# Chapter 35 – Quantum Computing and ML

## Through two doors at once

### 35.1 – Introduction

This introduction is will probably sound like a pharmaceutical advertisement. In that it will contain a lot of disclaimers. But unlike those ads, I'll try not to run through them at lightning speed, hushed tone, and end with saying the main side effect of this brand-new wonder drug is the very thing the wonder drug is supposed to cure. First disclaimer, I am not a physicist. And this chapter deals with the absolute frontiers of physics. There will be many simplifcations, but I have included sufficient reading materials in the reference section, geared at many different target audiences if you want to read more. Second, we will have to gloss over many over the intricate details make up the entire field of quantum theory. Again, please refer to the additional reading materials at the end. Third, if you are unfamiliar with the perculiar reality of our universe that quantum theory seems to suggest, I will ask you to take a leap of faith. Don’t necessarily dismiss an experimental obersvation, theory, or mathematical proof because it violates your entire world view. The foundation of quantum theory came through careful experimentation, mathematical modeling**,** and reasoning. To be honest, I think none of the scientist that developed these complex set of theories were keen on having to admit that the immtual laws of physics put forward by Newton, Maxwell and many others were indeed, not immutual. Something was off. Fourth, even with all the experimental findings it is still an incomplete theory and our best guess**,** and our best interpretation of what reality truly is. Or what reality means to us. And what our place is in it. To me, science isn’t about the truth. It’s about getting better and better in using our scientific tools to approximate our observations. A theory is only better than the one it replaces because it shows fewer discrepancies between what is and what the theory it should be. Newton’s laws of gravity are extremely accurate. At the scale of things of earth. But for things much larger, that’s where Einstein needed to revise the orginal Newtonian ideas. The same holds for quantum theory. The closer we looked, the more the stuff we thought were understood now, tomorrow and forever, stopped behaving in that way. All the above disclaimers are about physics. There are two more that have to then also be stated about using the mathematical framework of quantum theory to do intelligent work: to compute things. First, that the framework allows us to do it theorectically and that it provides the tools we need that conventional computing cannot provide. And of course, that it allows us to do it in practice.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | **Figure 35.1** – Like Quantum Computing, a lot of people didn’t quite think the Beatles were going to be big. |  |

### Right here and now, and with our own hands. Build machines with those abilities that work. You might not be convinved. You are not in bad company, there are many that think this field of quantum computing is a nothing more than a pipe dream. Highly regarded scientists who are experts in the field Quantum Computing is try to harness. But I am with on board because the gains, rewards and change it offers are worth giving it a shot. And I can quote many famous scientists that said no to ideas that later did became reality and established fact. But I prefer this one (although it is disputed it was ever said) by Dick Rowe, the manager of the record label Decca: ‘Guitar groups are on their way out, Mr. Epstein….’. The band in question. The Beatles trying to sign their first record deal.

|  |  |  |
| --- | --- | --- |
|  | A close-up of a machine  Description automatically generated with low confidence |  |
|  | **Figure 35.2** – Google’s Quantum Computer Prototype |  |

### 35.2 – Classical versus Quantum Physics

Imagine that infamous (properly non-existing) apple that fell on sir Isaac Newtown’s head, providing him with the insights into the laws of motion and gravity. Depending on the size of that apple, Sir Isaac must have felt some pain from the resulting impact, big wig or not. Now, what if the apple had been twice as big. Or twice as heavy. Twice the pain, probably. (we are ignoring a lot of neuroscience here, but that’s ok for the point we are trying to get across). What if the apple had been x times heavier? Let’s just put forth a simple equation. P (that’s pain measured by the amount of profenities Sir Isaac mutters between two clocktower strikes) = W (the weight of the apple). With x being a completely abtritrary number, but at least between two reasonable numbers given what we know about apples. Let’s say somewhere between 8 and 24 ounces. So 7 is out, and so is 24.1. Fine. But that still leaves an infinite number of weights in between. Because we are measuring values in real numbers. 10.1 works. And so does 13.34395834598234038420938420394823 ounces. One continuous value (weight) directly transformed into another continuous value. And that is the underlying assumption of the classical physical laws. F = MA always holds, no matter what the numbers are. [MORE]

### 35.3 – The Uncertainity Principle

|  |  |  |
| --- | --- | --- |
|  | A cartoon of a person holding a box  Description automatically generated |  |
|  | **Figure 35.3** - Why some people probably shouldn't own cats. |  |

### 35.4 – The Wave function, Superposition, Collapse, and a Cat

However, the quantum model and its associated uncertainties hint at a rather strange reality. You probably will have heard about Schrodinger's cat. It is classic thought experiment, in which Edwin Schrodinger imagined a cat in closed box, alive and well when we put it in there (although perhaps a bit miffed from having been manhandled into a box).

|  |  |  |
| --- | --- | --- |
|  | A cat in a box  Description automatically generated with medium confidence |  |
|  | **Figure 35.4** - Our theoretical experimental setup. A cat in a closed boxed, a photon emitting device and an angled mirror that has a 50% chance of reflecting or letting a photon pass through |  |

We now fire a single photon at a half-reflecting mirror. This mirror will either reflect the photon into a different direction, or let the photon pass unimpeded in its original direction, with both events being equally likely and completely up to chance. The diverted photon will hit a detector (A) which will trigger the release of a poison in the box, killing the cat. The unimpeded photon will activate another detector (B). We cannot see whether either switch has been triggered without a measurement of one of the two detectors. So, what is the true state of the cat AFTER we have fired the photon but BEFORE we open the box? Is the cat dead or alive?

|  |  |  |
| --- | --- | --- |
|  | A picture containing clock, clipart, illustration, design  Description automatically generated |  |
|  | **Figure 35.5** - Without any direct observation, we don't know whether the cat was in fact killed or not. So, is it alive? Or dead? Or both? |  |

Surely, the cat is one or the other. Truth is, we really don't know for sure, when it comes to things the size of cats that are comprised of trillions and trillions of atoms. However, experimental observations at the subatomic level (photons, electrons) have shown that up until we do a measurement of the outcome by looking at either detector, the photon is said to have been in superposition. It existed with equal probabilities in both realities. When we do look at the detectors, this superposition is said to collapse into a single outcome. The probability of observing any outcome depends on the precise superposition state of the photon. In our example, the 50% reflecting mirror we give us a 50% change of measuring the photon at A and B, but these probabilities can be manipulated by experimental manipulation. So, how does simply observing something and measuring it change its physical reality? At the subatomic level, there is no truly passive and objective way of measuring anything. To get a read on the state of particle, you need to see how it interacts with another particle. This interaction inherently changes the particle under investigation. How do these strange physical realities translate into a new way of computing? It turns out that these states of sub-atomic particles can be induced using specialized hardware.

|  |  |  |
| --- | --- | --- |
|  | A picture containing clipart, cartoon, illustration, design  Description automatically generated |  |
|  | **Figure 35.6** - Once we observe the output of one of the two sensors, we force the dual reality into one: the cat is dead, or it is not. As if our observation forces reality to decide between different alternatives. |  |

**35.5 – Entanglement**

Another oddity arising from careful measurement and the mathematical models is the phenomenon of entanglement,

|  |  |  |
| --- | --- | --- |
|  | A picture containing illustration, cartoon, art  Description automatically generated with medium confidence |  |
|  | **Figure 35.7** - Two particles can be entangled, meaning that once we measure one , forcing it to collapse back into a single state, the other particle does as well. Even if they are light years apart. Einstein referred to this to 'Spooky action at a distance' and he wasn't keen on it. |  |

### 35.6 – Quantum Computing

Quantum computing is the use of these before mentioned phenomena: superposition and entanglement to perform computations. Exploiting these phenomena gives quantum computers the ability to represent, iterate over as well as search through extremely large state spaces with relatively little computing resources, such as physical memory and circuitry. As such, the applications of quantum computing focus on improving algorithms that show exponential growth in complexity in classic computing environments. Key to the understanding of the quantum world, and therefore quantum computing, is that at a subatomic level, the state (described by its attributes spin, velocity, position, phase) of particle is probabilistic until it interacts with an observer by way of measurement. This is not an easy reality to accept, but careful and repeated measurements of subatomic particles continue to be in line with this uncertainty principle. But what it needs to work is a new way of representing data: Qubits.

|  |  |  |
| --- | --- | --- |
|  | A picture containing circle, diagram, text, line  Description automatically generated |  |
|  | **Figure 35.8** - Unlike a bit, a Qubit is a bit more involved in its representation. The state of a Qubit isn't either 0 or 1, rather it is a probability of being 0 and a probability of being 1 that add up to 1. |  |

### 35.7 – From Bits to Qubits

In classical computing, we use bits to represent our data and code. A classical bit is 0 or 1, so the number of different states we can represent with n bits is 2n. Classical computing involves passing bits through logic gates. Our data (bit strings) is operated on, one at a time, by a set of instructions, performing basic binary operations on them. However, a classical register of n bits can only take on one value at a time. At the time of writing, an average MacBook Pro runs at about 1 teraflop, making it capable of performing one trillion (1012) floating-point operations per second. Even at that incredible speed, to test every possible permutation of a 64-bit code, we would need have to run through 2e19 different combination. At 1 teraflop, that takes 20000000 seconds, or roughly 231 days. Quantum computers, on the other hand, use qubits. The qubit is to quantum computing what a bit is to classical computing. However, since the qubit comes with several interesting properties that cleverly designed algorithms can harness to yield extremely powerful tools. All of this because a qubit has an actual physical reality: a sub-atomic particle. If we harness and perfect our ability to create sub-atomic particles and put them in certain states, we can use them as Qubits, and use the properties or superposition and entanglement to perform certain computations instanteniously, computations that will take a classical computer more time than the universe has and will exist to perform.

### 35.8 – Understanding the Qubit - Superposition

Qubits can represent numerous possible combinations of 0 and 1at the same time. This ability to simultaneously be in multiple states is called superposition. To put qubits into superposition, researchers manipulate them using precision lasers or microwave beams. Thanks to this counterintuitive phenomenon, a quantum computer with several qubits can crunch through a vast number of potential outcomes simultaneously. The result of a calculation emerges only once the qubits are measured, which immediately causes their quantum state to collapse to either 0 or 1. More formally, the state of a single qubit can be described by a two-dimensional column vector of unit norm, that is, the magnitude squared of its entries must sum to 1. This vector, called the quantum state vector (1), holds all the information needed to describe the one-qubit quantum system. And it represents a qubit state if the vector satisfies (2). Pictured on the right are examples of valid quantum state vectors representing qubits. (3) and (4) are qubits in a classical state, taking on the value 0 or 1. (5) and (6) represent the qubit in a state of superposition of equal probabilities. Now that we know how to represent a qubit, we can gain some intuition for what these states represent by discussing the concept of measurement. A measurement corresponds to the informal idea of looking at a qubit, which immediately collapses the quantum state to one of the two classical states. When a qubit given by the quantum state vector (a) is measured we obtain the outcome 0 with probability and 1 with probabilities (7) and (8).

### 35.9 – Understanding the Qubit - Entanglement

We can also generate pairs of qubits that are entangled, which means the two members of a pair exist in a single quantum state. Changing the state of one qubit will instantaneously change the state of the other in a predictable way. Quantum computers harness entangled qubits in a kind of quantum daisy chain to work their magic. [MORE]

**35.10 – The big deal**

We are faced with ever larger data sets potentially providing new insights. In response, machine learning algorithms are becoming ever more complex themselves, with current deep belief networks having adjustable parameters that number in the millions and more. But this comes as a high price: the curse of dimensionality. Adding new dimensions or columns to our data creates an exponential growth in the number of possible states in a state space. And at some point, we will reach the physical limits of our hardware in terms of computation speed and storage capacity. However, quantum computing effectively overcomes this problem by turning any linear increase in the dimensionality of a state space, into a linear increase in the number of qubits required. What was once considered to be an intractable problem to solve for even a large group of supercomputers working in unison, now becomes reality with only a handful of Qubits. At the core, classical and quantum computing use the same paradigm: both operate on packages of data (bit strings) that can represent any type of information of interests (strings, numbers, images). However, qubit greatly expand the number of possible operations we can simultaneously perform on it. In addition, the unique properties of superposition and entanglement (that they inherit from their physical reality) allow for a new set of superpowers: logical operations not possible within the classical computing paradigm. Combined it reveals the true strength and promise of quantum computing: its ability to represent and evaluate many different states of a system at once. If we can prove our quantum computer can solve a known algorithmic problem faster than our conventional computing methods, we have achieved quantum supremacy. Such Quantum Supremacy was recently demonstrated by the Quantum team at Google. This demonstrates the real-world viability of quantum computing beyond an academic interest.

### 35.11 – Logical Gates as the building blocks of Quantum Computers

The machines' ability to speed up calculations using specially designed quantum algorithms is why there's so much buzz about their potential. However, in quantum computing, we still operate on data by comparing and transforming their states through logical gates. Only now, these gates have additional properties that utilize the potential of the Qubits being both in superposition and entangled. Below I detail briefly three such gates we will use into very simple application of Quantum Computing below.

### 35.12 – Controlled-NOT gate

The controlled-NOT gate (also C-NOT or CNOT) is a quantum gate that is an essential component in the construction of a quantum computer. It can be used to entangle and disentangle qubits and their states. Imagine Qubits x and y. a CNOT gate mainly functions as a conventional XOR gate, the output of the gate is 1 if x and y are not the same, 0 if they are. However, a CNOT gate has Qubit x operate on Qubit y, where y represents the traditional XOR output. Qubit x itself is retained and does not change. This allows the operation to be reversed, something not possible with a conventional XOR gate.

|  |  |  |
| --- | --- | --- |
|  | A picture containing circle, symbol, line, design  Description automatically generated |  |
|  | **Figure 35.9** - Symbolic representation of a Hadamard Gate |  |

### 35.13 – Hadamard Gate

To place a qubit into superposition, we use a Hadamard gate (H). This is another essential quantum computing operation as we will see later in a couple of Hello World examples.

If our qubit is currently not in a state of superposition, applying the H gate will do so. In both cases, the qubit goes from its classical definite state (0 or 1) into a superposition state of equal probability. However, depending on the state of the qubit prior to the operation, the resulting superposition states will differ in terms of phase. This phase difference allows the H gate to be its own inverse, in the same way a NOT gate is its own inverse in classical computing. If we apply the H gate twice, we end up with the exact same qubit state as we started with. This is essential to quantum computing and the algorithms we implement.

### 35.14 – Read-out Gate

Not really a gate, but more of an endpoint, read-out is the direct observation of the value of the Qubit. Should the Qubit not be in superposition, the read-out simply reflects the state the Qubit was in 0 or 1. Had the Qubit been in superposition, the read-out causes the collapse of the waveform and returns 0 or 1 with some probability.

|  |  |  |
| --- | --- | --- |
|  | A picture containing LEGO, cartoon, toy  Description automatically generated |  |
|  | **Figure 35.10 - Spy Vs Spy** |  |

### 35.15 - Spy Hunter Demo

Imagine I ask someone to deliver a handwritten note to another person, with the specific instruction that they cannot read it, or have anyone else by the recipient to read it. In the classical world without the quantum phenomena of superposition and entanglement, there is no way of ever knowing whether this has happened anyway by examining the note on arrival. You cannot tell from looking at the note whether someone else has read it before you. However, in the quantum world, reading the note as akin to an observation, a measurement. And as such, it has a potential lasting effect on the note itself, because such a measurement will collapse any part of the note that was in superposition. Currently, we cannot put entire pieces of paper into a state of superposition (although it has already been done with molecules. We can though, use qubits, instead of pieces of paper, to send our information. And these, we can put into a state of superposition. The following example highlights how the phenomenon of superposition used in quantum computing can provide a level of security unheard of before. More specifically, it allows use to determine whether a piece of sensitive data has been intercepted and copied during transfer. Imagine that Heidi and Ada want to exchange data but want to make sure the data isn't seen by anyone during transmission, especially not Thomas who has nefarious plans with the data and a means to tap the channel that Ada and Heidi are communicating on.

|  |  |  |
| --- | --- | --- |
|  | A picture containing LEGO, cartoon, toy, clipart  Description automatically generated |  |
|  | **Figure 35.11** - Using Quantum Computing to determine if sensititive information has been seen during transfer. Here we have Ada and Heidi communicating and we want to make sure Thomas isn't sneaking a peak. |  |

The following is a series of steps that will tell Ada and Heidi whether their information has been intercepted. Ada is the one to send a bit of information. To do so, she uses two qubits. The first she uses to flip a coin uses the outcome to decide whether to put the second one into a superposition or not. She logs the decision and sends the qubit, which can be |0> or |1> if she didn't apply the Hadamard or |+> or |-> if she did use the Hadamard.

|  |  |  |
| --- | --- | --- |
|  | A picture containing LEGO, cartoon, clipart  Description automatically generated |  |
|  | **Figure 35.12** - Ada's decision-making progress on whether to send a classic bit or qubit in superposition. One qubit acts as a random number generator producing a truly random 0 or 1 when collapsed after being put in superposition. This determines whether the qubit to be send should be put in superposition or not before being send to Heidi. |  |

Heidi will have to read the bit of information set gets from Ada. To do so, she also uses two qubits: one of her own, and the one she received through the channel. She uses her own qubit s to flip a coin to decide whether to pass the received qubit through a Hadamard gate, exactly like Ada did. And after that, whether it is passed through the H gate or not, Heidi reads the received qubit and logs the data and her decision to apply the Hadamard gate or not. Let's look at the all the possible scenarios that can play out in this process. From this, we will be able to conclude that Heidi can decode Ada's information, even without knowing, at the time (this is important) what Ada did to the qubit prior to sending it.

|  |  |  |
| --- | --- | --- |
|  | A picture containing LEGO, cartoon, clipart  Description automatically generated |  |
|  | **Figure 35.13** - Heidi's decision-making progress on whether to push the recevied Qubit through her own Hadamard gate. Like Ada, one qubit acts as a random number generator producing a truly random 0 or 1 when collapsed after being put in superposition. This determines whether the qubit to be received should be put through a hadamard port or simply collapsed as is. |  |

These choices by both sender and receiver lead to 4 different scenarios that can tell whether during transmission the qubit of information has been read by Thomas.

**Scenario 1**. In the first scenario, neither Ada nor Heidi decides to put the qubit into a state of superposition. This defaults to a classical computing situation: Heidi simply reads exactly what Ada has send.

|  |
| --- |
|  |
|  |

|  |  |  |
| --- | --- | --- |
|  | A black rectangle with a white background  Description automatically generated with low confidence |  |
|  | **Figure 35.14** - First scenario: neither Ada nor Heidi passes their qubit through a Hadamard gate. |  |

**Scenario 2**. Alternatively, both Ada and Heidi decide to put the qubit into a state of superposition. Remember that stringing two Hadamard gates to together has the second H gate undo the operation of the first. Therefore, if both apply the H gate, the qubit state that Heidi observes is the state Ada send.

|  |  |  |
| --- | --- | --- |
|  | A picture containing LEGO, cartoon  Description automatically generated |  |
|  | **Figure 35.15** - Second scenario. Both Heidi and Ada pass the Qubis through their Hadamard gates A qubit passing through two Hadamard gates in sucession returns the Qubit to its original state, so Heidi and Ada again agree on the state of Qubit in the classic sense: 0 or 1. |  |

**Scenario 3**. Heidi put the qubit in superposition, but Ada does not pass it through a H gate again. That means that when she reads the qubit, she collapses it state: it will be 0 in 50% of the cases, 1 in the other 50% of the cases. But more importantly, the state of the qubit Heidi observes is decorrelated completely from what Ada set it to.

|  |  |  |
| --- | --- | --- |
|  | A picture containing LEGO, cartoon, clipart  Description automatically generated |  |
|  | **Figure 35.16** - Third scenario. Only Ada put their Qubis in superposition using the Hadamard gate. Heidi collapses the qubit, which can now take on any state, completely decorrelated from what Ada send. Since Ada and Heidi are aware of that only one of them applied the Hadamard gate, they expect this, and they expect this. |  |

**Scenario 4**. Here, Ada does not apply the H gate before sending, but Heidi applies when receiving it before reading. Like scenario 3, this sequence of events breaks the correspondence to what Ada set the qubit to and what Heidi reads out. The qubit comes to Heidi in non-superposition state. However, by putting it into superposition before reading, Heidi basically recreates the random number outlines at beginning of this section. If Ada and Heidi compare notes about what they did to the qubit prior and after sending/receiving it, they should also agree on the bit state if they both applied or both did not apply the H gate operation. When this correspondence breaks down, it means something is interfering during transfer of the information.

|  |  |  |
| --- | --- | --- |
|  | A picture containing LEGO, cartoon  Description automatically generated |  |
|  | **Figure 35.17** - Fourth scenario scenario. Ada sends the Qubit like a classic bit, but Heidi applies a Hadamard gate to it. Again, this decorrelates what Heidi and Ada consider to be the actual state of the Qubit after collapsing it. |  |

Now, imagine that Thomas wants to steal the bit of information. He decides to secretly read the bits in transfer. He does not know whether Ada but the qubit in superposition, that information is not communicated through the channel until after the entire message has been send. Therefore, in reading it, he potentially collapses the superposition, leaving him with a random qubit state that has nothing to do with the information that Ada is sending to Heidi. Even more problematic, he cannot undo this operation, pretending the qubit was never read. Again, reading it alters the qubits state in some instances. He can therefor only send a new qubit but has no idea what he should set it to: classic or in superposition. No matter what clever trick he uses to send a new qubit to Heidi, pretending it was never read to begin with, Ada and Heidi will stop seeing an agreement between qubit states when they both apply or both not apply the Hadamard gate operation.

|  |  |  |
| --- | --- | --- |
|  | A lego figure with a question mark  Description automatically generated with low confidence |  |
|  | **Figure 35.18** - Thomas cannot in any way have a look at the data without Heidi and Ada noticing. Since he is unware whether the Qubit is in superposition or not, he can only guess what to send to Ada. |  |

### 35.16 - Some Use Cases of Quantum Computing

The following is a non-exhaustive list of potential use cases. They do share a common underling theme: all of them deal with a level computational complexity that scales poorly in classical computing frameworks.

### 5.17 - Use Case: Linear Systems

Key to many machine learning algorithms is the high dimensional algebra that needs to be performed. Most computationally expensive are matrix multiplications, dot products, inverses, and convolutions. With quantum computing, these large mathematical operations can be carried out all at once, on all the elements of our data simultaneously, rather than the serial process of our conventional CPU based computers. Similarly, Google's PageRank algorithm emulates random walks processes, used to compute the strength between links. Quantum computing can speed up the necessary calculating needed to compute PageRank and related graph metrics used by social media services, or graph databases.

|  |  |  |
| --- | --- | --- |
|  | A picture containing circle, graphics, screenshot, clipart  Description automatically generated |  |
|  | **Figure 35.19** - Multiplying high dimensional matrices, like those in PageRank is an excellent application for Quantum Computing |  |

### 3 35.18 - Use Case: Hidden Markov Models

Hidden Markov Models are used to capture the likelihood of transition between states. In natural language processing, they are used to predict the likelihood of a word, given the words already observed in the text, in some order. Because there are so many possible transitions, estimating the probabilities associated with them becomes exponentially harder with more states being added. Each probability is a weighted function of all other probabilities. Quantum computing would allow to quickly resolve this system of dependents, in the same way that it allows us to solve many linear equations in parallel.

|  |  |  |
| --- | --- | --- |
|  | A picture containing drawing, design, pattern, embroidery  Description automatically generated with medium confidence |  |
|  | **Figure 35.20** - Protein folding is one of the most challenging topics in machine learning and essential for finding new drugs to treat diseases like cancer. Again, it is the dimensionality of the solution space that is simply too large for classical computing methods. |  |

### 35.19 - Use Case: Protein Folding

The protein folding problem is the question of how a protein's amino acid sequence dictates its three-dimensional atomic structure. Our ability to efficiently design proteins with a specific structure, and therefore biological properties will push forward medical research in terms of drug development. Rather than expensive laboratory experiments, computer simulations can predict protein structure directly from low level features. However, current models predicting these structures are incomplete and therefore lack the required level of accuracy. The exponential growth of potential conformations (folding structures) with chain length makes the problem challenging for classical computers. For example, in one model, a chain of 20 amino acids has 109 potential conformations, and chains with 60 and 100 amino acids have 1028 and 1047 conformations, respectively. Quantum computing has the potential to overcome many of these computational challenges. For example, scoring the great number of possible structures and identifying the most promising one.

### 35.20 - Use Case: Genomes and outcomes

Continuing the same theme once more, finding cause and effect between the complexity of human genome and its additional epigenetic effects to an equally complex space of observable outcomes will require shifting through an extremely large space of possible interactions and directional effects.

|  |  |  |
| --- | --- | --- |
|  | A person pushing a key into a folder  Description automatically generated |  |
|  | **Figure 35.21** - Similarly, encryption, like secure hash algorithms that can generate extremely high dimensional keys that will take longer than the age of the universe to decrypt. In the theory, quantum computing could try every possible key at once, making encryption like blockchain obsolete. |  |

### 35.21 - Use Case: Encryption and Security

Quantum computing and its ability to explore large state spaces at high speed creates a problem for our current encryption methods. However, it also offers new ways to encrypt and secure sensitive data, at a level far deeper than current technologies can deliver on. We will demo this approach in the Hello World section below.

### 35.22 - Use Case: Machine Learning

### 35.23 - Resources

|  |  |  |
| --- | --- | --- |
|  | A picture containing text, screenshot, font, brand  Description automatically generated |  |
|  | **Figure 35.22** - There are several resources available already that either simulate or claim to allow you to use real quantum computers to test out algoithms. With these tools, several new programming languages have also been introduced. |  |

### 35.24 - Demos

### 35.25 - References