Flappy Bird Q-Learning Implementation

Similachi Andrei

January 12, 2025

1 Introduction

This report details the implementation of a Q-learning agent using deep neural networks to play Flappy Bird. The implementation uses raw pixel input which is preprocessed and fed to a CNN.

2 Architecture

2.1 Neural Network Structure

The neural network architecture consists of:

- Input Layer: 4 channels of 84x84 preprocessed frames
- Convolutional Layer 1: 32 filters of size 8x8 with stride 4
- Convolutional Layer 2: 64 filters of size 4x4 with stride 2
- Convolutional Layer 3: 64 filters of size 3x3 with stride 1
- Fully Connected Layer 1: 3136 $\rightarrow$ 512 neurons
- Output Layer: 512 $\rightarrow$ 2 neurons (representing actions)

ReLU activation functions are used throughout the network.

2.2 Input Processing

The raw game state is preprocessed by:

- Reshaping to 12x15 matrix
- Resizing to 84x84 pixels
- Normalizing pixel values
- Stacking 4 consecutive frames

3 Q-Learning Implementation

3.1 Key Parameters

- Discount Factor: 0.99
- Initial Epsilon: 0.2

- Final Epsilon: 0.0001
- Number of Iterations: 30,000
- Replay Memory Size: 10,000
- Minibatch Size: 32
- Learning Rate: 0.0001

3.2 Algorithm Components

- Experience Replay: Implemented using a deque with maximum size 10,000
- Epsilon-Greedy Policy: Linear decay from 0.2 to 0.0001
- Q-Value Updates: Using MSE loss and Adam optimizer

4 Training Process

The training process involves:

- Collecting experiences in replay memory
- Sampling random minibatches
- Computing target Q-values using the Bellman equation
- Updating network weights through backpropagation

The agent learns to map raw pixel inputs to Q-values for each possible action (flap or do nothing).

5 Results

5.1 Run 1: Training Parameters

The following parameters were used during training for Run 1:

- **Learning Rate (lr):** 0.0001
- **Number of Actions:** 2
- **Discount Factor :** 0.99
- **Initial Epsilon:** 0.2
- **Final Epsilon:** 0.0001
- **Number of Training Iterations:** 50,000
- **Replay Memory Size:** 10,000
- **Minibatch Size:** 32

5.2 Training Performance

Below is a snapshot of the training performance during Run 1:

Listing 1: Training Log (Run 1)

```
Episode: 760, Score: 23.600000000000268, Avg Score: 5.16, Epsilon: 0.0061, Best
Episode: 761, Score: -4.499999999999998, Avg Score: 5.10, Epsilon: 0.0059, Best
Checkpoint saved: Episode 762, Score 7.499999999999984
Episode: 762, Score: 7.499999999999984, Avg Score: 5.17, Epsilon: 0.0055, Best S
Checkpoint saved: Episode 763, Score 29.300000000000264
Episode: 763, Score: 29.300000000000264, Avg Score: 5.44, Epsilon: 0.0036, Best
Checkpoint saved: Episode 764, Score 15.799999999999962
Episode: 764, Score: 15.799999999999962, Avg Score: 5.50, Epsilon: 0.0029, Best
Checkpoint saved: Episode 766, Score 32.100000000000335
Episode: 766, Score: 32.100000000000335, Avg Score: 5.80, Epsilon: 0.0006, Best
```

The highest score achieved during training was **32.1**, with an average score of approximately **5.80** by the end of training. The agent's performance improved significantly as epsilon decayed, enabling it to exploit the learned policy.

5.3 Testing Performance

After training, the model was tested over 10 episodes. The results are as follows:

Listing 2: Testing Log (Run 2)

```
Test Episode 1: Score = 42.80000000000043
Test Episode 2: Score = 14.499999999999973
Test Episode 3: Score = 106.0999999999975
Test Episode 4: Score = 11.29999999999998
Test Episode 5: Score = 52.10000000000067
Test Episode 6: Score = 60.90000000000092
Test Episode 7: Score = 72.20000000000033
Test Episode 8: Score = 21.20000000000009
Test Episode 9: Score = 8.799999999999947
Test Episode 10: Score = 85.89999999999884
```

5.4 Performance Metrics

- **Average Test Score:** 47.58
- **Maximum Test Score:** 106.10
- **Minimum Test Score:** 8.80

The agent achieved an average score of **47.58** during testing, with a maximum score of **106.10**.

5.5 Run 2: Training Parameters

The following parameters were used during training for Run 2:

- **Learning Rate (lr):** 0.001
- **Number of Actions:** 2

- **Discount Factor :** 0.99
- **Initial Epsilon:** 0.2
- **Final Epsilon:** 0.0001
- **Number of Training Iterations:** 90,000
- **Replay Memory Size:** 10,000
- **Minibatch Size:** 32

5.6 Training Performance

Below is a snapshot of the training performance during Run 2:

Listing 3: Training Log (Run 2)

```
Episode: 1328, Score: 14.799999999999942, Avg Score: 7.90, Epsilon: 0.0048
Episode: 1329, Score: 43.10000000000034, Avg Score: 8.34, Epsilon: 0.0038
Episode: 1330, Score: 73.39999999999932, Avg Score: 9.00, Epsilon: 0.0012
Episode: 1331, Score: 5.999999999999993, Avg Score: 9.07, Epsilon: 0.0011
Episode: 1332, Score: 2.799999999999999, Avg Score: 9.06, Epsilon: 0.0010
Episode: 1333, Score: 26.700000000000156, Avg Score: 9.31, Epsilon: 0.0002

Model saved as: saved_models/model_ep_1333_avg_9.31_20250105_153524.pth
```

The highest score achieved during training was **73.40**, while the average score improved to **9.31** by the end of training.

5.7 Testing Performance

After training, the model was tested over 10 episodes. The results are as follows:

Listing 4: Testing Log (Run 2)

```
Test Episode 1: Score = 10.19999999999997
Test Episode 2: Score = 142.69999999999533
Test Episode 3: Score = 13.199999999999967
Test Episode 4: Score = 12.699999999999966
Test Episode 5: Score = 4.699999999999994
Test Episode 6: Score = 174.49999999999324
Test Episode 7: Score = 5.59999999999999
Test Episode 8: Score = 6.699999999999983
Test Episode 9: Score = 8.999999999999961
Test Episode 10: Score = 29.400000000000198
```

5.8 Performance Metrics

- **Average Test Score:** 40.87
- **Best Test Score:** 174.50
- **Worst Test Score:** 4.70

The model demonstrated moderate performance during testing, achieving an average score of **40.87**. The highest score obtained was **174.50**, while the lowest was **4.70**.

5.9 Run 3: Training Parameters

The following parameters were used during training for Run 3:

- **Learning Rate (lr):** 0.0001
- **Number of Actions:** 2
- **Discount Factor :** 0.99
- **Initial Epsilon:** 0.2
- **Final Epsilon:** 0.0001
- **Number of Training Iterations:** 12,000
- **Replay Memory Size:** 10,000
- **Minibatch Size:** 32

5.10 Training Performance

Below is a snapshot of the training performance during Run 3:

Listing 5: Training Log (Run 3)

```
Episode: 1719, Score: 12.699999999999948, Avg Score: 10.01, Epsilon: 0.0050
Episode: 1720, Score: 3.5999999999999988, Avg Score: 9.90, Epsilon: 0.0049
Episode: 1721, Score: 11.299999999999963, Avg Score: 10.02, Epsilon: 0.0047
Episode: 1722, Score: 33.60000000000022, Avg Score: 10.36, Epsilon: 0.0040
Episode: 1723, Score: 2.3, Avg Score: 10.36, Epsilon: 0.0038
Episode: 1724, Score: -2.3000000000000003, Avg Score: 10.33, Epsilon: 0.0037
Episode: 1725, Score: 20.400000000000134, Avg Score: 10.50, Epsilon: 0.0031
Episode: 1726, Score: 15.19999999999996, Avg Score: 10.61, Epsilon: 0.0028
Episode: 1727, Score: 4.69999999999999, Avg Score: 10.60, Epsilon: 0.0026
Episode: 1728, Score: 27.600000000000207, Avg Score: 10.74, Epsilon: 0.0019
Episode: 1729, Score: 41.30000000000043, Avg Score: 11.10, Epsilon: 0.0009
Episode: 1730, Score: 14.799999999999937, Avg Score: 11.20, Epsilon: 0.0005
Episode: 1731, Score: 9.099999999999971, Avg Score: 11.23, Epsilon: 0.0003

Model saved as: saved_models/model_ep_1731_avg_11.23_20250110_222552.pth
```

The highest score achieved during training was **41.30**, while the average score improved steadily to **11.23** by the end of training.

5.11 Testing Performance

After training, the model was tested over 10 episodes. The results are as follows:

Listing 6: Testing Log (Run 3)

```
Test Episode 1: Score = 196.199999999992
Test Episode 2: Score = 209.8999999999919
Test Episode 3: Score = 14.599999999999968
Test Episode 4: Score = 9.099999999999984
Test Episode 5: Score = 112.09999999999783
Test Episode 6: Score = 105.89999999999739
Test Episode 7: Score = 21.900000000000084
```

```
Test Episode 8: Score = 6.499999999999984
Test Episode 9: Score = 37.6000000000003
Test Episode 10: Score = 176.99999999999406
```

5.12 Performance Metrics

- **Average Test Score:** 89.08
- **Best Test Score:** 209.90
- **Worst Test Score:** 6.50

6 Code Structure

Key components of the implementation:

- Neural Network Class: Implements the CNN architecture
- Frame Preprocessing: Converts raw game state to suitable input
- Training Loop: Implements Q-learning with experience replay
- Testing Framework: Evaluates trained model performance