What does it take for an AI to beat a score of 100 in Flappy Bird?

A Case Study on Flappy Bird AI

Arno Törö

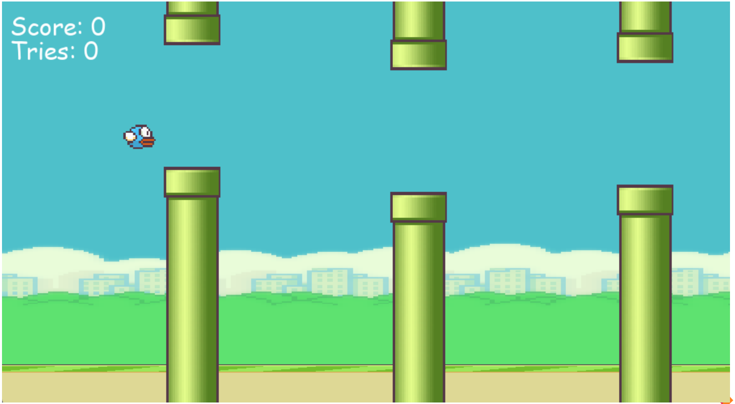
LUT University

December 11, 2024

1. Introduction
2. Methodology
3. Results
4. Conclusion

Introduction

- Flappy Bird was a popular game in 2014 and it became infamous for its simple but challenging gameplay.



Introduction

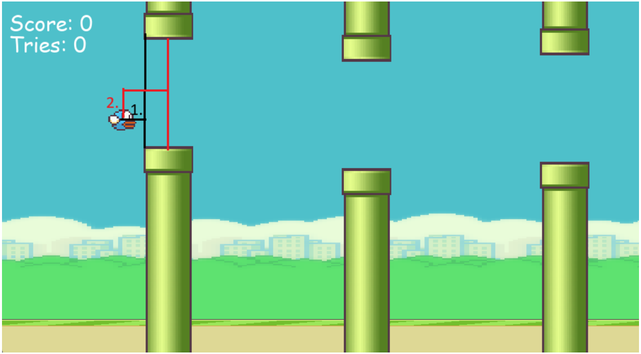
- The motivation for this case study was to have an intuitive but relatively simple problem and try to develop an AI agent to solve it.
- The agent is trained in the game environment through trial and error, using a reinforcement learning algorithm called deep Q-learning algorithm.

Environment setup

- Created a game environment of Flappy Bird in Python, which handles the game logic (collisions, rendering more pipes, etc.).
- There are two possible actions in the game: flap or don't flap
- Return the game state information of each frame to the agent.

Information to agent

- Information gathered for the agent:
  1. Horizontal distance to next pipe
  2. Vertical distance to next center of pipe gap
  3. Bird's current speed

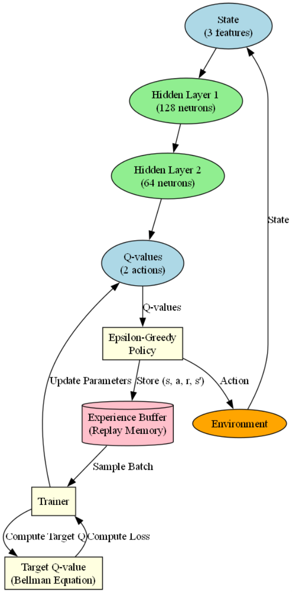


Deep Q-learning algorithm

- Utilized PyTorch
- Q-learning calculates a value for each possible action in a state.
- Deep Q-learning (DQN) uses neural networks to approximate Q-values for each possible action at given state.
- DQNs often utilize CNNs to approximate Q-values, but not used in this study.
- ϵ-greedy policy
  - Balance exploration and exploitation to prevent suboptimal results

$$\text{Action at time(t)} \begin{cases} \max Q_t(a) & \text{with probability } 1-\epsilon \quad \text{exploitation} \\ \text{any action (a)} & \text{with probability } \epsilon \quad \text{exploration} \end{cases}$$

Deep Q-learning algorithm structure



Used hyperparameters

- Hyperparameters used in this study
  - Learning rate: 0.001
  - γ: 0.95
  - Batch size: 128
  - Replay buffer size: 20 000
  - ϵ: decay form 1.0 to 0.1 over time.
- Rewards given for the agent:
  - Surviving: 0.01
  - Passing a pipe: 1
  - Failing: -10

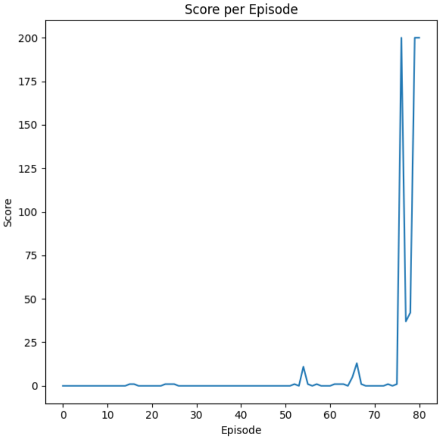
Results

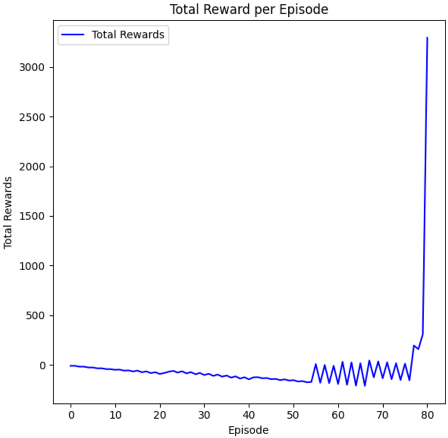
- Agent was trained for 250 iterations or until a score of 200 was achieved with two consecutive games.
- Training results of 10 agents:

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

Training session performance

- Agent's performance on one training session which lead to reaching a score of 100 points





ty

Conclusions

- Agent is capable of reaching 100 points of score rather quickly.
- Sometimes fails to converge to a solution, the first pipe pass is the most important one. Can be based on luck.
- Improvements in hyperparameters or learning algorithm structure could help overcome this issue.

Demovideo