Notes on Quantum Computing

Aditya Morolia

Last Updated: November 5, 2020

Contents

1 Preface(?!)

Well this began when I asked a senior of mine for notes on basics of Quantum Computing, and then I started typing short notes for things I kept reading because I realised one pass through text was never enough as things were too damn connected and ideas too complicated. Now I keep coming back to the references, looking at stuff from before to understand things better. Of course I'm too lazy to go back through the old stuff and type it up more formally.

2 Introduction

Quantum Information is any physical information held in a quantum system. Quantum computation and quantum information is the study of the information processing tasks that can be accomplished using quantum mechanical systems. Quantum mechanics is a mathematical framework or set of rules for the construction of physical theories. For example, there is a physical theory known as quantum electrodynamics which describes with fantastic accuracy the interaction of atoms and light.

2.1 Postulates

Following are some of the axioms of Quantum Mechanics:

- Every system has an associated Hilbert Space, which is a vector space where inner product is defined.
- A state is a complete description of a system. It is the maximum information provided by nature so that future measurements can be performed. A state is a unit vector in Hilbert Space of the system.
- An observable is a property of a physical system that in principle can be measured. In QM these are self adjoint operators. The observable have a spectral decomposition and represented as $A = \sum a_n |n\rangle \langle n|$, where $|n\rangle$ is eigenbasis of A and a_n is the corresponding eigenvalue.

- A measurement is a process in which information about the state of the physical system is acquired by an observer by the use of an observable. In QM, the measurement of an observable A lead the system to collapse onto one of the observable's eigenstate. If the state prior to measurement is $|\psi\rangle$ then outcome a_n is obtained with probability $|\langle\psi|n\rangle|$ and the state collapses to $|n\rangle$.

- Composite Systems: If the Hilbert space for system A is H_A and that of system B is H_B, then the Hilbert space for the composite system AB is the tensor product $H_A \bigotimes H_B$. If system A is prepared in state $|\psi\rangle_A$ and system B is prepared in the state $|\psi\rangle_B$ then the composite system state is $|\psi\rangle_A \bigotimes |\psi\rangle_B$.

2.2 Qubit

Whereas the indivisible unit of classical information is the bit, which takes one of the two possible values 0, 1, the corresponding unit of quantum information is the quantum bit or qubit. The smallest non-trivial Hilbert space is two-dimensional. We may denote an orthonormal basis for a 2D vector space as $\{|0\rangle, |1\rangle\}$. Then the most general normalized state can be expressed as $a|0\rangle + b|1\rangle$ where a and b are complex numbers satisfying $|a|^2 + |b|^2 = 1$. $|a|$ is the probability of finding $|\psi\rangle$ in state $|0\rangle$ and likewise for $|1\rangle$, when measured in the computational basis.

2.3 Pauli Matrices

In 2D Hilbert space the Pauli operators are $\sigma_z,\ \sigma_x,\ \sigma_y\ and\ I$. They have following matrix representations:

$\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

$\sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$

Eigenvalues for all of them are 1 and -1. But eigenvectors are not the same. For σ_z the eigenvectors are $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ with eigenvalues 1 and -1 respectively. Similarly for σ_x the eigenvectors are denoted by $|+\rangle$ and $|-\rangle$ for eigenvalues 1 and -1 respectively, where $|+\rangle = \sqrt{\frac{1}{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \sqrt{\frac{1}{2}}(|0\rangle - |1\rangle)$.

2.4 No cloning theorem

A routine task performed in information processing is making copies of data. In classical scenario copying bits is quite an easy process. But the same is not true in quantum world. We can never make an exact copy of an arbitrary qubit. This is the No-Cloning theorem.

Consider two arbitrary states $|\psi\rangle$ and $|\phi\rangle$. Let us take another state $|B\rangle$ on which we want to copy $|\psi\rangle$ and $|\phi\rangle$. Suppose that there exists an arbitrary unitary operator U such that

$U\,|\phi\rangle \bigotimes |B\rangle = |\phi\rangle \bigotimes |\phi\rangle \ldots 1$

$U\,|\psi\rangle \bigotimes |B\rangle = |\psi\rangle \bigotimes |\psi\rangle \ldots 2$

Now, taking inner products of the left side of the above equations, we get

$\langle\psi| \bigotimes \langle B|\, U^{\dagger}U\, |\phi\rangle\, |B\rangle = \langle\psi|\phi\rangle$

Now for the right hand side, we get

$|\langle\psi|\phi\rangle|^2$

Equating these two results we have the following equation, $\langle\psi|\phi\rangle = |\langle\psi|\phi\rangle|^2$

Which is only true of $\langle\phi|\psi\rangle = 0$, i.e. they are orthogonal or $\langle\phi|\psi\rangle = I$ i.e. $|\phi\rangle = |\psi\rangle$. Hence, there is no unitary operator U that can be used to clone an arbitrary quantum states. If we can copy a state, then we can also distinguish the state easily.

2.5 Quantum Teleportation

Teleportation is the task of transferring an unknown qubit from one party to another using classical communication channel and a shared entangled pair as the only resources.

The task is that Alice wants to transmit an unknown state to Bob, (who is physically separated from her). Let the state be a qubit (normalised) to be teleported be $|X\rangle = a|0\rangle + b|1\rangle$.

Let them share an entangled pair which is $|\phi^+\rangle = (|0_A 0_B\rangle + |1_A 1_B\rangle)/\sqrt{2}$

Here the first member of the pair (kets subscripted with A) belongs to Alice while the second belongs to Bob. Alice and Bob have no interaction whatsoever, except the entangled pair. So as we notice that Alice now has two qubits $|X\rangle$ and one qubit from the entangled pair.

Now, let us write the combined state including the three particle, we name it as

$|\psi_{123}\rangle = (a|0\rangle + b|1\rangle) \bigotimes ((|0_A 0_B\rangle + |1_A 1_B\rangle)/\sqrt{2})$

$= (a|000\rangle + b|100\rangle + a|011\rangle + b|111\rangle)/\sqrt{2}$

Alice decides to make measurements on her two particles in the bell's basis which has eigenvectors. So we rearrange the state in the following manner:

$$|\psi_{123}\rangle = \frac{1}{2}(a(\left|\phi^+\right\rangle+\left|\phi^-\right\rangle)|0_B\rangle+b(\left|\phi^+\right\rangle-\left|\phi^-\right\rangle)|1_B\rangle+a(\left|\psi^+\right\rangle+\left|\psi^-\right\rangle)|1_B\rangle+b(\left|\psi^+\right\rangle-\left|\psi^-\right\rangle)|0_B\rangle$$

$$= \frac{1}{2}(\left|\phi^+\right\rangle(a|0\rangle + b|1\rangle) + \left|\phi^-\right\rangle(a|0\rangle - b|1\rangle) + \left|\psi^+\right\rangle(a|1\rangle + b|0\rangle) + \left|\psi^-\right\rangle(a|1\rangle - b|0\rangle))$$

So if after measurement Alice collapses on $|\phi^+\rangle$, Bob's state collapses on $a|0\rangle + b|1\rangle$ (which Alice is supposed to teleport). Similarly, for the other bell states as follows:

Alice	Bob
$\|\phi^-\rangle$	$a\|0\rangle - b\|1\rangle$
$\|\psi^+\rangle$	$a\|1\rangle + b\|0\rangle$
$\|\psi^-\rangle$	$a\|1\rangle - b\|0\rangle$

So if now Bob applies Pauli operators he will get the required qubit we are needed to teleport.

Before Operation	Operators	After Operation
$a\|0\rangle + b\|1\rangle$	I	$a\|0\rangle + b\|1\rangle$
$a\|0\rangle - b\|1\rangle$	σ_z	$a\|0\rangle + b\|1\rangle$
$a\|1\rangle + b\|0\rangle$	σ_x	$a\|0\rangle + b\|1\rangle$
$a\|1\rangle - b\|0\rangle$	$\sigma_z\sigma_x$	$a\|0\rangle + b\|1\rangle$

Alice informs Bob her measurement outcome via the classical communication channel using two classical bits (since four possibilities are there) and Bob performs the corresponding operator to obtain the unknown qubit. In this way teleportation is achieved. It must be noted that the protocol involves classical communication at the end which cannot be achieved faster than the light and hence it is concluded that Teleportation does not violate the No-signaling principle.

2.6 Quantum Dense Coding

Let us suppose that Alice needs to convey the results of two matches (A vs B and C vs D) (mutually exclusive events) to Bob using classical channel only. It can be shown easily that she would require at least two classical bits. Quantum Dense coding achieves the same task using 1 qubit. The resources used in this protocol include an entangled pair and a quantum channel.

Let them share an entangled pair, say $|\phi^+\rangle$. Now after the matches are over, Alice depending on the results she applies unitary operations on her qubit as follows:

Match Results	Operators	Entangled State after operation
A and C wins	I	$\|\phi^+\rangle$
A and D wins	σ_z	$\|\phi^-\rangle$
B and C wins	σ_x	$\|\psi^+\rangle$
B and D wins	$\sigma_z\sigma_x$	$\|\psi^-\rangle$

Now, after unitary transformation she sends her qubit to Bob via a quantum channel. The mapping from match results to the four states is pre-decided and known to both of them. Alice has transformed $|\phi^+\rangle$ to one of the four states, which are orthogonal to one another, so Bob performs a measurement in the bell basis and would be able to retrieve the results of both the matches using the mapping. It must be noted that had the states not been orthogonal they could not have been distinguished reliably (No Cloning Theorem).

The protocol is quite intriguing as it achieves the same task using one qubit instead of two classical bits and hence has the name Superdense Coding.

2.7 Qauntum Cryptography (overview)

Cryptography involves transferring information between sender and receiver through a public (unsecure) channel in the presence of eavesdropper. Various classical algorithms exist to make this task possible. However the loophole in all classical algorithms is that they rely on computational hardness of some problems. For example, RSA relies on the NP-hardness of Factoring problem. However, Quantum Key Distribution (QKD) does not rely on the computational hardness of any problem. Famous QKD protocols include BB84, B92, E91 and similar variants. It is to be noted that these protocols also take into consideration certain assumptions are not completely free of attacks but are better than classical key distribution protocols since they use inherent indeterminacy of Quantum Mechanics rather than the computational hardness.

BB84 protocol

In the first phase, Alice communicates to Bob over a quantum channel. Alice begins by choosing a random string of bits and for each bit, Alice randomly chooses a basis, rectilinear or diagonal, by which to encode the bit. She transmits a photon for each bit with the corresponding polarization, as just described, to Bob. For every photon Bob receives, he measures the photon's polarization by a randomly chosen basis. If, for a particular photon, Bob chose the same basis as Alice, then in principle, Bob should measure the same polarization and thus he can correctly infer the bit that Alice intended to send. If he chose the wrong basis, his result, and thus the bit he reads, will be random. In the second phase, Bob notifies Alice over any insecure channel what basis he used to measure each photon. Alice reports back to Bob whether he chose the correct basis for each photon. At this point Alice and Bob discard the bits corresponding to the photons which Bob measured with a different basis. Provided no errors occurred or no one manipulated the photons, Bob and Alice now both have an identical string of bits which is called a sifted key. Before they are finished however, Alice and Bob agree upon a random subset of the bits to compare to ensure consistency. If the bits agree, they are discarded and the remaining bits form the shared secret key. In the absence of noise or any other measurement error, a disagreement in any of the bits compared would indicate the presence of an eavesdropper on the quantum channel. This is because the eavesdropper, Eve, were attempting to determine the key, she would have no choice but to measure the photons sent by Alice before sending them on to Bob. This is true because the no cloning theorem assures that she cannot replicate a particle of unknown state . Since Eve will not know what bases Alice used to encoded the bit until after Alice and Bob discuss their measurements, Eve will be forced to guess. If she measures on the incorrect bases, the Heisenberg Uncertainty Principle ensures that the information encoded on the other bases is now lost. Thus when the photon reaches Bob, his measurement will now be random and he will read a bit incorrectly 50% of the time. Given that Eve will choose the measurement basis incorrectly on average 50% of the time, 25% of Bob's measured bits will differ from Alice. If Eve has eavesdropped on all the bits then after n bit comparisons by Alice and Bob, they will reduce the probability that Eve will go undetected to $\frac{3}{4}^n$. The chance that an eavesdropper learned the secret is thus negligible if a sufficiently long sequence of the bits are compared. One of the major attacks possible in the BB84 protocol is the photon number splitting attack. In practice implementations use laser pulses attenuated to a very low level to send the quantum states. If the pulse contains more than one photon, then Eve can split off the extra photons and transmit the remaining single photon to Bob. This is the basis of the photon number splitting attack. This is resolved in the Decoy State Quantum Key Distribution.

3 Quantum Foundation

[[TODO]]

4 Quantum Computation

Quantum Computation is the application of quantum mechanical properties of a system to processing tasks. This can be done by constructing quantum circuits, composed of a sequence of gates, quantum states (quantum registers), etc. A few examples of basic circuits using Qiskit have been given below.

4.1 Basics of Quantum Computing using Qiskit

Basic gates, Measurement, Teleportation and Adder

December 4, 2019

1 Qiskit practice

1.1 1. Basic Gates and stuff

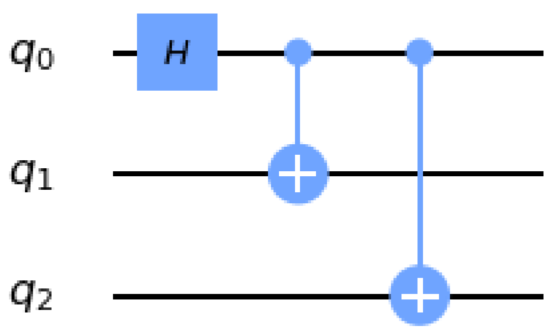
[1]:
```
import numpy as np
from qiskit import *

numQbits = 3
```

[16]:
```
circuit = QuantumCircuit(numQbits)
circuit.h(0)
circuit.cx(0, 1)
circuit.cx(0, 2)
```

[16]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7f60b744bac8>
```

[17]:
```
circuit.draw(output='mpl')
```

[17]:

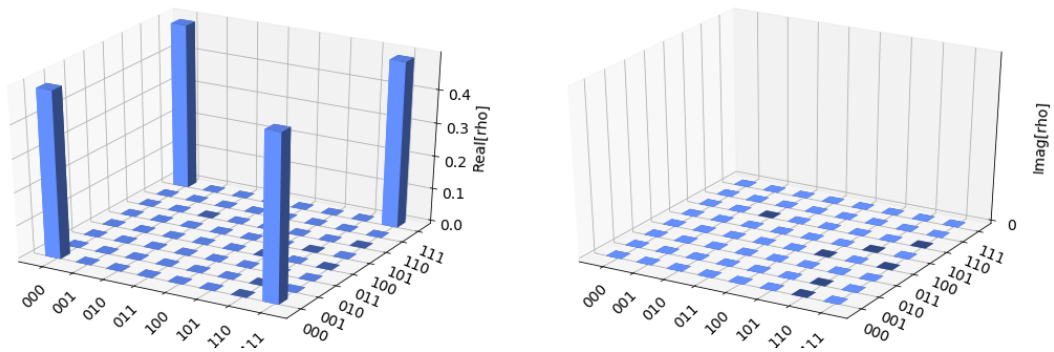


1.1.1 Statevector simulator

[18]:
```
backend = Aer.get_backend('statevector_simulator')
job = execute(circuit, backend)
result = job.result()
outputstate = result.get_statevector(circuit, decimals=3)
print(outputstate)
visualization.plot_state_city(outputstate)
```

```
[0.707+0.j 0.   +0.j 0.   +0.j 0.   +0.j 0.   +0.j 0.   +0.j 0.   +0.j
 0.707+0.j]
```

[18]:



1.1.2 Unitary Simulator

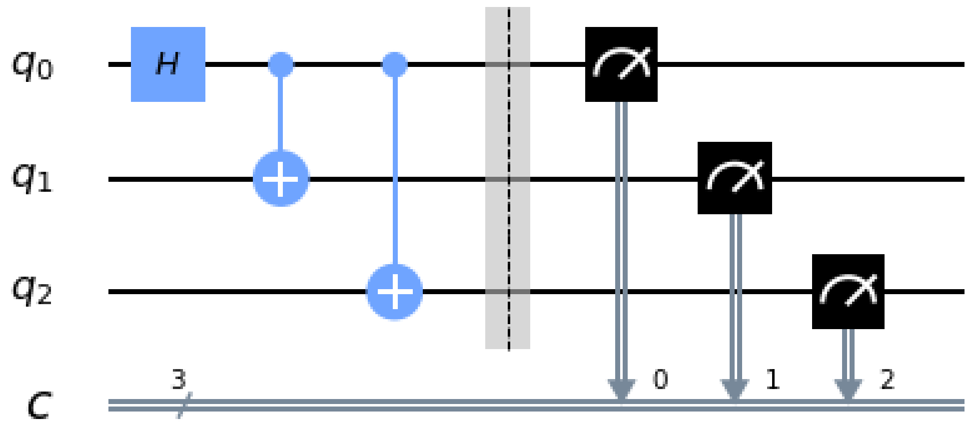
[19]:
```
backend = Aer.get_backend('unitary_simulator')
job = execute(circuit, backend)
result = job.result()
print(result.get_unitary(circuit, decimals=3))
```

```
[[ 0.707+0.j  0.707+0.j  0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j
   0.   +0.j  0.   +0.j]
 [ 0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j
   0.707+0.j -0.707+0.j]
 [ 0.   +0.j  0.   +0.j  0.707+0.j  0.707+0.j  0.   +0.j  0.   +0.j
   0.   +0.j  0.   +0.j]
 [ 0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j  0.707+0.j -0.707+0.j
   0.   +0.j  0.   +0.j]
 [ 0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j  0.707+0.j  0.707+0.j
   0.   +0.j  0.   +0.j]
 [ 0.   +0.j  0.   +0.j  0.707+0.j -0.707+0.j  0.   +0.j  0.   +0.j
   0.   +0.j  0.   +0.j]
 [ 0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j
   0.707+0.j  0.707+0.j]
 [ 0.707+0.j -0.707+0.j  0.   +0.j  0.   +0.j  0.   +0.j  0.   +0.j
   0.   +0.j  0.   +0.j]]
```

1.1.3 Measurement

[24]:
```
meas = QuantumCircuit(3, 3)
meas.barrier()
meas.measure(range(3),range(3))
qc = circuit+meas
qc.draw(output='mpl')
```

[24]:



[9]:
```
backend_sim = Aer.get_backend('qasm_simulator')
job_sim = execute(qc, backend_sim, shots=1024)
result_sim = job_sim.result()
counts = result_sim.get_counts(qc)
print(counts)
```

```
{'000': 507, '111': 517}
```

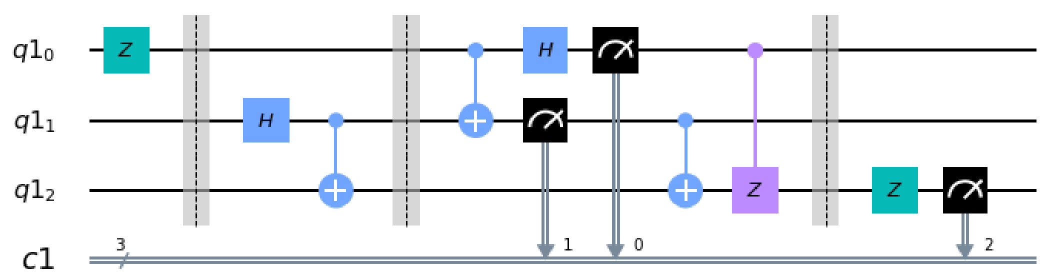
1.2 2. Teleportation

1. Resource Generation and circuit

[10]:
```
def apply_secret_unitary(secret_unitary, qubit, quantum_circuit, dagger):
    functionmap = {
                    'x':quantum_circuit.x,
                    'y':quantum_circuit.y,
                    'z':quantum_circuit.z,
                    'h':quantum_circuit.h,
                    't':quantum_circuit.t,
    }
    if dagger:
        functionmap['t'] = quantum_circuit.tdg
        [functionmap[unitary](qubit) for unitary in secret_unitary]
```

```
    else:
        [functionmap[unitary](qubit) for unitary in secret_unitary[::-1]]

qr = QuantumRegister(3) # qr1: To be teleported. qr2: First entangled qubit.␣
 ↪qr3: Second entangled qubit.
cr = ClassicalRegister(3)
qc = QuantumCircuit(qr, cr)
secret_unitary = 'z'

apply_secret_unitary(secret_unitary, qr[0], qc, dagger = 0)
qc.barrier()
qc.h(qr[1])
qc.cx(qr[1], qr[2])
qc.barrier()
# Resource generation done, one entangled pair shared between Alice and Bob.
qc.cx(qr[0], qr[1])
qc.h(qr[0])
qc.measure(qr[0], cr[0])
qc.measure(qr[1], cr[1])
qc.cx(qr[1], qr[2])
qc.cz(qr[0], qr[2])
qc.barrier()
apply_secret_unitary(secret_unitary, qr[2], qc, dagger=1)
qc.measure(qr[2], cr[2])
qc.draw(output='mpl')
```

[10]:

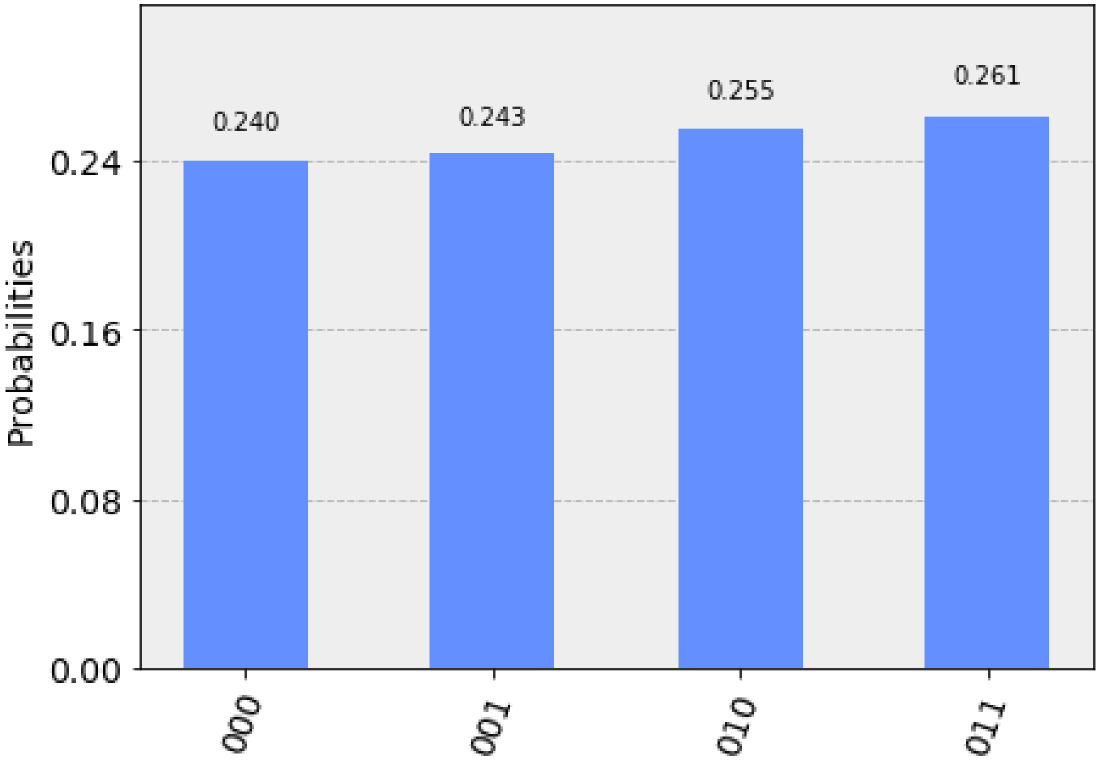


2. Simulating and testing

[12]:
```
backend = Aer.get_backend('qasm_simulator')
job_sim = execute(qc, backend, shots=2048)
sim_result = job_sim.result()
# print(sim_result)
measurement_result = sim_result.get_counts(qc)
print(measurement_result)
visualization.plot_histogram(measurement_result)
```

{'011': 535, '010': 523, '000': 492, '001': 498}

[12]:



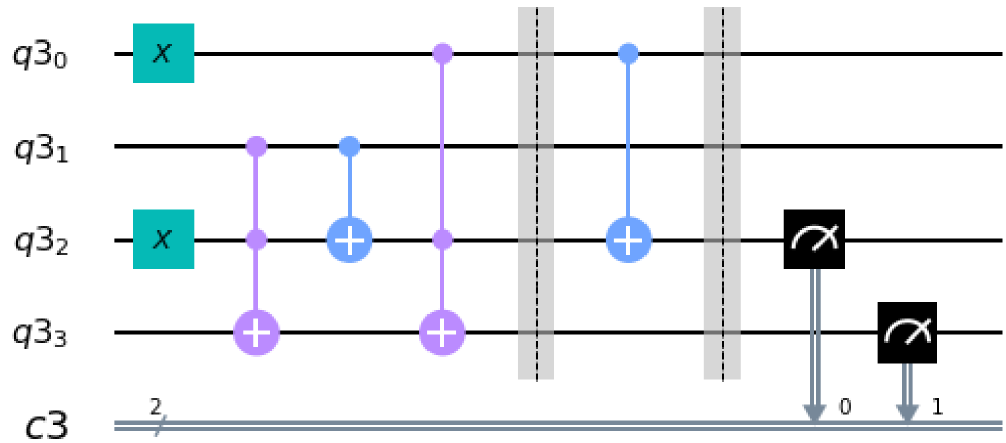
1.3 3. 1-Bit Adder Circuit

[27]:

```
numQBits = 4 #1 == A, 2 == B, 3 == C(i+1), output in B
qr = QuantumRegister(numQBits)
cr = ClassicalRegister(2)
circuit = QuantumCircuit(qr, cr)

# Initialization
'''Initial values of the registers are 0. Applying cx will flip the registers.
Hence, by applying cx as required, we can initialize the circuit to a desired␣
↪input'''
circuit.x(qr[0])  # 0 == carry
# circuit.x(qr[1])  # 1 == A
circuit.x(qr[2])  # 2 == B

# Carry
circuit.ccx(qr[1], qr[2], qr[3])
circuit.cx(qr[1], qr[2])
circuit.ccx(qr[0], qr[2], qr[3])
circuit.barrier()
```

```
# Sum
circuit.cx(qr[0], qr[2])
circuit.barrier()

circuit.measure(qr[2], cr[0]) # Output
circuit.measure(qr[3], cr[1]) # Carry

circuit.draw(output='mpl')
```

[27]:



[28]:
```
backend = Aer.get_backend('qasm_simulator')
job_sim = execute(circuit, backend, shots=1)
sim_result = job_sim.result()
measurement_result = sim_result.get_counts(circuit)
print(list(measurement_result.keys())[0])
# print(measurement_result)
```

10

1.4 4. n-Bit Adder

[76]:
```
def oneBitAdder(A, B, Cin):
    qr = QuantumRegister(4)
    cr = ClassicalRegister(2)
    circuit1 = QuantumCircuit(qr, cr)
#     circuit1.initialize([1, 0], qr[0])
#     circuit1.initialize([1, 0], qr[1])
#     circuit1.initialize([1, 0], qr[2])
#     circuit1.initialize([1, 0], qr[3])
```

```
    if int(A):
        circuit1.x(qr[1])
    if int(B):
        circuit1.x(qr[2])
    if int(Cin):
        circuit1.x(qr[0])
    # Carry
    circuit1.ccx(qr[1], qr[2], qr[3])
    circuit1.cx(qr[1], qr[2])
    circuit1.ccx(qr[0], qr[2], qr[3])
    circuit1.barrier()

    # Sum
    circuit1.cx(qr[0], qr[2])
    circuit1.barrier()

    circuit1.measure(qr[2], cr[0]) # Output
    circuit1.measure(qr[3], cr[1]) # Carry
#     print(circuit1)

    backend = Aer.get_backend('qasm_simulator')
    job_sim = execute(circuit1, backend, shots=1)
    sim_result = job_sim.result()
    measurement_result = sim_result.get_counts(circuit1)
    ret = list(measurement_result.keys())[0]
    return ret[1], ret[0]

a = '11011'
b = '11001'
cin = 0
idx = len(a) - 1
sum_ = ''
while idx >= 0:
#     print(a[idx], b[idx], cin)
    s, cin = oneBitAdder(a[idx], b[idx], cin)
#     print(s, cin)
    sum_ += s
    idx -= 1
sum_ += cin
print(sum_[::-1])
```

110100

[ ]:

```

```

5 Quantum Algorithms

A quantum algorithm is an algorithm which runs on a realistic model of quantum computation, the most commonly used model being the quantum circuit model of computation (as depicted earlier.) A classical algorithm is a finite sequence of instructions, or a step-by-step procedure for solving a problem, where each step or instruction can be performed on a classical computer. Similarly, a quantum algorithm is a step-by-step procedure for solving a problem (which can be either classical or quantum), where each of the steps can be performed on a quantum computer. Although all classical algorithms can also be performed on a quantum computer, the term quantum algorithm is usually used for those algorithms which seem inherently quantum, or use some essential feature of quantum computation such as quantum superposition or quantum entanglement. Problems which are undecidable using classical computers remain undecidable using quantum computers.

Two concepts need to be understood and differentiated here:

Quantum supremacy is the goal of demonstrating that a programmable quantum device can solve a problem that classical computers *practically* cannot.

Quantum advantage (a weaker goal) is the demonstration that a quantum device can solve a problem merely faster than classical computers.

Following are some examples of quantum algorithms:

- Deutsch-Jozsa algorithm
- Bernstein-Vazirani algorithm

- Simon's algorithm
- Quantum Phase estimation algorithm
- Grover's Search
- Shor's algorithm
- HHL algorithm

Here, I will explain the Deutsch-Jozsa algorithm, and a few in the next section of Quantum Machine Learning.

5.1 Deutsch-Jozsa algorithm

Problem Statement: Given is an oracle that implements some function $f : \{0,1\}^n \rightarrow \{0,1\}$, and that the input function is either constant or balanced, determine if the function is constant or balanced.
This problem was specifically designed to show separability between the complexity classes of EQP and P.

Algorithm:

The circuit begins with n+1 qubit state
$|\psi_0\rangle = |0\rangle^n \bigotimes |1\rangle$.
Next, Hadamard gate is applied to get
$|\psi_1\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle \left(|0\rangle - |1\rangle\right)$.

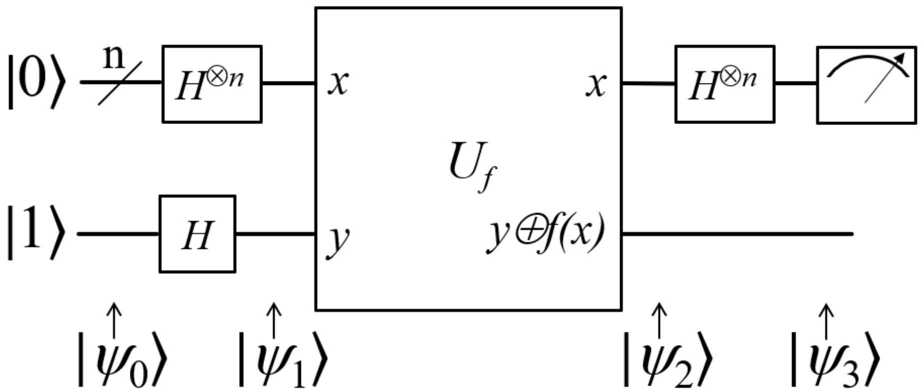


Figure 5.1: Quantum Circuit for Deutsch-Jozsa algorithm

The oracle U_f maps $|x\rangle\,|y\rangle$ to $|x\rangle\,|y \bigoplus f(x)\rangle$. Applying this, we get

$$|\psi_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle\,(|f(x)\rangle - |1 \bigoplus f(x)\rangle)$$

Since $f(x) \in \{0, 1\}$, we get

$$|\psi_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)}\,|x\rangle\,(|0\rangle - |1\rangle)$$

Now, ignore the last qubit. Applying Hadamard gate again, we get

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left[\sum_{y=0}^{2^n-1} (-1)^{x.y}\,|y\rangle \right]$$

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left[\sum_{x=0}^{2^n-1} (-1)^{f(x)+(x.y)} \right] |y\rangle$$

where $x.y = x_0y_0 \bigoplus x_1y_1 \bigoplus \cdots x_{n-1}y_{n-1}$

Then, we measure the probability of getting $|0\rangle^{\bigotimes n}$, which is $\left|\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\right|^2$ and gives 1 if $f(x)$ is constant and 0 if it is balanced.

6 Quantum Machine Learning

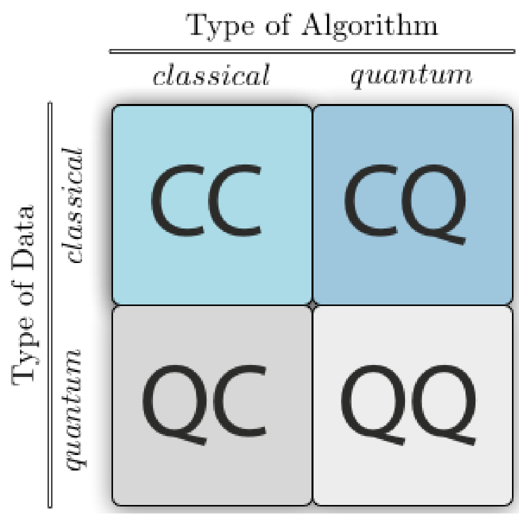


Figure 6.1: Different approaches to Quantum Machine Learning

Quantum machine learning is an emerging interdisciplinary research area at the intersection of quantum computing and machine learning. One such application is machine learning algorithms for the analysis of classical data executed on a quantum computer (quantum-enhanced machine learning). One prominent method in the NISQ (Noisy intermediate scale quantum) age are hybrid methods that involve both classical and quantum processing, where computationally difficult subroutines are outsourced to a quantum device. These routines can be more complex in nature and executed faster with the assistance of quantum devices. Additionally, quantum algorithms can be used to analyze quantum states instead of classical data. Beyond quantum computing, the term "quantum machine learning" is often associated with classical machine learning methods applied to data generated from quantum experiments (i.e. machine learning of quantum systems), such as learning quantum phase transitions or creating new quantum experiments. Quantum machine learning also extends to a branch of research that explores methodological and structural similarities between certain physical systems and learning systems, in particular neural networks (i.e., the relationship between quantum systems and neural networks.) Here, I will briefly explain the use of Parameterized quantum circuits as machine learning models and Quantum Circuit Learning.

Learning is the process of iteratively updating the set of parameters in the model and selecting the one attaining low loss.

6.1 Parameterized quantum circuits as machine learning models

This was proposed in a recent paper (June 2019) published with the same name by Marcello Benedetti, Erika Lloyd, and Stefan Sack. Here I will present a brief overview of the paper.

6.1.1 Introduction

Numerous advancements in quantum hardware and the theoretical foundations are being made continuously, but a fully functioning quantum computer is still far away. Meanwhile, it is being argued that noisy intermediate scale quantum (NISQ) devices may be useful commercially, as well as scientifically. Parameterized quantum circuits provide a way to demonstrate quantum supremacy in the NISQ era. The intuition here is that by outsourcing parts of the algorithm to classical hardware, we significantly reduce the burden on the quantum hardware. In particular, we reduce the required coherence time, circuit depth and number of qubits, hence allowing NISQ hardware to focus entirely on the computationally hard part of the problem. This hybrid algorithmic approach turned out to be successful in attacking scaled-down problems in chemistry, combinatorial optimization and machine learning.

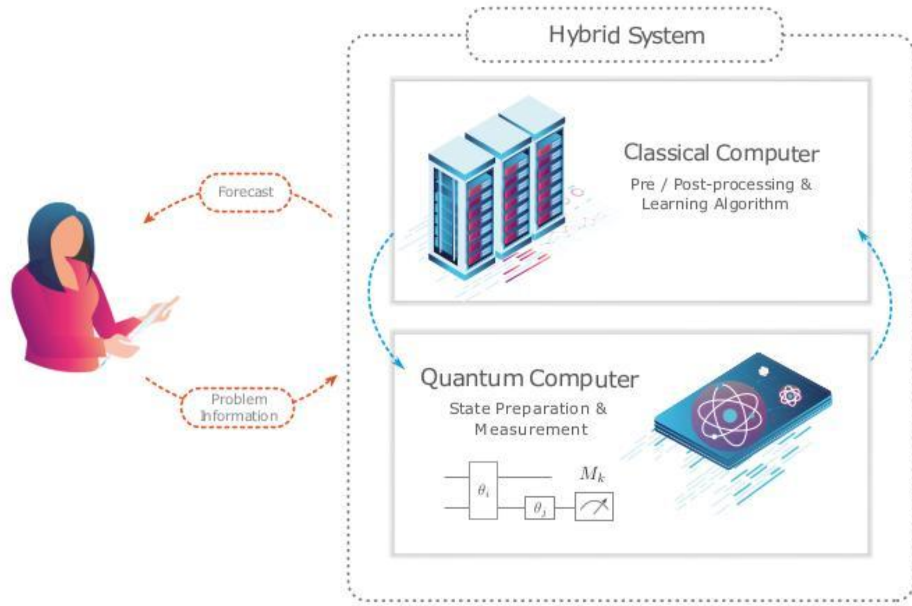


FIG. 1. High-level depiction of hybrid algorithms used for machine learning. The role of the human is to use the problem information to setup the model, assess the learning process, and use the forecasts. Within the hybrid system the quantum computer prepares quantum states according to a set of parameters. Using the measurement outcomes, the classical learning algorithm adjusts the set of parameters in order to minimize an objective function. The parameters, now defining a new quantum circuit, are fed back to the quantum hardware in a closed loop.

Figure 6.2: Hybrid Quantum Algorithms (Source: arXiv:1906.07682v)

The framework involves five major steps:

1. Data vector is sampled from the training set and transferred by classical preprocessing (e.g. decorrelation or standardization functions)

2. The transformed data point is mapped to the parameters of an encoder circuit $U_{\phi(x)}$.

3. A variational circuit U_θ implements the core operation of the model.

4. This is followed by the estimation of a set of expectation values $\{\langle M_k \rangle_{x,\theta}\}_{k=1}^{K}$

5. Then a post processing function f is applied to find the output values.

Here, steps 1 and 5 are performed using a classical computer, and the steps 2, 3 and 4 are performed on a quantum processor.

7 Exploratory readings

This is mostly a dump for things I keep coming back to, stuff that I'm currently too noob to understand yet.

7.1 MIP*=RE

Okay I can't do this. Should probably give up quantum computing. The reason I picked this up to read was not that I wanted to understand the result, but because I thought the exercise would lead me to a lot of older work on complexity theory, which is something that I am interested in. I did not read the whole paper, but only parts of it that I could understand. At first sight, I was totally lost and understood practically nothing. But I read a lot of pre-requisites and got back to the result, as well as the author's blog post. I plan on getting back to this after finishing Aaronson's notes.

References:

- https://www.scottaaronson.com/blog/?p=4512
- Original Paper: https://arxiv.org/pdf/2001.04383.pdf
- https://quantumfrontiers.com/2020/03/01/the-shape-of-mip-re/

7.2 Google Supremacy Result

Reading about the google supremacy result let me to reading more about practical challenges of quantum computing, noise, benchmarking quantum computers, as well as theoretical results on the subject. I also read about Gil Kalai's argument against quantum computing. In the end, it was a fun exercise, and introduced me to a lot of things currently happening in the field, but I did not get into any of these very deeply.

References:

- https://ai.googleblog.com/2019/10/quantum-supremacy-using-programmable.html
- https://www.scottaaronson.com/blog/?p=4317
- https://www.scottaaronson.com/blog/?p=4372

7.3 Quantum Game Theory

Quantum game theory was relatively easy to get into, since I knew a lot of prerequisites already. It basically deals with using quantum resource in the games defined by game theoretic definitions. It can be superposition or entanglement between states of the players, or superposition of strategies used by the players. This is definitely something I want to explore further in the future.

7.4 Solovay-Kitaev Theorem

Let $\mathcal{G}$ be a finite set of elements in $SU(2)$ containing its own inverses (so $g \in \mathcal{G} \implies g^{-1} \in \mathcal{G}$). Consider some $\varepsilon > 0$. Then there is a constant c such that for any $U \in SU(2)$, there is a sequence S of gates from $\mathcal{G}$ of length $O(\log^c(1/\varepsilon))$ such that $\|S - U\| \leq \varepsilon$. That is, S approximates U to operator norm error.

References:

- https://en.wikipedia.org/wiki/Solovay
- https://arxiv.org/pdf/quant-ph/0505030.pdf
- http://home.lu.lv/ sd20008/papers/essays/Solovay-Kitaev.pdf
- A Simple Proof that Toffoli and Hadamard are Quantum Universal

7.5 Spectral Theorem

Theorem. (Gelfand & Maurin)

Given an arbitrary set of commuting self-adjoint operators defined on the same dense subspace of a Hilbert space, there is always an isomorphic Hilbert space in which these operators are represented by multiplication with real-valued functions.

8 Quantum Machine Learning (MOOC)

8.1 Basics

- Quantum theory as a generalisation of classical probability theory
- Classical and Transverse field ising model
- Qubits:

  Logical and Physical. 1 logical qubit $\approx$ 100 to 1000 physical qubits
- Many body systeams:

  Imagine a classical ising model representing electron spins. Consider the problem of finding the lowest energy state. If number of particles N = 40, search space is $2^{40} \approx 7 * 10^{22} \approx$ total bits stored in all of world's computer. If N = 268, search space is $2^{268} \approx 10^{80} \approx$ number of particles in the known universe. Storing all states physically impossible.
- Strategies to solve many-body problems:
  - Analytical: Perturbation theory, neglecting interactions, etc. (approximation)
  - Numerical Approaches (approximation)
  - Truncation/compression: Reduce the size of hilbert space. e.g.: Density matrix re-normalisation group
  - Stochastic (MCMC)

8.2 Quantum Computing models

8.2.1 Gate model of quantum computing.

Software stack for quantum computers: Problem definition (Travelling salesman) →Quantum Algorithm (QAOA) →Quantum Circuit (Gates and unitary operators) →Quantum Compiler (Gate translation, if implemented gate not in hardware and Connectivity or interactions) →Simulation or running on quantum processor.

Solovay-Kitaev theorem: Finite set of gates can approximate and unitary operation (efficiently). Hence, gate model of quantum computing is Universal.

8.2.2 Adiabatic Quantum Computing:

Adiabatic theorem: A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.

$$H_0 = \sum_i \sigma_i^x$$

$$H_1 = -\sum_{<i,j>} j_{ij}\sigma_i^z\sigma_j^z - \sum_i h_i\sigma_i^z$$

$$H(t) = (1-t)H_0 + tH_1; t \in [0,1]$$

Slow change $\sim$ Adiabatic pathway

Speed Limit: $\frac{1}{min(\Delta(t))^2}$

Adiabatic Quantum Computing Hamiltonian:

$$H = -\sum_{<i,j>} j_{ij}\sigma_i^z\sigma_j^z - \sum_i h_i\sigma_i^z - \sum_{<i,j>} g_{ij}\sigma_i^x\sigma_j^x$$

This is also universal.