NTDS Project Proposal

Kiran Bacsa, Rabeeh Karimi Mahabadi, Adrian Löwenstein, Manuel Vonlanthen

November 28, 2017

1 Description

The project is based on the kaggle competition 'TensorFlow Speech Recognition Challenge'[1]. The objective of this competition is to design a speech recognition algorithm that is able to recognize simple speech commands such as 'up','down',etc... In order to participate in this kaggle competition, the open-source software library TensorFlow has to be used [2].

2 Dataset

TensorFlow recently released the Speech Commands Datasets. It includes 65,000 one-second long utterances of 30 short words, by thousands of different people. More specifically, this is a set of one-second .wav audio files, each containing a single spoken English word. These words are from a small set of commands, and are spoken by a variety of different speakers. The audio files are organized into folders based on the word they contain. The audio files were collected using crowdsourcing. The goal was to gather examples of people speaking single-word commands, rather than conversational sentences, so they were prompted for individual words over the course of a five minute session. Twenty core command words were recorded, with most speakers saying each of them five times. The core words are "Yes", "No", "Up", "Down", "Left", "Right", "On", "Off", "Stop", "Go", "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", and "Nine". To help distinguish unrecognized words, there are also ten auxiliary words, which most speakers only said once. These include "Bed", "Bird", "Cat", "Dog", "Happy", "House", "Marvin", "Sheila", "Tree", and "Wow".

3 Tentative approach

In our project, we want to design a solution to the given speech classification problem using the graph methods we learned during the lectures. A first approach we want to try would be to classify the different core words similarly to what we have seen in assignment 3, i.e. extracting speech features (e.g. mel-cepstral coefficients) from the audio signals, build a graph using a distance function on the feature vectors and finally extract the fiedler vectors from the Laplacian of the graph to cluster the graph. Additionally, we will look further into speech processing using semi-supervised learning on graphs to try to improve our solution.
Even though we will probably use TensorFlow as a tool (as required for the kaggle challenge), we will not constrain ourselves to this single library.

[1] https://www.kaggle.com/c/tensorflow-speech-recognition-challenge
[2] https://www.tensorflow.org/