Deep Learning Lab
Imitation Learning

Ayisha Ryhana Dawood
5653447

May 2, 2024

1 Introduction

The assignment focuses on Imitation Learning to train an autonomous driving agent using the Car-Racing environment provided by OpenAI Gym. Expert data is used to train a behaviour cloning agent using a CNN architecture. The evaluation of the agent over 15 episodes [1] resulted in an average score of 747.8 (std 101.7).

2 Expert data

2.1 Data collection

Data was collected by manually controlling the car through the environment with 5 actions - straight, accelerate, brake, left, right. 20000 samples were collected over 5 episodes with an average score of 650. The quality of the agent depends on the quality of the expert data used.

2.2 Data pre-processing

Each frame of the training dataset shows the environment, the tracks, the position of the car, the reward, and bar graphs indicating the current acceleration state. While the acceleration states are valuable information, the reward could add noise to the frame data. Thus the section of the frame containing the reward counter is set to all black pixels. The entire frame is converted to greyscale.

The data collected is highly imbalanced - around 60% of the data is straight. The actions are augmented to ensure a reasonable balance in action distribution. Augmentation increases the count of samples in the minority classes while the samples in the majority class are lost. Thus for each epoch, data is augmented to use a random subset of the majority samples.

[1] The YouTube video of the test run available here

2.3 History

The expert benefits from the knowledge of previous states and actions taken. The agent is given this advantage by including n previous frames from the run in addition to the preceding frame. This inclusion of episode history allows the agent to see the temporal evolution of the environment and better understand the dynamics of the car's movement, enabling it to make more informed decisions.

3 Agent

3.1 Model

The agent is trained using a CNN model with architecture and parameters as follows.

```
Model: CNN
    Hyperparameters:
        dimensions: 6x96x96
        n_minibatches: 300
        batch_size: 64
        learning_rate: 1e4
        epochs: 4
    Architecture:
        1. Conv2d: 16 filters,
            kernel=(3,3), stride=1, pad=1
        2. ReLU
        3. BatchNorm2d: 16
        4. MaxPool2d: kernel=(2,2)
        5. Conv2d: 32 filters,
        kernel=(3,3), stride=1, pad=1
    6. ReLU
    7. BatchNorm2d: 16
    8. MaxPool2d: kernel=(2,2)
    9. Flatten
    10. Linear: out=128
    11. RELU
    12. Linear: out=n_classes
```

3.2 Training and evaluation

The other training hyperparameters are as follows:

```
Training:
 history_length: {0,1,3,5}
 acceleration: 0.5
 left: -1.0
 right: +1.0
 brake: 0.1
 augmentation_class_sizes: {
        0: 3000,
        1: 3000,
        2: 3000,
        3: 3000,
        4: 200,
    }
Evaluation:
 history_length: 5
 acceleration: 0.2
 left: -0.3
 right: +0.3
 brake: 0.01
```

The model was trained with different values of history length - 0, 1, 3, 5. The model with history length 3 has the best valid loss in the given configuration, but it does not align with the test scores.

The agent navigates the world by predicting one of 5 action IDs based on the input sequence of previous frames. The action IDs are mapped to predefined action configurations in different directions. The best mean score over 15 episodes was observed with history length = 5.

History Length	Valid Loss	Test score - Mean (std)
0	0.49	747.8 (101.7)
1	0.47	724.5 (119.8)
3	0.42	364.4 (126.9)
5	0.44	741.5 (40.2)

Table 1: Training results

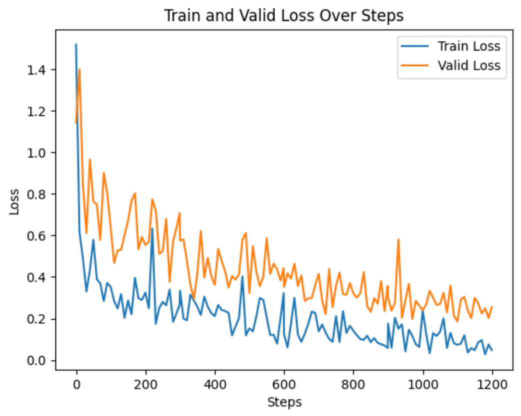

Figure 1: Training loss and Validation loss of the agent with history length =5

4 Configuration of the environment for test runs

The agent fails to accelerate in the initial frames in some episodes. This causes it to remain at the start position for the entire episode. To overcome this, the first 40 frames are manually accelerated. Training samples focussed on the initial frames (zooming in from a higher viewpoint) might help. The option has not been explored yet.

It is observed that the agent performs better when the acceleration speeds are lower than the expert run. For example, the expert run was configured to 0.5 while the test runs were configured at 0.2. This enables the agent to move slower and make more precise decisions in subsequent frames.

The left and right turns are very sharp during the expert turns (left = -1.0 and right = +1.0). The agent overshoots on the turns during test runs with these values. On reducing to -0.3 and +0.3 respectively, the agent turns in smaller angles, thus having more control over the navigation of the car. With the slower navigation, the car takes around 2000 timesteps to complete the track.

5 Conclusion

Imitation learning has been successfully used to train an agent using a CNN model to navigate a race track with a mean score of 747.8 (std 101.7). It has been observed that the quality of the expert data is an important factor in ensuring the performance of the network. While the CNN may achieve high accuracy during training, this does not directly reflect in the performance of the agent on the track. The test environment needs to be configured to optimize the performance of the agent.