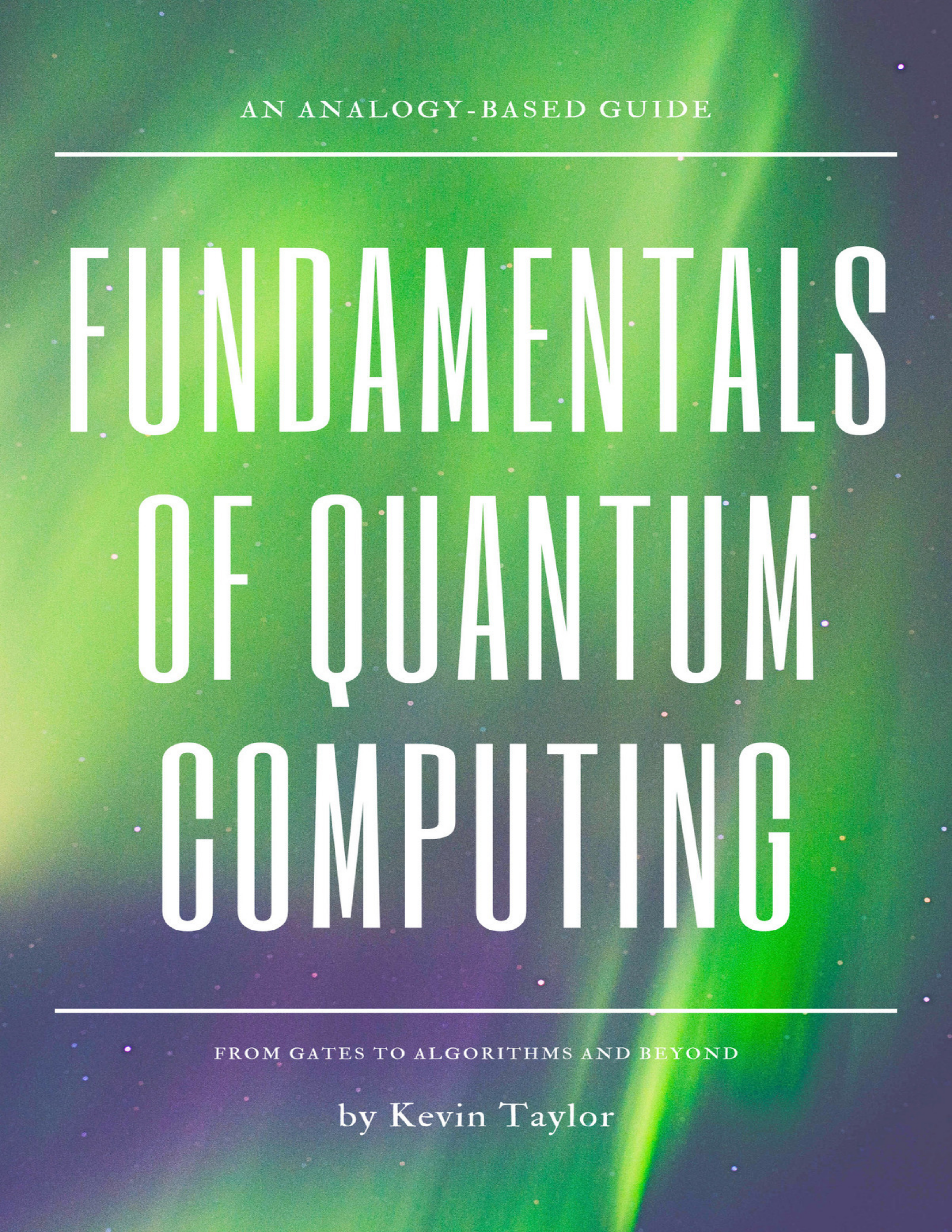


AN ANALOGY-BASED GUIDE

FUNDAMENTALS OF QUANTUM COMPUTING

FROM GATES TO ALGORITHMS AND BEYOND

by Kevin Taylor

CONTENTS

INTRODUCTION

Quantum computing may seem to most a heavy, tedious, and infinitely abstract field. However, as daunting as it may be, the field of quantum computing holds the answers for some of the world's most complex problems, problems that choke up even the most powerful supercomputers. This isn't to say quantum computers are objectively superior to classical computers, but simply different. If classical computers were cars, quantum computers would be planes. In some cases, it makes sense to use the car – it can be impractical to take a plane for the 3 miles you need to travel. Sometimes, though, it makes sense to fly for a few hours instead of trying to drive across an entire continent. Of course, this analogy is a simplification of the differences between the two, but it still serves as an important boundary nonetheless.

Although each of the systems have nuanced structures, they are similar in many ways. You might know of the classical bit, which is either 0 or 1. 1 is on, 0 is off, it's true vs false and active vs inactive. This is because, practically, either an electrical signal is on or off – there is no in between. Quantum computing fills in the gap where classical computing ends, and enters into a specialized zone where reality is suspended, and the future is simply a probability amplitude. There is no more 0 OR 1, there is both. A unit of data that exhibits this characteristic, called a qubit, or quantum bit, can simultaneously be 0 and 1 at the same time. In fact, it can represent even more than 0 or 1. This property of having multiple opposing states is called superposition, and it is critical to quantum computing.

Schrödinger's Cat

This phenomenon might seem counterintuitive, so we'll start with a simple example involving a very famous cat first proposed by Erwin

Schrödinger in 1935. Imagine you place a cat in a box with a very special jar. This jar has a 50% chance of releasing poison that would certainly kill the cat. You seal the box, leaving the cat and jar inside, and come back a week later. Obviously, in a classical sense, the cat is either dead or alive. But when we transcend the classical world of logic and certainty, we imagine the cat as both dead and alive, suspended in a state of superposition in the instant before we can be certain of any conclusion. But this paradoxical feline thought experiment isn't just a detached idea; it manifests in the form of quantum computers. These machines harness the power of superposition to explore multiple avenues of a given problem simultaneously. Where a classical computer would simply need to move sequentially and handle each case at a time, step by step, quantum computers expand their path of computation exponentially, gracefully solving unimaginably large problems. This fundamental advantage to quantum computers allows them to speed through calculations that would take your average computer eons to perform. In fact, quantum computers can solve in 4 minutes what would take the world's top supercomputers over 10,000 years to accomplish. In a sense, quantum computers work in parallel. Instead of performing 500 million operations individually, quantum computers can perform them all at once.

What to Expect

In the pages that follow, we will unravel the intricacies of qubits, explore quantum gates, and peek into the world of quantum computing. As we delve into quantum computing's folds, we'll keep our focus practical, bridging the gap between the abstract and the physical. This book isn't just about theoretical applications or high-level ideas of quantum computing, it's about practical application and analogous learning.

By the time you turn the final page, you'll have a firm grasp of the key principles behind quantum computing and the sophisticated ideas it holds. Even if you don't work in a tech-related field, and all of these complex ideas and solutions might seem distant and detached, this kind of technology will only be developed further. Even if all quantum computers crash and burn tomorrow, the ideas in this book can have a profound impact on the way you think, the way you see the world around you. Though whether it be five years or five decades, quantum computers will eventually intertwine themselves with our daily lives. With this new adoption will come extended use such as heightened calculation speeds, improved biological simulations of organisms, and even more accuracy and speed in artificial intelligence algorithms. Welcome to a new era of computation, where 1s, 0s, and the uncertain void between them aid us in solving problems of any magnitude.

CHAPTER 1

The Qubit and Superposition: A High-Level Overview

As we've seen in the introduction, the world of quantum computing offers a departure from the familiar realm of classical computing. Just as planes redefine the possibilities of travel, quantum computers redefine the limits of computation. But how exactly do these quantum marvels operate? Let's journey into the heart of quantum computing fundamentals.

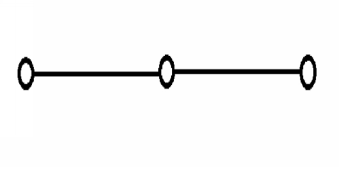
A qubit by its very definition is superpositional, meaning it can have both states of 0 and 1 at the same time. Just as Schrödinger's famous cat was both dead and alive at once, qubits themselves are both 1 and 0.

The Problem State

Before we dive into the intricacies of the qubit, it is important to understand in some abstract sense *what* the quantum computer is doing, before the math and quantum phenomena become too overwhelming.

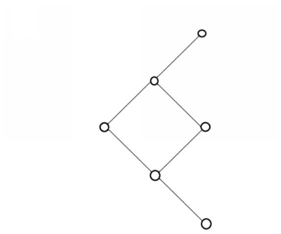
To grasp this, let's use visual aids.

Imagine you're running a program on a classical computer, say, searching a database. The computer performs operations like this:



Each dot on the line represents another step of the program, another piece of data analyzed. The classical computer moves step by step by step by step, only focusing on one thing at a time.

A quantum computer running the same task would exemplify the following diagram:



As you can see, the quantum computer exponentially navigates the solution space by essentially using superposition to cover all the bases. If the qubit is both 0 and 1, then the computer is able to take advantage of both states to check multiple scenarios at once, instead of moving linearly through the problem like classical computers. Liken this to a tree: the quantum computer branches out, and branches out from those branches. The classical computer, on the other hand, simply grows one branch and, in doing so, sacrifices the calculated efficiency of the quantum system.

A More Advanced Definition

Except how can we define a qubit? Of course, it isn't enough to simply say, "It is both 0 and 1" or "it is superpositional." As aforementioned, a qubit is defined as a probability amplitude, with a certain probability of being 0 and a certain probability of being 1. Let's take a look at the mathematical definition below:

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, $\psi =$ qubit state

$|\alpha|^2 + |\beta|^2 = 1$

This might seem like a lot, especially if you aren't heavily involved in mathematics. However, it is deceptively simple once you understand the notation and vocabulary.

- $|\psi\rangle$: This is the ket notation for the quantum state of the qubit. The $|\rangle$ symbol represents a "ket," which is used in quantum mechanics to denote a vector in a quantum state space. ψ, the variable inside the ket, indicates the state of a given qubit in this scenario.
- α & β: These are complex coefficients (also known as amplitudes) associated with the two basis states of the qubit. α & β measure the probability of the qubit being in the states 0 and 1 respectively.
- $|0\rangle$: This is the ket notation for the "0" state of the qubit. It represents one of the possible states that the qubit can be in. This state is often referred to as the "ground state."
- $|1\rangle$: This is the ket notation for the "1" state of the qubit. Like $|0\rangle$, it represents another possible state that the qubit can be in. This state is often referred to as the "excited state."
- State Vector: Imagine an arrow pointing off into space in a specific direction with a specific magnitude of 1 (the normalization of α and β sum to 1). That's a vector. A state vector is simply a vector whose characteristics indicate certain conclusions about a qubit's state.

Now that we understand the basic terms and definitions, the equations take on a much more intuitive approach. α represents the probability that the qubit is in the state 0 at any given time, while β represents the probability that the qubit is in the state 1 at any given time. We can view the exact probabilities (either α or β) of the qubit having a specific state in familiar terms by simply taking the absolute value and square of any individual variable:

$\alpha = \frac{1}{\sqrt{2}}$

$|\alpha|^2 = \frac{1}{\sqrt{2}^2} = \frac{1}{2}$

ψ has a $\frac{1}{2}$ chance of being 0, in other words, a 50% chance of being 0. This also means β must be $\frac{1}{\sqrt{2}}$ as well. This is because the absolute values of α and β squared and summed is equal to 1, as shown above, which is called the normalization condition. All the normalization condition does is ensure that all possible outcomes of the qubit state add up to 1, or the percent chances of the qubit having a state of 0 and 1 add up to 100%.

Quantum Phase

This is the other side of qubits: the phase. While the quantum state ψ is relatively easy to define as simple probabilities, quantum phase can be more nuanced and a bit tricky to understand, so we will explore the concept with examples.

Imagine you're in a realm where mysterious forces govern the behavior of objects. In this world, there are two kinds of "quantum magnets": one is exceptionally strong, hovering 10 feet above the ground, and the other is relatively weaker, attached to the floor. Somewhere between them, suspended in the air, there's a special pole. The end of the pole farthest from the magnets is fixed in space.

This pole exists in a unique state, much like a quantum bit, or qubit. It's as if the pole can choose between pointing up or down, representing the 0 or 1 state of a qubit. But here's where it gets interesting: the way this pole points is determined not just by the powerful magnet above or the weaker one below, but by the subtle interaction of their forces.

The strong magnet on the ceiling certainly has a more significant influence on the pole's orientation. However, the weaker magnet on the floor still manages to exert some pull, tugging the pole in its own direction. As a result, the pole doesn't point directly up or directly down, but somewhere in between. The extent to which the pole leans towards the stronger magnet depends on their relative strengths, which, in this case, is the stronger magnet above the pole.

Now, let's translate this into qubits. Imagine that the stronger magnet corresponds to the probability amplitude α (for the $|0\rangle$ state), and the weaker magnet corresponds to β (for the $|1\rangle$ state). The angle at which the pole tilts can be compared to the qubit's phase angle. This angle isn't a physical tilt, but instead a mathematical concept that influences the behavior of quantum states.

In this analogy, the phase angle represents the balance between the influences of α and β. If the top magnet's pull dominates, the phase angle tilts one way; if the bottom magnet's influence gains the upper hand, the phase angle tilts the other way. When the pole points straight ahead, α and β are equal to one another, balancing each other out.

Of course, leaving the analogy and returning to the reality of quantum mechanics, things get messier. The phase itself is actually associated with complex numbers($i = \sqrt{-1}$). Another key component of the phase is called the phase angle, or the angle that the imagined pole would point. An angle of $45\degree$ for example would indicate that our analogical pole would be pointing more up towards the top magnet than down towards the bottom magnet. The phase angle can be defined in the following way:

$$\theta = \arg(\alpha) - \arg(\beta)$$

$\arg(x)$ represents the angle between the positive real horizontal axis and the line connecting the complex number to the origin in the complex plane.

This might seem a bit unclear at first, so let’s take it slow. First, to understand this example, we need to realize another thing about α and β. They exist in the complex plane, meaning that they aren’t always going to be a nice, real number, like $\frac{1}{2}$. Instead, these amplitudes can take the form $a + bi$, where a is a real number and b is a coefficient to the complex root i. This form describes a complex number, meaning parts of the variables may not exist (i is quite literally an “imaginary number”). This means that α and β can look pretty messy at times, such as:

$$\alpha = \frac{1}{\sqrt{3}} + \frac{i}{\sqrt{6}}$$

$$\beta = \frac{1}{\sqrt{3}} - \frac{i}{\sqrt{6}}$$

Note that these values of α and β are valid because they satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$.

Now that we know the probability amplitudes α and β can consist of complex and real numbers, we can return to the $\arg(x)$ function. Also known as the $\arctan2(x,y)$ function calculates the angle between a probability amplitude and the real x-axis, taking complex factors into account. For example, if

$$\beta = \frac{1}{\sqrt{3}} - \frac{i}{\sqrt{6}}$$

And we called $\arctan2(\beta)$, that would be equivalent to calling $\arctan2(x,y)$ where x is equal to the real portion of β, or $x = \frac{1}{\sqrt{3}}$, and y is equal to the complex coefficient, or $y = -\frac{1}{\sqrt{6}}$(Note that y is the coefficient and not the complex term, meaning we do not include i in our definition of y).

So, again returning to the definition of θ, the angle between our two amplitudes, we can generalize it as the difference between the angles of α and β relative to the x-axis. The angle θ, the phase angle, is crucial to many aspects of quantum computing. In fact, it plays a role in determining how the probability amplitudes α and β interact, in an event known as quantum coherence, in which quantum systems attempt to maintain their superposition over time.

Visualizing the Qubit

Now that we've explored the true meaning of qubit and some key properties, let's talk a little bit more about how we can actually picture a model of it, instead of just imagining the ideas behind the concept. The most common method of visualizing a qubit is through what is known at the Bloch sphere, a 3-dimensional spherical canvas to paint our qubit. Let's imagine that the state $|0\rangle$ exists at the top of the sphere, directly north. $|1\rangle$ lives at the bottom, exactly south. Finally, let's introduce our state vector, an arrow originating from the center of our Bloch sphere. The position of the state vector depends on two conditions: the phase and the state.

For the sake of the example, α and β will take on the following values:

$\alpha = \frac{1}{\sqrt{5}}$

$\beta = \frac{2}{\sqrt{5}}$

This is to say there is a 20% chance the qubit will have a state of 0 and an 80% chance the qubit will have a state of 1. Already, we can use this information to imagine an aspect of our state vector.

Imagine that by increasing α, we rotate the state vector up towards the north pole, or towards $|0\rangle$. In contrast, increasing β, and thus, decreasing α, we rotate the state vector down towards the south pole, representing a shift towards the state $|1\rangle$. This concept should seem familiar from our previous example with the hovering pole between magnets: phase.

The degree of vertical rotation applied to the state vector is controlled by the phase angle. However, there are still more dimensions to consider: what about the left and right rotations?

If the phase angle is positive, the state vector will rotate counterclockwise around the Bloch sphere. If the phase angle is negative, the state vector will rotate clockwise around the Bloch sphere. However, even horizontal rotation affects the probability amplitudes of the qubit. A counterclockwise rotation (indicating a positive phase angle) increases the probability that the qubit is measured in a state of 0. Similarly, a clockwise rotation increases the probability that the qubit is measured in a state of 1.

It is important to recognize that this model only truly works when you consider one qubit at a time. If you were to attempt to imagine multiple qubits in the same system, the Bloch sphere would need to expand dimensionally, due to phenomena such as entanglement. Said phenomena will be discussed further along in the rest of the book.

However, as far as the single-qubit representation goes, it is very intuitive. The Bloch sphere is an great visualization tool for understanding quantum states because it offers a clear geometric representation of qubit states. By mapping the complex probability amplitudes onto a 3D sphere, it provides a tangible way to grasp concepts like superposition and phase. The sphere’s simplicity enables us to visualize qubit rotations and transformations, making it a powerful aid in comprehending the complex behaviors of quantum systems.

Conclusion

In this opening chapter, we’ve taken our first steps into the captivating world of quantum computing. We’ve discovered the remarkable nature of qubits, with their ability to exist in superposition, embodying multiple states at once. The enigmatic concept of phase has come into focus, guiding qubits’ interactions and behaviors. As we move forward in this journey, we’ll delve deeper, exploring quantum phenomena, quantum gates, and the elusive quantum algorithm.

CHAPTER 2

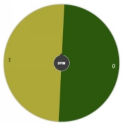
Deeper Quantum Phenomena

In this chapter, we will contemplate the complex and intricate ideas and theories buried just beneath the surface of quantum computing.

On the surface, these ideas may just be words ⁻ descriptions and abstract explanations whose effect may never be realized. Memorizing these ideas will get you nowhere without seeing them unfold before your mind's eye. True learning comes from deep realization and the communicative nature of ideas. While reading this chapter, do your best to connect every concept in here to some sort of hypothetical situation in your life, no matter how abstract or unfamiliar the idea might be. Even making an attempt to relate this information to your existence in any way aids growth.

Superposition and the Enigma of the Wavefunction Collapse

The term superposition has been discussed previously in the chapter, but let's dig a bit deeper into it. Right in the instant before we are certain of a qubit's state, it is suspended in a state of superposition, existing as both 0 and 1 until it crystallizes as either option. You might imagine spinning a picker wheel, like the one pictured below:



When the wheel stops spinning, and chooses a concrete state, it will remain that way. But right before it stops spinning, it fluctuates between 0 and 1 based on probability. As the wheel spins and begins to slow, it is still passing over 0 and 1, fluctuating between them. The wheel depicted above has an equal chance of landing on 0 and 1 because both 0 and 1 take up equal space in the wheel. This is superpositional because the qubit's state is not static up until the point of certainty, more, it is in flux. However, qubits don't always have to have a 50/50 chance of being 0 and 1. They can be split 30/70, 60/40, and even 99.999/0.001. However, no qubit can ever exist with either a probability of 0 for having the states 0 or 1, because then it wouldn't be superpositional. For example, if the probability of a qubit having a state of 1 was 0, then there was no way the qubit would ever be 0. In this case, said qubit wouldn't be a qubit at all, because a qubit is defined as superpositional, and a unit of data with only one attainable value is not superpositional but constant.

The wavefunction collapse is a process that signifies the end of a qubit's superpositional state, and symbolizes the start of the qubit's entry into the mundane realm of certainty.

Imagine a qubit suspended in a state of superposition, embodying the probability of both 0 and 1. As long as we don't measure its state, this qubit exists in a state of ambiguity, poised between the two possibilities. However, the moment we observe or measure the qubit, a transformation occurs—an event known as wavefunction collapse. In an instant, the once-fluid probabilities solidify into a singular, definite outcome. You might imagine this in the context of our wheel-spinning example. If we apply this phenomenon to that specific scenario, we might understand the solidification of qubit state, the collapse of the wavefunction, as the wheel finally landing upon a specific value.

In a complex sense, a qubit doesn't really have a value until we try to determine what that value is. This is because by measuring a qubit at a certain time, we force a certain outcome to present itself. A qubit constantly flips between 0 and 1, but measuring a qubit at a given time will only results in 0 or 1, serving to “collapse the wavefunction,” effectively demanding that the qubit relinquish its grasp on duality and accepting a singular value.

If you decide to stop the wheel while it just so happens to be moving over the sector marked “1”, then the qubit has a state of 1. Similarly, if you stop the wheel over “0”, and force the wavefunction to collapse, the qubit will have a state of 0. Of course, in the context of qubits, you wouldn’t actually see the wheel spinning, only the result. In this way, we begin to understand how the qubit as a unit can be superpositional.

Quantum Entanglement

Entanglement is a special, overarching concept that lies deep inside the heart of quantum computing. It is unique because it doesn’t just describe qubits, but actual particles themselves. Quantum entanglement is the idea that 2 particles are linked to each other no matter how far away they are.

Imagine two particles, let’s call them Particle A and Particle B, that have interacted and become entangled. The remarkable property of entanglement is that the quantum state of the combined system cannot be described independently for each particle; instead, the state of one particle is inherently tied to the state of the other, no matter how far apart they are. This connection persists even when the particles are separated by vast distances, defying the constraints of classical physics.

The entangled state of Particle A and Particle B can be described as a superposition of possible states, where the outcomes of measurements on one particle are intrinsically linked to the outcomes on the other. This means that if a measurement is performed on Particle A and it collapses into a specific state, the state of Particle B instantly becomes correlated, even though it might be light-years away. This instantaneous connection, termed “spooky action at a distance” by Einstein, Podolsky, and Rosen in their famous EPR paper, is a key trait of entanglement.

A classic example of entanglement involves particles with spin, an form of angular momentum. When two particles become entangled, their spins become correlated, regardless of the distance between them. If we measure the spin of Particle A along a certain axis and find it to be “up,” the spin of Particle B, when measured along the same axis, will instantaneously be found to be “down.” This correlation holds true even if the measurements are taken in different directions.

One classic example of quantum superposition is the jinx. Remember back in 8th grade, talking about how well your math test went was considered a “jinx,” meaning simply talking about it might bring bad luck upon you? The jinx isn’t real, of course, but the underlying idea is very intuitive and indeed replicated on a large scale throughout the quantum realm. Making one action seems to lead to a different outcome in a different situation. An important distinction to make here is that Particle A has no physical contact, direct or otherwise, with Particle B.

Since a hallmark trait of qubits is that they are entangled can hold and process information in ways that classical bits cannot, the entanglement of qubits enables quantum computers to perform complex calculations with a potential for exponential speedup. This opens doors to solving problems that were previously intractable for classical computers.

But beyond that, quantum entanglement has a philosophical value associated with it as well. If everything in the universe is somehow connected, no matter the distance, no matter the infinite logical laws against it, is anything truly random? Can the universe be described by an equation, a sequence of electrons influencing electrons influencing everything? If so, is the idea of an unchanging future, or fate, real? The lines get blurrier and blurrier the more we consider the concept of entanglement, which is why it is interesting food for thought.

Quantum Coherence

Quantum coherence is a principle strongly attached to the raw definition of the qubit: the probability amplitudes α and β and their relation to the state of a qubit. Coherent states have phases that are well-defined and synchronized. Incoherent states, on the other hand, have uncontrollable phases and tend to lose their quantum properties, becoming ordinary bits instead of remaining superpositional qubits.

Picture two dancers moving to the rhythm of a musical composition. The dancers represent qubits, and their synchronized steps embody quantum coherence. Quantum phase guides the dancers' movements, ensuring that they remain in harmony and perform a coherent, synchronized routine. The longer the dancers stay in sync, the longer the show can go on. If the dancers lose harmony, the magic is ruined, and the act essentially falls apart. Quantum phase is a vital dimension to qubits because it allows them to work together; if the phase is an incoherent or generally undesirable state then the qubit loses its special properties and the quantum computer depending on it may experience interference, leading to errors in quantum computations and can affect the reliability of quantum systems. Quantum coherence is delicate, easily disrupted by the outside world. It's as if our quantum dancer is performing on a stage is constantly brushed by invisible gusts of wind. The environment, with its vibrations, interactions, and fluctuations, can disturb the coherence, causing our qubits to lose their synchronized elegance. This phenomenon is known as decoherence. Simply put, without a special and fragile harmony, everything falls apart.

Therefore, a desirable phase is required for qubits to function and remain superpositional. In the case of this broken harmony, the α and β amplitudes may even cancel each other out, resulting in what is known as quantum interference.

Quantum Interference

Imagine a symphony hall where musicians play a multitude of instruments, each producing its distinct sound waves. As these waves converge, they can reinforce one another to create a harmonious crescendo or cancel out to bring about a moment of silence. This interplay of waveforms is a classical illustration of interference, a phenomenon that’s equally captivating in the realm of quantum mechanics.

In the world of quantum physics, interference transcends the familiar world of sound waves and instead represents state probability amplitudes, which can either work together or work against one another.

At its core, quantum interference is the phenomenon in which the probabilities of different quantum pathways combine, leading to enhanced or diminished outcomes upon measurement. This concept arises due to the wave-like nature of quantum particles, encapsulated in the wavefunction that describes a particle’s possible states and their associated probabilities.

Consider a quantum particle, such as an electron, in a superposition of states. This superposition is akin to our symphony hall, with multiple instruments producing distinct waveforms. When these waveforms overlap, they can either reinforce each other, leading to constructive interference and higher probabilities of certain outcomes, or cancel each other out, resulting in destructive interference and reduced probabilities. Teamwork makes the dreamwork, literally.

One of the most iconic experiments illustrating quantum interference is the double-slit experiment. In this setup, particles are fired at a barrier with two slits. As particles pass through the slits, their wavefunctions create an interference pattern on the screen behind the barrier. This pattern arises from the constructive and destructive interference of the wavefunctions emerging from the two slits. Even when particles are sent through one at a time, the interference pattern gradually emerges as each particle contributes to the overall probability distribution.

Quantum interference plays a pivotal role in quantum algorithms and computations. Quantum computers leverage interference to enhance desirable outcomes and suppress undesirable ones. Algorithms are able to exploit interference to amplify the probability of correct solutions while damping incorrect ones. This harnessing of interference allows quantum computers to search through vast solution spaces more efficiently than classical counterparts.

The beauty of quantum interference lies in its duality—simultaneously perplexing and awe-inspiring. It highlights the probabilistic nature of quantum particles, where different possibilities coexist until the moment of measurement, when interference either accentuates their probabilities or leads to their vanishing.

Quantum Tunneling

In the realm of classical physics, a ball rolling down a hill eventually reaches the lowest point of the slope due to the influence of gravity. This behavior adheres to the principles of classical mechanics, where particles obey well-defined trajectories determined by their energy and the forces acting upon them. However, when we venture into the world of quantum mechanics, this intuitive understanding begins to unravel, giving rise to a phenomenon known as quantum tunneling.

Quantum tunneling is a fascinating process in which particles appear to “tunnel” through energy barriers that, according to classical physics, they should not be able to overcome. It sounds to be straight out of a sci-fi movie. It challenges our classical intuitions by demonstrating that particles can traverse barriers that should be insurmountable based on their energy levels.

This phenomenon emerges due to the wave-like nature of quantum particles. Such particles, like electrons, are dictated by wavefunctions, which contains a number of possible positions and energy levels. However, as these particles approach what should be an unwavering barrier, their wavefunctions are able to extend past the classical realm into a sort of forbidden space. This means that there is some possibility, however small it may be, that particles are able to exist on the other side of such barriers.

These particles exist in a state of probability, simultaneously inhabiting various positions and energies. This probabilistic nature gives rise to quantum tunneling, as particles can almost “borrow” energy from their surroundings for the briefest of moments, allowing them to essentially tunnel through obstacles they wouldn’t normally be able to.

A classic analogy for quantum tunneling involves a particle approaching a barrier, much like a person trying to walk through a solid wall. In the quantum realm, however, there’s a finite probability that the particle will materialize on the other side of the barrier without having passed over it or broken it. It simply tunnels through it, as if passing through a multidimensional passage to simply *exist* on the other side.

Despite its seeming paradoxical nature, quantum tunneling has been extensively verified through experiments. While it might challenge our everyday understanding, quantum tunneling opens doors to further technological advancements, and serves as a reminder of the depth of the universe’s quantum fabric: the boundaries of possibility are far more nuanced and multidimensional than they appear on the surface.

Quantum Error Correction

Quantum computing is a delicate dance: qubits must not be caught off balance in their superpositional balancing acts. Quantum error correction (QEC) is the critical foundation upon which the reliability of quantum computers is built. It’s akin to the error-correction codes used in classical computing but adapted to the considerably more delicate world of quantum mechanics.

Quantum computers operate in a realm where information is stored and processed in quantum states, which are highly susceptible to disturbances from their surroundings. This sensitivity to environmental factors, known as quantum decoherence, can cause qubits to flip from their intended state, creating errors that cascade through quantum algorithms, rendering their results unreliable. This phenomenon poses a major obstacle to the development of practical quantum computers.

Quantum error correction is a groundbreaking concept that enables quantum computers to mitigate errors and maintain the integrity of their calculations. The core idea of QEC is to use specially designed quantum codes to encode and protect quantum information in a way that allows errors to be detected and corrected without directly measuring the qubits. This is crucial, because measuring qubits will shatter their superposition.

To start, QEC encodes logical qubits into larger groups of physical qubits. These groups, known as code words, provide redundancy. Errors in the physical qubits can be identified and rectified by comparing the state of the physical qubits to the expected state of the logical qubit. Think of this like checking your work; you compare your answer on the practice sheet to the answer key your teacher or professor provides you with.

Then, quantum error correction employs specialized quantum gates that are fault-tolerant, meaning they can operate correctly even when some of the physical qubits are in error. To detect errors, QEC involves taking measurements known as syndromes, which provide information about the presence and location of errors in the code words. These syndromes are calculated using ancillary qubits that interact with the code words.

Finally, once errors are detected through syndrome measurements, QEC algorithms determine how to correct the errors without directly measuring the logical qubits, which would risk collapsing their delicate quantum states. Correcting errors may involve applying controlled gates and other operations in order to shift the rogue qubits back into the realm we want them to be in.

Essentially, by grouping large numbers of qubits together QEC can handle rogues by comparing them to the larger population of qubits: if 1 or 2 qubits out of 30 are showing incorrect values, chances are that the 1 or 2 qubits are incorrect instead of the the 30 (unless your quantum computer is really bad). When these few rogue qubits are identified, they can more easily be shifted back into their rightful states in order to maximize the effectiveness of the quantum system.

Quantum No-Cloning

In the realm of classical computing, one of its fundamental characteristics is the ability to copy information effortlessly. Given a string of bits representing data, you can create a perfect duplicate with precise accuracy. This property is deeply ingrained in how classical computers function and serves as a cornerstone of digital data storage, transmission, and processing. If I type my name twice into my computer, it will be stored twice.

However, when you enter the enigmatic world of quantum computing, you encounter a fundamental restriction known as the no-cloning theorem. This theorem lays down a firm boundary, asserting that it’s impossible to create an exact copy of an arbitrary, unknown quantum state.

In classical computing, copying data is a straightforward operation. If you have a string of bits, you can simply replicate it, creating a precise duplicate. In quantum computing, things are drastically different. Quantum states, represented by qubits, can exist in a superposition of states, and their precise state cannot be measured without altering it. Attempting to copy a quantum state without disturbing it is where the challenge arises.

At the heart of the no-cloning theorem lies Heisenberg’s Uncertainty Principle, a fundamental concept in quantum mechanics. This principle tells us that we cannot simultaneously know both the position and speed of a particle with precision, because knowing one changes the other. In the context of quantum computing, it means that if we try to measure a qubit to obtain information about its state, say, the velocity, we actually modify that state.

This is powerful for many reasons, especially in the context of quantum-enhanced cryptography. For example, if a sender and receiver utilize quantum encryption in securing their data, there is no possible way for their communication to be breached; their quantum keys cannot be cloned.

Quantum Bell Measurements

Bell measurements are fascinating because they allow us to look at hidden relationships behind the scenes of quantum trickery. When you have two entangled qubits (whose states we don’t know because they are superpositional by definition), they are directly related in some way. This is where the Bell measurements come in.

Even though we might not know the exact state each qubit exists in, we can still determine how they are related. Bell measurements are obtained by recording specific measurements of each individual particle and using probability and statistical analysis in order to determine their correlation. This correlation will tell us exactly how one qubit relates to another; their differences, similarities, and unique properties – at least, in relation to each other.

Even though these measurements may not be superpositional, they don’t necessarily have to be. So long as the wavefunction on the actual particles themselves is allowed to continue after the Bell measurement tests have been performed, then all is well.

Bell measurements might feel abstract right now, but they are a surprisingly useful tool in processes like quantum teleportation.

Quantum Teleportation

This is a spooky one. Quantum teleportation isn’t exactly the teleportation of physical objects like in the movies. Instead it facilitates the transfer of quantum information from one location to another, instantly, without any physical transmission medium.

In classical communication, sending information from one location to another is straightforward. We can encode data in bits (0s and 1s), transmit it as electrical signals, light pulses, or radio waves, and decode it at the receiving end.

As we know, qubits can be in a state of 0, 1, or both 0 and 1 simultaneously. This presents a challenge: How can we transmit this superposition of information without destroying it?

The no-cloning theorem complicates matters further. It states that we cannot make an exact copy of an arbitrary unknown quantum state. So, we can’t simply measure the state of a qubit and send that information to recreate it elsewhere because measurement collapses the superposition, effectively destroying our qubit.

Here’s where quantum teleportation comes into play. It allows us to transmit the complete quantum state of one qubit to another qubit in a distant location, using two entangled qubits and classical communication. The steps are intricate but fascinating.

Firstly, the sender and receiver both share an entangled pair of qubits. Essentially, when one qubit is measured, the other will have a correlated state.

The sender will then perform Bell measurement to determine the correlation between each of the qubits in relation to each other.

The sender must then classically communicate the results of the Bell measurement to the receiver (no quantum methods are being used to transmit the message).

Given this Bell measurement data, the receiver can apply specific quantum gates to shift their qubit into the state of the sender regardless of whatever state each is in.

Since the Bell measurement provides data about how the pair is correlated, the receiver can essentially transform their qubit using that specialized data.

This might seem backhanded and not as exciting as the term “teleportation” alludes to, but it is teleportation: the receiver’s qubit is able to immediately take on the state of the sender’s qubit. This is, after all, instant transportation or teleportation of data.

Of course, the receiver and sender still need classical methods to communicate the Bell measurements, but the final transfer of data is different.

Keep in mind that this doesn’t violate quantum no-cloning because only the state is transferred.

Quantum Annealing

In the field of metallurgy, annealing refers to the cooling of a material in order to remove defects and help guide said material to a more stable state. In the quantum mechanical realm, annealing is similar: we can use it to gradually guide qubits away from errors and incorrect transformations while increasing their likelihood of taking on beneficial states and undergoing correct transformations. Essentially, it takes us from the beginning of a problem to the end, primarily in optimization-based problems. You might imagine optimization problems as finding two numbers whose sum is 26 and whose product is as small as possible. In this example, we optimize the product.

Over the course of quantum annealing, quantum gates will shift qubit states and favor low energy states, which is where we will usually mark our solution or objective. For example, if the problem is searching a database, we can flag our specific database item for search in a low energy state, so that it will be favored during annealing.

A key part of the annealing process is quantum tunneling. During annealing, the quantum system has the remarkable ability to “tunnel” through energy barriers, allowing it to escape local minima in the energy landscape and explore a broader solution space. This quantum tunneling effect enables quantum annealers to overcome challenges that often trap classical optimization algorithms, by quite literally passing through obstacles and barriers.

Annealing has the unique ability to dimensionally walk through incredibly complex problem spaces, making it an incredibly useful tactic for countless problems. We will look at some annealing gates later on.

Conclusion

Now that we have explored some critical phenomena, take a step back and reevaluate your mental quantum model. Remember that certainty is boring, and essentially worthless in the quantum realm. Nothing is for certain, and that opens up technically infinite possibilities in the field of quantum computing. Keep these concepts in mind for the next chapter, where we bring a more technical lense to the table.

CHAPTER 3

Quantum Gates

In classical computing, logic gates are the basic units responsible for actually processing and operating on binary data. Quantum gates are similar, but again are different in many ways. In order to better understand quantum gates, we will dissect the idea in a theoretical sense before looking at actual examples.

Imagine you're hosting a big party at your house. You want to make sure that only invited guests get in, while keeping out anyone who wasn't invited. To manage this, you hire a security guard who stands at the entrance.

Now, this security guard has a set of rules based on the invitations. Let's say you've invited your close friends and family, and you've also provided them with special passes. The security guard has two inputs: one for checking if the person has an invitation and another for whether they have the special pass.

Here's how the security guard's logic works:

1. Invitation Check: If a person shows up with an invitation, the guard's invitation check input is set to "1." If the person doesn't have an invitation, the input is set to "0."
2. Pass Check: If the person has the special pass, the guard's pass check input is set to "1." If the person doesn't have the pass, the input is set to "0."

Now, based on these inputs, the security guard follows a simple rule:

If the person has both an invitation and the special pass (inputs are “1” for both), the guard lets them in (output is “1”). If any of the inputs are “0” (no invitation or no pass), the guard denies entry (output is “0”).

This security guard is like a logic gate. It takes inputs (invitation and pass checks) and processes them based on a rule to produce an output, such as allowing or denying entry: 1 or 0. In the same way, logic gates in electronics take input signals and use predefined rules to produce an output signal. Just as the security guard’s decision determines whether someone can enter the party, logic gates’ decisions determine the behavior of digital circuits in computers and other electronic devices.

Dissociating from the analogy, applying logical gates takes some getting used to. We will start off with a few basic examples from classical computing, before delving into the complex realm of uncertain data.

OR Gate

Return 1 if either input is 1, else 0.

$$1 \text{ OR } 1 = 1$$

$$0 \text{ OR } 1 = 1$$

$$0 \text{ OR } 0 = 0$$

This gate is pretty self-explanatory: if either input is 1, the result is 1. Otherwise, the result is 0. But what if our input isn’t just 2 bits?

Larger Numbers

Classical gates applied to binary numbers with multiple bits follow this process:

$$x = 101$$

$$y = 110$$

$$x \text{ AND } y = (1 \text{ AND } 1) + (0 \text{ AND } 1) + (1 \text{ AND } 0)$$

$$= 100$$

In other words, the gate is applied to each positional pair of bits inside of larger binary numbers.

Quantum Gates

Now that we understand a rudimentary classical gate, we can examine some quantum ones. Similar to how classical gates were invented to simply modify the state of simple bits in varying ways, quantum gates were developed to modify the state and phase of qubits in order to prepare them for practical use. However, quantum gates can be multifaceted, addressing only certain aspects of the complex and multidimensional qubit. The quantum gates explained here may be abstract for the time being, but their importance and applications will become apparent as the book progresses. Remember that whenever you focus on an abstract and specific topic, it can't truly be understood without a view of the larger forces at play in the environment. Quantum computing is no different. As the book wraps up in the later chapters, this wider scope will be available to you. With that in mind, let's take a look at some quantum gates.

A Note About Gate Operations

Remember while reading this section that qubits don't have concrete values until we measure them. So even if a qubit happens to be 0 at a given time, and a quantum gate shifts that value to be 1, the qubit is still able to change back to being 0, so long as it does so before being measured. Quantum gates mostly operate on unknown data (since we can't tell the value of the qubit as it passes through the gate), so remember that the examples you will read are all just theoretical possibilities as to what would occur if a qubit with state X passes through a gate.

Pauli-X Gate

The Pauli-X gate takes an input of only 1 qubit and flips the quantum state around the x-axis. In other words, it swaps the values of α and β.

For example:

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$X|\psi\rangle \rightarrow \beta|0\rangle + \alpha|1\rangle$

In the Bloch sphere, you might visualize this gate by picturing the state vector of a subject qubit to rotate 180$\degree$ around a vertical axis. If the state vector was pointing directly north, it would face directly south after the Pauli-X gate is applied.

Pauli-Y Gate

The Pauli-Y gate is effectively a phase shift. $\alpha|0\rangle$ becomes $i\alpha|0\rangle$ and $\beta|1\rangle$ becomes $-i\beta|1\rangle$, effectively flipping the sign of β and making both α and β complex.

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$Y|\psi\rangle \rightarrow i\alpha|0\rangle - i\beta|1\rangle$

In the context of the Bloch sphere visualization, the Pauli-Y gate corresponds to a rotation of the qubit’s state vector either clockwise or counterclockwise. Additionally, this gate has the effect of altering the phase between α and β. As we know, both horizontal rotation and phase changes are critical factors in determining the qubit’s final state.

The Role of Qubit Value

In the gates to follow, you may wonder at some point “Why is β always singled out? Why never α?” In order to address this head-on, it is important to recognize what is known as a “controlled gate.” That is to say, a gate that will only operate on states of $|1\rangle$. Usually representing the probability amplitude of the $|1\rangle$ state, β will be operated on and essentially singled out. β, or the symbol representing any states of $|1\rangle$, will hereby be referred to as “active coefficients.”

Pauli-Z Gate

This controlled gate performs a phase flip operation, negating the value of β. For example:

)$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$Z|\psi\rangle \rightarrow \alpha|0\rangle - \beta|1\rangle$

This gate will cause the state vector to rotate about the z-axis (as the name suggests) in the Bloch sphere. If the vector was pointing straight ahead initially, it will point the straight backwards after the Pauli-Z gate is applied.

Hadamard Gate

The Hadamard gate was developed to initiate a superpositional state inside of a qubit. That is to say, it rotates the state vector to be in between $|0\rangle$ and $|1\rangle$ in the Bloch sphere (it is a controlled gate, which again means it will only apply to β in this situation).

The Hadamard gate accomplishes this task by first multiplying the value of $\frac{\alpha}{\sqrt{2}}$ against $|0\rangle$ and $|1\rangle$, and then adding the resultant to the value of $\frac{\beta}{\sqrt{2}}$ multiplied by $|0\rangle$ and $-|1\rangle$. Let’s walk through the process:

Given the definition of $|\psi\rangle$:

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

The Hadamard gate will act as follows:

$$H|\psi\rangle = \frac{\alpha(|0\rangle + |1\rangle)}{\sqrt{2}} + \frac{\beta(|0\rangle - |1\rangle)}{\sqrt{2}}$$

This gate might seem a bit strange, but it does indeed work every time, no matter the number. We can test this using concrete values for α and β:

$\alpha = 0.6$

$\beta = 0.8$

1. Act on the $|0\rangle$ state:

$$H(\alpha|0\rangle) = \frac{0.6(|0\rangle + |1\rangle)}{\sqrt{2}} = \frac{0.6}{\sqrt{2}}|0\rangle + \frac{0.6}{\sqrt{2}}|1\rangle$$

2. Act on the $|1\rangle$ state:

$$H(\beta|1\rangle) = \frac{0.8(|0\rangle - |1\rangle)}{\sqrt{2}} = \frac{0.8}{\sqrt{2}}|0\rangle - \frac{0.8}{\sqrt{2}}|1\rangle$$

3. Combine the calculated states:

$$H(\alpha|0\rangle) + H(\beta|1\rangle) = \frac{0.6}{\sqrt{2}}|0\rangle + \frac{0.6}{\sqrt{2}}|1\rangle + \frac{0.8}{\sqrt{2}}|0\rangle - \frac{0.8}{\sqrt{2}}|1\rangle$$

$$= \frac{1.4}{\sqrt{2}}|0\rangle - \frac{0.2}{\sqrt{2}}|1\rangle$$

Finally, α'(the new definition of α) becomes $\frac{1.4}{\sqrt{2}}$ and β' becomes $-\frac{0.2}{\sqrt{2}}$. We can also ensure the validity of these new variables using the familiar normalization condition.

$$\left|\frac{1.4}{\sqrt{2}}\right|^2 + \left|\frac{-0.2}{\sqrt{2}}\right|^2 = 1$$

$$\frac{1.96}{2} + \frac{0.04}{2} = \frac{2}{2} = 1$$

Thus, we can prove the function of the Hadamard gate.

CNOT Gate

This gate is pretty simple, although it extends the field of effect to 2 qubits. One qubit (usually the first one) is classified as a control qubit, and one is classified as the target qubit (usually the second one).

For example, say we have the following qubits (note that these are still qubits, because we haven’t measured them yet. We simply flipped whatever unknown state they were in):

$A = 0$

$B = 1$

Now, we run the gate.

$\text{CNOT}|A,B\rangle \rightarrow 01$

This gate returns 01, the same as the input, because the control qubit was set to 0. If control qubit is 1, however, the result changes.

$\text{CNOT}|B,A\rangle\rightarrow 11$

In this new case, B is in the control slot. Since the B qubit was measured as 1, the target qubit, 0, flips to also be 1.

Toffoli Gate

The toffoli gate, or the CCNOT gate, is the 3-qubit version of the CNOT gate. It works exactly the same as the CNOT gate, except there are 2 control qubits. For example

$A = 1$

$B = 1$

$C = 1$

$\text{CCNOT}|A,B,C\rangle \rightarrow 110$

Of course, if either A or B was 0 instead of 1 at the time of gate execution, then the qubit C would not flip but remain 1, if still in a state of 0.

Phase Gate(S Gate)

The phase gate has one purpose: shift the phase of the $|1\rangle$ probability by $\frac{\pi}{2}$ counterclockwise (a quarter turn). The phase gate, or S gate, does so by introducing the imaginary number i to the active coefficient.

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$S|\psi\rangle \ = \alpha|0\rangle + i\beta|1\rangle$

T Gate

Similar to the S gate, the T gate also shifts the $|1\rangle$ phase by $\frac{\pi}{4}$ counterclockwise and again leaving the $|0\rangle$ probability untouched. Instead of introducing a simple constant like i, the T gate multiplies the active coefficient by $e^{\frac{i\pi}{4}}$.

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$T|\psi\rangle \ = \alpha|0\rangle + e^{\frac{i\pi}{4}}\beta|1\rangle$

In terms of the visual representation of this gate in the Bloch sphere, the state vector will simply rotate counterclockwise $\frac{1}{8}$ of the full circle.

Controlled Phase Gate(Rk)

The Controlled Phase gate, or the Rk gate, is very similar to the phase and T gates in the sense that all of them shift the phase of the qubit passed through them. It is indeed a controlled gate, however, the “Controlled” term in the title of the gate means something else. It is unique from the gates we’ve examined because it takes in an input of more than just qubits. Instead, this gate also accepts a variable k, which is used to determine the phase shift applied to the target qubit. This parameter k is powerful because it lets us decide how exactly to modify the phase of our given qubit. The format these changes takes is as follows:

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

$k = 1$

$\theta = \frac{\pi}{2^k}$

You can think of k as controlling how many times π is cut in half. For example, cut it in half only 1 time and the gate will rotate $\frac{\pi}{2}$ radians(90$\degree$) counterclockwise. To apply our variables inside of the gate, all we need to do is call it and include θ in the same way we included $\frac{\pi}{4}$ in the T gate.

$Rk(\psi, k) = \alpha|0\rangle + e^{i\theta}\beta|1\rangle$

Even though we can't see θ being explicitly calculated in the calling of the gate, just remember how it is defined by our chosen value of k.

Multiple Qubits

Many of the gates we are looking at only have practical value when applied to more than 1 qubit, meaning that we need to learn how to expand our methods beyond the realm of a single qubit. With our entry in the multi-qubit realm comes a new notation. For every qubit represented in a given state $|\psi\rangle$, we will need 2^n probability amplitudes to represent every possible value of the state, where n is the number of qubits. For our example, we only want to look at 2 qubits, so we need $2^2 = 4$ probability amplitudes. Our 2 qubit state would look like this:

$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$

Notice how α and β represent the specific values $|00\rangle$ and $|01\rangle$ instead of $|0\rangle$ and $|1\rangle$. We also introduce γ and $\delta$to represent the remaining 2 states.

With this new notation comes a new normalization condition.

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$$

Again, since the quantum state needs to have a 100% chance of existing, the sum of all of the normalized probabilities needs to be equal to 100%, or 1.

However, there is another viable option for representing qubit states. Instead, we could keep individual qubits separate from each other until the end of the operation/gate we are performing and combine them. For example,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$\phi\rangle = \gamma|0\rangle + \delta|1\rangle$$

As mentioned, we can actually combine these qubit states using an interesting operator known as the tensor product.

Multiple Qubits and Active Coefficients

When more than 1 qubit is introduced, and states can become very complicated (having terms such as $|01\rangle$), the method of applying controlled gates becomes a bit different. If any of the values inside of the state ket are equal to 1, the control gate is activated. This means:

$$Rk(\alpha|01\rangle,0) = e^{i\pi}\alpha|01\rangle$$

and

$$Rk(\alpha|00\rangle,0) = \alpha|00\rangle$$

It is essentially the same idea as your normal 1 qubit controlled gates, except it is activated if any value of 1 is represented.

The Tensor Product

The tensor product is crucial to performing gates and other operations on multiple qubits, because it allows you to combine quantum states of multiple qubits to create a joint state that spans the entire composite system.

Let's look back on the qubits from the section above, $|\psi\rangle$ and $|\phi\rangle$. Both qubits can exist in a superposition of states $|0\rangle$ and $|1\rangle$. In order to describe a composite system of both qubits, we can employ the tensor product. It essentially combines the possibilities of each qubit's states. Mathematically, the tensor product of the qubits' states would be represented as:

$|\psi\rangle \otimes |\phi\rangle$

Here's how you calculate it:

If qubit ψ is in state $|0\rangle$ ($\alpha|0\rangle + \beta|1\rangle$) and qubit ϕ is in state $|1\rangle$ ($\gamma|0\rangle + \delta|1\rangle$), their combined state using the tensor product would be:

$|\psi\rangle \otimes |\phi\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle) = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$

If you remember the notation section well, this general format should be familiar to you. What is extremely important to remember about terms such as $\alpha\gamma$ and $\beta\delta$ is that they are still simple probability amplitudes. You can choose to represent them as $\alpha\delta$ or ϕ or even π if you like, it truly doesn't matter.

If you remember the FOIL method from algebra, the tensor product becomes a lot simpler. If not, just remember:

F - First

O - Outside

I - Inside

L - Last

You simply need these laws of multiplication to merge qubit states. You just need to push the states together in a way, so $\alpha|0\rangle \cdot \delta|1\rangle = \alpha\delta|01\rangle$. In the example above, we multiplied the first term in each factor ($\alpha|0\rangle$ and $\gamma|0\rangle$) to get $\alpha\gamma|00\rangle$. Then, we moved on to multiply the outside portions of each factor, then the inside, and so on.

This process can be extended to any number of qubits. For instance, with three qubits A, B, and C, the combined state $|A\rangle \otimes |B\rangle \otimes |C\rangle$ would result in eight possible states: $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$, and $|111\rangle$. The FOIL method becomes a bit more complicated, but it is doable.

In quantum computing, when applying gates to multiple qubits, the tensor product becomes essential. For example, if you want to apply a Hadamard gate to qubit ψ and an X gate to qubit ϕ, you would represent it as ($H \otimes X$) and apply it as follows:

$$(H \otimes X)(|\psi\rangle \otimes |\phi\rangle) = (H|\psi\rangle) \otimes X|\phi\rangle)$$

Similarly, if we have a 4-qubit state and want to apply, say, a Hadamard gate to it, we can do the following:

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

$$H(|\psi\rangle) = \alpha(\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle + |1\rangle)}{\sqrt{2}}) + \beta(\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}) + \gamma(\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle + |1\rangle)}{\sqrt{2}}) + \delta(\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle - |1\rangle)}{\sqrt{2}})$$

Thus, you can see the power of the tensor product. Of course, the tensor product can be expanded even to states like $|0101110101\rangle$ and so on.

The Identity Gate

The identity gate is the simplest gate in existence. It literally means just “ignore” the qubit.

$$I(|\psi\rangle) = |\psi\rangle$$

This may seem incredibly useless right now, but it can be helpful by allowing us to ignore certain aspects of the qubit state we don’t want to mess with.

Quantum Fourier Transform(QFT) Gate

The Quantum Fourier Transform gate is not really so much a gate itself, but a unique combination of other gates in order to highlight the frequency components of qubit states. To put it another way, the QFT gate converts information about the positions of the qubits into information about the frequencies(the phase angle θ and probability amplitudes α and β).

Imagine a group of people clapping their hands at different rhythms. Each person represents a different basis state($|0\rangle$ or $|1\rangle$) of a quantum system. The timing of each clap corresponds to the phase angle, where the strength of the clap represents the probability amplitudes α and β. The α clap is loud and clear, while the β clap is softer. The QFT is like analyzing the collective sound of all the claps, considering both the timing (phase angle) and the strength (probability amplitude) of each clap.

While the QFT doesn’t directly separate the claps, it helps identify the underlying rhythmic patterns and relationships between the claps, taking into account their timing and loudness. QFT orchestrates the claps in a way that highlights their harmonies and interactions, revealing the unique composition of the system’s performance. In qubit terms, its like dragging the chains that bond the probability amplitudes and the phase angle all the way to the surface; it reveals patterns rooted in the qubit states.

In essence, the QFT helps extract the phase information that describes how different basis states (claps) are “aligned” in terms of phase angles, while also considering their individual strengths (alpha and beta). This can be incredibly powerful for certain quantum algorithms that leverage both phase relationships and probability amplitudes to solve problems more efficiently.

This transformation is vital in order to solve problems that involve analyzing extracted phase information, such as factoring large numbers or solving certain mathematical equations.

QFT takes in n qubits, and unlike most other gates, won't have an influence on the Bloch sphere, at least not one we can discuss. Since it operates on multiple qubits, we wouldn't be able to accurately picture it in our 3-dimensional Bloch sphere due to the entangled states of the different qubits; the Bloch sphere is only designed for 1 qubit at a time.

With that being said, let's explore the inner workings of the QFT gate.

1. The Hadamard gate. The Hadamard gate is applied to each individual qubit, so n times to n qubits. To clarify, each qubit will only pass through the gate once. Remember, the Hadamard gate will put all n qubits in a state of superposition.
2. The phase gate. Each qubit will be applied a different variation of the phase gate; k will change based on the position of each individual qubit in the system of all n qubits. Essentially, the rightmost qubit will start with no controlled phase gates at all. The rightmost qubit will perform a controlled phase gate with a k of 0, which would be a rotation$\frac{\pi}{2}$ radians counterclockwise if it was alone in the Bloch sphere. The qubit to the immediate left of it would have a k of 1, and a θ of $\frac{\pi}{4}$. This pattern continues until the leftmost qubit has a k value of $n-1$.

Now, to begin the QFT process. First, we will define our qubit states, $|\psi\rangle$ and $|\phi\rangle$. We will keep them separate for now in order to ensure maximum simplicity:

$$|\psi\rangle \ = \alpha|0\rangle + \beta|1\rangle$$

$$|\phi\rangle = \gamma|0\rangle + \delta|1\rangle$$

To take it slow, we will only apply the Hadamard gate to the each qubit individually.

$$\mathrm{H}|\psi\rangle = \alpha\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\mathrm{H}|\phi\rangle = \gamma\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \delta\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Now, we can take a moment to combine them with the following tensor product operation:

$$|\tau\rangle = |\psi\rangle \otimes |\phi\rangle$$

$$|\tau\rangle = (\alpha \cdot \gamma)\frac{(|0\rangle + |1\rangle)(|0\rangle + |1\rangle)}{\sqrt{2} \cdot \sqrt{2}} + (\alpha \cdot \delta)\frac{(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)}{\sqrt{2} \cdot \sqrt{2}} + (\beta \cdot \gamma)\frac{(|0\rangle - |1\rangle)(|0\rangle + |1\rangle)}{\sqrt{2} \cdot \sqrt{2}} + (\beta \cdot \delta)\frac{(|0\rangle - |1\rangle)(|0\rangle - |1\rangle)}{\sqrt{2} \cdot \sqrt{2}}$$

Phew! That is a lot of math. Let's simplify, and our answer should look much nicer.

$$|\tau\rangle = \frac{\alpha\gamma}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) + \frac{\alpha\delta}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) + \frac{\beta\gamma}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) + \frac{\beta\delta}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)$$

Ok, still looks scary. How about this:

$$|\tau\rangle = \frac{\alpha\gamma + \alpha\delta + \beta\gamma + \beta\delta}{2}|00\rangle + \frac{\alpha\gamma - \alpha\delta + \beta\gamma - \beta\delta}{2}|01\rangle + \frac{\alpha\gamma + \alpha\delta - \beta\gamma - \beta\delta}{2}|10\rangle + \frac{\alpha\gamma - \alpha\delta - \beta\gamma + \beta\delta}{2}|11\rangle$$

It might not be any less scary, but at least it fits into one line! Let's break down what just happened.

First, we performed the standard Hadamard gate on $|\psi\rangle$ and $|\phi\rangle$. Next, we used the tensor product operator $\otimes$ with 2 qubit states($|\psi\rangle$ and $|\phi\rangle$), essentially FOILing them out and pushing the factors together; that's what 2. was: the raw product of the FOILing. In 3., we just simplified our answer down to something easier to write. We also put our answer in the 2-qubit composite state format we discussed earlier, of

$$|x\rangle = A|00\rangle + B|01\rangle + C|10\rangle + D|11\rangle.$$

In this format, $A = \frac{\alpha\gamma + \alpha\delta + \beta\gamma + \beta\delta}{2}|00\rangle$.

Now, let's use our composite state τ to complete the rest of the QFT gate. Hang in there!

If you're confused thus far, don't worry. Just go back and flip through the fuzzy concepts.

Time for the phase gates.

So, we will apply the phase gates starting with the rightmost qubit. K will begin at 0 and progress as we move left. Remember, since it is controlled, $|00\rangle$ will not be affected. Our final answer is:

$$(Rk(3) \otimes Rk(2) \otimes Rk(1) \otimes I)|\tau\rangle$$

$$= \frac{\alpha\gamma + \alpha\delta + \beta\gamma + \beta\delta}{2}|00\rangle + \frac{e^{\frac{i\pi}{4}}(\alpha\gamma - \alpha\delta + \beta\gamma - \beta\delta)}{2}|01\rangle + \frac{e^{\frac{i\pi}{2}}(\alpha\gamma + \alpha\delta - \beta\gamma - \beta\delta)}{2}|10\rangle$$
$$+ \frac{e^{i\pi}(\alpha\gamma - \alpha\delta - \beta\gamma + \beta\delta)}{2}|11\rangle$$

Feel free to try this example out with any placeholder numbers for α, β, γ, and δ in the original 2 qubits, but make sure they satisfy the respective qubit normalization conditions.

Custom Oracle Gate(Uf)

The oracle gate is essentially a marking tool; you might think of it as having the ability to flag specific states and encode classical information into quantum data. For example, in Grover's algorithm (a special search algorithm we look at later on), the state being searched for needs to be flagged by the oracle gate. This is so the computer is able to distinguish it from other states, rendering it useful to the algorithm. This process of "flagging" might seem intimidating, but it simply means to flip the sign of the selected state coefficient. This gate is not controlled, but does operate off of user-based input to determine which state to flag.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$Uf(|\psi\rangle,|0\rangle) = -\alpha|0\rangle + \beta|1\rangle$

As you can see, this gate is relatively straightforward and doesn't contain a large amount of theoretical knowledge. A quantum state goes in, parts of it are flipped to distinguish it from other values, all is good in the world.

Grover Diffusion Operator Gate(Uw)

The Grover diffusion operator gate, denoted by Uw, essentially raises the probability of the computer selecting the correct answer and lowers the probability of the computer selecting the wrong answer.

Pretend you're taking a multiple-choice quiz. It is very difficult, almost impossible. You don't know what any of the terms mean, and each of the answers look equally correct. Now, imagine you can request a hint. This hint will help you narrow down the choices; it will help you become more and more certain of the correctness of a certain answer while simultaneously becoming more certain of the incorrectness of other answers. Finally, after receiving enough hints, you're able to correctly select the right answer.

This hint mechanism is symbolic of the Grover diffusion operator gate. The gate is composed of 3 layers: an initial Hadamard gate, a custom oracle gate, and another Hadamard gate. This gate is particularly important in algorithms like Grover's(which is the gate's namesake), where narrowing the field of search is incredibly valuable.

Let's walk through an example.

$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$

We begin with the first Hadamard gate. If any of the following confuses you, flip back to refer to the section on Hadamard gates and tensor products.

$$|\psi\rangle = \frac{\alpha\gamma + \alpha\delta + \beta\gamma + \beta\delta}{2}|00\rangle + \frac{\alpha\gamma - \alpha\delta + \beta\gamma - \beta\delta}{2}|01\rangle + \frac{\alpha\gamma + \alpha\delta - \beta\gamma - \beta\delta}{2}|10\rangle + \frac{\alpha\gamma - \alpha\delta - \beta\gamma + \beta\delta}{2}|11\rangle$$

Then comes the custom oracle gate. We will flag the $|01\rangle$ state.

$$|\psi\rangle = \frac{\alpha\gamma + \alpha\delta + \beta\gamma + \beta\delta}{2}|00\rangle - \frac{\alpha\gamma - \alpha\delta + \beta\gamma - \beta\delta}{2}|01\rangle + \frac{\alpha\gamma + \alpha\delta - \beta\gamma - \beta\delta}{2}|10\rangle + \frac{\alpha\gamma - \alpha\delta - \beta\gamma + \beta\delta}{2}|11\rangle$$

Finally, the second Hadamard gate.

$$|\psi\rangle = \frac{\alpha\gamma + \alpha\delta - \beta\gamma + \beta\delta}{2}|00\rangle + \frac{\alpha\gamma + \alpha\delta + \beta\gamma - \beta\delta}{2}|10\rangle + \frac{-\alpha\gamma + \alpha\delta + \beta\gamma + \beta\delta}{2}|10\rangle + \frac{\alpha\gamma - \alpha\delta + \beta\gamma + \beta\delta}{2}|11\rangle$$

Now, this doesn't look like it really accomplished much, and to be honest, it didn't. The diffusion operator by itself only performed once will not have a very noticeable effect, especially if we don't use values for each of the probability amplitude coefficients. Not to mention, it was designed to work in collaboration with other gates. In the chapter about algorithms, however, we will be able to see the full force of the Grover diffusion operator in action.

Conclusion

Throughout the chapter, we have focused on some critical quantum logic gates and how they take effect. In the next chapter, we will explore the real-world applications of these abstract gates via specialized algorithms, which are similar to gates in the way that they follow a procedure to perform a task, but will likely be much more familiar as they aim to solve problems like factoring numbers and searching databases.

CHAPTER 4

The Quantum Algorithm

Thus far, we have covered a lot of ground in terms of how quantum computers function and the base of logic they stand on. But how can they be practically applied? What is their true purpose as of now, what problems are they truly spectacular at solving?

That is what this chapter aims to tackle: arguably the most important concept to emerge from quantum computers ⁻ the quantum algorithm. Quantum algorithms, conceptually similar to classical algorithms, are a list of steps implemented to solve a problem. The quantum algorithms leverage, of course, quantum gates, which is why they are quantum in the first place. As we know, phase is a critical part of quantum mechanics. We also know we can't really represent phase very easily in the same way we can represent state. For these reasons, we aren't able to walk through an implementation of every single algorithm with real examples of qubits as we did with gates, but we will still be able to gain a deep understanding of them in other ways. When possible, real qubit examples will be provided.

With the goal of understanding in mind, there is another classical computing concept that is worth understanding before we dive into the shifting realm of quantum algorithms.

Runtime Complexity

Runtime complexity is best understood in big-picture terms. To demonstrate this, we will start off with an example.

Let's say we have a list of items, and we are searching for a specific one. Since we are talking about the capabilities of classical computers(for now) we will need to scan each item one-by-one in order to determine whether it is the item we are searching for. Our list has a total of n elements, meaning we need a worst-case runtime of n in order to find our target item. This is worst-case because the item might be the first one we look it, but instead we assume it is the last one we look at, meaning we have to check all n elements to find it. The worst-case scenario is represented by the syntax $O(n)$, where n is the number of operations performed. Again, since we are searching a list of n values, we perform n operations.

Now, imagine we have two different lists. We need to search each of them in order to find matching items. For each item on the first list, we need to scan the entire second list in order to find if it has any matches. In this case, our runtime is $O(n^2)$, because for every value in n, we perform another n operations.

Finally, imagine we have the same two lists, but we are only looking for a value on each list. We will again assume that the value we are looking for is the very last one on the list, meaning we have to run through all n values in order to complete our goal. Since we are running the search operation 2 separate times, once on each list, our runtime is $O(2n)$. Here is where our focus on the bigger picture comes into play.

Essentially, as $n \rightarrow \infty$, n^2 is going to be a lot greater than $2n$. In fact, as values get larger and larger, n and $2n$ are going to look the same in comparison to n^2. For this reason, we disregard constant factors, meaning $O(2n) = O(n)$.

Also, if the runtime looks like $O(n + 1)$, the same big-picture mindset applies. As n grows larger and larger, the constant is dwarfed, meaning it is essentially equivalent to not existing at all. It also allows things to simplify much easier.

We will apply similar concepts to analyzing the runtime of some quantum algorithms to follow.

Deutsch's Algorithm

This algorithm is intuitive because it allows us to explore both possible outputs with only a single query, and even provides a conclusive response. It is also a very good introduction to the concept of quantum algorithms, because it demonstrates how abstract quantum gates can be used to accomplish a task.

Let's examine the problem.

We want to figure out whether an unknown function, also called a black box function, is constant or balanced.

A constant function is one in which all the outputs are the same, regardless of input. They are constant.

$$g(0) = g(1)$$

A balanced function is one where differing inputs equate to differing outputs.

$$g(1) \neq g(0)$$

g is our black box function. We will assume that the input (and output) domain is $\{0,1\}$.

Typically, classical computers can solve this problem by just checking if $g(0) = g(1)$. If true, then it's constant. Otherwise, it's balanced. Generally, this is a very reasonable and completely acceptable way to solve the problem. The quantum solution can actually speed this up further, although again, this really has no practical application.

The problem has an input of either 0 or 1, which can be represented by a single qubit. The output is also a binary classification; we only need one more qubit to represent the output. Both qubits will start out at $|0\rangle$ for the sake of simplicity.

$|\rho\rangle = |0\rangle$

$|\epsilon\rangle = |0\rangle$

As with most quantum algorithms, we need superposition. Why? Since we are trying to perform parallel operations simultaneously(check two conditions with a single function call), we need to create a superpositional state. To achieve superposition, we'll use the Hadamard gate, because we know it is designed to shift qubits into superposition.

$\mathrm{H}|\rho\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

$\mathrm{H}|\epsilon\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

This state should be pretty familiar, but it doesn't appear to do much for our problem.

We have a superpositional state, and now we need some way to call the function g and measure our results. To do this, we can utilize the idea of a black box oracle.

The black box oracle will flip the state of the second qubit, our output qubit, if and only if $g(\rho) = 1$. Our state might look like this:

$|\rho\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$

$|\epsilon\rangle = -\frac{|0\rangle + |1\rangle}{\sqrt{2}}$

That is, if $g(x) = 1$.

Then, we apply another Hadamard gate and measure our result of our first qubit. If you measure $|0\rangle$, the function is constant. If you measure $|1\rangle$, the function is balanced.

But something's missing. Just because $g(\rho) = 1$, the function doesn't necessarily have to be constant or balanced, right? This is where the quantum phenomena kick in. Quantum interference solves the problem for us. Again, since our qubits are superpositional, both values are called when the function g is run. If $g(0) = g(1)$, the probability amplitudes will cancel eachother out and leave a state of $|0\rangle$ behind. Otherwise, if the qubits aren't on the same wavelength, so to speak, they won't cancel out and we will get an existing state of $|1\rangle$.

In this way, we can effectively describe a function as balanced or constant using only 1 function call, in a swirling symphony of superposition.

Quantum Error Correction: The 3-Qubit Flip Code

To understand this algorithm, a few definitions first need clarification.

Physical qubit - a "real" qubit, susceptible to error, noise and decoherence

Logical qubit - a higher level abstraction of the qubit that is resistant to error by utilizing 3 qubits in order to mimic the behavior of a single qubit

Essentially, the logical qubit reduces error from noise or environmental factors by relying on 3 physical qubits and representing its state via the majority; If 2/3 of the qubits agree, then the shared state of those 2 qubits is most likely correct, thereby mimicking a "real" single qubit, free of error.

The three-qubit bit flip code itself is a simple quantum error correction code that can detect and correct errors caused by bit flips (flipping from 0 to 1 or vice versa) in a quantum system. It involves encoding a single logical qubit into three physical qubits using specific quantum gates. This encoding process introduces redundancy and allows for error detection and correction, enabling quantum computers to be more accurate and avoid falling victim to random noise or unexpected difficulties.

The algorithm begins with a single physical qubit that we want to protect from errors. For the sake of the example, we will imagine the qubit in a single state for now.

$|\psi\rangle = |\xi\rangle$

ξ represents either 0 or 1. The state is then transformed as follows.

$|0\rangle \rightarrow |000\rangle$

$|1\rangle \rightarrow |111\rangle$

Now, imagine we have a bit flip error. This may be due to quantum incoherence, random noise, or simply environmental factors.

$$|\psi\rangle = |010\rangle$$

The algorithm will measure the parity of the three qubits. In this case, the second qubit is observed to have an error, since it is the outlier from the other two.

To fix the error, all it takes is a simple CNOT gate using either of the other two qubits.

$\mathrm{CNOT}(|\psi\rangle) \rightarrow |000\rangle$

Thus, the error is corrected and the crisis averted.

Even though this example is simple, error correction using methods like this is critical to quantum computing. Error will always be an issue, but specialized algorithms like the 3-qubit flip code help to reduce this error to a manageable level.

Grover’s Algorithm

Invented in 1996 by Lov Grover, Grover’s algorithm is a quantum search algorithm, meaning it has the ability to search, say, a database, and find the correct item with incredible efficiency.

For the sake of the problem, we assume that the list is in no particular order. Typically, the fastest runtime achievable for classical computers given this problem is $O(n)$. Since the list is unsorted, we can't take advantage of any patterns to find our target item.

However, using quantum techniques, we can actually locate our targeted list item in only $O(\sqrt{n})$. If you think about the implications of that, it means we don't even need the time to look at every item in the list.

How can that be possible? We have truly no idea what else is in the list, only that we don't need to. This again reflects how quantum technology can move exponentially throughout the solution space, simply demolishing irrelevant items.

Let's take a look at the process of Grover's algorithm.

Firstly, the algorithm applies a Hadamard gate, creating an equal superposition of all states.

Then begins Grover's Iterations.

Grover's algorithm isn't just a mundane, fixed set of steps, but instead changes based on the list. Grover's iterations are as follows.

1. The oracle(Uf) gate. This gate "marks" the correct item by negating the probability amplitude coefficient in front of the correct state. The incorrect states are left untouched.
2. The Grover diffusion operator gate. This gate, designed to work in tandem with the oracle gate, decreases the probability of selecting the wrong answer and increases the probability of selecting the right answer by applying a Hadamard gate, a custom oracle gate to the correct item, and another Hadamard gate.

The above Grover Iteration is applied to the list approximately $\sqrt{n}$ times. After said iterations, measuring the qubit will most likely result in the correct state being selected.

You might imagine the process of Grover’s algorithm in the context of a piece of paper.

Imagine you have a piece of paper covered in small dots. There are dozens of them, covering both sides of the paper. The dot you are searching for is written in pen, while the other dots are written in pencil. Of course, you could find the dot by simply erasing each dot, one by one, until you find the dot you can’t erase.

But what if you had a special kind of quantum eraser, that could erase all the dots at once?

Shifting back to the field of quantum computing, and our “magic eraser” is Grover’s algorithm. By manipulating virtually the entire solution space at once, Grover’s algorithm is capable of searching for an item in an unsorted list faster than any other method known to man.

Although quantum computers are still in their primitive stages, Grover’s algorithm is proven to be extremely effective and will have broad-reaching effects as technology develops.

Shor’s Algorithm

A somewhat controversial algorithm(as will be addressed later), Shor’s can exponentially reduce the time needed to factor large numbers into their prime components.

Factoring large numbers is generally considered a very hefty task, at least to classical computers. The difficulty lies in the exponential time increase needed as the number to factor grows larger.

However, as we know, quantum computers tend to be able to run exponentially expanding tasks with no problem, and this sentiment holds true with Shor’s algorithm. The algorithm breaks down into 3 steps.

Let’s assume N is the digit we are going to factor, and a is an integer randomly chosen between 1 and $\mathrm{N}-1$.

Firstly, we apply a Hadamard gate in order to create a superposition of all possible states. Then, the problem of factoring large numbers is mapped to the problem of finding the period of a modular exponentiation function. The function $a^x \bmod N$ is performed for varying values of x. The goal is to find the smallest value of x so that

$$a^x \bmod N = 1$$

is true. This equation is the period we are looking for. Classically, this is the time-consuming part. We can avoid this barrier by leveraging quantum interference to test multiple values of x in parallel to solve the equation, with the QFT gate.

This gate reveals the periodicity hidden beneath the surface of the quantum state, essentially exposing the underlying patterns of the qubit. It transforms the state into a superposition of all possible periods of the modular exponentiation function.

All there is left to do is measure the state and process the result, so long as the measured state is even. If it’s odd, then the process may need to be repeated.

Again, the trick to this algorithm is that it allows us to leverage multiple values of x in parallel, solving an otherwise arduous task with effortless grace.

Quantum Phase Estimation(QPE)

We need to define some terms in order to understand how the QPE transformation.

Unitary operators are mathematical transformations that play a crucial role in quantum mechanics and quantum computing. All of the gates we have looked at so far are unitary. Unitary operators are characterized by several essential properties:

1. Preservation of Quantum State: Unitary operators do not fundamentally alter the quantum state they operate on. This means that key quantum properties such as superposition

and entanglement must be preserved after applying a unitary operator. In simpler terms, the resulting state should still exhibit the fundamental characteristics of quantum mechanics.

2. Conservation of Probability: One of the most fundamental principles in quantum mechanics is that probabilities must add up to 100%. Unitary operators adhere to this principle by ensuring that the normalization condition is maintained. In other words, the sum of the probabilities of all possible outcomes remains equal to 1.
3. Preservation of Probability Amplitudes: Unitary operators do not modify the probability amplitude coefficients of quantum states. For instance, if a quantum state is represented as a linear combination of basis states with coefficients α and β, a unitary operator will not change their values. These coefficients continue to describe the probability amplitudes of the quantum state.
4. Orthogonality Preservation: Unitary operators also maintain the orthogonality of quantum states. Orthogonality is a mathematical property where different quantum states are perpendicular or orthogonal to each other in a complex vector space. Applying a unitary operator does not change the relative angles between quantum states.

The quantum phase estimation algorithm can estimate the phase of a unitary operator such as a gate; in doing so it provides a clear example of how quantum computers outperform classical ones.

The algorithm first initializes two quantum registers(working and ancillary). The working register is where the phase information is encoded, and the ancillary register is used to perform quantum operations. A Hadamard gate is subsequently applied to the working register, followed by a controlled-U operation.

A controlled-U operation is essentially a CNOT gate, but if the the control qubit is measured as $|1\rangle$, it performs a given unitary operation instead of simply flipping the sign. For example, you could apply an oracle to the target qubit if the control qubit is $|1\rangle$.

In this case the ancillary register is used as input for the controlled-U operation, each state in the register determining whether those in the working register are applied the specified unitary operator.

Finally, the inverse QFT gate(QFT^{-1}) is applied. The inverse QFT gate undoes the QFT gate. For example, if $f(0) = 1$, then $f^{-1}(1) = 0$, and $f(f^{-1}(f(0))) = f(0) = 1$.

Let’s break down what just happened.

By applying a controlled-U operation, the QPE algorithm entangles the working register with the phase information encoded in the unitary operator. This entanglement is a key quantum phenomenon that allows for exponentially more efficient phase estimation compared to classical methods. The inverse QFT then is responsible for extracting the phase information from the amplitudes of the working register. It’s works in such a way that the probability of measuring a particular state is proportional to the corresponding phase. When we measure the working register after the inverse QFT operation, we obtain an estimate of the phase of the unitary operator. The precision of the estimate depends on the number of qubits used in the working register. More qubits lead to a more precise estimate.

The output may appear to be simply a sequence of bits, but will reveal specific quantum phase when decoded. Being able to obtain the phase is critical in many contexts as we will soon observe.

Quantum Amplitude Estimation(QAE)

The QAE algorithm is designed to estimate the numerical values for probability amplitude coefficients. Given the state below, the algorithm would attempt to find reasonable estimates of α and β.

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

This is accomplished through a series of amplitude amplification gates. It is a similar process as in Grover’s algorithm but with a twist.

Assume we want to estimate β. A typical Hadamard gate is applied to the state in order to create equal superposition.

Two iterations of the QPE gate are performed. Then an inverse QPE gate is applied. This strange flip-flop action seems pointless, but actually serves an important purpose.

In the initial QPE step, we apply the gate to estimate the phase angle (θ) associated with the target quantum state, meaning we only apply the gate to β. This phase angle is directly related to the variable we want to estimate. QPE gives us an estimate of θ, but it’s not yet a direct estimate of β. The inverse itself is critical because it lets us flip the state of the phase angle which allows us to correctly estimate β.

After the initial QPE, we apply a gate (e.g., a rotation gate) to invert the phase. This means we effectively flip the sign of the phase angle. This operation doesn’t change the magnitude of β, but inverts its sign. Now, we have a pretty good idea of $-\beta$.

We repeat the process of off-again-on-again QPE gates to the now inverted phase to get $-\theta$ and θ, both of which allude to the value of β. Again, all of these gates are being directly applied to β and whatever it is attached to.

To get there though, we need to apply the Grover diffusion operator gate that was explained earlier. This is designed to raise the probability of measuring the proper result while simultaneously decreasing the probability of measuring the improper result.

In the current context the diffusion operator chooses a value for β based on the phase angle. Keep in mind that this entire process, from QPE through diffusion operators, must be repeated a number of times in order to maximize accuracy.

Finally, we measure the result. An output of $|1\rangle$ means β is more approximately $|1\rangle$ than $|0\rangle$. A result of $|0\rangle$ would have the opposite implications.

Conclusion

Quantum algorithms may be primitive for now, but understand them at their early stage will prove incredibly advantageous as society advances. For example, being on the ground floor of the creation of ARPANET(1969), the first version of the internet, definitely inspired those who worked on it and understood it as the technology progressed. Their strong foundational knowledge grew alongside technology, giving them a unique perspective and a variety of opportunities to innovate. In fact, the lead manager of ARPANET, Bob Taylor, went on to invent the personal computer, inspiring graphical user interfaces in the Apple Lisa and Macintosh.

Carrying this unique and illuminating information about quantum computing here in the early days will benefit you in some way in the future.

CHAPTER 5

Quantum Neural Networks

At the time of this book, the world is in the “Golden Age” of artificial intelligence and neural networks. ChatGPT, DALL-E, Bard, and others are now gradually integrating themselves into our daily lives. But imagine a marriage between the quantum phenomena we’ve unraveled and the neural networks that drive many of today’s AI advancements. This union gives birth to a new breed of computational power: Quantum Neural Networks.

Before we dive into the intricacies of Quantum Neural Networks (QNNs), let’s understand why we need them. Classical neural networks have made remarkable strides in solving complex problems, from image recognition to natural language processing. However, there are computational tasks so demanding that even the most advanced classical computers struggle to provide quick and efficient solutions, such as extremely complex problems that may require complex, high-dimensional nonlinearities instead of nonlinear approximations. Here is where quantum computing steps in.

Quantum computers, as we’ve learned, harness the power of qubits and quantum gates to perform certain calculations exponentially faster than classical computers. QNNs leverage this quantum advantage to enhance the capabilities of neural networks, particularly in areas where classical machines fall short.

Let’s first understand classical neural networks.

Classical Neural Networks

While it may seem like magic to most (and still is to passionate researchers), neural networks are just math. Designed to simulate human minds, they consist of a layer of interconnected nodes(neurons) that perform calculations. These neurons, also called nodes, each have internal parameters called weights and biases.

For example, if I want to plan a trip, the most important factors would be cost and having a good time. An unimportant (but still present!) factor might be the weather. In this scenario, a neural network would place heavier weights on the variables of cost and fun, signifying their importance. The opposite is executed on the weather.

Now that we understand weights, it is critical to imagine them mathematically.

$y = mx + b$

Remember this from middle school? It's back! Your 6th grade teacher was right; you will be using this throughout your life.

Mathematically, m might be the weight of a certain variable, and b might be the bias.

So the weight amplifies the value of a variable, it extends its importance, and the bias adds a constant level of significance regardless of the value of x.

Essentially, m depends on x while b is independent, and both represent significance. Higher weights and biases = more important factors.

Now, in these individual nodes, weights and biases might not mean much. In fact, they mean nothing alone. But when you connect the layers and form the "brain" of the program, everything comes together.

You can't think with just a single neuron, it takes many of them all working together. The same is true for neural networks, which are really just artificial simulations of the mind.

But our neurons also have "activation" meaning some sort of process that incoming signals must deal with. These "activation functions" exist in machine learning as well. Activation functions decide whether the output of each neuron is significant enough to contribute to the system. The activation function introduces nonlinearity into the model, allowing neural networks to learn from and be applied to everything from videos and images to textual or audible conversations.

Whenever anything is passed into a neural network, from images to words, it is all transformed into numbers. We won’t get into the specifics but further independent research is strongly encouraged.

Now that we know what a neural network is, let’s talk about training. All we really know right now is a giant fancy math figure acts like a virtual brain. But how exactly does it function like one?

The specifics of the learning process involve heavy calculus, but essentially the network is fed testing data and updates its weights using derivatives, by shifting incorrect weights gradually towards being correct until eventually neural networks function as they should, and are ready to be tested on data they haven’t seen before.

It is also important not to overtrain models, because they become too used to the data. Models will adapt to the data they are being trained on, and will become too specific in their weights to be able to generalize their training to accurate predictions, leading to large amounts of error. The best models are trained on data enough times to be able to accurately recognize the data, but not so many times that they can’t apply the patterns to other data.

Quantum Neural Networks(QNNs)

At their core, Quantum Neural Networks are similar to classical neural networks. They consist of layers of interconnected nodes, where each node performs computations. However, what sets QNNs apart are the quantum gates used for these computations.

In a classical neural network, nodes perform mathematical operations like activations. In a QNN, these nodes are quantum nodes, each representing a qubit or a quantum state. These quantum nodes are manipulated using quantum gates, such as the Hadamard gate and CNOT gate.

The choice of quantum gates in a QNN is critical. Commonly used gates include:

1. Hadamard Gate (H): Since it places qubits in a superposition of states, it enables them to explore multiple possibilities simultaneously.
2. RX, RY, and RZ Gates: These gates allow rotations of qubits around the X, Y, and Z axes of the Bloch sphere, modifying their quantum states which could benefit training in a multidimensional sense.
3. CNOT Gate: The Controlled-NOT gate entangles qubits, meaning they can create complex quantum correlations between them that might aid in training.

Training a QNN involves adjusting the parameters of quantum gates to minimize a loss function, just like in classical neural networks. The difference lies in how these parameter adjustments are computed.

We will examine a number of different quantum machine learning models in order to better understand how the dynamic field of quantum computing can be hybridized with classical neural networks in order to optimize efficiency and accuracy.

Quantum Convolutional Neural Networks(QCNNs)

Classical convolutional neural networks, or CNNs, are responsible for interpreting image-based data. CNNs can perform everything from image classification to object detection, all by using pixel data. As we know, our computer screen is made up of tiny little dots called pixels, which can represent different colors. However pixels are not only physical but virtual as well. Each pixel is digitally represented by 3 numbers, each number having an inclusive range from [0,255].

This is the RGB (Red Green Blue) system and it serves to encode the color of specific entities into your computer so the screen knows how to display them. For example, a completely red pixel might look like this:

$(255,0,0)$

As you can see, the red component is on full charge, while the green and blue components are not being utilized. If all values are 0, the pixel is black, and if all are 255, the pixel is white.

But the interesting part of pixels is their ability to mix colors. For example, if I want my pixel to be neon yellow, I would have:

$(255, 240, 31)$

Except quantum machines tend to view things in grayscale, meaning the computer might extract an "intensity" value from this pixel, deeming it brighter or dimmer based on the overall value of the pixel. This is because encoding in color takes a lot of data, and it is currently under heavy research in the quantum computing community. For now, we will look at grayscale(colorless) examples for QNNs.

Classical CNNs actually process these pixels using convolutions, or layers designed to detect patterns in the vast array of pixels they are given. This quest for the pattern in images is transferred into quantum CNNs(QCNNs) as well, although of course there are some key differences.

The quantum convolutional neural network training process begins with the encoding of the input image into quantum states. In a classical CNN, each pixel's intensity is a numerical value. In QCNNs, we represent this pixel information using qubits (so we can perform gates on the data). Each qubit corresponds to a specific pixel or feature of the image.

For example, consider a grayscale image of 8x8 pixels. One common encoding method is amplitude encoding, where the amplitude of a qubit represents the pixel value. A representative quantum state might look like this:

$|00\rangle \rightarrow \text{Pixel intensity value } 0$

At the start of the actual convolutions, a Hadamard gate is applied to create superposition. This allows the QCNN to explore multiple feature combinations simultaneously. Then, Rk (controlled phase) gates are applied to either suppress or emphasize certain features in the target image based on their relative positions. For example, the prime target of the image might be emphasized, while the grainy fuzz in the background might be suppressed due to a lack of relevance.

The CNOT gate is then applied to introduce entanglement between certain qubits, letting the computer capture spatial relationships between the features. This serves to give the computer a more dimensional view of the image, letting it see how different pixels affect other pixels.

Finally, more entanglement gates are applied to enable the QCNN to capture complex feature dependencies.

It’s like letting the computer zoom out and see the effect of different pixels on the rest of the image.

Although every scenario is different based on the needs and goals of each model, most QCNNs follow this general format.

After this fascinating process occurs, QCNNs then go through pooling and subsampling to reduce dimensionality, a highly mathematical operation that essential just means “get to the important stuff.”

Of course this process is repeated countless times through different layers through the network. Remember we are talking about a very well connected virtual organism.

What makes QCNNs truly remarkable, though, is their ability to examine the big picture and almost mimic human eyes. When us humans look at an image, we don’t just see the floating balloons (or only any single pixel of them), but the way the setting sun complements the balloons as they drift higher and higher into the sky.

These networks have the unique ability to gain an incredible perspective, making them critical to the field of quantum computing and image processing due to their impressive ability to analyze images in a new fashion.

Quantum Natural Language Processing Networks(QNLP)

Natural language processing is the art of analyzing and classifying textual input. Just as with CNNs, the input is transformed to fit a number. But how do we turn text into numbers?

The answer is strange. We cut all the sentences into words, and after removing stopwords like “and,” “the,” and “is,” we reduce all forms to their roots. Thus, “running” becomes “run”. Every word becomes lowercase as well.

“I was running over to the store” → “run”, “store”

From here, there are a variety of methods used to encode the meaning of words, including by matrix representation, dense vector representation, or weighing words.

After this, text is fed to the neural networks in their simplified and numericized forms, and the input is treated just like any other (for the basic NLP models we will quantumly transform).

Trained models are able to classify text, answer questions about text, identify key information such as date and location, and even generate their own text.

QNLP algorithms again adapt the gist of this process. The textual data is still encoded, but in qubits instead of bits. One interesting concept is that multiple words can be encoded into few qubits because of their superpositional nature; qubits can represent more than one word simultaneously.

Such encoding happens with the use of Hadamard gates, for the obvious superpositional initialization, and the specialized quantum embedding gate, which encodes the meanings and contexts of words into quantum states.

What differentiates QNLP models from classical models is once again their superpositional capabilities: being able to have fluidity, almost a shapeless form, gives an immense power to predictive models because it lets them completely and totally adapt to the complex and multidimensional nature of text.

Textual analysis is generally considered to be incredibly nuanced and sophisticated: to understand even a simple sentence you need to know the definitions of all the terms involved, and even more so the implications behind specific terms. There is context embedded in every situation, not to mention humor and tone. Using quantum techniques in processing text allows computers to expand their "brains" dimensionally and adapt totally to the self-changing and volatile nature of language.

Quantum General Adversarial Networks(QGANs)

As with QNLP models and QCNNs before them, understanding of GANs is required to understand QGANs.

The GAN, or generative adversarial network, is a type of generative model that learns how to produce an accurate result through repeated competition. The GAN is actually two networks, a generator and discriminator. The generator is tasked with replicating images, or certain styles of text, or whatnot, and tries to create sophisticated enough predictions to trick the discriminator, which has the job of discerning real data from fake data generated by the discriminator.

The discriminator is surprisingly only fed random noise, in an attempt to make the outcome "random" and therefore less predictable.

Gradually, the generator and discriminator both improve, still challenging each other and pushing each other to be more accurate. It's a classic game of cat and mouse, where both parties improve and try to beat the other.

At the end of the training process, if all went well, the generator will be very well trained to generate realistic images or whatever you fed it; the discriminator is typically discarded.

In classical GANs, the generator creates data point by point, following a probability distribution. In QGANs superposition plays a critical role in allowing for an expanded solution space with added captured complexity, but the probabilistic nature of quantum techniques are a factor as well.

Since probabilities are kind of the “jam” for quantum techniques, the data fed to quantum models is likely more “random” than if it were generated by a classical algorithm, meaning the data generated by quantum generators is likely to be more realistic, as it relies on true randomness to create faux data. Classical computers, on the other hand, are forced to use repetitive and predictable pseudorandom generator algorithms in order to generate noise.

Not to mention, quantum computers provide a significant speedup when compared with the lengthy training time accompanied by classical GAN methods.

Essentially, the introduction of quantum methods to generative adversarial networks allows for countless possibilities and improvements in every way, including runtime, accuracy, and complexity depth.

Conclusion

While we have been deprived of the nitty-gritty details of how quantum neural networks function according to each individual step of the process and intricacies that follow, this chapter has hopefully given you an idea of the power that quantum computing has when merged with the virtual human mind. The power of AI is just being realized in this very moment, much less the power of quantum neural networks. It is, without a doubt, more powerful and shadowy than any other enigma. The explorations into the dark depths of quantum AI have been brief and shallow, but even then we have struck gold.

We will finish this chapter with one though-provoking concept: when made quantum, many neural networks become more accurate; they are able to accomplish their tasks better, able to simulate the human mind better. What does that say about our brains and how they work?

CHAPTER 6

Quantum Hardware

It goes without saying that quantum computing would be nothing without quantum hardware; qubits are just an idea, if that, without quantum computers to stand on.

For these reasons, quantum hardware needs no introduction.

Types of Quantum Systems

Superconducting Qubits

Superconducting qubits are typically considered the most widely used method of quantum computing, and also generally agreed upon as the simplest. Superconducting qubits are a remarkable manifestation of quantum mechanics that take advantage of the phenomenon of superconductivity. Superconductivity is a state of matter where certain materials, when cooled to extremely low temperatures (-459$\degree$, close to the lowest possible temperature attainably in the universe), can conduct electrical current with zero resistance. This means that if you run electricity through a wire, it would hypothetically remain there forever: no energy would be lost. This property opens the door to creating qubits, because energy retention is crucial the quantum computers in order to avoid decoherence.

At the heart of a superconducting qubit lies the Josephson junction, a tiny device that connects two superconducting materials separated by a thin insulating barrier. This junction allows for the flow of a supercurrent without any energy loss. Superconducting qubits also incorporate inductors, which store magnetic energy. These inductors, together with the Josephson junction, create an electrical circuit that behaves quantum mechanically. Capacitance elements store electrical energy as an electric field between two conductive plates. In superconducting qubits, capacitance is typically formed by the geometric arrangement of components.

There are several types of superconducting qubits, but the main takeaway from them is that they function by taking advantage of superconductivity. Each qubit is a physical device, an electrical circuit, that behaves quantum mechanically.

The qubits themselves can be measured using microwave pulses, which indicates either $|0\rangle$ or $|1\rangle$.

As mentioned previously, superconducting qubit-based quantum computers are very popular, especially with companies like IBM and Google, who have even developed their own.

Trapped Ion Qubits

Trapped ion qubits are a fascinating and highly promising quantum computing architecture that relies on the precise control and manipulation of individual ions—electrically charged atoms. These quantum systems literally are based in the electrical nature of real ions themselves. This approach has gained significant attention due to its remarkable coherence times and the potential scalability.

Trapped ion qubits operate based on the quantum states of individual ions. These ions are isolated from their environment and held in place using electromagnetic fields, hence the name “trapped ion qubits”. The entire setup usually consists of the following key components: the ion trap, the ions themselves, the laser system, and the electrodes.

The ion trap is the heart of the system. It creates a stable environment for ions to be captured and manipulated. There are various types of ion traps, such as linear Paul traps or Penning traps, each with its unique characteristics. The details differ as you approach the problem in closer examination, but that’s a question for professional physicists. The ion trap keeps the qubits in a stable place for us to use them.

The ions themselves. Typically, a specific isotope of an element is chosen for qubit operations. For example, in the case of ytterbium, the Yb+ ion is used. These ions have very well-defined energy levels (internal states), which is important for encoding quantum information.

High-precision laser systems are then employed to cool the ions down to their quantum ground state (near absolute zero temperature) and to manipulate their quantum states. This is because when things cool down, they slow down as well, letting us observe the ions without them dashing all over the place. The lasers are also used for state manipulation.

Finally, a set of precisely controlled electrodes generates the electromagnetic fields necessary for trapping and manipulating ions. These electrodes create the trapping potentials and facilitate the movement of ions within the trap. Essentially, the electrodes are able to influence the position and entrapment of the ions.

The qubits are able to measured by fluorescence. A certain laser illuminates a qubit, and its fluorescence gives away its state.

Trapped ion computers are especially good at maintaining coherence, which allows for longer calculation time and lets qubits keep their states. Overall, trapped ion computers are a very solid option for quantum systems.

Photonic Qubits

Photons, the fundamental particles that form light itself, make excellent qubits. They can transmit data over long distances and manipulate information with ease and are even resistant to some types of quantum error. Photons also naturally exist in superposition, where they may have multiple polarizations at once.

Polarization describes the orientations of oscillating electric and magnetic fields that propagate through space. Simply, polarization tells us whether the waves are horizontal or vertical.

Photons can also be easily entangled, and have very high coherence.

This is because photons tend to interact very little with their surroundings, reducing the possibility of contamination or corruption from unwanted environmental noise.

This type of quantum computer can be constructed starting with a mere laser. Simply split the laser into two entangled particles, and you can perform operations. Typically, operations tend to be carried out with waveplates, beam splitters, and phase shifters. Photonic qubits are typically measured using devices called avalanche photodiodes, which convert incoming photons into electrical current and measuring that. From this output, a definitive state determination can be made about a given qubit.

Photonic computers are unique because you don't need millions of dollars and an incredibly low-temperature environment in order to run them. The drawback is scalability: photonic systems are very difficult to grow. A small number of qubits works just fine, but handling much more becomes a significant challenge with photonic computers, much more so than systems like trapped ion and superconductive.

This is because photons are very delicate carriers, and can easily be lost as they pass through gates. Photons also don't play very nicely with each other; they have been known to interact weakly and refrain from unifying with other qubits.

While photonic qubits might not be the best choice for building the most high-level systems, they are great examples of how nature continues the pattern of embedding quantum mechanics in each of its creations.

Topological Qubits

Topological physics is a field that studies the properties of materials that remain unchanged even after continuous deformations. In short, if something is topological, it is able to remain in a stable, constant state even through a multitude of changes. An example is the Möbius strip, a 2d surface with only one edge and one side. No matter how you stretch or bend the strip, you will never distort the fact it only has one edge and side.

Topological qubits are no different in their efforts to avoid change: they can cut down on error and noise with ease, remaining stable and continuing quantum calculations. Data encoded into topological qubits will most likely be strongly rooted there.

But to understand topological qubits, we must first grasp the concept of topological insulators. Topological insulators are materials that conduct electricity on their surface but act as insulators in their interior. What makes these materials special is their topological order, which endows them with unique electronic properties, which forms a state of matter that is impervious to outside changes, allowing for a robust and steady state.

The keys to topological qubits are the Majorana zero modes, which facilitate superposition and introduce quantum mechanics to topological physics.

Majoranas are extremely stable and resistant to errors. They're like super-reliable switches that can handle a lot of noise and still give you the correct answer. This is essential for building powerful and error-free quantum computers.

The output for these types of quantum computers is measured through measured charge and statistics, which indicate certain quantum states.

Currently, Microsoft is one of the main players in the topological quantum computer industry.

They do have a lot of potential to be very powerful in the future, especially due to their resistant and unchanging nature yet still being able to hold superposition.

Diamond NV Centers

One of the most intriguing and promising quantum systems that has captured the attention of researchers is the use of diamond nitrogen-vacancy (NV) centers. Diamonds, known for their dazzling beauty, hold a secret within their crystal lattice that makes them valuable for quantum computing and sensing applications. This secret is the NV center, a naturally occurring defect in the diamond's structure that behaves as a unique qubit.

The NV center's quantum states primarily involve the spin of electrons associated with the nitrogen atom's atomic nucleus. They have two fundamental spin states: the ground state $|\uparrow\rangle$ and the excited state $|\downarrow\rangle$, akin to the binary representation of classical bits. The NV center can achieve superposition by adopting both the ground and excited states, making it eligible as a qubit.

Diamond NV center-based qubits can be passed through gates via the use of finely tuned microwave or radiofrequency pulses, which speak directly to the core of the diamond. These qubits can be measured with specialized lasers, which activate fluorescence not dissimilar to that given off by qubits in trapped ion computers.

Of course, this technique isn't without its challenges. While they have a decently long coherence time, scaling up the number of qubits is difficult, as with photonic computers. However, researchers may be able to make diamond NV center-based computing work in hybrid with other quantum computing methods.

Silicon Qubits

Silicon qubits capitalize on the unique properties of silicon, a material abundant in classical computing and electronics. In these quantum systems, qubits are encoded in the quantum states of specific atoms embedded within a silicon substrate(slices of highly purified silicon). Phosphorus atoms, in particular, have garnered significant attention.

Silicon qubits primarily utilize the electronic and nuclear spins of phosphorus atoms as the basis for quantum information encoding. These spins serve as the quantum analogs of the classical "0" and "1" bits. The electron spin represents the qubit, while the nuclear spin is employed for qubit readout and manipulation. Think of the electron spin as literally the direction that the electron spins in, and the nuclear spin as the orientation of the spin-generated magnetic field relative to an external magnetic field.

To build such a quantum computer, first doping is necessary. Doping is the process in which atoms of phosphorus are introduced to the silicon substrate. Said atoms are then isolated, and each is initially given an electron spin of either up or down.

Magnetic fields and microwave radiation are commonly used for qubit manipulation, such as the same old gates and algorithms we have viewed in the rest of the book.

Qubit state is typically derived by the interaction between the qubit's electron spin and nearby nuclear spins.

In short, silicon-based quantum computers are great because they are scalable, very coherent, and even compatible with existing classical computers(which are also based on silicon technology).

Quantum Circuits

Quantum circuits are the bedrock upon which quantum algorithms and computations are built. They are the quantum analogs of classical digital circuits, but instead of classical bits, quantum circuits manipulate quantum bits or qubits.

With the knowledge we have accumulated so far, quantum circuits should be pretty simple to grasp. Just as classical circuits apply certain steps or operations to bits, quantum circuits apply certain steps or operations to qubits. Algorithms like Shor's or Grover's were all implemented in quantum circuits, and thus they are the backbone of the quantum computing environment.

Typically, quantum circuits begin by initializing qubits in a certain state(usually $|0\rangle$) and perform gates. Controlled gates then create entanglement. The qubits are finally measured in order to produce the output of the gate, and classically transformed in order to represent usable data (look up “ASCII binary to text” for more information).

So while quantum circuits are a vital component of quantum computers, they are very intuitive and easy to grasp.

Conclusion

This chapter wasn’t designed to outline every single functional characteristic of quantum computers; that would have taken hundreds more pages. Instead, the main takeaways should be a general understanding of each type of computer, and the patterns present in each of the methods.

Quantum hardware such as that discussed is all in its infancy, and most of the people reading this book won’t get a physical glimpse at it for a long time, if ever.

But understanding how and why this technology exists holds power; however far off the future is, it still exists.

CHAPTER 7

Ethical Considerations

Quantum computing is a transformative technology that promises to revolutionize fields ranging from cryptography and drug discovery to optimization and artificial intelligence.

Like any disruptive new technology before it, such as the printing press, the cellphone, and Manhattan project, quantum computing will bring about major change to our lives. With this tidal wave of new problem-solving methods follows ethical concerns and responsibilities, which will be thoroughly covered in this chapter.

Cryptography and Security

Cryptographic algorithms defend systems ranging everywhere from popular video games and online movies to the FBI and the nuclear arsenal of the United States of America. Arguably, cryptography is one of the most vital computerized inventions every developed.

And it has developed at a slow and steady pace. The first cryptographic “algorithm” was invented in the 16th century: a simple cipher in which the letters of the alphabet were rearranged in order to hide the meaning of sentences. Since then, of course, technology has developed to revolve around mathematics and hidden keys.

Take RSA (Rivest-Shamir-Adleman) encryption, a protocol used to protect a variety of content, including US military secrets. The algorithm works using a set of keys, public and private. The public key is used to encrypt the data, while the private key is used to decrypt the data.

The whole algorithm rests upon one seemingly unbreakable security measure: RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers.

See a problem yet?

RSA made the (reasonable at the time) assumption that *nothing* could divide huge numbers into prime factors.

Enter Shor's algorithm.

Although no quantum machines can perform it to such a scale today, technology is increasing steadily. One day, it will be a reality that the might of RSA, AES, SHA-256 – will be defeated by a clever scientist and a couple of spinning particles.

So is quantum computing ethical? Can we continue to employ this type of computing in good faith if it is going to do so much harm?

Whether or not quantum computing is ethical is a pure matter of opinion.

Quantum computing is likely as ethical as classical computers were when they were first introduced. Just as we fear for our security with potential quantum-powered encryption-breaking, most people were afraid of their computers when they first arrived on the scene in the late 20th century.

Even though quantum computers are threatening to disrupt our digital security, the extent of their effect will simply be just that: disruption. Quantum computers can be used to shatter encryption, yes, but they can also build it. Using quantum phenomena such as no-cloning, as discussed earlier, can be used to fashion a sort of quantum cryptographic system in which quantum keys cannot be cloned or copied by anyone else, not even individuals with quantum computers.

So in short, no, the quantum revolution won't majorly harm the world in terms of digital security. No nuclear weapons are going to be hijacked, and our military documents are safe. Widespread adoption of quantum techniques will definitely require most organizations and governments to redesign their security, but quantum computers shouldn't leave a lasting mark on digital security and data protection.

Social Media

As quantum algorithms advance, they have the potential to unlock unprecedented insights from data, but this power also brings ethical dilemmas regarding the analysis of individuals' personal information and the responsible use of data.

We know quantum computers are truly superior at naturally detecting relationships between two things, linking them even when they aren't directly connected(entanglement). When applied to larger-scale data analytics, quantum methods can potentially expose more reliable information about an individual or group than classical algorithms.

Social media addiction affects an estimated population of more than 200 million individuals worldwide, according to a University of Michigan study. Classical neural networks are already great at predicting what kind of content people want to interact with on social media: who they want to "friend," what they want to watch, and what they want to buy. But what happens when this predictive technology is too good, when serious addiction is a growing threat?

This is a question of knowledge and consent. It is like smoking: we know the effects of smoking on the human body and lungs, but if you know and accept the risks, its up to you whether to smoke or not. With the quantum power of in-depth analysis beyond our shallow-dimensional "flat" neural networks, quantum predictive techniques are likely to greatly exceed their current classical capacity. But again, this doesn't necessarily need to be a bad thing. So long as companies and organizations are honest and open about the predictive modeling techniques they use to analyze user data, and each user is informed accurately and gives informed consent, quantum predictive techniques in social media would generally be as ethical as an informed and willing human being smoking a cigarette.

Of course, this begs an argument over whether allowing people to choose in the first place is ethical, or adopting a tactic that could potentially be very addictive to users, which is again of opinion.

So in summary, it depends largely on whether you personally accept the idea of potentially very accurate predicative models to be applied to social media and entertainment platforms. This is a choice that each individual must make for themselves, and it is critical to be informed of both sides of the argument.

Social Inequality

The rapid development of quantum computing technology has the potential to exacerbate existing social inequalities, raising significant ethical dilemmas. As we delve into this complex issue, we encounter various dimensions of social inequality impacted by quantum computing.

Widespread access to technology is already a growing issue. The term “digital divide,” a widely used phrase, aptly describes the problem at hand with classical computers.

The issue is that highly developed countries have the resources to obtain classical computers and internet, while developing countries might not have as easy access to them. Classical computer/internet access was initially concentrated in only the most privileged individuals. Of course, there is a large-scale effort to change this, with entities such as the Alliance for Affordable Internet(A4AI) working to bring these technologies to previously ignored parts of the world.

When examined in a quantum lens, this problem is magnified tenfold. Quantum computers require extremely specific environmental conditions and hardware in order to operate, which won’t be easily replicable without a vast array of resources. Quantum computers are at risk of developing as a luxury only for the wealthiest of the wealthy, with a 50 qubit quantum machine costing around $50,000,000.

So how can we introduce quantum computing to socially unequal situations? How can we be ethical in developing technologies that are essentially closed off to lower-income users inherently?

Quantum computing, as discussed earlier, is still in its infancy. For now, excluding high-level research institutions, quantum computing is closed off to everyone, exuberantly wealthy or not. This gives us time to prepare for the eventual quantum inequality, to begin educating the public.

Quantum simulators are a great way to do. Simulators can be accessed online, and some particularly good ones are STAQ, QuEST, and Qrack, which are all free to use. Keep in mind that users need to know some classical programming languages before using this software. However, simulator software still needs to be run on computers, meaning it is generally difficult to access in low-income situations, not to mention having the luxury to learn programming languages. The only way to fix this is continued education and outreach programs.

As quantum technology advances, it is important to continue programs to educate developing countries and supply them with computers, which in turn allow users access to the world of both classical and quantum computing, at least at this stage.

We must not allow the barrier of ignorance to fall between us. If we can accomplish that, quantum computing will grow and be connected to everyone, not just the powerful and extremely wealthy.

In short, preventing low-income individuals and groups from falling behind in classical computing will prevent them from falling behind in terms of quantum computing, allowing everyone to be able to have a part in the future of quantum systems. Now more than ever, it is imperative to spread technology and educate people of all social and economic classes.

Quantum Supremacy

The Cold War was a dark period in the histories of the US and Russia following World War II, in which both countries rushed to develop as many nuclear weapons as they could. This “arms race” lead to heightened political tensions in the regions, as well as unhealthy competition and aggression.

The same may occur in quantum computing unless action is taken against it.

Countries like the US and China are both racing to develop their quantum programs faster than the other. Currently, the US is ahead in quantum computer development and quantum sensing capabilities, but China is leading in quantum communications and total number of quantum technology patents.

If this quantum competition is allowed to continue, the field of quantum computing may be marked forever by a vicious rivalry between the US and China.

If a rivalry does indeed develop, tensions may also lead the countries involved to misuse of quantum computers, turning them into weapons instead of tools.

We have covered how quantum computers may be misused in security purposes, but governments won't have ample time to defend against such misuse in the case of a quantum race.

So how do we prevent this from happening?

One viable solution is the formation of international agreements and regulations in order to maintain the ethics of quantum computing. This may not be a very popular solution, because it will restrict potential military use on both sides of the agreement, but it will overall benefit society at large by preventing abuse and potentially dangerous acts perpetrated using quantum machines.

In short, by forming an ethical contract between competing nations, we can keep the competition civil and light, not “life-or-death.”

Job Disruptions

The advent of quantum computing heralds a new era of technology with vast potential, but like any disruptive innovation, it raises ethical concerns. One such concern centers around job disruptions caused by the automation and optimization capabilities of quantum computers.

Will quantum computers pose a threat to our jobs?

With every new technological advancement comes fear of economic impact. However, quantum computing will probably open up more career opportunities than it will close.

We will still need janitors, garbage collectors, accountants, and teachers. But with quantum technology, careers as quantum hardware and software engineers will blossom, followed by quantum algorithm and policy jobs.

Quantum computing will also significantly enhance existing industries. With quantum computers, material engineers simulate molecular interactions, aiding in materials discovery and development. Medical researchers can accelerate drug discovery and protein folding simulations. Logistics specialists can optimize supply chain operations and logistics, potentially altering how goods are distributed.

Quantum computing won’t take jobs from human beings; it will integrate into these disciplines and harmonize with workers.

However, this isn’t always the case.

Quantum neural networks have the potential to outperform classical ones, which is a completely different story. Classical neural networks are already encroaching into data entry and processing jobs, as well as administrative tasks, telemarketing and customer support jobs, and even financial bookkeeping jobs. Their quantum counterparts will likely perform even better, and will get more and more cost-effective as quantum technology develops and becomes standardized.

Overall, the future is somewhat cloudy, but it would appear that the influx of new, quantum-enhanced jobs will likely overtake the deficit of quantum replaced jobs to some degree, with a probably positive net effect on the job market and economy.

Conclusion

While the natural power of quantum computers may inspire a large amount of fear and apprehension, remember to focus on the positive. All problems have a solution, and at such an early stage, optimal outcomes are definitely attainable. As with all technology, ethical considerations are critical even in the early stages of development, and questions about the ethical viability of quantum systems will persist throughout the rest of your life.

Remember, new things are scary, but humanity will adapt and grow alongside quantum machines.

CHAPTER 8

The Future

Quantum computing will continue to be shaped and evolved, as does every field, but there are some critical factors to consider. How will quantum computing develop? Where is it headed right now? How far from true quantum adaptation are we as a society?

All of these are valid questions, and are the object of this final chapter.

Scaling Quantum Hardware

A large focus of experts in the field is the number of qubits each machine has. Quantum computers can’t be useful until they have the capacity to store larger amounts of data.

Comparable to this dilemma, performing 5,012,214 x 43,239,009 would be faster on a calculator than by hand, if the particular calculator can store numbers that large.

Right now, the current qubit world record is held by IBM with a grand total of 433 qubits. By the end of the year IBM hopes to release a 1,000 qubit processor. Scaling quantum computers is just about the most important focus in the field right now, and we won’t see really any practical use emerge until we expand quantum technology to be able to store our problems.

Another problem to consider is the physical limitations of quantum computers. The ultimate goal for these machines is to standardize them and use them in professional environments, which won’t really be possible until the invention of the room-temperature superconductor.

As previously discussed, the superconducting quantum computers are the most widely used and adopted. But since they have to be wrapped in giant machines and restricted heavily by temperature controls, the prices and convenience of the devices has increased significantly.

That’s not to say quantum computers will ever be cheap, but with the invention of room-temperature superconductors and the removal of the bulky refrigerators that protect them, quantum computing will be much more available for larger scale production and standardization.

In the future, we might see a steady increase in quantum capabilities until the release of the room-temperature superconductor, when quantum computing will erupt as a field. It is inevitable that such superconductors will be developed, though whether they exist during our lifetimes are a matter of speculation.

Error Correction and Fault Tolerance

Quantum computers are incredibly error prone, as you know. Error codes are expected to continue to develop and protect quantum computers from environmental noise and disruptions, until their accuracy approaches 100%.

Right now, quantum computers are on shaky ground. To even have a chance of attaining superposition, much less a usable superpositional state, you need a fortune and a team of researchers. Eventually, as quantum computing evolves, error codes and fault tolerance will stabilize and be perfected (or improved to a close enough state) to where quantum computers are just as reliable as classical computers.

For some specialized algorithms not discussed in the book, such as QAOA(quantum approximate optimization) algorithms, their performances will be enhanced.

QAOA algorithms estimate combinatorial optimization problems faster than classical algorithms, an example being the Knapsack problem.

Imagine you have a knapsack. You want to walk away with the most money you can. However, the knapsack can only hold a certain weight. Each item to collect has a certain weight and price, and only a certain combination of them results in the highest price yield.

The QAOA algorithm estimates this peak price; it gives an approximate “guess” of what that number might be. For example if I had a knapsack with a maximum price yield of 10, a reasonable approximation given by QAOA methods might be anywhere from 8-12(including decimals).

But as error correction codes continue to develop and become more effective, quantum approximation algorithms will approach perfection and eventually be “good enough” to the point where an approximation will most always be useful enough to reach a definitive result. The range that the approximations will fall around the correct value will shrink and become insignificant, and quantum computers will have reached the limit of precision.

Quantum Cloud Services

Quantum computing is becoming increasingly accessible via online methods such as quantum simulators. The future will likely see an increase in these simulators, as quantum methods for problem solving become more and more popular.

While such simulators aren’t the “real deal,” and can only accurately simulate few qubits at the moment, advancements will give simulators the edge over even supercomputers.

Even if it is resource-heavy to simulate quantum systems, it may be a worthy tradeoff to accomplish tasks and solve problems that much faster. At any rate, it is likely going to be more efficient than buying an actual quantum computer, at least for the time being.

Quantum cloud computing will also makes its way into business, but much more gradually. Businesses can certainly benefit from quantum technology, as can every industry, but it isn’t as necessary a commodity as things like internet and email servers.

It isn’t common in most businesses to need to search an unordered list in $O(\sqrt{n})$ time instead of $O(n)$ time, for example.

Rarely is it that Fortune 500 companies require the prime factors of 3,912,563,834.

So excluding research-based companies and highly technical companies, most businesses won't require much of cloud-based quantum computing for a while.

Hybrid Quantum-Classical Systems

Strictly quantum computers are useful, but not in an everyday sense. In fact, quantum computers are arguable expensive paperweights without specific problems and contexts to perform.

Classical computers are the opposite; they are generally useful in everything, but lack in specific areas with specific contexts.

The intersection of quantum and classical technology could bridge the gap; it could allow users to check their emails, watch internet videos, and enable quantum-enhanced calculations simultaneously.

Very few of these systems exist, and their designs are as formless as water for now. This type of technology is definitely possible, even if less imagined due to the issue of quantum computing's everyday practicality.

Quantum Algorithms and Applications

The development of quantum algorithms and the bridge between theoretical and practical will continue to be prevalent in the spread of quantum machines.

Most new applications will again only result from an upgrade in quantum hardware.

However, assuming we have achieved such quantum hardware, we would be able to create a sort of secure quantum networking system using quantum encryption methods, quantum repeaters, and even quantum satellites.

Applying the principles of quantum mechanics unbounded by hardware constraints will introduce a new wave of unimaginably powerful technologies. Every human endeavor would be enhanced.

Quantum computers could accurately simulate complex molecular structures and chemical reactions, revolutionizing drug discovery and materials design.

Quantum computers could perform highly detailed climate simulations, enabling precise long-term weather forecasts and climate change modeling.

Quantum computers could decipher the intricate process of protein folding, leading to breakthroughs in understanding diseases and designing targeted therapies.

Quantum computing could optimize traffic flow in real time, reducing congestion and emissions in urban areas.

Quantum-enhanced sensors and navigation systems could improve spacecraft trajectory calculations and enable precise interplanetary travel.

Quantum simulations could facilitate the study of artificial life forms and evolutionary processes in ways that classical computers could never replicate.

Again, all of these speculations should be considered in the context of "quantum systems *could*," meaning there is a lot riding on the development of hardware to reach that stage and the specific nature of how these things work.

What we can be sure of is the quantum potential to solve each and every one of these problems, or at least enhance their solutions.

Conclusion

As we conclude our exploration into the future of quantum computing, it is clear that we stand on the precipice of a transformative era in human history.

Quantum computing, once a realm of theoretical musings and laboratory experiments, is now on the verge of practical utility.

In the not-so-distant future, quantum computers with thousands, millions, or even billions of qubits will unravel complex problems in seconds that would stymie classical supercomputers for millennia. They will decipher the fundamental building blocks of our universe, revolutionize cryptography, and design groundbreaking materials.

So in conclusion, the future of quantum computing is both blindingly bright (but also dim at the same time).

FINAL THOUGHTS

Throughout the course of this book, we have explored the mysterious phenomena that govern the quantum world, the abstraction of quantum gates and algorithms, and even quantum hardware and ethical considerations. Finally, after looking forward to the next frontier of quantum technology, our voyage is done.

Each type of quantum system brings its own unique advantages and challenges, contributing to the vibrant tapestry of quantum technology.

But the path forward is not without its challenges. Quantum error correction, fault tolerance, and ethical considerations loom large on the horizon. Yet, the quantum community remains undeterred, working tirelessly to overcome these obstacles.

The future of quantum computing is a symphony of quantum bits, a dazzling display of entanglement, and an exploration of computational frontiers. It holds the promise of solving problems previously deemed insurmountable, of securing communications in an increasingly interconnected world, and of unraveling the mysteries of the quantum universe.

As we conclude this book and peer into the quantum future, one thing is abundantly clear: quantum computing is not a destination but a journey—an exhilarating expedition into uncharted territory. The quantum revolution is underway, and its impact on science, technology, and society will be nothing short of ultimate.