Fundamentals of AI: Project Proposal

Who are we?	NUID
Mario Anson	002931419
Aditya Shanmugham	002738073

Problem Description:

a) **Problem to solve:**
To explore the advantages of using Deep-Q-Learning over Q-learning in terms of runtime and space complexity. To produce a rule-of-thumb on the scenarios where the Deep Q learning can be used over the Q-learning.

b) **Description: Who is Mario?**
We are planning to explore the realm of Reinforcement learning by using the Super Mario Game (originally released for NES) with the help of simulator.

Fig: Mario realizing that Wario Cheats

By empowering Mario with the supernatural power of Reinforcement learning, we can empower his quality of actions and his love for princess peach. By providing the ability to remember his actions and outcome of his earlier life, we empower super Mario to take calculated risks to reach his goal.

Since Mario lives in a beautiful vast world with many possibilities where he can explore (***states***) though he can move in only four directions (***actions***) there are so many possibilities for him to consider (***state-action table***). Once he obtains the power of learning from his mistakes in his past life (***Q-Learning),*** we will help him to remember better using Supernatural Power v2 ***(Deep Q-Learning***), where we use a neural network to draw an action using various parameters: Current State, Mario's Enemies, Power-ups, Princess Peach Position (Final State).

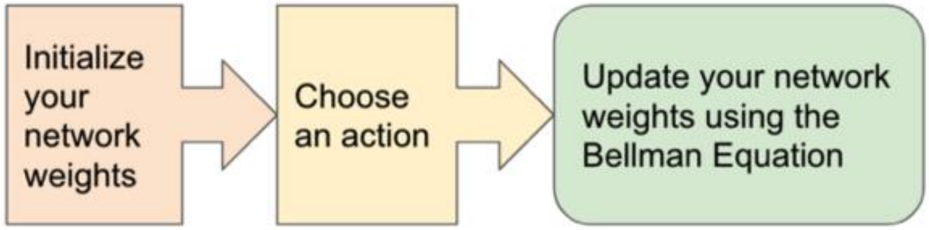
c) Computationally Speaking...:

We are planning to compare the computational efficiency of Q-Learning vs Deep Q-Learning using the Super Mario game.

- **Q-Learning:** By initializing the Q-table with constant values, we see various lifecycles of Super Mario playing the game. The actions are chosen based on the exploration-exploitation (epsilon greedy exploitation strategy). Once the action is taken, the Q-table is filled using the Bellman Equation.

$$Q(S_t, A_t) = (1 - \alpha)\, Q(S_t, A_t) + \alpha * (R_t + \lambda * max_a\, Q(S_{t+1}, a))$$

- **Deep Q-Learning:** In Deep Q-Learning, we perfect the computational complexity of the Q-learning by using a Deep Neural network instead of a state-action table to select an action.


Initialize your network weights
Choose an action
Update your network weights using the Bellman Equation

The algorithm initializes Main and Target Neural Networks, chooses an action based on the exploration and exploitation strategy (Epsilon Greedy Exploration Strategy) and finally updates the network weights using the Bellman Equations.

$$(R_t + \lambda * \; max_a\, Q(S_{t+1}, a))$$

d) What about the I/O?

We will be running the Super Mario game using the open-ai gym wrapper library for Super Mario on the *NES_PY* library (simulator) where we take the attributes of the game states (pos_x, pos_y, reward function, enemy position, etc.) and the output would be the Boolean check whether the current state matches the final state of the game.

e) Is it Interesting though?

The whole agenda behind our project is to learn the necessity of Deep Q-learning in the land of reinforcement learning. What does it bring to the table? What problems does it solve? We believe that this project will help us to understand the significance of exploration-exploitation factor, mathematical significance of discount factor, solving the problems we face while encoding the state-action table in terms of neural network.

On hindsight, it is always fascinating to watch an algorithm (created by us) which plays a game (created by us) according to the rules defined (created by us) which scores higher than us.

f) Data Source:

The data we are going to use is a simulator from the nes-py emulator using open ai. It is a python library that supplies an environment emulator for Super Mario Bros which will help us to implement the algorithms.

- Source Link: https://pypi.org/project/gym-super-mario-bros/
- Open AI source: https://github.com/openai/gym
- NES Emulator: https://github.com/Kautenja/nes-py

Approach / Algorithms

a) Algorithms

The following are the algorithms that will be implemented

- Q-Learning
- Deep-Q-Learning
- Q-Learning VS Deep-Q-Learning Analysis

b) Approach

- Implement the two algorithms in the Mario Bros environment.
- Run the simulation of Mario Bros using both the algorithms.
- Analyze both the time and space complexity of the two algorithms.
- Find insights about when one algorithm performs better over the other and create a report about it.

The comparison we have planned is to understand the space/time complexity/limitation of one algorithm over the other.

Work partition

Task	Member
Implementation of Q-Learning	Aditya Shanmugham
Implementation of Deep-Q-Learning	Maria Anson
Comparison of Q-Learning vs Deep-Q-Learning	Aditya Shanmugham, Maria Anson
Project Report Draft	Aditya Shanmugham, Maria Anson

Prediction of the Future:

If all the above-mentioned scope is achieved, we will additionally compare the limitation of an adversarial search over reinforcement learning.