COMP9444 Project Summary

Deep Q-learning for Flappy Bird

z5213002 Junming Cai z5295964 Yi Fu

z5212799 SiyuanTang z5259101 Jeffrey Seto

1. Introduction

Deep Reinforcement Learning is a type of machine learning technique in which an agent learns about the environment thus making decisions aiming to optimise the objectives. This report aims to implement the experience replay methodology of Deep Q-Learning into a flappy bird game, by studying the image of each frame and thus making decisions attempting to achieve the highest possible score. This report is just a summary of our project. A detailed explanation of our project is in train1.ipynb(main explanation) and train2.ipynb(additional classes explanation).

2. Methods

(Method chosen)Experience replay methodology of deep-q-learning is used in this report. Since the efficiency and effectiveness of experience replay can be affected by the variable learning rate, initial epsilon (percentage of choosing a random action at the beginning), memory size and sample size, we decided to test out 4 different scenarios, one base model, and 3 each with only 1 variable changed.

(Literature Review)We have researched and trained the DQN experience replay model(main model), DDQ (double Q model) and Q-learning model. Below is some of the useful links and references we used:

1. Oroojiooyjadid, A, Nazari, M, Snyder, L & Takac, M 2022, ‘A Deep Q-Network for the Beer Game: Deep Reinforcement Learning for Inventory Optimization’, *Manufacturing & service operations management*, vol. 24, no. 1, pp. 285–304.
2. Bui, Y-H, Hussain, A & Kim, H-M 2020, ‘Double Deep Q -Learning-Based Distributed Operation of Battery Energy Storage System Considering Uncertainties’, *IEEE transactions on smart grid*, vol. 11, no. 1, pp. 457–469.

3. Experimental Setup

a. Dataset for RL

There is no pre-existing data-set for our Reinforcement learning agent. Instead, our agent learns while playing the game and gets data generated from the environment as an image, studies the bird position, gap position, and pipe position. After performing an action, a new image, reward and a boolean of game terminated or not, is generated.

We used the game base from

https://github.com/uvipen/Flappy-bird-deep-Q-learning-pytorch, written by Uvipen.

b. Pre-process of image

To increase the efficiency of our model, we perform pre-processing to reduce the image dimension. We would extract images from the current state, and convert the resultant 512x288 pixels image into grayscale, then rescale it to 90x90 pixels and normalize it from [0, 1] to [0,255]. The resultant image is then used as the input for our network.

c. Game rules and game reward(The rewarding system of the game)

-	if bird is alive:	reward = 0.1
-	else if the bird is dead:	reward = -1
-	else if the bird passes through a pipe.	reward = 1

d. Hyper-parameter Choosing

Name	Leaning rate	initial epsilon	memory_size	discount_factor	sample size
Base Scenario	1E-5	0.2	50	0.99	30
Decrease learning rate(Best)	5E-6	0.2	50	0.99	30
Increase memory size	1E-5	0.2	100	0.99	85
Increase initial epsilon	1E-5	0.6	50	0.99	30

e. Evaluation Strategy (method used for examining in result eg. reward and its equation)

We used an average reward(per 100 episodes) and the time taken to train the model to evaluate how our agent performed, and we compared the performance between each model.

4. Results

Our result suggests that overall, experience replay is a good approach in training agents to learn the environment thus attempting to achieve objectives. Experience replay gets better rewards (than Q learning) and is more stable and thus faster (than double q) for training compared to the other two models, and can aim to achieve the highest possible score.

However, it is possible to further improve the performance by altering variables. In our scenario, we found out that by decreasing the learning rate, the average reward is increased significantly by roughly 4 times, while training time is not increased.

5. Conclusions

The experience replay overall is a good deep q-learning method in reinforced learning, agents can learn how to play the game in a short period of time, however, it comes with limitations that as the memory size increases, the time it takes to train the agent increases dramatically. Moreover, as each variable (Learning rate, initial epsilon, discount factor, memory size, batch size) affects the performance of the model, it requires multiple trials until we can obtain the best variable value, both in terms of maximum reward and timeliness. As a result, it is recommended that in the future, we could attempt to find a correlation between variables and input size, further experiment and solve issues found in other models, and also consider additional methods such as the advantage actor-critic method.