Quantum Fourier Transform

Quantum computing has brought groundbreaking advancements by efficiently performing tasks that are beyond the reach of classical computers. One remarkable achievement is the ability to factor large numbers exponentially faster than classical algorithms. While classical methods require an impractical amount of time, a quantum algorithm, such as the quantum Fourier transform, can accomplish the same task with significantly fewer operations.

The quantum Fourier transform is a key component in quantum factoring and other intriguing quantum algorithms. It allows for the efficient computation of Fourier transforms of quantum mechanical amplitudes. This transformative algorithm opens doors to phase estimation, which approximates eigenvalues of unitary operators under specific conditions.

By leveraging phase estimation, we can tackle challenging problems like order-finding and factoring, which were previously considered infeasible on classical computers. Additionally, the quantum Fourier transform combined with the quantum search algorithm enables us to count solutions to search problems.

The quantum Fourier transform also holds potential for solving the hidden subgroup problem, a generalization of phase estimation and order-finding. It even offers an efficient quantum algorithm for the discrete logarithm problem, which remains a formidable challenge for classical computers.

Definition

The quantum Fourier transform is the classical discrete Fourier transform applied to the vector of amplitudes of a quantum state, which usually has length

$N=2^n$

The **classical Fourier transform** acts on a vector $\left(x_0,\ x_1, \cdots\ x_{N-1}\right) \in C^N$ and maps it to the vector$\left(y_0,\ y_1, \cdots\ y_{N-1}\right) \in C^N$ according to the formula :

$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \omega_N{}^{-nk}, k = 0, 1, 2, ..., N-1$$

Where $\omega_N = e^{\frac{2\Pi i}{N}}$ and ω_N^n is an N-th root of unity.

Similarly, the quantum fourier transform acts on a quantum state $|j\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$ and maps it to a quantum state $\sum_{k=0}^{N-1} y_k |k\rangle$ according to the formula

$$|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi ijk}{N}}$$

where the amplitudes y_k are the discrete Fourier transform of the amplitudes x_j .

Generally, we write the state $|j\rangle$ using the binary representation $j = j_1 j_2, \cdots j_n$ which is equals to $j = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0$. It is also convenient to adopt the notation $0.j_l j_{l+1} \cdots j_m$ to represent the binary fraction

$$j_{\frac{l}{2}} + j_{\frac{l+1}{4}} \dots j_{\frac{m}{2^{m-l+1}}}$$

.

The quantum fourier transform can be represented as

$$|j_1 j_2 j_3 \dots j_n\rangle \rightarrow (|0\rangle + e^{2\pi i 0.j_n} |1\rangle)(|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle) \dots \dots (|0\rangle + e^{2\pi i 0.j_1 j_2 j_3 \dots j_n} |1\rangle)$$

To obtain this unitary transformation, we can use the unitary gate <R sub{k}>

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}$$

To understand quantum fourier transform more clearly, let's walk through each step of the transformation when the input state is $| j_2, \cdots j_n\rangle$:

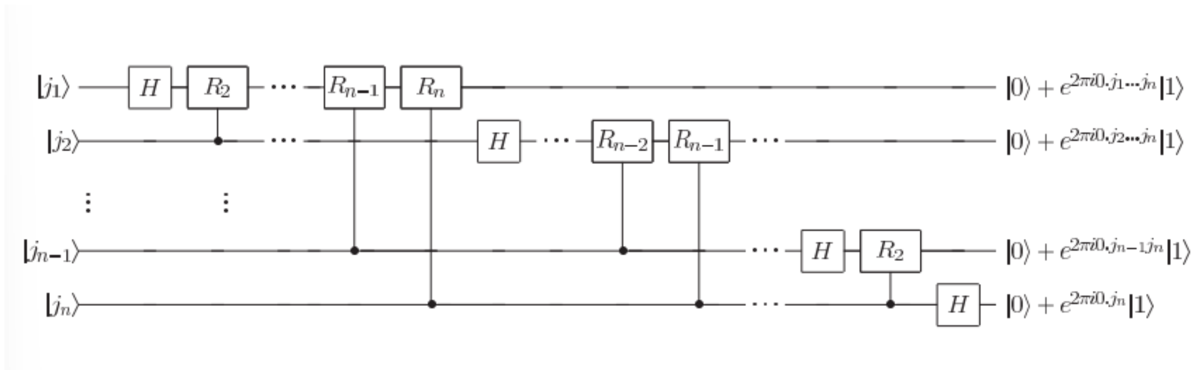

Figure - 1.1

1 : After applying the Hadamard gate to the first qubit produces the state

$$\frac{1}{2^{\frac{1}{2}}}\left(|0\rangle + e^{2\pi i 0.j_1}\ |1\rangle\right)|j_2 j_3 \cdots j_n\rangle$$

2 : After applying the first controlled- R_2 gate on the first qubit produces the state

$$\frac{1}{2^{\frac{1}{2}}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2}\ |1\rangle\right)|j_2 j_3 \cdots j_n\rangle$$

3 : After applying the last controlled- R_2 gate on the first qubit produces the state

$$\frac{1}{2^{\frac{1}{2}}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n}\ |1\rangle\right)|j_2 j_3 \cdots j_n\rangle$$

4 : Next, we add a Hadamard gate to the second qubit, which puts us in the state

$$\frac{1}{2^{\frac{2}{2}}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n}\ |1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2}\ |1\rangle\right)|j_3 j_4 \cdots j_n\rangle$$

5 : After applying the controlled-R_2 through R_{n-1} gates gives us the state

$$\frac{1}{2^{\frac{2}{2}}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n}\ |1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2 \ldots j_n}\ |1\rangle\right)|j_3 j_4 \cdots j_n\rangle$$

We continue in this fashion for each qubit, giving a final state

$$\frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \ldots j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_2 \ldots j_n}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)$$

Swap operations , omitted from *figure 1.1* for clarity, are then used to reverse the order of the qubits. After the swap operations, the state of the qubits is

$$\frac{1}{2^{n/2}}\left(|0\rangle + e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle + e^{2\pi i 0.j_{n-1} j_n}|1\rangle\right)\ldots\left(|0\rangle + e^{2\pi i 0.j_1 j_2 \cdots j_n}|1\rangle\right)$$

Which is also represented as

$$\frac{1}{2^{\frac{n}{2}}}(|0\rangle + e^{\frac{2\pi ij}{2}}|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi ij}{2^2}}|1\rangle) \otimes \ldots \otimes (|0\rangle + e^{\frac{2\pi ij}{2^{n-1}}}|1\rangle) \otimes (|0\rangle + e^{\frac{2\pi ij}{2^n}}|1\rangle)$$

This is the required output from the quantum Fourier transform.This construction also proves that the quantum Fourier transform is unitary, since each gate in the circuit is unitary.

Number of gates :

We start by doing a Hadamard gate and n − 1 conditional rotations on the first qubit – a total of n gates. This is followed by a Hadamard gate and n − 2 conditional rotations on the second qubit, for a total of n + (n − 1) gates. Continuing in this way, we see that n+ (n−1) +···+1 = n(n+ 1)/2 gates are required,plus the gates involved in the swaps. At most n/2 swaps are required, and each swap can be accomplished using three controlled-NOT gates.

Summary :

The Fourier transform plays a critical role in various real-world data processing applications. For instance, in computer speech recognition, the initial step involves Fourier transforming the digitized sound to facilitate phoneme recognition. However, when it comes to leveraging the quantum Fourier transform for accelerating these computations, it poses challenges. Unfortunately, there is currently no known method to achieve this.

The fundamental issue lies in the fact that the amplitudes within a quantum computer cannot be directly accessed through measurement. As a result, it becomes impossible to determine the Fourier-transformed amplitudes of the original state. Additionally, efficiently preparing the original state for the Fourier transformation is generally not feasible. These limitations imply that discovering practical applications for the quantum Fourier transform requires a more nuanced approach than initially anticipated.

- *Content written by- Akash*