Deep Learning Lab - Exercise 3

Baohe Zhang

04.12.2018, Uni Freiburg

1 Introduction

In this exercise we base on Gym environment by OpenAI and implement a convolutional neural network model by Tensorflows[1], which will be used to accomplish imitation learning. Besides that, we will try preprocessing our data set, constructing different architectures, analysis the impact of training epochs and tuning hyperparameters using Hpbandster[2]. The code is based on the framework provided by AIS Lab of University Freiburg.[3]

2 Preprossing Data set

Due to the feature of Car racing game, it's often to get an unbalanced data set. Achieving a balanced data set will be very helpful for us. Thus we proposal a weighted sampling method. Its effect can be shown in the following figure:

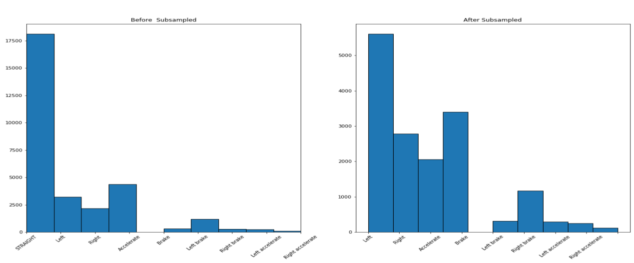

Figure 1: Sampling Method Impact

[1] Martin Abadi et al. "Tensorflow: a system for large-scale machine learning." In: *OSDI*. vol. 16. 2016, pp. 265–283.

[2] Stefan Falkner, Aaron Klein, and Frank Hutter. "BOHB: Robust and efficient hyperparameter optimization at scale". In: *arXiv preprint arXiv:1807.01774* (2018).

[3] Andreas Eitel. *Deep Learning Lab*. `https://github.com/aisrobots/dl-lab-2018`. 2018.

3 CNN Structure Comparison

Here we compare the performance of two CNN structures. The first one has 3 convolutional layer and 3 fully connected layer and finally output 3 float values(regression). The second one has an additional convolutional layer. After testing the model with 15 episodes, the reward we get is:

Here we can see that weak model can't capture enough for our agent, thus its result is also very frustrating. Once we add another layer the result gets much better.

LAYERS	REWARD	STD
3	-82.5	+-2.6
4	516.1	+-203

4 Training Epoch Impact

The number of training epochs is one of the most important element for training a neural network. Here we investigate the resulting error during training procedure: We can find after cretain epochs (in this case 20), our model tends

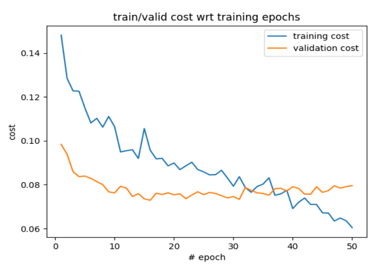

Figure 2: Learning Curve of Training Epochs Growing

to overfit the data.

5 History Length

In the real life, when we need to make driving decisions, usually we are basing on the previous situation. Therefore adding history is also very intuitive for us. Here we compare the impact of history Length, where history length $\in \{1, 3, 5\}$, the rest hyperparameters stay the same. We can see that with hisotry length grows, the model gives worse result. This perhaps because we should increase our model capacity.

Figure 3: (a) Length 1 (b) Length 3 (c) Length 5

6 Long short-term memory

Long short-term memory[4] is published in 1997 as a special unit of RNN, which is mainly used to deal with time-serial data. It consists of cell, input gate, output gate and forget gate. With this structure it can handle long time-serial data easily and can give us a better result. In this experiment we use history length 5:

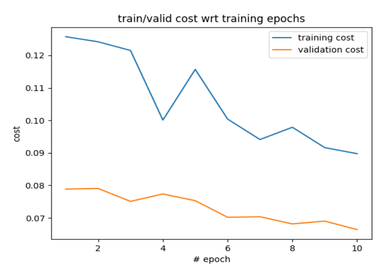

Figure 4: Learning Curve of LSTM

6.1 Automatically Hyperparameter Tuning

Hyperparameter Tuning is always the most time-consuming procedure. But with the help of tuning Hpbandster we can easily find the best combination from our configuration space. LR: Learning rate, BS: Batch size, NF: Number of Filters, HL: History Length, NS: Number of Subsamples

LR	BS	NF	HL	NS	LSTM
0.000322	58	74	1	16000	NO
0.000134	28	41	2	16000	YES

[4]Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

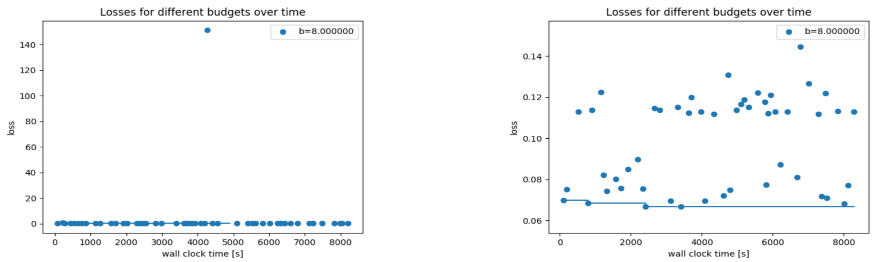

Figure 5: (a) Without LSTM (b) With LSTM

7 Conclusion

Imitaion learning is an very intuitive technique and can give us a good result in special environment. Building up a suitable architecture resulting an amazing result. With the help of LSTM and AAD techniques we could also easily deal with time-serial data and find the corresponding best hyperparameter combination. But its limitation is also quite clear after this experiment.

TYPE	REWARD	STD
LSTM	826.7	+-139.8
NO LSTM	540.6	+-188.64
MANUALLY	811.4	+-53.679