Disciplina:
Prof. Bernardo Barancelli Schwedersky (bernardo.schwedersky@inf.ufpel.edu.br)

Trabalho Prático

Parte 1 - Filtragem de sinais com a convolução em tempo discreto

A operação de convolução em tempo discreto é um conceito fundamental no processamento de sinais e na análise de sistemas. Ela desempenha um papel importante em diversas áreas, como processamento de imagens, processamento de áudio, telecomunicações e outras disciplinas relacionadas. A convolução é uma técnica matemática que permite combinar dois sinais para criar um terceiro sinal, que pode representar informações resultantes da interação entre os dois sinais originais.

Na sua essência, a convolução em tempo discreto envolve deslizar uma das sequências sobre a outra, multiplicando seus valores correspondentes e somando os produtos. Isso resulta em um novo sinal, chamado de saída da convolução. No caso unidimensional (1D), a operação considera duas sequências discretas: uma é a entrada $x[n]$ e a outra é a resposta ao impulso $h[n]$ do sistema. A operação de convolução em tempo discreto pode ser formalmente definida como:

$$c[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} x[m]h[n-m] \tag{1}$$

Devido as sequências serem discretas, a implementação computacional dessa operação é simples. Cada elemento do sinal resultante pode ser obtido iterativamente, obtendo a soma de todos os elementos da multiplicação entre as sequências, sendo que uma das sequências é revertida no tempo e deslocada a cada iteração.

Essa operação aparece em diversas aplicações. Uma delas é a filtragem de sinais em tempo discreto. Por exemplo, um filtro passa-baixas pode ser usado para eliminar ruídos de frequência mais alta que o sinal de interesse. Em tempo discreto, o filtro passa-baixas mais simples pode ser representado pela seguinte função de transferência:

$$H(z) = \frac{(1-a)}{(z-a)}, \tag{2}$$

onde o parâmetro a define a frequência de corte do filtro, sendo que, quanto mais próximo de 1, menor será a frequência de corte do filtro.

Dessa forma, implemente na sua linguagem de preferência a operação de convolução entre um sinal corrompido pelo ruído e o filtro em tempo discreto. O sinal corrompido pelo ruído deve ser representado por $x[n] = s[n] + r[n]$, onde o sinal é $s[n] = sen(0,05n) + 0.5 * sen(0,15n)$ e o ruído é representado por um sinal aleatório, com distribuição uniforme e amplitude máxima $0,2$. A resposta ao impulso $h[n]$ pode ser obtida por meio da transformada Z inversa, sendo $h[n] = (1-a)(a)^n \cdot u[n]$. Realize a simulação deste processo de filtragem considerando $0 \leq n < 200$, discutindo o impacto da escolha do valor de a.

Apresente a implementação desse processo de filtragem, os gráficos para o sinal original $s[n]$, o sinal corrompido pelo ruído, $x[n]$, e o sinal filtrado, o qual é o resultado da convolução, para diferentes valores de a.

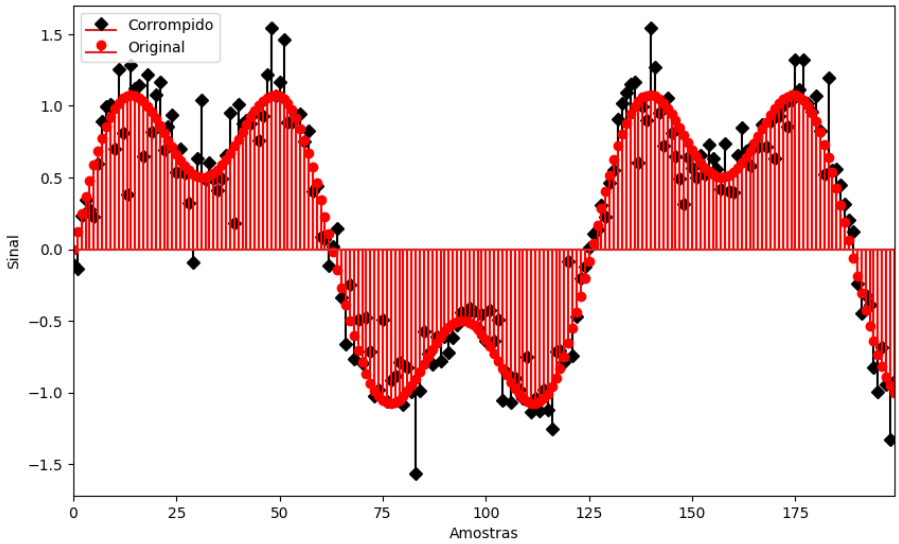

Figura 1: Exemplo do sinal e do sinal corrompido.

Parte 2 - Convolução 2D em Tempo Discreto

A operação de convolução em tempo discreto pode ser realizada para dados bidimensionais (2D). Esse tipo de operação é aplicado na construção filtros convolucionais para o processamento de imagens e visão computacional. A convolução 2D permite realizar transformações nos dados de uma imagem, destacando características relevantes e realçando informações específicas. Essa abordagem é amplamente utilizada em tarefas como detecção de bordas, suavização, realce de detalhes e muito mais.

A convolução 2D segue o mesmo princípio básico da convolução em tempo discreto com sinais unidimensionais, mas agora aplicado a sinais bidimensionais. Nesse contexto, a imagem original atua como a sequência de entrada, a qual é aplicada à um filtro bidimensional (também conhecido como kernel). A imagem é deslizada sobre a resposta ao impulso do filtro, e, a cada posição do deslizamento, os valores do filtro são multiplicados pelos valores correspondentes da região da imagem coberta pelo filtro. A soma desses produtos resulta no valor da saída da convolução para aquela posição.

A operação de convolução 2D é definida como:

$$c[x,x] = I[x,y] * K[x,y] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} I[i,j]K[x-i, y-j], \tag{3}$$

onde $I[x,y]$ representa a imagem; $K[x,y]$ representa a resposta ao impulso do filtro (kernel); x e y representam as coordenadas de cada pixel da imagem; e $c[x,y]$ representa o resultado da operação de convolução 2D.

Os filtros convolucionais desempenham um papel crítico em várias aplicações de processamento de imagens. Por exemplo, Um filtro de suavização (ou de média) pode ajudar a reduzir o ruído da imagem, homogeneizando pequenas variações e criando uma aparência mais uniforme. Para alcançar esse efeito, podemos utilizar o kernel:

$$K_{suavizao} = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{4}$$

Outro exemplo é uma implementação rudimentar de um filtro de detecção de bordas, cuja tarefa é realçar as mudanças bruscas de intensidade na imagem, destacando as bordas entre diferentes regiões. Esse filtro pode ser alcançado aplicando um filtro de suavização e após um filtro considerarmos o kernel:

$$K_{bordas} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}. \tag{5}$$

Para facilitar a visualização da detecção, podemos usar um limitador após a convolução, fazendo com que os pixels que, após a filtragem, tiverem valores negativos sejam mapeados para 0, e os com intensidade positiva sejam mapeados para 255.

A tarefa consiste em implementar o processo de convolução 2D para filtragem de duas imagens. As duas imagens podem ser escolhidas livremente, sendo que o aconselhado é trabalhar com imagens em escala de cinza. Dessa forma, a entrada do filtro, $I[x, y]$ será uma matriz bidimensional. Cada imagem deve ser filtrada com o filtro de detecção de bordas, e o filtro de suavização. O aluno deve gerar um relatório discutindo a implementação do código, e apresentado as imagens originais e filtradas.

A seguir, é apresentado um exemplo de imagem filtrada com ambos filtros.

Figura 2: Exemplos do processo de filtragem