Week 4: Fourier Based Methods

Contents

1 Quantum Fourier Transform

Classically, the Fourier transform (FT) decomposes a signal (a function of time) into its constituent frequencies. The Fourier transform is not limited to functions of time, but the domain of the original function is commonly referred to as the time domain. For example, in quantum mechanics the Fourier Transform allows us to transform between the momentum and position representations of a wave function. The motivation behind using Fourier Transforms is that linear operations performed in the original domain have corresponding operations in the Fourier domain, which are sometimes easier to perform.
The Fourier Transform is essentially a mapping from the original space to a "Fourier" space.

$$\sum_{x=0}^{N-1} a_x|x\rangle \xrightarrow{QFT} \sum_{y=0}^{N-1} \tilde{a}_y|y\rangle \tag{1}$$

QFT takes the coefficients (a_x) of a quantum state vector $|\phi\rangle$ with respect to the standard basis vectors ($|x\rangle$, such that $x \in [0, N-1]$) and produces new coefficients ($\tilde{a}_y$) with respect to the Fourier basis vectors ($|y\rangle$, such that $y \in [0, N-1]$). The most basic QFT is the Hadamard Transform for $N = 2$.
The individual basis states are transformed as follows

$$|x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \omega_N^{-xy}|y\rangle \tag{2}$$

From equations 1 and 2, we get an expression of the QFT operator as

$$U_{\text{QFT}} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \omega_N^{-xy}|y\rangle\langle x| \tag{3}$$

Before we dive into the circuit construction of QFT, it is important to note that unlike in the classical case, QFT does not give us the individual coefficients, but rather gives us the resulting quantum state.
We have seen that

$$\text{QFT}_2 = H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Similarly, for the general case we have

$$\text{QFT}_N = \frac{1}{\sqrt{N}} \begin{bmatrix} . & . & . & . & . \\ . & . & . & . & . \\ . & . & \omega_N^{jk} & . & . \\ . & . & . & . & . \\ . & . & . & . & . \end{bmatrix} \tag{4}$$

where $\omega_N = e^{2\pi i/N}$. QFT_N is unitary and hence we can create a quantum circuit for QFT. We create an efficient circuit for QFT using Hadamard gates and controlled R_s gates.

$$R_s = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^s}} \end{bmatrix} \tag{5}$$

Suppose we have a number $a = a_1 a_2 .... a_n$, where a_i is a bit with a_1 being the most significant bit and a_n being the least significant bit. Now, we can write $a/2^n = 0.a_1 a_2 .... a_n = \sum_{k=1}^{n} a_k 2^{-k}$. Lets look at a numeric example. Let $a = 31$ and $n = 5$.

$$\begin{aligned}
\frac{31}{32} &= \frac{(16+8+4+2+1)_{10}}{(32)_1 0} \\
&= \frac{(11111)_2}{2^5} \\
&= (1)_2 * 2^{-1} + (1)_2 * 2^{-2} + (1)_2 * 2^{-3} + (1)_2 * 2^{-4} + (1)_2 * 2^{-5} \\
&= (0.1)_2 + (0.01)_2 + (0.001)_2 + (0.0001)_2 + (0.00001)_2 \\
&= 0.11111
\end{aligned}$$

The Fourier transform can then be expressed as

$$\begin{aligned}
F_N|y\rangle &= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{jy} |j\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i jy/N} |j\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i jy/2^n} |j\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i (\sum_{k=1}^{n} j_k 2^{-k}) y} |j_1 j_2 .... j_n\rangle \\
&= \frac{1}{\sqrt{2}} \sum_{j=0}^{N-1} \prod_{k=1}^{n} e^{2\pi i j_k 2^{-k} y} |j_1 j_2 .... j_n\rangle \\
&= \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i y/2} |1\rangle\right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i y/2^2} |1\rangle\right) .... \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i y/2^n} |1\rangle\right) \\
&= \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 0.y_n} |1\rangle\right) \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 0.y_{n-1} y_n} |1\rangle\right) .... \otimes \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i 0.y_1 ..... y_{n-1} y_n} |1\rangle\right)
\end{aligned} \tag{6}$$

where the i^{th} term in the tensor product refers to the i^{th} qubit, and the first term is the most significant bit. This is an useful form of the QFT for $N = 2^n$ and we will use it to construct the actual circuit in the exercises.

2 Quantum Phase Estimation

Suppose we can have a unitary operator U and we are given an eigenvector of U, namely $|\psi\rangle$.

$$U|\psi\rangle = \lambda|\psi\rangle$$

Now the task at hand is to approximate the corresponding eigenvalue λ. Since U is unitary, $|\lambda| = 1$ (as we have seen in an earlier exercise), and therefore we can write λ as a root of unity $e^{2\pi i\phi}$, where $\phi \in [0, 1)$.
To estimate the eigenvalue λ the only requirement is to estimate the phase ϕ. The algorithm is detailed below

- Start with initial state $|0\rangle^{\otimes n}|\psi\rangle$.
- Applying an n-bit Hadamard Gate on the first qubit gives us

$$|0\rangle^{\otimes n}|\psi\rangle \xrightarrow{H^{\otimes n}\otimes I} \frac{1}{\sqrt{2^n}}\left(|0\rangle + |1\rangle\right)^{\otimes n}|\psi\rangle$$

- Applying the unitary operator U 2^j times would yield an eigenvalue

$$\begin{aligned} U|\psi\rangle &= \lambda|\psi\rangle = e^{2\pi i\phi}|\psi\rangle \\ \implies U^{2^j}|\psi\rangle &= e^{2\pi i\phi 2^j}|\psi\rangle \end{aligned}$$

- Now if we applied a controlled unitary C-U^{2^j} on the second register by defining the j^{th} qubit of the first register as the control for the unitary C-U^{2^j}, we get the following state

$$\Bigg(\underbrace{\underbrace{\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\phi 2^0}|1\rangle\right)}_{\text{nth qubit}} \otimes \underbrace{\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\phi 2^1}|1\rangle\right)}_{\text{(n-1)th qubit}} .... \otimes \underbrace{\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\phi 2^n}|1\rangle\right)}_{\text{1st qubit}}}_{\text{1st register}}\Bigg) \otimes \underbrace{|\psi\rangle}_{\text{2nd register}}$$

$$= \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2\pi i\phi k}|k\rangle \otimes |\psi\rangle$$

- Now we apply inverse QFT on the first register

$$\begin{aligned} \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2\pi i\phi k}|k\rangle \otimes |\psi\rangle &\xrightarrow{\text{QFT}^\dagger\otimes I} \frac{1}{N}\sum_{x=0}^{N-1}\sum_{k=0}^{N-1} e^{2\pi i\phi k}e^{-2\pi ikx/2^n}|x\rangle \otimes |\psi\rangle \\ &= \frac{1}{N}\sum_{x=0}^{N-1}\sum_{k=0}^{N-1} \exp\left(\frac{-2\pi ik}{2^n}\left(x - 2^n\phi\right)\right)|x\rangle \otimes |\psi\rangle \end{aligned}$$

- We can write $2^n\phi$ in terms of the nearest integer α as $2^n\phi = \alpha + 2^n\beta$, where $0 \leq |2^n\beta| \leq 0.5$. Then our phase in the first register changes to

$$\sum_{k=0}^{N-1} \exp\left(\frac{-2\pi ik}{2^n}(x-a)\right) \exp\left(2\pi ik\beta\right)$$

  Let us designate this phase as Φ_x.

- Now we perform a measurement in the computational basis state on the first register

$$\begin{aligned}
\Pr[a] &= \left| \left\langle a \left| \frac{1}{N} \sum_{x=0}^{N-1} \Phi_x \right| x \right\rangle \right|^2 \\
&= \left| \frac{1}{N} \sum_{x=0}^{N-1} \Phi_x \langle a|x\rangle \right|^2 \\
&= \frac{1}{N^2} \left| \sum_{k=0}^{N-1} e^{(2\pi ik\beta)} \right|^2
\end{aligned}$$

  In the third step, since $|a\rangle$ and $|x\rangle$ are computational basis states, either $x = a \implies \langle a|x\rangle = 1$, or, $x \perp a \implies \langle a|x\rangle = 0$. Thus we can rewrite Φ_x as

$$\Phi_x = \sum_{k=0}^{N-1} e^{(2\pi ik\beta)}$$

  If the approximation is precise, i.e. $2^n\phi = \alpha$ and $\beta = 0$, $\Pr[a] = 1$. Even when the approximation is not precise we can show that $\Pr[a] \geq \frac{4}{\pi^2}$ (We shall explore this in the exercise).

3 Shor's Algorithm

Imagine we are given a composite number N whose input size is $n = log_2(N)$ bits. A very important question with respect to x is: *Is it possible to find a factor of N*?

Finding a factor is easy if N is even because 2 will be a factor of any even N. What if N is odd? How hard is it to find atleast one factor? A naive approach is as follows - for each number $a < \frac{N}{2}$ check greatest common divisor (gcd[1]) of a and N. If $\gcd(a, N)$ is not equal to 1, then a is a factor of N. This will take $O(N)$ computations.

Now, assume that we are given N as a product of just two numbers, say p and q. Can we do any better than $O(N)$ computations? Observe that since $N = p \cdot q$, one of the p and q will be $\leq \sqrt{N}$. It is so because if both p and q are $> \sqrt{N}$, then the product $p \cdot q$ will be greater than N. So as an improvement, we just need to check the gcd of N with the numbers that are less than or equal to $\sqrt{N}$. This has reduced our computational complexity from $O(N)$ to $O(\sqrt{N})$. However, in terms of the input size, the complexities are exponential in n. There are better classical algorithms. The fastest known classical algorithm is the quadratic sieve field that solve this problem in $\tilde{O}\big(exp\big\{\big(log(n)\big)^{\frac{1}{2}} \cdot \big(log(log(n))\big)^{\frac{2}{3}}\big\}\big)$. Despite being faster that the naive approach, the computational complexity of sieve field still exponential in n.

Factoring problem is a problem of huge importance because the security of cryptographic protocols like RSA depends on the assumption that factoring is classically intractable. As a consequence, RSA is one of the most widely used encryption protocol in the world currently. If an algorithm can solve the factoring problem in polynomial time, then RSA would cease to exist as a viable cryptographic protocol.

As in the quantum setting, in contrast to the classical algorithms, Shor's algorithm helps solve the factoring problem in time polynomial in n. More specifically, Shor's algorithm helps solve the factoring problem in

$$O\big(\big(log(N)\big)^2\big(log\ log(N)\big)\big(log\ log\ log(N)\big)\big)$$

steps. Note that the quantum algorithm given by Shor does not directly compute a factor of the given composite number N. Rather the algorithm is a period finding algorithm. We will now see what a period finding problem is and how the factoring problem can be reduced to period finding problem.

Suppose we are given an N that is odd and is not a prime power. Now, let us randomly choose $x \in \{2, ..., \lceil \frac{N}{2} \rceil\}$. If x is not coprime to N, then $gcd(x, N)$ is a non-trivial factor of N. Now, if x is coprime to N, then we have $gcd(x, N) = 1$. Then x is a member of the multiplicative group Z_N^*. Let r be the order of x in Z_N^*. Then, we have $x^r = 1 \bmod N$. Since N is odd and is not a prime power, we have that the period is even such that $x^{r/2} + 1$ and $x^{r/2} - 1$ are not multiples of N with probability $\geq \frac{1}{2}$. Then, we have, $x^r = 1 \bmod N \iff (x^{r/2} + 1)(x^{r/2} - 1) = 0 \bmod N$. Since $x^{r/2} + 1$ and $x^{r/2} - 1$ are not multiples of N, $gcd(x^{r/2}+1, N)$ and $gcd(x^{r/2}-1, N)$ are some non-trivial factors of N. But these are subject to certain condition on x. Trying out different x, we will obtain a factor of N with high probability. Let $f(a)$ be a function defined as $f(a) = x^a \bmod N$. Then, we can see that our goal is to find r such that $f(a) = f(a + (k \cdot r))$. Hence the factoring problem is reduced

[1]Computation of $gcd(a, N)$ can be done using Euclid's algorithm in just $O(log(N))$ which is computationally very cheap.

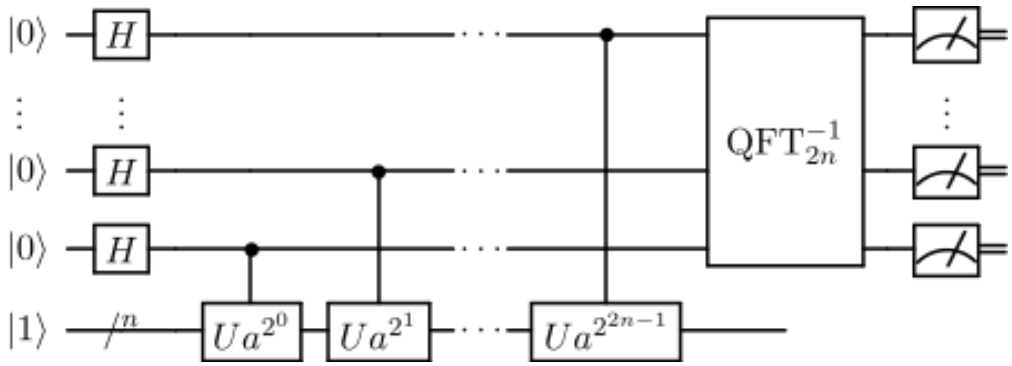

Figure 1: Circuit of Shor's Algorithm (Image credits: Wikipedia)

to the period find problem. The period finding problem can be, more formally, defined as the problem of finding $r \in \{0, ..., N-1\}$ given a function $f : \mathbb{N} \longrightarrow \{0, ..., N-1\}$ with the property that $f(x) = f(y)$ if and only if $x = y \bmod r$ for all $x, y \in \mathbb{N}$.

Given a black box, say O_f, that computes f, the circuit of the Shor's period finding algorithm is given in Figure 1. Let $q = 2^k$ be such that $N^2 < q < 2N^2$. The circuit starts with the state $\left|0^k\right\rangle |0\rangle$ where $n = \lceil log(N) \rceil$. First, we implement the quantum Fourier transform on the first register of $\left|0^k\right\rangle$ qubits. Then the resultant state will be

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{2^n-1} |a\rangle |0^n\rangle.$$

Next we apply the oracle O_f to get the state

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{2^n-1} |a\rangle |f(a)\rangle.$$

Suppose we obtained the state $f(s)$ for some $s < r$. Then first register in the post measurement state will be a superposition of all the states of the form $|l \cdot r + s\rangle$ where $l \cdot r + s \in \{0, ..., q-1\}$. Let m be the number of states of the form $|l \cdot r + s\rangle$. The the post measurement state ignoring the second register is given as

$$\tfrac{1}{\sqrt{m}} \textstyle\sum_{l=0}^{m-1} |l \cdot r + s\rangle.$$

Applying QFT again on register 1, we get the state

$$\begin{aligned} &\frac{1}{\sqrt{m}} \sum_{i=0}^{m-1} \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} e^{2\pi i \frac{(l \cdot r+s)j}{q}} |j\rangle \\ &= \frac{1}{\sqrt{mq}} \sum_{j=0}^{q-1} e^{2\pi i \frac{sj}{q} \left(\sum_{l=0}^{m-1} e^{2\pi i \frac{lrj}{q}} \right)} \end{aligned}$$

Now, we have

$$\textstyle\sum_{l=0}^{m-1} e^{2\pi i \frac{lrj}{q}} = \begin{cases} m, & \text{if } e^{2\pi i \frac{rj}{q}} = 1 \\ (1 - e^{2\pi i \frac{mrj}{q}})/(1 - e^{2\pi i \frac{rj}{q}}), & \text{if } e^{2\pi i \frac{rj}{q}} \neq 1 \end{cases}$$

Let us first look at the case where r divide q. Note that $e^{2\pi i \frac{rj}{q}} = 1$ if and only if rj/q is an integer iff j is a multiple of q/r. Now the superposition is such that we have non-zero amplitudes only for j that are integral multiples of q/r. So on measuring, we get some j such that $j = cq/r$ for some random c. Since c will be coprime to r with probability $\Omega(1/\log\log \mathrm{r})$, we need an expected number of $O(\log\log \mathrm{r})$ repetitions to obtain a $j = cq/r$ with c coprime to r. Using that j, we can obtain r by reducing b/q to its lowest terms.

Now if r does not divide q, then algorithm will still output a j close to a multiple of q/r with high probability. We can write

$$\frac{\left|1 - e^{2\pi i \frac{mrj}{q}}\right|}{\left|1 - e^{2\pi i \frac{rj}{q}}\right|} = \frac{\left|sin(\pi mrj/q)\right|}{\left|sin(\pi rj/q)\right|} \text{ using the equality } \left|1 - e^{i\theta}\right| = 2\left|sin(\theta/2)\right| .$$

From the denominator of the RHS of the above equation, we can see that when j is close to a multiple of q/r, the above ratio is large and vice-versa. More precisely, we can show that we will obtain with high probability a j such that

$$\left|\frac{j}{q} - \frac{c}{r}\right| \leq \frac{1}{2q}.$$

Similar to the previous case, we have known j and q while c and r are unknown. We then use a classical method called "continued fraction expansion" to obtain r.

4 Exercises

1. Give the Fourier Basis states for the Hadamard Transform.
2. Show that the eigenvalues of a unitary matrix have modulus 1.
3. We claim that the most significant qubit in Equation 6 can be obtained by using only a Hadamard gate. Can you explain why?
4. Using nothing but H and R_s gates, derive a circuit that implements the form of equation 6 for 2 qubits. Extend this to 3 qubits now.
5. When $2^n\phi \neq a$, show that the probability of obtaining $\Pr[a]$ is approximately $\frac{4}{\pi^2}$.
6. Give the circuit for Quantum Phase Estimation using Hadamard gates, controlled Unitaries, and inverse QFT as a unitary.