Dungeons & Qubits

An Adventurer's Tale Beyond the Quantum Computing Tutorials

Brian N. Siegelwax
Independent Quantum Algorithm Designer

2021-06-06

Contents

1 Challenge

I'd like to thank QubitWrangler (@QubitW on Twitter) for this idea. I had just tweeted, "I will be giving [this book] away for free to anyone willing to read it" when he replied "if you're not willing to read it, how much do you have to pay?"

This instantly reminded me of the ALS Ice Bucket Challenge. Consequently, I will be challenging people via social media to either read this book for free or to donate to Sisters of Mary Congregation via Fr. Al's Children Foundation, Inc. (FACFI) at:
https://www.facfi.org.ph/

"The Sisters of Mary Congregation was founded by Msgr. Aloysius Schwartz, an American diocesan priest now known as Venerable Aloysius Schwartz, in 1964 in Busan, South Korea to provide free educational and vocational welfare for the underprivileged children/youth, medical care to the sick and the dying, and shelter service to the homeless and helpless."

2 Prologue

I was born in the greatest city in the world. Within several years, however, my family moved to the suburbs of another great city, closer to where my father could practice his arcane traditions. On our big table, under a fairy light, sat his leather-bound spellbook. Along its spine were engraved the mystic symbols ”HP 3000;” whatever could that mean? The strange words adorning its vellum pages were spells written in an exotic language I would later learn is called “BASIC.”

Through my early years, with countless hours of self-study, I went on to learn other exotic languages, such as Logo, HTML, VBScript, SQL, Javascript, and Visual Basic. I have probably long since forgotten others. And I gained proficiency in skills such as operating systems, networking, word processing, spreadsheets, databases, desktop publishing, software development, web development, and project management. My mastery was so great with some of the tools of these trades that I instructed and lectured working professionals at multiple academies. And, of course, I could build and repair spellbooks.

Then began several years of military apprenticeship, during which I incessantly practiced the evocation spell magic missile. I fondly recall using an arcane focus known as an M4 "thunderstick," which I uncreatively named Mjolnir. I gained proficiency in skills such as [redacted], [redacted], [redacted], and [redacted].

Self-study resumed with a focus on the arcane tradition of Artificial Intelligence, specializing in Machine

Learning and specializing further in Deep Learning. I learned even more exotic languages, in descending order of significance: NASM, C, Python, and quite a few others. I began publicly advocating that once you learn a few languages, irrespective of whether those languages are spoken or arcane, it becomes quicker and easier to learn new ones. You don't even need to cast a 1st-level Comprehend Languages divination spell.

Over several months, I gained insight into the mystic symbols of Linear Algebra and Calculus. New proficiencies included skills such as sentiment analysis, text summarization, keyword ranking, and Google web scraping, as well as tools such as the Twitter API and IBM Watson. My arcane research included pseudorandom number generation, code that writes code, code that edits code, quines, Monte Carlo simulations, and neural networks for regression and classification, as well as Long Short-Term Memory (LSTM) and ensembles.

And then, one day, while exploring an ethereal village called YouTube, I learned of quantum computing and set forth on a grand new adventure....

Throughout this tome, by the way, “z” is pronounced “zee.” [29]

3 Acknowledgements

It's customary to mention somewhere near the start of a book that all responsibility for errors and omissions are born by the authors. I'm editing and distributing this version by myself, so I couldn't point at anyone else even if such behavior was within my character.

Nope. Every word and punctuation mark was placed by me. So, if you think my writing is completely unintelligible, that's on me. And, if you think one of my statements is grossly inaccurate, well, that's on me, too. Or, if you think one of my jokes is actually funny,... okay, that's on you.

That said, I'd like to thank, in chronological order:

- My parents. It would have been really, really hard to write this book had I never been born. And, I couldn't have asked for better.

- Whoever introduced me to Dungeons & Dragons. [41] If this book should have only two readers and both of them think it's stupid, at least I thoroughly enjoyed writing it.

- SoloLearn. My understanding of Quantum Machine Learning is heavily due to the very simple classical neural networks that I found within this Android app. I translated these neural networks, within this same app, from Python to C to force myself to deeply understand how they work. It is this deep understanding that enabled quantum-related revelations later.

- YouTube algorithm. I had never heard of quantum computing. And, even if I had had, I would never have presumed any hope of ever playing with quantum processors without a Ph.D. in physics, mathematics, or perhaps engineering. Everything changed when the YouTube algorithm "thought" I might want to know that IBM would allow me to do exactly that via the cloud absolutely for free.

- Dr. Seth Lloyd. The notion of mapping a terabit of data to only 40 qubits instantly captured my imagination. And this curiosity intensified over time as questionee after questionee not only couldn't explain it but, in some cases, replied that it couldn't be done. Well, it can be done. It is shockingly easy to do. And, I will share my discovery of how to do it within these pages.

- IBM. I have run well over 1,000 jobs on IBM quantum processors. Well, a percentage of those were run on the quantum computing simulator, of course. But, the important thing here is the price tag. The total cost of running 1,000+ jobs within the IBM Quantum Experience has been exactly 0 silver pieces. They haven't asked me for any real-world money either. And, it is my understanding that these devices cost many millions of dollars each, and I've always had a selection of multiple devices. Consequently, I try to regularly express my fandom across social media platforms.

- Arnaldo Gouveia. Suggesting I write a book is a huge compliment, and I am grateful beyond words. Irony intended. I also appreciate his referrals for jobs at IBM, even though IBM seems to require a Ph.D. even for janitorial careers.

- Dr. Robert Sutor. I approached Packt because of "Dancing with Qubits." [37] I asked him all sorts of questions about authoring and Packt, and every time he responded I told my wife how big of a deal it is that someone of his stature is acknowledging me at all. So, I owe a lot about the mechanics of this book, such as how I created my LaTeX circuit diagrams, to Dr. Sutor's valuable time.

- Ravit Jain. At one point, his was the only name at Packt that I knew due to his promotion of "Dancing with Qubits." [37] I thank him for introducing me to Sunith Shetty.

- Sunith Shetty. I greatly appreciate the opportunity, especially as a first-time author, to work with a traditional book publisher. Ultimately, however, I decided that if I am going to invest such a substantial amount of time in any kind of endeavor, I need to retain full creative freedom and I need to have fun with it. But, if you have an idea for a technical book and you're willing to work within a publisher's guidelines, you can reach out to him on LinkedIn.
- My wife. Chronologically, we married before all of this. But, it was on Christmas Eve 2020 that she gave me the green light to officially start this project.
- My kids. For sleeping. Most of this book was written while at least one of my kids was unconscious, and often all of them. In fact, my writing was just interrupted so that I could tell a funny horror bedtime story about baby firetrucks. That's not a joke. That actually just happened.
- VerbTex. I wrote this book on an Android smartphone using the VerbTex app. I don't know if

it's the only way to manage citations while writing on a smartphone, but it's the only way that I could find. Of all the things I do exclusively on smartphones, including machine learning and quantum computing, managing citations for a full-length book stands out as being particularly challenging, especially when the list of potential sources starts approaching 100.

- Quantikz. I learned about this LaTeX package from Dr. Sutor and it is a relatively straightforward way to build quantum circuit diagrams. As I write this the arXiv BibTek citation isn't working, but the arXiv identifier is 1809.03842.

- Overleaf. There remain limitations to writing a book on a smartphone. The final PDF document that you are reading right now was finished in my browser's "desktop mode" with Overleaf.

- Grammarly. I apparently have a few words and phrases I like to use too often. Plus, this free smartphone keyboard caught a few legitimate grammatical errors.

- Pixabay. If it doesn't look like a circuit diagram, it was a free download from Pixabay.

4 0th Edition

What's new in this edition? Everything! Comments and constructive criticism are welcome!

- 0. First publicly-available draft. 2021-06-06

"Hit me with your best shot. Fire away."
- Pat Benetar

5 Apprentice Adventuring

I was visiting an ethereal village called YouTube, which I had been prone to do way too often, when its town crier made a bold attempt to draw my attention. It worked.

Through an intricate set of minor illusions, a sage named Doug McClure, representing the IBM faction, claimed that the quantum computing magic school was allowing free arcane research through ethereal resources. My imagination was captured instantly, and I vowed to someday cast a single spell, simply to boast that I had had. There was just one problem: casting quantum computing spells must be especially difficult, right?

So, I did something I had never done before: I studied. I even enrolled at an ethereal academy and audited a course.

I was inquisitive. I was studious. I memorized incantations and gestures. I listened to experts such as Dr. Peter Wittek, who I believe introduced me to the concept of variational quantum spells. Variational spells are better characterized as hybrid classical-quantum spells, and I generally avoid them these days. I prefer to craft and cast pure quantum spells and avoid classical components to the extent possible. [3, 21]

Dr. Wittek also virtually introduced me to Dr. Seth Lloyd, who launched my quest to discover how to map a terabit of data to only 40 qubits. I seriously thought I had heard him incorrectly. Consequently, I re-listened to the audio and re-read the transcript multiple times just to make sure I hadn't failed a saving throw against some kind of unfamiliar confusion effect.

I no longer remember if it was weeks or if it was months, but the passage of time finally led to the confidence I needed to register for the IBMQ Experience.

So, I registered with IBMQ – which has since been rebranded as IBM Quantum – and quickly found a set of tutorials. The first tutorial could not have been better explained, and I wish it still existed. Sadly, it has been modified over time. But, I digress, the somatic components were clearly and concisely described in Common tongue, as was the guidance for determining whether or not the spell was cast successfully.

The goddess Mystra must've frowned upon me, however, because there was a second problem: my technology, a six-year-old Samsung Galaxy Note 3, was seemingly incompatible with the drag-and-drop quantum spell builder. I now know that a browser's "desktop mode" might've solved the problem, but I hadn't thought of that at the time. I must thank Dr. James Wootton, one of the authors of the original IBMQ tutorials, for later giving me the idea to try that.

For the moment, though, I was disappointed. I was saddened. My quantum computing adventure seemed to have ended before it even had a chance to begin.

But then I noticed a nearby partially-obscured scroll. It was blank except for just a few strange words, one of which was "OpenQASM." [10] Though I had never encountered this language before, it seemed quite familiar to me. Aesthetically it reminded me of NASM, [40] which turned out to be no coincidence since the "ASM" in each name is short for "assembly language." But, the tutorial made no mention of OpenQASM, and I couldn't find any language resources anywhere within the IBMQ Experience, so where could I learn just enough OpenQASM to run this one experiment?

I scoured manuscripts and studied scrolls. The system of notation seemed very straightforward. It seemed as if some of the scribes had been fluent in Assembly Language and other scribes had been fluent in C, and the resultant OpenQASM language became a hybrid of the two. Since I found this new language to be quite readable, I regained my confidence that it was once again time to start manipulating the fabric of quantum magic.

SPELLCRAFTING

The first few lines of the incantation were already provided, so I decided not to alter them in any way:

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[5];
creg c[5];
```

If I could have followed the tutorial, the first somatic component required was the "Hadamard" gate. Although I never found an OpenQASM command called "Hadamard," I did quickly find an "h" command. I took a gamble and transcribed the following:

```
h q[0];
```

The next somatic component was called a "CNOT" gate. I honestly don't remember at this point if I knew that “CNOT” and “CX” gates were the same thing or if I only knew that “NOT” and “X” gates were the same, but I do remember finding a “cx” command and being reasonably confident, compared to the “h” command, that it was correct. I transcribed the following:

cx q[0],q[1];

Finally, I knew that I had to measure the qubits. I found a “measure” command and, for this one, wondered, “it can’t be that easy, right?” Nonetheless, I transcribed the following two lines:

measure q[0] − > c[0];
measure q[1] − > c[1];

I would later learn that this spell is not optimized. In fact, I haven’t seen optimization mentioned in any tutorials anywhere, nor have I seen since a comprehensive approach to quantum spell optimization. Instead, it's something that is learned little by little and is quite often incorrect. As a devotee of the IBM faction, I'll go buy a few rounds of ale at Qiskit Slack Tavern and get real answers as to whether newfound techniques will benefit my spells or not.

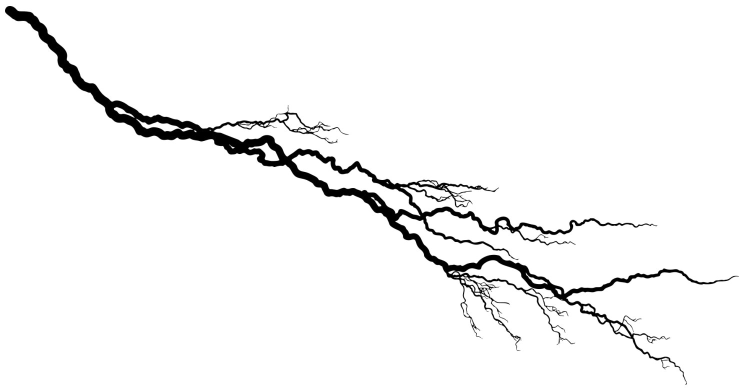
But, I didn't know anything about optimization at this point, and I just wanted to cast my first quantum spell. I called on Mystra and began to intertwine, twist, and fold the threads of The Weave. I fixed the spell in my mind, taxed myself physically and mentally, channeled my energy, and with one forceful gesticulation, I tapped the blue "run" button.

Never ask me what I rolled for initiative. Don't do it. My order of combat was so low that I paced like an expectant father for about half an hour. Remarkably, despite being a novice, I seemed to have made my very first ability check. I don't recall that ever happening with any other language. But, I read and I re-read the tutorial, and my confidence grew that I had just cast my first-ever quantum computing spell: "hello, quantum world!"

I went on to learn spells crafted from esteemed sages such as Dr. Grover, Dr. Deutsch, Dr. Jozsa, Dr. Simon, and Dr. Shor. Readers of this tome all have these spells in your spellbooks by now. I skipped, in case you hadn't noticed, a spell from Dr. Bernstein and Dr. Vazirani. It seemed to have been presented out of sequence and thus learning it seemed unnecessary.

Finally, I learned two particularly interesting spells. The 7th-level Teleport conjuration spell was initially disappointing due to science-fiction expectations, but I copied this one into my spellbook for potential future use anyway. And then there was the Quantum Fourier Transform, [17, 29, 31, 33] the power of which I would go on to underestimate for quite some time. But, I guess both spells demonstrate why only higher-level wizards learn higher-level spells.

The "leading edge" and "bleeding edge" were no longer good enough. To me, quantum computing became the "hemorrhaging edge" of magic and precisely where I want to be.



6 Fireballs & Lightning Bolts

Upon completing the tutorials, it was time to venture beyond the safety of the city walls; there was nothing more to learn within their confines.

As I departed, my spellbook contained not only the tutorial spells but also my earliest attempts to craft new spells. I had discovered, for example, a full adder spell that added $1 + 1$ and I had successfully modified it to add $2 + 2$.

SPELLCRAFTING

I had found a quantum spell for adding two ones, so first I tried to modify it to add two twos. Admittedly, I couldn't get it to work. And, unfortunately, this took place so long ago that I no longer remember what I tried that failed. But, I do remember thinking back to the tutorials and remembering that any spell that can be cast classically can also be cast quantumly.

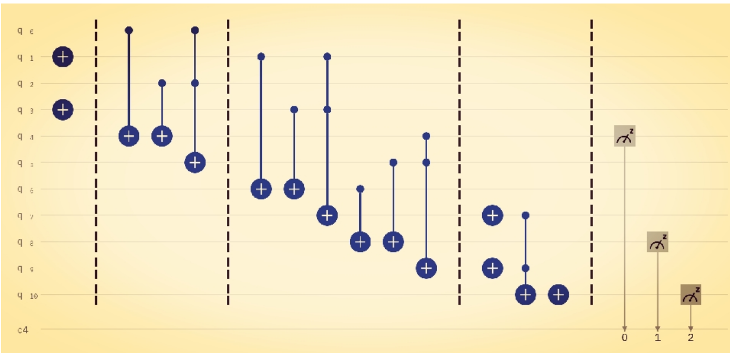
My interpretation of that was to research classical adder circuitry. I wanted to know the ingredients – the logic gates – required to cast a classical full adder spell. I then swapped these classical ingredients for their quantum counterparts. Or, so I thought.

What I really did is I tried to literally and faithfully recreate a classical full adder circuit using quantum logic gates. Toffoli gates are AND gates, so that was easy. Then, I recreated the classical XOR gate using two CNOTs and an ancilla qubit and the OR gate out of an AND gate with inverted inputs and an inverted output.

Basically, once I got the classical truth tables [30] to work with quantum gates, I immediately grabbed my quill and transcribed the commands needed for casting them.

I tell this tale not simply because it happened, but as a cautionary tale of what NOT to do. The thinking that went into crafting this spell was entirely classical. At the time, I was merely interested in getting a quantum full adder spell to work; I hadn't considered whether anyone should even bother doing that, which, of course, you shouldn't. There will never be justification, in regards to computational advantage anyway, to add numbers together using quantum spells.

This, therefore, has become a major reason why I don't spend time crafting or even learning variational spells, spells that mix classical thinking and quantum thinking together. Some of that hybridization, admittedly, is due to efficiency; again, don't bother adding numbers together quantumly. However, too much of that hybridization is due to overreliance on classical thought. If a quantum solution is too challenging or too counter-intuitive, however you want to characterize it, you can still shove square classical pegs into round quantum holes. But, you shouldn't. Unfortunately, it works; consequently, too many researchers fall back on it.



This tome isn't about classical logic gates or truth tables, so I won't delve any deeper into the crafting of this particular spell, but I will add one final note about inefficiency. To demonstrate how little I understood quantum computing at this time, I used two CNOTs and an ancilla qubit to craft a quantum XOR logic gate where a single CNOT gate should've sufficed. If you make a truth table for a CNOT gate, you should notice that the output of the target qubit is an XOR operation on the two inputs.

Go ahead. I'll wait.

Anyway, I didn't make this mistake just once. I did this a second time when I possibly misinterpreted an interview with Dr. Seth Lloyd. Critics at Reddit Tavern thought I must've heard something about factoring and I was confused. After all, factoring 15 into 3 and 5 is definitely a thing. However, I'm fairly certain I heard Dr. Lloyd say, "multiply three times five." Because of what I thought I heard, I went ahead and crafted a multiplication spell.

SPELLCRAFTING

At this stage of my journey, the easiest way to encode the multiplicand, in this case the number three, and the multiplier, in this case the number five, was to use binary. Therefore, I needed two qubits to encode binary 11 for decimal 3 and three qubits to encode binary 101 for decimal 5.

I then essentially performed multiplication as if I was doing it on paper.

```
  11
x101
----
.  11
. 00
11
----
1111
```

Using truth tables, I realized that binary multiplication is nothing more than AND gates: the product of two ones is one, and all other products are zero. The quantum component for that is the Toffoli gate.

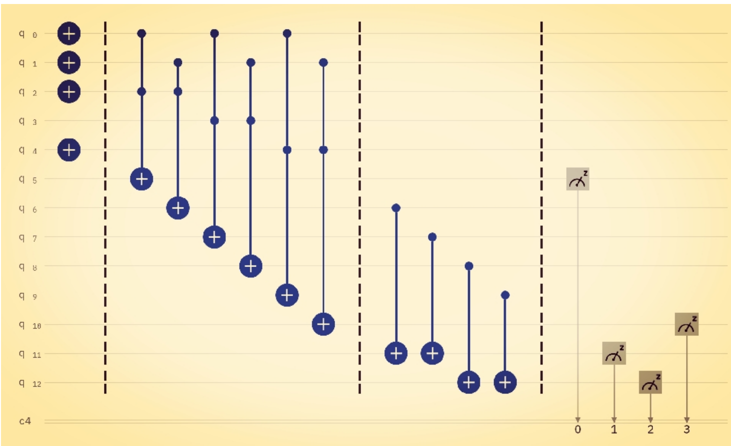
0*0=0
0*1=0
1*0=0
1*1=1

I also remembered that binary addition is nothing more than XOR gates; the sum of zero and one, in either order, is one, and the sum of two of the same digits is zero. The quantum component for that, I had discovered by this point, is the CNOT gate.

0+0=0
0+1=1
1+0=1
1+1=0

The only trick involved in casting this spell was in measuring the qubits in the correct sequence.

This multiplication spell has multiple issues. Like the aforementioned addition spell, it is classical thinking implemented with quantum components. There's nothing inherently "quantum" about it. Also, in its current configuration, it breaks down if you try to multiply numbers with a product higher than 15. Consequently, this spell is doubly worthless.



Although my "quantum" arithmetic spells functioned, both are examples of square classical pegs shoved into round quantum holes. But, even if they were round quantum pegs, they would still be pointless. Spoiler alert: classical calculators are here to stay, and there will never be any computational advantage to casting quantum versions of these spells.

Nonetheless, these spells were important baby steps along my journey. They helped me to learn how to recognize classical thought disguised as variational and even "quantum" algorithms. They also helped me to understand that just because you can cast a quantum spell doesn't mean that you should. If a classical spell is already fast, like a calculator adding $2+2$, it doesn't need a quantum counterpart. Quantum spells will not be useful until they can finally solve extremely challenging, time-consuming, real-world problems. [44]

In addition to crafting new spells, flaws notwithstanding, I had also begun to combine spells. I cannot reveal the most intricate mixture due to its potential usage in a pending quest, however, the important takeaway is that you can "craft" new spells by extending and combining spells you've already learned.

The most significant example of that, of course, is Shor's Factoring Algorithm. Setting aside the classical components for a moment, you should've learned from its tutorial that Dr. Shor crafted the quantum component of this legendary spell from two notable existing quantum computing spells: Simon's Period-Finding Algorithm [1] and the Quantum Fourier Transform (QFT). In fact, there's a good chance your most powerful spells will have the QFT or its inverse in them. [9]

Secondary to the QFT is amplitude amplification. [45] Yes, I'm talking about the diffusion operator of Grover's algorithm. If you read enough scrolls about quantum advantage, you'll find that the QFT and amplitude amplification are universally acknowledged as components that boost the power of quantum spells beyond that of classical spellcasting. [1] However, the power gained from the QFT (exponential) is more pronounced than the power gained from amplitude amplification (quadratic, or quartic with QEC). [5, 37]

Except for the one spell that I can't yet reveal, I didn't craft any combinations of significance during this period. My apprenticeship years were characterized mostly by trial-and-error, and just trying to figure out how everything worked. And, except for my full adder and multiplication spells, I only experimented with the tutorial spells. I tried to extend them and combine them in different ways, but most of the tutorial spells don't have any innate utility anyway, so only the process of experimentation would prove valuable.

6.1 Early Failures

We learn from our failures, and I sure made some epic fails along my journey. I'm sure what follows is not a comprehensive list, but it is a list of my most memorable early failures.

My first epic fail was a scholarly paper. I thought I had discovered the secret to superluminal communication, aka "faster-than-light" or "FTL" communication, but I really only discovered that I had a fundamental misunderstanding of the 7th-level Teleport conjuration spell. I thank Dr. James Wootton of the IBM faction for gently enlightening me.

To be honest, this failure still haunts me. I understand and accept that the textbook spell requires classical communication channels, but I can't escape the feeling that researchers are probably unfamiliar with all the different ways secure communication is made possible. I would love to work in a laboratory where everyone has a security clearance and I could ping away with what-if questions until either someone has an epiphany or I get fired for gross incompetence.

Either way would be fun.

My second epic fail was attempting to factor the number 21. The problem might have been that I had zero interest in the classical components of Shor's Factoring Spell and was trying to cast only the quantum components. But, I re-read an old blog article I wrote about it, and the real problem was I had no idea what I was doing. There was no component of Shor's Factoring Spell that I understood. So, the real, real problem was that I was going through the tutorials too quickly; I needed to slow down and digest each spell, and not just race to the next exciting one.

Over time, I have gained an understanding of most of the spell's components, both quantum and classical. However, I have not since revisited this spell and I have, therefore, not corrected this long ago failure. To my knowledge, 21 is still the largest number that has been factored with Shor's Factoring Spell, and, at this point, it doesn't make sense to channel energy into not contributing toward progress.

And my favorite epic fail was a Role Playing Game

(RPG) partial character generator. This spell was inspired by the aforementioned Dr. Wootton, who I refer to as "the Dungeon Master of Quantum Gaming." After reading his chronology of the tradition, I wanted to attempt to craft something myself. But, I didn't want to do anything resembling anything anyone else has already done, so I needed to craft a quantum gaming spell with a personal twist.

SPELLCRAFTING

When you think about role-playing games, the gold standard is unquestionably Dungeons & Dragons. Whatever other RPGs or MMORPGs you might play, they all trace their lineage to D & D.

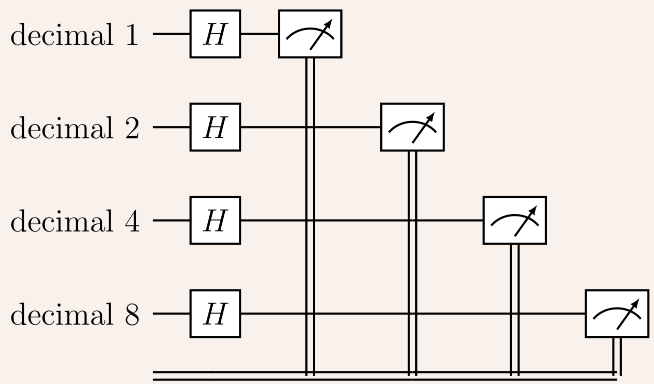
D & D characters have six ability scores: strength, dexterity, intelligence, wisdom, constitution, and charisma. These attributes are so important to the game that they're a logical starting point for a random character generator. With exceptions that are beyond the scope of this tome, each ability score ranges from 3 to 18.

At this point, I still only understood binary encoding. I could generate random binary numbers, sure, but I faced a bit of a conundrum: how could I "roll" an 18 without risking rolling 19 through 31?

I quickly realized that, besides not needing numbers above 18, I also did not need numbers below 3. The easiest way to not "roll" below 3 was to manually add 3 to 0, which also meant adding 3 to all other "rolls." The consequence of adding 3 to all other "rolls" is that 15, the maximum decimal value of four qubits, becomes 18. Therefore, the binary 0000 (decimal 0) of four qubits becomes the minimum 3 and the binary 1111 (decimal 15) becomes the maximum 18, and all other "rolls" fall in between.

I confess: I used Qiskit for this one. I didn't have 24 qubits available, plus you couldn't implement reset gates or mid-circuit measurements on real hardware back then. So, I used the Python language to cast this spell six times in succession and then to format the results appropriately.

The results simply showed the six ability scores, in order, with their randomly generated values.



It is an understatement to tell you I was excited.

After casting this spell, I had a quantum-generated partial D & D character with truly random ability scores. Okay, it was a very partial character, but at least it was a start. Unfortunately, it was also wrong.

I thought I had demonstrated a masterful proficiency in nerd skills until I boasted about my new spell at the Reddit Tavern. There I met a true master of nerd skills; sadly I've long since forgotten the name.

The problem with my spell is that every number from 3 to 18 has an equal probability of being "rolled." If this statement seems counterintuitive, feel free to cast the spell a few thousand times yourself. However, the exclusive use of Hadamard gates means that every binary bit of the output has its own 50/50 chance of being either a 0 or a 1.

And, therein lies the problem. If you roll 3 6-sided dice a few thousand times, you would not see this uniform distribution. With each roll, you have only 1 chance to roll 3 (3 1's) and only 1 chance to roll 18 (3 6's), but you have multiple chances, for example, to roll 11 (6-4-1, 6-3-2, 5-5-1, etc.). Therefore, a quantum spell for generating ability scores needs to have a higher probability of measuring 1000 (which is decimal 8, but we're adding 3 to it), for example, than either 0000 or 1111.

I have not since revisited this spell and tried to correct it; by the time I was sitting in Reddit Tavern with proverbial egg on my face, I had already moved on to my next project. However, I invite readers of this tome to accept such a challenge. It would be exciting to go on an adventure someday with a fully-quantum-generated character.

6.2 Ancient Texts

Once outside the castle walls, I finally heard about the legendary arXiv repository. What a treasure trove of knowledge! How had I never heard of it before? And there aren't even any monsters guarding it or traps surrounding it!

The sheer enormity of the repository should seem overwhelming to anyone newly finding it. I overcame this feeling by committing myself to read – and digesting – just one scroll per day; a practice I maintained until starting to write this tome. According to my diary, my earliest readings included Quantum Error Correction (QEC) and Quantum Neural Networks (QNN).

Sadly, much of the available research has turned out to be disappointing. Naivety suggests that every scroll should bestow great wisdom upon the reader. Instead, you find many authors citing themselves in unsatisfying ways. And, perhaps worse, are what I call "epic rap battles of science," which demonstrate that there is disagreement over some subject matter rather than consensus.

On the other hand, I stumbled upon the evolution of the “no cloning theorem;” that was a fascinating series of scrolls. In short, a researcher proposed quantum cloning. And, although that spell cannot be cast, the review of its components resulted in the formalization of this theorem. The researcher ended up contributing positively to the tradition, even though not as originally intended.

The moral of that story is to go ahead and share your research. In a perverted twist on superposition, you can fail and succeed at the same time.

Another benefit to learning of the arXiv repository was learning about peer-reviewed journals. I remember asking in Reddit Tavern about which journals are the most prestigious; I learned instead the significance of attending conferences. In fact, I was invited early on along my journey to be a guest speaker at a conference and talk about any research I had been working on, but, sadly, I had already discovered the failure of my superluminal communication spell and I did not have a new scroll in development worthy of presentation at a conference.

Around this time I also began reading more into variational spells, also known as hybrid classical-quantum spells. Fortunately, I heard the powerful words of the great sage Dr. Peter Shor, who said something to the effect of not being impressed by the Variational Quantum Eigensolver (VQE) or the Quantum Approximate Optimization Algorithm (QAOA). That was enough for me, and I've kept almost exclusively to purely-quantum spells since then. Not only are variational spells unimpressive, but they are also arguably obsolescent, unlikely to be of use beyond the NISQ era. [26]

Finally, I began learning more about the other quantum computing factions. The only one to even open their gate for me was D-Wave. I appreciate that. Sadly, however, their quantum annealing spells have a high learning curve only to be no more powerful than classical spells. Like the aforementioned words of Dr. Shor in regards to VQE and QAOA, I first heard about quantum annealing's lack of proven advantage from Dr. John Preskill. Again, that was good enough for me. And, I have not read or heard suggestions to the contrary since then, at least not outside D-Wave marketing scrolls anyway.

Other early requests for membership included Rigetti, IonQ, and quite a few others. To this day, none have responded. And while the business case for that is certainly understandable, it seems to me that the IBM faction has been contributing the most to developing the quantum ecosystem. Not only can anyone get free and unlimited ethereal access to multiple quantum processors and simulators, but their sages are also shockingly accessible and extraordinarily helpful. In addition to having had many questions answered, I've been assisted by the powerful wizard [redacted] in casting a spell that I didn't have powerful enough components to cast myself. The spell didn't work as intended, but, I hadn't been able to confirm that on a quantum computing simulator or any of the hardware directly accessible to me. So, pardon my fandom, if you will; it has been earned.

6.3 New Knowledge

As this phase of my journey drew to a conclusion, I reflected upon all the new knowledge that I had gained, starting with circuit optimization. Though straightforward in many ways, you eventually encounter the Quantum Error Correction (QEC) versus circuit optimization paradox. [5, 37] The paradox is that as you try to detect and correct NISQ-related errors, you add circuit width and depth that expose your spells to even more NISQ-related errors. Optimization strategies, on the other hand, try to reduce errors by making circuits smaller. Since you can't add entanglement and reduce entanglement at the same time, nor can you delay measurements and simultaneously shorten time to measurements, the paradox is hopefully fully evident. Further paradoxically, however, you could apply optimization strategies to QEC to minimize the paradoxical increase in noise it introduces.

Because I learned optimization strategies in bits and pieces, both here and there, I don't have a coherent tale to tell about it. Instead, that seems worthy of an appendix at the end of this tome.

During this phase of my journey I also began work on a quantum portfolio optimization spell for finding efficient frontiers. If you don't know what that is, it's an approach to maximizing your investment return for your specific risk tolerance. I mean, who doesn't want a pouch full of gold pieces, am I right?

Anyway, the project was initially a collaboration, but I ultimately withdrew because the potential for quantum advantage was unclear. There wasn't a problem with the quantum spell, per se, but rather that known classical spells are too powerful to make crafting a quantum equivalent worthwhile. We were looking at a classical runtime of a relatively large dataset of less than a minute. And that's an important lesson to apprentice adventurers out there, as I stated earlier in this tome: if there is no advantage gained in using a quantum spell, you should just use a classical spell.

That said, I gained valuable insights into quantum mixed states from this portfolio optimization research. Other than brief mentions of their existence, I learned nothing about mixed states from tutorials. For that matter, I haven't found much actionable information about them since. Almost everything I have read and heard about quantum states has referenced pure states, with the exception being mathematics-heavy scrolls that don't explicitly declare that they are referencing mixed states; you have to learn first how to recognize mixed states by their notation.

I've heard Dr. John Preskill illustrate this problem perfectly by stating that the Bloch sphere should really be referred to as a Bloch ball. A sphere is an outer shell, and that's where we find our pure quantum states. [44] But, a ball consists of its outer shell plus the whole area inside it; here we find our mixed quantum states. [1] So, yes, mixed states include pure states.

Because I couldn't find comprehensible answers anywhere, I independently formulated two approaches to working with mixed states. Both approaches are mathematical. That means, unfortunately, that classical spellcasting cannot be entirely avoided.

I described the first approach as casting a shadow.

Because the efficient frontier is two-dimensional, I imagined shining a light directly above a point on a Bloch sphere. This point casts a shadow onto the Bloch sphere's x-y equatorial plane. Each qubit, therefore, represented a single portfolio of financial assets with a shadow on an imagined (not imaginary) plane, from which the quantum spell would hopefully determine the efficient frontier.

The second approach is more useful when using all three dimensions of the Bloch sphere. I took two pure states, and each represented one point on the surface of a Bloch sphere. I then imagined a line running through the inside of the Bloch sphere and connecting those two points. The midway point of that line represents the mixed state.

After independently formulating these approaches, I would later discover scrolls in support thereof. [1] In all likelihood, previously incomprehensible scrolls suddenly made sense. The scrolls were still written in Physics tongue with mathematical inscriptions, but I could newly recognize them as describing what I had done.

There is nothing quantum about the trigonometry involved, so I leave that to tomes dedicated to that mathematics branch. My proficiency had deteriorated substantially since high school, but that was ameliorated by the fact that a Bloch sphere is a unit sphere, so the length of the hypotenuse is always 1; that noticeably simplifies the equations.

I also began work on an ambitious quantum computing simulator. Started in the C language, my goal is to translate it into NASM as part of a bare-bones operating system. My vision is to have a box dedicated to simulating quantum computing spells and incapable of executing anything else. Unfortunately, that vision remains unrealized at this time. My focus has been on crafting quantum spells, and while this project certainly deepens one's understanding of quantum spellcasting, it is nonetheless inherently classical.

Finally, I learned about quantum-inspired spells, including the Legend of Ewin Tang. The bards tell the story of a quantum recommendation spell that was more powerful than any known classical recommendation spell. That was, until, Ms. Tang studied the quantum variant and created an even more powerful classical spell. [32, 37]

The lesson here, as previously noted twice already, is that if you have a competitive classical spell in your spellbook, or if you can craft a competitive new classical spell, you should craft and cast that classical spell. Maybe your use case doesn't require a quantum spell, after all.

7 6th-Level Spells

As a quantum spellcrafter, this is when the adventure really began. The spells at this level begin to have real-world applications, which is to say they might be useful in the real world if and when the spell components become sufficiently powerful. Of course, there is no guarantee that quantum computing will ever prove to be advantageous over classical computing, but I believe that way too many smart people are working in the field for it to go nowhere.

Of all the potential applications of quantum computing, such as chemistry and cryptography, the one most familiar to me is machine learning. Consequently, of all the scrolls I have read to acquire knowledge, the scrolls about quantum machine learning have easily been the most intuitive. Although I have crafted quantum spells for other applications, I have naturally gravitated in the direction of QML, and that will probably become evident as you read this tome in its entirety.

7.1 Quantum Machine Learning

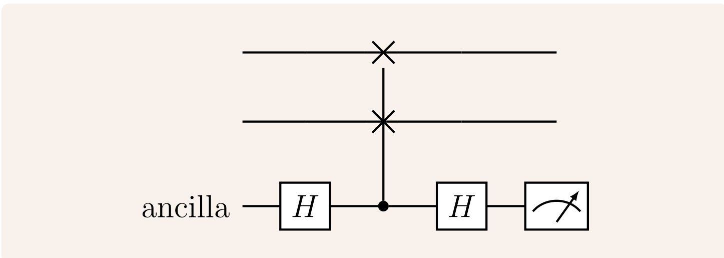
Before I set forth on this quantum computing adventure, I studied classical machine learning. And, because I wanted to deeply understand how neural networks function, I translated a few simple Python spells into C. My goal in doing so had nothing to do with increasing the power of the spells, but rather in fully understanding each and every component of each and every spell. By crafting the spells from scratch, and by using the unforgiving C language, I was forced to achieve a relatively deep understanding of neural networks to get them to function properly.

Coincidentally, after studying classical machine learning and before researching quantum machine learning, I discovered something called the SWAP Test. Of all the scrolls I was studying from the arXiv repository as part of my daily readings, I found the SWAP Test to be unusually comprehensible. Although I didn't deep dive into the SWAP Test as I did with neural networks, I went on to do extensive experimentation with it, and I became quite familiar with what it did and how it could be used.

7.1.1 SWAP Test

The SWAP Test is a powerful quantum spell component that I first learned about as a lower-level adventurer. [2] Also known as "distance measure," "kernel method," and "inner product," [19] because everything quantum has to be unnecessarily confusing, it can be found in the catacombs under the arXiv repository. If you're lucky, the beholder will be sleeping while you sneak in there.

I recognized its significance immediately and began extensive testing.



As its simplest application, the SWAP Test determines if two quantum states are identical. Looking at the scroll fragment above, the top two qubits are the ones you want to compare. The measurement of the ancilla qubit, which serves as the control qubit for the controlled-SWAP operation, measures 0 when the states are identical. More importantly, when the states are not identical, there is an equal probability of measuring 0 or 1. [15, 16]

Experimentation revealed that we can learn so much more than simply whether or not the states are identical. By casting the spell thousands of times, we can learn the degree of similarity. Whereas identical states measure 0 every time, maximally opposite states – imagine two points on exact opposite sides of a Bloch sphere – measure 0 only 50% of the time. All other distances between those two points measure 0 less than 100% of the time but more than 50% of the time.

But, that's not random. Two close quantum states measure 0 closer to 100% of the time, while two distant states measure 0 closer to 50% of the time. This discovery alone made me realize that the SWAP Test must have great potential.

What else can we use the SWAP Test for? Well, borrowing from quantum state tomography, we can determine how similar two states are in each of the measurement bases, not just overall. We can also test the output of some operation to determine if an entanglement spell has been successfully cast. [14, 16] And, we can do Quantum Machine Learning (QML).

So, an important question is whether or not the SWAP Test provides any potential for computational advantage over classical spellcasting. After all, I've already besmirched everything except the QFT and amplitude amplification. Well, since I'm not a mathematician, I simply defer to the scrolls of other researchers who have written in support of the SWAP Test's potential to achieve a quantum advantage. [2, 8, 18, 24, 25, 35, 34] I have not read any scrolls disavowing their claims so, unlike VQE, QAOA, and D-Wave, I continue to use the SWAP Test.

The only experiment that I can think of to do with the SWAP Test and that I have not done yet is to test its destructiveness. [16] This is something I read about and accepted, but never actually tested. But, if the SWAP Test, or one of its variations, can measure distance non-destructively – meaning that compared quantum states can still be used after being compared – maybe some spells crafted with it can be modified to require fewer components.

One final note before moving on: as just briefly mentioned, the SWAP Test has alternatives. The Hadamard Test [2, 8] works for very basic comparisons. Its advantage is that it adds far less circuit depth, which is an important NISQ-era consideration, but its disadvantage is its very limited usefulness. There is also an unnamed alternative that substitutes one CNOT gate and one Toffoli gate for the Fredkin gate, [14] but I have not tested this configuration. The idea behind that proposal is to achieve the same results as the SWAP Test while minimizing the amount of additional circuit depth required to achieve it. But, again, confirmation of that would require significant testing. In the meantime, the SWAP Test simulates well and both approaches, despite having their own error detection and mitigation process, [20] would fare too poorly on NISQ devices to compare their performances there.

7.1.2 Quantum Neural Networks (QNN)

It was late at night in arXiv; perhaps calling it predawn in the morning would be more accurate. The aisles were mostly empty. The tables were mostly bare. And my candle was down to its final drips of wax. I turned for the exit.

Suddenly, a flicker of light illuminated some parchment that had fallen onto a chair. The lines, the colors, the markings,... they were maps! But, had I found what I was looking for?

An adventurer in Twitter Tavern had been boasting about discovering a dungeon called Quantum Neural Networks (QNN). [38] The arcane energies within were told to be young but of great potential. Its depths might host components for functional, albeit rudimentary, new spells.

I ventured out immediately. Surely my knowledge of classical neural networks would be of some advantage, at least on the upper levels of the dungeon.

But, excitement soon led to disappointment. The dungeon posed no challenge. No evocation spells were cast, no dagger was drawn, and coveted scrolls were easy to find.

The worst part was that the scrolls didn't seem worth finding. As I perused them, I grew increasingly troubled that the "quantum" spells written thereon seemed shockingly familiar. "What's with all these classical components," is the family-friendly translation of what I muttered to myself. These incantations and gestures so closely resembled those of classical spells, what made them any more "quantum" than my full adder and multiplication spells? [7, 39, 38, 42]

Saddened, I returned to the nearby village. The journey was long but uneventful. I went to my temporary quarters at the inn, finished off some old rations, and passed out.

Upon awakening, I ordered a mug of qoffee and resumed my studies. Of course, I had taken a few select scrolls from the dungeon with me. I had awakened with the belief that if Quantum Machine Learning is to be a viable school of magic, surely quantum neural networks must be capable of more than what I had seen.

So, I sat in a reasonably-comfortable chair and I pondered. I thought back to my earlier studies and how neural networks fundamentally work. I pulled out the scrolls that I had written in C tongue and I began to read them line by line.

Fortune smiled upon me and epiphany did not delay: the SWAP Test has precisely the same function as neural networks.

7.1.3 Classification

Out of the four simple neural networks that I had translated from Python to C, the second and fourth were classifiers. I had not set out deliberately to figure out quantum classification before quantum regression, but as this tale unfolds you'll hopefully see how my classification epiphany retrospectively seems obvious.

> SPELLCRAFTING
>
> The simplest classification neural network has two classifications. Numerically, one classification is represented by a zero and the other classification is represented by a one.
>
> The simplest classification neural network also has only one feature. Both classical implementations that I translated from Python to C had three features, but I selected one for simplicity. If I couldn't figure out quantum classification for one feature, I almost certainly wasn't going to figure it out for three features.

Although the feature I selected ranged in value from double-digit to triple-digit numbers, my classical implementation normalized the values to range between zero and one. The reason for this was simply that the classical model hadn't been training properly, and normalization fixed it. For testing purposes, the results that I wanted became the results that I saw.

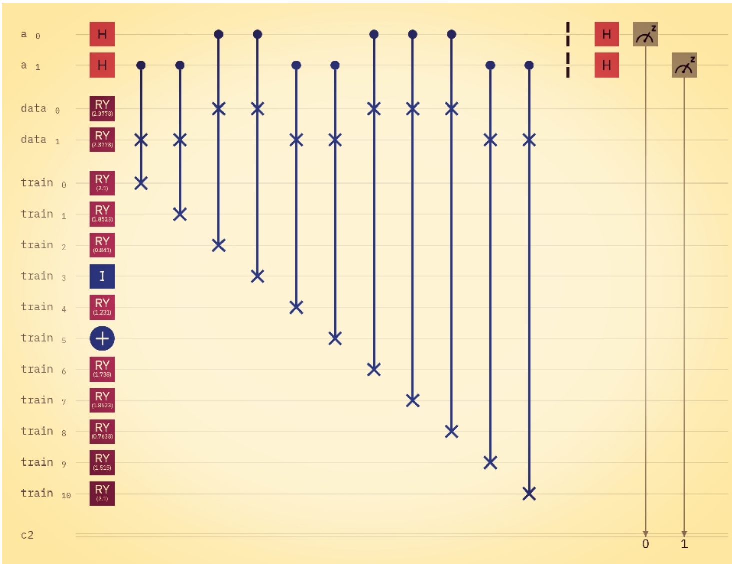
With such a grossly over-simplified model, the inner workings of a classification neural network become transparent: values closer to zero might be classified as the classification represented by zero, and values closer to one might be classified as the classification represented by one. In other words, the proximity of test data to each classification determines the recommended classification. And, once you start thinking about how close two values are together, or, conversely, how far apart, the term "distance" starts to circulate in the subconscious mind.

Distance. I was concentrating on quantum spellcrafting as I began to wonder: where had I seen that word before?

To re-craft a simple classical classification neural network as a quantum classification neural network, I first designated one classification as $|0 >$ and the other classification as $|1 >$. I then normalized some test data, but instead of normalizing the selected feature's values between zero and one, I normalized them between $0 * \pi$ and $1 * \pi$. Up to this point, I was still trying to shove a square classical peg into a round quantum hole.

Now, just like with an oversimplified classical neural network, you can just eyeball the results; you don't need to cast the spell at all. If the rotation of the test data is less than $0.5 * \pi$ then the recommended classification will be $|0 >$ and if the rotation of the test data is more than $0.5 * \pi$ then the recommended classification will be $|1>$. But, how could I make that determination after I start adding complexity and can no longer just eyeball it? I would need to be able to add more classifications and more features to cast any practical spell.

And, that there was the epiphany: all it is is a "distance measure." I realized that the fundamental quantum spell component for quantum classification spells just happens to be my all-time favorite quantum spell component: the SWAP Test.



I was still working with oversimplification at this point, but it was striking to me that this worked at all. After all, I hadn't trained a model. There were no weights. And, Mystra forbid anyone should ever craft a quantum sigmoid function. Mystra forbid.

I had crafted a very simple spell for quantum classification that in only a very small way resembled its classical inspiration. And, it worked.

Because I had experimented so extensively with the SWAP Test, I sure thought I had figured it all out. I now knew I had not yet even begun to put it through its paces.

7.1.4 MNIST

My quantum classification experiments culminated with crafting a Quantum MNIST spell. The idea originated with a drunken patron at Twitter Tavern, but I liked it and I ran with it. It is still my most viewed and most-read article on Medium. My goal in retelling that tale here is to go more behind-the-scenes into how I crafted that spell. Really, the goal of this whole tome is to explore the thought process of quantum spellcrafting in hopes of demystifying it a bit.

I'll go off on a quick tangent here because I'm writing about quantum machine learning. Hopefully, you've noticed that classical machine learning is an old friend. I'm not a chemist, so you won't read about me crafting quantum chemistry spells.

I think that's an important distinction. Start with what you know. I already told the tale of a portfolio optimization spell, but I neglected to mention how I first had to read up on what the efficient frontier is. While I know quite a bit about finance, it is not my profession, and crafting that spell required some research. I will later tell a few tales about quantum cryptography, and although I have some real-world experience with high-security computer networks, it is important to note that I was collaborating with someone with classical expertise.

Therefore, again, my advice to apprentice adventurers is to start with what you know. Now, back to my originally planned tale....

MNIST, for those unfamiliar, is a dataset of handwritten digits: the numbers zero through nine. It is practically unavoidable if you study classical machine learning and get into convolutional neural networks (CNN). The idea is to train a model to correctly recognize handwritten digits as the numbers they are intended to be.

Honestly, I skipped it. You can find classical MNIST spells everywhere, so it's a problem not in need of a solution. Plus, Python libraries like Keras take all the fun out of Deep Learning spellcrafting. So, I moved on to other projects and didn't look back.

And, that's why the challenge of crafting a quantum MNIST spell captured my attention. The only quantum MNIST spell I was aware of at the time was in the publicly available Tensorflow Quantum spellbook. At the time, the spell was limited to classifying the digits 3 and 6 only, plus both digits had to be compressed into 4x4 pixel images. That's very limited, and I wondered if I could outdo the entire Google faction.

SPELLCRAFTING

I decided, first and foremost, to use all ten digits. I had already crafted spells with more than two classifications, so I merely continued that expansion out to ten classifications. This spell was never going to work on any NISQ device anyway, so I might as well test the limits of the IBMQ simulator.

My next decision was to use grayscale. I think every QNN scroll I had read up until that point, including the Tensorflow Quantum scroll, used black and white; consequently, every pixel was binary encoded as only $|0>$ or $|1>$. So, I decided to use angle encoding to represent grayscale. I knew from experimentation that the SWAP Test would compare any quantum state to any other quantum state, and not just to the computational basis states.

Normalization is subtracting the minimum value from the input and dividing by the range of the values, resulting in a value between zero and one: ((input - min) / (max - min)). Multiplying the output by π provides the angle of rotation around the y axis. Another grayscale value can simultaneously be encoded via rotation around the z axis, but more on that in a bit.

I then decided to compress each image down to 16 pixels. I didn't want to, mostly because Tensorflow Quantum used 16 pixels, but I had to consider the limitations and performance characteristics of the IBMQ simulator.

I compressed the images classically. Had I known then what I know now, I would've tried to do this differently, but I essentially calculated mean digits. So, the mean zero has these mean values per pixel, and the mean one has these mean values per pixel, and so forth. I then summed the mean pixels of each mean digit into 16 blocks. Finally, I normalized all the blocks of all the digits between $0 * \pi$ and $1 * \pi$; these values would be my rotations.

Mathematically, each block represented 49 original pixels, each of which ranged in value from 0 to 255. The sum of a block, therefore, could be anywhere from 49 * 0 to 49 * 255. These sums, one per block, were the values that were normalized. Each rotation, therefore, represented 0 to 12,495 in original total pixel intensity.

Angle encoding, by the way, has always struck me as inefficient. Rotate around the y axis or rotate around the x axis, you still have a z axis you can work with. So, I decided to try this with double angle encoding. By rotating around the y axis to represent one block and by rotating around the z axis to represent another block, I could map all 16 blocks of 1 digit to only 8 qubits.

Because of how some digits resemble certain other digits, I devised a pattern to map the pairs of blocks to the qubits. I didn't want the two, three, eight, and nine, for example, to all have two nearly-identical qubits representing the four similar blocks representing the similarly curved tops of these four digits. Consequently, I mapped the blocks in a diagonal pattern to try to capture more of the differences between the digits.

My concern at this point, by the way, was the degree of the compression. I had opted to not use binary black-and-white pixels, but I had to wonder if mean grayscale blocks of pixels would just blur all the digits together.

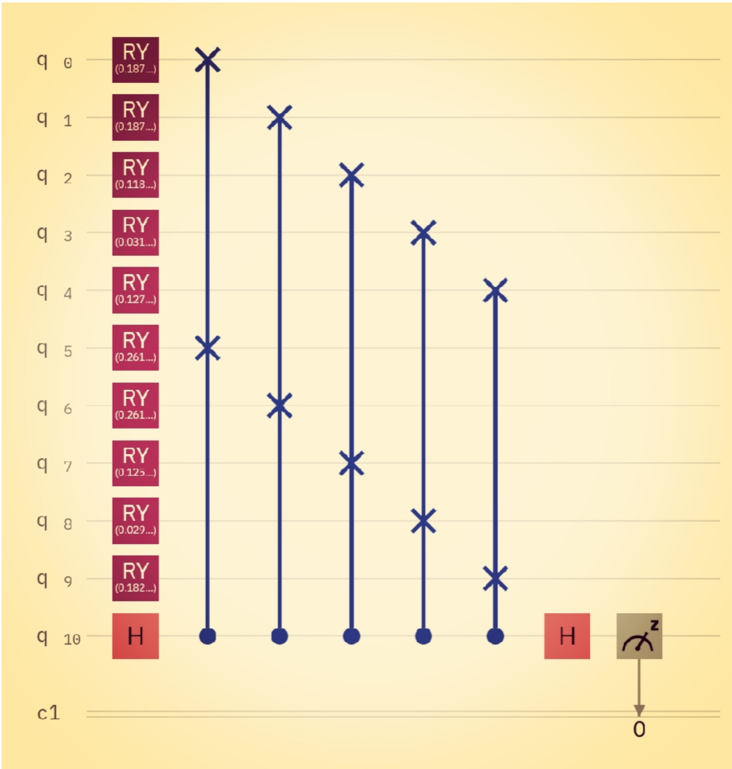
I pressed on anyway, and I added the main ingredient: the SWAP Test. Lots of them. I had already prepared a test zero in the same way that I had prepared the mean digits, except using the actual values instead of the means of values; the test zero was mapped to eight double-encoded qubits.

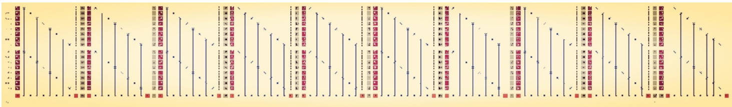
I used one ancilla qubit to compare the corresponding qubits of the test zero to the mean zero. Each Fredkin gate compared a specific test digit qubit to a specific mean digit qubit so that the two compressed blocks of the test digit were compared to the corresponding compressed blocks of the mean digit. That's important because I wouldn't want to compare the top of a seven to the bottom of a seven and get an inaccurate result.

I then reset the qubits and repeated this process to compare the test zero to the mean one. In total, I reset the qubits nine times and compared the test digit to all ten of the mean digits.

Despite the considerable extension of the SWAP Test from its canonical implementation, there remained only one measurement per ancilla qubit and only one ancilla qubit per comparison. Consequently, there were precisely ten measurements taken overall.

Due to the limited supply of spell components available, each comparison would have to take place sequentially; ideally, all ten comparisons could run in parallel. Fortunately, the simulator allowed resetting and recycling, something real hardware wasn't doing at the time. This would allow me to cast one high-level spell, which is less taxing than casting ten lower-level spells.


q 0
q 1
q 2
q 3
q 4
q 5
q 6
q 7
q 8
q 9
q 10
c1
RY (0.187...)
RY (0.187...)
RY (0.118...)
RY (0.031...)
RY (0.127...)
RY (0.261...)
RY (0.261...)
RY (0.125...)
RY (0.029...)
RY (0.182...)
H
H
z
0



I tested a random zero, and it worked. Of all ten measurements, the mean zero comparison measured zero with the highest probability, indicating that the mean zero was closer to the test zero than to the other nine mean digits.

Quite frankly, I was amazed. With all that compression, I would've been satisfied by simply avoiding obvious wrong answers such as one, four, and seven. Had a zero been classified as an eight, I could've taken that in stride as slightly wrong.

In the interest of full disclosure, I did not do extensive spell testing. The primary reason was that I had challenged myself to not write in Python tongue. I must concede, however, that that would've made the spellcrafting much, much easier. Instead, I wrote in OpenQASM and used Microsoft Excel for the classical pre-processing and post-processing.

Furthermore, I do not consider this spell to be optimized. Yes, this would never run on a NISQ device. And, yes, the IBMQ simulator has limitations. However, I confess there are days when I consider circling back to this spell, experimenting with different compression and encoding strategies that I have learned since, and hoping the simulator can still handle it.

My favorite thing about this spell is its potential future applications in such fields as astronomy, [28] neuroscience, and, well, anything important to do with imagery. These applications are no different than for classical machine learning, but it's exciting to think about what might be discovered on a quest to prove a quantum computational advantage.

Finally, it is important to note that considerable time passed between crafting a basic classification spell and this MNIST spell. Comparing eight qubits to eight other qubits using only one ancilla qubit was not an epiphany. I took many baby steps between these spells, and this spell merely took full strides in multiple directions. That's why I didn't have to ponder quite as much to craft this spell, nor did I have to ponder with the interim spells, for that matter. If you look closely at the spell, all I really did was play with Fredkin gates like they were puzzle pieces until their arrangement produced the desired results. MNIST simply had more jigsaw pieces than its predecessor experiments.

7.1.5 Clustering

Clustering is a natural follow-up to classification, especially after using the SWAP Test as a distance measure. With previous spells, I had the equivalent of classical training data. So, all these points are this classification, and all those points are that classification, and so forth. I simply compared test data points, via the SWAP Test, to the training data points; the classification of the nearest training data points is the recommended classification for the test data point.

But, what if I just have data points and they're not logically grouped? I have to group them first. Whatever the values represent, points closest to each other are more similar than points that are far away. Even more so than classification at a glance, clustering visually screams "distance measure."

As was with classification, oversimplified clustering can be eyeballed. But, and again like classification, it doesn't take much to add complication. I decided to see if the SWAP Test could correctly cluster groups of related data points.

SPELLCRAFTING

To craft my first quantum classification spell, I had thought about the simplest possible neural network. So, to craft a quantum clustering spell, I thought about the simplest possible clusters of points.

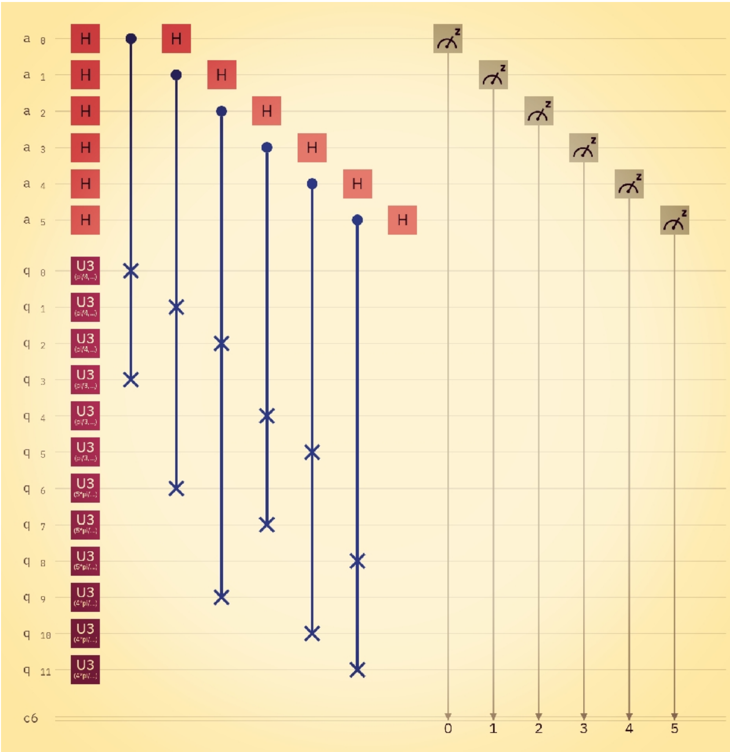
For testing purposes, the clusters needed to be blatantly obvious. Therefore, I rotated two qubits so that their quantum states would be near each other, and then I rotated two more qubits so that their quantum states would be near each other on the opposite side of the Bloch sphere. Visually, the two groups could not be more obvious, so the effects of this spell could only be correct or very wrong.

So, imagine points A and B together on one side of a Block sphere and points C and D together on the opposite side of the Bloch sphere.
I then set up six SWAP Tests. Each SWAP

Test measured the distance between two of the four quantum states until every possible combination was tested:

A-B
A-C
A-D
B-C
B-D
C-D

Because of the configuration of the quantum states, I expected two measurements to be zero with high probability, and the other four measurements to be zero with noticeably lower probabilities. That would indicate that only two groups of two quantum states were close together, just as I rotated them to be.



This was a very simple proof-of-principle, but at least it worked. I'm sure it could be developed further, a process through which it could hopefully be optimized, but I need a real-world problem to challenge myself with. For some reason, classification challenges have been far easier to come by, which is how crafting a simple quantum classification spell resulted in crafting the aforementioned Quantum MNIST spell.

And, like with classification, we have to ask whether or not quantum clustering has the potential to achieve a quantum computational advantage over classical approaches. I have found scrolls that suggest it does, [36, 28, 44] and I have found no scrolls that suggest it does not. Therefore, I intend to continue developing this spell.

7.1.6 Clustering & Classification

The only idea I've independently come up with for my quantum clustering spell was a new spell that could combine both clustering and classification into one spell. So, the spell would first create groups from existing data points, and then it would classify a new data point as belonging to one of those two groups. As I did with classification and then with clustering, I chose to create a very obvious problem; the results would have to either be correct or very wrong.

Another reason I kept the spell so simple was that I didn't have powerful enough ingredients to cast the combined spell with any respectable level of power. I knew before I started that I would have to work within the limitations of the IBMQ simulator.

SPELLCRAFTING

I began this new spell with my existing clustering spell. I even kept the quantum states exactly the same.

Skipping forward a bit, the principles of classification are the same. Therefore, the clustering and classification aspects of the spell were admittedly not inherently innovative; the innovation becomes apparent in combining the two spells.

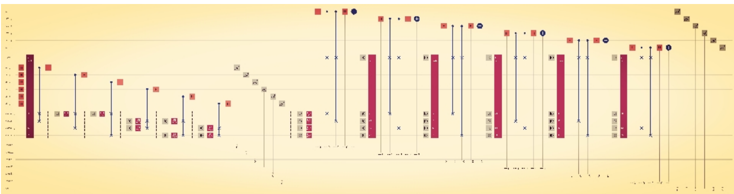
The first innovation was the use of syndrome bits. I had only seen them implemented in conjunction with quantum error correction (QEC), but they're not limited to that application. I noticed that they were used to detect errors, then conditional logic was used to correct those errors. So, I decided to use syndrome bits to detect clusters, and then I used conditional logic to improve the classification results. More on the latter in a bit....

The second innovation was the use of functions. Honestly, they added inefficiency to the spell. However, they made crafting the spell easier. And my modest goal was simply to get it to work, so I wasn't concerned with inefficiency for this particular spell. I generally avoid OpenQASM functions because they show up as non-descript rectangles on circuit diagrams, causing confusion and forcing me to have to explain what they are; that's never the case when the component gates are visible. If I didn't share my spells, however, I would be implementing OpenQASM functions like any Qiskit Advocate would use Python functions.

The third innovation was made possible by using conditional logic on the syndrome bits. I had six measurements; the higher the probability of measuring zero, the higher the probability that the qubit pair was a cluster. So, if a measurement was zero, I compared the test data point to that cluster. The higher the probability of that measurement being zero, the higher the probability that the test data should be classified with that cluster.

But, that wasn't the real innovation. I realized that the qubit pairs that were not clustered would still measure zero sometimes, and the test data would consequently be compared to the non-existing clusters. Therefore, I realized I could skew the results to minimize that. If a qubit pair measured one, indicating they were not a cluster, I could apply a Pauli-X instead of a SWAP Test. The Pauli-X would measure as a one, meaning that the test data should not be classified as that non-cluster. So, while the non-clusters would still measure zero sometimes and offer the potential for errant classification, the measuring of ones could be used to guarantee measuring ones for the second round of measurements.

My mathematical logic was in the way the SWAP Test indicates distance. Identical quantum states measure zero all the time. Maximally opposite states, however, still measure zero half the time. Therefore, I applied the logic that if a qubit pair is not a cluster, the classification measurement will be one every time the clustering measurement was one, plus the remaining measurements will be zero, at the most, half the time. Overall, there would be more one measurements, indicating that the test data should not be classified with non-clustered points.



It worked. I used one of the training data points as the test data point, which meant that the test data point should either be classified with that cluster, or something went very wrong. Not only was the test data point classified with the correct cluster with the highest probability, but the second-highest probability was also noticeably lower. Most importantly, the comparisons to the non-clusters were even lower than that.

The spread in the measurements builds confidence that the result is correct. If you have a greater-than-90% chance that a data point should be classified with an obvious cluster and all other probabilities are noticeably less than that, then that looks like a pretty good result. That was especially true in this case when the data was intentionally selected to either be obviously correct or fundamentally wrong.

On one last personal note, this consolidated spell got some great feedback in Twitter Tavern. Like a classical neural network, feedback is weighted. Some patrons just raised a glass, but their stature in the school makes such a gesture a huge compliment. After all, they could've continued with their conversations and ignored my announcement. Or, they could also have been critical of it. So the thumbs-ups, even from across the room, makes this spell a particularly fond one for me.

Plus, I would go on to independently discover other creative ways to use syndrome bits and conditional logic. My favorite, a tale I will tell later in this tome, is a way to do something that supposedly couldn't be done at the time.

7.2 Imperfect Cloning

I enjoyed reading the evolution of the no-cloning theorem. But, I couldn't help wondering, even if you can't perfectly clone an unknown quantum state, how close can we get?

I later found out either in Reddit Tavern or LinkedIn Tavern that there is an official answer to this question, but I am an experimentalist, so I probably would've crafted this spell anyway. I prefer casting spells and observing actual results.

SPELLCRAFTING

I had learned about quantum state tomography and how to measure any quantum state in the x, y, and z bases. [19] So, I wondered if I could use trigonometry on thousands of measurements to clone a state imperfectly, but to get it really, really close.

I first added a CNOT so that the second qubit would measure the unknown qubit in the z basis.

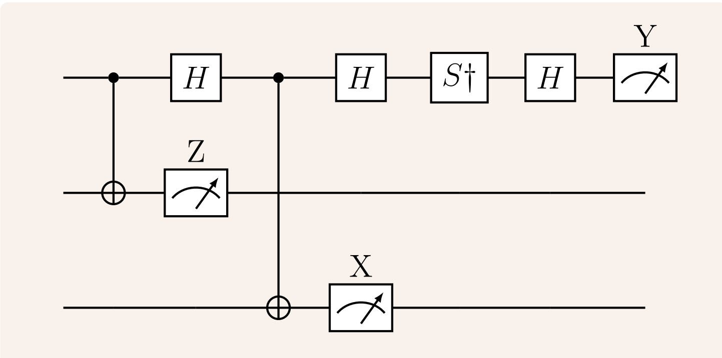
I then sandwiched a CNOT between two Hadamard gates so that the third qubit would measure the unknown qubit in the x basis. The second Hadamard gate simply rotates the qubit back to the z basis to show a textbook rotation to the y basis.

Finally, I added an S dagger gate and a Hadamard gate to measure in the y basis.

Mathematically, only the z basis measurements and one other basis's measurements are needed, but I measured all three bases because this whole exercise was for solely educational purposes anyway.

The trigonometry was similar to that of my financial portfolio optimization spell, but a little backward. The probability of measuring zero on any axis provides the length of the adjacent side of a right triangle, and the hypotenuse of the unit sphere is one. Therefore, you have enough information to calculate the rotation that results in the measurements that were already obtained.

In this case, I had created an unknown quantum state by applying a bunch of alternating Hadamard and T gates. Therefore, I was able to recreate that unknown state, plus a second state from the results of the trigonometry, and then compare the two qubits using a SWAP Test. Measuring zero at 100% would indicate a perfect clone, but that's impossible, right?



Using the maximum precision of every number available to me, the trigonometry-generated qubit measured 99.768% identical to the unknown quantum state. That's not perfect, but that's pretty darn close.

7.3 Encoding

There are quite a few ways to map classical data to qubits. Throughout my quantum computing adventure, I have developed an affinity for three of them. I learned none of them from tutorials, which seems to make them worth sharing.

7.3.1 Double Angle Encoding

I have used double angle encoding more than any other encoding method. Because of using normalization with classical neural networks, it seemed logical to multiply normalized values by π and encode values as rotations between $|0>$ and $|1>$.

The equation to normalize values, again, is ((input - min) / (max - min)). Simply multiply by π to use the output as a rotation.

I normally rotate around the y axis, however, I have since seen rotations around the x axis. But, the meridian that includes $|+>$ seems like a more logical starting point for rotations around the z axis, which is the second angle of double angle encoding.

This encoding method is quite practical. With it, as already noted within this tome, I can map classical data to qubits and perform distance measures with SWAP Tests. This approach has worked accurately for quantum classification and quantum clustering spells.

Recently, I discovered that this encoding method may be referred to as “dense angle encoding”. [23] I think “double” is more descriptive and more precise, but if “dense” is some sort of standard naming convention then feel free to roll with it. The important thing is that it’s useful.

7.3.2 Amplitude Encoding

I am sharing this particular tale because of the length of time that this technique eluded me. Before I cast my first quantum spell, I heard Dr. Seth Lloyd talk about mapping a terabit of data to only 40 qubits. Did my ears deceive me? I listened to the audio again and again. He definitely used the word "terabit" and the phrase "40 qubits."

I asked quite a few people. It couldn't be done, they said. Maybe only Dr. Lloyd can figure it out, one said.

Then one day, many, many months after I first heard Dr. Lloyd's fateful words, I read a sentence. It was just a random sentence in a random paragraph, but it explained in plain English how to amplitude encode a small number onto a small number of qubits.

> *"To encode the [number 5] on 4 qubits, we rotated the leftmost qubit by* $5/2n = 5/16$ *full turns (*$5/16 * 2\pi$ *radians). The next qubit is turned double this (*$10/16 * 2\pi$ *radians,*

> *or* 10/16 *full turns), this angle is then doubled for the qubit after, and so on."* [4]

I concentrated on it for quite some time. The encoding didn't have to be a small number at all. It could've been a much bigger number.

I realized that anything – everything, in fact – can be converted into binary. That binary number has a decimal value that can be encoded, just as that small number was, on any arbitrary number of qubits. I don't know for sure the rationale behind 40, but I guess that it's because 40 qubits have $2^4$0 dimensions and a terabit is $2^4$0 bits.

Amplitude encoding is associated with exponential storage advantage, [31] albeit with a tradeoff in performance; it takes time to prepare the quantum states. [6, 23, 25] Another disadvantage is that you can't decode a terabit of data mapped to 40 qubits; you can only measure 40 bits. [3]

For tasks such as classification, however, decoding isn't necessary. A terabit of data on 40 qubits can be compared to a terabit of data on another 40 qubits, and all you would need is 1 ancilla qubit, 2 Hadamards, and 40 Fredkin gates.

Too easy.

7.3.3 Binary Expansion

This is my favorite, albeit the most thoroughly impractical, encoding technique. I don't know how many times I passed it while reading "Mike & Ike," but one day I passed it again, did a double-take, and went back to read it again.

> *"How much information is represented by a qubit? Paradoxically, there are an infinite number of points on the unit sphere, so that in principle one could store an entire text of Shakespeare in the infinite binary expansion of θ."* [29]

Unfortunately, there is no accompanying explanation for how to do this. However, one method could be to take your binary string – again, everything can be converted to binary – and for each position n calculate the value of the bit as $1/2^n * 0/1$. Sum them all up and the result is a number between zero and one that can be multiplied by π and used as an angle of rotation.

Could this be used for Natural Language Processing (NLP)? Every word would have a different associated angle of rotation, with the most weight given to the first letter and less weight given to each subsequent letter. So, a word starting with "a" starts close to all other words starting with "a" and relatively far from words starting with any other letter. The differences in the angles of rotation get trivial quickly, though, so comparing anything more than one character might result in false identical matches.

Even if there is to be no practical application for this, you could always have fun encoding someone's name on a single qubit, measuring it a few thousand times, and presenting the resultant histogram as a birthday card. It's just cool, would be the bottom line.

7.4 Quantum Advantage

I don't know Big-O notation, [6, 29] and I have no desire to learn it. The number of steps required to solve a problem seems irrelevant when the size of the steps taken are not equal; I usually see proverbial apples being compared to proverbial oranges.[5, 22, 32, 36, 43, 45]

Instead, I consider the actual runtime. I've heard you can find this information somewhere, but it's more fun to test it yourself. You can cast arbitrary cantrips and arbitrary powerful spells with the same processor and that will give you, in the results, the range of runtime for that processor. Consequently, the only other information needed is the actual runtime of the classical equivalent. Compare the quantum runtime to the classical runtime; is there a quantum advantage or not?

If a spell cannot be cast, for whatever reason or reasons, then what do the scrolls say? My favorite quantum spell component, as you may have noticed, is the SWAP Test. I have quite a few scrolls in my library that propose that SWAP Tests will someday prove advantageous in regards to quantum machine learning. [2, 8, 18, 24] So if anyone would like to argue about whether or not my spells could ever prove to have an advantage, I just hold up my scroll collection and let the authors speak on my behalf. Feel free to argue with them; I'll be busy crafting my next spell.

One last guide is the number 50. Once your quantum states require around 50 qubits, the classical simulation would push the limits of today's most powerful supercomputers. [37, 32] So if you're only using five qubits, you might as well just cast a classical spell. But, if you need more than 50 qubits, especially if we're talking about post-NISQ fault-tolerant logical qubits, then quantum spells may indeed present computational advantages.

Finally, and even though I don't want to weigh in on the debate over the source of quantum computational advantage, I'll just share with you that you can find scrolls that attribute it to entanglement [1, 21, 33] as well as scrolls that state it can't be attributed to entanglement. [21, 33] Astute observers might notice that some of the citations on both sides of the argument are identical; interpret that as you wish.

7.5 Quantum Volume

Herein lies the most detailed explanation of quantum volume any adventurer needs to know: bigger numbers are better.

This is fun to experiment with, by the way. Cast a simple spell using different processors with different QV scores and, for the most part, the differences in error rates will be apparent. Take caution because that is not always the case – you could use the highest-error qubits and connections of a better device and the lowest-error qubits and connections of a worse device – but it is generally the case.

7.6 Pseudo-Conditional Logic

I'm including this tale in this tome because this spell was not officially possible when I crafted it.

The story goes that some IBMQ processors had been recently updated to allow the execution of reset gates. I must've been asking about future updates, including conditional logic. The official response was that conditional logic would not become executable on real hardware – it worked on the simulator [10] – until the release of OpenQASM3.

SPELLCRAFTING

I thought about how OpenQASM2 conditional logic works on the simulator; it's just an IF statement. If the syndrome bit or bits equal a specified value, execute the specified command.

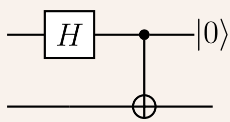
And then I thought about how controlled operations work. Let's look at the fundamental controlled operation: the controlled-NOT, or CNOT. If the control qubit measures zero, do nothing. But, if the control qubit measures one, then go ahead and bit flip the target qubit.

That looks like conditional logic to me!

There's one problem, though. And it's a biggie. The two qubits end up entangled. That can have unintended consequences later in the spell.

But, what if I can break the entanglement? What effect would that have?

So, I applied a reset gate to the control qubit.



It worked. Additional experimentation showed that a control qubit can be used to perform conditional operations on a target qubit, and resetting the control qubit breaks the entanglement while leaving the target qubit in its new state.

Therefore, I used conditional logic on a NISQ processor before it was officially possible.

8 The Fate of the World

The bards must've been telling my tale, because patrons at LinkedIn Tavern occasionally began pulling up chairs and joining me at my corner table. Some merely asked questions about my travels. Others, however, sought deeper knowledge.

Spells. They sought spells not found in any known spellbooks. They asked me to craft new spells out of unfamiliar ingredients. In most cases, the challenge was intoxicating and I accepted.

8.1 Machine Learning

The bards told tales of classification and clustering, so it is unsurprising that inquiries might relate to quantum machine learning.

The very first inquiry, in fact, was presented as a classification challenge. It should've been a simple matter of casting my quantum classification spell, but something was amiss in the results. My ability check had been successful, so an investigation was warranted. I soon discovered impurities in the ingredients.

A manual sifting through the ingredients showed an overlap that would similarly negate any equivalent classical spell. The problem was that each record could be classified as up to three different classifications. You would have to look at the training data to know which classification is correct, but you could look at neighboring classifications and see a fit.

The lesson here is that, just like with classical machine learning, the quality of the data is critically important. I had been used to experimenting with care-fully selected data because I needed to test that my spells were working. And, even in the case of MNIST, that is a highly-used dataset that has probably had numerous issues reported and resolved. This was my first time working with real-world data and, sure enough, we encountered a real-world issue.

Fortunately, I was able to suggest a way to reduce the problem somewhat and increase the probability of correct classification. But, it admittedly wasn't perfect.

8.1.1 Quantum Restricted Boltzmann Machines (QRBM)

Another inquiry concerned Quantum Restricted Boltzmann Machines (QRBM). I was able to take the provided data ingredients, modify the somatic component of my quantum classification spell appropriately, and produce correct results. However, the challenge remained to somehow craft and cast a QRBM spell.

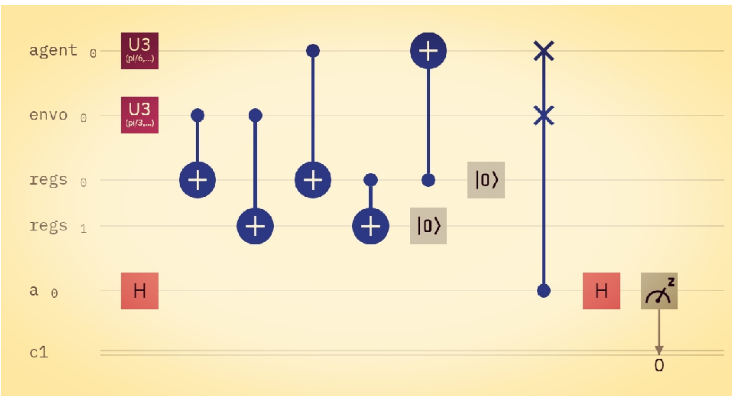
As I admitted earlier in this tome, not everything on this journey has been successful. Despite having dedicated most of my research to quantum machine learning, I look at QRBM and I just don't get it. I've had conversations with other researchers about it, and I still don't see any reason to delve any deeper into it short of the sheer personal challenge of making it happen. But, because my classification spell worked, the motivation is not quite there to make QRBM work for the sake of making it work.

8.1.2 Quantum Reinforcement Learning (QRL)

The most shocking inquiry to date has concerned Quantum Reinforcement Learning (QRL). Although I had not done reinforcement learning either classically or quantumly before, I quickly found a scroll with an appropriate spell. [12]

SPELLCRAFTING

I searched arXiv and found an existing spell. I copied the spell into my spellbook and it worked.



The moral of this particular tale is that you don't need to craft or modify spells if can find them ready to go in some dusty tome somewhere.

8.2 Cryptography

Although I have some real-world experience with high-security computer networks, my background does not include cryptography, network intrusion detection, nor key distribution. I hadn't experimented with super-dense coding yet, either.

What this means is that I occasionally misinterpreted and misunderstood the desired spell requirements. So while I cannot share in this tome the spells I crafted that were accepted, I can share the spells that were rejected. Though not useful for their intended purposes, maybe they can be used or modified for use for other applications.

8.2.1 GHZ Superdense QKD

This spell might just be my all-time favorite misunderstanding. The way I first misinterpreted the requirements, I wasn't sure it would be possible. And then the first way I thought to craft the spell, it didn't work. So, I had to do some troubleshooting to get it to work, which made it feel more rewarding when it finally worked.

Unfortunately, it wasn't what was needed.

But, that doesn't necessarily make the spell worthless. I'm sharing this tale in case anyone can find any other application for it.

My challenge was for Alice to share keys with Bob and Charlie. My misunderstanding was that I thought Alice would be sharing the same key with Bob and Charlie. But, no, she just needed to share keys at the same time, not to share the same keys.

SPELLCRAFTING

The spell starts as a textbook superdense coding spell. But, then I gave Alice an additional qubit and I gave Charlie a qubit and I entangled the qubit pair. At that point, Alice has two entangled pairs, one shared with Bob and one shared with Charlie.

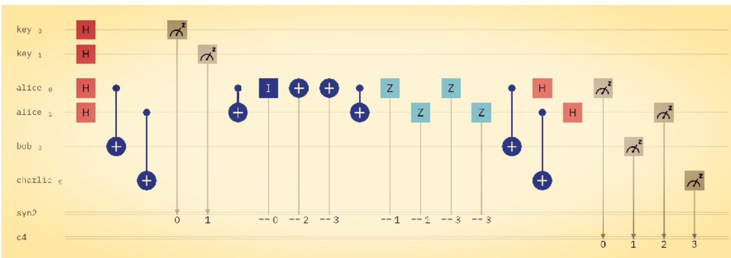
I then added a random key generator. It's just a couple of Hadamard gates being measured to a couple of syndrome bits. The advantage of doing this, though, is in the testing; every way Alice could encode the qubits is measured at the same time.

Syndrome bits and conditional logic are not the only way to do this, by the way. But, I chose this approach to simulate Alice manually deciding on some arbitrary key and then superdense coding it.

I then used conditional logic on the syndrome bits to implement identical superdense coding on Alice's qubits. This is the point where I had misunderstood the requirements for this spell; Alice's qubits should not have been identical.

But, I got Alice's qubits to be identical, even though I wasn't sure that I could. A CNOT worked for making the qubits identical in the z basis, but I just couldn't get it to work in the x basis. Sometimes entanglement draws a dagger and puts up a fight. My workaround was to apply conditional logic to both qubits, instead of just to one.

The spell finishes up as if it is two superdense coding spells cast in parallel.



The spell worked as I misunderstood it. Bob and Charlie did receive the same random key from Alice. Is there a practical application for this misunderstanding? Hopefully, a future update for this tome will reveal something in the affirmative.

8.2.2 Double Key Generator

This spell was probably rejected due to having unnecessary additional message security. After all, quantum cryptography is already secure. But, I didn't make this up out of the blue; I misunderstood a requirement, and this was the result.

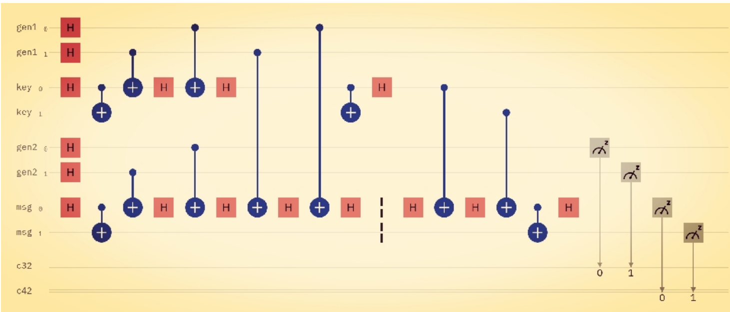
The misunderstanding was that the key itself should be scrambled. So, Bob couldn't just unscramble Alice's message using the textbook approach, because that message would be scrambled by a second key, which Bob would also need.

I recall proposing that additional layers could be added. Bob would no doubt be one seriously confused message recipient, but keys could be nested within keys until there are no more available qubits to work with. However, that was unnecessary given that even one scrambled key was not desired.

SPELLCRAFTING
Okay, okay, this text box isn't really necessary.

I'm just including it for consistency.

This spell is just random randomness, as evident by its exclusive use of Hadamards and CNOTs. It's random in both the z and x bases for superdense coding purposes, but the spell is really just "hello, quantum world" on steroids.



Like the spell resulting from my previous misunderstanding, this spell also worked as I misunderstood it. Also, like the previous spell, I don't know if there's any practical application for it. But, it doesn't do nothing, and perhaps that's something?

8.3 Image Processing

Sadly, the assembled party never left the tavern. We had a diverse group of talented characters gathered in a private room, ready to venture, we believed, where no adventurers had ventured before. There was palpable excitement and enthusiasm, but then nothing happened. The room went silent. Perhaps everyone was lost in their thoughts, but I snuck back out into the tavern and still, no one seems to have noticed. Or, maybe they've all since left, too? I may never know.

At least the time I had invested into this project had not been fully wasted. I discovered a tome containing powerful image processing spells. [13] I still remember the sense of shock as I turned each page and discovered how beautifully explained everything was, as well as how complete. QIMP is not some theorized potential future application; the spells I found can be cast today. The spells would be limited to very small pixel or voxel counts, but that's still enough to cast a proof-of-principle demonstration spell.

What's worse, considering we never proceeded, is that we had a fantastic application selected. I don't want to reveal too much, just in case the team eventually decides to venture off without me, but it involves medical imaging in a way that does not seem to have been explored. I would find it thoroughly rewarding to someday help craft a quantum spell that has medical applications.

Beyond the general challenge of medical imaging classically, let alone quantumly, this project seemed ideal for someday demonstrating quantum computational advantage.

8.4 Confidential

Rule One of working on confidential projects is: I am not working on any confidential projects.

8.5 Tentative

As I wrote this tome, a new stranger pulled up a chair and sat down at my table. I have no details, but the potential for a new adventure is always exciting.

This page is a placeholder for hopefully telling a grand new tale in a future edition of this tome.

8.6 Regression

One of the challenges I set for myself in writing this tome was to craft a new spell in real-time. In other words, instead of recollecting about spellcrafting and transcribing my memories, I wanted to craft a new spell while a bard observes with quill and parchment.

Unfortunately, I am now rushing to finish this tome in anticipation of a new project. Rest assured, if the feedback on this tome is sufficiently positive, I definitely want to craft a quantum regression spell for a second edition. I will transcribe my thoughts in real-time, hopefully resulting in a thoroughly detailed "spellcrafting" callout.

8.7 Wish List

Where do I want to go next on this grand adventure? Well, I thoroughly enjoy the challenges presented to me by the patrons at LinkedIn Tavern. But, that has historically allowed me some free time, so I've thumbed through my notes and assembled my current wish list of quantum spellcrafting ideas that have not already been presented earlier in this tome.

8.7.1 Music

I don't know much about music, but the practice of quantum music – akin to quantum image processing – is surprisingly well developed. [27] It might be fun to collaborate with a musically-inclined partner to see what we can do together in this area.

8.7.2 Decision Trees

I don't use classical decision trees all that often, but this seems like something that shouldn't be all that difficult to do quantumly. I would want to do some research on the possibility of achieving quantum advantage with it, but doing it should be relatively easy.

8.7.3 GHZ Double Key Generator

In a variation of my rejected spell in which Alice sends duplicate keys to Bob and Charlie, I had intended at the time to modify the spell so that Bob and Charlie each shared GHZ states with Alice. For entertainment purposes only, I would like to solve that puzzle someday.

8.7.4 GHZ Double Scramble

This would also be more of a puzzle than it would likely be of any practical value, whether to quantum cryptography or anything else. Since much of what I do is based on the thrill of just getting it to work, this is something that I think I can get to work.

Imagine generating a truly-random key. Except, we don't need that, so it could represent a string or a number or anything. But, I'm going to keep referring to it as a key for simplicity.

Then imagine generating a key that scrambles the first key. The idea is that you need the first key, but the first key is scrambled, and you can't unscramble it without the second key. The way I crafted the simple version of this spell allows for nesting so that you could conceivably need an nth key to unscramble and (nth-1) key that you would need to unscramble and (nth-2) key, and so forth. And, it would use GHZ states.

8.7.5 Database

This is inspired by a quantum dictionary that stores key-value pairs. [17] The paper made me wonder if the concept could be extended to build something resembling a database table, or even a database with multiple tables.

8.7.6 Binning

This is inspired by mapping classical mean values to qubits. Using SWAP Tests to measure distances to these mean values compares points to points, but classical data is often not best represented by single points.

One way to do this could be to classically bin the data, calculate the means of the bins, and then map the means of the bins to qubits. For expeditiousness, sadly, I've already done that. However, this feels like shoving a square classical peg into a round quantum hole. I suspect that a round quantum peg is available.

9 OpenQASM3

"OpenQASM 3: A broader and deeper quantum assembly language" [11] was released while I was already editing this tome. I expect it to play a central role in any update to this tome.

10 Epilogue

I hope you had as much fun reading this tome as I had writing it. Quantum computing is far too interesting to be relegated to boring books, papers, blogs, videos, and whatever. Honestly, maintaining a creative theme over so many pages is challenging; I tried my best.

I've had positive reviews of my literary creativity when the documents have been limited to several pages, and so I gambled that I could maintain that level of creativity throughout an entire book. I had to simultaneously try to ensure that I got my actual quantum spellcasting points across, so hopefully the real meat and potatoes weren't buried under the garnishes.

And this brings me to the bottom line: the heart of this book is quantum algorithm design. I focused primarily on quantum machine learning because that's what I understand the best. Hopefully, my thought processes make sense outside of the confines of my head.

TL;DR: if you understand how a classical algorithm works fundamentally and you understand how the quantum building blocks work, you can hopefully design a quantum algorithm that achieves a desired effect with quantum computational advantage.

References

[1] Scott Aaronson. Introduction to quantum information science lecture notes. https://www.scottaaronson.com/blog/?p=3943, 2018.

[2] Blank C. Park D.K. Rhee JK.K. et al. Quantum classifier with tailored quantum kernel. *npj Quantum Inf*, 6(41), 2020.

[3] P.S. Emani J. Warrell A. Anticevic et al. Quantum computing at the frontiers of biological sciences. *Nat Methods*, 2021.

[4] Abraham Asfaw, Antonio Corcoles, Luciano Bello, Yael Ben-Haim, Mehdi Bozzo-Rey, Sergey Bravyi, Nicholas Bronn, Lauren Capelluto, Almudena Carrera Vazquez, Jack Ceroni, Richard Chen, Albert Frisch, Jay Gambetta, Shelly Garion, Leron Gil, Salvador De La Puente Gonzalez, Francis Harkins, Takashi Imamichi, Hwajung Kang, Amir h. Karamlou, Robert Loredo, David McKay, Antonio Mezzacapo, Zlatko Minev,

Ramis Movassagh, Giacomo Nannicini, Paul Nation, Anna Phan, Marco Pistoia, Arthur Rattew, Joachim Schaefer, Javad Shabani, John Smolin, John Stenger, Kristan Temme, Madeleine Tod, Stephen Wood, and James Wootton. Learn quantum computation using qiskit, 2020.

[5] Ryan Babbush, Jarrod McClean, Craig Gidney, Sergio Boixo, and Hartmut Neven. Focus beyond quadratic speedups for error-corrected quantum advantage, 2020.

[6] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, Sep 2017.

[7] Yudong Cao, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. Quantum neuron: an elementary building block for machine learning on quantum computers, 2017.

[8] Lukasz Cincio, Yiğit Subaşı, Andrew T Sornborger, and Patrick J Coles. Learning the quan-

tum algorithm for state overlap. *New Journal of Physics*, 20(11):113022, Nov 2018.

[9] Antonio D. Corcoles, Maika Takita, Ken Inoue, Scott Lekuch, Zlatko K. Minev, Jerry M. Chow, and Jay M. Gambetta. Exploiting dynamic quantum circuits in a quantum algorithm with superconducting qubits, 2021.

[10] Andrew W. Cross, Lev S. Bishop, John A. Smolin, and Jay M. Gambetta. Open quantum assembly language, 2017.

[11] Andrew W. Cross, Ali Javadi-Abhari, Thomas Alexander, Niel de Beaudrap, Lev S. Bishop, Steven Heidel, Colm A. Ryan, John Smolin, Jay M. Gambetta, and Blake R. Johnson. Openqasm 3: A broader and deeper quantum assembly language, 2021.

[12] F. A. Cárdenas-López, L. Lamata, J. C. Retamal, and E. Solano. Multiqubit and multilevel quantum reinforcement learning with quantum technologies. *PLOS ONE*, 13(7):e0200455, Jul 2018.

[13] Salvador E. Venegas-Andraca Fei Yan. *Quantum Image Processing*. Springer, 2020.

[14] Steph Foulds, Viv Kendon, and Tim Spiller. The controlled swap test for determining quantum entanglement, 2021.

[15] S´ebastien Gambs. Machine learning in a quantum world, 2006.

[16] Juan Carlos Garcia-Escartin and Pedro Chamorro-Posada. swap test and hong-ou-mandel effect are equivalent. *Physical Review A*, 87(5), May 2013.

[17] Austin Gilliam, Charlene Venci, Sreraman Muralidharan, Vitaliy Dorum, Eric May, Rajesh Narasimhan, and Constantin Gonciulea. Foundational patterns for efficient quantum computing, 2021.

[18] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, Mar 2019.

[19] Abhijith J., Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, Andreas Bärtschi, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, David Gunter, Satish Karra, Nathan Lemons, Shizeng Lin, Alexander Malyzhenkov, David Mascarenas, Susan Mniszewski, Balu Nadiga, Daniel O'Malley, Diane Oyen, Scott Pakin, Lakshman Prasad, Randy Roberts, Phillip Romero, Nandakishore Santhi, Nikolai Sinitsyn, Pieter J. Swart, James G. Wendelberger, Boram Yoon, Richard Zamora, Wei Zhu, Stephan Eidenbenz, Patrick J. Coles, Marc Vuffray, and Andrey Y. Lokhov. Quantum algorithm implementations for beginners, 2020.

[20] Tyson Jones, Suguru Endo, Sam McArdle, Xiao Yuan, and Simon C. Benjamin. Variational quantum algorithms for discovering hamiltonian spectra. *Phys. Rev. A*, 99:062304, Jun 2019.

[21] Richard Jozsa and Noah Linden. On the role of entanglement in quantum-computational speedup. *Proceedings of the Royal Society of London.*

Series A: Mathematical, Physical and Engineering Sciences, 459(2036):2011–2032, Aug 2003.

[22] Vipin Kumar. *Introduction to Parallel Computing*. Addison-Wesley Longman Publishing Co., Inc., USA, 2nd edition, 2002.

[23] Ryan LaRose and Brian Coyle. Robust data encodings for quantum classifiers. *Physical Review A*, 102(3), Sep 2020.

[24] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning, 2013.

[25] Danijela Marković and Julie Grollier. Quantum neuromorphic computing. *Applied Physics Letters*, 117(15):150501, Oct 2020.

[26] Olivia Di Matteo, Anna McCoy, Peter Gysbers, Takayuki Miyagi, R. M. Woloshyn, and Petr Navrátil. Improving hamiltonian encodings with the gray code, 2021.

[27] Eduardo R. Miranda. Quantum computer: Hello, music!, 2020.

[28] Wiebe Nathan, Kapoor Ashish, and Svore Krysta M. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Info. Comput.*, 15(3–4):316–356, 2015.

[29] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[30] Noam Nisan and Shimon Schocken. *The Elements of Computing Systems*. The MIT Press, Cambridge, Massachusetts, 2005.

[31] Anupam Prakash. *Quantum Algorithms for Linear Algebra and Machine Learning*. PhD thesis, EECS Department, University of California, Berkeley, Dec 2014.

[32] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.

[33] Eleanor Rieffel and Wolfgang Polak. *Quantum Computing*. The MIT Press, Cambridge, Massachusetts, 2011.

[34] Maria Schuld. Quantum machine learning models are kernel methods, 2021.

[35] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical Review Letters*, 122(4), Feb 2019.

[36] Steven S Skiena. *The Algorithm Design Manual*. Springer, London, 2008.

[37] Robert S. Sutor. *Dancing with Qubits*. Packt Publishing, 2019.

[38] Francesco Tacchino, Panagiotis Barkoutsos, Chiara Macchiavello, Ivano Tavernelli, Dario Gerace, and Daniele Bajoni. Quantum implementation of an artificial feed-forward neural network. *Quantum Science and Technology*, 5(4):044010, Oct 2020.

[39] Francesco Tacchino, Chiara Macchiavello, Dario Gerace, and Daniele Bajoni. An artificial neuron implemented on an actual quantum processor. *npj Quantum Information*, 5(1), Mar 2019.

[40] The NASM Development Team. Nasm – the netwide assembler. https://nasm.us/docs.php, 2017.

[41] Wizards RPG Team. *Dungeons & Dragons: Basic Rules Version 0.3*. Wizards of the Coast, 5th edition, 2015.

[42] Andrew W. Trask. *grokking Deep Learning*. Manning Publications Co., Shelter Island, NY, 2019.

[43] G Wendin. Quantum information processing with superconducting circuits: a review. *Reports on Progress in Physics*, 80(10):106001, Sep 2017.

[44] P. Wittek. *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. Elsevier Insights. Elsevier Science, 2014.

[45] N.S. Yanofsky and M.A. Mannucci. *Quantum Computing for Computer Scientists*. Cambridge University Press, 2008.

A Spellbook

The illustrations presented throughout this tome are my original spells, as I crafted them at the time I crafted them. Like all spellcrafters, I look back at my early work and see room for improvement. In some cases, I see room for quite a bit of improvement.

Therefore, this page is a placeholder. Depending on how the first edition of this tome is received, I might revisit select spells, update them, and share how I should have crafted them, had I known then what I know now.

The prime candidate, just off the top of my head, would be my Quantum MNIST spell. My goal at the time was just to get it to work. But, the extent of the classical pre-processing and post-processing, the way I mapped the classical values, and even the way I did the comparisons all show potential for improvement.

Not only have I acquired new knowledge since crafting these spells, but the passage of time is also great enough that I can look upon my spellbook with fresh eyes, be self-critical, and hopefully craft new spells worthy of the digital paper they're not actually printed on.

B Optimization

I have never found a comprehensive guide to quantum circuit optimization. I've picked up a technique over here and a strategy over there, but the information is scattered.

What's worse, much of the information is incorrect. At least two papers come to mind that suggested specific ways to reduce circuit noise, but I've been able to have those methods debunked in Qiskit Slack Tavern.

This subsection is reserved for a comprehensive guide to circuit optimization to be included in the next edition of this tome. My disclaimer will be the same as the preceding paragraph, however: check with your faction, your preferred library, and your specific device for all the optimization options available to you.