

MCS 2020
Algebra lineare numerica
Compressione di immagini tramite la DCT

Silva Edoardo 816560, Zhigui Bryan 816335, Marchetti Davide 815990

21/05/2020

Abstract

Si vuole avere un confronto dei tempi d'esecuzione della DCT2 della libreria scelta con la nostra implementazione. I problemi da risolvere sono delle matrici quadrate nella quale si definisce un offset di partenza che definirà la matrice iniziale e quelle avvenire fino a una dimensione massima denotata di default. I risultati ottenuti saranno riportati graficamente in una scala semi-algoritmica.

1 Librerie

1.1 Calcoli matematici

Il pacchetto **Numpy** è una libreria open source che contiene diverse funzioni e metodi utili per il calcolo scientifico. In particolar modo, per il calcolo vettoriale e matrici multidimensionale in maniera efficiente e veloce.

Il pacchetto **SciPy** è una libreria open source che utilizza la funzionalità di Numpy per fornire un pacchetto di calcolo scientifico General-purpose. In particolare, in questo progetto si utilizza la estensione Fast Fourier Transform (fftpack) per lavorare con le `dct` e `dctn(dct2)`.

1.2 Grafici

Il pacchetto **Pandas** consente la manipolazione di dati in formato tabellare o sequenziale.

Il pacchetto **Matplotlib** consente la generazione di grafici 2D. è costituita da 3 componenti importanti:

1. La **matplotlib** API: viene utilizzata per inserire funzionalità per la creazione di grafici nei propri script python
2. Il modulo **pyplot**: è una interfaccia in relazione con la componente precedente ed è progettato per emulare le stesse funzionalità grafiche rispetto a MATLAB.
3. Ouput back-end: produce un output dei grafici su varie tipologie di GUI e su diversi formati di file

Il pacchetto **Seaborn** serve per la creazione di grafici statistici. Legato a Matplotlib e strettamente integrato con le strutture dei dati Pandas.

2 Funzione DCT2 custom

La DCT2 è stata creata grazie all'utilizzo della DCT che viene applicata inizialmente sulle colonne e successivamente sulle righe. Inoltre, sapendo che lavora solo su vettori si è utilizzata la libreria numpy.

2.1 Funzione DCT2

Si ha in input una matrice M di dimensione $N \times N$ sulla quale viene creata una secondaria C con le stesse dimensioni ma inizializzata tutta a 0.

Inizialmente si effettua la DCT sui vettori colonna della matrice originale salvando i risultati ottenuti in colonna sulla matrice C .

Successivamente, viene applicata la DCT sui vettori riga della matrice C sovrascrivendo i risultati ottenuti sulle righe della stessa matrice. Infine, in output risiede la nuova Matrice C .

2.2 Funzioni DCT e IDCT

La funzione DCT prende in input un vettore V di dimensione n avente come valori i coefficienti $f(k)$. Viene creato un vettore C di dimensione n inizializzata tutta a 0. Ogni coefficiente letto viene applicata la formula (come spiegata a lezione) ricavando così i coefficienti $c(k)$ che saranno salvati tutti sul vettore C dato in output.

Viene effettuato lo stesso flusso rispetto alla IDCT.

3 Oggetto custom python(Dizionario?)

Si è deciso di creare un oggetto customizzato in modo tale da avere i risultati in un formato standard per ogni problema risolto dalle due funzioni messe a confronto.

L'oggetto ha le seguenti caratteristiche:

- È una matrice principale che contiene un numero di record = $2 * (\text{\#problemi_risolti})$.
- Ogni vettore riporta le seguenti informazioni:
 - Dimensione della matrice risolta (o problema);
 - Tempo di esecuzione: che si calcola grazie alla libreria time di python (`time.perf_counter()`);
 - Tipo di libreria usata: scipy oppure custom.

4 Creazione grafico

Viene creata una figura avente due assi cartesiani con:

- Le ordinate che rappresentano i tempi di esecuzione in scala logaritmica;
- Le ascisse che riportano la dimensione dei problemi risolti.

I dati presi dall'oggetto custom vengono trasformati in Dataframe mediante la libreria **Pandas**.

Ogni dataframe viene rappresentato come una “pallina” nel grafico e viene congiunto agli altri mediante una linea. Da notare che i risultati vengono differenziati rispetto al tipo (implemented oppure scipy) e poi viene visualizzato tutto graficamente con l'utilizzo della libreria Seaborn.

5 Conclusioni

Dallo studio grafico possiamo notare che la DCT2 customizzata ha un tempo di esecuzione $O(N^3)$ infatti nel grafico continua a crescere la linea mentre la DCT2 della libreria Scipy ha un tempo di esecuzione $O(N^2 \log N)$ ovvero migliore alla nostra.

C'è da considerare alcuni aspetti però:

- Noi utilizziamo una DCT2 che si basa sull'esecuzione della DCT per ben due volte e questo porta a un radico rallentamento del tempo di esecuzione, basandoci sulle formule basilari.
- La DCT2 della libreria oltre al fatto di effettuare un'operazione del tutto compatta utilizza delle formule matematiche (FAST FOURIER) che ne permette il calcolo molto più veloce.