Reinforcement Learning – Project 3

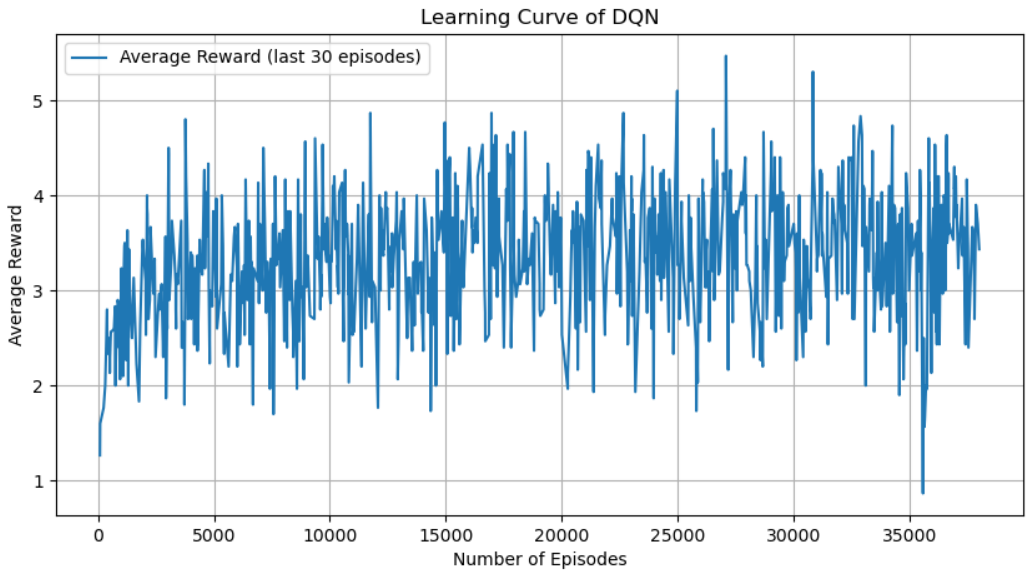
-Rutwik S. Kulkarni

1. Screenshot of the Test Results.

```
  self.policy_net.load_state_dict(torch.load("outputs_final/models_final/model_final.pth", map_location=self.device))
100%|                                                                                | 100/100 [01:28<00:00,  1.13it/s]
Run 100 episodes for 5 lives each
Mean: 60.12
rewards [51.0, 63.0, 63.0, 63.0, 51.0, 44.0, 51.0, 63.0, 74.0, 44.0, 74.0, 74.0, 63.0, 44.0, 74.0, 51.0, 63.0, 63.0, 74.0, 63.0,
63.0, 63.0, 63.0, 74.0, 63.0, 51.0, 74.0, 74.0, 63.0, 44.0, 74.0, 74.0, 74.0, 74.0, 44.0, 51.0, 74.0, 51.0, 63.0, 44.0, 44.0, 44.
0, 44.0, 51.0, 44.0, 51.0, 63.0, 63.0, 44.0, 74.0, 63.0, 63.0, 74.0, 74.0, 74.0, 63.0, 63.0, 51.0, 44.0, 63.0, 74.0, 51.0, 74.0,
63.0, 44.0, 63.0, 51.0, 63.0, 44.0, 63.0, 74.0, 51.0, 51.0, 63.0, 63.0, 44.0, 74.0, 51.0, 51.0, 44.0, 63.0, 63.0, 74.0, 51.0, 51.
0, 63.0, 63.0, 74.0, 44.0, 74.0, 74.0, 74.0, 63.0, 63.0, 74.0, 44.0, 74.0, 44.0, 44.0, 63.0]
running time 88.68120408058167
running time: 89.73707628250122
```

- **Mean: 60.12**

2. Visualization: Learning Curve of the DQN



- X-axis: number of episodes
- Y-axis: average reward in the last 30 episodes.

NOTE: The model initially trained for approximately 30,000 episodes. However, due to a computer crash, the intermediate training data was lost, though the model weights were saved in model_old.pth. The current visualization represents the learning curve from episode 30,000 onward.

3. Set of Experiments Performed.

1. Model Architecture:

- **Layers**
  - 2 convolutional layers (32 and 64 channels) with kernel size 5 and stride 2, followed by 2 fully connected layers (100 and 4 neurons).
- **Activations**: *ReLU* is applied after each layer.
- **Dropout:** Applied at 0.25 after fully connected layers to prevent overfitting.

2. Hyperparameters:

- **Episodes:** Planned for 100,000; training resumed from episode 30,000 due to a crash.
- **Batch Size:** 16.
- **Learning Rate:** 0.0001.
- **Discount Factor (Gamma):** 0.99.
- **Epsilon (Exploration):** Starts at 1.0, decays to 0.025 over 1000 steps.
- **Gradient Clipping:** 1.
- **Target Network Update Rate (Tau):** 0.001.

3. Weight Initialization and Activations:

- **Initialization:** Default PyTorch initialization.
- **Activation Function:** ReLU throughout.

4. Loss Function and Optimizer:

- **Loss Function:** Smooth L1 Loss.
- **Optimizer:** *AdamW* with *AMSGrad* (learning rate 0.0001) for stability and regularization.