
# Lecture 12:
# Advanced Value Based Methods

B. Ravindran

# Recall Double Q-learning

**Double Q-learning, for estimating $Q_1 \approx Q_2 \approx q_*$**

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q_1(s, a)$ and $Q_2(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, such that $Q(\mathit{terminal}, \cdot) = 0$

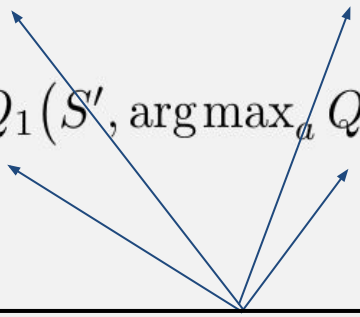
Loop for each episode:
  Initialize $S$
  Loop for each step of episode:
    Choose $A$ from $S$ using the policy $\varepsilon$-greedy in $Q_1 + Q_2$
    Take action $A$, observe $R$, $S'$
    With 0.5 probabilility:

$$Q_1(S, A) \leftarrow Q_1(S, A) + \alpha\Big(R + \gamma Q_2\big(S', \arg\max_a Q_1(S', a)\big) - Q_1(S, A)\Big)$$

    else:

$$Q_2(S, A) \leftarrow Q_2(S, A) + \alpha\Big(R + \gamma Q_1\big(S', \arg\max_a Q_2(S', a)\big) - Q_2(S, A)\Big)$$

    $S \leftarrow S'$
  until $S$ is terminal



Different Q functions for selecting an action and estimating its value


# Extending to Double DQN

- Add replay memory and target network
- Make the Q-network a Neural Network
- Add two more networks (as in double Q-learning), one each for the online network and the target network to prevent overestimation????


# Extending to Double DQN

- Add replay memory and target network
- Make the Q-network a Neural Network
- Add two more networks (as in double Q-learning), one each for the online network and the target network to prevent overestimation????

We can use the target network to estimate the value while the online network is used for selecting the action!

# Extending to Double DQN

Unlike the Q-learning algorithm, where both the networks are updated with equal probability, in Double DQN, only the online network will be updated since the target network is updated towards online network as according to the DQN algorithm.

The equation for the target is therefore:

$$Y_t^{\text{DoubleDQN}} \equiv R_{t+1} + \gamma Q(S_{t+1}, \underset{a}{\text{argmax}}\, Q(S_{t+1}, a; \boldsymbol{\theta}_t), \boldsymbol{\theta}_t^-)$$

Online network parameters

Target network parameters

# Prioritized Experience Replay

Idea: Use more important samples more often to converge faster.

# Prioritized Experience Replay

Idea: Use more important samples more often to converge faster.

Prob. of sampling transition $i$

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$$

Where

- $p_i$ is the priority of transition $i$
- $\alpha$ determines the prioritisation

# Prioritized Experience Replay

Idea: Use more important samples more often to converge faster.

Prob. of sampling transition $i$

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$$

Note: $\alpha = 1$ is the default uniform case

Where

- $p_i$ is the priority of transition $i$
- $\alpha$ determines the prioritisation

# Prioritized Experience Replay

Choices for $p_i$

Case 1: $p_i = |\delta_i| + \epsilon$

Where $\epsilon$ is a small positive constant to prevent $p_i$=0

Case 2: $p_i = \frac{1}{\text{rank}(i)}$

Where rank(i) is the rank of transition $i$ when the replay memory is sorted acc. to $|\delta i|$.

# Prioritized Experience Replay

Problem with *P(i):*

E(V) expects the updates to come from the same distribution as the samples but PER changes this distribution in an uncontrolled fashion => bias

# Prioritized Experience Replay

Problem with *P(i):*

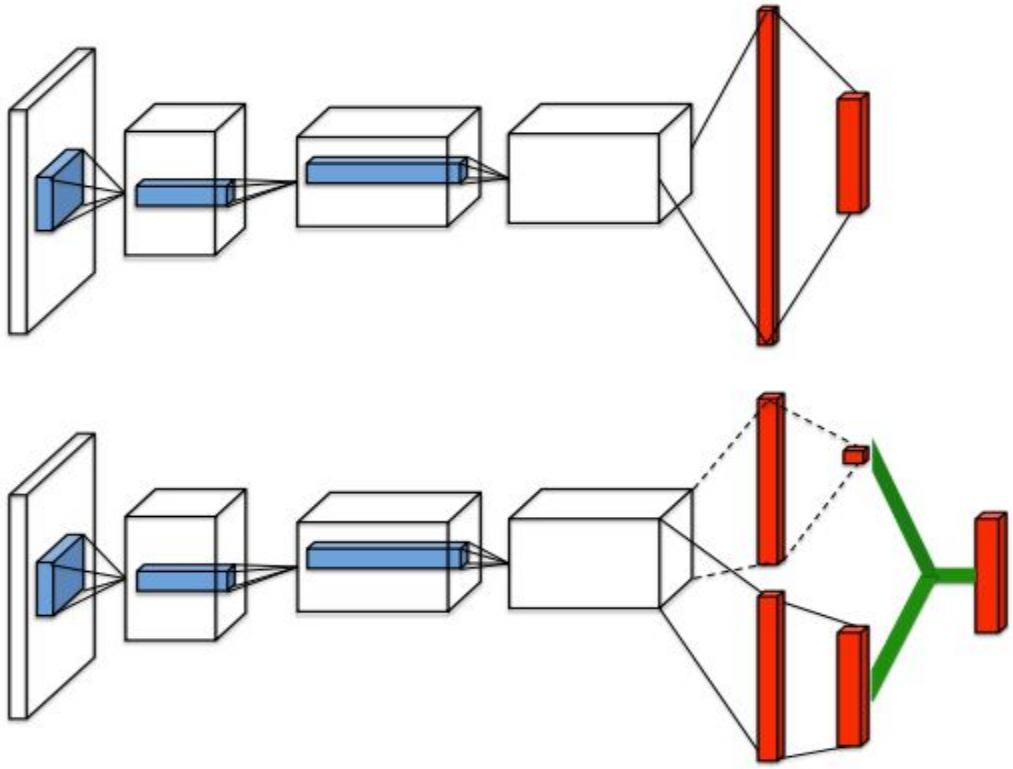
E(V) expects the updates to come from the same distribution as the samples but PER changes this distribution in an uncontrolled fashion => bias

Fixing the bias using *weighted* importance sampling:

$$w_i = \left( \frac{1}{N} \cdot \frac{1}{P(i)} \right)^{\beta}$$

Use $w_i \delta_i$ instead of $\delta_i$ in the Q-learning update

# Dueling Network Architecture



Source: [3]

*Figure 1.* A popular single stream $Q$-network (**top**) and the dueling $Q$-network (**bottom**). The dueling network has two streams to separately estimate (scalar) state-value and the advantages for each action; the green output module implements equation (9) to combine them. Both networks output $Q$-values for each action.


# Dueling Network Architecture

The dueling network outputs two streams:

1. Value function (V)
2. Advantage (A)

The aggregation of the two above gives us the Q function, so the network is essentially a Q-network. Thus, we can directly apply Q-learning techniques such as DDQN and PER with the dueling architecture.


# Dueling Network Architecture

Aggregation:

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + A(s, a; \theta, \alpha)$$

Where α and β are parameters for the two streams of the dueling network.

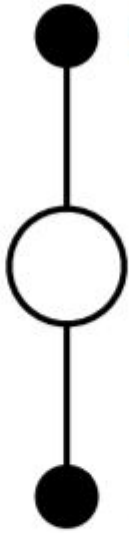
Does it work?

# Dueling Network Architecture

Aggregation:

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left( A(s,a;\theta,\alpha) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a';\theta,\alpha) \right)$$

# Recall SARSA



$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

# Recall SARSA

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

Problem:

$a_{t+1}$ introduces variance which slows convergence

# Expected SARSA

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \mathbb{E}_\pi[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}] - Q(S_t, A_t)\Big]$$
$$= Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\Big],$$

Using the expectation of Q value reduces the variance in the update => we can increase the α to increase the rate of learning.

# Expected SARSA

**Algorithm 1** Expected Sarsa

Initialize $Q(s, a)$ arbitrarily for all $s$,$a$
**loop** {over episodes}
  Initialize s
  **repeat** {for each step in the episode}
    choose $a$ from $s$ using policy $\pi$ derived from Q
    take action $a$, observe $r$ and $s'$
    $V_{s'} = \sum_a \pi(s', a) \cdot Q(s', a)$
    $Q(s, a) \leftarrow Q(s, a) + \alpha \left[ r + \gamma V_{s'} - Q(s, a) \right]$
    $s \leftarrow s'$
  **until** $s$ is terminal
**end loop**

Source: [4]


# References:

[1] DDQN https://arxiv.org/pdf/1509.06461.pdf

[2] PER https://arxiv.org/pdf/1511.05952.pdf

[3] Dueling https://arxiv.org/pdf/1511.06581.pdf

[4] Expected Sarsa http://www.cs.ox.ac.uk/people/shimon.whiteson/pubs/vanseijenadprl09.pdf