UNIVERSIDADE FEDERAL DO CEARA
CAMPUS DE SOBRAL
ENGENHARIA DA COMPUTAÇÃO E ENGENHARIA ELÉTRICA

Processamento Digital de Sinais (SBL0085)

Prof.: C. Alexandre Rolim Fernandes

Trabalho 1 – Transformada de Fourier para Detecção de Comandos de Voz

- Trabalho Individual

- Apenas simulações, sem trabalho escrito

- Os códigos devem estar bem organizados e comentados, para que seja possível entendê-los e corrigi-los. Códigos que estejam desorganizados ou sem os devidos comentários explicativos terão penalização na nota.

- Fazer todas as questões em **um só arquivo.**

- O seu código deve **gerar automaticamente todos os gráficos e resultados** solicitados.

- Enviar no SIGAA o **código cujo nome do arquivo deve ser igual ao seu nome.**

- Não enviar código em PDF.

- Prazo e forma de entrega: dia 22/09/23 às 23h59, no SIGAA.

Detecção de Comandos de Voz usando Transformada de Fourier e Transformada de Fourier de tempo curto:

Foram fornecidos diferentes arquivos de áudio, no format .mat, correspondentes às palavras “sim” e “não”. O objetivo desta prática é utilizar Transformada de Fourier (TF) e Transformada de Fourier de tempo curto (short-time Fourier transform – STFT) para detectar, de forma automática, se um áudio corresponde à palavra “sim” ou à palavra “não”. Em outras palavras, realizaremos o reconhecimento automático dos comandos de voz “sim” e “não” usando o espectro de Fourier. O propósito final é determinar se a TF e a STFT são eficientes ferramentas para caracterizar sinais de áudios para esta aplicação específica.

Os sinais de áudio estão divididos em dois grupos. O arquivo InputDataTrain.m

contém 10 sinais de áudio, sendo os 5 primeiros correspondentes à palavra “não” e os 5 últimos à palavra “sim”. Estes sinais serão usados para o chamado treinamento da técnica de classificação, ou seja, serão extraídos alguns parâmetros destes sinais que caraterizam os áudios “sim” e “não”.

A segunda etapa corresponde à classificação (ou detecção) dos áudios. Esta fase usará o arquivo InputDataTest.m, que contém 7 arquivos de áudio, sendo os 3 primeiros correspondentes à palavra “não” e os 4 últimos à palavra “sim”. Nesta etapa, você irá comparar as características extraídas de cada um dos 7 áudios com aquelas extraídas na fase de treinamento, para se chegar a uma decisão final.

1) Carregue os 10 sinais de áudio de InputDataTrain.m e gere os gráficos destes sinais, em 2 figuras separadas. Uma figura deve conter os áudios “sim” e a outra deve conter os áudios “não”

2) Divida cada um destes 10 sinais em 80 blocos de N/80 amostras, em que N é o número de amostras de cada um dos sinais de áudio. Calcule a energia de cada um destes blocos e gere os gráficos com as energias de destes 80 blocos no eixo y e o índice do bloco no eixo x, em 2 figuras separadas. Uma figura deve conter os áudios “sim” e a outra deve conter os áudios “não”. Caso N/80 não seja inteiro, ignore as casas decimais de N/P.

3-) Calcule o módulo ao quadrado da Transformada de Fourier (TF) dos 10 sinais de áudio de InputDataTrain.m e gere os gráficos destas TFs, em 2 figuras separadas. Uma figura deve conter os áudios “sim” e a outra deve conter os áudios “não”. O eixo x deste gráficos deve corresponder às frequências entre -pi e pi.

4) Note que o módulo ao quadrado da TF é simétrico em relação à frequência zero. Além disso, pode-se perceber que o espectro dos sinais é concentrado nas baixas frequências. Desta forma, você deve eliminar as redundâncias das TFs calculadas no Item 3 descartando as frequências negativas e as frequências acima de pi/2. Em outras palavras, recalcule estas TFs considerando apenas as frequências entre 0 e pi/2. Gere os gráficos destas TFs, em 2 figuras separadas. Uma figura deve conter

os áudios “sim” e a outra deve conter os áudios “não”. O eixo x deste gráficos deve corresponder às frequências entre 0 e pi/2.

5-) Divida cada uma das 10 TFs do Item 4 em 80 blocos de N/320 amostras (N/4 é o número de amostras de cada uma das TFs). Calcule a energia de cada um destes blocos e gere os gráficos com as energias de destes 80 blocos no eixo y e o índice do bloco no eixo x, em 2 figuras separadas. Uma figura deve conter os áudios “sim” e a outra deve conter os áudios “não”. Caso N/320 não seja inteiro, ignore as casas decimais de N/320.

6-) Agora, divida cada um dos sinais de áudio (no domínio do tempo) em 10 blocos de N/10 amostras e calcule o módulo ao quadrado da TF de cada um destes blocos. Os 10 espectros resultantes de cada áudio correspondem à Transformada de Fourier de tempo curto (short-time Fourier transform – STFT). Tal como no Item 4, recalcule estas STFTs considerando apenas as frequências entre 0 e pi/2. Gere os gráficos destas STFTs, em 2 figuras separadas, mas apenas para um sinal do tipo “sim” e um sinal do tipo “não”. O eixo x deste gráficos deve corresponder às frequências entre 0 e pi/2. Note que, no Itens 3 e 4, você deve calcular a TF do sinal inteiro (com todas as amostras), enquanto nos Itens 5 e 6 você deve calcular a TF dos sub-sinais gerados (cada um com N/10 amostras).

7) Divida estas STFTs em 8 blocos de N/320 amostras (N/40 é o número de amostras de uma das STFTs). Calcule a energia de cada um destes blocos. Pra cada sinal de áudio, você deve obter 8 energias para cada uma das 10 STFTs, totalizando 80 energias para cada sinal de áudio. Não precisa gerar nenhum gráfico neste Item 7.

Note que, nos itens 2, 5 e 7, foram calculadas as energias de 80 blocos em 3 diferentes domínios. No Item 2, foi usado o domínio do tempo, no Item 5 foi usada a TF e no Item 7 a STFT. Estas 80 energias serão usadas para caracterizar o sinal de áudio (em cada um dos 3 domínios). O objetivo desta prática é saber, para o caso específico do reconhecimento dos comandos de voz “sim” e “não”, qual destes 3 domínios (tempo, TF e STFT) é o mais eficiente.

8-) Para cada um dos 3 domínios, organize as 80 energias calculadas em um vetor e tire a média destes vetores para as 5 amostas da classe “sim” e para as 5 amostras da classe “não”. No final, você deverá obter 2 vetores de tamanho 80x1 (representandos o “sim” e o “não”) para cada um dos 3 domínios. Estes vetores são chamados de centroides. Não precisa gerar nenhum gráfico neste Item 8.

9-) Repita os procedimentos do Itens 1 a 7 para calcular os vetores de tamanho 80x1 com as energias, só que desta vez usando os 7 sinais de teste do arquivo InputDataTest.m e sem gerar nenhum gráfico. Ou seja, apenas gere, para um dos 3 domínios, os 7 vetores de tamanho 80x1 com energias dos blocos. Não precisa gerar nenhum gráfico neste Item 9.

10-) Para realizar a detecção dos comandos de voz (classificação), usaremos o chamado Algoritmo do Centroide, que compara os vetores de energia dos áudios de teste gerados no Item 9 com os centroides que caracterizam cada classe, gerados no Item 8. Desta forma, calcule a distância Euclidiana entre cada um dos 7 vetores gerados no Item 9 e os 2 centroides gerados no Item 8, totalizando 14 distâncias Euclidianas.

Para cada um dos 7 vetores de teste, realize uma escolha entre as classes “sim” e “não” baseado no parágrafo anterior e calcule a quantidade de acertos usando cada um dos 3 domínios (tempo, TF e STFT). Lembre-se que o arquivo InputDataTest.m contém 7 arquivos de áudio, sendo os 3 primeiros correspondentes à palavra “não” e os 4 últimos à palavra “sim”.

Qual domínio obteve um maior número de acertos? Qual obteve o menor número de acertos? Tente escrever uma justificativa (em forma de comentário no código) para a ordem de desempenho dos domínios.