Machine Learning Engineer Nanodegree

Capstone Proposal – Speech Recognition Using Tensorflow

Avinash Benki
October 8th, 2019

Proposal

Domain Background

Speech Recognition systems have a history of around 50 years. The very first speech recognition system named Audrey was developed in 1952. Audrey was only designed to recognize digits. IBM ShoeBox was launched 10 years later by IBM which was capable of recognizing 16 words including digits. Major contribution was made by Defense Advanced Research Projects Agency (DARPA). DARPA developed Harpy which was able to recognize 1011 words.

Major breakthrough in 1980s, the Hidden Markov Model (HMM) was applied which was a statistical model used to model the problems involving sequential information. In the year 2001, Google introduced Voice search in its search engine. 2011 saw the launch of Siri by Apple which offered a real-time, faster and easier way to interact with Apple devices by using voice. Personally, I will be moving into a new role as a speech engineer this project will motivate me to take my first step towards the role.

Problem Statement

To build an algorithm that understands simple spoken English commands using the Speech Commands Dataset. Speech Detector is hard to build using free, open data and code. Neural Network model require preprocessing of the voice datasets. Tensorflow released the Speech Command Datasets to counter the problems faced while building a simple model for speech recognition.

Datasets and Inputs

Dataset was obtained from Tensorflow. It includes 65,000 one-second long utterances of 30 short words, by thousands of different people. The audio files were collected using crowdsourcing [1].

The dataset contains a set of one-second .wav audio files, each containing a single spoken English word. These words were spoken by a variety of different speakers and the words are from a small set of commands. This dataset is best suited for speech recognition as it is well organized and easy to start with enough data size.

The core words which were collected are "Yes", "No", "Up", "Down", "Left", "Right", "On", "Off", "Stop", "Go", "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", and "Nine". Each word was recorded five times by each speaker. Ten auxiliary words were are also recorder to help distinguish unrecognized words. These include "Bed", "Bird", "Cat", "Dog", "Happy", "House", "Marvin", "Sheila", "Tree", and "Wow" which were recorded only once by each speaker.

Solution Statement

The solution proposed is an algorithm containing recurrent neural network (RNN) to accurately predict the words. The neural network architecture used the convolutions to extract the short term dependencies, RNNs and attention to extract long-term dependencies.

The main intention of using Long Short-Term Memory (LSTM) layers which are a type of recurrent neural network architecture is that they remember past information at the current time point which intern influences their output. Speech recognition is temporal in nature and the LSTM's are suitable for such contexts.

Benchmark Model

I will be using the following accuracy table as the benchmark models from the*]. Res results from Warden (2018) [2]. ConvNet on raw WAV results from Jansson (2018 [3]. Depthwise Separable Convolutional Neural Network (DS-CNN) from Zhang et al. (2017)[4].*

Table 1: Google Speech Command Dataset accuracy results.

Model	Accuracy(%) – 20 Commands
Res15	95.8
Res26	95.2
Res8	94.1
DSCNN	95.4
ConvNet on raw WAV	89.4

Evaluation Metrics

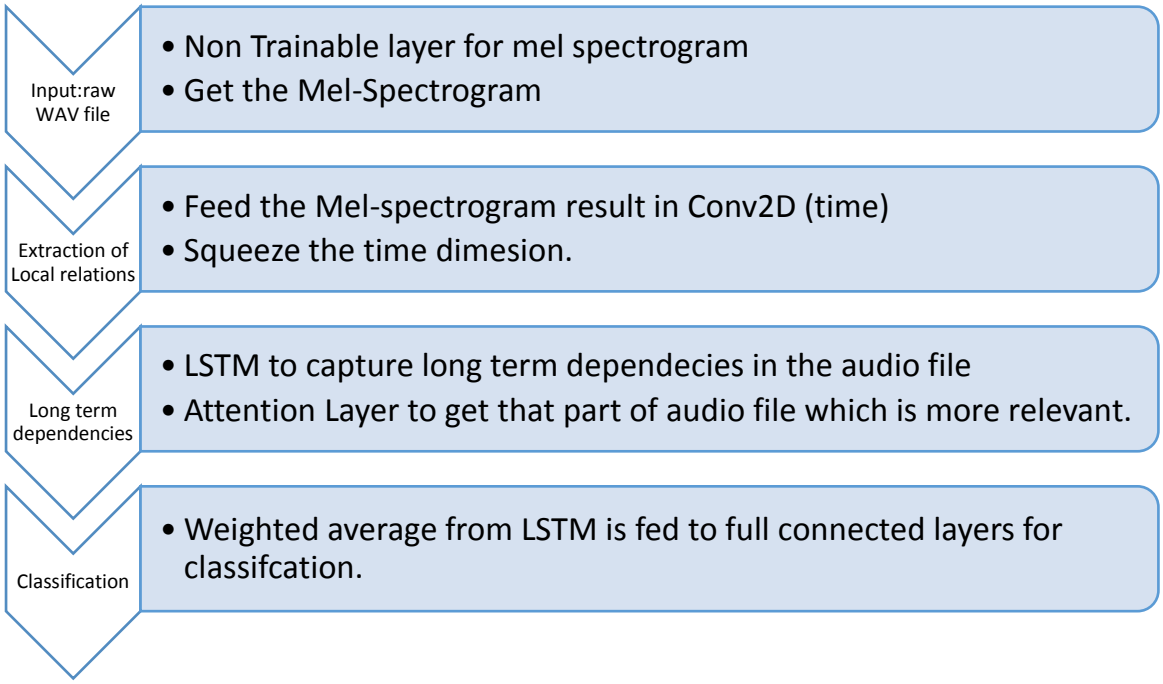
The evaluation metrics for this algorithm would be the multiclass accuracy. The multiclass accuracy is the average number of observations with correct label. There are 12 possible labels for the Test set: yes, no, up, down, left, right, on, off, stop, go, silence, unknown. The algorithm has to classify a label as unknown if the command is not one of the first 10 labels or that is not silence.

Project Design

Theoretical Workflow

- Gathering and loading the Speech Command Dataset provided by the Tensorflow. The dataset provided has WAVE format with 16000 sample rate and a duration of 1 second for each utterance.
- Data Visualization in order to have a look on the features of the dataset. Wave and spectrogram along with MFCC will be used for the visualizations along with some other techniques.
- The audio clips haven't been separated into training, test, and validation sets explicitly, but by convention a hashing function is used to stably assign each file to a set. Hence division of dataset into training, testing and validation sets.
- Defining the model architecture.
  I will be using the RNNs with attention mechanism. The model starts by computing the mel-scale pectrogram of the audio using non trainable layers. Input to the model is raw WAV file with sampling rate of 16Khz. Mel –scale spectrogram is computed using 80-band mel scale and 1024, discrete Fourier transform points and hop size of 8 seconds.

- Compile and Train the model. Get the classification accuracy and based on the validation set accuracy tweak the parameters accordingly in order to improve accuracy.
- Calculate the Accuracy on Test Set.



Intended Work

- Build a RNN model for speech recognition.
- Explore different LSTMs and apply the best suited LSTMs for audio files.
- Complete the Model Architecture and improve the accuracy. An accuracy of >80% should be expected result.

References

[1]. aiyprojects.withgoogle.com/open_speech_recording – Crowdsourcing by Google.

[2]. Warden, P., 2018. Speech commands: A dataset for limited-vocabulary speech recognition. CoRR abs/1804.03209. URL http://arxiv.org/abs/1804.03209

[3]. Jansson, P., 2018. Single-word speech recognition with convolutional neural networks on raw waveforms. URL https://www.theseus.fi/bitstream/handle/10024/144982/Jansson_Patrick.pdf?sequence= 1

[4]. Zhang, Y., Suda, N., Lai, L., Chandra, V., 2017. Hello edge: Keyword spotting on microcontrollers. CoRR abs/1711.07128. URL http://arxiv.org/abs/1711.07128