Report for Artificial Neural Network HW4

Jiayi Weng
Department of Computer Science
Tsinghua University
wengjy16@mails.tsinghua.edu.cn

1 Network Architecture and Hyperparameter Setting

According to the provided code and guideline, I keep the other hyperparameters the same as before. I only tune the number of layers of network, the type of RNN cell, and the optimize machine.

Therefore, in each experiment set, there are 6 network architecture listed in Table 1.

Table 1: Network Architecture

Name	#Layer	RNN unit
RNN1	1	RNN
RNN2	2	RNN
LSTM1	1	LSTM
LSTM2	2	LSTM
GRU1	1	GRU
GRU2	2	GRU

The optimization methods include original gradient decent and SGD with momentum, namely GD (no momentum), MM5 (momentum=0.5) and MM9 (momentum=0.9).

2 Experiment

2.1 Experiment Setting

All of these experiments are conducted on a Linux server. The CPU infomation is "Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz", with 8 NVIDIA GeForce GTX 1080 Ti.

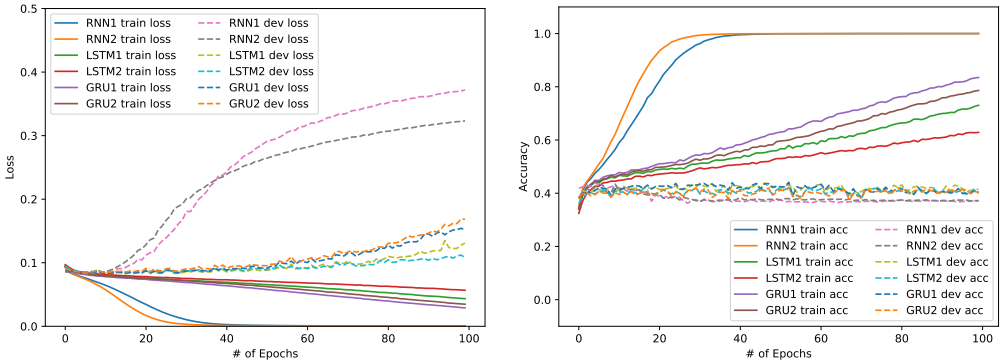
2.2 Origin Gradient Decent (GD)

Using GD method, the result can be found in Figure 1 and Table 2. From the result, when using GD, these networks have the following characteristics:

Converge Speed BasicRNNCell converges faster, then GRUCell, and BasicLSTMCell is the slowest.

Overfit Both three types of RNN cell has the problem of overfitting. BasicRNNCell faces this problem seriously than others. We could see in Figure 1 that after $\sim$30 epochs the valid accuracy no longer goes up. Thus with RNN we must early stop to get a better performance. BasicLSTMCell overfits the data less than other methods.

Layers Increase the layer of network does not enhance the performance. On the contrary, the network seems easier to overfit the data.

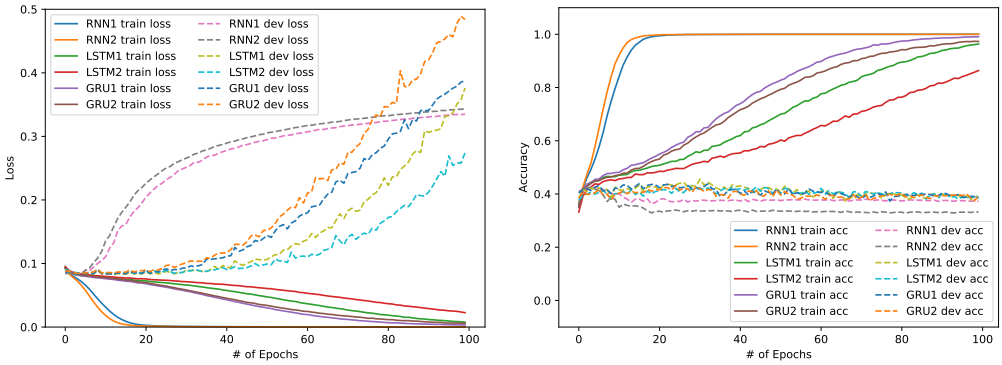


(a) Loss (b) Accuracy

Figure 1: Result of applying GD method in different network architecture.

Table 2: GD Numeric Result

Name	Min Train Loss	Max Train Acc	Min Valid Loss	Max Valid Acc
RNN1	0.000380	0.999883	**0.082691**	0.438692
RNN2	**0.000292**	**1.000000**	0.082897	0.434151
LSTM1	0.043414	0.730454	0.082907	0.438692
LSTM2	0.056667	0.628394	0.084614	0.423252
GRU1	0.028998	0.835089	0.083249	**0.441417**
GRU2	0.034662	0.786049	0.084942	0.421435



(a) Loss (b) Accuracy

Figure 2: Result of applying MM5 method in different network architecture.

2.3 SGD with 0.5 Momentum (MM5)

Using MM5 method, the result can be found in Figure 2 and Table 3. From the result, when using MM5, these networks have the following characteristics:

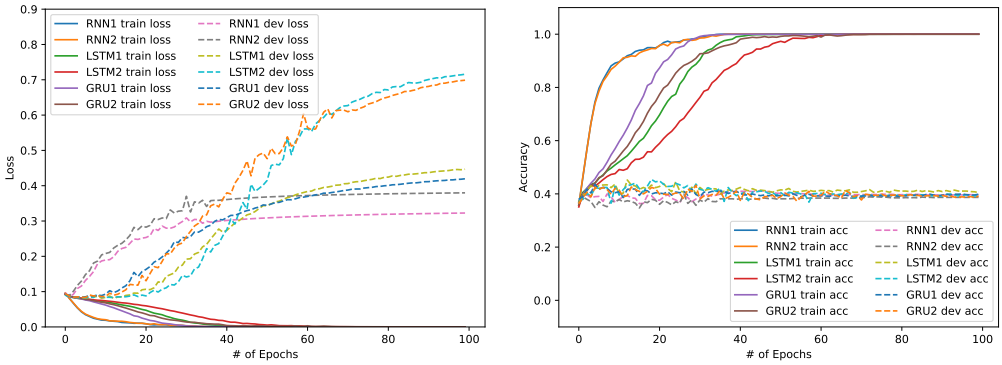
Converge Speed Same as GD.

Overfit BasicRNNCell faces the overfit problem most seriously. After changing GD to MM5, its converge speed turns faster than before. However, after about 80 epochs the GRUCell overfits more seriously than LSTM. Its training loss is higher than LSTM's.

Layers Conclusion of GD can be also applied here, except LSTM. The two layer LSTM's training loss is the lowest at 100 epochs.

Table 3: MM5 Numeric Result

Name	Min Train Loss	Max Train Acc	Min Valid Loss	Max Valid Acc
RNN1	0.000205	**1.000000**	0.082831	0.425976
RNN2	**0.000172**	**1.000000**	0.082491	0.425068
LSTM1	0.007586	0.963717	**0.082454**	**0.455949**
LSTM2	0.022608	0.863530	0.084072	0.426885
GRU1	0.003309	0.990637	0.082743	0.439600
GRU2	0.005851	0.973783	0.084521	0.432334



(a) Loss (b) Accuracy

Figure 3: Result of applying MM9 method in different network architecture.

Table 4: MM9 Numeric Result

Name	Min Train Loss	Max Train Acc	Min Valid Loss	Max Valid Acc
RNN1	0.000102	**1.000000**	0.085168	0.411444
RNN2	**0.000094**	**1.000000**	0.089313	0.399637
LSTM1	0.000220	**1.000000**	**0.080949**	0.452316
LSTM2	0.000223	**1.000000**	0.082294	**0.454133**
GRU1	0.000229	**1.000000**	0.084048	0.435967
GRU2	0.000249	**1.000000**	0.083409	0.432334

2.4 SGD with 0.9 Momentum (MM9)

Using MM9 method, the result can be found in Figure 3 and Table 4. From the result, when using MM9, these networks have the following characteristics:

Converge Speed Same as GD and MM5. An interesting thing is that all of the network's maximal training accuracy is 100%, hence tuning momentum parameter can lead to faster converge speed and more serious overfit problem.

Overfit Almost the same as MM5, but performance of LSTM and GRU seems the same. Thus the complexity of network determines the degree of overfitting.

Layers After about 70 epochs, the two-layer LSTM and GRU overfit most, then single-layer LSTM and GRU, finally RNN.

3 Conclusion

According the above analysis, I finally choose the LSTM1 model with MM5 optimization method. To avoid overfitting problem I choose the first 20 epochs result. The result is under the "codes" folder.