Quantum Fourier Transform

Exponential speedup by quantum computation

1. It is easy to compute classically, then there is an efficient reversible circuit = there is an efficient quantum circuit
   on input $\sum_{x \in \{0,1\}^n} x |x\rangle$, output is $\sum_{x \in \{0,1\}^n} f(x) |x\rangle$
   example: $n=1$, $f(x) = x$
   input: $\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$, output: $\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$
   - just a CNOT gate

2. Quantum Fourier transform

   Discrete Fourier transform : modulo $N$, on $N \times N$ unitary matrix, $\omega_{N}^{xk}$

   $\begin{pmatrix}
   1 & 0 & \cdots & 0 \\
   0 & \omega_{N} & \cdots & 0 \\
   \vdots & \vdots & \ddots & \vdots \\
   0 & 0 & \cdots & \omega_{N}^{N-1}
   \end{pmatrix}$

   - standard basis
   - Fourier basis
   - inner product between $A^t$ and $B^t$

   $\begin{pmatrix}
   \omega_{N}^{k}\omega_{N}^{jk} \\
   \omega_{N}^{k}\omega_{N}^{jk} \\
   \vdots \\
   \omega_{N}^{k}\omega_{N}^{jk}
   \end{pmatrix}

   $e^{j\theta} = 1$ unless $\theta = \frac{2\pi}{N}$

   since if $\theta \neq 0 \mod N$, then $1 + e^{\frac{2\pi}{N}j} + \cdots + e^{\frac{2\pi}{N}N} j = 0$

   if $\theta = 0 \mod N$, then $1 + e^{\frac{2\pi}{N}j} + \cdots + e^{\frac{2\pi}{N}N} j = N$

   computing the discrete Fourier transform is essential for digital signal processing -- naive matrix multiplication takes $\Theta(N^2)$ steps
   - the FFT takes only $\Theta(N \log N)$ steps (1)

   how? Assume $N = 2^k$. Split the matrix into four parts, namely
   \[ \begin{pmatrix}
   x_{0} & x_{1} \\
   x_{2} & x_{3}
   \end{pmatrix} \]

   by columns into even and odd ones

   \[ \begin{pmatrix}
   x_{0} & x_{2} \\
   x_{1} & x_{3}
   \end{pmatrix} \]

   write $\omega = e^{j\frac{2\pi}{N}}$

   time to solve problem of size $N \cdot a$

   $T(N) = 2T(\frac{N}{2}) + O(N) \rightarrow T(N) = O(N \log N)$
Classical

O(n log n) steps

Quantum

log n levels, each taking O(n) time → O((n log n)

\[ N = 2^n \]

C-state of a qubit

will find a quantum circuit of size

O(n^3) = O((log n)^4).

exponential speedup

What's the catch? The output is \[ \sum_k x_k |k \rangle \], whereas classically you'd get the whole list \( x_1, x_2, \ldots, x_N \). All we can do is Fourier sampling: measure to get |x⟩ with probability \( \frac{1}{N} \). Even with this restriction, the quantum Fourier transform is quite powerful.

We want \[ |a⟩ → \frac{1}{\sqrt{N}} \sum_{a=0}^{N-1} |a⟩ |b⟩ \]

\[ a \]

\[ a_2 \]

\[ a_3 \]

\[ a_{00} \]

\[ a_{01} \]

\[ a_{10} \]

\[ a_{11} \]

\[ b_0 \]

\[ b_1 \]

\[ \frac{1}{N} \sum_{a=0}^{N-1} e^{2 \pi i a \cdot b} |a⟩ \]

What is the coefficient of \[ |b⟩ \]?

Why does the circuit do this?

\[ a_0 = 0 \]

\[ a_1 = 1 \]

\[ b_0 = 0 \]

\[ b_1 = 1 \]