

This handout includes space for every question that requires a written response. Please feel free to use it to handwrite your solutions (legibly, please). If you choose to typeset your solutions, the | README.md | for this assignment includes instructions to regenerate this handout with your typeset L^AT_EX solutions.

1.b

Based on Jensen's inequality, we have:

$$\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$$

for random variable X and a convex function f .

The maximum function is a convex function and we consider Q-values for all actions in a given state as a random variable. Applying Jensen's inequality, we get:

$$\mathbb{E}[\max_{a \in \mathcal{A}} Q(s, a)] \geq \max_{a \in \mathcal{A}} (\mathbb{E}[Q(s, a)])$$

We are given that $\mathbb{E}[Q(s, a)] = Q^*(s, a)$, so

$$\mathbb{E}[\max_{a \in \mathcal{A}} Q(s, a)] \geq \max_{a \in \mathcal{A}} Q^*(s, a)$$

2.a

The partial derivative of a dot product is simply the other vector in the dot product (i.e. $\partial(u \cdot v)/\partial u = v$), so

$$\nabla_{\theta} Q_{\theta}(s, a) = \frac{\partial Q_{\theta}(s, a)}{\partial \theta_{s, a}} = \frac{\partial \theta_{s, a} \cdot \delta(s, a)}{\partial \theta_{s, a}} = \delta(s, a)$$

Substitute this and into the linear update equation:

$$\theta \leftarrow \theta + \alpha(r + \gamma \max_{a'} Q_{\theta}(s', a') - Q_{\theta}(s, a)) \nabla_{\theta} Q_{\theta}(s, a)$$

We get:

$$\theta \leftarrow \theta + \alpha(r + \gamma \max_{a'} Q_{\theta}(s', a') - Q_{\theta}(s, a)) \delta(s, a)$$

Shows that only the single θ component correspond to (s, a) will be updated and thus identical to:

$$\theta \cdot \delta(s, a) \leftarrow \theta \cdot \delta(s, a) + \alpha(r + \gamma \max_{a'} Q_{\theta}(s', a') - Q_{\theta}(s, a)) \delta(s, a)$$

And therefore

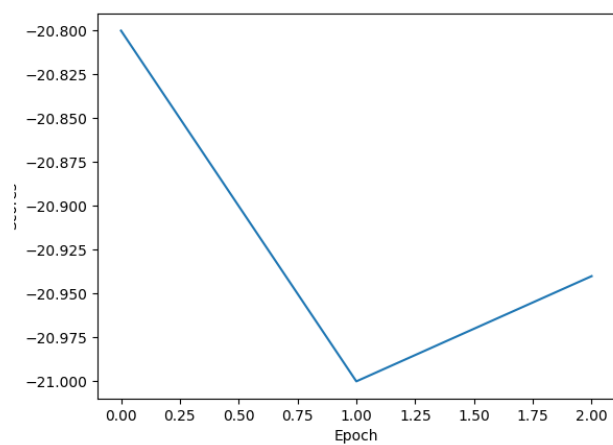
$$Q_{\theta}(s, a) \leftarrow Q_{\theta}(s, a) + \alpha(r + \gamma \max_{a'} Q_{\theta}(s', a') - Q_{\theta}(s, a))$$

3.c

The linear model is obviously learn quicker and give you more stable results while DQN can sometimes fail to reach the optimal policy with the epoch we choose. DQN might need even more training steps to reach the optimal policy.

In conclusion, for the simple test environment, the linear model is the better choice due to its efficiency and appropriate complexity. The DQN is unnecessarily powerful for this task.

4.a



The performance of the linear model on the Pong-v5 environment is extremely poor and shows no evidence of learning.

4.c

The DQN is obviously improving its performance over time while the linear model doesn't seem to improve.

The DQN has a deep CNN layer that are specifically designed for image data. It gives the DQN the understanding of the game state directly from pixels, which is necessary for high-level strategic play.

On the other hand, the linear model treat each pixels as an independent feature and attempts to find a simple linear relationship between pixel values and strategy, which is insufficient.

4.d

No, the performance of DQN is improving in the long run but it fluctuates a lot locally. It could be the ϵ -greedy exploration where it randomly chooses the actions sometimes that causes the fluctuation. Also, the deep neural network is a highly non-linear function approximation. A small parameter change can lead to significant change in the policy.