Reinforcement Learning Algorithms

Code Implementation

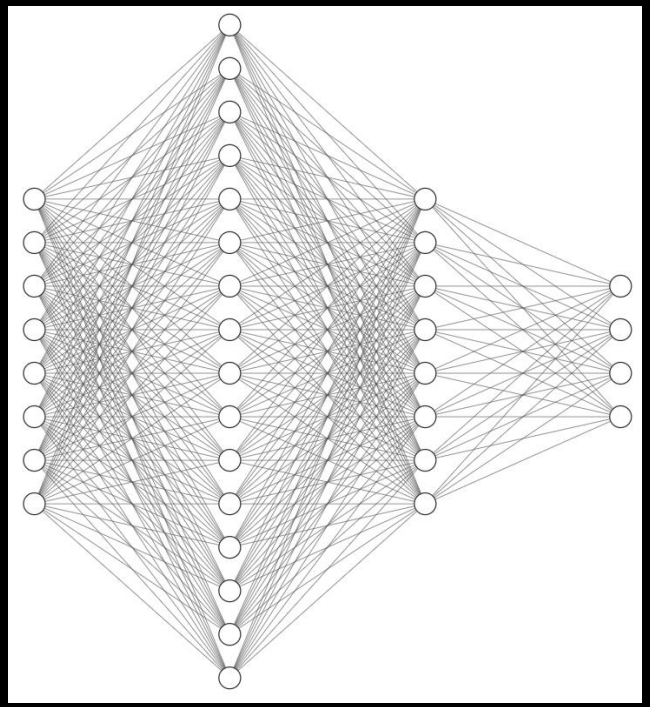
Saif Al Wahaibi
PhD Candidate

July 12th, 2023

Q-Learning

- Intuition:
  - *Estimate $Q_{\boldsymbol{\pi}}(s,a)$ via function approximation*

$s \rightarrow$



$\rightarrow \hat{Q}_{\boldsymbol{\theta}}(s,a_1)$

$\vdots$

$\rightarrow \hat{Q}_{\boldsymbol{\theta}}(s,a_n)$

- Math:

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{\pi}\left[\left(\delta_{TD} - \hat{Q}_{\boldsymbol{\theta}}(s,a)\right)^2\right] = \mathbb{E}_{\pi}\left[\left(r + \gamma \max_{a'} \hat{Q}_{\boldsymbol{\theta}}(s',a') - \hat{Q}_{\boldsymbol{\theta}}(s,a)\right)^2\right]$$

Q-Learning Algorithm

- Pseudocode:

Initialize $\hat{Q}_{\boldsymbol{\theta}}(s, a)$ *with random weights*
for *episode = 1, 2, 3, …, E* ***do***
Initialize environment $s_{\mathbf{0}}$
for *t = 0, 1, 2, …, T* ***do***
Select action a_t *randomly with probability* ϵ*, otherwise*

$$a_t = \underset{a_t}{\operatorname{argmax}}\, \hat{Q}_{\boldsymbol{\theta}}(s_t, a_t)$$

Execute action a_t *in environment and observe* r_{t+1}, s_{t+1}, *and terminal or truncate flags*
Set TD target $\delta_{TD} = r_{t+1}$ *if terminal or truncate, otherwise*

$$\delta_{TD} = r_{t+1} + \gamma \max_{a_{t+1}} \hat{Q}_{\boldsymbol{\theta}}(s_{t+1}, a_{t+1})$$

Perform a gradient descent step on

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{\pi}\left[\left(\delta_{TD} - \hat{Q}_{\boldsymbol{\theta}}(s_t, a_t)\right)^2\right]$$

Set s_{t+1} *as current state*
end for
end for

Q-Learning Example

- “CartPole-v1” Gym Environment: