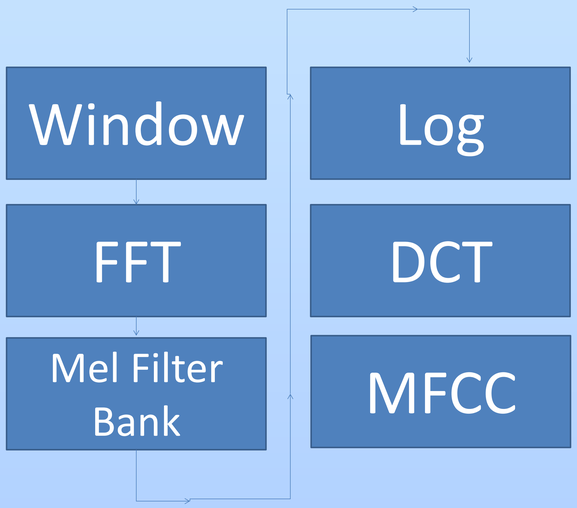
Speech Recognition System
by Isaac Mosebrook

Abstract

One of the most prominent applications of audio signal processing in the modern era is speech recognition. To perform speech recognition, we must acquire an audio input, evaluate its features, and then feed it into some kind of model, such as a neural net. The output of this system will give the word spoken, such as “on” or “off”.

Feature Extraction

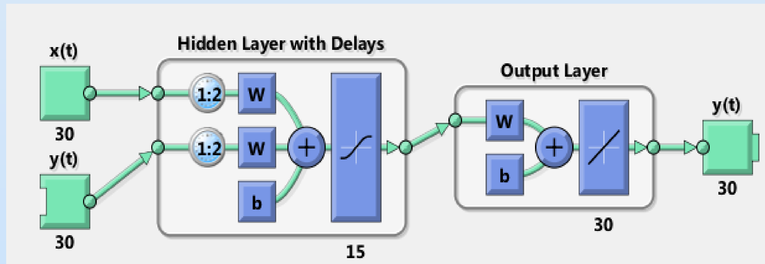
The raw audio waveform is too specific and dependent on time, so we must transform the input to be more general. The most common and effective feature set chosen for speech applications is Mel-Frequency Cepstral Coefficients (MFCC).



Neural Networks

Our recognition system is a neural network. This is a series of interconnected nodes where each node is a weighted sum of all the previous nodes. We also use the previous output as a secondary input. By training the network, it will learn the optimal weights and bias.

$$y = \sum w_1 * x_i + w_2 * y_{i-1} + b$$



In the case of this project, the input at each time step is a vector of 30 MFC Coefficients. The output is vector of length 30. Each node represents the probability of a particular command being stated. We use only one hidden layer, so there are a total of three weight matrices and two biases.

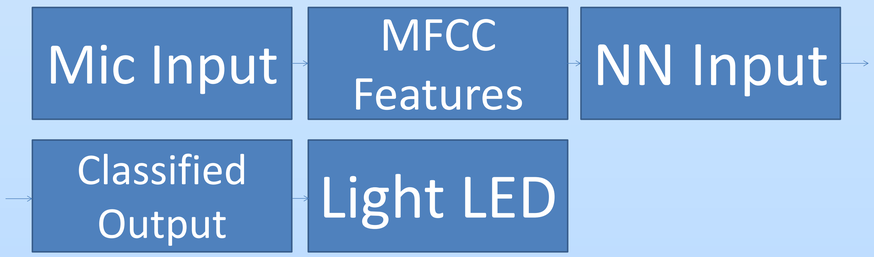
Neural Network Training

To train the neural network, I used Google’s speech command dataset, which contains about 2000 recordings per command. Matlab’s neural network toolbox was used to perform gradient backpropagation on the dataset. The model was trained until the classification error was under 5%. Note that no training is done in real time.

Acquisition / Hardware

To get the speaker’s input, we must interface a microphone to our digital system. We can use an Arduino to capture microphone data in real time and send it to Matlab. Additionally, Matlab can access the Arduino’s hardware to give real life feedback (lighting an LED) when a command is recognized.

Final Processing Flow



Future Work

- Add pre-processing to de-noise microphone input
- Train on larger set of commands
- Implement more tasks for system to perform after recognizing command