

Speech Recognition Using Artificial Neural Network

MCE 6646 Intelligent Control System

Shovan Chowdhury
Department of Mechanical Engineering
Idaho State University
Advisor: Dr. Marco Schoen

OBJECTIVE

The main objective of my project is to detect speech using artificial neural network. There are many other way to recognize voice such as Biometrics, Dynamic time wrapping etc. But machine learning is becoming more and more popular in the world. That's why I decided to recognize speech using artificial neural network. In this project, my target was to detect the speech of speaker. There are some other approach of detection where they showed detection of speaker by recognizing their voice. But my approach was different. I wanted to detect speech what speaker says. This project shows the method of detecting speech of speaker by using previous trained voice sample of the speaker. This training was done by artificial neural network. For training purpose, we have used Matlab as coding platform. With this project, one can learn about voice recording in Matlab, extract various feature of voice signal using digital signal processing, how to convert time domain to frequency domain, how to use frequency as training input of neural network and finally detailed algorithm of artificial neural network method.

Methodology

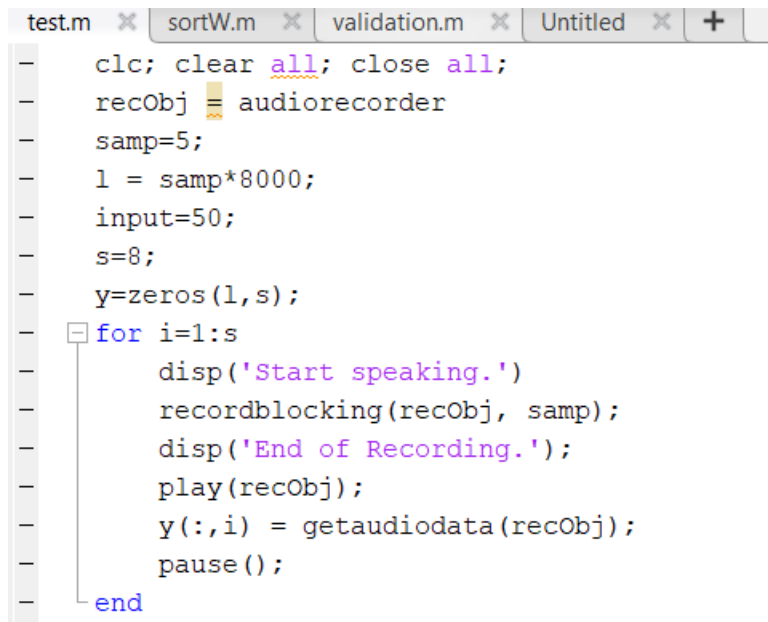
1. Introduction:

As I planned to detect speaker speech, I needed to plan what speech I should detect. I selected '0' and '1' for detection. It is like that if a speaker says '0', then our system should detect that speaker says '0' and vice-versa. I worked with just one speaker voice. If I want to detect all universal voice, then I need to take huge voice sample. As there are limitation for the project, I just worked with my voice. That means system just detect my voice. It will just detect '0' and '1' said by me. It will work perfectly if this speech is said by me. In this whole project, I firstly recorded my voice sample in Matlab, then extract feature from that voice signal for each sample, then trained the sample with my desired output and finally validation of the neural network system. So I divided this project into four phases. They are as follows.

- a) Recoding voice in Matlab
- b) Feature extraction and sorting using digital signal processing
- c) Neural network training
- d) Validation of the build up network

a) Recording Voice in Matlab:

First of all, I needed to know how to record voice in Matlab. I needed to record some sample voice for providing training input in neural network. So I planned to take 8 voice sample of me by saying '0'; and '1'. I said four times '0' and four times '1'. The recording procedure is shown below

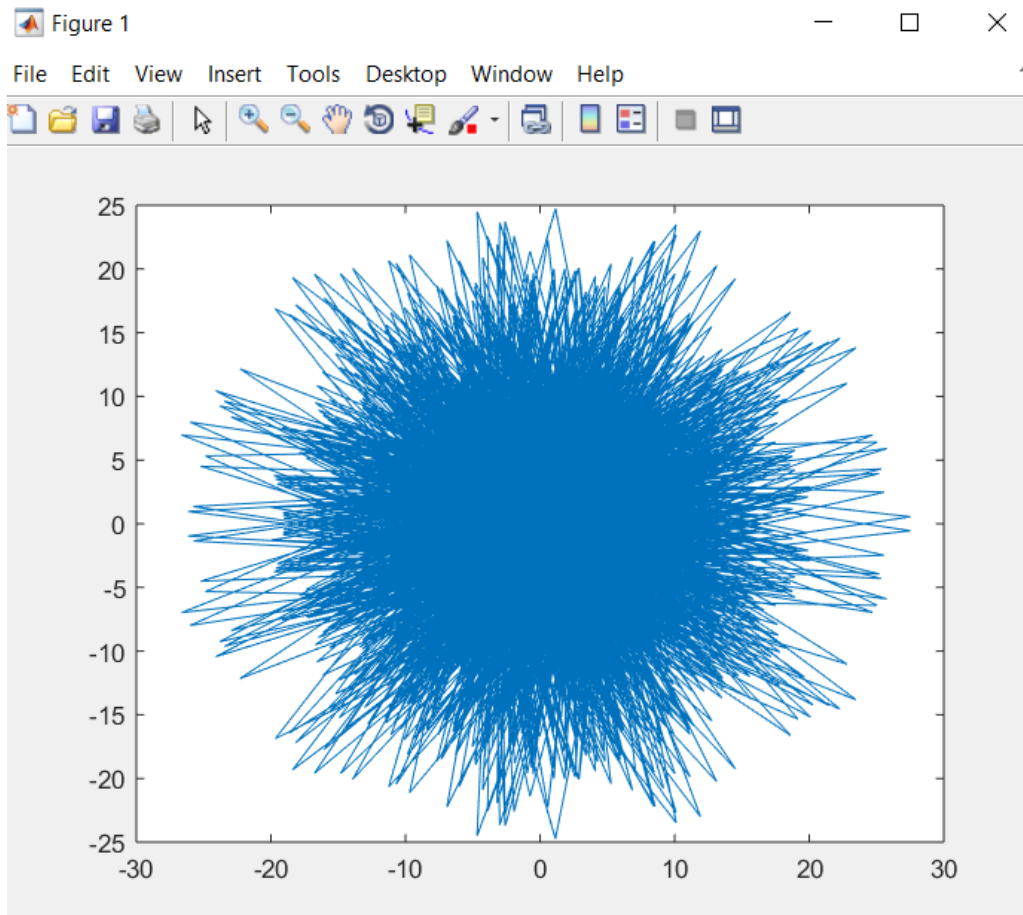


```
test.m x sortW.m x validation.m x Untitled x +
-   clc; clear all; close all;
-   recObj = audiorecorder
-   samp=5;
-   l = samp*8000;
-   input=50;
-   s=8;
-   y=zeros(1,s);
-   for i=1:s
-       disp('Start speaking.')
-       recordblocking(recObj, samp);
-       disp('End of Recording. ');
-       play(recObj);
-       y(:,i) = getaudiodata(recObj);
-       pause();
-   end
```

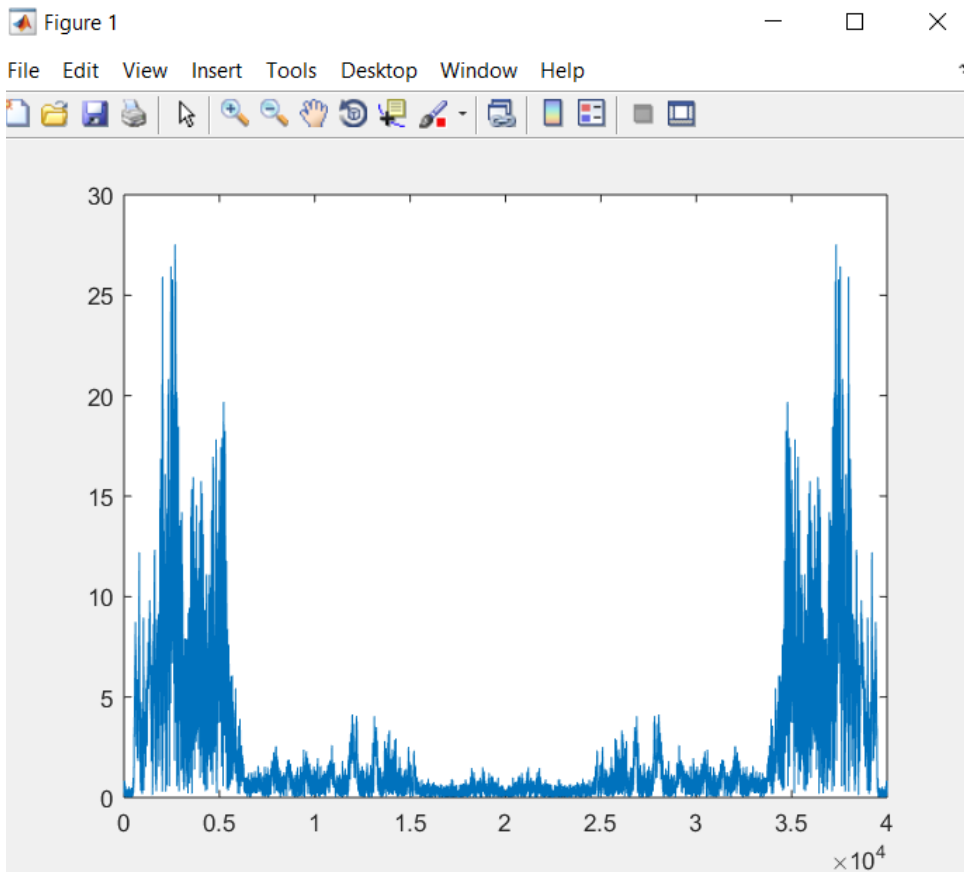
In this coding, 'samp' means how long you record your voice. I took 5 second for each voice sample to record. That's why I took the value of "samp" is 5. And I took 8 voice sample and for that reason the value of "s" is 8. For each second of voice signal, it contains 8000 bit. So it takes 40000 bit for each 5 second of voice sample. Each voice sample is stored in a row matrix 'y'. I used this sample matrix as input in neural network. But before that we need to covert each voice signal from time domain to frequency domain.

b) Feature extraction and sorting using digital signal processing:

In this phase of the project, I needed to extract features of the voice signal. For doing that, first of all I needed to take each signal from time domain to frequency domain by fast fourier transformation. The following figure I have got from the fast fourier transform.



This figure contains complex portion of the signal. But we want only real value of the signal. That's why we use 'abs' command in matlab for taking only absolute real value. After doing that, we have got a good frequency graph. The figure is as belows.



For converting all voice sample signal to this frequency domain and taking only absolute value, I did the following coding in matlab.

```
in=zeros(input,s);
for i=1:s
    W=abs(fft(y(:,i)));
    in(:,i)=sortW(W,1,input);
end
```

Now we can not train all 40000 value as it will take so many space. So I decided to take highest 50 frequency as an input. It is now a challenging part how we can sort regarding highest frequency. We can use 'sort' build up command by matlab. But by doing that, it will be sorted by amplitude. It will not be a smart technique because speaker can give speech loudly or slowly. So if we train speech with loud voice, then speech can not be perfectly detected when speaker speaks slowly. So using amplitude while sorting is not an efficient approach in this phase. If we can use frequency, then it will not be a matter

whether speaker speaks loudly or slowly. For that reason, I used frequency as training input. Now for sorting frequency for higher to lower, I created an own 'sort' function in matlab which is given below.

```
function [f]=sortW(W,l,in)

    w=[1:1]';
    temp=0;
    for i=1:l
        for j=1:l-1
            if W(j+1) > W(j)
                temp=W(j+1);
                W(j+1)=W(j);
                W(j) =temp;
                temp=w(j+1);
                w(j+1)=w(j);
                w(j) =temp;
            end
        end
    end
    f=zeros(in,1);
    for i=1:2:in*2-1
        f((i/2)+0.5)=w(i);
    end
end
```

By this function, I sorted frequency from highest to lowest. But there are two same values we have got from the absolute value graph. So we need to take only one value from the same two. That's why I take only top 50 odd value from the sorted order. I used 'for loop' for taking only first 50 odd value so that one value is not repeated twice. Now our value for neural network training is ready. We have 8 sample voice signal which is stored in a (50×8) matrix. Screenshot of some of the value of this matrix is given below:

| in | | | | | | | | | |
|-------------|------|------|------|------|------|------|------|------|---|
| 50x8 double | | | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 1 | 5001 | 2681 | 2088 | 4983 | 5360 | 2029 | 5256 | 2674 | |
| 2 | 5002 | 2682 | 2089 | 4985 | 5361 | 2030 | 5255 | 2673 | |
| 3 | 5003 | 2680 | 2495 | 4979 | 5359 | 2028 | 5257 | 2675 | |
| 4 | 5000 | 2317 | 2496 | 4981 | 5362 | 2031 | 5254 | 2672 | |
| 5 | 4999 | 2316 | 2090 | 4980 | 5358 | 2027 | 5258 | 2676 | |
| 6 | 5004 | 2683 | 2087 | 4982 | 5363 | 2032 | 5253 | 2671 | |
| 7 | 4998 | 2679 | 2497 | 4978 | 5357 | 2026 | 5259 | 2456 | |
| 8 | 5005 | 2315 | 2494 | 4988 | 5356 | 2033 | 5252 | 2457 | |
| 9 | 4172 | 2318 | 2091 | 4986 | 5364 | 2054 | 5260 | 2455 | |
| 10 | 4997 | 2572 | 2086 | 4977 | 5355 | 2053 | 5251 | 2018 | |
| 11 | 5006 | 2573 | 2498 | 4975 | 5365 | 2025 | 5261 | 2677 | |
| 12 | 4169 | 2684 | 2493 | 4984 | 5354 | 2055 | 5250 | 2017 | |
| 13 | 4170 | 2571 | 2085 | 4976 | 5353 | 2052 | 5262 | 2670 | |
| 14 | 4171 | 2574 | 1927 | 4987 | 5366 | 1064 | 5249 | 2545 | |
| 15 | 4173 | 2678 | 2499 | 2469 | 5352 | 1065 | 5263 | 2458 | |
| 16 | 4168 | 2314 | 1926 | 4989 | 5351 | 1055 | 5248 | 2544 | |
| 17 | 4174 | 2319 | 1928 | 2467 | 5367 | 1063 | 5264 | 2709 | |
| 18 | 4167 | 2570 | 2092 | 4991 | 5350 | 1054 | 5247 | 2546 | |
| 19 | 4175 | 2575 | 2470 | 2468 | 5349 | 2056 | 5265 | 2710 | |
| 20 | 5007 | 2685 | 2492 | 2465 | 5348 | 1056 | 5246 | 2019 | |
| 21 | 4176 | 2521 | 1925 | 2471 | 5347 | 2080 | 5266 | 2708 | |
| 22 | 4166 | 2569 | 2084 | 2466 | 5368 | 2079 | 5245 | 2454 | |
| 23 | 4996 | 2520 | 1924 | 4973 | 5346 | 2005 | 6161 | 2016 | |

That means, 8 column represent 8 input and 50 rows of each column represent top 50 frequency value of each recorded sample. Now we can move forward for neural network training.

c) Neural network training:

I trained neural network by supervised learning. As this is supervised training, I needed to first define output as regarding input. I gave speech as following sequence.

```

in=zeros(input,s);
for i=1:s
    W=abs(fft(y(:,i)));
    in(:,i)=sortW(W,1,input);
end
o = [1 0 0 1 1 0 1 0];

```

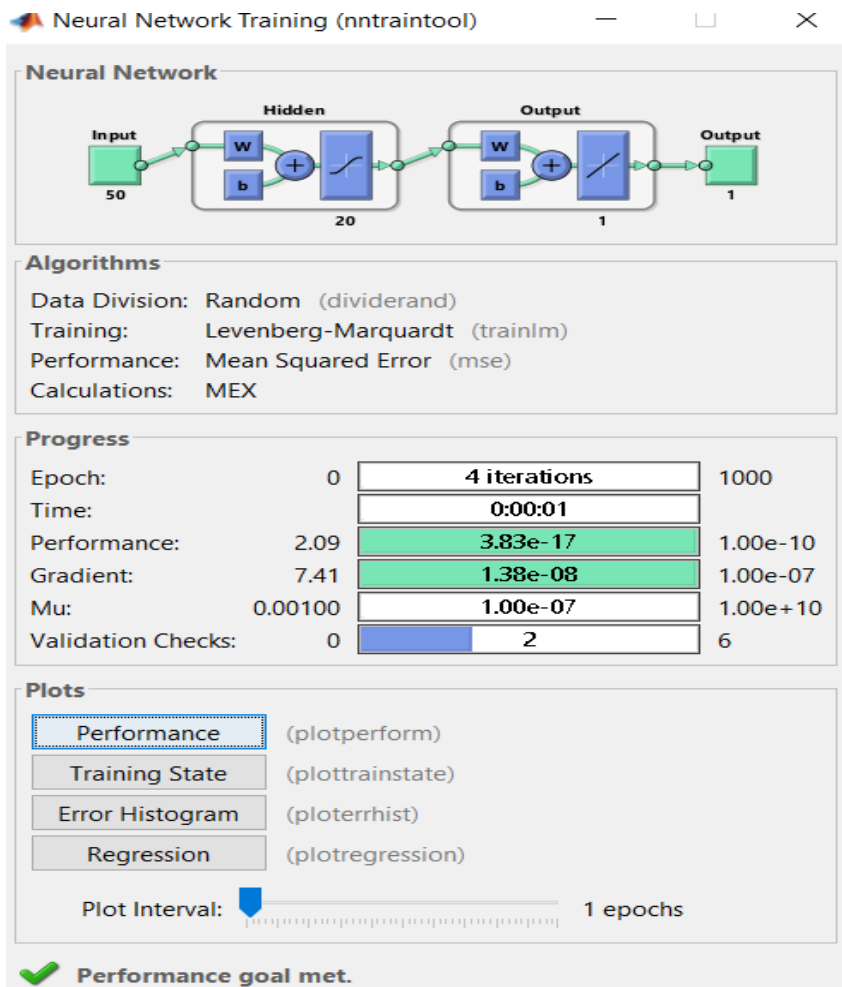
Now, input and output both are ready. Now we need neural network toolbox to train the sample input. I selected maximum number epoch and also set the value of error so that training can stop when meet any of the two criterion. I also selected 20 hidden neuron and finally stored trained network in 'net'. The command for training the network in Matlab is given below:

```

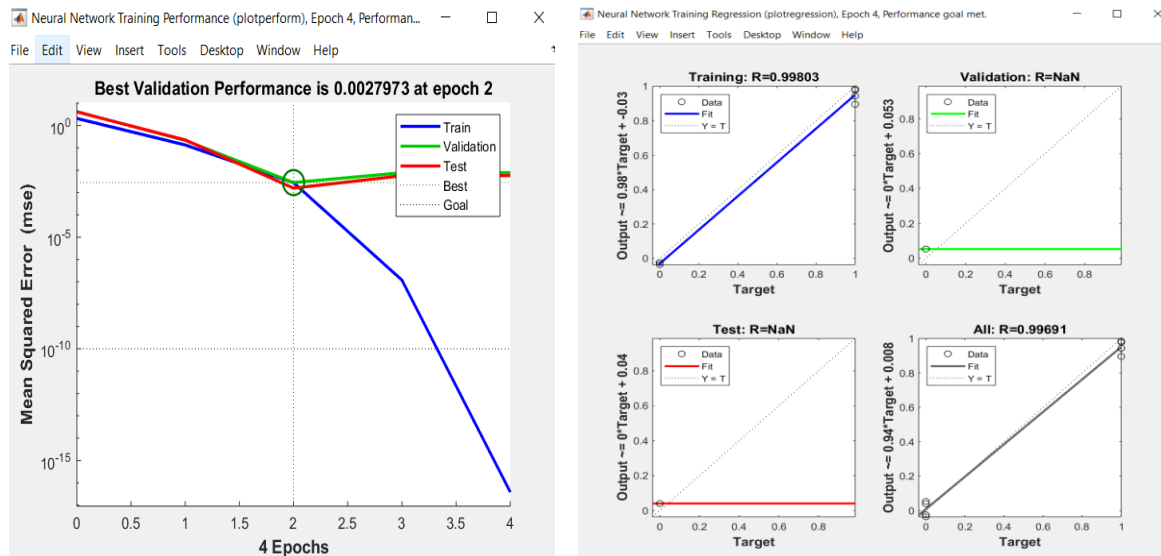
22
23 - net = feedforwardnet(20);
24 - net.trainParam.epochs=1000;
25 - net.trainParam.goal=1e-10;
26 - net = train(net,in,o);
27 - pause()
28

```

I have used default training method which is Levenberg-Marquardt method. Now we need to check the performance graph and regression graph. I am including some some screenshot and figures after training.



The following figures are performance graph and regression graph of the neural network:



d) Validation:

We need to now verify that our trained network works fine or not. For validation purpose, I again took a voice sample of 5 second. Doing all same procedure of feature extraction, we just need to give it as input in our trained network to see the output. This time computer don't know the output. It will give output by comparing its previous trained network. This is called actually machine learning. If computer can recognize perfectly my speech, then we can say that training was good. Now it's time to check the output. As I trained network just for '0' and '1', I need to give only this two speech as input. The coding for validation is given below:

```
test.m x sortW.m x validation.m x Untitled x +
1 % Validation
2 - clc;
3 - disp('Start speaking.')
4 - recordblocking(recObj, samp);
5 - disp('End of Recording. ');
6 - play(recObj);
7 - q = getaudiodata(recObj);
8 - W=abs(fft(q));
9 - pause();
10 - f=sortW(W,40000,50);
11 - x=net(f);
12
13 - if x<0.5
14 -     x=0;
15 - else
16 -     x=1;
17 - end
18
19 - fprintf('I hope you said %d\n',x);
20
```

For getting an integer value, I used 'if' loop in coding. So for the validation purpose I said '0' and it successfully detect it. Here is the output screenshot given below:

```
Command Window
Start speaking.
End of Recording.
I hope you said 0
fx >> |
```

I tried with different input and I found the system detecting speech correctly 95 percent of the time. Sometimes it failed to detect correctly if it can not record your voice perfectly.

CONCLUSION

We have trained our network perfectly with sample input and successfully got the desired output. We have learned to take voice as input in Matlab and trained them in neural network by converting signal from time domain to frequency domain. The neural network build up in Matlab environment. A success rate of 95% achieved while just give '0' and '1' as input. Finally, we can say that neural network system successfully detect the speech which I wanted.

LIMITATION AND FUTURE WORK

In this project, we just recognized two number which was '0' and '1' and we just trained 8 voice sample. But we can use more voice sample for getting better result. The more the sample we can use, the more better output will come. We just only worked with one speaker voice sample but for future work we can work with more speaker voice sample and make it universal system. We detect speech in our system but we can also work for speaker detection. We can build a system which detect speaker by detecting their voice. And if we can detect speaker then it can be used in various security purpose such as a phone can be unlocked by detecting owner's voice. We can use it for national security instead of finger print or image recognition.

REFERENCE

1. Ganesh K Venayagamoorthi, Viresh Moonasar and Kumbes Sandrasegaran "Voice Recognition using Artificial Neural Network"
2. P Krauss, L Shure, J N Little, "MATLAB Signal Processing Toolbox User's Guide", The MathworksInc.1996
3. Chee Peng Lim, Siew Chan Woo, Aun Sim Loh and Rohaizan Osman "Speech Recognition Using Artificial Neural Networks"
4. Wouter Gevaert, Valeri Mladenov and Georgi Tsenov "Neural networks used for speech recognition" Journal of Automatic Control, January 2010

