The Neural Qubit: Biologically Inspired Quantum Deep Learning

Siraj Raval
School of AI Research
hello@sirajraval.com

Abstract -- Deep learning has resulted in state of the art performance for automated tasks in the fields of natural language processing, computer vision, autonomous driving, and many other subfields of Artificial Intelligence. However, if the goal is to create a system that is capable of learning any task, given an objective function, I hypothesis that it's necessary to reconsider classical neural network architecture to incorporate certain properties of quantum mechanics, namely superposition and entanglement. Building on the work of Fisher [12], I surmise that Phosphorus-31 enables both of these properties to occur within neurons in the human brain. In light of this evidence, quantum information processing in the context of digital neural networks is an area that deserves further exploration. As such, I present a novel quantum neural network architecture, similar to the continuous variable archicteture by Killoran et al. [11] that can accurately predict credit card fraud for a web-based business. My aim is that this will provide a starting point for more research in this space, ultimately using this technology to drive more innovation in every Scientific discipline, from Pharmacology to Computer Science.

Introduction

A Biological Neural Networks

Our brain's neural networks are what inspired the advent of digital neural networks more than 50 years ago, but I use the term inspired loosely here. Human neurons both send and receive data in analog frequencies consisting of combinations and even overlays of tones, chirps, and strobes. The symbology of these transmissions are unique for each person, like a fingerprint. These frequencies can ride each other over a synapse and dendrite, so that one message might reach a target, while another is shunted to another area.

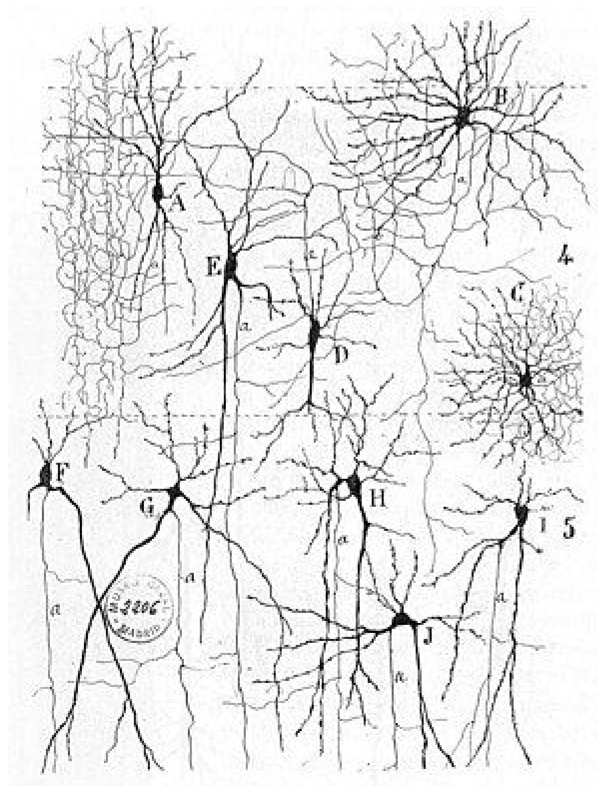

Figure 1 - Different neuronal cell types in the brain

These behaviors constantly change as the city of activity inside the neuron are all making their own decisions about how to process these signals.

B Digital Neural Networks

Digital neural networks however, have no corollary for this kind of signal processing complexity. One of the simplest types of neural networks, the perceptron[4], receives two floating point numbers and performs a multiplication operations between the two then adds a bias value, and if the result passed through an activation function trips the minimum value, it sends it along either a boolean or a numerical output

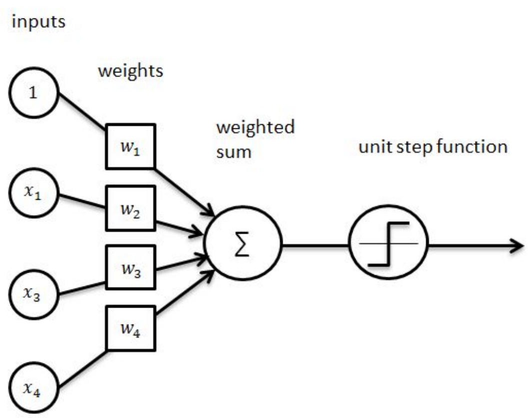

Figure 1.1 - The classical perceptron network

Despite this huge difference between the neuron in biology and the neuron in silicon, neural networks are still capable of performing incredibly challenging tasks like image captioning, essay writing, and vehicle driving. Despite this, they are still limited in their capability. For example, they can't yet perform causal reasoning, meaning making connections between disparate concepts, Recent progress in Transformer networks trained on text data have shown it's possible to generate a compelling essay, but it still has not been shown that the network understands what the concepts it's writing about are or how they're related. It's repeatedly making likely next word predictions using a probability model. Self driving cars use neural networks for computer vision, but are still vulnerable to adversarial attacks which can sometimes make them misclassify an image if a single pixel is replaced. In order to improve these efficiencies, it's widely acknowledged that we'll need better algorithms and more computational power. I hypothesize that quantum hardware and software can help us do both. They can allow us to solve problems of enormous time complexity in a feasible time span. if we can simulate our universe on a machine, chemical, physical, and biological interactions, we can build a simulated lab in the cloud that scales, ushering in a new era of Scientific research for anyone to make discoveries using just their computer. More specifically, if we incorporate quantum computing into machine learning to get higher accuracy scores, that'll enable innovation in the private sector to create more efficient services for every industry, from agriculture to finance.

The Limits of Classical Computation

Although the limits of the classical computing theory theory have been well known since its beginnings, computers can do so much nowadays that it is easy to forget that they have been invented for what they could not do. Alan Turing defined the notion of a universal digital computer (a Turing machine) in his 1937 paper, "On computable numbers, with an application to the decision problem," and his goal was to show that there were tasks that even the most powerful computer machine could not perform (for example, the stop problem). According to the now-famous Church-Turing thesis, these issues are merely beyond traditional computing. Kurt Gödel acquired a comparable outcome around the same time. In 1931, through an ingenious device known as Gödel numbering, Gödel discovered a way to assign natural numbers to arithmetic statements themselves in a distinctive manner,effectively turning numbers into talking about numbers. This allowed him to demonstrate a theorem of incompleteness that basically says there are real statements of mathematics (theorems) that we can never officially understand are true. It is interesting that while both Turing and Gödel have shown that the entire body of human knowledge can not be obtained by formal computation alone, considering the intrinsic limitations of the method, they seem to give distinct reasons for how the human mind could accomplish the feat. Comparing a Turing machine to a human mind is unfair, according to Turing–the former runs algorithmically and never makes mistakes, and the latter does "trial-and-error" and always makes wild guess. "If a machine is to be unfailing, it can't be smart as well." And only if it makes no pretense of infallibility can a device become smart and human-like. On the other hand, Gödel, on the consistency of human knowledge, did not want to give up. He suggested that "it remains possible for a theorem-proving machine to exist (and even be empirically discovered) that is in fact equivalent to mathematical intuition, but can not be proven to be so, nor can it be demonstrated to produce only right theorems of the theory of finite numbers."

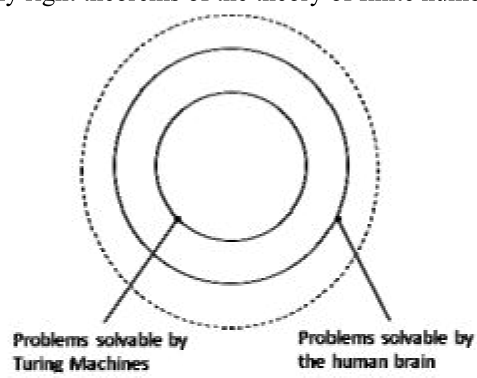

Figure 2 - An example problem complexity space

At the end of his now famous 1960 article entitled "The unreasonable efficacy of mathematics in the natural sciences," Eugene Wigner wondered "if one day we could establish a[mathematical] theory of consciousness or biology phenomena that would be as coherent and convincing as our present theories of the inanimate world." Apparently, half a century later, the hope of Wigner was not fulfilled. The list of human "cognitive biases" gets longer and longer as we discover more of them. In all these cases, of course, irrationality is described as human deviation from a Turing-like computational representation of the issue to be solved, and decades of psychological studies seem to demonstrate without doubt that such deviation is systematic and abundant. The discrepancy shown in Figure 2 suggests an explanation: we may have used mathematics that is not strong enough to capture all of human cognition's exciting complexity. Efforts have been made to extend the classical computing theory–its history is nearly as long as the classical computing theory itself. Various computation models, often called hypercomputation or super-Turing computation, have been suggested that can calculate functions that are not effectively computable in the Church-Turing sense of the thesis. The BlumShub-Smale (BSS) machine [14], also called a real computer, is a remarkable hyper-computational model.

A classical computer, as we understand, relies on discrete symbols (e.g., 0s and 1s) for encoding data and presupposes that all the fundamental sets are countable (one-to -one correspondence to natural numbers N). Real numbers (R, a continuum) can be handled by a real computer and can therefore answer uncountable subset questions (e.g., "is the Mandelbrot set decisive?"). While it has been shown that real computation can be applied straight to numerical analysis and scientific computing issues, it is not evident whether it substantially decreases the discrepancy shown in Figure 2. I claim that the expansion from N to R, however important it may seem, remains insufficient to deal with some of the cognitive science's toughest issues (X-problems).

"X-problems" of Human Cognition

As I will later attempt to define a technique to more accurately simulate human cognition programmatically, it's helpful to distinguish two classes of cognitive science issues[12]. One class can be called Z-problems (for puZZle), referring to those empirical results that are puzzling but in classical computational terms somewhat explainable. Examples of the Z-problem include the difference between short-term memory and long-term memory, the idea of working memory ability, skill development through the formation and tuning of when-then manufacturing laws, and attention through bottom-up and top-down control.

Another class of issues can be called X-problems (for paradoXes), referring to those empirical results that are so magical and mysterious that they seem to defy classical descriptions of mathematics. Examples of this category include consciousness and awareness, intuition, sensation, visual perception gestalt phenomena, and, to name but a few, multiple so-called "cognitive biases" in human judgement and decision making. It seems insufficient to discredit them as ephemeral and indignant, or merely to label them as "human biases and heuristics," or to suggest ad-hoc patched explanations. These phenomena are human brain and body functions resulting from millions of years of evolution and adaptation. It is reasonable to say that cognitive science has so far been brief in offering solutions to X-problems with more strict and more systematic responses. In order to offer solutions to Z-problems, I surmise that it's necessary to explore the possibilities that quantum mechanics play in giving rise to human cognition.

Quantum Mechanics in the Brain

Tegmark's theory on the importance of quantum decoherence in brain processes was insightful towards this end[14]. In it, he demonstrated how Quantum mechanics processes only continue to show their effects when the relevant information doesn't leak into the environment, also called decoherence. His argument was that decoherence timescales take place over a time of the order 10^-13 to 10^-20 seconds. But the timescales over which neurons fire is different, it's .001 to .1 seconds, meaning it must be a classical computer. However, Fisher more recently wrote that that there's a loophole in that argument [12]. Yes, quantum computation would indeed require isolation from the thermal bath of the human brain, but that's what nuclear spins can help with. Because it is a charged particle in motion i.e spinning, an atomic nucleus creates a magnetic field. The "north pole" of this spin-generated magnetic field points a certain direction in space. The nuclear spin state is the orientation of the spin-generated magnetic field relative to an external magnetic field. He proposed that there must be a common biological element with a very isolated nuclear spin inside of a neuron that can serve as a kind of neural qubit. He calculated that it would have to be an atom with a nuclear spin of 1/2 to be able to maintain quantum coherence.

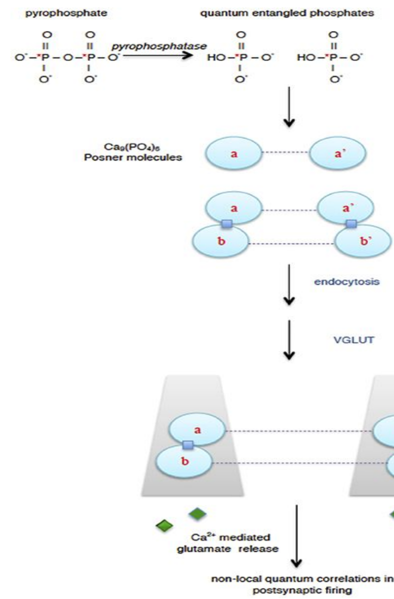

Figure 3 - Quantum Coherence in the P-31 isotope

He found that there is only one that does so, Phosphorous, specifically it's only stable isotope, Phosphorus-31. P-31 is likely the only possible neural qubit, which when entangled with others like can maintain quantum coherence for up to 106 seconds[12]. There's much more research to be done here, but given this evidence, we must examine how computation in the brain would work in a quantum framework.

The Quantum Coin Toss

In classical theory, tossing a coin that could land on head (H) or tail (T) is modeled by a bit.. There are two points (on R), 0 (for H) and 1 (for T) in the entire phase space. A state with binary decisions, H and T, each with a 1⁄2 likelihood, can be represented by tossing a fair coin. Formally,

$$S = ½ H + ½ T$$

In quantum theory, a coin tossing can be depicted by a Qubit (a quantum bit) living in a complicated 2-dimensional Hilbert space. Every point on the sphere of the Bloch is a possible Qubit state. A Qubit is comparable to a classic bit in that it turns out to be 0 or 1. However, approximately speaking, a Qbit includes much more data (quantum) than a classic bit–each point (there are endless numbers) on the Bloch sphere's equator unit circle reflects a feasible state of fair coin throwing. Thus, the following two states (among infinitely others) can equally well represent throwing a fair coin.

$$|\psi 1> \;= \frac{1}{\sqrt{2}} \times |H> + \frac{1}{\sqrt{2}} \times \;|T>,$$
$$|\psi 2> \;= \frac{1}{\sqrt{2}}\, i \times |H> + \frac{1}{\sqrt{2}}\, i \times \;|T>$$

These representations are called wavefunctions, and these coefficients are complex numbers whose squared modulus represents a corresponding likelihood.

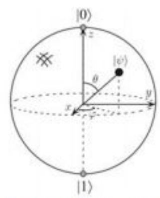

Figure 4 - Bloch Sphere representation of a Qubit

You may wonder how these extra degrees of freedom are being used. They appear to depict phase information, which is intended to live in a different dimension. Traditionally, the resulting states are explained roughly by the phrase "the coin is simultaneously in a superposition of heads and tails." For non-quantum physicists, this kind of description creates a lot of confusion. It seems that our natural language lacks the required expressive authority to even describe, let alone differentiate, these closely confused states. The issue is, can our brain do that? What about our mind (conscious or unconscious)? And what, if any, is the phase's feasible psychological significance? Any fair-minded psychologist would probably reject this nonsense quickly: "Why do I need an infinite number of possible states to represent a simple half-and-a-half-coin tossing that I don't even know how to tell them apart?"

This is a reasonable question, of course, and here are some counterarguments. First of all, the brain has so many neurons, and we don't understand how each operates and what it really represents, depending on our present understanding of neuroscience. Certainly, the data of the stage may be represented by neurons or neuron groups somehow. Second, unlike experimental quantum physicists who design advanced and costly devices that enable them to perform multiple accurate and fine-grained measurements of quantum particles in their laboratories, psychologists are far less able to assess neuronal or mental states. Tools in their arsenal, including introspection, verbal reporting, reaction time, or even neuroimaging (fMRI or EEG), often result in rough, gross, imprecise, and indirect action. Thus, this type of framing of cognition can be beneficial for those looking to explore quantum deep learning as a viable alternative to classical deep learning. If the brain does use quantum mechanics for cognition, then replicating that ability could improve the accuracy of different types of neural network architectures, we'll explore that in the next section.

Quantum Deep Learning

A Classical Deep Learning

The basic building for deep learning is the neural feedforward network (also known as the multilayer perceptron)[2]. a multilayer framework where each layer is a linear transformation preceded by a nonlinear' activation' feature. Mathematically, a single layer performs the transformation for an input vector

$$\mathcal{L}(\mathbf{x}) = \varphi(W\mathbf{x} + \mathbf{b}), \qquad (1)$$

Where *W* is a matrix, *b* is a vector, and φ is a nonlinear function. The objects *W* and *b*, respectively referred to as the weight matrix and the bias vector, have their own parameters that are learned. The activation function ϕ typically does not contain any free parameters and acts on its inputs element-wise.

The' deep' in deep learning emerges from stacking together various layers of this sort, so that one layer's output is used as an input for the next. Overall, each layer will have its own autonomous parameters of weight and bias. A *N*-layer neural network model is used to summarize all model parameters by the specified parameter.

$$\mathbf{y} = f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathcal{L}_N \circ \cdots \circ \mathcal{L}_1(\mathbf{x}), \qquad (2)$$

Building machine learning models with multilayer neural networks is well motivated due to different theorems of universality. These theorems ensure that feedforward neural networks can approximate to an arbitrary degree of precision any continuous function on a closed and bounded subset of real numbers given sufficiently free parameters. While the original theorems showed that two layers were sufficient for the approximation of universal function, deeper networks with the same number of parameters can be more powerful and efficient than shallower networks.

The theorems of universality[9] demonstrate the neural network model's authority to approximate features, but those theorems do not say anything about how this approximation can be found. The feature to be equipped is typically not clearly recognized, but its input-output relationship is to be inferred from the information. How can we tailor the parameters of the network to fit the information? The workhorse is the stochastic gradient descent algorithm for this task, which suits information with a neural network model by estimating derivatives of the parameters of the model–the weights and biases–and by using gradient descent to minimize some appropriate objective feature. Neural networks trained through stochastic gradient descent, combined with a sufficiently big dataset, have shown notable efficiency for a multitude of assignments across many application fields.

B Continuous Variable Quantum Computing

The classical bit's quantum analog is the qubit.
A multi-qubit system's quantum states are standardized vectors in a complex Hilbert space. Over the years, several attempts have been produced to encode neural networks and neural-network-like structures into qubit systems, with different degrees of success. Two strategies can be distinguished. There are approaches that encode inputs into a multi qubit state's amplitude vector and interpret unitary transformations as layers of neural networks. These models involve indirect methods to introduce the vital nonlinearity of the activation function[10], often leading to the algorithm's undeniable likelihood of failure.
Other methods encode each input bit into a distinct qubit but have an overhead that stems from the need to binarize the ongoing values. In addition, the typical neural network design of multiplication of matrices and nonlinear activations becomes cumbersome to translate into a quantum algorithm, and the benefits of doing so are not always evident. Because of these limitations, qubit architectures may not be the most flexible quantum frameworks for encoding neural networks with constant real-evaluated inputs and outputs.

Fortunately, qubits are not the only accessible medium for processing quantum information. An alternative architecture of quantum computing, the continuous variable (CV) model[5], fits much better with the ongoing image of neural networks underlying computation. CV formalism has a lengthy history and can be realized physically using optical devices, microwave, and ion traps. In the CV model, data is transmitted in quantum states of bosonic modes, often called qumodes, which form a quantum circuit's ‘wires’.

Continuous-variable quantum data can be encoded using the representation of the wavefunction and the formulation of quantum mechanics in phase space. In the former, I specify a single continuous variable, say x, and represent the qumode state by means of a complex-evaluated function of this variable called the wavefunction $\psi(x)$. In concrete terms, we can interpret x as a coordinate of position, and $\psi(x)$^2 as the density of probability of a particle at x. I can also use a wavefunction based on a conjugate momentum variable, π(p), from basic quantum theory. Instead of position and momentum, x and p can be pictured equally as the real and imaginary parts, such as light, of a quantum field.

I treat the conjugate variables x and p on an equal footing in the phase space picture, giving a connection to the classical Hamiltonian mechanics. The state of a single qumode is thus encoded with two real-value variables (x, p). Qumode states are depicted in the phase space called *quasiprobability distributions* as real-valued features $F(\boldsymbol{x}, \boldsymbol{p})$. Quasi relates to the reality that with classical probability distributions, these features share some, but not all, characteristics. In particular, the features of quasiprobability can be negative.

While standardization forces qubit systems to have a unitary geometry, standardization gives the CV image a much looser constraint, namely that the ***F(x, p)*** function has an integral unit over phase space. Qumode states also have a representation as vectors or density matrices in infinite Hilbert space. These basic states depict the particle-like nature of qumode structures[5], with the amount of electrons being denoted by *n*. This is similar to how it is possible to expand square-integrable features using a countable basis set like sines or cosines. Equivalent projections are given by the phase space and Hilbert space formulations. Thus, from both a wave-like and particle-like viewpoint, CV quantum systems can be studied. We're going to focus primarily on the former.

In the CV model, there's a key difference between Gaussian quantum doors and non-Gaussian ones. The Gaussian gates are the "easy" operations for a quantum computer with a CV in many ways. The easiest Gaussian single-mode doors are *rotation*, *displacement*, and *squeezing*. The fundamental Gaussian two-mode door is the (phaseless) beamsplitter (which can be understood as a two-qumod rotation). More explicitly, the following phase space transformations are produced by these Gaussian doors:

$$R(\phi): \quad \begin{bmatrix} x \\ p \end{bmatrix} \mapsto \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix}, \tag{3}$$

$$D(\alpha): \quad \begin{bmatrix} x \\ p \end{bmatrix} \mapsto \begin{bmatrix} x + \mathrm{Re}(\alpha) \\ p + \mathrm{Im}(\alpha) \end{bmatrix}, \tag{4}$$

$$S(r): \quad \begin{bmatrix} x \\ p \end{bmatrix} \mapsto \begin{bmatrix} e^{-r} & 0 \\ 0 & e^{r} \end{bmatrix} \begin{bmatrix} x \\ p \end{bmatrix}, \tag{5}$$

$$BS(\theta): \quad \begin{bmatrix} x_1 \\ x_2 \\ p_1 \\ p_2 \end{bmatrix} \mapsto \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & \cos\theta & -\sin\theta \\ 0 & 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ p_1 \\ p_2 \end{bmatrix}. \tag{6}$$

Notice that most of these Gaussian operations have names suggestive of a linear character. Indeed, there is a natural correspondence between Gaussian operations and affine transformations on phase space. For a system of N modes, the most general Gaussian transformation has the effect

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} \mapsto M \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_i \end{bmatrix}, \tag{7}$$

This native affine structure will be the key for building quantum neural networks.

$$\Omega = \begin{bmatrix} 0 & \mathbb{1} \\ -\mathbb{1} & 0 \end{bmatrix} \tag{8}$$

is the *2N* × *2N* symplectic form. A generic symplectic matrix *M* can be split into a type of singular-value decomposition – known as the Euler or Bloch-Messiah decomposition – of the form

$$M = K_2 \begin{bmatrix} \Sigma & 0 \\ 0 & \Sigma^{-1} \end{bmatrix} K_1, \tag{9}$$

A matrix *K* with these two properties must have the form

$$K = \begin{bmatrix} C & D \\ -D & C \end{bmatrix}, \tag{10}$$

With

$$CD^T - DC^T = 0 \tag{11}$$

$$CC^T + DD^T = \mathbb{1}. \tag{12}$$

I will also need later the fact that if C is an arbitrary orthogonal matrix, then C ⊕ C is both orthogonal and symplectic. Importantly, the intersection of the symplectic and orthogonal groups on 2N dimensions is isomorphic to the unitary group on N dimensions. This isomorphism allows us to perform the transformations via the unitary action of passive linear optical interferometers. Every Gaussian transformation on N modes (Eq. (7)) can be decomposed into a CV circuit containing only the basic gates mentioned above. Looking back to Eqs. (3)- (6), I can recognize that interferometers made up of *R* and *BS* gates are sufficient to generate the orthogonal transformations, while S gates are sufficient to give the scaling transformation *Σ* ⊕ *Σ −1* . Finally, displacement gates complete the full affine transformation.

Classical	CV quantum computing
feedforward neural network	CV variational circuit
weight matrix W	symplectic matrix M
bias vector $\boldsymbol{b}$	displacement vector $\boldsymbol{\alpha}$
affine transformations	Gaussian gates
nonlinear function	non-Gaussian gate
weight/bias parameters	gate parameters
variable x	operator $\hat{x}$
derivative $\frac{\partial}{\partial x}$	conjugate operator $\hat{p}$
no classical analogue	superposition
no classical analogue	entanglement

TABLE I. Conceptual correspondences between classical neural networks and CV quantum computing. Some concepts from the quantum side have no classical analogue.

D Network Architecture

In this section, I present a scheme for quantum neural networks using this CV framework. It is inspired from two sides. The main idea is the following: the fully connected neural network architecture provides a powerful and intuitive approach for designing variational circuits in the CV model. I will introduce the most general form of the quantum neural network, which is the analogue of a classical fully connected network. In Table I, I give a high-level matching between neural network concepts and their CV analogues.

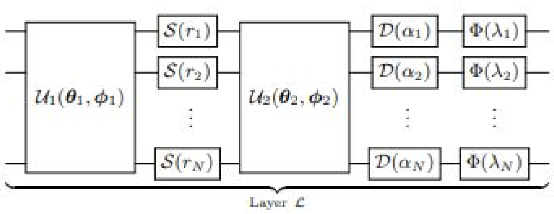

FIG. 1. The circuit structure for a single layer of a CV quantum neural network: an interferometer, local squeeze gates, a second interferometer, local displacements, and finally local non-Gaussian gates. The first four components carry out an affine transformation, followed by a final nonlinear transformation.

A general CV quantum neural network is built up as a sequence of layers, with each layer containing every gate from the universal gate set. Specifically, a layer L consists of the successive gate sequence shown in Fig. 1:

$$\mathcal{L} := \Phi \circ \mathcal{D} \circ \mathcal{U}_2 \circ \mathcal{S} \circ \mathcal{U}_1, \tag{16}$$

The collective gate variables *(θ, φ, r, α,λ)* form the free parameters of the network, where λ can be optionally kept fixed.
The sequence of Gaussian transformations shown above is sufficient to parameterize every possible unitary affine transformation on N qumodes. This sequence thus has the role of a 'fully connected' matrix transformation. Interestingly, adding a nonlinearity uses the same component that adds universality: a non-Gaussian gate Φ. Using *z* = ***(x, p)***, I can write the combined transformation in a form reminiscent of Eq. (1), namely

$$\mathcal{L}(\mathbf{z}) = \Phi(M\mathbf{z} + \boldsymbol{\alpha}). \tag{17}$$

Thanks to the CV encoding, I get a nonlinear functional transformation while still keeping the quantum circuit unitary. Similar to the classical setup, I can stack multiple layers of this type end-to-end to form a deeper network. The quantum state output from one layer is used as the input for the next. Different layers can be made to have different widths by adding or removing qumodes between layers. Removal can be accomplished by measuring or tracing out the extra qumodes. This architecture can also accept classical inputs. I can do this by fixing some of the gate arguments to be set by classical data rather than free parameters. This scheme can be thought of as an embedding of classical data into a quantum feature space [10]. The output of the network can be obtained by performing measurements and/or computing expectation values. The choice of measurement operators is flexible; different choices (homodyne, heterodyne, photon-counting, etc.) may be better suited for different situations.

Experiment & Results

I tested the continuous variable quantum model by selecting a threshold probability necessary to classify transactions as real or fake from a Kaggle dataset using the "Strawberry Fields" quantum simulation framework in Python. A receiver operating characteristic (ROC) curve can be built by varying the classification threshold, where each point in the curve is parameterized by a threshold value. An optimal classifier has a true negative rate of 1 and a false negative rate of 0. My classifier has a region below the 0.953 ROC curve compared to the ideal 1. It is imperative to minimize the false negative rate to detect fraudulent credit card transactions. i.e. the rate of misclassification of a fraudulent transaction as genuine. In contrast, minimizing the false favorable rate is less essential–these are the instances where real transactions are classified as fraudulent. Typically, such instances can be resolved by sending cardholders verification emails.

The findings here demonstrate a classical-quantum neural hybrid proof-of-principle network capable of performing classification for a real practical interest issue. While it is simple to build a classical neural network to outperform this hybrid model, owing to the need to simulate the quantum network on a classical computer, this network is limited in width and depth. Exploring the efficiency of hybrid networks in combination with a physical quantum computer are the next steps in the research pipeline.

Conclusion

In this paper, I posit that the intersection of quantum information processing and machine learning is an exciting, unexplored area of research that deserves further investigation. The cognition behind Biological neural networks are still largely not understood for “X-Problems”, and I support the need for a new type of mathematics that address non-commutativity to help solve them. Based on the findings of Fisher[11], I support the theory that the brain uses quantum mechanical properties like superposition and entanglement for cognition, something that Turing computers are not capable of doing with sufficient time complexity. To demonstrate a proof of concept architecture in this vein, I Provided a quantum neural network architecture that leverages and explores the continuous-variable formalism of quantum computing through both theoretical exposure and numerical tests. This system can be regarded as an analog of latest neural network proposals encoded using classical light[31], with the extra component that we leverage the electromagnetic field's quantum characteristics. I expect that specific neural networks will also be solely inspired from the quantum side in future job. I evaluated the efficiency of quantum neural network models numerically and showed promise in the activities I considered, namely fraud detection. Finally, exploring the role that basic ideas of quantum physics–such as symmetry, interference, enmeshment, and the principle of uncertainty–perform within the context of quantum neural networks would be an interesting research direction.

References

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

[2] Peter Wittek. Quantum machine learning: what quantum computing means to data mining. Academic Press, 2014.

[3] Maria Schuld and Francesco Petruccione. Quantum computing for supervised learning. Upcoming monograph.

[4] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. Physical Review Letters, 109(5):050505, 2012.

[5] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. Nature Physics, 10(9):631, 2014.

[6] Guang Hao Low, Theodore J Yoder, and Isaac L Chuang. Quantum inference on bayesian networks. Physical Review A, 89(6):062315, 2014.

[7] Nathan Wiebe and Christopher Granade. Can small quantum systems learn? arXiv:1512.03145, 2015.

[8] Ashley Montanaro. Quantum speedup of monte carlo methods. Proc. R. Soc. A, 471(2181):20150301, 2015.

[9] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. Physical Review Letters, 113(13):130503, 2014.

[10] Maria Schuld and Nathan Killoran. Quantum machine learning in feature Hilbert spaces. arXiv:1803.07128, 2018.

[11] Nathan Killoran,Thomas R. Bromley, Juan Miguel Arrazola, Maria Schuld,Nicolas Quesada, and Seth Lloyd. Continuous-variable quantum neural networks arXiv: 1806.06871, 2018

[12] Mathew P. A. Fisher. Quantum Cognition: The possibility of processing with nuclear spins in the brain. arXiv: 1508.05929 , 2015

[13] Hongbin Wang, Jack W. Smith, Yanlong Sun. Simulating Cognition with Quantum Computers. ArXiv: 1905.12599, 2019

[14] Max Tegmark. The importance of quantum decoherence in brain processes. arXiv:9907009 ,1999