Dog Vs Labrador Vs German Shepherd

PyData Meetup 11,
Mumbai, Aug 11, 2018

Pratik Bhavsar

Senior Data Scientist

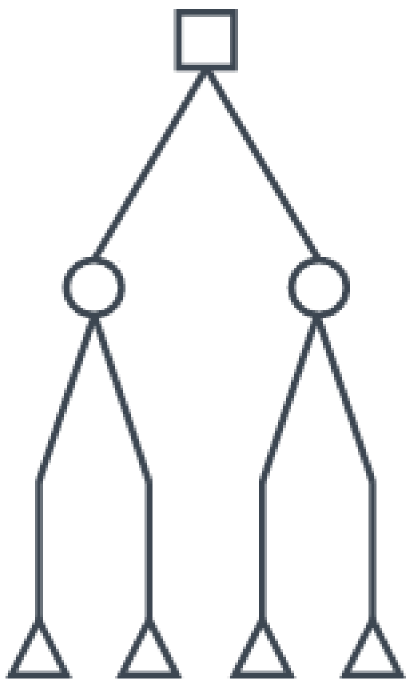
Morningstar

~~Dog Vs Labrador Vs German Shepherd~~

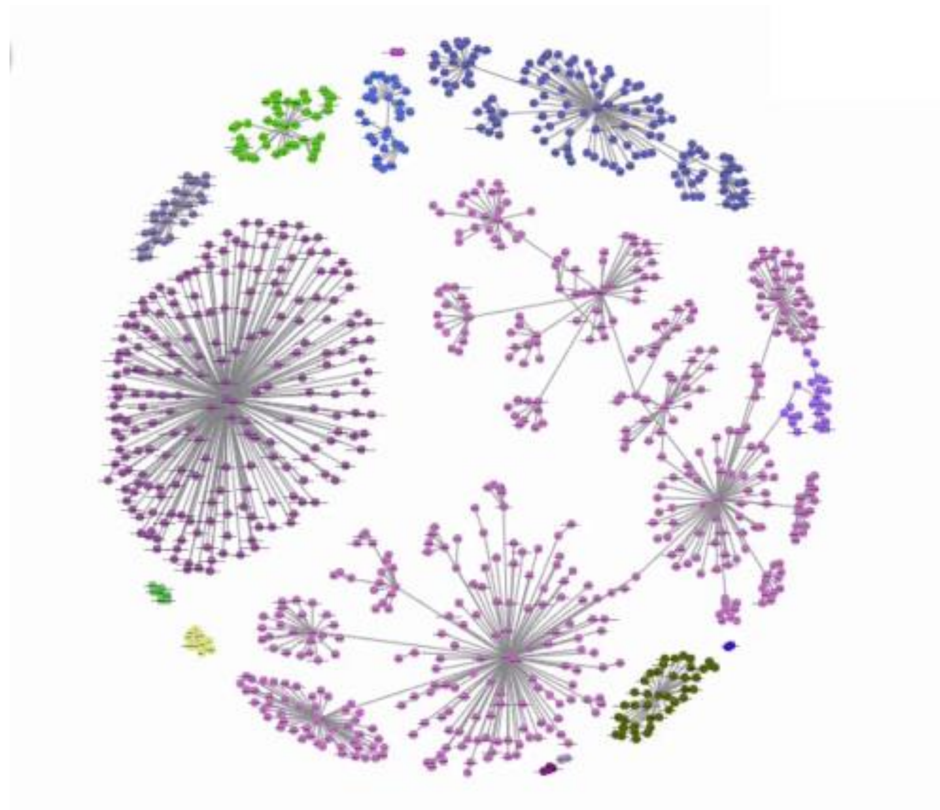
Machine Learning Vs Deep Learning
Vs Reinforcement Learning

Machine Learning

Supervised

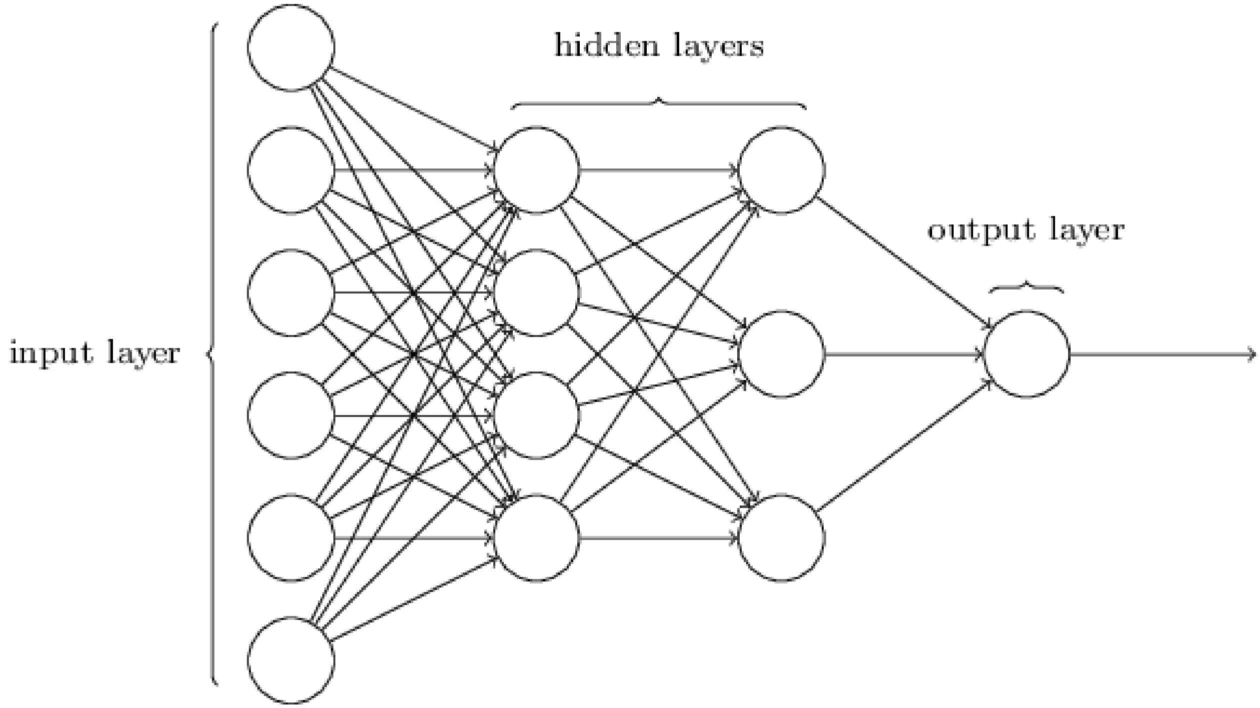


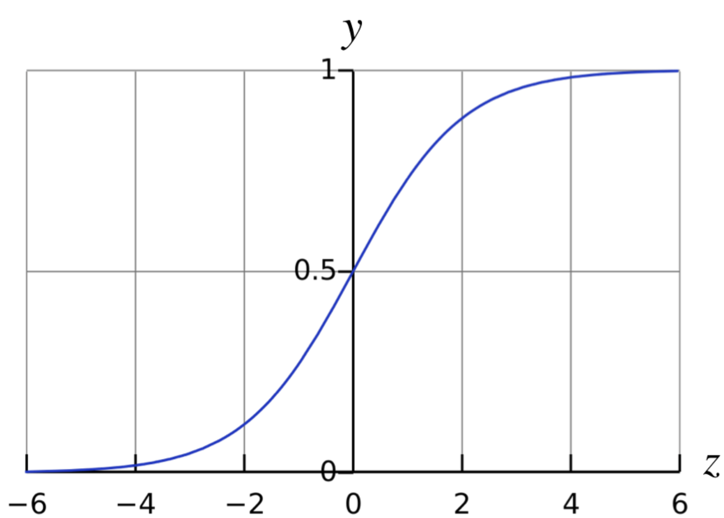
Unsupervised



Deep Learning

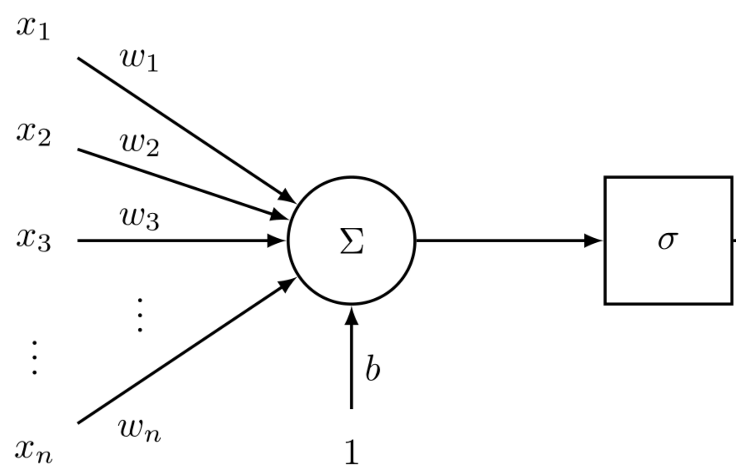
- [Universal approximation theorem]()
- [XOR function]()





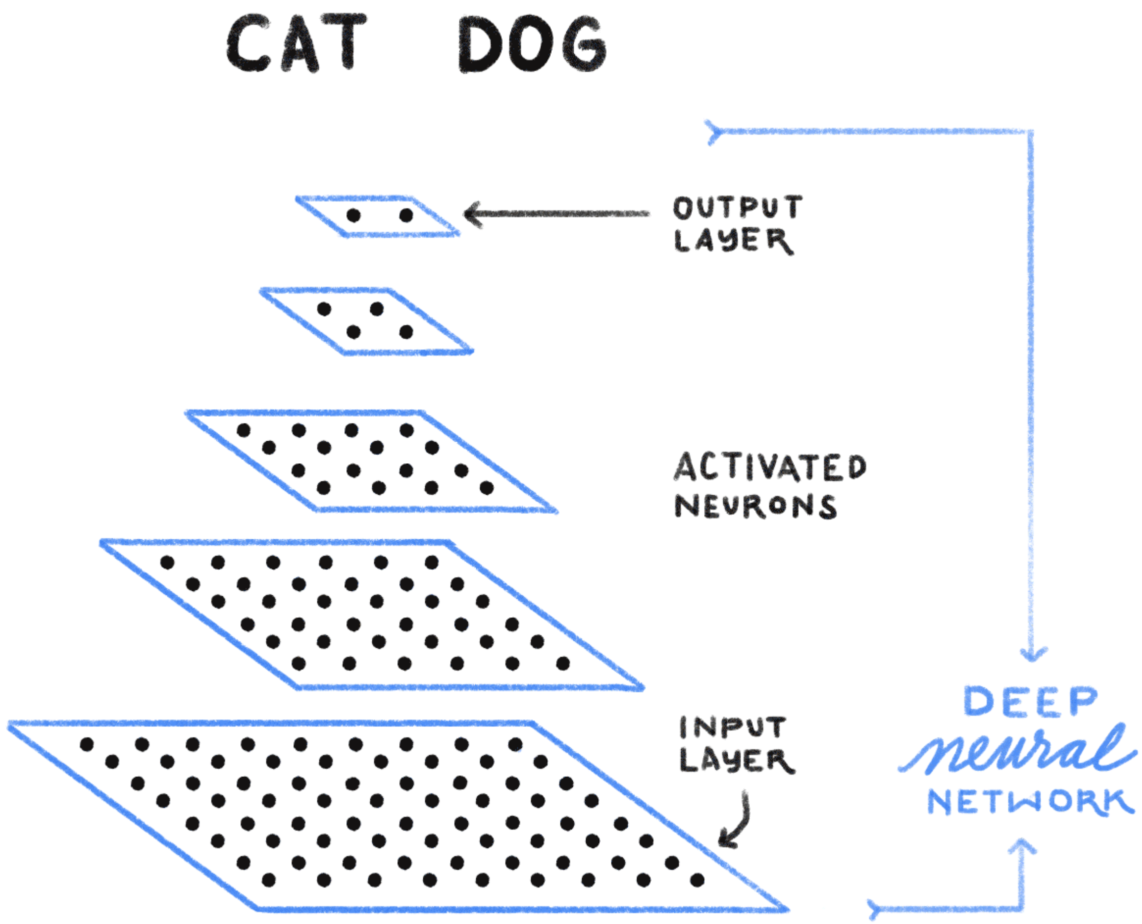
$$z = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

$$y = \frac{1}{1 + e^{-z}}$$



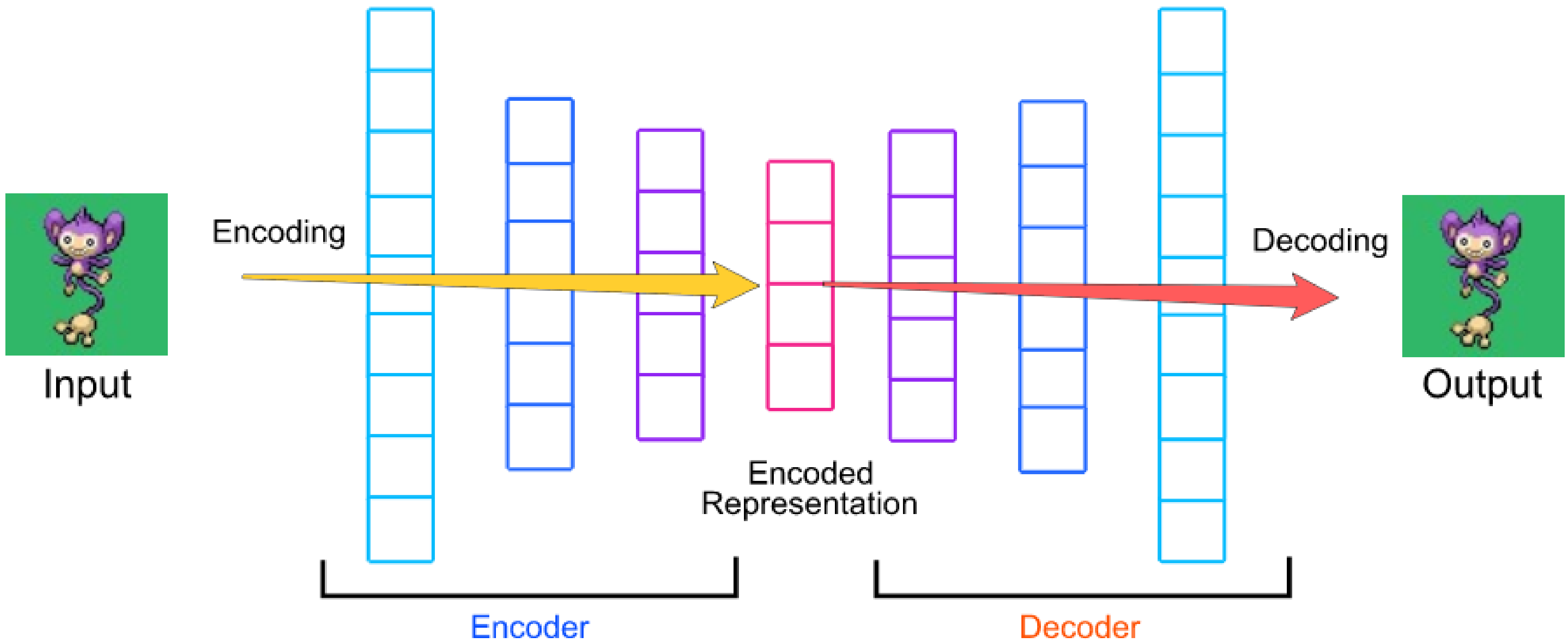
Deep Learning - Supervised

$$f_{W,b}(x) \approx y$$



Deep Learning - Unsupervised

$$f_{W,b}(x) \approx x$$



Reinforcement Learning

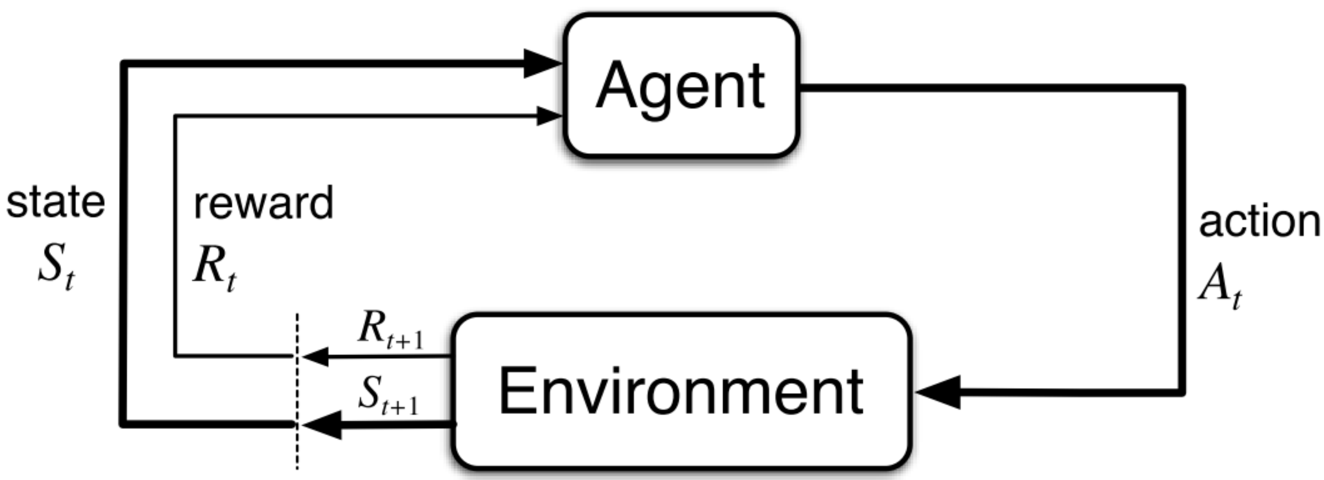
- Supervised Or Unsupervised?

Instruction based

- Supervised ML

Evaluation based

- Reinforcement learning



n-Armed Bandit Problem – A stationary problem

- Exploration Vs Exploitation

Agent's goal is to maximize the reward it receives in the long run.

How might this be formally defined?

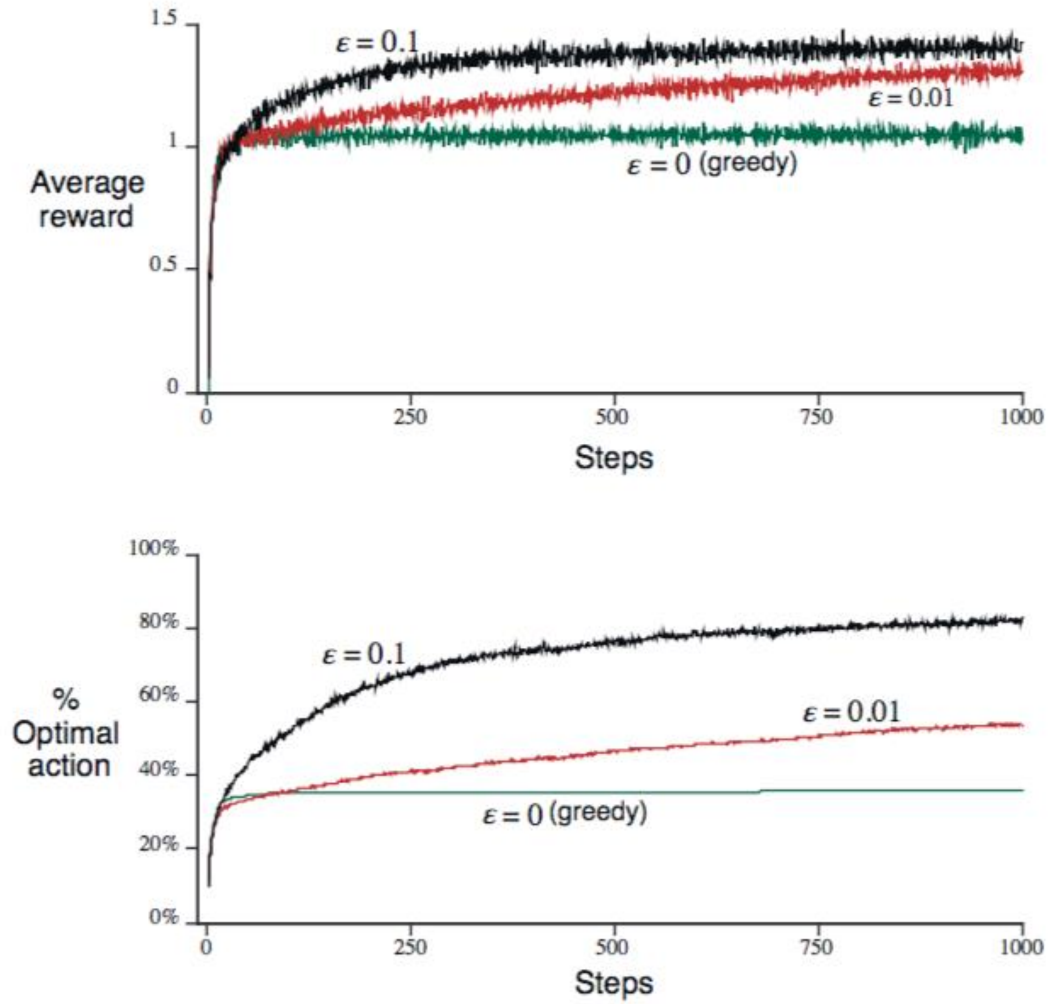
$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}$$

Average performance of ε-greedy action-value methods on the 10-armed testbed

n-Armed Bandit Problem – A stationary problem

- Exploration Vs Exploitation
  - Exploring restaurants

$$Q_t(a) = \frac{R_1 + R_2 + \cdots + R_{K_a}}{K_a}$$



Average performance of ε-greedy action-value methods on the 10-armed testbed

Reinforcement Learning Tasks

- Episodic tasks
  - Mario

- Continuous tasks
  - pubg

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T,$$

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

The Markov Property

- A stochastic process has the **Markov property** if the conditional probability distribution of **future** states of the process (conditional on both past and present states) **depends** only upon the **present** state, not on the sequence of events that preceded it.

- TLDR: Future can be predicted by just the present state. History is irrelevant.

$$\mathbf{Pr}\{R_{t+1} = r, S_{t+1} = s' \mid S_t, A_t\}$$

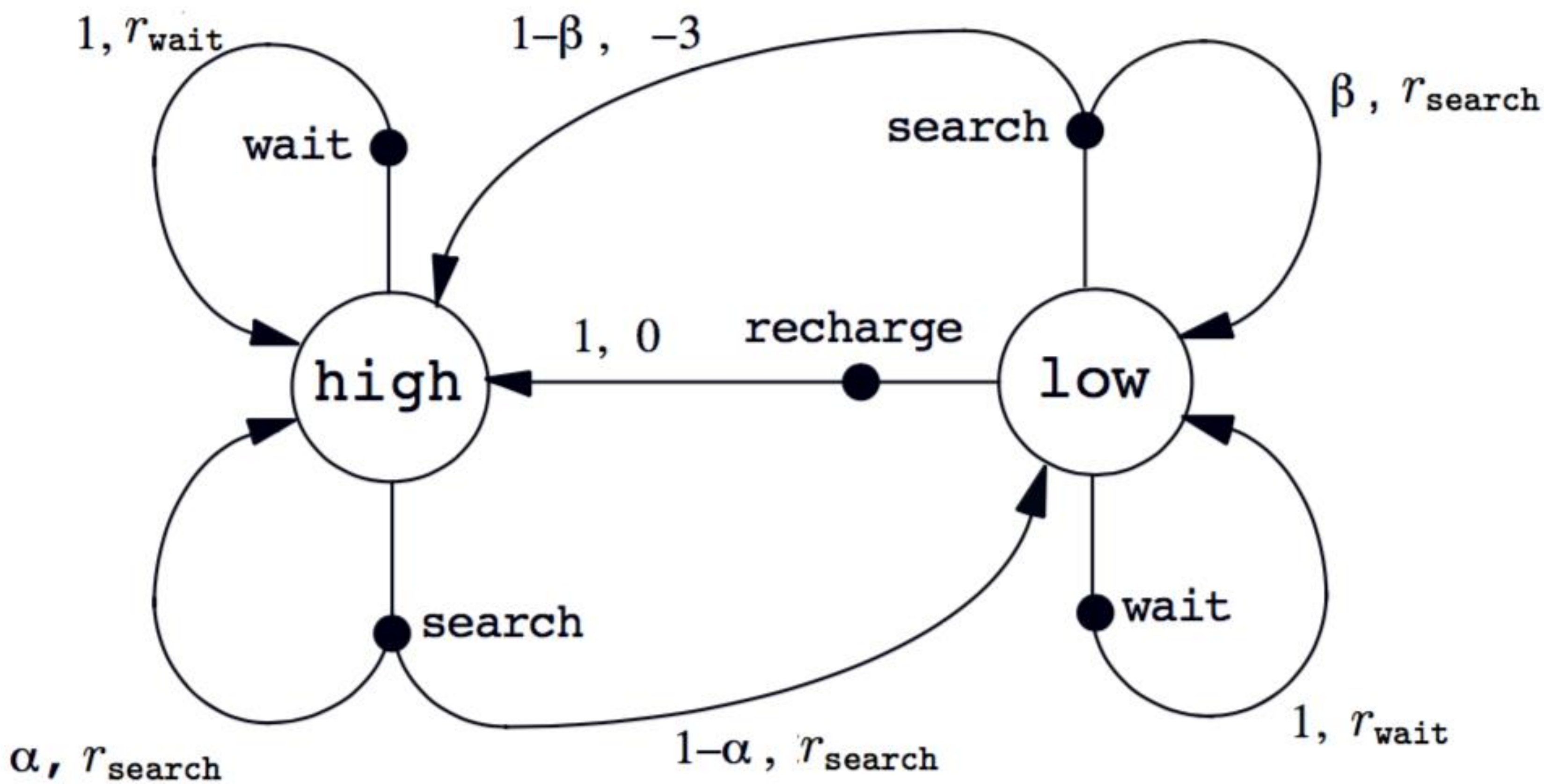
Recycling Robot MDP

1. Actively search for a can
2. Remain stationary and wait for someone to bring it a can
3. Go back to home base to recharge its battery.

s	s'	a	$p(s'\|s,a)$	$r(s,a,s')$
high	high	search	α	$r_{\texttt{search}}$
high	low	search	$1-\alpha$	$r_{\texttt{search}}$
low	high	search	$1-\beta$	-3
low	low	search	β	$r_{\texttt{search}}$
high	high	wait	1	$r_{\texttt{wait}}$
high	low	wait	0	$r_{\texttt{wait}}$
low	high	wait	0	$r_{\texttt{wait}}$
low	low	wait	1	$r_{\texttt{wait}}$
low	high	recharge	1	0
low	low	recharge	0	0.

$$A(\text{high}) = \{\text{search, wait}\}$$
$$A(\text{low}) = \{\text{search, wait, recharge}\}$$

Recycling Robot MDP



Transition graph for the recycling robot example

Value functions

- Value function = state–action pairs
  - Predict how good it is for the agent to perform a given action in a given state
  - Goodness is defined in terms of future reward that can be expected

$$v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \;\middle|\; S_t = s\right]$$

Reward of B.Tech

State-value function for policy π

Choosing career

What to do after B.Tech?

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \;\middle|\; S_t = s, A_t = a\right]$$

State-action function for policy π

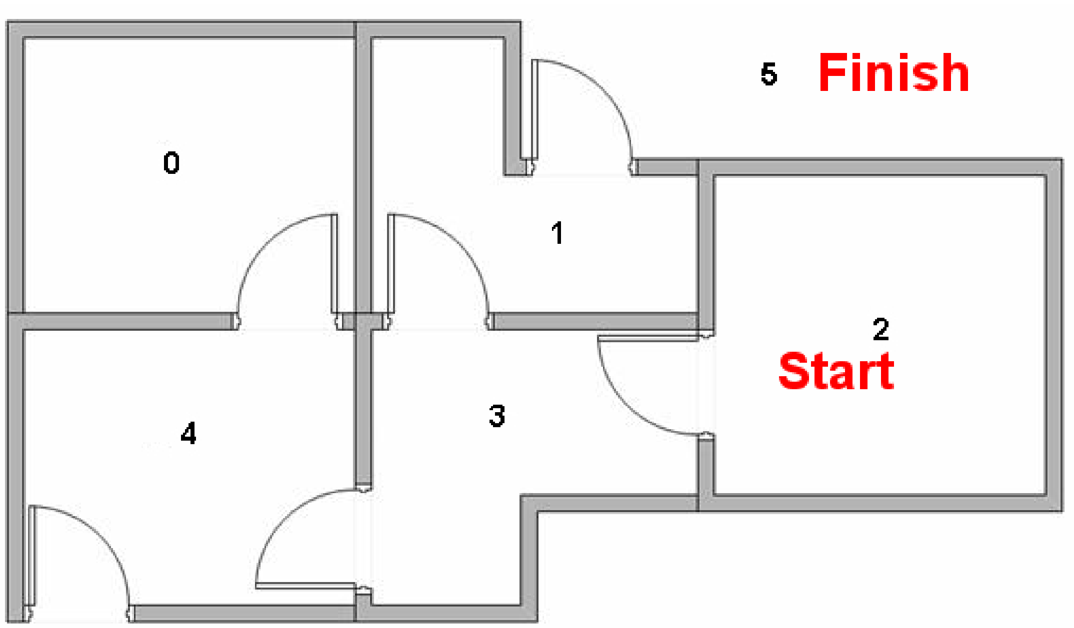
Policy?

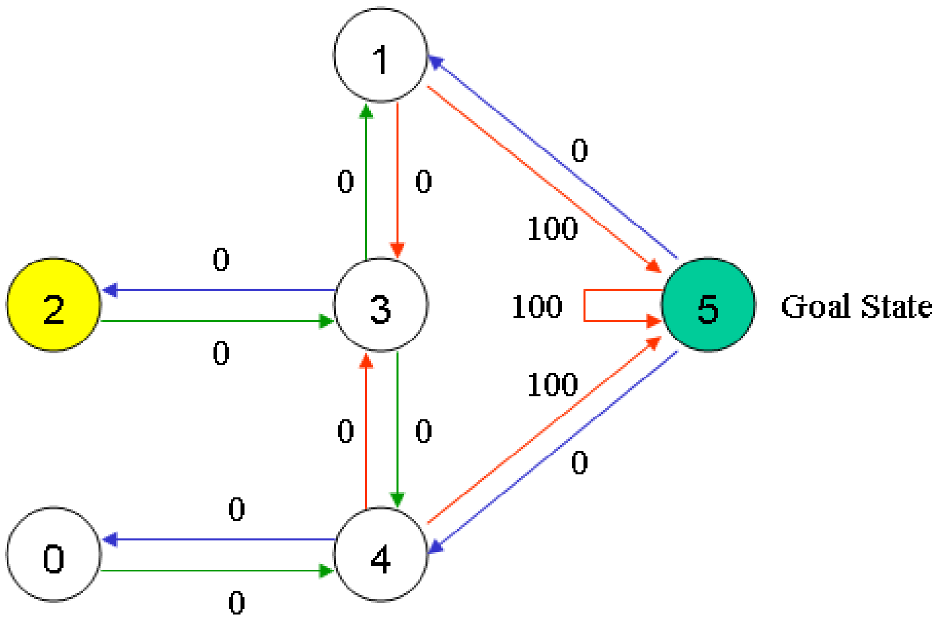
$$q_\pi(s,a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]$$

$$R = \begin{array}{c} \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \text{Action} \\ \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 64 & 0 & 100 \\ 0 & 0 & 0 & 64 & 0 & 0 \\ 0 & 80 & 51 & 0 & 80 & 0 \\ 64 & 0 & 0 & 64 & 0 & 100 \\ 0 & 80 & 0 & 0 & 80 & 100 \end{bmatrix} \end{array}$$

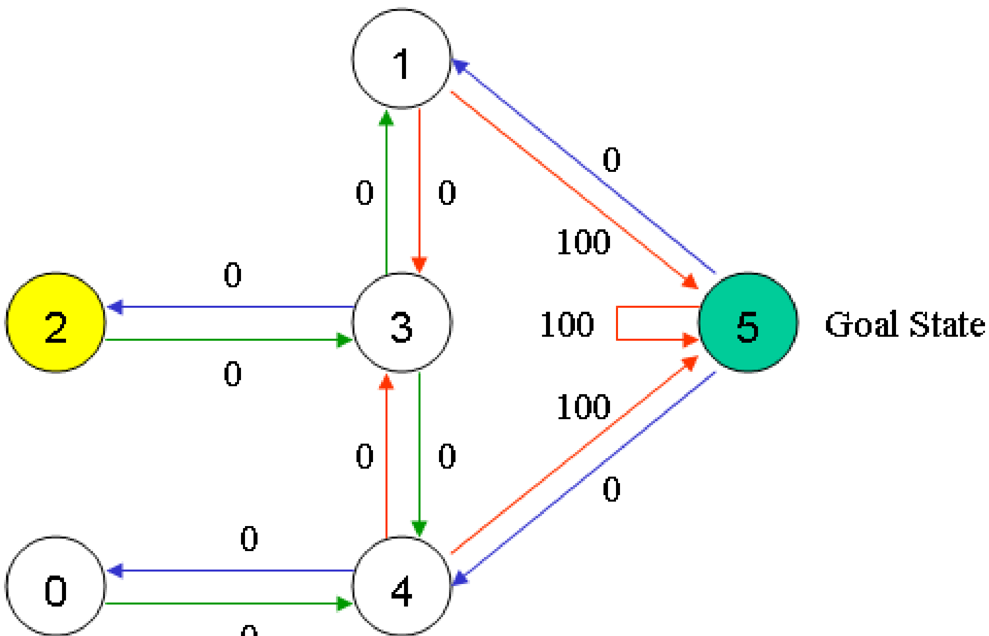
Reinforcement Learning – Q Learning





Reinforcement Learning – Q Learning

Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]

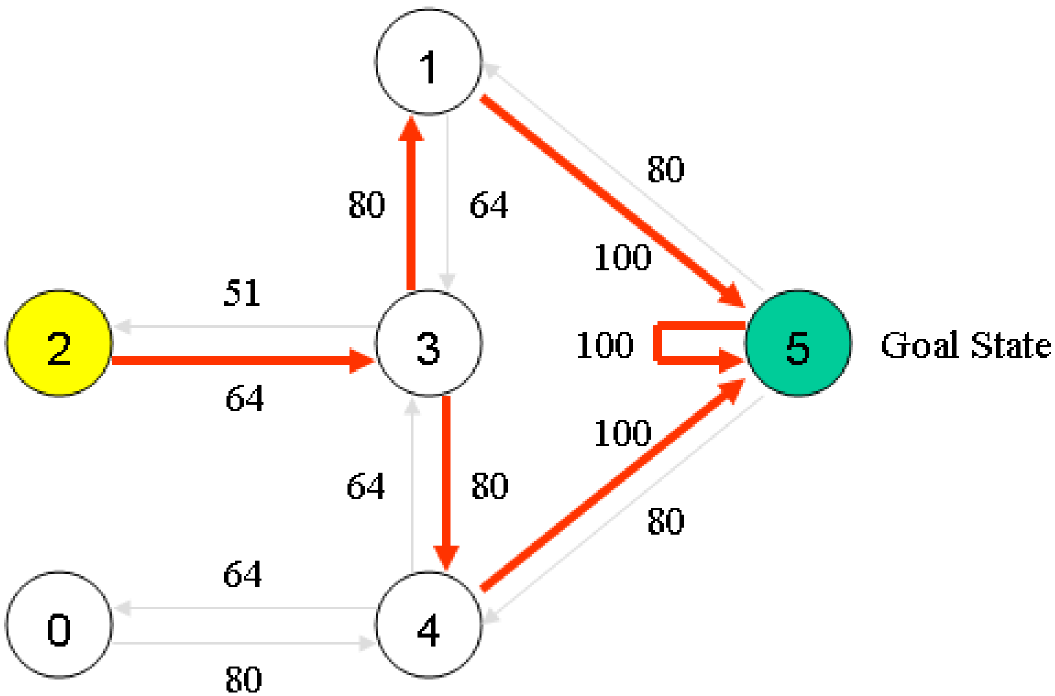


$$
\begin{array}{cc}
 & \text{Action} \\
\text{State} & \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\
R = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix}
\end{array}
$$

Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]

http://mnemstudio.org/path-finding-q-learning-tutorial.htm

Reinforcement Learning – Q Learning



$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 64 & 0 & 100 \\ 0 & 0 & 0 & 64 & 0 & 0 \\ 0 & 80 & 51 & 0 & 80 & 0 \\ 64 & 0 & 0 & 64 & 0 & 100 \\ 0 & 80 & 0 & 0 & 80 & 100 \end{array} \right] \end{array}$$

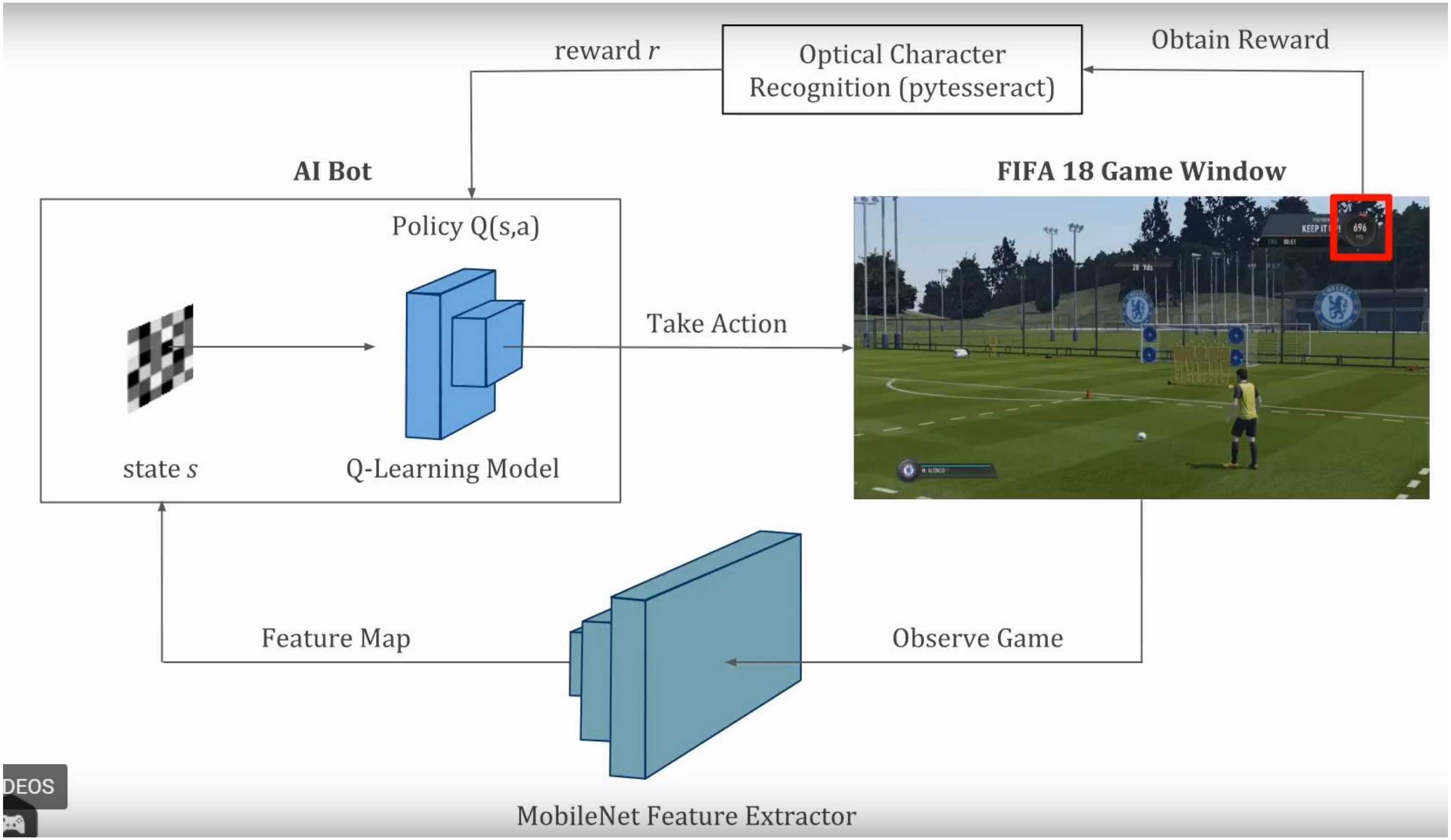
Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]

http://mnemstudio.org/path-finding-q-learning-tutorial.htm

Reinforcement Learning

- Applications
  - Finance
  - Game Theory and Multi-Agent Interaction
  - Robotics
  - Vehicular Navigation

Free Kicks in FIFA 2018 - Reinforcement Learning



What make Reinforcement Learning special?

Q/A Session