**Machine Learning Engineer Nanodegree**
**Capstone Proposal**
**Geoffrey Hung**

## Domain Background

In this Kaggle competition, I will need to use the Speech Commands Dataset from Google to build an algorithm that understands simple spoken commands such as "yes", "no", "up", "down". By improving the recognition accuracy of open-sourced voice interface tools, we can improve product effectiveness and their accessibility.

This problem becomes more relevant to IoT products such as Google Home and Amazon Echo. Most of the in-room device requires the products to understand the human voice in order for them. As such, this project will try to build the algorithms to predict the words based on audio data.

## Problem Statement

The problem is to train the model to classify the audio to different short commands, there are different approachs to this problem, I prefer to try it out with deep learning where I can use what I learned in lessons to do prediction. Also, audio data can be transferred to spectrogram which I can combine CNN as part of the toolkit models.

## Datasets and Inputs

datasets:
The audio files were collected using crowdsourcing by Google, see aiyprojects.withgoogle.com/open_speech_recording for some of the open source audio collection code they used. 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".

Reference:
https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/data

**Solution Statement**
The problem will be to train the model to classify the audio to different short commands, there are different approachs to this problem, I prefer to try it out with deep learning where

audio data can be transferred to spectrogram and I can use CNN as part of the toolkit models.

**Benchmark Model**
According to Google tutorial, their models after hours of training on GPU can achieve accuracy between 85% and 90%, the goal for this project will be to achieve the similar accuracy level as Google did with the model I developed myself.

reference: https://www.tensorflow.org/versions/master/tutorials/audio_recognition

**Evaluation Metrics**
Metric will be classification accuracy, this is to count percentage of correctness in classifying the short audio. Accuracy = #(classify correctly) / #(test size)

**Project Design**
My approach to this problem will be divided into following steps,

1. data exploration
In this part, I will refer to different online materials on how to handle audio data. Audio data is different from image or text which I need another representation to convert audio data. I will also explore and visualize the patterns of different words at this stage.

2. data preparation
In this part, I will split the data to training, validation and test set so I can test my final model fairly. Also, I will apply the representation I picked in part 1 to convert audio data model building

3. model prototype
In this part, I will try to build different model in small size to confirm the model can at least overfit the data in small size so I know it maybe able to learn something from data.

4. model building
In this part, I will use GPU to train the model to full set of training data. I will monitor the performance along with validation result and test the final model with test set.

5. error analysis and conclusion
In this part, I will analyse the error source, are they distributed evenly and is there any correlation b/w them. I will compare my result with Google tutorial benchmark at last as a final comparison