Abstract

In the past a few months, I and David were together working on a series of tasks, including simple Audio to Text conversion, software tests on android device, Cloud Infrastructure, Kaldi platform setup, increasing training data, neural network pruning. We also spent time on Iflytek SDK development, but it will not to be discussed in this report. At current stage, we are facing challenge in increasing data and dropping connection.

Task1: Simple Audio to Text conversion

Our first task is to train a model to recognize 10 Chinese words from the Chinese class vocabulary using TensorFlow. The idea of training audio processing model with CNN architecture comes from the revolutional keyword spotting method of Google Inc. New York.[1] With this method, we trained a CNN model using spectrogram[2]. We trained 105,000 WAVE audio datasets through 18,000 epochs. The final accuracy is 88.2%(N=4890). The outcome confusion matrix is given below (Fig1.1). We test this model on android device. It is proved to be able to recognize 10 English words with a relatively high accuracy.

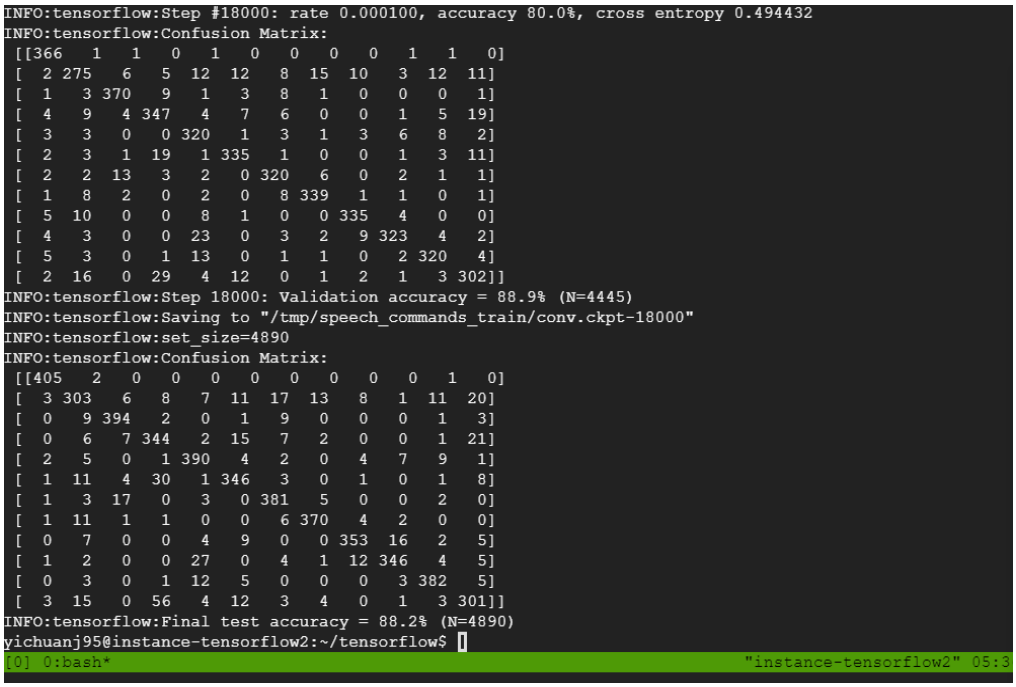

Fig1.1 outcome confusion matrix

Task2: Cloud Infrastructure

Since the training process takes days to complete, we use Google Cloud's GPU to finish the long-time training process. Also, it's more secure to store our model on the cloud. We store our model in different instances across multiple private zones. Tmux is used to master the session in the training process. We keep snapshots for models.

Task3: Network Pruning and connection dropping

We tried to do weight pruning with Keras API. But very soon we realized that Keras weight pruning API only targets MNIST model[3]. However, our trained model is MobileNetV1. It means we can not use it to do compression. Also, we are now struggling how to set

the initial_sparsity, begin_step, end_step and frequency in the pruning_schedule. [4]

https://stackoverflow.com/questions/56470335/is-there-any-keras-code-to-reproduce-the-weight-pruning-of-mobilenet

Task4: Increasing Data

This section focused on making artificial voice data. Since voice data is deficient, generating artificial data becomes a necessary prerequisite for further model training. Using the code from
We are still struggling with debugging the code. The code is based on python 2.7 which will be expired by 2020. Some of the libraries are not compatible in python3.6 and windows environment, so we tried two solutions. The first one is to use another code which is for python3, the second one is to use the linux environment and python2.7 on the virtual machine.

Notations:

[1]: https://www.tensorflow.org/tutorials/sequences/audio_recognition

[2]: Convolutional Neural Networks for Small-footprint Keyword Spotting, ara N. Sainath, Carolina Parada Google, Inc. New York, NY, U.S.A

[3]: https://github.com/tensorflow/tensorflow/issues/29163

[4]: https://stackoverflow.com/questions/56470335/is-there-any-keras-code-to-reproduce-the-weight-pruning-of-mobilenet

References:

1. Convolutional Neural Networks for Small-footprint Keyword Spotting, ara N. Sainath, Carolina Parada Google, Inc. New York, NY, U.S.A
2. J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, ""Your word is my command": Google search by voice: A case study," in Advances in Speech Recognition, A. Neustein, Ed. Springer US, 2010, pp. 61–90.
3. G. Chen, C. Parada, and G. Heigold, "Small-footprint Keyword Spotting using Deep Neural Networks," in Proc. ICASSP, 2014