Motivation: Previously, Deep Q Network (DQN) <Mnih et al., 2015> has shown its success on many different Atari video games by estimating the action value. However, DQN can only handle discrete and low dimensional action spaces. When the action is continuous, that will result a high dimensional action spaces. Deep Deterministic Policy Gradient (DDPG) is a combination of DQN and Policy Gradient <Silver et al., 2014> to overcome the problem of high dimensional action spaces.

Methods:

1. Create 2 online network: actor $\mu(s|\theta^\mu)$ and critic $Q(s,a|\theta^Q)$ with weights θ^μ and θ^Q.
2. Create 2 target network: actor μ' and critic Q' with weights $\theta^{\mu'} \leftarrow \theta^\mu$, $\theta^{Q'} \leftarrow \theta^Q$.
3. Initialize replay buffer R.
4. For episode = 1 $\rightarrow$ M do
    a. Initialize a random process N for action exploration.
    b. Receive initial observation state s_1.
    c. For t = 1 $\rightarrow$ T do
        - Select action $a_t = \mu(s_t|\theta^\mu) + N_t$ by epsilon greedy.
        - Execute action a_t and observe reward r_t and observe new state s_{t+1}.
        - Store transition (s_t, a_t, r_t, s_{t+1}) in R.
        - Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R.
        - Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$.
        - Update critic by minimizing the L2 loss:

$$L = \frac{1}{N} \sum\nolimits_{i}(y_i − Q(s_i, a_i|\theta^Q))^2$$

        - Update actor policy using sampled policy gradient from Silver et al., 2014:

$$\triangledown_{\theta^\mu}J \approx \frac{1}{N} \sum\limits_i \triangledown_a Q(s, a| \theta^Q)|_{s=s_i, a=\mu(s_i)} \triangledown_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

        - Update target networks with $\tau \ll 1$ to slowly adapt to new policy and keep the stability of learning:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

References:

- Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning." *arXiv preprint arXiv:1509.02971*(2015).
- Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." *Nature* 518.7540 (2015): 529.
- Silver, David, et al. "Deterministic policy gradient algorithms." *ICML*. 2014.

Previously, Deep Q Networks (DQN) (Mnih et al., 2015) have shown its success on many different Atari video games by estimating the action value. However, DQN can only handle discrete and low dimensional action spaces. When the action is continuous, that will result a high dimensional action spaces which requires new learning framework that can deal the continuous action. Deep Deterministic Policy Gradient (DDPG) is a combination of DQN and Policy Gradient (Silver et al., 2014) to overcome the problem of high dimensional action spaces.

DDPG uses an actor-critic approach based on Deterministic Policy Gradient algorithm (Silver et al., 2014) . DDPG maintains a parameterized actor function $\mu(s|\theta^{\mu})$ which maps a states to a specific action a with a parameter θ^{μ}. The actor is updated by applying the chain rule to the expected return from the start distribution J with respect to the actor parameters:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N} \sum_{i} \theta_a Q(s, a|\theta^{Q})|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|_{s_i}$$

.

The critic $Q(s, a)$ learns the weight theta using Bellman equation as in Q-learning. The critic is updated by minimizing the L2 loss:

$$L = \frac{1}{N} \sum\nolimits_{i} (y_i - Q(s_i, a_i|\theta^{Q}))^2$$

.

Target networks is used to calculate y_i because practically learning from large, non-linear function approximators tends to be unstable <Mnih et al., 2015>. The target networks are slowly updated:

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta' \text{ with } \tau \ll 1.$$

Slowly updating target network will improve the stability of learning. Neural networks for reinforcement learning assume the samples are independently and identically distributed. Therefore, the authors use a replay buffer and minibatch training to address the sequential exploration issue of online learning. With batch normalization, the algorithm learns effectively across many different tasks with different types of units. An Ornstein-Uhlenbeck process (Uhlenbeck & Ornstein, 1930) is used to select temporally correlated action for exploration in physical control problems with inertia. That results in adding Gaussian noise $\mathcal{N}$ to the target actor:

$$\mu'(s_t) = \mu(s_t|\theta_t^{\mu}) + \mathcal{N}$$

.

DDPG framework is not only capable of providing end-to-end solution, which means generating a solution from raw pixel input, but also capable of getting a solution for a high dimensional and continuous action space. However, the quality of experimental results in the paper is not convincing. The algorithm only runs 5 times for each game and records the average and best observation from those 5 runs. The difference

between average and best is quite high even though they are normalized. Thus, from this observation, we doubt the stability of the algorithm.