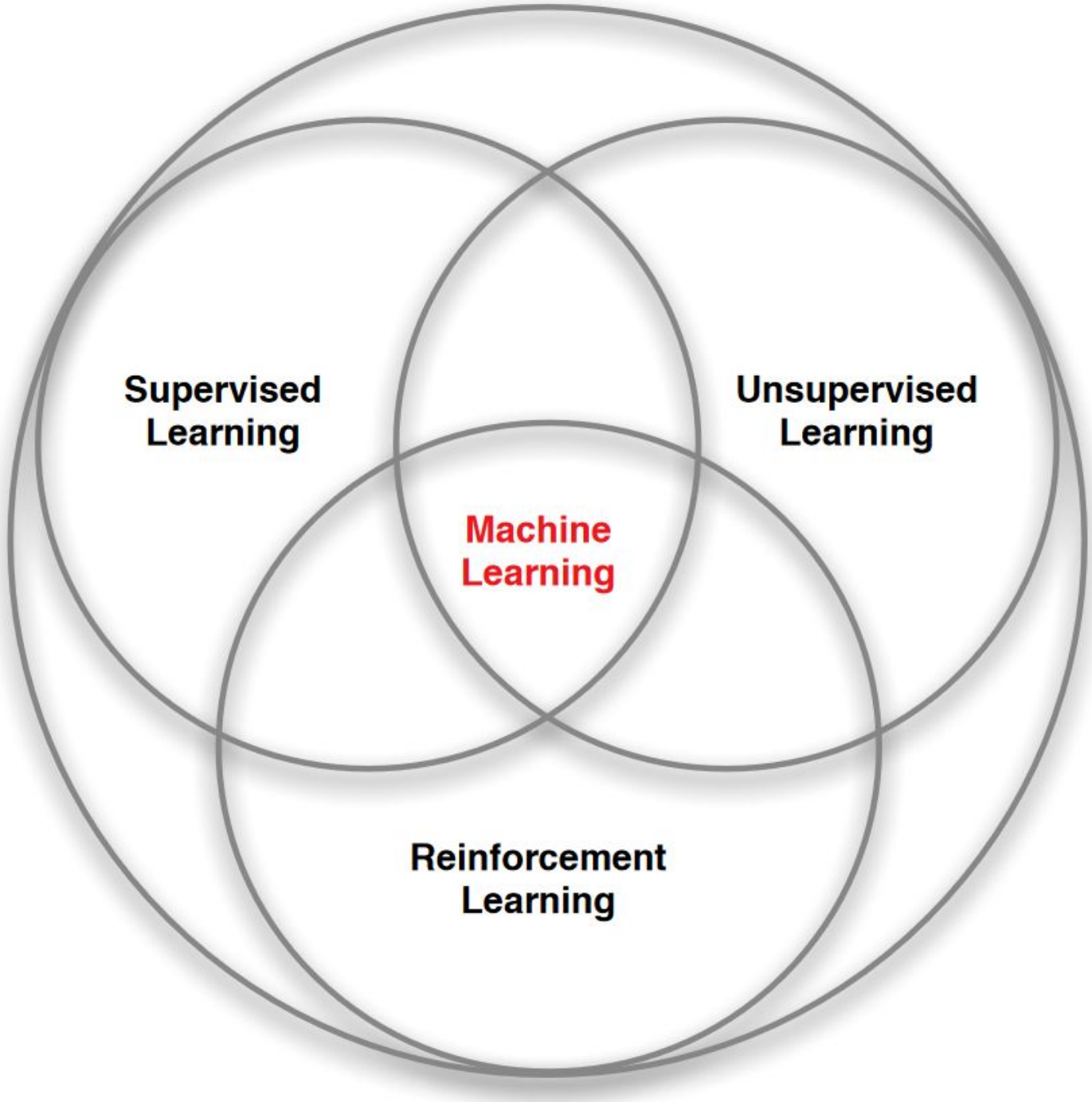
Intro to Reinforcement Learning

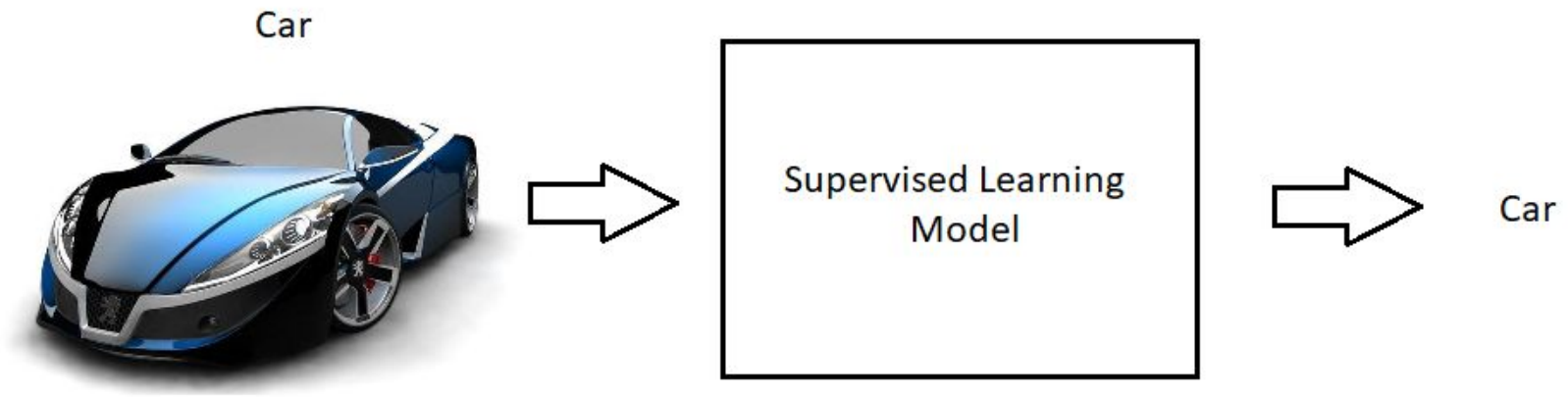
Roman Kopšo, AMPLYFI

PyData Meetup, Cardiff 11th December 2018


Supervised
Learning
Unsupervised
Learning
Machine
Learning
Reinforcement
Learning

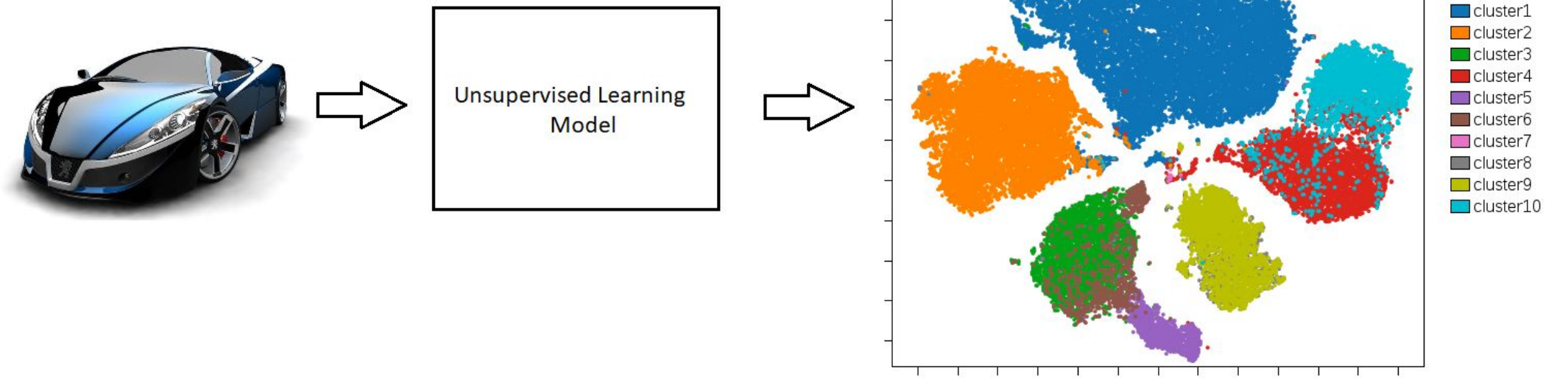
Supervised Learning

- Data comes with labels
- We give immediate feedback to model
- Classification purpose
- Used in: **Image tagging**, **Patient diagnosis**, **Spam filtering**



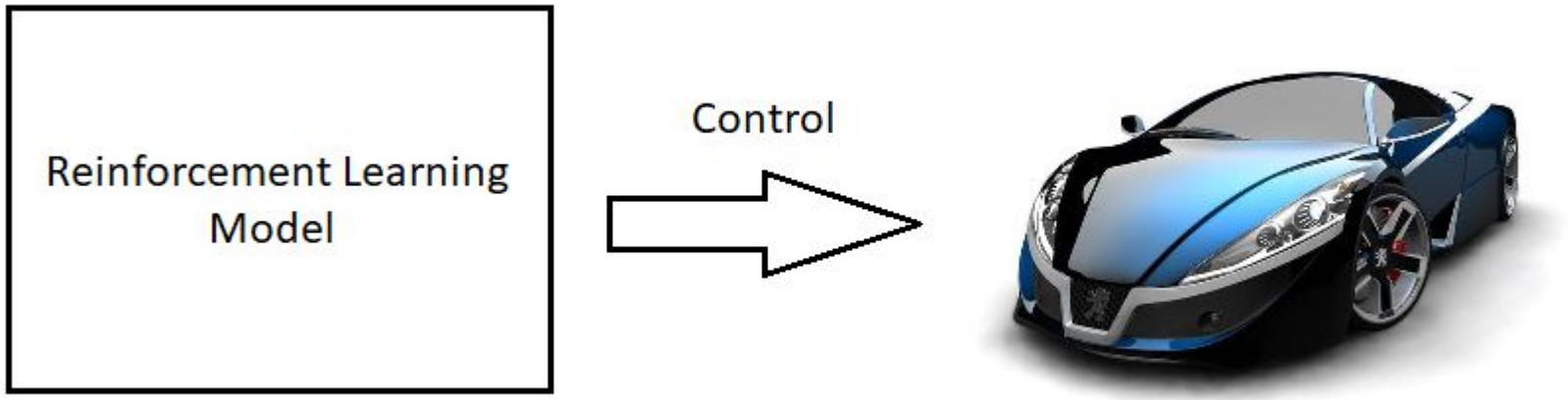
Unsupervised Learning

- Data has no labels
- We give no feedback to model
- Clustering purpose
- Used in: **Customer segmentation**, **Sentiment analysis**, **Recommender systems**



Reinforcement Learning

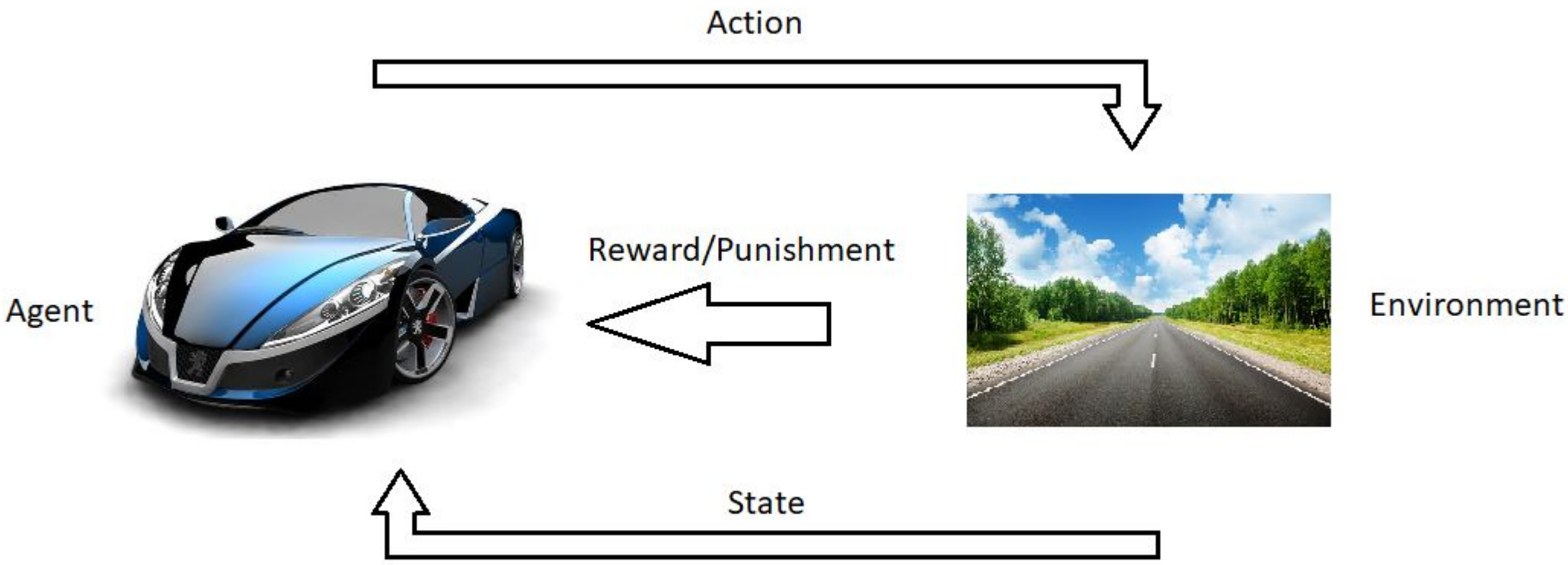
- Data is evaluated using reward signals
- Feedback can be delayed
- Automatization purpose
- Used in: **Game AI**, **Real-time decisions**, **Robot control**



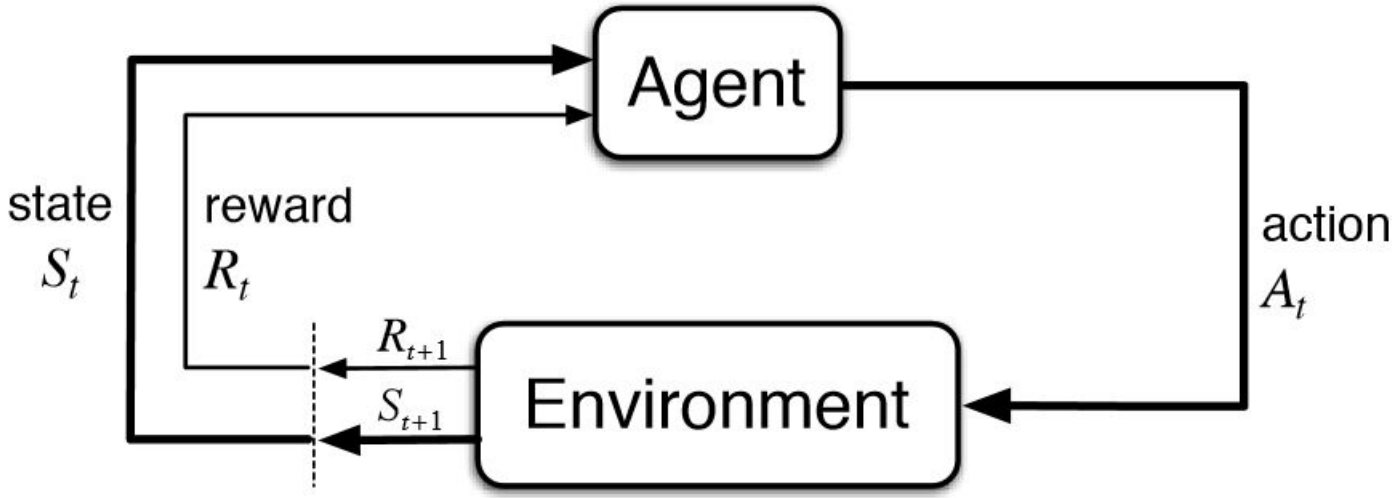
Keywords in RL

- Agent
  - Car
- Action
  - Start the engine, turn wheels, brake
- State
  - Speed of the car, location
- Reward
  - Keep the car on the road, not crashing
- Environment
  - Highway, city streets

Visualization of RL Keywords



Reinforcement Learning Setup



Major Components of an RL Agent

- Policy: agent's behaviour function
- Value function: how good is each state and/or action

Policy

- A policy is the agent's behaviour
- It is a map from state to action

$$a = \pi(s)$$

Value function

- Value function is a prediction of future reward
- Used to evaluate the goodness/badness of states
- And therefore to select between actions

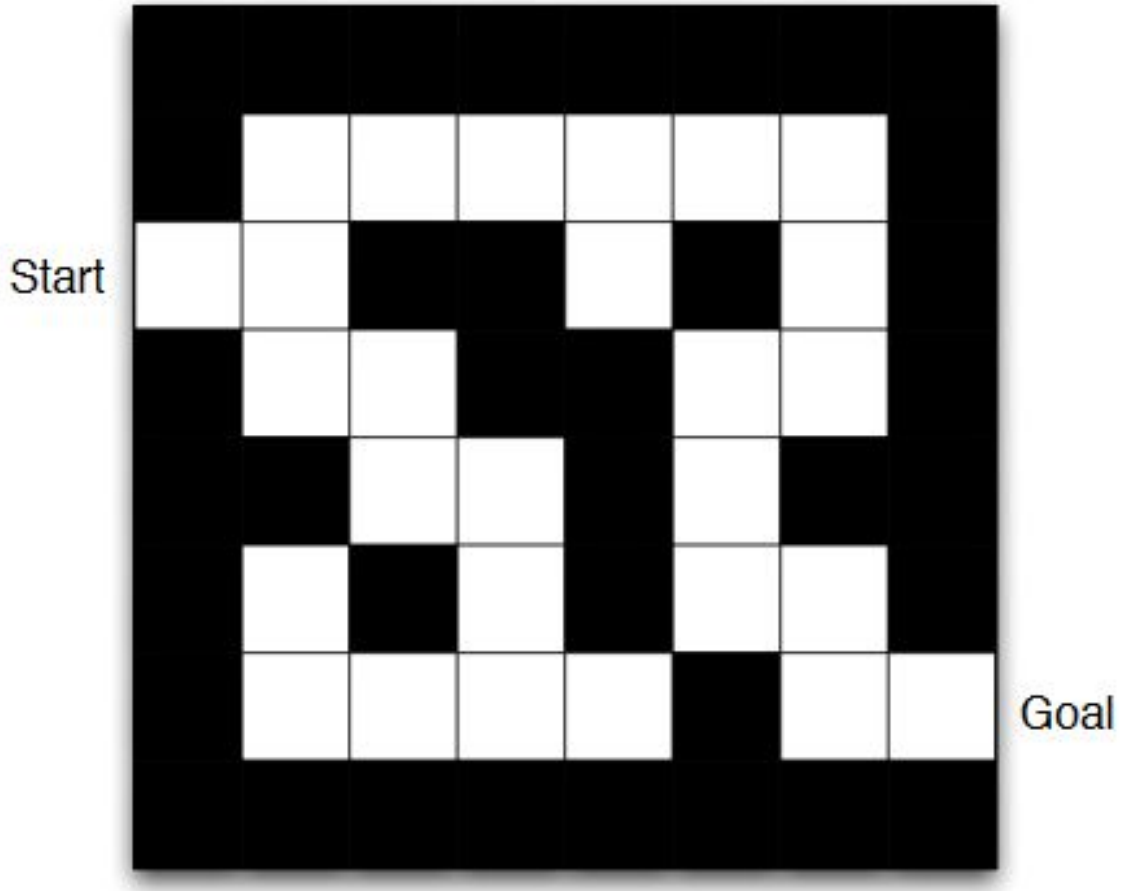
$$v_\pi(s) = \mathbb{E}_\pi \left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s \right]$$

Maze example

Rewards: -1 per timestep

Actions: N, S, W, E

States: Agent's location

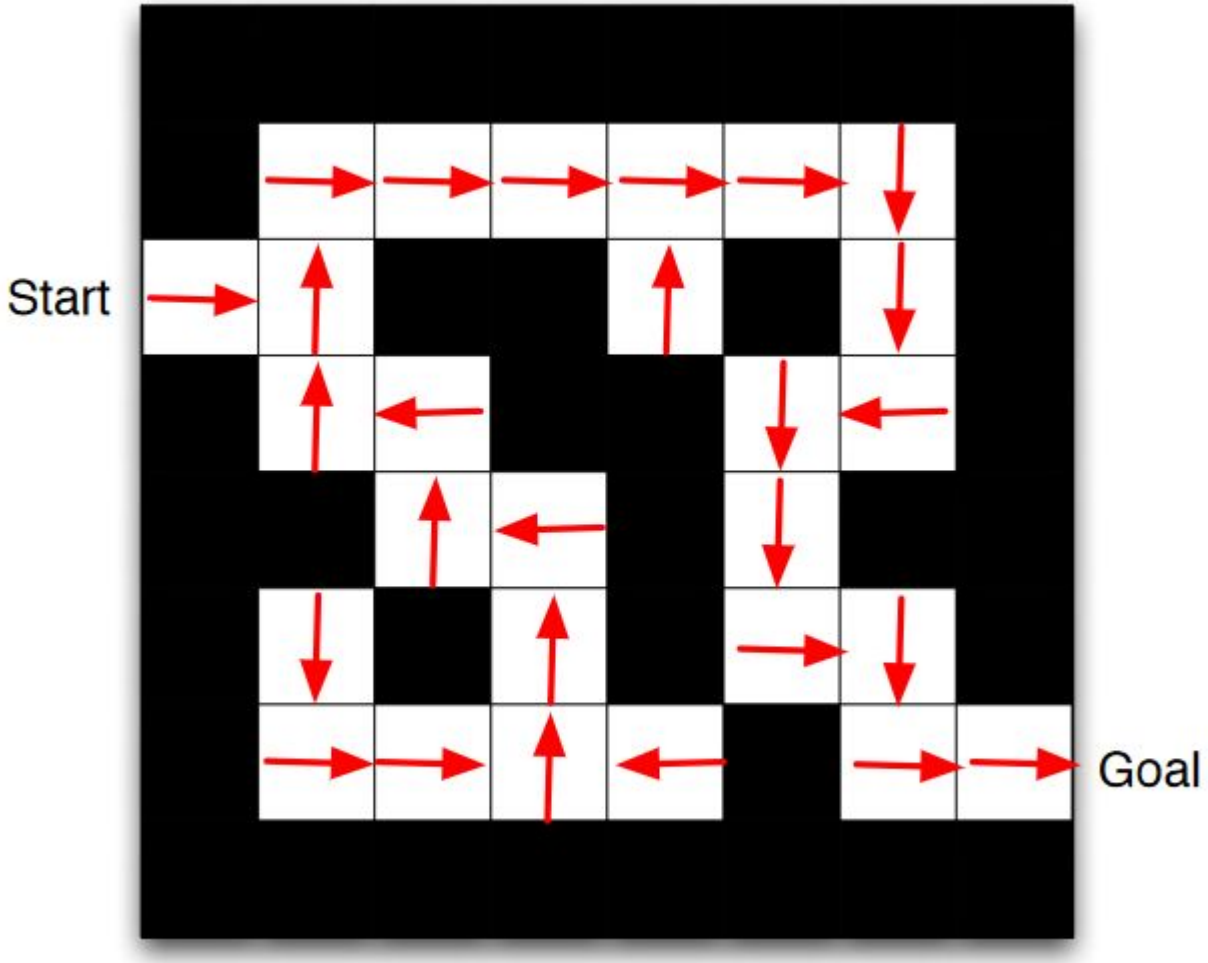


Maze example: Policy

Rewards: -1 per timestep

Actions: N, S, W, E

States: Agent's location

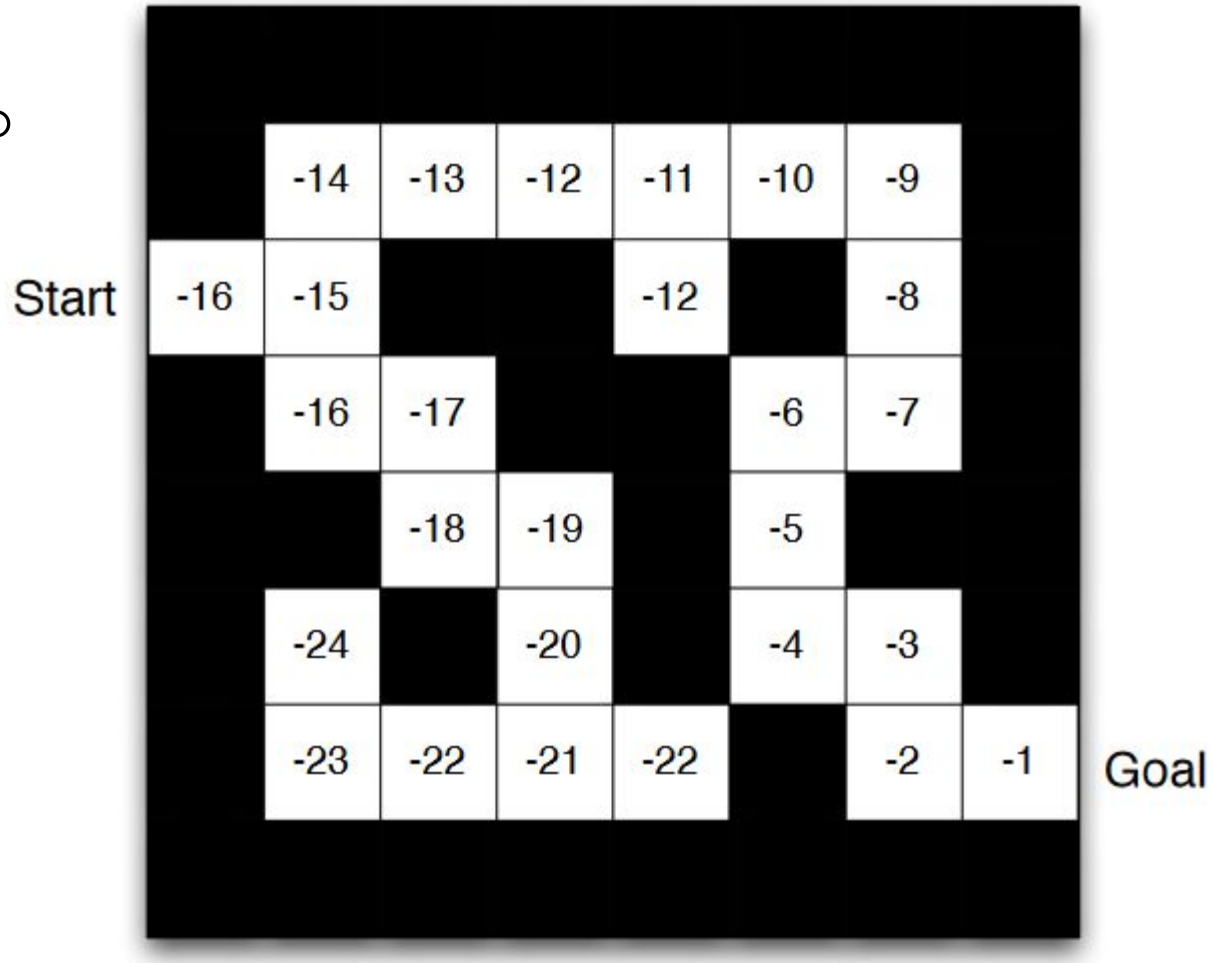


Maze example: Value function

Rewards: -1 per timestep

Actions: N, S, W, E

States: Agent's location



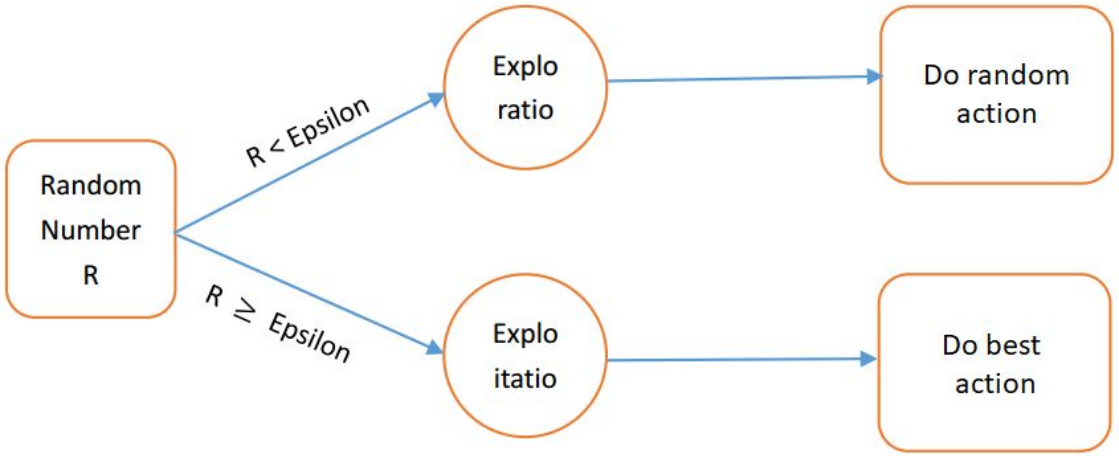
Q-learning

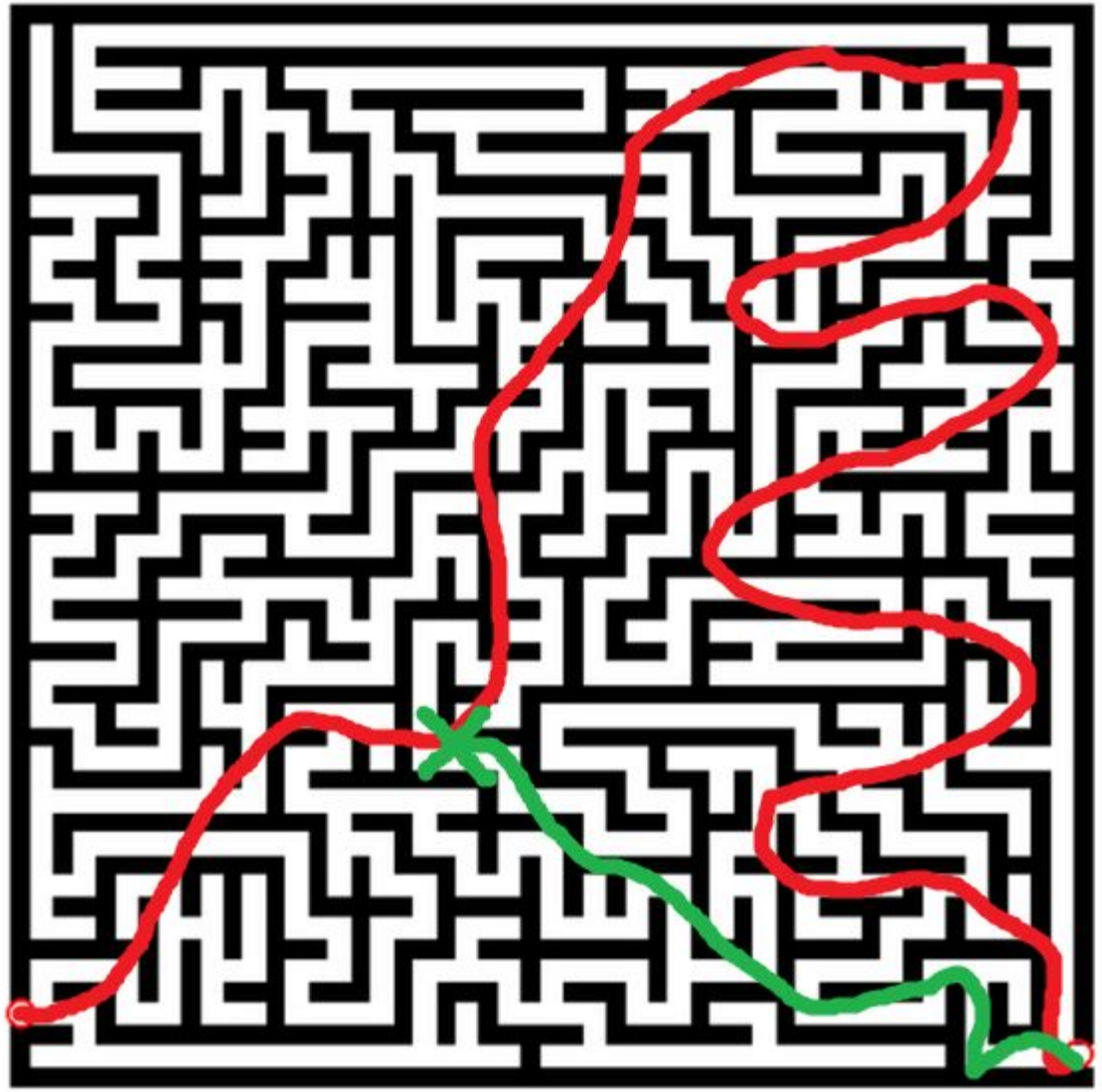
- Extension of Value function
- Takes into account values of specific actions in specific states
- Uses also discount factor gamma (0, 1)

$$Q^{\pi}(s_t, a_t) = E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... | s_t, a_t]$$

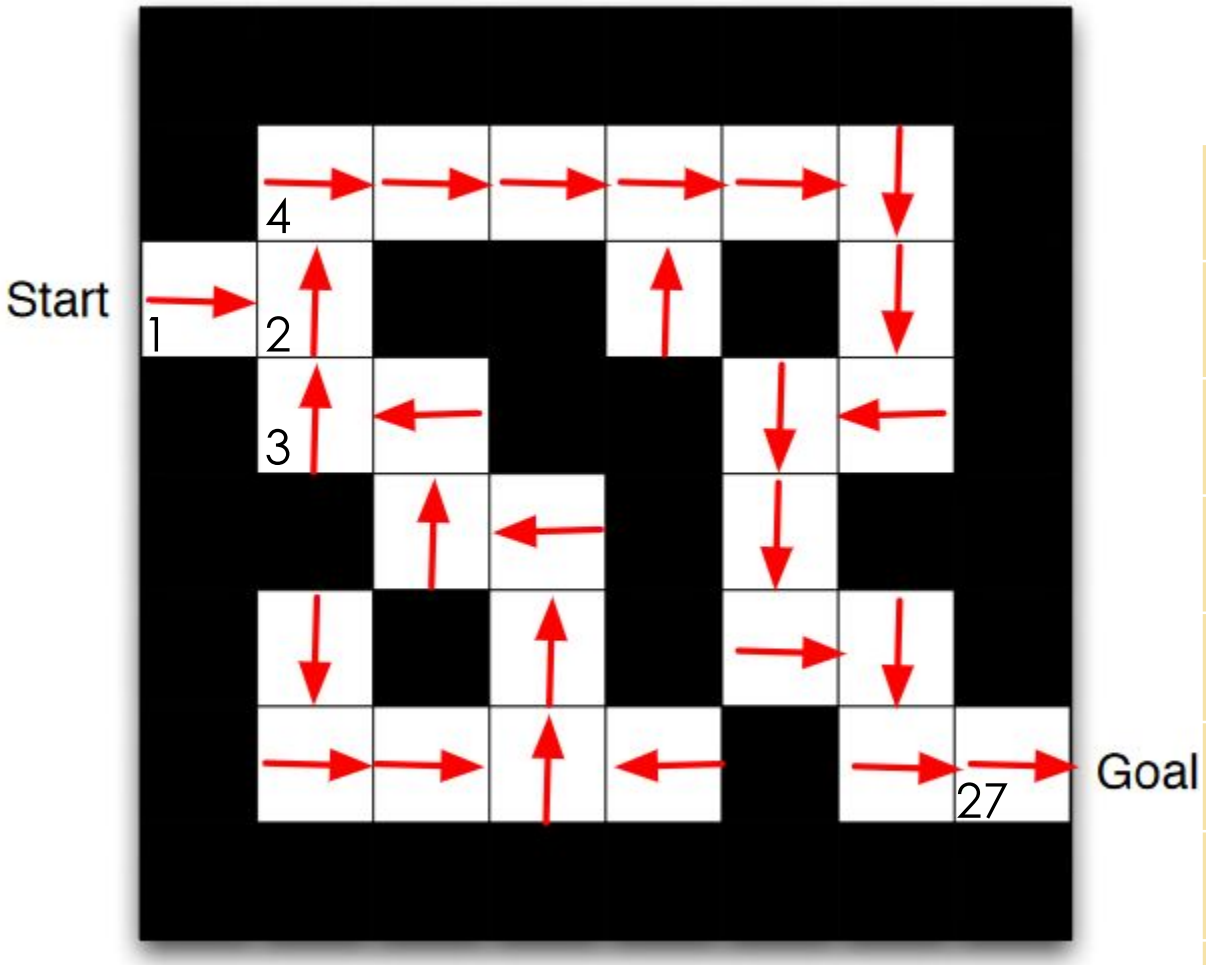
Exploration vs exploitation

Epsilon-greedy algorithm

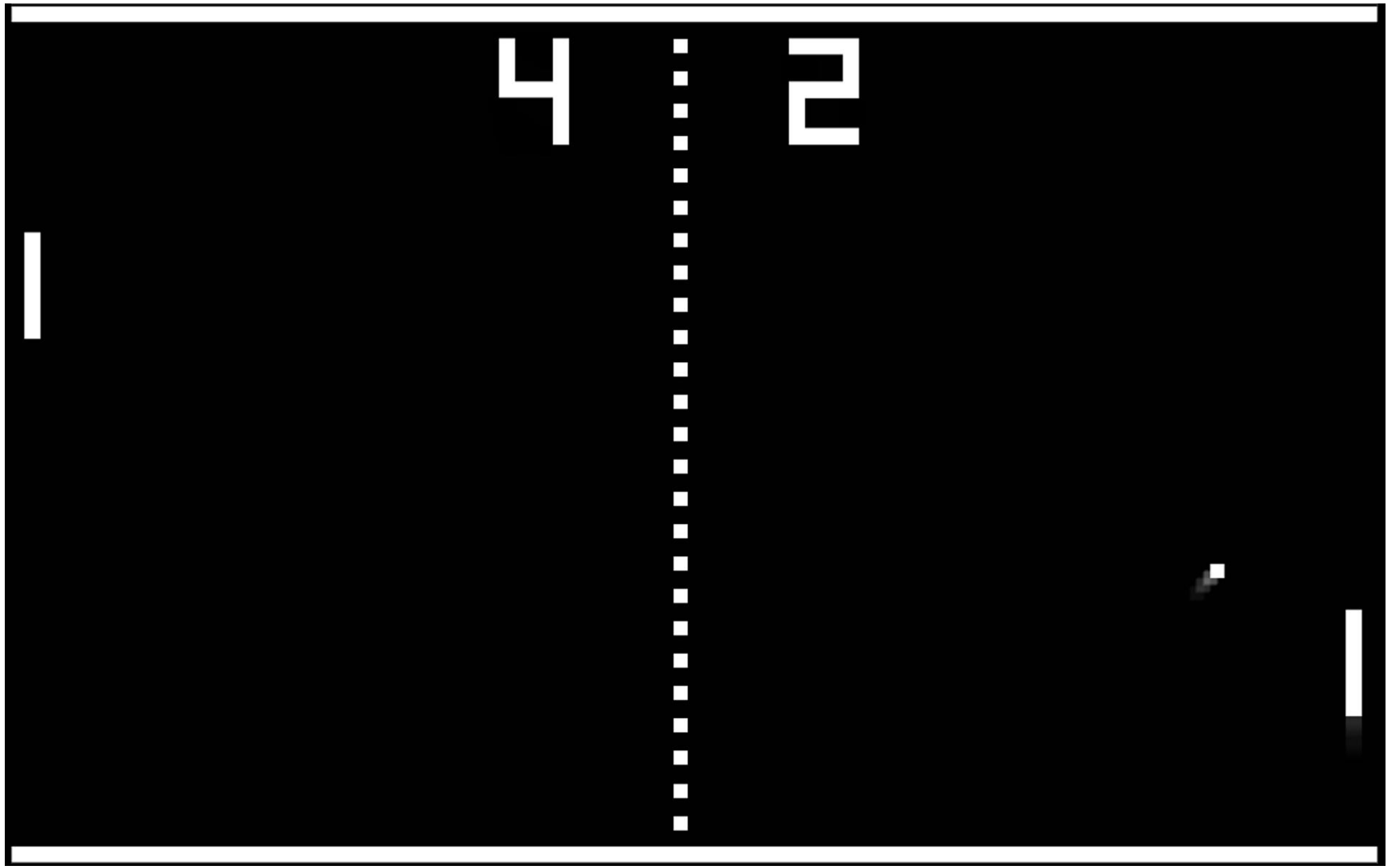




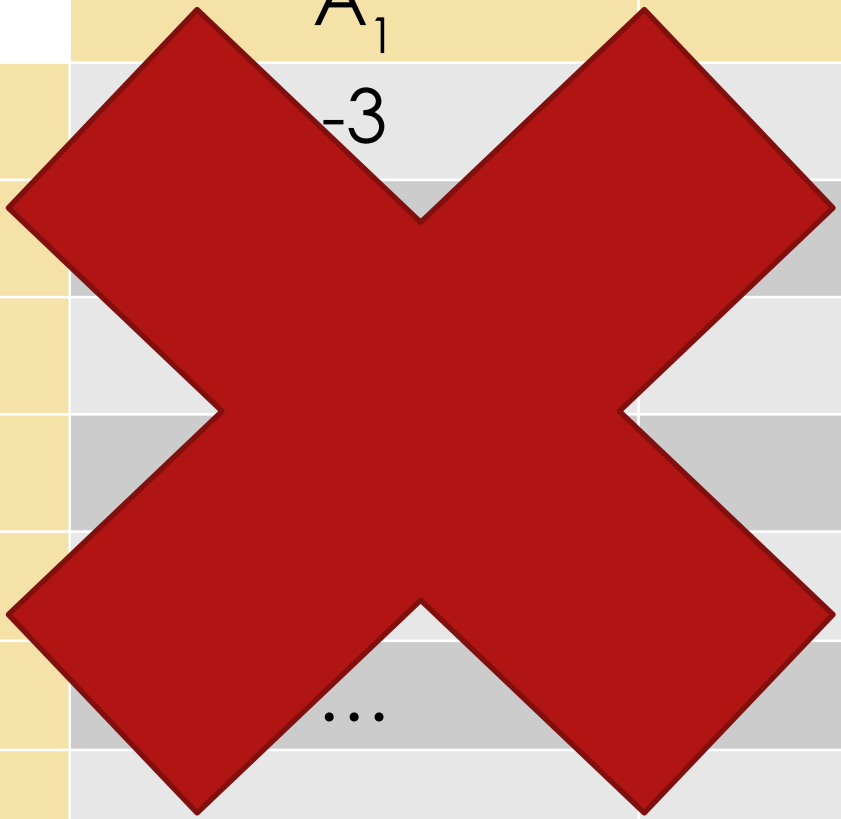
Saving optimal policy



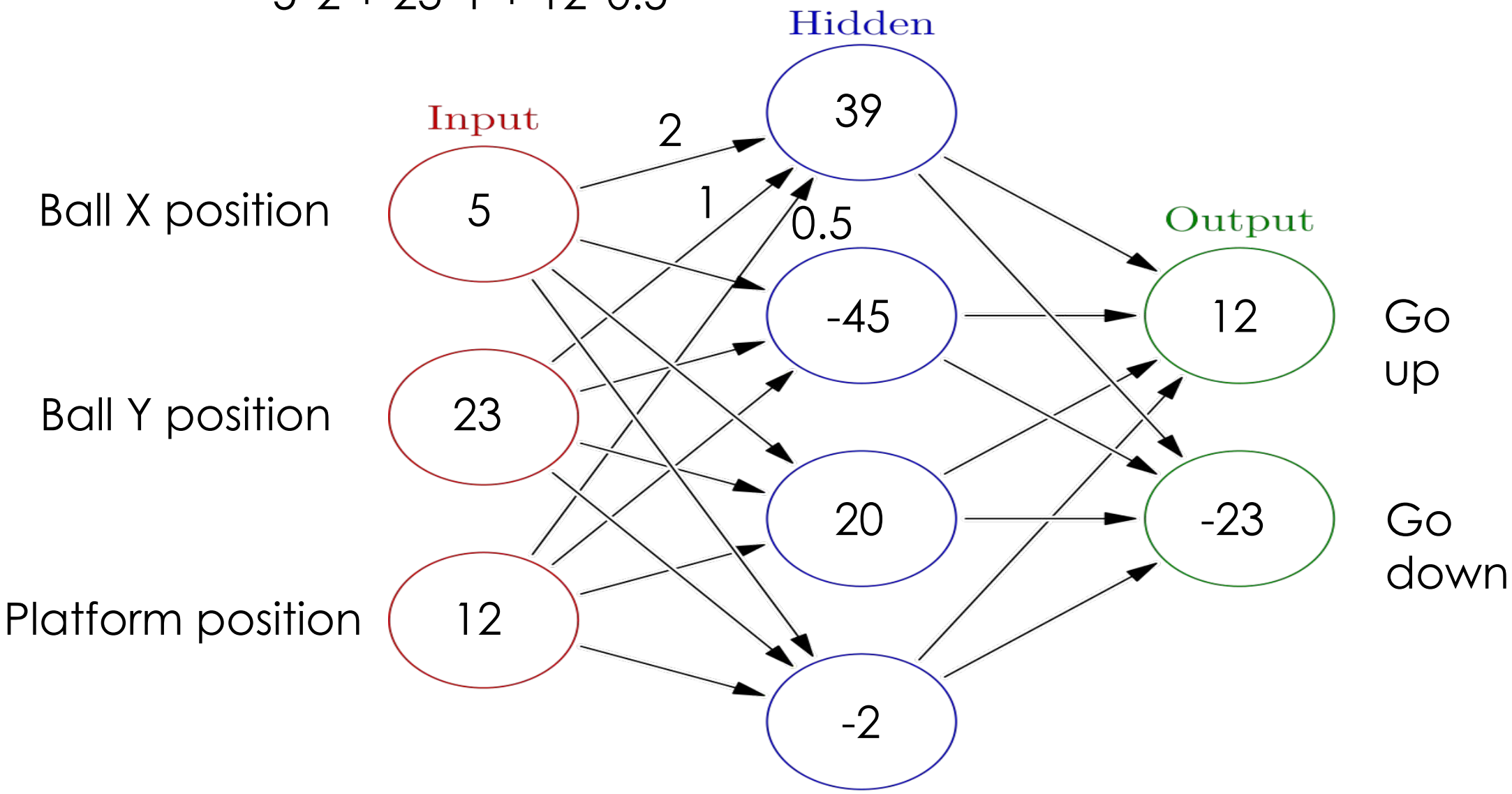
	A_1 N	A_2 S	A_3 W	A_4 E
S_1	/	/	/	-16
S_2	-15	-17	-17	/
S_3	-16	/	/	-18
S_4	/	-16	/	-14
….	…	…	…	…
….	…	…	…	…
….	…	…	…	…
S_{27}	/	/	-3	-1


4
2

	A_1	A_2
S_1	-3	2
S_2		0
S_3		4
S_4		5
….		…
….	…	…
….	…	…
S_N	8	-3

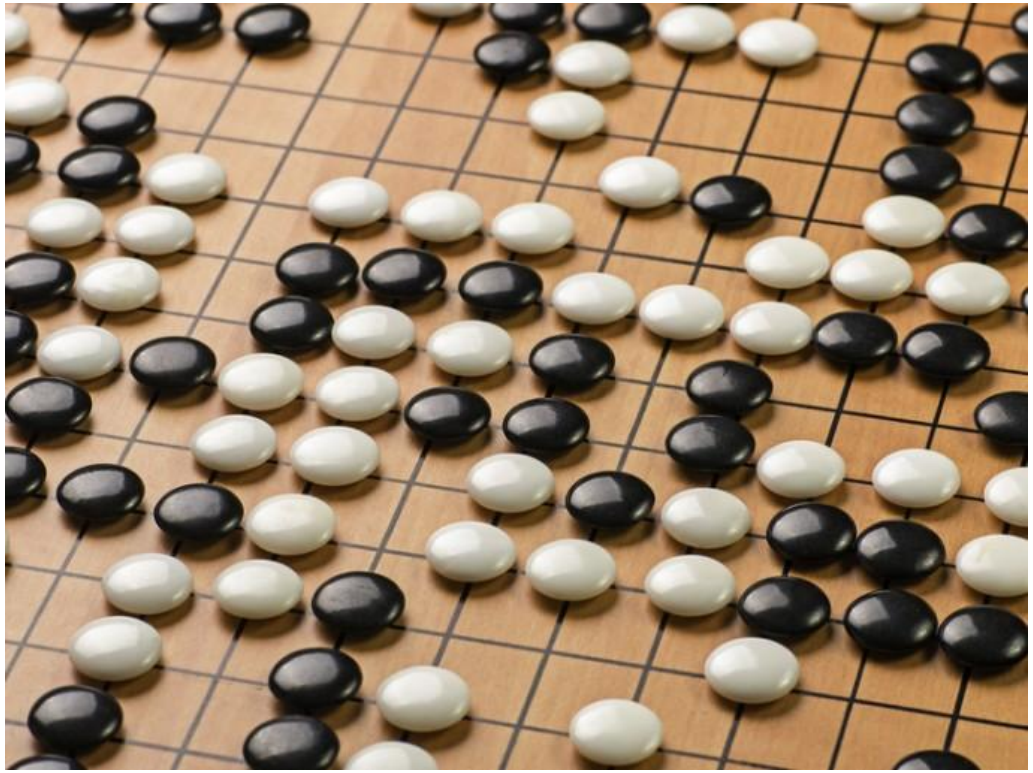


5*2 + 23*1 + 12*0.5


Hidden
Input
Output
2
1
0.5
39
-45
20
-2
5
23
12
12
-23
Ball X position
Ball Y position
Platform position
Go up
Go down

DeepMind

- British AI company
- Founded in September 2010
- Acquired by Google in 2014
- Mostly known for AlphaGo



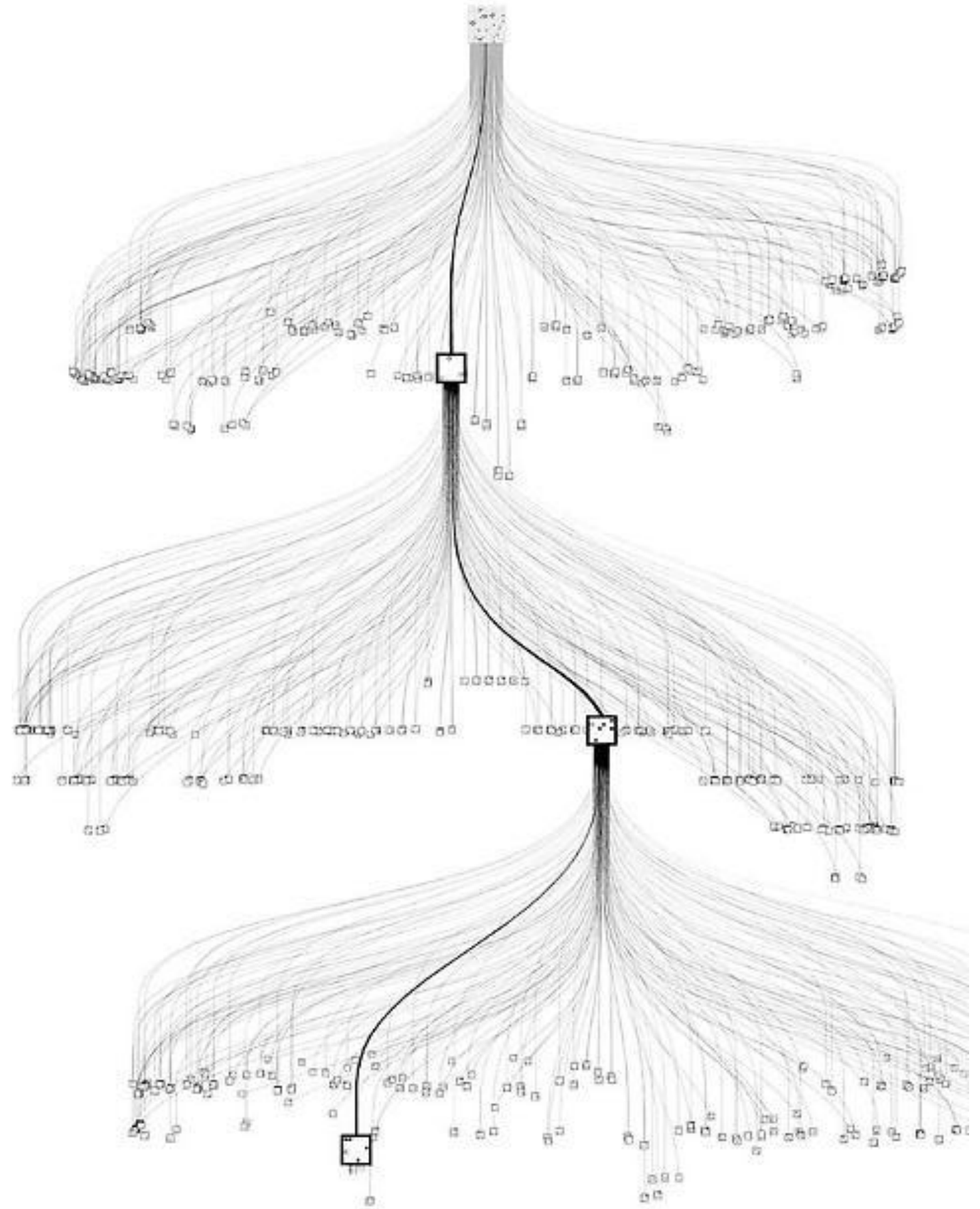
AlphaGo

- Program that uses deep reinforcement learning
- Applied in the game of Go
- First, supervised learning is used
- Next, reinforcement learning is used
- Game is played thanks to Policy and Value network
- Defeated Lee Sedol 1 to 4 in a 5-game match

Victory of AI in game of Go

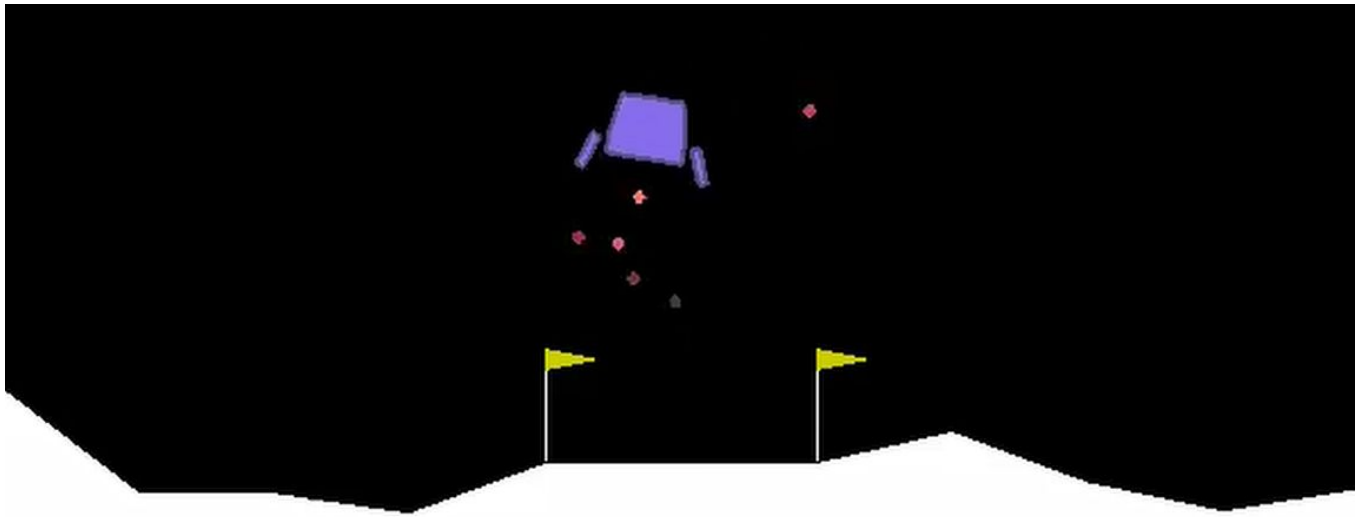
- It was predicted that solving Go using AI is approx. decade away
- Game of Go itself is simple… BUT!

- The number of board positions is about 10^{170}
- In our universe there are 10^{80} atoms

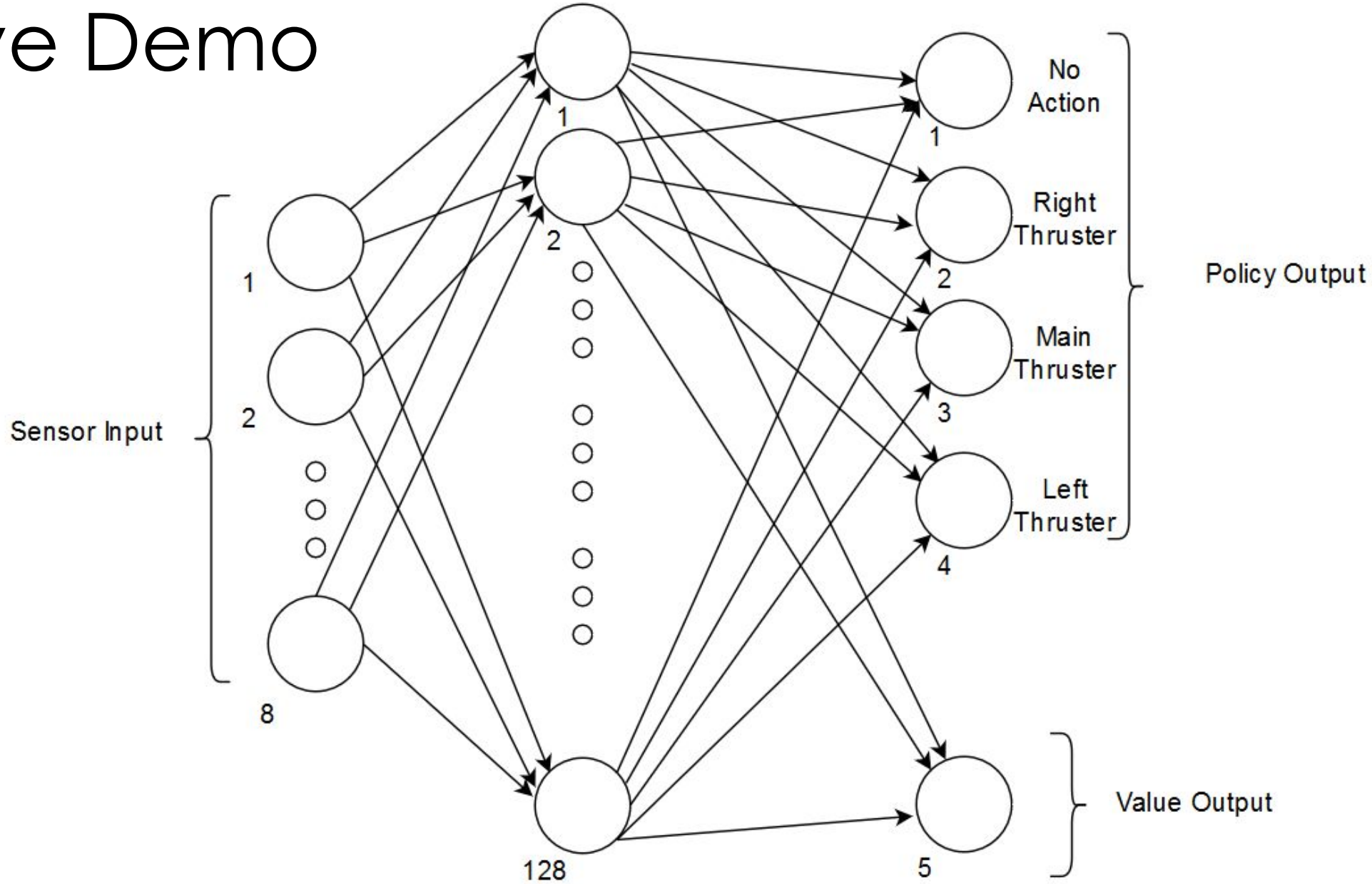


Live Demo

- Task -> land spaceship
- Input -> 8 sensors
- Output -> 4 actions



Live Demo


Sensor Input
1
2
8
1
2
128
No Action
Right Thruster
Main Thruster
Left Thruster
1
2
3
4
Policy Output
5
Value Output

Conclusion

- Reinforcement learning scenario is described by
  - States, actions, rewards, agent and environment
- Major components of RL agent are policy and value function
- Q-learning is extension of value function
- AlphaGo made breakthrough using novel RL method
- We have tested actual RL agent

Materials

- David Silver's lectures http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Teaching.html
- An Introduction to Reinforcement Learning, Sutton and Barto, 1998 http://incompleteideas.net/book/bookdraft2017nov5.pdf
- Python machine learning libs: Tensorflow (https://www.tensorflow.org/), PyTorch(https://pytorch.org/)

Thank you very much