Report

Shubham Shrivastava

February 21, 2021

- **OVERVIEW:**

  If your mum asks you to call your dad and tell him to bring some required groceries while you're busy doing dishes with your music on. Let me ask you, how are you gonna do that? I know, right? Google assistant came to your mind or Siri if you can afford an apple product. Speech recognition is not that much of an alien concept in this modern era. It involves various methodologies and technologies which allow recognising and translating spoken words into text speech from a verbal format to a text one with the help of various machine learning algorithms which first train it. It provides a plethora of advantages which make our life much simpler which involves to make a phone call, play music and even complex tasks like financial transactions.

- **Model Used:**

  Speech recognition can be done by various methods like convolution, LTM etc. We have used Simple Feed Forward neural network architecture to train our data.

  It is commonly known as a multi-layered network of neurons, feed forward neural networks are called so due to the fact that all the information travels only in the forward direction.

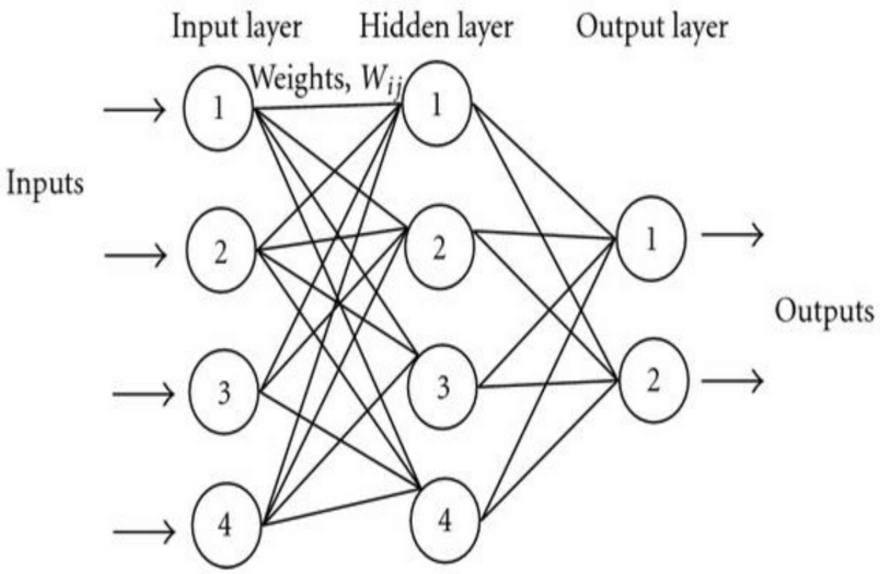
  The information first enters the input nodes, moves through the hidden layers, and finally comes out through the output nodes. The network contains no connections to feed the information coming out at the output node back into the network.

Feedforward neural networks are meant to approximate functions. Here's how it works.

* There is a classifier y = f*(x).

* This feeds input x into category y.

* The feedforward network will map y = f (x;). It then memorizes the value of that approximates the function the best.



Feature Extraction and acuuracy

- **MFCC(Mel-Frequency Cepstral-Coefficients) :** MFCC is a feature extraction technique. In audio analysis this process is largely based on finding components of an audio signal that can help us distinguish it from other signals.

  We have understood the importance of MFCC up untill now. It facilitates the process of feature extraction which in turn increase the accuracy because we can't rely on factors like amplitude, frequency etc to calculate accuracy. Preprocessing truncates the data prior having to do MFCC which helps a lot because after that we can extract important features more easily.

- **Visualization of audio signals :**

  Audio signals as data are not easy to processed having some factors we know nothing about. Machine doesn't understand human voice or texts as some ordinary human would think. As data analyst, we need a mechanism which would unravel its properties.

  Spectrogram : A spectrogram is a visual way of representing the signal strength, or "loudness", of a signal over time at various frequencies present in a particular waveform. Not only can one see whether there is more or less energy at, for example, 2 Hz vs 10 Hz, but one can also see how energy levels vary over time.
  A spectrogram is usually depicted as a heat map, i.e., as an image with the intensity shown by varying the color or brightness.

  In our code, we used pylab(python library used for visualization) to get our desired spectrograms. Code for required purpose is provided.

Description:

- **Layers Of Feed Forward Neural Architecture :**

* **Input layer :**
This layer consists of input data which is being fed to the neural network. This layer is projected as like it contains neurons but they are not same as artificial neurons with computational abilities.

* **Hidden layer :**
  This is the layer that consists of the actual artificial neurons. In deep neural network,output of first hidden layer is generally fed to next hidden layer. The hidden layers are used to increase the non-linearity and change the representation of the data for better generalization over the function.

* **Output layer :**
  This layer is used to represent the output of the neural network. The number of output neurons depends on number of output that we are expecting in the problem at hand.

- **Weights and Bias :**
  The neurons in the neural network are connected to each other by weights. Apart from weights, each neuron also has its own bias. Weights ranges from 0 to 1.

* Our model works on the basis of epocs. An **Epoch** is defined as one complete cycle through the training dataset and indicates the number of passes that the machine learning algorithm has completed during the training. For better accuracy, we have used 50 epochs. You're welcome if you can increase its accuracy by changing the number of epochs. We leave that part upto you.

- **Backpropagation:**

* During the training phase, the neural network is initialized with random weight values. Training data is fed to the network and the network then calculates the output. This is known as a forward pass.

* The calculated output is then compared with the actual output with the help of loss/cost function and the error is determined.

* Now comes the backpropagation part where the network determines how to adjust all the weights in its network so that the loss can be minimized.

* This weight adjustment starts happening from the rear end of the network. The error is propagated in the backward direction to the front layers till the end and the neurons across the network start adjusting their weights. Hence the name backpropagation.

- **Activation functions used :**

* **ReLU :** Rectified linear unit is the most commonly used activation function in neural networks. ReLU is linear (identity) for all positive values, and zero for all negative values.

* **Softmax :** Softmax is a type of activation function that turns numbers into probabilities that sum to one. Softmax function outputs a vector that represents the probability distributions of a list of potential outcomes.

* **Loss Function Used :**

We used the **Categorica Cross entropy** loss function. Categorical crossentropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one.

Formally, it is designed to quantify the difference between two probability distributions.

* **For getting output in human readable format :**

  We humans may have developed big time. But there are some limitations when it comes to competing with machines(no offense though). The output of our model is going to be in numerical format. Thus, we will be needing a method for this very purpose.

  The method is provided in code. In that, we're doing no rocket science. Model is saving unique ids of inputs it is being fed and converting categorical data to non-categorical(basically texts.)

- **What kind of Data do we have :**

  1: we have set 16KHz as sampling rate.
  2: We have 80 utterances of each command.

* Data/forward.
* Data/back.
* Data/left.
* Data/right.
* Data/stop.

- **Language Used :**

  PYTHON 3

- **Libraries Used:**

  Librosa, scikit learn, sound device, winsound, matplotlib, tensorflow and some other useful modules.

- **Did I have fun? let's find out.**

* I had a lot of fun doing this project. When it gave the outputs having done a lot of brain storming, that was a moment anyone could die for(might be exaggerated I know). Although research part had been a bit tricky but I learned new things that added icing on the cake.