REPORT FOR ARTIFICIAL NEURAL NETWORK ASSIGNMENT 3

Xiang Zhang
Department of Computer Science and Technology
Tsinghua University
xiang-zh17@mails.tsinghua.edu.cn

November 3, 2019

ABSTRACT

In this assignment, two types of basic RNN cells, i.e. LSTM and GRU are implemented and constitute a bidirectional RNN structure to perform fine-grained sentence-level sentiment classification task on Stanford Sentiment Treebank (SST) dataset. The network is evaluated in terms of number of layers and self attention, and the effects of these factors are analyzed.

1 Network Architecture

Basic network architecture is derived from original codes provided, with some minor changes, which is shown in the following table

Table 1: Network architectures in this experiment

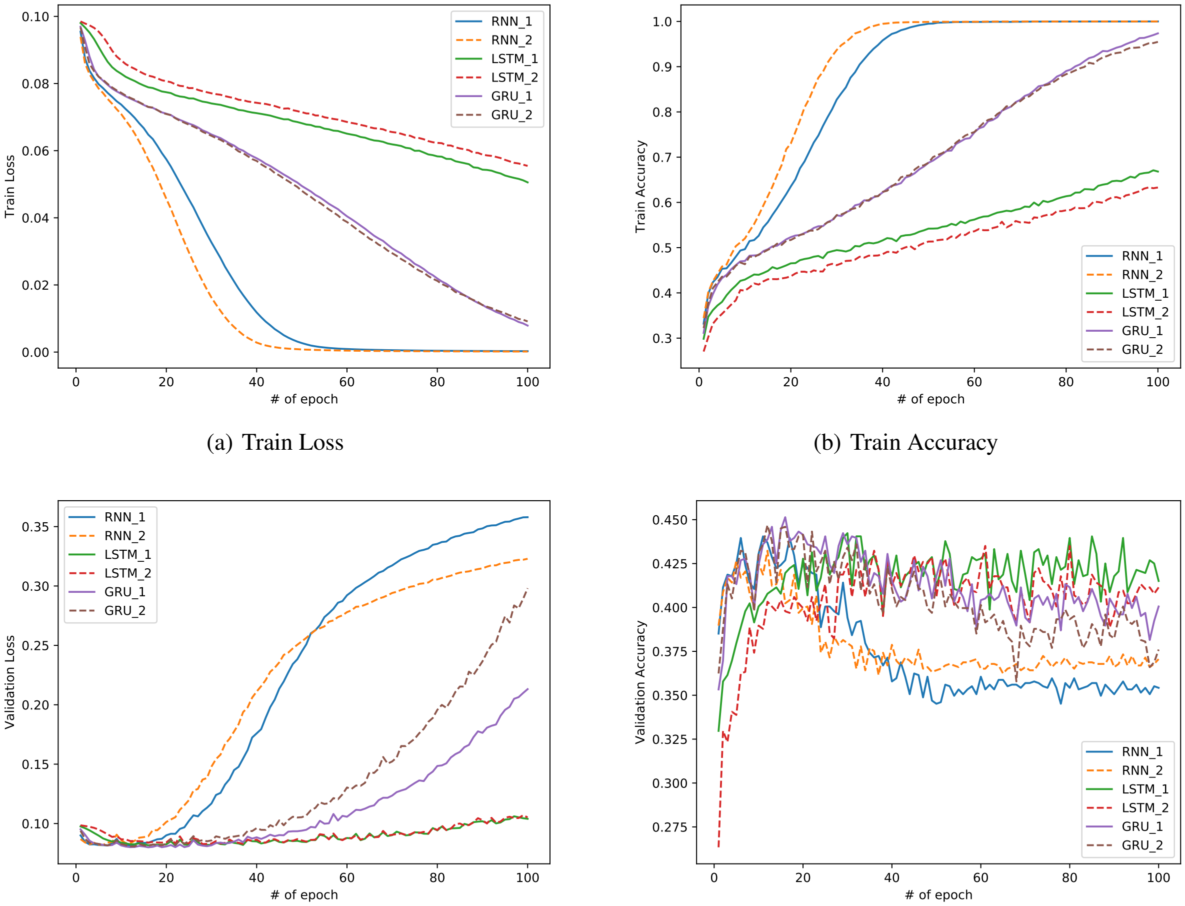
Name	# Recurrent Layers	RNN Unit
RNN_1	1	RNN
RNN_2	2	RNN
LSTM_1	1	LSTM
LSTM_2	2	LSTM
GRU_1	1	GRU
GRU_2	2	GRU

Besides, the number of units of recurrent layer is tuned to determine its effect on the performance, in which case, an extra suffix will be added to the name of the network, e.g. RNN_1_256 indicates the network has one recurrent layer with 256 units. Unless otherwise stated, the default value is 512.

2 Experiment Result and Analysis

2.1 Choice of RNN Cells

In this setting, all networks are trained through specified Gradient Descent (GD) optimizer. Other hyperparameters are left to their default values. The results are shown in **Fig. 1** and **Table 2**.



(a) Train Loss

(b) Train Accuracy

(c) Validation Loss

(d) Validation Accuracy

Figure 1: Training and validation loss/accuracy against each epoch

Table 2: Numerical performance of different networks

Name	Best Train Loss	Best Train Acc	Best Val Loss	Best Val Acc
RNN_1	0.000262	**1.000000**	0.081461	0.440509
RNN_2	**0.000159**	**1.000000**	0.082071	0.432334
LSTM_1	0.050557	0.670997	0.081586	0.442325
LSTM_2	0.055464	0.633193	0.082744	0.435059
GRU_1	0.007888	0.973666	**0.080008**	**0.451408**
GRU_2	0.009124	0.954822	0.080782	0.446866

From all the results above, the training of RNN is fastest among these three types, rendering it more susceptible to over-fit after 30 epochs, in which case, the final validation loss is two times larger than the initial value, and the validation accuracy drops simultaneously. The LSTM trains most slowly owing to its complex structure compared to GRU and RNN, making it suffers from over-fitting the least. In this experiment, its validation loss begins to increase after 80 epochs, when most of the training process has elapsed. GRU has a structure with intermediate complexity, which accounts for its training time at the median.

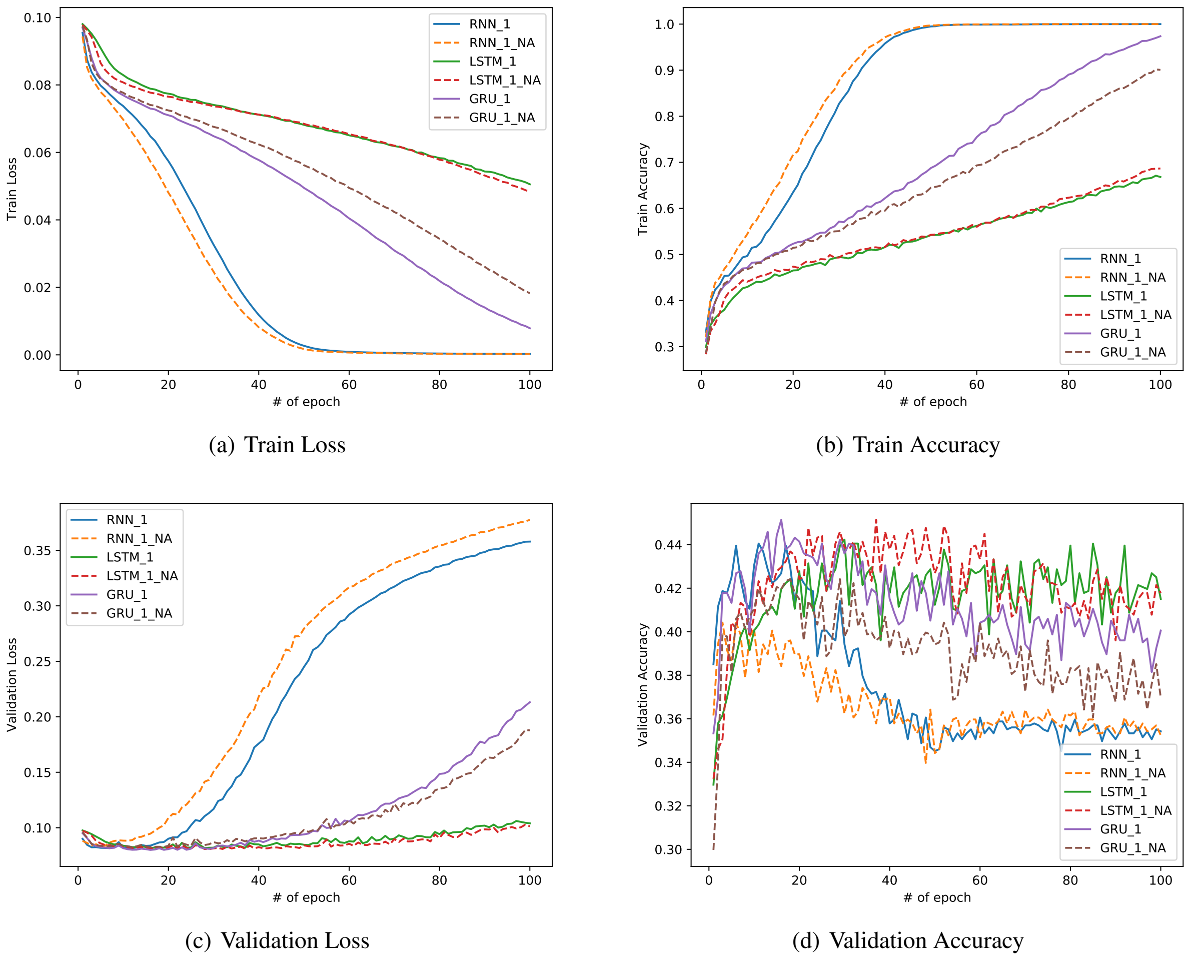
In terms of performance, except for the overfitting occurs during the training, GRU is on par with LSTM and better than simple RNN. Thanks to the built-in gates in GRU and LSTM, they are more capable of "remembering" information in a sequence, in comparison with a typical RNN cell.

2.2 Number of Recurrent Layers

From **Fig. 1** and **Table 2**, increasing the number of recurrent layers only facilitates the training process of RNN by accelerating this procedure, yet causing it to over-fit prematurely. As for the validation, adding an extra layer merely impairs its performance by $\sim 1\%$. One possible explanation is the bolstered over-fitting brought about by more layers can harm the performance during training process. For this reason, in the following experiments, only networks with a single recurrent layer will be adopted.

2.3 Self Attention Mechanism

In this experiment, three networks without self-attention mechanism are also implemented in contrast with those mentioned in **Section 2.1**. The networks are formed by taking the last output of bidirectional RNN layer, concatenating them and connecting to an MLP classifier. The results are illustrated in **Fig. 2** and **Table 3** (suffix **_NA** of the network name means **N**ot **A**ttention).



(a) Train Loss

(b) Train Accuracy

(c) Validation Loss

(d) Validation Accuracy

Figure 2: Training and validation loss/accuracy against each epoch w/ or w/o self-attention

Table 3: Numerical performance of networks w/ or w/o self-attention

Name	Best Train Loss	Best Train Acc	Best Val Loss	Best Val Acc
RNN_1	0.000262	**1.000000**	0.081461	0.440509
RNN_1_NA	**0.000197**	**1.000000**	0.083706	0.404178
LSTM_1	0.050557	0.670997	0.081586	0.442325
LSTM_1_NA	0.048324	0.686681	0.080714	**0.451408**
GRU_1	0.007888	0.973666	**0.080008**	**0.451408**
GRU_1_NA	0.018280	0.901685	0.082440	0.424160

From the results, we can conclude that self-attention bolsters performance before RNN or GRU entering over-fitting stage, as is expected. For LSTM cells, self-attention comes with little help, and even the one without attention mechanism outperforms its counterpart, though the differences in between are remarkably minor.

3 Final Architecture

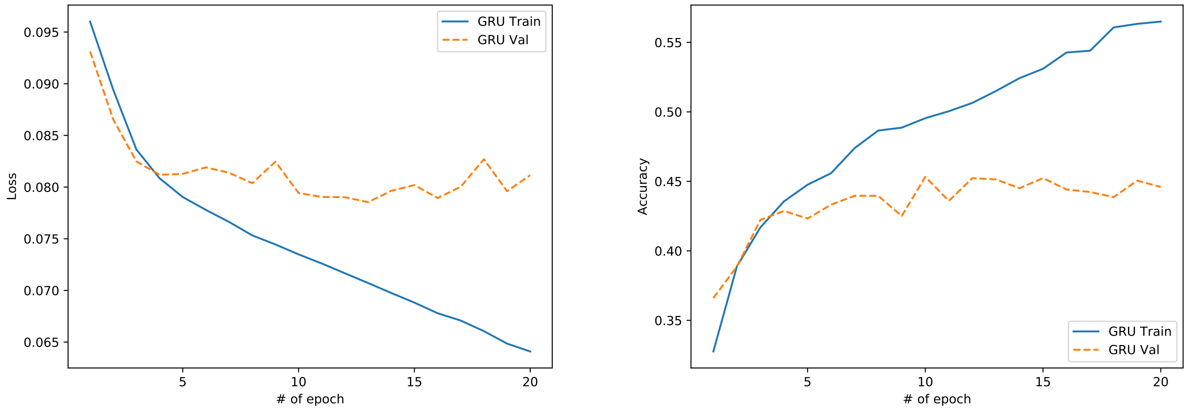
Based on aforementioned experiment results, GRU cell is chosen as the building block of final architecture, with 256 units, half of the value in original code. Plus, the global optimizer is replaced by SGD with momentum to accelerate the training process. The hyperparameters are

`learning_rate=0.005, num_epoch=20, batch_size=16, momentum=0.5.`

Table 4: Architecture of final network

Layer name	Spec
WordEmbedding	
RNN	GRU Cell (units=256)
SelfAttentionLayer	
MLP	

The training results are as follows:



(a) Loss

(b) Accuracy

Figure 3: Training and validation loss/accuracy against each epoch

Table 5: Numerical performance of best epoch (#9)

Train Loss	Train Acc	Val Loss	Val Acc
0.073486	0.495435	0.079429	0.453224

Since this final model utilizes SGD with momentum optimizer, the training process is shortened compared to aforementioned GRU_1 network, which employs GD optimizer. The validation accuracy is a little better than GRU_1, which is related to the fine tuning of parameters decisive to network architecture, i.e. number of units of GRU cell. Self-attention is enabled here, for it enhances the performance when used together with GRU. Plus, early stop technique plays a significant role during training process. From the analysis in previous sections, unlike LSTM, GRU can still be susceptible to over-fitting, though it is capable to cope with this situation better than basic RNN cell. Excessive epochs will harm the performance and finally results in the degration of accuracy, for which reason epoch #9 is opted as the final model.

4 Conclusion

In this assignment, three types of RNN cells, i.e. RNN, LSTM and GRU, along with several networks built upon them are implemented to complete the classification task of SST. It is found that GRU cell performs best in terms of the environment and parameters during the process of the experiment, and self-attention can help to bolster the accuracy in this setting. With regard to the number of recurrent layers, increasing it can impair the performance due to the elevated probability of over-fitting. Based on these observations, the final model is based on RNN with fine tuning of parameters, which can obtain an accuracy of 45.32% on this task.