# PLAYING LUNAR LANDER WITH DEEP REINFORCEMENT LEARNING

**Carlos Souza***
Georgia Institute of Technology
São Paulo, SP, Brazil
souza@gatech.edu

June 28th, 2019

**ABSTRACT**

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

***K*eywords** Reinforcement learning · Deep Reinforcement Learning · Deep Q-Networks

## 1 Introduction

Reinforcement learning (RL) theory provides mathematical background and classic algorithms to enable agents to learn optimally control an environment. However, to successfully apply those algorithms in problems with real-world complexity, several challenges must be tackled, must notably i) how to derive efficient representations of the environment from high-dimensional sensory inputs, and ii) how to use these to generalize past experience to new situation (Mnih et al. 2015).

Recent advances in deep neural networks (DNNs) originated a new type of agent, known as Deep Q-Network agent, capable of overcoming these challenges, which enabled RL algorithms to perform effectively. All the many recent successes in applying RL to complex sequential decision-making problems were kick-started by the Deep Q-Network algorithm (DQN; Mnih et al. 2015). Since then, many extensions have been proposed to improve its stability and/or speed. Double DQN (DDQN; Hasselt, Guez, and Silver 2016) addresses the problem of overestimation of action values, by decomposing the max operation in the target into action selection and action evaluation. The Dueling Network architecture (Dueling DDQN; Wang et al. 2016) better generalize learning accross actions by proposing an architecture that explicitly separates the representation of state values and (state-dependent) action advantages. Prioritized Experience Replay (PER; Schaul et al. 2016) improves data efficiency, by replaying more often transitions from which there is more to learn.

Each of these algorithms enables substantial performance improvements in isolation and combined, since they build on a shared framework. In this paper we propose to study an agent that combines all the aforementioned improvements, exploring its performance in Lunar Lander environment from OpenAI gym (Brockman et al. 2016).

*Latest Git hash: INCLUDE LATEST GITHASH

### 1.1 Background

In the standard reinforcement learning setting, agent interacts with an environment over discrete time steps. At each time step $t$, the agent perceives a state $s_t \in \mathcal{S}$, chooses an action from a discrete set $a_t \in \mathcal{A}$ accordingly to a policy $\pi$, where $\pi$ is a mapping from state space $\mathcal{S}$ to action space $\mathcal{A}$, and observes a reward $r_t$ and a next state $s_{t+1}$. This process continues until the agent reaches a terminal state, after which the process is restarted. The purpose of the agent is to maximize the expected discounted reward $R_t = \sum_{k=1}^{\infty} \gamma^k r_{k+t}$, where $\gamma \in (0, 1]$ is the discount factor that trades-off the importance of immediate vs future rewards. This interaction between the agent and the environment is formalized as a *Markov Decision Process* (MDP), described by $\langle \mathcal{S}, \mathcal{A}, T, r, \gamma \rangle$ tuple, where $T(s, a, s') = P[s_{t+1} = s' | s_t = s, a_t = a]$ is the stochastic transition function.

For an agent following a stochastic policy $\pi$, the true value of an action $a$ in a state $s$, and the value of that state $s$ are:

$$Q^{\pi}(s, a) = \mathbb{E}[R_t | s_t = s, a_t = a], \text{and} \tag{1}$$

$$V^{\pi}(s) = \mathbb{E}_{a \sim \pi(s)}[Q^{\pi}(s, a)] \tag{2}$$

The optimal state-value function is defined as $Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$. Under deterministic policy $a = \arg\max_{a' \in \mathcal{A}} Q^*(s, a')$, it follows that $V^* = \max_a Q^*(s, a)$. Also, the optimal state-value function $Q$ satisfies the Bellman equation:

$$Q^*(s, a) = \mathbb{E}_{s'} \left[ r + \gamma \max_{a'} Q^*(s', a') \mid s, a \right] \tag{3}$$

A common way of deriving a new policy from a state-action value function is to act $\epsilon$-greedly with respect to the action values, i.e. taking the action with the highest value with probability $(1 - \epsilon)$ or otherwise acting randomly with probability $\epsilon$. The optimal policy is easily derived from the optimal state-action value function by selecting the highest-valued action in each state.

One of the early breakthroughs in RL was the development of an off-policy Temporal Difference control algorithm known as *Q-learning* (Watkins 1989), defined by:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t) \right], \tag{4}$$

where $\alpha$ is a constant step-size parameter, or learning rate. The learned action-value function $Q$ directly approximates the *optimal* action-value function $Q^*$, independent of the policy being followed.

However, large state and/or action spaces make it tractable to learn $Q$ value estimates for each state-action pairs independently. To solve this challenge, DQN (Mnih et al. 2015) successfully used deep neural networks to approximate the state-action value function as $Q(s, a; \theta)$, where $\theta$ are the parameters of the network. For an $n$-dimensional state space and an action space containing $m$ actions, the neural network is a function from $\mathbb{R}^n$ to $\mathbb{R}^m$.

Two important components of the DQN algorithm proposed by Mnih et al. 2015 are i) the use of a target network, and ii) the use of experience replay. At each time step, the agent selects an action $\epsilon$-greedily with respect to the action values, and adds a transition $\langle s_t, a_t, r_t, s_{t+1} \rangle$ to the experience replay memory, that holds millions of experience tuples. The parameters of the neural network are then optimized by using stochastic gradient descent to minimize the loss function at iteration $i$:

$$L_i(\theta_i) = \mathbb{E}_{s,a,r,s'} \left[ (y_i - Q(s, a; \theta_i))^2 \right] \tag{5}$$

$$y_i = r + \gamma \max_{a'} Q(s', a'; \theta^-) \tag{6}$$

where $\theta^-$ are the parameters of a fixed and separate *target network*. The parameters of the target network $Q(s', a'; \theta^-)$ are frozen for a fixed number of iterations while updating the *online network* $Q(s, a; \theta_i)$ by gradient descent. The optimization is performed on mini-batches sampled uniformly at random from the experience replay memory. These two additions greatly improves the stability of the algorithm, leading to super-human performance on several Atari games.

### 1.2 DQN Extensions

**Double Q-learning.** The max operator in standard Q-learning and DQN uses the same values both to select and to evaluate an action. This makes it more likely to select overestimated values, resulting in overoptimistic value estimates. To prevent this, we can decouple the selection from the evaluation. This is the idea behind Double Q-learning. It is

possible to effectively combine this with DQN using the loss

$$L_i(\theta_i) = \mathbb{E}_{s,a,r,s'}\left[(y_i - Q(s,a;\theta_i))^2\right] \tag{7}$$

$$y_i = r + \gamma Q(s', \arg\max_{a'} Q(s', a'; \theta_i); \theta^-), \tag{8}$$

which was shown to improve performance by reducing value overestimates present in vanilla DQN.

**Dueling networks.** The dueling network is a DNN architecture designed for value based RL. The key insight behind it is that, for many states, it is unnecessary to estimate the value of each action choice. After the first DQN original layers, two streams of fully-connected layers are constructed such that they have the capability of providing separate estimates of the value and advantage functions, being the *advantage function* defined as $A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s)$. Finally, the two streams are combined to produce a single output $Q$ function

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s,a';\theta,\alpha)\right), \tag{9}$$

where $\theta$ denotes the parameters of the previous layers, while $\alpha$ and $\beta$ are the parameters of the two streams of fully-connected layers. As the dueling architecture shares the same input-output interface with standard Q networks, we can recycle all learning algorithms with Q networks to train the dueling architecture.

**Prioritized experience replay.** DQN samples uniformly from the replay memory. Their key idea behind prioritized experience replay is to increase the replay probability of experience tuples from which there is much to learn. Schaul et al. 2016 proposes the last encountered absolute *TD error* as a proxy for learning potential, which is the central component of the algorithm.

Several implementation challenges arise from the greedy TD-error prioritization and the bias it introduces by changing the distribution of experience samples. To overcome those, the algorithm implements *stochastic prioritization* and *weighted importance-sampling*.

Prioritized experience replay is shown to deliver both faster learning and better final policy quality across most games of the Atari benchmark suite, as compared to uniform experience replay.

### 1.3 Lunar Lander Environment

Rocket trajectory optimization is a classic topic in Optimal Control. OpenAI Gym proposes an environment where an agent has to control a "Lunar Lander" with the purpose of landing it successfully in a landing pad. The agent has four possible actions: do nothing, fire the left orientation engine, fire the main engine, or fire the right orientation engine. The state space is a 8-dimensional continuous space: at each time step, environment returns Lunar Lander position (x, y), velocities (horizontal and vertical), angle, angular velocity, and whether left/right legs are in contact with the ground. Landing pad coordinates are (0,0). Reward for moving from the top of the screen to landing pad and zero speed is about 100..140 points. If lander moves away from landing pad it loses reward back. Episode finishes if the lander crashes or comes to rest, receiving additional -100 or +100 points. Each leg ground contact is +10. Firing main engine is -0.3 points each frame. Firing side engine is -0.03 points each frame. The problem is considered to be solved when the agent achieves 200 score over past 100-episodes rolling average.

## 2 Methods

To investigate how the integrated agent that combines all described improvements perform, using Lunar Lander environment as test-bed, we propose 5 different experiments detailed below.

**Experiment 1: Impact of DQN extensions.** We start our study by analyzing the impact of each DQN extension, comparing all possible combinations. The simplest agent implements vanilla DQN, while the most sophisticated, the integrated agent, combines all extensions described earlier. We also compared these with agents containing only one and/or two extensions, to understand the impact of each extension on their performance.

While varying the extensions, we kept all other hyper-parameters frozen. We use RMSProp as the optimization algorithm, with $\alpha$ (learning rate) of $0.0001$. $\gamma$ (discount factor) is set at $0.99$. Replay memory capacity is set to $10,000$ experience tuples, and training is performed using 32 as mini-batch size. The network architecture consists of 2 fully-connected hidden layers with 512 neurons each, both with rectified linear unit (ReLU) activation functions (for

dueling network: we added a fully-connected layer as described previously). For prioritized replay, we annealed the bias by progressively increasing $\beta$ (exponent in importance-sampling weights) from $0.4$ to $1$. As exploration strategy, we decayed $\epsilon$ following $\epsilon(t) = e^{-t/70}$, where $t$ is the current episode. If $\epsilon$ falls below $0.05$, it stays constant at $0.05$ from that point on.

**Experiment 2: Effect of different gradient descent optimization algorithms and learning rates.** Next, we assess the effect of different optimization algorithms and their optimal learning rates on the integrated agent, which combines all DQN extensions. We simulate RMSProp, Adaptive Moment Estimation (Adam), and Stochastic Gradient Descent (SGD). For those 3 different algorithms, we test 30 different $\alpha$ (learning rates), evenly spaced on a log scale from $10^{-6}$ to $10^{-1}$. All other hyper-parameters are kept frozen as described in experiment 1.

**Experiment 3: Discount factor sensitivity.** In this experiment, we investigate the effect of $\gamma$ in the integrated agent's performance. We test all $\gamma \in [0.8, 0.85, 0.9, 0.92, 0.94, 0.96, 0.98, 0.99, 1.0]$. As usual, all other hyper-parameters are kept frozen.

**Experiment 4: Evaluation of different exploration strategies.** Following, we evaluate the impact of 10 different $\epsilon$ decay curves. We assess two types of curves: exponential decay functions, $\epsilon(t) = e^{-kt}$; and inverted logistic functions, $\epsilon(t) = 1/(1 + e^{-k_1(t-k_2)})$. Again, all other hyper-parameters are kept frozen.

**Experiment 5: Effect of kick-starting replay memory with human experience.** Finally, we investigate how kick-starting the replay memory with human experience can speed-up agent learning. To do it, we generate experience tuples $\langle s, a, s', r \rangle$ by playing Lunar Lander by hand, and use this data to pre-populate the replay memory. Then, we set our integrated agent to start learning procedure, with $\epsilon$ constant at $0.05$.

## 3 Results

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa. Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu,

sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa. Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer. Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

## 4 Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## References

[1] C. J. C. H. Watkins. “Learning from Delayed Rewards”. PhD thesis. King’s College, Oxford, 1989.

[2] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 00280836. URL: http://dx.doi.org/10.1038/nature14236.

[3] Greg Brockman et al. *OpenAI Gym*. cite arxiv:1606.01540. 2016. URL: http://arxiv.org/abs/1606.01540.

[4] Hado van Hasselt, Arthur Guez, and David Silver. "Deep Reinforcement Learning with Double Q-Learning". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, 2016, pp. 2094–2100. URL: http://dl.acm.org/citation.cfm?id=3016100.3016191.

[5] Tom Schaul et al. "Prioritized Experience Replay". In: *CoRR* abs/1511.05952 (2016).

[6] Ziyu Wang et al. "Dueling Network Architectures for Deep Reinforcement Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1995–2003. URL: http://proceedings.mlr.press/v48/wangf16.html.