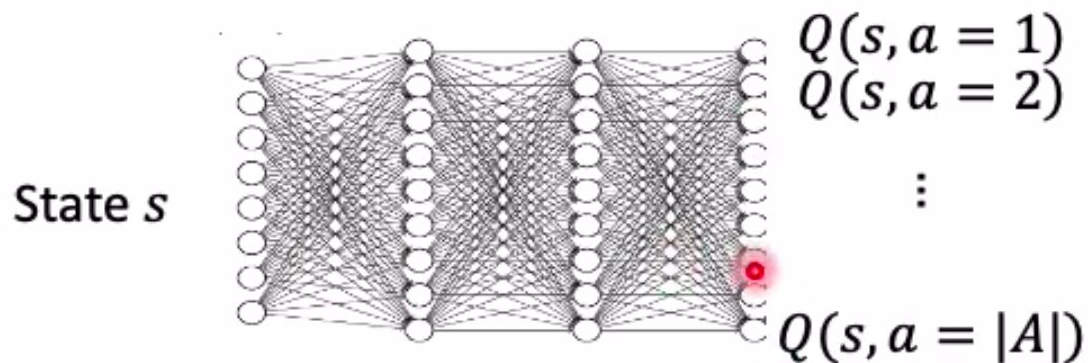


Extending DQN to Continuous Action Spaces

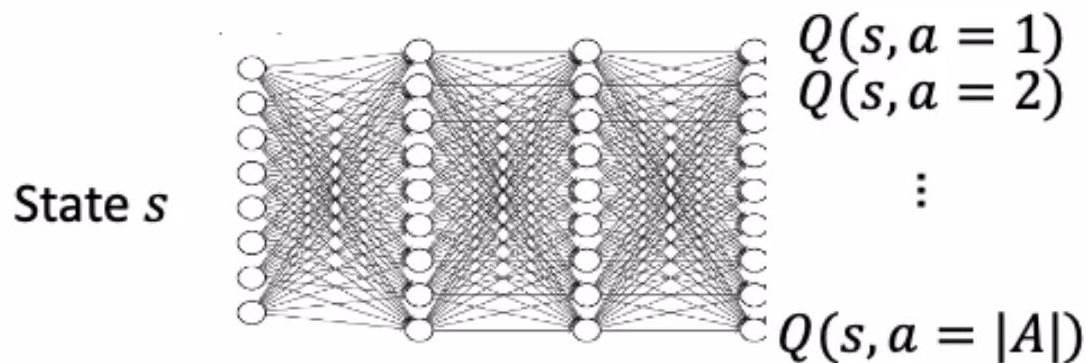
Selecting Optimal Actions with DQN



Policy:

$$\pi(s) = a^* = \underset{a}{\operatorname{argmax}} Q(s, a)$$

Selecting Optimal Actions with DQN



Policy:

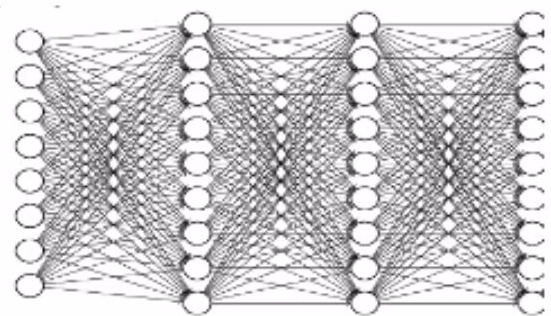
$$\pi(s) = a^* = \underset{a}{\operatorname{argmax}} Q(s, a)$$

Recall, this is also used in setting the Q targets at training time:

$$y_i = r_i + \gamma \max_{a'} Q_{\phi}(s'_i, a'_i) = r_i + \gamma Q_{\phi}(s'_i, \pi(s'_i))$$

DQN with Continuous Actions?

State s



~~$Q(s, a = 1)$~~

~~$Q(s, a = 2)$~~

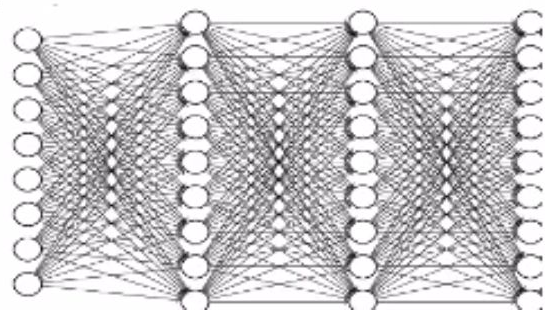
 ~~\vdots~~

~~$Q(s, a = |A|)$~~

Can't simply enumerate $Q(s, a)$ for all actions a any more.

State s

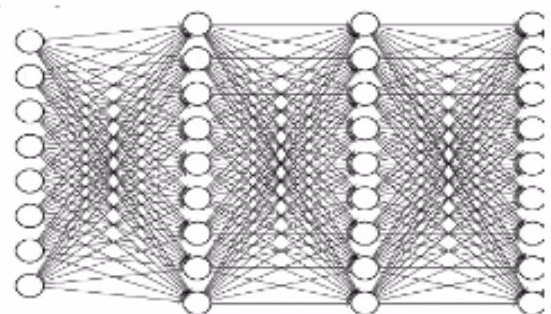
Action a



$Q(s, a)$

DQN with Continuous Actions?

State s



~~$Q(s, a = 1)$~~

~~$Q(s, a = 2)$~~

 ~~\vdots~~

~~$Q(s, a = |A|)$~~

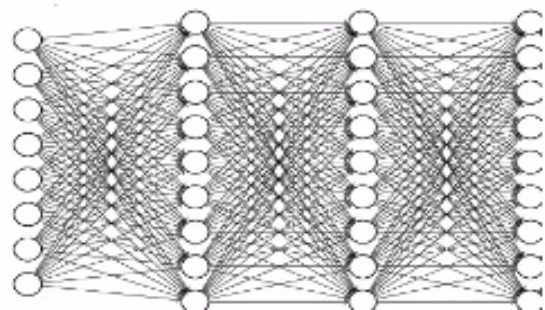
Can't simply enumerate $Q(s, a)$ for all actions a any more.

Can't simply set $\pi(s) = a^* = \underset{a}{\operatorname{argmax}} Q(s, a) \dots$

Treat $\underset{a}{\operatorname{argmax}} Q(s, a)$ as an optimization problem for every action and solve with, say, gradient descent?

State s

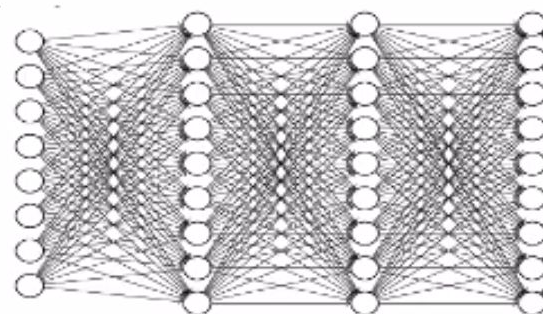
Action a



$Q(s, a)$

DQN with Continuous Actions?

State s



$$Q(s, a = 1)$$

$$Q(s, a = 2)$$

\vdots

$$Q(s, a = |A|)$$

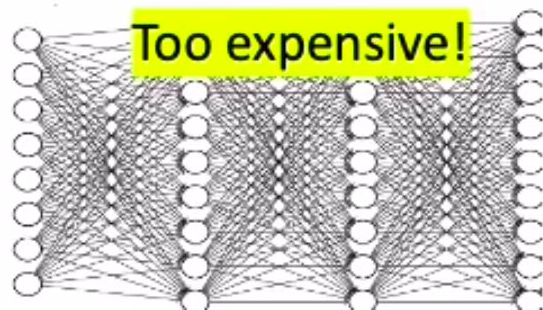
Can't simply enumerate $Q(s, a)$ for all actions a any more.

Can't simply set $\pi(s) = a^* = \underset{a}{\operatorname{argmax}} Q(s, a) \dots$

Treat $\underset{a}{\operatorname{argmax}} Q(s, a)$ as an optimization problem for every action and solve with, say, gradient descent?

State s

Action a



Too expensive!

$$Q(s, a)$$

DQN with Continuous Actions?

Could we train a neural network to produce the output of this optimization problem?

$$a^*(s) = \operatorname{argmax}_a Q(s, a)$$

