Introduction à l'Intelligence Artificielle

Cours 5 – mercredi 26 octobre 2022

Adrien Revault d'Allonnes

ara@up8.edu

Université Paris 8 – Vincennes à Saint-Denis

IIA – sept. à déc., 2022

À l'épisode précédent...

- Apprentissage
  - au lieu de programmer un ordinateur *manuellement*,

  $\Rightarrow$ lui donner les moyens de se programmer **lui-même**
- Pourquoi ?
  - problèmes trop complexes
  - pas d'expert, ou trop coûteux
  - personnalisation de l'information
  - grande quantité d'information
  - parce que c'est rigolo...

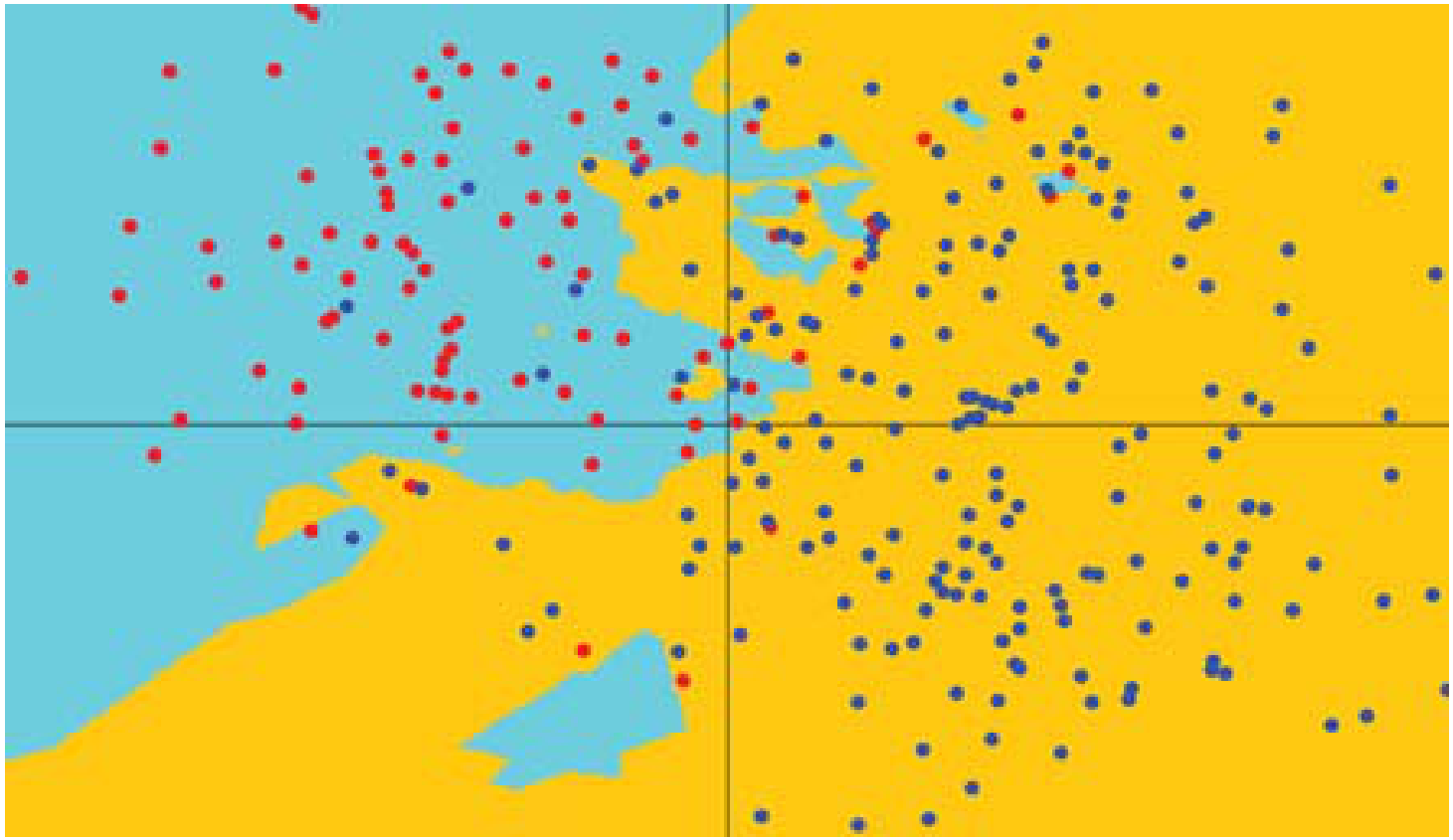
À l'épisode précédent...

- Apprentissage
  - au lieu de programmer un ordinateur *manuellement*,

  $\Rightarrow$ lui donner les moyens de se programmer **lui-même**
- Apprentissage supervisé
  - on se limite à la problématique de la classification binaire
  - fournir au système un **ensemble d'exemples étiquetés**
  - fournir au système un **ensemble d'apprentissage**
  - $f_\theta : \mathbb{R}^N \longrightarrow [-1, +1]$

  $\Rightarrow$ se programmer **lui-même** : trouver les **paramètres optimaux**

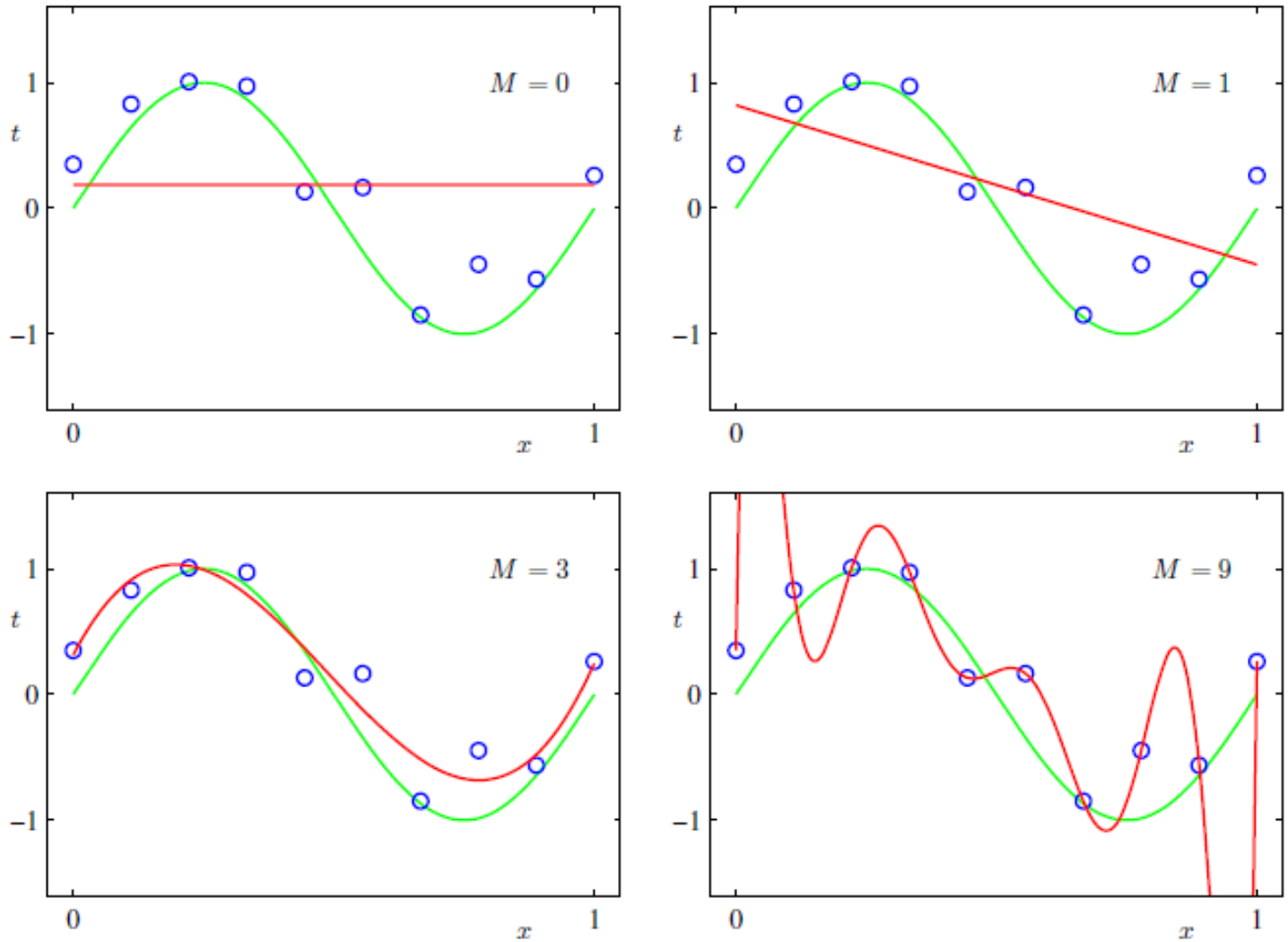
Exemple : kppv



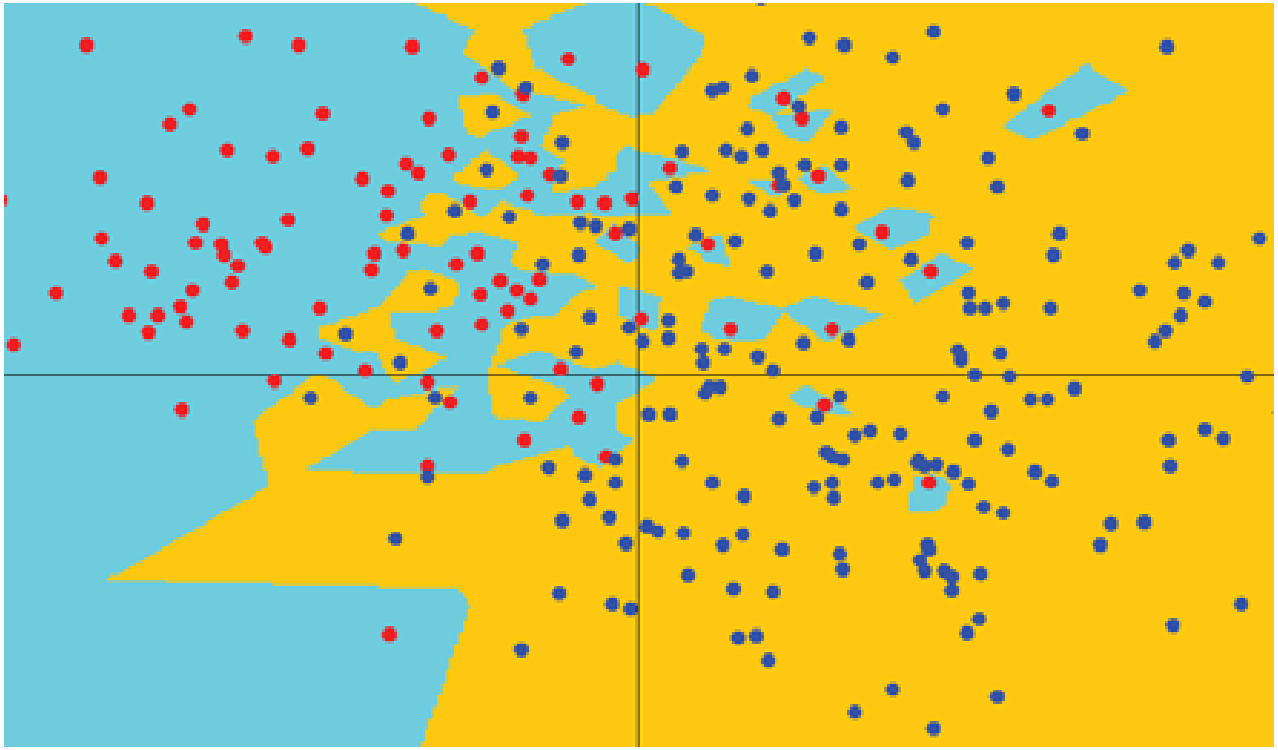
- Paramètres
  - $f_\theta =$ k-plus-proches-voisins
  - $\theta =$ ensemble d'exemples d'apprentissage

Généralisation

- Un bon modèle est un modèle qui **généralise bien :**
  - évaluation du modèle sur une **base de test**
  - compromis entre un modèle **compliqué** et un modèle **simple**

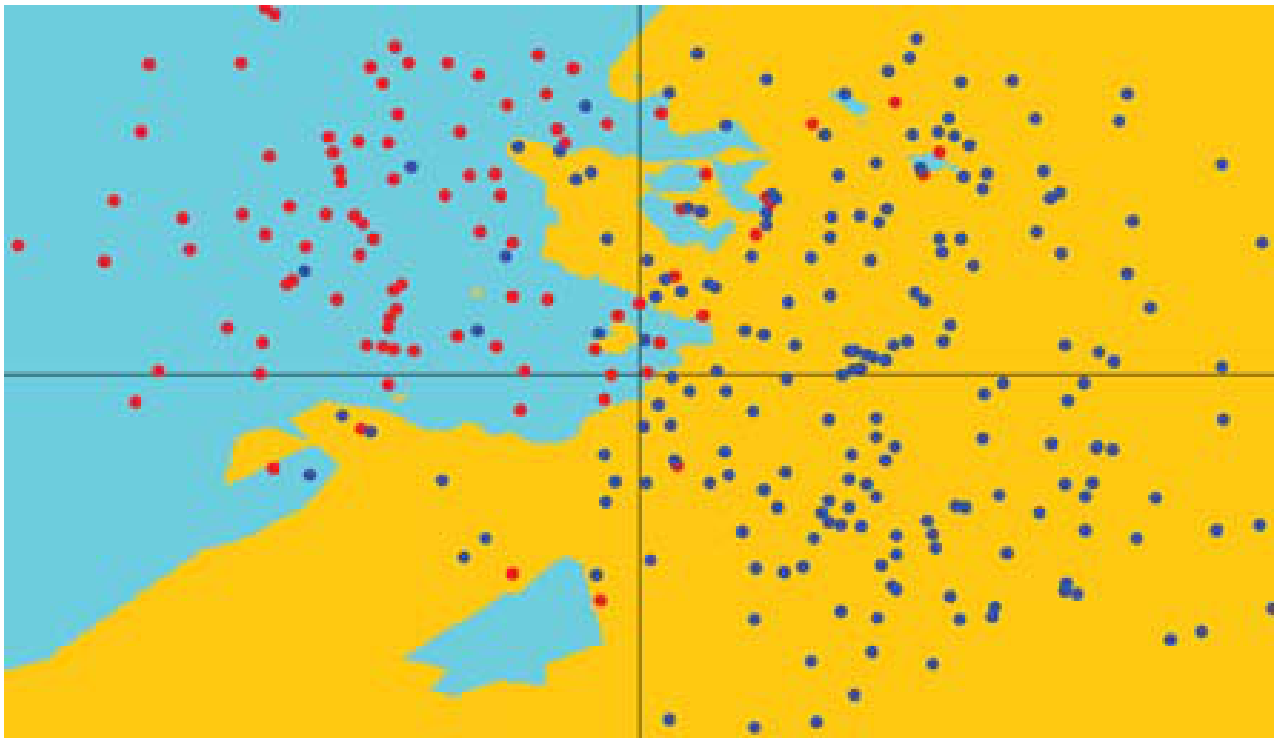


Généralisation



$k = 1$

Généralisation

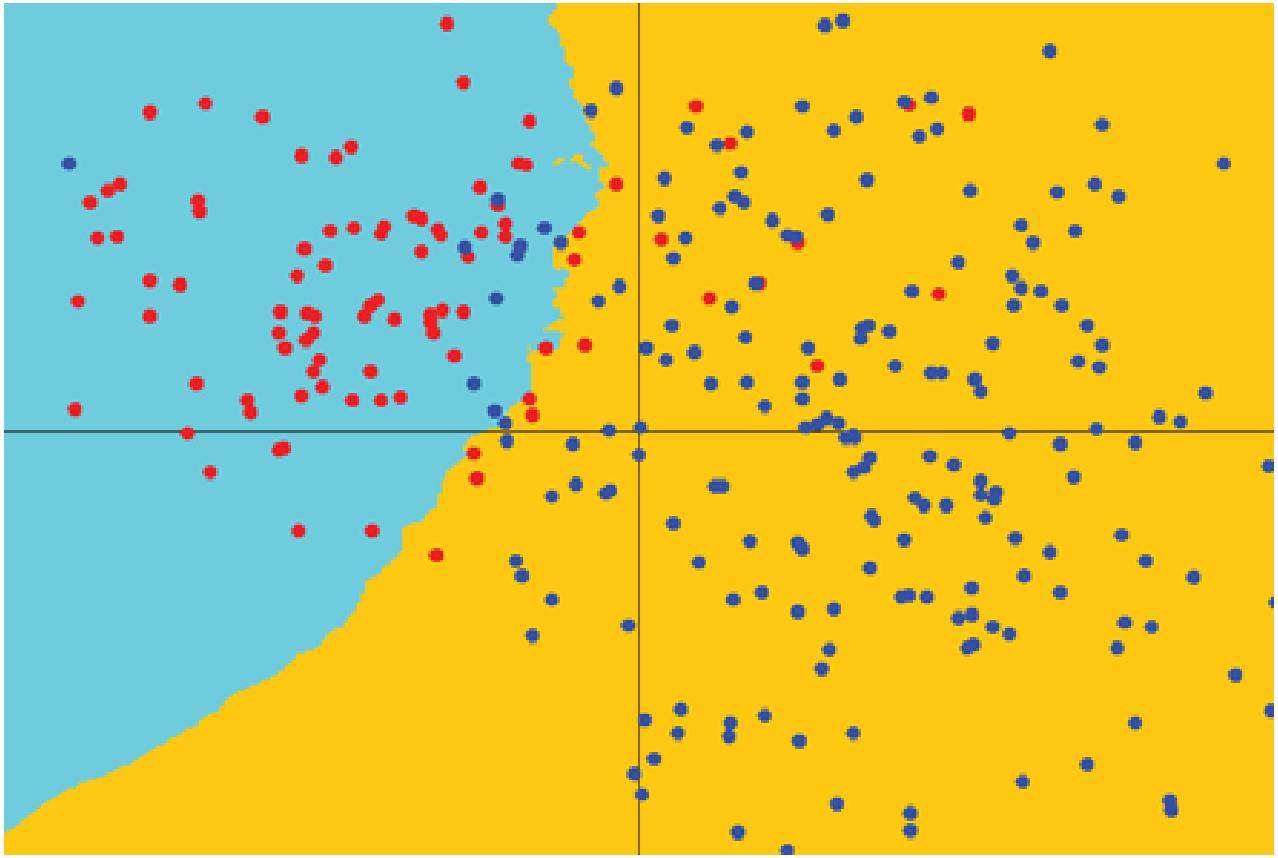


$k = 3$

Généralisation

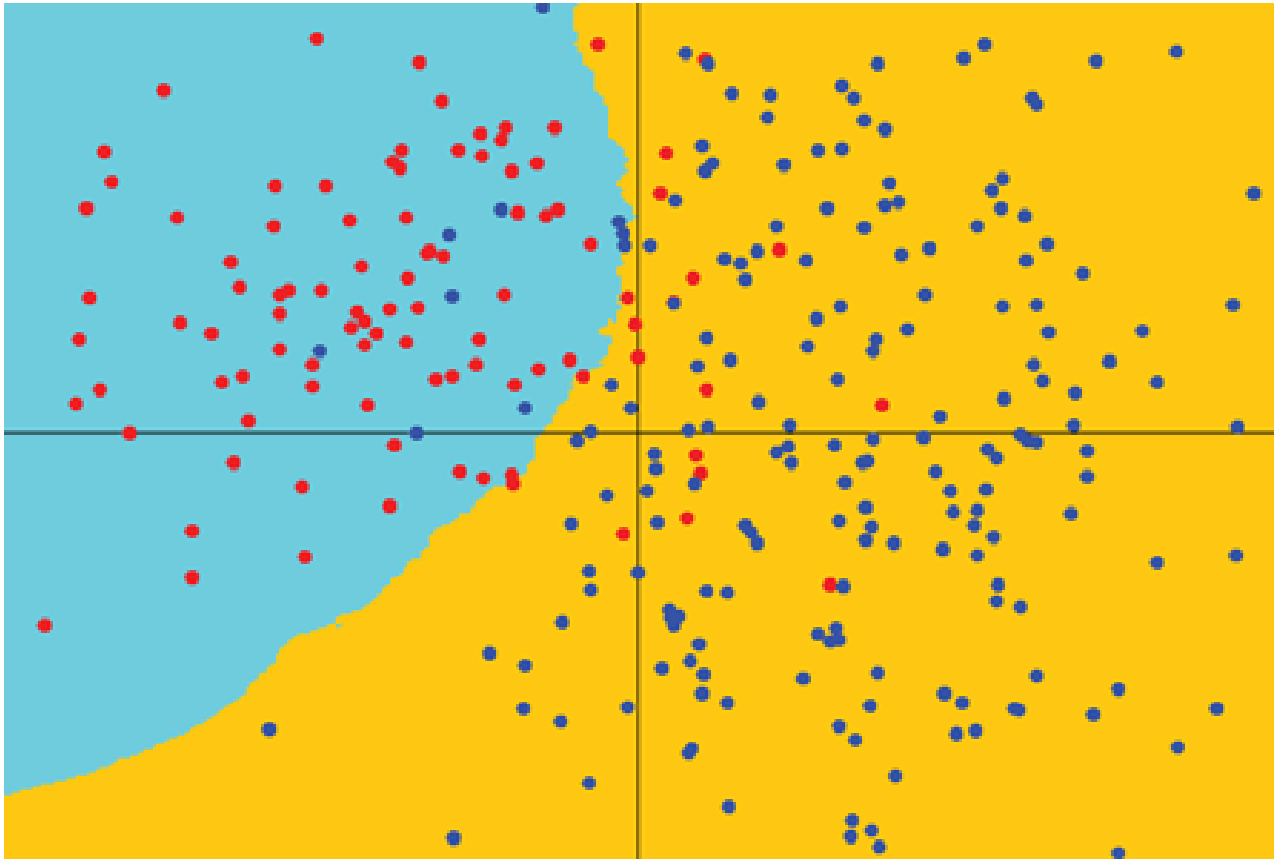
$k = 5$

Généralisation



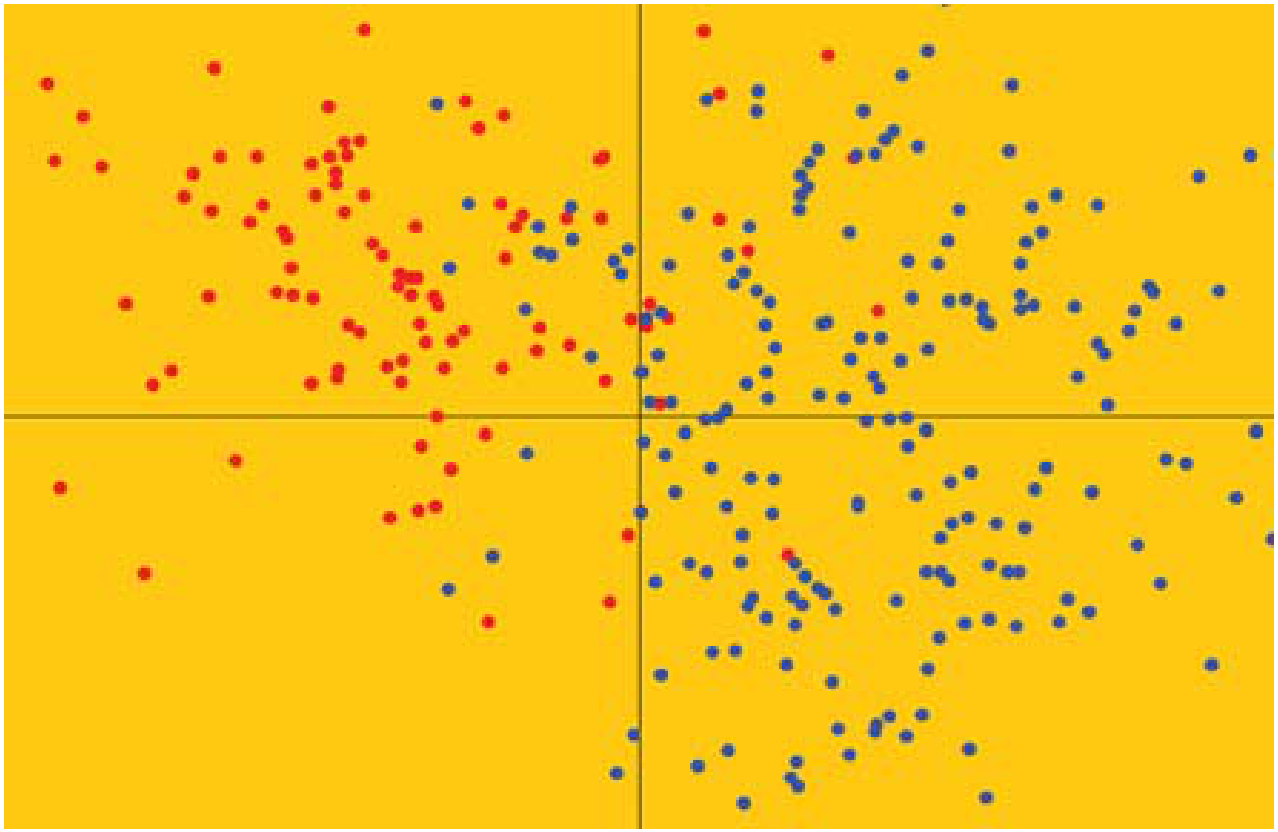
$k = 20$

Généralisation



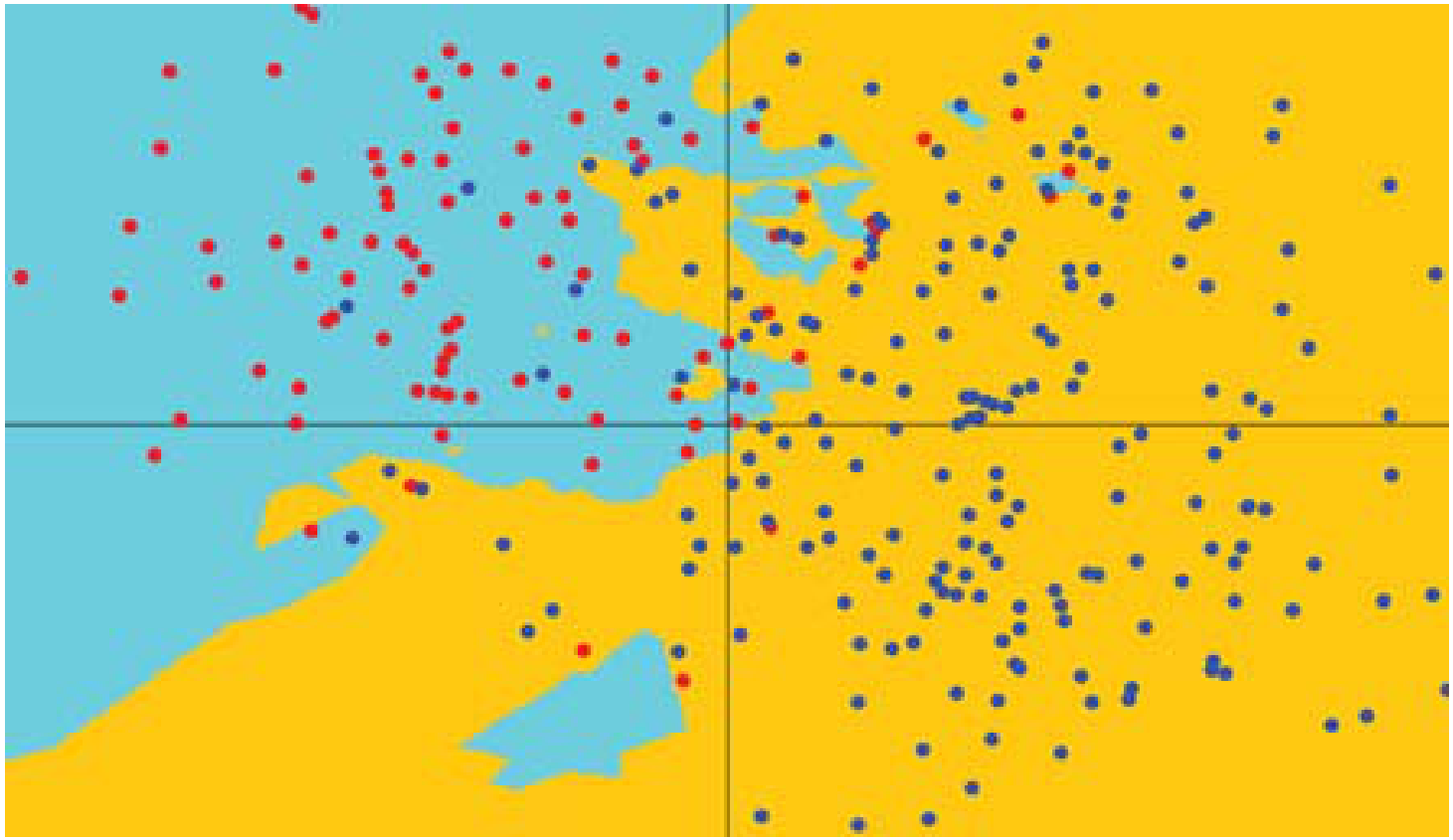
$k = 100$

Généralisation



$k = 200$

Exemple : kppv



- Paramètres
  - f_θ = k-plus-proches-voisins
  - θ = ensemble d'exemples d'apprentissage

Problématique du jour

- Constat
  - 1ppv est un modèle
    - trop complexe
    - avec trop de paramètres
- Problématique
  - quel modèle utiliser qui soit
    - simple
    - avec peu de paramètres
    - efficace

Inspiration biologique

- Neurone
  - machine simple (en apparence)
  - qui permet d'apprendre

Inspiration biologique

- Le cerveau naturel : modèle très séduisant
  - robuste et tolérant aux fautes
  - flexible, facilement adaptable
  - s'accomode d'informations
    - incomplètes,
    - incertaines,
    - vagues,
    - bruitées
    - ...
  - massivement parallèle
  - capable d'apprentissage

Inspiration biologique

- Le cerveau naturel : modèle très séduisant
  - robuste et tolérant aux fautes
  - flexible, facilement adaptable
  - s'accomode d'informations
    - incomplètes,
    - incertaines,
    - vagues,
    - bruitées
    - ...
  - massivement parallèle
  - capable d'apprentissage
- Neurones
  - 10^{11} neurones dans le cerveau humain
  - 10^4 connexions (synapses + axones) / neurone
  - potentiel d'action / période réfractaire / neuro-transmetteurs
  - signaux excitateurs / inhibiteurs

Neurones

- Les attraits pratiques
  - calculs parallélisables
  - implantables directement sur circuits dédiés
  - robustes et tolérants aux fautes
    (calculs et représentations distribués)
  - algorithmes simples
  - d'emploi très général
- Les défauts
  - opacité des raisonnements
  - opacité des résultats

Historique

- Prémisses
  - Mc Cullock & Pitts (1943) : premier modèle de neurone formel.
    Rapport neurone et calcul logique : base de l'intelligence artificielle
  - Hebb (1949) : apprentissage par renforcement du couplage synaptique
    Règle : *'Cells that fire together, wire together'*

Historique

- Prémisses
  - Mc Cullock & Pitts (1943) : premier modèle de neurone formel. Rapport neurone et calcul logique : base de l'intelligence artificielle
  - Hebb (1949) : apprentissage par renforcement du couplage synaptique
    Règle : *'Cells that fire together, wire together'*
- Premières réalisations
  - Widrow-Hoff (1960) : Adaline
  - Rosenblatt (1958–1962) : Perceptron
  - Minsky & Papert (1969) : 'Perceptrons : an introduction to computational geometry'

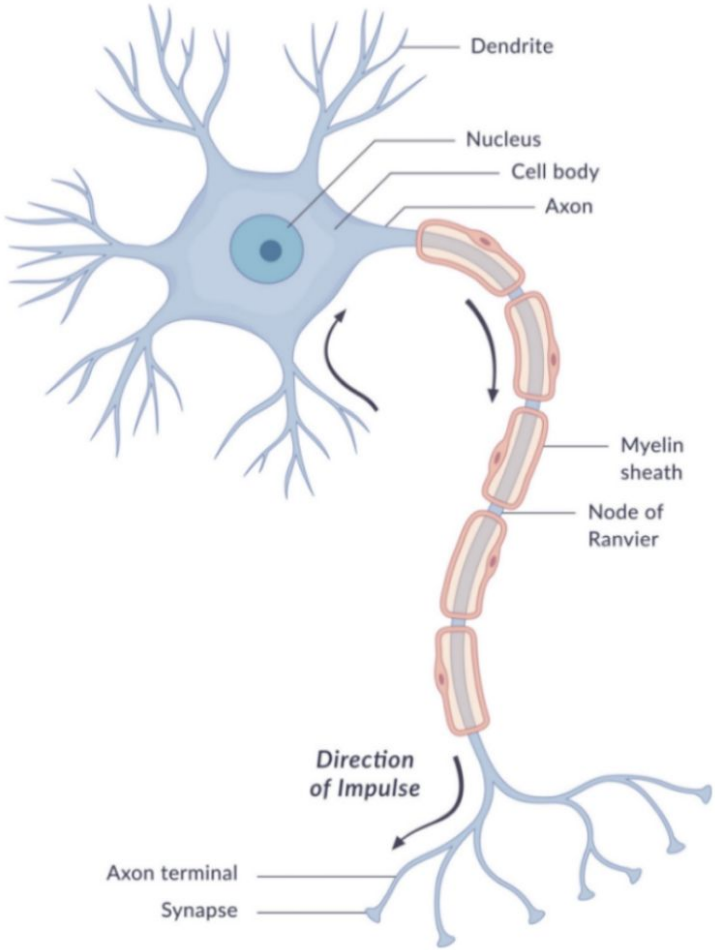
Historique

- Prémisses
  - Mc Cullock & Pitts (1943) : premier modèle de neurone formel. Rapport neurone et calcul logique : base de l'intelligence artificielle
  - Hebb (1949) : apprentissage par renforcement du couplage synaptique
    Règle : *'Cells that fire together, wire together'*
- Premières réalisations
  - Widrow-Hoff (1960) : Adaline
  - Rosenblatt (1958–1962) : Perceptron
  - Minsky & Papert (1969) : 'Perceptrons : an introduction to computational geometry'
- Nouveaux modèles
  - Kohonen (1984) : apprentissage compétitif
  - Hopfield (1982) : réseau bouclé
  - Rumelhart et al. (1986) : perceptron multicouche

Historique

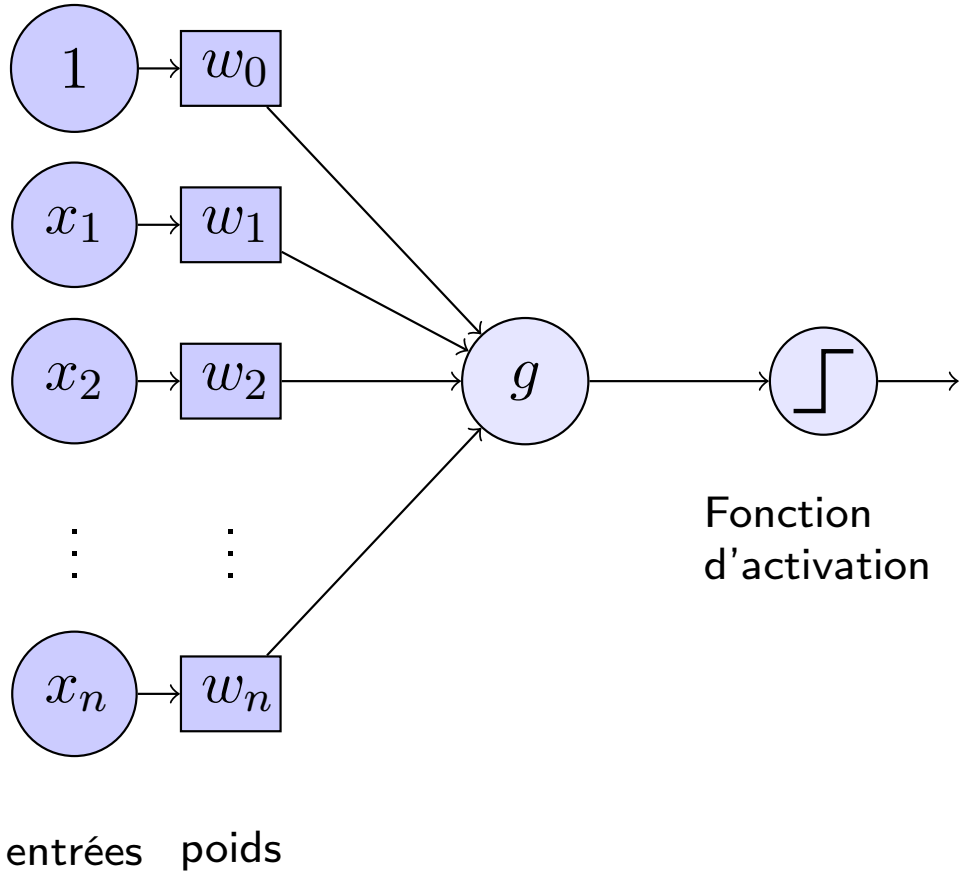
- Prémisses
  - Mc Cullock & Pitts (1943) : premier modèle de neurone formel. Rapport neurone et calcul logique : base de l'intelligence artificielle
  - Hebb (1949) : apprentissage par renforcement du couplage synaptique
    Règle : *'Cells that fire together, wire together'*
- Premières réalisations
  - Widrow-Hoff (1960) : Adaline
  - Rosenblatt (1958–1962) : Perceptron
  - Minsky & Papert (1969) : 'Perceptrons : an introduction to computational geometry'
- Nouveaux modèles
  - Kohonen (1984) : apprentissage compétitif
  - Hopfield (1982) : réseau bouclé
  - Rumelhart et al. (1986) : perceptron multicouche
- Analyse et développement
  - Vapnik (1981) : théorie du contrôle, de la généralisation

Inspiration biologique



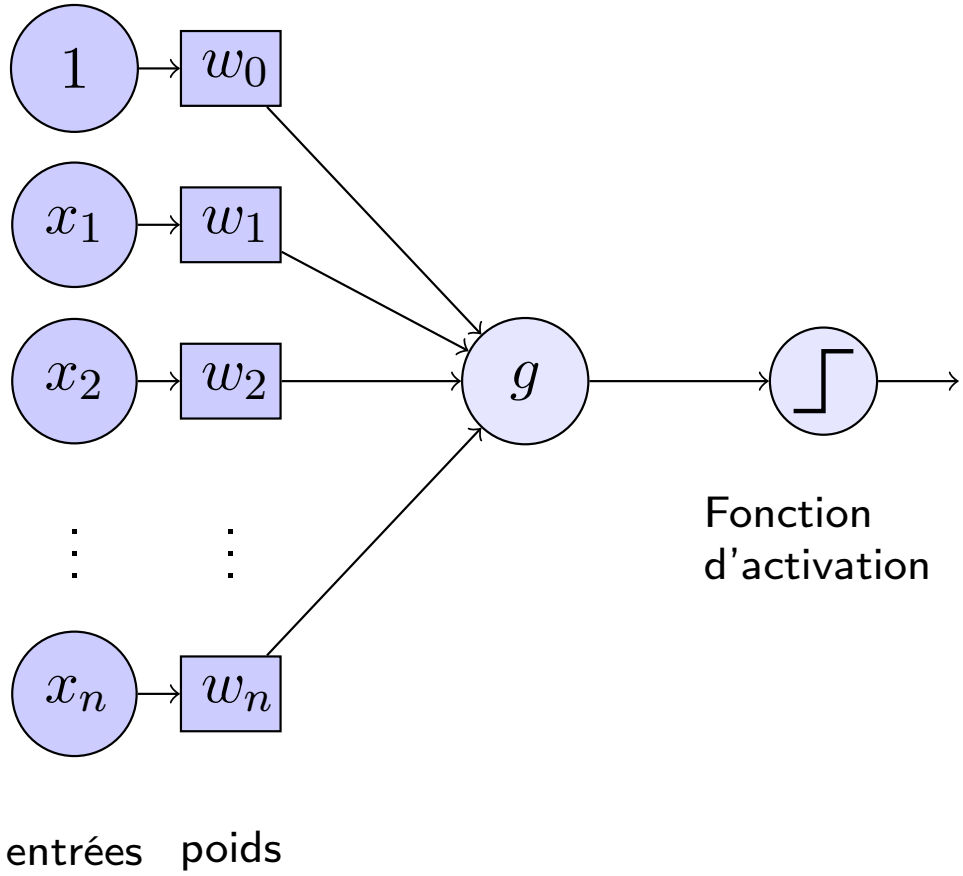
- Neurone
  - machine simple (en apparence)
  - qui permet d'apprendre

Perceptron de Rosenblatt (1960)



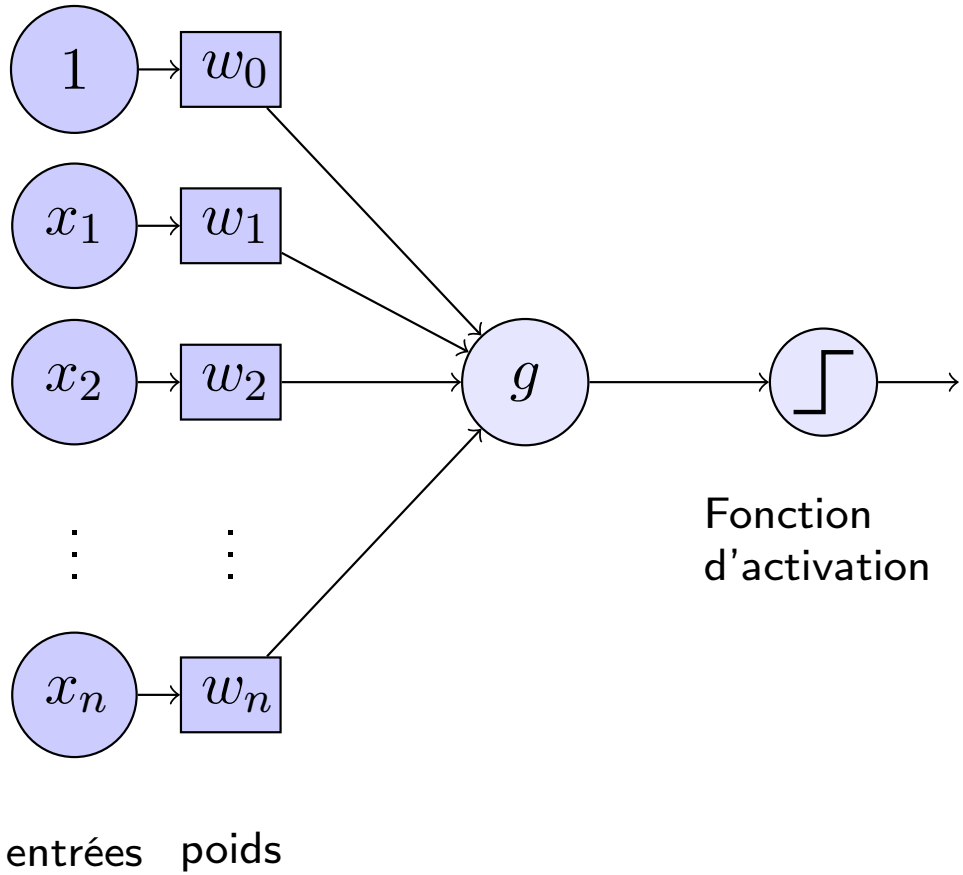
- Premier système artificiel capable d'apprendre
  - par expérience
  - y compris lorsque son instructeur commet quelques erreurs
    (ce en quoi il diffère d'un système d'apprentissage logique formel)

Perceptron de Rosenblatt (1960)



- Fonction de décision
  - $f_\theta(x) = g(w_0 + x_1 \times w_1 + x_2 \times w_2 + \ldots + x_n \times w_n)$
  - $\theta = \{w_0, w_1, \ldots, w_n\}$
  - décision : $f_\theta(x) > 0$?

Perceptron de Rosenblatt (1960)



- Fonction de décision
  - $f_\theta(x) = g(\sum_{i=0}^{n} x_i \times w_i)$ où $x_0 = 1$
  - Si $g(x) = x$, $f_\theta(x) = \langle x \cdot \theta \rangle$

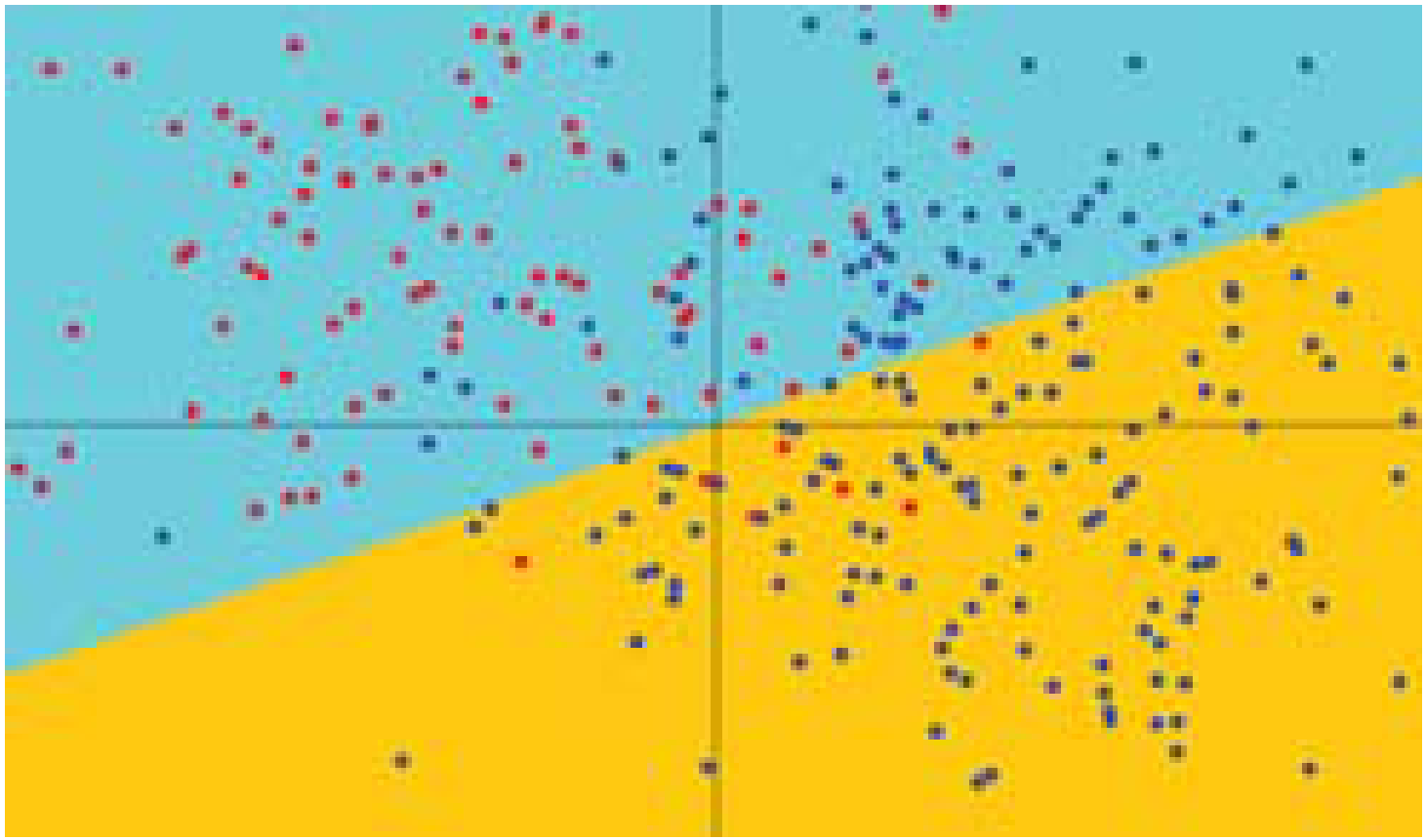
Perceptron de Rosenblatt (1960)

- Fonction de décision
  - $f_\theta(x) = \langle x \cdot \theta \rangle$

$\theta = (1; 1)$

Perceptron de Rosenblatt (1960)

- Fonction de décision
  - $f_\theta(x) = \langle x; \theta \rangle$

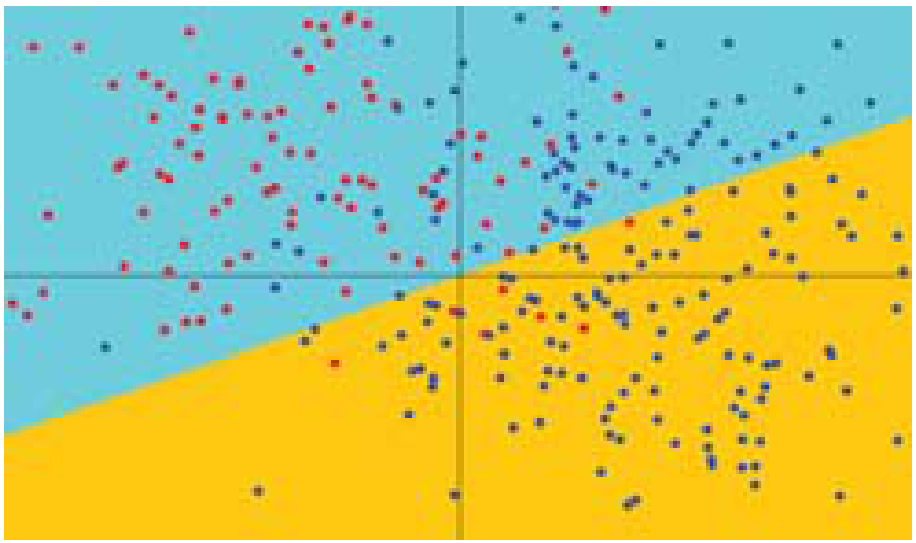


$\theta = (0.3; -0.5)$

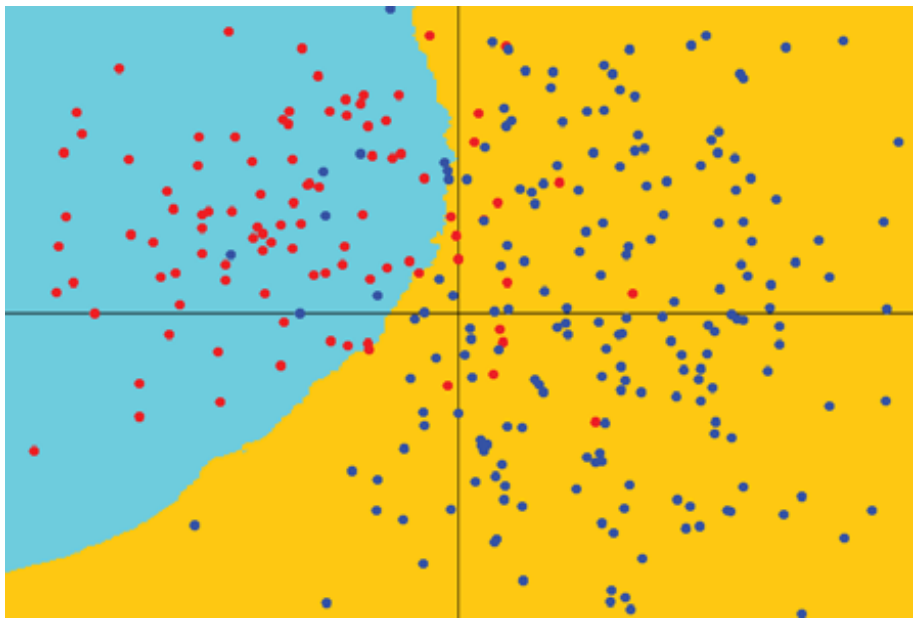
Perceptron de Rosenblatt (1960)

- Le perceptron est
  - une **machine de classificattion linéaire**
  - il modélise un **hyperplan** [1] dans une espace vectoriel

[1] un truc tout droit qui coupe en deux



perceptron

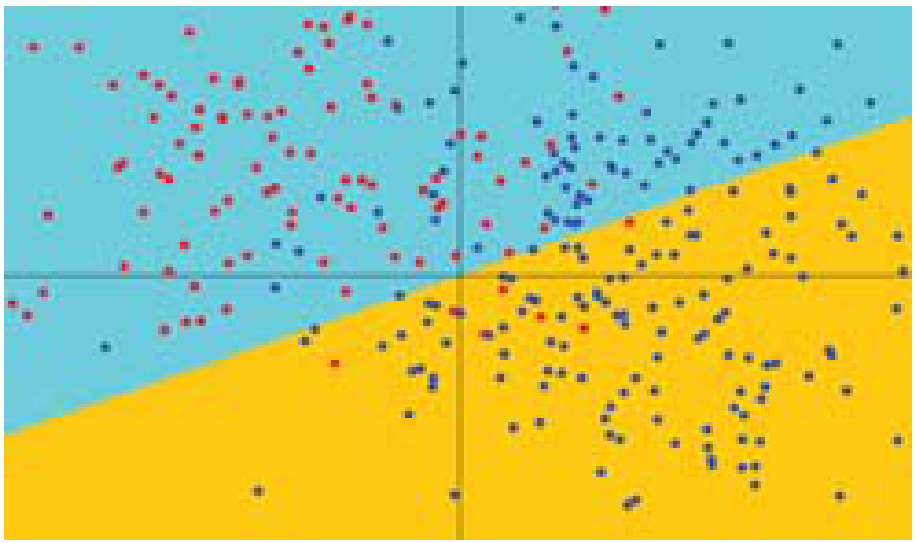


kppv

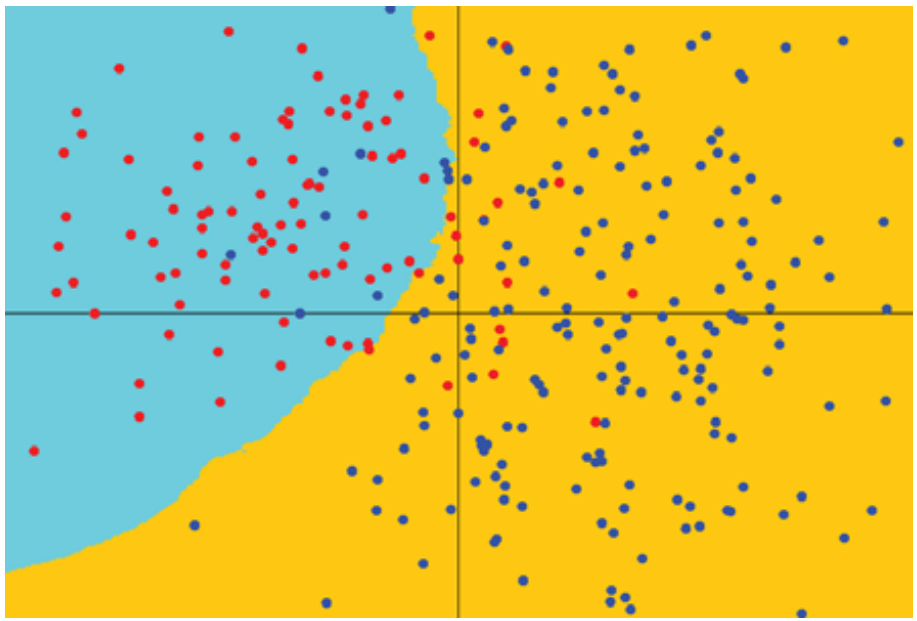
Perceptron de Rosenblatt (1960)

- Le perceptron est
  - une **machine de classificattion linéaire**
  - il modélise un **hyperplan** [1] dans une espace vectoriel

[1] un truc tout droit qui coupe en deux

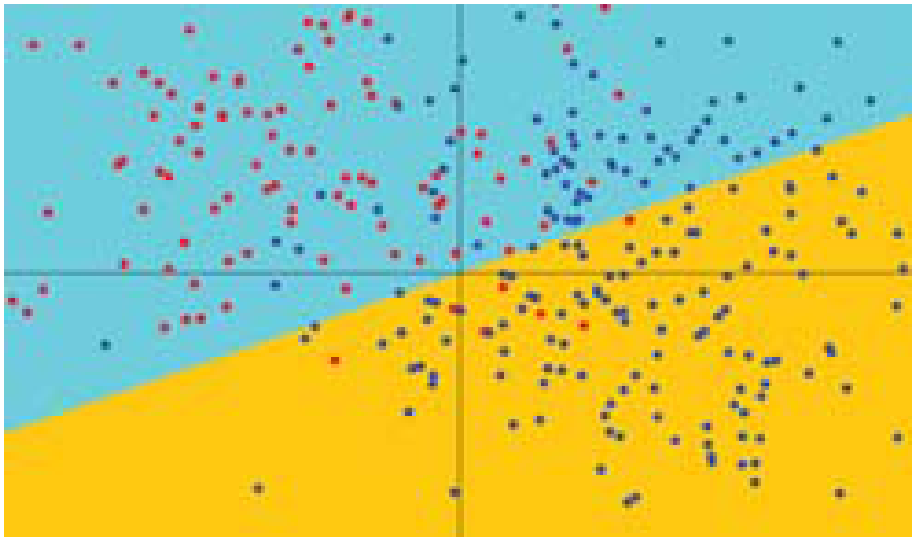


2 paramètres

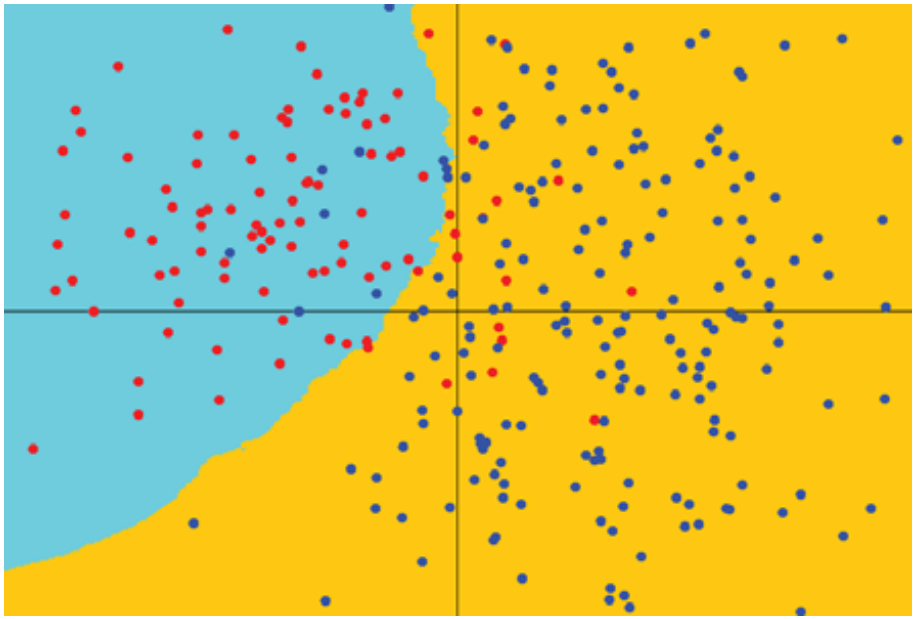


600 paramètres

Séparabilité linéaire



temps de calcul très court

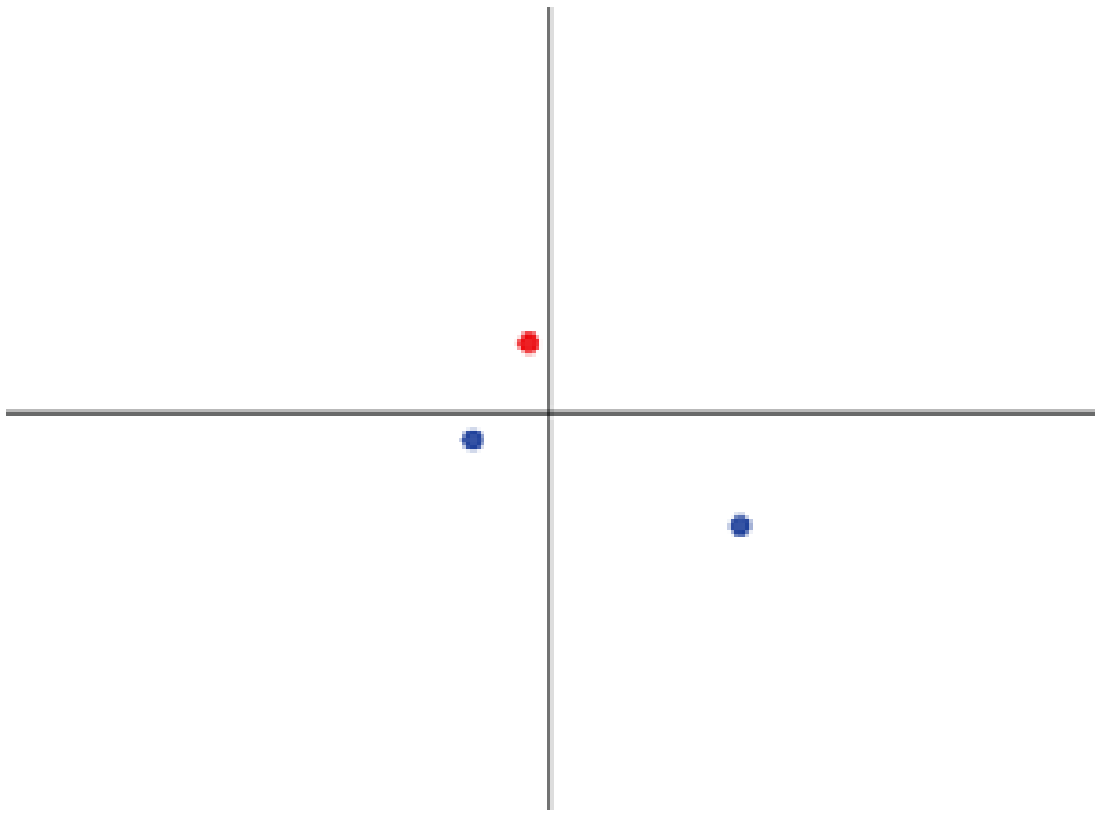


temps de calcul prohibitif

- Problème
  - le perceptron modélise une **frontière linéaire**
  - est-ce que ça peut apprendre qqc ?

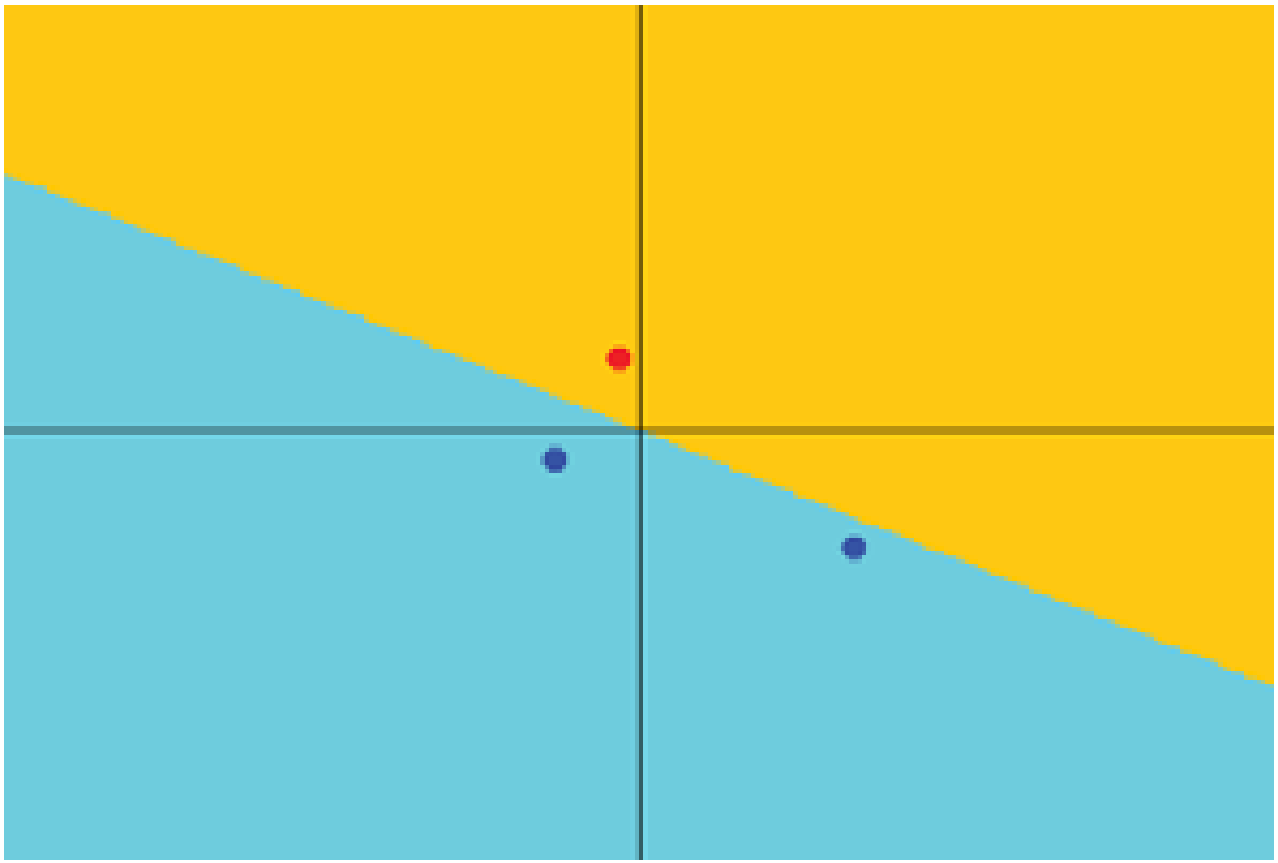
Séparabilité linéaire

- Problème
  - le perceptron modélise une **frontière linéaire**
  - est-ce que ça peut apprendre qqc ?



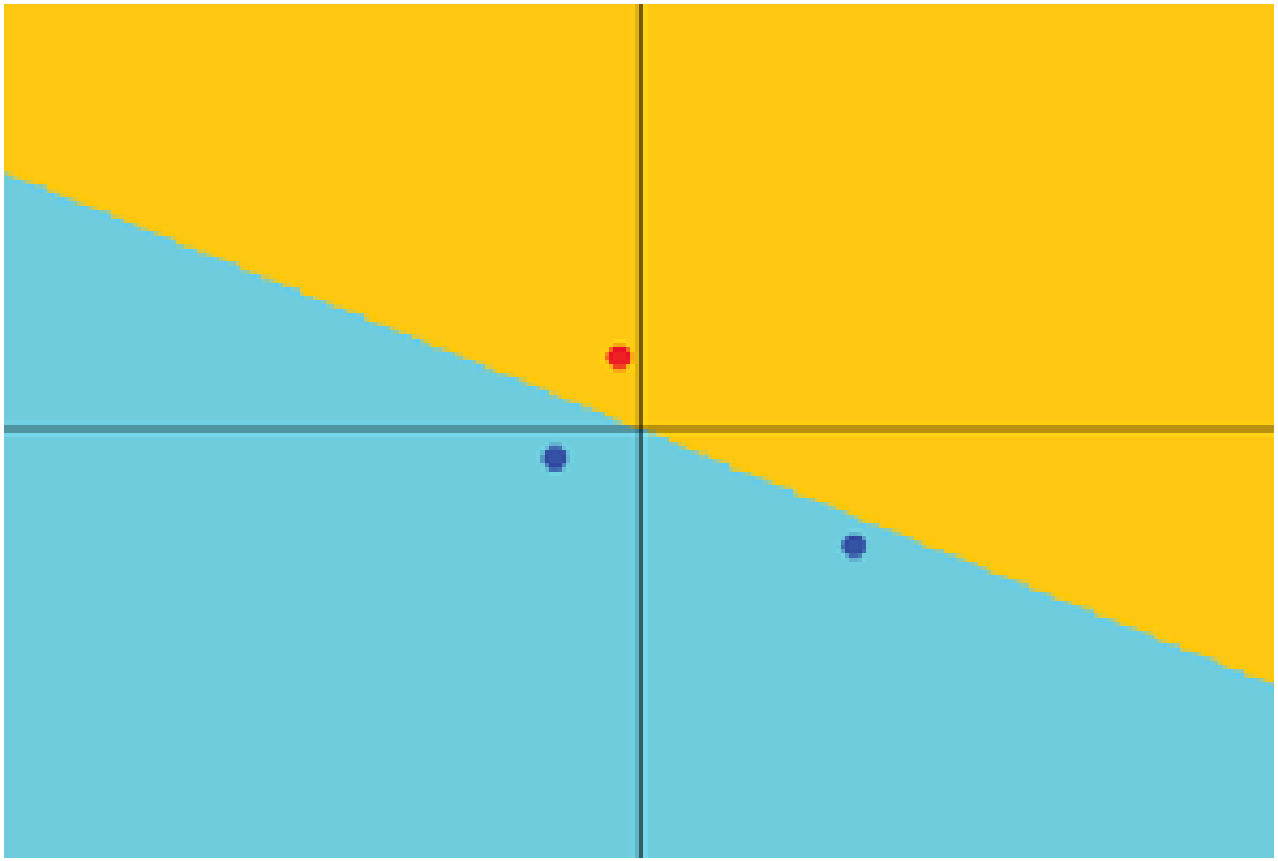
Séparabilité linéaire

- Problème
  - le perceptron modélise une **frontière linéaire**
  - est-ce que ça peut apprendre qqc ?



Séparabilité linéaire

- Problème
  - le perceptron modélise une **frontière linéaire**
  - est-ce que ça peut apprendre qqc ?



- 3 points [1] en dimension 2 sont toujours **linéairement séparable**

1. non alignés

Séparabilité linéaire

- Généralisation
  - 3 points[1] en dimension 2 sont toujours **linéairement séparable**
  - 4 points[1] en dimension 3 sont toujours **linéairement séparable**
  - ...
  - $n+1$ points[1] en dimension n sont toujours **linéairement séparable**

- Perceptron : modèle adapté à l'apprentissage dans les **espaces de grande dimension**
  - texte
  - image
  - séries financières
  - ...

1. non alignés

Algorithme du Perceptron

- Initialiser θ aléatoirement
- Répéter
  - pour $i = 1$ à N
    - si $y^i \times \langle \theta \cdot x^i \rangle \leq 0$ alors $\theta = \theta + \eta \cdot y^i \cdot x^i$
- Jusqu'à convergence

- Algorithme de correction d'erreur
- η peut être fixe ou variable (décroissant)

Théorème de convergence du Perceptron : Novikov 1962

- Si
  - $\exists R \,|\, \forall x, ||x|| \leq R$
  - les données peuvent être séparées avec une marge ρ

$$\sup_{\theta} \min_{i} y^i \times \langle x^i \cdot \theta \rangle \geq \rho$$

  - l'ensemble d'apprentissage peut être présenté suffisamment de fois
- Alors

$\Rightarrow$ L'algorithme converge après, au plus, $\frac{R^2}{\rho^2}$

Propriétés de l'erreur de généralisation

- Si
  - les données sont séparables
  - elles sont en nombre infini
  - après la $k^{\text{ème}}$ correction, les $m_k = \frac{1+2\ln(k)-\ln(\eta)}{-\ln(1-\varepsilon)}$ données présentées sont reconnues correctement

- Alors

  $\Rightarrow$ le perceptron converge en $I \leq \frac{1+4\ln(\frac{R}{\rho})-\ln(\eta)}{-\ln(1-\varepsilon)} . \frac{R^2}{\rho^2}$ étapes

  $\Rightarrow$ avec une probabilité $1-\eta$, l'erreur de test est $\leq \varepsilon$

Le Perceptron : résumé

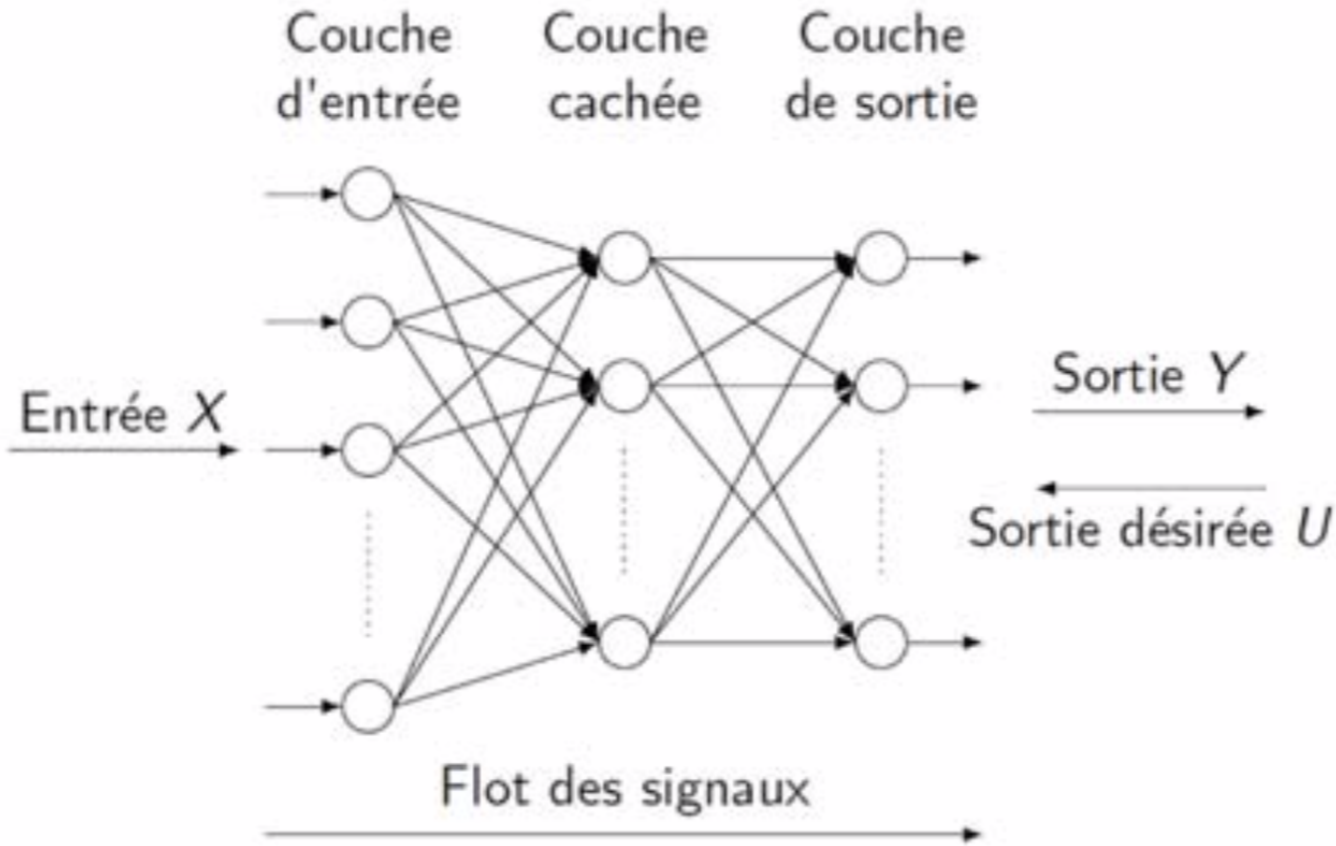
- Perceptron
  - inventé en 1960 – premier modèle d'apprentissage...
  - ...dont on a une preuve de convergence (il apprend qqc) ;
  - permet de faire de la **classification linéaire**...
  - ...ce qui est bien suffisant, en grande dimension ;
  - algorithme itératif d'apprentissage...
  - ...qui ne corrige que s'il y a une erreur.

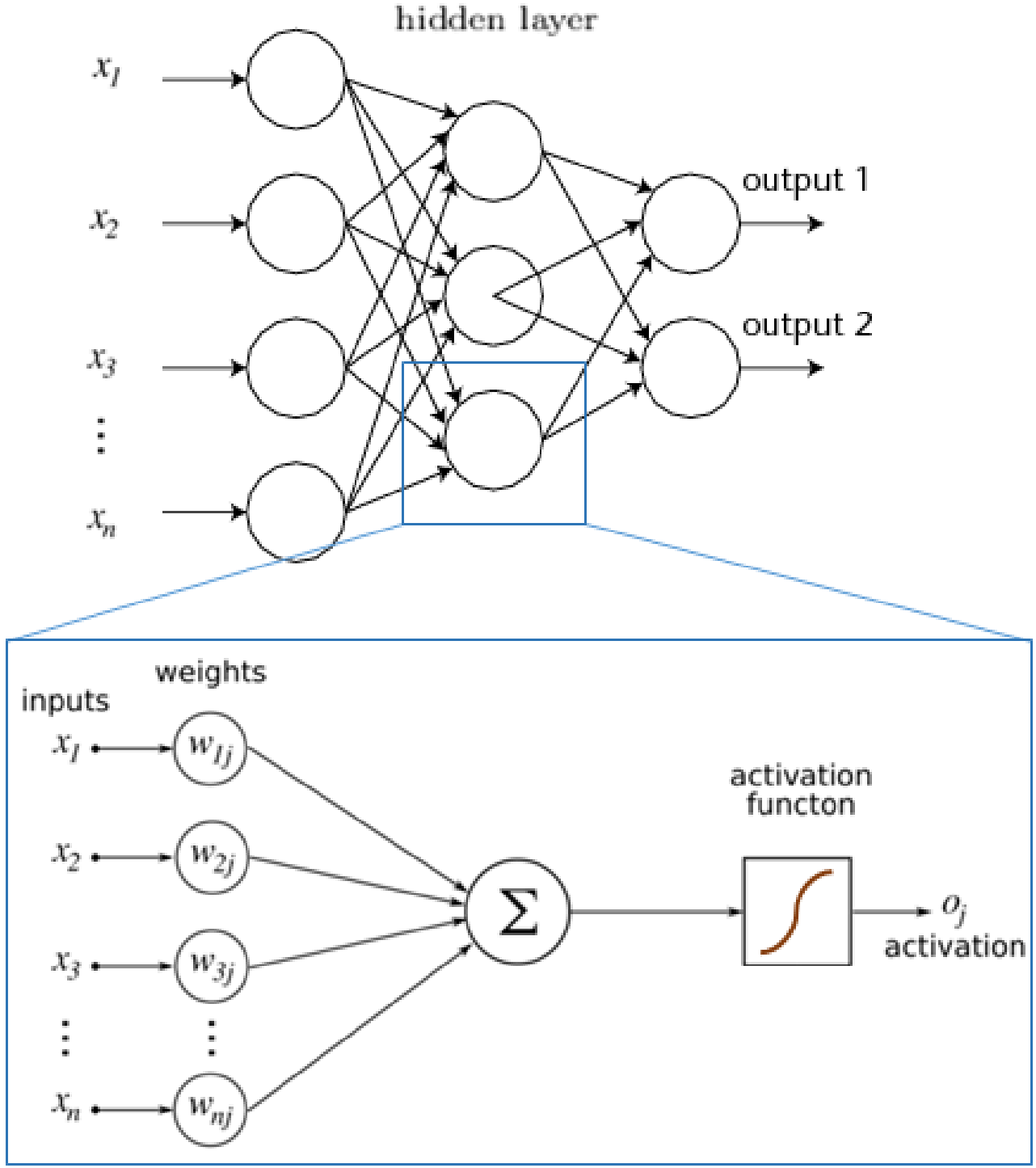
Réseau de neurones

- Idée
  - si on en mettait plusieurs bout à bout ?

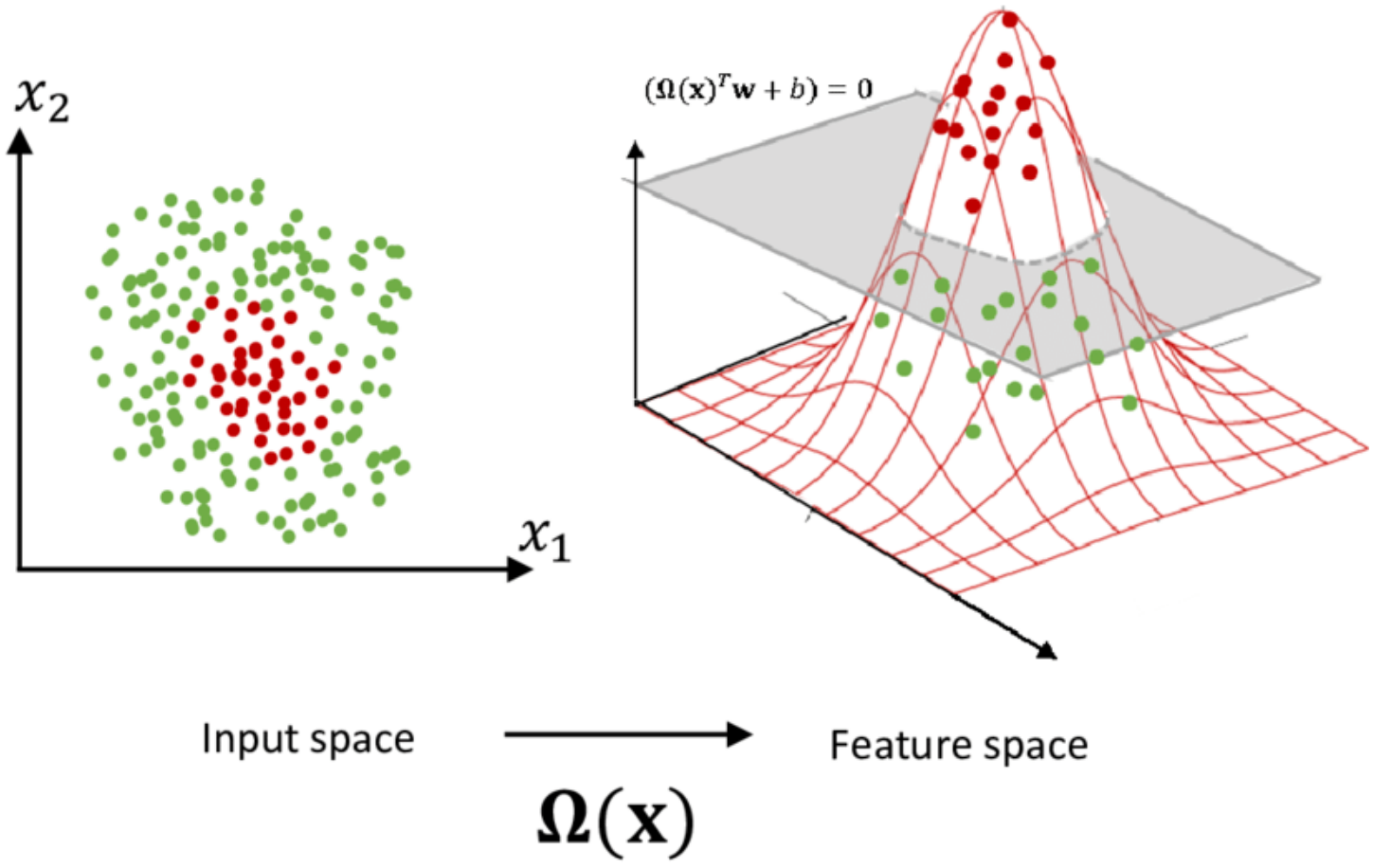
- Réseau de neurones
  - apparition en 1986
  - constitué de plusieurs couches de neurones

Réseau de neurones



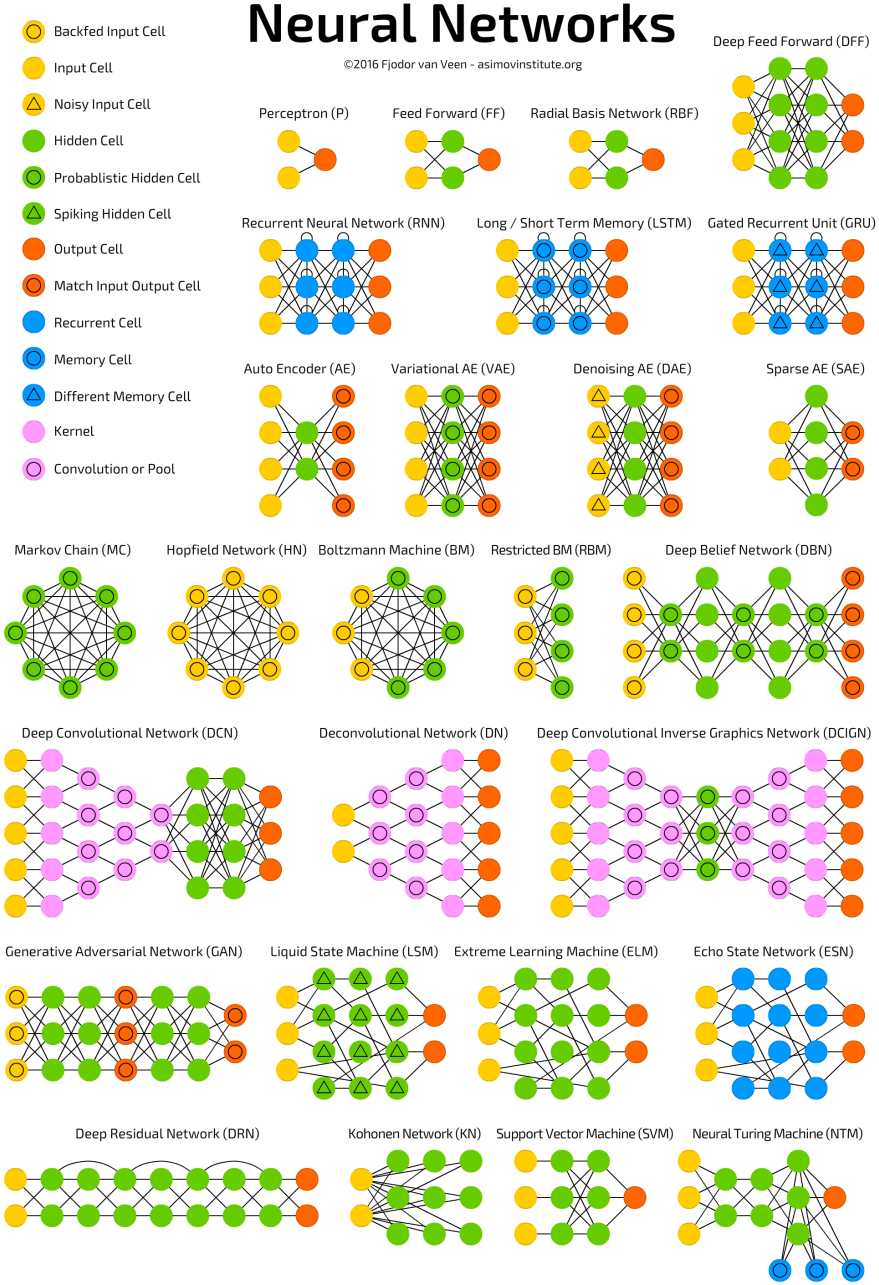

hidden layer
x_1
x_2
x_3
x_n
output 1
output 2
inputs
weights
w_{1j}
w_{2j}
w_{3j}
w_{nj}
Σ
activation
functon
o_j
activation

Réseau de neurones



A mostly complete chart of

Neural Networks



www.asimovinstitute.org/neural-network-zoo/