Introduction to quantum computing: Grover search

Erik Spence

SciNet HPC Consortium

6 June 2023

Today's code and slides

You can get the slides and code for today's class at the SciNet Education web page.

`https://scinet.courses/1290`

Click on the link for the class, and look under "Lectures", click on "Grover search".

Breaking locks

One of the classic problems quantum computing has been applied to is breaking combination locks. How would we frame this problem?

- Rather than having a combination lock which has numbers, we will have a lock that consists of bits, either 0 or 1.
- We can assume this without loss of generality.
- As the number of bits increases, the number of possible combinations grows quickly. For n bits there are 2^n possible combinations.
- As you might imagine, solving this by brute-force is not tenable.
- If there are 80 bits then there are $2^{80} \simeq 10^{24}$ (about 1 septillion) combinations.
- Using an petascale supercomputer (10^{15} operations per second) this would take about 13,922 days to completely search.

Obviously this brute force approach is not worth considering.

A quantum approach

As we discussed last class, quantum computers use qubits.

- Because qubits can be in a superposition of quantum states $|0\rangle$ and $|1\rangle$ simultaneously, they may be in a better position to break the lock than classical approaches, where each bit is either 0 or 1.
- If we put each individual qubit into a uniform superposition:

$$|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

- then the total state of the system will be the tensor product of all n qubits:

$$\begin{aligned} |\psi\rangle &= |+\rangle \otimes |+\rangle \otimes \ldots \otimes |+\rangle \\ &= |+\rangle^{\otimes n} \\ &= \frac{1}{\sqrt{2^n}} \sum_{\mathrm{x}\in\{0,1\}^n} |\mathrm{x}\rangle \end{aligned}$$

A quantum approach, continued

We start off with our qubits in a uniform superposition.

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathrm{x} \in \{0,1\}^n} |\mathrm{x}\rangle$$

What does that look like in practice? To give a specific example, if $n = 3$, then

$$|\psi\rangle = \frac{1}{\sqrt{2^3}} \left(|000\rangle + |100\rangle + |010\rangle + |001\rangle + |110\rangle + |101\rangle + |011\rangle + |111\rangle\right)$$

Where the various states are the tensor products of the possible states of the constituent qubits ($|010\rangle = |0\rangle \otimes |1\rangle \otimes |0\rangle$), as we discussed last class.

The oracle

We need a means of determining if any given set of bits opens the lock.

- We will create a function to fill this role.
- The function will act like us, dialing the particular combination into the lock and seeing if it opens.
- If we assume that the combination that opens the lock is given by $|\mathrm{s}\rangle$, then a simple lock-testing function might be

$$f\left(\mathrm{x}\right) = \begin{cases} 1 & \mathrm{x} = \mathrm{s} \\ 0 & \text{otherwise} \end{cases}$$

- This function is known as the "oracle".
- Each use of the function is known as a "query".

The oracle, continued

It would be convenient if we could encode the oracle into a Unitary operator of some kind.

- This is usually formalized by raising the function's output to a phase.

$$\begin{aligned} U_f \left|x\right\rangle &= (-1)^{f(x)} \left|x\right\rangle \\ &= \begin{cases} -\left|x\right\rangle & x = s \\ \left|x\right\rangle & \text{otherwise} \end{cases} \end{aligned}$$

- Since this acts on a computational basis state and doesn't change the amplitude of the state, this is obviously Unitary.
- Note that this operator has no effect on the probability of measuring the correct state.
- The oracle can be equivalently written as

$$U_f = I - 2\left|s\right\rangle\left\langle s\right|$$

A quantum approach, continued

Ok, now what?

- Our initial quantum state is the uniform superposition of n qubits, $|\psi\rangle$.
- The answer, $|\mathrm{s}\rangle$, is in there somewhere.
- (0100111010, is a 10-bit example)
- But the probability of using the oracle and randomly picking the correct answer is $1/2^n$, which is the same as brute-force guessing.
- While it may appear that the superposition of n qubits can do an huge number of calculations at once, we actually only measure a randomly-sampled state of all the possible states.

So now what?

A different approach

The simplest quantum approach is no better than classical. Where does that leave us?

- One can come up with tricks that can be used to do moderate speedup from random guessing, such as testing in pairs, but there are better ways.
- What we would like is to figure out a way to measure the correct answer, $|s\rangle$, with high probability.
- To do this, we need to increase the probability amplitude of the solution state, relative to the other $2^n - 1$ possible states.
- This is known as "amplitude amplification".
- If the probability amplitude of $|s\rangle$ is large compared to the other possible states the probability of measuring it will be high.

This approach should be more effective at finding the solution.

Grover search (1996)

How can we increase the amplitude of the correct solution, so that we measure the correct answer with high probability?

- Let us consider all $N = 2^n$ possible states simultaneously.
- First, we start with $|\psi\rangle$, the uniform distribution of all possible states.
- Next, apply the oracle. This will flip the sign of the amplitude of the solution state, $|\mathrm{s}\rangle$.
- Flipping the sign alone isn't enough to help, however, since the amplitude of the state has not changed, we've only introduced a relative phase change.
- Recall that the probability of measuring a state goes like the magnitude of the amplitude squared, so relative phase changes don't help.
- What are we left with at this point?

$$\begin{aligned} U_f |\psi\rangle &= (I - 2 |\mathrm{s}\rangle \langle \mathrm{s}|) |\psi\rangle \\ &= |\psi\rangle - 2 \langle \mathrm{s}|\psi\rangle |\mathrm{s}\rangle \end{aligned}$$

which is the uniform superposition, with the correct answer flipped in sign.

Grover search, continued

If we examine the inner product of the solution and our new state, we find

$$\langle \mathrm{s}| U_f |\psi\rangle = \langle \mathrm{s}|\psi\rangle - 2\langle \mathrm{s}|\psi\rangle \langle \mathrm{s}|\mathrm{s}\rangle = -\langle \mathrm{s}|\psi\rangle$$

So we've flipped the sign of the correct answer, now what?

- Create a new operator, D, called the diffusion operator, defined as

  $$D = 2|\psi\rangle\langle\psi| - I$$

  where $|\psi\rangle$ is the usual uniform superposition of states.
- Note that
  - $D|\psi\rangle = |\psi\rangle$
  - $D|\phi\rangle = -|\phi\rangle$ if $|\phi\rangle$ is orthogonal to $|\psi\rangle$.
- Applying this operator to the above state has an interesting affect.

The diffusion operator spreads amplitude around the uniform state.

Grover search, continued more

$$U_f |\psi\rangle = |\psi\rangle - 2\langle s|\psi\rangle |s\rangle$$

Applying the Diffusion operator ($D = 2|\psi\rangle\langle\psi| - I$) the inner product with the solution now becomes:

$$\begin{aligned}
\langle s| DU_f |\psi\rangle &= \langle s| D\left(|\psi\rangle - 2\langle s|\psi\rangle |s\rangle\right) \\
&= \langle s| \left(2|\psi\rangle\langle\psi| - I\right)\left(|\psi\rangle - 2\langle s|\psi\rangle |s\rangle\right) \\
&= \langle s| \left(2|\psi\rangle\langle\psi| - I\right)|\psi\rangle - 2\langle s| \left(2|\psi\rangle\langle\psi| - I\right)\langle s|\psi\rangle |s\rangle \\
&= 2\langle s|\psi\rangle\langle\psi|\psi\rangle - \langle s|\psi\rangle - 4\langle s|\psi\rangle\langle\psi|s\rangle\langle s|\psi\rangle + 2\langle s|\psi\rangle\langle s|s\rangle \\
&= \langle s|\psi\rangle - 4\langle s|\psi\rangle^3 + 2\langle s|\psi\rangle \\
&= 3\langle s|\psi\rangle - 4\langle s|\psi\rangle^3
\end{aligned}$$

Since $\langle s|\psi\rangle = 1/\sqrt{2^n} < 1$, the projection of our working state onto the solution state has increased by applying the DU_f operator combination!

Grover search, continued even more

So where are we now?

- We've found an operator combination, DU_f, that performs amplitude amplification. Which is what we were after.
- This operator, $G = DU_f$, is known as the Grover operator.
- If we apply this operator many times to our state, the amplitude of the answer will grow, and the amplitude of all other states will shrink.
- If we then measure the final state, it will be probable that we will measure $|s\rangle$, the solution state.

Let's see how this algorithm looks in practice.

Grover search, example

How does this look in practice? Let's try out this algorithm by hand.

We will set the problem size to 3 bits.

We will choose the solution to be $|101\rangle$, which corresponds to an index of 5.

The first thing we need to do is build the oracle, U_f. This, you will recall, will just flip the sign of the solution, and leave all other states unchanged.

```
In [1]: import numpy as np
In [2]:
In [2]: nbits = 3
In [3]: sindex = 5
In [4]:
In [4]: Uf = np.identity(2**nbits)
In [5]: Uf[sindex, sindex] = -1
In [6]: Uf
Out[6]:
array([[ 1., 0., 0., 0., 0., 0., 0., 0.],
       [ 0., 1., 0., 0., 0., 0., 0., 0.],
       [ 0., 0., 1., 0., 0., 0., 0., 0.],
       [ 0., 0., 0., 1., 0., 0., 0., 0.],
       [ 0., 0., 0., 0., 1., 0., 0., 0.],
       [ 0., 0., 0., 0., 0., -1., 0., 0.],
       [ 0., 0., 0., 0., 0., 0., 1., 0.],
       [ 0., 0., 0., 0., 0., 0., 0., 1.]])
```

Grover search, example, continued

Now we need to build our diffusion operator $D = 2\left|\psi\right\rangle\left\langle\psi\right| - I$.

Recall that $\left\langle\psi\right| = \frac{1}{\sqrt{2^3}}\left[1, 1, 1, 1, 1, 1, 1, 1\right]$

There are 2^n elements in $\left|\psi\right\rangle$, where n is the number of bits.

$$\left|\psi\right\rangle\left\langle\psi\right| = \frac{1}{\sqrt{2^3}}\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}\frac{1}{\sqrt{2^3}}\begin{bmatrix}11111111\end{bmatrix} = \frac{1}{2^3}\begin{bmatrix}11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\\11111111\end{bmatrix}$$

Grover search, example, continued more

Now that we've got a matrix form for $|\psi\rangle\langle\psi|$, building the rest of D is straightforward.

$$D = 2\,|\psi\rangle\langle\psi| - I$$

Now that we have D, we can prepare our initial state, the uniform superposition state, $|\psi\rangle$.

```
In [7]:
In [7]: two_n = 2**nbits
In [8]:
In [8]: psipsi = np.ones((two_n, two_n)) / two_n
In [9]:
In [9]: D = 2 * psipsi - np.identity(two_n)
In [10]:
In [10]: psi = np.ones(two_n) / np.sqrt(two_n)
In [11]:
In [11]: psi
Out[11]:
array([0.35355339, 0.35355339, 0.35355339, 0.35355339,
       0.35355339, 0.35355339, 0.35355339, 0.35355339])
In [12]:
```

Grover search, example, continued some more

Now we apply the operations DU_f to $|\psi\rangle$. We'll do this in two steps.

Applying this operation amplifies the amplitude, as expected. Repeating the operation DU_f amplifies it even more.

```
In [12]:
In [12]: temp_psi = Uf.dot(psi)
In [13]:
In [13]: temp_psi
Out[13]:
array([0.35355339, 0.35355339, 0.35355339, 0.35355339,
       0.35355339, -0.35355339, 0.35355339, 0.35355339])
In [14]:
In [14]: psi2 = D.dot(temp_psi)
In [15]:
In [15]: psi2
Out[15]:
array([0.1767767, 0.1767767, 0.1767767, 0.1767767,
       0.1767767, 0.88388348, 0.1767767, 0.1767767])
In [16]:
```

Grover search, example, continued even more

But notice what happens if we apply the Grover operator too many times.

Clearly there's a limit to how many times we should apply the operator. We'll come back to this question in a few slides.

```
In [16]:
In [16]: psi3 = D.dot(Uf.dot(psi2))
In [17]:
In [17]: psi3
Out[17]:
array([-0.08838835, -0.08838835, -0.08838835, -0.08838835,
       -0.08838835, 0.97227182, -0.08838835, -0.08838835])
In [18]:
In [18]: psi4 = D.dot(Uf.dot(psi3))
In [19]:
In [19]: psi4
Out[19]:
array([-0.30935922, -0.30935922, -0.30935922, -0.30935922,
       -0.30935922, 0.57452426, -0.30935922, -0.30935922])
In [20]:
```

Encoding the oracle

This is all well and good, but how do we implement the oracle, U_f, as a circuit?

$$U_f \left|\mathrm{x}\right\rangle = \begin{cases} -\left|\mathrm{x}\right\rangle & \mathrm{x} = \mathrm{s} \\ \left|\mathrm{x}\right\rangle & \text{otherwise} \end{cases}$$

Let us consider storing the phase information generated by $U_f \left|\mathrm{s}\right\rangle$ in an "auxiliary" qubit. Consider a new operator, $C^{(\mathrm{s})}X$:

$$C^{(\mathrm{s})}X \left|\mathrm{x}, y\right\rangle = \begin{cases} \left|\mathrm{x}\right\rangle \otimes X \left|y\right\rangle & \mathrm{x} = \mathrm{s} \\ \left|\mathrm{x}, y\right\rangle & \text{otherwise} \end{cases}$$

where X is the Pauli X gate, and $\left|y\right\rangle$ is our auxiliary qubit. This operator is like our previous CNOT gate, but the control is activated by the value of s, and the NOT operation applies to the auxiliary qubit only.

Encoding the oracle, continued

What's the point? Well, if $|y\rangle = |-\rangle$, then we get this behaviour:

$$C^{(\mathrm{s})}X\,|\mathrm{x},-\rangle = \begin{cases} |\mathrm{x}\rangle \otimes X\,|-\rangle = -\,|\mathrm{x},-\rangle & \mathrm{x} = \mathrm{s} \\ \qquad\qquad\qquad |\mathrm{x},-\rangle & \text{otherwise} \end{cases}$$

Which is exactly the behaviour that we want for the oracle.

$$U_f\,|\mathrm{x}\rangle = \begin{cases} -\,|\mathrm{x}\rangle & \mathrm{x} = \mathrm{s} \\ |\mathrm{x}\rangle & \text{otherwise} \end{cases}$$

This lays out a means of building a circuit that can behave like the oracle.

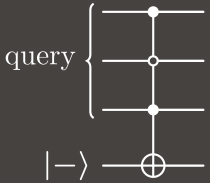
The oracle, as a circuit

To implement the oracle as a circuit, we use the $C^{(s)}X$ operator, using $|-\rangle$ as the auxiliary qubit.

The $C^{(s)}X$ operator is simply a multi-control-qubit CNOT gate. Rather than control on values of 1 only, as we did previously, here we control on a customized combination of ones and zeros. In this case, the control is activated for $|101\rangle$. Solid circles indicate a "1", hollow circles a "0".

The original qubits are known as the "query".

This technique is known as the "kickback trick".



Encoding the diffusion operator

Ok, now what about the diffusion operator, $D = 2\left|\psi\right\rangle\left\langle\psi\right| - I$? Recall that

- $D\left|\psi\right\rangle = \left|\psi\right\rangle$
- $D\left|\phi\right\rangle = -\left|\phi\right\rangle$, if $\langle\phi|\psi\rangle = 0$.

To reach this behaviour, we note a few things:

- $\left|\psi\right\rangle = H^{\otimes n}\left|0\right\rangle$
- $H^{\otimes n} \cdot H^{\otimes n} = I$
- $H^{\otimes n}\left|\psi\right\rangle = H^{\otimes n} \cdot H^{\otimes n}\left|0\right\rangle = \left|0\right\rangle$

As such, if we apply $H^{\otimes n}$ to the query qubits, and the result is $\left|0\right\rangle$, we know that we started with $\left|\psi\right\rangle$. If we then apply a $C^{(0)}X$ gate to our auxiliary qubit $\left|-\right\rangle$, as we did previously with the oracle, we will get a negative sign. We can then apply $H^{\otimes n}$ again to revert to the original state.

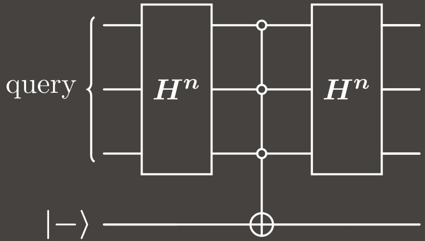
Encoding the diffusion operator, continued

Again, we use an auxiliary qubit to hold the phase information.

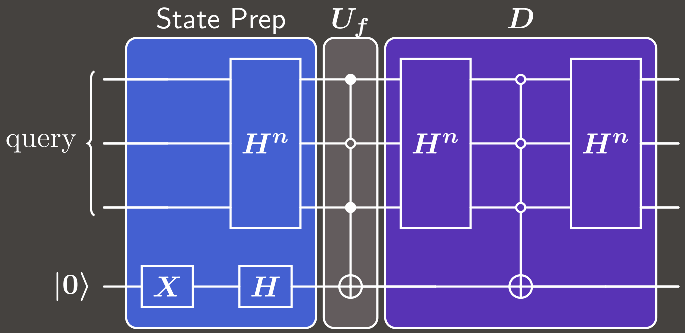
If we consider the operation $H^{\otimes n} C^{(0)} X H^{\otimes n}$ we see that .

$$H^{\otimes n} C^{(0)} X H^{\otimes n} |\mathrm{x}, -\rangle = \begin{cases} -|\mathrm{x}, -\rangle & |\mathrm{x}\rangle = |\psi\rangle \\ |\mathrm{x}, -\rangle & \text{otherwise} \end{cases}$$

Note that this is not D, but rather $-D$ that has been implemented.



Grover search, the full circuit



The green and red operations are repeated as many times as desired. Again, this particular implementation is for the case $|s\rangle = |101\rangle$.

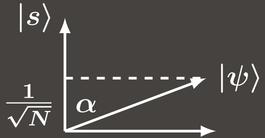
The number of iterations

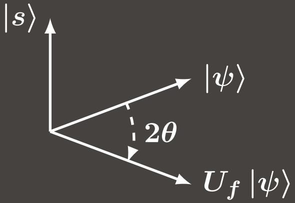
How many iterations of the Grover operator should we apply? If $|\psi\rangle$ is the uniform super-position of states, and $|s\rangle$ is our solution state, we then have that

$$\cos\alpha = \langle s|\psi\rangle = \frac{1}{\sqrt{N}}$$

where $N = 2^n$ is the total number of states. We know that the oracle U_f has the effect of reversing the sign of the component of the state parallel to $|s\rangle$.

If $\theta = \pi/2 - \alpha$, then this corresponds to a rotation of 2θ, clockwise.





The number of iterations, continued

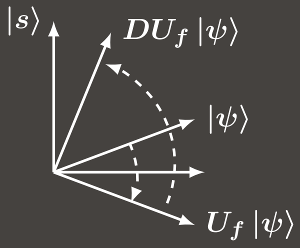
We now apply the diffusion operator. This has the effect of flipping the state around $|\psi\rangle$. Which results in a counter-clockwise rotation of 4θ.

Each application of G results in a net counter-clockwise rotations of 2θ. If we have that

$$\cos\alpha = \cos\left(\frac{\pi}{2} - \theta\right) = \sin\theta = \frac{1}{\sqrt{N}}$$

If N is large then $\theta \simeq \frac{1}{\sqrt{N}}$, and the total number of applications of G should be

$$S = \frac{\alpha}{2\theta} = \frac{\frac{\pi}{2} - \theta}{2\theta} = \frac{1}{2}\left[\frac{\pi}{2\theta} - 1\right] \simeq \frac{\pi\sqrt{N}}{4}$$



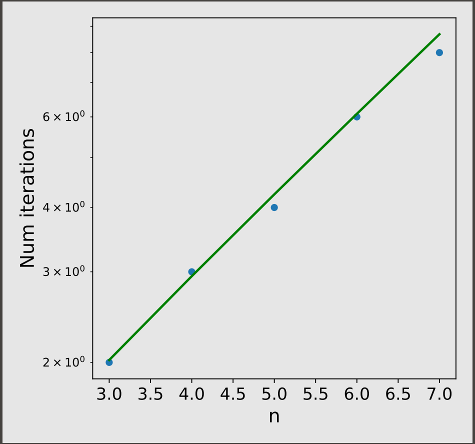
Thus, the number of calculations now goes like $\mathcal{O}(N^{\frac{1}{2}})$, rather than $\mathcal{O}(N)$.

The number of iterations, empirically

How many iterations of the Grover operator should we apply?

We can also calculate the scaling empirically. If you calculate the number of Grover steps for $n = \{3, 4, 5, 6, 7\}$ bits you'll get $S = \{2, 3, 4, 6, 8\}$. A simple fit indicates that the number of steps is scaling like $\sqrt{N}$.

For our 80-bit example, rather than 2^{80} times we need only run the Grover operator $\frac{\pi}{4}\sqrt{2^{80}} \simeq 2^{40}$ times.



But don't we need to know the answer?

An obvious question arises: don't we need to know the answer to search for the answer?

- The short answer is: yes. This obviously limits its usefulness.
- While we do need to know the answer to encode the answer in the circuit, Grover himself was not interested in this problem.
- Grover was interested in the polynomial reduction in query complexity, not implementability.
- This algorithm is more interesting from a theoretical point of view than a practical one.
- That being said, modern approaches to this problem have attempted to address this shortcoming, though they eventually run into the same limitation.

At the end of the day, you will need some means of feeding information into the circuit. You could imagine that to be a black box.

Summary

A summary of this section of the course.

- We can apply our quantum computer to the problem of searching for a lock combination.
- We first create an oracle operator, U_f, which flips the sign of the solution state, but leaves other states unchanged.
- We create a diffusion operator, D, which projects the state onto a component proportional to the uniform distribution, and a part orthogonal to it.
- Applying both of these operators on the uniform superposition results in amplitude amplification of the solution state, $|\mathrm{s}\rangle$.
- Unlike classical approaches which scale like $\mathcal{O}(N)$, Grover search scales like $\mathcal{O}(\sqrt{N})$, which is a significant improvement.

Grover search is a good example of the speedup one can see from using a quantum algorithm.

Grover search: hands on

Try this problem: write a function which implements Grover search, for n qubits, with the Grover operator applied k times, for some combination string ("010", for example):

- Initialize your qubits. Initializing the query qubits is done most easily with the PennyLane "broadcast" function:

  qml.broadcast(qml.Hadamard, wires = range(nbits), pattern = 'single')

- Build the oracle, which is most-easily implemented using the qml.MultiControlledX function. This function will implement an arbitrarily controlled PauliX operation on a given wire.
- Build the diffusion operator.
- Repeat the Grover operator steps k times.
- Return the probabilities associated with the query qubits.

Play with the number of qubits and the number of iterations. Confirm some of the numbers from slide 24.

Linky goodness

Grover search:

- https://arxiv.org/abs/quant-ph/9605043

Don't you already need to know the answer?

- https://quantumcomputing.stackexchange.com/questions/7138/in-grovers-algorithm-does-the-exact-solution-need-to-be-given-to-the-oracle
- https://quantumcomputing.stackexchange.com/questions/2110/whats-the-point-of-grovers-algorithm-if-we-have-to-search-the-list-of-elements
- https://www.quora.com/If-the-Oracle-from-Grovers-Algorithm-must-already-know-the-location-it-is-searching-for-why-bother-using-a-quantum-computer-to-fi