

Reinforcement Learning Approach to Play Flappy Bird

What Does It Take for an AI to Learn to Beat a Score of 100 in Flappy Bird?

Project group members:

A. Törö., Everything

Introduction

Flappy Bird is a simple yet challenging arcade game in which the player controls a bird that must navigate between moving pipes. Despite the game looking easy, the game's non-intuitive controls make it a tough task for humans and an interesting challenge for AI.

Motivation and Challenges:

- Develop an understanding of reinforcement learning algorithms through a fun and interactive project.
- A perfect use case for RL, requiring the agent to learn precise control over the bird's movement while adapting to moving level layouts in real time.
- Building a Flappy Bird environment where a reinforcement learning model can continuously improve through trial and error.

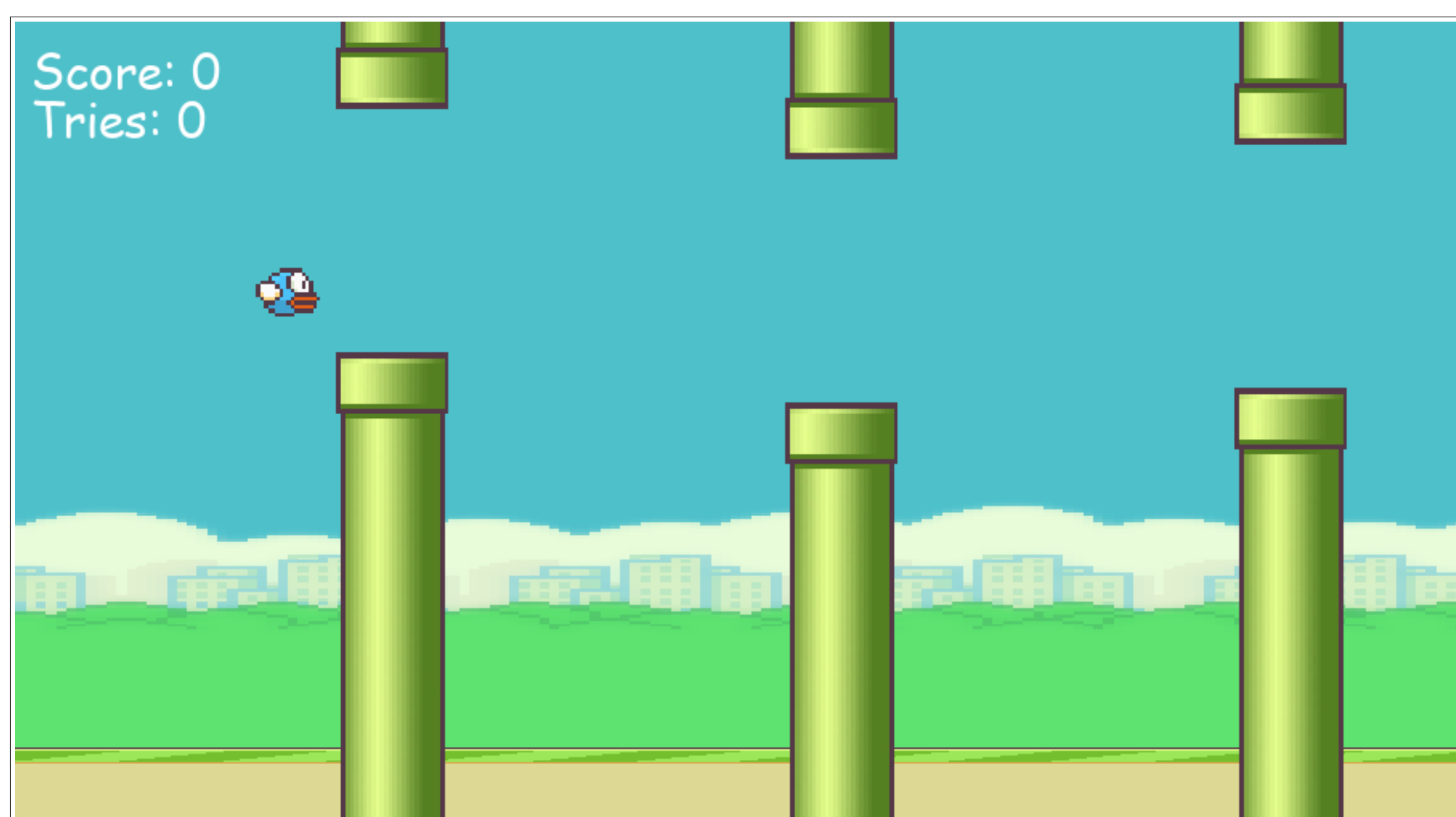


Figure 1: Flappy Bird game environment.

Methods

The agent learns to play in the environment using a deep Q-learning algorithm (DQN) [Dut18]. The training process involves three main steps:

1. **Collecting states and actions:** The agent observes the environment (pipe's and bird's positions) and takes actions (jump or not to jump).
2. **Training the network:** The agent trains a neural network to approximate the Q-value function, guiding its decisions.
3. **Evaluating performance:** The agent's performance is tested by playing the game and scoring its ability to achieve high scores without crashing.

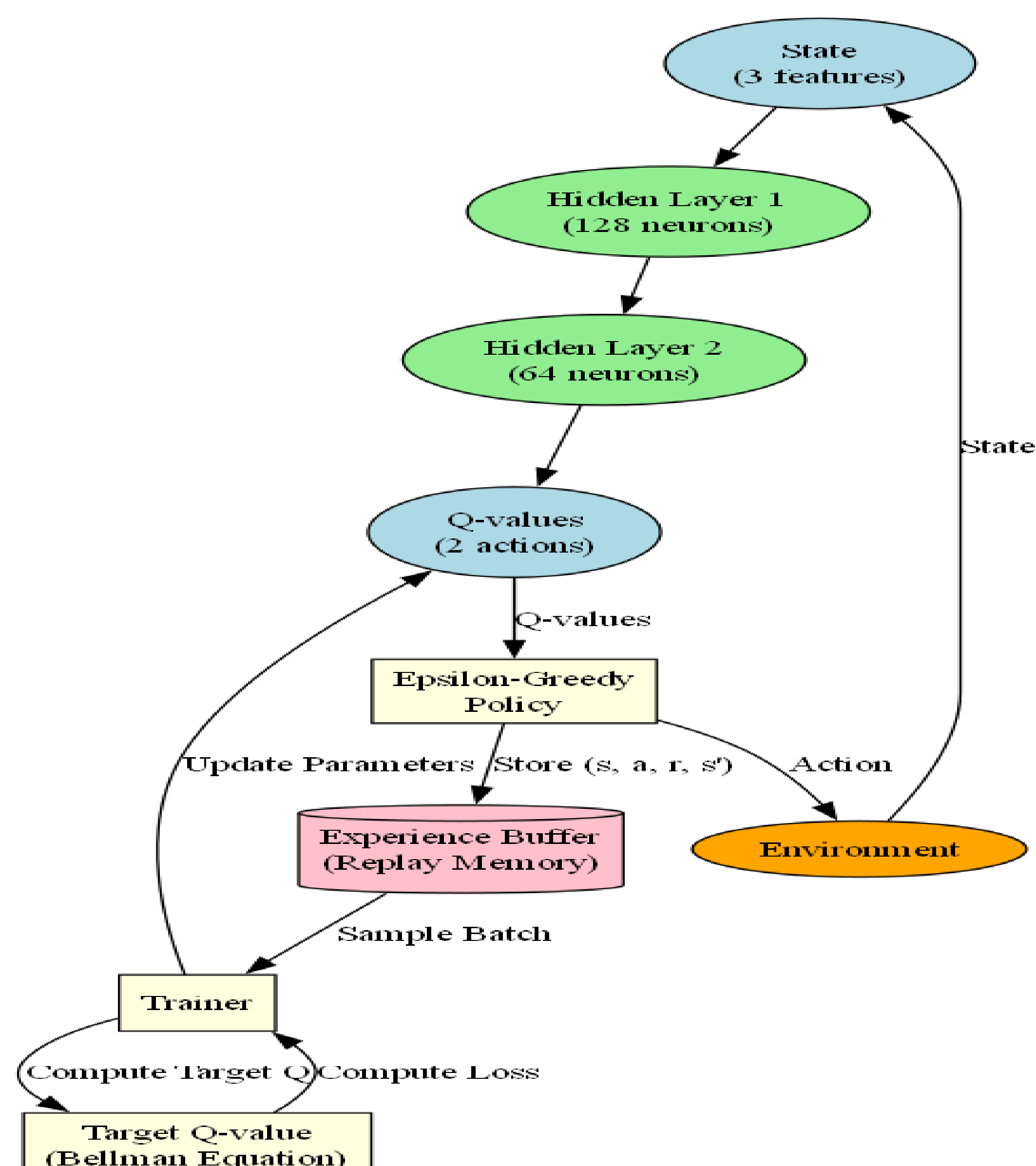


Figure 2: Method schematic showing the DQN process.

Applications

The agent was trained for 250 iterations, or until it achieved two consecutive games with scores over 100 points. Given the simple action space of Flappy Bird, the algorithm converges relatively quickly. During training, the agent optimizes its decision-making by balancing exploration and exploitation of high-reward actions.

The following hyperparameters were used:

- Learning rate: 0.001
- γ : 0.95
- Batch size: 128
- Replay buffer size: 20 000
- ϵ : decay from 1.0 to 0.1 over time.

However, the agent may diverge, resulting in suboptimal behavior as shown in Table 1

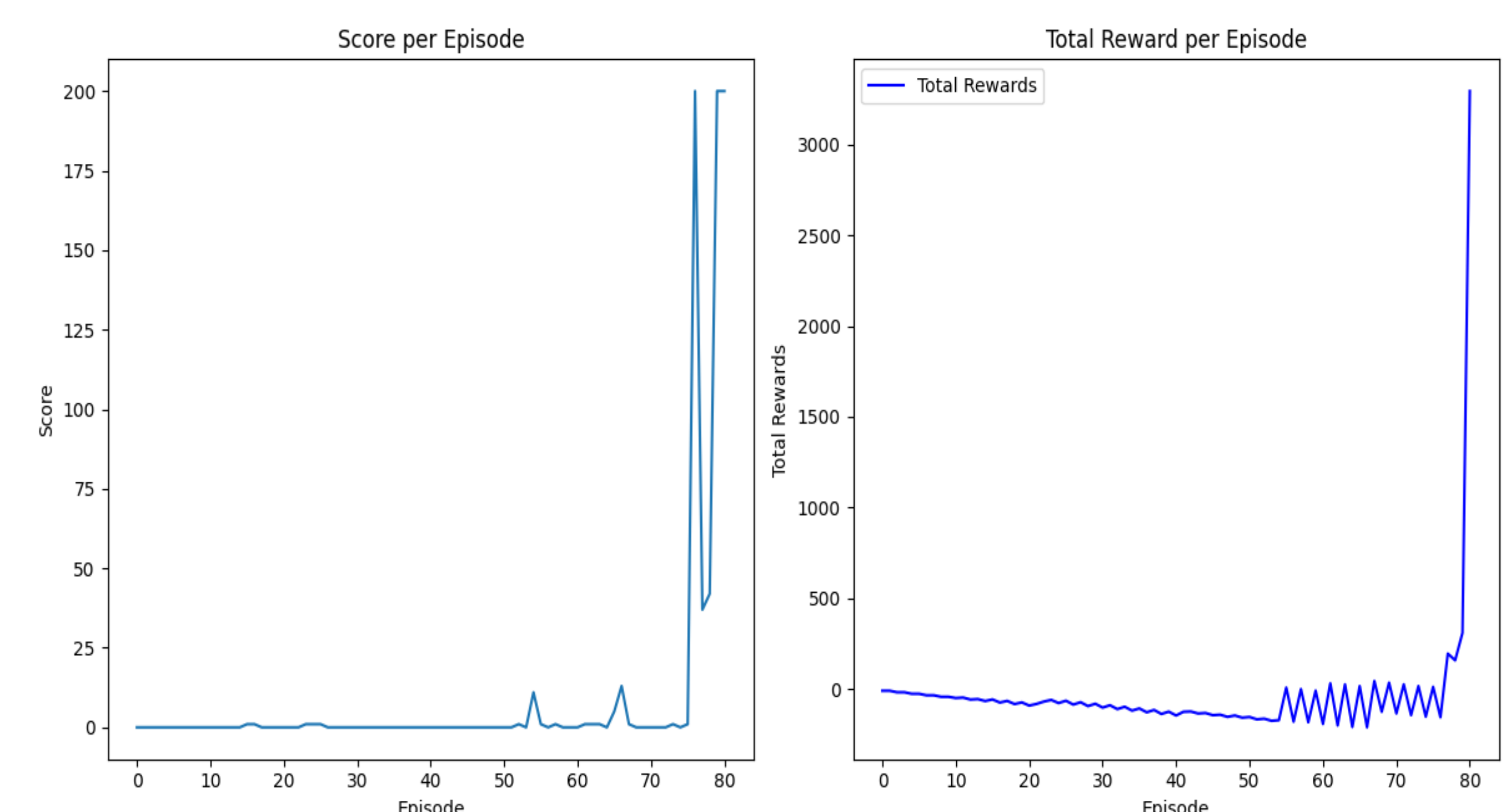


Figure 3: Agent performance: Scores and rewards over training iterations.

Table 1: Training results: Iterations required for the agent to reach 100 points or fail (i.e., 250 iterations).

Training Session	Iterations to Reach 100 Points or Fail
1	99
2	250
3	250
4	90
5	250
6	250
7	105
8	95
9	94
10	97
Average	158

Conclusions

- Results for this case study show that it took on average 158 iterations for an AI to learn to fly between 100 pipes in Flappy Bird
- Flappy Bird is challenging game but not a match for modern reinforcement learning algorithms.
- The agent converged quickly to a optimal solution about 60% of the time, other times it failed miserably.
- Future work should focus on fining the algorithm by optimizing the hyperparameters.

References

- S. Dutta, *Reinforcement learning with tensorflow: A beginner's guide to designing self-learning systems with tensorflow and openai gym*, 1st ed., Packt Publishing, Birmingham, 2018.