Deep Learning Course Assignment Report

Bicheng Gao,[1] *Songyu Ke,*[2] *Yuhao Zhou*[3]

[1]*volz.kz.g@gmail.com*
[2]*songyuke@sjtu.edu.cn*
[3]*yuhao_zhou95@sjtu.edu.cn*

14 ACM class, Shanghai Jiao Tong University

ABSTRACT ******edit(accuracy)

The purpose of this research is to classify the tone using the neural network and compare the differences between three neural network frameworks — `Torch`, `MXNet` and `Theano`. Since it seems hard for the neural network to extract the feature from raw data automatically, it is necessary to preprocess the data with some proper methods. We've tried two ways of data preprocessing: eliminating the noise in all datasets or add some noise in training data to fit the environment of test data. With the processing data and Convolutional Neural Networks, we finally achieved the accuracy of _______% in `test_new` dataset. After reaching a better performance, we keep working on comparing the frameworks and provide some major factors listing on this report.

1 Introduction ******edit

The project of *Deep Learning Course* aims at constructing a proper neural network model to solve tone classification with different frameworks and discuss factors which may have an influence on the performance of each framework.

With only the raw data, the file: `train.engy`, `train.f0`, `test.engy`, `test.f0`, `test_new.engy`, `text_new.f0`, it's a rather tough task for neural network itself to extract proper feature that may have a good performance in the test dataset. Therefore, preprocessing the data is an appropriate way to change the data, which is much easier to train a better network. We eliminate the low energy noise and use cubic spline interpolation to fix the data with a proper length. Then with those data (in directory `data`) ******edit(data processing procedure)**
Since the test data is recording in the noisy environment, which is totally different from the situation of training data, the other thought is to add some white noise in the training data to fit the environment of the test data. We also tried this method to evaluate the performance.
And in the 2^{nd} *Section*, we will provide all the details in our data preprocessing.

After preprocessing the data, another major work need to be decided is the model of neural networks. With different frameworks, the implementation of each framework is different. Therefore, with the same parameters, the performance of three frameworks may have some slight changes. To choose an appropriate neural network model is complicated in many works. In this project, the total size of training data set is 400 with the fixed data shape: $[120, 1]$ ([width, height]). Therefore a small model may work in a small collection of data. Fully-connected Networks and Convolutional Neural Networks are our choices, since it has excellent performance in many tasks. Meanwhile, we also try some complicated models with more layers, and the results of models comparison are also shown in the report.
All the information details of this part is shown in the 3^{rd} *Section* **Model of Neural Networks**

In the 4^{th} *Section* **Background of the Frameworks**, we will provide the elementary information of three frameworks: `Torch`, `Theano`, `MXNet`. We will first give out a fundamental analysis of three frameworks through their own description and then compare with our own evaluations by the experiment of the tone classification in some aspects, which shown in the 5^{th} *Section* **Evaluation**. ********edit(some factors, reason)**

For the 6^{th} *Section* **Future Work** Part, we're going to show some vulnerabilities of our works and the improvement that can be implemented in the future, and giving a **conclusion** in the 7^{th} *Section*. ******** edit (some detail future work lists)**

2 Data Preprocessing

****** edit**

3 Model of Neural Networks

In this part, we're going to show some models we used to solve tone classification. During the data processing, we've changed some parameters and models to fit the data to have a good training performance. With all models we used, the following three models can be the most representative ones for our experiment. Some may performance out of expectation, and others may have no improvement compared with simple ones. The models we shown are as follows:

- Three layers of Fully Connected Networks.
- Variant of the Lenet.
- Variant of VGG.

Three layers of Fully Connected Networks is the most simple model we used, and then improve it to the Lenet

3.1 Three layers of Fully Connected Networks

3.2 Variant of the Lenet

3.3 Variant of VGG

3.4 Evalution *****edit

4 Background of the Frameworks

We now keep working on the second major part of our *DL_project*. Before we compare with the three models, we first list some standard information, from which we may derive inspiration of deciding the comparative factors.
Most of the background information are from their *Official Site*.

4.1 `Torch`

The summary of core features of Torch can be listed as follows:

4.2 `Theano`

4.3 `MXNet`

5 Evaluation ***** edit

In this part, we're going to give a formal evaluation system for three frameworks — `Torch`, `Theano`, `MXNet`. With the following aspects and the general idea to these frameworks, it is rational for us to give a brief impression and further analysis to the frameworks. And the factors we use are listed as follows:

5.1 Learning and Implementation

5.2 Training Effect Evaluation

5.3 Training Time Evaluation

6 Future Work

As we have spent most of dealing with the dataset, there're still many works need to be accomplished. As for the model part, thought we've tried the several neural networks and choose *Three-layers Fully Connected Networks*, *Variant of Lenet* and *VGG* to represent most performance of our model, there're still many networks may work well, but we haven't tried yet.

Recurrent Neural Network performances well in many cases of Natural Language Processing in recent works. We guess it may also improve our model to some extent. And like Alex et al.,[14] work on Speech Recognition, RNNs may have a result out of expectation.

For three frameworks, there're still some problems remain to be solved. Just like MXNet, many researchers will suffer from its poor documents. Therefore, some of the effect or implementation of the embedded functions must be learnt from its open-source code. Meanwhile, MXNet has a well-performed encapsulation and module, therefore, their still many embedded functions is left to do some research. Maybe it can be another aspect for the framework evaluation.

7 Conclusion

8 Reference

[14] Alex Graves, Navdeep Jaitly, *Towards End-to-End Speech Recognition with Recurrent Neural Networks*