Esercitazione 6

DFT (terza parte)

2 Maggio 2022

1. Ottimizzazione numerica: Fast Fourier Transform - FFT

   (a) Considerare il segnale x di lunghezza 128 in cui la prima metà dei campioni vale 1 e la seconda metà vale 0. Calcolare a mano il valore dei coefficienti pari della DFT (utilizzare la serie geometrica). Nota bene: i coefficienti sono indicizzati a partire da 0.

   (b) Calcolare la DFT del segnale x utilizzando la funzione creata nelle esercitazioni precedenti e fare un plot del valore assoluto dei coefficienti ottenuti utilizzando la funzione `stem` della libreria `matplotlib`.

   (c) Calcolare la fase utilizzando il comando `angle` della libreria `numpy` (utilizzare come input i coefficienti calcolati al punto 2) Fare un plot del risultato utilizzando la funzione `stem`. Confrontare il risultato con quello ottenuto utilizzando come input della funzione i coefficienti della DFT calcolati con la funzione `fft` di `scipy`. A cosa è dovuta la differenza?

2. FFT: Truncating and Zero Padding

   (a) Considerare il segnale:

   $$y = \sin\left(\frac{80\pi t}{1000}\right) + \sin\left(\frac{100\pi t}{1000}\right),$$

   dove t è un vettore equispaziato di 4000 punti compresi tra 0 e 3999. Calcolare la DFT di y e fare un plot del suo valore assoluto.

   (b) Il comando `fft` della libreria `scipy` accetta anche un secondo parametro opzionale M che indica quanti campioni di y vengono utilizzati per calcolare la FFT. Se M è minore della lunghezza del segnale, la DFT viene calcolata utilizzando solo i primi M campioni di y. Se M è maggiore della lunghezza del segnale, al segnale vengono aggiunti degli zeri al fondo per ottenere la lughezza desiderata. Calcolare la DFT di y utilizzando solo i primi 500 campioni e fare un plot del suo valore assoluto. Ripetere la stessa operazione utilizzando solo i primi 50 campioni di y. Qual è la differenza?

(c) Considerare il vettore di coefficienti ottenuto utilizzando solo i primi 50 campioni di y. Ricostruire il segnale utilizzando la DFT inversa. Il segnale ottenuto ha lunghezza 50, calcolarne la DFT utilizzando $M = 4000$ e fare un plot del suo valore assoluto. Qual è la differenza rispetto a quanto ottenuto al punto precedente?

3. Considerare il seguente segnale
$$x_f = \cos\left(\frac{2\pi f t}{8000}\right).$$
Il parametro f definisce la frequenza della sinusoide. Impostando adeguatamente il parametro f possiamo rappresentare le varie note musicali. In particolare, consideriamo le seguenti frequenze:

- C4 = 261.63,
- D4 = 293.66,
- F4 = 349.23,
- G4 = 392.00,

che rispondono rispettivamente alle note Do, Re, Fa e Sol. Definiamo i seguenti segnali:

- $x_1 = \cos\left(\frac{2\pi C4 t_1}{8000}\right)$,
- $x_2 = \cos\left(\frac{2\pi C4 t_2}{8000}\right)$,
- $x_3 = \cos\left(\frac{2\pi D4 t_3}{8000}\right)$,
- $x_4 = \cos\left(\frac{2\pi C4 t_3}{8000}\right)$,
- $x_5 = \cos\left(\frac{2\pi G4 t_3}{8000}\right)$,
- $x_6 = \cos\left(\frac{2\pi F4 t_4}{8000}\right)$,

dove t_1 è un vettore di 2400 punti equispaziati da 1 a 2400, t_2 è un vettore di 1600 punti equispaziati da 1 a 1600, t_3 è un vettore di 3200 punti equispaziati da 1 a 3200 e t_4 è un vettore di 4000 punti equispaziati da 1 a 4000. Calcolare la DFT dei sequenti segnali e fare un plot del valore assoluto dei primi 500 coefficienti. Salvare i segnali come file audio utilizzando il comando `scipy.io.wavfile.write('sound1.wav',8000,x1)` e

provare ad ascoltarli.
Concatenare i segnali in un unico vettore $x = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6]$ (utilizzare la funzione `concatenate` della libreria `numpy`), calcolare la sua DFT e fare un plot del suo valore assoluto. Salvare il segnale x come file audio utilizzando il comando

```
scipy.io.wavfile.write('happy_birthday.wav',8000,x)
```

e provare ad ascoltarlo.