

# Apprentissage Artificiel

mise en perspective d'un demi-siècle d'évolution



Antoine Cornuéjols

AgroParisTech – INRA MIA 518

# Une science ... son objet

---

« **How** can we **build** *computer systems* that automatically improve with experience, and  
**what** are the **fundamental laws** that govern *all learning processes*? »

*Tom Mitchell, 2006*

Mais d'où viennent  
les algorithmes d'apprentissage ?

# Plan

---

1. Préhistoire et **Cybernétique**
2. Apprendre : c'est **s'adapter**
3. L'âge de **raison**
4. Double coup de butoir et **changement de perspective**
5. Un **paradigme triomphant**
6. La fin de l'histoire ... **et après ?**

---

# Préhistoire et Cybernétique

(~ 1943 - ~1956)

# Avant la cybernétique

---

## ■ Théorie de l'esprit

- Le territoire de la philosophie
- Mais *les pensées ne peuvent pas être mesurées*

## ■ Esprit et cerveau

- Trépanations / effets des lésions : **modularité** de l'esprit
- 1ères mesures des processus de pensée

## ■ Prégnance de la physique

- **Vitesse, force** et **intensité** d'un **signal** dans un milieu continu
- Causes et effets. Causes **avant** les effets.

# La cybernétique

---

*« Cybernetics is not merely another branch of science. It is an **Intellectual Revolution** that rivals in importance the earlier Industrial Revolution »*

Isaac Asimov, 1956

# La cybernétique : machines programmables

---

## ■ La « machine » de Turing (1936)

- Opère sur des **symboles** (*vs. objets avec une masse*)
  - Table de comportements
  - Pas de distinction entre nombres et fonctions opérant sur les nombres
1. Toutes les **fonctions calculables** sont réalisables sur une machine de Turing
  2. **La notion de science change ...**



# La cybernétique

---

- Science = activité de **modélisation**

- Abstraction
- Prédiction



- Machine de Turing (programmée)

- **Modèle** du monde
- « **Effectif** » : capable de *reproduire le monde*

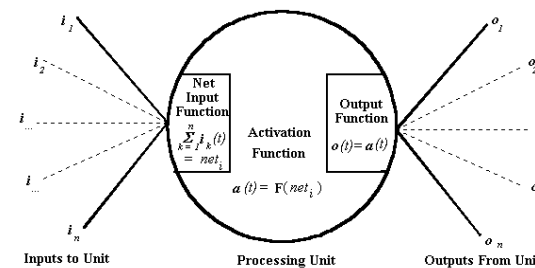
# La cybernétique

## ■ Shannon

- Master's thesis (1937)
  - Étude de circuits électriques
  - Utilise une représentation logique (vs. En termes de quantités électriques)
- 1. La **logique** pour **décrire** des **circuits**
- 2. Des **circuits** pour « **effectuer** » la **logique**

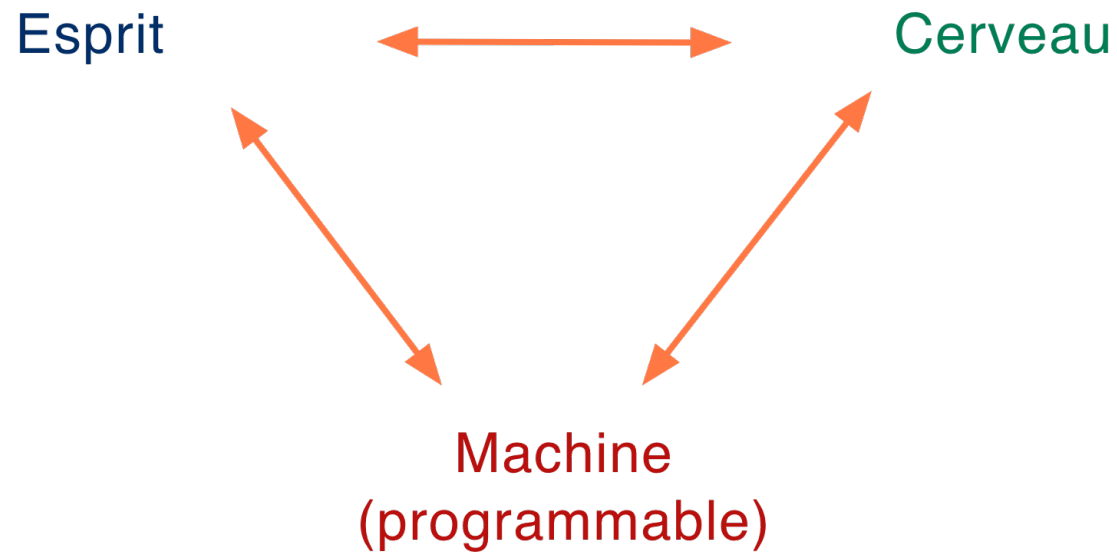
## ■ McCulloch et Pitt (1943)

- **Modèle logique** du neurone



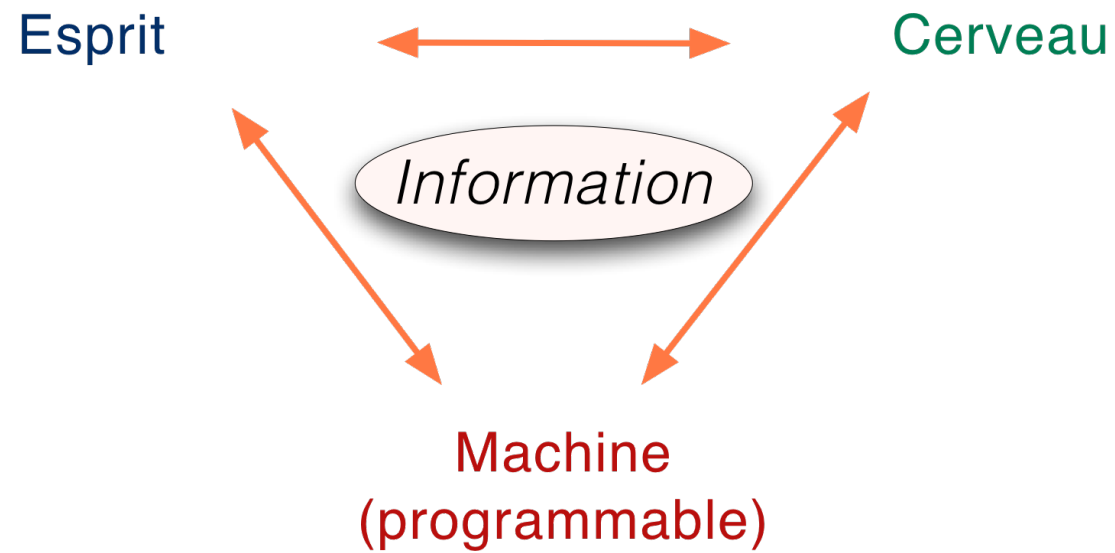
# La cybernétique

---



# La cybernétique

---



# La cybernétique

---

## ■ La notion d'**information**

### – Se construit à partir de :

- La notion de consigne et de **boucle de rétro-action**
  - *Norbert Wiener* (« Cybernetics », 1948)
- La thermodynamique : information et **néguentropie**
  - *Léon Brillouin* (« Science et théorie de l'information », 1959)
- La **théorie de la communication** et du codage
  - *Claude Shannon* (« A mathematical theory of communication », 1948)

### – L'**information** devient :

- La nouvelle **unité de compte**
- Et de **mesure**

# La cybernétique

---

## ■ **Machines programmables**

### – **Algorithmes**

- Machine de Turing : Application de règles sur des symboles

### – **Traitement de l'information**

- Ce qui « est cause de » est de l'information
- Toute interaction est communication

### – **Programme = données => auto-modifiable**

Tout est TRÈS difficile à programmer => **L'apprentissage** est fondamental

# La cybernétique : son projet

---

- Mémoire,  
adaptation,  
apprentissage,  
raisonnement,  
représentation

- Qu'est-ce que c'est ?
- Comment ça fonctionne ?

*Algorithmes*  
*Échanges d'information*

Dans tous les êtres vivants

# Une série de conférences extraordinaires

---

- Les 10 **conférences Macy** (1943 – 1953)
- **Hixon Symposium** on Cerebral Mechanisms in Behavior (1948)
- **Session on Learning Machines** (1955)
- **Dartmouth Summer School** on Artificial Intelligence (1956)
- **Symposium on the "Mechanization of Thought Processes"** (1958)



# Session on Learning Machines (1955)

---

- Wesley Clark and Belmont Farley : « [Generalization of Pattern Recognition in a Self-Organizing System](#) »

*Some pattern-recognition experiments on networks of **neuron-like elements**. Règle de **Hebb**. Allusion à capacité de **généralisation**.*

- Gerald Dinneen (1924- ) : « [Programming Pattern Recognition](#) ».

*Computational techniques for processing images. Suggère d'utiliser des **filtres** sur des **images pixelisées** en niveaux de gris.*

- Oliver Selfridge (1926-2008) : « [Pattern Recognition and Modern Computers](#) ».

*Techniques for highlighting features in clean-up images" (coins, carrés, triangles)*

- Allen Newell (1927-1992) : « [The Chess Machine: An Example of Dealing with a Complex Task by Adaptation](#) ».

*About programming a computer to **play chess**. Notions de buts, de recherche dans un espace de sous-but, de recherche en meilleur d'abord, d'heuristique, de calcul des prédicats.*

# Dartmouth Summer School on Artificial Intelligence (1956)

---

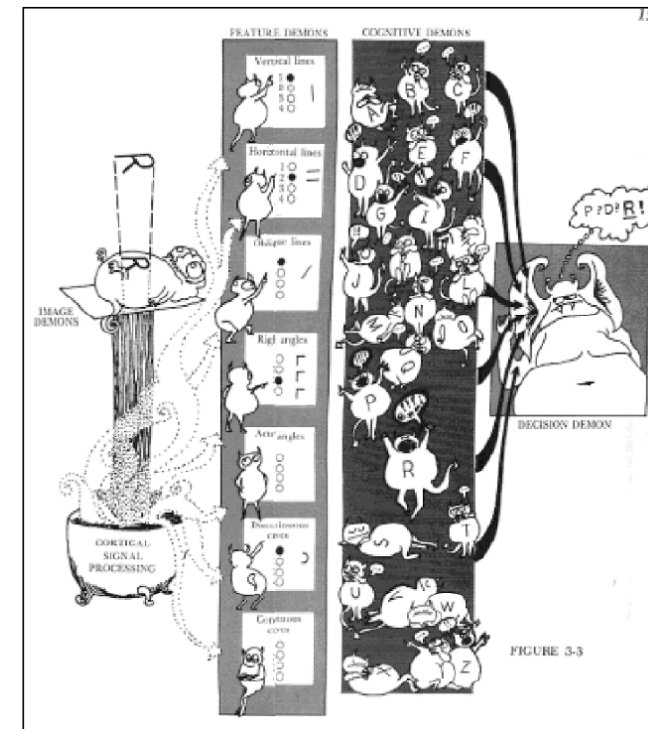
- John McCarthy : [Language de la pensée.](#)  
Précurseur de Lisp
- Allen Newell and Herbert Simon : [Logic Theorist.](#)
- Marvin Minsky : [Réseaux connexionnistes -> Approche symbolique.](#)

*« Consider a machine that would tend to build up within itself **an abstract model of the environment** in which it is placed. If it were given a problem it would **first explore solutions** within the internal abstract model of the environment and then **attempt external experiments.** »*

# Dartmouth Summer School on Artificial Intelligence (1956)

- Marvin Minsky : Méthodes pour la planification, l'apprentissage et la reconnaissance des formes.
- John McCarthy : Logique des prédicats et Lisp.
- Oliver Selfridge : « Pandemonium: A Paradigm for Learning ».

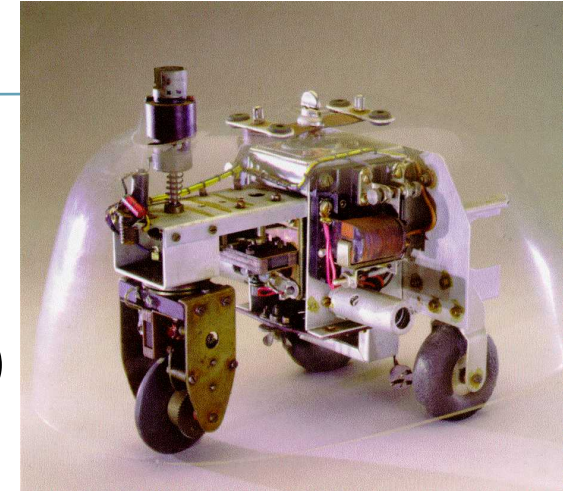
Une architecture hiérarchique de « démons » pour résoudre des problèmes + la suggestion d'un mécanisme d'apprentissage



# Apprentissage : autres explorations

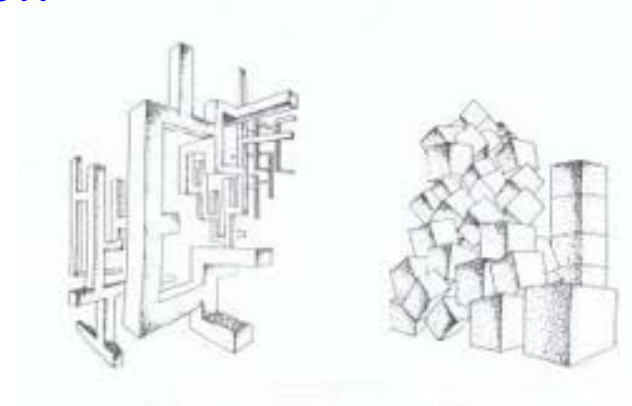
## ■ Apprentissage **par renforcement**

- *Tortues cybernétiques* de Grey Walter (~1950)



## ■ Idées d'**homéostasie** et d'**auto-organisation**

- E.g. « *Order from noise* »
- [Ashby, 1947], [von Foerster, ~1960]



## ■ Question émergente :

### – L'apprentissage de **représentations internes**

- *Assemblées de neurones et règle d'apprentissage*  
[« *The Organization of Behavior* », Hebb, 1949]

# L' « héritage » de la cybernétique

## ■ **Avant** : Idéal de la **physique**

- Forces ; causes et effets ; énergie
- Signal

## ■ **Après** : tout est

- Algorithmes
- Communication

## ■ Abandon de la **géométrie**

## ■ Pour le **calcul** sur des messages



---

# Apprendre : c'est s'adapter

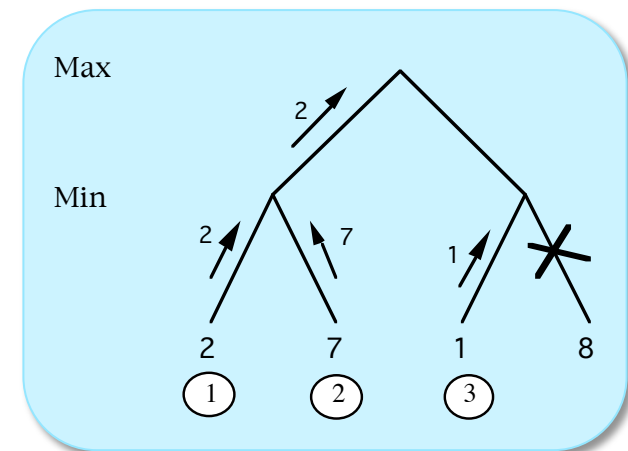
(~ 1956 - ~1970)

# L'exemple de CHECKER

## ■ Combinaison de descripteurs et attribution de mérite

- Arthur Samuel. IBM, 1952 (IBM-701), 1954 (IBM-704), avec apprentissage : 1956 ...
- Modélisation MinMax du jeu
- Apprentissage de la **fonction d'évaluation**

$$\text{valeur}(\text{position}) = \sum_{i=1}^n w_i \phi_i$$



## ■ Deux problèmes

1. Sélectionner de bonnes fonctions de base :  $\phi_i$
2. Pondérer l'importance de ces fonctions :  $w_i$

# L'exemple de CHECKER

---

## ■ **Pondération** des fonctions de base

- Apprentissage de la fonction d'évaluation dans une approche MinMax.
- **Fonction linéaire de 38 attributs** (n'utilisant que les 16 meilleurs).
- Principe : **modifier les poids** pour que **l'évaluation à la racine soit plus proche de celle ramenée par MinMax**.
  - Précurseur de la méthode des différences temporelles [Sutton] en apprentissage par renforcement.
- **Apprentissage par cœur** de la valeur de certaines positions pour des parties jouées.

<http://www.fierz.ch/samuel.html>



# L'exemple de CHECKER

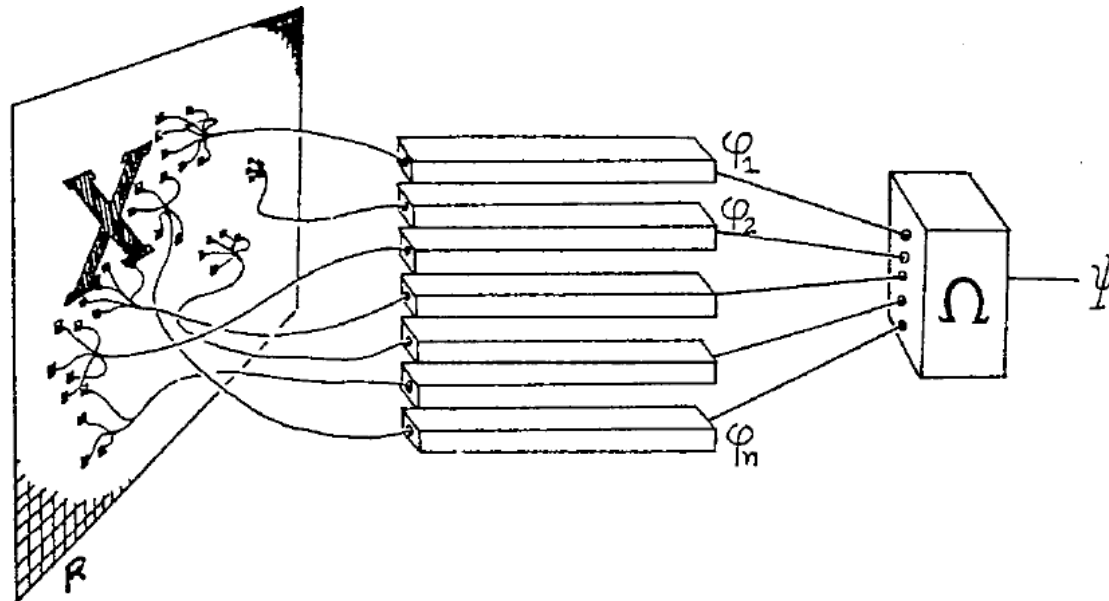
---

- **Recherche** de bonnes fonctions de base
  - Choix aléatoire de 16 fonctions parmi 38.
  - À chaque fois qu'une fonction de base a eu la moins bonne pondération :  $\text{score} := \text{score} + 1$ .
  - Quand  $\text{score} > 32$  : fonction éliminée et remplacée par une autre du pool

Jugé peu satisfaisant par Samuel qui voudrait pouvoir **inventer** de nouvelles fonctions de base

# Premier connexionnisme : le perceptron

- Frank Rosenblatt (1958 – 1962)



$$\Psi(\mathbf{x}) = \sum_{i=1}^n w_i \phi_i(\mathbf{x})$$

# Premier connexionnisme : 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

---

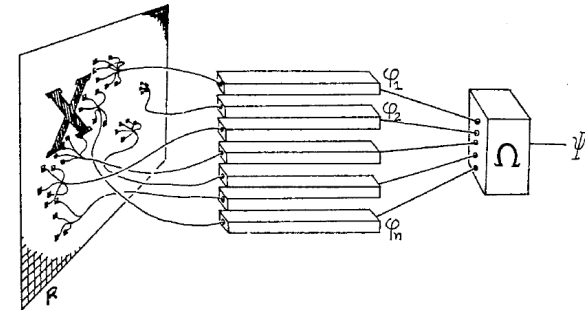
```
tant que non convergence faire  
  | si la forme d'entrée est correctement classée alors  
  |   | ne rien faire  
  | sinon  
  |   |  $w(t + 1) = w(t) - \eta x_i y_i$   
  | fin  
  | Passer à la forme d'apprentissage suivante  
fin
```

---

# Premier connexionnisme : le perceptron

## ■ Propriétés

- Algorithme **en-ligne**
- **Ne pouvait pas tout apprendre !?**
  - Car **ne peut pas tout représenter**
  - Il faut avoir de **bonnes fonctions de base** (détecteurs locaux)
  - Il faut **savoir les combiner**

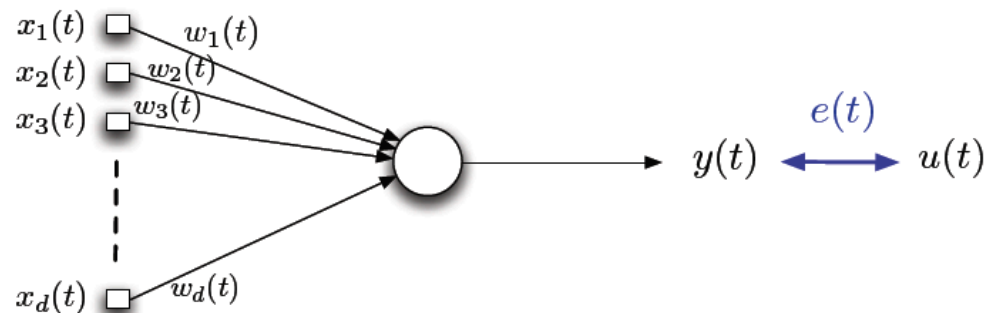
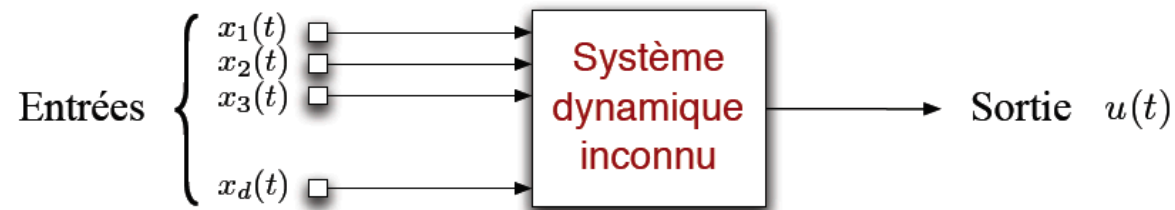


→ Blocage

# La règle de Widrow-Hoff

Conçue dans le cadre du **filtrage adaptatif**.

Chercher un **modèle linéaire d'un signal temporel** :  $y(t) = \sum_{k=1}^M w_k(t)x_k(t)$



# La règle de Widrow-Hoff

B. Widrow and M. Hoff. *Adaptive Switching Circuits*. IER WESCON Conv. Rec. Pt.4, pp; 96-104

The problem of adjusting the h's is not trivial, because their effects upon performance interact. Suppose that the predictor has only two impulses in its impulse response,  $h_1$  and  $h_2$ . The mean square error for any setting of  $h_1$  and  $h_2$  can be readily derived:

$$\begin{aligned} \epsilon(m) &= f(m) - h_1 f(m-1) - h_2 f(m-2) \\ \overline{\epsilon^2}(m) &= \phi_{ff}(0)h_1^2 + \phi_{ff}(0)h_2^2 - 2\phi_{ff}(1)h_1 - 2\phi_{ff}(2)h_2 \\ &\quad + 2\phi_{ff}(1)h_1h_2 + \phi_{ff}(0) \end{aligned} \quad (1)$$

The discrete autocorrelation function of the input is  $\phi_{ff}(j)$ .

The mean square error given by equations (1) is what the mean square meter would read if it were to average over very large sampled size. The mean square error is a parabolic function of the predictor adjustments  $h_1$  and  $h_2$ , and, in general, can easily be shown to be a quadratic function of such adjustments, regardless of how many there are.

The optimum n-impulse predictor can be derived analytically by setting the partial derivatives of  $\epsilon^2$  of equation (1) equal to zero. This is the discrete analogue of Wiener's optimization<sup>7</sup> of continuous filters. Finding the optimum system experimentally is the same as finding a minimum of a paraboloid in n dimensions. This could be done manually by having a human operator read the meter and set the adjustment, or it could be done automatically

# Dérivation par optimisation : règle de Widrow-Hoff

$$\ell(\mathbf{w}) = \frac{1}{2} e^2(t)$$

Méthode de gradient :

$$\frac{\partial \ell(\mathbf{w})}{\partial \mathbf{w}} = e(t) \frac{\partial e(t)}{\partial \mathbf{w}}$$

$$e(t) = u(t) - \mathbf{x}^\top(t) \mathbf{w}(t) \quad \text{d'où :} \quad \frac{\partial e(t)}{\partial \mathbf{w}(t)} = -\mathbf{x}(t)$$

$$\frac{\partial \ell(\mathbf{w})}{\partial \mathbf{w}(t)} = -\mathbf{x}(t) e(t)$$

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \mathbf{x}(t) e(t)$$

[Widrow-  
Hoff:60]

B. Widrow and M. Hoff. *Adaptive Switching Circuits*. IRE WESCON Conv. Rec. Pt.4, pp.96-104.

# Les pionniers : un bilan

- Démarche inspirée du traitement du signal et de l'adaptation
- **Pas de principe normatif** et générique sous-jacent
  - Mais ...
- Importance de **la représentation**
  - Quelle **expressivité** ?  
Comment réaliser des **fonctions** entre **entrées** et **sorties** ?
    - Fonctions de base
    - Combinaison (hiérarchique)
  - **Convergence** ?



# Et pendant ce temps ... la reconnaissance des formes

- S'intéresse à des processus sub-conscients de **perception**
- Adopte un **point de vue bayésien**

## Applications

- Reconnaissance de **caractères**
- Reconnaissance de la **parole**
- Reconnaissance de **gestes** (lecture sur les lèvres)
- Reconnaissance de **particules** (trajectoires dans les chambres à bulles)
- ...

```
.....  
..... DIMENSION IMACHL(2)  
20 ..... ACCEPT 31, I, J  
31 ..... FORMAT(215)  
..... IF[I] 79, 99, 40  
40 ..... IF[I-IMACHL] 50, 50, 60  
50 ..... IMACH[I]=J  
60 ..... GO TO 20  
99 ..... RETURN  
.....
```

# Et pendant ce temps ... la reconnaissance des formes

- On cherche à associer une décision à une forme d'entrée :  $\mathbf{x}_\ell \mapsto C_i$

- **Théorie de la décision bayésienne :**

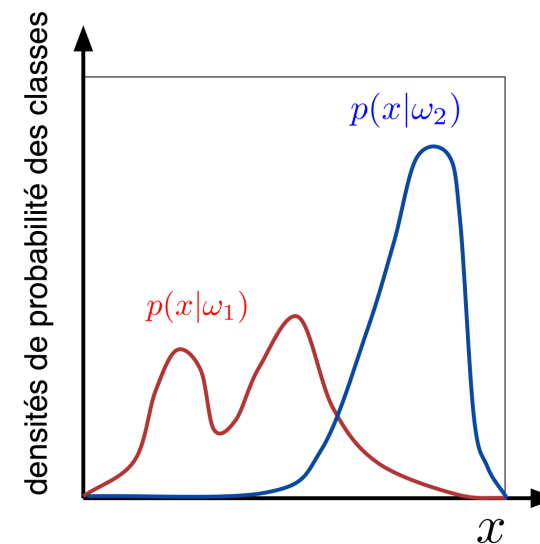
$$\begin{aligned} C^* &= \operatorname{Argmax}_{C_i \in \mathcal{C}} \mathbf{p}(C_i | \mathbf{x}) \\ &= \operatorname{Argmax}_{C_i \in \mathcal{C}} \frac{\mathbf{p}(\mathbf{x} | C_i) \cdot \mathbf{P}(C_i)}{\mathbf{p}(\mathbf{x})} \\ &= \operatorname{Argmax}_{C_i \in \mathcal{C}} \mathbf{p}(\mathbf{x} | C_i) \cdot \mathbf{P}(C_i) \end{aligned}$$

# La reconnaissance des formes : bilan

- Focalisation sur l'**apprentissage supervisé**
- **Apprentissage = estimation de paramètres**  $p(\mathbf{x}|C_i)$  et  $P(C_i)$ 
  - En général familles paramétrées distributions de probabilité conjuguées

- Un problème d'**optimisation**

- Pour faire les calculs, on suppose des **données i.i.d.**
- **Approche générative**



# Bilan (~ fin des années 60)

## ■ On a précisé **la tâche**

- Apprentissage supervisé
- Généralisation (*échantillon d'apprentissage ; échantillon de test*)

## ■ On cherche **des méthodes**

- De dérivation de règles d'apprentissage
  - E.g. Widrow-Hoff
  - Critère de décision optimale (Bayes)
  - Critères inductifs (MAP ; MLE)

## ■ Nouveaux présupposés (RF)

- Tout s'exprime en termes de **distribution de probabilités** sous-jacentes
- Données i.i.d.



# Bilan

- La reconnaissance de formes **ne permet pas** :
  - D'apprendre des **descriptions** (v.s. des règles de décision)  
[McCarthy, Stanford, 1971]
  - D'apprendre des **règles** d'un système expert
  - D'apprendre des **descriptions structurées**

---

# L'âge de raison

(~ 1970 - ~1984)

# La raison triomphante

---

- Le (1<sup>er</sup>) connexionnisme est mort

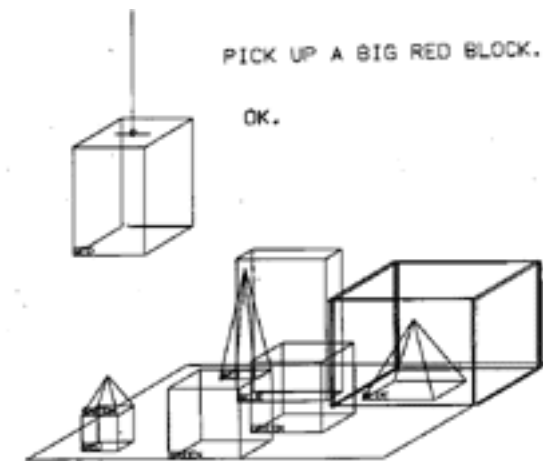
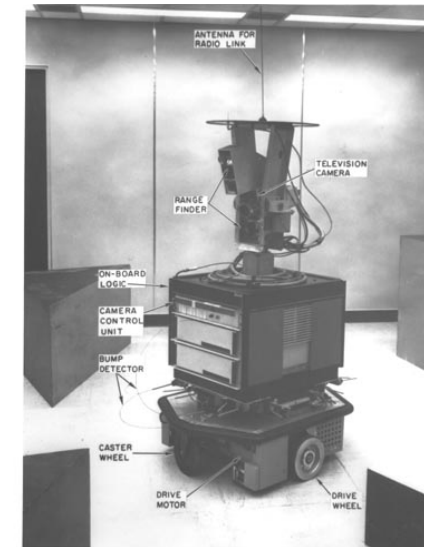
[« Perceptrons » Minsky & Papert, 1969]

- Et même les robots pensent

*The symbol system hypothesis* [Newell & Simon, 1972]

# Même les robots pensent : Shakey

- 1<sup>er</sup> robot mobile contrôlé par ordinateur.  
(Stanford Research Institute, 1967-1972)
  - **Vision** : Thèse de *David Waltz*  
(reconnaissance de polyèdres en 3D à partir d'une image 2D)
  - **Contrôle et planification** : STRIPS, ABSTRIPS (puis NOAH, ...)
  - **IHM** : SHRLDU [Thèse de *Terry Winograd*, MIT, 1968-1970]
  - **Apprentissage** : ARCH [Thèse de *Patrick Winston*, 1970, 1975]

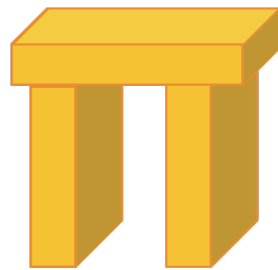




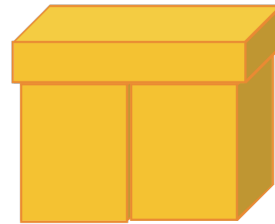
# ARCH [Winston, 1970]

---

- Apprentissage de concept (e.g. arche) dans un monde de blocs



(a)



(b)



(c)

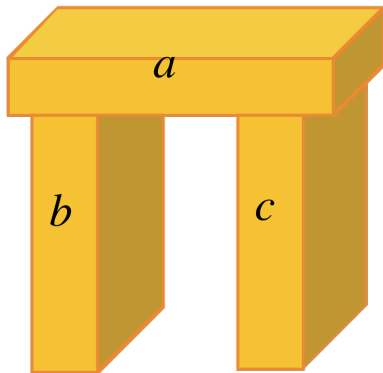


(d)

# ARCH [Winston, 1970]

---

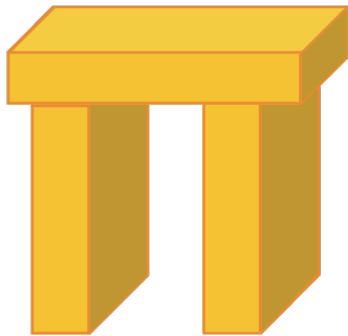
- **Apprentissage de concept** (e.g. arche) dans un monde de blocs



# ARCH [Winston, 1970]

---

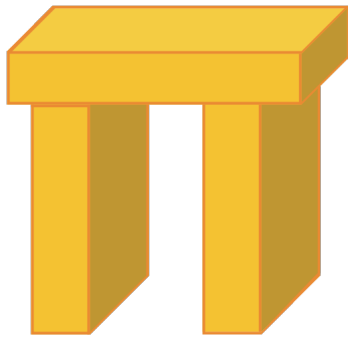
- Heuristique : « **require-link** »



# ARCH [Winston, 1970]

---

- Heuristique : « **forbid-link** »



# ARCH [Winston, 1970]

---

## ■ Caractéristiques

### – Cas « réalisable » : représentation adéquate

- Professeur  $\equiv$  Élève

### – Représentation différenciée + théorie

- Réseau sémantique
- Liens must et must-not

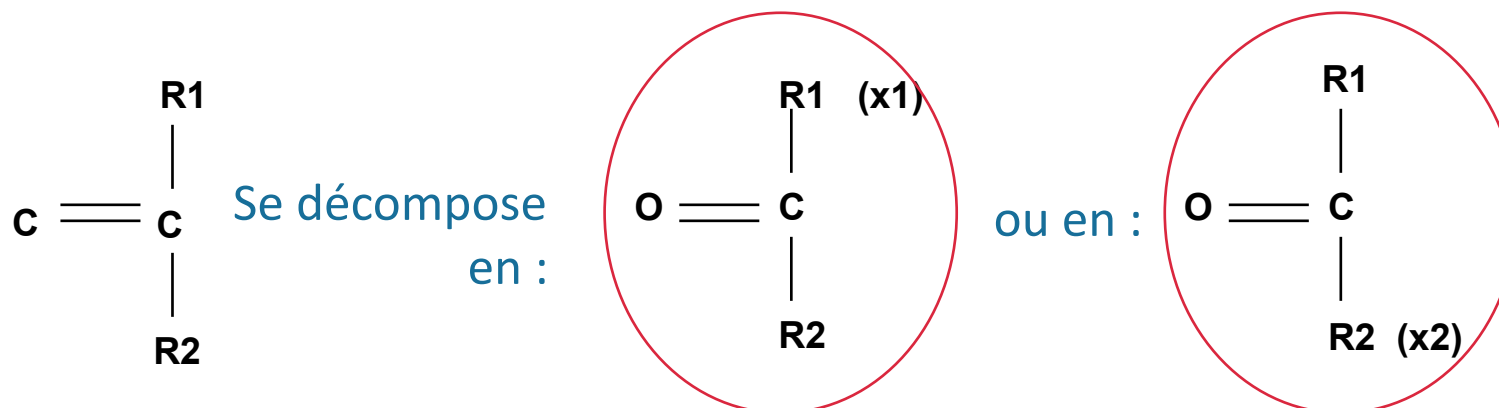
### – Apprentissage incrémental

- Séquentiel
- Constructif : ex + (*généralisation*) ; ex – (*spécialisation*)
- Séquence d'exemples bien choisie
  - Exemples négatifs <--> near-misses



# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

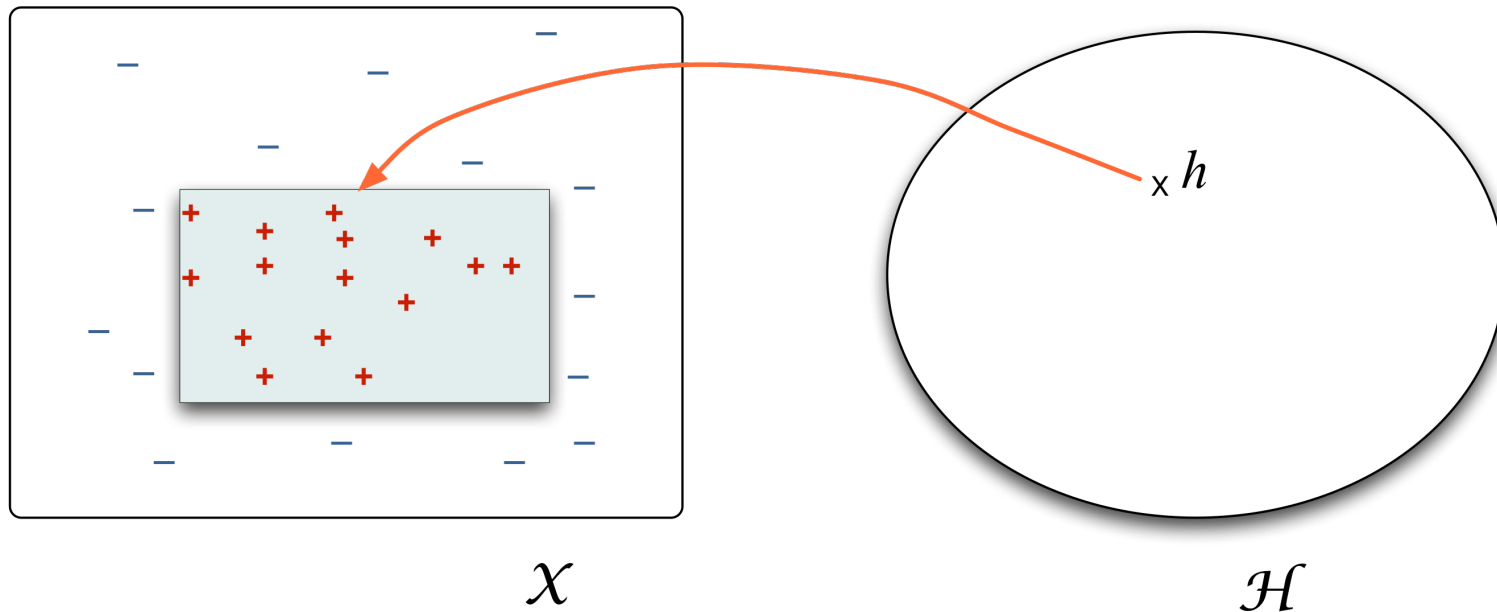
- **Apprentissage de règles** pour le système expert Meta-Dendral
  - Descriptions relationnelles de **sous-structures moléculaires** ayant probablement produit les fragments mesurés dans un spectrogramme de masse.



mais pas en ...

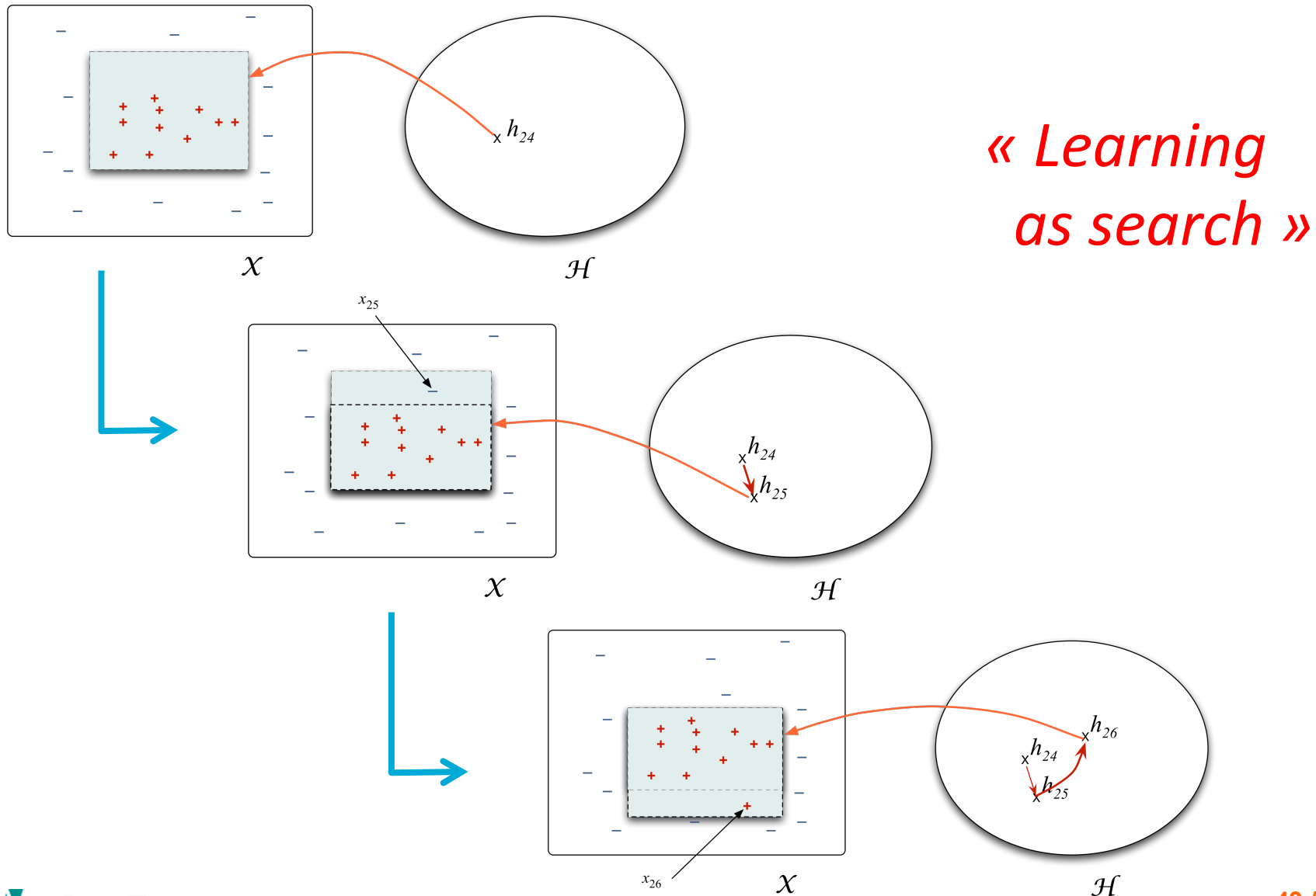
# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

- Introduit explicitement l'idée de **recherche dans un espace d'hypothèses**





# Apprentissage de l'espace des versions [Tom Mitchell, 1979]



« Learning as search »

# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

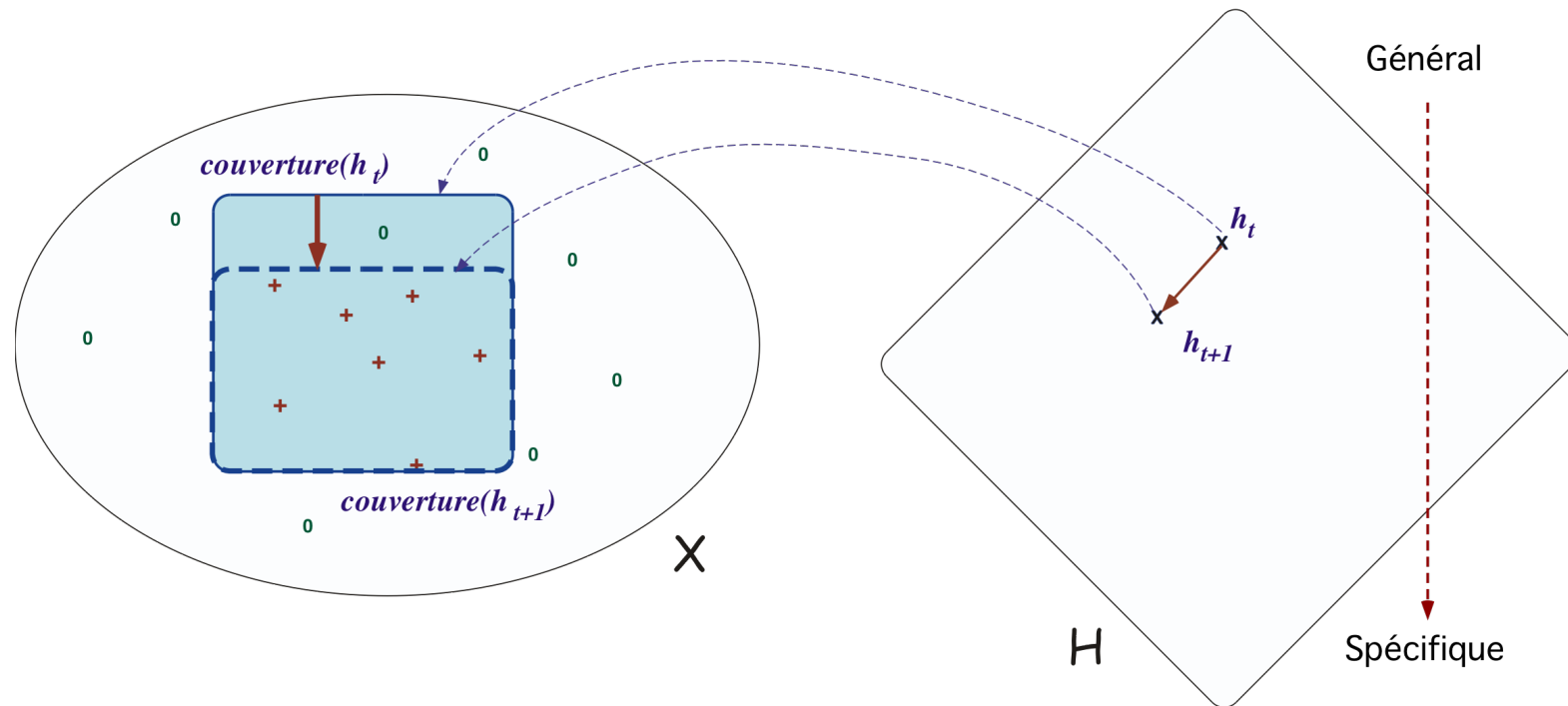


Figure: La relation d'inclusion dans  $\mathcal{X}$  induit la relation de généralisation dans  $\mathcal{H}$ . Ici,  $h_{t+1} \preceq h_t$ .

# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

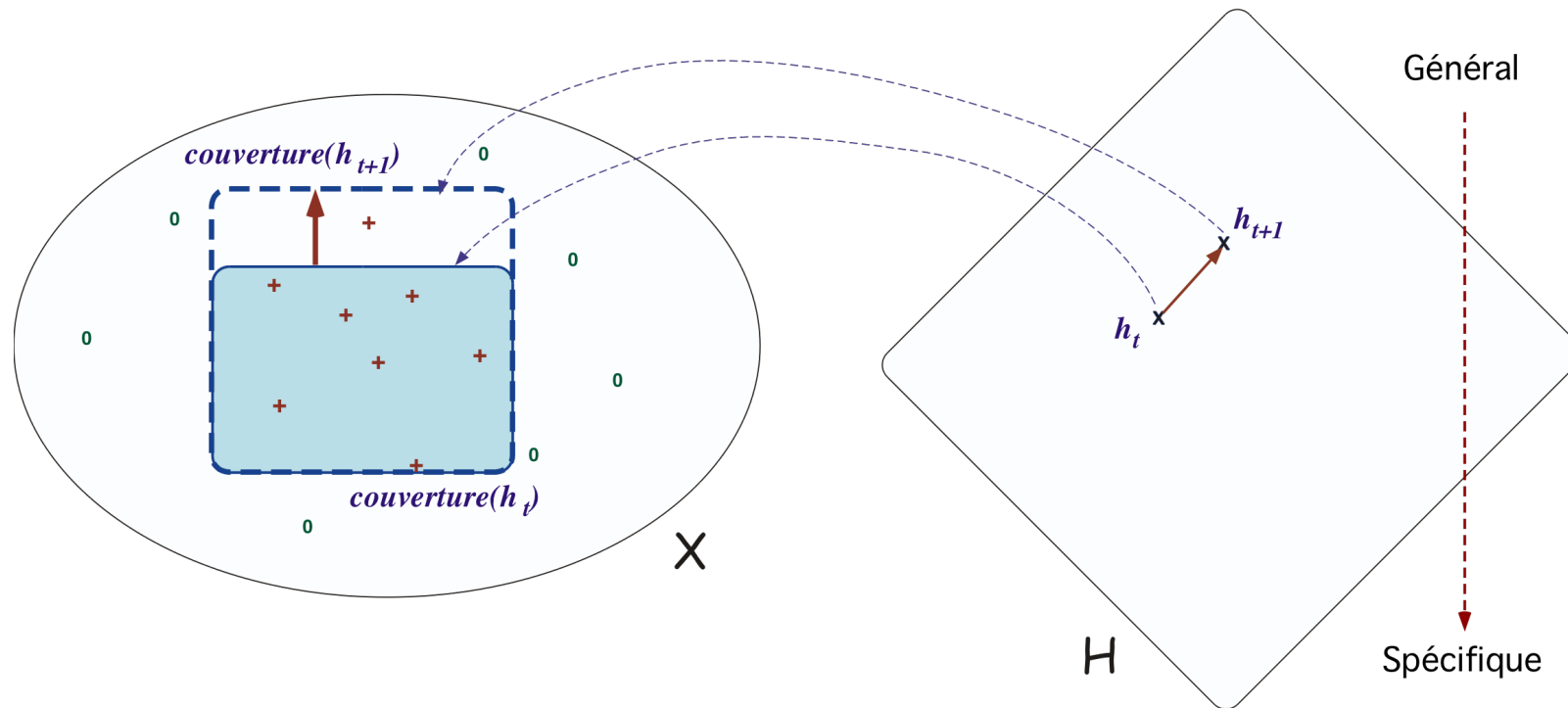


Figure: La relation d'inclusion dans  $\mathcal{X}$  induit la relation de généralisation dans  $\mathcal{H}$ . Ici,  $h_{t+1} \succeq h_t$ .

# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

## ■ Opérateurs de **généralisation** (spécialisation)

### – Abandon de conjonction

•  $A \& B \rightarrow C \quad \Rightarrow \quad A \rightarrow C$

### – Ajout d'alternative

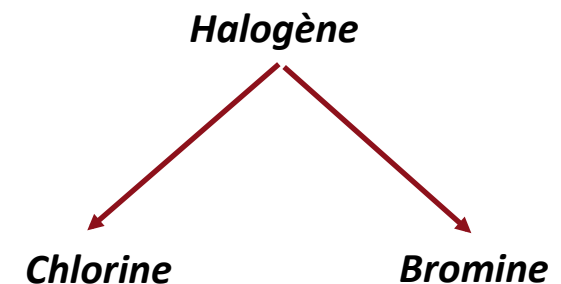
•  $A \text{ ou } B \rightarrow C \quad \Rightarrow \quad A \text{ ou } B \text{ ou } D \rightarrow C$

### – Ascension dans une hiérarchie de concepts

•  $\text{corrosif \& bromine} \rightarrow \text{toxique}$   
 $\Rightarrow \text{corrosif \& halogène} \rightarrow \text{toxique}$

### – Inversion de la résolution

– ...



# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

---

## ■ Opérateurs de **généralisation / spécialisation**

### – Généralisation

- Transforme une **description** en une **description plus générale** (au sens de l'inclusion dans  $\mathcal{X}$ )
- (*Souvent équivalent à produire une **conséquence logique** de la description initiale*)

### – Spécialisation

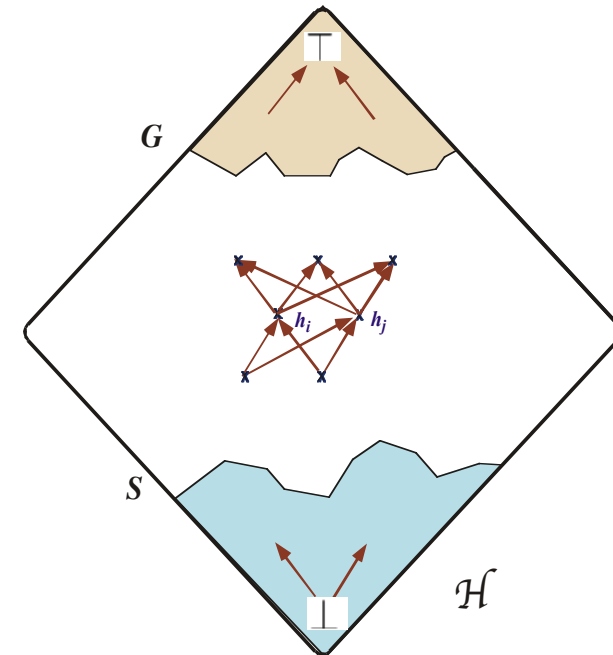
- Duale

# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

## Observation fondamentale :

L'espace des versions structuré par une relation d'ordre partiel peut être représenté par :

- sa **borne supérieure** : le *G-set*
- sa **borne inférieure** : le *S-set*



- *G-set* = Ensemble de toutes les hypothèses **les plus générales** cohérentes avec les exemples connus
- *S-set* = Ensemble de toutes les hypothèses **les plus spécifiques** cohérentes avec les exemples connus

# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

---

## Apprentissage

*... par mise à jour de l'espace des versions*

Idée :

maintenir le S-set

et le G-set

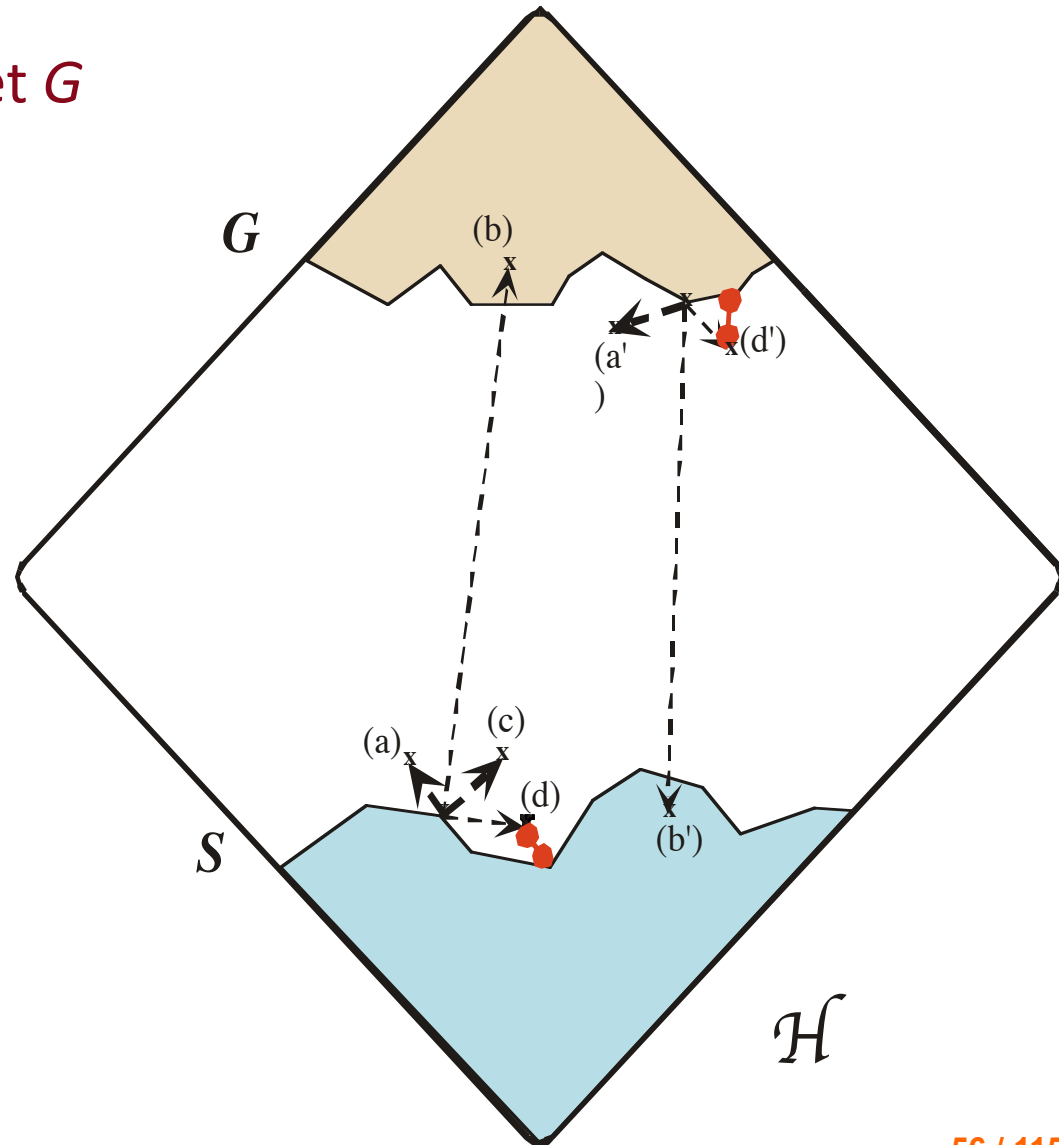
après chaque nouvel exemple



*Algorithme d'élimination des candidats*

# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

Mise à jour des bornes  $S$  et  $G$





# Apprentissage de l'espace des versions [Tom Mitchell, 1979]

## Algorithme 3 : Algorithme d'élimination des candidats.

**Résultat :** Initialiser  $G$  comme l'hypothèse la plus générale de  $\mathcal{H}$

Initialiser  $S$  comme l'hypothèse la moins générale de  $\mathcal{H}$

**pour chaque** *exemple*  $x$  **faire**

si  $x$  est un exemple positif **alors**

Enlever de  $G$  toutes les hypothèses qui ne couvrent pas  $x$

pour chaque hypothèse  $s$  de  $S$  qui ne couvre pas  $x$  **faire**

Enlever  $s$  de  $S$

**Généraliser**( $s, x, S$ )

c'est-à-dire : ajouter à  $S$  toutes les généralisations minimales  $h$  de  $s$  telles que :

- $h$  couvre  $x$  et
- il existe dans  $G$  un élément plus général que  $h$

Enlever de  $S$  toute hypothèse plus générale qu'une autre hypothèse de  $S$

**fin**

sinon

Enlever de  $S$  toutes les hypothèses qui couvrent  $x$

pour chaque hypothèse  $g$  de  $G$  qui couvre  $x$  **faire**

Enlever  $g$  de  $G$

**Spécialiser**( $g, x, G$ )

c'est-à-dire : ajouter à  $G$  toutes les spécialisations maximales  $h$  de  $g$  telles que :

- $h$  ne couvre pas  $x$  et
- il existe dans  $S$  un élément plus spécifique que  $h$

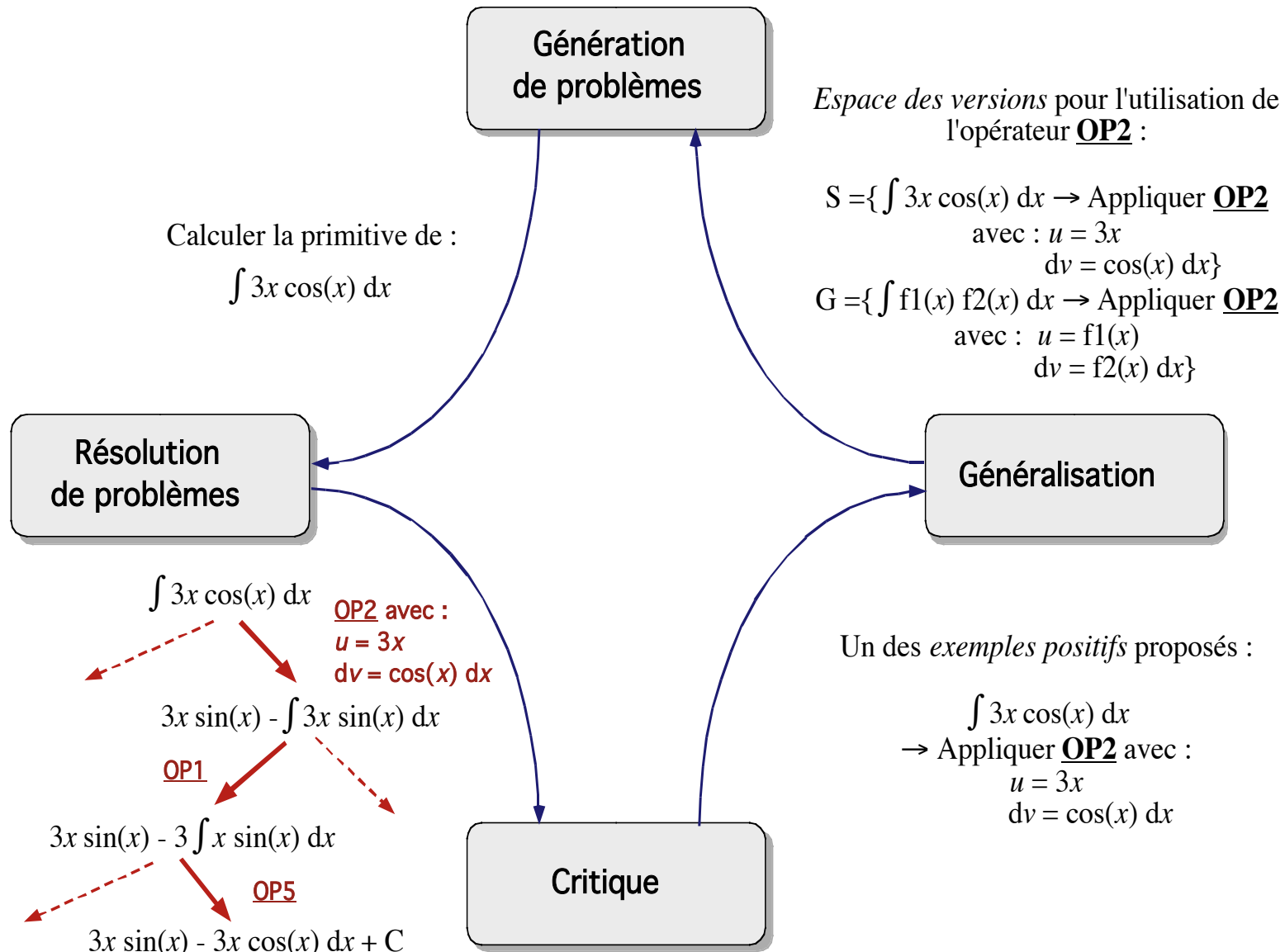
Enlever de  $G$  toute hypothèse plus spécifique qu'une autre hypothèse de  $G$

**fin**

**fin**

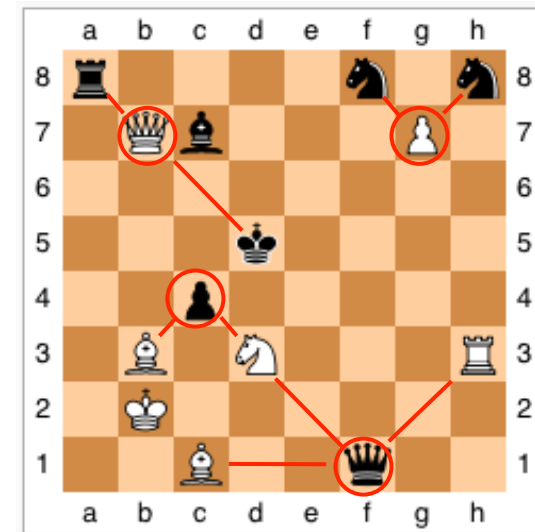
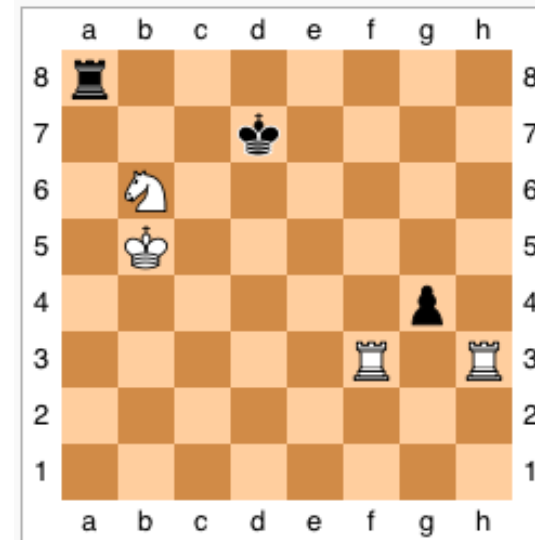
**fin**

# Le système LEX [Tom Mitchell, 1983]



# Explanation-Based Learning

1. Un exemple unique
2. Recherche de la preuve de la « fourchette »
3. Généralisation



# Explanation-Based Learning

Ex : **apprendre le concept** `empilable(Objet1, Objet2)`

## ■ Théorie :

(T1) : `poids(X, W) :- volume(X, V), densité(X, D), W is V*D.`

(T2) : `poids(X, 50) :- est-un(X, table).`

(T3) : `plus-léger(X, Y) :- poids(X, W1), poids(X, W2), W1 < W2.`

## ■ Contrainte d'opérationalité :

- Concept à exprimer à l'aide des prédicats *volume, densité, couleur, ...*

## ■ Exemple positif (solution) :

`sur(obj1, obj2).`

`est_un(objet1, boîte).`

`est_un(objet2, table).`

`couleur(objet1, rouge).`

`couleur(objet2, bleu).`

`matériau(objet2, bois).`

`volume(objet1, 1).`

`volume(objet2, 0.1).`

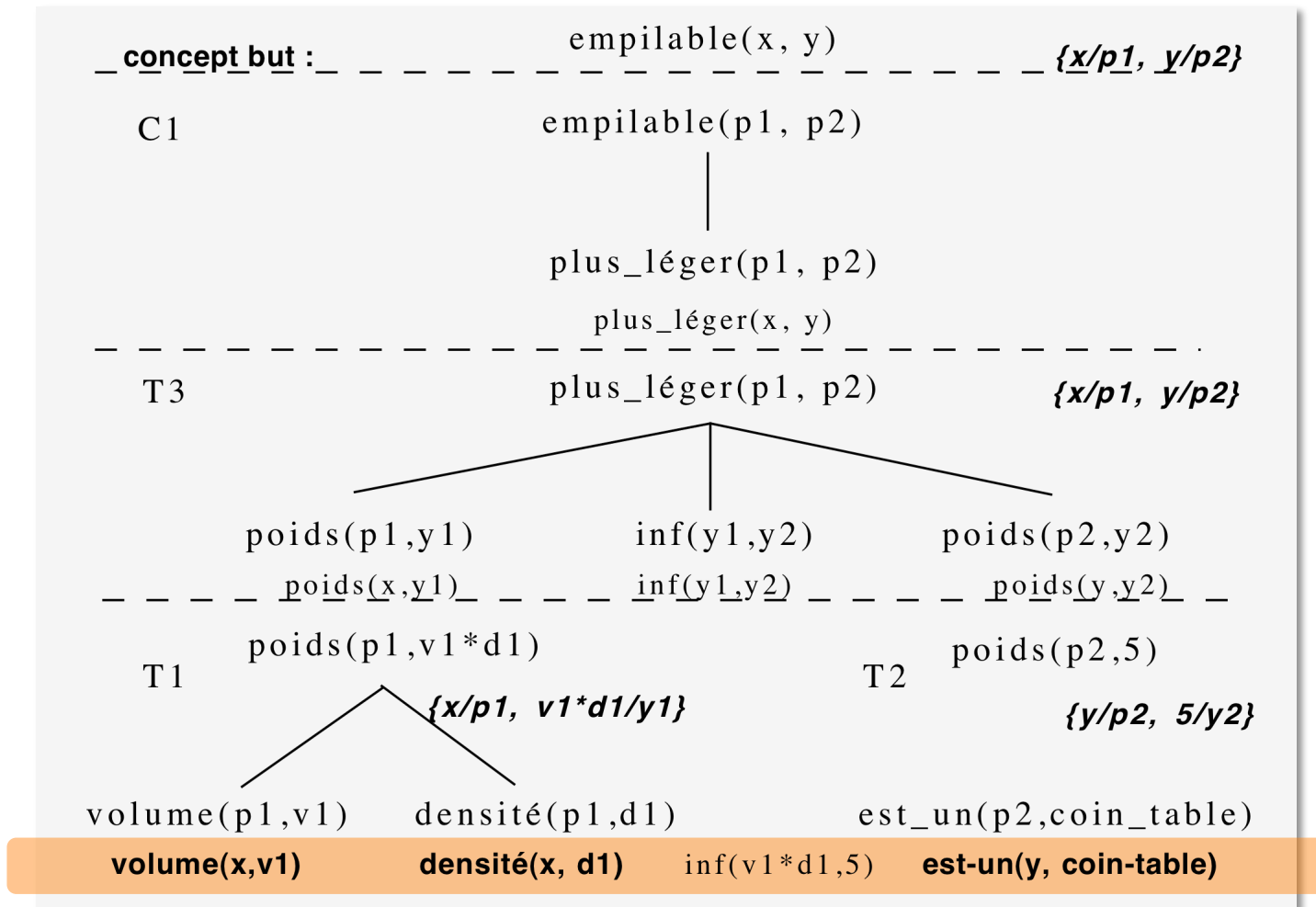
`propriétaire(objet1, frederic).`

`densité(objet1, 0.3).`

`matériau(objet1, carton).`

`propriétaire(objet2, marc).`

# Explanation-Based Learning



Arbre de preuve généralisé obtenu par **régression du concept cible dans l'arbre de preuve** en calculant à chaque étape les littéraux les plus généraux permettant cette étape.

# Explanation-Based Learning

---

- Induction à **partir d'un seul exemple**
  - ... et d'une **théorie forte du domaine**
- Langage de la logique
- **Opérateurs** de raisonnement (déduction, ...)
  
- *Maintenant utilisées dans les « solveurs » de problèmes SAT.*

# L'« âge de raison » : bilan

## ■ Modélisation « cognitive »

- élève = professeur (même appareil cognitif)
- Concepts des sciences cognitives
  - **Représentation** logique / réseau sémantique
  - Mémoire *court-terme / long-terme / de travail*
  - Connaissances *procédurales vs. déclaratives*
  - **Raisonnement** : induction ; déduction ; analogie ; ...

## ■ Apprentissage de connaissances structurées / théories

- EBL
- SOAR / Chunking
- PRODIGY

# L'« âge de raison » : bilan

## ■ Algorithmes

1. Manipulations de **représentations**
2. Par des **opérateurs**
  - *spécialisation ; généralisation ; reformulation*
3. **Satisfaction de contraintes**
  - posées par les exemples
4. Très inspirés et en dialogue avec les **sciences cognitives**



# L'« âge de raison » : bilan

## ■ Limites

- Requiert une **forte théorie du domaine**
  - Difficulté pour l'acquérir 🙄
  - Suspicion de manque de généralité (ad hoc) 🙄
  - Mais se contente de peu d'exemples 😊
- Pas adapté à des **données bruitées**
  - Cas « non réalisable »
- Le « **passage à l'échelle** » n'est pas évident

# L'« âge de raison » : âge adulte ?

---

## ■ Orientation vers les **applications**

### – Bases d'exemples

- De **moins en moins structurés**
- Incrémental -> Traitement **batch**

### – Performances

- Explication / compréhension -> **Taux d'erreur**



Mécanisme d'apprentissage -> **optimisation**

---

**Double coup de butoir**  
**Et changement de perspective**

(~ 1984 - ~1995)

# Deux nouveautés

---

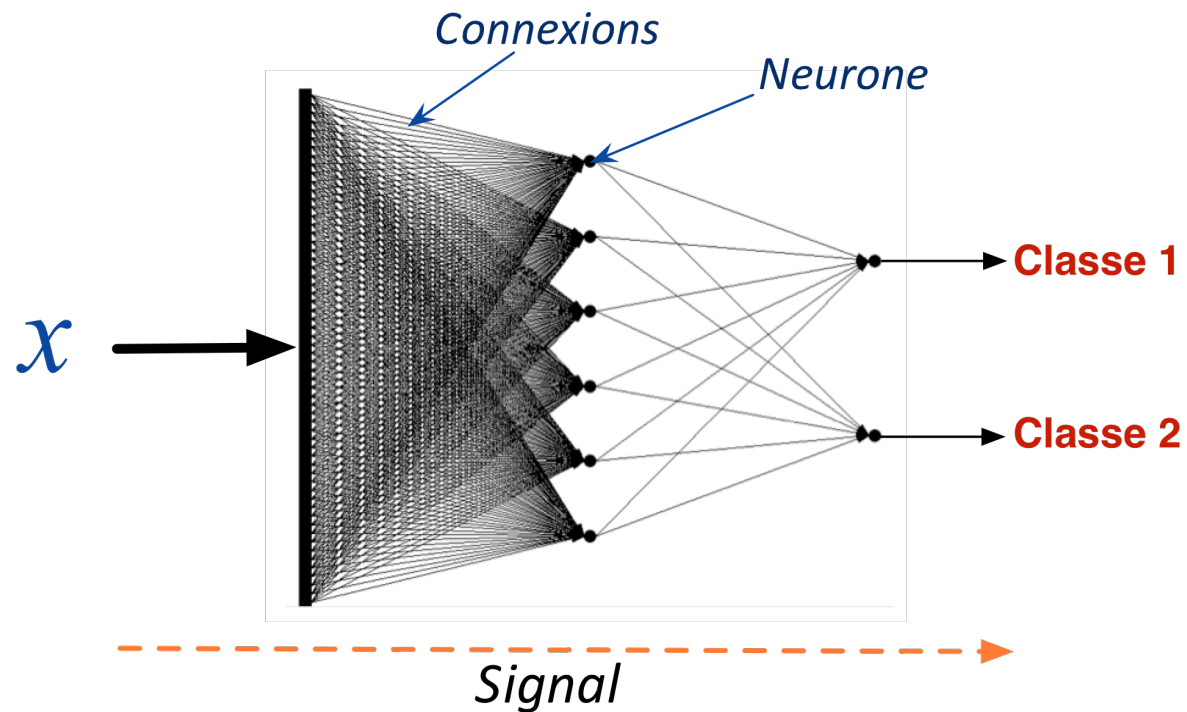
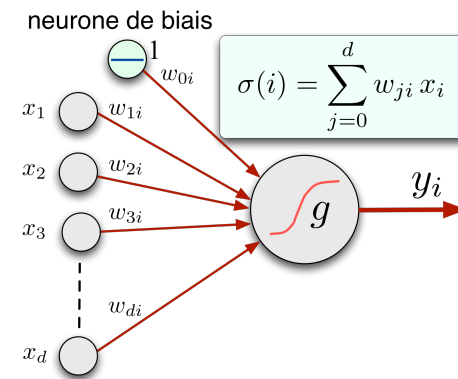
- La **théorie PAC** de l'apprentissage

- [Valiant, 1984]

- L'avènement du **2° connexionnisme**

- (Parallel Distributed Processing, [McLelland & Rumelhart, 1986])

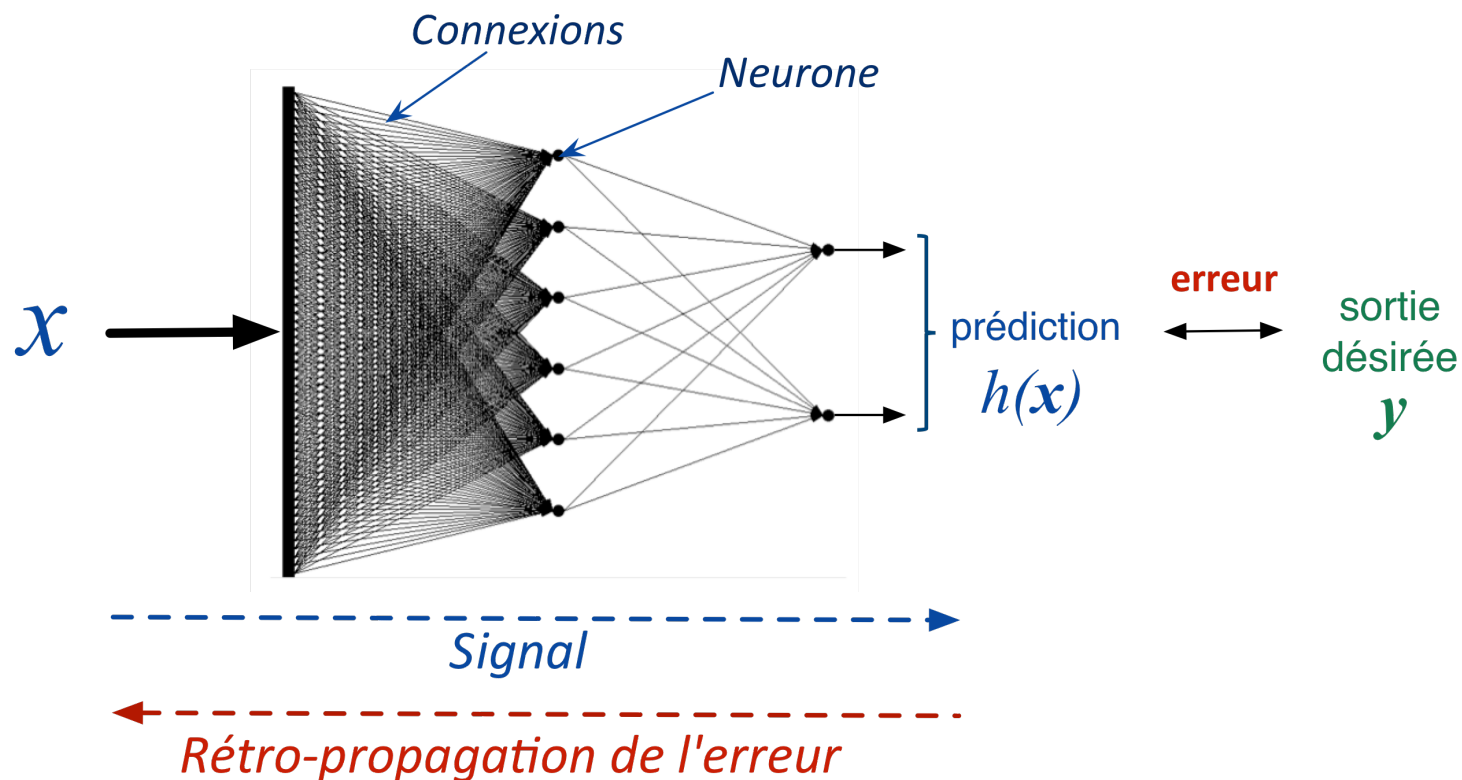
# Le 2<sup>ème</sup> connexionnisme



# Le 2<sup>ème</sup> connexionnisme

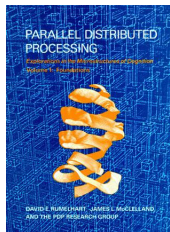
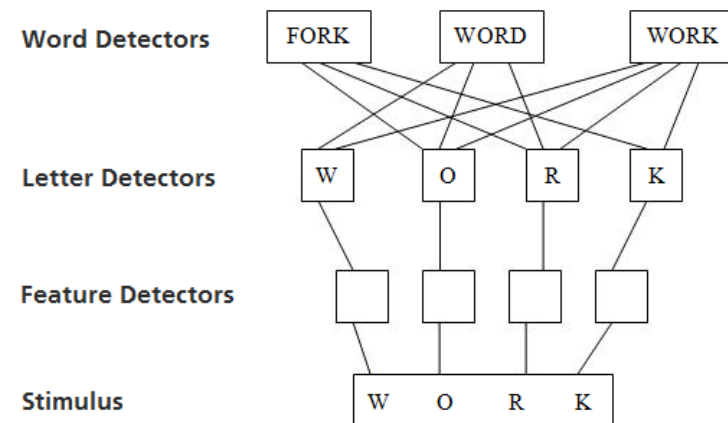
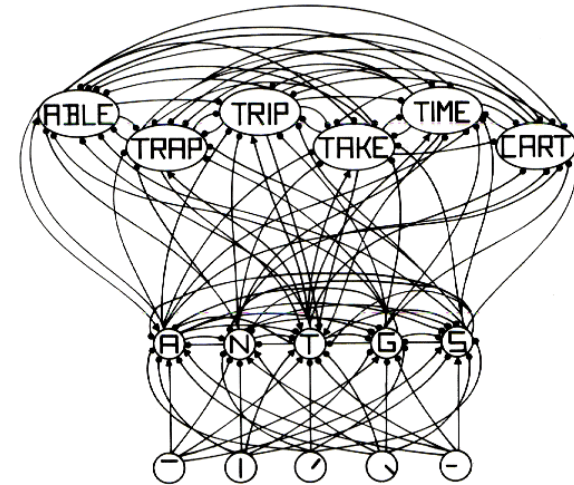
## ■ Questions :

- Comment apprendre les **paramètres** (poids des connexions) ?
- Comment déterminer l'**architecture** du réseau ?



# Le (premier) 2<sup>ème</sup> connexionnisme

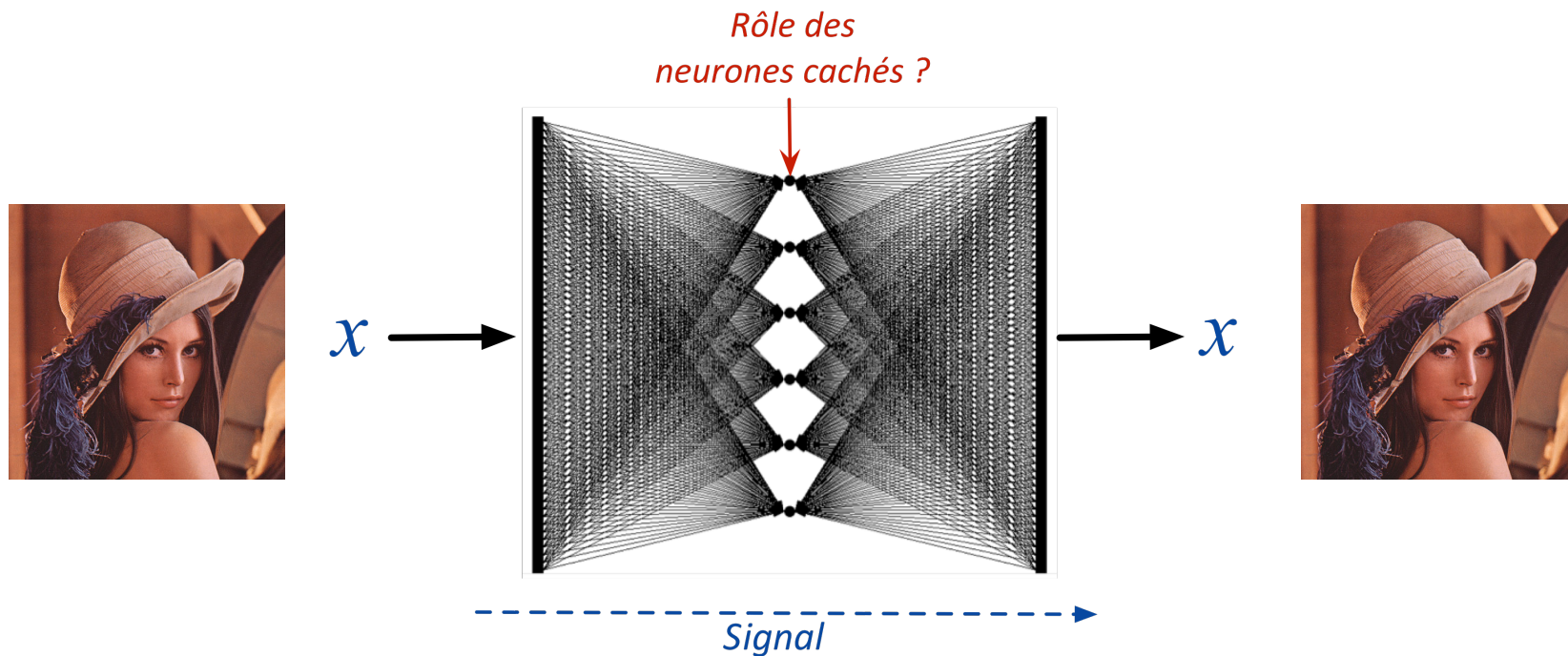
- « ... *discover appropriate representations for a given task ...* »
- « ... *good set of underlying features ...* »
- « ... *Widrow-Hoff rule is a learning rule which is designed to capture second order structure. It is therefore limited to easy learning ...* »
- « ... *designing feature detectors ...* »



[Rumelhart & McClelland,  
« Parallel Distributed Processing »,  
MIT Press, 1986]

# Le 2<sup>ème</sup> connexionnisme : **une question de représentation**

- Les questions à ce moment là (1985-1990)
  - **Quelle représentation** (variables latentes) ?
  - **Comment choisir l'architecture** ?





# Nouveau concept : le(s) biais

---

- **Biais de représentation**

- Langage de représentation des hypothèses
- Espace des hypothèses

- **Biais de recherche**

- Exploration de l'espace de recherche

- **Questions**

- **Choix** du (des) biais
- **Quantification** du biais

# Le 2<sup>ème</sup> connexionnisme : un problème d'optimisation

## ■ Algorithme de **rétro-propagation de gradient**

$$\frac{\partial E^l}{\partial w_{ij}}$$

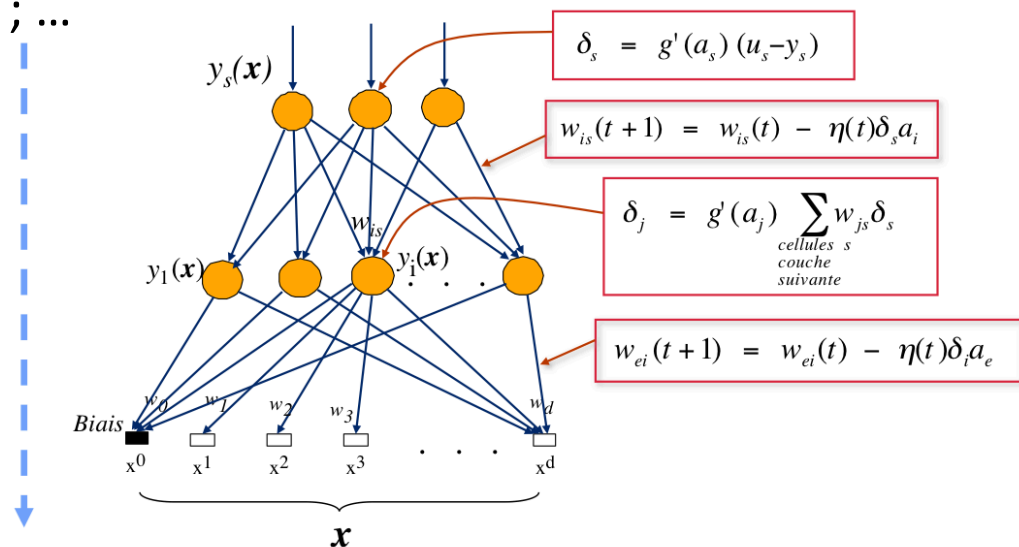
– Algorithme **itératif**

- Gradient stochastique ou total

– **Local**

– Valide pour **tout type d'apprentissage supervisé**

- Classification ; régression ; ...
- Toute mesure d'erreur



# Le 2<sup>ème</sup> connexionnisme : l'architecture

---

- Les questions à ce moment là (1985-1990)
  - Quelle représentation (variables latentes) ?
  - Comment **choisir l'architecture** ?
  
- Si architecture trop pauvre
  - **Mauvaise performance** en apprentissage et en généralisation
  
- Si architecture trop « riche »
  - Bonne performance en *apprentissage*
  - Mauvaise performance en *généralisation*
  - > **Sur-apprentissage**



# L'analyse style « PAC learning »

---

- Mesure le **lien** entre **risque empirique** et **risque réel**
- Plus précisément :
  - En fonction du biais inductif de  $\mathcal{H}$ , *quelle est la probabilité de sélectionner une hypothèse mauvaise* (risque réel  $> \varepsilon$ ) **alors que la performance apparente est bonne** (risque empirique = 0) ?

# L'analyse « PAC learning »

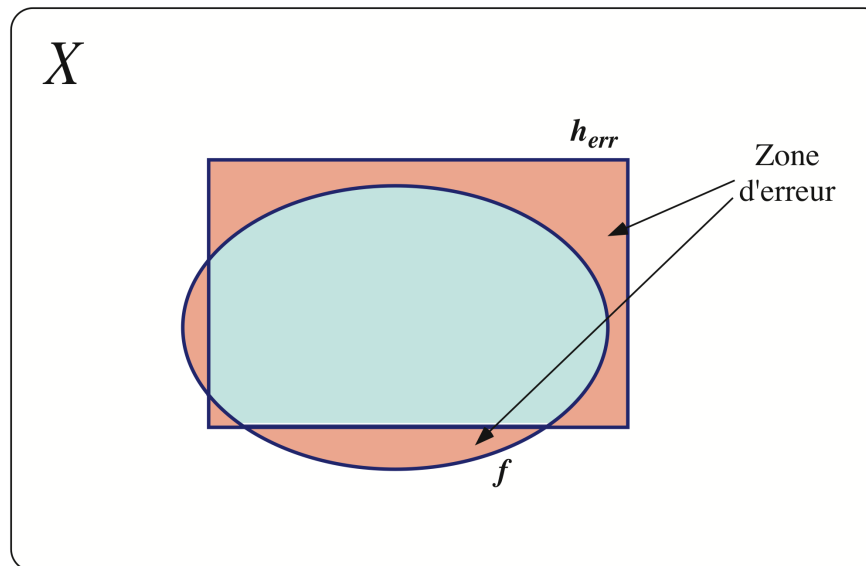
---

## ■ Supposons

- Le **concept cible**  $f \in \mathcal{H}$
- L'**espace des concepts** est de taille finie :  $|\mathcal{H}| < \infty$
- Données **non bruitées**
- **Apprentissage de concept** (une classe vs. tout le reste)
- **Exemples** tirés **i.i.d.** selon une loi de distribution  $P_X$

# L'analyse « PAC learning »

- Puisque  $f \in \mathcal{H}$ , à tout instant il existe **au moins une hypothèse** dans l'espace des versions (**d'erreur nulle**)
  - Je choisiss(\*) une hypothèse apparemment sans erreur :  $h_{err}$
  - La probabilité d'erreur de  $h_{err}$  est égale à la probabilité de tirer un exemple dans la zone d'erreur (différence entre  $f$  et  $h_{err}$ )



# L'analyse « PAC learning »

- Quelle est la probabilité que je choisisse **une hypothèse  $h_{\text{err}}$  de risque réel  $> \varepsilon$**  **et que je ne m'en aperçoive pas** après l'observation de  $m$  exemples ?

- Probabilité de survie de  $h_{\text{err}}$  **après 1 exemple** :  $(1 - \varepsilon)$

- Probabilité de survie de  $h_{\text{err}}$  **après  $m$  exemples** :  $(1 - \varepsilon)^m$

- Probabilité de survie d'**au moins une hypothèse dans  $\mathcal{H}$**  :  $|\mathcal{H}| (1 - \varepsilon)^m$

– On utilise la probabilité de l'union  $\mathbf{P}(A \cup B) \leq \mathbf{P}(A) + \mathbf{P}(B)$

- On veut que la **probabilité qu'il reste au moins une hypothèse de risque réel  $> \varepsilon$**  dans l'espace des versions soit **bornée par  $\delta$**  :

$$|\mathcal{H}| (1 - \varepsilon)^m < |\mathcal{H}| e^{(-\varepsilon m)} < \delta$$

$$\log |\mathcal{H}| - \varepsilon m < \log \delta$$

$$m > \frac{1}{\varepsilon} \log \frac{|\mathcal{H}|}{\delta}$$



# L'analyse « PAC learning »

---

- On arrive à :

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

# L'analyse « PAC learning »

---

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

- *Nouveau critère inductif :*

- Le **risque empirique régularisé**

1. Satisfaire les contraintes posées par les **exemples**
2. Choisir le meilleur **espace d'hypothèses** (capacité de H)

# L'analyse « PAC learning »



- On arrive à :

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

- *Remarques :*

- **Analyse dans le pire cas**
  - Contre toute distribution des exemples  $P_X$
  - Vrai pour toute hypothèse
- Fait intervenir
  - la « **richesse** » de l'espace des hypothèses
  - Le **nombre d'exemples** (tous d'égale importance (i.i.d.))



# L'analyse « PAC learning »

## ■ **Avant** : motivation de Leslie Valiant

- Montrer que la **classe des concepts apprenables** (correspondant à des **classes de représentations logiques** (e.g.  $k$ -DNF)) est **non vide mais limitée**
- D'où **nécessité d'un apprentissage cumulatif, hiérarchique et guidé**

## ■ **Après**

- Apprentissage **batch** à partir d'**exemples tirés i.i.d.**
- **Algorithme d'apprentissage** escamoté (**optimisation** « magique »)
- Espace de **concepts** -> espace de **fonctions**
- **Biais** -> mesure de « **capacité** » de  $\mathcal{H}$ 
  - E.g. dimension de Vapnik-Chervonenkis

# Le 2<sup>ème</sup> connexionnisme

## ■ Avant :

- Quelle **représentation** des connaissances ?
- Quel processus (**algorithme**) d'apprentissage ?
- **Limites : difficile et peut paraître ad hoc**

## ■ Après

- Le système construit lui-même les **descripteurs intermédiaires** nécessaires (opacité)
- Apprentissage = **descente de gradient**
- Problèmes :
  - Choix de l'architecture :  $\mathcal{H}$
  - Optima locaux

# Des glissements progressifs ...

## ■ ... qui finissent par **tout changer**

### – L'algorithme d'apprentissage

- Reposant sur des *raisonnements*
- > Devient un **algorithme d'optimisation** omnipotent
  - Parfait
  - Tous usages

### – L'espace des concepts

- Associé à un *langage de représentation*
- > Devient un **espace de fonctions**
  - Dont la seule structure est celle mesurée par le biais (e.g.  $d_{VC}$ )

### – L'apprentissage

- Séquentiel et *incrémental* (d'une théorie)
- > Devient **apprentissage batch**, à partir d'exemples i.i.d.

---

# Un paradigme triomphant

(~ 1995 - ~2012)

# Nouvelle perspective

## ■ Poser un problème d'apprentissage, c'est :

1. L'exprimer sous forme d'**un critère inductif** à optimiser

- **Risque empirique**

- avec une **fonction d'erreur** adéquate

- Un **terme de régularisation**

- exprimant les contraintes

- et connaissances a priori

$$h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} \left[ R_{\text{Emp}}(h) + \lambda \text{reg}(h) \right]$$

2. Trouver un **algorithme d'optimisation** adapté



# Nouvelles idées et adaptation au/du paradigme

---

- Le **boosting**
- Les **séparateurs à Vastes Marges** et les **méthodes à noyaux**

# Le boosting

---

## ■ Source

- Une question : *l'apprentissage « faible » peut-il être « fort » ?*
- Réponse : **oui !!** [Shapire, 1989, 1990]

## ■ Procédés et esprit de la preuve (constructive)

- Algorithme par filtrage [Shapire, 1989, 1990]
- Algorithme en-ligne « weighted majority » [Freund, 1990]

# Le boosting : illustration

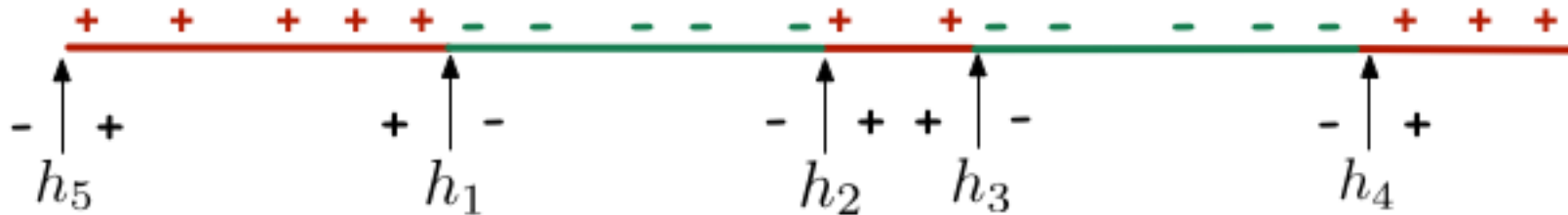


*Et si je pouvais combiner avec un autre séparateur linéaire ? Ou même plusieurs autres !*

Par exemple en utilisant un **vote pondéré** :

$$H(\mathbf{x}) = \text{sign} \left\{ \sum_{i=1}^l \alpha_i h_i(\mathbf{x}) \right\}$$

# Le boosting : illustration



$$H(x) = \text{sign}\{ 0.549 h_1(x) + 0.347 h_2(x) + 0.310 h_3(x) + 0.406 h_4(x) + 0.503 h_5(x) \}$$

- Comment arriver à ce genre de combinaison ?

**Algorithme du boosting**

# L'algorithme AdaBoost

---

- Algorithme itératif glouton

$$\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right) > 0$$

$$D_{t+1} = \frac{D_t}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$

→  $H(\mathbf{x}) = \text{sign} \left\{ \sum_{i=1}^T \alpha_t h_t(\mathbf{x}) \right\}$

# Le boosting s'inscrit dans le paradigme

## ■ Re-dérivation du boosting

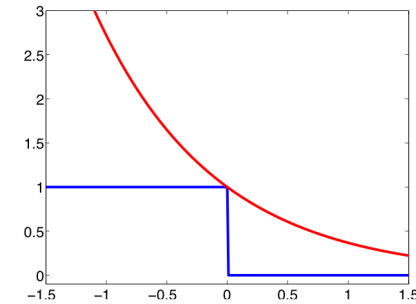
- En choisissant une *fonction de perte surrogée* de forme exponentielle

Soit :  $H_{T-1} = \alpha_1 h_1(\mathbf{x}) + \alpha_2 h_2(\mathbf{x}) + \dots + \alpha_{T-1} h_{T-1}(\mathbf{x})$

On veut ajouter :  $\alpha_T h_T(\mathbf{x})$

$$\begin{aligned}
 R_{\text{Emp}}(H_T) &= \sum_{i=1}^m e^{-y_i [H_{T-1}(\mathbf{x}) + \alpha_T h_T(\mathbf{x})]} \\
 &= \sum_{i=1}^m e^{-y_i H_{T-1}(\mathbf{x})} \cdot e^{-\alpha_T h_T(\mathbf{x})} \\
 &= \sum_{i=1}^m \underbrace{W_{T-1}(\mathbf{x}_i)}_{\text{Poids de } \mathbf{x}_i \text{ à } T-1} \cdot \underbrace{e^{-\alpha_T h_T(\mathbf{x})}}_{\text{à optimiser}}
 \end{aligned}$$

$$\frac{\partial R_{\text{Emp}}(H_T)}{\partial \alpha} \propto e^{-\alpha} \underbrace{(1 - \varepsilon_T)}_{\text{poids des exemples correctement prédits}} + e^{\alpha} \underbrace{\varepsilon_T}_{\text{poids des exemples incorrectement prédits}}$$

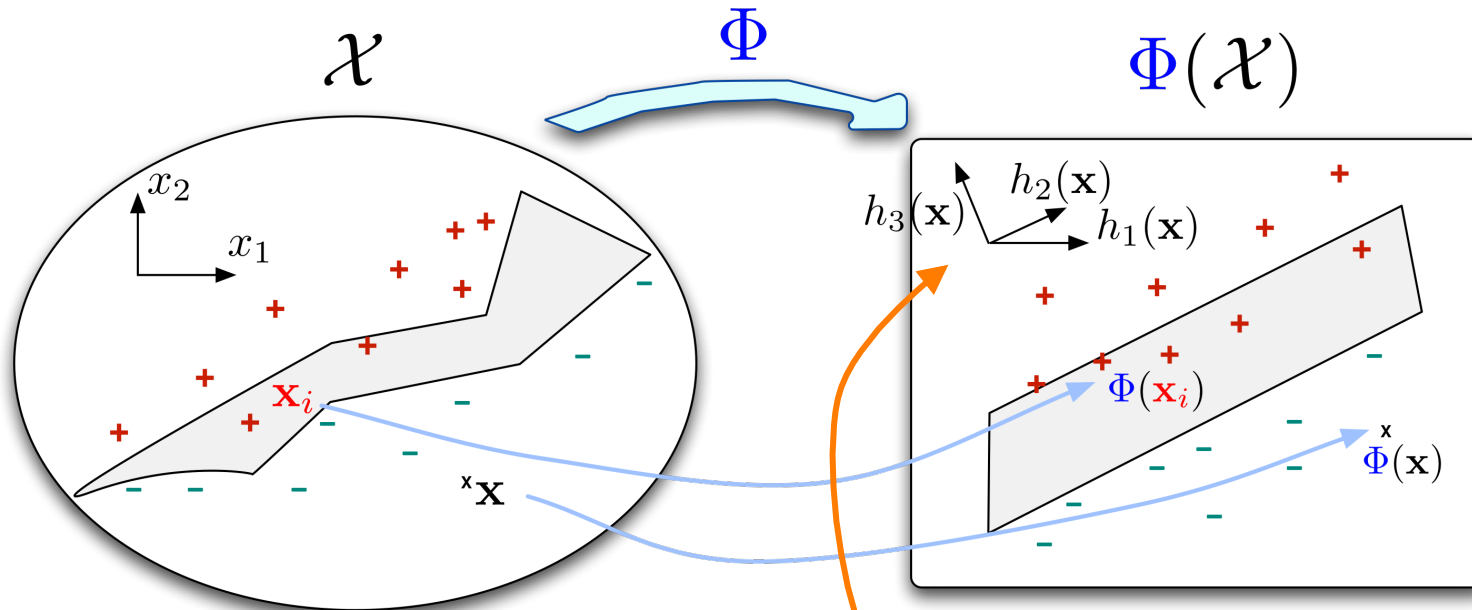


$$l(h(\mathbf{x}), y) = e^{-y \cdot h(\mathbf{x})}$$

Algorithme de gradient conjugué

$$\alpha_T = \frac{1}{2} \log \frac{1 - \varepsilon_T}{\varepsilon_T}$$

# Boosting et redescription

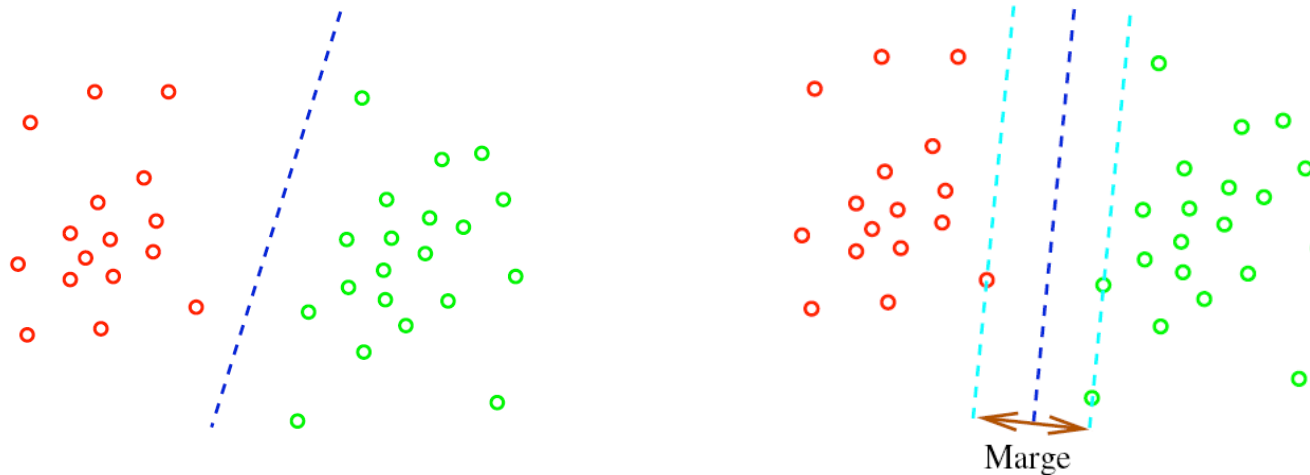


$$H(\mathbf{x}) = \text{sign} \left\{ \sum_{i=1}^T \alpha_i h_i(\mathbf{x}) \right\}$$

- Construction **itérative** de l'espace de redescription

# SVM et méthodes à noyaux

## ■ Séparateur linéaire à plus **V**aste **M**arge



- Plus robuste à variations de l'échantillon d'apprentissage
- Validé par analyse théorique
  - bornes de convergence fonction de la marge

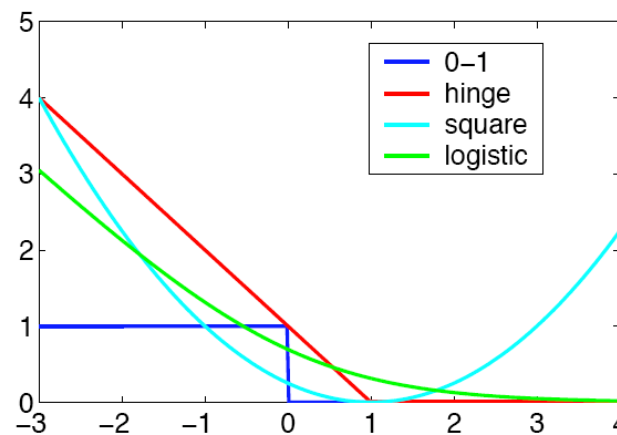


# SVM et méthodes à noyaux

- La recherche de la marge maximale conduit au **critère** :

$$\mathbf{w}^* = \underset{\mathbf{w}}{\text{ArgMin}} \left[ \underbrace{\sum_{i=1}^m |1 - y_i h(\mathbf{x}_i)|_+}_{\text{Risque empirique}} + \underbrace{\frac{1}{2} \mathbf{w}^\top \mathbf{w}}_{\text{Marge}} \right]$$

Fonction de *perte de substitution* (surrogate loss)



# SVM et méthodes à noyaux

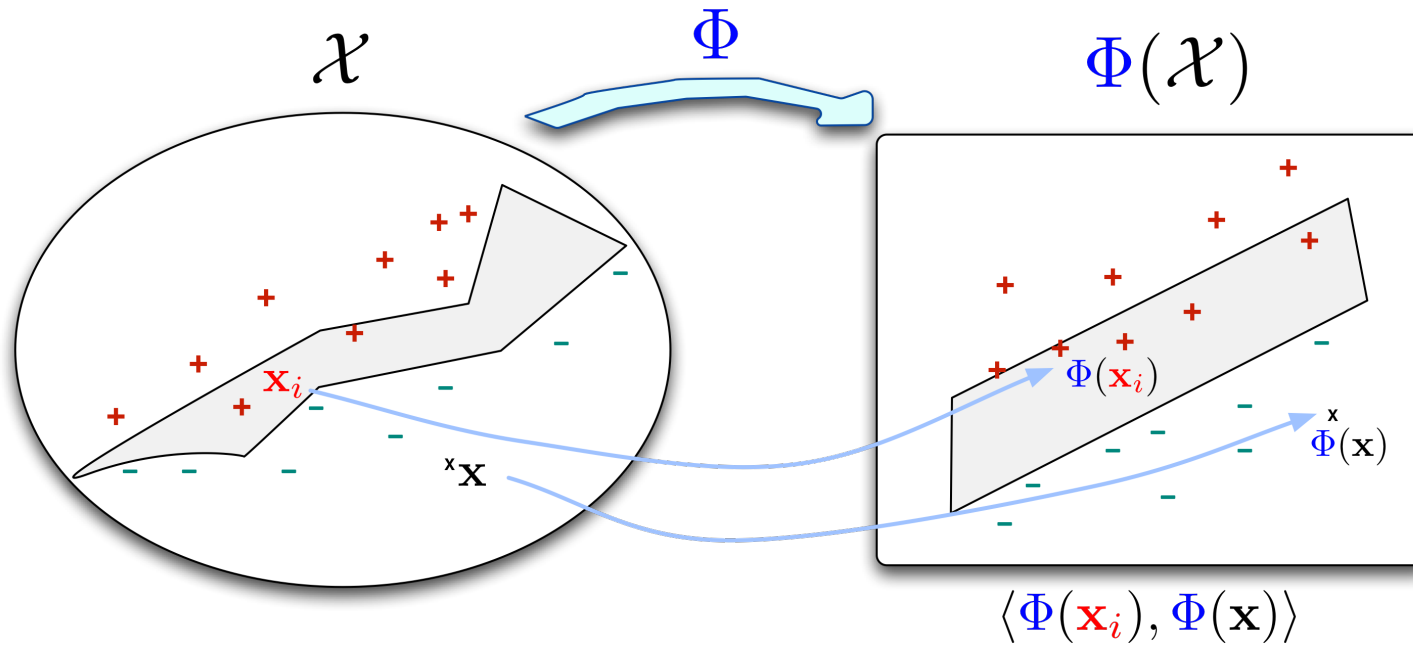
---

- Expression de l'**hypothèse** (fameuse « forme duale »)

$$h^*(\mathbf{x}) = \text{sign} \left\{ \sum_{i \in \mathcal{P}_S} \alpha_i^* y_i \langle \mathbf{x}_i, \mathbf{x} \rangle + w_0^* \right\}$$

- Trois idées
  - Hypothèses comme combinaison **linéaire**
  - Directement **fonction des exemples** (*exemples support*)
  - Minimise un **risque régularisé** dans lequel **la marge** mesure la versatilité de l'hypothèse

# SVM et méthodes à noyaux



$$h^*(\mathbf{x}) = \text{sign} \left\{ \sum_{i \in \mathcal{P}_S} \alpha_i^* y_i \kappa(\mathbf{x}_i, \mathbf{x}) + w_0^* \right\}$$

# Recherches actuelles : démarche générale

- Un **critère inductif** approprié

$$h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} \left[ R_{\text{Emp}}(h) + \lambda \text{reg}(h) \right]$$

$$h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} \left[ \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i) + \lambda \text{reg}(h) \right]$$

- Éventuellement une ré-expression pour **faciliter l'optimisation**
  - Convexité
    - E.g. Fonction de perte surrogée

# Le triomphe du paradigme

- Apprentissage = **problème inverse mal-posé**
  - Exprimer le problème sous forme d'**un critère régularisé**
  - Si possible **convexe**
- Le **boosting** et les **SVM** peuvent s'en dériver
- Le **choix de la description** disparaît
- Modèles additifs **linéaires**
  - Que l'on maîtrise bien
- **Extraordinaire généralité** du cadre

# « Traduction » : sélection de descripteurs

- Recherche d'hypothèse linéaire parcimonieuse

$$h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} \left[ \frac{1}{m} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i) + \lambda \text{reg}(h) \right]$$

$$h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} \left[ \frac{1}{m} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i) + \lambda \|h\|_1 \right]$$

$$\text{Norme } l_1: \quad \|\mathbf{w}\|_1 = \sum_{j=1}^p |w_j|$$

- Méthodes de type LASSO

# « Traduction » : classification semi-supervisée

- $l$  données étiquetées,  $u$  données non étiquetées

$$\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_l, y_l), \mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+u}\}$$

$$\mathbf{h} = [h(\mathbf{x}_1), h(\mathbf{x}_2), \dots, h(\mathbf{x}_{l+u})]$$

Mesure de régularité sur les données  $\mathbf{h}^\top \mathcal{L} \mathbf{h} = \frac{1}{2} \sum_{i,j=1}^{l+u} W_{ij} (h(\mathbf{x}_i) - h(\mathbf{x}_j))^2$

$$h^* = \underset{h \in \mathcal{H}}{\text{Argmin}} \left\{ \frac{1}{l} \sum_{i=1}^l (y_i - h(\mathbf{x}_i))^2 + \lambda_1 \|h\|_2 + \lambda_2 \mathbf{h}^\top \mathcal{L} \mathbf{h} \right\}$$

# « Traduction » : apprentissage multi-tâches

- $T$  tâches de classification binaire définies sur  $\mathcal{X} \times \mathcal{Y}$

$$\mathcal{S} = \left\{ \{(\mathbf{x}_{11}, y_{11}), (\mathbf{x}_{21}, y_{21}), \dots, (\mathbf{x}_{m1}, y_{m1})\}, \dots, \{(\mathbf{x}_{1T}, y_{1T}), (\mathbf{x}_{2T}, y_{2T}), \dots, (\mathbf{x}_{mT}, y_{mT})\} \right\}$$

$$h_j(\mathbf{x}) = \mathbf{w}_j \cdot \mathbf{x} \quad \text{Hypothèses linéaires}$$

*Partage entre tâches*       $\mathbf{w}_j = \mathbf{w}_0 + \mathbf{v}_j$

$$h_1^*, \dots, h_T^* = \underset{\mathbf{w}_0, \mathbf{v}_j, \xi_{ij}}{\text{Argmin}} \left\{ \sum_{j=1}^T \sum_{i=1}^m \xi_{ij} + \frac{\lambda_1}{T} \sum_{j=1}^T \|\mathbf{v}_j\|^2 + \lambda_2 \|\mathbf{w}_0\|^2 \right\}$$



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  $\rho_w$  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 :

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

Ainsi, trouver la solution optimale de l'équation (7.5) revient à chercher le vecteur  $\mathbf{w}$  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  $\ell_{\text{Erf}_{\text{cx}}}(\cdot)$  (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 :

---

**La fin de l'histoire ...**  
**... ou un accident de l'histoire ?**  
(2013 - ...)

# La fin de l'histoire

- Données i.i.d.
  - > **Pas** de dépendances entre les exemples
- Apprentissage batch
  - > **Pas** de séquence
- Critère convexe
  - > **Pas** de dépendance du résultat sur les accidents de l'exploration
- Modèles additifs
  - > **Pas** de construction de concepts intermédiaires



*Il n'y a plus d'histoire*

# La fin de l'histoire

---

Et pourtant ...

# Pas le choix

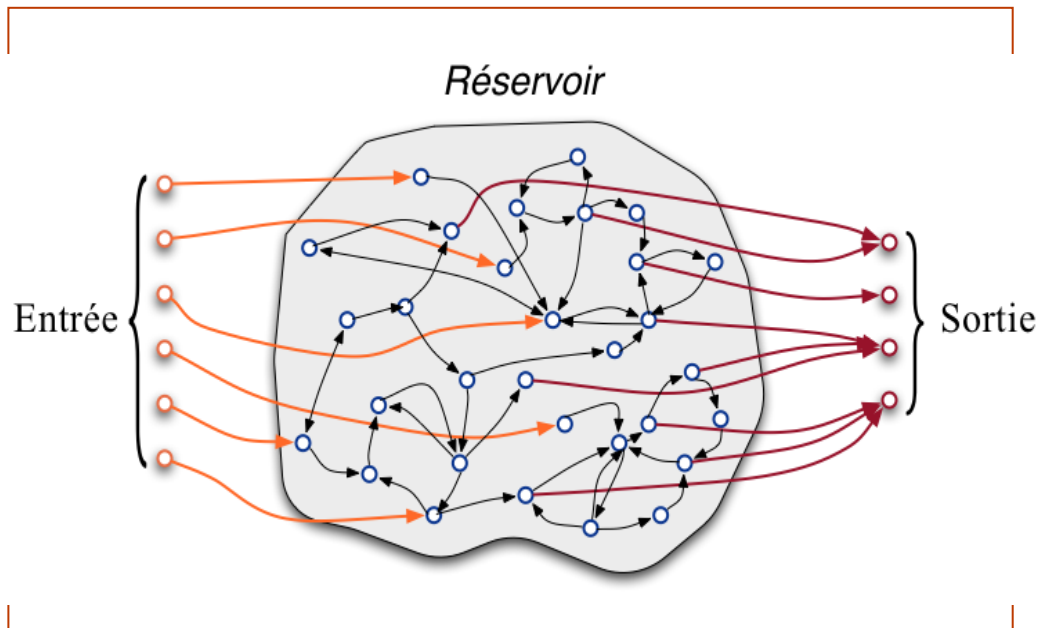
---

- Très grandes *masses de données*
  - Impossible d'optimiser en un coup -> **histoire**
- Critère *non convexe*
  - Exploration de l'espace de recherche -> sensible à l'**histoire**
- Apprentissage « *long-life* » et *multi-tâches* : par *transfert*
  - Séquence de tâches -> sensible à l'**histoire**

# Une idée intrigante : le « reservoir computing » »

## ■ Idée :

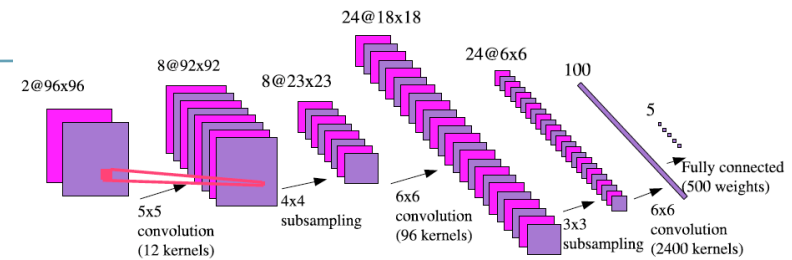
- Utiliser un réseau récurrent sans l'entraîner explicitement
- Entraîner une seule couche de sortie



- Ré-introduit une « dynamique »
  - ▣ Séries temporelles

# Apprentissage hiérarchique

## ■ Et si ...



- [Bengio & Le Cun, 07] « *Scaling Learning Towards AI* »
  - [Bengio et al., 09] « *Curriculum Learning* »
  - [Bottou, 11] « *From Machine Learning to Machine Reasoning* »
  - [Valiant, 00] « *A Neuroidal Architecture for Cognitive Computation* »
  - Étonnement que nos méthodes n'apprennent pas mieux avec des **exemples typiques bien choisis** plutôt qu'avec beaucoup de données !?
  - ...
- pointaient vers des **limites du paradigme dominant**  
et soulignaient des **questions incontournables**

# Rétrospective

1943

Notion d'information + machine programmable

1960

**Pionniers** : algorithme + adaptation

- **Représentation** (prédicats)
- **Paramètres** (credit assignment problem)

1970

**Reconnaissance des Formes**

- **Critère d'optimalité** général (MAP, MLE)
- **Estimation de paramètre** / (sélection de modèle)

**Intelligence Artificielle**

- **Représentation** structurée / logique
- Induction = **Généralisation**
- **Exploration d'un espace discret**

1985

*Appauvrissement de la tâche -> triomphe de l'approche statistique*

**Apprentissage = problème inverse mal-posé**

- Risque régularisé
- Algorithme d'optimisation

2010

Extraordinaire puissance du paradigme MAIS évacuation de l'histoire

*Et après ?*



---

## ■ Apprentissage

1. Algorithme
2. Algorithme qui **adapte**
  - Améliorations *locales*
3. Algorithme qui **raisonne**
  - Inférences *locales*
4. Algorithme qui **optimise**
  - Critère *général*
    - Batch
    - Sur  $\mathcal{H}$  (convexité)

La suite ?

Une suite ?

---

## ■ Algorithmes d'apprentissage

1. **Algorithme**
2. Algorithme qui **adapte**
  - Améliorations *locales*
3. Algorithme qui **raisonne**
  - Inférences *locales*
4. Algorithme qui **optimise**
  - Critère *général*
    - Batch
    - Sur  $\mathcal{H}$  (convexité)

Apprentissage  
d'algorithme ?

# Quelques références

---



[A. Cornuéjols & L. Miclet](#)

Apprentissage Artificiel. Concepts et algorithmes

[Eyrolles \(2° éd.\), 2010](#)



[J. Johnston](#)

The Allure of Machinic Life. Cybernetics, Artificial Life and the New AI

[MIT Press, 2011](#)



[T. Mitchell](#)

The Discipline of Machine Learning

[CMU-ML-06-108, 2006](#)



[N. Nilsson](#)

The Quest for Artificial Intelligence. A History of Ideas and Achievements

[Cambridge University Press, 2010](#)



[J. Shavlik & T. Dietterich](#)

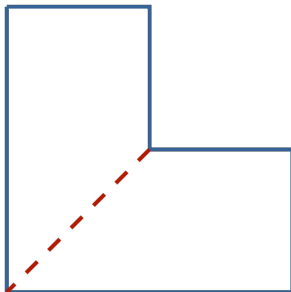
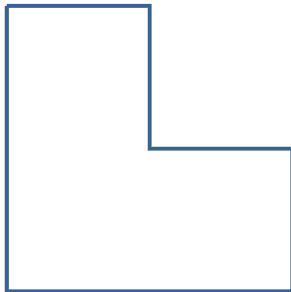
Readings in Machine Learning

[Morgan Kaufmann, 1990](#)

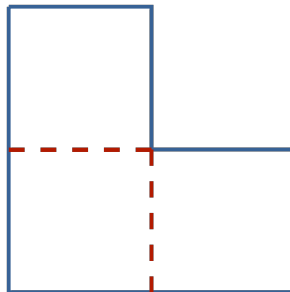
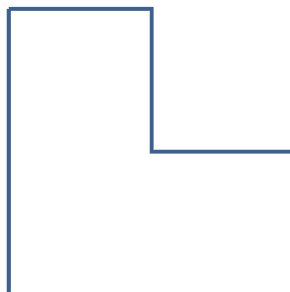
# Effets de séquences

- Consigne : découper la figure suivante en  $n$  parties superposables

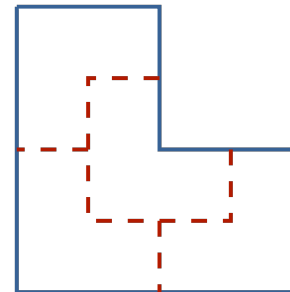
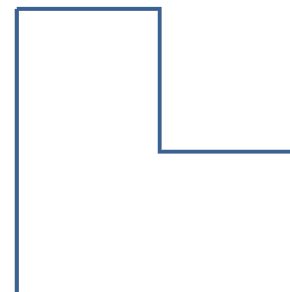
En 2



En 3



En 4



En 5

