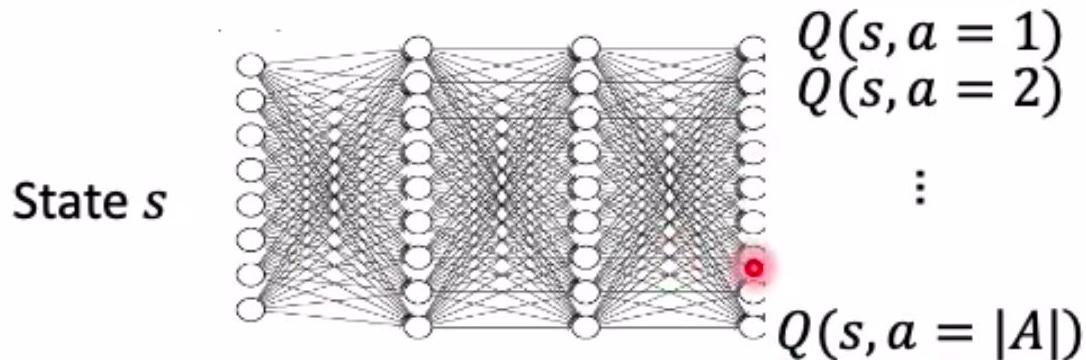


Extending DQN to Continuous Action Spaces

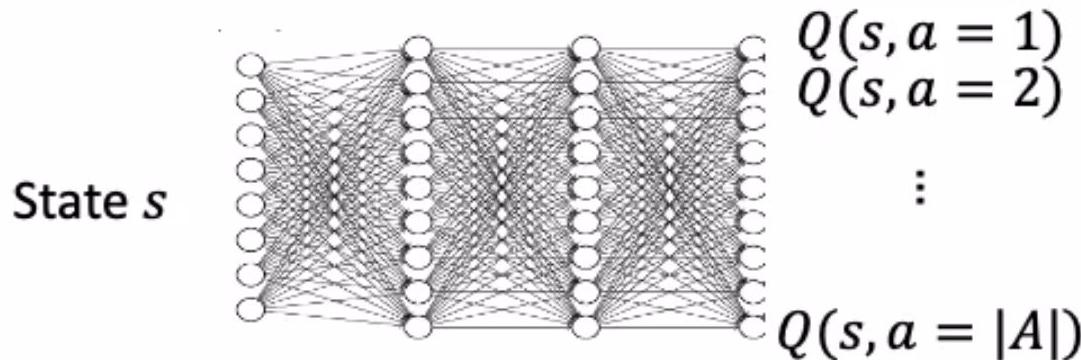
Selecting Optimal Actions with DQN



Policy:

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

Selecting Optimal Actions with DQN



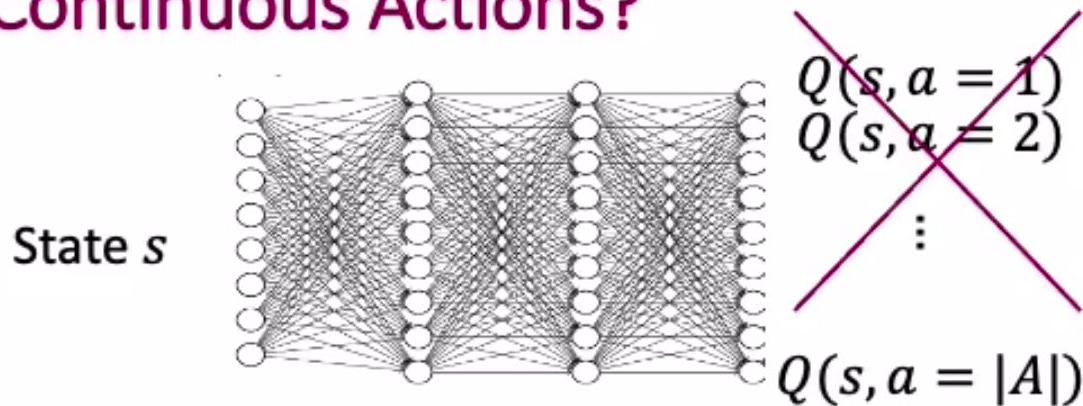
Policy:

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

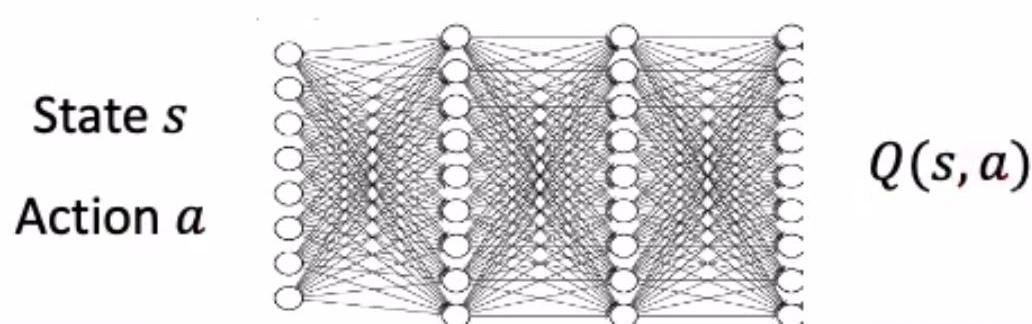
Recall, this is also used in setting the Q targets at training time:

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

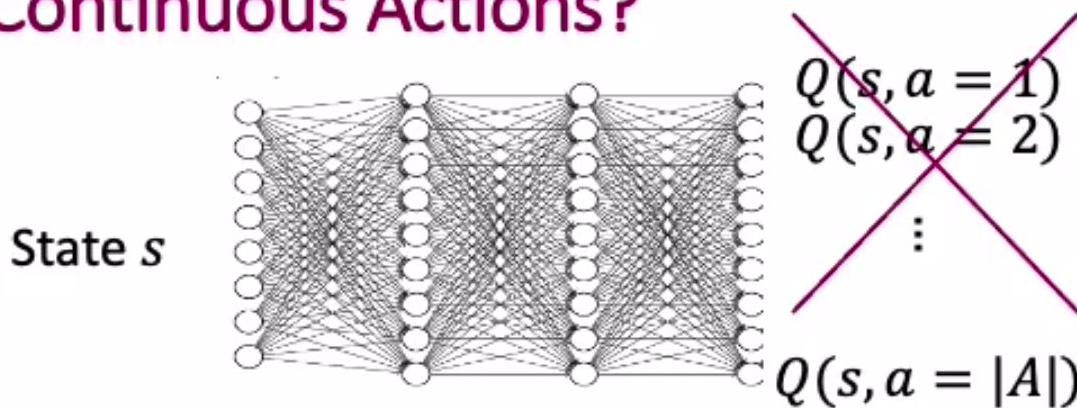
DQN with Continuous Actions?



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



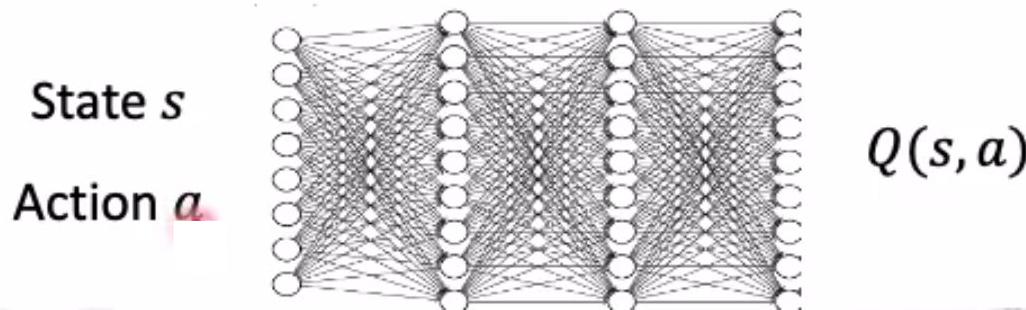
DQN with Continuous Actions?



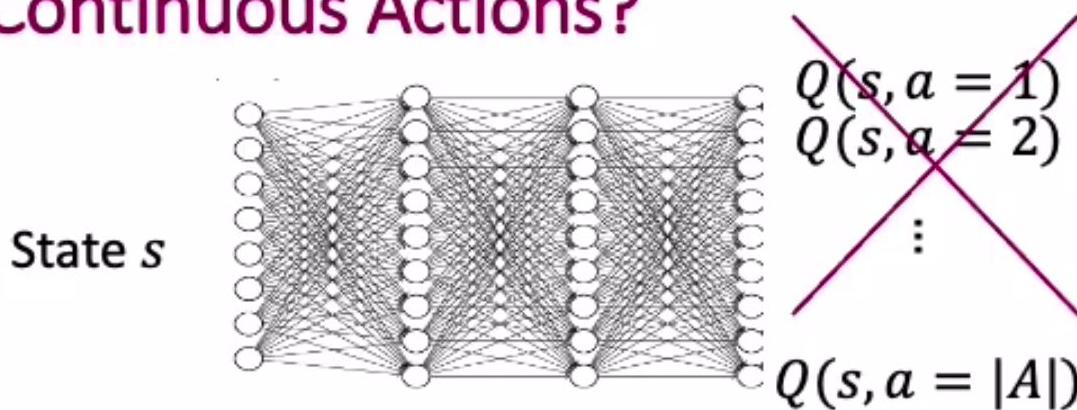
Can't simply enumerate $Q(s, a)$ for all actions a any more.

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?



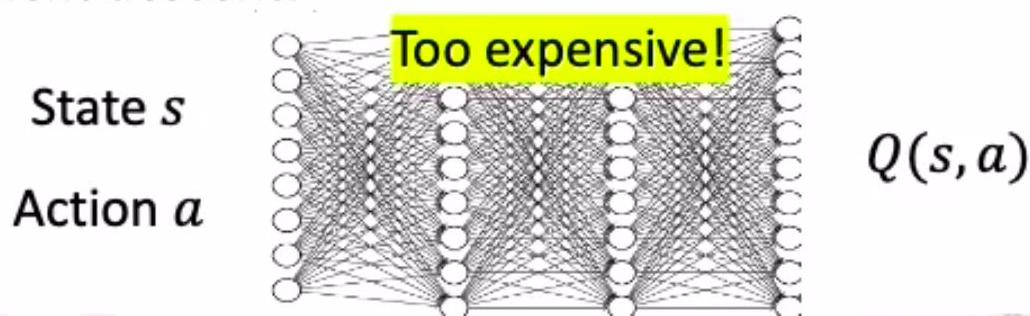
DQN with Continuous Actions?



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?



DQN with Continuous Actions?

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

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

