92-quantum-shor-discrete-fourier-transform-03

Quantum Computing

Discrete Fourier Transform (DFT) e Quantum Fourier Transform (QFT)

1

1

trasformata di Fourier

- data una funzione, lo scopo della trasformata di Fourier è quello di decomporre la funzione nella somma di funzioni sinusoidali a diverse frequenze
  - innumerevoli applicazioni

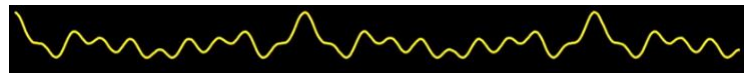
2

2

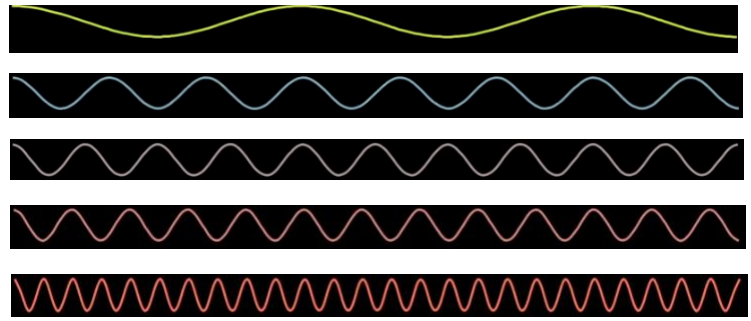
92-quantum-shor-discrete-fourier-transform-03

trasformata di Fourier

- es: data



- si vuole decomporla nella somma di

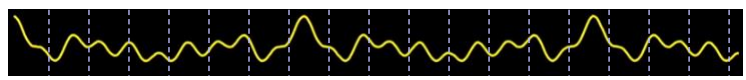


3

3

trasformata di Fourier discreta

- nella maggior parte delle applicazioni la funzione da decomporre non è nota in ogni istante ma soltanto in una sequenza di istanti equi-distanziati



- in questi casi la decomposizione viene calcolata con la DFT: Discrete Fourier Transform

4

4

calcolo della DFT

- dato il vettore di valori di una funzione in N istanti, i valori delle componenti della funzione alle diverse frequenze si ottengono moltiplicando il vettore dei valori per la matrice:
- $$\frac{1}{\sqrt{N}}\begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)^2} \end{pmatrix}$$
- dove $\omega = e^{2\pi\frac{i}{N}} = \cos\frac{2\pi}{N} + i\sin\frac{2\pi}{N}$

5

5

un esempio per il valore di ω

- $\omega = e^{2\pi\frac{i}{N}} = \cos\frac{2\pi}{N} + i\sin\frac{2\pi}{N}$
- es. se $N = 8$ abbiamo $\omega = \cos\frac{2\pi}{8} + i\sin\frac{2\pi}{8} = \cos\frac{\pi}{4} + i\sin\frac{\pi}{4} = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}i$

6

6

92-quantum-shor-discrete-fourier-transform-03

definizione della matrice

- per ricordare i valori degli elementi della matrice basta numerare righe e colonne da 0 a $N-1$
- l'elemento in posizione j, k ha valore ω^{jk}
- $$\frac{1}{\sqrt{N}}\begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)^2} \end{pmatrix}$$
- la matrice che definisce la DFT è usata anche per la trasformata di Fourier nel quantum computing (Quantum Fourier Transform) e si denota con QFT_N 7

7

formula di Eulero

- nota come l'uguaglianza $\omega = e^{2\pi\frac{i}{N}} = \cos\frac{2\pi}{N} + i\sin\frac{2\pi}{N}$ discenda dalla *formula di Eulero* che stabilisce
$$e^{i\alpha} = \cos\alpha + i\sin\alpha$$

8

8

92-quantum-shor-discrete-fourier-transform-03

esempio – calcolo di QFT_4

- calcoliamo QFT_4
- per farlo calcoliamo $\omega = e^{2\pi\frac{i}{N}} = \cos\frac{2\pi}{N} + i\sin\frac{2\pi}{N}$ con $N = 4$
- abbiamo $\omega = \cos\frac{2\pi}{4} + i\sin\frac{2\pi}{4} = 0 + i = i$
- quindi $\omega^0 = 1$, $\omega^1 = i$, $\omega^2 = -1$ e $\omega^{N-1=3} = -i$
- $\mathrm{QFT}_4 = \frac{1}{2}\begin{pmatrix} \omega^0 & \omega^0 & \omega^0 & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \omega^3 \\ \omega^0 & \omega^2 & \omega^4 & \omega^6 \\ \omega^0 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}$

9

9

proprietà generale

- notare come tutti i coefficienti della matrice QFT_4 in modulo al quadrato abbiano valore 1
- questa è una proprietà generale degli elementi di QFT_N

10

10

92-quantum-shor-discrete-fourier-transform-03

calcolo di ω^M per $M > N$

- dato che $\omega = e^{2\pi\frac{i}{N}}$ abbiamo che $\omega^N = e^{2\pi i}$
- per la formula di Eulero abbiamo $e^{2\pi i} = \cos 2\pi + i \sin 2\pi = 1$
- quindi $\omega^N = e^{2\pi i} = 1$
- per calcolare ω^M con $M > N$ basta dividere M per N e calcolarne il resto R, quindi scrivere $M = QN + R$ che equivale a dire $R = M \pmod N$
- allora $\omega^N = \omega^{QN+R} = \omega^R$

11

11

uso di QFT_4 per calcolare la DFT

- se $N = 4$ possiamo applicare QFT_4, ignorando il quantum computing, per calcolare la DFT di una funzione f nota in 4 punti
- $$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}$$
- gli α_i sono i valori di f negli istanti di campionamento e i β_i sono i coefficienti che danno le componenti di f alle diverse frequenze

12

12

quanto costa calcolare la DFT?

- calcolare la DFT richiede di eseguire un prodotto riga per colonna
  - dato un certo valore di N una implementazione semplice ha complessità $O(N^2)$
  - esiste un celebre algoritmo, noto come FFT (Fast Fourier Transform) che ha complessità $O(N \log N)$
    - tra i più importanti algoritmi dal punto di vista applicativo

13

13

uso di QFT_4 per calcolare la QFT

- se applichiamo QFT_4 nel quantum computing abbiamo che gli α_i sono i valori di ampiezza di *due* qubit in input e i β_i sono i valori di ampiezza di due qubit in output
- $\alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$
- $$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}$$

14

14

un miglioramento esponenziale

- gli N valori in input alla DFT possono essere codificati nel quantum computing come le $2^n = N$ ampiezze che corrispondono ad un registro di n qubit
- analogo ragionamento vale per l'output
- il circuito che calcola la QFT ha un numero di gate da uno o due qubit pari a $O(n^2) = O(\log N^2)$! e la componente quadratica può essere migliorata !

15

15

un miglioramento esponenziale

- ad esempio, se nel calcolo tradizionale $N = 256$ i 256 valori in input e in output sono rappresentati da vettori di $N = 256$ elementi
- nel quantum computing possiamo codificare gli $N = 256$ valori in input e in output come le 2^8 ampiezze di 8 qubit

16

16

92-quantum-shor-discrete-fourier-transform-03

tutto perfetto?

- nel calcolo classico l'input è costituito da N numeri complessi e anche l'output
- nel quantum computing l'input è rappresentato nello stato di n qubit e anche l'output
- se è vero che il calcolo si fa in tempo $O(n^2) = O((\log N)^2)$, è anche vero che tutto ciò che è visibile quando si misura è uno stato che corrisponde ad un numero j con probabilità β_j
- una primitiva potentissima rischia di risultare inutile

17

17

la QFT è una unitary transformation?

- piuttosto che dimostrarlo in generale verifichiamolo per QFT_4 calcolandone il prodotto con la sua trasposta coniugata
- $$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \times \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

18

18

92-quantum-shor-discrete-fourier-transform-03

esempio – calcolo di QFT_N di $|00\cdots0\rangle$

- se calcoliamo QFT_N con input $|00\cdots0\rangle$ (n qubit in input tutti a $|0\rangle$) otteniamo in output

$$\frac{1}{\sqrt{N}}\begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)^2} \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- cioè

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$$

19

19

esempio – calcolo di QFT_N di $|00\cdots0\rangle$

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$$

- si noti come x nella somma sia il numero intero corrispondente alla codifica di uno stato base
- abbiamo messo in uno stato di equilibrio gli N numeri interi dati da tutti i possibili stati base degli n qubit

20

20