

Programming Homework 3 LSTM

Yuheng ZHAO

November 2022

1 Contents and File Structure

According to the requirements, this program includes:

1. `LSTM.py`, source code of the analyser. We imply a sequence model with LSTM to conduct sentiment analysis on SST-2 dataset with positive and negative sentiments. The loss and accuracy of the model of each epoch is reported the same way as the project implemented before.
2. `plot.py` is used to plot the results of the analyser, including losses of each epoch.
3. `./img` folder contains the output of `plot.py`.
4. `./log` folder stores the log data for the training process, including the average loss and accuracy of each iteration. In this approach, epoch size is set to 20, and the last piece of the log data contains the final accuracy of the model.

2 Result

In this part we illustrate the model structure (see subsection 2.1), the result of model training, including the average loss of each epoch and the final accuracy (see subsection 2.2), in which the data can be found the data in `./log` folder.

2.1 Network Structure

We use single layer BiLSTM for processing, the input word is embedded as input vectors through the embedding layer, then pass through BiLSTM, concat the output of last hidden layer of LSTM and reverse LSTM together as the input of the fully-connected layer, then use the output of the fully-connected layer as the output of the model. We use the loss function same as the logistic model for binary classification in project 1, and SGD as optimizer with momentum of 0.9 and learning rate of 0.01.

Some parameters of the model are shown as follows: vocabulary size is 18003, embedding dimension is 128, input size is the max length of sentences in each

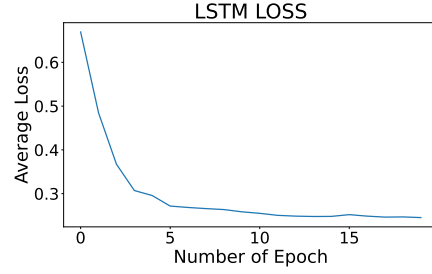


Figure 1: Loss of each epoch of LSTM

batch, hidden size is 128, number of LSTM layers is 1, input dim and output dim of fully-connected layer is 256 and 1, respectively.

2.2 Loss and Accuracy

We trained the models with SGD optimizer and a momentum of 0.9, a learning rate of 0.01. We report average loss after each epoch of the training data. Figure 1 shows the change of loss in each epoch. The loss of each iteration decrease and finally converge at a low value as the number of iteration increases, and the final accuracy of the model is approximately 78.034%.