Udacity's Deep Reinforcement Learning Nanodegree Project 2 Report:

December 24, 2018

1 Introduction

2 Learning Algorithm

The Deep Deterministic Policy Gradient (DDPG) algorithm [1, 2] is implemented in this project:

```
Initialize replay memory $D$ with capacity $N$
Initialize critic network $\hat{q}$ with random weights $w^q$
Initialize actor network $\mu$ with weights $w^\mu$
Initialize target critic weights $w^{q-} \leftarrow w^q$
Initialize target actor weights $w^{\mu-} \leftarrow w^\mu$
for the episode $e \leftarrow 1$ to $M$ do
    Initialize a random process $\mathcal{N}$ for action exploration
    Receive initial input state $S$
    for time step $t \leftarrow 1$ to $T$ do
        Choose action $A = \mu(S|w^\mu) + \mathcal{N}$
        Take action $A$, observe reward $R$, and next input frame $S'$
        Store experience $(S, A, R, S')$ in replay memory $D$
        $S \leftarrow S'$
        Obtain minibatch of tuples $(s_j, a_j, r_j, s_{j+1})$ from $D$ of size $K$.
        Set target $y_j = r_j + \gamma\hat{q}(s_{j+1}, \mu(S', w^{\mu-}), w^{q-})$
        Update $w^q$: $\Delta w^q = -\alpha\frac{1}{N}\sum_j (y_j - \hat{q}(s_j, a_j, w^q))\nabla_{w^q}\hat{q}(s_j, a_j, w^q)$
        Update $w^\mu$ with policy gradient:
            $$\nabla_{w^\mu} J \approx \frac{1}{N}\sum_i \nabla_a \hat{q}(s_i, a\mu(s_i))\nabla_{w^\mu}\mu(s_i|w^\mu)$$
        Soft update $w^{q-}$: $w^{q-} \leftarrow (1-\tau)w^{q-} + \tau w^q$
        Soft update $w^{\mu-}$: $w^{\mu-} \leftarrow (1-\tau)w^{\mu-} + \tau w^\mu$
```

3 Results and Plots of Rewards

4 Ideas for Future Work

References

[1] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, sep 2015.

[2] Udacity. Deep Reinforcement Learning Nanodegree Course Material, 2018.