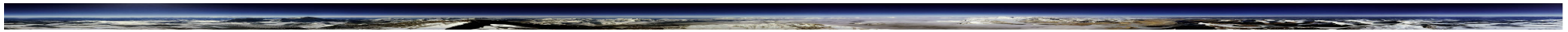
Apprentissage (et) statistique

de l'âge de raison à l'empire des normes



A. Cornuéjols

AgroParisTech – INRA MIA 518

Trame

- Comment aborder l'apprentissage
  - Comment cela a été fait
- Les paradigmes et leur évolution
  1. Théorie du contrôle … symbolique
  2. L'âge de raison
  3. Transition
  4. L'empire des normes
- Et maintenant … ?

C'est quoi l'apprentissage ?

Deux **conceptions**

- Contrôle / adaptation
  - *Jeu d'échecs ; robots ; …*
  - **Reconnaissance** des formes
  - Apprentissage pour la **décision**
  - Apprentissage par **renforcement**
  - **Évolution** simulée

- Connaissances / modélisation
  - *Apprentissage de « théories », d'explications, …*
  - **Explanation** Based-Learning
  - Apprentissage de **catégories**
  - Apprentissage de **concepts**

Objets d'une discipline de l'apprentissage

- Science de l'**ingénieur**
  - Réaliser
- **Lois** fondamentales
  - Comprendre
  - Prédire

« **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

Perspective historique

Héritage de la cybernétique

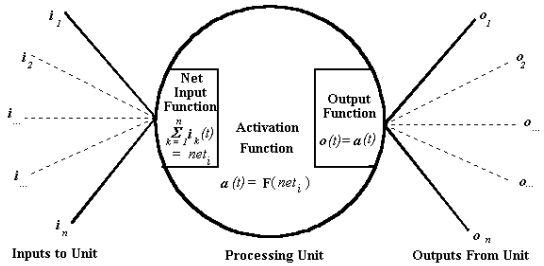
- **Avant**
  - Les pensées ne **peuvent pas être mesurées**
  - Prégnance des **modèles physiques**
    - *Vitesse*, *force* et *intensité* d'un **signal** dans un milieu continu
    - Causes et effets. Causes *avant* les effets
- **Après**
  - Prégnance du **concept d'information**
    - La machine de Turing opère sur des **symboles**
    - Toute interaction est **communication**
    - Le **feedback** est central :
      - l'effet désiré **précède** la cause à produire
      - Le feedback est de **l'information**

La **cybernétique** : illustration

- 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



IA : deux directions

- Est-ce qu'une machine peut penser ?
  - Comme un humain adulte -> difficile

- Est-ce qu'une machine peut apprendre ?
  - Construisons un « enfant » (ou un organisme élémentaire)
  - Faisons le apprendre
  - Nous aurons un adulte pensant

Turing (1950) « *Computing Machinery and Intelligence* »,
Mind, vol. LIX, N° 236, pp. 433-460.

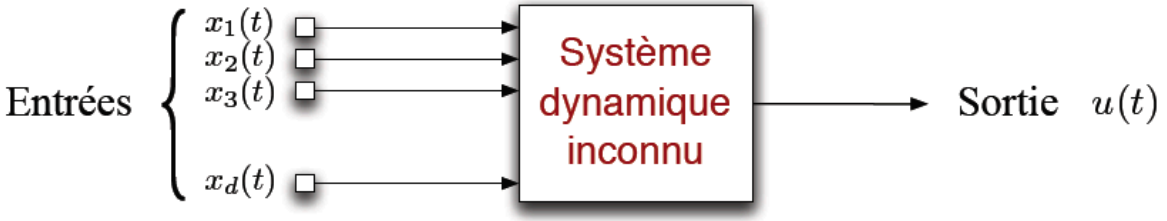
Apprendre : c'est s'adapter

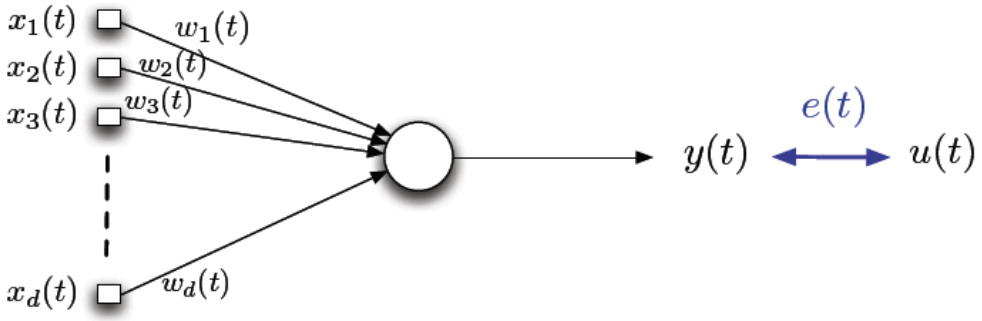
(~ 1956 - ~1970)

Connexionnisme : 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, 1960

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:

$$\epsilon(m) = f(m) - h_1 f(m-1) - h_2 f(m-2)$$

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

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 $\overline{\epsilon^2}$ of equation (1) equal to zero. This is the discrete analogue of Wiener's optimization[7] 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 :

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

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

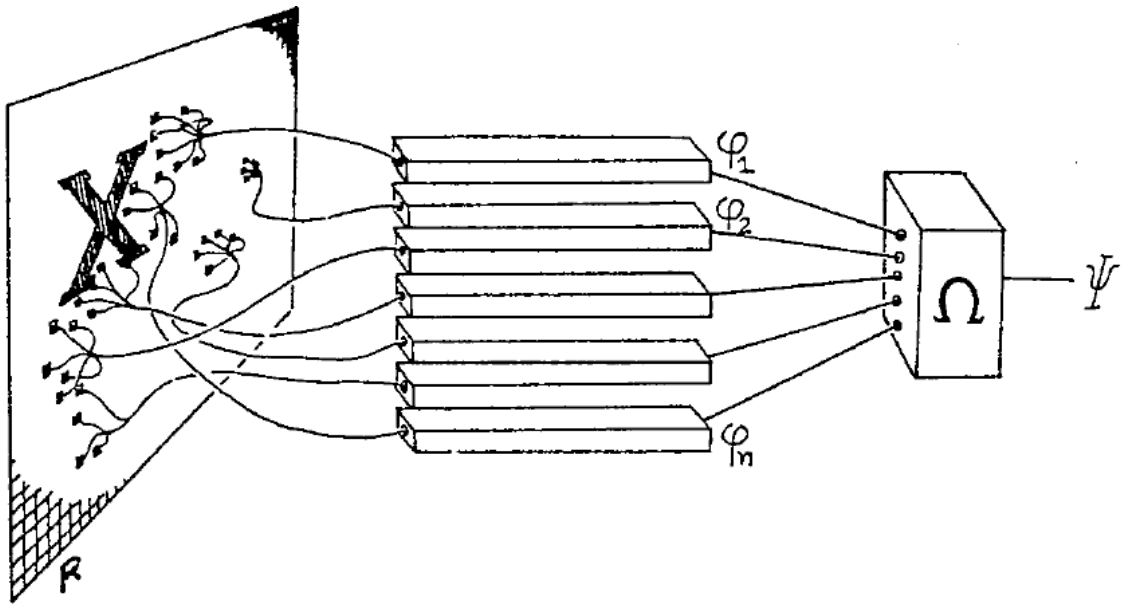
$$\frac{\partial \ell(\mathbf{w})}{\partial \mathbf{w}(t)} = -\mathbf{x}(t)\, e(t)$$

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

[Widrow-Hoff:60] B. Widrow and M. Hoff. *Adaptive Switching Circuits.* IRE WESCON Conv. Rec. Pt.4, pp.96-104.

Connexionnisme : le perceptron

- Frank Rosenblatt (1958 – 1962)



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

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

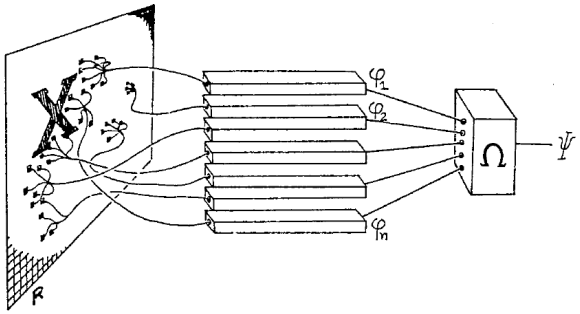
Règle du perceptron : apprendre seulement en cas d'échec

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
    $\mathbf{w}(t+1) = \mathbf{w}(t) - \eta\, \mathbf{x}_i\, y_i$
  fin
  Passer à la forme d'apprentissage suivante
fin

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)
    - Pour une dernière étape de combinaison linéaire

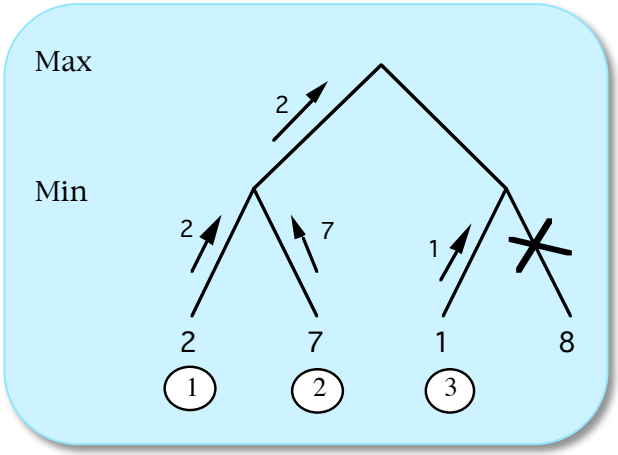


→ Blocage

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**

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



- Deux problèmes
  1. **Sélectionner** de bonnes fonctions de base : ϕ_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 : score := score +1.
  - Quand 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

Bilan

Des algorithmes simples

- Importance de la représentation
  - Espaces discrets
  - Etats / opérateurs
  - Représentations
    - Descripteurs de haut niveau (Checker)
    - Représentations complexes (Perceptron)

- Mais comment apprendre une bonne représentation ?

Or …. pendant ce temps

- General Problem Solver (1959 - …)
  - Théorie du contrôle symbolique
- Prouveurs de théorèmes
  - Principe de résolution [Robinson, 1965]
- DENDRAL (1965 - …)
  - Précurseur des systèmes experts
- ANALOGY (Evans, 1963)

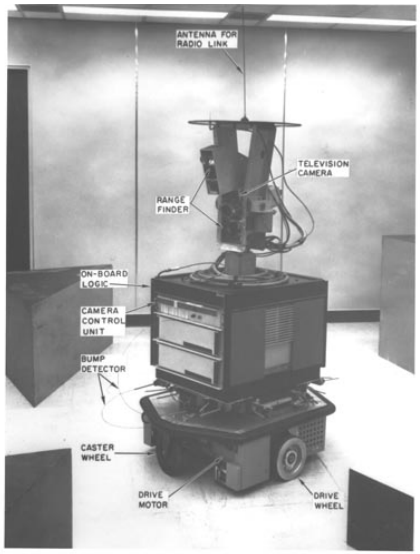
L'âge de raison

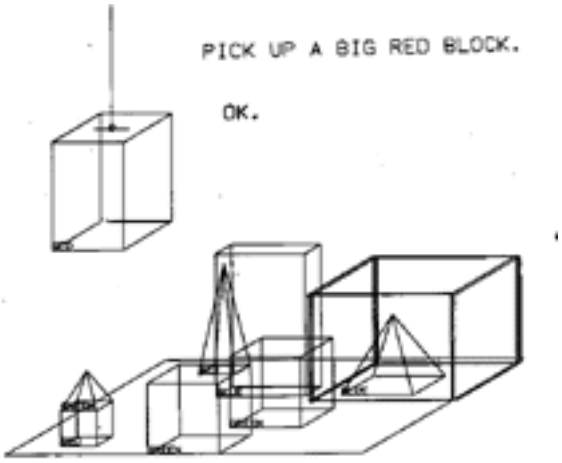
Apprendre en **pensant**

(~ 1970 - ~1984)

Même les robots pensent : Shakey

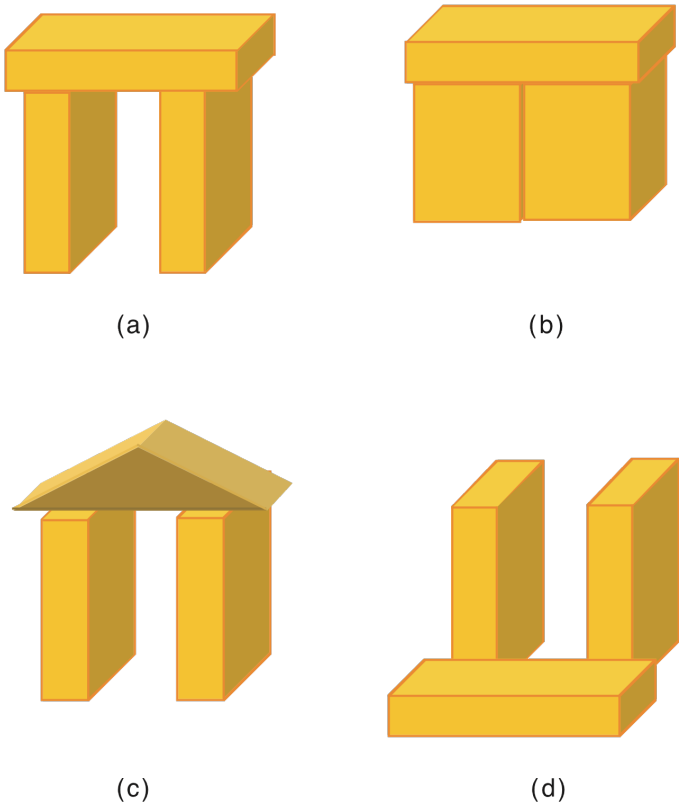
- 1er robot mobile contrôlé par ordinateur.
  (Stanford Research Institute, 1967-1972)
  - **Vision** : Thèse de *David Waltz* (reconnaissance de polyèdres en 3D à partir d'une image 2D)
  - **Contrôle et planification** : STRIPS, ABSTRIPS (puis NOAH, …)
  - **IHM** : SHRLDU [Thèse de *Terry Winograd*,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





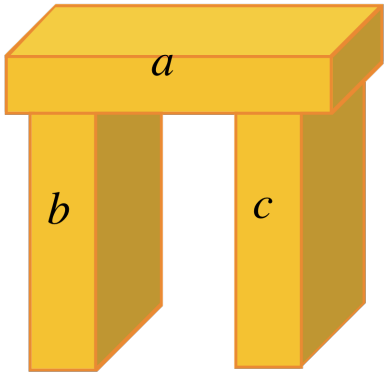
ARCH [Winston, 1970]

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



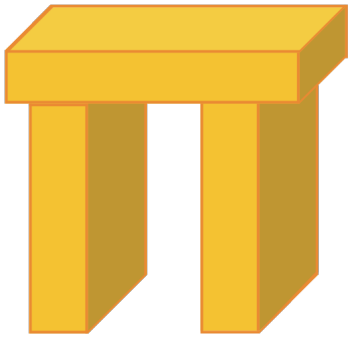
ARCH [Winston, 1970]

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



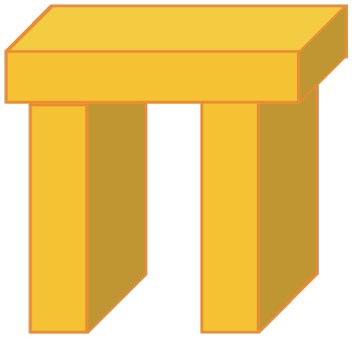
ARCH [Winston, 1970]

- Heuristique : « require-link »



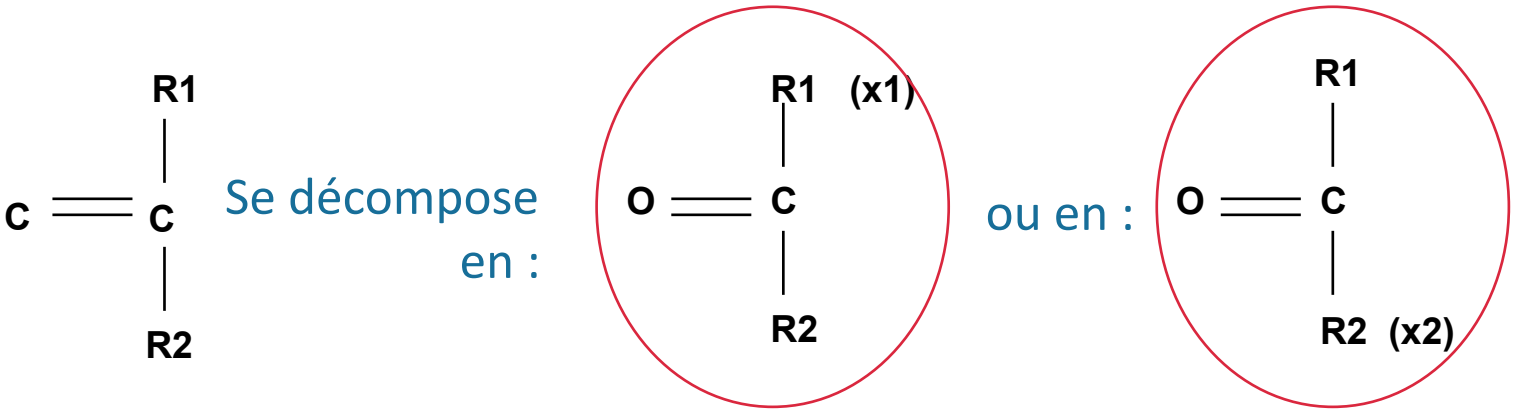
ARCH [Winston, 1970]

- Heuristique : « forbid-link »



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

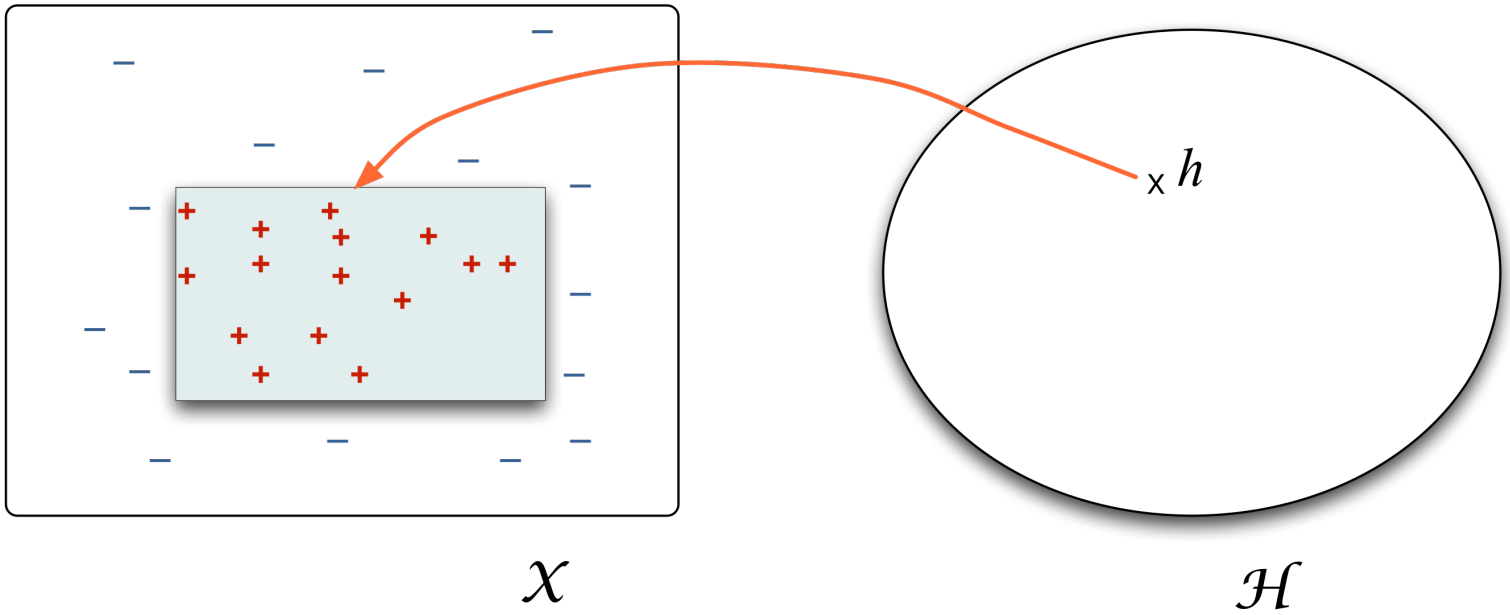
- **Apprentissage de règles** pour le système expert Meta-Dendral
  - Descriptions relationnelles de **sous-structures moléculaires** ayant probablement produit les fragments mesurés dans un spectrogramme de masse.



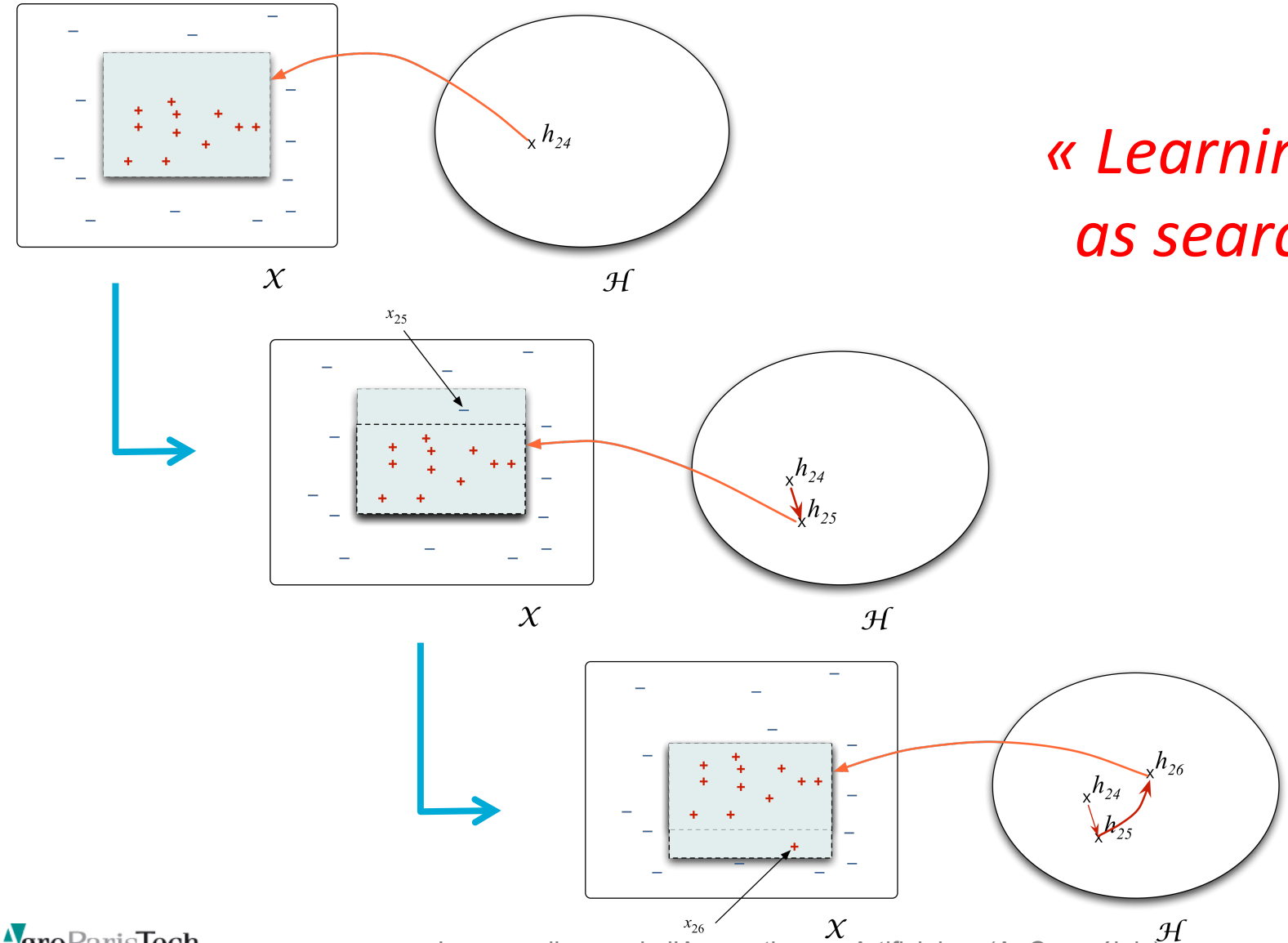
mais pas en …

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]

- 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*)

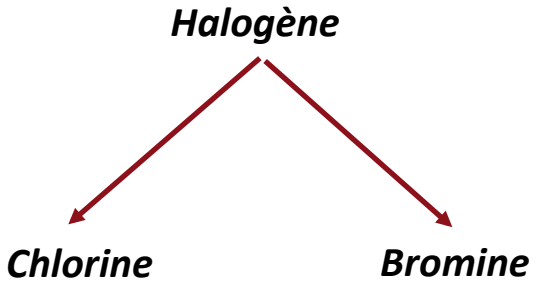
  - **Spécialisation**

    - Duale

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

- Opérateurs de **généralisation** (spécialisation)
  - **Abandon de conjonction**
    - A & B -> C => A -> C
  - **Ajout d'alternative**
    - A ou B -> C => A ou B ou D -> C
  - **Ascension** dans une **hiérarchie de concepts**
    - corrosif & bromine -> toxique

      => corrosif & halogène -> toxique



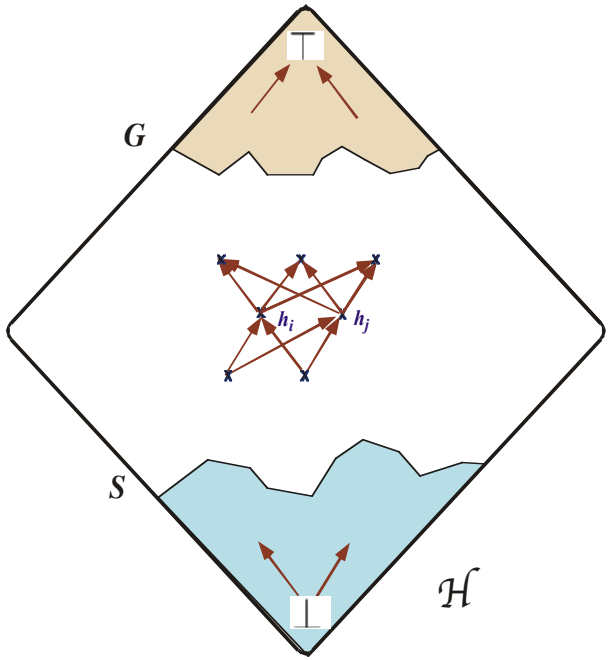
  - **Inversion de la résolution**
  - …

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

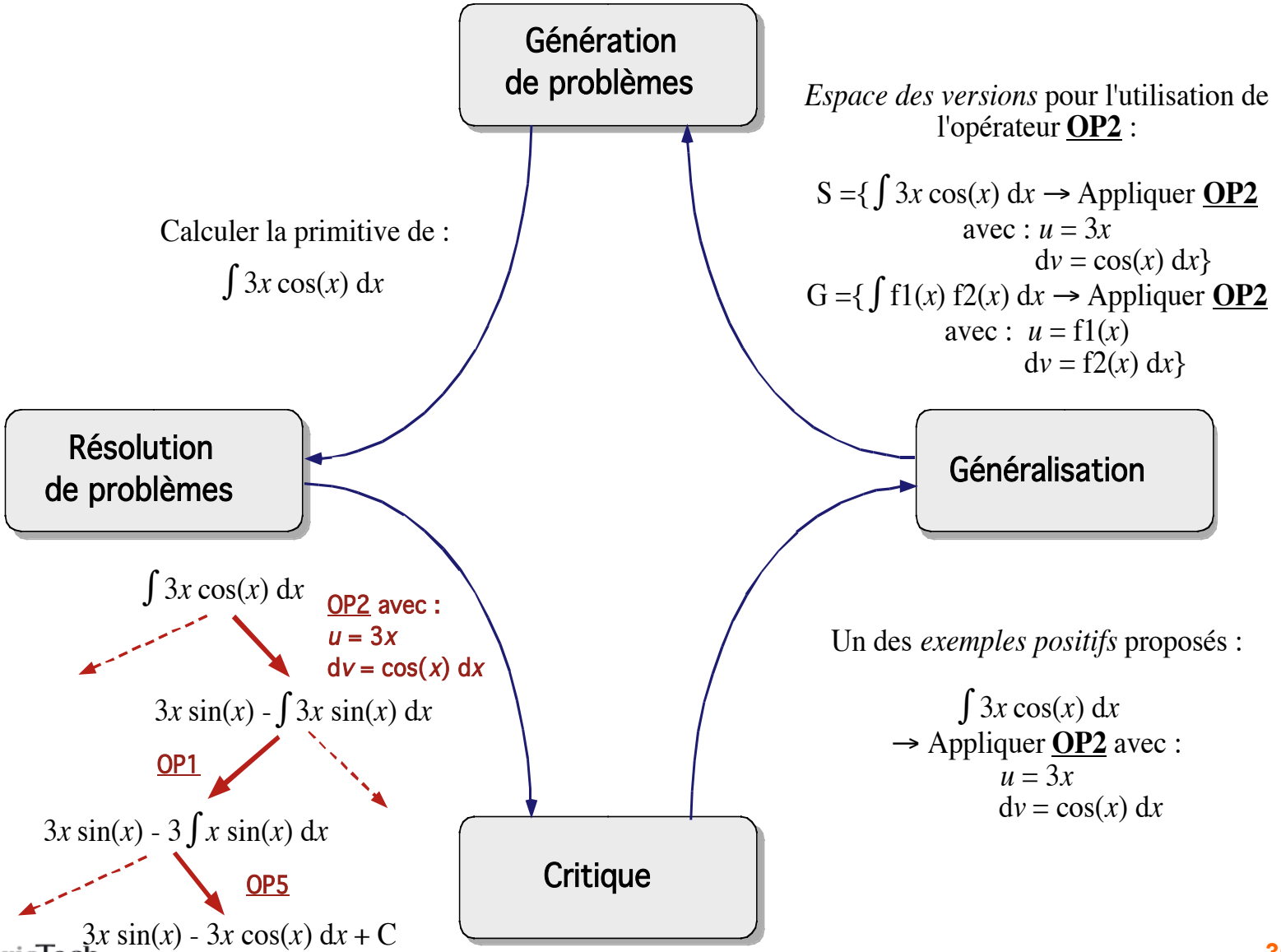
➥ *Algorithme d'élimination des candidats*

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

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* $\mathbf{x}$ **faire**
  si $\mathbf{x}$ *est un exemple positif* **alors**
    Enlever de G toutes les hypothèses qui ne couvrent pas $\mathbf{x}$
    pour chaque *hypothèse* s *de* S *qui ne couvre pas* $\mathbf{x}$ **faire**
      Enlever s de S
      Généraliser(s,$\mathbf{x}$,S)
      c'est-à-dire : ajouter à S toutes les généralisations minimales h de s telles que :
        - h couvre $\mathbf{x}$ 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 $\mathbf{x}$
    pour chaque *hypothèse* g *de* G *qui couvre* $\mathbf{x}$ **faire**
      Enlever g de G
      Spécialiser(g,$\mathbf{x}$,G)
      c'est-à-dire : ajouter à G toutes les spécialisations maximales h de g telles que :
        - h ne couvre pas $\mathbf{x}$ 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]

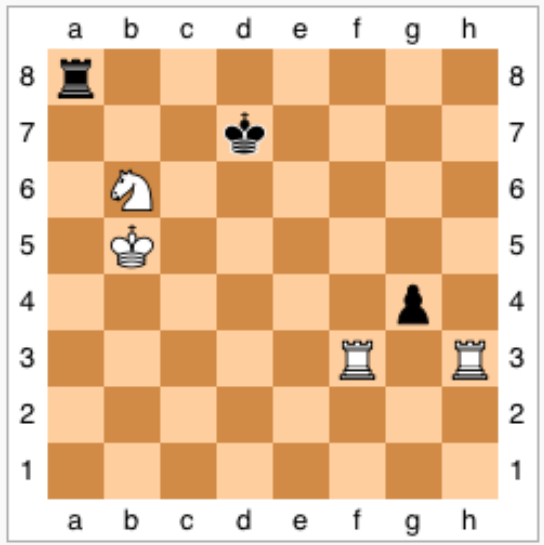


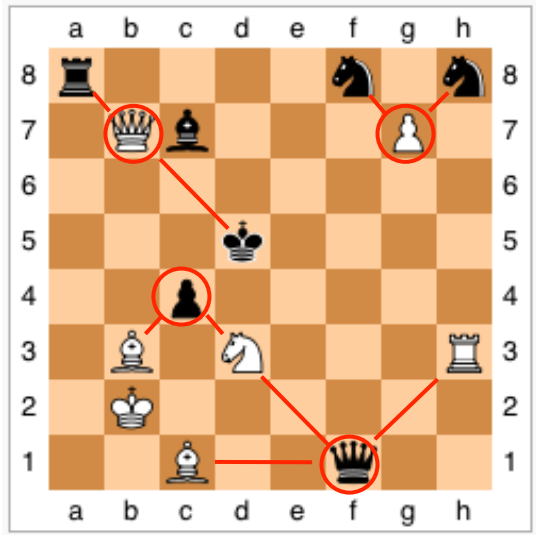
Explanation-Based Learning

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** :

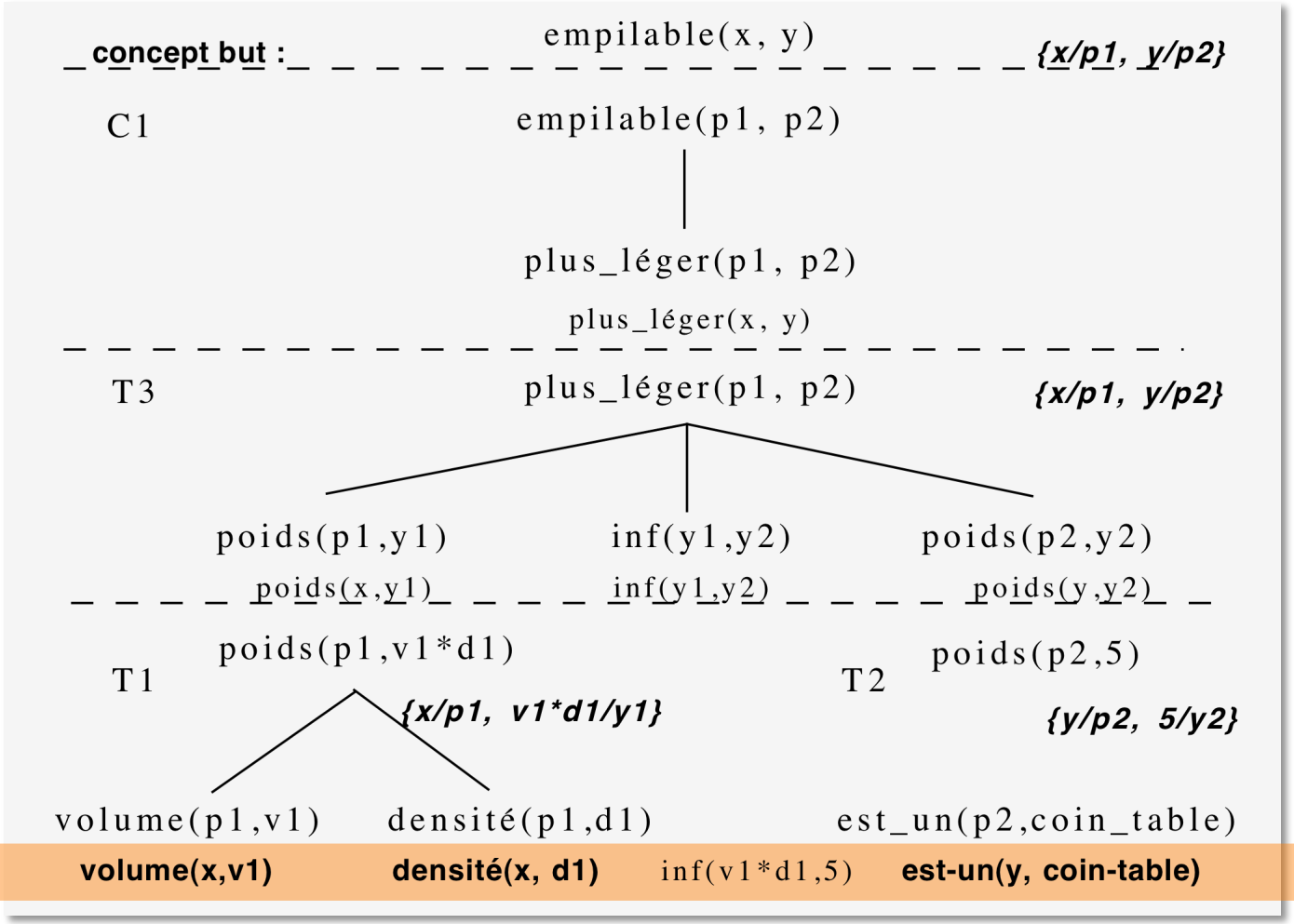
```
(T1) : poids(X, W) :- volume(X, V), densité(X, D), W is V*D.
(T2) : poids(X, 50) :- est-un(X, table).
(T3) : plus-léger(X, Y) :- poids(X, W1), poids(X, W2), W1 < W2.
```

- **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

- **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**
   - Mémoire *procédurale*, *déclarative*, de *travail*
   - Raisonnement : *déduction, induction, analogie*, …

L'« âge de raison » : les limites

- Requiert une forte **théorie du domaine**
  - Difficulté pour **l'acquérir** 
  - Suspicion de **manque de généricité** (ad hoc) 
  - Mais **se contente de peu d'exemples** 
- Pas adapté à des données **bruitées**
- Le « passage à l'**échelle** » n'est **pas évident**
- **Pas** de critère d'**évaluation** général et quantifiable
  - Permettant une comparaison entre systèmes

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 retournement de perspective

(~ 1984 - ~1995)

Deux mouvements indépendants …

- … arrivent à maturité quasi simultanément

- juste au moment où l'âge de raison s'essouffle

Deux mouvements indépendants …

- … arrivent à maturité quasi simultanément

1. Une nouvelle **théorisation** de l'apprentissage
   - Valiant (1984) : le *PAC learning*
   - Vapnik (1974-1989) : *conditions nécessaires et suffisantes pour la consistance du principe de minimisation du risque empirique*

2. Le renouveau du **connexionnisme** (1986 -)
   - Perceptrons multi-couches et rétro-propagation de gradient

Nouvelle théorisation

Une théorie de l'adaptation à la « consigne »

- Notion de **fonction cible** $f : y = f(\mathbf{x}) + \varepsilon$
- Consigne connue par l'expérience (supposée stochastique)
  - Échantillon d'apprentissage (i.i.d.)

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_m, y_m)\}$$

- Idéalement (le ***risque réel***)

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

- Empiriquement (le ***risque empirique***)

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

Nouvelle théorisation

- Le principe de minimisation du risque empirique (ERM)

  … est-il sain ?

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

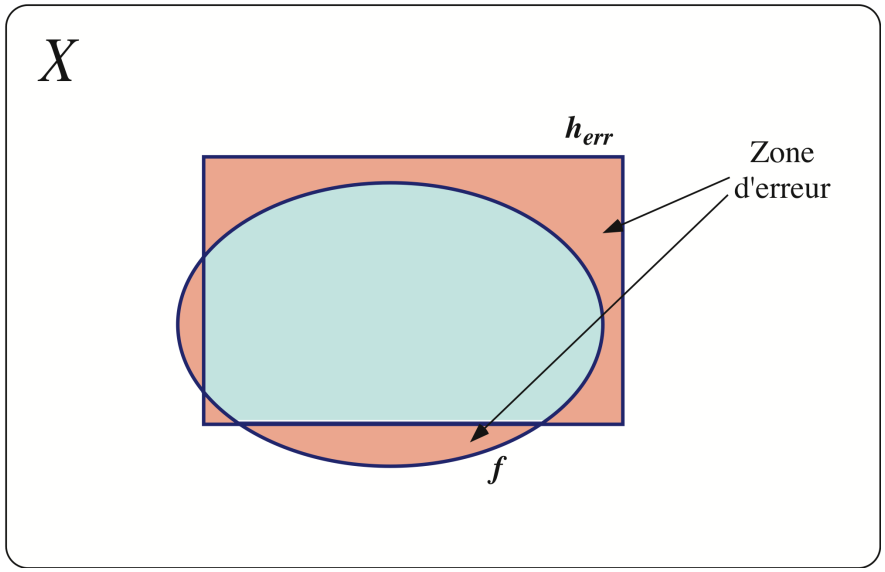
  – Est-ce que *h* est bonne relativement au risque réel ?

$$h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} \ R(h)$$

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

L'analyse « PAC learning »

- Supposons $f \in \mathcal{H}$. À tout instant il existe **au moins une hypothèse** dans l'espace des versions (**d'erreur nulle**)
  - Je choisis(*) 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_{err} de risque réel > ε **et que je ne m'en aperçoive pas** après l'observation de *m* exemples ?
- Probabilité de survie de h_{err} **après 1 exemple** : $(1-\varepsilon)$
- Probabilité de survie de h_{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 > ε dans l'espace des versions soit bornée par δ :

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

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

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

L'analyse « PAC learning »

- On arrive à :

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

Le principe de minimisation du risque empirique

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

L'analyse « PAC learning »

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

- *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*)

Nouveautés séduisantes

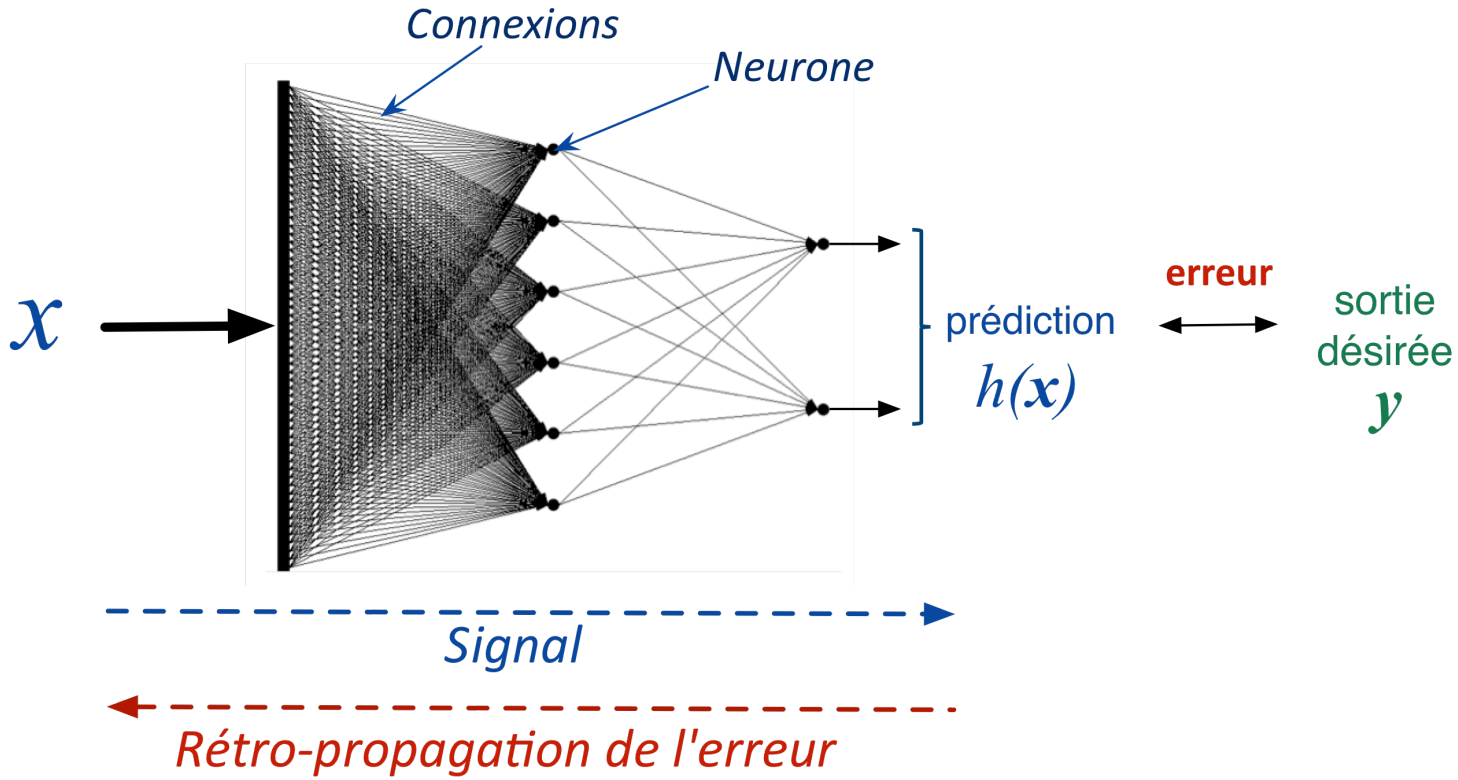
- **Algorithme d'apprentissage**
  - Générique : ***minimisation du risque empirique régularisé***
  - Apprentissage = optimisation

- **Faible a priori** sur le monde
  - Suppose données (et questions) **i.i.d.**
  - $f \in H$ ou $f \notin H$
  - **Valable dans le pire cas** : contre toute distribution cible

- **Bornes** en généralisation
  - Formalisation mathématique supportant le bien-fondé

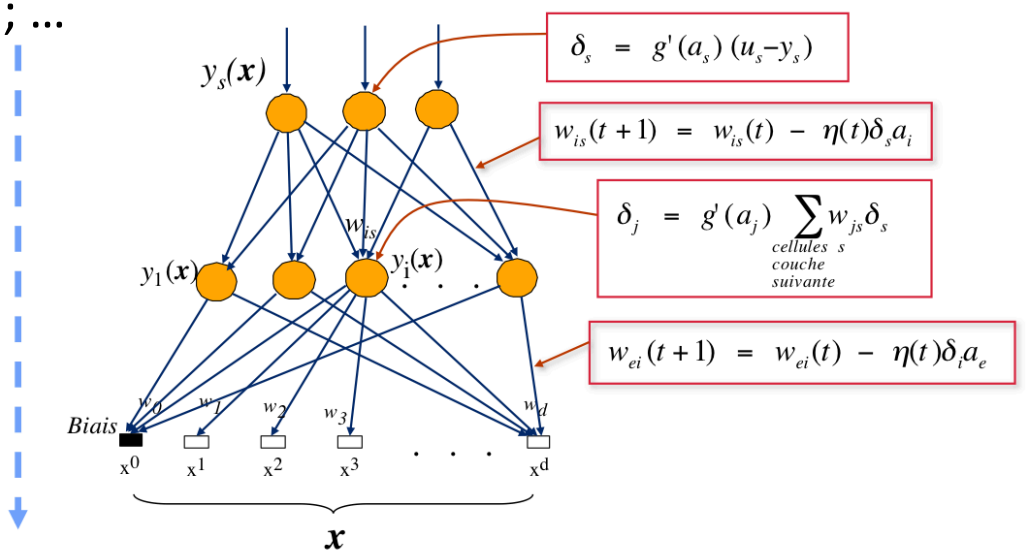
Le 2ème connexionnisme

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



Le 2ème connexionnisme : un problème d'optimisation

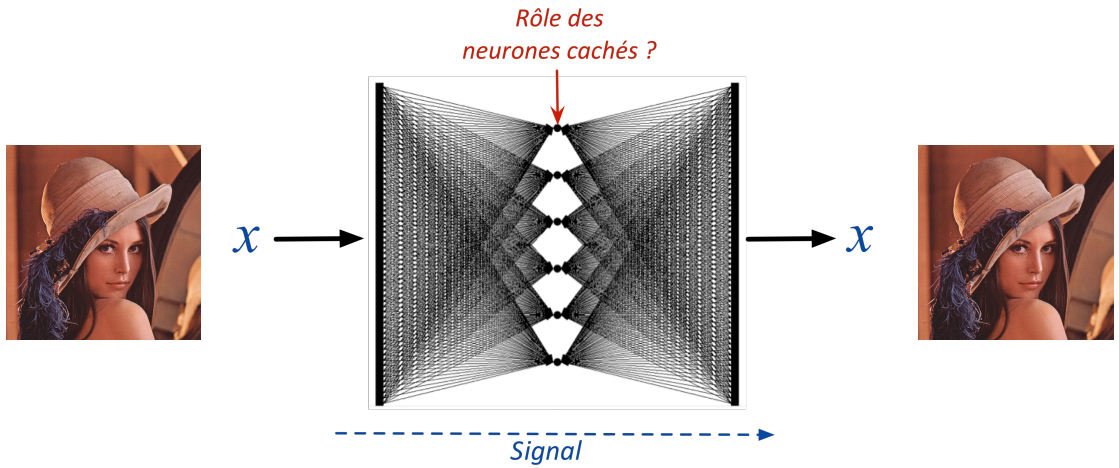
- Algorithme de **rétro-propagation de gradient** $\frac{\partial E^l}{\partial w_{ij}}$
  - Algorithme **itératif**
    - Gradient stochastique ou total
  - **Local**
  - Valide pour **tout type d'apprentissage supervisé**
    - Classification ; régression ; …
    - Toute mesure d'erreur



Remarques sur cette transition

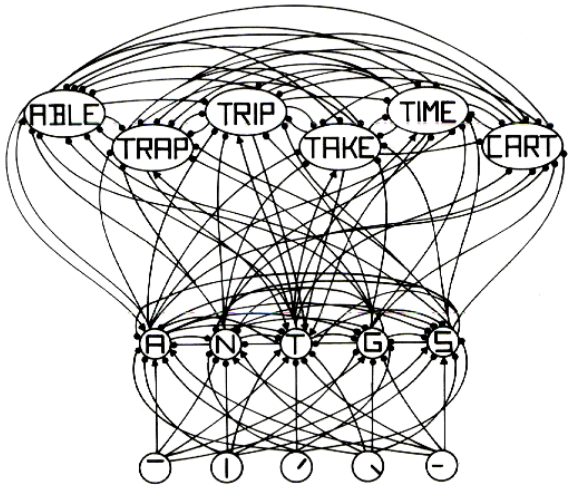
Le nouveau connexionnisme

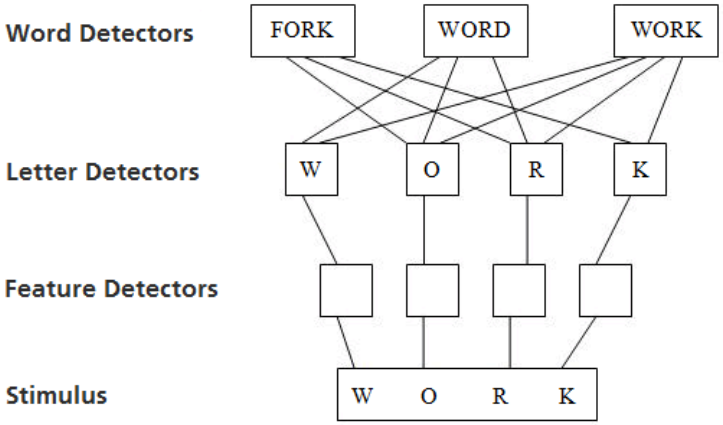
- Au début
  - Questions sur la représentation
    - Que « représentent » les neurones en couches cachées ?
    - Comment apprendre de bonnes représentations ?

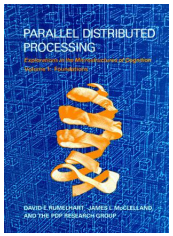


Le 2ème connexionnisme : témoin d'une **transition**

- *« … 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 2ème 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 : contraintes sur $\mathcal{H}$
    - Optima locaux

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

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 **optimisation** d'un critère stochastique (exemples i.i.d.)

Désormais

- Nécessite **beaucoup** d'exemples
  - Supposés équivalents (i.i.d.)

- **Pas** de construction d'un **modèle du monde** ou d'une théorie élaborés

- **Non** intégré avec du **raisonnement**

Un paradigme triomphant

Apprentissage = choix de normes + optimisation

(~ 1995 - ~20??)

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
     - Si possible conduisant à problème convexe

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

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

Un paradigme extraordinairement performant

- Le **boosting**

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

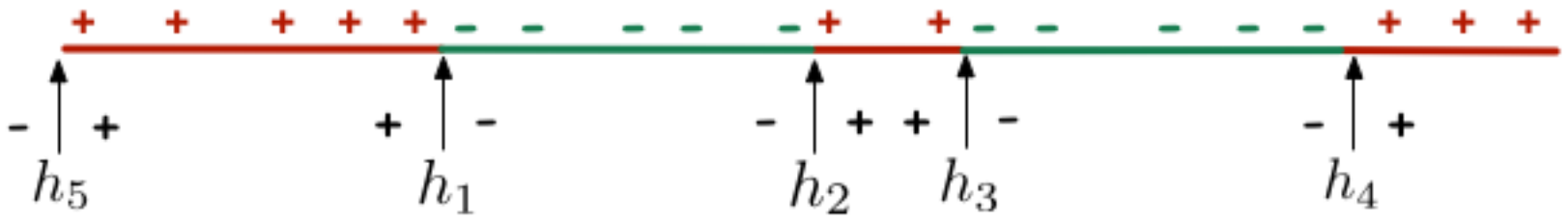
Le boosting : illustration

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

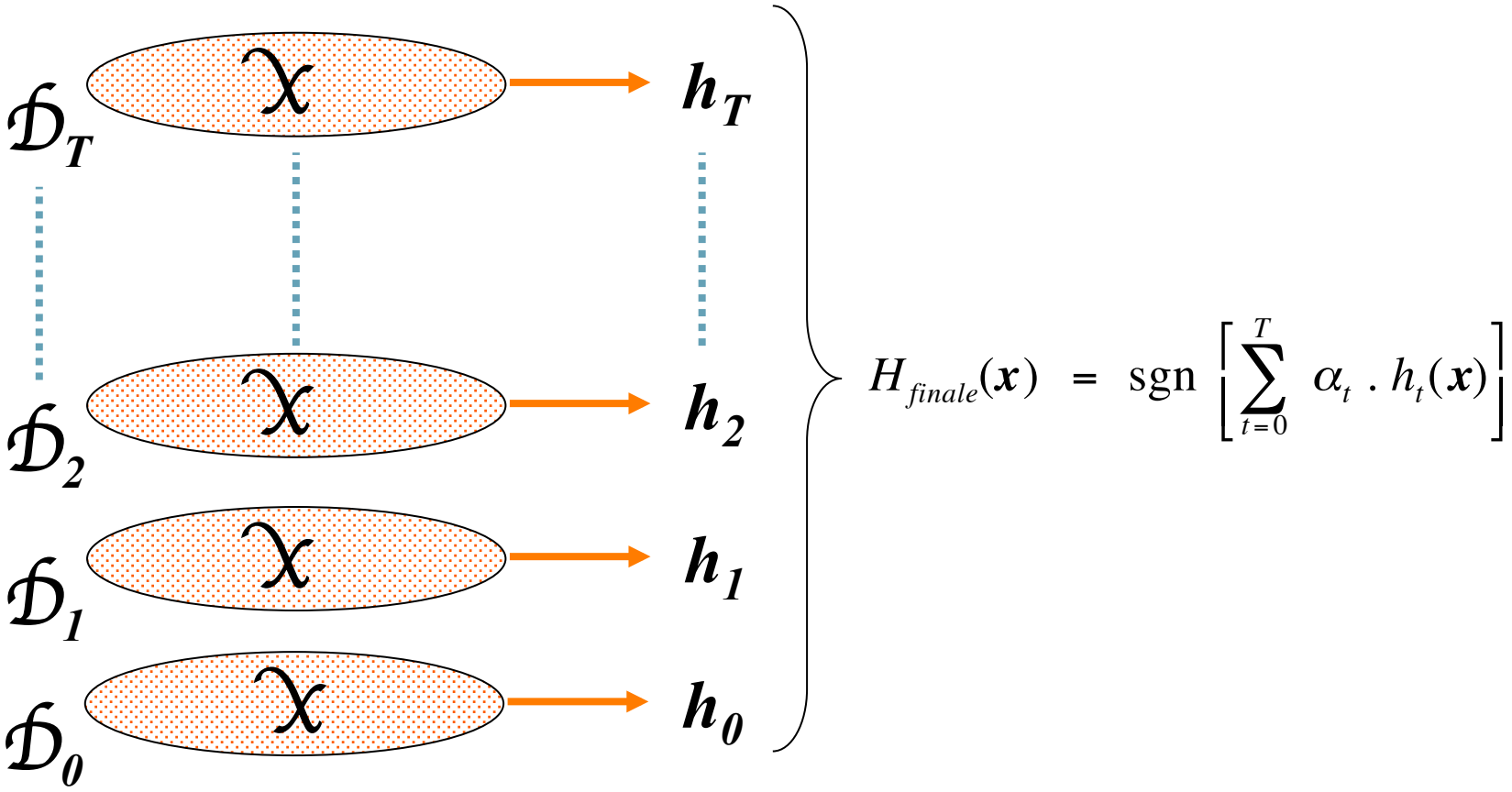


H(x) =sign{ 0.549 $h_1(x)$ + 0.347 $h_2(x)$ + 0.310 $h_3(x)$ + 0.406 $h_4(x)$
+ 0.503 $h_5(x)$ }

- Comment arriver à ce genre de combinaison ?

Algorithme du boosting

Le principe général



- **Comment passer de $\mathcal{D}_t$ à $\mathcal{D}_{t+1}$?**

- **Comment calculer la pondération α_t ?**

L'algorithme AdaBoost

Algorithme itératif glouton

Procédés et esprit de la preuve (constructive)

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

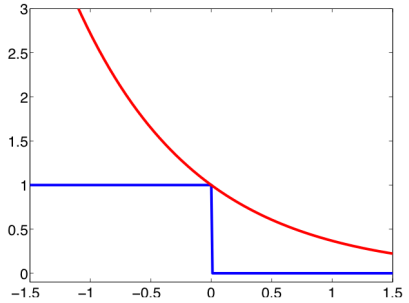
$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) > 0$$

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

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

Le boosting s'inscrit dans le paradigme

- **Re-dérivation du boosting**
  - En choisissant une *fonction de perte surrogée* de forme exponentielle



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

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

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

$$R_{\text{Emp}}(H_T) \;=\; \sum_{i=1}^{m} e^{-y_i \left[H_{T-1}(\mathbf{x}) + \alpha_T\, h_T(\mathbf{x})\right]}$$

$$= \sum_{i=1}^{m} e^{-y_i\, H_{T-1}(\mathbf{x})} \cdot e^{-\alpha_T\, h_T(\mathbf{x})}$$

$$= \sum_{i=1}^{m} \underbrace{W_{T-1}(\mathbf{x}_i)}_{\text{Poids de } \mathbf{x}_i \text{ à } T-1} \cdot \underbrace{e^{-\alpha_T\, h_T(\mathbf{x})}}_{\text{à optimiser}}$$

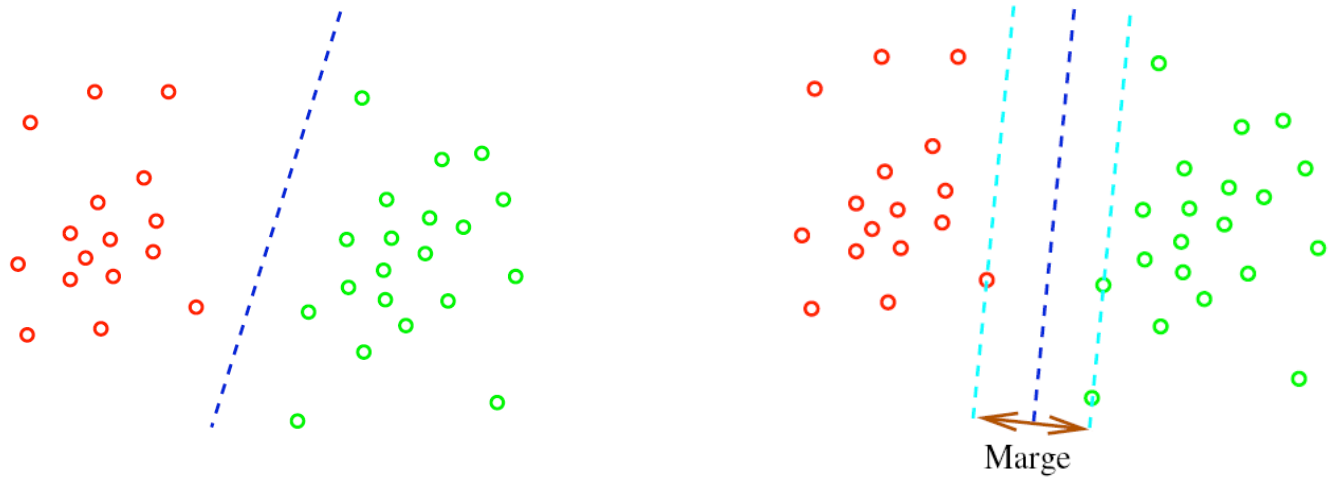
Algorithme de gradient conjugué

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

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

SVM et méthodes à noyaux

- **S**éparateur linéaire à plus **V**aste **M**arge



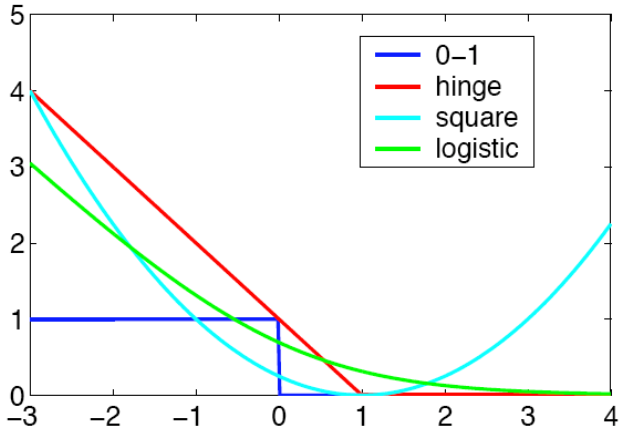
  - Intuitivement : plus robuste à variations de l'échantillon d'apprentissage
  - Validé par analyse théorique
    - bornes de généralisation fonction de la marge

SVM et méthodes à noyaux

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

$$\mathbf{w}^\star = \underset{\mathbf{w}}{\text{ArgMin}} \Big[\underbrace{\sum_{i=1}^{m} \max\{1 - y_i h(\mathbf{x}_i)\}}_{\text{Risque empirique (surrogé)}} + \lambda \underbrace{\frac{1}{2}\mathbf{w}^\top \mathbf{w}}_{\text{Marge}} \Big]$$

Fonction de *perte de substitution* (surrogate loss)



SVM et méthodes à noyaux

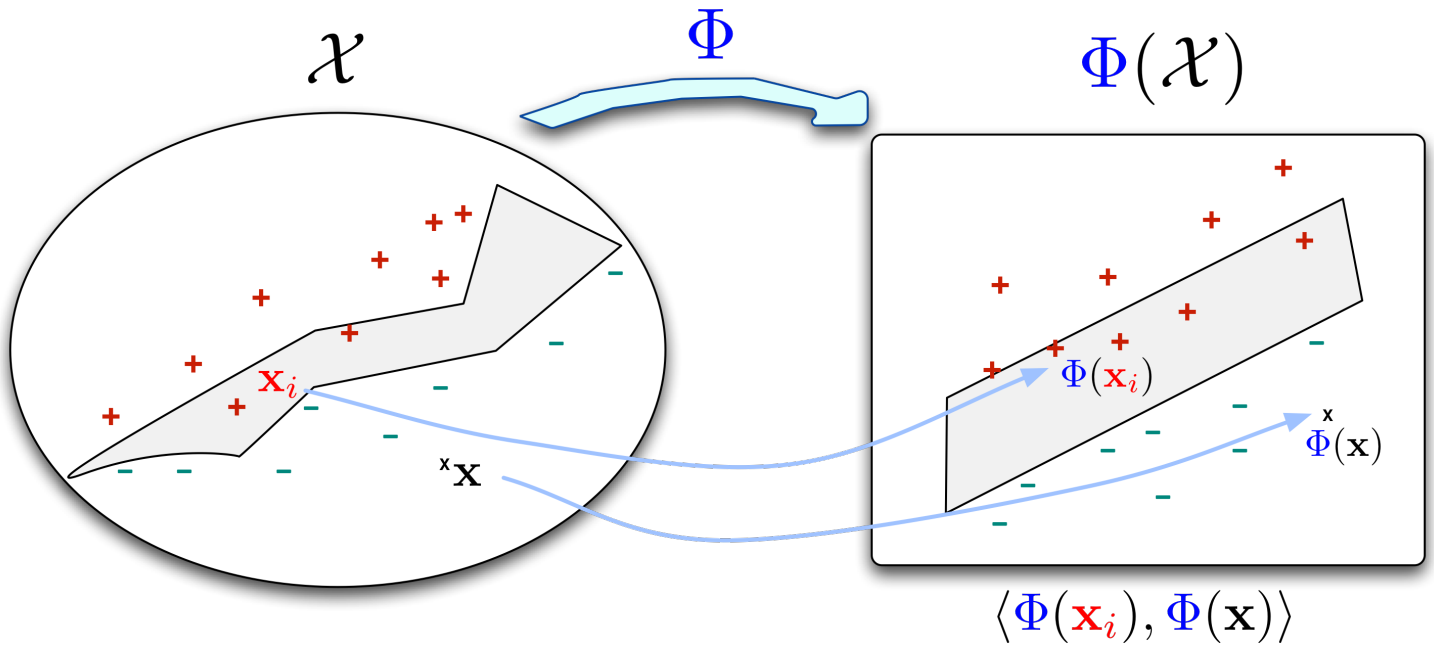
- Expression de **l'hypothèse**

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

$$\mathbf{w}^\star = \sum_{i \in \mathcal{P}_S} \alpha_i^* \, y_i \, \mathbf{x}_i$$

- Apprentissage (de **w***)
  - Optimisation d'une *fonction convexe* : la **marge**
  - Sous *contraintes linéaires* : les **exemples**
  - Passage par forme duale

SVM et méthodes à noyaux



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

Recherches actuelles : démarche générale

- Un **critère inductif** approprié

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

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

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

« Traduction » : sélection de descripteurs

- Recherche d'**hypothèse linéaire** parcimonieuse

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

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

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

- Méthodes de type LASSO

« Traduction » : classification semi-supervisée

- *l* données **étiquetées**, *u* données **non étiquetées**

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

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

*Mesure de **régularité sur les données***

$$\mathbf{h}^\top \mathcal{L}\, \mathbf{h} = \frac{1}{2} \sum_{i,j=1}^{l+u} W_{ij} \left(h(\mathbf{x}_i) - h(\mathbf{x}_i)\right)^2$$

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

« Traduction » : apprentissage multi-tâches

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

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

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

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

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

3.3 du chapitre 3. Ainsi, étant donnés 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 :

$$\min_{w} \; c\,m\,R_S(G_{\rho_w}) + a\,m\,\mathrm{dis}_{\rho_{w'}}(S_u, T_u) + \mathrm{KL}(\rho_w \| \pi_0), \tag{7.5}$$

où $\mathrm{dis}_{\rho_{w'}}(S_u, T_u) = \left| \mathop{\mathbb{E}}_{(h,h')\sim\rho_{w^2}} R_{S_u}(h,h') - \mathop{\mathbb{E}}_{(h,h')\sim\rho_{w^2}} R_{T_u}(h,h') \right|$ 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 $\ell_{\mathrm{dis}}(x) = 2\,\ell_{\mathbf{Erf}}(x)\,\ell_{\mathbf{Erf}}(-x)$ (illustrée sur la figure 7.1) , pour toute distribution D sur X, on a :

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

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

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

L'équation précédente est fortement non convexe. Afin de rendre sa résolution plus facilement contrôlable, nous remplaçons la fonction $\ell_{\mathbf{Erf}}(\cdot)$ par sa relaxation convexe $\ell_{\mathbf{Erf}_{\mathrm{cvx}}}(\cdot)$ (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 :

Industrie des bornes en généralisation

On peut étendre la démarche du PAC learning

Pour obtenir des bornes

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R(h) \leq \hat{R}(h) + \sqrt{\frac{\Omega(\text{satisfaction attentes})}{m}} \right] > 1 - \delta$$

- Si $\widehat{\text{err}}(h) = 0$ (ou petite)
- Si attente sur le monde vérifiée (ou presque)
- Avec échantillon assez grand
- Alors $\text{err}(h) < \varepsilon$ (en probabilité)

Bilan : l'empire des normes

- Une **démarche générique** et générale
  - Définition d'un **risque régularisé**
    - Traduisant des attentes sur les régularités d'intérêt
    - Assurant problème convexe
  - Algorithme d'**apprentissage** = algorithme d'**optimisation**
- Un **certificat d'excellence**
  - Bornes en généralisation, bien mathématiques
- Des **présupposés** supposés **modestes**
  - Et adaptés au « big data »

Données et questions **i.i.d.**

Mais …

Limites

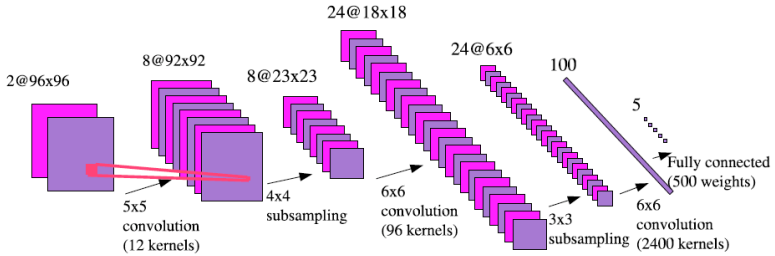
- Apprentissage **passif** et **données et questions i.i.d.**
  - Agents situés : le monde n'est pas i.i.d.
- Requiert **beaucoup** d'exemples
  - Nous sommes beaucoup plus efficaces
  - « Producteurs de théories », théories que nous testons ensuite
- Pas adapté à la recherche de **causalités**
- Pas **intégré** avec un **raisonnement**

Les machines **apprenantes** ne sont pas des machines **pensantes**

Des **barbares** aux portes

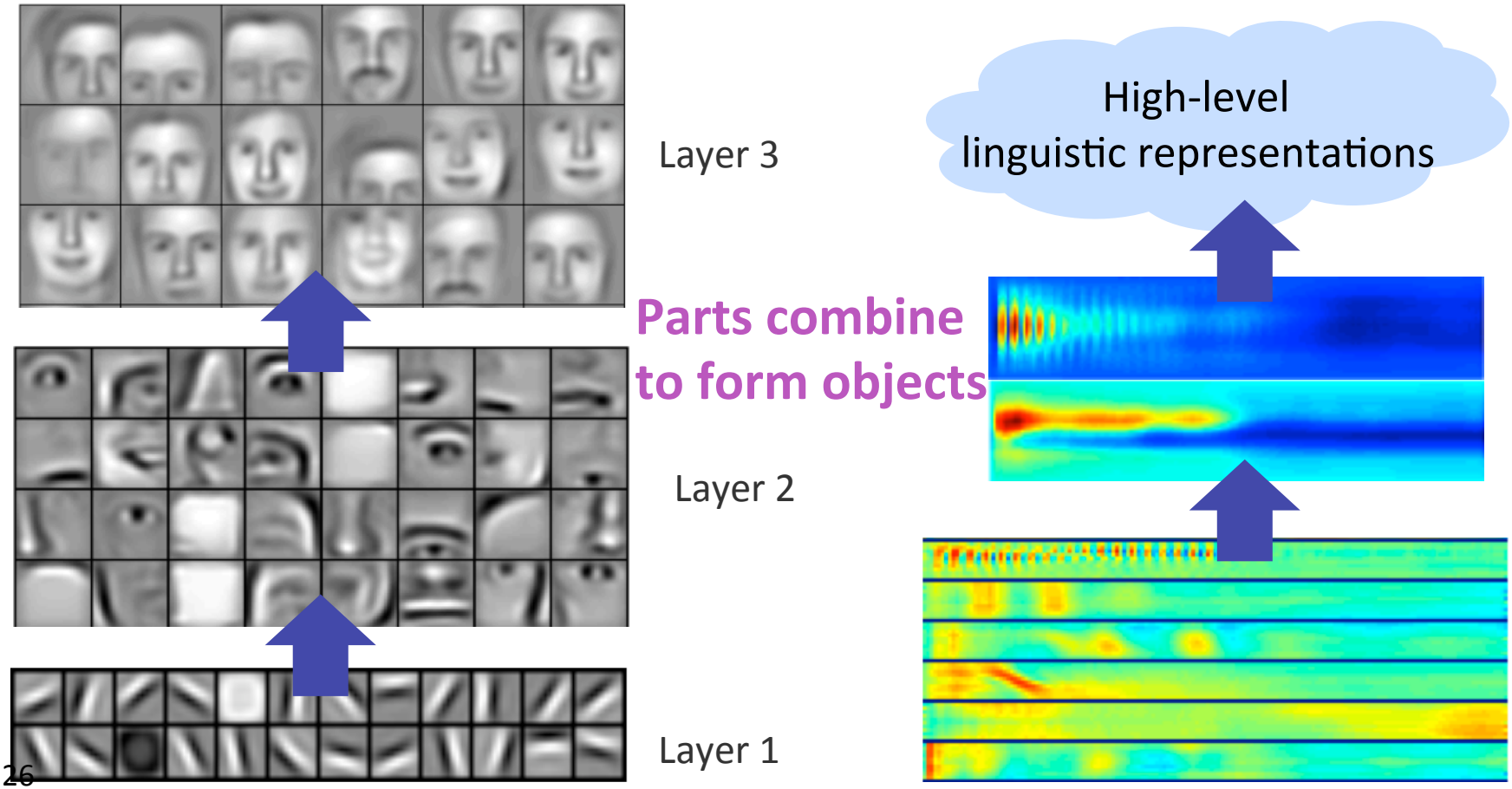
de l'empire des normes **?**

Apprentissage de connaissances structurées



- Réseaux de neurones profonds
  - Ré-introduisent la notion de **représentations abstraites**
- [Bengio & Le Cun, 07] « *Scaling Learning Towards AI* »
- [Bengio et al., 09] « *Curriculum Learning* »
- [Bottou, 11] « *From Machine Learning to Machine Reasoning* »
  - Ré-introduit la **modularité** et une **structuration** des concepts appris
- [Valiant, 00] « *A Neuroidal Architecture for Cognitive Computation* »
- [Domingos, 11-15] Statistical relational learning

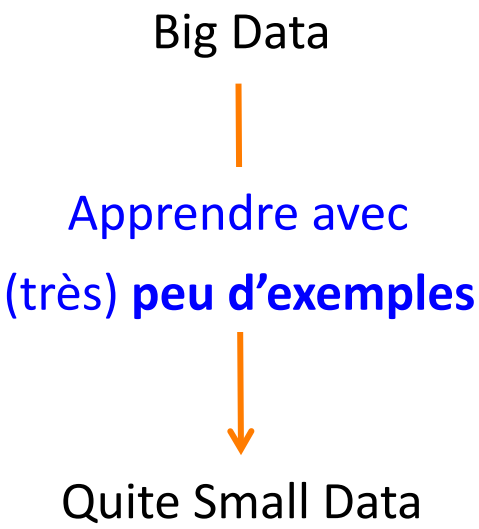
Apprentissage de représentations hiérarchiques



16

Apprentissage et **histoire** (non i.i.d.)

- Apprentissage de **causalités**
  - expériences sur le monde
  - raisonnement
- [Bengio et al., 09] « *Curriculum Learning* »
  - Choix des exemples ; choix de la séquence
- **Transfert** entre tâches
  - Quelle information partager ou transmettre ?
  - Quel ordre de tâches ?
- **Long-life** learning
  - Séquence de tâches



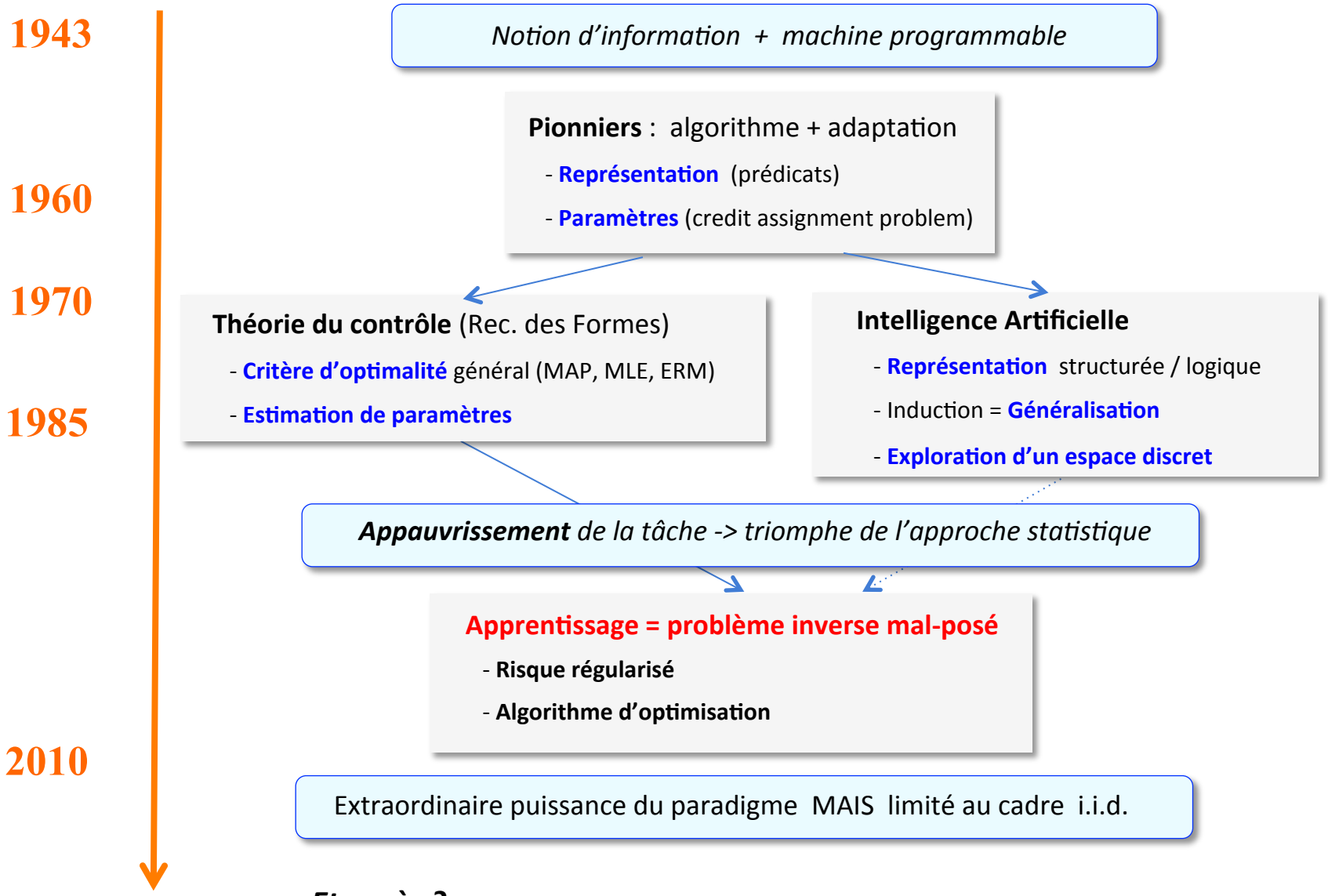
Construire de nouveaux paradigmes : nouveaux et rigoureux

- Exemples non i.i.d.
- Et pas très nombreux

analogie ; transfert ; …

- Comment **fonder** une **nouvelle théorisation** adaptée ?
  - Quels présupposés ?
  - Quel critère de performance ?
  - Quels outils formels de prédiction et de preuve ?

Rétrospective



Et après ?