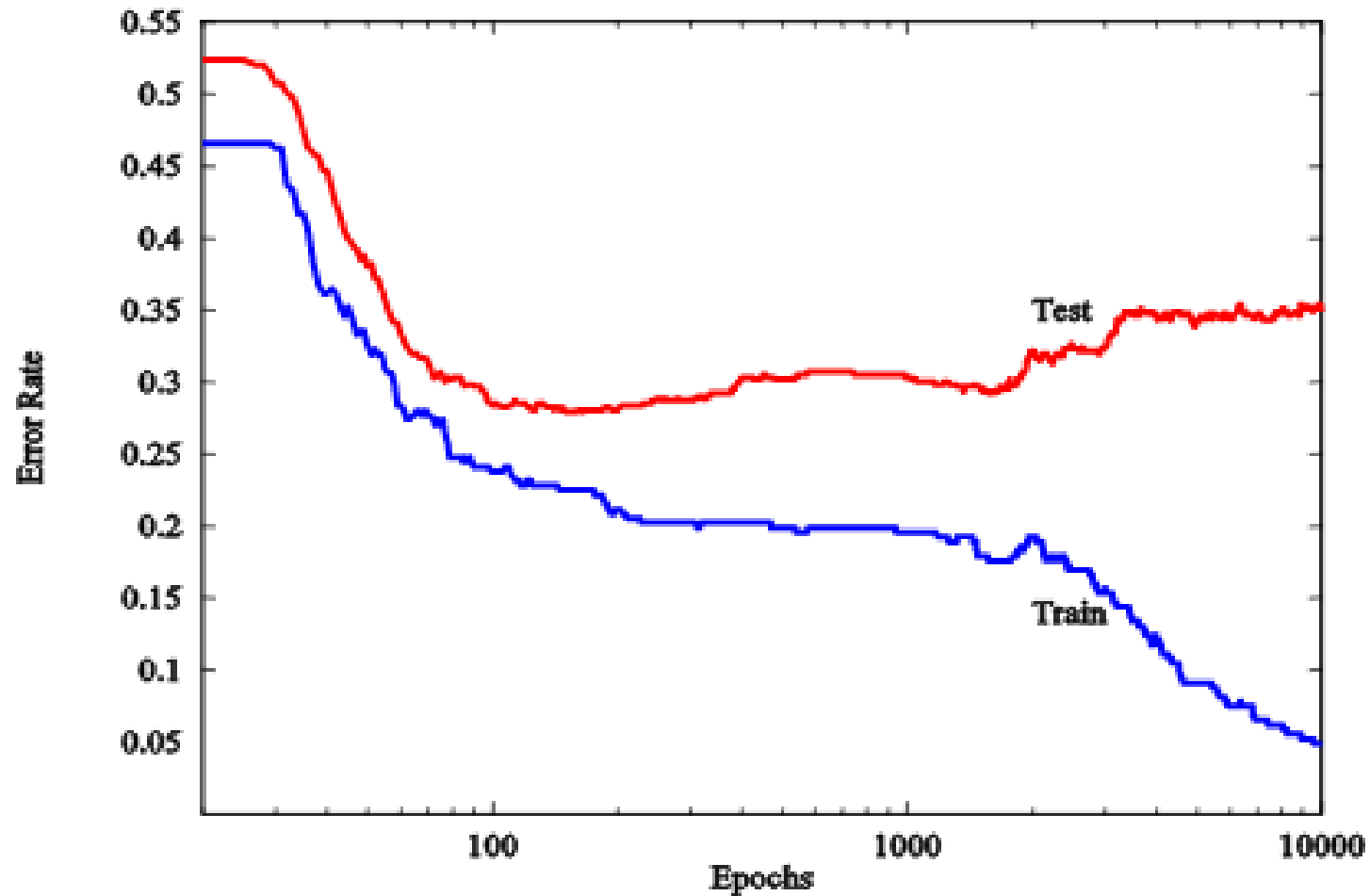


Sous-apprentissage

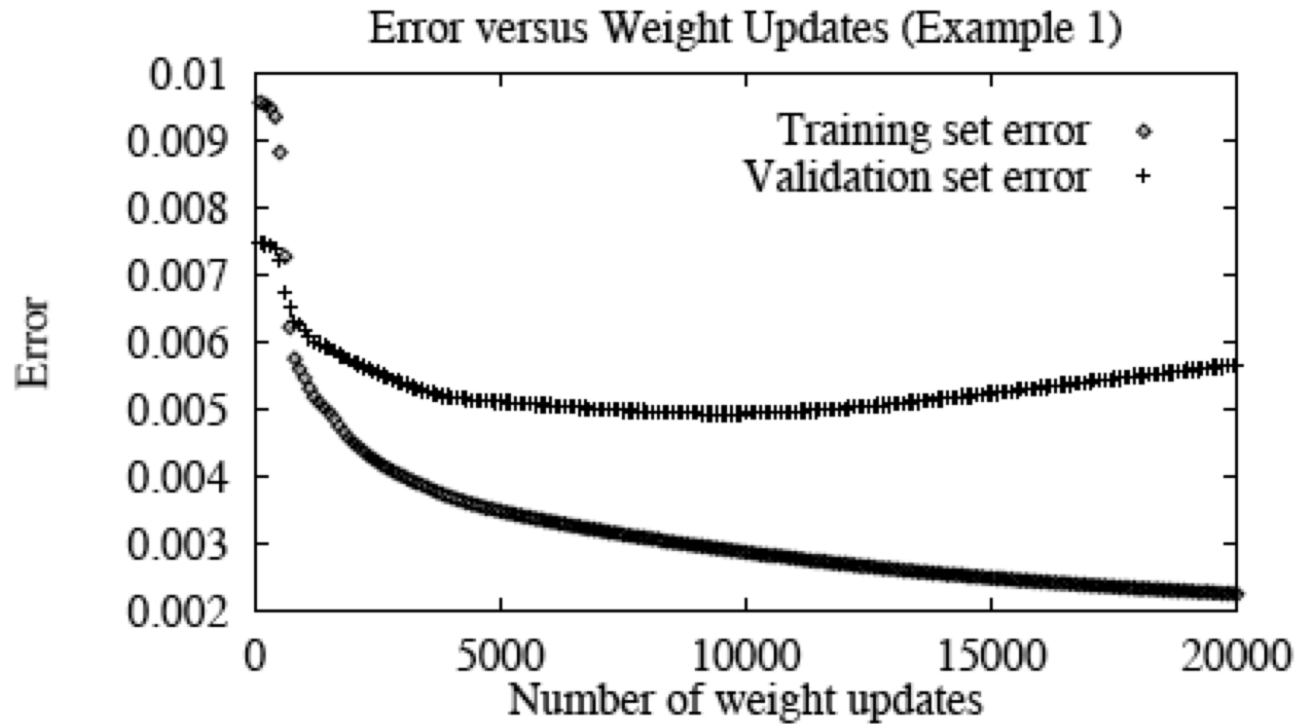
Modèle « correct »

Sur-apprentissage

Sur-apprentissage



Sur-apprentissage (RN)



- Courbes pour 1 000 exemples
- *Quelles courbes si on avait 2 000 exemples ?*

Les trois ingrédients de l'apprentissage artificiel

Trois ingrédients

1. Le choix de l'**espace des hypothèses** H
 - Généralement $H \neq F$
2. Le **critère inductif**
 - Comment évaluer chaque hypothèse en fonction de S
3. La méthode d'**exploration** de H
 - Comment trouver une bonne (optimale ?) hypothèse

Plan

1. Pourquoi toute cette excitation ?
2. Grands types d'apprentissage
3. Apprentissage prédictif par réseaux de neurones
4. Quelles garanties ?
5. Recette pour créer des algorithmes d'apprentissage
6. Les réseaux de neurones profonds
7. Ce que l'on sait faire et les défis à relever

Il est temps de se poser la question

*Mot d'ordre : **pas de pensée magique !***

Comment **fonder** l'induction ?

Illustration : apprendre à classer des exemples

- Comment faire ?

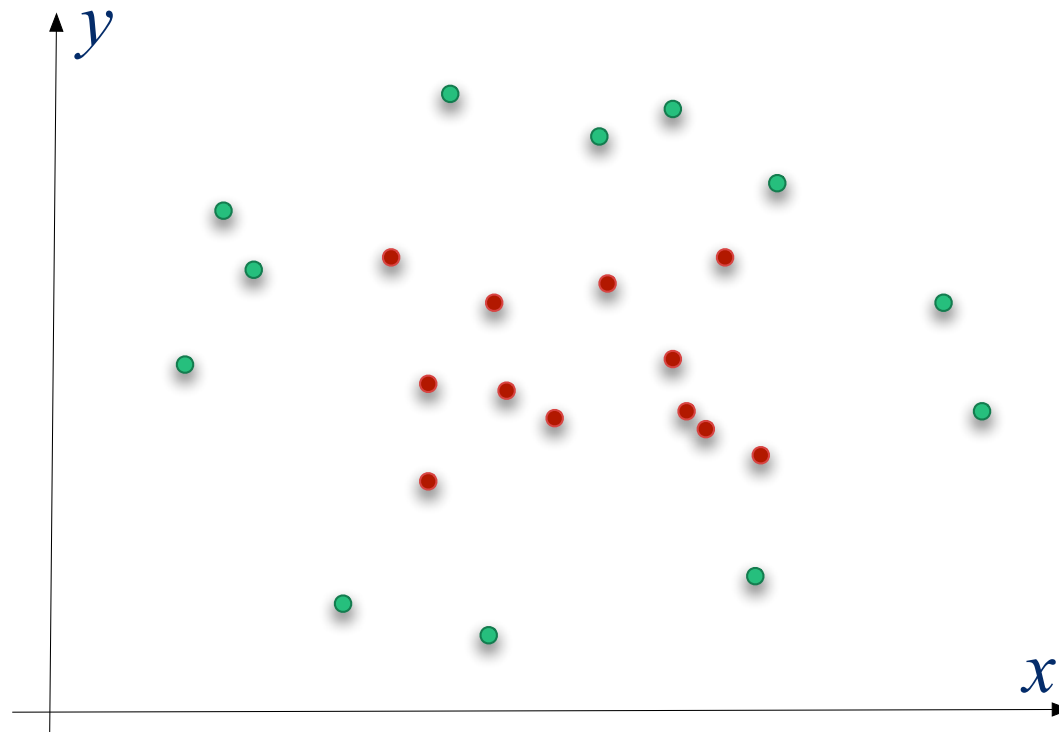


Illustration : apprendre à classer des exemples

- Comment faire ?

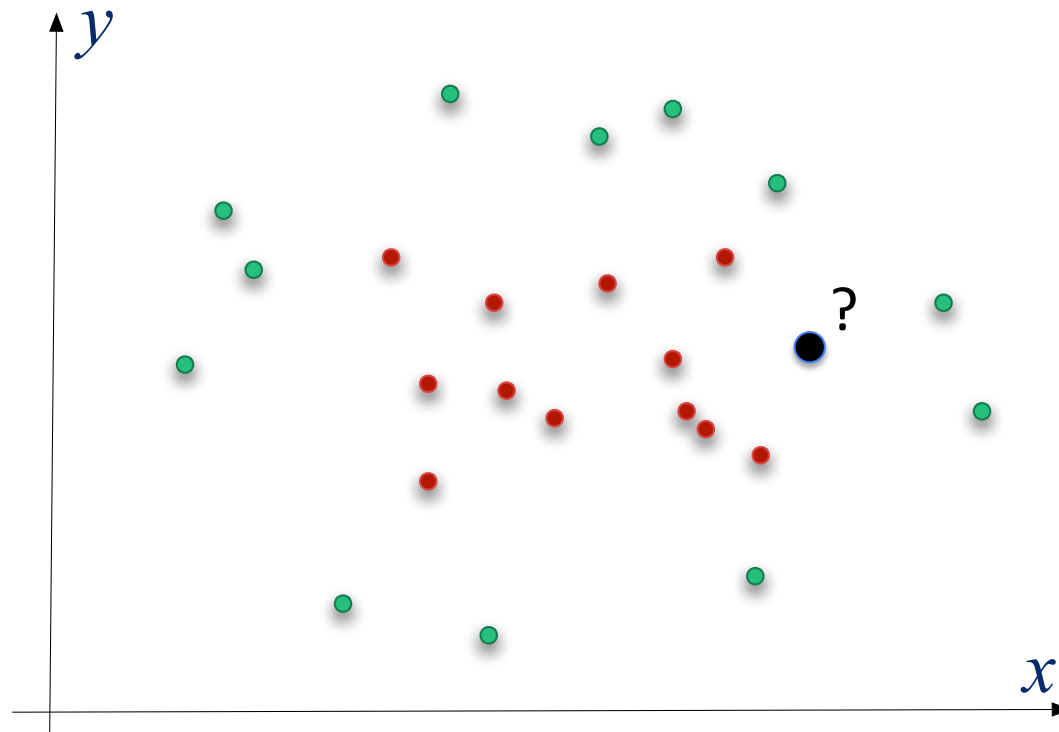
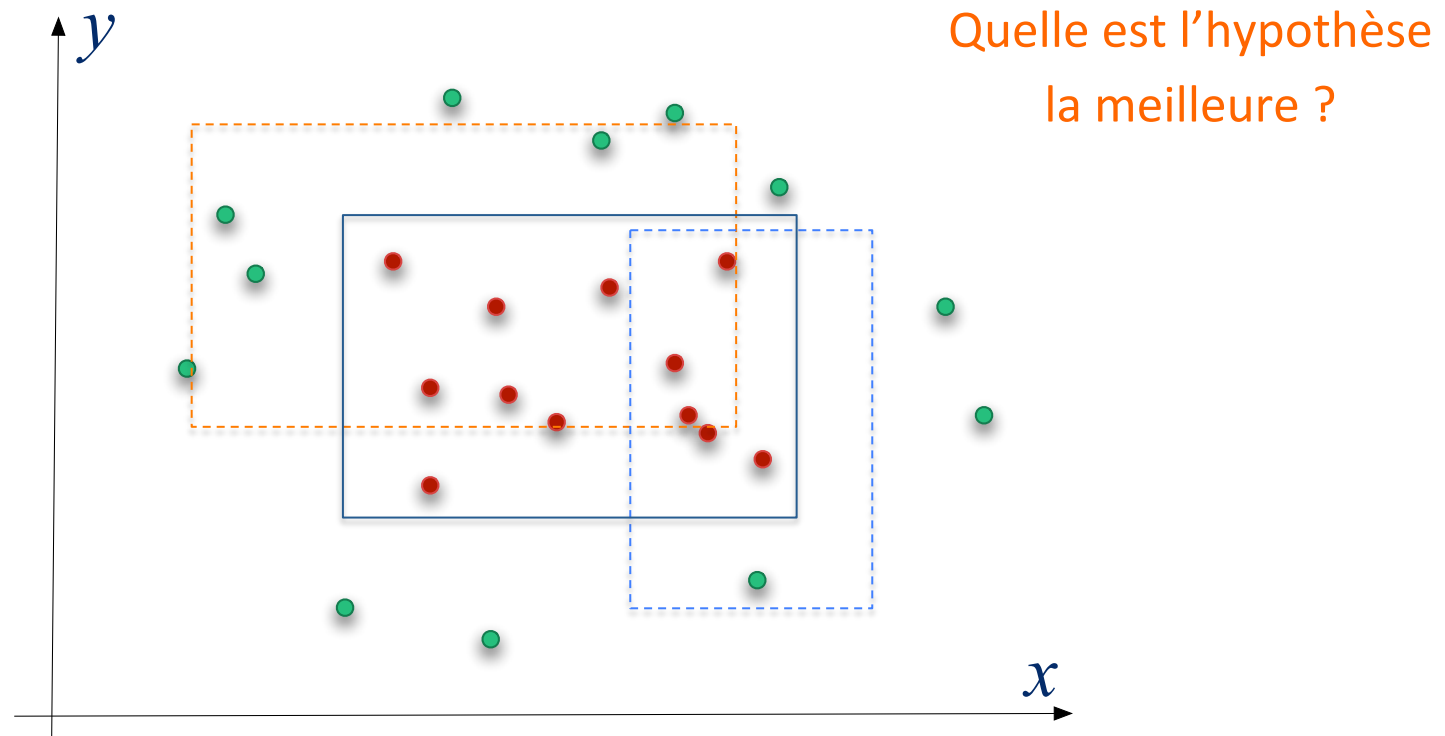


Illustration : apprendre à classer des exemples en 2D

- Comment faire ?



Quelle hypothèse choisir ?

Quelle **qualité** pour **chaque hypothèse candidate** ?

Le « **critère inductif** »

Quelle hypothèse choisir ?

Quelle **performance** ?

- Coût d'une erreur de prédiction
 - La **fonction de perte**

$$\ell(h(\mathbf{x}), y)$$

Quelle hypothèse choisir ?

Quelle **performance** ?

- Coût d'une erreur de prédiction
 - La **fonction de perte**

$$\ell(h(\mathbf{x}), y)$$

- Quel **coût à venir** (espérance) si je choisis h ?
 - Espérance de coût : le « **risque réel** »

$$R(h) = \int_{\mathcal{X} \times \mathcal{Y}} \ell(h(\mathbf{x}), y) \mathbf{p}_{\mathcal{X}\mathcal{Y}}(\mathbf{x}, y) d\mathbf{x} dy$$

Quelle hypothèse choisir ?

Comment trouver h^* (ou une bonne hypothèse) alors que l'on n'a accès qu'à un **échantillon d'apprentissage limité** ?

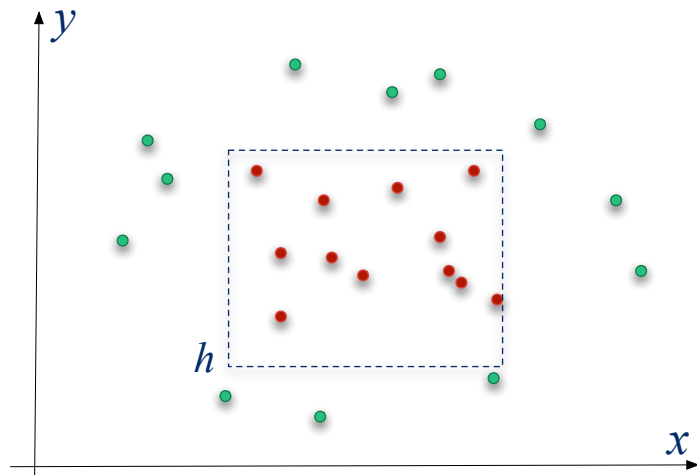
Quelle hypothèse choisir ?

Comment trouver h^* (ou une bonne hypothèse) alors que l'on n'a accès qu'à un **échantillon d'apprentissage limité** ?

- Critère inductif : $\mathcal{H} \times S \rightarrow \text{valeur}(h)$
- Le plus naturel : ERM
 - La Minimisation du Risque Empirique

Quelle hypothèse choisir ?

- Quelle performance attendue pour h ?
 - **Erreur moyenne** sur l'échantillon d'apprentissage S



Le « risque empirique »

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i)$$

- A-t-on raison d'utiliser l'ERM ?

Un exemple qui en dit beaucoup ...

- Exemples décrits par :

Number (1 or 2); **size** (small or large); **shape** (circle or square); **color** (red or green)

Description	Your prediction	True class
1 large red square		-
1 large green square		
2 small red squares		
2 large red circles		
1 large green circle		
1 small red circle		

Un exemple qui en dit beaucoup ...

- Exemples décrits par :

Number (1 or 2); **size** (small or large); **shape** (circle or square); **color** (red or green)

Description	Your prediction	True class
1 large red square		-
1 large green square		+
2 small red squares		
2 large red circles		
1 large green circle		
1 small red circle		

Un exemple qui en dit beaucoup ...

- Exemples décrits par :

Number (1 or 2); **size** (small or large); **shape** (circle or square); **color** (red or green)

Description	Your prediction	True class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		
1 large green circle		
1 small red circle		

Un exemple qui en dit beaucoup ...

- Exemples décrits par :

Number (1 or 2); **size** (small or large); **shape** (circle or square); **color** (red or green)

Description	Your prediction	True class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		
1 small red circle		

Un exemple qui en dit beaucoup ...

- Exemples décrits par :

Number (1 or 2); **size** (small or large); **shape** (circle or square); **color** (red or green)

Description	Your prediction	True class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		

Un exemple qui en dit beaucoup ...

- Exemples décrits par :

Number (1 or 2); **size** (small or large); **shape** (circle or square); **color** (red or green)

Description	Your prediction	True class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		+

Un exemple qui en dit beaucoup ...

- Exemples décrits par :

Number (1 or 2); **size** (small or large); **shape** (circle or square); **color** (red or green)

Description	Your prediction	True class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		+

How many possible functions altogether from X to Y ?

$$2^{2^4} = 2^{16} = 65,536$$

How many functions do remain after 6 training examples?

$$2^{10} = 1024$$

Un exemple qui en dit beaucoup ...

- Examples described using:

Number (1 or 2); **size** (small or large); **shape** (circle or square); **color** (red or green)

Description	Your prediction	True class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		+
1 small green square		-
1 small red square		+
2 large green squares		+
2 small green squares		+
2 small red circles		+
1 small green circle		-
2 large green circles		-
2 small green circles		+
1 large red circle		-
2 large red squares	?	

15

How many remaining functions?



?

Induction: impossible de gagner ?

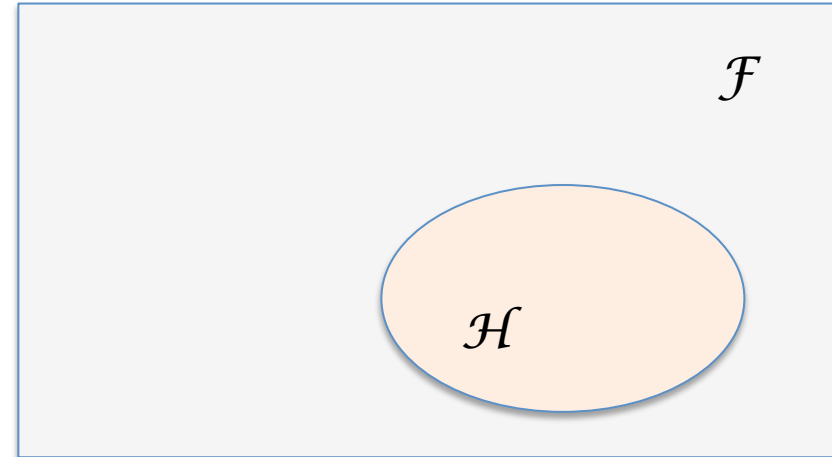
- **Un biais est nécessaire**
- **Types de biais**
 - **De représentation** (déclaratif)
 - **De recherche** (procédural)

Induction: impossible de gagner ?

- **Un biais est nécessaire**

- **Types de biais**

- **De représentation** (déclaratif)
- **De recherche** (procédural)



Induction: impossible de gagner ?

- **Un biais est nécessaire**

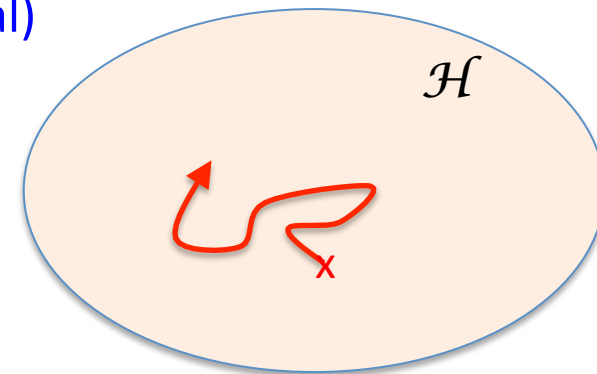
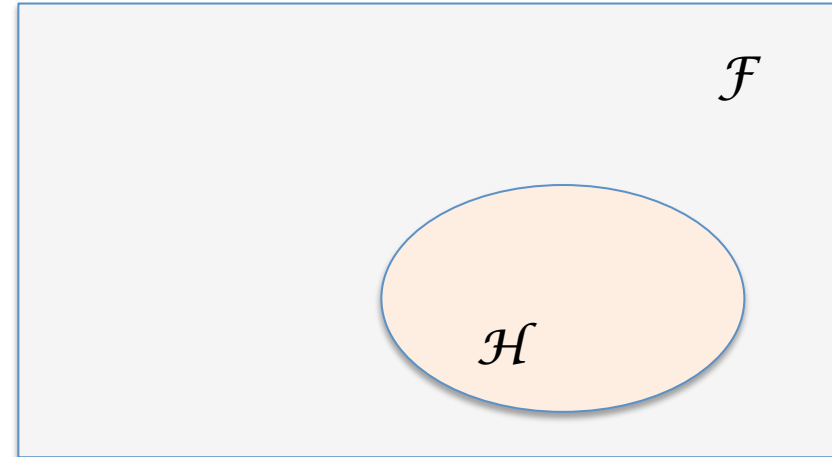
- **Types de biais**

- **De représentation**

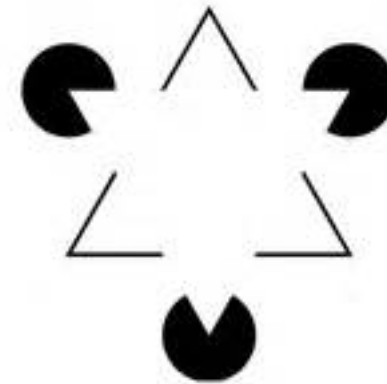
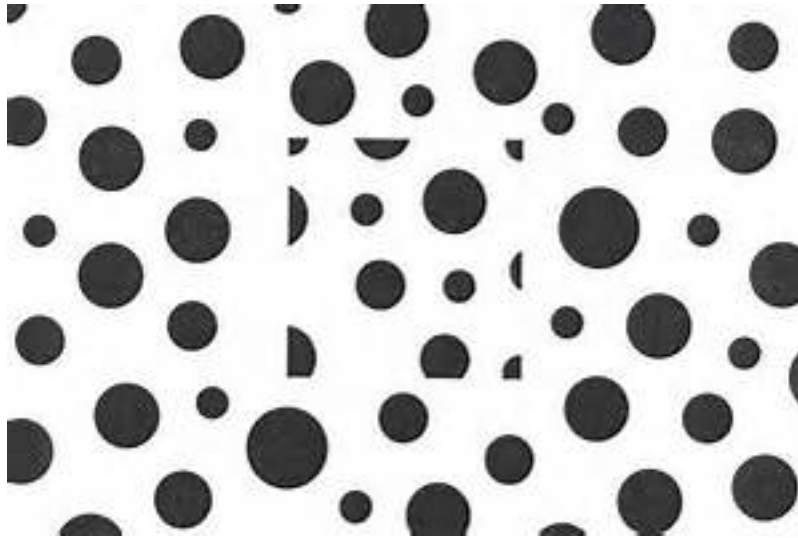
(déclaratif)

- **De recherche**

(procédural)



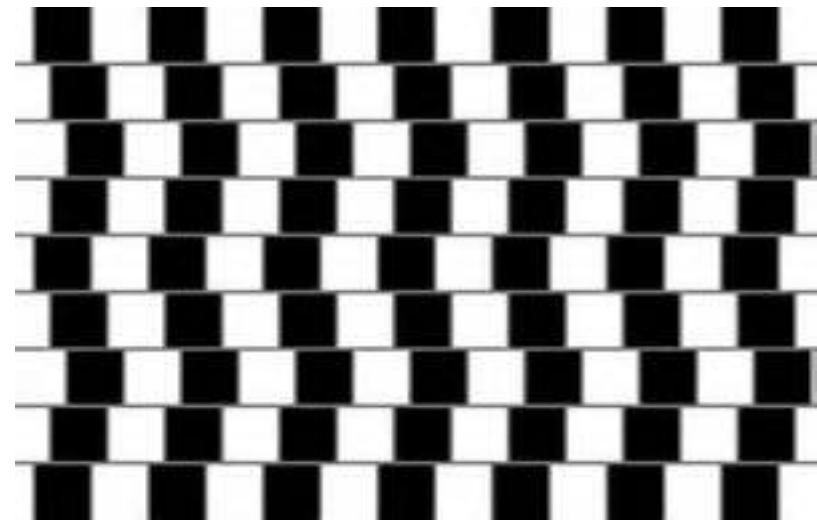
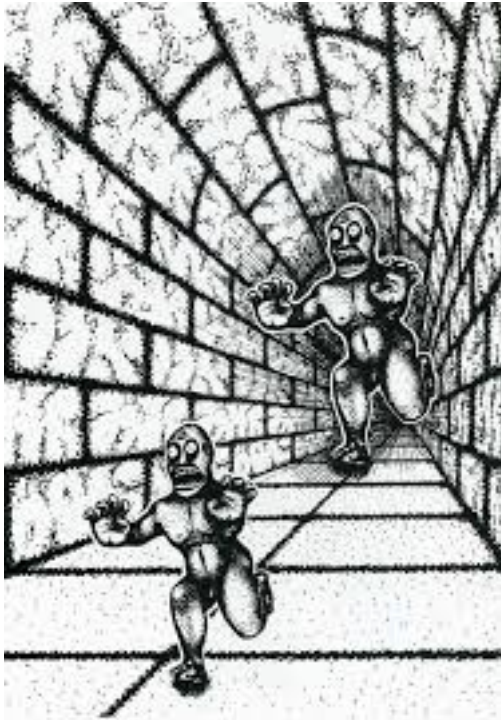
Learning – completing data with necessary a priori



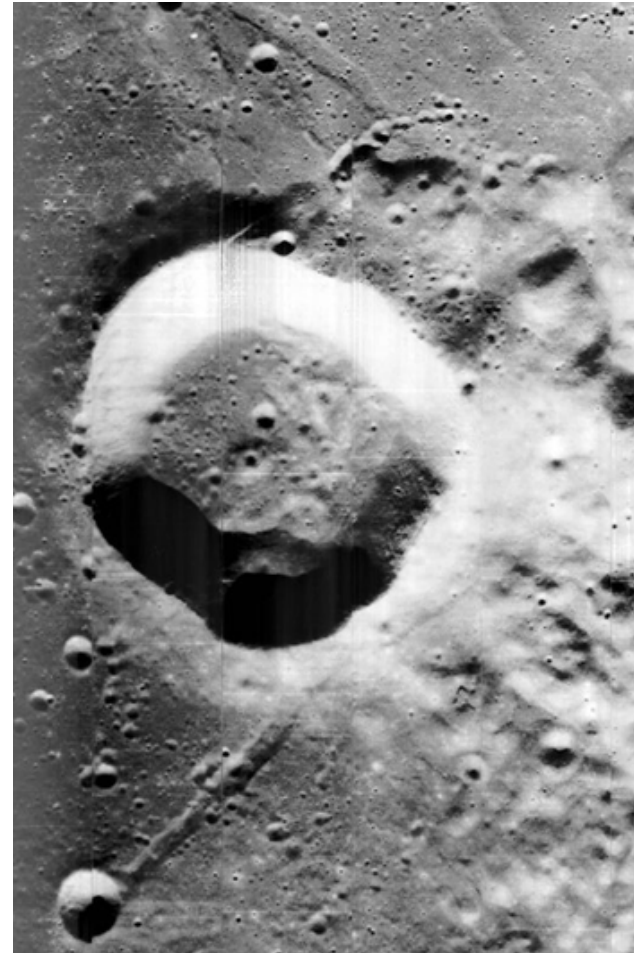
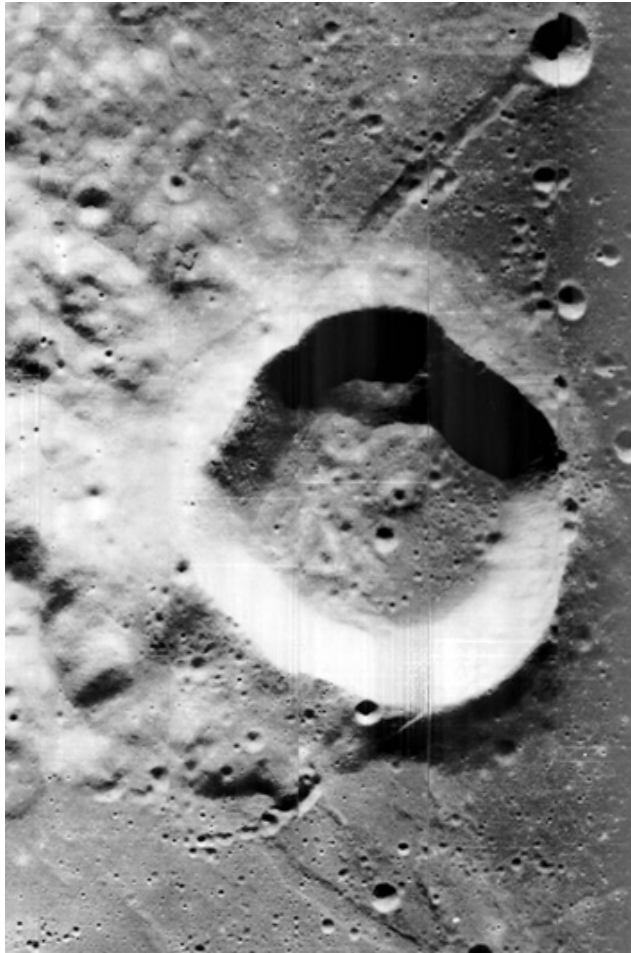
Interpreting – completion of percepts



Induction and its illusions



Induction and illusions



Crater *or* hill?

Un exemple qui en dit beaucoup ...

- Examples described using:

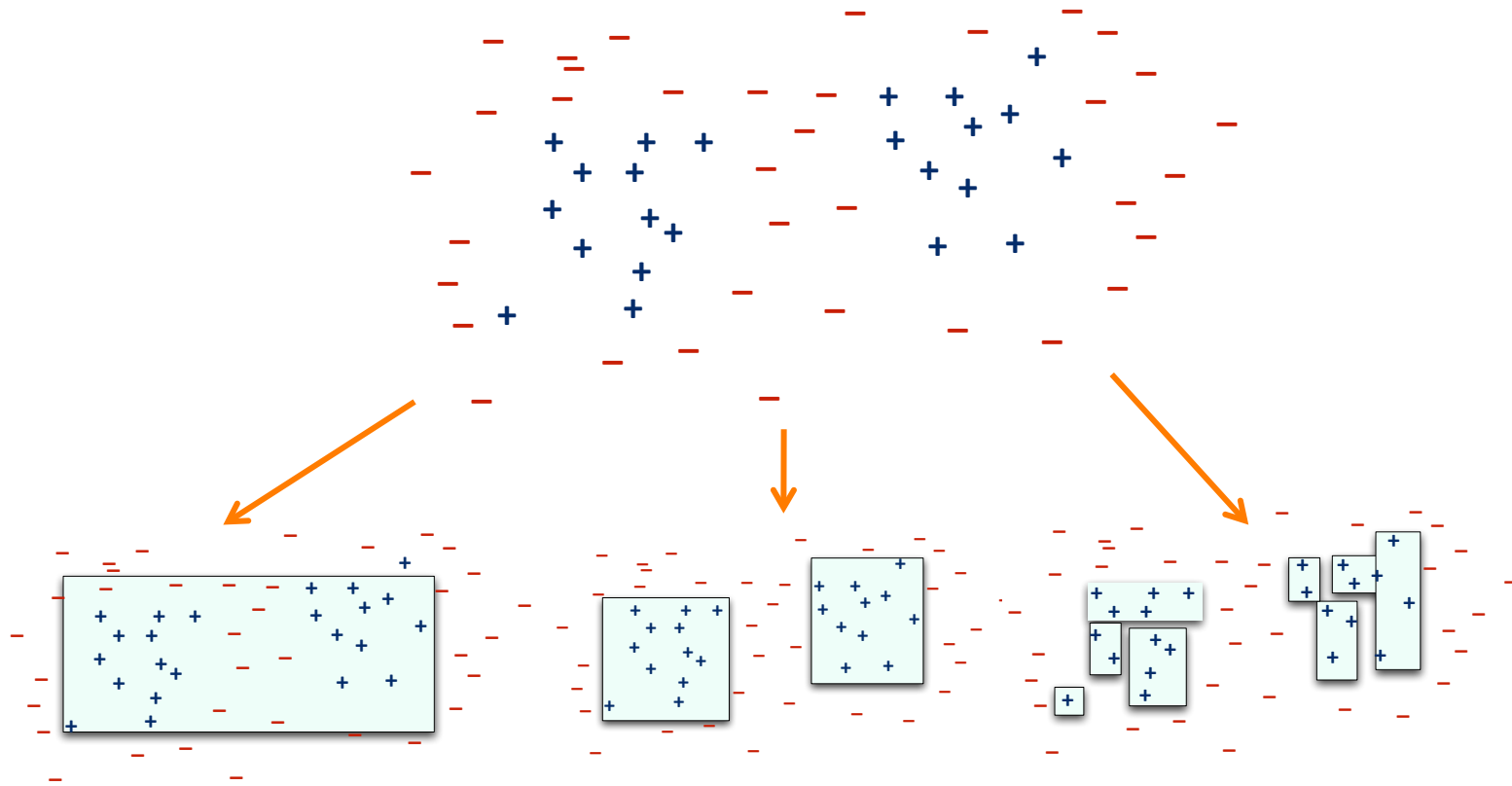
Number (1 or 2); **size** (small or large); **shape** (circle or square); **color** (red or green)

Description	Your prediction	True class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		+

How many possible functions with 2 descriptors from X to Y ? $2^{2^2} = 2^4 = 16$

How many functions do remain after 3 \neq training examples? $2^1 = 2$

Comment garantir un niveau de performance ?



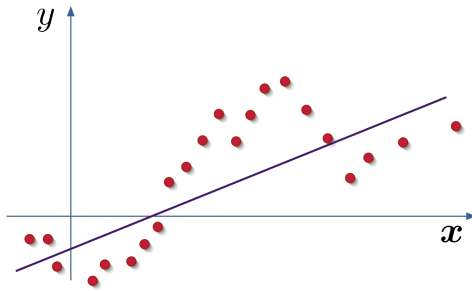
Sous-apprentissage

Bon-apprentissage

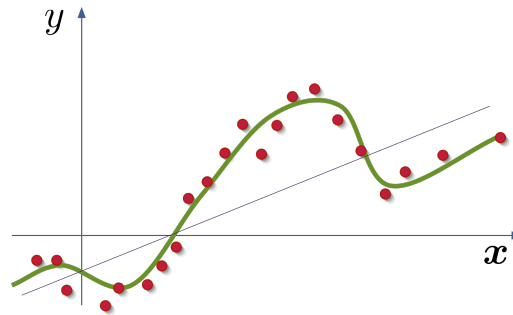
Sur-apprentissage

x

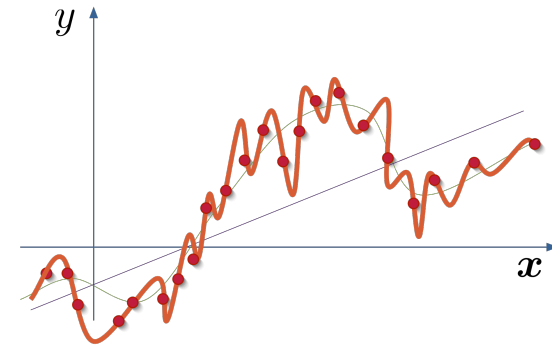
Comment garantir un niveau de performance ?



Sous-apprentissage



Bon-apprentissage



Sur-apprentissage

Question centrale : le principe inductif

- Le principe de **minimisation du risque empirique** (ERM)
... est-il sain ?

— **Si** je choisis h telle que $\hat{h} = \underset{h \in \mathcal{H}}{\text{ArgMin}} \hat{R}(h)$

— **Est-ce que** h est bonne relativement au risque réel ?

$$\hat{R}(\hat{h}) \overset{?}{\longleftrightarrow} R(\hat{h})$$

— **Est-ce que** j'aurais pu faire beaucoup mieux ? $h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} R(h)$

$$R(h^*) \overset{?}{\longleftrightarrow} R(\hat{h})$$

L'analyse « PAC learning »

- On arrive à :

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \mathbf{P}^m \left[R_{\text{Réal}}(h) \leq R_{\text{Emp}}(h) + \overbrace{\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{m}}^{\varepsilon} \right] > 1 - \delta$$

Le principe de minimisation du risque empirique

n'est **sain que si** il y a des contraintes sur l'espace des hypothèses

Plan

1. Pourquoi toute cette excitation ?
2. Grands types d'apprentissage
3. Apprentissage prédictif par réseaux de neurones
4. Quelles garanties ?
5. **Recette pour créer des algorithmes d'apprentissage**
6. Les réseaux de neurones profonds
7. Ce que l'on sait faire et les défis à relever

Recette pour ... **concevoir des algorithmes** d'apprentissage

1. Définir un **critère inductif régularisé**

- a. Exprimer le coût d'erreur de prédiction en une **fonction de perte**
- b. Définir un **terme de régularisation** qui exprime les attendus sur les régularités du monde
- c. Si possible, rendre convexe le problème d'**optimisation** résultant

$$h_{opt} = \underset{h \in \mathcal{H}}{\text{ArgMin}} \left[\underbrace{\frac{1}{m} \sum_{i=1}^m l(h(\mathbf{x}_i), y_i)}_{\text{empirical risk}} + \lambda \underbrace{\text{reg}(\mathcal{H})}_{\text{bias on the world}} \right]$$

2. Utiliser ou développer un **algorithme d'optimisation efficace**

Learning **sparse linear** approximator

- The **hypothesis** is of the form $h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$
- **A priori assumption**: few non zero coefficients

Ridge regression

$$\mathbf{w}_{\text{ridge}}^* = \underset{\mathbf{w}}{\text{Argmin}} \left\{ \sum_{i=1}^m (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_2^2 \right\}$$

Lasso regression

$$\mathbf{w}_{\text{lasso}}^* = \underset{\mathbf{w}}{\text{Argmin}} \left\{ \sum_{i=1}^m (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 + \lambda \|\mathbf{w}\|_1 \right\}$$

Risque
empirique
régularisé

3.3 du chapitre 3. Ainsi, étant donné un échantillon source étiqueté $S = \{(x_i^s, y_i^s)\}_{i=1}^m$ constitué de m exemples *i.i.d.* selon P_S et un échantillon cible non étiqueté $T = \{(x_i^t)\}_{i=1}^m$ composé de m exemples *i.i.d.* selon D_T , en posant $S_u = \{x_i^s\}_{i=1}^m$ l'échantillon S privé de ses étiquettes, on veut minimiser :

$$\min_w c m R_S(G_{\rho_w}) + a m \text{dis}_{\rho_w}(S_u, T_u) + \text{KL}(\rho_w \| \pi_0), \quad (7.5)$$

où $\text{dis}_{\rho_w}(S_u, T_u) = \left| \mathbb{E}_{(h, h') \sim \rho_w^2} R_{S_u}(h, h') - \mathbb{E}_{(h, h') \sim \rho_w^2} R_{T_u}(h, h') \right|$ est le désaccord empirique entre S_u et T_u spécialisé à une distribution ρ_w sur l'espace \mathcal{H} des classifieurs linéaires considéré. Les réels $a > 0$ et $c > 0$ sont des hyperparamètres de l'algorithme. Notons que les constantes A et C du théorème 7.7 peuvent être retrouvées à partir de n'importe quelle valeur de a et c . Étant donnée la fonction $\ell_{\text{dis}}(x) = 2 \ell_{\text{Erf}}(x) \ell_{\text{Erf}}(-x)$ (illustrée sur la figure 7.1), pour toute distribution D sur X , on a :

$$\begin{aligned} \mathbb{E}_{(h, h') \sim \rho_w^2} R_D(h, h') &= \mathbb{E}_{x \sim D} \mathbb{E}_{(h, h') \sim \rho_w^2} \mathbf{I}[h(x) \neq h'(x)] \\ &= 2 \mathbb{E}_{x \sim D} \mathbb{E}_{(h, h') \sim \rho_w^2} \mathbf{I}[h(x) = 1] \mathbf{I}[h'(x) = -1] \\ &= 2 \mathbb{E}_{x \sim D} \mathbb{E}_{h \sim \rho_w} \mathbf{I}[h(x) = 1] \mathbb{E}_{h' \sim \rho_w} \mathbf{I}[h'(x) = -1] \\ &= 2 \mathbb{E}_{x \sim D} \ell_{\text{Erf}}\left(\frac{\langle \mathbf{w}, \mathbf{x} \rangle}{\|\mathbf{x}\|}\right) \ell_{\text{Erf}}\left(-\frac{\langle \mathbf{w}, \mathbf{x} \rangle}{\|\mathbf{x}\|}\right) \\ &= \mathbb{E}_{x \sim D} \ell_{\text{dis}}\left(\frac{\langle \mathbf{w}, \mathbf{x} \rangle}{\|\mathbf{x}\|}\right). \end{aligned}$$

Expression de
substitution
du risque
régularisé

Ainsi, trouver la solution optimale de l'équation (7.5) revient à chercher le vecteur w qui minimise :

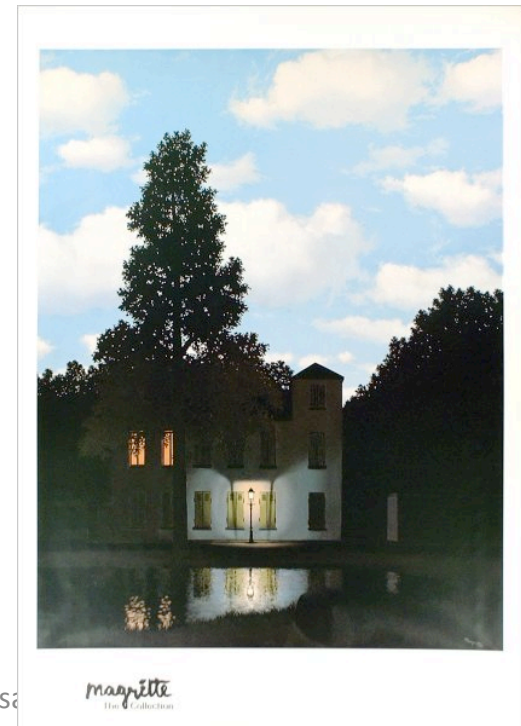
$$c \sum_{i=1}^m \ell_{\text{Erf}}\left(y_i^s \frac{\langle \mathbf{w}, \mathbf{x}_i^s \rangle}{\|\mathbf{x}_i^s\|}\right) + a \left| \sum_{i=1}^m \left[\ell_{\text{dis}}\left(\frac{\langle \mathbf{w}, \mathbf{x}_i^s \rangle}{\|\mathbf{x}_i^s\|}\right) - \ell_{\text{dis}}\left(\frac{\langle \mathbf{w}, \mathbf{x}_i^t \rangle}{\|\mathbf{x}_i^t\|}\right) \right] \right| + \frac{\|\mathbf{w}\|^2}{2}. \quad (7.6)$$

L'équation précédente est fortement non convexe. Afin de rendre sa résolution plus facilement contrôlable, nous remplaçons la fonction $\ell_{\text{Erf}}(\cdot)$ par sa relaxation convexe $\ell_{\text{Erf}_{\text{cvx}}}(\cdot)$ (comme pour PBGD3 et illustrée sur la figure 7.1). L'optimisation se réalise ensuite par une descente de gradient. Le gradient de l'équation 7.6 étant :

Un paradigme général

- Boosting
- Arbres de décisions (random forests)
- Régression logistique
- Réseaux de neurones
- Séparateurs à Vastes Marges (SVM)
- ...

Des garanties « de lampadaire »



Des garanties « de lampadaire »

(*Quasi*) garantie que :

- **Si** le monde satisfait **mes attentes** sur lui
- **Alors** l'algorithme d'apprentissage produira une bonne hypothèse (proche de la vraie)



Des garanties « de lampadaire »

(Quasi) garantie que :

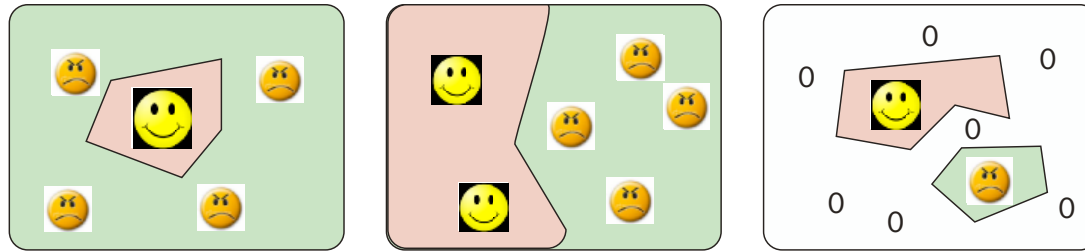
- **Si** le monde satisfait **mes attentes** sur lui
- **Alors** l'algorithme d'apprentissage produira une bonne hypothèse (proche de la vraie)

- **Autrement** l'apprentissage peut conduire à de très mauvaises hypothèses
(ex. *Si le monde n'est pas parcimonieux*)



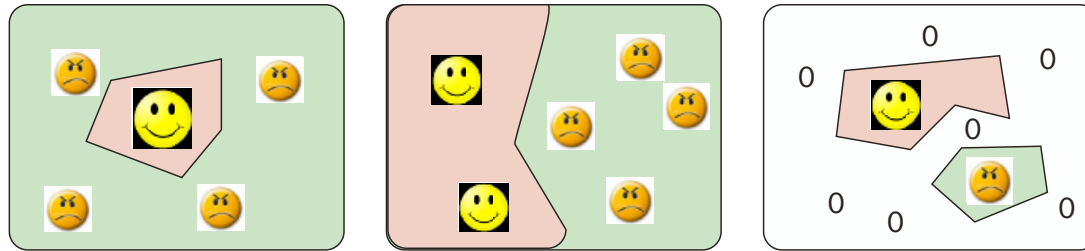
Le no-free-lunch theorem

Possible

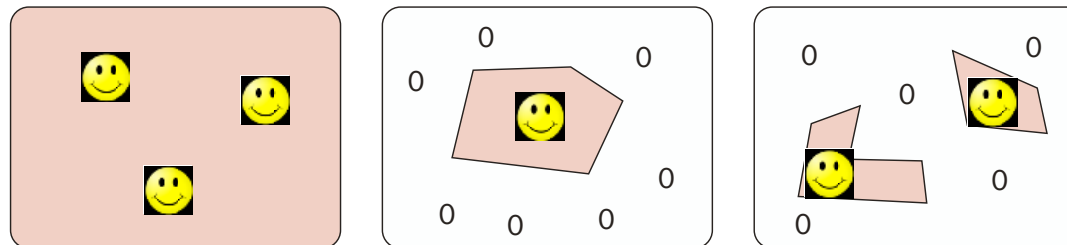


Le no-free-lunch theorem

Possible



Impossible



Il faut **choisir** le **bon** **algorithme** pour la **classe de problèmes** étudiée

Plan

1. Pourquoi toute cette excitation ?
2. Grands types d'apprentissage
3. Apprentissage prédictif par réseaux de neurones
4. Quelles garanties ?
5. Recette pour créer des algorithmes d'apprentissage
6. Les réseaux de neurones profonds
7. Ce que l'on sait faire et les défis à relever

Ce que sont les RNs profonds.

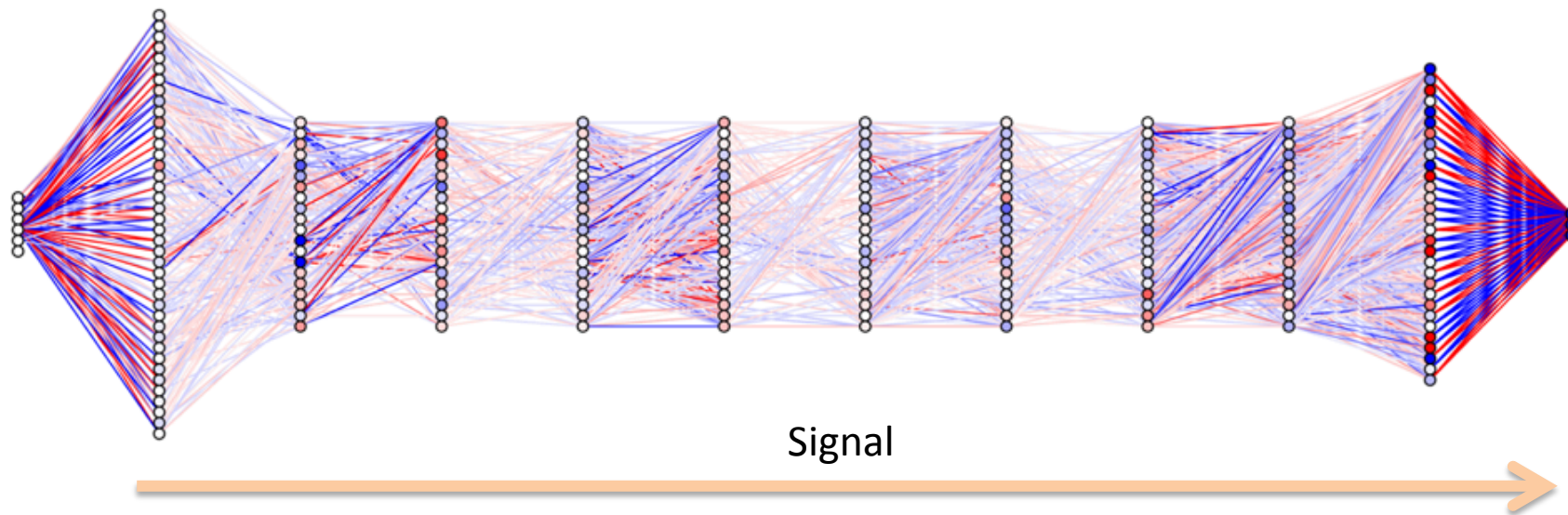
Comment ça marche : le problème de l'optimisation.

Pourquoi ça marche : des risques et une énigme.

Ce que sont les réseaux de neurones **profonds**

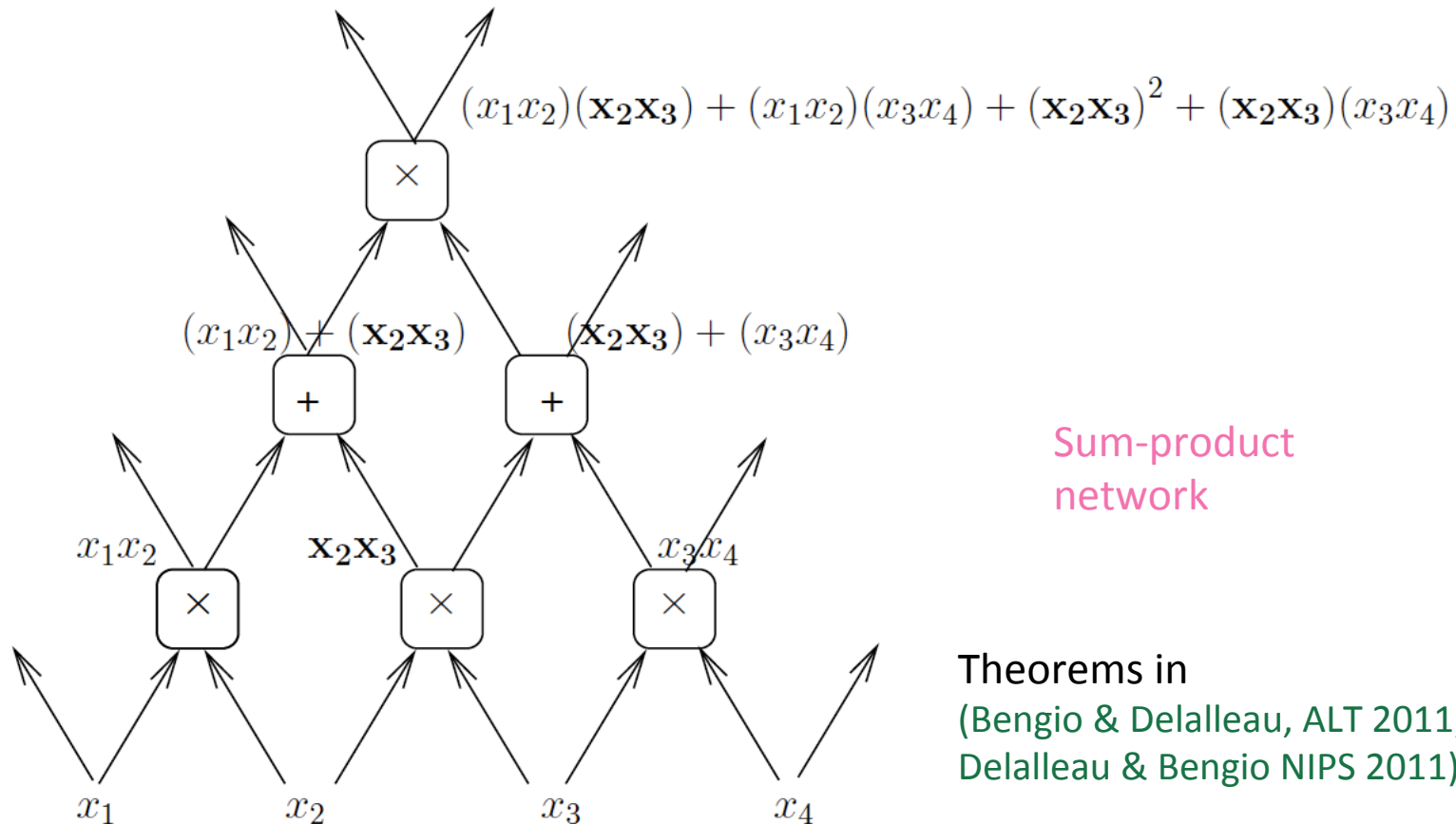
Les « réseaux de neurones **profonds** »

- Des réseaux de neurones artificiels
 - à grand nombre de couches (parfois > qqs 100)
 - et **très grand nombre de paramètres** (qqs $10^7 - 10^{11}$ paramètres)



Shallow vs. Deep architectures

- Polynomial expressed with shared components: advantage of depth may grow exponentially

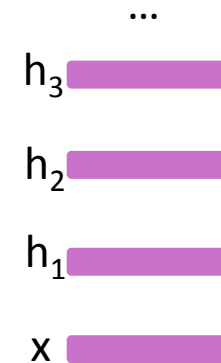


Sum-product network

Theorems in
(Bengio & Delalleau, ALT 2011;
Delalleau & Bengio NIPS 2011)

Représentations profondes

- Idée
 - Apprendre des niveaux de représentation de **plus en plus abstraits** par mise en relation des niveaux inférieurs



- Motivations
 - Permet potentiellement un **gain exponentiel dans la puissance expressive**
 - Le **cerveau humain** utilise des structures en couches successives
 - Les **connaissances** et **théories** humaines sont compositionnelles
 - **Marche très bien expérimentalement** sur certaines tâches

Les « réseaux de neurones **profonds** »

- Des réseaux de neurones artificiels
 1. à grand nombre de couches
 2. et **très grand nombre de paramètres**
 3. qui apprennent des **représentations hiérarchiques**
 4. et **décomposent les calculs**

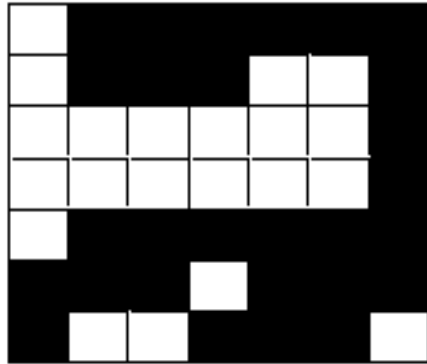
Ce que sont les réseaux de neurones **profonds**

La base de données

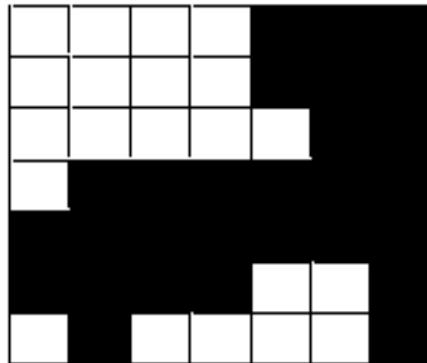
65473 60198 68544
70065 70117 19032^{AP} 96720
27260 61820 19559
74136 ~~19137~~ 63101
20878 60521 38002
48640-2398 20907 14868

How to code the inputs

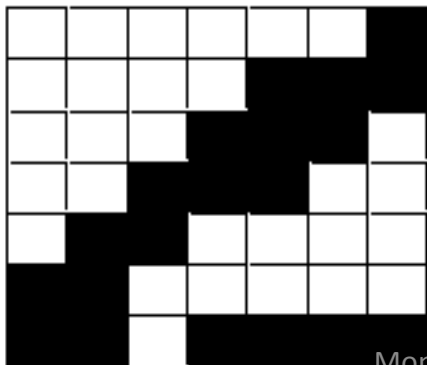
Learning is easy when we know what to look for



- Yes

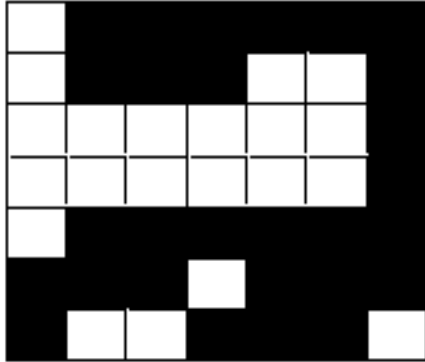


- Yes



- No

Inputs and prior knowledge

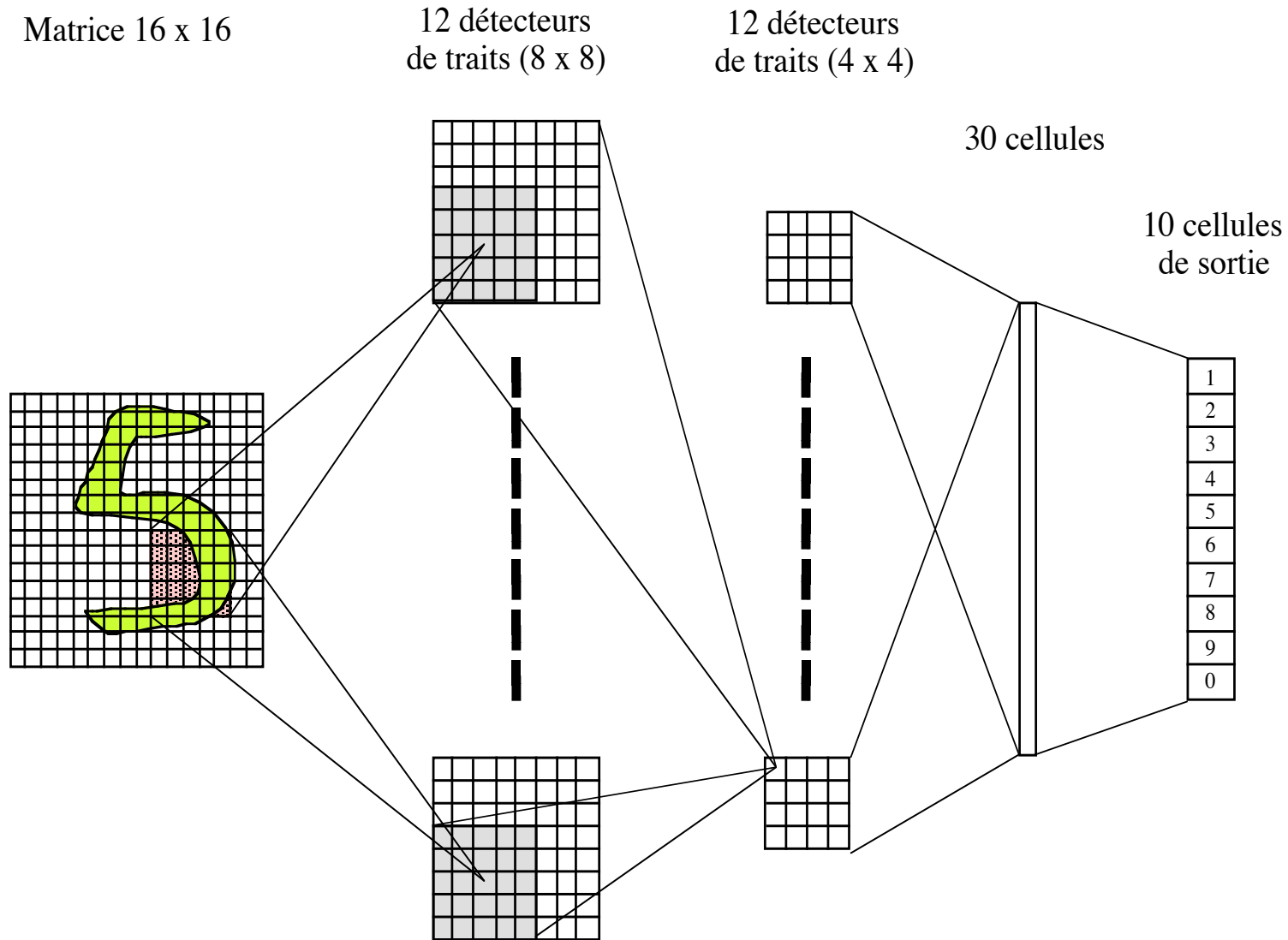


- Is it a pattern recognition task? A character recognition task? ...
- *How to code the examples?*

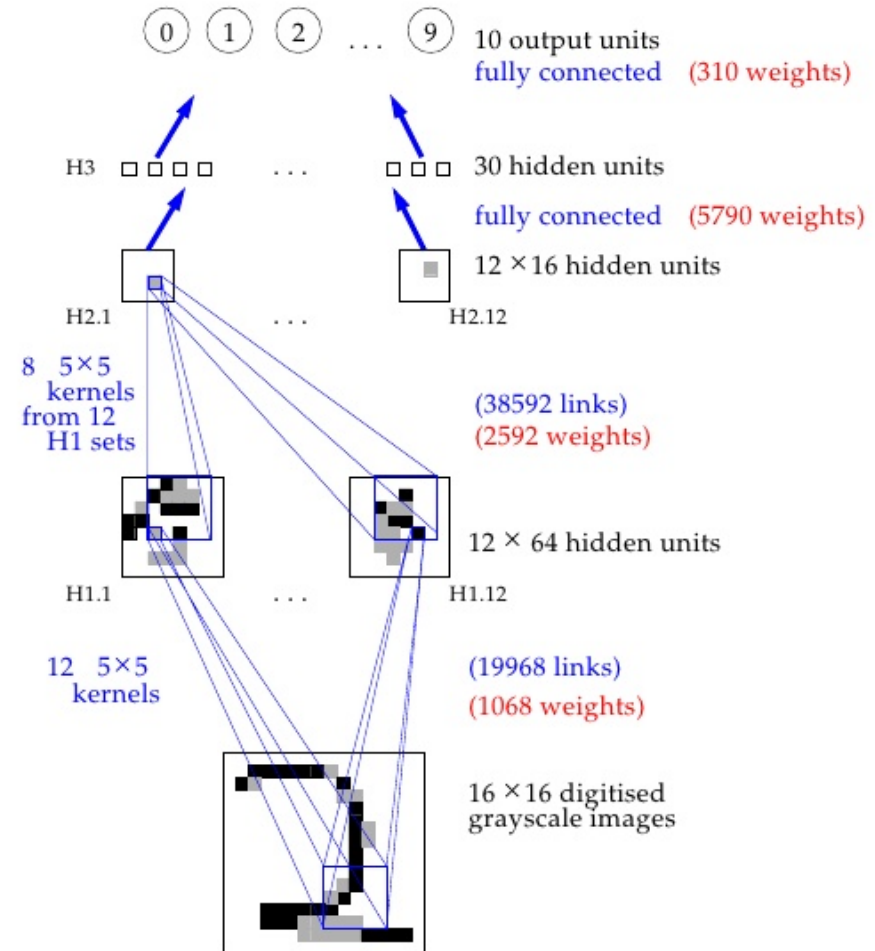
0 1 1 1 1 1 1 0 1 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 0

- *A right choice of representation can render the learning task trivial*
 - ⇒ *But how can we know the right representation?*

Réseaux à convolution : Application aux codes postaux



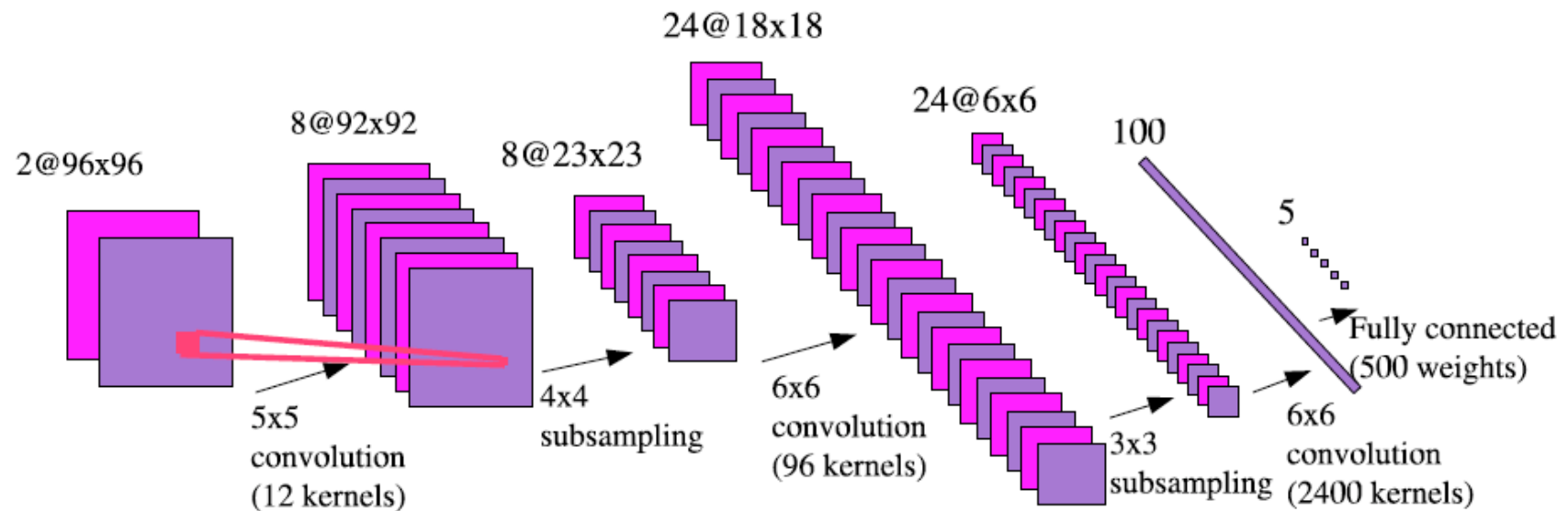
Réseaux à convolution : Application aux codes postaux



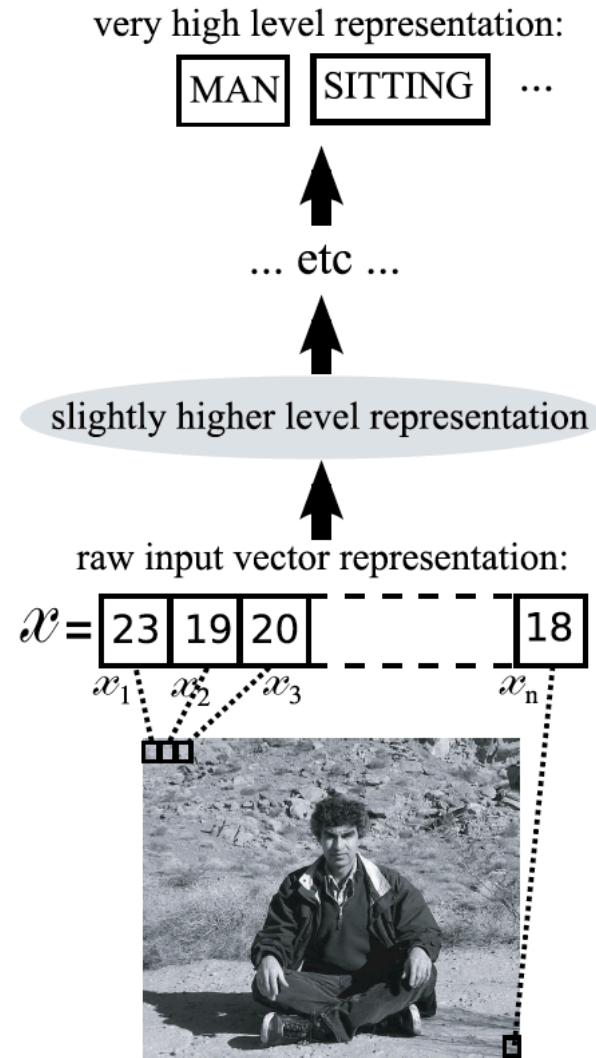
Before weight sharing 64660 links
After weight sharing 9760 weights

- Taken from <http://image.slidesharecdn.com/bp2slides-090922011749-phpapp02/95/the-back-propagation-learning-algorithm-10-728.jpg?cb=1253582278>

Réseau connexionniste pour la reconnaissance de chiffres manuscrits



« Deep belief networks »

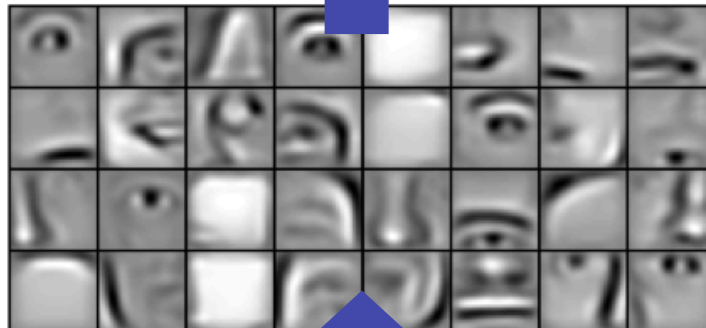


Apprentissage de représentations hiérarchiques

- Apprentissage de représentations hiérarchiques



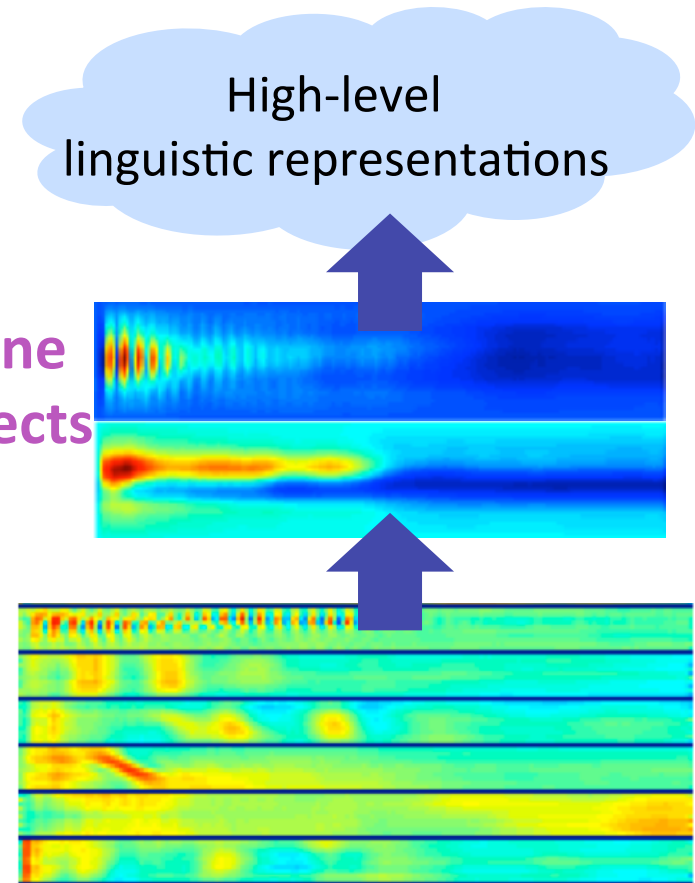
Layer 3



Layer 2



Layer 1



26

Illustration : ImageNet

La compétition ImageNet

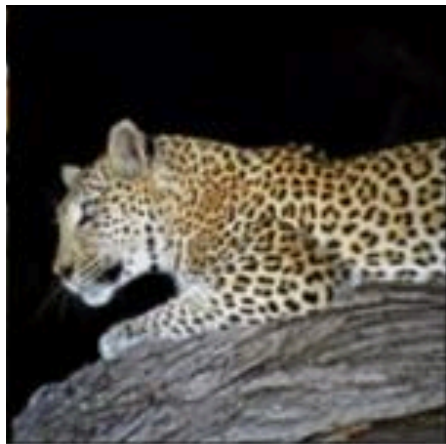
- Plus de **15M d'images** haute résolution étiquetées
- Environ **22K catégories**
- Récoltées sur le Web et étiquetées par Amazon Mechanical Turk



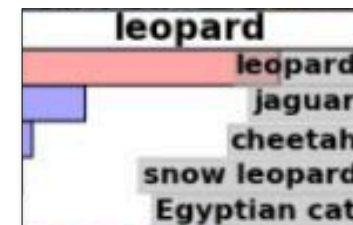
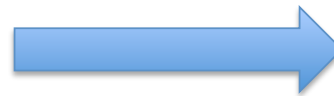
Illustration : ImageNet

La compétition ImageNet

- Plus de **15M d'images** haute résolution étiquetées
- Environ **22K catégories**
- Récoltées sur le Web et étiquetées par Amazon Mechanical Turk



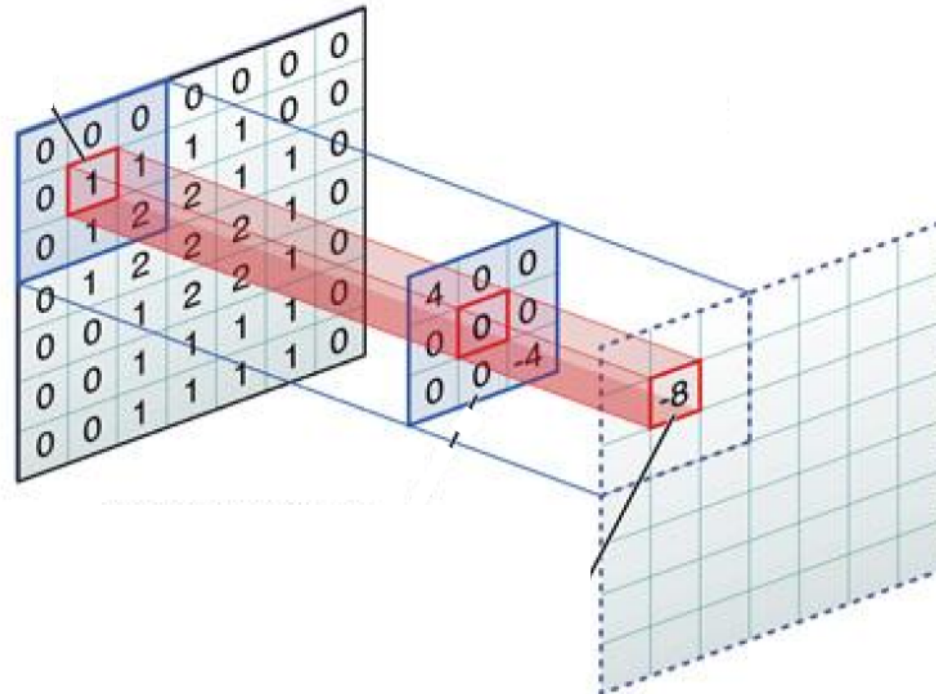
Classification



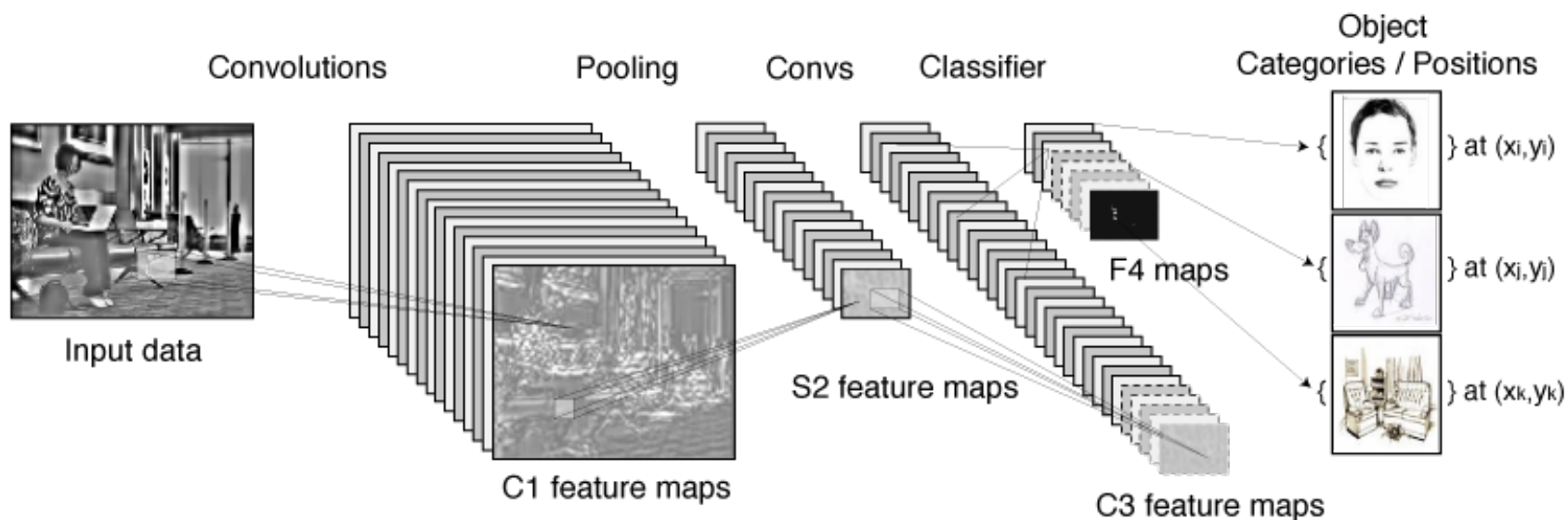
Convolutions

Same dimension between layers

$$(4 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (0 \times 1) + (-4 \times 2) = -8$$



La structure des réseaux de convolution



- 1) Le **pooling** consiste réduire la résolution des images filtrées, par exemple en moyennant les pixels contigus (*ex : 4 pixels deviennent 1 seul pixel qui contient une valeur moyenne*) - étape S2
- 2) **Convolution** : des filtres sont appris sur les nouvelles images - étape C3
- 3) **Classifieur** : les dernières couches sont entièrement connectées (*i.e. MLP*) et apprennent à prédire la classe à partir des filtres appris (*i.e. des descripteurs générés automatiquement*) - étape F4