The Deutsch-Jozsa Algorithm

Carter M. Gustin

May 2021

1 Important Quantum Gates

1. Exchange (X) Gate:

$$\hat{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

This matrix exchanges (flips) the qubits i.e. $\hat{X}\left|0\right\rangle = \left|1\right\rangle, \hat{X}\left|1\right\rangle = \left|0\right\rangle$

2. Hadamard Gate (H):

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

This matrix turns qubits from the computational basis, $\{\left|0\right\rangle, \left|1\right\rangle\}$ into linear combinations i.e. $\hat{H}\left|0\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle)$, $\hat{H}\left|1\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$. We can write this in a compact form:

$$\hat{H}\left|x\right\rangle = \frac{1}{\sqrt{2}}(\left|0\right\rangle + (-1)^x \left|1\right\rangle)$$

3. The Z Gate (Z):

$$\hat{Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

This matrix is the standard Pauli-Z gate which can be written in a compact form:

$$Z\left|x\right\rangle = (-1)^x \left|x\right\rangle$$

4. Controlled Gates:

Controlled gates are a two-qubit gate. One qubit serves as the *control* qubit, while the other is the *target* qubit. Essentially, the gate applies a unitary operator to the target provided that the control qubit is in the state $\left|1\right\rangle$. The general form for controlled gates are:

$$cU = \left|0\right\rangle\left\langle 0\right| \otimes \mathbf{1} + \left|1\right\rangle\left\langle 1\right| \otimes U$$

- Controlled NOT Gate: $CNOT\,|a,b\rangle = |a, b \oplus a\rangle$. This gate is analogous to the XOR gate of two bits in classical computations.
- Controlled Z Gate: $cZ\,|a,b\rangle = (-1)^{a \cdot b}\,|a,b\rangle$

2 Quantum Algorithms

2.1 Quantum Programs

A quantum computer's program can be described by a quantum circuit (combinations of the quantum gates above.) We assume that all of the input qubits are in the basis state $|0\rangle$ which give the initial state $|0^n\rangle$ and the output (after the unitary evolution) is measured in the standard $\{|0\rangle, |1\rangle\}$ basis.

Now, assume f is a mathematical function which takes a string a of k qubits as input, calculates a single qubit as output, $f(a)$ and uses this qubit as the control qubit. These gates are called *f-controlled gates* where the control qubit is $f(a)$ (f operates on the input string writes the output $f(a)$ on the last qubit).

For example, the *f-controlled NOT* gate writes $f(a)$ on the last input qubit and uses this as the control qubit on for the target qubit i.e.

$$U_f\,|a,b\rangle = |a, b \oplus f(a)\rangle$$

Another $f - controlled$ gate is the $f - controlled - Z$ gate which can be summarized by

$$fcZ\,|a,b\rangle = (-1)^{f(a)}\,|a,b\rangle$$

using the rules of the cZ gate above.

2.2 Quantum Oracles

If we want to understand what the function f does on our input string (is it constant and outputs a string of all 0's or 1's or is it balanced and outputs a string of an equal number of 0's or 1's?), we must send a string in and evaluate what the outcome is. The subroutine that calculates f is an oracle for the function f. The problem of determining how f behaves is known in quantum computing as the oracle problem. Classically, if we have an oracle, we could learn after two evaluations if f is balanced or constant (if $f(0) \neq f(1)$ then f must be balanced). However, we may need to evaluate the function 2^k times for k input bits. If k is very large, this will take a very long time to run. The Deutsch-Jozsa Algorithm shows that a quantum computer can solve the oracle problem with only *one* coherent evaluation of f.

2.3 An Important Result

Suppose k qubits start out in the basis state $|a\rangle$, where a is a k-bit string. We apply the Hadamard gate $\hat{H}$ to each of the qubits. We know for one qubit, $\hat{H}|x\rangle = \frac{1}{\sqrt{1}}(|0\rangle + (-1)^x |1\rangle)$ which we can write even more compactly as:

$$\hat{H}|x\rangle_1 = \frac{1}{\sqrt{2}} \sum_{k=0}^{1} (-1)^{k*x} |k\rangle_1$$

If we apply the Hadamard gate to a string $|a\rangle$, we have to take a tensor product for each qubit:

$$(\hat{H}^{(1)} \otimes \hat{H}^{(2)} \otimes \cdots \otimes \hat{H}^{(k)})|a\rangle = \hat{H}^{\otimes k}|a\rangle$$
$$= \frac{1}{\sqrt{2^k}} \sum_{c_1=0}^{1} \sum_{c_2=0}^{1} \cdots \sum_{c_k=0}^{1} (-1)^{c_1*a_1+\ldots c_k*a_k} |c_1\rangle_1 \otimes \cdots \otimes |c_k\rangle_k$$

which can be simplified to:

$$\hat{H}^{\otimes k}|a\rangle = \frac{1}{2^{k/2}} \sum_{c} (-1)^{a\cdot c} |c\rangle$$

where $a \cdot c = a_1c_1 + \ldots a_kc_k$ and $|c\rangle$ is the output string of qubits.

2.4 The Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm shows that given an input qubit $|0^k, 0\rangle$, the quantum oracle can be determined with only one coherent evaluation of the function f. The Deutsch-Jozsa circuit that the algorithm uses is as follows:

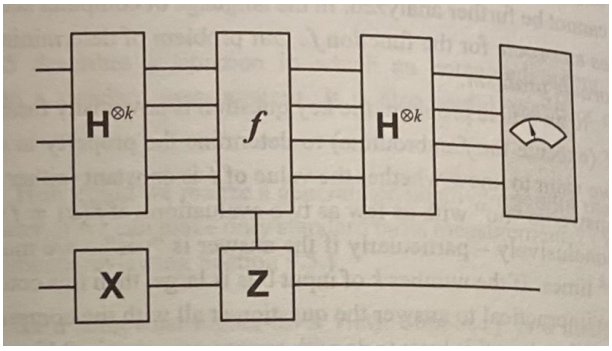

Figure 1: The Deutsch-Jozsa Circuit

The algorithm is as follows:

- The input state is prepared by applying a Hadamard gate ($\hat{H}$) to each of the k input qubits (the string of qubits) and the $\hat{X}$ gate is applied to the bottom qubit to convert it into $|1\rangle$:

$$(\hat{H}^{\otimes k} \otimes \hat{X}^{(k+1)}) \left|0^k, 0\right\rangle = \frac{1}{2^{k/2}} \sum_a |a, 1\rangle$$

- The function f is evaluated once using the f-controlled Z (fcZ) gate:

$$(fcZ)\frac{1}{2^{k/2}} \sum_a |a, 1\rangle = \frac{1}{2^{k/2}} \sum_a (-1)^{f(a)} |a, 1\rangle$$

  by the rules above for the $f - cZ$ gate.

- We once more perform the Hadamard gate on all of the k input qubits by invoking the result above for operating $\hat{H}^{\otimes k} |a\rangle$:

$$\begin{aligned}(\hat{H}^{\otimes k})\frac{1}{2^{k/2}} \sum_a (-1)^{f(a)} |a, 1\rangle &= \frac{1}{2^k} \sum_a (-1)^{f(a)} \left(\sum_c (-1)^{a \cdot c | c, 1\rangle} \right) \\ &= \sum_c \left(\frac{1}{2^k} \sum_a (-1)^{a \cdot c + f(a)} \right) |c, 1\rangle\end{aligned}$$

- Finally, measure the k input qubits in the standard computational basis $\{|0\rangle, |1\rangle\}$ and determine an experimental value of the bit string c.

Now, we must determine what the probability is that the final measurement of c is 0^k. We calculate this by standard probability rules:

$$P(0^k) = \left| \frac{1}{2^k} \sum_a (-1)^{f(a)} \right|$$

If the function f is constant, the 2^k terms in the probability amplitude are all of the same sign, and $P(0^k) = 1$. On the other hand, if the function is balanced, then the terms in the probability amplitude exactly cancel out, so that $P(0^k) = 0$. The measurement of c either yields $00 \ldots 0$ or something else which determines, with certainty, whether f is constant or balanced.

3 Sources

- *Quantum Processes, Systems, and Information*
- https://researcher.watson.ibm.com/researcher/files/us-nannicini/qc_lecture_2.pdf
- https://arxiv.org/pdf/1110.2998.pdf