

Deep Learning Lab

Exercise 03

Megan Klaiber

December 4, 2018

1 Introduction

In this exercise a behavioral cloning agent will be implemented and its performance will be evaluated on the CarRacing control task of OpenAi Gym. Therefore data were collected by driving on the track. For training 45000 samples were used and for testing 5000.

The distribution of actions in the dataset ist imbalanced. To avoid problems with that during training the actions were uniformly sampled.

2 First Agent

For the first try the architecture of the last exercise were used. So it is a CNN with two convolutional layers with 16 5x5 filters, each followed by a Relu activation function and a max pooling layer of size 2. A fully connected succeed with 128 units and a softmax layer is used subsequent. The network is optimized with cross-entropy loss and stochastic gradient descent. It was trained over 1000 minibatches each of size 64. Learning rate was 0.0001 and the input was only the current image.

In Figure 1 you can see the validation and train error with this architecture. This doesn't look so bad but the evaluation of the performance over 15 test episodes isn't so good. Table 1 shows the mean achieved reward and the standard deviation of the evaluation.

REWARD	STD
- 57.46	52.94

Table 1: Mean reward and standard deviation.



Figure 1: Change of validation and train error during 1000 training minibatches and first architecture.

3 Improved Agent

To improve the bad result of section 2 another architecture is tested. This CNN has four convolutional layer with 76 filters and ReLu activation function. Then three fully connected layer and an output layer follow. The network is optimized with mean squarred error and Adam optimizer. It was trained over 1000 minibatches each of size 64. Learning rate was 0.0003 and the input was only the current image. In Figure 2 you see the validation and train error with this architecture. Here the validation error isn't that good. But if you have a look at Table 2 you see that the agent can achieve a much better result during evaluation than the first agent in section 2.

So if you leave the max pooling layer off and increase the filter size you do better in this task. Important information about the state get lost through pooling layer and in the task of CarRacing this has a bad impact on learning. In addition more details can be learned during training with a higher number of filters. Also the Adam optimizer seems more stable than the stochastic gradient descent.

REWARD	STD
501.99	316.34

Table 2: Mean reward and standard deviation.

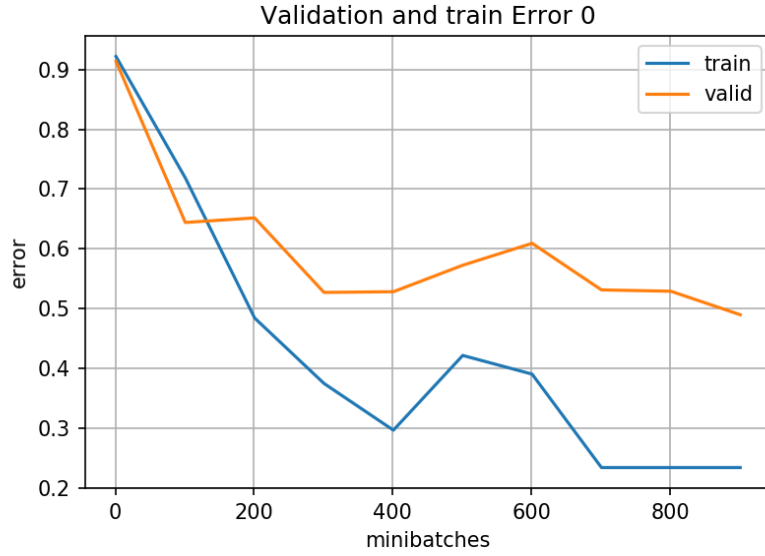


Figure 2: Change of validation and train error during 1000 training minibatches and improved architecture.

This architecture is used in the following experiments.

4 Impact of Batch Size

In this experiment the batch size is evaluated. So the batch size represent the number of demonstrations during training. The following values are tested: {32, 64, 128}

In Figure 3 you see the validation error during training and with the different batch sizes. Here the highest value achieve the lower validation error. Also during 15 test episodes (Table 3) the model trained with the highest batch size get the best result. This is the impact of the higher number of demonstrations. So the result of batch size = 64 should be higher than the one of batch size = 32.

BATCH SIZE	REWARD	STD
32	754.81	124.57
64	501.99	316.34
128	835.65	115.88

Table 3: Mean reward and standard deviation.



Figure 3: Change of validation and train error during 1000 training minibatches and batch size of 32, 64 and 128.

5 Impact of History length

In this experiment the history length of the input is evaluated. This means that the history of the last images are append to the input. The following values are tested: {1, 3, 5}

In Figure 4 you see the validation error during training and with the different history lengths. Here the highest length has the lowest validation error. If you compare this to the results of the testing episodes in Table 4 you see that here the highest lenght achieve the worst result. It was expected that the results with higher history length are better.

HISTORY LENGTH	REWARD	STD
1	804.03	127.49
3	503.55	165.58
5	391.83	271.65

Table 4: Mean reward and standard deviation.

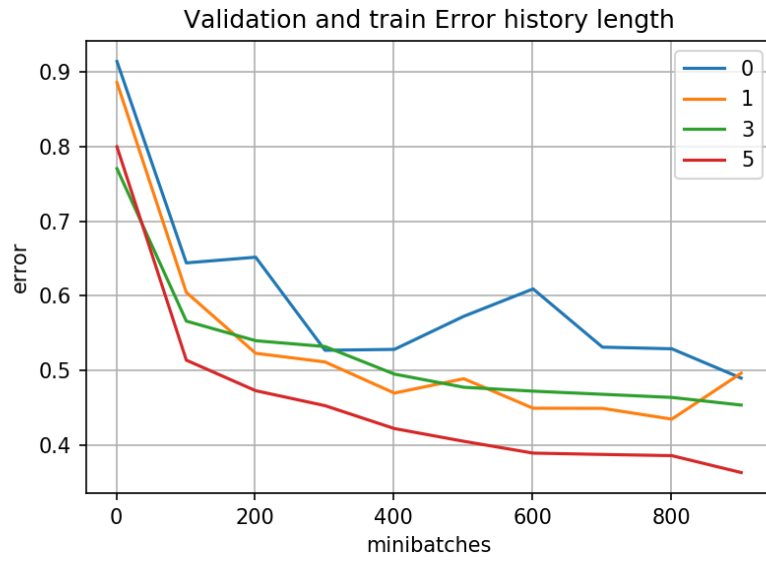


Figure 4: Change of validation and train error during 1000 training minibatches and history length of 0, 1, 3 and 5.

6 Conclusion

With this exercise you see that it is important to adjust the architecture individually to the given task. Also the hyperparameter tuning is important and you see that every hyperparameter has their individually impact on the training.