# Comparison of Double, Duelling and Prioritised DQN in Solving the Lunar Lander Problem

**Ziad Al-Ziadi** and **Behzad Javaheri**

## Introduction

Herein, a comparative study of DQN and two flavours (Duelling and Prioritised DQN) to solve the Lunar Lander problem was performed. We will provide a description and underlying equation to each algorithm and briefly describe their differences.

**Double DQN:** in a conventional DQN, the target value computed by one Q-value may overestimate the Q-value for the next state-action pair. Overestimation in most cases will not allow identification of the optimal action due to noise that is present in the estimated Q-value [21]. This is reported to occur primarily due to the maximisation steps that prefer overestimated values to be underestimated (Fig. 1). The max operator uses the same values to both select and evaluates action [1].

$$y = r + \gamma max_{a'} Q_\theta(s', a')$$

Figure 1. DQN Target Function

Double DQN resolves this issue by introducing an additional Q, in which this additional Q function is parameterised by the target network $\theta'$ used to compute the Q-value itself whilst the first $Q_\theta$ is utilised for evaluation of the greedy policy [18]. Ultimately, two Q-functions compute the target value as depicted in Figure 2.

$$y = r + \gamma Q_{\theta'}(s', argmax_{a'} Q_\theta(s', a'))$$

Figure 2. Double DQN Target Function

**Duelling DQN:** the notion of an *advantage function* is defined as the difference between the $Q^\pi$-function and the value-function $V^\pi$ allowing to measure the significance of each action. The architecture of the Duelling DQN consists of two separate streams; the first derives the value-function and the second the advantage-function (Fig. 3). Simply, for any state input, the value stream provides the value of a state and the advantage stream provides the advantage of all possible actions in a state [2].

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$$

Figure 3. Advantage Function

**Prioritised DQN:** The key idea revolves around the notion that an online RL network learns efficiently from certain transitions stored in its memory replay buffer, in particular those that are prioritised. Schual *et al*., employed this approach exploiting the temporal-difference (TD) $\delta$ error to prioritise some transitions. TD error $\delta$ is defined as the difference between the target and the predicted value (Fig. 4) [3].

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

Figure 4. TD Error

Essentially, the TD error acts as a proxy to measure how unexpected a given transition is and a transition with the largest TD error will be prioritised and replayed by the agent. Intuitively, the agent selects the transition with the highest TD error to learn better and therefore minimise the error. Simply, the agent will learn more from transitions with higher TD than those with lower TD errors and therefore assigns more priority lesser-known transitions. Transitions in Prioritised DQN is derived from $(s, a, r, s', p)$ where $p$ is the priority [3].

## Implementation

In our implementation, we define three respective replay buffers in which the Double and Duelling DQN share the same architecture. The replay buffer for Prioritised DQN has a

different architectural approach to incorporate the addition of the TD error $\delta$. Each model contains a greedy epsilon search in which the agent, under the influence of the given DQN model, will choose to explore with epsilon probability. The temporal different loss is calculated for each network to evaluate performance in which backwards propagation is used. All three models trained for 10,000 episodes.

Our data show an interesting relationship between each model and their respective loss values. The Double DQN achieves the highest reward of 4.27 with a relatively inconsistent pattern: we can see that the algorithm does not follow a distinct pattern. With respect to loss, the Double DQN achieves a relatively poor performance until ~6000 episodes. This error loss becomes apparent at the end episodes (Fig. 5, top two graphs)

The Dueling DQN model achieves a lower performance compared to Double DQN with a -81.29 reward at the end of training. Similarly, the loss score of the Dueling DQN performed inconsistently with the network achieving a consistent loss score after several thousand episodes.

Lastly, the Prioritised DQN model achieved the lowest reward out of all three networks with a score of -177.49. Nonetheless, the Prioritised DQN attained the lowest loss with a score close to zero achieved after ~ 500 episodes. The network maintained this low loss score until completion.

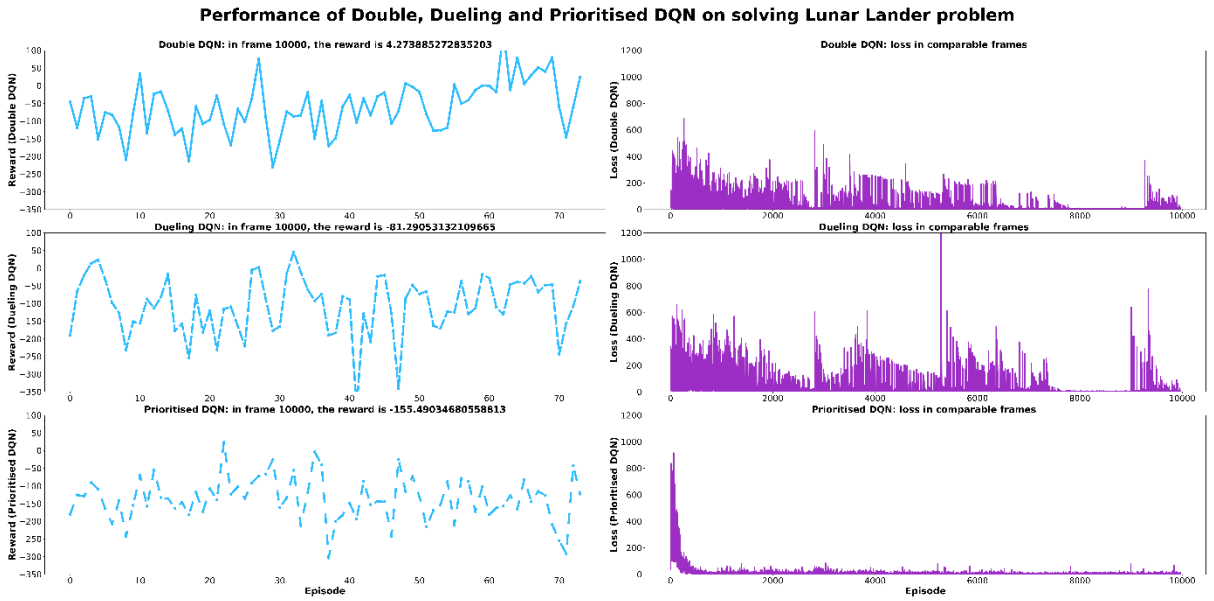


Figure 5. The performance of Double, Duelling and Prioritised DQN in solving the Lunar lander problem. The graphs on the left show the reward obtained for each algorithm in blue and the graphs on the right illustrate the loss for corresponding implementations.

**Conclusions**

Ultimately, as far as rewards are concerned, the Double DQN model performs better than both, Duelling and Prioritised DQN, however with costly and unstable loss score. Our Prioritised DQN model achieved the lowest reward score whilst maintaining a stable and low loss score. Further refining of the hyperparameters may allow reaching better performance for the algorithms used in this task.

## References


[1] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning. CoRR abs/1509.06461 (2015)," *arXiv preprint arXiv:1509.06461,* 2015.

[2] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*, 2016: PMLR, pp. 1995-2003.

[3] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952,* 2015.