

Résolution du problème du labyrinthe par apprentissage par renforcement avec un réseau neuronal

Nicolas Yax

Sous la direction de *Philippe Preux*

Equipe SequeL – INRIA Lille

ENS Paris-Saclay

Sommaire

- Introduction
 - Équipe Sequel
 - Contexte du Stage
- Contexte scientifique
 - Problème du labyrinthe
 - Architecture du machine Learning
 - Réseau de neurones artificiels
- Contribution
 - Mécanique des fluides
 - Formes de réseau de neurones
 - Résultats et perspectives

Environnement



Equipe SequeL – INRIA Lille

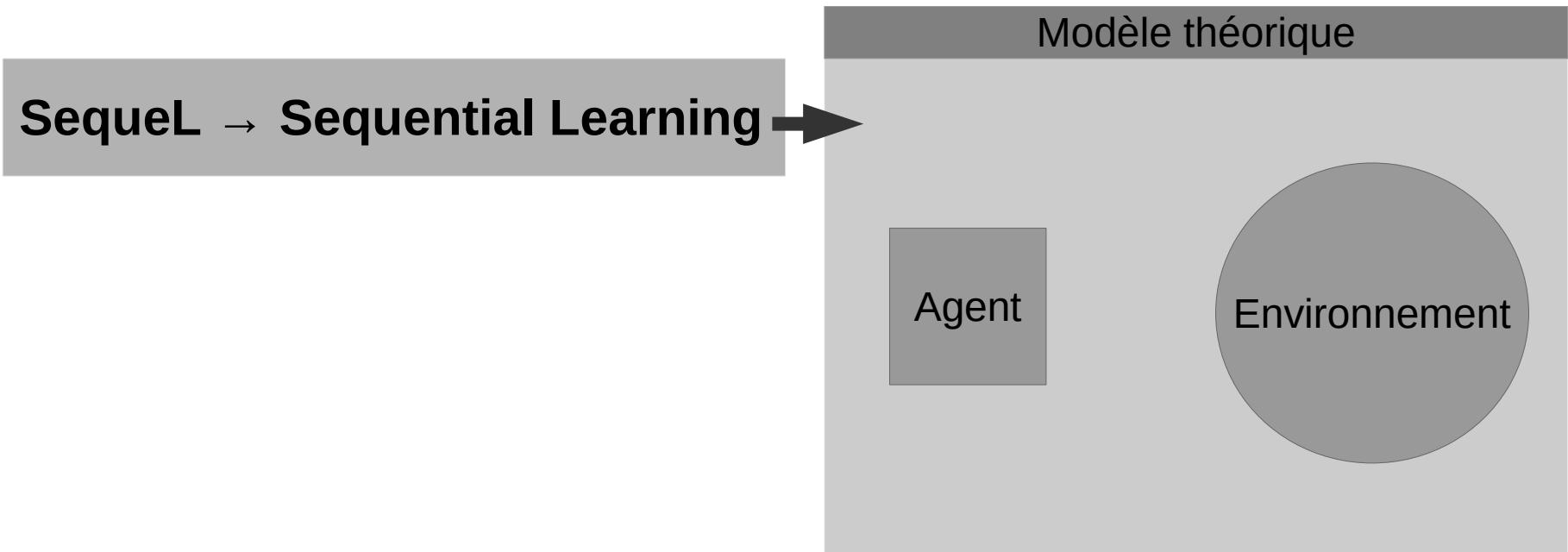
SequeL → Sequential Learning

Equipe SequeL – INRIA Lille

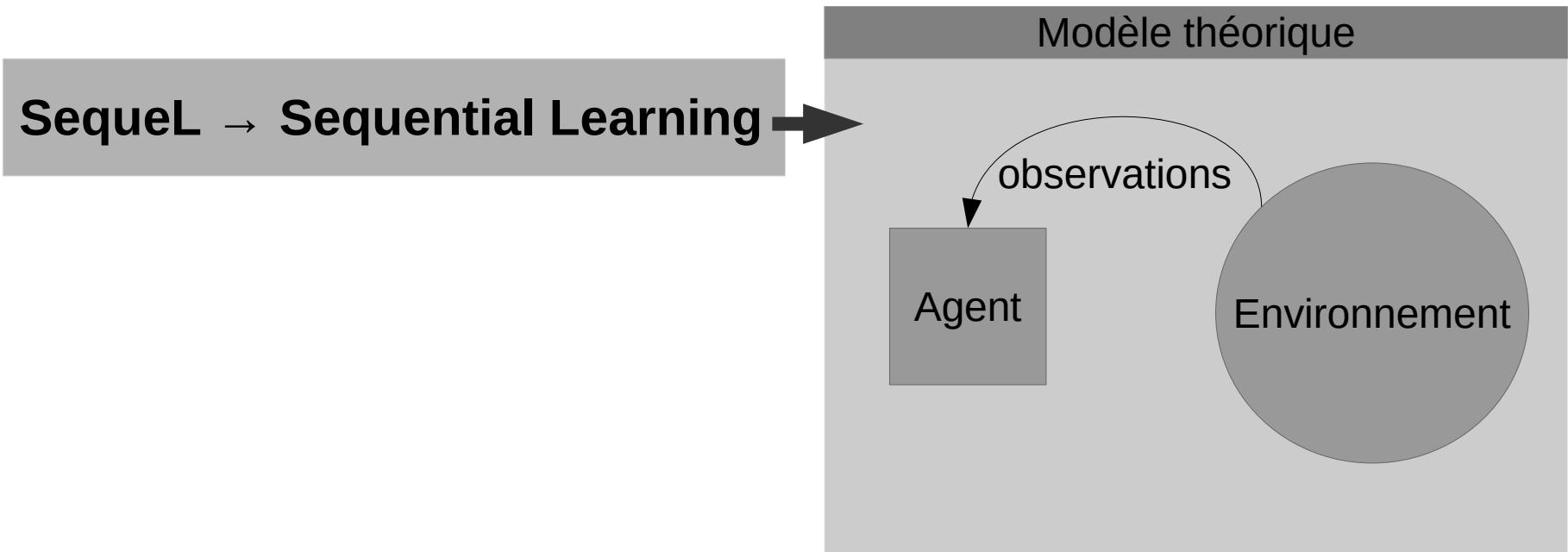
Modèle théorique

SequeL → Sequential Learning →

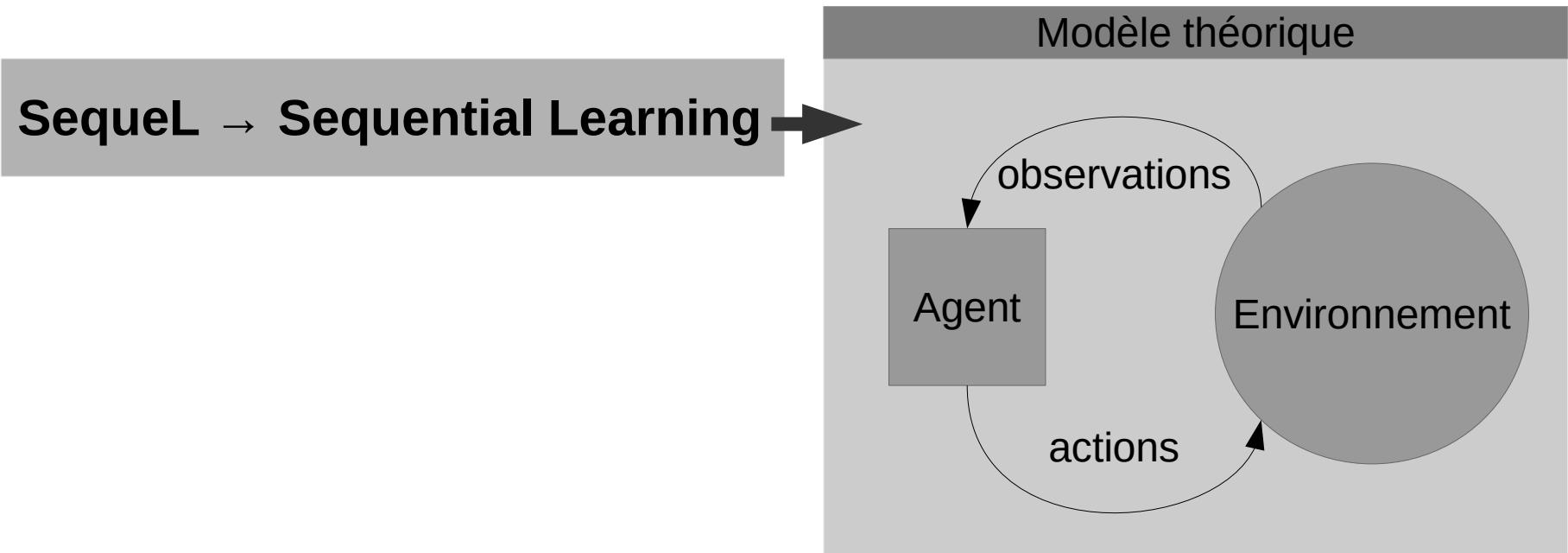
Equipe SequeL – INRIA Lille



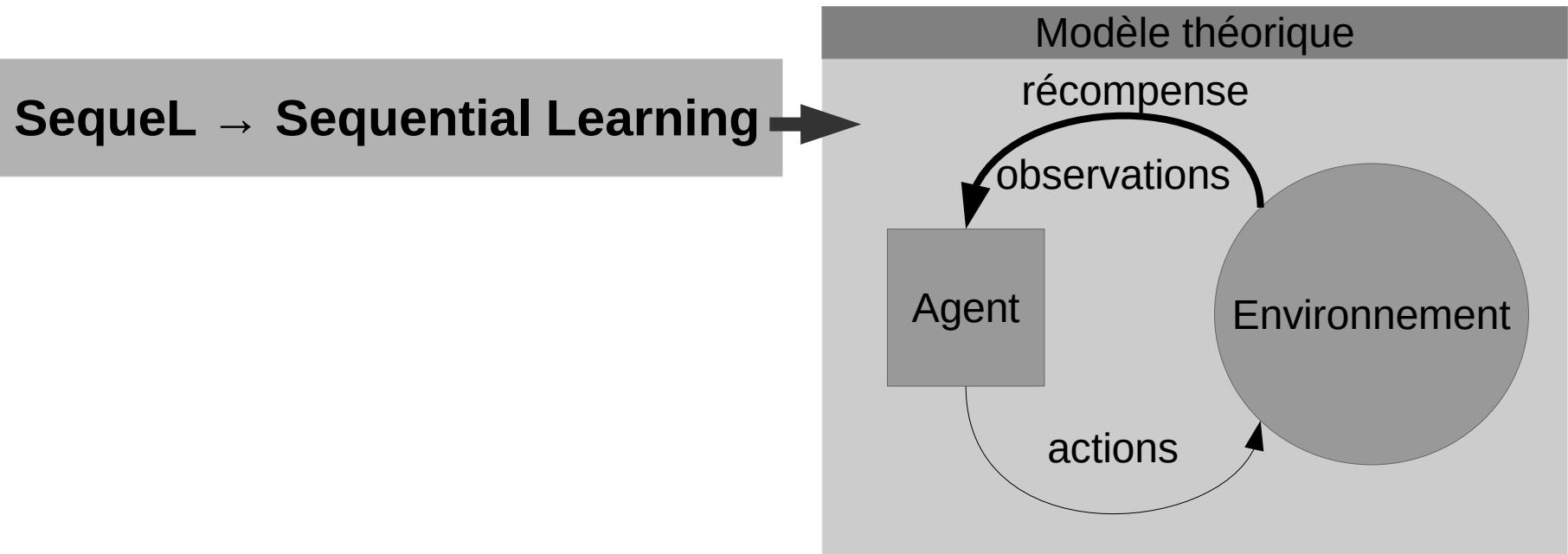
Equipe SequeL – INRIA Lille



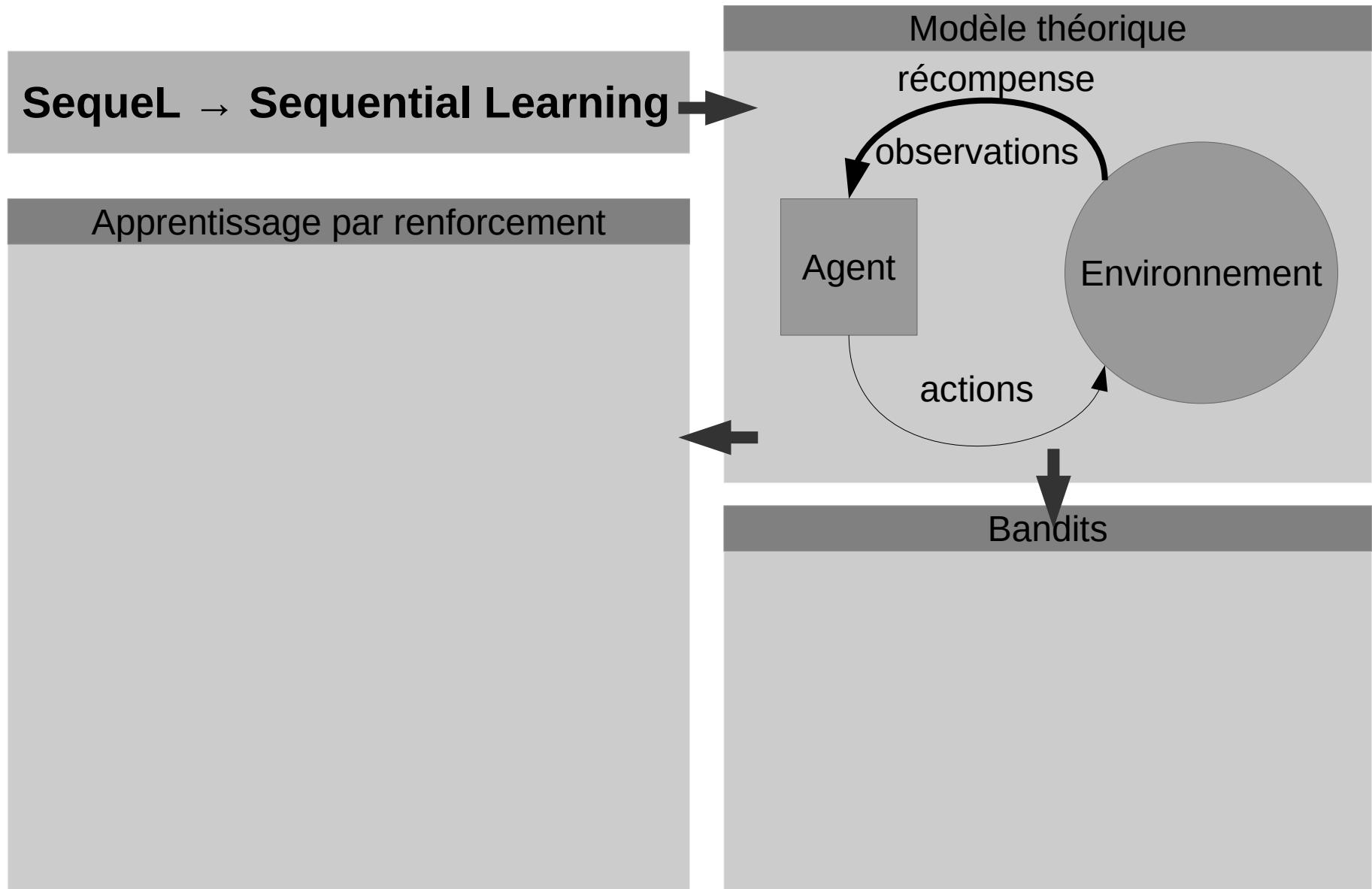
Equipe SequeL – INRIA Lille



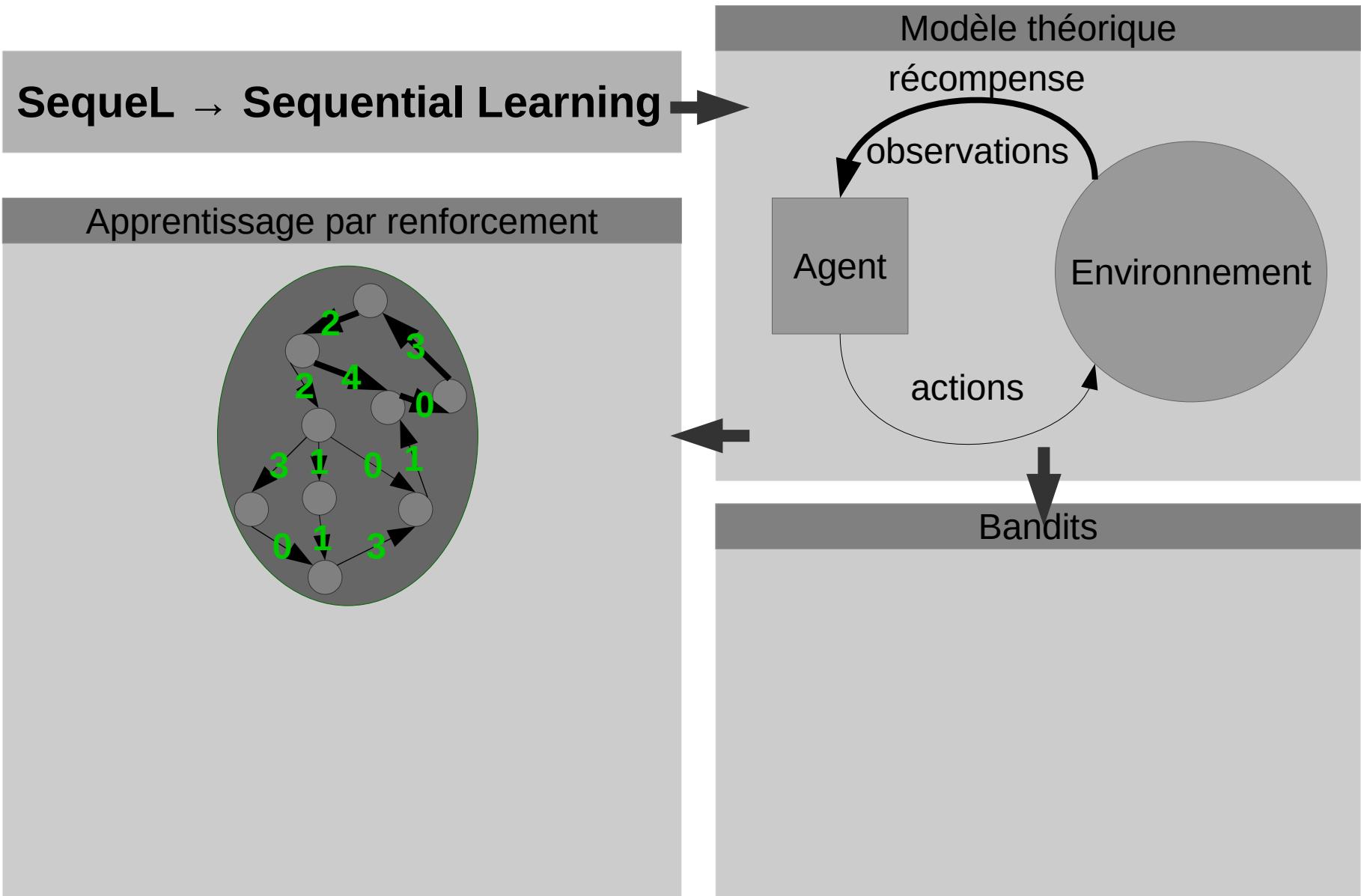
Equipe SequeL – INRIA Lille



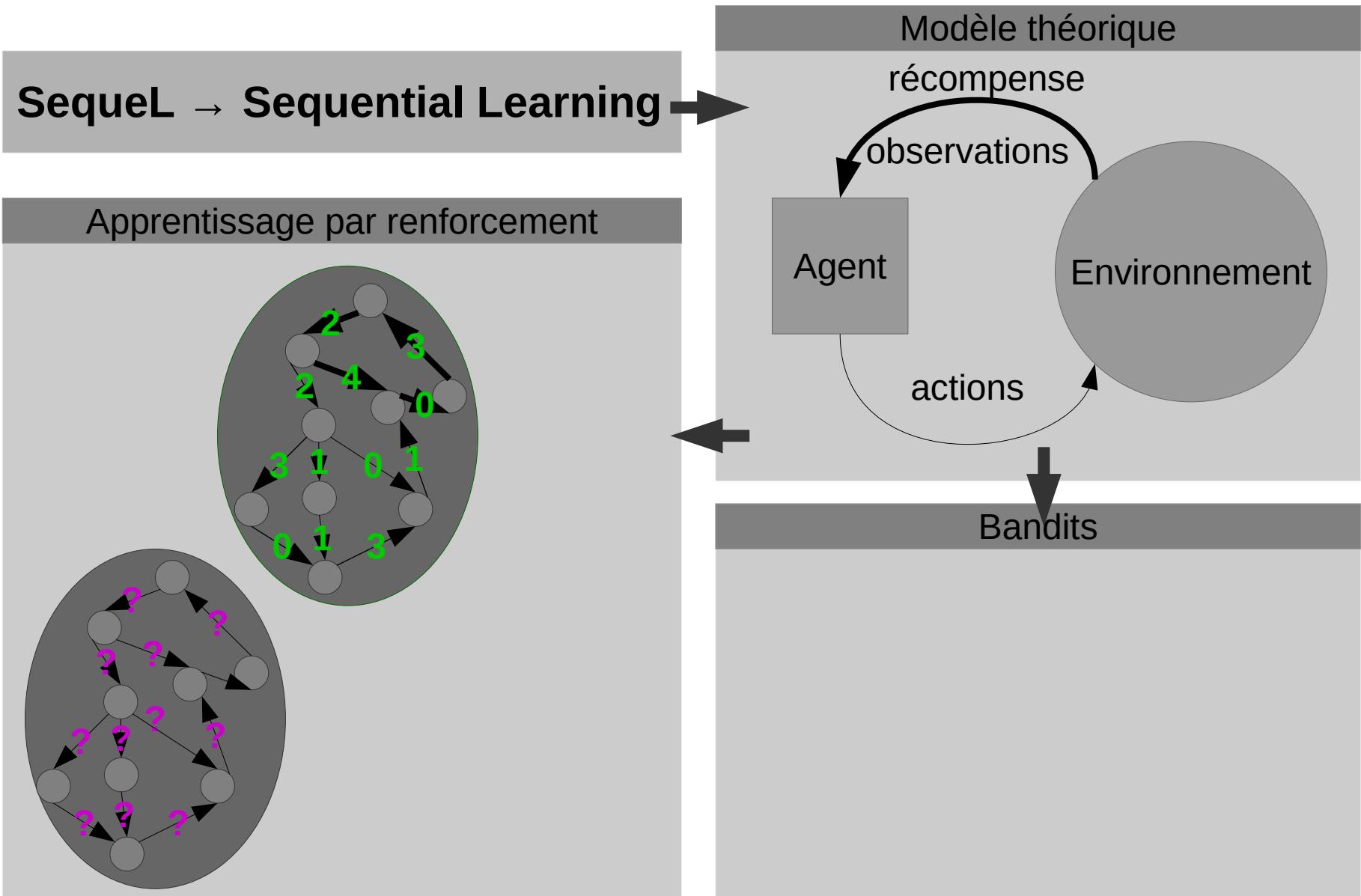
Equipe SequeL – INRIA Lille



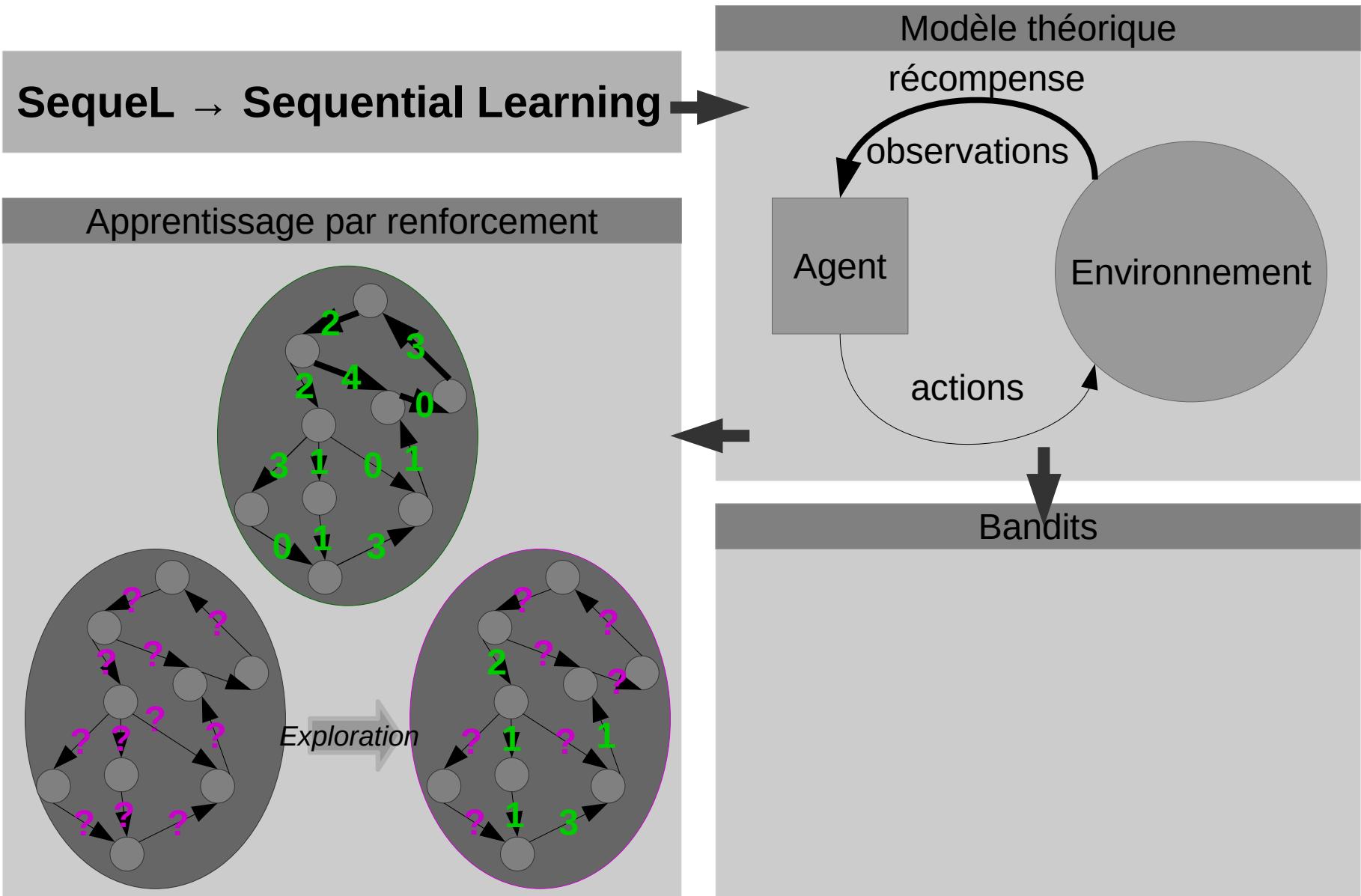
Equipe SequeL – INRIA Lille



Equipe SequeL – INRIA Lille

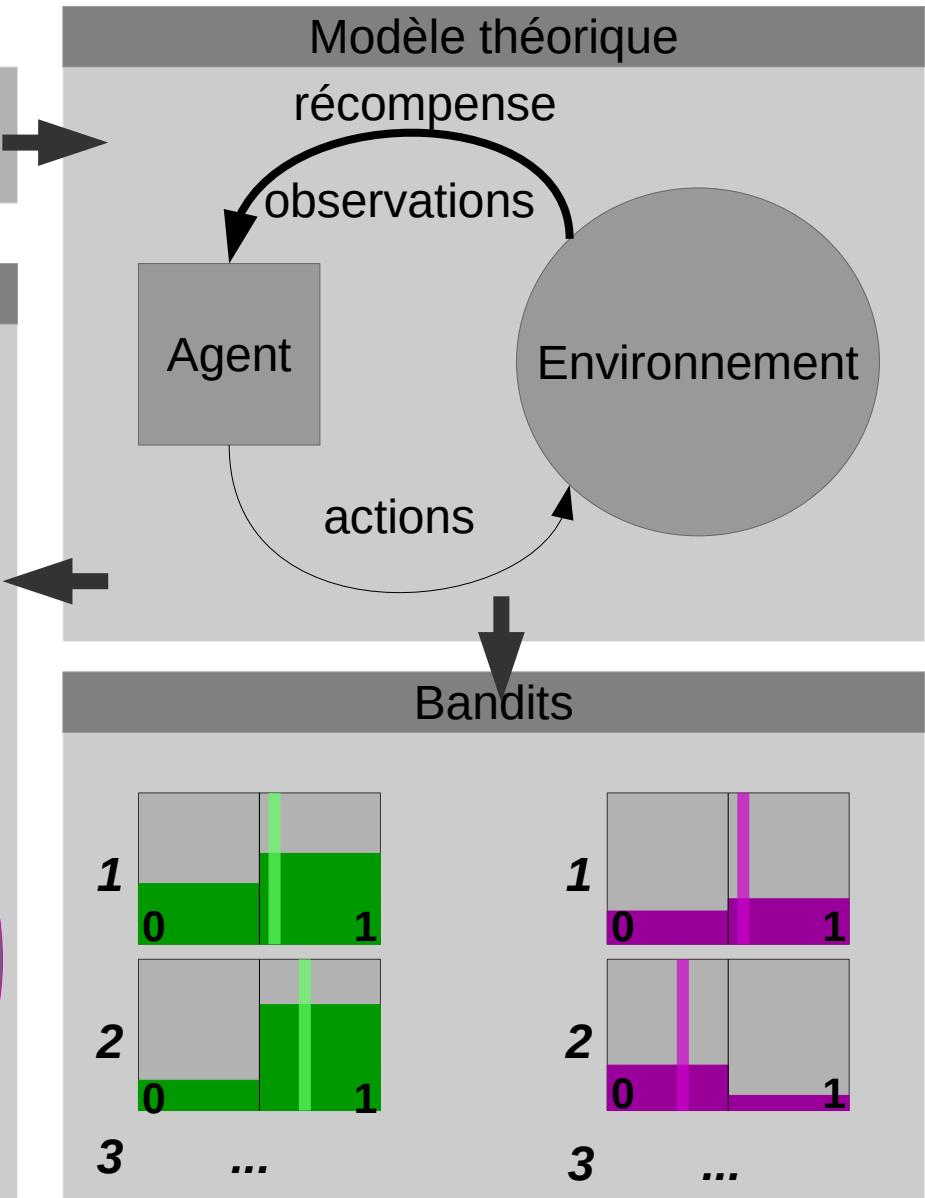
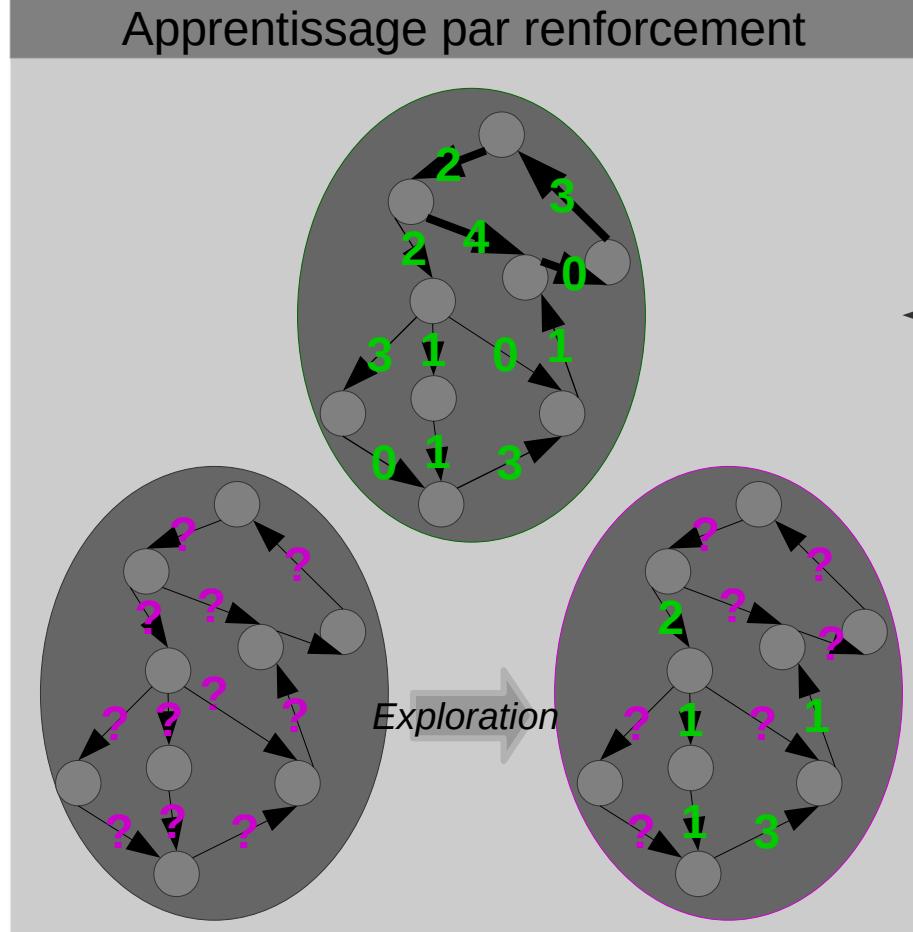


Equipe SequeL – INRIA Lille

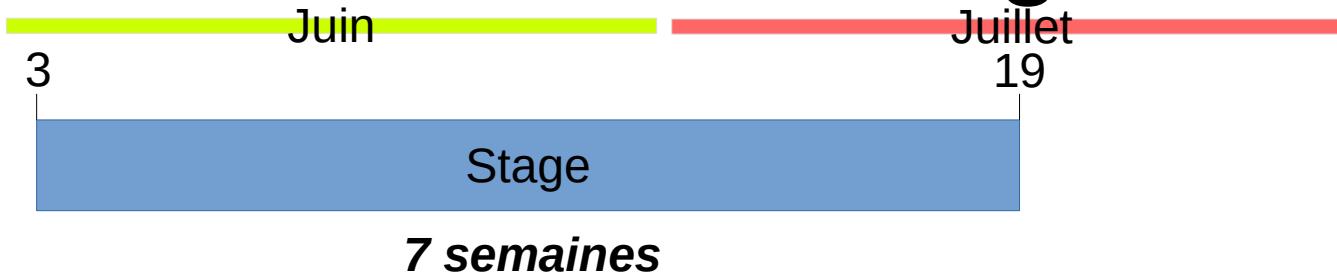


Equipe SequeL – INRIA Lille

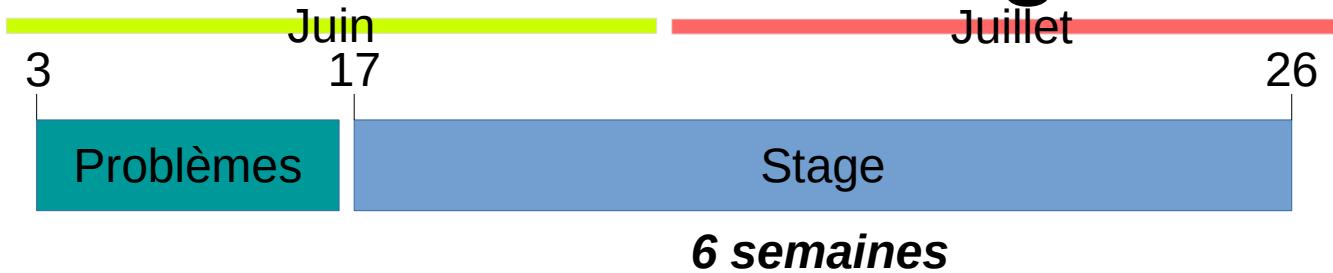
SequeL → Sequential Learning



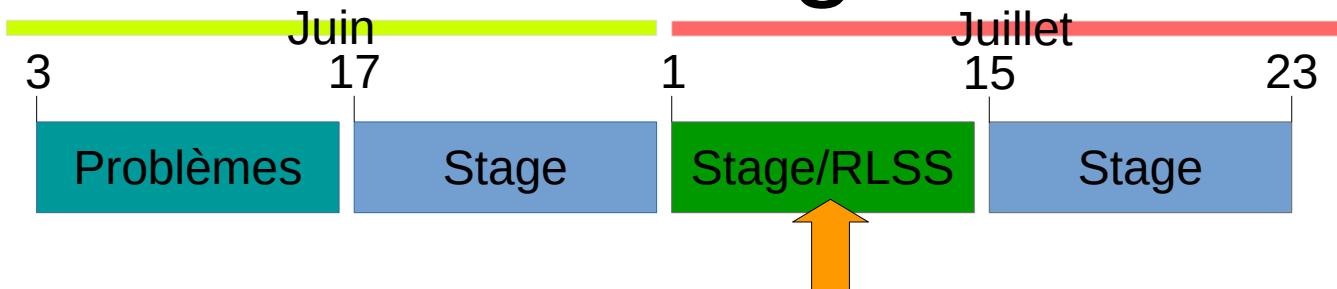
Contexte du Stage



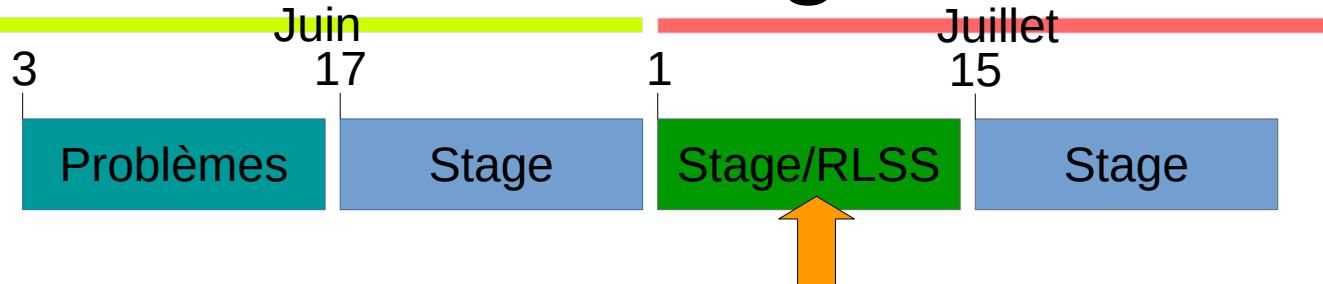
Contexte du Stage



Contexte du Stage - RLSS

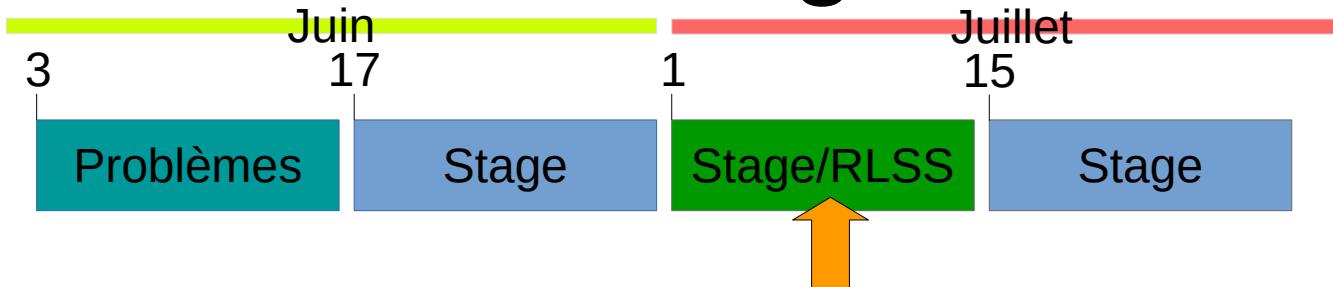


Contexte du Stage - RLSS

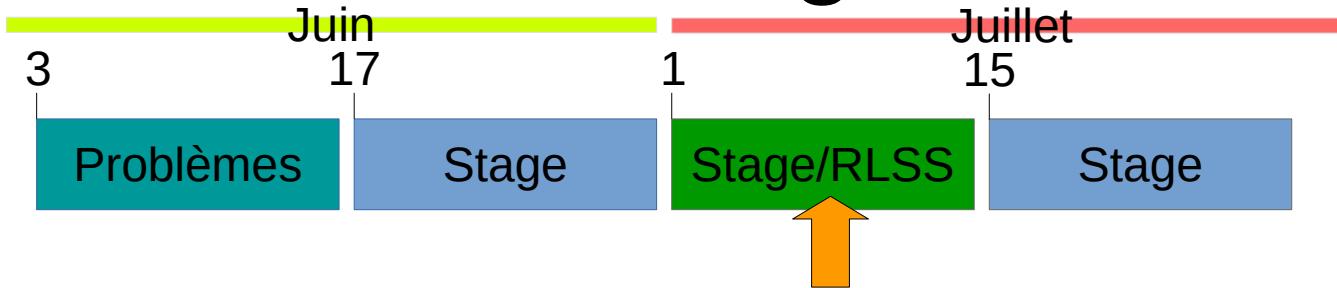


07-01 MONDAY					07-02 TUESDAY					07-03 WEDNESDAY					07-04 THURSDAY					07-05 FRIDAY					07-08 MONDAY					07-09 TUESDAY					07-10 WEDNESDAY					07-11 THURSDAY					07-12 FRIDAY				
09:00	Registration	Stochastic Bandits Emilie Kaufmann Slides	MDP Bruno Scherrer Slides	Introduction RL Olivier Pietquin Slides	Policy Gradient Matteo Pirotta Slides	09:00	Structured Bandits Odalric-Ambrym Maillard Slides	Exploration Alessandro Lazaric Slides on RL	Safe RL Felix Berkenkamp Slides	MCTS Tristan Cazenave Slides	Final Project Max Lapan																																						
10:30	Opening remarks	Coffee Break	Coffee Break	Coffee Break	Coffee Break	10:30	Coffee Break	Coffee Break	Coffee Break	Coffee Break	Coffee Break																																						
10:45	Basic Statistics Guillaume Obozinski Slides on Concentration inequalities	Structured Bandits Odalric-Ambrym Maillard Slides	MDP Bruno Scherrer Slides	Introduction RL Olivier Pietquin Slides	Policy Gradient Matteo Pirotta Slides	10:45	Bandits for Optimization Emilie Kaufmann Slides	Exploration Alessandro Lazaric Slides on DRL	Learning without a Reward Bilal Piot	Multi-agent Marc Lanctot Slides	Final Project Max Lapan																																						
12:15	Lunch Break	Lunch Break	Lunch Break	Lunch Break	Lunch Break	12:15	Lunch Break	Lunch Break	Lunch Break	Lunch Break	Lunch Break																																						
14:00	Basic Statistics Guillaume Obozinski Slides on Markov chains Slides on Supervised learning	Healthcare Audrey Durand Slides	Rec. Systems Jérémie Mary Slides	Energy Gabriel Dulac-Arnold Slides	RL Good practice Vincent François-Lavet Slides	14:00	Video Games Oriol Vinyals Slides	Language Florian Strub Slides	Practical Session DRL 2 Solutions	Neuroscience Eleni Vasilaki Slides	Final Project Max Lapan																																						
15:00	Coffee Break	Coffee Break	Coffee Break	Coffee Break	Coffee Break	15:00	Advanced Deep RL Matteo Hessel Slides	Coffee Break	Practical Session DRL 3 Solutions	Coffee Break	Coffee Break																																						
15:30	Coffee Break	Practical Session Bandits Solutions: Colab Notebook	Practical Session Rec. Systems (zip) Solutions	Neural Networks Ludovic Denoyer	Practical Session RL Solutions	15:30	Advanced Deep RL Matteo Hessel Slides	Practical Session DRL 1 Solutions	Robotics Gerhard Neumann Slides	Final Project Max Lapan	Final Project Max Lapan																																						
16:00	Stochastic Bandits Emilie Kaufmann Slides					16:30	Coffee Break		1. Intro	1. Intro																																							
18:00	Welcoming cocktail			Dinner at Couvent des Minimes		17:00	Advanced Deep RL Matteo Hessel Slides	Practical Session DRL 1 Solutions	2. TextWorld	2. TextWorld	Final Project Max Lapan																																						
18:30						17:30	Advanced Deep RL Matteo Hessel Slides	Practical Session DRL 1 Solutions	3. MiniWoB: - standalone - colab	3. MiniWoB: - standalone - colab																																							
						18:30		Dinner and Party at Gare St. Sauveur	Poster Session																																								

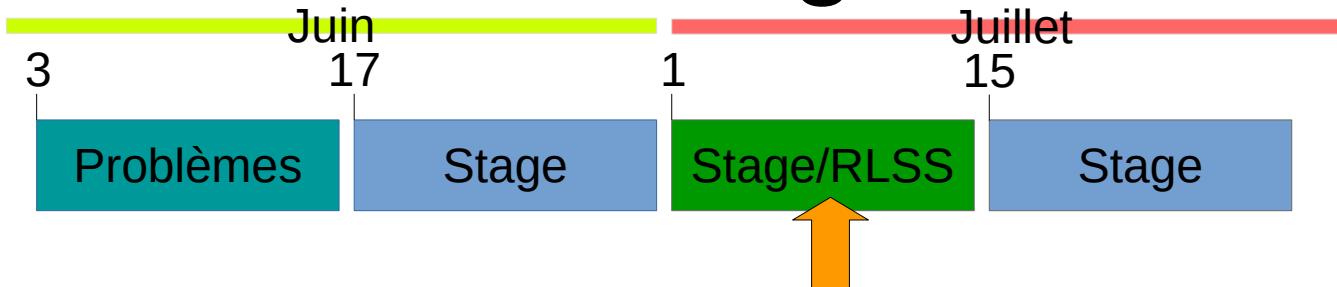
Contexte du Stage - RLSS



Contexte du Stage - RLSS



Contexte du Stage - RLSS



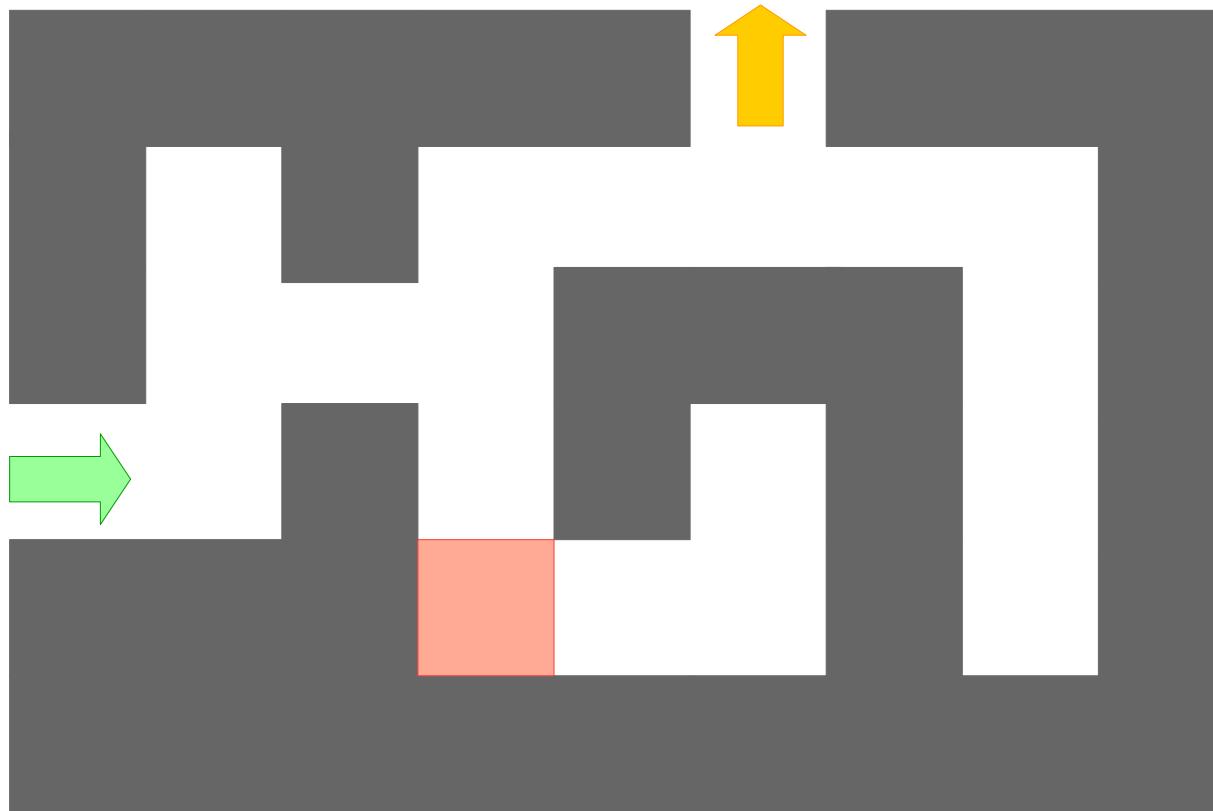
Sujet du Stage

Résolution du problème du labyrinthe

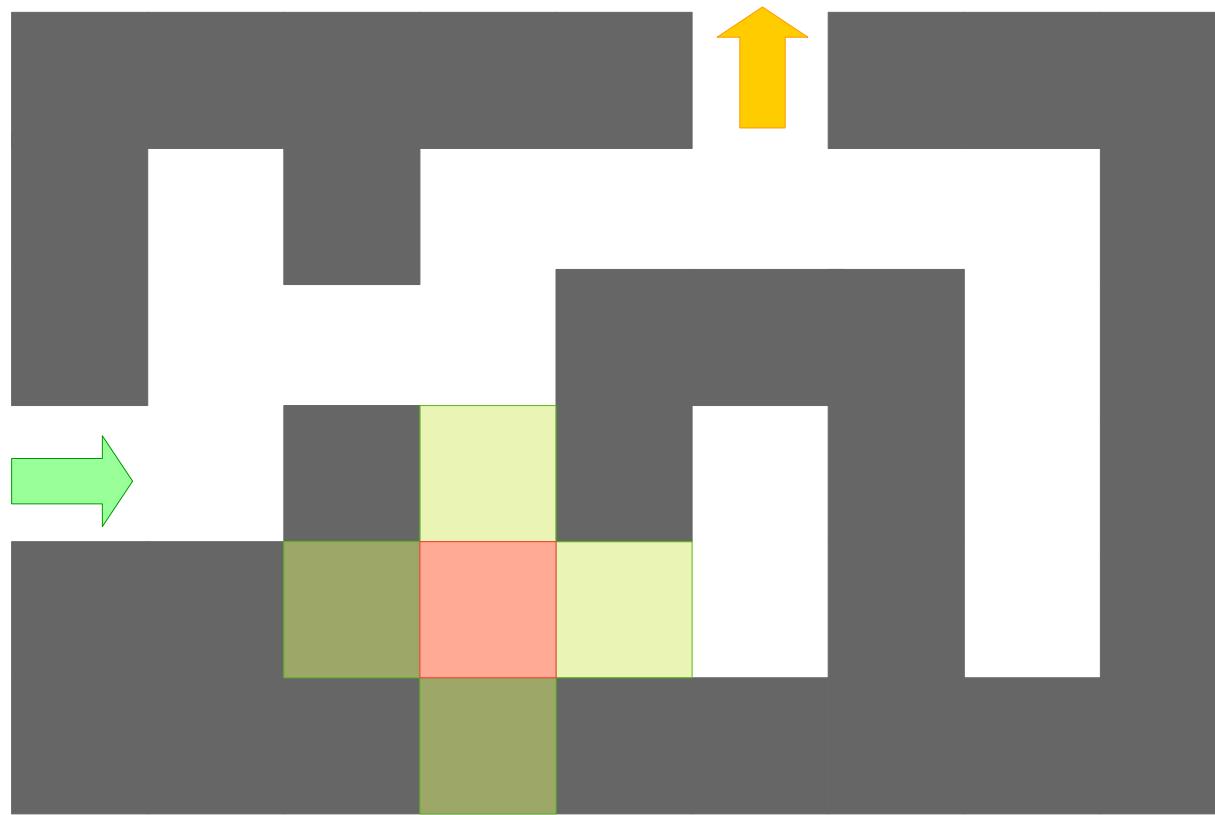
Le problème du labyrinthe



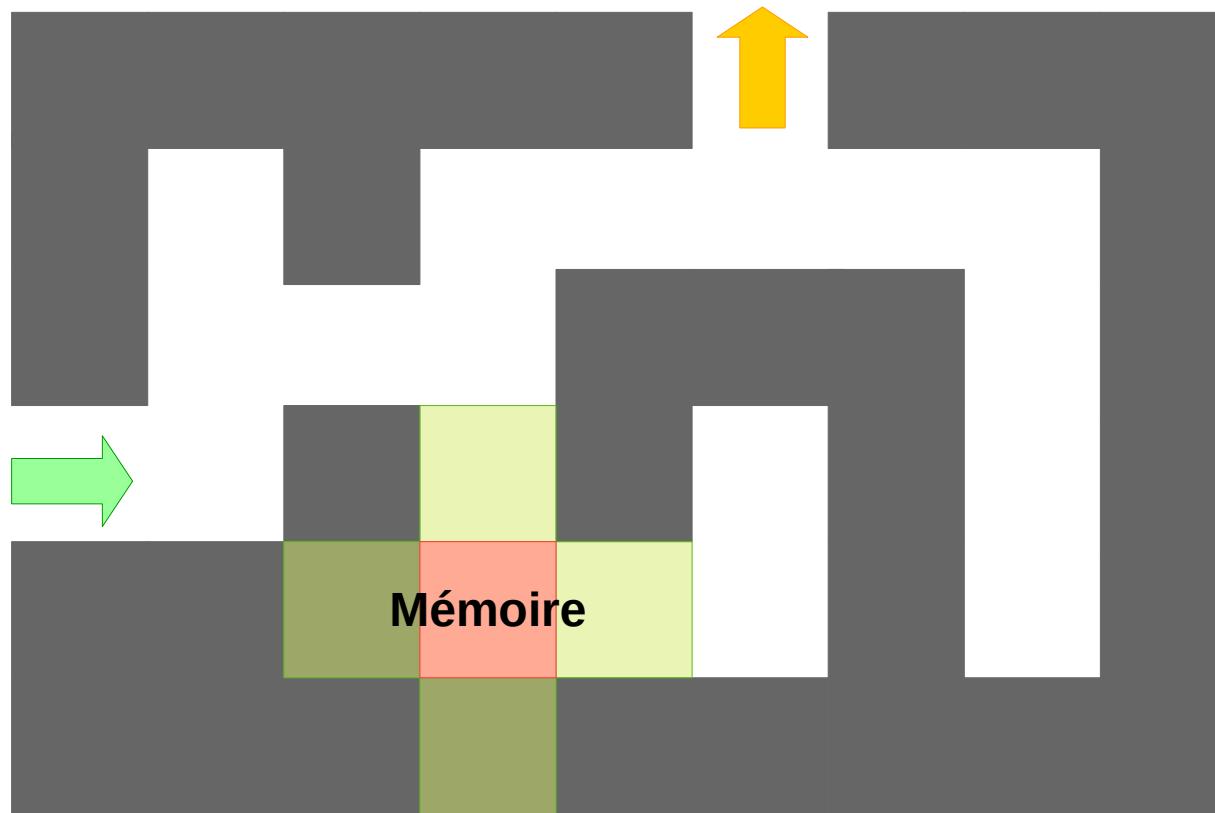
Le problème du labyrinthe



Le problème du labyrinthe

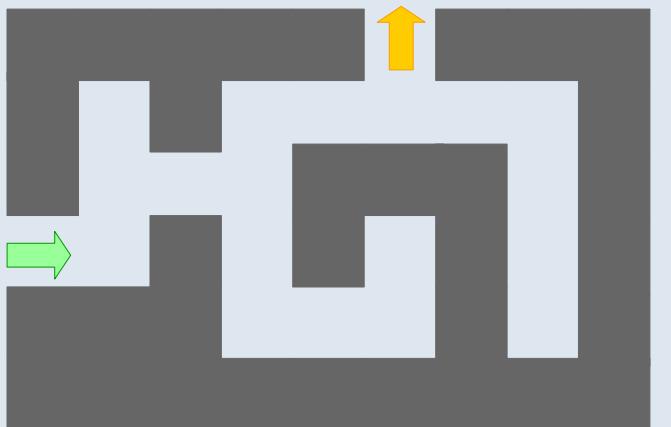


Le problème du labyrinthe



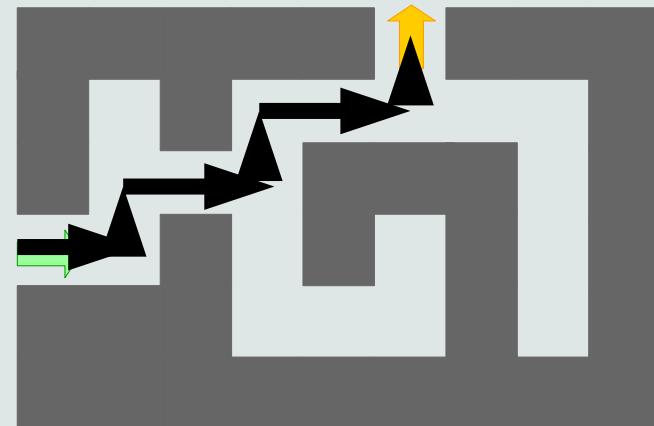
Le problème du labyrinthe

Donnée

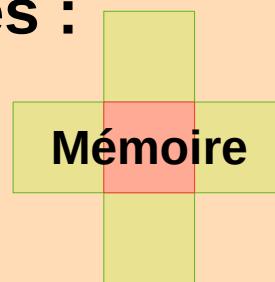


Problème

Trouver la solution optimale pour la résolution du labyrinthe

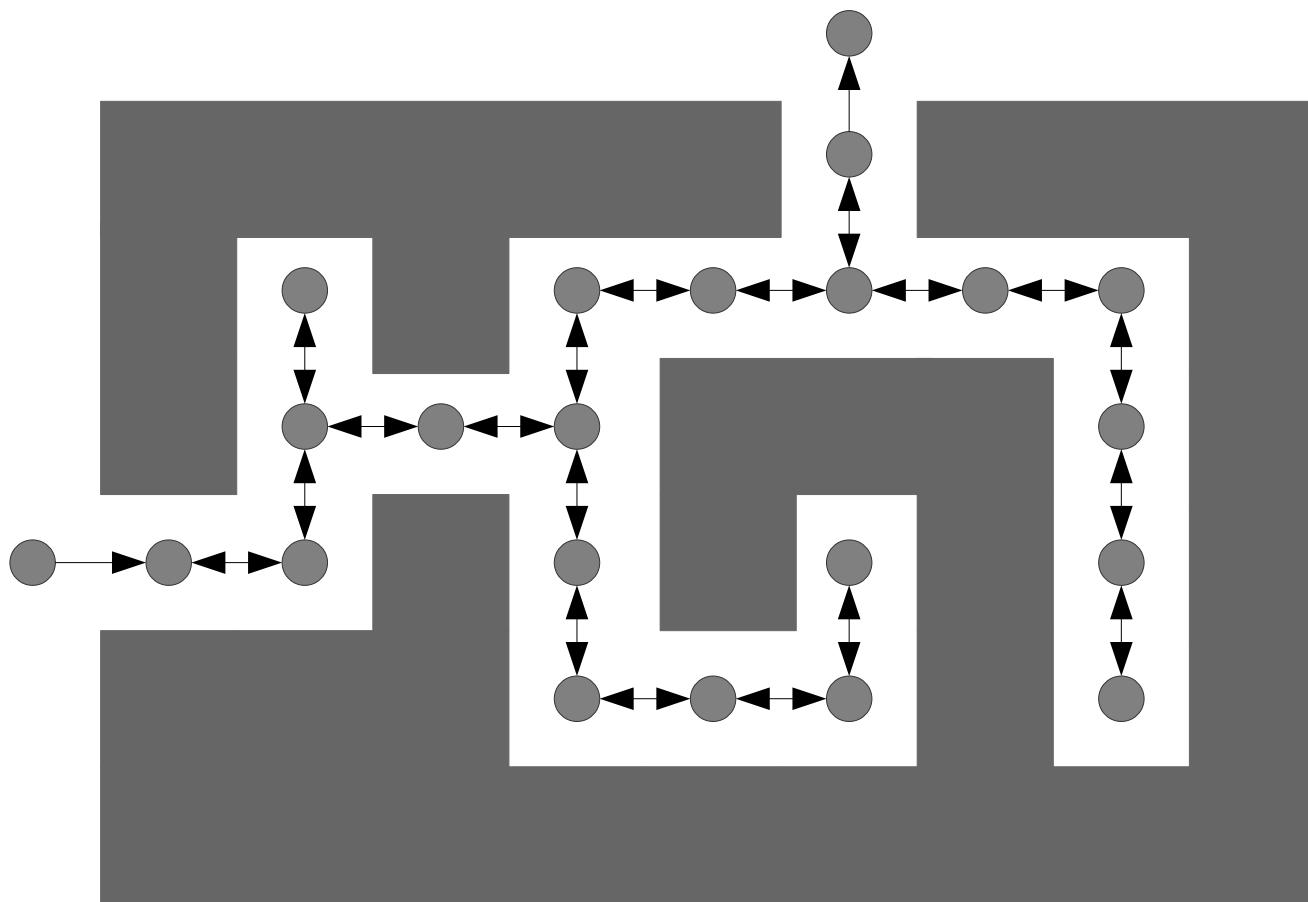


Contraintes :

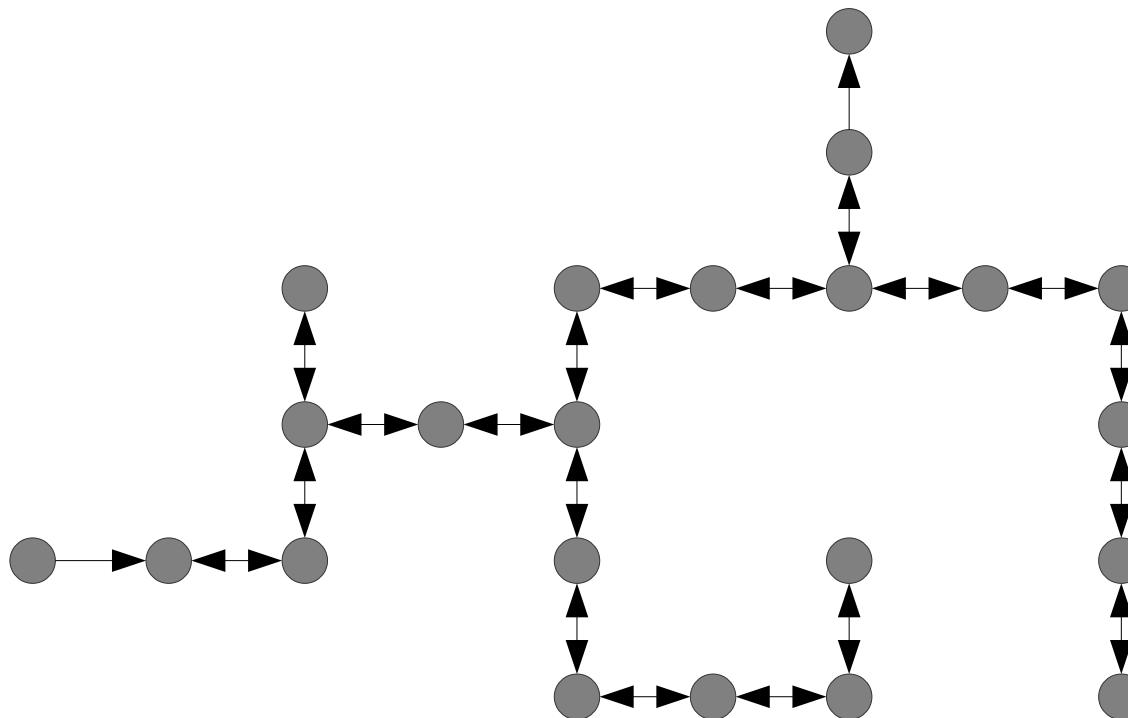


- (1) Trouver la solution le plus vite possible
- (2) Utiliser le moins d'espace mémoire possible

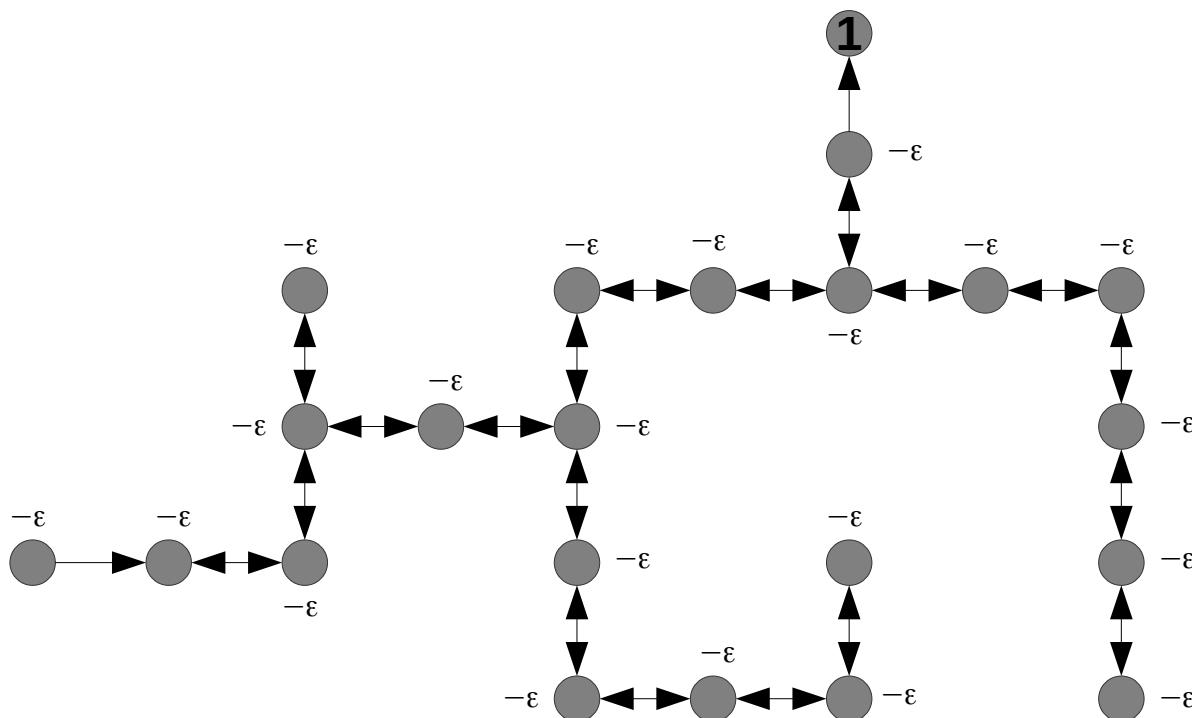
Apprentissage par renforcement



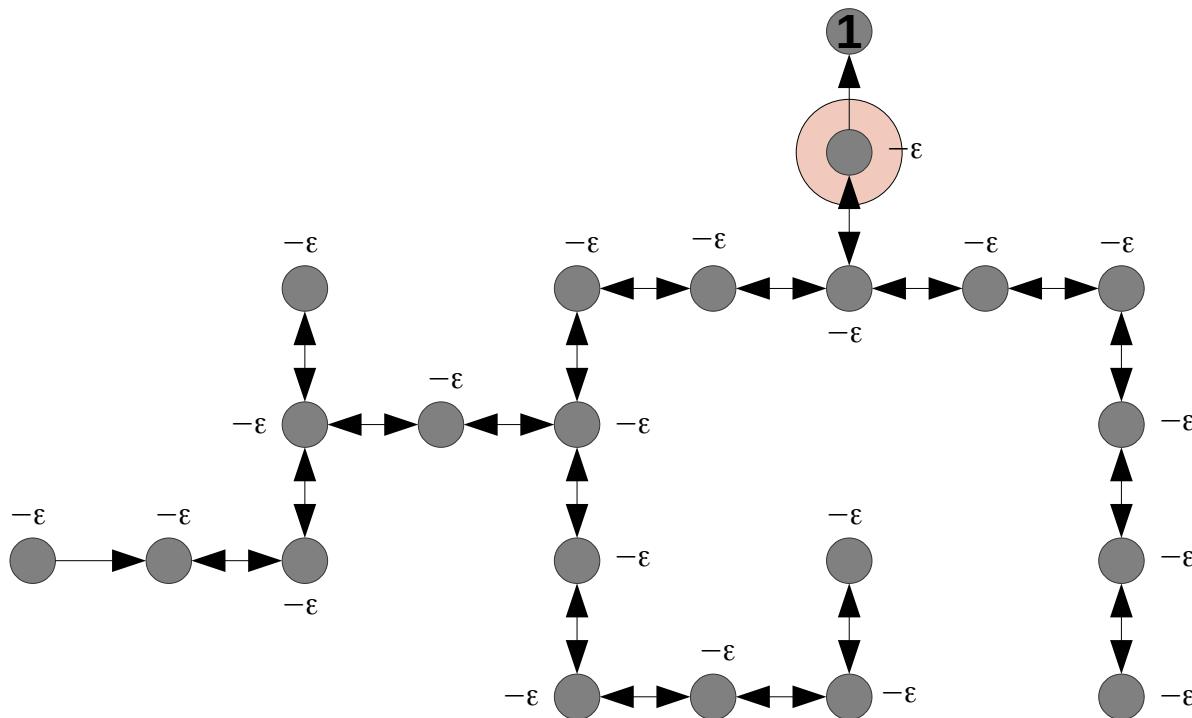
Apprentissage par renforcement



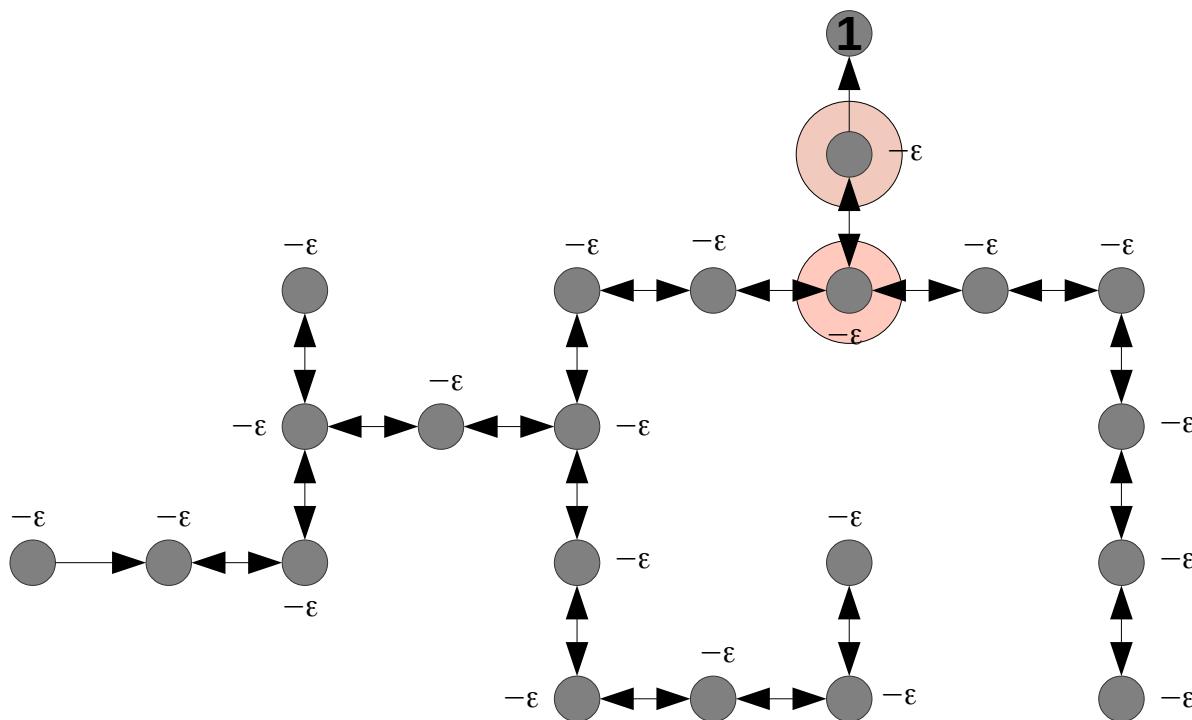
Apprentissage par renforcement



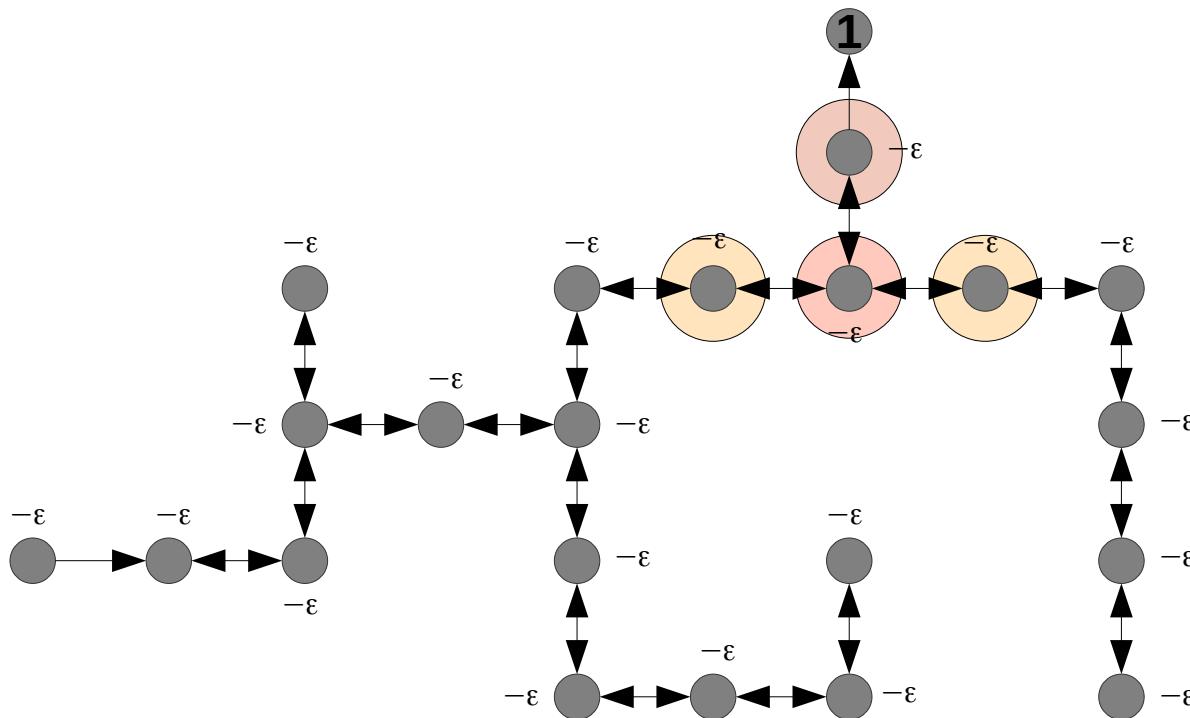
Apprentissage par renforcement



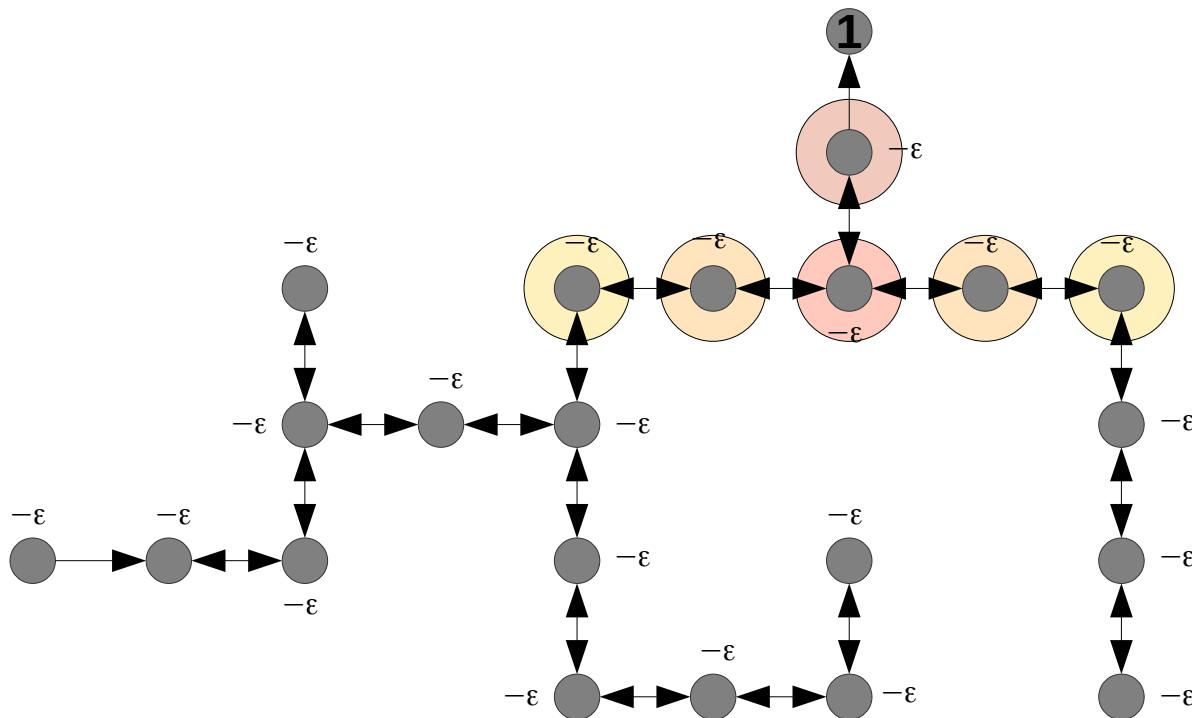
Apprentissage par renforcement



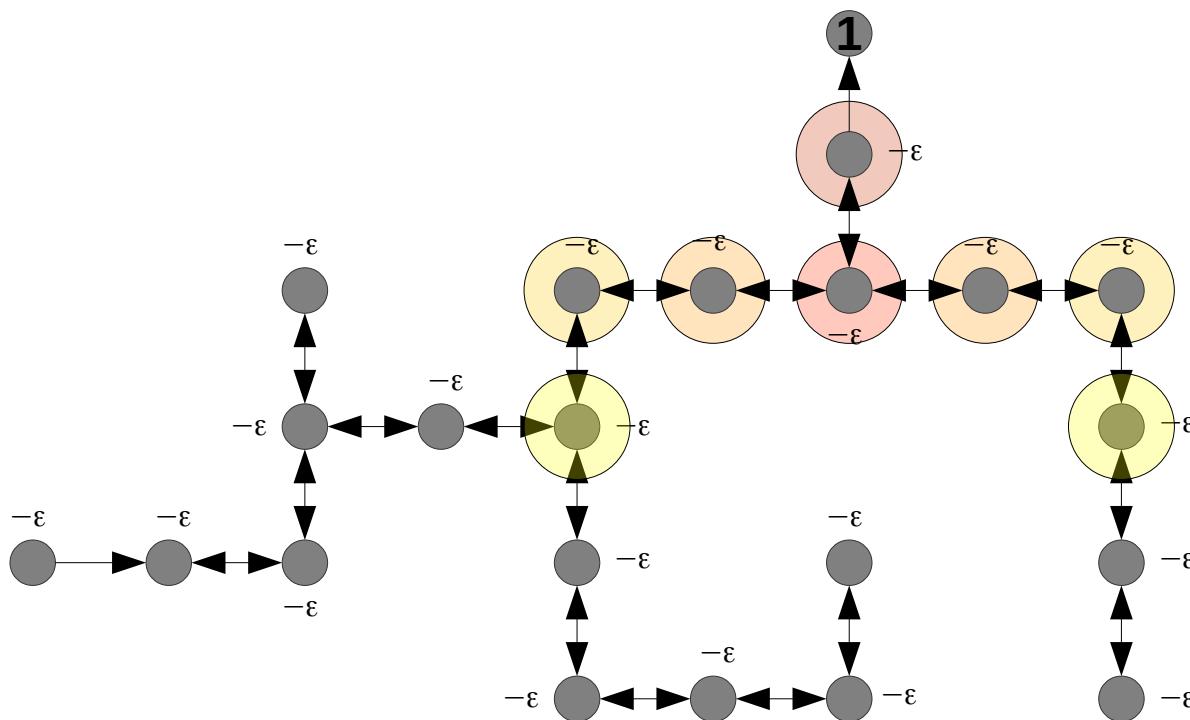
Apprentissage par renforcement



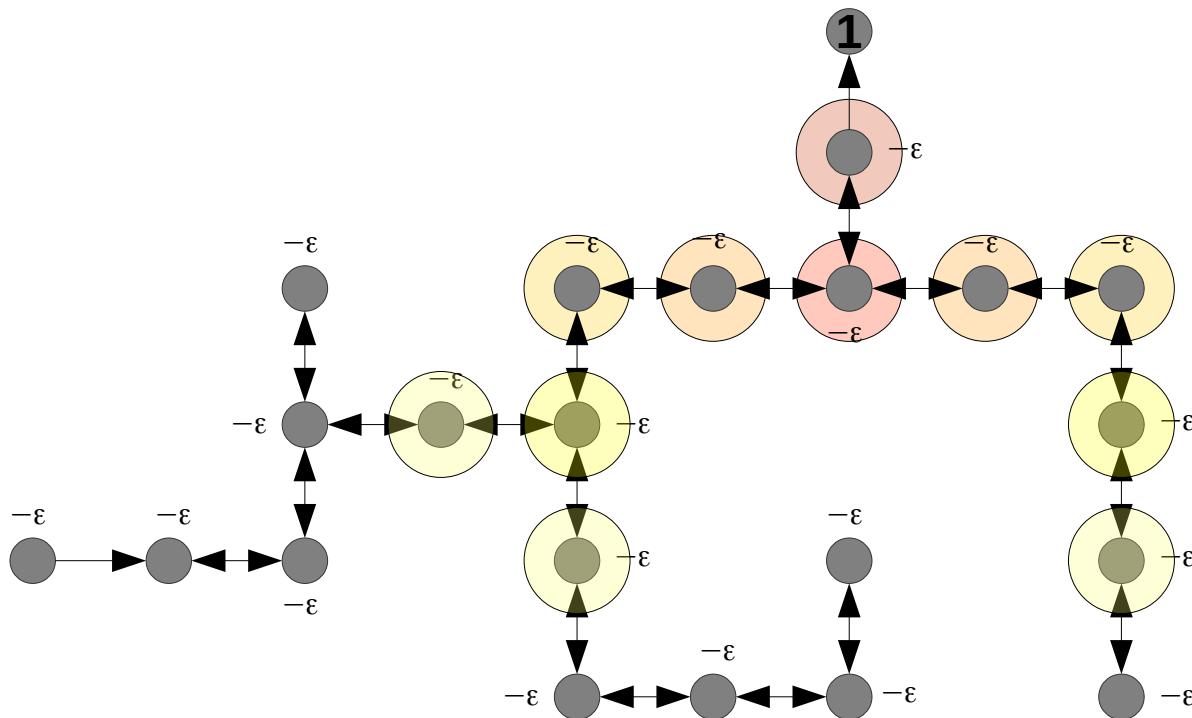
Apprentissage par renforcement



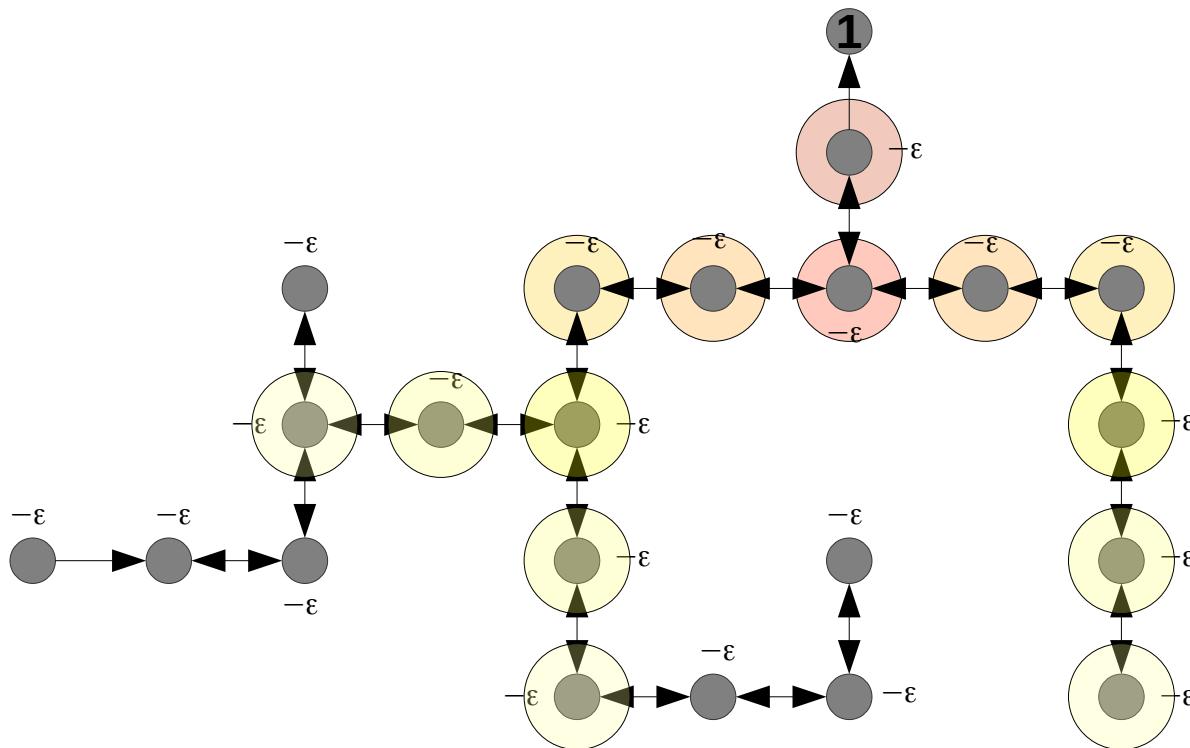
Apprentissage par renforcement



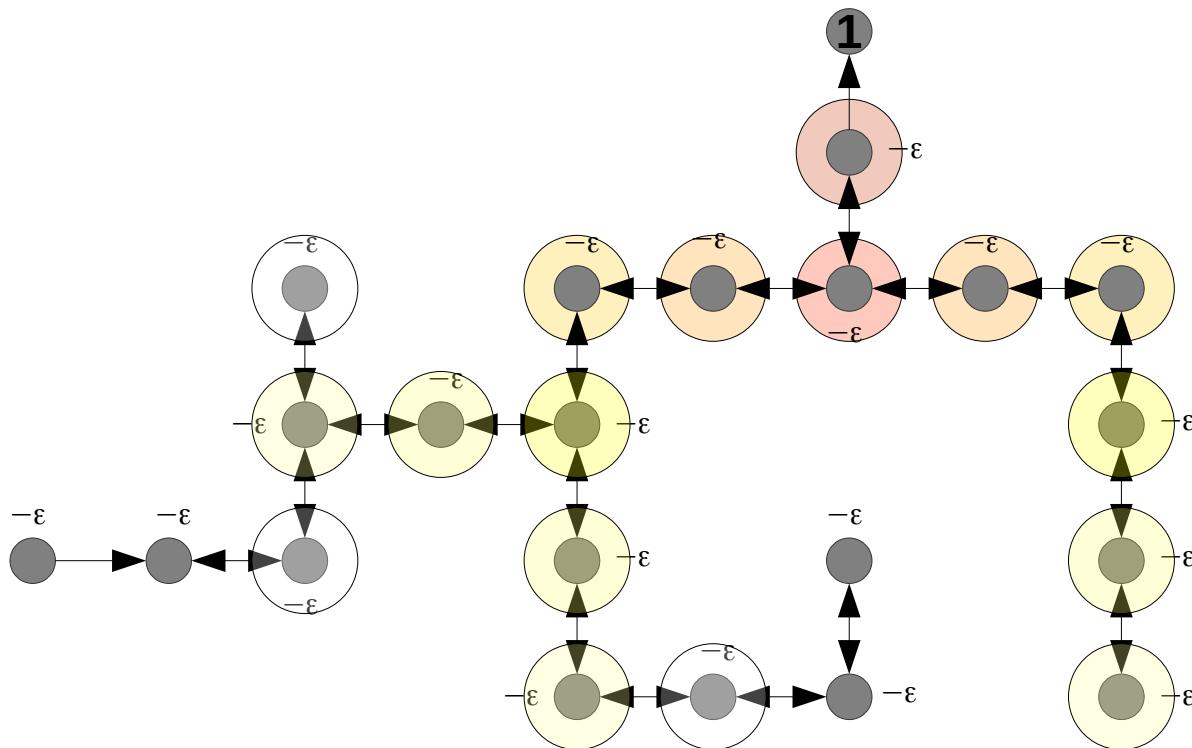
Apprentissage par renforcement



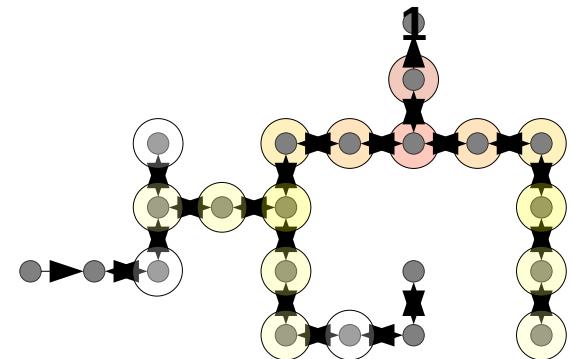
Apprentissage par renforcement



Apprentissage par renforcement

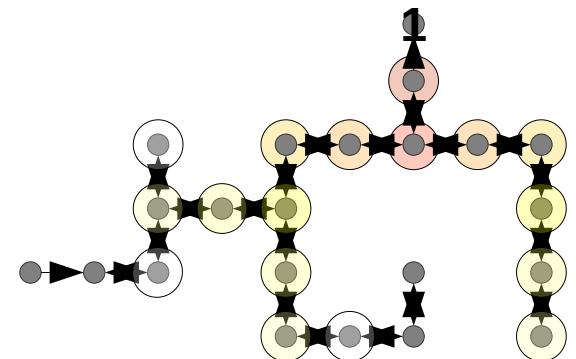


Q-Learning tabulaire



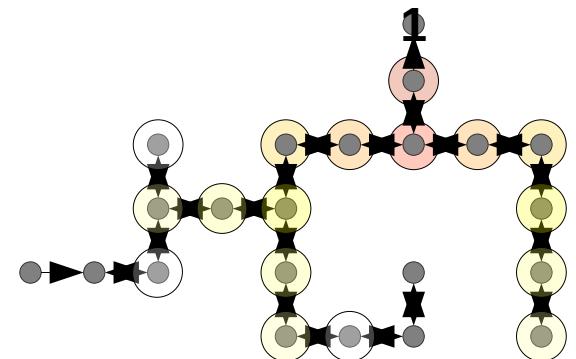
$Q[s] += \alpha($

Q-Learning tabulaire



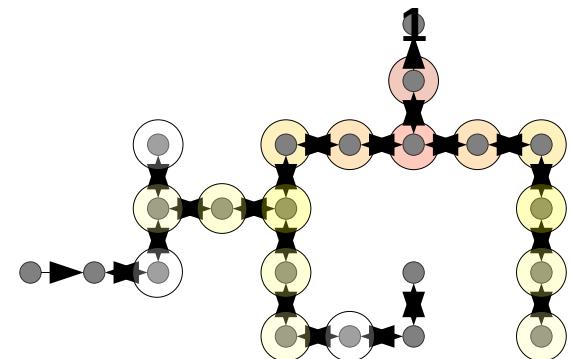
$$Q[s] += \alpha(r - Q[s])$$

Q-Learning tabulaire



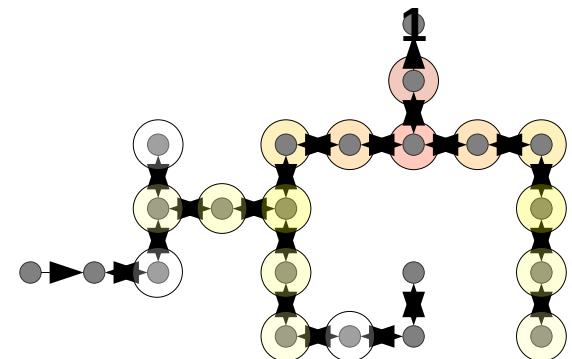
$$Q[s] += \alpha(r - Q[s] + \max_d(Q[\text{move}(s,d)]))$$

Q-Learning tabulaire



$$Q[s] += \alpha(r - Q[s] + \gamma \max_d(Q[\text{move}(s,d)]))$$

Q-Learning tabulaire



$$Q[s] += \alpha(r - Q[s] + \gamma \max_d(Q[\text{move}(s,d)]))$$

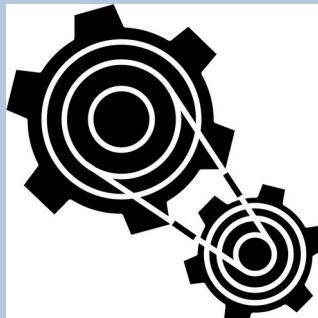
0.3 0.92

Architecture du Machine Learning

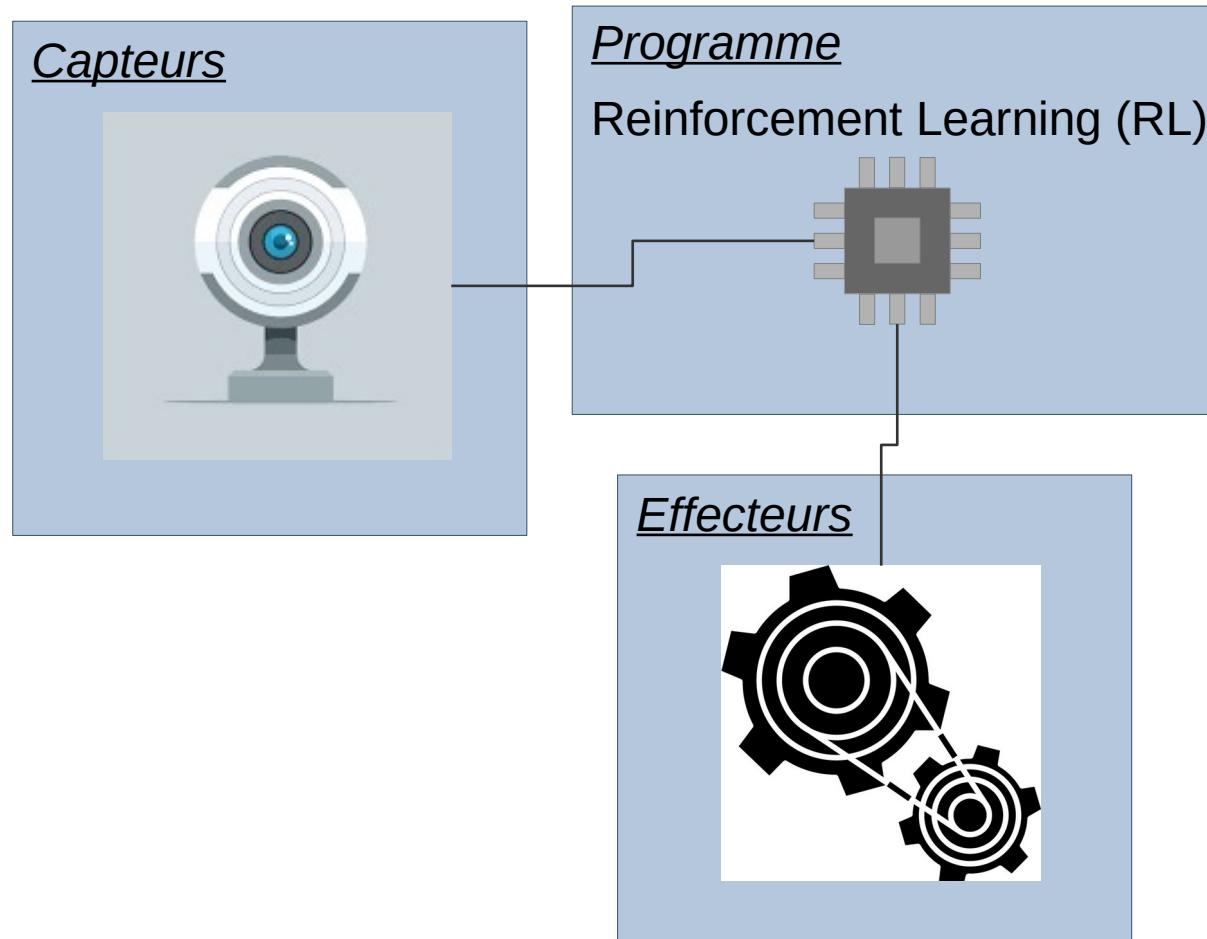
Capteurs



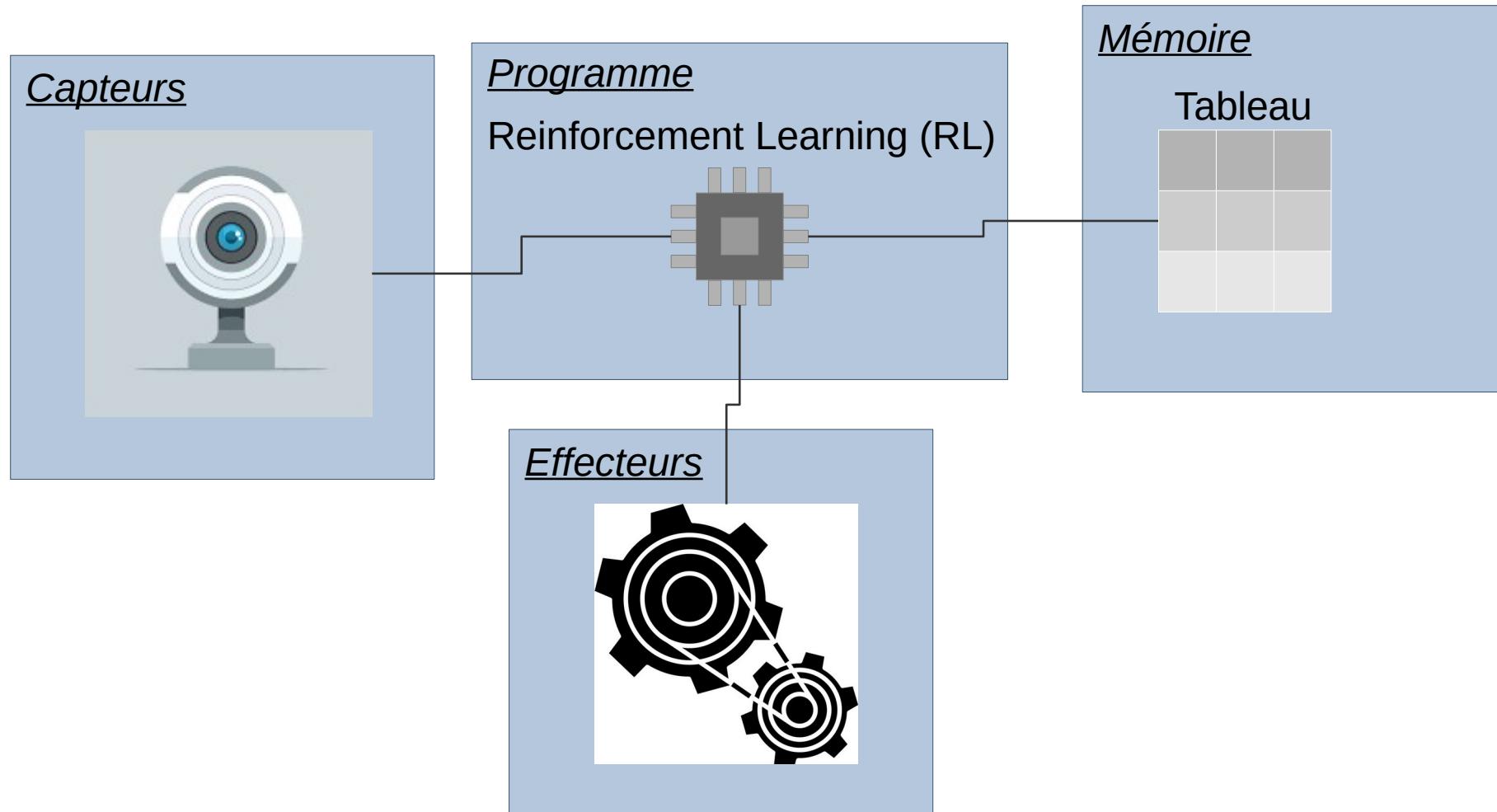
Effecteurs



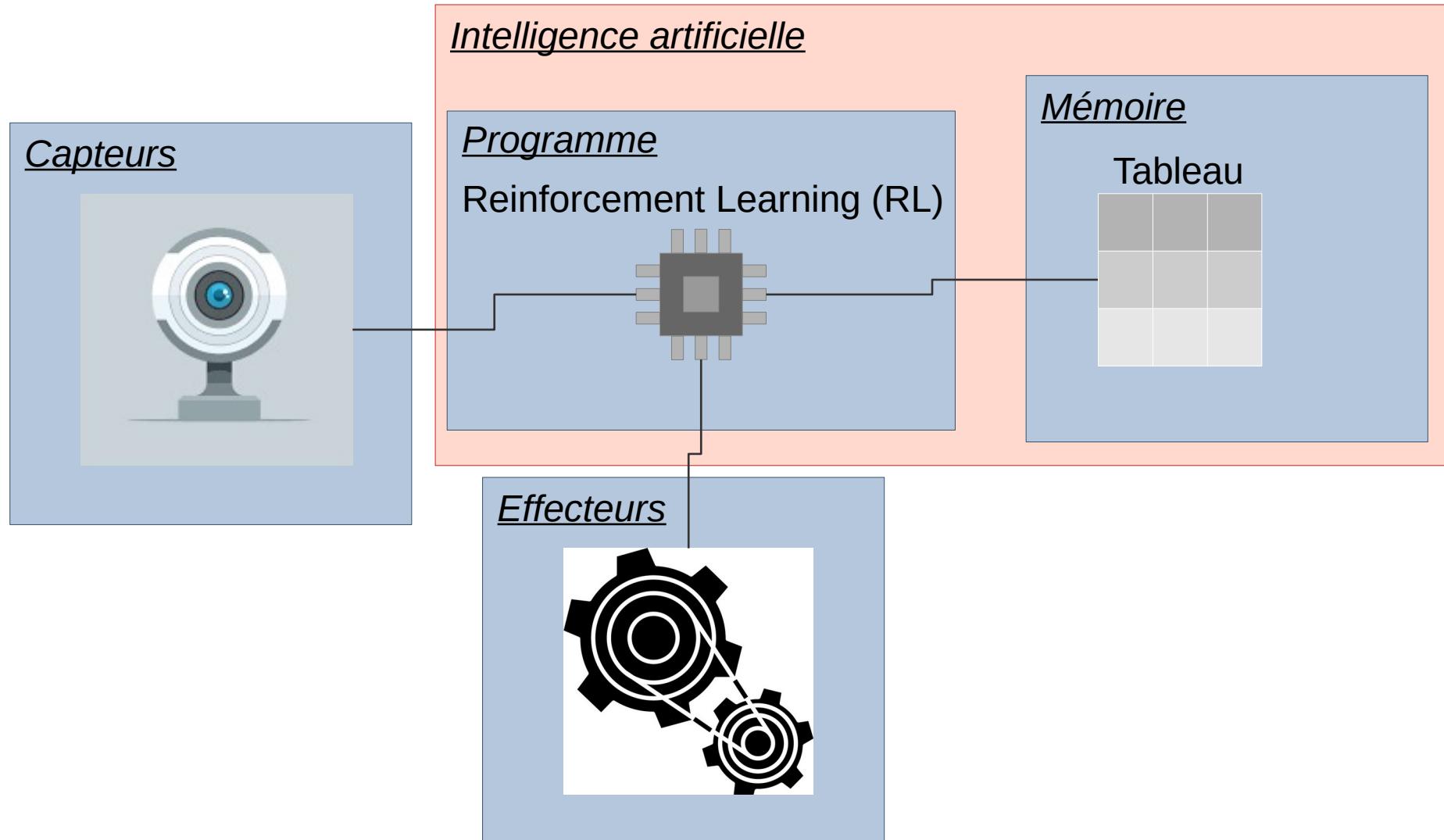
Architecture du Machine Learning



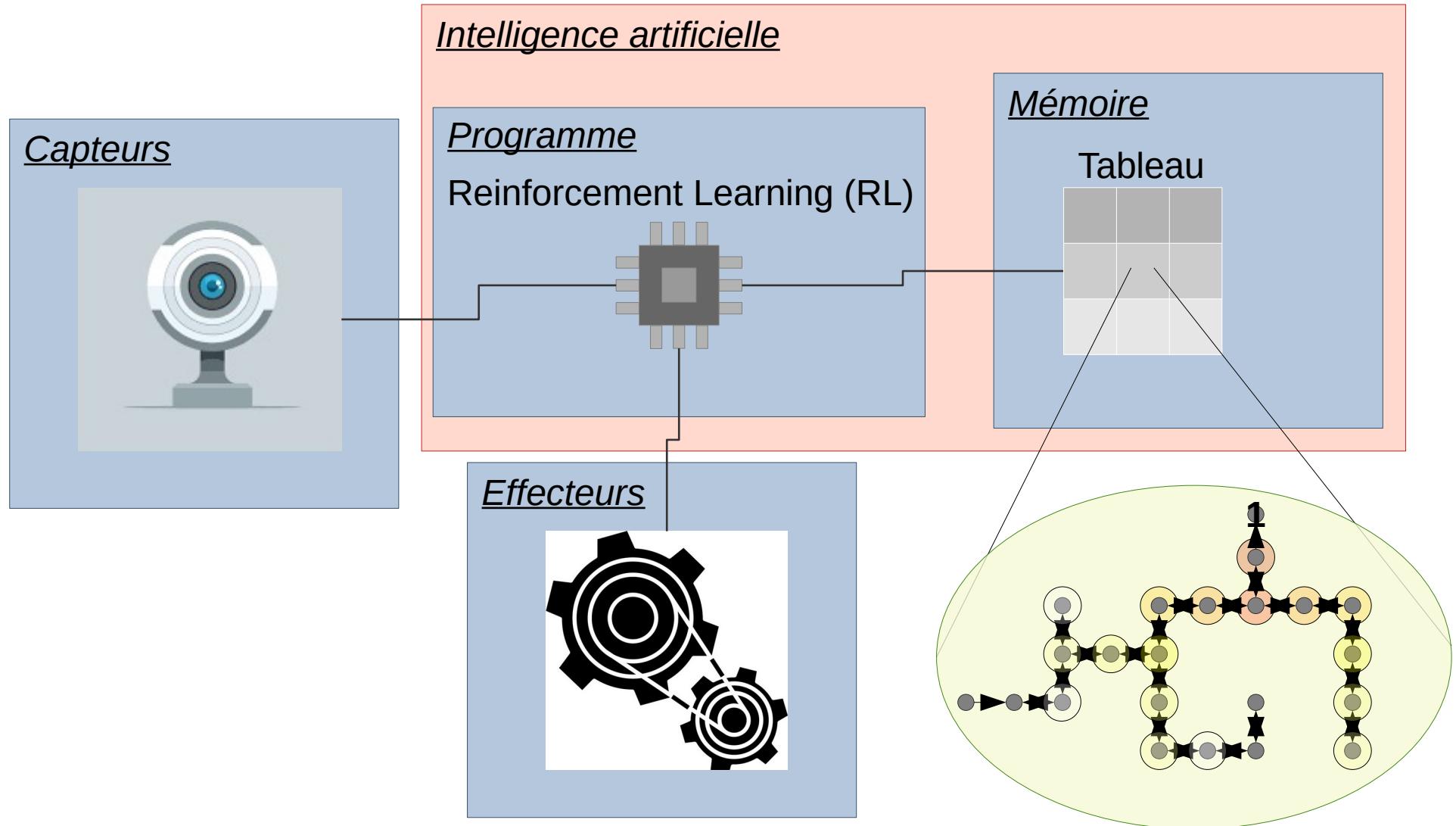
Architecture du Machine Learning



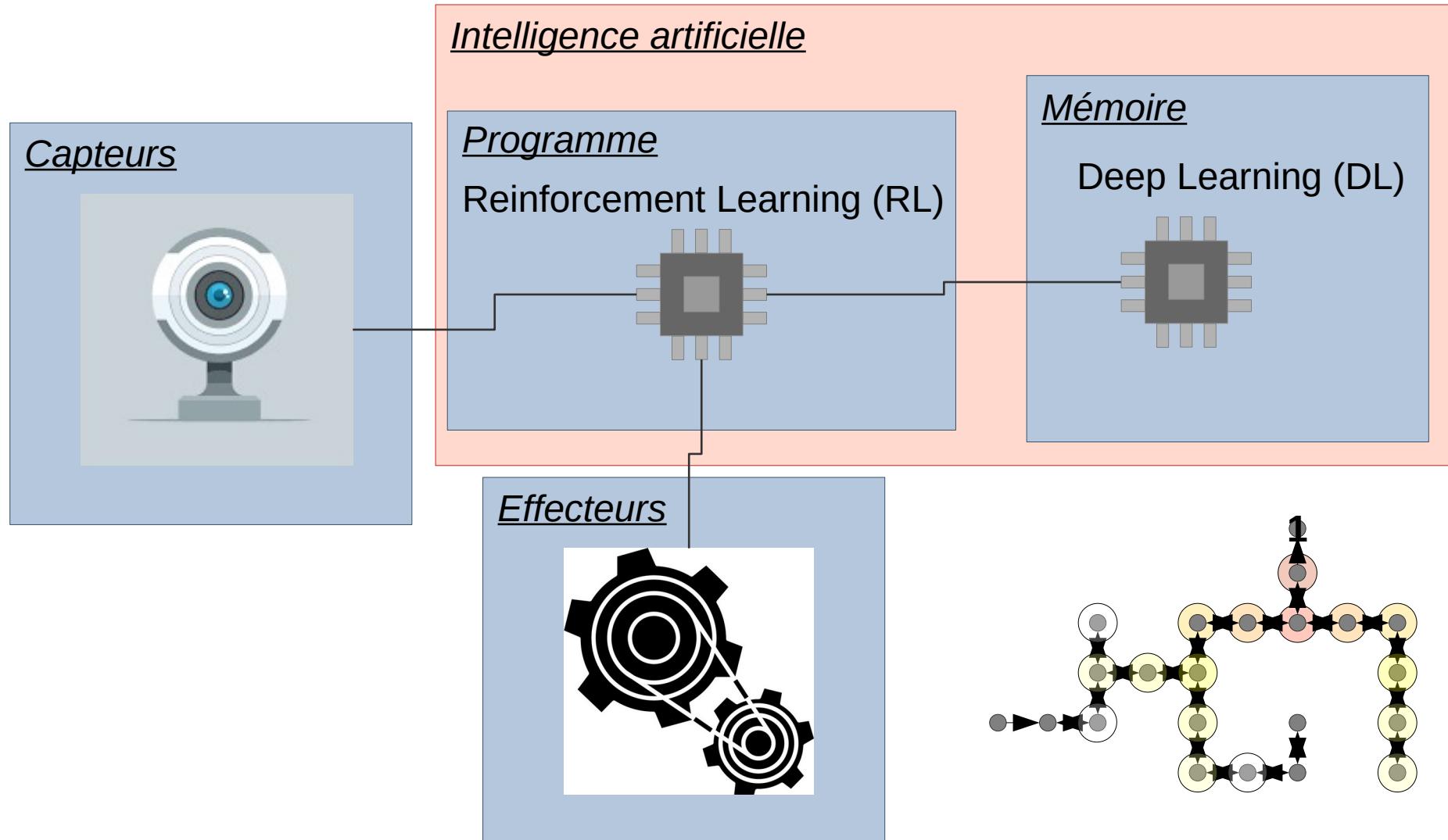
Architecture du Machine Learning



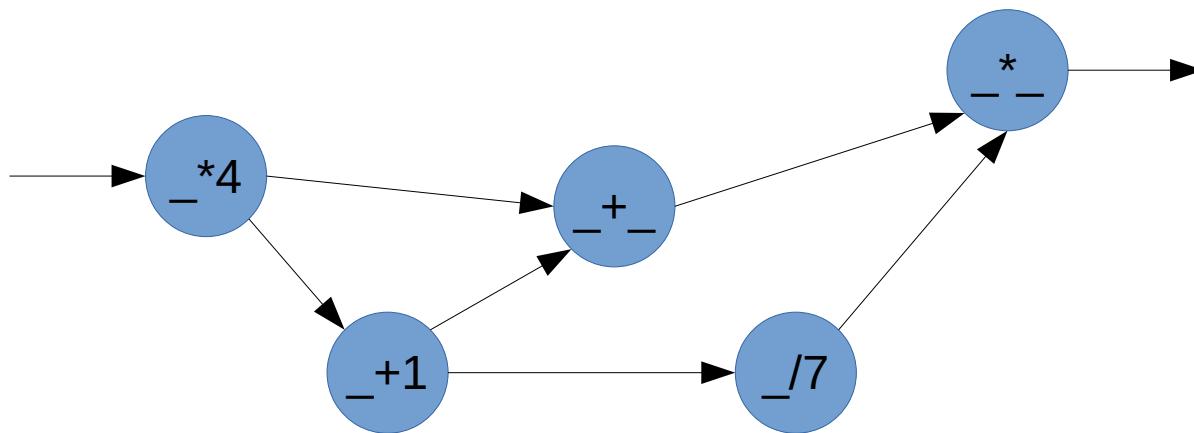
Architecture du Machine Learning



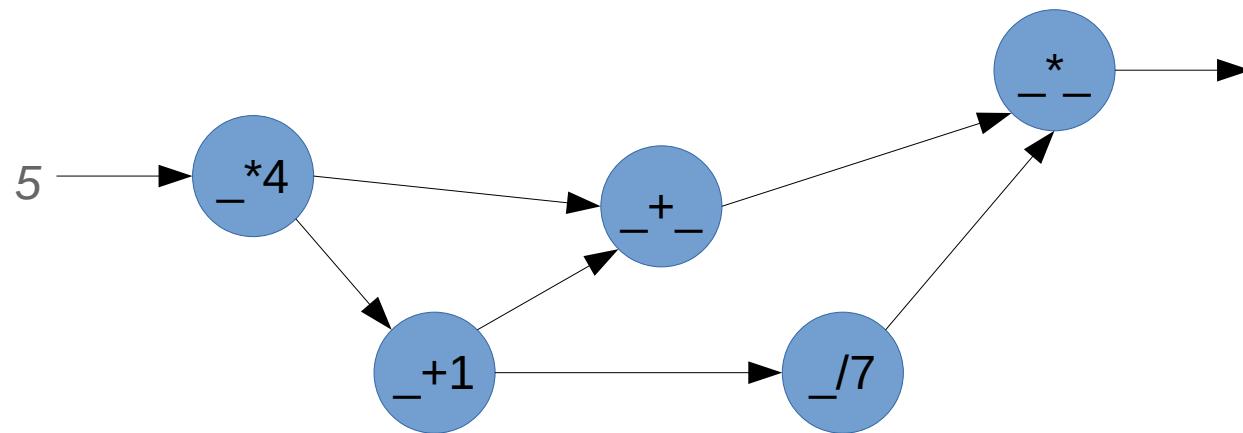
Architecture du Machine Learning



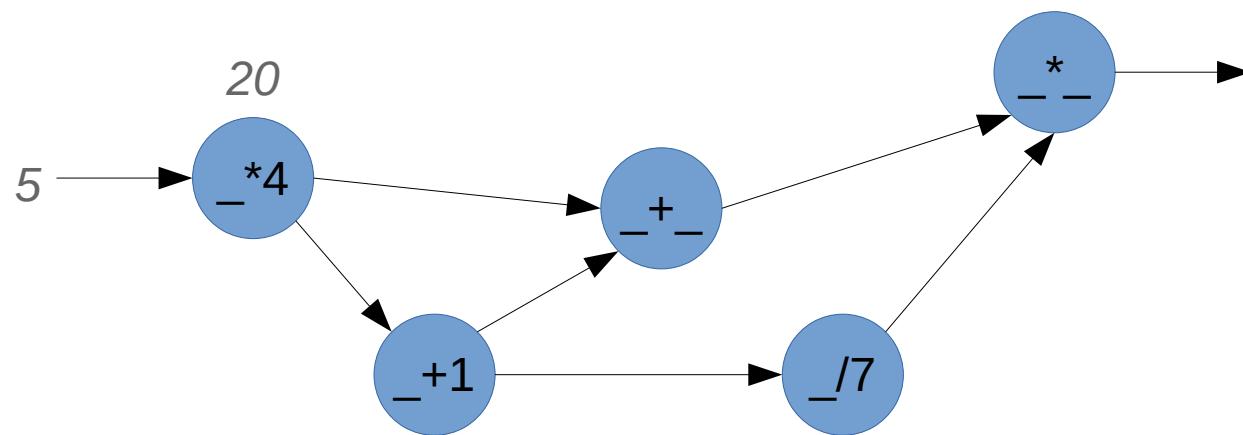
Calcul sur un graphe



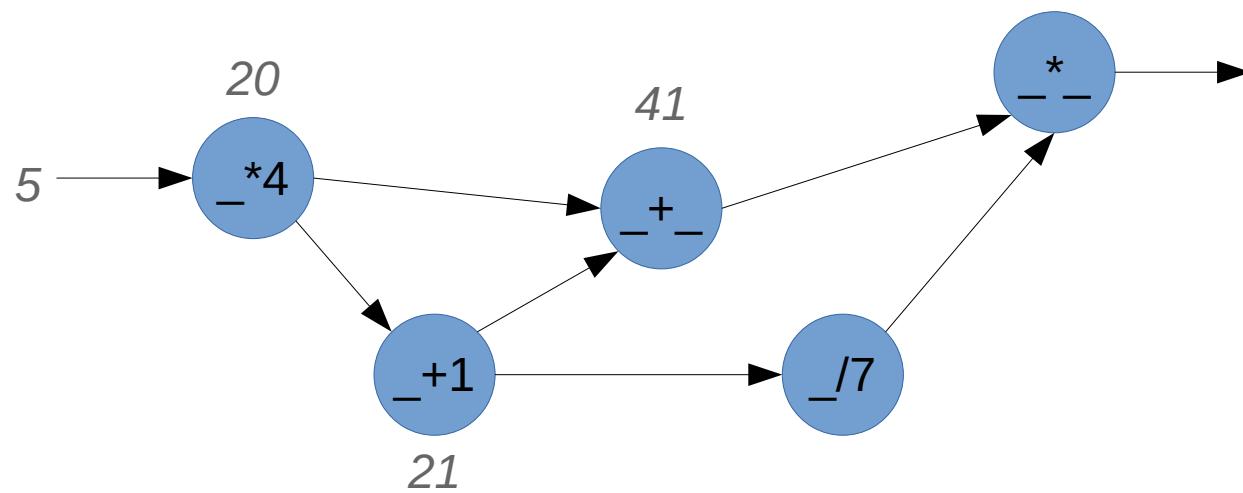
Calcul sur un graphe



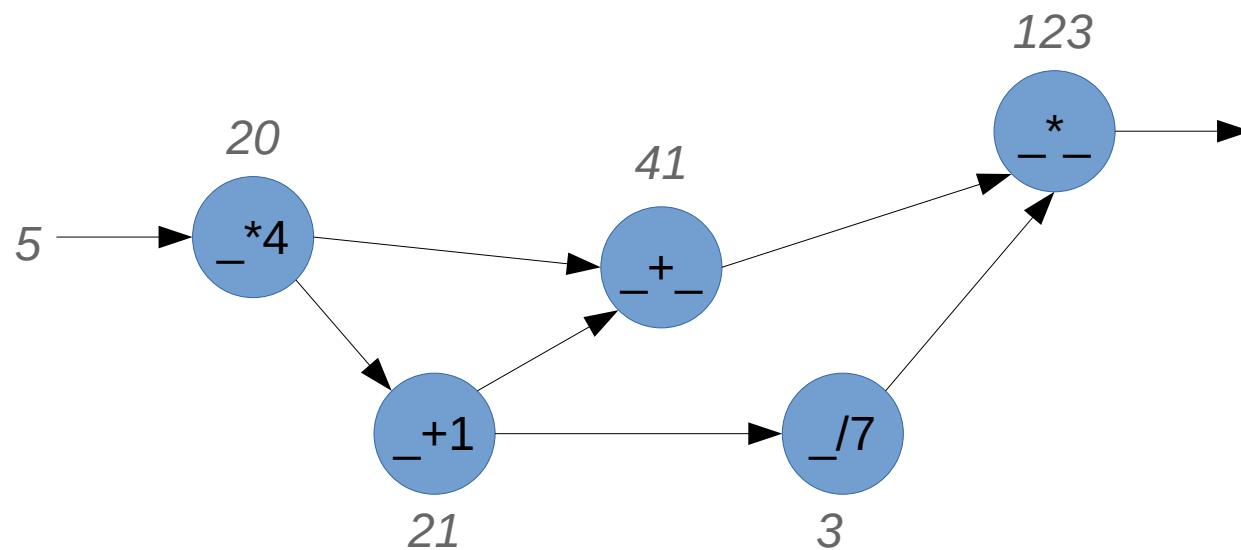
Calcul sur un graphe



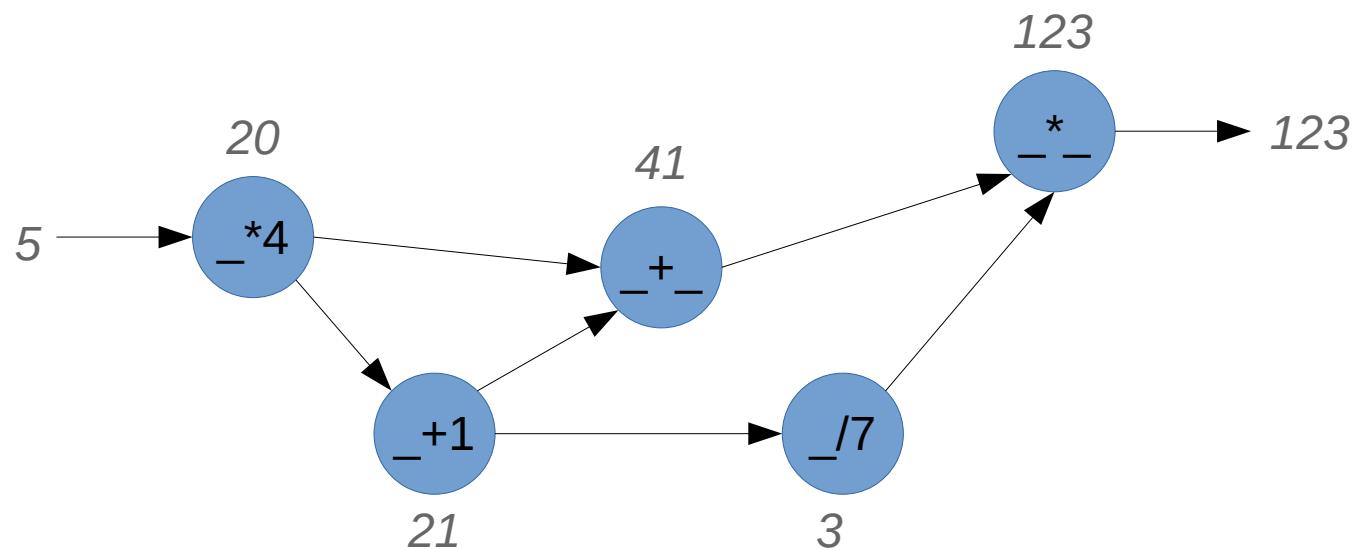
Calcul sur un graphe



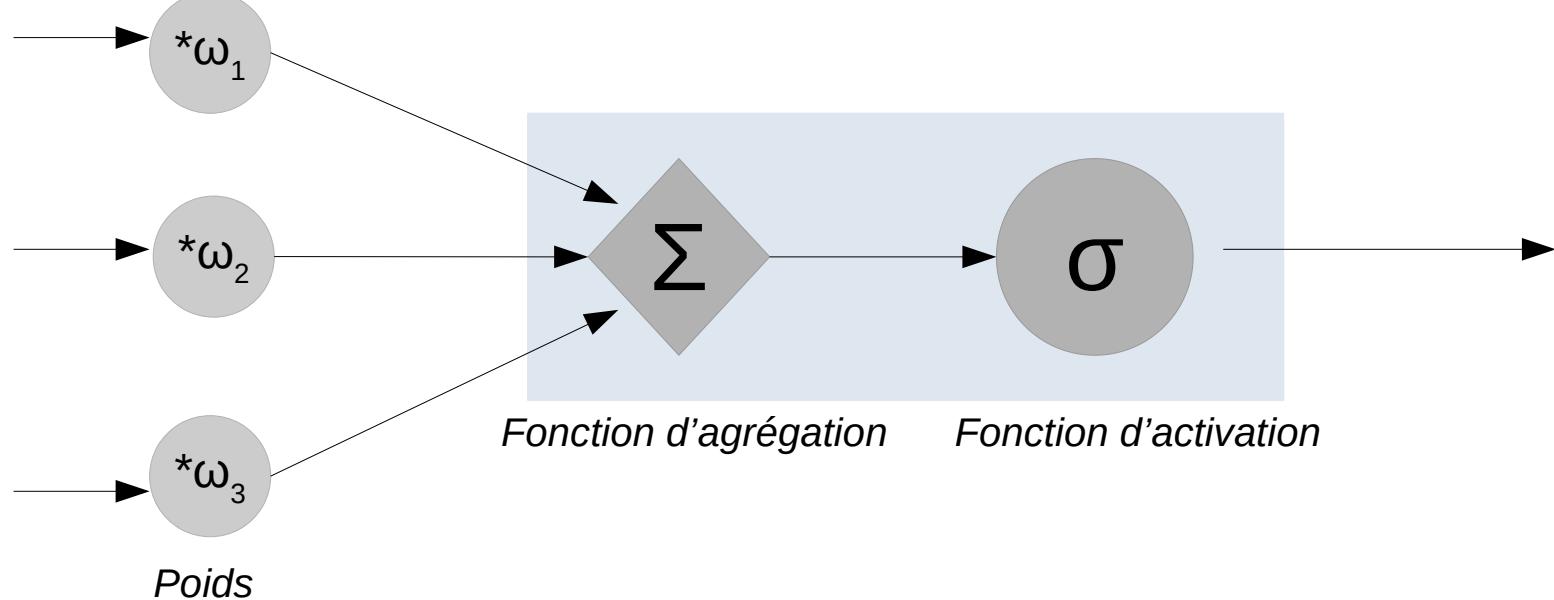
Calcul sur un graphe



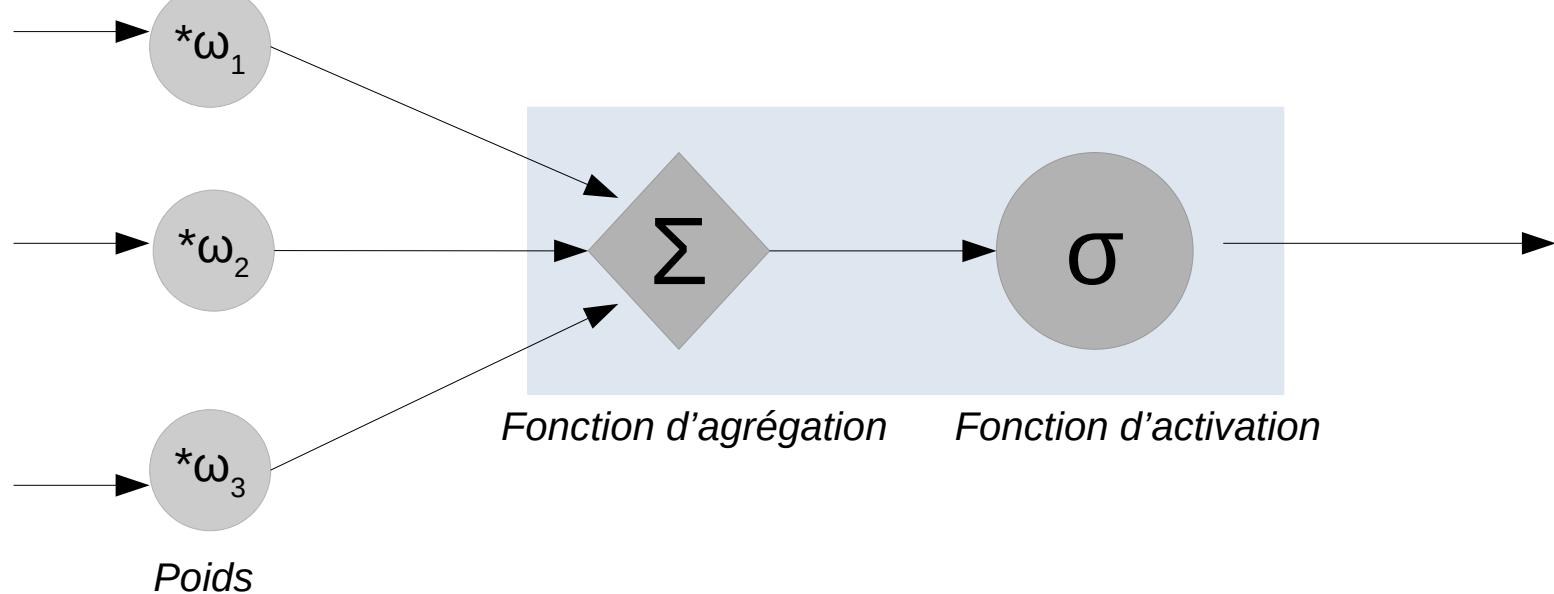
Calcul sur un graphe



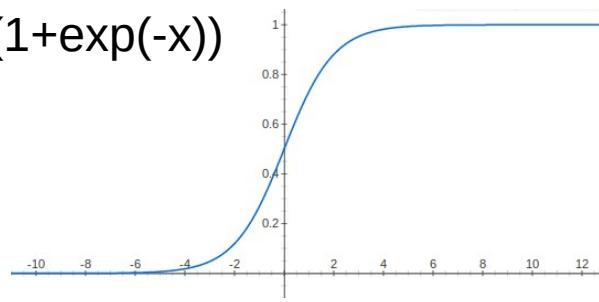
Le neurone classique



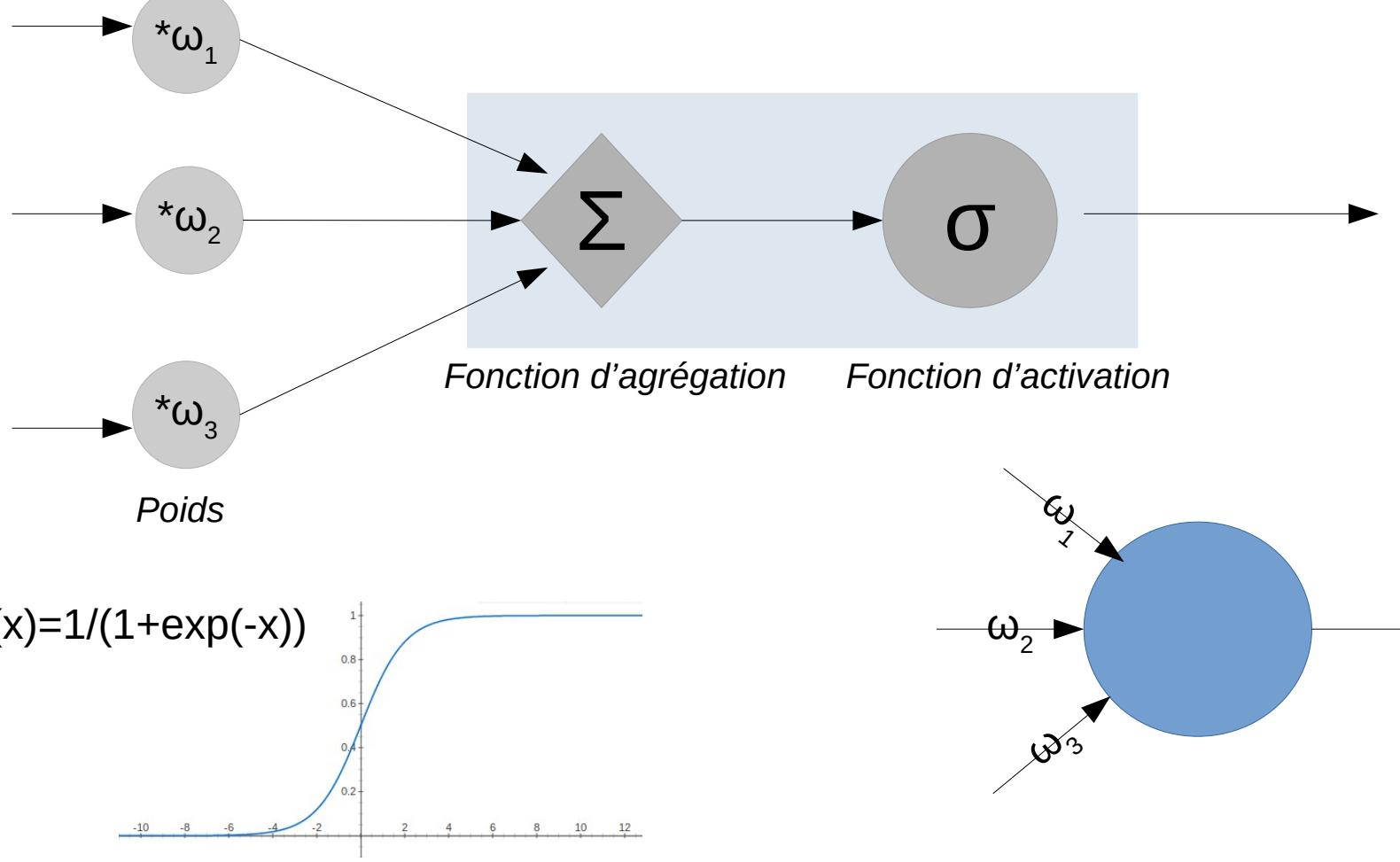
Le neurone classique



$$\sigma(x) = 1/(1+\exp(-x))$$

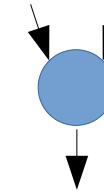


Le neurone classique



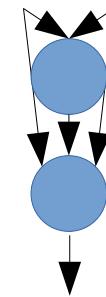
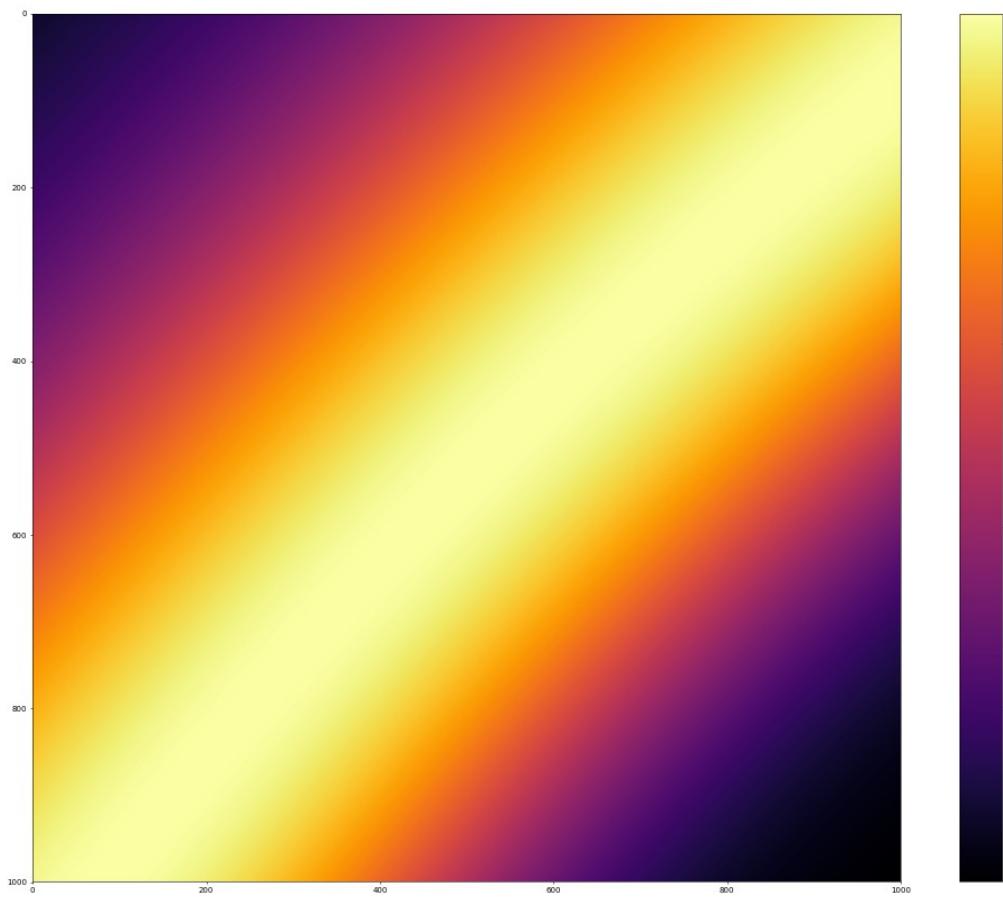
Limites des neurones

OR

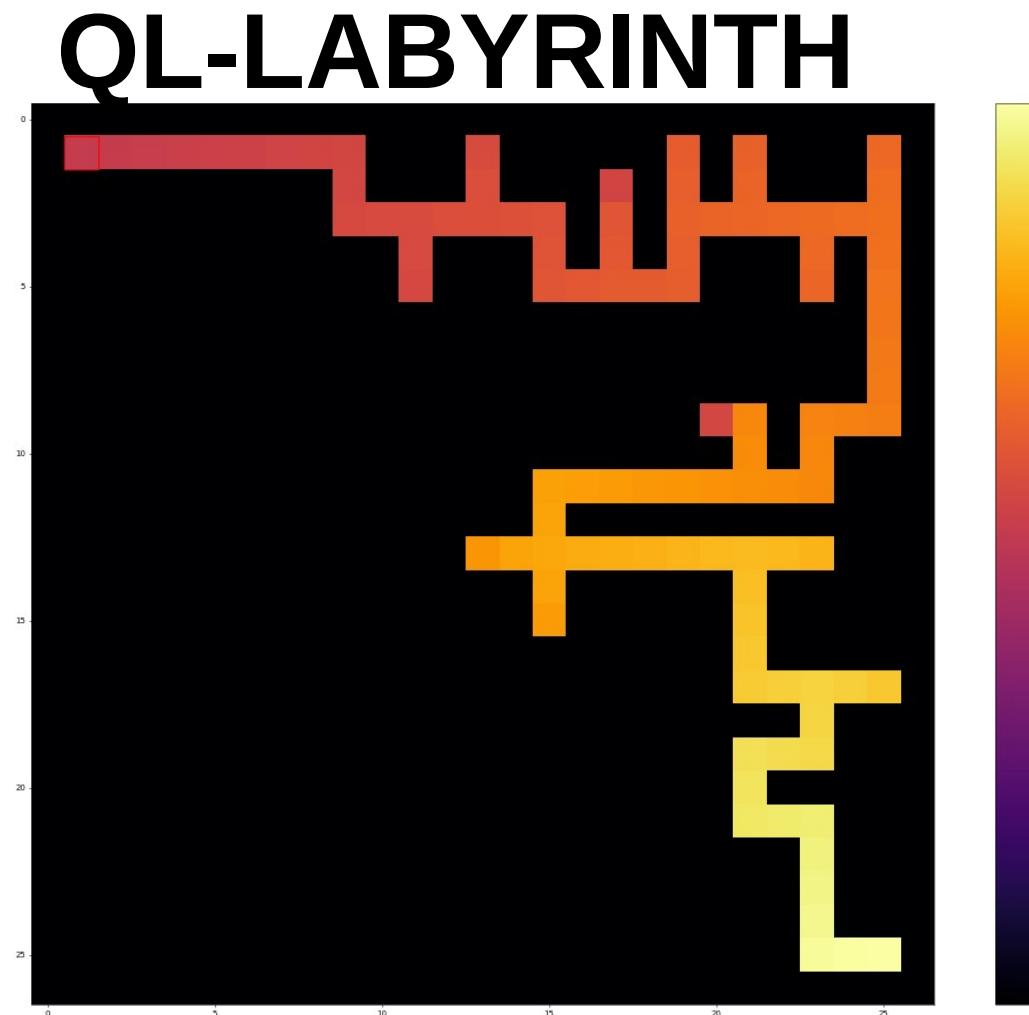


Limites des neurones

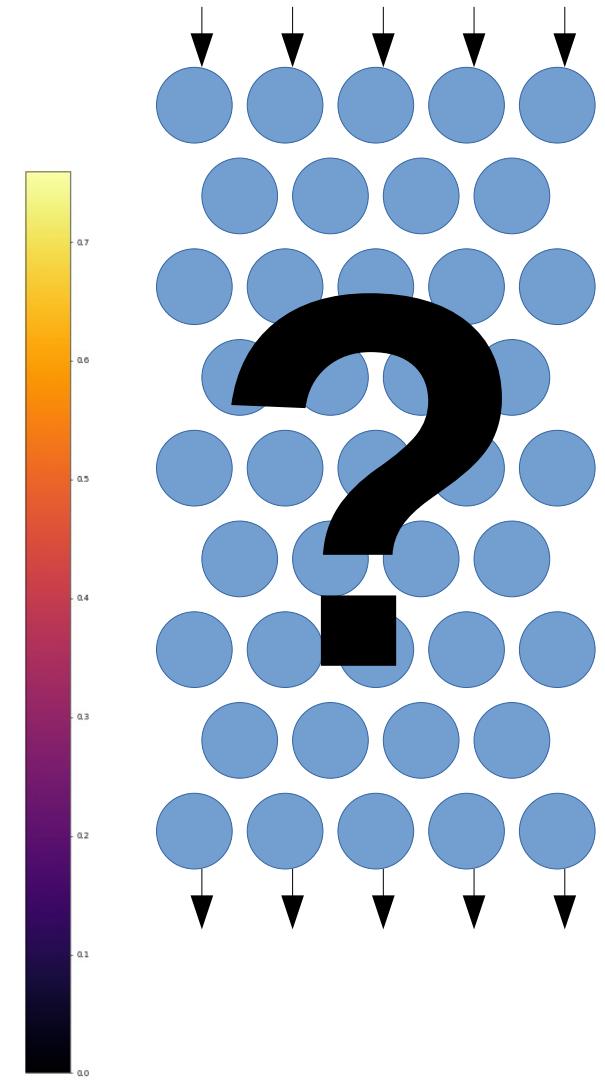
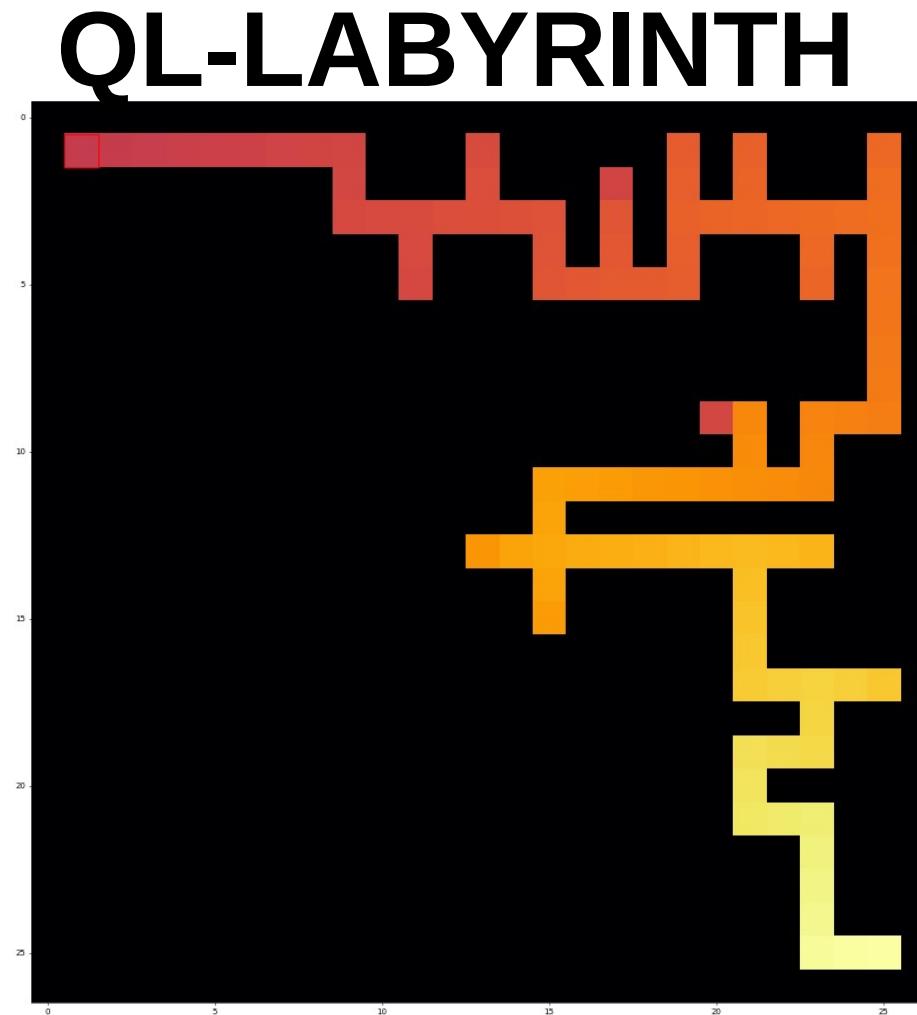
XOR



Limites des neurones



Limites des neurones

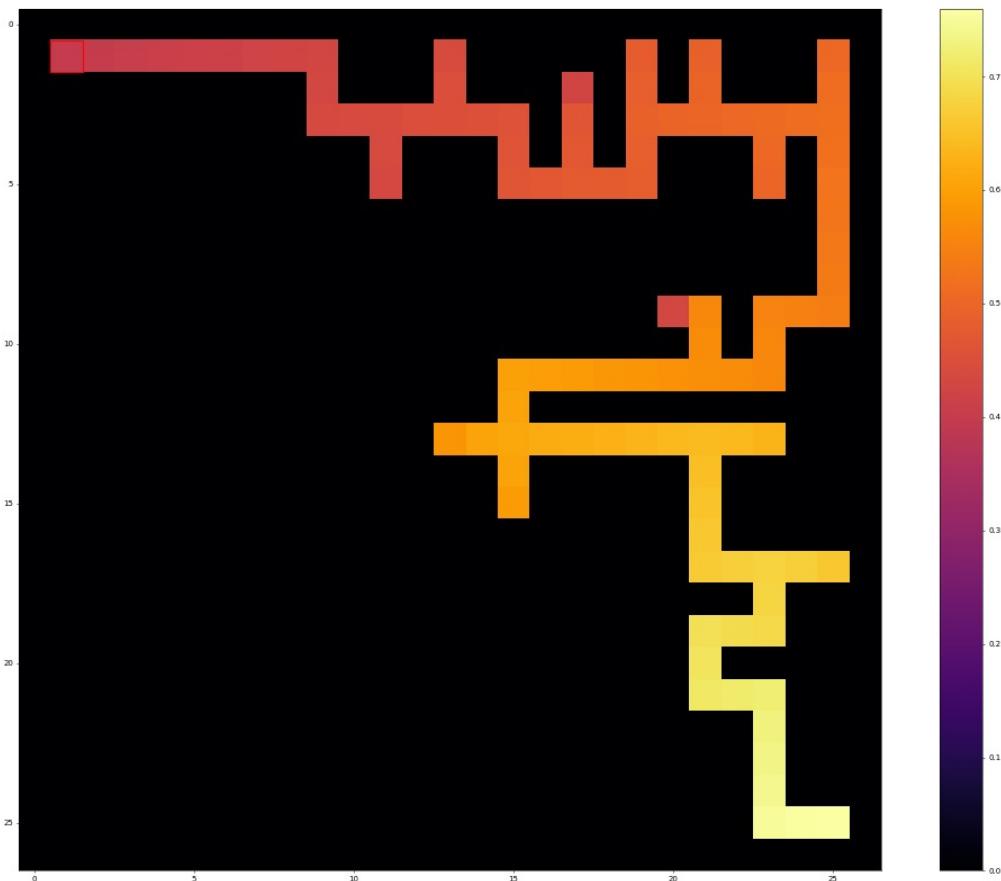


Sujet du Stage

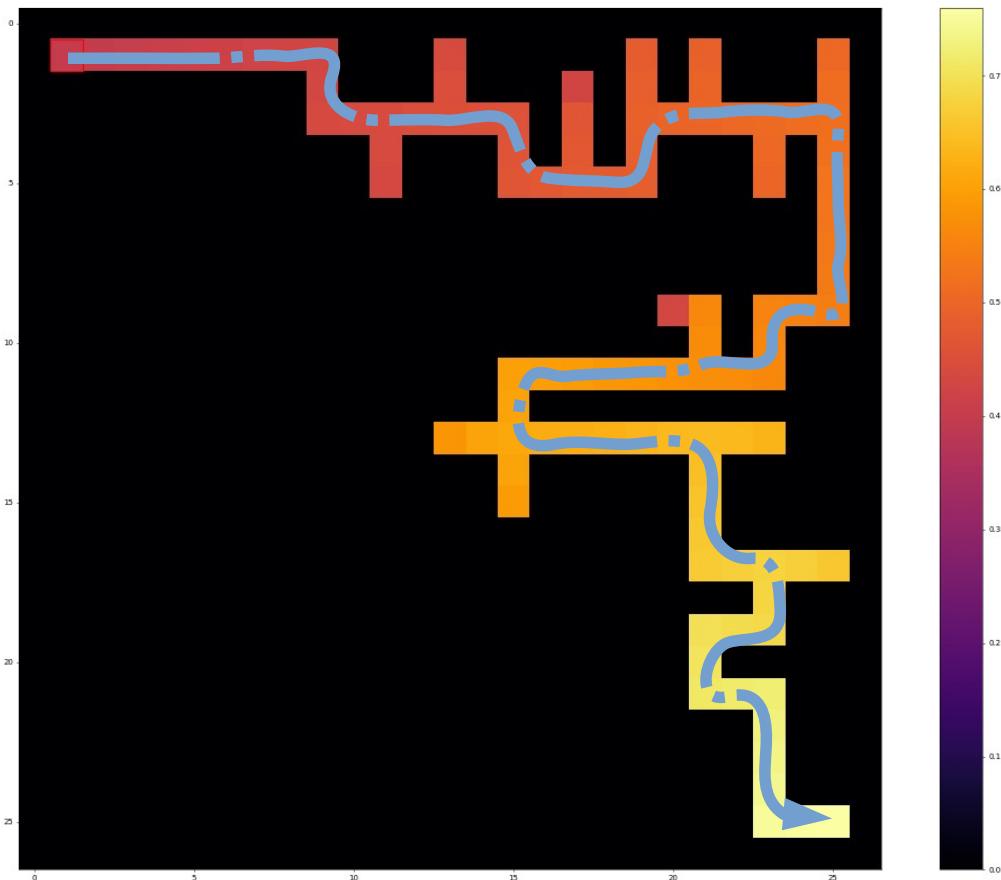
Résolution du problème du labyrinthe

- par **apprentissage par renforcement**
- en utilisant un **réseau de neurones**.

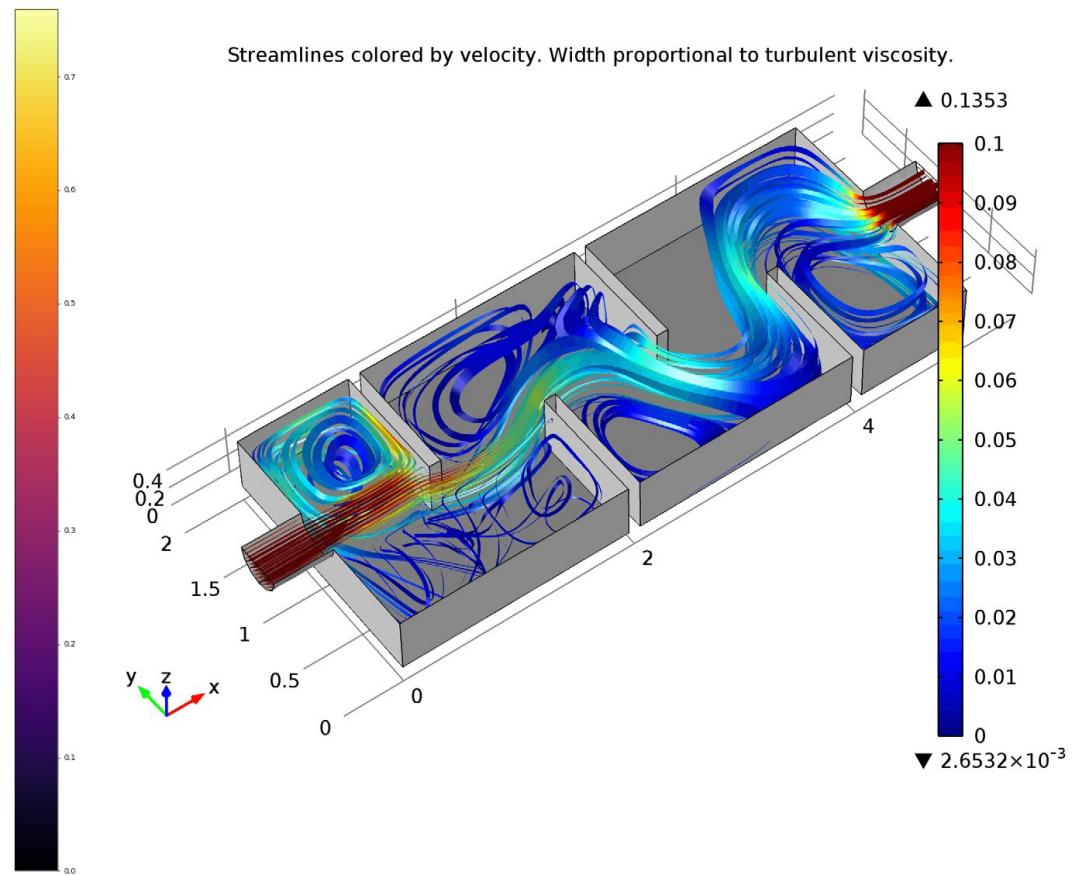
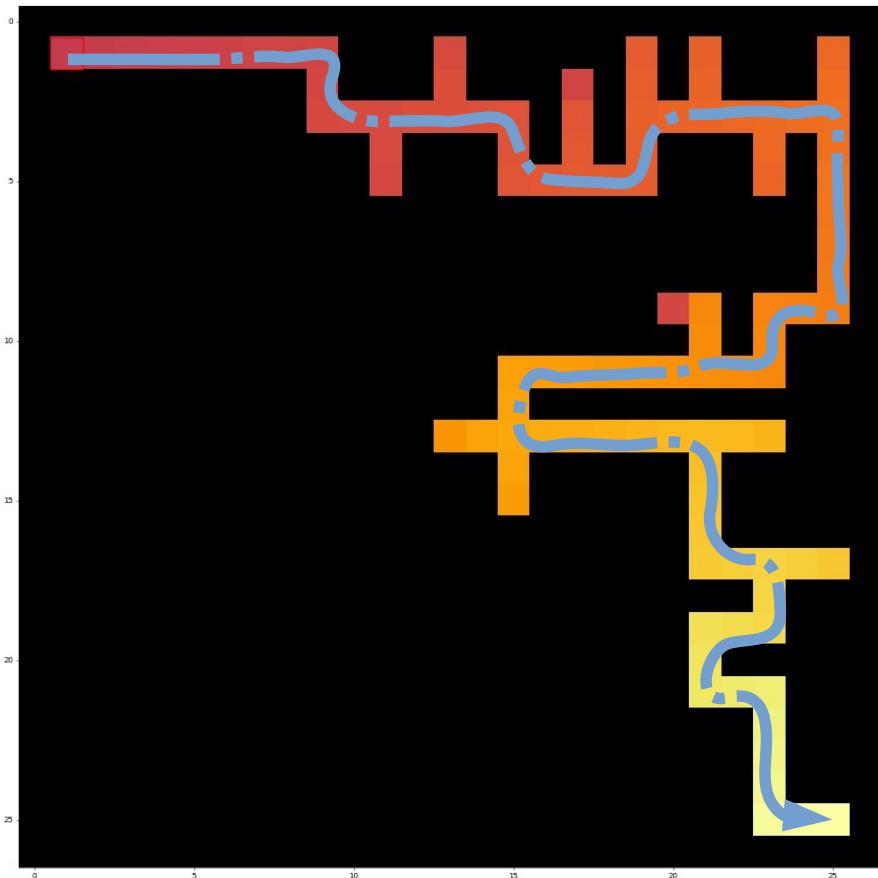
Idée de résolution



Idée de résolution



Idée de résolution



Equations de Navier Stokes

$$\operatorname{div} \vec{v} = \vec{0}$$

$$\rho(\vec{v} \cdot \vec{\nabla}) \cdot \vec{v} = -\vec{\nabla} p$$

Objectif

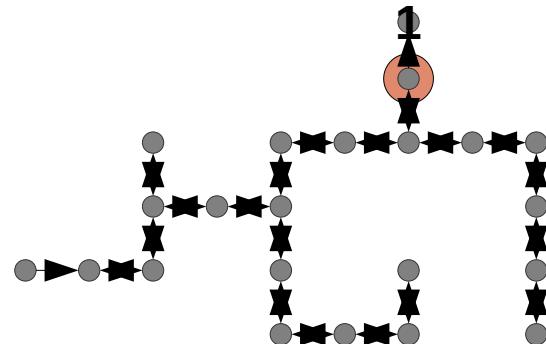
- Obtenir un réseau de neurones ne pouvant apprendre **que** des solutions de ces équations

Objectif

- Obtenir un réseau de neurones ne pouvant apprendre **que** des solutions de ces équations
 - Résoudre les équations sous forme de calcul sur un graphe (de manière modulaire)
 - En extraire un réseau de neurones

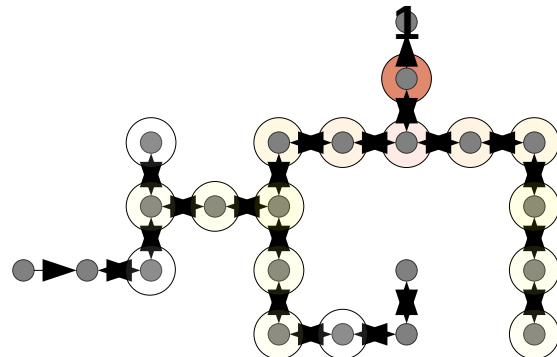
Objectif

- Obtenir un réseau de neurones ne pouvant apprendre **que** des solutions de ces équations
 - Résoudre les équations sous forme de calcul sur un graphe (de manière modulaire)
 - En extraire un réseau de neurones



Objectif

- Obtenir un réseau de neurones ne pouvant apprendre **que** des solutions de ces équations
 - Résoudre les équations sous forme de calcul sur un graphe
 - En extraire un réseau de neurones



Changer la forme d'apprentissage

$$\operatorname{div} \vec{v} = \vec{0}$$

$$\rho(\vec{v} \cdot \vec{\nabla}) \cdot \vec{v} = -\vec{\nabla} p$$

Changer la forme d'apprentissage

$$\operatorname{div} \vec{v} = \vec{0}$$
$$\rho(\vec{v} \cdot \vec{\nabla}) \cdot \vec{v} = -\vec{\nabla} p$$

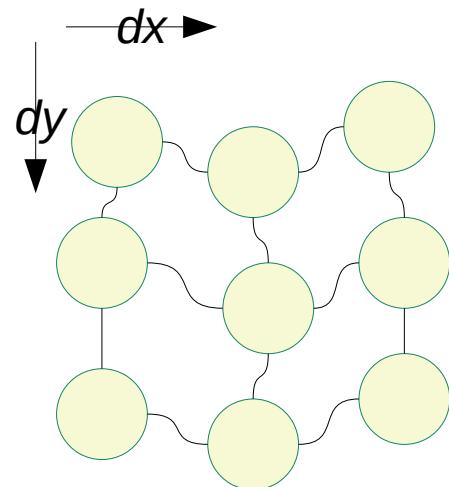

Réseau sous forme de plan
(tableau de neurones)
(tableau vivant)

Changer la forme d'apprentissage

$$\operatorname{div} \vec{v} = \vec{0}$$

$$\rho(\vec{v} \cdot \vec{\nabla}) \cdot \vec{v} = -\vec{\nabla} p$$

Réseau sous forme de plan
(tableau de neurones)
(tableau vivant)



Changer la forme d'apprentissage

$$\operatorname{div} \vec{v} = \vec{0}$$

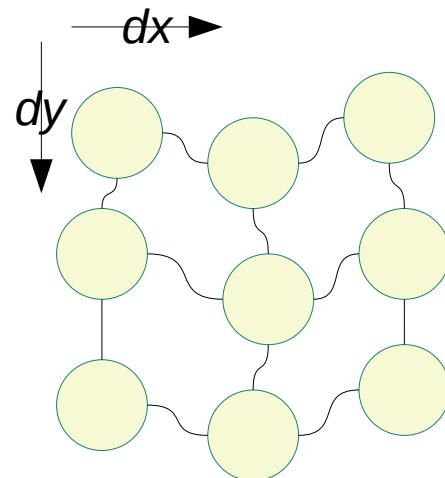
$$\rho(\vec{v} \cdot \vec{\nabla}) \cdot \vec{v} = -\vec{\nabla} p$$



Réseau sous forme de plan
(tableau de neurones)



Poids dans les neurones (forme
d'apprentissage originale)



Équations de pseudo-Navier-Stokes

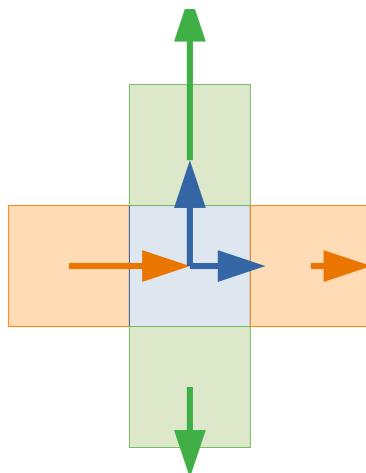
$$\operatorname{div}(\vec{v}) = \begin{cases} \alpha & \text{à l'entrée} \\ -\alpha & \text{à la sortie} \\ 0 & \text{sur le reste des cases} \end{cases}$$

$$\vec{v} = \overrightarrow{ave} \vec{v} + \lambda \overrightarrow{v_{learn}} + \overrightarrow{dev}$$

Équations de pseudo-Navier-Stokes

$$\operatorname{div}(\vec{v}) = \begin{cases} \alpha & \text{à l'entrée} \\ -\alpha & \text{à la sortie} \\ 0 & \text{sur le reste des cases} \end{cases}$$

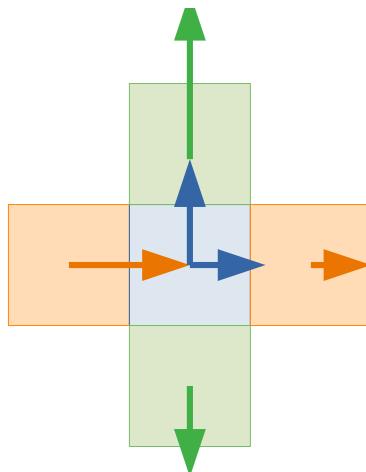
$$\vec{v} = \overrightarrow{ave} \vec{v} + \lambda \overrightarrow{v_{learn}} + \overrightarrow{dev}$$



Équations de pseudo-Navier-Stokes

$$\operatorname{div}(\vec{v}) = \begin{cases} \alpha & \text{à l'entrée} \\ -\alpha & \text{à la sortie} \\ 0 & \text{sur le reste des cases} \end{cases}$$

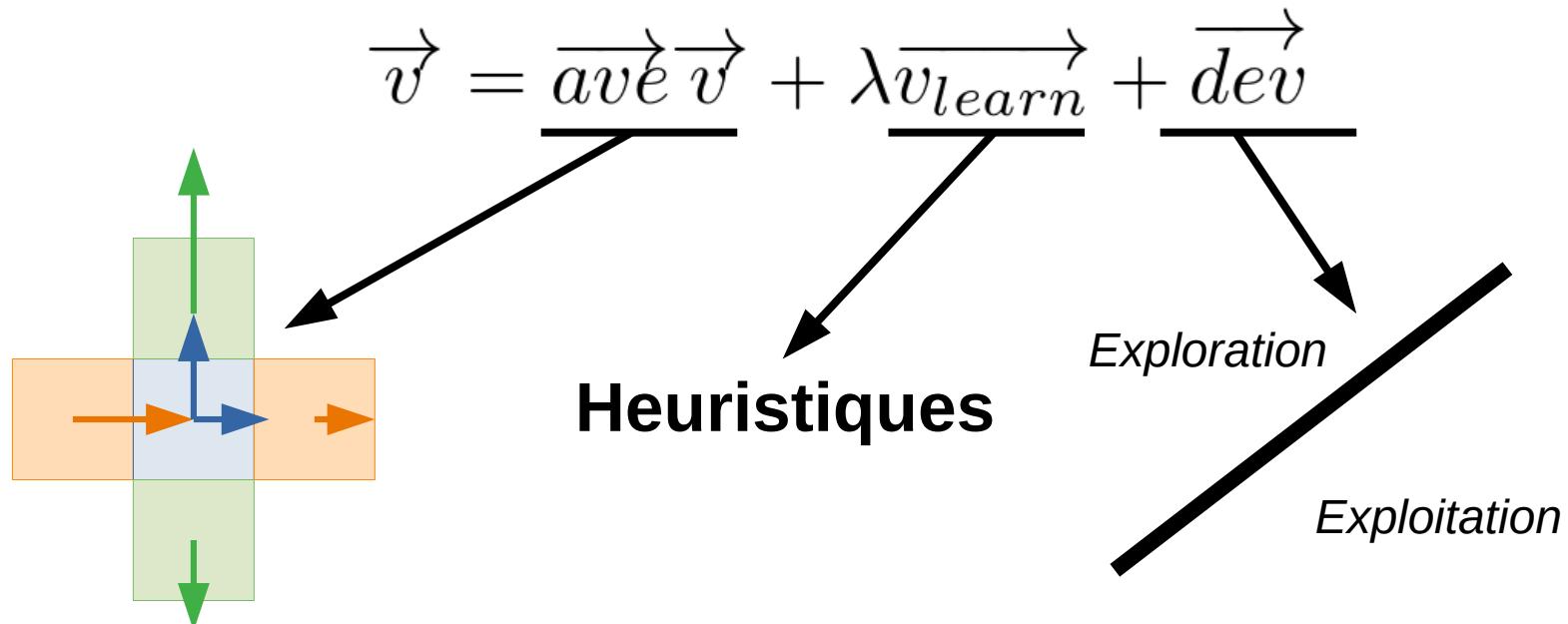
$$\vec{v} = \overrightarrow{ave} \vec{v} + \lambda \overrightarrow{v_{learn}} + \overrightarrow{dev}$$



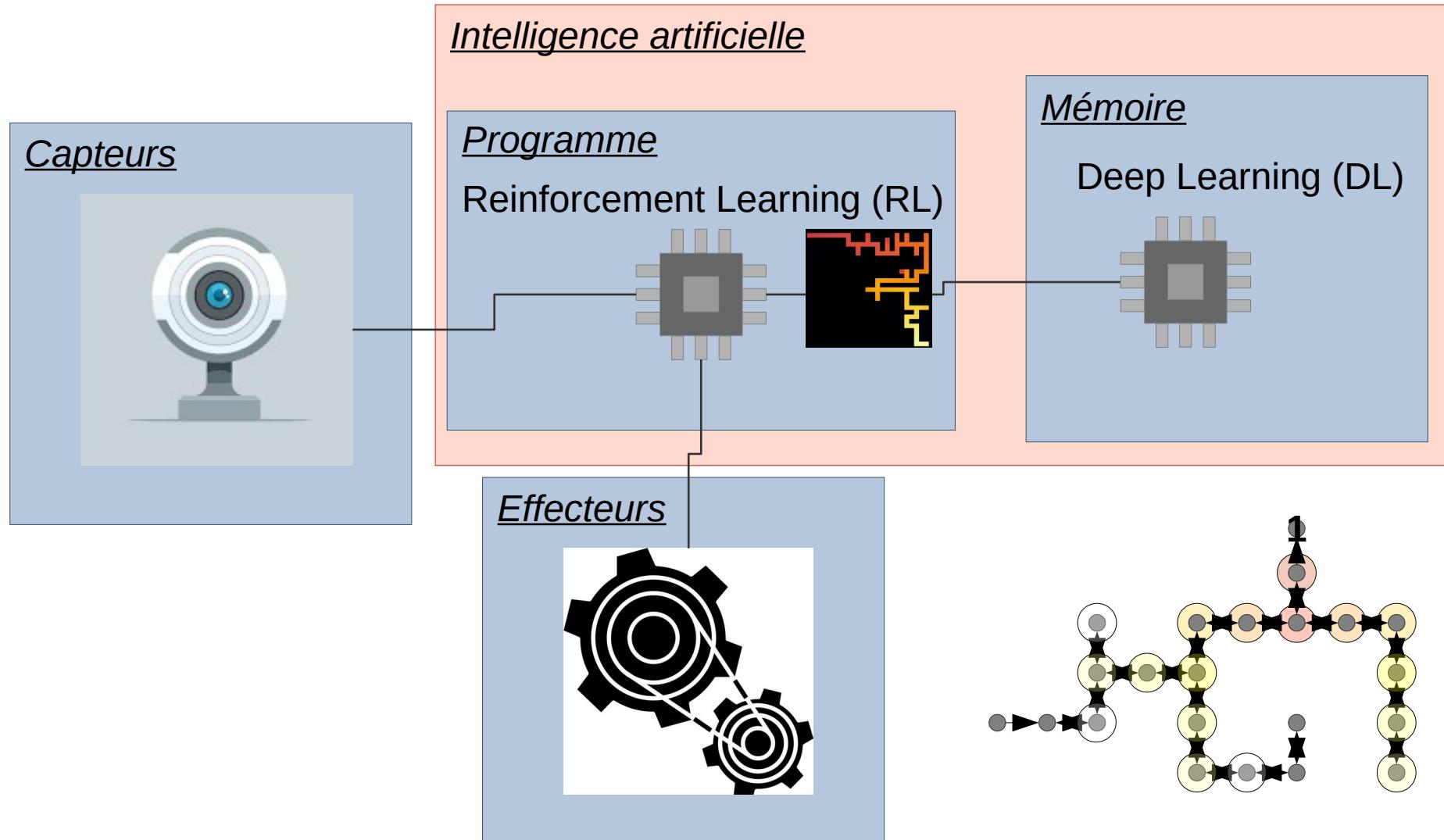
Heuristiques

Équations de pseudo-Navier-Stokes

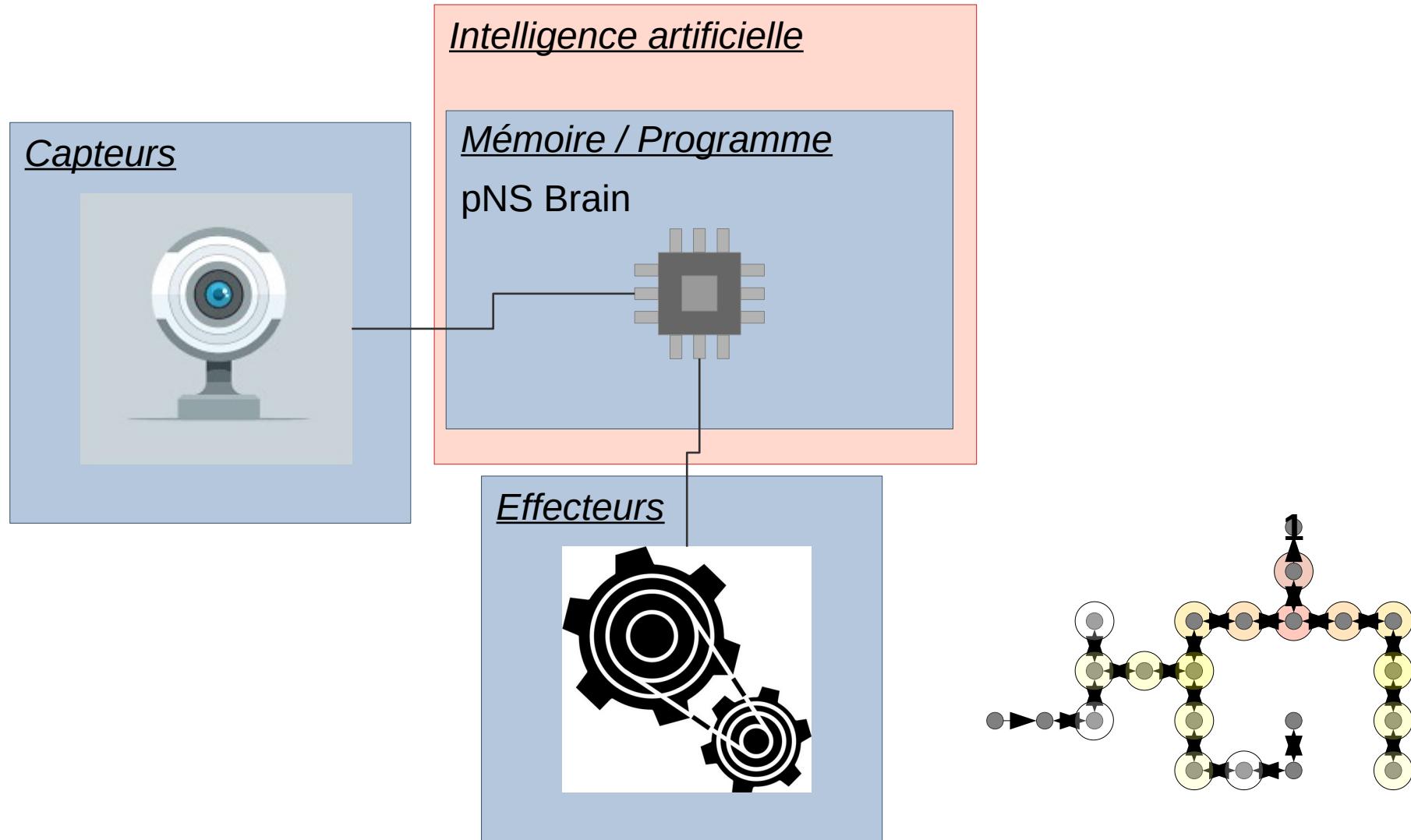
$$\operatorname{div}(\vec{v}) = \begin{cases} \alpha & \text{à l'entrée} \\ -\alpha & \text{à la sortie} \\ 0 & \text{sur le reste des cases} \end{cases}$$



Architecture du Machine Learning

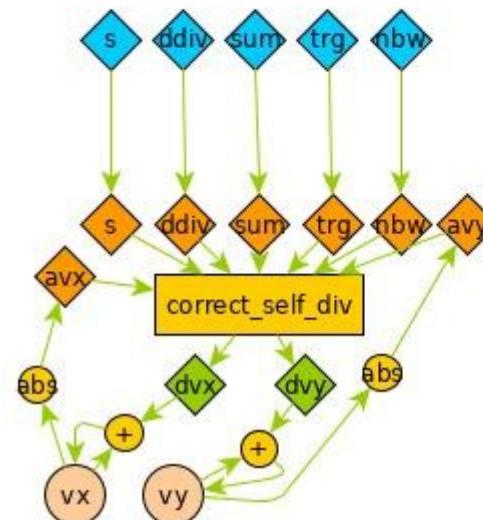


Architecture du Machine Learning

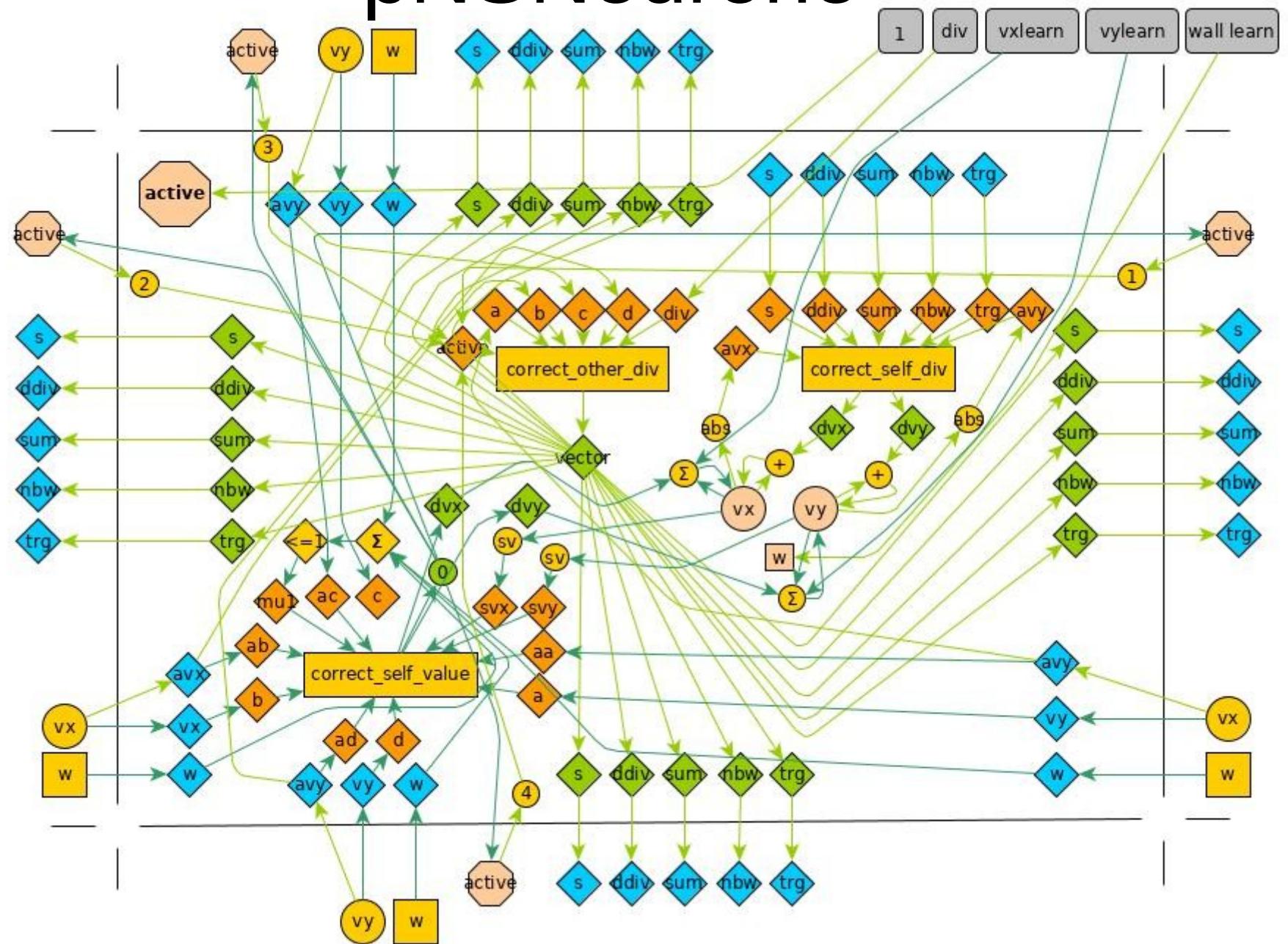


pNSNeurone

1 div vxlearn vylearn wall learn



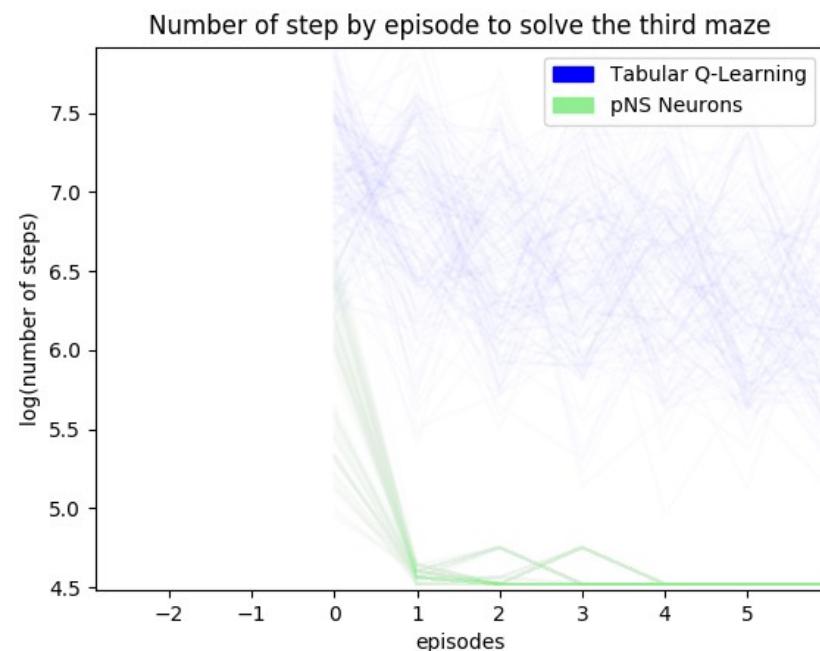
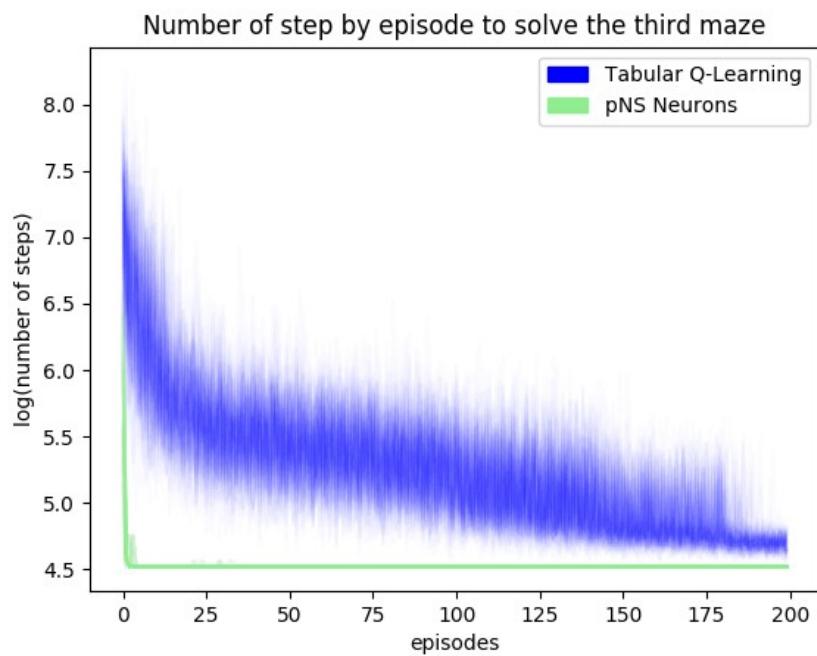
pNSNeurone



Démo

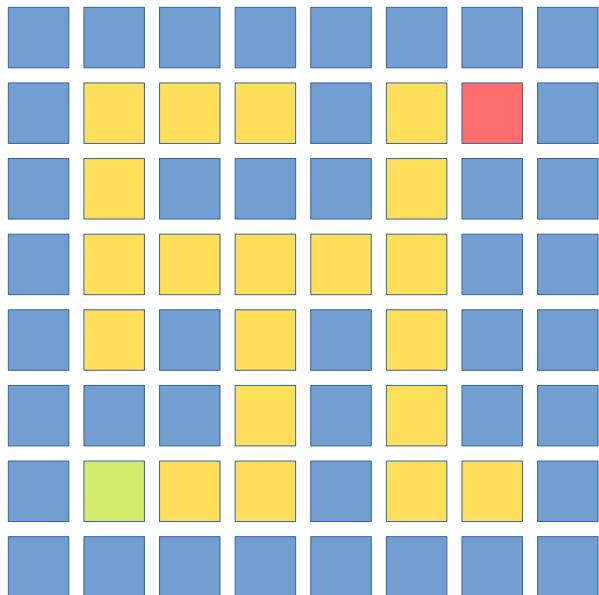
<https://github.com/Yaaxx/maze-resolution-with-pNS-neurons>

Résultats



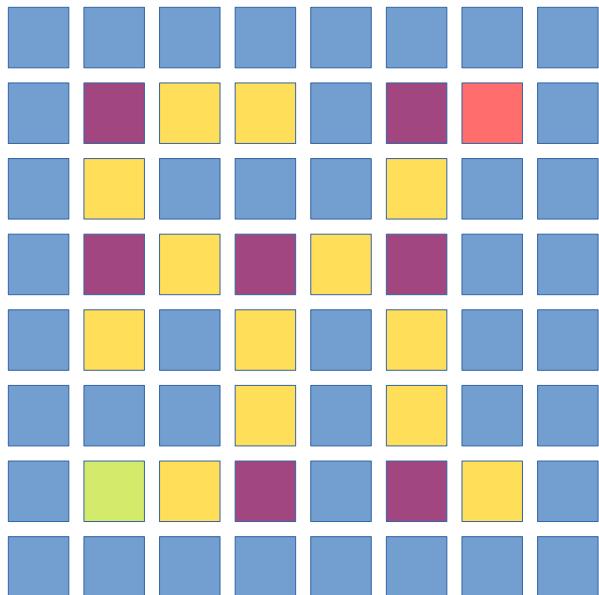
Faire mieux en espace

pNSNetwork



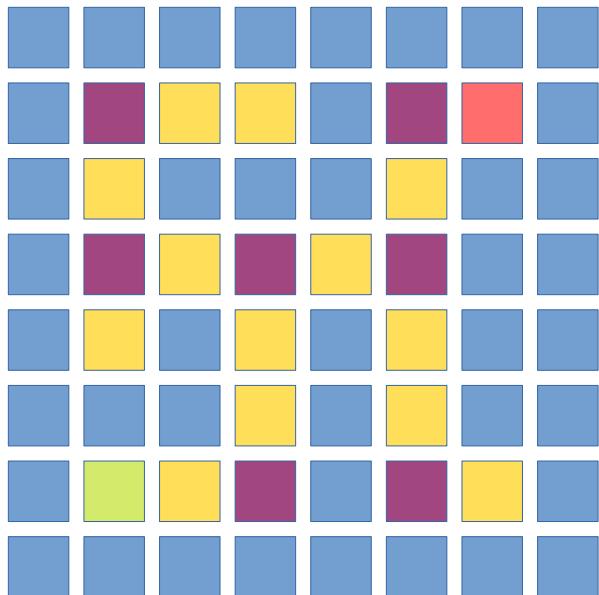
Faire mieux en espace

pNSNetwork



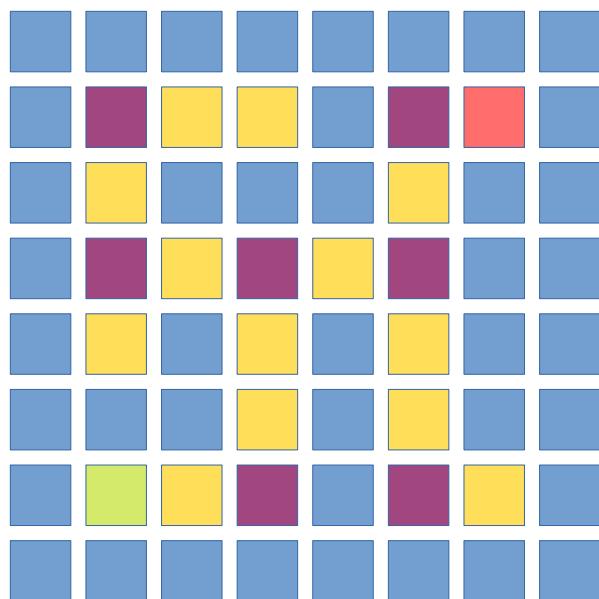
Faire mieux en espace

pNSNetwork

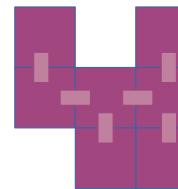


Faire mieux en espace

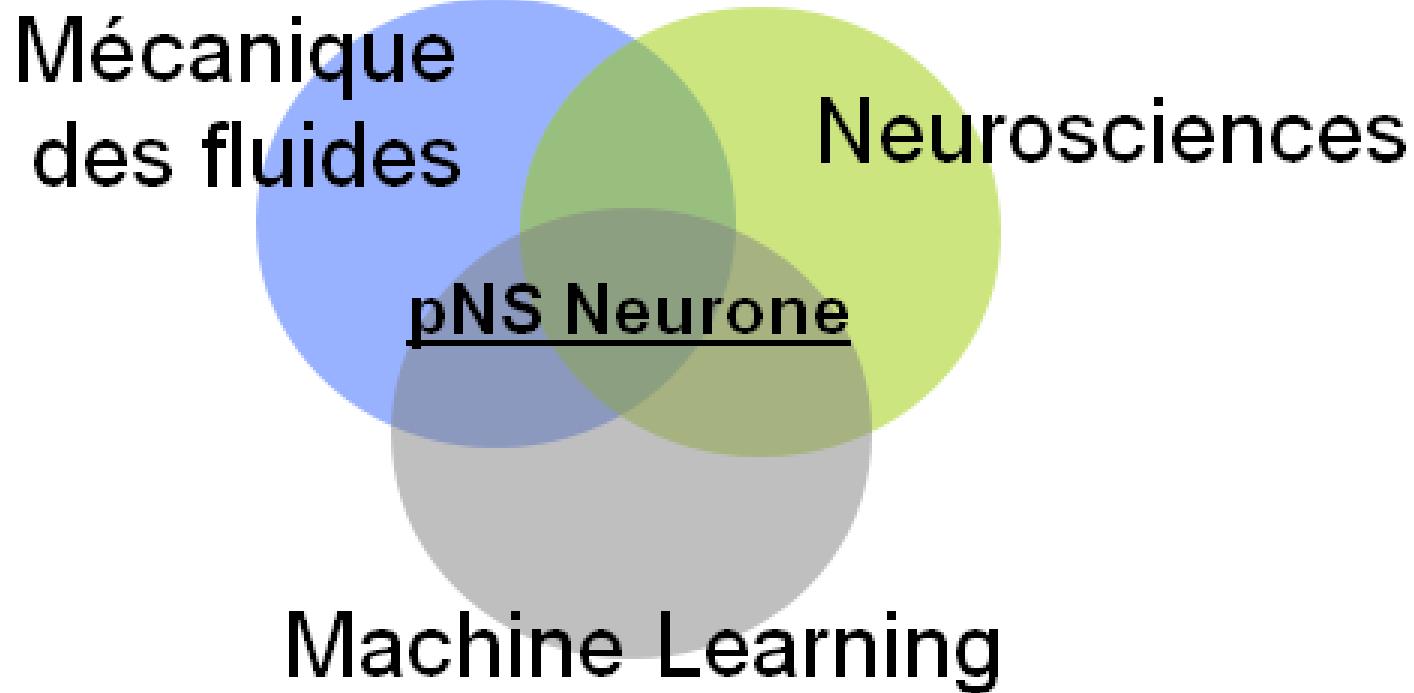
pNSNetwork



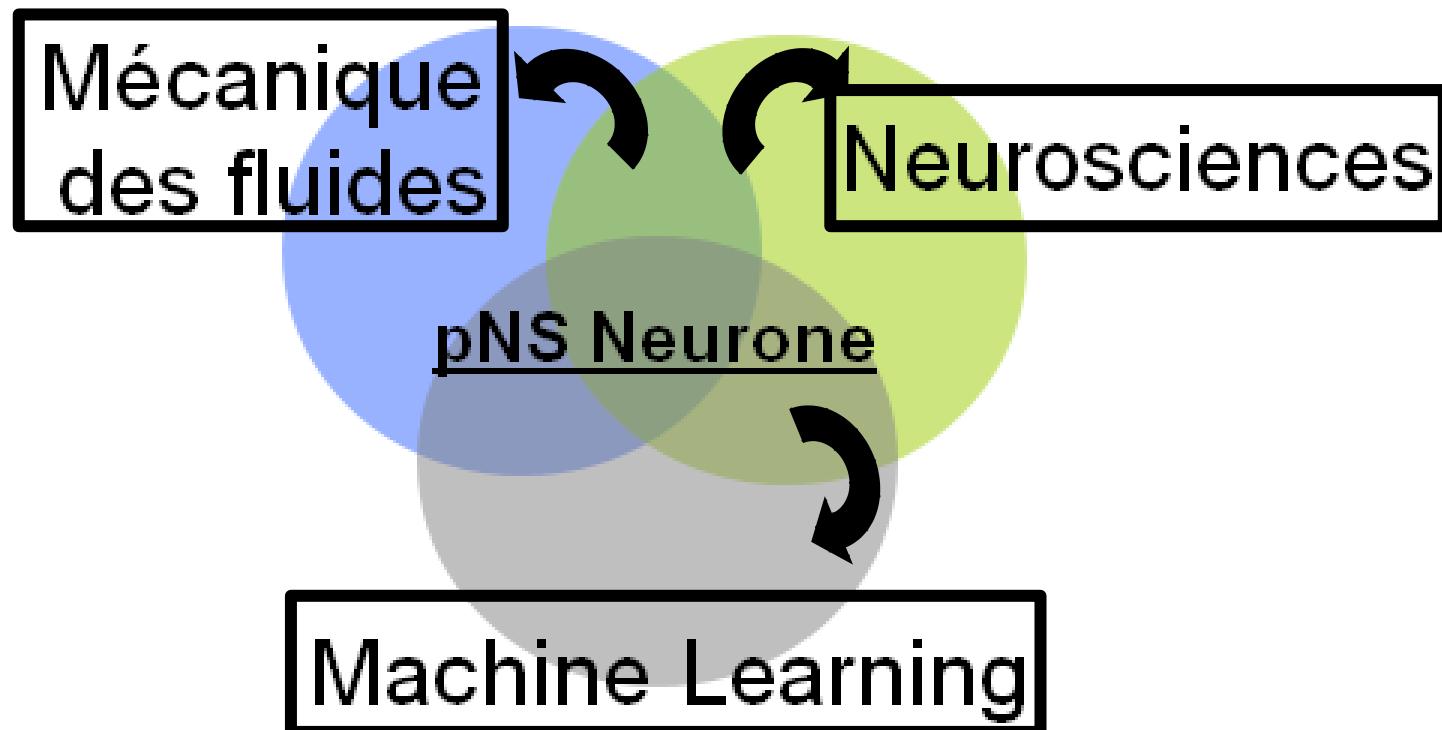
pNSNetwork++



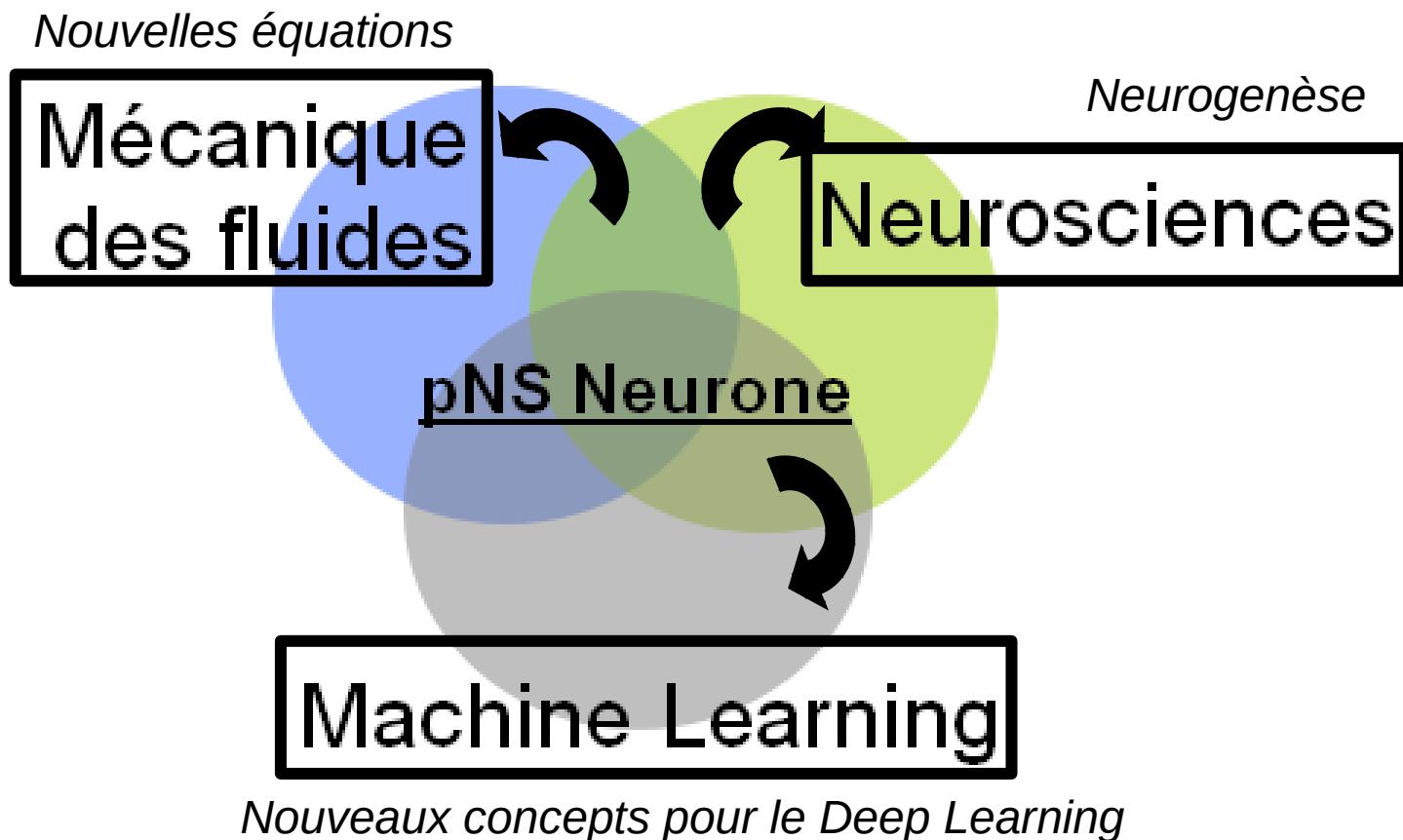
pNSNeurone



pNSNeurone



pNSNeurone



Pour aller plus loin

- Résoudre les labyrinthes à plusieurs solutions
- Faire des cerveaux mixtes
- Développer le secteur de recherche sur les neurones à organites

Merci

Des questions ?