Apprentissage automatique et applications

Matthieu Vilain

Stage L2 laboratoire ETIS

Tuteur : Pierre Andry

Etat de l'art

1951 – M.Minsky : premier réseau de neuron

1957 – F.Rosenblatt : Perceptron

1969 – Minsky & Papert : livre « Perceptrons »

Limites réseaux de neurones → AI winter

1986 – Rumrlhart & Hinton : backpropagation

1989 – Watkins & Sutton : reinforcment learning

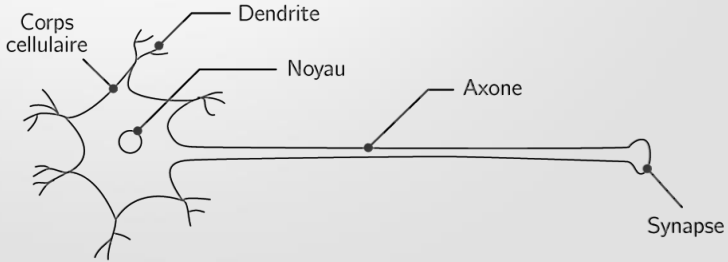
1997 – IBM : DeepBlue bat Kasparov aux échecs

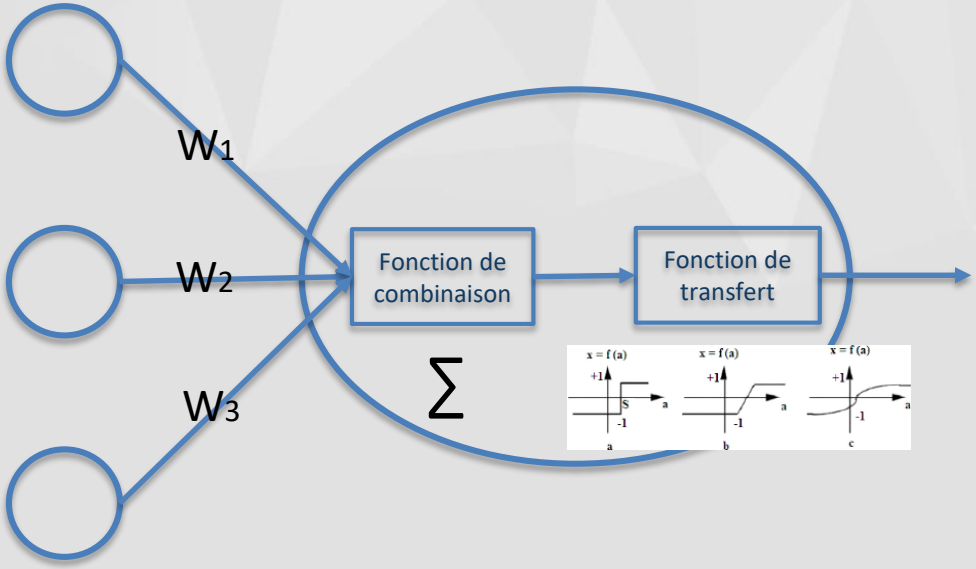
1998 – Y.LeCun : Deep learning

2016 – Google : AlphaGo bat Lee Sedol aux Go

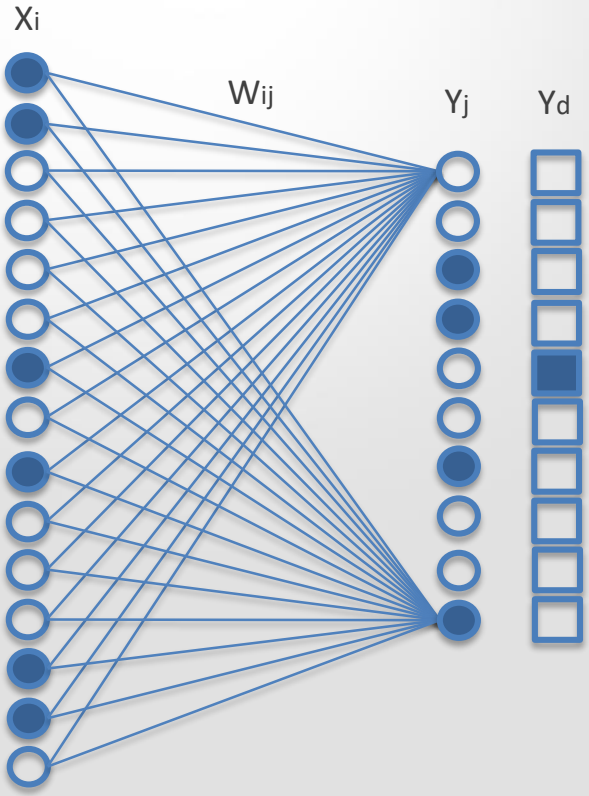
Réseaux de neurones

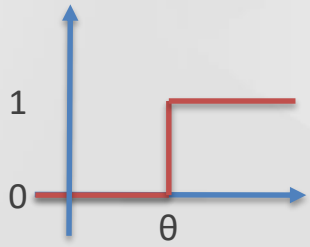
Modélisation simpliste du fonctionnement d'un neurone biologique



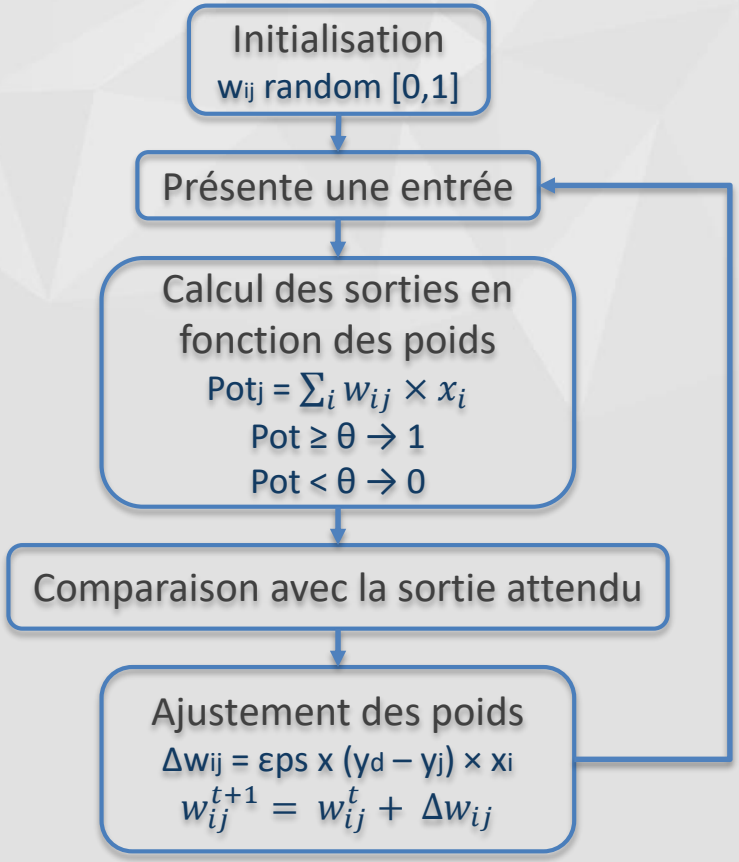


Fonctionnement perceptron simple



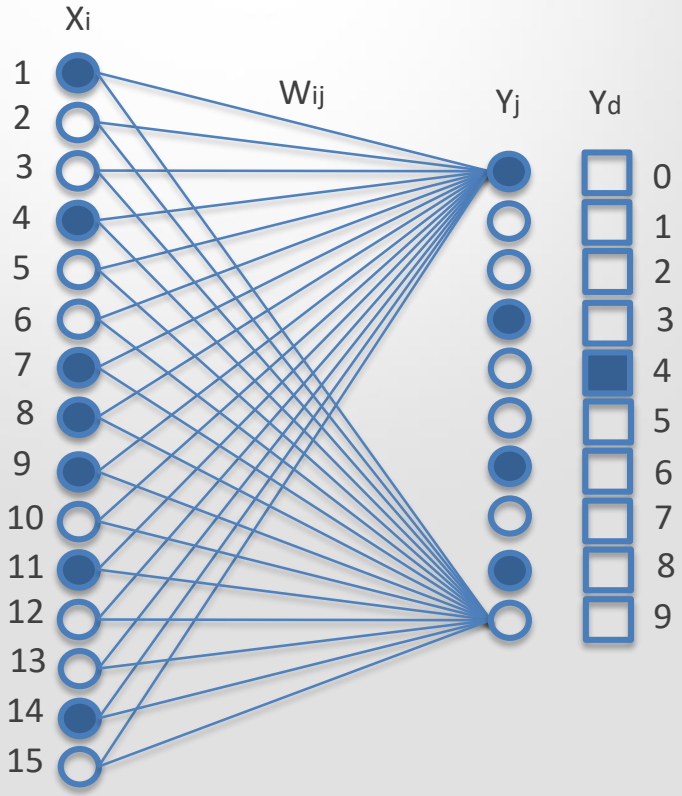


$0 \leq \varepsilon ps \leq 1$



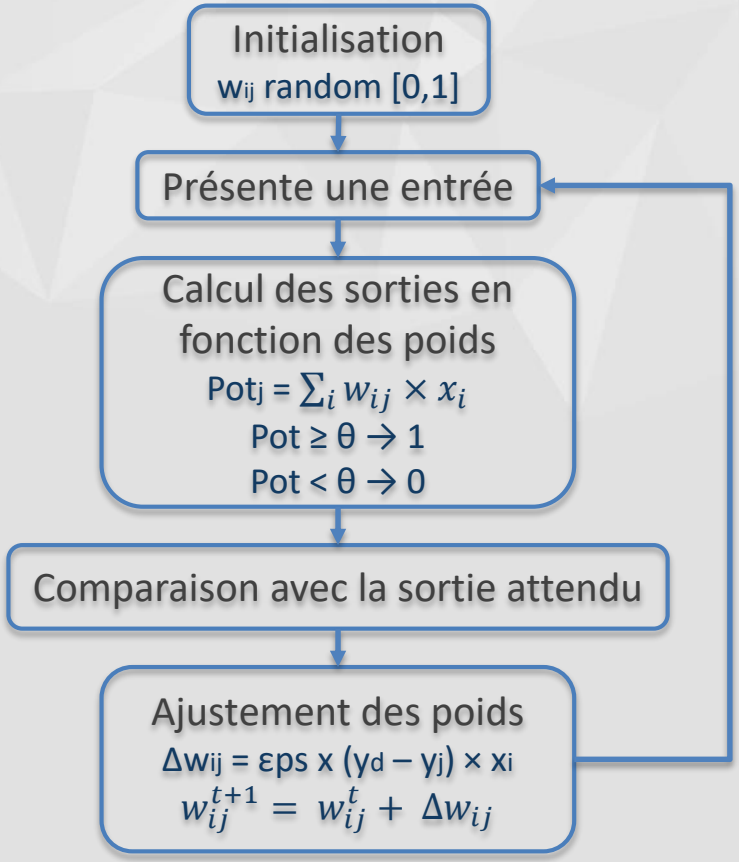
F.Rosenblatt 1957

Fonctionnement perceptron simple

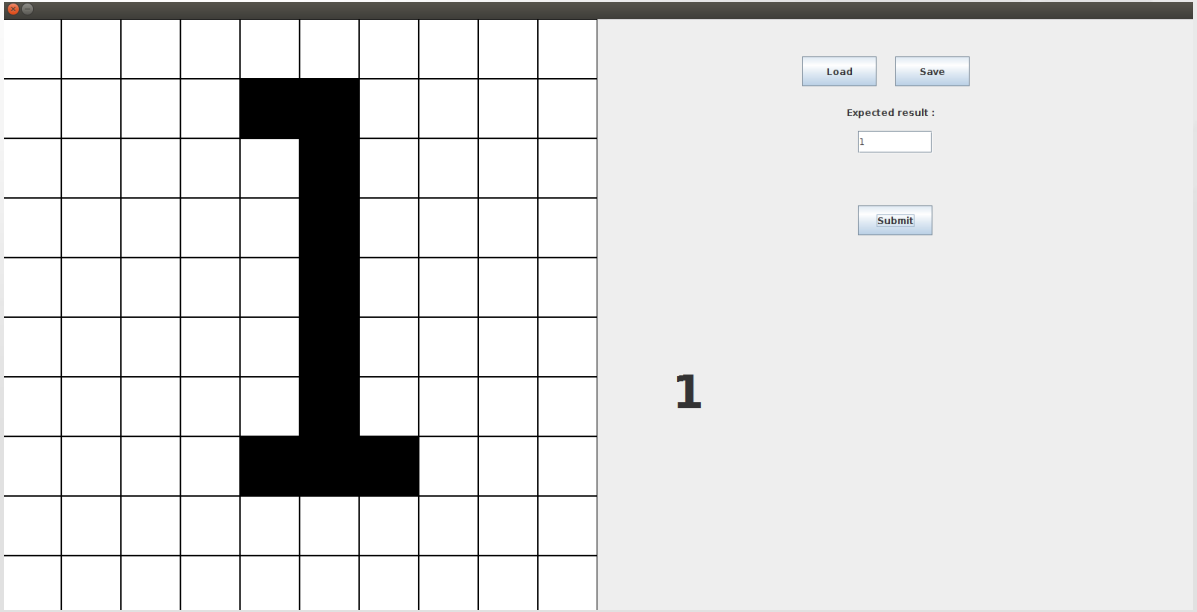


1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

F.Rosenblatt 1957



Application



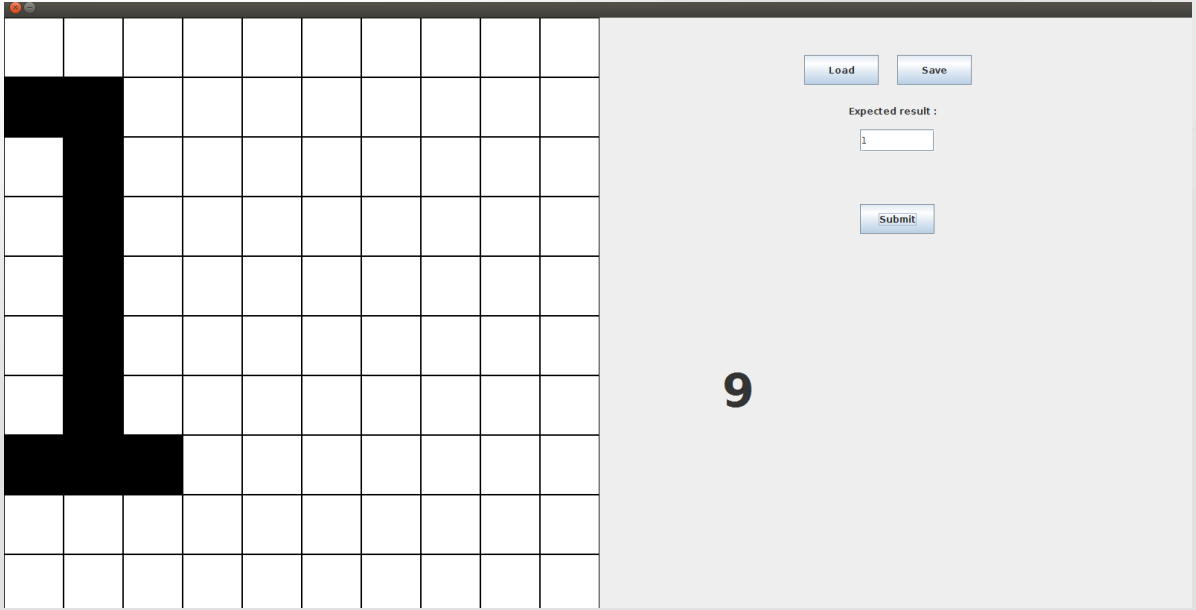
Limites

- Position dans la grille
- Temps d'apprentissage
- Non ergonomique

Rétine : 100 neurones

Sortie : 10 neurones

Application



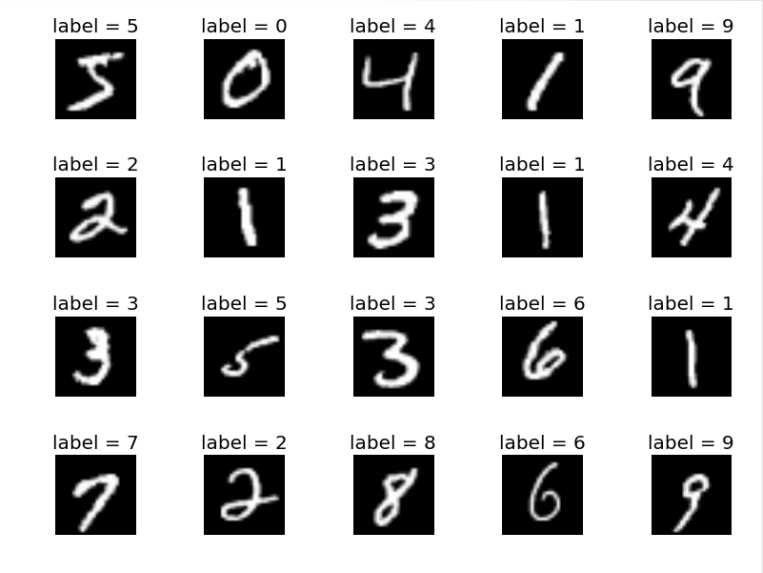
Limites

- Position dans la grille
- Temps d'apprentissage
- Non ergonomique

Rétine : 100 neurones

Sortie : 10 neurones

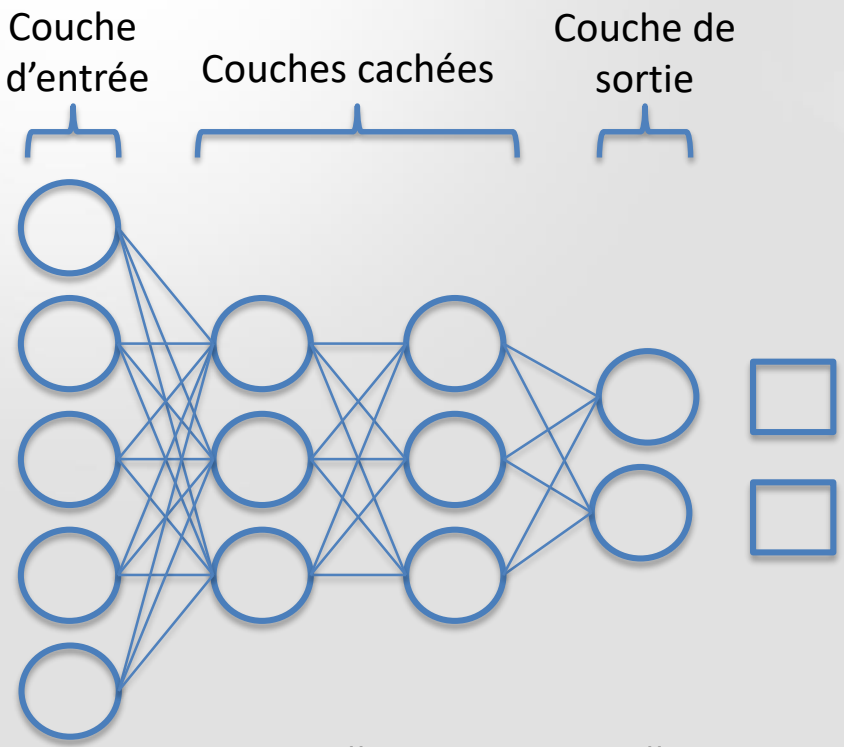
Base de données MNIST



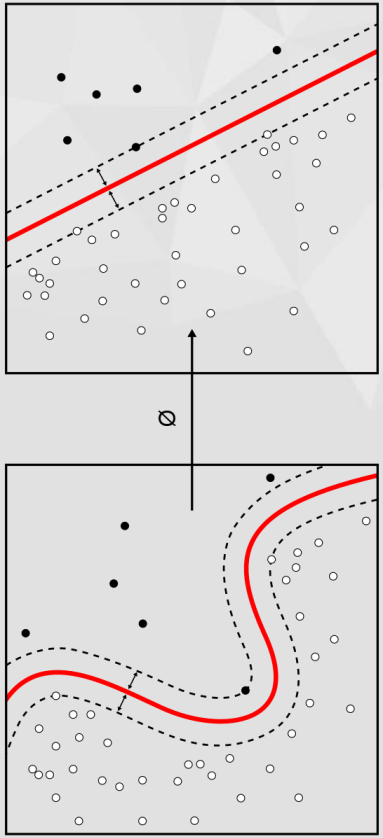
- Images 28x28
- Labélisées
- Base d'entrainement : 60 000 exemples
- Base de test : 10 000 exemples
  - 5000 normaux
  - 5000 bruités

Yann LeCun & Corinna Cortes & J.C. Burges

Fonctionnement perceptron multicouches



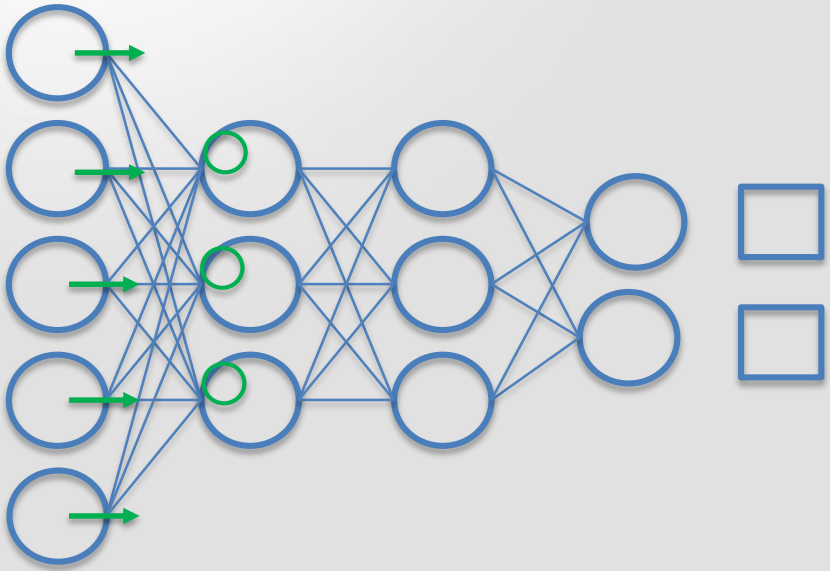
Rumelhart, Hinton, Williams 1986



Fonctionnement perceptron multicouches

Propagation de l'information

$$Pot_j = \sum_i w_{ij} \times x_i$$



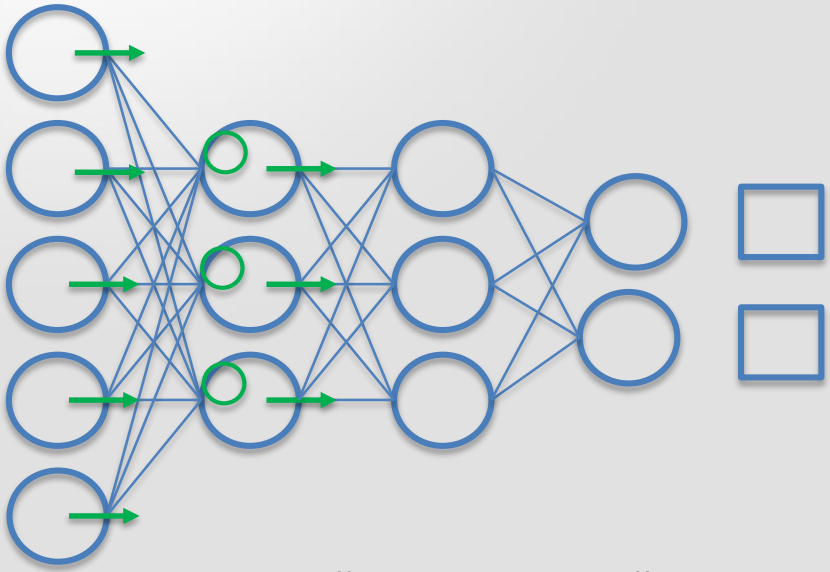
Rumelhart, Hinton, Williams 1986

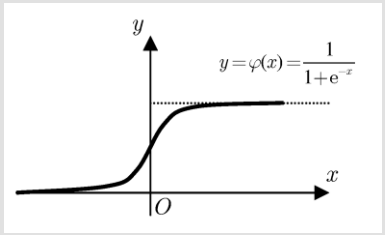
Fonctionnement perceptron multicouches

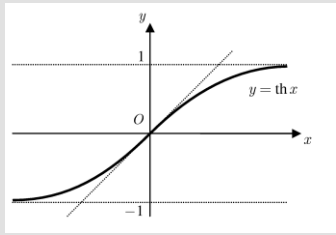
Propagation de l'information

$$Pot_j = \sum_i w_{ij} \times x_i$$

$$Signal_{out} = sigmoid(Pot)$$







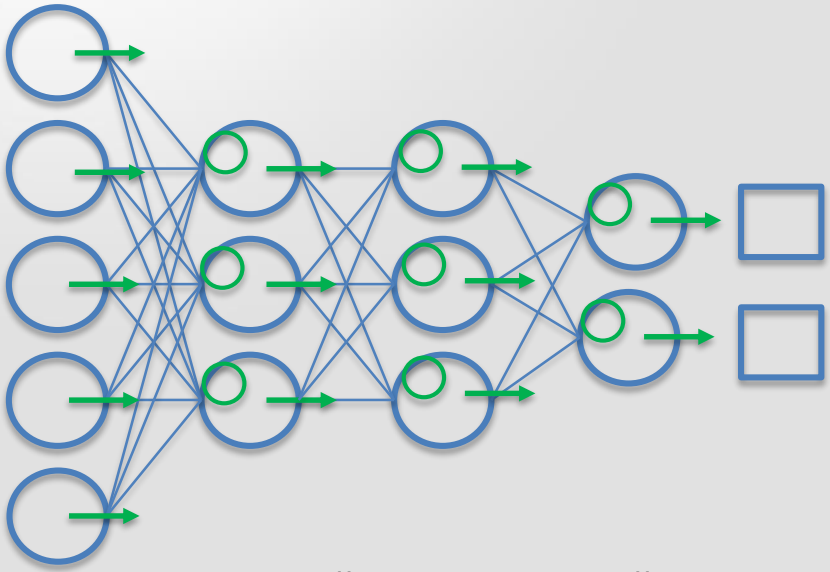
Rumelhart, Hinton, Williams 1986

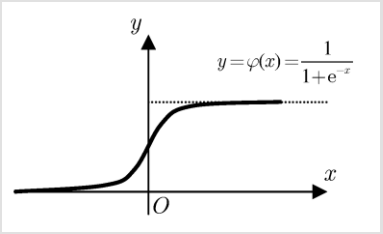
Fonctionnement perceptron multicouches

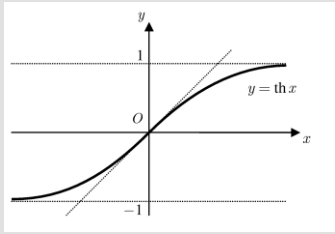
Propagation de l'information

$$Pot_j = \sum_i w_{ij} \times x_i$$

$$Signal_{out} = sigmoid(Pot)$$

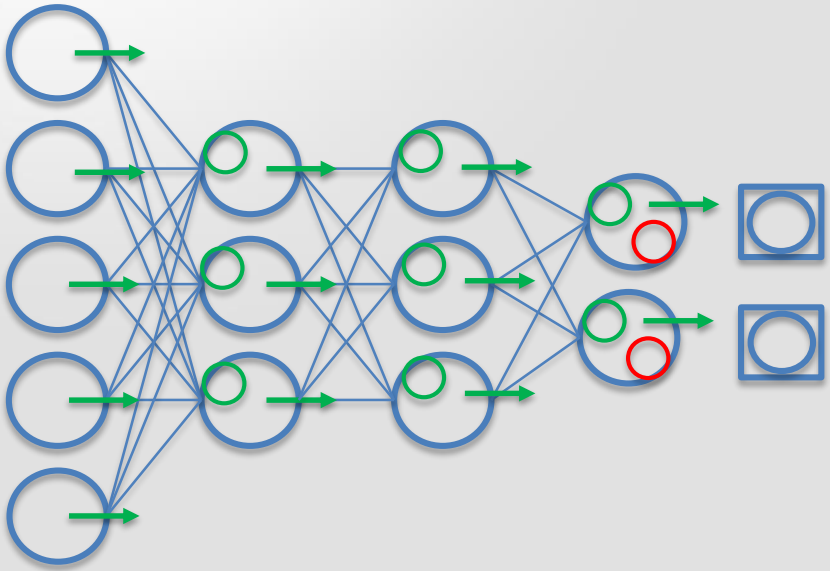






Rumelhart, Hinton, Williams 1986

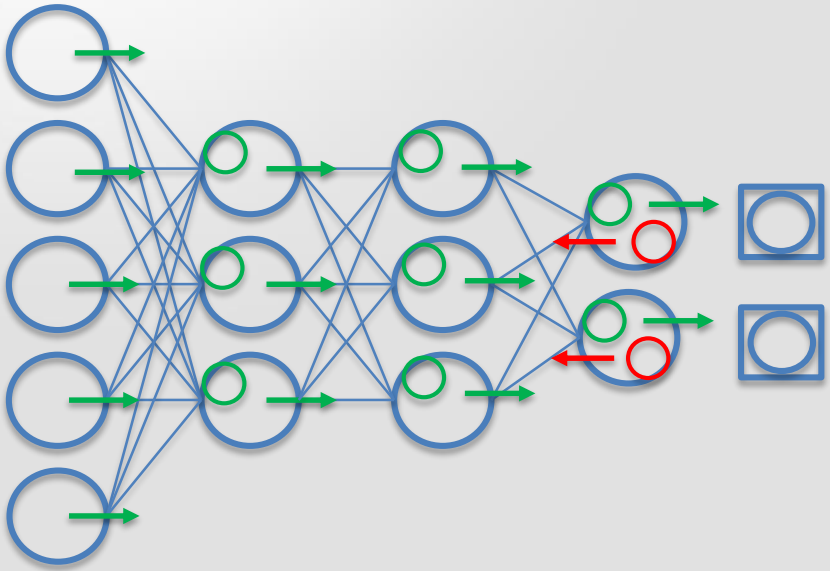
Calcul de l'erreur



$$E_i^{out} = superviseur_i^d - sig_i^{out}$$

Rumelhart, Hinton, Williams 1986

Calcul du signal d'erreur

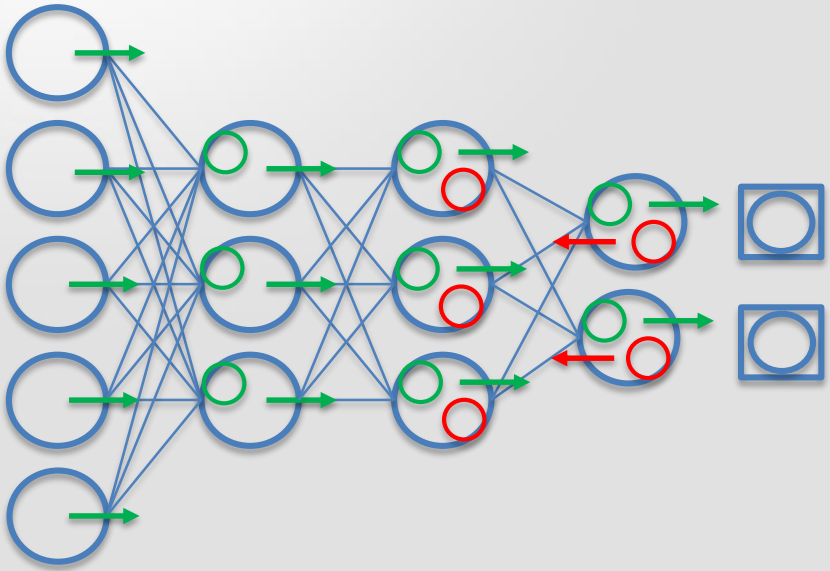


$$E_i^{out} = superviseur_i^d - sig_i^{out}$$

$$\delta^{out} = sig^{out} \circledast (1 - sig^{out}) \circledast E^{out}$$

Rumelhart, Hinton, Williams 1986

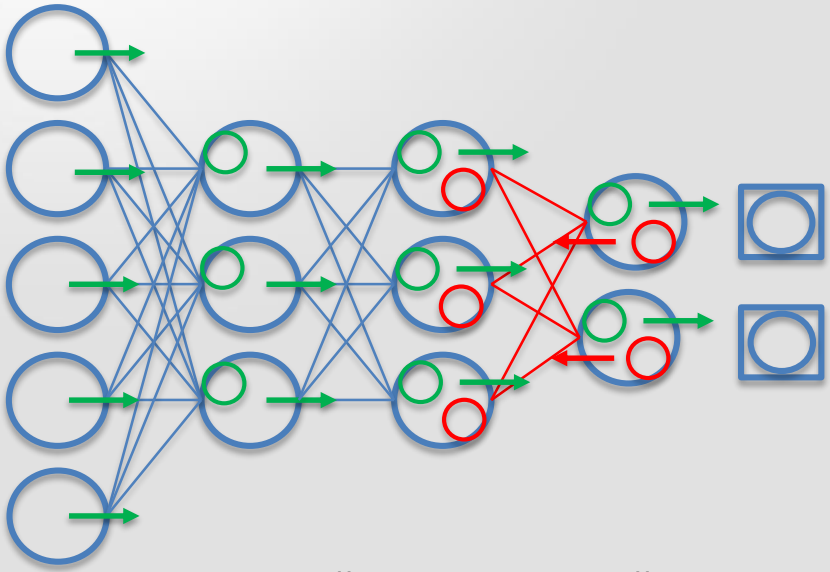
Propage signal d'erreur



$$E_i^n = \sum_j W_{ij} \times \delta_j^{n+1}$$

Rumelhart, Hinton, Williams 1986

Mise à jour des poids

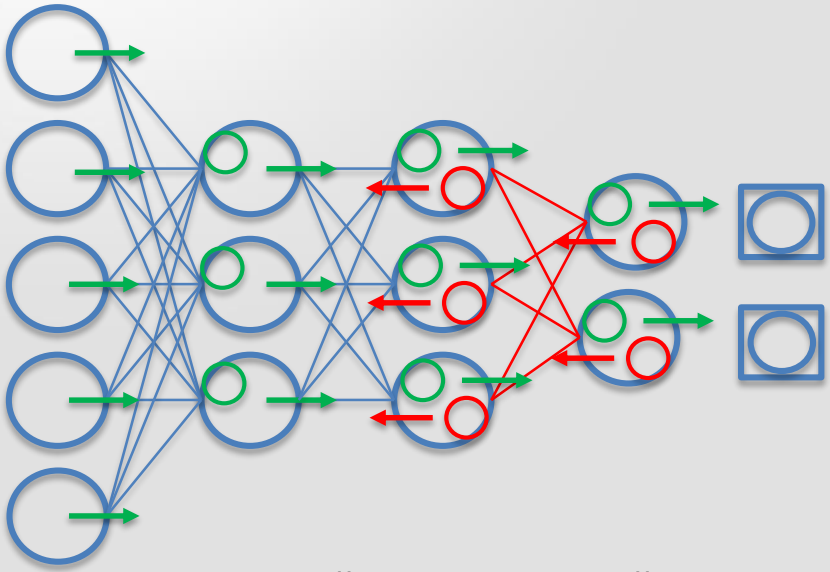


$$E_i^n = \sum_j W_{ij} \times \delta_j^{n+1}$$

$$W_{ij}^{t+1} = W_{ij}^t + \eta \times sig_i^n \times \delta_j^{n+1}$$

Rumelhart, Hinton, Williams 1986

Calcul du signal d'erreur



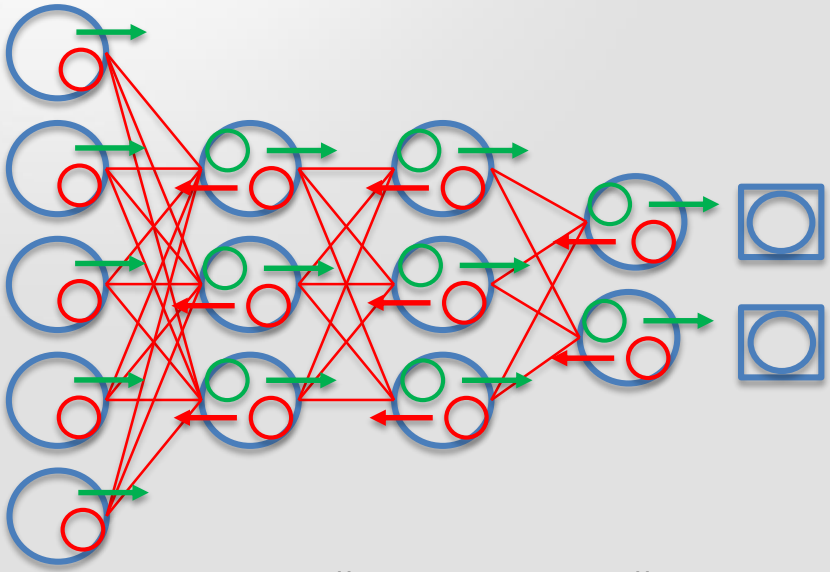
Rumelhart, Hinton, Williams 1986

$$E_i^n = \sum_j W_{ij} \times \delta_j^{n+1}$$

$$W_{ij}^{t+1} = W_{ij}^t + \eta \times sig_i^n \times \delta_j^{n+1}$$

$$\delta^n = sig^n \circledast (1 - sig^n) \circledast E^n$$

Propagation jusqu'au début



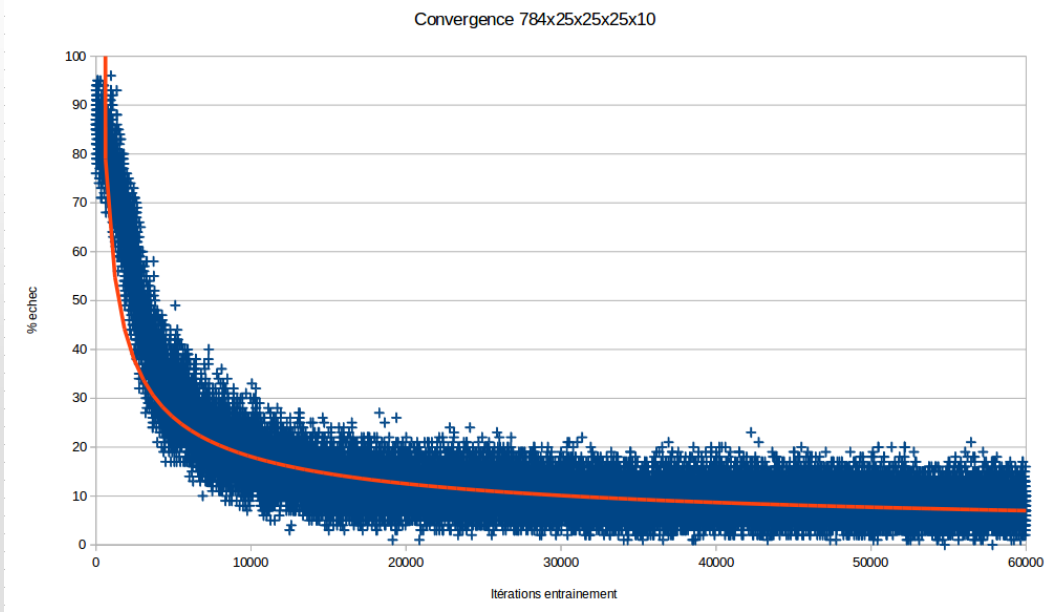
Rumelhart, Hinton, Williams 1986

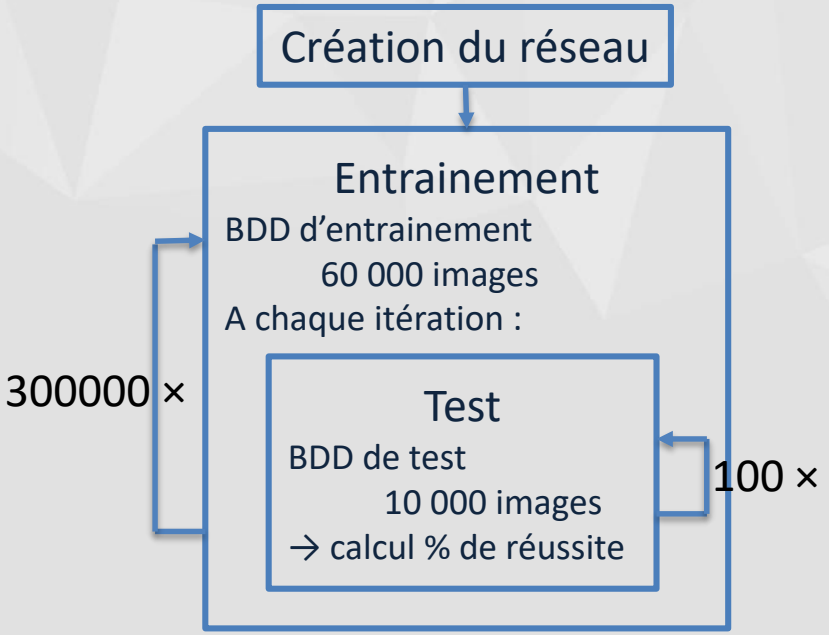
$$E_i^n = \sum_j W_{ij} \times \delta_j^{n+1}$$

$$W_{ij}^{t+1} = W_{ij}^t + \eta \times sig_i^n \times \delta_j^{n+1}$$

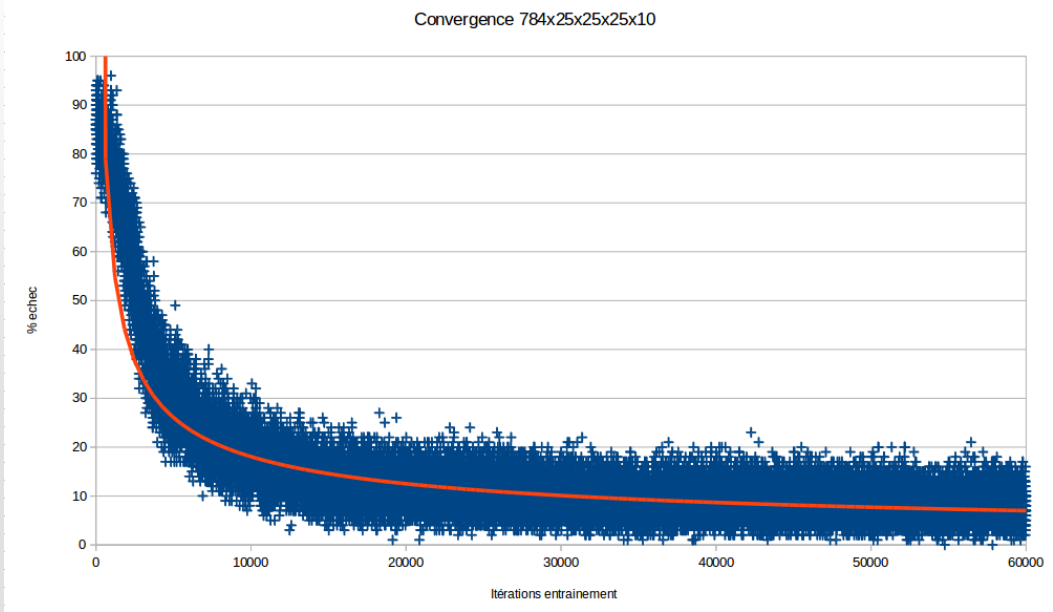
$$\delta^n = sig^n \circledast (1 - sig^n) \circledast E^n$$

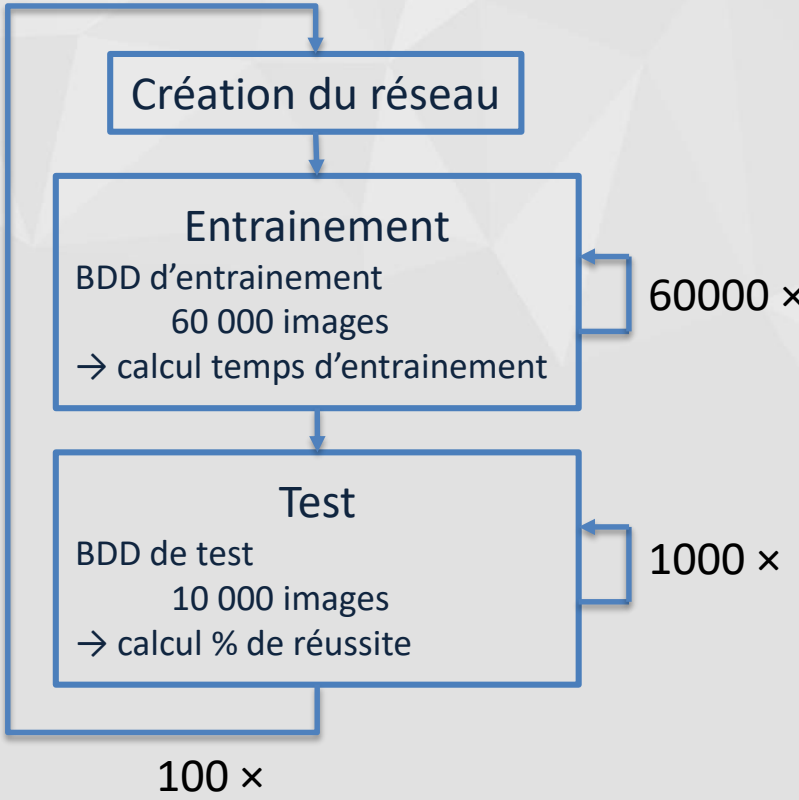
Résultats


Convergence 784x25x25x25x10
% echec
100
90
80
70
60
50
40
30
20
10
0
0
10000
20000
30000
40000
50000
60000
Itérations entrainement

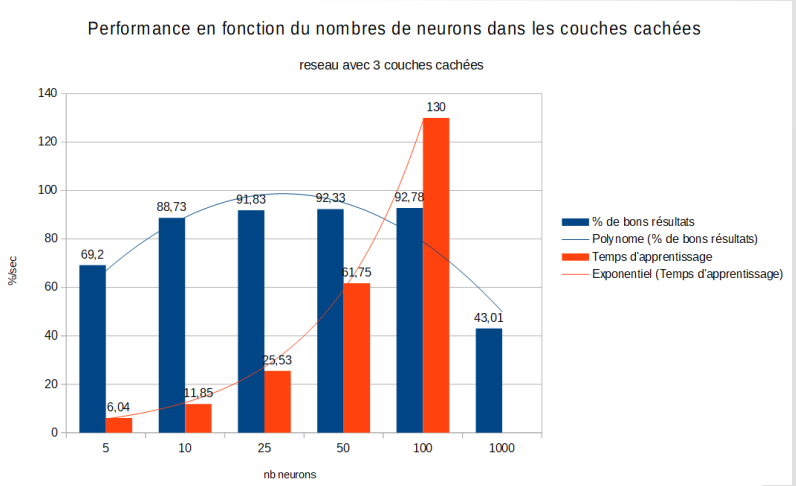

Création du réseau
Entrainement
BDD d'entrainement
60 000 images
A chaque itération :
300000 ×
Test
BDD de test
10 000 images
→ calcul % de réussite
100 ×

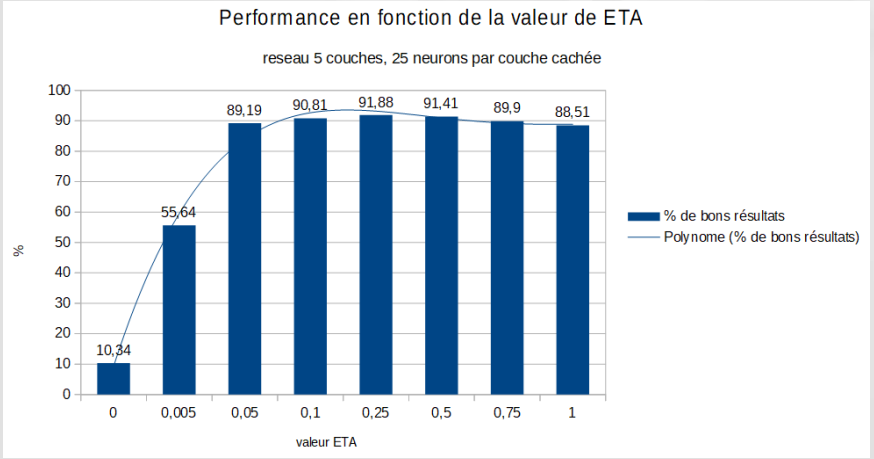
Résultats

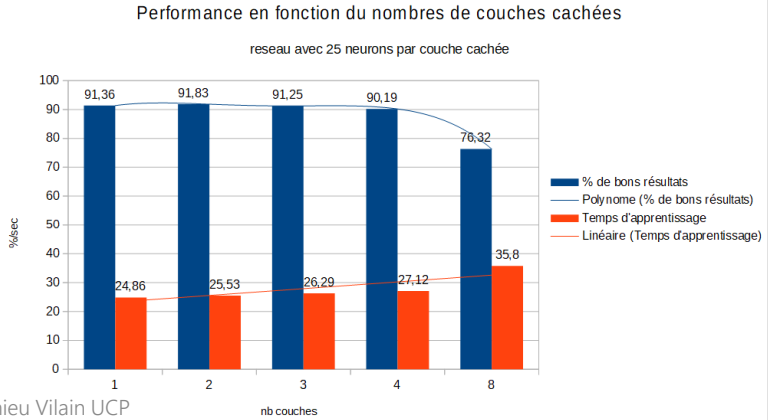

Convergence 784x25x25x25x10
% echec
Itérations entrainement
0
10
20
30
40
50
60
70
80
90
100
0
10000
20000
30000
40000
50000
60000


Création du réseau
Entrainement
BDD d'entrainement
60 000 images
→ calcul temps d'entrainement
60000 ×
Test
BDD de test
10 000 images
→ calcul % de réussite
1000 ×
100 ×

Perceptron multicouches
Optimisation

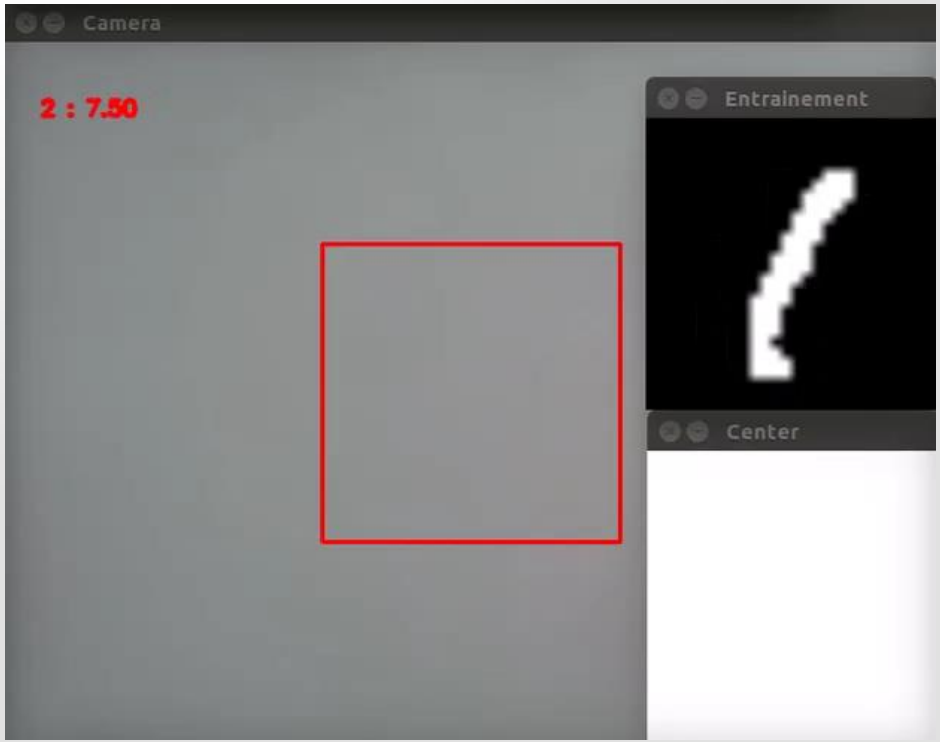






150 000 – 200 000 training
95% de réussite
100 sec de training

Application

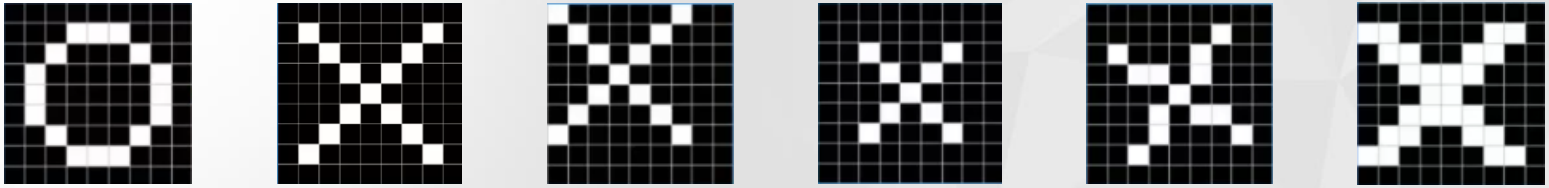


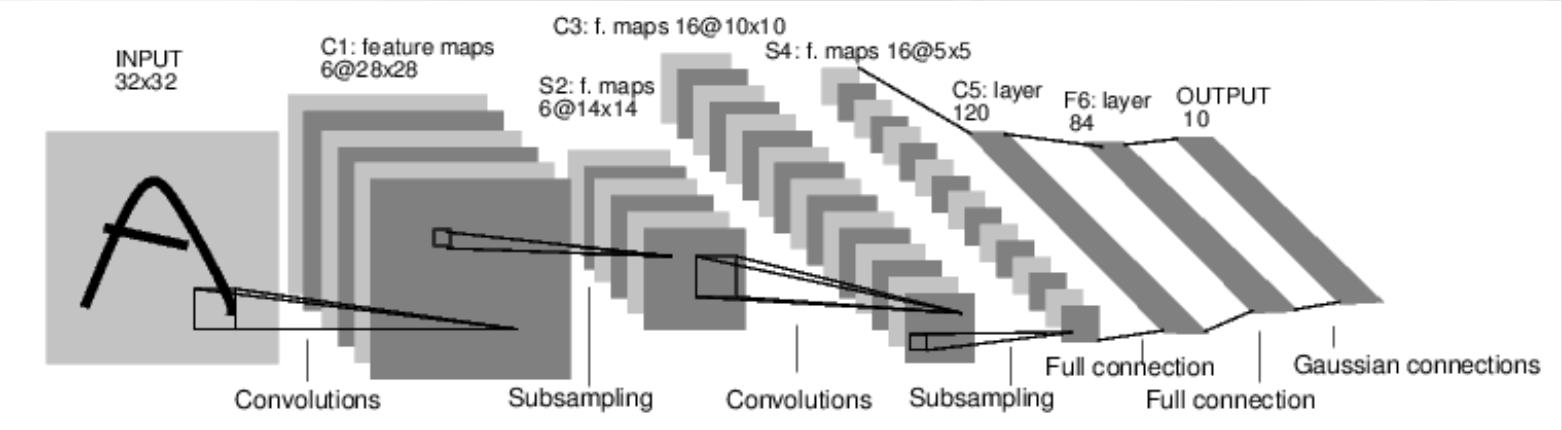
Problème

Non résistant aux :

- Translation
- Zoom
- Rotation
- Position
- Bruit

Fonctionnement du CNN





Réseau de neurones convolutif
Yann LeCun 1998

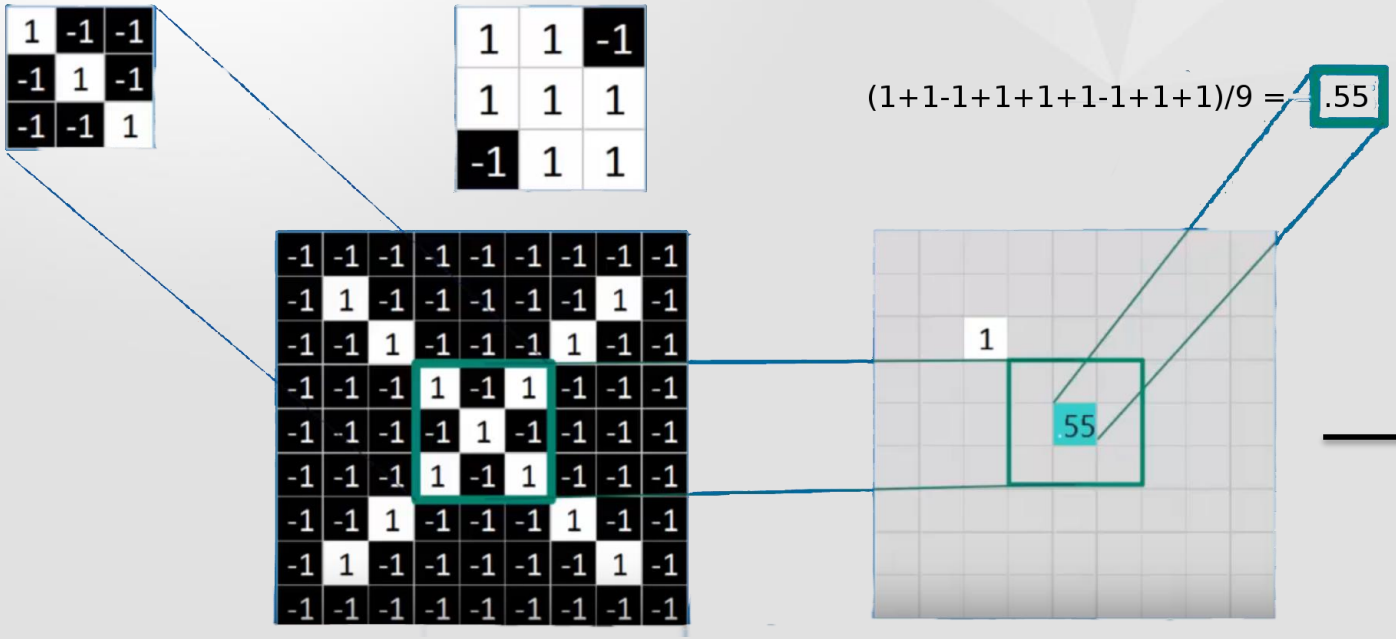
Couches de convolution

Filtres :

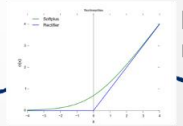
1	-1	-1
-1	1	-1
-1	-1	1

1	-1	1
-1	1	-1
1	-1	1

-1	-1	1
-1	1	-1
1	-1	-1



Couches de normalisation



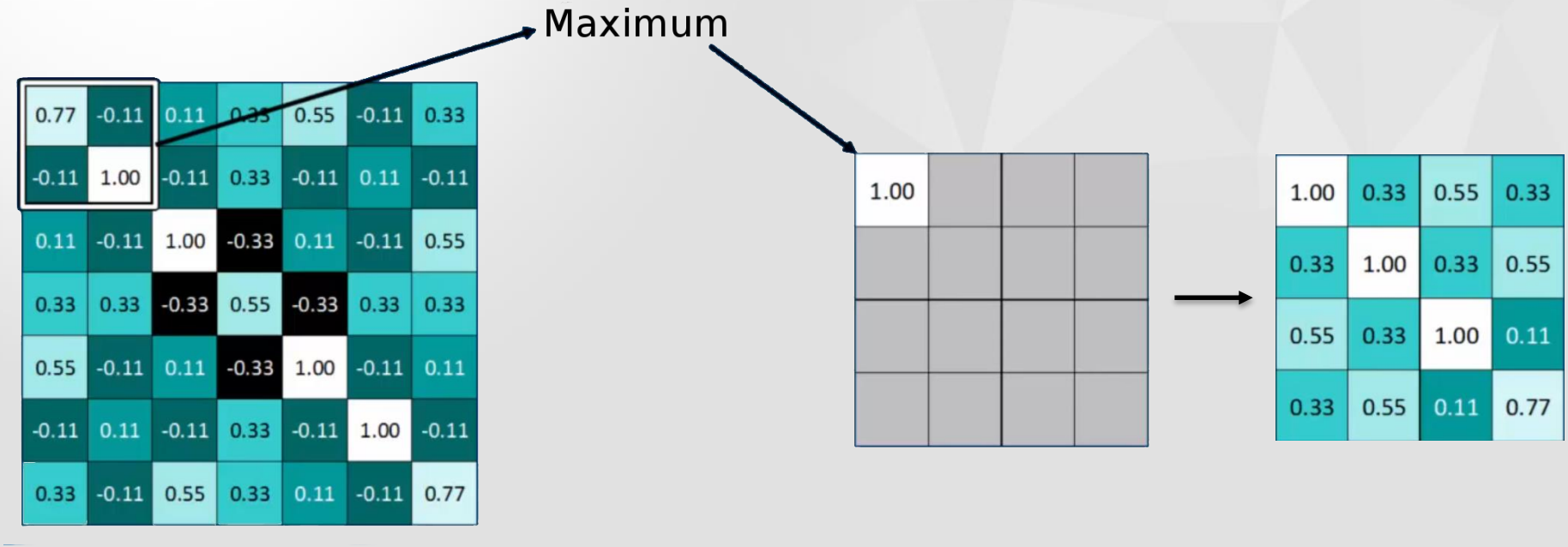
log(exp(x) +1
max(o, x)

0.77	-0.11	0.11	0.33	0.55	-0.11	0.33
-0.11	1.00	-0.11	0.33	-0.11	0.11	-0.11
0.11	-0.11	1.00	-0.33	0.11	-0.11	0.55
0.33	0.33	-0.33	0.55	-0.33	0.33	0.33
0.55	-0.11	0.11	-0.33	1.00	-0.11	0.11
-0.11	0.11	-0.11	0.33	-0.11	1.00	-0.11
0.33	-0.11	0.55	0.33	0.11	-0.11	0.77

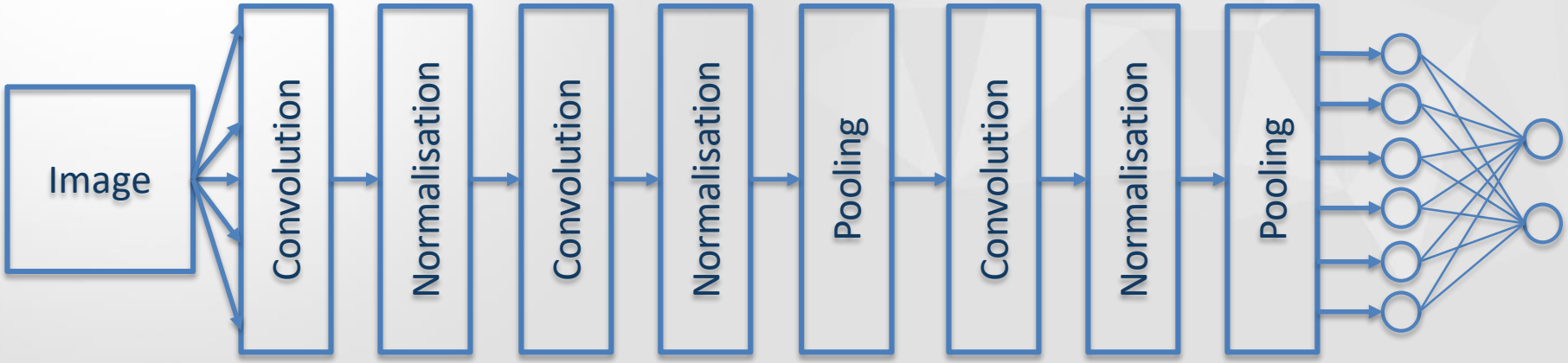
0.77						

0.77	0	0.11	0.33	0.55	0	0.33
0	1.00	0	0.33	0	0.11	0
0.11	0	1.00	0	0.11	0	0.55
0.33	0.33	0	0.55	0	0.33	0.33
0.55	0	0.11	0	1.00	0	0.11
0	0.11	0	0.33	0	1.00	0
0.33	0	0.55	0.33	0.11	0	0.77

Couches de pooling



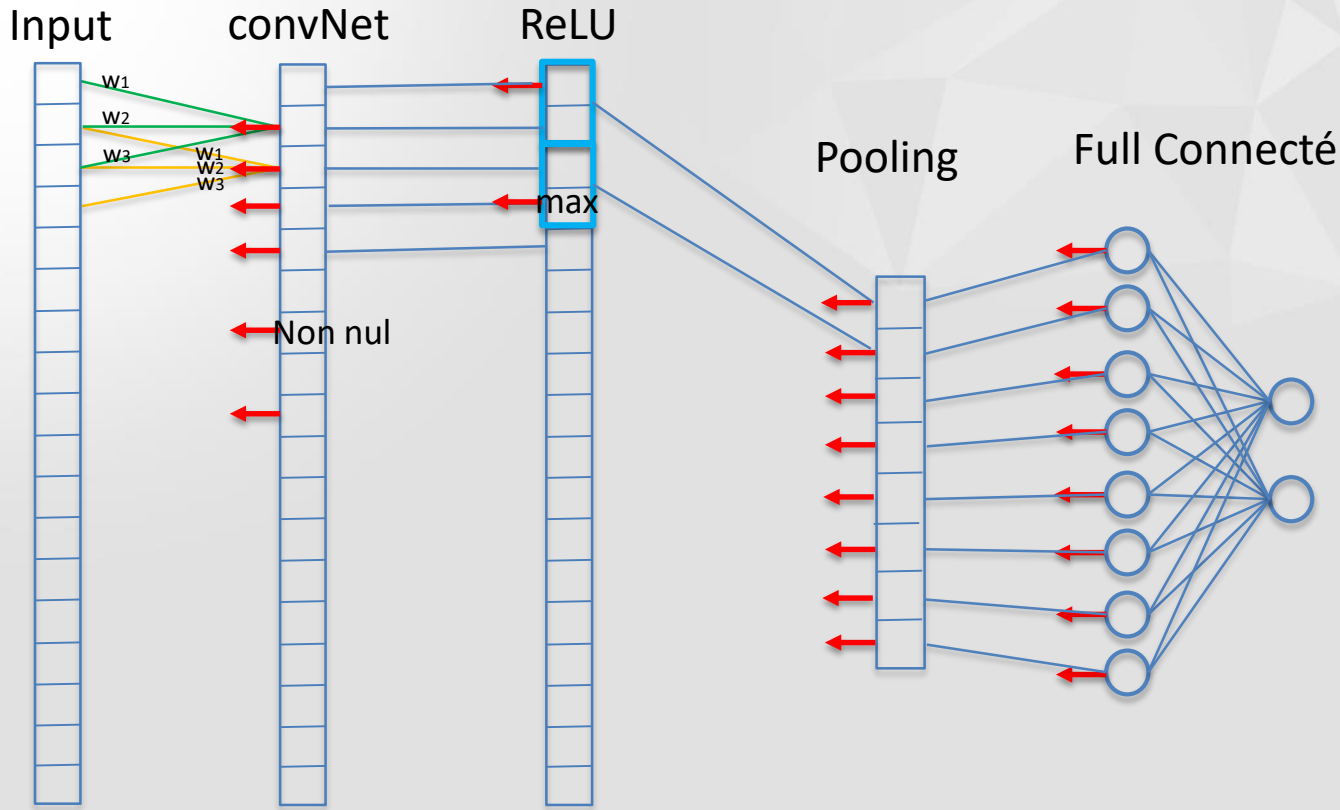
Organisation des couches



$$INPUT \rightarrow \big((CONV \rightarrow ReLU) \times N \rightarrow POOL\big) \times M \rightarrow (FC \rightarrow RELU) \times K \rightarrow FC$$

Avec $0 \leq N \leq 3$, $M \geq 0$, $0 \leq K < 3$

Backpropagation



Résultats

LeNet-5

99,28% de réussite

26 minutes de training

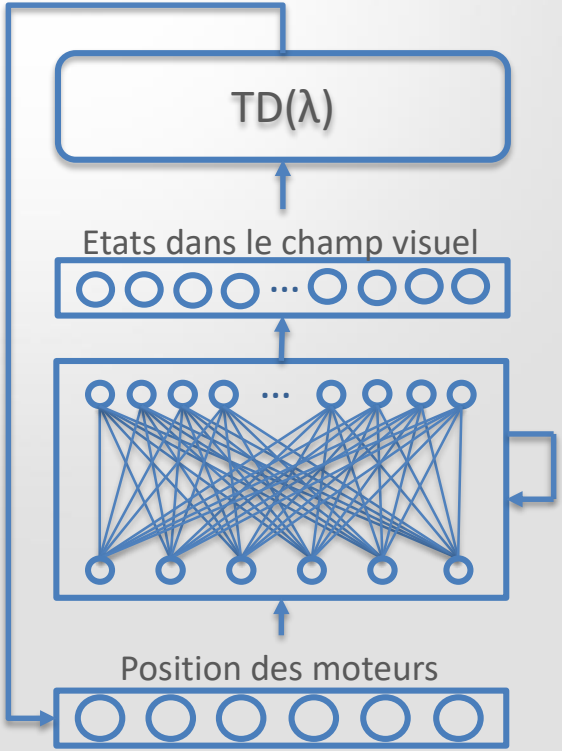
20 000 training

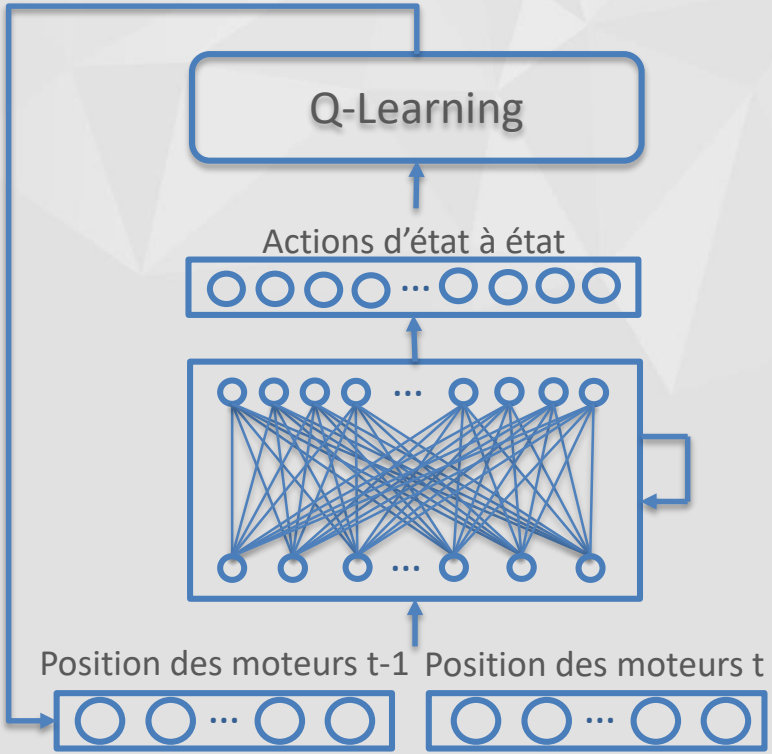
CPU - 8 cœurs

Le katana©

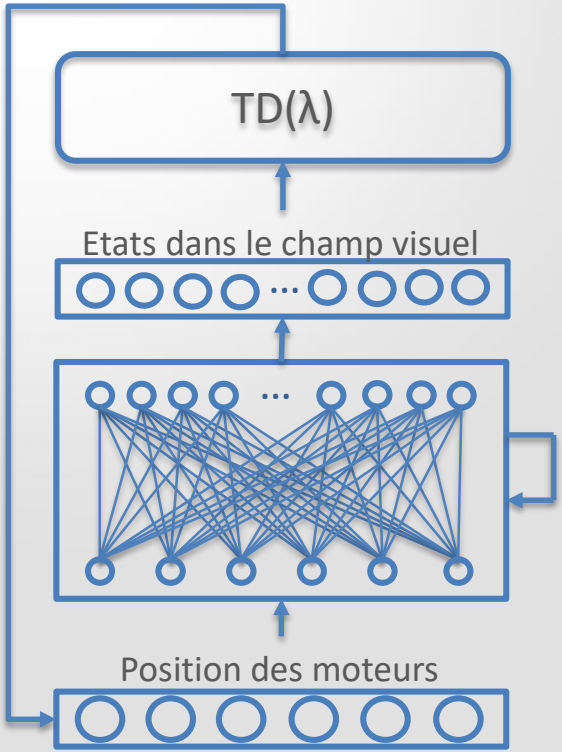
Apprendre à un bras robotique à attraper des objets

Le model

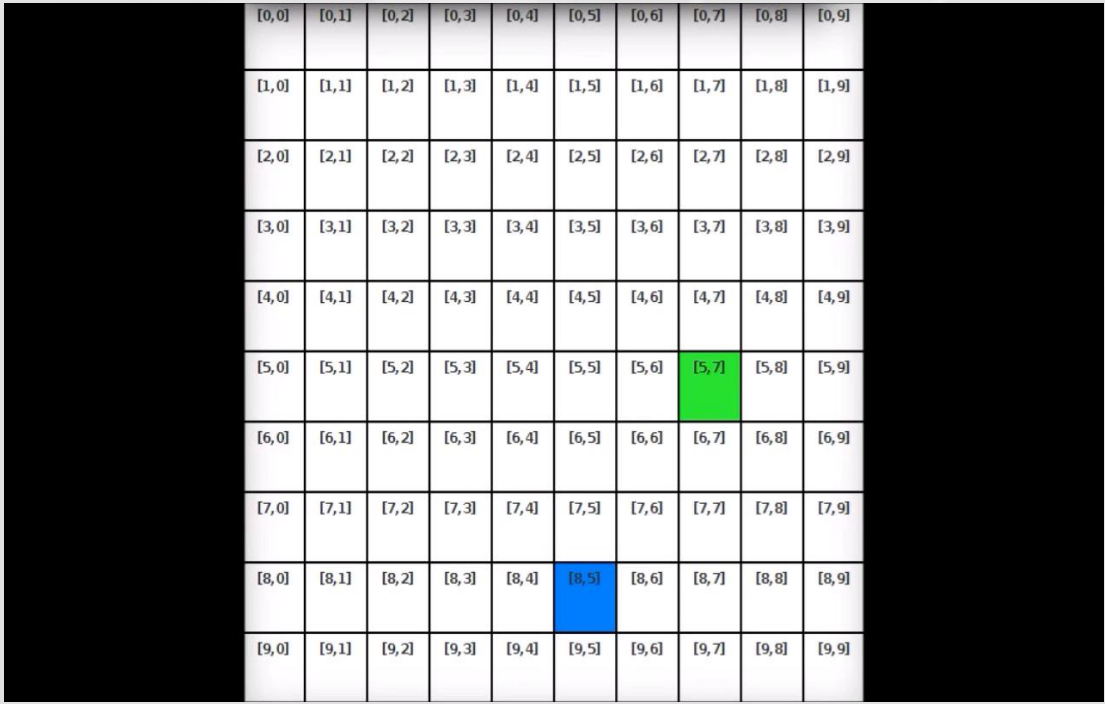

TD(λ)
Etats dans le champ visuel
Position des moteurs


Q-Learning
Actions d'état à état
Position des moteurs t-1
Position des moteurs t

Le model

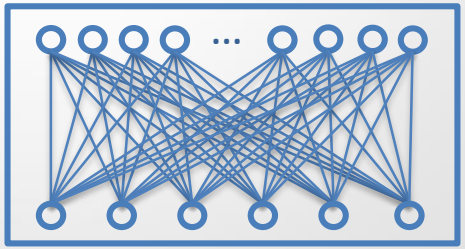


$$V(s_t) = V(s_t) + \alpha \times [R_{t+1} + \gamma\, V(s_{t+1}) - V(s_t)]$$

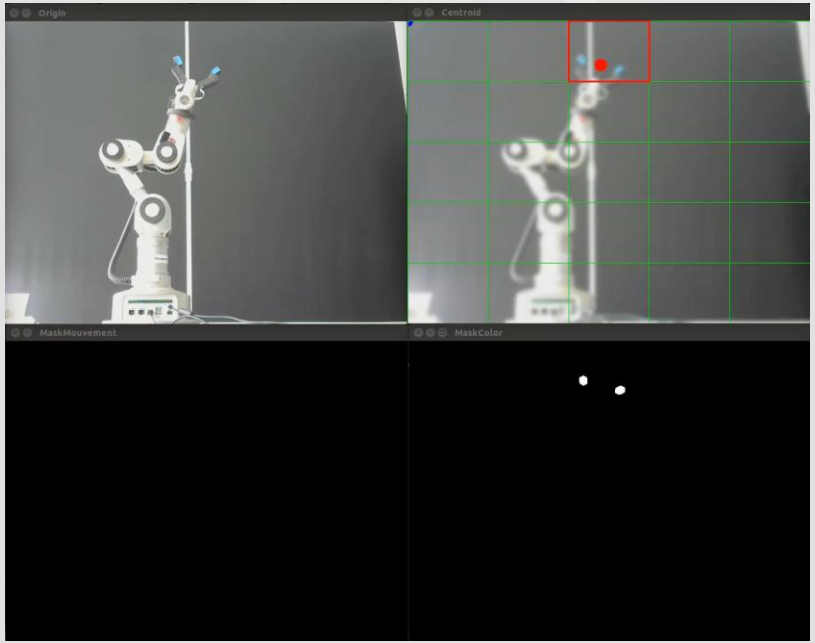


Implémentation

Réseau de neurones ✓



Traitement d'images ✓



Renforcement TD ✗

TD(λ)

Bilan du stage

- ✓ Appris énormément sur les réseaux de neurones
- ✓ Conforté dans mon choix d'orientation
- ✓ Plein de nouvelles idées de projets

Regret :

Pas avoir eu le temps de plus appliquer la théorie

Continuité du stage :

Projet de voiture autonome L3

Finir l'apprentissage sur le Katana