QUANTUM DEFORMED NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

We develop a new quantum neural network layer designed to run efficiently on a quantum computer but that can be simulated on a classical computer when restricted in the way it entangles input states. We first ask how a classical neural network architecture, both fully connected or convolutional, can be executed on a quantum computer using quantum phase estimation. We then deform the classical layer into a quantum design which entangles activations and weights into quantum superpositions. While the full model would need the exponential speedups delivered by a quantum computer, a restricted class of designs represent interesting new classical network layers that still use quantum features. We show that these quantum deformed neural networks can be trained and executed on normal data such as images, and even classically deliver modest improvements over standard architectures.

1 INTRODUCTION

Quantum mechanics (QM) is the most accurate description for physical phenomena at very small scales, such as the behavior of molecules, atoms and subatomic particles. While QM does not seem to describe any phenomena at the scale that we are familiar with, it does have a huge impact on our every day lives through technologies such as lasers, transistors (and thus microchips) and MRI.

A recent view of QM has formulated it as a (Bayesian) statistical methodology that only describes our subjective view of the (quantum) world, and how we update that view in light of evidence (i.e. measurements) (Hooft, 2016; Fuchs & Schack, 2013). This is in perfect analogy to the classical Bayesian view, a statistical paradigm extensively used in artificial intelligence where we maintain probabilities to represent our beliefs for events in the world.

The philosophy of this paper will be to turn this argument on its head. If we can view QM as just another consistent statistical theory that happens to describe nature at small scales, then we can also use this theory to describe *classical* signals by endowing them with a Hilbert space structure. In some sense, the 'only' difference with Bayesian statistics is that the positive probabilities are replaced with complex 'amplitudes'. This however has the dramatic effect that, unlike in classical statistics, interference between events now becomes a possibility. In this paper we show that this point of view uncovers new architectures and potential speedups for running neural networks on quantum computers.

We shall restrict our attention here to binary neural networks. We will introduce a new class of quantum neural networks and interpret them as generalizations of probabilistic binary neural networks, identifying potential speedups by running the models on a quantum computer. Then we will devise classically efficient algorithms to train the networks for a restrict set of quantum circuits. We present results of classical simulations of the quantum neural networks on real world data sizes and related gains in accuracy due to the quantum deformations. Contrary to almost all other works on quantum deep learning, our quantum neural networks can be simulated for practical classical problems, such as images or sound. The quantum nature of our models is there to increase the flexibility of the model-class and add new operators to the toolbox of the deep learning researcher, some of which may only reach their full potential when quantum computing becomes ubiquitous.

1.1 RELATED WORK

In Farhi & Neven (2018) variational quantum circuits that can be learnt via stochastic gradient descent were introduced. Their performance could be studied only on small input tasks such as classifying 4×4 images, due to the exponential time of classical simulations of those circuits. Other works on variational quantum circuits for neural networks are Verdon et al. (2018); Beer et al. (2019). Their focus is similarly on the implementation on near term quantum devices and these models cannot be efficiently run on a classical computer. Exceptions are models which use tensor network simulations (Cong et al., 2019; Huggins et al., 2019) where the model can be scaled to 8×8 image data with 2 classes, at the price of constraining the geometry of the quantum circuit (Huggins et al., 2019). The quantum deformed neural networks introduced in this paper are instead a class of variational quantum circuits that can be scaled to the size of data that are used in traditional neural networks as we demonstrate in section 4.2.

Another line of work directly uses tensor networks as full precision machine learning models that can be scaled to the size of real data (Miles Stoudenmire & Schwab, 2016; Liu et al., 2017; Levine et al., 2017; 2019). However the constraints on the network geometry to allow for efficient contractions limit the expressivity and performance of the models. Further these works do not directly relate to implementations on quantum computers.

A large body of work in quantum machine learning focuses on using quantum computing to provide speedups to classical machine learning tasks (Biamonte et al., 2017; Ciliberto et al., 2018; Wiebe et al., 2014), culminating in the discovery of quantum inspired speedups in classical algorithms (Tang, 2019). In particular, (Allcock et al., 2018; Cao et al., 2017; Schuld et al., 2015; Kerenidis et al., 2019) discuss quantum simulations of classical neural networks with the goal of improving the efficiency of classical models on a quantum computer. Our models differ from these works in two ways: i) we use quantum wave-functions to model weight uncertainty, in a way that is reminiscent of Bayesian models; ii) we design our network layers in a way that may only reach its full potential on a quantum computer due to exponential speedups, but at the same time can, for a restricted class of layer designs, be simulated on a classical computer and provide inspiration for new neural architectures. Finally, quantum methods for accelerating Bayesian inference have been discussed in Zhao et al. (2019b;a) but only for Gaussian processes while in this work we shall discuss relations to Bayesian neural networks.

2 GENERALIZED PROBABILISTIC BINARY NEURAL NETWORKS

Binary neural networks are neural networks where both weights and activations are binary. Let $\mathbb{B} = \{0, 1\}$. A fully connected binary neural network layer maps the N_ℓ activations $\boldsymbol{h}^{(\ell)}$ at level ℓ to the $N_{\ell+1}$ activations $\boldsymbol{h}^{(\ell+1)}$ at level $\ell + 1$ using weights $\boldsymbol{W}^{(\ell)} \in \mathbb{B}^{N_\ell N_{\ell+1}}$:

$$h_j^{(\ell+1)} = f(\boldsymbol{W}^{(\ell)}, \boldsymbol{h}^{(\ell)}) = \tau\left(\frac{1}{N_\ell + 1}\sum_{i=1}^{N_\ell} W_{j,i}^{(\ell)} h_i^{(\ell)}\right), \quad \tau(x) = \begin{cases} 0 & x < \frac{1}{2} \\ 1 & x \geq \frac{1}{2} \end{cases}. \tag{1}$$

We divide by $N_\ell + 1$ since the sum can take the $N_\ell + 1$ values $\{0, \ldots, N_\ell\}$. We do not explicitly consider biases which can be introduced by fixing some activations to 1. In a classification model $\boldsymbol{h}^{(0)} = \boldsymbol{x}$ is the input and the last activation function is typically replaced by a softmax which produces output probabilities $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W})$, where $\boldsymbol{W}$ denotes the collection of weights of the network.

Given M input/output pairs $\boldsymbol{X} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^M), \boldsymbol{Y} = (\boldsymbol{y}^1, \ldots, \boldsymbol{y}^M)$, a frequentist approach would determine the binary weights so that the likelihood $p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}) = \prod_{i=1}^M p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{W})$ is maximized. Here we consider discrete or quantized weights and take the approach of variational optimization staines2012variational, which introduces a weight distribution $q_\theta(\boldsymbol{W})$ to devise a surrogate differential objective. For an objective $O(\boldsymbol{W})$, one has the bound $\max_{\boldsymbol{W} \in \mathbb{B}^N} O(\boldsymbol{W}) \geq \mathbb{E}_{q_\theta(\boldsymbol{W})}[O(\boldsymbol{W})]$, and the parameters of $q_\theta(\boldsymbol{W})$ are adjusted to maximize the lower bound. In our case we consider the objective:

$$\max_{\boldsymbol{W} \in \mathbb{B}^N} \log p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W}) \geq \log \mathbb{E}_{q_\theta(\boldsymbol{W})}[p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W})] = \sum_{i=1}^M \log q(\boldsymbol{y}_i|\boldsymbol{x}_i), \tag{2}$$

where $q(\boldsymbol{y}|\boldsymbol{x}) = \mathbb{E}_{q_\theta(\boldsymbol{W})}[p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{W})]$. To uncouple the data terms in the last equality we can formally introduce different $\boldsymbol{W}_i$ for each data-case and integrate over $\prod_i q(\boldsymbol{W}_i)$ where the distributions q are all the same. While the optimal solution to equation 2 is a Dirac measure, one can add a regularization term $\mathcal{R}(\theta)$ to keep q soft. While we focus in this work on $\mathcal{L} = \log \mathbb{E}_{q_\theta(\boldsymbol{W})}[p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W})]$, we remark that we could also have considered $\mathcal{L} = \mathbb{E}_{q_\theta(\boldsymbol{W})}[\log p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{W})]$ and apply this framework to Bayesian deep learning – see appendix A for background on Bayesian deep learning and the relation to the variational objective.

In both variational Bayes and variational optimization frameworks for binary networks, we have a variational distribution $q(\boldsymbol{W})$ and probabilistic layers where activations are random variables. We consider an approximate posterior factorized over the layers: $q(\boldsymbol{W}) = \prod_{\ell=1}^{L} q^{(\ell)}(\boldsymbol{W}^{(\ell)})$. If $\boldsymbol{h}^{(\ell)} \sim p^{(\ell)}$, equation 1 leads to the following recursive definition of distributions:

$$p^{(\ell+1)}(\boldsymbol{h}^{(\ell+1)}) = \sum_{\boldsymbol{h}\in\mathbb{B}^{N_\ell}} \sum_{\boldsymbol{W}\in\mathbb{B}^{N_\ell N_{\ell+1}}} \delta(\boldsymbol{h}^{(\ell+1)} - f(\boldsymbol{W}^{(\ell)}, \boldsymbol{h}^{(\ell)}))p^{(\ell)}(\boldsymbol{h}^{(\ell)})q^{(\ell)}(\boldsymbol{W}^{(\ell)})\,. \tag{3}$$

We can identify the prediction $q(\boldsymbol{y}|\boldsymbol{x})$ with the distribution after L levels: $q(\boldsymbol{y}|\boldsymbol{x}) = p^{(L)}(\boldsymbol{y})$.

In previous works (Shayer et al., 2017; Peters & Welling, 2018), the approximate posterior was taken to be factorized: $q(\boldsymbol{W}^{(\ell)}) = \prod_{ij} q_{i,j}(W_{i,j}^{(\ell)})$, which results in a factorized activation distribution as well: $p^{(\ell)}(\boldsymbol{h}^{(\ell)}) = \prod_i p_i^{(\ell)}(h_i^{(\ell)})$. (Shayer et al., 2017; Peters & Welling, 2018) considered the Bayesian framework and sampling of activations at each layer using the local reparametrization trick.

The quantum neural network we introduce below will naturally give a way to sample efficiently from complex distributions and in view of that we here generalize the setting: we act with a stochastic matrix $\boldsymbol{S}_\phi(\boldsymbol{h}', \boldsymbol{W}'|\boldsymbol{h}, \boldsymbol{W})$ which depends on parameters ϕ and correlates the weights and the input activations to a layer as follows:

$$\pi_{\phi,\theta}(\boldsymbol{h}', \boldsymbol{W}') = \sum_{\boldsymbol{h}\in\mathbb{B}^N} \sum_{\boldsymbol{W}\in\mathbb{B}^{NM}} \boldsymbol{S}_\phi(\boldsymbol{h}', \boldsymbol{W}'|\boldsymbol{h}, \boldsymbol{W})p(\boldsymbol{h})q_\theta(\boldsymbol{W})\,. \tag{4}$$

To avoid redundancy, we still take $q_\theta(\boldsymbol{W})$ to be factorized and let $\boldsymbol{S}$ create correlation among the weights as well. The choice of $\boldsymbol{S}$ will be related to the choice of a unitary matrix in the quantum circuit of the quantum neural network. A layer is now made of the two operations, $\boldsymbol{S}_\phi$ and the layer map f, resulting in the following output distribution:

$$p^{(\ell+1)}(\boldsymbol{h}^{(\ell+1)}) = \sum_{\boldsymbol{h}\in\mathbb{B}_\ell^N} \sum_{\boldsymbol{W}\in\mathbb{B}^{N_\ell N_{\ell+1}}} \delta(\boldsymbol{h}^{(\ell+1)} - f(\boldsymbol{W}^{(\ell)}, \boldsymbol{h}^{(\ell)}))\pi_{\phi,\theta}^{(\ell)}(\boldsymbol{h}^{(\ell)}, \boldsymbol{W}^{(\ell)})\,. \tag{5}$$

which allows one to compute the network output $q(\boldsymbol{y}|\boldsymbol{x})$ recursively. Both the parameters ϕ and θ will be learned to solve the following optimization problem:

$$\min_{\theta,\phi} \mathcal{R}(\theta) + \mathcal{R}'(\phi) - \mathcal{L}\,. \tag{6}$$

where $\mathcal{R}(\theta), \mathcal{R}'(\phi)$ are regularization terms for the parameters θ, ϕ. We call this model a generalized probabilistic binary neural network, with ϕ deformation parameters chosen such that $\phi = 0$ gives back the standard probabilistic binary neural network. In a variational Bayesian setup we would replace the first and last term with the ELBO.

To study this model on a classical computer we need to choose S which leads to an efficient sampling algorithm for $\pi_{\phi,\theta}$. In general, one could use Markov Chain Monte Carlo, but there exists situations for which the mixing time of the chain grows exponentially in the size of the problem (Levin & Peres, 2017). In the next section we will show how quantum mechanics can enlarge the set of probabilistic binary neural networks that can be efficiently executed and in the subsequent sections we will show experimental results for a restricted class of correlated distributions inspired by quantum circuits that can be simulated classically.

3 Quantum implementation

Quantum computers can sample from certain correlated distributions more efficiently than classical computers (Aaronson & Chen, 2016; Arute et al., 2019). In this section, we devise a quantum circuit

that implements the generalized probabilistic binary neural networks introduced above, encoding $\pi_{\theta,\phi}$ in a quantum circuit. This could lead to a potential exponential speedup for this model, opening up the study of more complex probabilistic neural networks with quantum computers.

A quantum implementation of a binary perceptron was introduced in Schuld et al. (2015) as an application of the quantum phase estimation algorithm (Nielsen & Chuang, 2000). However, no quantum advantage of the quantum simulation was shown. Here we will extend that result in several ways: i) we will modify the algorithm to represent the generalized probabilistic layer introduced above, showing the quantum advantage present in our setting; ii) we will consider the case of multi layer percetrons as well as convolutional networks.

3.1 Introduction to quantum notation and quantum phase estimation

As a preliminary step, we introduce notations for quantum mechanics. We refer the reader to Appendix B for a more thorough review of quantum mechanics. A qubit is a normalized vector $|\psi\rangle \in \mathbb{C}^2$. N qubits form the vector space $(\mathbb{C}^2)^{\otimes N} \cong \mathbb{C}^{2^N}$ spanned by all N-bit strings, $|b_1, \dots, b_N\rangle \equiv |b_1\rangle \otimes \cdots \otimes |b_N\rangle$, $b_i \in \mathbb{B}$. Quantum circuits are unitary matrices on this space. The probability of a measurement with outcome ϕ_i is given by matrix element of the projector $|\phi_i\rangle\langle\phi_i|$ in a state $|\psi\rangle$, namely $p_i = \langle\psi|\phi_i\rangle\langle\phi_i|\psi\rangle = |\langle\phi_i|\psi\rangle|^2$, a formula known as Born's rule.

Next, we describe the quantum phase estimation (QPE), a quantum algorithm to estimate the eigenphases of a unitary $\boldsymbol{U}$. Denote the eigenvalues and eigenvectors of $\boldsymbol{U}$ by $\exp\left(\frac{2\pi i}{2^t}\varphi_\alpha\right)$ and $|v_\alpha\rangle$, and assume that the φ_α's can be represented with a finite number t of bits: $\varphi_\alpha = 2^{t-1}\varphi_\alpha^1 + \cdots + 2^0\varphi_\alpha^t$. (This is the case of relevance for a binary network.) Then introduce t ancilla qubits in state $|0\rangle^{\otimes t}$. Given an input state $|\psi\rangle$, QPE is the following unitary operation:

$$|0\rangle^{\otimes t} \otimes |\psi\rangle \xrightarrow{\text{QPE}} \sum_\alpha \langle v_\alpha|\psi\rangle\, |\varphi_\alpha\rangle \otimes |v_\alpha\rangle \ . \tag{7}$$

Appendix B.1 reviews the details of the quantum circuit implementing this map, whose complexity is linear in t. Now using the notation τ for the threshold non-linearity introduced in equation 1, and recalling the expansion $2^{-t}\varphi = 2^{-1}\varphi^1 + \cdots + 2^{-t}\varphi^t$, we note that if the first bit $\varphi^1 = 0$ then $2^{-t}\varphi < \frac{1}{2}$ and $\tau(2^{-t}\varphi) = 0$, while if $\varphi^1 = 1$, then $2^{-t}\varphi \geq \frac{1}{2}$ and $\tau(2^{-t}\varphi) = 1$. In other words, $\delta_{\varphi^1,b} = \delta_{\tau(2^{-t}\varphi),b}$ and the probability $p(b)$ that after the QPE the first ancilla bit is b is given by:

$$\Big(\sum_\alpha \overline{\langle v_\alpha|\psi\rangle}\, \langle\varphi_\alpha| \otimes \langle v_\alpha|\Big)\Big[\,|b\rangle\langle b| \otimes \mathbf{1}\Big]\Big(\sum_\beta \langle v_\beta|\psi\rangle\, |\varphi_\beta\rangle \otimes |v_\beta\rangle\Big) = \sum_\alpha |\langle v_\alpha|\psi\rangle|^2 \delta_{\tau(2^{-t}\varphi_\alpha),b} \ , \tag{8}$$

where $\Big[\,|b\rangle\langle b| \otimes \mathbf{1}\Big]$ is an operator that projects the first bit to the state $|b\rangle$ and leaves the other bits untouched.

3.2 Definition and advantages of quantum deformed neural networks

Armed with this background, we can now apply quantum phase estimation to compute the output of the probabilistic layer of equation 5. Let N be the number of input neurons and M that of output neurons. We introduce qubits to represent inputs and weights bits:

$$|\boldsymbol{h}, \boldsymbol{W}\rangle \in \mathcal{V}_h \otimes \mathcal{V}_W \ , \quad \mathcal{V}_h = \bigotimes_{i=1}^N (\mathbb{C}^2)_i \ , \quad \mathcal{V}_W = \bigotimes_{i=1}^N \bigotimes_{j=1}^M (\mathbb{C}^2)_{ij} \ . \tag{9}$$

Then we introduce a Hamiltonian $\boldsymbol{H}_j$ acting non trivially only on the N input activations and the N weights at the j-th row:

$$\boldsymbol{H}_j = \sum_{i=1}^N \boldsymbol{B}_{ji}^W \boldsymbol{B}_i^h \ , \tag{10}$$

and $\boldsymbol{B}_i^h$ ($\boldsymbol{B}_{ji}^W$) is the matrix $\boldsymbol{B} = |1\rangle\langle 1|$ acting on the i-th activation (ji-th weight) qubit. Note that $\boldsymbol{H}_j$ singles out terms from the state $|\boldsymbol{h}, \boldsymbol{W}\rangle$ where both $h_j = 1$ and $W_{ij} = 1$ and then adds them

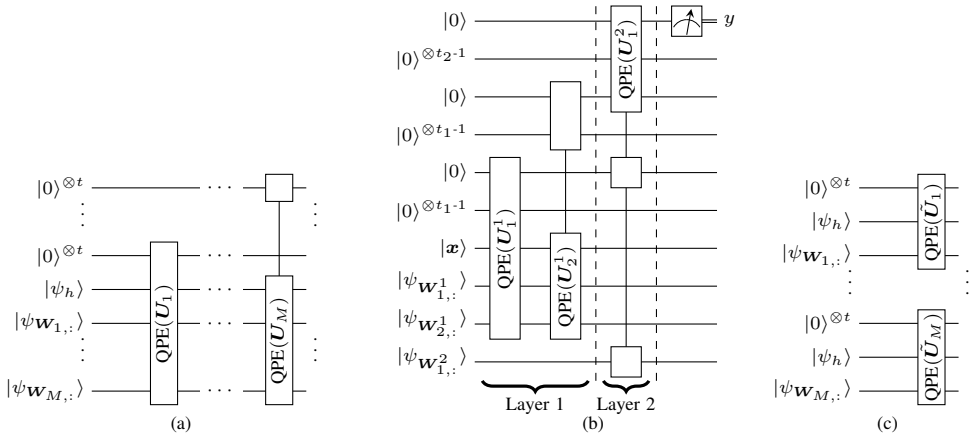

Figure 1: (a) Quantum circuit implementing a quantum deformed layer. The thin vertical line indicates that the gate acts as identity on the wires crossed by the line. (b) Quantum deformed multilayer perceptron with 2 hidden quantum neurons and 1 output quantum neuron. $|\boldsymbol{x}\rangle$ is an encoding of the input signal, y is the prediction. The superscript ℓ in $\boldsymbol{U}_j^\ell$ and $\boldsymbol{W}_{j,:}^\ell$ refers to layer ℓ. We split the blocks of t_ℓ ancilla qubits into a readout qubit that encodes the layer output amplitude and the rest. (c) Modification of a layer for classical simulations.

up, i.e. the eigenvalues of $\boldsymbol{H}_j$ are the preactivations of equation 1:

$$\boldsymbol{H}_j \left|\boldsymbol{h}, \boldsymbol{W}\right\rangle = \varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:}) \left|\boldsymbol{h}, \boldsymbol{W}\right\rangle \ , \quad \varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:}) = \sum_{i=1}^{N} W_{ji} h_i \ . \tag{11}$$

Now define the unitary operators:

$$\boldsymbol{U}_j = \boldsymbol{D} \mathrm{e}^{\frac{2\pi i}{N+1} \boldsymbol{H}_j} \boldsymbol{D}^{-1} \ , \tag{12}$$

where $\boldsymbol{D}$ is another generic unitary, and as we shall see shortly, its eigenvectors will be related to the entries of the classical stochastic matrix $\boldsymbol{S}$ in section 2. Since $\boldsymbol{U}_j \boldsymbol{U}_{j'} = \boldsymbol{D} \mathrm{e}^{\frac{2\pi i}{N+1}(\boldsymbol{H}_j + \boldsymbol{H}_{j'})} \boldsymbol{D}^{-1} = \boldsymbol{U}_{j'} \boldsymbol{U}_j$, we can diagonalize all the $\boldsymbol{U}_j$'s simultaneously and since they are conjugate to $\mathrm{e}^{\frac{2\pi i}{N+1} \boldsymbol{H}_j}$ they will have the same eigenvalues. Introducing the eigenbasis $\left|\boldsymbol{h}, \boldsymbol{W}\right\rangle_{\boldsymbol{D}} = \boldsymbol{D} \left|\boldsymbol{h}, \boldsymbol{W}\right\rangle$, we have:

$$\boldsymbol{U}_j \left|\boldsymbol{h}, \boldsymbol{W}\right\rangle_{\boldsymbol{D}} = \mathrm{e}^{\frac{2\pi i}{N+1} \varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:})} \left|\boldsymbol{h}, \boldsymbol{W}\right\rangle_{\boldsymbol{D}} \ . \tag{13}$$

Note that $\varphi \in \{0, \ldots, N\}$ so we can represent it with exactly t bits, $N = 2^t - 1$. Then we add M ancilla resources, each of t qubits, and sequentially perform M quantum phase estimations, one for each $\boldsymbol{U}_j$, as depicted in figure 1 (a). We choose the following input state

$$\left|\psi\right\rangle = \left|\psi\right\rangle_h \otimes \bigotimes_{j=1}^{M} \left|\psi\right\rangle_{\boldsymbol{W}_{j,:}} \ , \quad \left|\psi\right\rangle_{\boldsymbol{W}_{j,:}} = \bigotimes_{i=1}^{N} \left[\sqrt{q_{ji}(W_{ji} = 0)} \left|0\right\rangle + \sqrt{q_{ji}(W_{ji} = 1)} \left|1\right\rangle \right] \ , \tag{14}$$

where we have chosen the weight input state according to the factorized variational distribution q_{ij} introduced in section 2. In fact, this state corresponds to the following probability distribution via Born's rule:

$$p(\boldsymbol{h}, \boldsymbol{W}) = |\left\langle \boldsymbol{h}, \boldsymbol{W} | \psi \right\rangle|^2 = p(\boldsymbol{h}) \prod_{j=1}^{M} \prod_{i=1}^{N} q_{ji}(W_{ji}) \ , \quad p(\boldsymbol{h}) = |\left\langle \boldsymbol{h} | \psi \right\rangle_h|^2 \ . \tag{15}$$

The state $\left|\psi_h\right\rangle$ is discussed below. Now we show that a non-trivial choice of $\boldsymbol{D}$ leads to an effective correlated distribution. The j-th QPE in figure 1 (a) corresponds to equation 7 where we identify $\left|v_\alpha\right\rangle \equiv \left|\boldsymbol{h}, \boldsymbol{W}\right\rangle_{\boldsymbol{D}}$, $\left|\varphi_\alpha\right\rangle \equiv \left|\varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:})\right\rangle$ and we make use of the j-th block of t ancillas. After M steps we compute the outcome probability of a measurement of the first qubit in each of the M registers of the ancillas. We can extend equation 8 to the situation of measuring multiple qubits, and recalling that the first bit of an integer is the most significant bit, determining whether $2^{-t}\varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:}) = (N+1)^{-1}\varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:})$ is greater or smaller than $1/2$, the probability of outcome $\boldsymbol{h}' = (h'_1, \ldots, h'_M)$ is

$$p(\boldsymbol{h}') = \sum_{\boldsymbol{h} \in \mathbb{B}^N} \sum_{\boldsymbol{W} \in \mathbb{B}^{NM}} \delta_{\boldsymbol{h}', f(\boldsymbol{W}, \boldsymbol{h})} \left|\left\langle \psi | \boldsymbol{h}, \boldsymbol{W} \right\rangle_{\boldsymbol{D}}\right|^2 \ , \tag{16}$$

where f is the layer function introduced in equation 1. We refer to appendix C for a detailed derivation. Equation 16 is the generalized probabilistic binary layer introduced in equation 5 where $\boldsymbol{D}$ corresponds to a non-trivial $\boldsymbol{S}$ and a correlated distribution when $\boldsymbol{D}$ entangles the qubits:

$$\pi(\boldsymbol{h}, \boldsymbol{W}) = |\langle\psi| \boldsymbol{D} |\boldsymbol{h}, \boldsymbol{W}\rangle|^2 \ . \tag{17}$$

The variational parameters ϕ of $\boldsymbol{S}$ are now parameters of the quantum circuit $\boldsymbol{D}$. Sampling from π can be done by doing repeated measurements of the first M ancilla qubits of this quantum circuit. On quantum hardware $\mathrm{e}^{\frac{2\pi i}{N+1}\boldsymbol{H}_j}$ can be efficiently implemented since it is a product of diagonal two-qubits quantum gates. We shall consider unitaries $\boldsymbol{D}$ which have efficient quantum circuit approximations. Then computing the quantum deformed layer output on a quantum computer is going to take time $O(tMu(N))$ where $u(N)$ is the time it takes to compute the action of $\boldsymbol{U}_j$ on an input state. There exists $\boldsymbol{D}$ such that sampling from equation 17 is exponentially harder classically than quantum mechanically, a statement forming the basis for quantum supremacy experiments (Aaronson & Chen, 2016; Arute et al., 2019). Examples are random circuits with two-dimensional entanglement patterns, which from a machine learning point of view can be natural when considering image data. Other examples are $\boldsymbol{D}$ implementing time evolution operators of physical systems, whose simulation is exponentially hard classically, resulting in hardness of sampling from the time evolved wave function.

Even better, on quantum hardware we can avoid sampling intermediate activations altogether. At the first layer, the input can be prepared by encoding the input bits in the state $|\boldsymbol{x}\rangle$. For the next layers, we simply use the output state as the input to the next layer. One obtains thus the quantum network of figure 1 (b) and the algorithm for a layer is summarized in procedure 1 in appendix F. Note that all the qubits associated to the intermediate activations are entangled. Therefore the input state $|\psi_h\rangle$ would have to be replaced by a state in $\mathcal{V}_h$ plus all the other qubits, where the gates at the next layer would act only on the subspace $\mathcal{V}_h$ in the manner described in this section. (An equivalent and more economical mathematical description is use the reduced density matrix ρ_h as input state.) We envision two other possible procedures for what happens after the first layer: i) we sample from equation 16 and initialize $|\psi_h\rangle$ to the bit string sampled in analogy to the classical quantization of activations; ii) we sample many times to reconstruct the classical distribution and encode it in $|\psi_h\rangle$. In our classical simulations below on factorized distributions we will be able to actually calculate the probabilities and can avoid sampling.

3.3 Modifications for classical simulations

In this paper we will provide classical simulations of the quantum neural networks introduced above for a restricted class of designs. We do this for two reasons: first to convince the reader that the quantum layers hold promise (even though we can not simulate the proposed architecture in it's full glory due to the lack of access to a quantum computer) and second, to show that these ideas can be interesting as new designs, even "classically" (by which we mean architectures that can be executed on a classical computer).

To scale up those simulations, we do the modifications to the setup just explained which are depicted in figure 1 (c). We clone the input activation register M times, an operation that quantum mechanically is only approximate (Nielsen & Chuang, 2000) but exact classically. Then we associate the j-th copy to the j-th row of the weight matrix, thus forming pairs for each $j = 1, \ldots, M$:

$$|\boldsymbol{h}, \boldsymbol{W}_{j,:}\rangle \in \mathcal{V}_h \otimes \mathcal{V}_{W,j} \ , \quad \mathcal{V}_{W,j} = \bigotimes_{i=1}^{N} (\mathbb{C}^2)_{ji} \tag{18}$$

Fixing j, we introduce the unitary $\mathrm{e}^{\frac{2\pi i}{N+1}\boldsymbol{H}_j}$ diagonal in the basis $|\boldsymbol{h}, \boldsymbol{W}_{j,:}\rangle$ as in equation 10 and define the new unitary:

$$\tilde{\boldsymbol{U}}_j = \boldsymbol{D}_j \mathrm{e}^{\frac{2\pi i}{N+1}\boldsymbol{H}_j} \boldsymbol{D}_j^{-1} \ , \tag{19}$$

where w.r.t. equation 12 we now let $\boldsymbol{D}_j$ depend on j. We denote the eigenvectors of $\tilde{\boldsymbol{U}}_j$ by $|\boldsymbol{h}, \boldsymbol{W}_{j,:}\rangle_{\boldsymbol{D}_j} = \boldsymbol{D}_j |\boldsymbol{h}, \boldsymbol{W}_{j,:}\rangle$ and the eigenvalue is $\varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:})$ introduced in equation 11. Supposing that we know $p(\boldsymbol{h}) = \prod_i p_i(h_i)$, we apply the quantum phase estimation to $\tilde{\boldsymbol{U}}_j$ with input:

$$|\psi_j\rangle = |\psi_h\rangle \otimes |\psi\rangle_{\boldsymbol{W}_{j,:}} \ , \quad |\psi_h\rangle = \bigotimes_{i=1}^{N} \left[\sqrt{p_i(h_i = 0)}\, |0\rangle + \sqrt{p_i(h_i = 1)}\, |1\rangle \right] \ , \tag{20}$$

and $|\psi\rangle_{\boldsymbol{W}_{j,:}}$ is defined in equation 14. Going through similar calculations as those done above shows that measurements of the first qubit will be governed by the probability distribution of equation 5 factorized over output channels since the procedure does not couple them: $\pi(\boldsymbol{h}, \boldsymbol{W}) = \prod_{j=1}^{M} |\langle\psi_j| \boldsymbol{D}_j |\boldsymbol{h}, \boldsymbol{W}_{j,:}\rangle|^2$. In this setting, a quantum device could give an exponential advantage for sampling this distribution even on noisy, intermediate scale quantum computers that are already available (Arute et al., 2019). So far, we have focused on fully connected layers. We can extend the derivation of this section to the convolution case, by applying the quantum phase estimation on images patches of the size equal to the kernel size as explained in appendix D.

4 Classical simulations for low entanglement

4.1 Theory

It has been remarked in (Shayer et al., 2017; Peters & Welling, 2018) that when the weight and activation distributions at a given layer are factorized, $p(\boldsymbol{h}) = \prod_i p_i(h_i)$ and $q(\boldsymbol{W}) = \prod_{ij} q_{ij}(W_{ij})$, the output distribution in equation 3 can be efficiently approximated using the central limit theorem (CLT). The argument goes as follows: for each j the preactivations $\varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:}) = \sum_{i=1}^{N} W_{j,i} h_i$ are sums of independent binary random variables $W_{j,i} h_i$ with mean and variance:

$$\mu_{ji} = \mathbb{E}_{w\sim q_{ji}}(w)\mathbb{E}_{h\sim p_i}(h)\,, \quad \sigma_{ji}^2 = \mathbb{E}_{w\sim q_{ji}}(w^2)\mathbb{E}_{h\sim p_i}(h^2) - \mu_{ji}^2 = \mu_{ji}(1-\mu_{ji})\,, \tag{21}$$

We used $b^2 = b$ for a variable $b \in \{0, 1\}$. The CLT implies that for large N we can approximate $\varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:})$ with a normal distribution with mean $\mu_j = \sum_i \mu_{ji}$ and variance $\sigma_j^2 = \sum_i \sigma_{ji}^2$. The distribution of the activation after the non-linearity of equation 1 can thus be computed as:

$$p(\tau(\tfrac{1}{N+1}\varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:})) = 1) = p(2\varphi(\boldsymbol{h}, \boldsymbol{W}_{j,:}) - N > 0) = \Phi\left(-\tfrac{2\mu_j - N}{2\sigma_j}\right)\,, \tag{22}$$

Φ being the CDF of the standard normal distribution. Below we fix j and omit it for notation clarity.

As reviewed in appendix B, commuting observables in quantum mechanics behave like classical random variables. The observable of interest for us, $\boldsymbol{D}\boldsymbol{H}\boldsymbol{D}^{-1}$ of equation 19, is a sum of commuting terms $\boldsymbol{K}_i \equiv \boldsymbol{D}\boldsymbol{B}_i^W \boldsymbol{B}_i^h \boldsymbol{D}^{-1}$ and if their joint probability distribution is such that these random variables are weakly correlated, i.e.

$$\langle\psi| \boldsymbol{K}_i \boldsymbol{K}_{i'} |\psi\rangle - \langle\psi| \boldsymbol{K}_i |\psi\rangle \langle\psi| \boldsymbol{K}_{i'} |\psi\rangle \to 0\,, \quad \text{if } |i - i'| \to \infty\,, \tag{23}$$

then the CLT for weakly correlated random variables applies, stating that measurements of $\boldsymbol{D}\boldsymbol{H}\boldsymbol{D}^{-1}$ in state $|\psi\rangle$ are governed by a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with

$$\mu = \langle\psi| \boldsymbol{D}\boldsymbol{H}\boldsymbol{D}^{-1} |\psi\rangle\,, \quad \sigma^2 = \langle\psi| \boldsymbol{D}\boldsymbol{H}^2\boldsymbol{D}^{-1} |\psi\rangle - \mu^2\,. \tag{24}$$

Finally, we can plug these values into equation 22 to get the layer output probability.

We have cast the problem of simulating the quantum neural network to the problem of computing the expectation values in equation 24. In physical terms, these are related to correlation functions of $\boldsymbol{H}$ and $\boldsymbol{H}^2$ after evolving a state $|\psi\rangle$ with the operator $\boldsymbol{D}$. These can be efficiently computed classically for one dimensional and lowly entangled quantum circuits $\boldsymbol{D}$ (Vidal, 2003). In view of that here we consider a 1d arrangement of activation and weight qubits, labeled by $i = 0, \ldots, 2N-1$, where the even qubits are associated with activations and the odd are associated with weights. We then choose:

$$\boldsymbol{D} = \prod_{i=0}^{N-1} \boldsymbol{Q}_{2i,2i+1} \prod_{i=0}^{N-1} \boldsymbol{P}_{2i+1,2i+2}\,, \tag{25}$$

where $\boldsymbol{Q}_{2i,2i+1}$ acts non-trivially on qubits $2i, 2i+1$, i.e. onto the i-th activation and i-th weight qubits, while $\boldsymbol{P}_{2i,2i+1}$ on the i-th weight and $i+1$-th activation qubits. We depict this quantum circuit in figure 2 (a). As explained in detail in appendix E, the computation of μ involves the matrix element of $\boldsymbol{K}_i$ in the product state $|\psi\rangle$ while σ^2 involves that of $\boldsymbol{K}_i\boldsymbol{K}_{i+1}$. Due to the structure of $\boldsymbol{D}$, these operators act locally on 4 and 6 sites respectively as depicted in figure 2 (b)-(c). This implies that the computation of equation 24, and so of the full layer, can be done in $O(N)$ and easily parallelized. Appendix E contains more details on the complexity, while appendix F contains an algorithm box for the classical simulation discussed here.

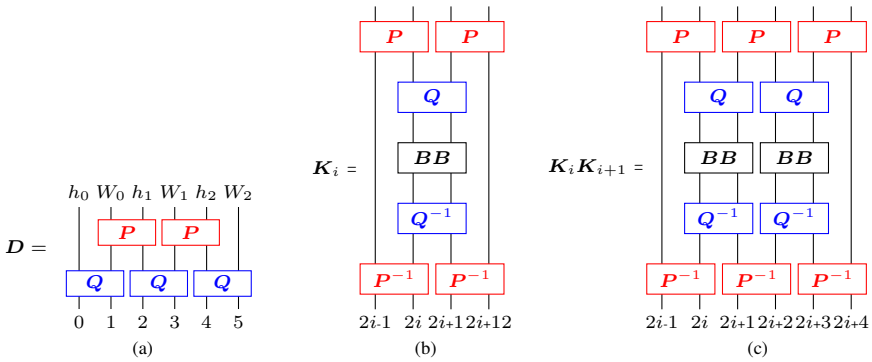

Figure 2: (a) The entangling circuit $\boldsymbol{D}$ for $N = 3$. (b) $\boldsymbol{K}_i$ entering the computation of μ. (c) $\boldsymbol{K}_i\boldsymbol{K}_{i+1}$ entering the computation of σ^2. Indices on $\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{B}$ are omitted for clarity. Time flows downwards.

4.2 Experiments

We present experiments for the model of the previous section. At each layer, q_{ij} and $\boldsymbol{D}_j$ are learnable. They are optimized to minimize the loss of equation 6 where following (Peters & Welling, 2018; Shayer et al., 2017) we take $\mathcal{R} = \beta \sum_{\ell,i,j} q_{ij}^{(\ell)}(1 - q_{ij}^{(\ell)})$, and $\mathcal{R}'$ is the L_2 regularization loss of the parameters of $\boldsymbol{D}_j$. $\mathcal{L}$ coincides with equation 2. We implemented and trained several architectures with different deformations. Table 4.2 contains results for two standard image datasets, MNIST and Fashion MNIST. Details of the experiments are in appendix G together with additional results.

The general deformation ([PQ]) performs best in all cases. In the simplest case of a single dense layer (A), the gain is $+3.2\%$ for MNIST and $+2.6\%$ for Fashion MNIST on test accuracy. For convnets, we could only simulate a single deformed layer due to computational issues and the gain is around or less than 1%. We expect that deforming all layers will give a greater boost as the improvements diminish with decreasing the ratio of deformation parameters over classical parameters (q_{ij}). The increase in accuracy comes at the expense of more parameters. This is in line with common practice in deep learning where overparametrized networks perform better.

Table 1: Test accuracies for MNIST and Fashion MNIST. With the notation $cKsS - C$ to indicate a conv2d layer with C filters of size $[K, K]$ and stride S, and dN for a dense layer with N output neurons, the architectures (Arch.) are A: d10; B: c3s2-8, c3s2-16, d10; C: c3s2-32, c3s2-64, d10. The deformations are: [/]: $\boldsymbol{P}^j_{i,i+1} = \boldsymbol{Q}^j_{i,i+1} = \mathbf{1}$ (baseline (Peters & Welling, 2018)); [PQ]: $\boldsymbol{P}^j_{i,i+1}, \boldsymbol{Q}^j_{i,i+1}$ generic; [Q]: $\boldsymbol{P}^j_{i,i+1} = \mathbf{1}, \boldsymbol{Q}^j_{i,i+1}$ generic.

Arch.	Deformation	MNIST	Fashion MNIST
A	[/]	91.1	84.2
	[PQ]	**94.3**	**86.8**
	[Q]	91.6	85.1
B	[/, /, /]	96.6	87.5
	[PQ, /, /]	**97.6**	**88.1**
	[Q, /, /]	96.8	87.8
C	[/, /, /]	98.1	89.24
	[PQ, /, /]	**98.3**	**89.6**
	[Q, /, /]	98.3	89.5

5 Conclusions

In this work we made the following main contributions: 1) we introduced quantum deformed neural networks and identified potential speedups by running these models on a quantum computer; 2) we devised classically efficient algorithms to train the networks for low entanglement designs of the quantum circuits; 3) we simulated the quantum neural networks on real world data sizes for the first time in the literature and showed modest gains in accuracy due to the quantum deformations. Running these models on a quantum computer will allow one to explore efficiently more general deformations, even those that cannot be approximated by the central limit theorem when the Hamiltonians will be sums of non-commuting operators. Another interesting future direction is to incorporate batch normalization and pooling layers in quantum neural networks.

REFERENCES

Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments, 2016.

Jonathan Allcock, Chang-Yu Hsieh, Iordanis Kerenidis, and Shengyu Zhang. Quantum algorithms for feedforward neural networks. *arXiv e-prints*, art. arXiv:1812.03089, December 2018.

Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J. Osborne, Robert Salzmann, and Ramona Wolf. Efficient Learning for Deep Quantum Neural Networks. *arXiv e-prints*, art. arXiv:1902.10445, February 2019.

Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. , 549(7671):195–202, September 2017. doi: 10.1038/nature23474.

Yudong Cao, Gian Giacomo Guerreschi, and Alán Aspuru-Guzik. Quantum Neuron: an elementary building block for machine learning on quantum computers. *arXiv e-prints*, art. arXiv:1711.11240, November 2017.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations. *arXiv e-prints*, art. arXiv:1806.07366, Jun 2018.

Carlo Ciliberto, Mark Herbster, Alessand ro Davide Ialongo, Massimiliano Pontil, Andrea Rocchetto, Simone Severini, and Leonard Wossnig. Quantum machine learning: a classical perspective. *Proceedings of the Royal Society of London Series A*, 474(2209):20170551, January 2018. doi: 10.1098/rspa.2017.0551.

Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, August 2019. doi: 10.1038/s41567-019-0648-8.

Edward Farhi and Hartmut Neven. Classification with Quantum Neural Networks on Near Term Processors. *arXiv e-prints*, art. arXiv:1802.06002, February 2018.

Christopher A. Fuchs and Rüdiger Schack. Quantum-Bayesian coherence. *Reviews of Modern Physics*, 85(4):1693–1715, Oct 2013. doi: 10.1103/RevModPhys.85.1693.

G. Hooft. *The Cellular Automaton Interpretation of Quantum Mechanics*. Fundamental Theories of Physics. Springer International Publishing, 2016. ISBN 9783319412856. URL `https://books.google.co.uk/books?id=ctlCDwAAQBAJ`.

William Huggins, Piyush Patil, Bradley Mitchell, K. Birgitta Whaley, and E. Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and Technology*, 4 (2):024001, Apr 2019. doi: 10.1088/2058-9565/aaea94.

Iordanis Kerenidis, Jonas Landman, and Anupam Prakash. Quantum algorithms for deep convolutional neural networks, 2019.

Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick, 2015.

David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

Yoav Levine, David Yakira, Nadav Cohen, and Amnon Shashua. Deep Learning and Quantum Entanglement: Fundamental Connections with Implications to Network Design. *arXiv e-prints*, art. arXiv:1704.01552, April 2017.

Yoav Levine, Or Sharir, Nadav Cohen, and Amnon Shashua. Quantum Entanglement in Deep Learning Architectures. , 122(6):065301, February 2019. doi: 10.1103/PhysRevLett.122.065301.

Ding Liu, Shi-Ju Ran, Peter Wittek, Cheng Peng, Raul Blázquez García, Gang Su, and Maciej Lewenstein. Machine Learning by Unitary Tensor Network of Hierarchical Tree Structure. *arXiv e-prints*, art. arXiv:1710.04833, October 2017.

E. Miles Stoudenmire and David J. Schwab. Supervised Learning with Quantum-Inspired Tensor Networks. *arXiv e-prints*, art. arXiv:1605.05775, May 2016.

M.E. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000. ISBN 9780521635035. URL https://books.google.co.uk/books?id=aai-P4V9GJ8C.

Jorn W. T. Peters and Max Welling. Probabilistic binary neural networks, 2018.

Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. Simulating a perceptron on a quantum computer. *Physics Letters A*, 379(7):660–663, Mar 2015. ISSN 0375-9601. doi: 10.1016/j.physleta.2014.11.061. URL http://dx.doi.org/10.1016/j.physleta.2014.11.061.

Oran Shayer, Dan Levi, and Ethan Fetaya. Learning discrete weights using the local reparameterization trick. *arXiv preprint arXiv:1710.07739*, 2017.

Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pp. 217–228, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367059. doi: 10.1145/3313276.3316310. URL https://doi.org/10.1145/3313276.3316310.

Guillaume Verdon, Jason Pye, and Michael Broughton. A Universal Training Algorithm for Quantum Deep Learning. *arXiv e-prints*, art. arXiv:1806.09729, June 2018.

Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical review letters*, 91(14):147902, 2003.

Nathan Wiebe, Ashish Kapoor, and Krysta M. Svore. Quantum Deep Learning. *arXiv e-prints*, art. arXiv:1412.3489, December 2014.

Zhikuan Zhao, Jack K. Fitzsimons, and Joseph F. Fitzsimons. Quantum-assisted gaussian process regression. *Phys. Rev. A*, 99:052331, May 2019a. doi: 10.1103/PhysRevA.99.052331. URL https://link.aps.org/doi/10.1103/PhysRevA.99.052331.

Zhikuan Zhao, Jack K. Fitzsimons, Michael A. Osborne, Stephen J. Roberts, and Joseph F. Fitzsimons. Quantum algorithms for training gaussian processes. *Phys. Rev. A*, 100:012304, Jul 2019b. doi: 10.1103/PhysRevA.100.012304. URL https://link.aps.org/doi/10.1103/PhysRevA.100.012304.