Quantum Error Correction

Diego Arcelli

June 25, 2022

Università di Pisa

Contents

1 Quantum Error Correction

The hardware devices used to implement quantum circuits are sensitive to noise (much more than classical computers). This might cause the information carried by the circuit to be modified in an unpredictable way. For this reason being able to detect and correct these errors it's a topic of major importance.
Sadly handling errors in the quantum case it's more troublesome than classical computing: first, we cannot copy arbitrary qubits (because of the no cloning theorem), then we cannot measure a qubit to check its state because this would destroy the superposition and, moreover, errors are continuous (while in classical computing we only have bits flip errors).

2 General Framework

The general framework applied for quantum error correction is similar to the one used in classical error correction. The first step is defining and error model, which is a description of the type of error we want to defend against. Once we have the error model, in order to defend a quantum state $|\psi\rangle$ from the error that might occur, we'll define an encoding operator U_{enc} which, by adding some extra ancillary qubits to the state, will produce an encoded state $|\psi_{enc}\rangle$:

$$U_{enc}\,|\psi\rangle\,|00\ldots0\rangle = |\psi_{enc}\rangle$$

The space of all the possible encoded space will form a sub-space of the Hilbert space that we'll call the Quantum Error Correcting Code (QECC). If $|\psi\rangle$ is composed by k qubit, and with the encoding we add n ancillary qubits, then the code $\mathcal{C}$ will be a sub-space of the 2^{k+n} Hilbert space. The idea of adding ancillary qubits, is to use them to store the information about the error that occurred, which we call the **error syndrome**. Then we define a recovery operator $\mathcal{R}$ that, if an error occurs on the encoded state $|\psi_{enc}\rangle$, will undo the effect of the error, based on the information gained with the error syndrome. Finally we'll decode $|\psi_{enc}\rangle$ back to $|\psi\rangle$ that, with high probability, will not contain effects of the error.
In general given a quantum code $\mathcal{C} \subset \mathcal{H}$, we say that a noise operator $\mathcal{E}$ is recoverable it there exists a recovery operator $\mathcal{R}$ such that for every density matrix ρ associated with a state $|\psi\rangle \in \mathcal{C}$:

$$\mathcal{R}(\mathcal{E}(\rho)) = \rho$$

In the following sections we'll analyze some of the most relevant type of errors and we'll show circuits which can detect and correct the errors, using the framework we just explained.

3 Correcting single qubit errors

3.1 Correcting bit flip errors

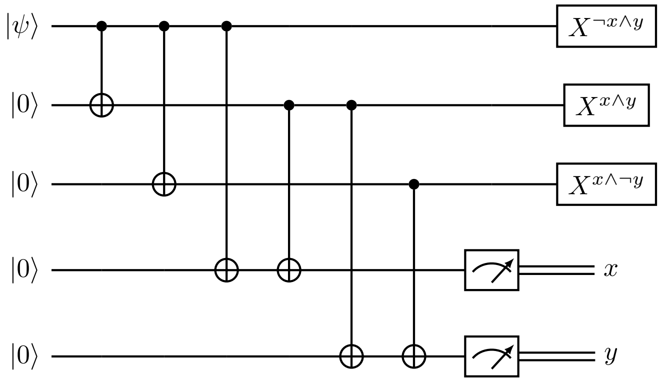
Let's start with the simplest error that we might have, the bit flip error, where a generic qubit in state $\alpha\,|0\rangle + \beta\,|1\rangle$ it's changed into $\alpha\,|1\rangle + \beta\,|0\rangle$ with probability p. This can be seen as an application of the X Pauli's gate to the qubit. To correct this kind of errors we use the following three bit encoding:

$$|0\rangle \to |000\rangle\,, \quad |1\rangle \to |111\rangle$$

which applied to a generic qubit is:

$$\alpha\,|0\rangle + \beta\,|1\rangle \to \alpha\,|000\rangle + \beta\,|111\rangle$$

So we encoded physical qubits to a sequence of logical qubits. Note that this is not a violation of the no cloning theorem since we're not mapping $\alpha\,|0\rangle + \beta\,|1\rangle$ to $(\alpha\,|0\rangle + \beta\,|1\rangle)^{\otimes 3}$.
The idea we want to apply is to check if the value of the three qubit is the same, if not, we impose the value of the majority bit. As we said earlier we cannot do that by measuring the state, otherwise the superposition will be destroyed. What we can do instead is using some extra qubits, to store in them the information of which qubit of the state has been flipped. All this can be done with the following circuit:



The first two CNOT gates are used to apply the three bits encoding. Then with the second two CNOT gates we check if the first two qubits are equal. If they are equal and they're both $|0\rangle$ then the first ancilla quibt is not flipped. If they are equal an they're both $|1\rangle$ the first ancilla qubit it's flipped twice (so it goes back to 0). If they're different the ancilla qubit is flipped and so it is set to $|1\rangle$. With the last two CNOT gates we repeat the same procedure, but this thy we compare the second and the third qubits and we act on the second ancillary qubit. Finally we measure the ancillary qubits and we get the values x and y. The value of x tells us if the first and the second qubit were the same, while the value of y tells us if the second and the third qubit were the same.

Input state	Measurement	Effect
$\alpha\|000\rangle + \beta\|111\rangle$	$x = 0, y = 0$	No qubit is flipped
$\alpha\|100\rangle + \beta\|011\rangle$	$x = 0, y = 1$	The third qubit is flipped
$\alpha\|010\rangle + \beta\|101\rangle$	$x = 1, y = 0$	The first qubit is flipped
$\alpha\|001\rangle + \beta\|110\rangle$	$x = 1, y = 1$	The second qubit is flipped

Note that if two or all the three qubits are flipped, this circuit will produce a wrong state. In general this strategy can be extended by using an encoding of length n.

3.2 Correcting phase flip errors

A more tricky type of error which we might have in quantum computing is the phase flip error where a generic qubit $\alpha|0\rangle + \beta|1\rangle$ is changed into $\alpha|0\rangle - \beta|1\rangle$. This can be seen as an application of the Z Pauli gate. It turns out that solving phase flip errors is very similar to solve bit flip errors, since phase flip errors in the canonical basis become bit flip errors in the Hadamard basis since $X = HZH$ and because of that, $Z|+\rangle = |-\rangle$ and $Z|-\rangle = |+\rangle$. We can exploit this fact by using the following encoding:

$$\alpha|0\rangle + \beta|1\rangle \to \alpha|+++\rangle + \beta|---\rangle$$

which can be obtain from the bit flip encoding, by applying to each one of the three qubit the H gate. Suppose that with this encoding a phase flip error occurs in the second qubit. Then, since in the Hadamard basis, a phase flip acts like a bit flip, the encoded state becomes:

$$\alpha|+-+\rangle + \beta|-+-\rangle$$

if we now apply and H gate to each qubit of the state, it becomes:

$$\alpha|010\rangle + \beta|101\rangle$$

which is like if a bit flip error occurred in the second bit of the encoding for the bit flip error. Therefore we can simply run the circuit, for the bit flip correction, and we'll correctly recover the initial phase flip error.

3.3 Nine-qubits correcting code

We have seen a circuit which can correct bit flip errors and a circuit which can correct phase flip errors, but what if a qubit is affected by both the errors? We can also easily solve those errors, using the 9-qubit encoding defined by Shor [1], which can be derived from the previous encoding as follows: first we start from the encoding for the phase flip errors:

$$\alpha |0\rangle + \beta |1\rangle \rightarrow \alpha |+++\rangle + \beta |---\rangle$$

then we apply to each qubit of the encoded sequence the circuit of the bit flip encoding and we get:

$$\alpha |0\rangle + \beta |1\rangle \rightarrow \frac{\alpha}{\sqrt{8}}(|000\rangle + |111\rangle)^{\otimes 3} + \frac{\beta}{\sqrt{8}}(|000\rangle - |111\rangle)^{\otimes 3}$$

So we have three block of three qubits each. Within each block, all the qubit are supposed to be the same, so we can run the bit flip error correction circuit in every one of the block to protect each block by bit flip errors. Now, if we consider the thee block, the sign of every block should be the same, but if there's been a phase flip error in one of the block, the sign on that block will be flipped. So we can run the phase flip correction circuits on the three block to correct the phase error.

$$(\alpha |0\rangle + \beta |1\rangle) |00000000\rangle$$
$$\alpha |000000000\rangle + \beta |100000000\rangle$$
$$\alpha |000000000\rangle + \beta |100100100\rangle$$
$$\alpha |+00 + 00 + 00\rangle + \beta |-00 - 00 - 00\rangle$$

3.4 Correcting generic errors

Since now we only saw bit flip and phase flip errors, but how do we deal with other type of errors, like a generic rotation of the qubit in the Bloch sphere. Consider a generic error on a single qubit defined by a matrix $E \in \mathbb{C}^{2\times 2}$, and consider the four Pauli matrices I, X, Y and Z. As we already saw X and Z represent respectively bit flip and phase flip errors. Since $Y = iXZ$, it can be seen as bit flip error followed by a phase flip error (the global phase factor i does not matter), while the gate I can be simply seen as the no-error gate. These four matrices span the space of the 2×2 complex matrices, so the before error matrix E can be written as:

$$E = \alpha_0 I + \alpha_1 X + \alpha_2 Y + \alpha_3 Z, \quad \alpha_0, \alpha_1, \alpha_2, \alpha_3 \in \mathbb{C}$$

Basically we've rewritten E as a linear combination of bit flip and phase flip errors, which we already know how to solve. This is important because there's a theorem which state that if a code $\mathcal{C}$ corrects a set of errors $\mathcal{E} = \{E_i\}$, then $\mathcal{C}$ corrects also the span of $\mathcal{E}$. So if we're able to correct bit flip and phase flip errors, then we're able to correct any kind of error.

References

[1] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.