Segundo Trabalho Métodos Computacionais em Física

11/06/2024

Data de entrega: 18/06/2024

Instruções gerais

- Esse trabalho é individual, o que significa que cada estudante deverá entregar seu próprio relatório, mas encorajo fortemente que ele seja discutido com seus colegas e que seja feito de forma colaborativa.
- O trabalho deve ser entregue no formato de um breve relatório, em formato pdf, apresentando as **considerações que tiveram que ser feitas** para resolver de cada um dos problemas elencados abaixo, os códigos computacionais utilizados e uma **análise crítica dos resultados** encontrados e das **dificuldades** enfrentadas na execução do trabalho.

Problemas:

1. **Transformadas de Fourier de formas de onda de exemplo:**

Estão sendo fornecidos dois arquivos piano.txt e trompete.txt, que contêm dados que representam a forma de onda de uma única nota, tocada, respectivamente, em um piano e em um trompete.

(a) Escreva um programa que carregue uma forma de onda de um desses arquivos, faça um gráfico dela, depois calcule sua transformada discreta de Fourier e represente graficamente as magnitudes dos primeiros 10.000 coeficientes. Observe que você terá que usar uma transformação rápida de Fourier (FFT) para o cálculo porque há muitas amostras nos arquivos para fazer as transformações de maneira lenta em qualquer período de tempo razoável.

Aplique seu programa às formas de onda do piano e do trompete e discuta brevemente o que se pode concluir sobre o som do piano e do trompete a partir dos gráficos dos coeficientes de Fourier.

(b) Ambas as formas de onda foram gravadas à taxa padrão da indústria de 44.100 amostras por segundo e ambos os instrumentos tocavam a mesma nota musical quando as gravações foram feitas. A partir dos resultados da transformada de Fourier, calcule a nota que eles estavam tocando. (Dica: a nota musical dó central tem uma frequência de 261 Hz.)

Sugiro apresentar seu relatório usando um Jupyter Notebook contendo seus gráficos dos coeficientes de Fourier para cada uma das duas formas de onda, suas conclusões sobre o que você observou e seus cálculos e resultados da parte (b).

2. **Filtragem e suavização de Fourier:**

Junto da tarefa você encontrará o arquivo dow.txt. Ele contém o valor de fechamento diário de cada dia útil, desde o final de 2006 até o final de 2010, do Dow Jones Industrial Average, que é uma medida dos preços médios no mercado de ações dos EUA.

Escreva um programa para fazer o seguinte:

(a) Leia os dados de dow.txt e coloque-os em um gráfico.

(b) Calcule os coeficientes da transformada discreta de Fourier dos dados usando a função rfft de numpy.fft, que produz uma matriz de N/2 + 1 números complexos.

(c) Agora defina todos os elementos desta matriz, exceto os primeiros 10%, como zero (ou seja, defina os últimos 90% como zero, mas mantenha os valores dos primeiros 10%).

(d) Calcule a transformada inversa de Fourier da matriz resultante, com zeros e tudo, usando a função irfft, e plote-a no mesmo gráfico dos dados originais. Pode ser necessário variar as cores das duas curvas para garantir que ambas apareçam no gráfico.

Comente o que você vê. O que acontece quando você define os coeficientes de Fourier como zero?

(e) Modifique seu programa para que ele defina todos os coeficientes, exceto os primeiros 2%, como zero e execute o cálculo novamente. Faça um gráfico que mostre todas as três curvas nos mesmos eixos – os dados brutos, a curva de 10% e a curva de 2%.

O conjunto de dados específico aqui estudado é especial num sentido: o valor do Dow no final do período era quase o mesmo que no início, pelo que a função é, grosso modo, periódica. Também está sendo fornecido outro arquivo chamado dow2.txt, que também contém dados sobre o Dow, mas para um período de tempo diferente, de 2004 a 2008. Durante este período, o valor mudou consideravelmente de um nível inicial em torno de 9.000 para um nível final em torno de 14.000.

(f) Modifique seu programa para ler os dados de dow2.txt e realizar a análise de Fourier novamente, traçando apenas duas curvas desta vez, para os dados brutos e a versão "2%". Você deverá ver que agora existe um artefato adicional na curva de 2%. No início e no final do gráfico você deverá ver grandes desvios da verdadeira função. Isso ocorre porque a função deve ser periódica – seu último valor deve ser igual ao primeiro - portanto, ele precisa desviar-se substancialmente do valor correto para fazer com que as duas extremidades da função se encontrem. Em algumas situações (incluindo esta) este comportamento é insatisfatório. Preferiríamos não introduzir artefatos deste tipo.

(g) Uma forma de resolver esse problema é utilizar uma transformada discreta de cosseno. Pesquise e descreva brevemente as diferenças entre a transformada de Fourier e a transformada de cosseno, ressaltando as vantagens e desvantagens da última em relação à primeira.

Modifique seu programa para repetir a mesma análise usando uma transformada discreta de cosseno. Transforme os dados e descarte novamente todos os coeficientes, exceto os primeiros 2%, depois inverta a transformação e represente graficamente o resultado. Você deverá ver uma melhoria significativa, com menos distorção da função nos finais do intervalo. Isso ocorre porque a transformada de cosseno não força o valor da função a ser o mesmo nas duas extremidades. Dica: A função dct faz a transformação do cosseno e idct faz a transformação inversa. É por causa dos artefatos introduzidos pela periodicidade estrita da DFT que a transformada cosseno é favorecida para muitas aplicações tecnológicas, como a compressão de áudio. Os artefatos podem degradar a qualidade do som do áudio compactado e a transformação do cosseno geralmente fornece melhores resultados.

Novamente, sugiro apresentar seu relatório usando um Jupyter Notebook. Faça um gráfico mostrando todas as três curvas que ele calcula nos mesmos eixos (dados brutos, 10% e 2%) e sua resposta à pergunta na parte (d). Entregue também seus dois gráficos das partes (f) e (g) mostrando sua análise do segundo conjunto de dados usando a DFT e a DCT respectivamente (ou você pode combinar (f) e (g) em um único gráfico, se preferir) .

3. **Compressão de imagem:**

Neste problema você escreverá seu próprio programa para compactar uma imagem fotográfica no estilo JPEG.

Primeiro baixe o arquivo house.txt. O arquivo contém dados para uma imagem de uma casa em forma de grade simples – uma matriz bidimensional de números que representa a intensidade de pixels na imagem.

(a) Escreva um programa Python que leia os dados do arquivo em um array bidimensional e então faça um gráfico de densidade do array, mostrando a imagem na tela.

Você deve usar o esquema de cores em escala de cinza para seu gráfico de densidade, para obter uma fotografia em preto e branco com aparência sensata.

(b) Agora crie outro array bidimensional do mesmo tamanho do array de imagens e inicialmente vazio. Percorra a matriz de imagens em blocos de 16×16 e execute uma transformação discreta de cosseno 2D dos dados em cada bloco, produzindo uma matriz 16 × 16 de coeficientes de Fourier (reais) e, em seguida, armazene esses coeficientes no bloco correspondente da nova matriz. Você pode realizar os DCTs usando a função dct2 do arquivo dcst.py. Para obter os blocos 16 × 16, você precisará fazer um “fatiamento” bidimensional nas matrizes. Ao terminar todos os blocos, você terá um novo array do mesmo tamanho do antigo, totalmente cheio de coeficientes de Fourier.

(c) Agora analise os coeficientes de Fourier um por um e defina como zero todos os coeficientes cujo valor absoluto seja menor que 10. Em outras palavras, todos os coeficientes no intervalo de -10 a +10 devem ser definidos como zero.

(d) Quando enviamos uma imagem pela Internet, transmitimos os coeficientes de Fourier, não a imagem em si, e só precisamos transmitir os coeficientes que são diferentes de zero. Conte quantos coeficientes foram definidos como zero em seu cálculo e use isso para calcular e imprimir um valor de quanto você comprimiu a imagem - quão menor é o conjunto de números que você teria que enviar pela Internet do que o conjunto original de o arquivo house.txt? Este valor é chamado de taxa de compressão.

(e) Quando os coeficientes de Fourier são recebidos na outra extremidade, o receptor realiza uma transformada inversa para recuperar a imagem. Embora não estejamos realmente transmitindo a nossa imagem neste caso, ainda podemos realizar esta segunda parte do cálculo para ver o que obteríamos. Adicione linhas ao seu programa para percorrer mais uma vez a matriz de coeficientes de Fourier em blocos de 16 × 16 e realizar uma DCT 2D inversa em cada um deles, armazenando os resultados novamente na matriz de dados original (ou em uma terceira, nova matriz, se você prefere). Esta é a “descompressão” da imagem. Você pode usar a função idct2 para os DCTs inversos.

(f) Faça um gráfico de densidade da imagem descompactada. Você deve descobrir que ela é essencialmente indistinguível da imagem original, embora a imagem tenha sido bastante comprimida – um número significativo de coeficientes de Fourier foi descartado ao defini-los como zero.

(g) Aumente o valor limite abaixo do qual os coeficientes são definidos como zero. Em vez de 10, tente 20, ou 50, ou 100 ou mais. Veja quão grande é a taxa de compressão que você pode atingir e ainda assim a imagem terá praticamente a mesma aparência. Novamente, entregue o seu programa final e as imagens “antes” e “depois” que ele produz, junto com sua resposta para a parte (d) e suas conclusões da parte (g).