Voice Recognition through Machine Learing

Raktim Gautam Goswami[1], Abhishek Bairagi[2] & G V V Sharma[3]

Contents

***Abstract*—This manual shows how to develop a voice recognition algorithm and use it to control a toycar.**

1 Dataset

1.1 Record 'forward' 80 times using you phone and save as 'forwardi.wav' for $i = 1, \ldots, 80$.

1.2 Repeat by recording 'left', 'right', 'back' and 'stop'. Make sure that the audio files for each command are in separate directories. Download the following directory for reference

```
svn checkout https://github.com/gadepall/
    EE1390/trunk/AI−ML/audio_dataset
```

1.3 Use the following script to generate a dataset for 'back' command. Explain through a block diagram.

```
https://raw.githubusercontent.com/gadepall/
    EE1390/master/AI−ML/codes/250files.py
```

The authors are with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India . e-mail: 1. ee17btech11004@iith.ac.in, 2. ee17btech11051@iith.ac.in, 3. gadepall@iith.ac.in

Solution: to generate the dataset needed for training. The following diagram explains how this is done for the back command.

back (80 files) $\overset{250files.py}{\rightarrow}$ $25000files$.

1.4 Suitably modify the above script to generate similar datasets for 'left', 'right', 'stop' and 'forward'.

1.5 Store the complete dataset in a directory and run **code.py** from within the directory. Note that this should be done on a powerful workstation. This will generate two files **W1.out** and **b.out**.

2 Implementation

2.1 Execute **record.py** and issue any of the commands 'forward', 'left', 'right', 'back' and 'stop'. The output will be as per Table **??**.

2.2 Install Google API "Arduino Bluetooth Controller" using google play-store

2.3 Open the app and connect to HC-05.

2.4 Open voice control section in the app and tap to give following commands.
Left, Right, Forward, Back & Stop

3 Building the neural network

3.1 Problem Statement

3.1.1 Consider $\mathbf{x}$ be 4043×1 to be human voice issuing either 'forward', 'left', 'right', 'back' and 'stop'. Let $\mathbf{W}$ be 4043×5 and $\mathbf{b}$ be 5×1. $\mathbf{W}$ and $\mathbf{b}$ are the machine parameters. Then the machine makes a decision based on

$$\hat{\mathbf{y}} = \mathbf{x}^T\mathbf{W} + \mathbf{b} \tag{3.1}$$

3.1.2 The problem is to estimate $\mathbf{W}$ and $\mathbf{b}$. This is done by considering

$$\min_{\mathbf{W},\mathbf{b}} J(\mathbf{W},\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \hat{\mathbf{y}}\|^2 \tag{3.2}$$

3.2 Solution: Gradient Descent

3.2.1 $\mathbf{W}$ and $\mathbf{b}$ can be estimated from (3.2) using

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \frac{\alpha}{2}\frac{\partial J(\mathbf{W},\mathbf{b})}{\partial \mathbf{W}} \tag{3.3}$$

$$\mathbf{b}(n+1) = \mathbf{b}(n) - \frac{\alpha}{2}\frac{\partial J(\mathbf{W},\mathbf{b})}{\partial \mathbf{b}} \tag{3.4}$$

Show that (3.3) can be expressed as

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \alpha\left[\mathbf{x}^T(n)\mathbf{x}(n)\mathbf{W}(n) + \mathbf{x}^T(n)\mathbf{b}(n) - \mathbf{x}^T(n)\mathbf{y}(n)\right] \tag{3.5}$$

$$\mathbf{b}(n+1) = \mathbf{b}(n) - \alpha\left[\mathbf{x}\mathbf{W} - \mathbf{b} - \mathbf{y}\right] \tag{3.6}$$

Solution: From (3.2) and (3.1),

$$J(\mathbf{W},\mathbf{b}) = \frac{1}{2}\|\mathbf{y} - \hat{\mathbf{y}}\|^2 \tag{3.7}$$

$$= (\mathbf{x}\mathbf{W} + \mathbf{b} - \mathbf{y})^T(\mathbf{x}\mathbf{W} + \mathbf{b} - \mathbf{y}) \tag{3.8}$$

$$= \left(\mathbf{W}^T\mathbf{x}^T + \mathbf{b}^T - \mathbf{y}^T\right)(\mathbf{x}\mathbf{W} + \mathbf{b} - \mathbf{y}) \tag{3.9}$$

$$= \mathbf{W}^T\mathbf{x}^T\mathbf{x}\mathbf{W} + \mathbf{W}^T\mathbf{x}^T\mathbf{b} - \mathbf{W}^T\mathbf{x}^T\mathbf{y} \tag{3.10}$$

$$+ \mathbf{b}^T\mathbf{x}\mathbf{W} + \mathbf{b}^T\mathbf{b} - \mathbf{b}^T\mathbf{y} - \mathbf{y}^T\mathbf{x}\mathbf{W} \tag{3.11}$$

$$- \mathbf{y}^T\mathbf{b} + \mathbf{y}^T\mathbf{y} \tag{3.12}$$

Using

$$\frac{\partial}{\partial \mathbf{W}}\mathbf{W}^T\mathbf{x}^T\mathbf{x}\mathbf{W} \tag{3.13}$$

3.3 Python code

https://github.com/raktimgg/ML-algorithm-for-speech-recognition/blob/master/code.py
This is the full code that is used for training. The accuracy we are getting is around 98 percent.

3.4 Dataset

We have made our own dataset by recording 25 samples of each word. Each of these samples are recreated by adding empty elements in the front and back in many different cobinations to create a dataset of 6250 samples for each word. All the audio files are imported to an array in the code and converted to mfcc format before training. For creating training dataset we recorded 25 audio file of each of the following word -
1)Forward
2)Left
3)Right
4)Back
5)Stop
The code for generating 6250 samples for each word from 25 samples can be found in the github link attached.
https://github.com/abhishekbairagi/Making-Dataset-for-ML/bl

4 Transfering the weights to Raspberry Pi (Yet to be done)

The weight(W1 and B) are saved in a file at the end of the code. These weights will be transferred to the raspberry pi and a simple program written, will record audio on the raspberry pi, do the calculations using the weights and predict the text output. This output will be sent,using bluetooth, to the toy car, which will move accordingly.