

CPEN 400Q Lecture 09

The quantum Fourier transform (QFT)

Monday 6 February 2023

Announcements

- Quiz 4 today
- Assignment 1 due tonight
- Next literacy assignment coming this week

[HOME](#)[EVENTS](#)[PRIZES](#)[HACKATHON](#)[JOB BOARD](#)[FAQ](#)[SIGN IN](#)

Join the Party

Call it fan expo,
hackathon, or scientific
conference.

QHACK — the one-of-a-kind
celebration of quantum computing
returns!

Online | February 13–28, 2023 | Free

[Register now](#)

Register through Xanadu Cloud.

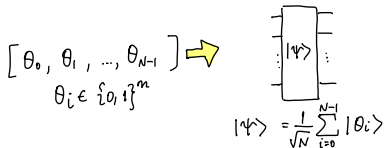
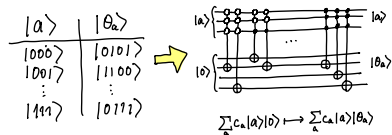
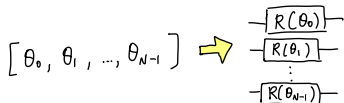
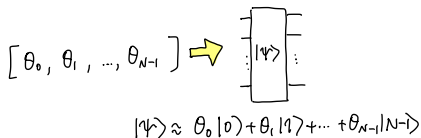
Need help? [Click here](#).



Visit qhack.ai for more details.

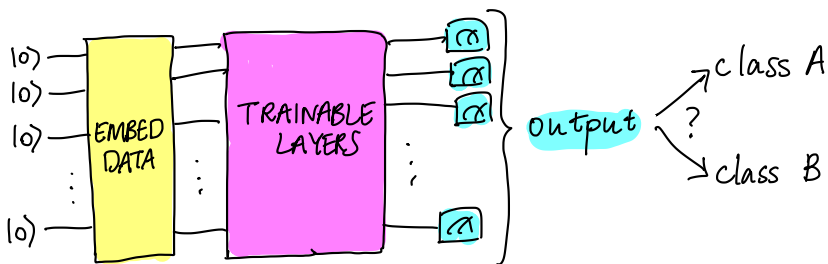
Last time

We explored different ways of encoding data into quantum circuits.



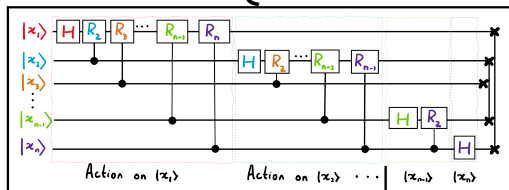
Last time

We successfully implemented a variational classifier in PennyLane.

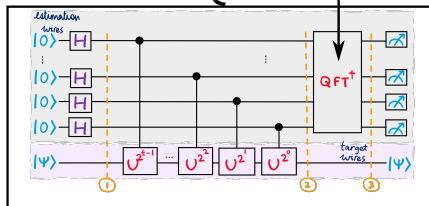


Where are we going?

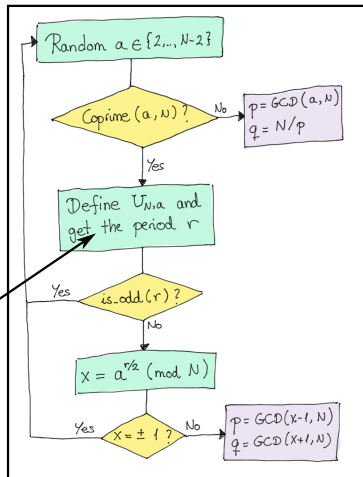
1. QFT



2. QPE



3. Shor



- Express floating-point values in fractional binary representation
- Describe the behaviour of the quantum Fourier transform
- Implement the quantum Fourier transform in PennyLane

Today there will be lots of MATH.

The discrete Fourier transform

From ELEC 221¹:

$$DFT = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \bar{\omega} & \bar{\omega}^2 & \dots & \bar{\omega}^{N-1} \\ 1 & \bar{\omega}^2 & \bar{\omega}^4 & \dots & \bar{\omega}^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \bar{\omega}^{N-1} & \bar{\omega}^{2(N-1)} & \dots & \bar{\omega}^{(N-1)(N-1)} \end{pmatrix}$$

where $\bar{\omega} = e^{-2\pi i/N}$.

Note: sometimes there a prefactor, depends how inverse is defined.

¹See Lecture 13: <https://github.com/glassnotes/ELEC-221>

The discrete Fourier transform

The DFT (and the fast Fourier transform which implements it efficiently) are standard tools in digital signal processing to convert between time and frequency domain.

Given a signal $x[n]$ in the time domain, the DFT computes

The discrete Fourier transform

The inverse DFT computes

$$\text{where } \omega = e^{2\pi i/N} = \bar{\omega}^{-1}$$

The DFT invertible; its matrix is unitary (up to a prefactor).
Seems like a good candidate for a quantum computer...

Quantum Fourier transform

The quantum Fourier transform (QFT) is the quantum analog of the **inverse DFT**.

Let $|x\rangle$ be an n -qubit computational basis state, $N = 2^n$.

The QFT sends

We are sending individual computational basis states to another basis, which is made up of linear combinations of computational basis states with complex exponential coefficients.

Quantum Fourier transform

The QFT has the following action on the basis states:

Check that this works...

Quantum Fourier transform

As a matrix, it looks a lot like the DFT:

$$QFT = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{pmatrix}$$

But... can we implement this unitary efficiently? How do we *synthesize* a circuit for it?

Quantum Fourier transform

Let's start with some special cases... suppose $n = 1$ ($N = 2$).

Here, $e^{2\pi i/2} = e^{i\pi} = -1$, so

Look familiar?

Quantum Fourier transform

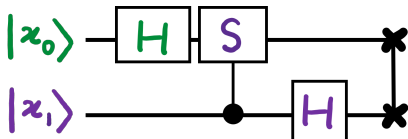
Suppose $n = 2$ ($N = 4$).

Here $\omega = i$, and $\omega^2 = -1$, $\omega^4 = 1$. So

Quantum Fourier transform

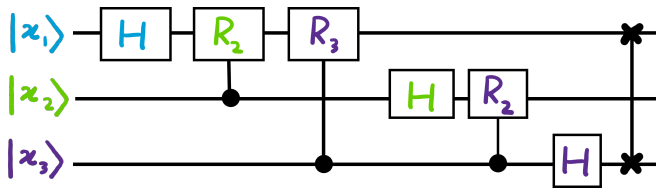
If we apply a SWAP, familiar things show up...

Top blocks are H , bottom are HS . Can show that the following circuit implements this QFT:



Quantum Fourier transform

Do the same for $n = 3$ ($N = 8$) but things get nasty... can show that the structure of the circuit that implements it is



Here, $R_2 = S$ and $R_3 = T$.

Image credit: Xanadu Quantum Codebook node F.3

Quantum Fourier transform

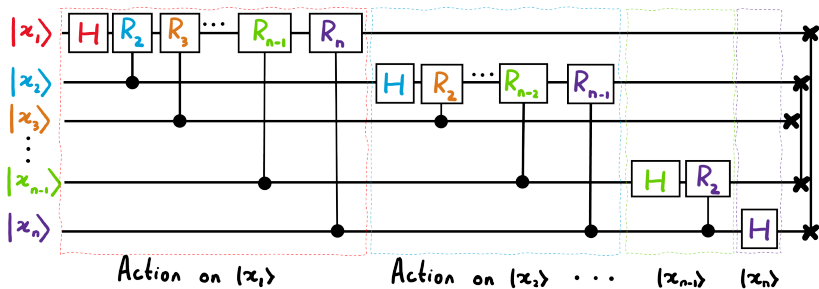


Image credit: Xanadu Quantum Codebook node F.3

A circuit for the QFT

Consider the expression

Here x and k are represented as integers.

They are n -qubit computational basis states so they also have binary equivalents $|x\rangle = |x_1 \cdots x_n\rangle$, $|k\rangle = |k_1 \cdots k_n\rangle$:

and similarly for k .

A circuit for the QFT

Recall that $\omega = e^{2\pi i/N}$.

We are working with

$$\omega^{xk} = e^{2\pi i x(k/N)}$$

with $N = 2^n$.

We can write a fraction $k/2^n$ in a 'decimal version' of binary:

Exercise: let $k = 0.11010$. What is the numerical value of k ?

A circuit for the QFT

Using the fractional decimal expression for k/N , we will work through the specification of the Fourier transform and see how we can reshuffle and *factor* the output state to get something that will make clear a circuit. Brace yourselves.

A circuit for the QFT

(keeping the last equation from the previous slide)

A circuit for the QFT

(keeping the last equation from the previous slide)

A circuit for the QFT

So...

$$|x\rangle \rightarrow \frac{(|0\rangle + e^{2\pi i 0.x_n} |1\rangle) (|0\rangle + e^{2\pi i 0.x_{n-1}x_n} |1\rangle) \dots (|0\rangle + e^{2\pi i 0.x_1 \dots x_n} |1\rangle)}{\sqrt{N}}$$

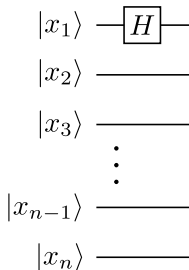
Believe it or not, this form reveals to us how we can design a circuit that creates this state!

A circuit for the QFT

Starting with the state

$$|x\rangle = |x_1 \cdots x_n\rangle,$$

apply a Hadamard to qubit 1:

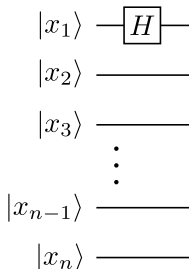


A circuit for the QFT

$$\frac{1}{\sqrt{2}} (|0\rangle + e^{2\pi i 0 \cdot x_1} |1\rangle) |x_2 \cdots x_n\rangle$$

If $x_1 = 0$, $e^0 = 1$ and we get the $|+\rangle$ state.

If $x_1 = 1$, $e^{2\pi i(1/2)} = e^{\pi i} = -1$
and we get the $|-\rangle$ state.



A circuit for the QFT

We are trying to make a state that looks like this:

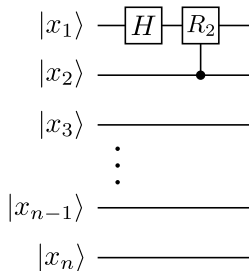
$$|x\rangle \rightarrow \frac{(|0\rangle + e^{2\pi i 0.x_n}|1\rangle)(|0\rangle + e^{2\pi i 0.x_{n-1}x_n}|1\rangle) \cdots (|0\rangle + e^{2\pi i 0.x_1 \cdots x_n}|1\rangle)}{\sqrt{N}}$$

Every qubit has a different *phase* on the $|1\rangle$ state. We are going to need some way of creating this.

We define the gate:

A circuit for the QFT

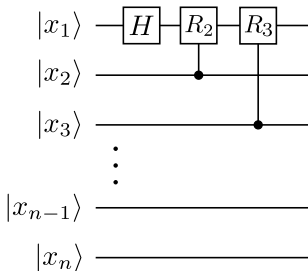
Apply controlled R_2 from qubit
 $2 \rightarrow 1$



First qubit picks up a phase:

A circuit for the QFT

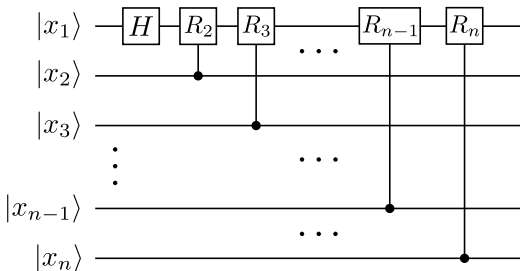
Apply controlled R_3 from qubit
 $3 \rightarrow 1$



First qubit picks up another phase:

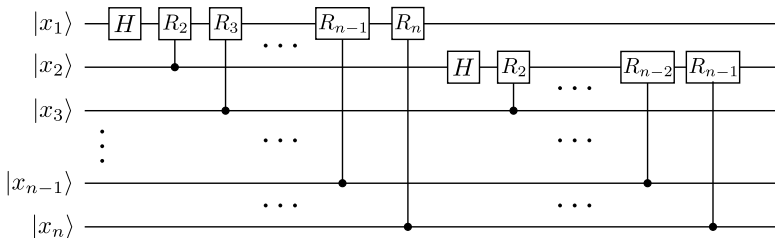
A circuit for the QFT

Apply a controlled R_4 from $4 \rightarrow 1$, etc. up to the n -th qubit to get



A circuit for the QFT

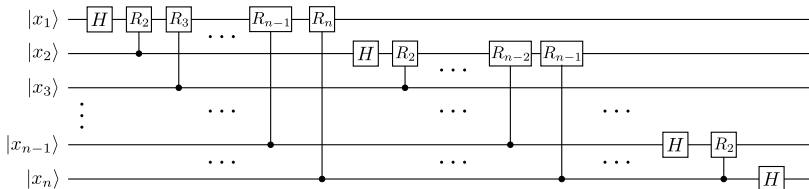
Next, do the same thing with the second qubit: apply H , and then controlled rotations from every qubit from 3 to n to get



A circuit for the QFT

Do this for all qubits to get that big ugly state from earlier:

$$|x\rangle \rightarrow \frac{(|0\rangle + e^{2\pi i 0 \cdot x_n} |1\rangle) (|0\rangle + e^{2\pi i 0 \cdot x_{n-1} x_n} |1\rangle) \dots (|0\rangle + e^{2\pi i 0 \cdot x_1 \dots x_n} |1\rangle)}{\sqrt{N}}$$

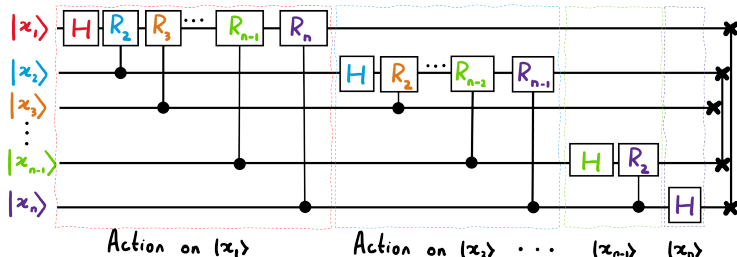


(though note that the order of the qubits is backwards - this is easily fixed with some SWAP gates)

Quantum Fourier transform

Gate counts:

- n Hadamard gates
- $n(n-1)/2$ controlled rotations
- $\lfloor n/2 \rfloor$ SWAP gates if you care about the order



The number of gates is *polynomial* in n !

Next time

Content:

- Quantum phase estimation

Action items:

1. Finish Assignment 1

Recommended reading:

- Codebook module F
- Nielsen & Chuang 5.1
- Codebook module P (for next class)