UNIVERSIDADE FEDERAL DO PARANÁ

CURSO DE ENGENHARIA ELÉTRICA

RECONHECIMENTO DE LOCUTOR ATRAVÉS DE ANÁLISE ESPECTRAL E REDE NEURAL ARTIFICIAL

Aluno: André Heidemann Iarozinski

Professor: Eduardo Parente Ribeiro, Dr.

CURITIBA

2015

1. Introdução

O processamento digital de sinais dedicado a reconhecimento de voz está em constante evolução e a cada dia se buscam formas mais eficientes e diferenciar fonemas em um sinal de áudio. Os sons das palavras são facilmente reconhecidos pelas pessoas, mas não é algo tão simples para uma máquina compreender pequenas diferenças entre os fonemas ocasionadas pelo sotaque, timbre, a velocidade em que uma palavra é pronunciada e até mesmo o estado emocional do locutor.

Atualmente a maioria dos algoritmos existentes se baseia em dados estatísticos para reconhecimento de voz o que resulta em erros em reconhecer palavras menos comuns.

2. Objetivo

O objetivo deste trabalho é desenvolver um sistema capaz de diferenciar diferentes locutores emitindo um mesmo som, como por exemplo a vogal “a”. Uma analogia seria o reconhecimento de diferentes instrumentos musicais emitindo a mesma nota musical, como cada instrumento possui seu próprio timbre, o espectro de frequência de um instrumento emitindo uma nota musical será sempre diferente de outro instrumento emitindo a mesma nota. Espera-se obter uma taxa de acerto igual ou superior a 95%.

3. Metodologia

Tudo será desenvolvido no software MATLAB por meio de algoritmos e em um software para gravação de áudio. Serão obtidas amostras de cinco locutores diferentes, digitalizadas em formato .WAV com taxa de amostragem de 44.1KHz.

Os sons de cada locutor serão gravados em um software para gravação de áudio e neste mesmo software, as alterações necessárias serão feitas como: filtros para elevar a relação sinal-ruído e normalização das amplitudes. No MATLAB as amostras serão convertidas para o domínio da frequência através da transformada rápida de Fourier (FFT).

Para determinar as características distintivas dos locutores serão calculadas as energias das faixas de frequências mais evidentes das FFT´s (faixas de frequência posteriormente determinadas) ou serão obtidos picos das frequências mais relevantes em cada amostra. Em seguida os dados serão utilizados para criação de uma rede neural que será treinada com o objetivo de conseguir uma taxa de acerto igual ou superior ao objetivo do trabalho.

4.Desenvolvimento

Primeiramente foram gravados áudios da minha voz (André) e de alguns colegas. O objetivo era criar um algoritmo que conseguisse diferenciar a minha voz dos demais locutores através de uma rede neural.

Como referência, no início do algoritmo o MATLAB gerava um tom senoidal com frequência 196Hz antes do início de cada gravação de voz.

```
amp=5;
fs=11025;  % frequencia de amostragem
duration=0.3;
freq=196;
values=0:1/fs:duration;
a=amp*sin(2*pi* freq*values);
sound(a,11025)
```

Em seguida foram adquiridas várias amostras e criada uma rede neural da seguinte forma:

```
%%%%%%%% criação da matriz data
d = [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20 a21
a22 a23 a24 a25 a26 a27 a28 a29 a30 b1 b2 b3 b4 b5 b6 b7 b8 b9 b10];  %% b´s
bruno,kaio,jake

%%%%%%% criação da matriz resposta
r = [zeros(1,30) ones(1,10)];
```

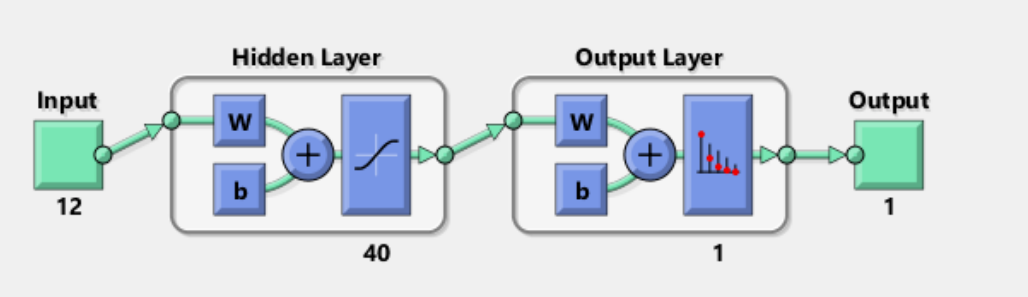
A matriz de entrada formada por 40 vetores coluna dos quais a1 até a30 são da minha voz e o restante de outras pessoas. Todas as amostras foram gravadas da mesma forma, com taxa de amostragem de 11025Hz e com duração de 1 segundo.

A medida que cada amostra foi gravada, foi feita sua FFT e através da função “findpeaks” do matlab se encontrou os picos de frequência em cada amostra criando um vetor de tamanho 12 para cada amostra.

```
z = getaudiodata(recc);
z=abs(fft(z));
z=z(1:3000);
[pks,locs] =
findpeaks(z,'MinPeakProminence',20,'Threshold',4,'MinPeakDistance',30);
size(locs);
if size(locs)<12
    locs=[locs; zeros(12-length(locs),1)];
else
locs=locs(1:12);
end
cmd = ['c' num2str(i) '= locs;'];
eval(cmd);
```

Caso a função encontrasse um numero inferior de 12 picos no sinal, o scrip preenche com zeros o vetor até atingir tamanho 12. Na terceira linha do código é feito um descarte de certa parte do espectro pois as frequências de interesse são somente as próximas de 200Hz.

Com a matriz data e a matriz resposta montou-se a rede neural com 40 neurônios. Dos dados de entrada, 80% das amostras foram utilizadas para treinamento, 10% para validação e 10% para teste.



Anteriormente foram feitas várias redes, com parâmetros diferentes a fim de se obter a rede com melhor matriz de confusão e com menor erro. Após o treinamento da rede foi gerado com código do matlab para a rede e nomeado como `rede_x2` para ser utilizado como uma função no algoritmo principal.

Com a rede pronta, foram feitos vários testes para medir o resultado do algoritmo.

5.Conclusões

Foram encontrados vários problemas durante o período de realização do trabalho, destacando os principais sendo a qualidade de captação do microfone do notebook, que possui uma baixa relação sinal ruído (verificado graficamente no matlab). Outro problema principal e que causou maior erro no desempenho foi que o vetor de entrada não tinha parâmetros fixos:

Quando uma pessoa gravava a primeira vez, o matlab criava o seguinte vetor (por exemplo):

B1 = [12 56 91 122 197 220 450 0 0 0 0 0];

E na segunda gravação criava :

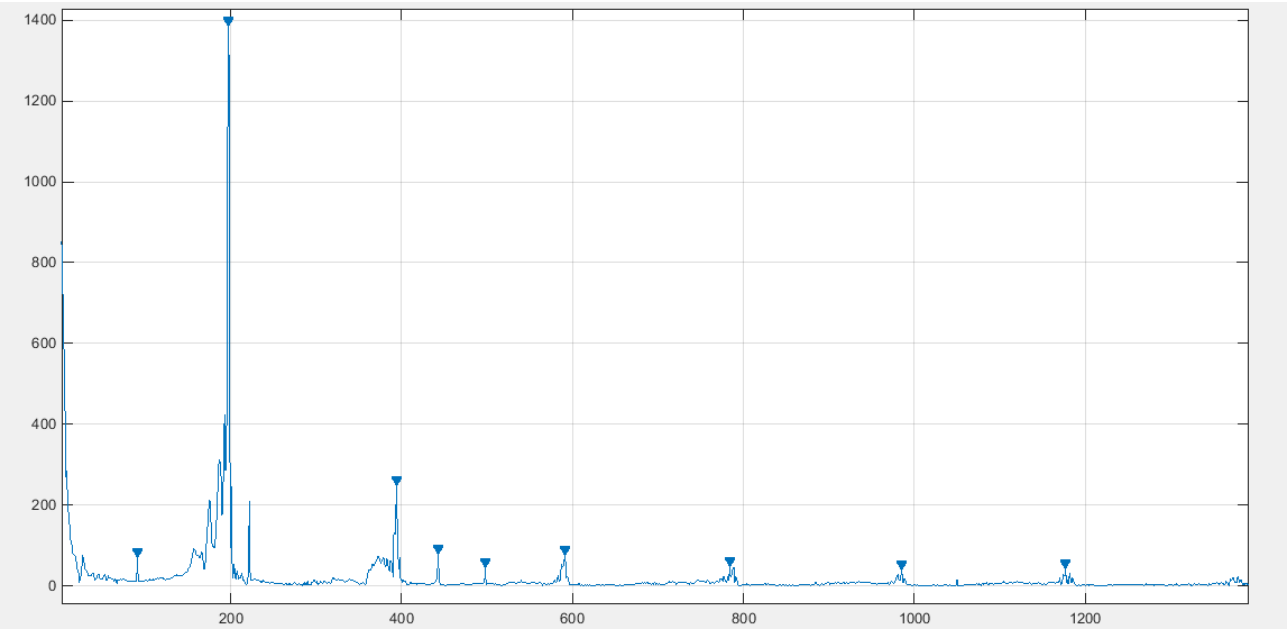
B2 = [12 54 130 196 213 447 0 0 0 0 0 0];

O pico de frequência de 91Hz, característico do timbre do locutor, não foi captado na segunda gravação e faz mudar a posição das demais frequências. Isso foi a maior causa de erro na interpretação da rede pois a cada gravação, mesmo obtendo as mesmas frequências, elas estavam em posições diferentes no vetor.

Deveria ter sido feito alguma modificação no script para eliminar este problema e ainda seria interessante ter adicionado as amplitudes de cada frequência como outro parâmetro de comparação.

Outro ponto importante foi o ajuste dos parâmetros da função *findpeaks* que tiveram de ser ajustados por tentativa e erro, verificando-se em cada gráfico se os picos obtidos eram picos relevantes para uma análise comparativa na rede.

- Picos obtidos com a função findpeaks com parâmetros: MinPeakProminence',20,‘ Threshold',4,‘ MinPeakDistance',30’



A seguir o código completo no MATLAB:

```
%%%%%%%%%% Identificador de Locutor André Heidemann Iarozinski
%%%%%%%%%%

prompt = 'bem vindo, aperte "enter" para fazer login ';
input(prompt);

%%%%% gerando som
amp=5;
fs=11025;  % frequencia de amostragem
duration=0.3;
freq=196;

values=0:1/fs:duration;
a=amp*sin(2*pi* freq*values);
sound(a,11025)
```

No início é apresentado para o locutor algo como um simulador de “login” para o locutor. Gera-se o tom de referência e se realizam 3 gravações consecutivas.
Após é feita uma média das três e o vetor é armazenado na variável “c” para ser utilizado como entrada na rede neural.

```
%%%%%%%%%% armazenando amostras para verificação

for i=1:3

prompt = 'para gravar sua voz - aperte "enter"';
input(prompt);

recc = audiorecorder(11025, 16, 1);
disp('inicio da gravacao')
recordblocking(recc, 1);
disp('fim da gravacao');

z = getaudiodata(recc);
z=abs(fft(z));
z=z(1:3000);

[pks,locs] =
findpeaks(z,'MinPeakProminence',20,'Threshold',4,'MinPeakDistance',30);
size(locs);

if size(locs)<12
    locs=[locs; zeros(12-length(locs),1)];
else
    locs=locs(1:12);
end

cmd = ['c' num2str(i) '= locs;'];
eval(cmd);

end

cf=(c1+c2+c3)/3;

c=ceil(cf);                  %%%ENTRADA NA REDE

%%%%%%%%% criação da matriz data
d = [a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15 a16 a17 a18 a19 a20 a21
a22 a23 a24 a25 a26 a27 a28 a29 a30 b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 u1 u2 u3 u4
u5 u6 u7 u8 u9 u10];  %% b´s bruno,kaio,jake

%%%%%%% criação da matriz resposta
r = [zeros(1,30) ones(1,20)];

d=d';
r=r';
%%%%criacao da rede neural "redezz"(10 neuronios) com as amostras

%%%%%%%%% verificação final
f= rede_x2(c);

if f<=0.4

    disp('login efetuado com sucesso! ');
else

    disp('acesso negado!');
end
```

6.Referências

1 - Simon Haykin, Neural Networks - A comprehensive Foundation.

2 - Oppenhein, A. V., Schafer, R. W. Discrete Time Signal Processing, 2.ed.

3 – Introdução ao "Neural Network Toolbox" no MatLab

http://www.mathworks.com/videos/getting-started-with-neural-network-toolbox-68794.html

Acessado em 27/10/2015

4- L.R.Rabiner, R.W.Schafer, Digital Processing of Speech Signals,Prentice Hall, 1978.