Speech Recognition Project Proposal

Phillip Spratling

Speech recognition technology has come a long way in the last couple decades. Until recently, it was predominantly performed using methods such as hidden Markov models, but now deep learning neural networks have proven to be much more effective. Because of the improvement in performance, speech recognition in devices is begging to show up everywhere. Previously used mostly in military applications and telephone call processing, you can now find sophisticated speech recognition being used in smartphones, cars, televisions, and even refrigerators.

Despite the recent surge in popularity in speech recognition technology, most independent makers and entrepreneurs have had difficulty building a simple speech detector. Neural networks require a significant amount of data to be effective, and gathering (not to mention processing and cleaning) voice clips of hundreds or thousands of people saying multiple phrases multiple times is not exactly easy. Thankfully, TensorFlow recently released the Speech Commands Datasets, which include 65,000 one-second long utterances of 30 short words, by thousands of different people.

In this project, I will be using this dataset to build a speech recognition algorithm that understands 10 of these words and will understand whether a sound clip is silent as well. It will also know to classify sounds that aren't one of these 11 categories as "unknown". These words were chosen to be used as a speech interface, so they are mostly commands such as "yes," "no," "stop," "go," etc. The goal is to build an algorithm that could be used in a simple speech recognition application, or for the foundation of a more sophisticated application.

To build the algorithm, I will process the sound clips as spectrograms and into Mel-frequency cepstral coefficients for use as input into my neural networks. These can be treated as images, and thus will be used as inputs for a convolutional neural network. A recurrent neural network composed of long short-term memory (LSTM) units will be built as well,

and the better performing model will be kept. I will use TensorFlow with Keras to build the neural networks.

To improve model performance and better simulate speech in noisy environments, I will also create new samples from the original voice clips mixed with some background noise. I will also create samples with a “time shift” to simulate the voice beginning in earlier or later times in the clip. These will effectively quadruple the size of the input data, and will hopefully produce better accuracy as a result.

Deliverables:

- All the code written throughout the project delivered in IPython notebooks
- Final Report detailing all my findings and methods
- A PowerPoint presentation