Quantum Neural Networks: Algorithms, and Applications

Ibad Momin, Karl Azangue

The University of Texas at Dallas
Computer Science

Abstract:
Quantum neural networks (QNNs) are an emerging field that combines principles of quantum computing with artificial neural networks, potentially enabling more efficient and powerful machine learning models. This paper provides an overview of QNNs, exploring their algorithms, applications, implementation approaches, and challenges. QNNs leverage quantum phenomena such as superposition and entanglement to perform computations, with the potential for exponential speedups over classical neural networks. Various quantum algorithms for encoding data, performing convolutions, and backpropagation in QNNs are discussed. Promising applications span quantum chemistry, materials science, optimization, cryptography, and quantum simulation. However, realizing large-scale practical QNNs faces significant hurdles, including maintaining quantum coherence, managing interconnections, developing miniaturization technologies, and scaling to larger numbers of qubits while mitigating errors. Physical implementation approaches like nuclear magnetic resonance, quantum dots, and other systems are explored, each with its own advantages and limitations. Despite the challenges, QNNs offer compelling theoretical advantages such as exponential memory capacity, faster learning, and the ability to solve classically intractable problems, motivating continued research in this rapidly evolving field at the intersection of quantum computing and machine learning.

I. Introduction

As quantum computing and quantum computers become increasingly popular, we are learning new applications for them and finding ways to utilize their extraordinary computing powers. Currently, there are just a few handfuls of quantum computers in the world, however soon there will be many more, and every big technology company will get on board to take part in the race, just like right now we are seeing the AI and ML boom. Following Moore's law, we can make the assumption that within a few decades, quantum computers will be cheap enough for more and more companies to start building their version of computers and applications for those computers.
One of the applications of quantum computers will be quantum machine learning. Machine learning is the subset of artificial intelligence that learns from previous experiences. Today machine learning is being used in every industry from education to healthcare to solve challenging problems. Using the quantum computer's superposition property of being in 0 or 1 state at the same time, we build more efficient quantum neural networks that will open our doors for quantum machine learning models. In this paper, we will discuss the current application of quantum neural networks, quantum neural networks to build quantum machine learning, and some real-world use cases of quantum neural networks. The algorithms of quantum neural networks that are currently

being used to design neural networks. We will also compare quantum neural networks to the classical neural networks to find the effectiveness of quantum neural networks compared to classical models and the problems of using quantum neural networks.

II. Applications of Quantum Neural Networks

While quantum neural networks are still in the early research stages, they hold significant potential for various applications due to their ability to process quantum data and leverage quantum parallelism. However, it's important to note that practical, large-scale implementations of quantum neural networks are still years away, as they require substantial advancements in quantum hardware and error correction techniques.

Quantum Chemistry and Materials Science
One of the most promising applications of quantum neural networks is in the field of quantum chemistry and materials science. Quantum neural networks can be used to model and simulate the behavior of molecules and materials at the quantum level, which is crucial for understanding their properties and designing new materials with desired characteristics. Traditional classical methods for simulating quantum systems often struggle with the exponential complexity of these problems, making quantum neural networks a potentially more efficient and accurate approach.

Quantum Machine Learning
Quantum neural networks can also be applied to quantum machine learning tasks, such as quantum data classification, quantum pattern recognition, and quantum data compression. By leveraging quantum parallelism and quantum entanglement, quantum neural networks may be able to outperform classical neural networks in certain tasks, particularly those involving high-dimensional data or complex quantum systems.

Quantum Optimization
Quantum neural networks can be employed for solving complex optimization problems, such as combinatorial optimization, scheduling, and routing problems. These problems are often computationally intractable for classical computers, but quantum neural networks may be able to find better solutions by exploiting quantum effects like superposition and entanglement.

Quantum Cryptography and Communication
Quantum neural networks could also find applications in quantum cryptography and secure communication. By leveraging quantum properties like entanglement and superposition, quantum neural networks may be able to develop more robust and secure cryptographic protocols, as well as improve the efficiency of quantum communication channels.

Quantum Simulation and Modeling
Beyond chemistry and materials science, quantum neural networks can be used for simulating and modeling various quantum systems, such as quantum computing devices, quantum sensors, and quantum control systems. This could lead to a better understanding and design of these systems, as well as the development of new quantum technologies.
It's important to note that while these applications are theoretically possible, their practical realization will depend on overcoming significant challenges in quantum hardware development, error correction, and the scalability of quantum neural network architectures.

III. Algorithms of Quantum Neural Networks

Algorithms

https://arxiv.org/pdf/2011.00027

Quantum neural networks are a subclass of variational quantum algorithms, consisting of quantum circuits that contain parameterized gate operations. Information is first encoded into a quantum state via a state preparation routine or feature map. The choice of feature map is usually geared toward enhancing the performance of the quantum model and is typically neither optimized nor trained. Once data is encoded into a quantum state, a variational model containing parameterized gates is applied and optimized for a particular task. This happens through loss function minimization, where the output of a quantum model can be extracted from a classical post-processing function that is applied to a measurement outcome.

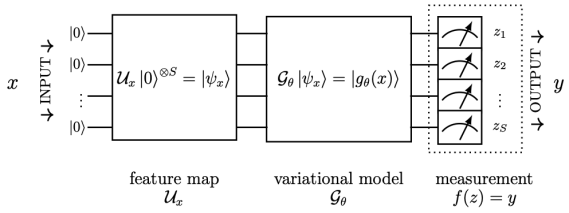

Figure 1: Overview of the quantum neural network used in this study. The input x ∈ Rsin is encoded into an S-qubit Hilbert space by applying the feature map |ψx⟩ := Ux |0⟩⊗S. This state is then evolved via a variational form |gθ(x)⟩ := Gθ |ψx⟩, where the parameters θ ∈ Θ are chosen to minimize a certain loss function. Finally, a measurement is performed whose outcome z = (z1,...,zS) is post-processed to extract the output of the model y = f(z).

First, Hadamard gates are applied to each qubit. Then, normalized feature values of the data are encoded using RZ-gates with rotation angles equal to the feature values of the data. This is then accompanied by RZZ-gates that encode higher orders of the data, i.e. the controlled rotation values depend on the product of feature values. The RZ and RZZ-gates are then repeated. Once data is encoded, the model optimizes a variational circuit Gθ containing parameterized RY-gates with CNOT entangling layers between every pair of qubits, where θ ∈ Θ denotes the trainable parameters. The post-processing step measures all qubits in the σz basis and classically computes the parity of the output bit strings. For simplicity, we consider binary classification, where the probability of observing class 0 corresponds to the probability of seeing even parity, and similarly, for class 1 with odd parity.

QNN -

https://ieeexplore.ieee.org/abstract/document/9528698?casa_token=eZQgaL5N

The way a QNN processes data is as follows. First, the input data is encoded into the corresponding qubit state of an appropriate number of qubits. Then, the qubit state is transformed through the parameterized rotation gates and entangling gates for a given number of layers. The transformed qubit state is then measured by obtaining the expected value of a Hamiltonian operator, such as Pauli gates. These measurements are decoded back into the form of appropriate output data. The parameters are then updated by an optimizer like Adam optimizer. A neural network constructed in the form of VQC can perform various roles in various forms, which will be explored as quantum neural networks.

QCNN -

https://www.nature.com/articles/s41567-019-0648-8

QCNNs are a hybrid approach that combines classical convolutional neural network (CNN) architectures with quantum computing principles. According to these papers [5], [19], the QCNN circuit computation proceeds as follows. The first step is the same as any other QNN model, but the input data (e.g. images) encoded into a qubit state with rotation operator gates, which is then processed through alternating quantum convolutional and quantum pooling layers with quasi-local unitary gates filter the input data into a feature map. The convolutional layers use parameterized quantum gates to capture the spatial relationships between the input data, leveraging quantum phenomena like superposition and entanglement. The pooling layers then reduce the dimensionality of the quantum circuit by repeating this process sufficiently, the fully connected layer acts on the qubit similar to how classical CNNs use pooling to reduce the size of feature maps. Then the measurement of the qubit state is decoded into output data with desired sizes with the circuit parameters being updated with a gradient descent-based optimizer after each measurement. This hybrid approach allows QCNNs to potentially take advantage of quantum speedups while retaining the powerful feature extraction capabilities of classical CNNs.

Forward Passing

In this section, we will design quantum procedures for the usual operations in a CNN layer. The forward pass algorithm for the QCNN is given as Algorithm 1. First, to perform a convolution product between an input and a kernel, we use the mapping between convolution of tensors and matrix multiplication, which can further be reduced to inner product estimation between vectors. The output will be a quantum state representing the result of the convolution product, from which we can sample to retrieve classical information to feed the next layer.

Algorithm 1 QCNN Layer

Require: Data input matrix A^ℓ and kernel matrix F^ℓ stored in QRAM. Precision parameters ϵ and η, a non linearity function $f : \mathbb{R} \mapsto [0, C]$.

Ensure: Outputs the data matrix $A^{\ell+1}$ for the next layer, result of the convolution between the input and the kernel, followed by a non linearity and pooling.

1: **Step 1: Quantum Convolution**

1.1: Inner product estimation

Perform the following mapping, using QRAM queries on rows A^ℓ_p and columns F^ℓ_q, along with Theorems 4.1 and 4.2 to obtain

$$\frac{1}{K}\sum_{p,q}|p\rangle|q\rangle \mapsto \frac{1}{K}\sum_{p,q}|p\rangle|q\rangle|\overline{P}_{pq}\rangle|g_{pq}\rangle\,, \tag{10}$$

where $\overline{P}_{pq}$ is ϵ-close to $P_{pq} = \frac{1+\langle A^\ell_p | F^\ell_q\rangle}{2}$ and $K = \sqrt{H^{\ell+1}W^{\ell+1}D^{\ell+1}}$ is a normalisation factor. $|g_{pq}\rangle$ is some garbage quantum state.

1.2: Non linearity

Use an arithmetic circuit and two QRAM queries to obtain $\overline{Y}^{\ell+1}$, an ϵ-approximation of the convolution output $Y^{\ell+1}_{p,q} = \langle A^\ell_p, F^\ell_q\rangle$ and apply the non-linear function f as a boolean circuit to obatin

$$\frac{1}{K}\sum_{p,q}|p\rangle|q\rangle|f(\overline{Y}^{\ell+1}_{p,q})\rangle|g_{pq}\rangle\,. \tag{11}$$

2: **Step 2: Quantum Sampling**

Use Conditional Rotation and Amplitude Amplification to obtain the state

$$\frac{1}{K}\sum_{p,q}\alpha'_{pq}|p\rangle|q\rangle|f(\overline{Y}^{\ell+1}_{pq})\rangle|g_{pq}\rangle\,. \tag{12}$$

Perform ℓ_∞ tomography from Theorem 4.5 with precision η, and obtain classically all positions and values $(p, q, f(\overline{Y}^{\ell+1}_{pq}))$ such that, with high probability, values above η are known exactly, while others are set to 0.

3: **Step 3: QRAM Update and Pooling**

Update the QRAM for the next layer $A^{\ell+1}$ while sampling. The implementation of pooling (Max, Average, etc.) can be done by a specific update or the QRAM data structure described in Section 5.2.2.

Feed Forwarding Neural Network Propagating

FFNN is essentially composed of a set of individual *ni* nodes, or artificial neurons, arranged in a sequence of successive *L j* layers. Information flows through the network from the input layer to the output layer, traveling through connections between neurons. Artificial neurons combine the input ($\rightarrow i$) and the weight ($\rightarrow i$). ($\rightarrow w$), providing an activation that depends on their$\rightarrow i \cdot \rightarrow w$. Assuming that the quantum register of N-qubits is initially in the rest configuration, $|0\rangle\otimes N$, the quantum state of the input vector with a unitary operation is prepared Ui such that $|\psi i >= Ui\,|0\rangle\otimes N$. Next, the weight factors of the $\rightarrow$w vector are applied w on the input state by implementing another unitary transformation, Uw, subject to the constraint $|1\rangle\otimes N = Uw\,|\psi w\rangle$. When several copies of the quantum register run in parallel, these, and the result of measurements made on them, can be used to feed information about the processing of the input weight to a successive layer.

Back Propagating

In this section, we want to give a quantum algorithm to perform backpropagation on layer l, and detail the impact on the derivatives, given by (Algorithm 2) The entire QCNN is made of multiple layers. For the last layer's output, we expect only one possible outcome, or a few in the case of a classification task, which means that the dimension of the quantum output is very small. A full tomography can be performed on the last layer's output in order to calculate the outcome. The loss L is then calculated, as a measure of correctness of the predictions compared to the ground truth. As the classical CNN, our QCNN should be able to perform the optimization of its weights (elements of the kernels) to minimize the loss by an iterative method.

Algorithm 2 Quantum Backpropagation

Require: Precision parameter δ. Data matrices A^ℓ and kernel matrices F^ℓ stored in QRAM for each layer ℓ.

Ensure: Outputs gradient matrices $\frac{\partial \mathcal{L}}{\partial F^\ell}$ and $\frac{\partial \mathcal{L}}{\partial Y^\ell}$ for each layer ℓ.

1: Calculate the gradient for the last layer L using the outputs and the true labels: $\frac{\partial \mathcal{L}}{\partial Y^L}$

2: **for** $\ell = L-1, \cdots, 0$ **do**

3: **Step 1 : Modify the gradient**
With $\frac{\partial \mathcal{L}}{\partial Y^{\ell+1}}$ stored in QRAM, set to 0 some of its values to take into account pooling, tomography and non linearity that occurred in the forward pass of layer ℓ. These values correspond to positions that haven't been sampled nor pooled, since they have no impact on the final loss.

4: **Step 2 : Matrix-matrix multiplications**
With the modified values of $\frac{\partial \mathcal{L}}{\partial Y^{\ell+1}}$, use quantum linear algebra (Theorem 4.4) to perform the following matrix-matrix multiplications

$$\begin{cases} (A^\ell)^T \cdot \frac{\partial \mathcal{L}}{\partial Y^{\ell+1}} \\ \frac{\partial \mathcal{L}}{\partial Y^{\ell+1}} \cdot (F^\ell)^T \end{cases} \tag{46}$$

to obtain quantum states corresponding to $\frac{\partial \mathcal{L}}{\partial F^\ell}$ and $\frac{\partial \mathcal{L}}{\partial Y^\ell}$.

5: **Step 3 : ℓ_∞ tomography**
Using the ℓ_∞ tomography procedure given in Algorithm 3, estimate each entry of $\frac{\partial \mathcal{L}}{\partial F^\ell}$ and $\frac{\partial \mathcal{L}}{\partial Y^\ell}$ with errors $\delta \left\| \frac{\partial \mathcal{L}}{\partial F^\ell} \right\|$ and $\delta \left\| \frac{\partial \mathcal{L}}{\partial Y^\ell} \right\|$ respectively. Store all elements of $\frac{\partial \mathcal{L}}{\partial F^\ell}$ in QRAM.

6: **Step 4 : Gradient descent**
Perform gradient descent using the estimates from step 3 to update the values of F^ℓ in QRAM:

$$F^\ell_{s,q} \leftarrow F^\ell_{s,q} - \lambda \left(\frac{\partial \mathcal{L}}{\partial F^\ell_{s,q}} \pm 2\delta \left\| \frac{\partial \mathcal{L}}{\partial F^\ell} \right\|_2 \right) \tag{47}$$

7: **end for**

IV. Implementation of QNN

How can quantum neural networks be implemented as real physical devices? First, let us mention briefly some of the difficulties we might face in the development of a physical realization of quantum neural networks. Coherence. One of the most difficult problems in the development of any quantum computational system is the maintenance of the system's coherence until the computation is complete. This loss of coherence (decoherence) is due to the interaction of the quantum system with its environment. In quantum cryptography this problem may be resolved using error-correcting codes. What about quantum neural networks? It has been suggested that if fact these systems may be implemented before ordinary quantum computers will be realized because of significantly lower demands on the number of qubits necessary to represent network nodes and also because of the relatively low number of state transformations required during data processing in order to perform useful computation. Another approach to the problem of decoherence in quantum parallel distributed processing proposed by Chrisley excludes the use of superposition states at all and suggests the use of quantum systems for implementing standard neural paradigms, i.e. multilayer neural systems trained with backpropagation learning. This model however, takes no advantage of the use of quantum parallelism. A more promising approach to the implementation of quantum associative memory based on the use of Grover's algorithm is provided by bulk spin resonance computation (see below) connections. The high density of interconnections between processing elements is a major difficulty in the implementation of small-scale integration of computational systems. In ordinary neurocomputers, these connections are made via wires. In (non-superpositional) quantum neurocomputers they are made via forces. In the quantum associative memory model discussed here, these connections are due to the entanglement of qubits. Physical systems. Now we can outline what kind of physical systems might be used to develop real quantum neural networks and how these systems address the problems listed above. Nuclear Magnetic Resonance. A promising approach to the implementation of quantum associative memory

based on the use of Grover's algorithm is provided by bulk spin resonance computation. This technique can be performed using Nuclear Magnetic Resonance systems for which coherence times on the order of thousands of seconds have been observed. Experimental verification of such an implementation has been done by Gershenfeld and Chuang (among others), who used NMR techniques and a solution of chloroform (CHCl3) molecules for the implementation of Grover's search on a system consisting of two qubits – the first qubit is described by the spin of the nucleus of the isotope C13, while second one is described by the spin of the proton (hydrogen nucleus). Rather interestingly, this approach to quantum computation utilizes not a ingle quantum system but rather the statistical average of many copies of such a system (a collection of molecules). It is precisely for this reason that the maintenance of system coherence times is considerably greater than for true quantum implementations. Further, this technology is relatively mature, and in fact coherent computation on seven qubits using NMR has recently been demonstrated by Knill, et al. This technology is most promising in the short term, and good progress in this direction is possible in the early 21st century.

Quantum dots. These quantum systems basically consist of a single electron trapped inside a cage of atoms. These electrons can be influenced by short laser pulses. Limitations to these systems that must be overcome include 1) short decoherence times due to the fact that the existence of the electron in its excited state lasts about a microsecond, and the required duration of a laser pulse is around a nanosecond; 2) the necessity of developing a technology to build computers from quantum dots of very small scale (10 atoms across); 3) the necessity of developing special lasers capable of selectively influencing different groups of quantum dots with different wavelengths of light. The use of quantum dots as the basis for the implementation of QNN is being investigated by Behrman and co-workers.

Other systems. There are many other physical systems that are now being considered as possible candidates for the implementation of quantum computers (and therefore possibly quantum neurocomputers). These include various schemes of cavity QED (quantum electrodynamics of atoms in optical cavities), ion traps, SQUIDs (superconducting quantum interference devices), etc. Each has its own advantages and shortcomings with regard to decoherence times, speed, possibility of miniaturization, etc. More information about these technologies can be found in paper by S.K Jaswal..

V. Comparison

Can QNN outperform classical neural networks? It is now known that quantum computing gives us unprecedented possibilities in solving problems beyond the abilities of classical computers. For example Shor's algorithm gives a polynomial solution (on a quantum computer) for the problem of prime factorization, which is believed to be classically intractable. Also, as previously mentioned, Grover's algorithm provides super-classical performance in searching an unsorted database. What about quantum neural networks? Will they give us some advantages unattainable by either traditional von Neumann computation or classical artificial neural networks? Compared to the latter, quantum neural networks will probably have the following advantages:
• exponential memory capacity; • higher performance for a lower number of hidden neurons; • faster learning; • elimination of catastrophic forgetting due to the absence of a pattern interference; • single-layer network solution of linearly inseparable problems; • absence of wires; • processing speed (1010 bits/s); • small scale (1011 neurons/mm3); •

higher stability and reliability; These potential advantages of quantum neural networks are indeed compelling motivation for their development. However, the more remote future possibilities of QNN maybe even more exciting.

VI. Problems

While quantum neural networks and quantum computing hold immense potential, there are significant challenges that must be overcome for their practical realization. One of the most formidable obstacles is maintaining the coherence of the quantum system throughout the computation process, as decoherence caused by interaction with the environment can disrupt the delicate quantum states. Implementing quantum neural networks as physical devices also faces difficulties such as maintaining coherence times, managing the high density of interconnections between processing elements, and developing technologies for miniaturization and specialized lasers. Certain physical systems like quantum dots suffer from extremely short decoherence times, limiting their viability. Moreover, scaling up quantum neural networks and quantum computers to handle larger and more complex problems remains a daunting challenge, as increasing the number of qubits while maintaining coherence and control is a major hurdle. Developing effective error-correction techniques is also crucial, as quantum systems are inherently fragile and prone to errors. Ultimately, the widespread adoption of quantum neural networks and quantum computing hinges on significant advancements in quantum hardware, which is still in its early stages. Despite their theoretical advantages, such as exponential memory capacity and faster learning, the practical applications of quantum neural networks are currently limited by the state of quantum hardware and the various challenges mentioned above.

VII. Conclusion

In conclusion, while quantum neural networks hold significant theoretical promise, their practical realization faces numerous formidable challenges. Maintaining coherence, overcoming decoherence, implementing physical systems with long coherence times, managing interconnections, developing miniaturization technologies, and scaling to larger numbers of qubits are just some of the major obstacles that must be surmounted. Effective error correction techniques and substantial advances in quantum hardware are also critical requirements. Nonetheless, the potential advantages of quantum neural networks, such as exponential memory capacity, faster learning, and the ability to solve classically intractable problems, provide a compelling motivation for continued research and development in this field. As quantum technologies continue to evolve and mature, overcoming these challenges may unlock unprecedented computational capabilities that could revolutionize various domains, including quantum chemistry, materials science, optimization, cryptography, and simulation. While the path ahead is arduous, the tantalizing prospects of quantum neural networks and quantum computing make this a pursuit worthy of the scientific community's unwavering efforts.

References

M. Sc. K. Beer, Quantum Neural Networks,

https://arxiv.org/pdf/2205.08154.pdf (accessed May 10, 2024).

J. Landman, Quantum algorithms for unsupervised machine learning ..., https://arxiv.org/pdf/2111.03598.pdf (accessed May 5, 2024).

D. P. Garcia, J. Cruz-Benito, and F. Garcia-Penalvo, Computer science review, https://arxiv.org/pdf/2201.04093 (accessed May 10, 2024).

A. Ezhov and D. Ventura, Quantum Neural Networks, https://axon.cs.byu.edu/papers/ezhov.fdisis00.pdf (accessed May 10, 2024).

S. K. Jeswal and S. Chakraverty, "Recent developments and applications in Quantum Neural Network: A review - archives of computational methods in engineering," SpringerLink, https://link.springer.com/article/10.1007/s11831-018-9269-0 (accessed May 5, 2024).

A. Anshu, N. P. Breuckmann, and C. Nirkhe, NLTS Hamiltonians from Good Quantum Codes, https://arxiv.org/pdf/2206.13228.pdf (accessed May 5, 2024).

R. Feynman, Plenty of room at the bottom - Richard P. Feynman, https://web.pa.msu.edu/people/yang/RFeynman_plentySpace.pdf (accessed May 5, 2024).

A. Zeguendry, Z. Jarir, and M. Quafafou, "Quantum Machine Learning: A Review and Case Studies," MDPI, https://www.mdpi.com/1099-4300/25/2/287 (accessed May 5, 2024).

Y. Kwak, W. J. Yun, S. Jung, and J. Kim, Quantum Neural Networks: Concepts, applications, and Challenges, https://arxiv.org/pdf/2108.01468.pdf (accessed May 10, 2024).