Midterm Report for 11785

Yilang Liu
Mechanical Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213
yilangl@andrew.cmu.edu

Yao Lu
Mechanical Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213
yaolu2@andrew.cmu.edu

Wuzhou Zu
Mechanical Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213
wuzhouz@andrew.cmu.edu

Xupeng Shi
Electrical and Computer Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213
xupengs@andrew.cmu.edu

Abstract

This is the project proposal for 11-785 course. Our project is to use RNN specifically Long short-term memory (LSTM) to transform audio, video input into transcripts and determine who are saying. The data sets are captured from Japanese anime "君の名は" Development will be facilitated by open source solutions such as the NLP From Scratch tutorial by PyTorch, where our primary contribution will be either labeling the speaker or segmenting the transcript to create visually-aware subtitles. We will choose a dataset that is reasonable for our hardware requirements. We expect a model with 20000 parameters to take 30 minutes long to perform (training / validation) on one batch of data. Therefore, for the entire dataset, we expect (training / validation) to take 180 minutes. This is where the 1.2 $ estimate comes from, since (3h x 0.4 $/hour = 1.2 $). Our main resource package is PyTorch. We will try to explore more open source methods in NLP with PyTorch to train our model. The TA mentor for our project is Joseph Konan. The full code can be seen in our GitHub
https://github.com/YilangLiu/11785_Project.git

1 Literature Review

In recent decades, speech recognition has been drawing much attention in applications of deep learning. As of today, the accuracy and reliability of automatic speech recognition have improved a lot. Recurrent neural networks (RNN) has been playing an important role in this remarkable progress.

RNN is a kind of neural network that contains a hidden state which can link the input and previous state together. Based on Long short term Memory Networks (LSTM), a variant category of RNN, many automatic speech recognition models have been developed and proven to be accurate. Graves et al. introduced a hybrid model, HMM-Deep Bidirectional LSTM (DBLSTM) that improves frame-level accuracy.[1] DBLSTM makes use of previous and future contexts by setting two separate hidden layers and processing the data in both directions. After proper training, this network shows promising results on TIMIT experiments.

In the work done by Liu, Chaojun, et al.[2], the extensive studies of speaker adaptation of LSTM-RNN models for speech recognition are performed. With different adaptation methods combined with KL-divergence based regularization, observations and analysis indicate that higher performance can be achieved over LSTM baseline model. According to the authors, in a large vocabulary speech recognition task, by adapting only 2.5% of the model parameters using 50 utterances per speaker, we obtained 12.6% WERR on the dev set and 9.1% WERR on the evaluation set, over a strong LSTM baseline model.

The main goal of this study is to find a way to adapt the speaker-independent model effectively while keeping the number of speaker-specific parameters to minimal. There are two approaches investigated: adapting existing network components and adapting inserted affine transformation between layers. The first approach tends to adapting weight matrices inside the recurrent loop and the second one is to insert an affine transform on top of each LSTM layer. Both methods are effective but the first method is adapting the top layer projection matrix gives a large improvement of 9.1% WERR with only 50 unsupervised utterances.

A widely applied example of automatic speech recognition is the automatic captions feature on Youtube. In general, such system works by first recognizing the speech in audio and matching it with an existed vocabulary as much as possible. Then, the generated text is modified according to the context to increase the closeness with original meanings.[3]. We found that reversing the order of the words in all source sentences (but not target sentences) improved the LSTM's performance markedly,[4] because doing so introduced many short term dependencies between the source and the target sentence which made the optimization problem easier. The crucial step is to transform the network outputs into a conditional probability distribution over label sequences. The network can then be used a classifier by selecting the most probable labelling for a given input sequence.[5]

The project will utilize Long short term Memory Networks (LSTM), a variant category of RNN, as the core model.The architecture uses CNNs to reduce the spectral variation of the input feature, and then passes this to LSTM layers to perform temporal modeling, and finally outputs this to DNN layers, which produces a feature representation that is more easily separable[6]. The reason why we use the LSTM model to generate transcripts is that LSTM works better than standard RNNs for speech recognition and image captioning. Due to the property of this dataset which is continuous and lengthy, the LSTM(RNN) fits the task perfectly since the structure of RNN is continuous and RNNs effectively have an internal memory that allows the previous inputs to affect the subsequent predictions. It's much easier to predict the next word in a sentence with more accuracy, if the previous words are known.

2 Project Overview

2.1 Problem Address

In order to enable content creators and media providers to create better NLP applications for animated film, we selected this project to create and train a LSTM model focused on NLP and Computer Vision in anime Kimi no Na Wa.[7] [4] Our main job in the project is to generate a labeled transcript based on the given anime video as the dataset. In the transcript, we first need to translate the dialogue between characters. There is lots of similar research and application on it, but the main problem of us is to improve the efficiency and performance of the translation with multimodal approaches. We also need to distinguish the speakers of the dialogue. During the project, we will show our process of establishing and training our neural networks. In the end, we will report our results for reproducible research. In the process of our project, we have to overcome many engineering problems with putting forward the end-to-end deeping learning solutions.

2.2 Dataset

For this project, the dataset is mainly composed by two parts: the audio section and subtitle section. For audio section, the data type is spectrum which has dimension (128, 1370582); For subtitle section, the data type is a complete sentence, some with labels on who is speaking. There are certain preprocessing needs for this dataset, firstly, the output of the LSTM cannot be a sentence directly, so we need to prepare a letter list as a reference and separate the sentences into characters and transfer every single character into corresponding sequence number in the letter list. The team also needs to add the labels for who is speaking separately since the current label is merged in the subtitles.

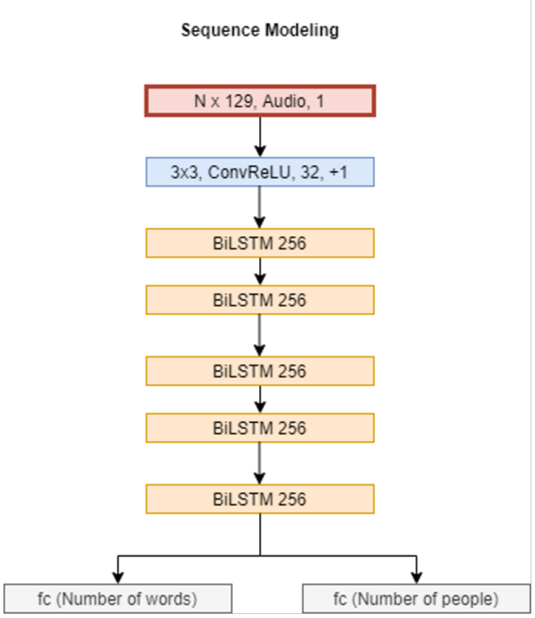

Figure 1: Model Flow chart

2.3 Evaluation Metric

In order to describe the prediction accuracy of our model, we will use CTC loss to evaluate the model performance. After we set up the model and run it successfully, we will have time synchronized outputs given a set of padded input. This output will contain blanks as well as the actual letters. The CTC loss is defined as following:

$$DIV = -\sum_t \sum_r \gamma(t,r) log y_t^{S(r)} \tag{1}$$

In this equation, $\gamma(t,r)$ is the posterior and it can be calculated by forward and backward probability.

$$\gamma(t,r) = \frac{\alpha(t,r)\beta(t,r)}{\sum_{r'} \alpha(t,r')\beta(t,r')} \tag{2}$$

The derivative of the divergence with respect to the output $Y(t)$ can be calculated and backpropagate through the layers.

$$\nabla_{Y_t} DIV = [\frac{dDIV}{dy_t^{S_0}} \frac{dDIV}{dy_t^{S_1}} .... \frac{dDIV}{dy_t^{S_{L-1}}}] \tag{3}$$

The objective of the model is to minimize the divergence between the output and the padded target. The preliminary loss that we currently have is around 2. So, instead of adding more layers and number of hidden size. We decide to use attention structure to better utilize the past knowledge and prioritize the important information to achieve better results.

2.4 Working Baseline

The baseline model chosen for this project is a seq2seq model based on CNN-LSTM architecture. There is one convolutional layer, one batch normalization layer, and one ReLU later. The convoluted data was then fed into a LSTM network for further processing. The LSTM chosen has a structure of bi-directional setting, there are totally 5 layers of the LSTMs. The loss function chosen in the baseline model is CTC Loss Function and the optimizer being chosen is ADAM with 5e-5 of weight decay. The flow chart of the model is shown in Figure 1:

2.5 Current Experiment

We tested the baseline model on the project dataset and successfully get a result, though the initial accuracy is not optimal at all. The baseline model achieves 2.00 validation loss after 30 epochs, which

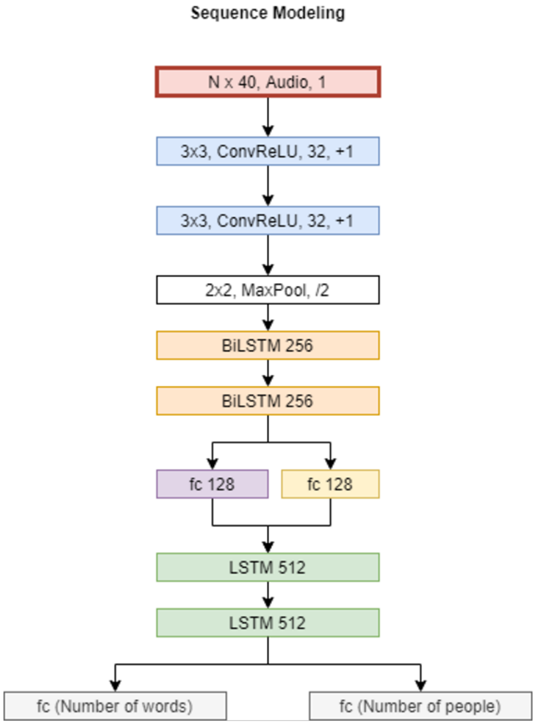

Figure 2: Model Structure

is not performing well. The team realized that the due to the difference between the complexities of the project dataset and the homework dataset(project dataset contains 81 different labels, homework dataset contains 42 different labels), current baseline barely satisfies the requirements, thus the team redesigned a new model which has following structure in Figure2:

2.6 Timeline

From 8 March to 14 March, we have firstly done the literature review on the RNN, LSTM and so on combining detailing reviewing and analysing our dataset. From 15 March to 10 April, after analysing our data and problem, we chose the baseline model based on what we learn, which is the CNN, Bidirectional LSTM and Linear layers to predict and form the transcript of the Kimi no Na wa movie audio record. We have successfully loaded our data and began to train the model. Currently, we have loss value around 2 and try to predict the letters based on the voice input. However, the accuracy is not ideal to precisely generate organized sentence. More work needs to be done to increase the accuracy. Next, we will continue analysing our model and result, and improve the model performance based on changing the parameters of the model, like adding more layers, changing CNN's embedding size and input size, changing dropout parameters and trying different learning rate.

References

[1] A. Graves, N. Jaitly, and A. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–278, 2013.

[2] Chaojun Liu, Yongqiang Wang, Kshitiz Kumar, and Yifan Gong. Investigations on speaker adaptation of lstm rnn models for speech recognition. pages 5020–5024, 03 2016.

[3] Anthony Romero. How does automated closed captioning work?, Nov 2018.

[4] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[5] Alex Graves, Santiago Fernández, and Faustino Gomez. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *In Proceedings of the International Conference on Machine Learning, ICML 2006*, pages 369–376, 2006.

[6] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4580–4584, 2015.

[7] Ralf C. Staudemeyer and Eric Rothstein Morris. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 2019.