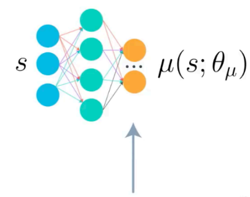
DRLND Project 2: Continuous Control
Report

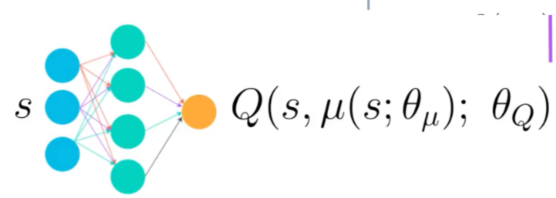
DEEP DETERMINISTIC GRADIENT POLICY (DDPG) ALGORITHM

We have two NNs - an actor and critic

Actor: Learns to approximate the optimal policy deterministically, $\mu(s \mid \theta_\mu)$ - we want it to output to best action for any given state
It's basically learning the $\arg\max_a Q(s, a)$



Critic: Learns to approximate the optimal action value function by using the actors's best believed action, $Q(s, \mu(s;\theta_\mu); \theta_Q)$



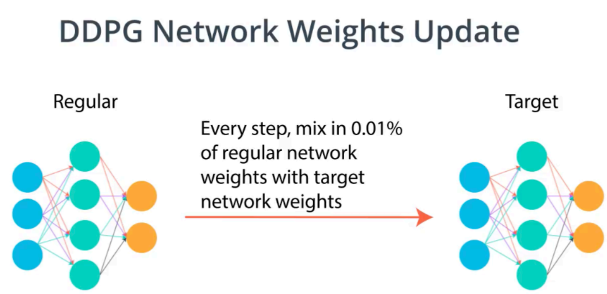
DDPG NETWORK WEIGHTS UPDATE

We have two copies of the network weights for each network:

- Regular Network for Actor
- Regular Network for Critic
- Target Network for Actor
- Target Network for Critic

The target networks are updates using a soft update strategy
This consists of slowly mixing the regular network weights into the target network weights
At every time step, we make the target network consist of,

$$\text{Target Network Weights} = \begin{cases} 99.99\,\% \text{ Target Network Weights} \\ 0.01\,\% \text{ Regular Network Weights} \end{cases}$$



MODEL ARCHITECTURE & HYPERPARAMETERS

Actor:

1. ReLU + Linear Layer: $33 \rightarrow 128$
2. ReLU + Linear Layer: $128 \rightarrow 128$
3. Tanh + Linear Layer: $128 \rightarrow 4$

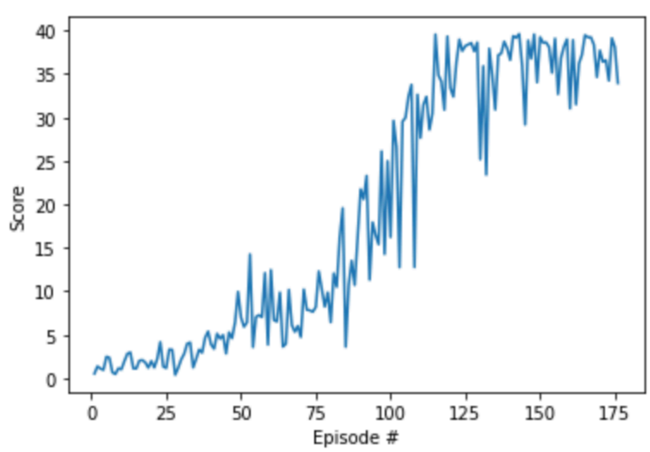
Critic:

1. ReLU + Linear Layer: $33 \rightarrow 128$
2. Concatenate Dimensions: $128 \rightarrow 132$
3. ReLU + Linear Layer: $132 \rightarrow 128$
4. Linear Layer: $128 \rightarrow 1$

Hyperparameters:

- Replay Buffer Size: e^5
- Mini-Batch Size: 128
- Discount Factor: 0.99
- Interpolation Parameter: e^{-3}
- Actor's Learning Rate: $2e^{-4}$
- Critic's Learning Rate: $2e^{-4}$
- Weight Decay: 0

PLOT OF REWARDS



FUTURE IDEAS TO IMPROVE PERFORMANCE

To improve learning stability, we can try different algorithms such as Trust Region Policy Optimization (TRPO), Truncated Natural Policy Gradient (TNPG), Proximal Policy Optimization (PPO), and Distributed Distributional Deterministic Policy Gradients (D4PG)