Introduction

What are quantum computers good for?

> Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.
>
> *Simulating Physics with Computers*, Feynman (1981)

Quantum computers harness the principles of quantum theory such as superposition and entanglement to solve information-processing tasks. In the last 42 years, quantum computing has gone from theoretical speculations to the implementation of machines that can solve problems beyond what is possible with classical means. This section will sketch a brief and biased history of the field and of its future challenges.

In 1980, Benioff [Ben80] takes the abstract definition of a computer and makes it physical: he designs a quantum mechanical system whose time evolution encodes the computation steps of a given Turing machine. In retrospect, this may be taken as the first proof that quantum mechanics can simulate classical computers. The same year, Manin [Man80] looks at the opposite direction: he argues that it would take exponential time for a classical computer to simulate a generic quantum system. Feynman [Fey82; Fey85] comes to the same conclusion and suggests a way to simulate quantum mechanics much more efficiently: building a quantum computer!

So what are quantum computers good for? Feynman's intuition gives us a first, trivial answer: at least quantum computers could simulate quantum mechanics efficiently. Deutsch [Deu85] makes the question formal by defining quantum Turing machines and quantum circuits. Deutsch and Jozsa [DJ92] design the first quantum algorithm and prove that it solves *some* problem exponentially faster than any classical *deterministic*[1] algorithm. Simon [Sim94] improves on their result by designing a problem that a quantum computer can solve exponentially faster than any classical algorithm. Deutsch-Jozsa and Simon relied on oracles[2] and promises[3] and their problems have little practical use. However, they inspired Shor's algorithm [Sho94] for prime factorisation and discrete logarithm. These two problems are believed to require exponential time for a classical computer and their hardness is at the basis of the public-key cryptography schemes currently used on the internet.

[1] A classical *randomised* algorithm solves the problem in constant time with high probability.
[2] An oracle is a black box that allows a Turing machine to solve a certain problem in one step.
[3] The input is promised to satisfy a certain property, which may be hard to check.

In 1997, Grover provides another application for quantum computers: "searching for a needle in a haystack" [Gro97]. Formally, given a function $f : X \to \{0, 1\}$ and the promise that there is a unique $x \in X$ with $f(x) = 1$, Grover's algorithm finds x in $O(\sqrt{|X|})$ steps, quadratically faster than the optimal $O(|X|)$ classical algorithm. Grover's algorithm may be used to brute-force symmetric cryptographic keys twice bigger than what is possible classically [BBD09]. It can also be used to obtain quadratic speedups for the exhaustive search involved in the solution of NP-hard problems such as constraint satisfaction [Amb04]. Independently, Benett et al. [Ben+97] prove that Grover's algorithm is in fact optimal, adding evidence to the conjecture that quantum computers cannot solve these NP-hard problems in polynomial time. Chuang et al. [CGK98] give the first experimental demonstration of a quantum algorithm, running Grover's algorithm on two qubits.

Shor's and Grover's discovery of the first real-world applications sparked a considerable interest in quantum computing. The core of these two algorithms has then been abstracted away in terms of two subroutines: phase estimation [Kit95] and amplitude amplification [Bra+02], respectively. Making use of both these subroutines, the HHL[4] algorithm [HHL09] tackles one of the most ubiquitous problems in scientific computing: solving systems of linear equations. Given a matrix $A \in \mathbb{R}^{n \times n}$ and a vector $b \in \mathbb{R}^n$, we want to find a vector x such that $Ax = b$. Under some assumptions on the sparsity and the condition number of A, HHL finds (an approximation of) x in time logarithmic in n when a classical algorithm would take quadratic time simply to read the entries of A. This initiated a new wave of enthusiasm for quantum computing with the promise of exponential speedups for machine learning tasks such as regression [WBL12], clustering [LMR13], classification [RML14], dimensionality reduction [LMR14] and recommendation [KP16]. The narrative is appealing: machine learning is about finding patterns in large amounts of data represented as high-dimensional vectors and tensors, which is precisely what quantum computers are good at.

However, the exponential speedup of HHL comes with some caveats, thoroughly analysed by Aaronson [Aar15]. Two of these challenges are common to many quantum algorithms: 1) the efficient encoding of classical data into quantum states and 2) the efficient extraction of classical data via quantum measurements. Indeed, what HHL really takes as input is not a vector b but a quantum state $|b\rangle = \sum_{i=1}^{n} b_i |i\rangle$ called its amplitude encoding. Either the input vector b has enough structure that we can describe it with a simple, explicit formula. This is the case for example in the calculation of electromagnetic scattering cross-sections [CJS13]. Or we assume that our classical data has been loaded onto a quantum random-access memory (qRAM) that can prepare the state in logarithmic time [GLM08]. Not only is qRAM a daunting challenge from an engineering point of view, in some cases it also requires too much error correction for the state preparation to be efficient [Aru+15]. Symmetrically, the output of HHL is not the solution vector x itself but a quantum state $|x\rangle$ from which we can measure some observable $\langle x|M|x\rangle$. If preparing the state $|b\rangle$ requires a number of gates exponential in the number of qubits, or if we need exponentially many measurements of $|x\rangle$ to compute our

[4]Named after its discoverers Harrow, Hassidim and Lloyd.

classical output, then the quantum speedup disappears.

Shor, Grover and HHL all assume *fault-tolerant* quantum computers [Sho96]. Indeed, any machine we can build will be subject to noise when performing quantum operations, errors are inevitable. We need an error correcting code that can correct these errors faster than they appear. This is the content of the *quantum threshold theorem* [AB08] which proves the possibility of fault-tolerant quantum computing given physical error rates below a certain threshold. One noteworthy example of such a quantum error correction scheme is Kitaev's toric code [Kit03] and the general idea of topological quantum computation [Fre+03] which offers the long-term hope for a quantum computer that is fault-tolerant "by its physical nature". However this hope relies on the existence of quasi-particles called Majorana zero-modes, which as of 2021 has yet to be experimentally demonstrated [Bal21].

The road to large-scale fault-tolerant quantum computing will most likely be a long one. So in the meantime, what can we do with the noisy intermediate-scale quantum machines we have available today, in the so-called NISQ era [Pre18]? Most answers involve a hybrid classical-quantum approach where a classical algorithm is used to optimise the preparation of quantum states [McC+16]. Prominent examples include the quantum approximate optimisation algorithm (QAOA [FGG14]) for combinatorial problems such as maximum cut and the variational quantum eigensolver (VQE [Per+14]) for approximating the ground state of chemical systems. These variational algorithms depend on the choice of a parameterised quantum circuit called the *ansatz*, based on the structure of the problem and the resources available. Some families of ansätze such as instantaneous quantum polynomial-time (IQP) circuits are believed to be hard to simulate classically even at constant depth [SB09], opening the door to potentially near-term NISQ speedups.

Although the hybrid approach first appeared in the context of machine learning [Ban+08], the idea of using parameterised quantum circuits as machine learning models went mostly unnoticed for a decade [BLS19]. It was rediscovered under the name of quantum neural networks [FN18] then implemented on two-qubits [Hav+19], generating a new wave of attention for quantum machine learning. The idea is straightforward: 1) encode the input vector $x \in \mathbb{R}^n$ as a quantum state $|\phi_x\rangle$ via the ansatz of our choice, 2) initialise a random vector of parameters $\theta \in \mathbb{R}^d$ and encode it as a measurement M_θ, again via some choice of ansatz 3) take the probability $y = \langle\phi(x)|M_\theta|\phi(x)\rangle$ as the prediction of the model. A classical algorithm then uses this quantum prediction as a subroutine to find the optimal parameters θ in some data-driven task such as classification.

One of the many challenges on the way to solving real-world problems with parameterised quantum circuits is the existence of *barren plateaus* [McC+18]: with random circuits as ansatz, the probability of non-zero gradients is exponentially small in the number of qubits and our classical optimisation gets lost in a flat landscape. One can help but notice the striking similarity with the vanishing gradient problem for classical neural networks, formulated twenty years earlier [Hoc98]. Barren plateaus do not appear in circuits with enough structure such as quantum convolutional networks [Pes+21], they can also be mitigated by structured initialisation strategies [Gra+19]. Another direction is to avoid gradients altogether and use

kernel methods [SK19]: instead of learning a measurement M_θ, we use our NISQ device to estimate the distance $|\langle\phi_{x'}|\phi_x\rangle|^2$ between pairs of input vectors $x, x' \in \mathbb{R}^n$ embedded in the high-dimensional Hilbert space of our ansatz. We then use a classical support vector machine to find the optimal hyperplane that separates our data, with theoretical guarantees to learn quantum models at least as good as the variational approach [Sch21].

Random quantum circuits may be unsuitable for machine learning, but they play a crucial role in the quest for *quantum advantage*, the experimental demonstration of a quantum computer solving a task that cannot be solved by classical means in any reasonable time. We are back to Feynman's original intuition: sampling from a random quantum circuit is the perfect candidate for such a task. The end of 2019 saw the first claim of such an advantage with a 53-qubit computer [Aru+19]. The claim was almost immediately contested by a classical simulation of 54 qubits in two and a half days [Ped+19] then in five minutes [Yon+21]. Zhong et al. [Zho+20] made a new claim with a 76-photon linear optical quantum computer followed by another with a 66-qubit computer [Wu+21; Zhu+21]. They estimate that a classical simulation of the sampling task they completed in a couple of hours would take at least ten thousand years.

Now that quantum computers are being demonstrated to compute something beyond classical, the question remains: can they compute something *useful*?

Why make NLP quantum?

> A girl operator typed out on a keyboard the following Russian text in English characters: "Mi pyeryedayem mislyi posryedstvom ryechi". The machine printed a translation almost simultaneously: "We transmit thoughts by means of speech." The operator did not know Russian.
>
> *New York Times* (8th January 1954)

The previous section hinted at the fact that quantum computing cannot simply solve any problem faster. There needs to be some structure that a quantum computer can exploit: its own structure in the case of physics simulation or the group-theoretic structure of cryptographic protocols in Shor's algorithm. So why should we expect quantum computers to be any good at natural language processing (NLP)? This section will argue that natural language shares a common structure with quantum theory, in the form of two linguistic principles: *compositionality* and *distributionality*.

The history of artificial intelligence (AI) starts in 1950 with a philosophical question from Turing [Tur50]: "Can machines think?" reformulated in terms of a game, now known as the Turing test, in which a machine tries to convince a human interrogator that it is human too. In order to put human and machine on an equal footing, Turing suggests to let them communicate only via written language: his thought experiment actually defined an NLP task. Only four years later, NLP goes from philosophical speculation to experimental demonstration: the

IBM 701 computer successfully translated sentences from Russian to English such as "They produce alcohol out of potatoes." [Hut04]. With only six grammatical rules and a 250-word vocabulary taken from organic chemistry and other general topics, this first experiment generated a great deal of public attention and the overly-optimistic prediction that machine translation would be an accomplished task in "five, perhaps three" years.

Two years later, Chomsky [Cho56; Cho57] proposes a hiearchy of models for natural language syntax which hints at why NLP would not be solved so fast. In the most expressive model, which he argues is the most appropriate for studying natural language, the parsing problem is in fact Turing-complete. Let alone machine translation, merely deciding whether a given sequence of words is grammatical can go beyond the power of any physical computer. Chomsky's parsing problem is a linguistic reinterpretation of an older problem from Thue [Thu14], now known as the *word problem for monoids*[5] and proved undecidable by Post [Pos47] and Markov [Mar47] independently. This reveals a three-way connection between theoretical linguistics, computer science and abstract algebra which will pervade much of this thesis. But if we are interested in solving practical NLP problems, why should we care about such abstract constructions as formal grammars?

Most NLP tasks of interest involve natural language *semantics*: we want machines to compute the *meaning* of sentences. Given the grammatical structure of a sentence, we can compute its meaning as a function of the meanings of its words. This is known as the *principle of compositionality*, usually attributed to Frege [Fre84] although he never stated it[6]. It was already implicit in Boole's *laws of thought* [Boo54] and then made explicit by Carnap [Car47]. From a philosophical principle, compositionality became the basis of the symbolic approach to NLP, also known as *good old-fashioned artificial intelligence* (GOFAI) [Hau89]. The meanings of words is first encoded in a machine-readable format, then the computer can answer complex questions by composing these meanings together. This approach culminated in 2011 with IBM Watson defeating a human champion at *Jeopardy!* [LF11].

The same year, Apple deploy their virtual assistant in the pocket of millions of users, soon followed by internet giants Amazon and Google. While Siri, Alexa and their competitors have made NLP mainstream, none of them make any explicit use of formal grammars. Instead of the complex grammatical analysis and knowledge representation of expert systems like Watson, the AI of these next-generation NLP machines is powered by deep neural networks and machine learning of big data. Although their architecture got increasingly complex, these neural networks implement a simple statistical concept: *language models*, i.e. probability distributions over sequences of words. Instead of the compositionality of symbolic AI, these statistical methods rely on another linguistic principle, *distributionality*: words with similar distributions have similar meanings.

This principle may be traced back to Wittgenstein's *Philosophical Investigations*:

[5] Historically, Thue, Markov and Post were working with *semigroups*, i.e. unitless monoids.

[6] What Frege did state is now known as the *context principle*: "it is enough if the sentence as whole has meaning; thereby also its parts obtain their meanings". This can be taken as a kind of dual to compositionality: the meanings of the words are functions of the meaning of the sentence.

"the meaning of a word is its use in the language" [Wit53], usually shortened into the slogan *meaning is use*. It was then formulated in the context of computational linguistics by Harris [Har54], Weaver [Wea55] and Firth [Fir57], who coined the famous quotation: "You shall know a word by the company it keeps!" Before deep neural networks took over, the standard way to formalise distributionality had been *vector space models* [SWY75]. We have a set of N words appearing in a set of M documents and we simply count how many times each word appears in each document to get a $M \times N$ matrix. We normalise it with a weighting scheme like tf-idf (term frequency by inverse document frequency), factorise it (via e.g. singular value decomposition or non-negative matrix factorisation) and we're done! The columns of the matrix encode the meanings of words, taking their inner product yields a measure of word similarity which can then be used in tasks such as classification or clustering. This method has the advantage of simplicity and it works surprisingly well in a wide range of applications from spam detection to movie recommendation [TP10]. Its main limitation is that a sentence is represented not as a sequence but as a *bag of words*, the word vectors will be the same whether the corpus contained "dog bites man" or "man bites dog". A standard way to fix this is to compute vectors not for words in isolation but for n-grams, windows of n consecutive words for some fixed size n. However the fix has its own limits: if n is too small we cannot detect any long-range correlations, if it is too big then the matrix is so sparse that we cannot detect anything at all.

In contrast, the recurrent neural networks (RNNs) of Rumelhart, Hinton and Williams [RHW86] are inherently sequential and their internal state can encode arbitrarily long-range correlations. At each step in the process, the network processes the next word in a sequence and updates its internal state. This internal memory can then be used to predict the rest of the sequence, or fed as input to another network e.g. for translation into another language. Once the obstacles to training were overcome (such as the vanishing gradients mentioned above), RNN architectures such as long short-term memory (LSTM) [HS97] set records in a variety of NLP tasks such as language modeling [SMH11], speech recognition [GMH13] and machine translation [SVL14]. The purely sequential approach of RNNs turned out to be limited: when the network is done reading, the information from the first word has to propagate through the entire text before it can be translated. Bidirectional RNNs [SP97] fix this issue by reading both left-to-right and right-to-left, with successful implementations such as BERT [Dev+19]. Bidirectionality is somewhat unsatisfactory from a cognitive perspective (humans manage to understand text without reading backward) but also harder to use in online settings where words need to be processed one at a time.

Attention mechanisms provide a much more elegant solution: instead of assuming that the "company" of a word is its immediate left and right neighbourhood, we let the neural network itself learn which words are relevant to which. First introduced as a way to boost the performance of RNNs on translation tasks [BCB16], attention has then become the basis of the *transformer model* [Vas+17]: a stack of attention mechanisms which process sequences without recurrence altogether. Transformers have now replaced RNNs as the state-of-the-art NLP model, culminating with the

GPT-3 language generator authoring its own article in *The Guardian* [GPT20]: "A robot wrote this entire article. Are you scared yet, human?"

Indeed *why* should we be scared? Because we are ignorant of *how* the robot wrote the article and we cannot explain what in its billions of parameters made it write the way it did. Transformers and neural networks in general are *black boxes*: we can probe the way they map inputs to outputs, but if we look at the terabytes of weights in between, we find no interpreation of the mapping. Moreover without explanability there can be no fairness: if we cannot explain how its decisions are made, we can hardly prevent the network from reproducing the discriminations present both in the datasets and in the assumptions of the data scientist who uses them. We argue that explainable AI requires to make the distributional black boxes transparent by endowing them with a compositional structure, we need *compositional distributional* models.

- neurosymbolic AI
- an alternative approach toward essentially the same goal: DisCoCat

What this thesis will *not* talk about:

- causality
- contextuality

References

[Aar15] Scott Aaronson. “Read the Fine Print”. In: *Nature Physics* 11.4 (Apr. 2015), pp. 291–293.

[AB08] Dorit Aharonov and Michael Ben-Or. “Fault-Tolerant Quantum Computation with Constant Error Rate”. In: *SIAM Journal on Computing* 38.4 (Jan. 2008), pp. 1207–1282.

[Amb04] A. Ambainis. “Quantum Search Algorithms”. In: *ACM SIGACT News* 35.2 (June 2004), pp. 22–35.

[Aru+15] Srinivasan Arunachalam, Vlad Gheorghiu, Tomas Jochym-O’Connor, Michele Mosca, and Priyaa Varshinee Srinivasan. “On the Robustness of Bucket Brigade Quantum RAM”. In: *New Journal of Physics* 17.12 (Dec. 2015), p. 123010.

[Aru+19] Frank Arute et al. “Quantum Supremacy Using a Programmable Superconducting Processor”. In: *Nature* 574.7779 (Oct. 2019), pp. 505–510.

[BCB16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *arXiv:1409.0473 [cs, stat]* (May 2016). arXiv: `1409.0473 [cs, stat]`.

[Bal21] Philip Ball. *Major Quantum Computing Strategy Suffers Serious Setbacks.* https://www.quantamagazine.org/major-quantum-computing-strategy-suffers-serious-setbacks-20210929/. Sept. 2021.

[Ban+08] Jeongho Bang, James Lim, M. S. Kim, and Jinhyoung Lee. “Quantum Learning Machine”. In: *arXiv:0803.2976 [quant-ph]* (Mar. 2008). arXiv: `0803.2976 [quant-ph]`.

[BLS19] Marcello Benedetti, Erika Lloyd, and Stefan Sack. “Parameterized Quantum Circuits as Machine Learning Models”. In: *arXiv:1906.07682 [quant-ph]* (June 2019). arXiv: `1906.07682 [quant-ph]`.

[Ben80] Paul Benioff. “The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines”. In: *Journal of Statistical Physics* 22.5 (May 1980), pp. 563–591.

[Ben+97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. “Strengths and Weaknesses of Quantum Computing”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1510–1523.

[BBD09] Daniel J Bernstein, Johannes Buchmann, and Erik Dahmén. *Post-Quantum Cryptography.* Berlin: Springer, 2009.

[Boo54] George Boole. *An Investigation of the Laws of Thought on Which Are Founded the Mathematical Theories of Logic and Probabilities*. London : Walton and Maberly, 1854.

[Bra+02] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. "Quantum Amplitude Amplification and Estimation". In: *arXiv:quant-ph/0005055* 305 (2002), pp. 53–74. arXiv: `quant-ph/0005055`.

[Car47] Rudolf Carnap. *Meaning and Necessity: A Study in Semantics and Modal Logic*. University of Chicago Press, 1947.

[Cho56] Noam Chomsky. "Three Models for the Description of Language". In: *IRE Transactions on Information Theory* 2.3 (Sept. 1956), pp. 113–124.

[Cho57] Noam Chomsky. *Syntactic Structures*. The Hague: Mouton and Co., 1957.

[CGK98] Isaac L. Chuang, Neil Gershenfeld, and Mark Kubinec. "Experimental Implementation of Fast Quantum Searching". In: *Physical Review Letters* 80.15 (Apr. 1998), pp. 3408–3411.

[CJS13] B. D. Clader, B. C. Jacobs, and C. R. Sprouse. "Preconditioned Quantum Linear System Algorithm". In: *Physical Review Letters* 110.25 (June 2013), p. 250504.

[Deu85] David Deutsch. "Quantum Theory, the Church–Turing Principle and the Universal Quantum Computer". In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.

[DJ92] David Deutsch and Richard Jozsa. "Rapid Solution of Problems by Quantum Computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (Dec. 1992), pp. 553–558.

[Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv:1810.04805 [cs]* (May 2019). arXiv: `1810.04805 [cs]`.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. "A Quantum Approximate Optimization Algorithm". In: *arXiv:1411.4028 [quant-ph]* (Nov. 2014). arXiv: `1411.4028 [quant-ph]`.

[FN18] Edward Farhi and Hartmut Neven. "Classification with Quantum Neural Networks on Near Term Processors". In: *arXiv:1802.06002 [quant-ph]* (Aug. 2018). arXiv: `1802.06002 [quant-ph]`.

[Fey82] Richard P. Feynman. "Simulating Physics with Computers". In: *International Journal of Theoretical Physics* 21.6 (June 1982), pp. 467–488.

[Fey85] Richard P Feynman. "Quantum Mechanical Computers". In: *Optics news* 11.2 (1985), pp. 11–20.

[Fir57] John R Firth. "A Synopsis of Linguistic Theory, 1930-1955". In: *Studies in linguistic analysis* (1957).

[Fre+03] Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. "Topological Quantum Computation". In: *Bulletin of the American Mathematical Society* 40.1 (2003), pp. 31–38.

[Fre84] Gottlob Frege. “The Foundations of Arithmetic”. In: *JL Austin. New York: Philosophical Library* (1884).

[GLM08] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. “Quantum Random Access Memory”. In: *Physical Review Letters* 100.16 (Apr. 2008), p. 160501.

[GPT20] GPT-3. “A Robot Wrote This Entire Article. Are You Scared yet, Human?” In: *The Guardian* (Sept. 2020).

[Gra+19] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. “An Initialization Strategy for Addressing Barren Plateaus in Parametrized Quantum Circuits”. In: *Quantum* 3 (Dec. 2019), p. 214.

[GMH13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. “Speech Recognition with Deep Recurrent Neural Networks”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Ieee. 2013, pp. 6645–6649.

[Gro97] Lov K. Grover. “Quantum Mechanics Helps in Searching for a Needle in a Haystack”. In: *Physical Review Letters* 79.2 (July 1997), pp. 325–328. arXiv: `quant-ph/9706033`.

[Har54] Zellig S. Harris. “Distributional Structure”. In: *WORD* 10.2-3 (Aug. 1954), pp. 146–162.

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. “Quantum Algorithm for Linear Systems of Equations”. In: *Physical Review Letters* 103.15 (Oct. 2009), p. 150502.

[Hau89] John Haugeland. *Artificial Intelligence: The Very Idea*. MIT press, 1989.

[Hav+19] Vojtech Havlicek et al. “Supervised Learning with Quantum Enhanced Feature Spaces”. In: *Nature* 567.7747 (Mar. 2019), pp. 209–212. arXiv: `1804.11326`.

[Hoc98] Sepp Hochreiter. “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions”. In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 06.02 (Apr. 1998), pp. 107–116.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural computation* 9 (Dec. 1997), pp. 1735–80.

[Hut04] W John Hutchins. “The Georgetown-Ibm Experiment Demonstrated in January 1954”. In: *Conference of the Association for Machine Translation in the Americas*. Springer. 2004, pp. 102–114.

[KP16] Iordanis Kerenidis and Anupam Prakash. “Quantum Recommendation Systems”. In: *arXiv:1603.08675 [quant-ph]* (Mar. 2016). arXiv: `1603.08675 [quant-ph]`.

[Kit95] A. Yu Kitaev. “Quantum Measurements and the Abelian Stabilizer Problem”. In: *arXiv:quant-ph/9511026* (Nov. 1995). arXiv: `quant-ph/9511026`.

[Kit03] A. Yu. Kitaev. “Fault-Tolerant Quantum Computation by Anyons”. In: *Annals of Physics* 303.1 (Jan. 2003), pp. 2–30.

[LF11] Adam Lally and Paul Fodor. “Natural Language Processing with Prolog in the IBM Watson System”. In: *The Association for Logic Programming (ALP) Newsletter* 9 (2011).

[LMR13] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum Algorithms for Supervised and Unsupervised Machine Learning”. In: *arXiv:1307.0411 [quant-ph]* (Nov. 2013). arXiv: 1307.0411 [quant-ph].

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum Principal Component Analysis”. In: *Nature Physics* 10.9 (Sept. 2014), pp. 631–633. arXiv: 1307.0401.

[Man80] Yuri Manin. “Computable and Uncomputable”. In: *Sovetskoye Radio, Moscow* 128 (1980).

[Mar47] A Markov. “On Certain Insoluble Problems Concerning Matrices”. In: *Doklady Akad. Nauk SSSR*. Vol. 57. 6. 1947, pp. 539–542.

[McC+16] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. “The Theory of Variational Hybrid Quantum-Classical Algorithms”. In: *New Journal of Physics* 18.2 (Feb. 2016), p. 023023.

[McC+18] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. “Barren Plateaus in Quantum Neural Network Training Landscapes”. In: *Nature Communications* 9.1 (Dec. 2018), p. 4812. arXiv: 1803.11173.

[Ped+19] Edwin Pednault, John A. Gunnels, Giacomo Nannicini, Lior Horesh, and Robert Wisnieff. “Leveraging Secondary Storage to Simulate Deep 54-Qubit Sycamore Circuits”. In: *arXiv:1910.09534 [quant-ph]* (Oct. 2019). arXiv: 1910.09534 [quant-ph].

[Per+14] Alberto Peruzzo et al. “A Variational Eigenvalue Solver on a Photonic Quantum Processor”. In: *Nature Communications* 5.1 (July 2014), p. 4213.

[Pes+21] Arthur Pesah et al. “Absence of Barren Plateaus in Quantum Convolutional Neural Networks”. In: *Physical Review X* 11.4 (Oct. 2021), p. 041011.

[Pos47] Emil L. Post. “Recursive Unsolvability of a Problem of Thue”. In: *Journal of Symbolic Logic* 12.1 (Mar. 1947), pp. 1–11.

[Pre18] John Preskill. “Quantum Computing in the NISQ Era and Beyond”. In: *Quantum* 2 (Aug. 2018), p. 79.

[RML14] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. “Quantum Support Vector Machine for Big Data Classification”. In: *Physical Review Letters* 113.13 (Sept. 2014), p. 130503.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning Representations by Back-Propagating Errors”. In: *Nature* 323.6088 (Oct. 1986), pp. 533–536.

[SWY75] G. Salton, A. Wong, and C. S. Yang. “A Vector Space Model for Automatic Indexing”. In: *Commun. ACM* 18.11 (1975), pp. 613–620.

[Sch21] Maria Schuld. “Quantum Machine Learning Models Are Kernel Methods”. In: *arXiv:2101.11020 [quant-ph, stat]* (Jan. 2021). arXiv: 2101.11020 [quant-ph, stat].

[SK19] Maria Schuld and Nathan Killoran. "Quantum Machine Learning in Feature Hilbert Spaces". In: *Physical Review Letters* 122.4 (Feb. 2019), p. 040504. arXiv: 1803.07128.

[SP97] M. Schuster and K.K. Paliwal. "Bidirectional Recurrent Neural Networks". In: *IEEE Transactions on Signal Processing* 45.11 (Nov. 1997), pp. 2673–2681.

[SB09] Dan Shepherd and Michael J. Bremner. "Temporally Unstructured Quantum Computation". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 465.2105 (May 2009), pp. 1413–1439.

[Sho96] Peter W Shor. "Fault-Tolerant Quantum Computation". In: *Proceedings of 37th Conference on Foundations of Computer Science*. IEEE. 1996, pp. 56–65.

[Sho94] P.W. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. Santa Fe, NM, USA: IEEE Comput. Soc. Press, 1994, pp. 124–134.

[Sim94] D. Simon. "On the Power of Quantum Computation". In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 1994, pp. 116–123.

[SMH11] Ilya Sutskever, James Martens, and Geoffrey E Hinton. "Generating Text with Recurrent Neural Networks". In: *ICML*. 2011.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to Sequence Learning with Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 27. Curran Associates, Inc., 2014.

[Thu14] Axel Thue. *Probleme Über Veränderungen von Zeichenreihen Nach Gegebenen Regeln*. na, 1914.

[Tur50] A. M. Turing. "Computing Machinery and Intelligence". In: *Mind* LIX.236 (Oct. 1950), pp. 433–460.

[TP10] P. D. Turney and P. Pantel. "From Frequency to Meaning: Vector Space Models of Semantics". In: *Journal of Artificial Intelligence Research* 37 (Feb. 2010), pp. 141–188.

[Vas+17] Ashish Vaswani et al. "Attention Is All You Need". In: *arXiv:1706.03762 [cs]* (Dec. 2017). arXiv: 1706.03762 [cs].

[Wea55] Warren Weaver. "Translation". In: *Machine translation of languages* 14.15-23 (1955), p. 10.

[WBL12] Nathan Wiebe, Daniel Braun, and Seth Lloyd. "Quantum Algorithm for Data Fitting". In: *Physical Review Letters* 109.5 (Aug. 2012), p. 050505.

[Wit53] Ludwig Wittgenstein. *Philosophical Investigations*. Oxford: Basil Blackwell, 1953.

[Wu+21] Yulin Wu et al. "Strong Quantum Computational Advantage Using a Superconducting Quantum Processor". In: *Physical Review Letters* 127.18 (Oct. 2021), p. 180501.

[Yon+21] Yong et al. “Closing the "Quantum Supremacy" Gap: Achieving Real-Time Simulation of a Random Quantum Circuit Using a New Sunway Supercomputer”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Nov. 2021), pp. 1–12. arXiv: 2110.14502.

[Zho+20] Han-Sen Zhong et al. “Quantum Computational Advantage Using Photons”. In: *Science* 370.6523 (Dec. 2020), pp. 1460–1463. arXiv: 2012.01625.

[Zhu+21] Qingling Zhu et al. “Quantum Computational Advantage via 60-Qubit 24-Cycle Random Circuit Sampling”. In: *Science Bulletin* (Oct. 2021).