Applying Deep Q-Learning to Atari Games

Harsha Vardhan Reddy Bonthu
Anudeep Kumar Reddy Guntaka
Bharath Kumar Reddy Kusuluru
Siddhaarth Mandava

Introduction

- Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention.
- Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization.
- In this project we demonstrate Deep Q-learning in the Arcade Learning Environment (ALE), a challenging framework composed of dozens of Atari 2600 games used to evaluate general competency in AI.

Reinforcement Learning

- The fundamental challenge in today’s artificial intelligence and machine learning is learning to make good decisions under uncertainty.

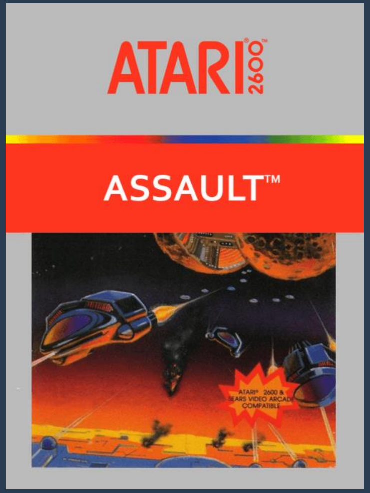
- Reinforcement learning is a machine learning training method based on rewarding desired behaviors and/or punishing undesired ones.

- In this type of learning the algorithm learns a policy on how to act in given situation and every action has some impact on the environment which in turn provides rewards to the algorithm to guide it and make better decisions.

Atari Games

- Atari Games Corporation has created many loved arcade games such as Space Invaders, Pong, Missile Command, Pitfall, etc.
- We are going to perform our Reinforcement Learning on the frames of the following game:

<u>Assault:</u> Assault is a 1983 fixed shooter video game developed and published by Bomb for the Atari 2600



Methods

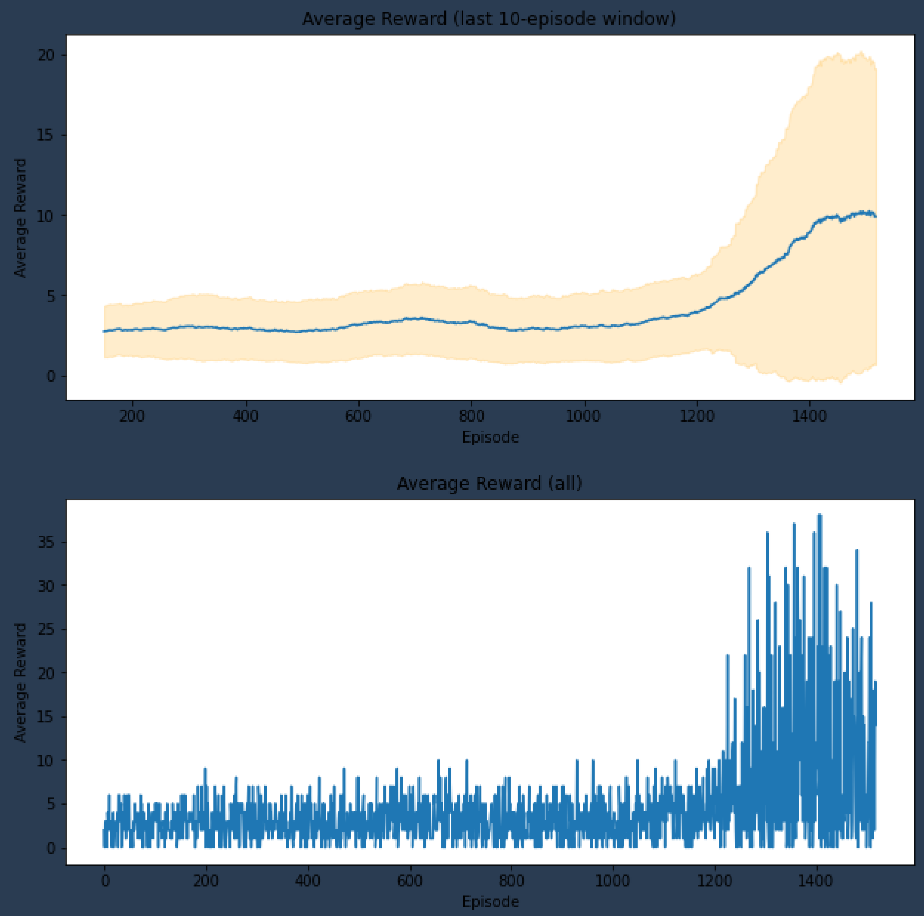
Deep Q Learning

- The core differences between Deep Q learning and vanilla Q learning is the implementation of the Q-table.
- This type of learning replaces the regular Q-table with the neural network to map input states to (‘action’, ‘Q value’) pairs.
- The initial state is fed into the neural network and it returns the Q-value of all the possible actions as an output
- Basically all the past experience is stored in the memory and the next action is determined by the maximum output of the Q-network. The loss function here is the mean-squared error of the predicted Q-value and target Q-value(Q*).

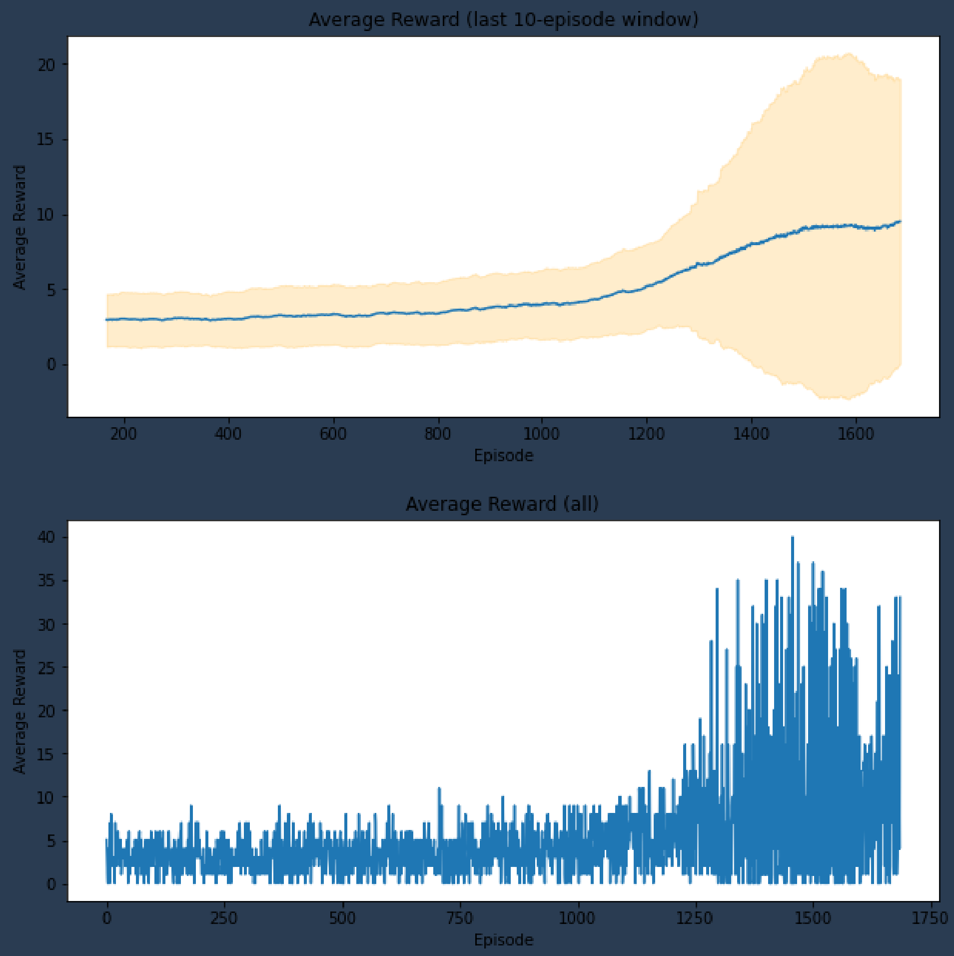
Double Deep Q Learning

- This idea was postulated in the paper “Deep Reinforcement Learning with Double Q-Learning” by H. van Hasselt and his team.
- The idea of Double Q-learning is to reduce overestimations by decomposing the max operation in the target into action selection and action evaluation.
- DDQN maintains two Q-value functions QA and QB, each one gets update from the other for the next state. The update consists of finding the action a* that maximises QA in the next state (Q(s’, a*) = Max Q(s’, a)), then use a* to get the value of QB(s’, a*) in order to update QA(s, a)

Results and Analysis



Rewards vs Episodes : DQN

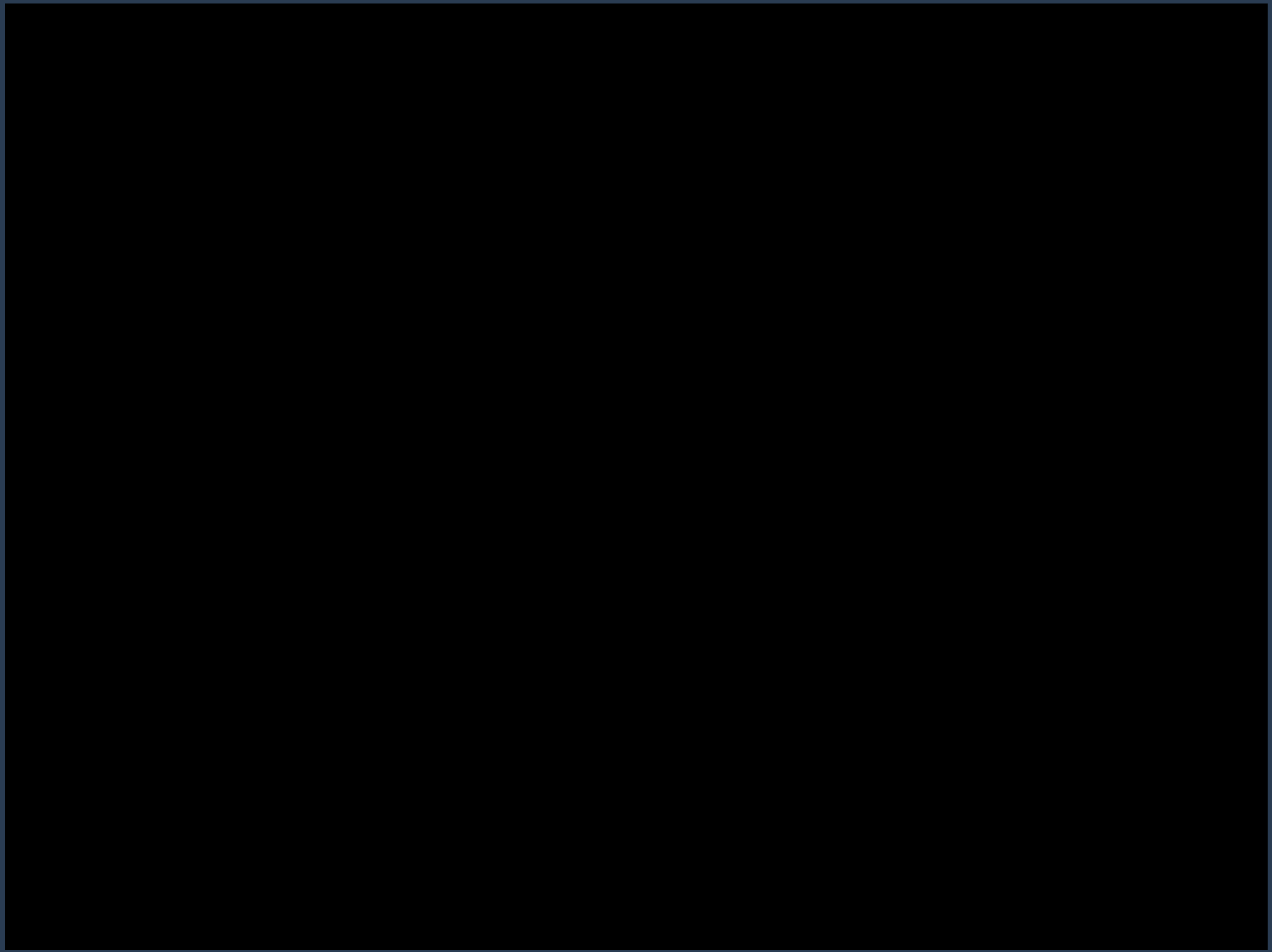


Rewards vs Episodes : DDQN

Rewards: DQN vs DDQN

Model	**Reward**	**Mean**
DQN Network	52.00	205.4
DDQN Network	124.00	252.21

Before Training



After Training

Thank You!