Deep Learning Lab 2018

Srdjan Milojevic

Exercise 3

1 Overview of the task

The main goal of this exercise is to implement a behavioural cloning agent and evaluate its performance on the CarRacing control task from the OpenAI Gym benchmark suite. This first required the author to drive the car manually and collect training data. In the next section author describes how the data needed to be processed prior to the training.

2 Preprocesing the data

The collected data were unevenly distributed so the author randomly chose the data giving advantage to the ones less probable to appear in the whole dataset. This was important because the agent would grow a bias towards the most common action (in this case no action state appeared in the dataset ten times more often then other actions). Also, input data (frames of the game) were converted to gray scale and adapted to have optional history.

3 Hyperparameters and architecture

The author analysis different hyperparameters and architectures of the neural network and experimentally concludes that a CNN with four layers provides the best results. Instead of using max pooling, the author replaces it with strided convolutional layers (stride 2) and scarifies speed for significantly smaller memory footprint. AdamOptimizer proved itself to be reasonably better than GradienDescentOptimizer with a learning rate of 0.0001.

The author focuses on filter size and its influence on performance. In the first layer, too small filter would provide too much details while a bigger one looks more at a "global" features (in our case track and its borders). That is why the author uses a filter size 10 in the first layer, while in other layers kernel of size 5. In every layer there are 32 filters.

4 Results and Conclusion

The final test accuracy of a CNN-LSTM architecture is shown on Figure 1.

For comparison, training accuracy for different values of history length is presented in the following table:

Neural Network Variations	
History lengths	Accuracy
History 1	75.66%
History 3	76.17%
History 5	79.72%
LSTM	83.23%

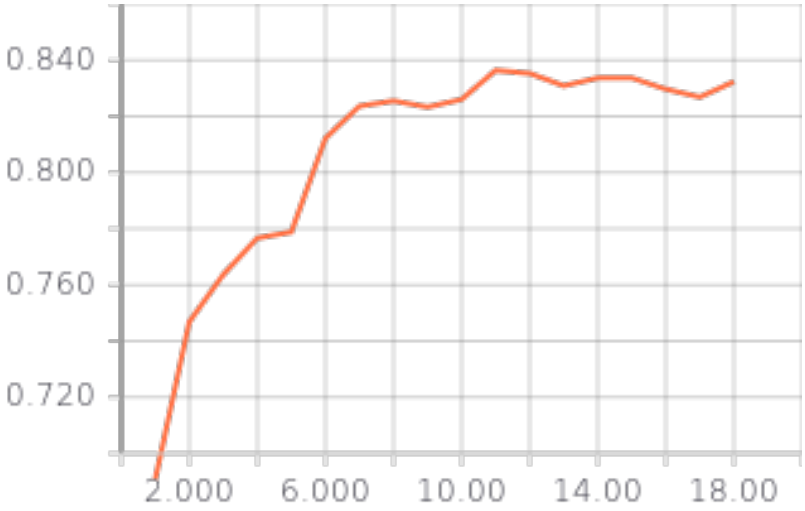

Figure 1: Learning curves of the realized neural network.

Test accuracy for other variation of the realized network are depicted on Figure 2, 3 and 4.

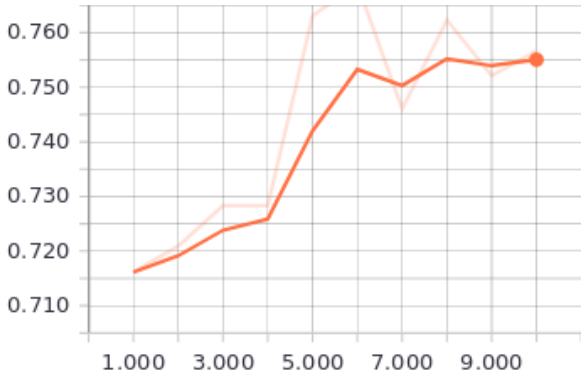

Figure 2: Learning curves of the history length 1.

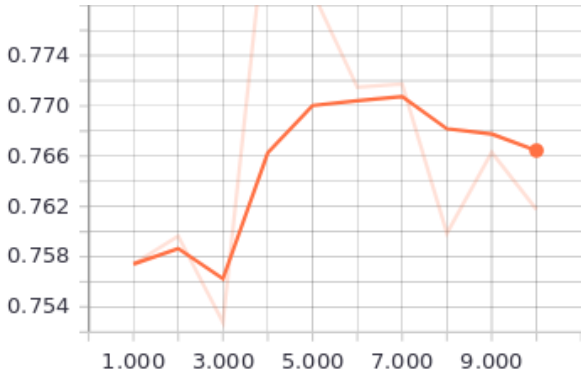

Figure 3: Learning curves of the history length 3.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series, just like in the given case. Therefore, it is noticeable that LSTM significantly improves the performance of the created neural network and achieves an accuracy of 83.23%.

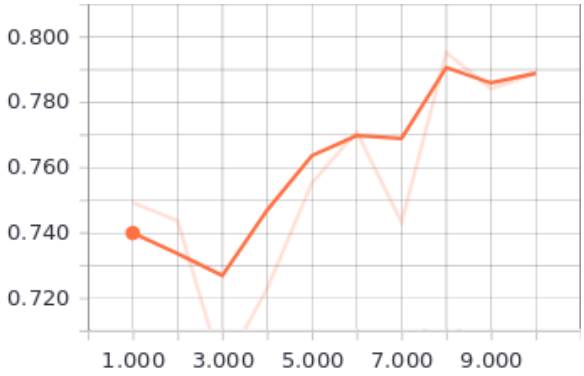

Figure 4: Learning curves of the history length 5.

The final code can be found here.