ADL HW5 Report

資工所 碩一 R05922021 陳翰浩

- Experience Replay

  為了避免在training的時候，每一筆training data之間有相依的關係以及相同的分佈情況，我們需要透過取樣的方式從先前的training data中隨機的取出一些sample來做training，所以我們需要先train一段的時間並把每一次的結果丟進一個pool中，之後再從中隨機選出一定的數量來做training，由於training data是透過隨機取樣的方式取得，data間的相依性就不會存在於該筆data中。

- Target Network

$$L(w) = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a', w) - Q(s, a, w)\right)^2\right]$$

  而在Target network中，我們要避免的是，在做backpropagation的時候會同時更新Q-network以及target network的問題，這會造成我們training的震盪，我們要求的Q function可能會在極大或極小間，不容易找到一個好的權重W，要解決這個問題，我們需要在更新的時候只更新target network，而在Q-network取值的時候，使用前一個週期所更新出來的值來做預測，這一來我們可以避免training的震盪的問題。

- Epsilon Greedy

  我們在選取agent的不同行為時可以透過一個機率讓agent可以隨機的選取一個任意的行為，而不依照model所預測的結果去做，讓我們的agent有機會去探索不同的行為模式。

- Clip Reward

  在每一輪所獲得的reward，我們會將它的極大以及極小值限定在一個區間[-1, 1]，這一來我們要最佳化的目標的變化量變小了，可以讓我們在taining的過程更容易找到一個好的最佳解。min(1, max(-1, reward))

- If a game can perform two actions at the same time, how will you solve this problem using DQN algorithm ?

  由於原先的DQN model在選擇每一次預測的action時，只是將output layer中值最大的那項取出來當作該輪的預測結果，因此我們可以透過一個簡單的方式直接將，Q(s, a)變成有兩個input action的pair Q(s, (a1, a2))，而在我們的network的output的選項在預測的結果的類別的數量的總數，設為C(類別數, 2) + 類別數，每次選分數最高的那一個類別，當作是預測的結果，因此就可以預測出兩個action(加上類別數的目的是為了讓他，有執行單項action的可能)。

- Reference

  https://github.com/gtoubassi/dqn-atari