Error Correction - Lecture 1

with Dan Browne

March 26, 2016

Contents

1 Introduction

Error correction is at the core of realistic quantum computing. If you think of quantum computation, you should also think of error correction. Any vision of quantum computing without error correction is fundamentally flawed. This course will introduce all the various notions of quantum error correction that are common in the field.

2 Classical Repetition Code

In this section, we will first have a look at classical repetition code in order to understand how classical error correction is done.

For the classical bit x, which can take on values $x \in [0, 1]$, there are two possible errors. Either the bit gets flipped, such that $0 \to 1$ or $1 \to 0$, or the bit is lost.

We can decompose most codes in terms of these errors. To protect from these errors, we **encode** our bits into several bits, such that

$$0 \to 000 \tag{1}$$

$$1 \to 111 \tag{2}$$

Then, an error on the 2nd bit would cause

$$000 \to 010 \tag{3}$$

$$111 \to 101 \tag{4}$$

If we query the values of the bits, we can take a majority vote and then correct the bit accordingly. This would work for the error described above. However, for two errors on two of the qubits, we find

$$000 \xrightarrow{error} 110 \xrightarrow{correction} 111 \xrightarrow{decoding} 1 \tag{5}$$

$$111 \xrightarrow{error} 001 \xrightarrow{correction} 000 \xrightarrow{decoding} 0 \tag{6}$$

As we can see, the correction and decoding process are unable to correctly negate the error.

We call the expression for the encoded bit a **codeword**. We also make a distinction between detection and correction of an error. It is always true that correction is easier than correction. Finally, consider

$$000 \xrightarrow{error} 111 \tag{7}$$

$$111 \xrightarrow{error} 000 \tag{8}$$

This error cannot be corrected because we cannot detect it. Whenever an error maps a codeword into another codeword, we become unable to correct it. It is a very general concept.

3 Code distance

We can define the **Hamming distance**. Given two n bit strings, A and B, the Hamming distance is the number of bits that you need to flip in order to go from A to B. As an example, let $A = 010$ and $B = 001$. We require two flips to go from A to B. Thus, $D_H = 2$.

It also follows that the Hamming distance is the number of errors that we cannot correct for.

Let us define another important concept, the **Hamming weight**. The Hamming weight is the number of nonzero bits in our codeword.

Finally, the **code distance** is the maximum distance of the code, which in turn is the smallest Hamming distance between codewords.

Example: For an n bit repetition code, the distance of the code is n. A number of $n - 1$ bitflips are detectable.

Ultimately, the error correction that we perform is limited by our choice of correction method. Consider the following table of errors on the 000 codeword:

No. of errors	Codeword
0	000
1	001
2	011
3	111

We have drawn the line at the point where we can no longer correct for errors. We can draw the Hamming distance around a codeword as a sphere. Every error inside the sphere will be mapped onto the codeword at its centre, but once an error takes us out of the sphere, it will be corrected into a codeword different from its original one.

In other words, we don't want the spheres to overlap. This leaves us with a number of $\frac{n-1}{2}$ correctable errors for an n bit repetition code. Note that when $n-1$ is odd, we must always round down to the nearest integer.

4 Computations

Consider the encoded NOT gate. It has the following function on the encoded bits:

$$0 \xrightarrow{NOT} 1 \tag{9}$$

$$1 \xrightarrow{NOT} 0 \tag{10}$$

In the repetition code, flipping all n bits implements a **logical NOT gate**. By logical, we mean that the gate affects the entire codeword that we have stored to mimic the effect of a single bit.

5 Going Quantum

Would the same concepts carry over without any problem into the quantum regime? Unfortunately not. The repetition code is a prime example where the **no-cloning theorem** prevents us from copying arbitrary quantum states into multiples of themselves.

The following aspects significantly complicate the transition to quantum error correction:

1 The no-cloning theorem forbids arbitrary qubit states from being copied.

2 Measurements change the state, which complicates detection.

3 Quantum errors are much more complicated than classical errors - there are more ways in which they can occur.

The last point might require some clarification. Any unitary transformation can become an error. In addition, we have the various ways in which a quantum state can decohere. Together, they add up to an infinite amount of errors, compared to just the single classical error!

6 The Quantum Repetition Code

This is an example of the most naive way in which we can start considering the quantum correction of errors. However, as we shall see later, even the quantum repetition code is an example of a more powerful formalism.

We take the basis states, and we clone the computational basis states, such that

$$|0\rangle \to |000\rangle = |0\rangle_L \tag{11}$$

$$|1\rangle \to |111\rangle = |1\rangle_L \tag{12}$$

Here, $|0\rangle_L$ and $|1\rangle_L$ are the logical qubits. We also say that the codewords, $|000\rangle, |111\rangle$ live in a **codespace**. We say that the codespace is spanned by the codewords.

For an arbitrary superposition, we obtain

$$\alpha|0\rangle + \beta|1\rangle \to \alpha|000\rangle + \beta|111\rangle \tag{13}$$

Note that since we are only copying the basis states, it is not cloning and therefore perfectly allowed.

6.1 The bit-flip error

The bit-flip error takes us

$$|0\rangle \to |1\rangle \tag{14}$$

and vice versa. It is equivalent to the Pauli X matrix

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{15}$$

So an error on the 2nd qubit of our three-qubit example would look like

$$I \otimes X \otimes I = \alpha|010\rangle + \beta|101\rangle \tag{16}$$

How can we correct for this error? We are not allowed to measure the qubits individually, since this measurement would tell us about the state and thus collapse the superposition. Instead, we introduce the concept of **parity**. The state $|00\rangle$ has even parity, and $|01\rangle$ has odd parity. We can then measure without collapsing the superposition, because the states in the above superposition, for example, have the same parity. We can measure a property as long as it belongs to every state in the superposition.

For the three-qubit code example, we have that $|000\rangle$, and $|111\rangle$ are examples of even parity states, and $|010\rangle, |110\rangle$ are examples of odd parity states.

So, we need a measurement that tells us about the parity without distinguishing the states. In order to perform this measurement, we require a CNOT gate and an ancilla qubit. Then, we apply the CNOT gate to each of our two (or three) qubits and the ancilla, and finally we read off the ancilla. Thus,

$$|00\rangle \to |0\rangle_A \tag{17}$$

$$|11\rangle \to |0\rangle_A \tag{18}$$

$$|01\rangle \to |1\rangle_A \tag{19}$$

$$|10\rangle \to |1\rangle_A \tag{20}$$

where $|0\rangle_A$ is the ancilla. However, we can also perform this measurement in a simpler way, that uses fewer gates. For example, the Z gate will measure Pauli Z in the computational basis. This will tell us about the parity of the qubits.

We must consider the parity of all the pair of the code. For the three-qubit code, the $Z \otimes Z$ measurement will have outcome $+1$ for even parity, and -1 for odd parity when applied to two of the three qubits.

So, for the three-qubit repetition code, we can think of three measurements that allows us to check the parity.

$$Z \otimes Z \otimes I \tag{21}$$

$$Z \otimes I \otimes Z \tag{22}$$

$$I \otimes Z \otimes Z \tag{23}$$

But note that since $ZZ = I$, the last measurement can actually be formed as a product of the other two. This is a sign of an underlying group-theoretical structure which we will explore more thoroughly in upcoming lectures.

7 Correcting quantum errors

Assume that there is only one error on our arbitrary superposition,

$$Z \otimes I \otimes I \left(\alpha \left|000\right\rangle + \beta \left|111\right\rangle\right) = \alpha \left|100\right\rangle + \beta \left|011\right\rangle \tag{24}$$

Then, if we apply the parity measurements defined above, we find

$$Z \otimes Z \otimes I \rightarrow -1 \tag{25}$$

$$Z \otimes I \otimes Z \rightarrow -1 \tag{26}$$

$$I \otimes Z \otimes Z \rightarrow +1 \tag{27}$$

and similarly for every other error.

We can write down a so-called **syndrome table** for all possible one-qubit X errors.

$X \otimes I \otimes I$	$I \otimes X \otimes I$	$I \otimes I \otimes X$
$-$	$-$	$+$
$-$	$+$	$-$
$+$	$-$	$-$

So, our correction strategy should be:

- Identify where the error occurred
- Apply a bit-flip operation where the error occurs

8 More general errors

Let us consider a general error

$$U = e^{-i\theta X} = \cos\theta I - i \sin\theta X \tag{28}$$

So that for our general superposition, we have

$$\alpha \left|000\right\rangle + \beta \left|111\right\rangle = \cos\theta \left(\alpha \left|000\right\rangle + \beta \left|111\right\rangle\right) - i \sin\theta \left(\alpha \left|100\right\rangle + \beta \left|011\right\rangle\right) \tag{29}$$

But now, measuring with $Z \otimes Z \otimes I$ collapses the superposition to either of the terms. But, if $\theta \ll 1$, then detecting the error will correct it!

This simple example is very powerful. It tells us that our error correction strategy will work in general.

9 The encoding circuit

In order to encode our qubits into logical qubits, for the case of the repetition code we want a circuit that gives us

$$\alpha |0\rangle \to \alpha |000\rangle \tag{30}$$

This cannot be just a unitary. Instead, we need an **isometry**, which is a unitary plus a Hilbert space expansion. The following circuit encodes the qubit:

stuff

All encoding is based on CNOT gates. In summary, we have that

- The encoding circuit encodes the state
- Two measurements detect the errors
- X gates corrects the errors

10 Logical operators

Consider the logical X operator. It flips the computational basis state. How can we achieve this on our encoded qubit? One stupid way to do this would be to decode our codewords, such that

$$\alpha |000\rangle + \beta |111\rangle \to \alpha |0\rangle + \beta |1\rangle \tag{31}$$

apply a single gate and then re-encode them. This is not a good way, since it requires a lot of gates.

The good way would be to apply the operator

$$X \otimes X \otimes X = \bar{X} \tag{32}$$

which we call the logical X operator. Clearly

$$\bar{X} |0\rangle_L = \bar{X} |000\rangle = |111\rangle = |1\rangle_L \tag{33}$$

which is what we want.

Similarly for the logical Z operator, we want

$$\alpha |000\rangle + \beta |111\rangle \to \alpha |000\rangle - \beta |111\rangle \tag{34}$$

But now, there are many different ways in which we can do this. We have ZII, IZI, IIZ, and ZZZ that all achieve the right thing (note that

we have dropped the tensor product notation). Each of the first three operators has weight 1. This means that they are indistinguishable from a single Z error.

Recalling what we said before about code distance and the Hamming distance, it means that the code cannot protect against a Z error, also called a phase error.

11 Distance of a Quantum Code

The logical operators are equal to the smallest undetectable error. We can define the minimum distance d in a quantum code as the smallest weight of a non-identity logical operator. That is, the smallest weight is the undetectable error.

So, given what we know about the repetition code, we find that it has distance $d = 1$, which is very bad.

Note on notation: we will use so called (n, k, d) notation, where
n is the number of physical qubits used to write one logical qubit,
k is the number of encoded qubits
d is the quantum distance of the code

It follows that $k < n$ always.

12 Correcting phase errors

Can we think of a way to correct a single Z error? We first note that

$$Z = HXH \tag{35}$$

and we know that the repetition code can correct for X error. Thus, we try applying $H \otimes H \otimes H$ on the repetition code. We find

$$\alpha |000\rangle + \beta |111\rangle \rightarrow \alpha |+++\rangle + \beta |---\rangle \tag{36}$$

Then, we measure with the operators XXI, XIX and IXX. Again note that the last measurement can be made up of the previous two. We then measure the parity in the $\{|+\rangle, |-\rangle\}$ basis. We again make use of CNOT and an ancilla to record the measurement.

We get positive outcomes for even parity, and negative outcomes for odd parity. This works exactly the same as in the $\{|0\rangle, |1\rangle\}$ basis.

But just like the previous code failed for phase errors, this code fails for bit-flip errors. So is there a way to combine both approaches?

13 The Shor code

The code idea is: encode a bit-flip code in a phase flip code. That is, take

$$|+\rangle \to |+++\rangle \tag{37}$$

$$|-\rangle \to |---\rangle \tag{38}$$

Take each qubit and encode them in the bit-flip code. This is called **code concatenation**. We encode each codeword into more codewords.

So we get the following code

$$|0\rangle_L \to |+\rangle_{bitflip} |+\rangle_{bitflip} |+\rangle_{bitflip} = (|000\rangle + |111\rangle)^{\otimes 3} \tag{39}$$

$$|1\rangle_L \to |-\rangle_{bitflip} |-\rangle_{bitflip} |-\rangle_{bitflip} = (|000\rangle - |111\rangle)^{\otimes 3} \tag{40}$$

Note here that we have neglected to write out the normalisation. This is intentional, as it would quickly get very messy.

The logical operators are then for this code

$$\bar{X} = ZII \otimes ZII \otimes ZII = ZIIZIIZII \tag{41}$$

$$\bar{Z} = XXXIIIIII \tag{42}$$

The distance of the Shor code is 3, and the error that we can correct for is

$$\frac{3-1}{2} \sim 1 \tag{43}$$