Neural Network Theory and Applications | Assignment#4

Shangyu Liu (020033910052)

May 18, 2021

1 Introduction

Human emotions usually change as time goes by. In the same way, the EEG signal which represents the emotions of experiment participants is a classical temporal-sequential signal. The Long Short-Term Memory (LSTM) model is a neural network suitable to process temporal-sequential signal. In this assignment, we implemented an LSTM model to deal with the three-class emotion classification problem over the SEED dataset. We adopted SVM and DNN as baseline models. From the experiment result, it is observed that using LSTM model could considerably improve the training accuracy but conversely failed on testing performance.

2 LSTM

LSTM is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feed-forward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).

The general architecture of LSTM is shown in Figure 1. Each LSTM layer consists of repeated cells which are corresponding to the input at specific time steps. Each cell takes in the input features and the hidden state from the last cell, and output hidden features to next layer and hidden state to the next cell. There are lots of gates in each cell to control the information flow. They determine how much information in the last hidden state will be passed on and also determine the importance of the current input. This is why the model is called long-short term memory.

The function of each gate in LSTM cells can be described as

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c_t} &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t * c_{t-1} + i_t * \tilde{c_t} \\
h_t &= o_t * \sigma_h(c_t)
\end{aligned}
$$

Where x_t denotes the input vector to the LSTM cell, f_t is the forget gate's activation vector, i_t is the input/update gate's activation vector, o_t is the output gate's activation vector, h_t is the hidden state vector (also known as output vector of LSTM cell), $\tilde{c_t}$ is cell input activation vector, c_t is the cell state vector. W, U and b is the weight matrices and bias vector parameters which need to be learned during training. The $*$ operation is Hadamard product, i.e. element-wise product.

3 Implementation

3.1 Building Dataset

For LSTM model, the SEED training-set is first divided into 11 parts based on different

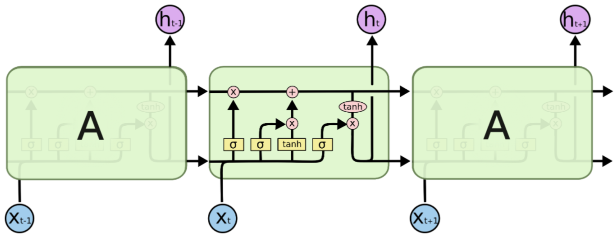

Figure 1: Overview of LSTM

subjects and then each part is divided into 15 segments based on different videos. We use a sliding window, the size of which is 60 samples, to extract small sequences as the training inputs. As the sequence length is not always the multiple of the window size, we combine the last 60 samples into a new sample. In this way, we can assure that all new training samples are extracted from the sequence of the same video and all raw samples are included.

Each new sample in the training set is a sequence consisting of 60 time steps and at each time step there is an input with 310 dimensional features. As the labels at each time step are the same, we set any of them as the new training label.

For baseline models without sequential inputs, we use a subset of the original training data to assure the fairness of comparison. The new training set is collected by sampling the old one every 60 samples.

3.2 LSTM settings

The LSTM model is implemented using Keras framework. The input data sequence is fed into an LSTM layer with 128 units and then activated by a tanh function and passed to an fully-connected layer with 3 units. The outputs are transformed by a softmax operation. The cross-entropy loss function is adopted.

3.3 Baseline Models

To boost the performance of LSTM, we implemented a three-layer fully connected network as the baseline model. The fist layer consists of 256 units and is activated by ReLU function. THe second layer consists of 128 units and is activated by ReLU as well. The output layer has 3 units and is activated by a softmax operation.

We also compared the LSTM model with SVM method which we have implemented in Assignment 2. We select the min-max-modular SVM model with prior knowledge as the baseline model, because it has the best performance among all SVM variants.

4 Results

Figure 2 plots the training and testing accuracy over time. It's obvious that LSTM has the best training accuracy over 0.9, but the worst testing accuracy around 0.5. We have discussed the different data distribution of the training and testing set. That's why the performance is definitely different between the two scenarios. But for the phenomenon that LSTM defeated by baseline DNN over the testing set, we guess the training sequence might be wrongly formulated. The LSTM model obviously doesn't learned the sequential knowledge from the training set. And what's worse, as the number of parameters in LSTM is much larger, it's very likely that the model is overfitted.

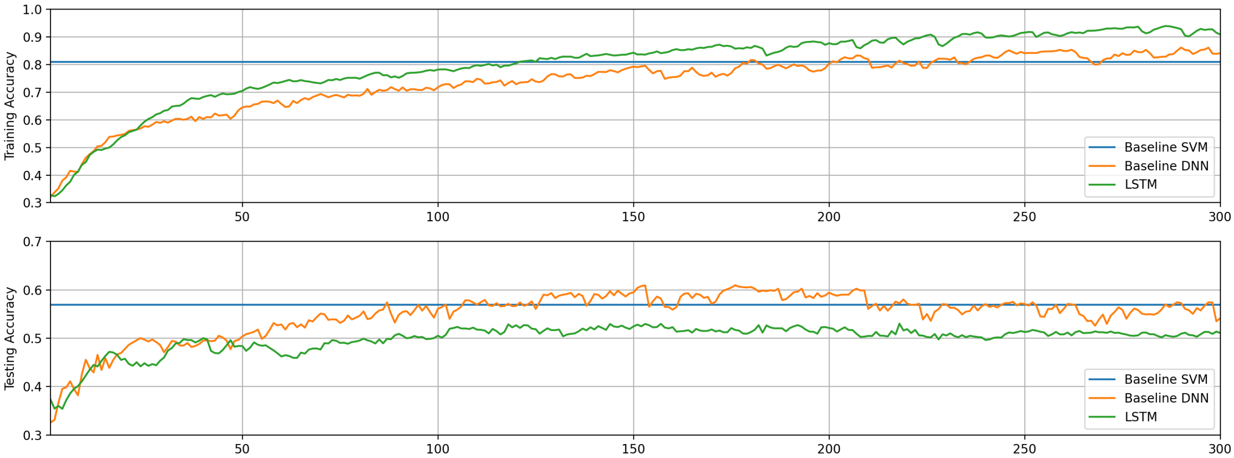

Figure 2: Training and Testing Accuracy among different models