Machine Learning Engineer Nanodegree

Capstone Proposal

Raafat Abualazm

27/10/2019

Domain Background

Speech recognition is an integral part of human-computer interfaces (HCI). They are present in personal assistants like Google Assistant, Microsoft Cortana, Amazon Alexa and Apple Siri to self-driving car. It is also an easily accessible method of interacting with the computers for people with disability. An accurate speech recognition is vital to avoid unnecessary hardships and to provide a safer experience for critical applications. Past work in this area used something called Hidden Markov Models to predict speech and the command. The project falls under the category of speech recognition. I am personally motivated for this project, because I am working on a NLP project at college and we need a speech recognition step.

Problem Statement

The problem to be solved is rather simple, however, it serves as the basis of any more complicated system. In short, human speech is composed of short utterances and silent sounds, to be able to recognize various commands, one must be able to recognize each utterance, around 1 second each. I, here, confine myself to recognition of each utterance and individual words and maybe very simple sentences. I need to build a model that predicts as much as possible right words from voice. This is a classification problem in which the algorithm takes a sound file and produces an output, a word that is most likely to have been in the file.

Dataset

The dataset used here is Google Speech Commands dataset. It contains over 65,000 utterances, around 1s each, of around 30 words from thousands of speakers, collected through Google's AIY website. The recordings are recorded under various circumstances and from many devices, all in a quiet room though to simulate the actual setting in which speech recognition might be used. It's released under a Creative Commons BY 4.0 license. It is an excellent starting point to learn how to develop a speech recognition model. The dataset will be used to train a Deep Learning model, as intended and authorized by Google.

Solution Statement

The solution will be quite unorthodox. For this, rather simple, problem, I will be using a CNN to be able to predict the word from its spectrogram. That is, we will visually represent each word with a unique spectrogram and train a CNN on it, which is widely used for image classification. We reduced the speech recognition problem to an image classification problem! The expected output will be a word, that is our label. We have 30 words in this dataset and a successful model should classify utterances into one of these words, our classes, as correctly as possible.

Benchmark model

The model will be benchmarked against Google’s Neural Network trained on this same dataset. It achieved an accuracy score of 94%. We can run an accuracy scorer on my model and Google’s model and compare the results of each.

Evaluation metrics

The metric used for evaluation will be accuracy score. That is: score = $\frac{Right\ predictions}{Total\ sample\ size}$ as we are interested in increasing the sheer number of right predictions and the need for other more involved scoring methods is not needed due to the nature of the problem.

Project Design

The project shall follow the following strategy:

1. Split the data into training and validation datasets using train test split method.
2. Pre-process every wav file to compute its log spectrogram through computing Short Time Fourier Transform then calculating the log for amplitudes. (Using Scipy.signal.stft and np.log)
3. Feed the resultant spectrogram into the CNN to be trained on it. (An input layer followed by 2 to 3 convolutional layers interleaved by Average or Max Pooling layers, then 1 to 2 dense layers that will act as an output layers)
4. A Convolutional Neural Network will be used to classify words.
5. Score the training using accuracy score.
6. Run the validation set and score the result.
7. Improve upon the design of the CNN till satisfactory performance.