Quantum Neuron Selection: Finding High Performing Subnetworks With Quantum Algorithms

Tim Whitaker
timothy.whitaker@colostate.edu
Colorado State University
Fort Collins, Colorado, USA

ABSTRACT

Gradient descent methods have long been the de facto standard for training deep neural networks. Millions of training samples are fed into models with billions of parameters, which are slowly updated over hundreds of epochs. Recently, it's been shown that large, randomly initialized, neural networks contain subnetworks that perform as well as fully trained models. This insight offers a promising avenue for constructing accurate neural networks by simply pruning weights from large, random models. However, neuron selection is combinatorically hard and there are no classical algorithms that can find the best subnetwork efficiently. In this paper, we explore the efficacy of several quantum algorithms for this neuron selection problem. Additionally, we introduce two methods for local quantum neuron selection that reduce the entanglement complexity that large scale neuron selection would require, making this problem more tractable for current quantum hardware.

KEYWORDS

neural networks, quantum computing, machine learning, neuron selection, nk landscapes

ACM Reference Format:
Tim Whitaker. 2022. Quantum Neuron Selection: Finding High Performing Subnetworks With Quantum Algorithms. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Deep neural networks have demonstrated inordinate success across many difficult machine learning tasks in computer vision, natural language processing, and reinforcement learning. Some of the recent advances in these fields have come about in part thanks to larger datasets and larger models. These state of the art models contain billions of parameters and train for thousands of GPU days [4, 8, 36]. The growing cost of training deep neural networks has led to an increased interest in alternative training methodologies.

Meanwhile, quantum computing continues to garner interest by machine learning researchers as quantum computers have the potential to solve large classes of difficult problems at significantly reduced time complexity compared to classical computers [2]. The mathematical foundations of both machine learning and quantum computing are fundamentally linearly algebraic and there is a lot of hope that quantum methods could enable new ways to drastically improve performance.

Deep neural networks are typically trained with variations of gradient descent. Training data is passed through a network, some loss is determined between the network's output and a target, and the loss is backpropagated through the network according to the gradient with respect to the weights. Modern neural networks repeat this process over millions of training samples for hundreds of epochs.

Recently, there has been some interesting work investigating the subnetworks of these deep neural networks. It's long been known that fully trained deep neural networks can be significantly pruned and maintain high performance after small amounts of tuning [3]. However, when those subnetworks are trained from scratch, they fail to achieve the same accuracy they had after dense training and subsequent pruning. A popular line of research titled The Lottery Ticket Hypothesis explored this problem and introduced a method for finding specific subnetworks, in certain network architectures, that train as well as the full size network it is derived from [14].

The Lottery Ticket Hypothesis has since spurred many follow up works investigating high performing subnetworks. In particular, some works have introduced methods for finding subnetworks in random models that perform well *without any training* [30, 37]. This approach, which we call *neuron selection*, offers an exciting paradigm for building neural networks in the future by simply pruning weights from large random models.

Discovering the best subnetwork in a large neural network is computationally hard due to exponential combinatoric complexity. Despite the difficulty for finding the *best* subnetwork, these large networks contain so many potential subnetworks that there are likely many that would perform well enough. Current approaches often use heuristics and pseudo-training algorithms to find acceptable subnetworks, where solutions are found by assigning scores to edges and optimizing network graphs with something akin to gradient descent [30, 35].

In this paper, we explore applications of quantum algorithms to the neuron selection problem. We demonstrate how neuron selection could be formulated for quantum gate computers, quantum annealing computers, and hybrid quantum-classical computers. We note that the current state of quantum hardware is not developed enough to apply quantum neuron selection at the scale of modern neural networks, so we introduce two local neuron selection algorithms that make these ideas tractable for current quantum systems.

Conference'17, July 2017, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND

Training neural networks by slowly adjusting the weights between neurons has long been the defacto standard in deep learning. However, there are several reasons that alternative training methods may be worth investigating for future neural networks. Insufficient information in the gradients, low signal to noise ratio, architecture bias, and flatness in the activations are all potential classes of problems that may prevent effective learning in gradient based methods [32].

There also exists strong biological motivation for exploring neural network construction via neuron selection. There is a natural stage of brain development in which a massive amount of neurons, axons, and synapses proliferate. This overabundance creates a competitive environment as these brain cells compete for resources. Many millions of neurons and connections are invariably killed and specialized subnetworks develop as a result. Neural Darwinism, a theory of neuronal group selection, explores how brain function evolves as a result of these selective processes acting on and between groups of neurons [10].

Pruning deep neural networks to find effective subnetworks is an old and established method for optimizing model size and cost [23]. However, applying neuron selection purely as a means for training networks is a relatively new idea deep learning. Neural architecture search and neuroevolution are tangentially related as many of these works aim to optimize network architecture along with weights.

Weight Agnostic Neural Networks are one such approach to neural architecture search that forgoes the dependence of weight tuning. This work demonstrates how architecture alone could encode solutions to complex problems where networks are constructed where all weights between neurons are fixed to a single shared value [16].

NK Echo State Networks employ neuron selection as a training method by selecting optimal combinations of neurons from a fixed, random, and recurrent reservoir layer [34]. The complex dynamics of the echo state reservoir enables neuron selection to model dynamic time dependent problems without any weight updates. Finding the optimal combinations of neurons is done by connecting a neuron selection layer (probe filter) to an output layer according to an NK-landscape and optimizing this landscape with dynamic programming. In this case, the NK-landscape is modeled where N neurons in the probe filter layer connect to exactly K neurons in the output layer, reducing the computational complexity from $O(2^N)$ to $O(2^K N)$.

The Lottery Ticket Hypothesis spawned a large amount of interest in neural network pruning over the last few years. This work introduces an approach to finding extremely sparse subnetworks that train well. First, a dense network is fully trained. Then the network is pruned according to the magnitude of the final weights. The network's weights are then rewound back to the initial values while keeping the sparse structure in tact. The resulting sparse network should then train as well as the dense network it is derived from [14].

A followup work, Deconstructing Lottery Tickets, explored the properties of these lottery ticket subnetworks and found that there were subnetworks that were able to perform "better than chance" without any subsequent weight tuning of the lottery ticket. The supermasks that describe these subnetworks could also be applied to randomly initialized neural networks. Zhoa et al. find that these supermasks significantly outperform random masks or random networks, suggesting that architecture does encode some form of inductive bias [37].

Recent work has since shown that subnetworks can be found in much larger and more modern neural network architectures than those explored in the original lottery ticket papers. Ramanujan et al. introduce an algorithm to find subnetworks in a randomly weighted WideResNet-50 that matches the performance of a trained ResNet-34 on ImageNet [30]. This work empirically demonstrated that neuron selection could be applied to problems of modern scale and result in extremely accurate networks with absolutely no weight tuning.

Malach et al. laid the theoretical foundations for this line of work in a paper titled, Proving The Lottery Ticket Hypothesis. Assuming some bounded constraints for the norms of weights and inputs, the authors show that a ReLU network of polynomial width and a depth of $2l$ contains a randomly initialized subnetwork that will approximate any trained neural network of width d and depth l [25].

THEOREM 1. *Fix some $\epsilon, \delta \in (0,1)$. Let F be some target network of depth l such that for every $i \in [l]$ we have $||W^{F(i)}||_2 \leq 1, ||W^{F(i)}||_{max} \leq \frac{1}{\sqrt{n_i n}}$ (where $n_{in} = d$ for $i = 1$ and $n_{in} = n f or i > 1$). Let G be a network of width $poly(d, n, l, \frac{1}{\epsilon}, log\frac{1}{\delta})$ and depth $2l$, where we initialize $W^{G(i)}$ from $U([-1, 1])$. Then, w.p. at least $1 - \delta$ there exists a weight-subnetwork $\tilde{G}$ of G such that:*

$$\sup_{x \in X} |\tilde{G}(x) - F(x)| \leq \epsilon$$

Furthermore, the number of active (non-zero) weights in $\tilde{G}$ is $O(dn + n^2 l)$ [25].

This bound is however much larger than empirically observed in related works. Pensia et al. and Orseau et al. both published proofs that remove the strict assumptions that Malach et al. require and significantly tighten the bounds for the random network to a logarithmic factor [27, 28].

THEOREM 2. *A randomly initialized ReLU network with width $O(dlog(dl/min\{\epsilon, \delta\}))$ and depth $2l$, with probability at least $1 - \delta$, can be pruned to approximate any neural network with width d and depth l up to error ϵ [28].*

Several works have also demonstrated that sufficiently overparameterized networks contain many winning lottery ticket subnetworks [9, 15, 25]. These results are important as they significantly reduce the computational complexity required to find a single best subnetwork. This is especially important for certain quantum search algorithms, in which multiple solutions can drastically reduce the number of iterations required.

The application of quantum algorithms to neuron selection offers a lot of potential opportunities for the future of machine learning. We hope that this work introduces a foundation for which new training methods and network architectures could develop, at a scale that would be infeasible for classical algorithms.

3 QUANTUM NEURON SELECTION

There is a lot of interest and excitement in the application of quantum algorithms to machine learning. Quantum computers have the potential to solve extremely large and difficult problems that would be intractable on classical computers.

Quantum Computers use probabilistic state vectors to describe the state of quantum systems. While classical computers use bits as the basic computational unit, quantum computers use qubits, which represent some superpositional state between $|0\rangle$ and $|1\rangle$.

These state vectors are normalized with coefficients α and β which represent the probabilities of measuring either $|0\rangle$ or $|1\rangle$.

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

This notation also extends to multiple-qubit systems.

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$$

Quantum Annealing and Quantum Gate Computing are the two primary paradigms under which quantum computers operate [19]. Quantum annealers are specialized pieces of hardware that are specifically tailored to specificly formulated optimization problems. Quantum gate computers are analagous to classical gate circuits, yet they make use of special quantum gates that act on qubits.

The state of quantum hardware is very much in its infancy at this stage. Quantum gate computers are limited to small numbers of qubits. The largest quantum gate computer at the time of this writing is the IBM Eagle processor which consists of 127 qubits with a a 64 bit quantum volume [7]. Quantum annealers are able to operate with far more qubits, with the current state of the art being the D-Wave Advantage, which consists of 5,640 qubits [26]. However, quantum annealers do not have the same computational complexity or generality of quantum gate computers and it's unclear whether they can offer the same types of computational advantages over specialized classical algorithms.

We introduce quantum neuron selection and explore how we could frame this problem in both a quantum annealing and quantum circuit framework. In the sections below, we discuss quantum neuron selection as a means for finding high performing subnetworks in randomly weighted neural networks. That is, we apply these quantum algorithms at a parameter level as opposed to a neuronal level. While these algorithms could also apply at a neuronal level, it's been shown that parameter selection is a far more computationally expressive approach, and that neuron subnetworks do not result in the same performance guarantees as parameter subnetworks [25].

We do note however, that neuron subnetworks may be sufficient for certain tasks and this can result in significant computational savings as there is an exponentially fewer number of neurons than parameters in any given fully connected network. We also note that the principles behind neuron selection very much apply to several kinds of neural architectures. We focus on fully connected and convolutional networks, but we note that neuron selection could theoretically be applied to recurrent and attention based networks as well.

3.1 Quantum Optimization Methods

Neuron selection can be thought of as a search or combinatorial optimization problem, in which given a neural network F consisting of N parameters, our goal is to find a binary bitmask $M \in \{0,1\}^N$ such that the network with the mask applied $F \circ M$ performs well on some test dataset. This problem fits well with quantum computing as many quantum algorithms often operate directly in quantum space where qubits collapse to basis states $|0\rangle$ and $|1\rangle$. Below we explore the most popular quantum optimization methods using quantum gates, quantum annealing, and hybrid quantum-classical approaches.

3.1.1 Amplitude Amplification. Grover's algorithm is a prominent quantum algorithm for unstructured search. Consider searching through a list of N items. In order to find a specific, unique item, one would need to check on average $N/2$ locations on a classical computer. In the worst case, one would need to check all N locations. On a quantum computer, Grover's algorithm can find the marked item in approximately $\sqrt{N}$ iterations. Grover's algorithm offers quadratic runtime speedup over naive classical methods and can save significant computation with large search spaces [18].

The generality of this approach lies in the fact that it is often difficult to find the correct solution, but easy to verify a solution. Grover's algorithm begins by placing all qubits into a uniform superposition.

$$|s\rangle = \frac{1}{N}\sum_{x=1}^{N} |x\rangle$$

An oracle is queried which flips the amplitude of some marked state(s). The oracle is implemented as a unitary operator that returns -1 for any marked state ω and 1 otherwise.

$$U_\omega\,|x\rangle = \begin{cases} |x\rangle & if\ x \neq \omega \\ -|x\rangle & if\ x = \omega \end{cases}$$

In our case we can implement the oracle with a function f. The oracle will take a proposed subnetwork bitmask x and run the neural network with the bit mask applied to a validation set. The function will return 0 if the subnetwork performs better than some threshold and return 1 if it does not.

$$U_f\,|x\rangle = (-1)^{f(x)}\,|x\rangle$$

This result is then passed through a diffusion operator which flips all qubit amplitudes about the mean.

$$U_d = 2\,|s\rangle\,\langle s| - I$$

The oracle and diffusion operations are then repeated several times to amplify the probability that the marked state(s) will be measured. After t iterations, the state of the system can be described as

$$|\psi_t\rangle = (U_f U_d)^t\,|s\rangle$$

The optimal number of iterations is dependent on the number of matched states. In the case that there is only a single potential subnetwork, the optimal number of iterations to run Grover's algorithm is $\frac{\pi}{4}\sqrt{N}$, where N is the total number of subnetworks.

However, as the potential number of subnetworks in a given model is 2^N, it's likely that large networks may have several subnetworks that are acceptable. By tuning the oracle's threshold, we can reduce the optimal number of iterations to $\frac{\pi}{4}\sqrt{\frac{N}{k}}$ where k is the number of matching subnetworks.

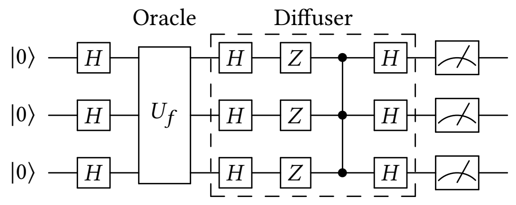

Figure 1: A 3-qubit circuit implementation of Grover's algorithm.

While Grover's algorithm offers provably quadratic speedup over classical methods, the search space of subnetworks in deep neural networks is still extremely large. Some set of constraints or heuristic information to guide search will be necessary to apply Grover's algorithm on large scale neural networks.

Additionally, Grover's algorithm would require a quantum circuit implementation of an oracle that can determine the validity of a given subnetwork. That is, we need to create a circuit that can store data, perform matrix multiplication, and apply activation functions on floats which consist of many bits per value. The current state of software makes this impractical. However, it has been shown that any classical computation can be compiled down into reversible boolean logic gates [2]. While these circuits would be quite large for neural networks of practical size, this does show that these methods can be implemented on quantum systems once the state of quantum compilation develops further.

3.1.2 Adiabatic Computation. Many combinatorial optimization problems in quantum computing are solved with methods based on the adiabatic theorem, which states that a quantum system will remain in its instantaneuos eigenstate if a given perturbation is acting on it slowly enough [12]. These approaches slowly evolve a system over time from some initial quantum state towards a corresponding ground state that encodes the solution to the problem.

These approaches are closely related to Ising models, which are used to study magnetic dipole moments of atomic spins [5]. In these models, discrete variables are organized into a lattice and described as spins which can be in either an up or down state (+1 or −1). The local structure of Ising models assumes that spin states interact only with their neighbors. The Hamiltonian of such models can be formulated with an operator of the following form.

$$H = -\sum_{i,j} J_{ij} s_i s_j - \sum_i h_i s_i$$

Where J is a coupling matrix which describes interaction strength between spin states, and h is a transverse magnetic field acting on all spins s. This formulation makes adiabatic methods very popular for satisfiability problems where the interactions of local neighborhood structures can be exploited.

Quantum Annealing is the primary means for performing combinatorial optimization in which a problem Hamiltonian H is constructed that describes the energy state of a system. In combinatorial optimization problems, this can be defined via some cost function C.

$$H|x\rangle = C(x)|x\rangle$$

The cost function $C(x)$ in our case can be described as the total loss of a neural network masked with the given bitstring $x \in \{0,1\}^n$ and evaluated on some validation set. In quantum annealing computers, a problem independent Hamiltonian H_0, called a mixing Hamiltonian, is applied every iteration t to slowly evolve the system over a long time period T according to.

$$H(t) = (1 - \frac{t}{T})H_0 + \frac{t}{T}H_C$$

While the mixing Hamiltonian can vary, the point of it is to apply weak perturbations to the system to encourage exploration of feasible states. A standard example is the bit flip mixer which flips the state of a node if all neighboring nodes are in the same state as the target node.

$$H_M = \sum_{v \in V(G)} \frac{1}{2^{d(v)}} X_v \prod_{w \in N(v)} (\mathbb{I} + (-1)^b Z_w)$$

where $V(G)$ is the set of vertices of some graph G, $d(v)$ is the degree of vertex v, and $N(v)$ is the neighborhood of vertex v. Z_v and X_v are the Pauli-Z and Pauli-X operators on vertex v, and I is the identity operator [20].

In general, optimization needs to formulated in a way that they can be ran on quantum annealing hardware. This generally means that the Hamiltonian and cost function needs to be specified as a Quadratic Unconstrained Binary Optimization (QUBO) problem [17]. In particular, this means formulating our cost function in a manner that accounts for evaluation of a neural network, including matrix multiplication, summation, loss over multiple data samples, and nonlinearities. This formulation would vary for networks of different types, and special network architectures may be needed to efficiently optimize structure with quantum annealers. Sasdelli et al. introduce how one would formulate a binary neural network for QUBO optimization [31].

3.1.3 Variational Circuits. Hybrid quantum-classical methods are perhaps the most popular approach for optimization on current quantum hardware. Variational Circuits combine quantum states with classical optimization by first preparing some initial quantum state $|\psi_0\rangle$. That quantum state is then transformed by a parameterized quantum circuit $U(\theta)$. The state of the quantum system is then measured. The parameters θ are then optimized using a classical algorithm. This process repeats until some convergence criteria is met [6].

Variational Quantum Eigensolver (VQE) and Quantum Approximate Optimization Algorithm (QAOA) are perhaps the two most popular variational methods [11, 29]. These approaches are closely related to the adiabatic methods based on quantum annealing defined above, however these methods operate with quantum logic gates. VQE is a variational approach to quantum phase estimation and QAOA can be thought of as discretized and variational version of quantum annealing [33].

As in quantum annealing, both methods rely on constructing a problem Hamiltonian H that describes the energy state of a system.

$$H|x\rangle = C(x)|x\rangle$$

VQE works by first preparing an initial state according to some set of parameters $|\psi(\theta)\rangle$, often called an ansatz. The expectation of

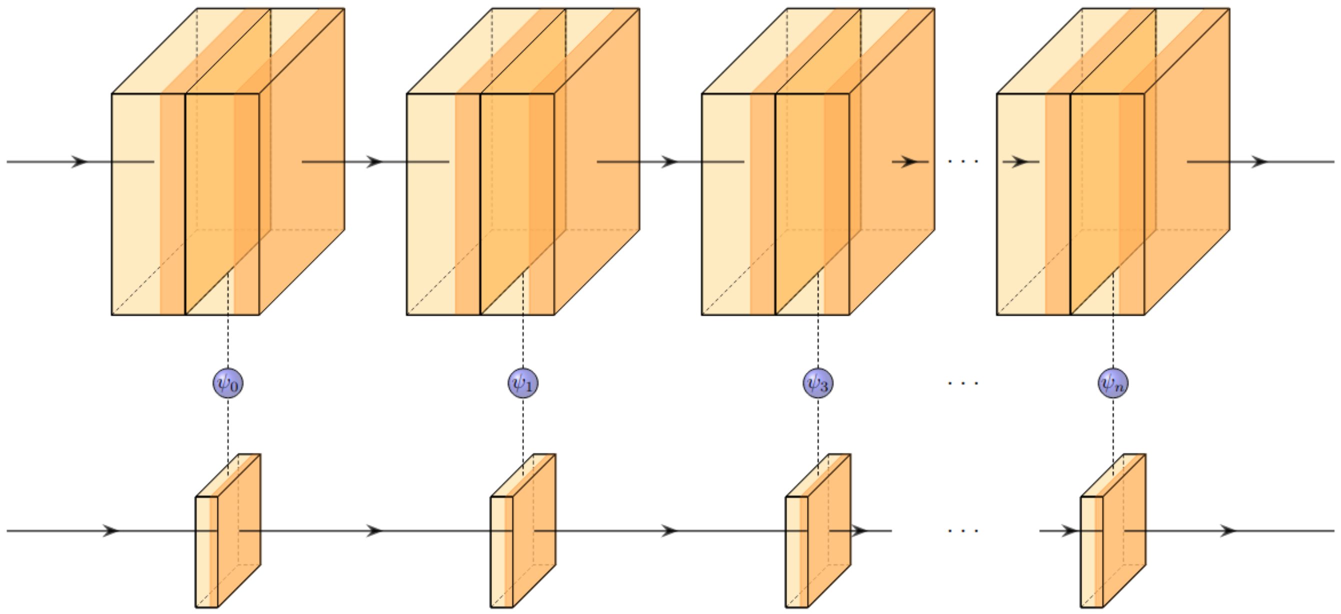

Figure 2: An illustration of local quantum neuron selection via knowledge distillation. A random student network (top) is successively optimized with a quantum neuron selection algorithm ψ_i according to the layerwise representations of a trained teacher network (bottom).

the quantum state is measured $\langle\psi(\theta)| H |\psi(\theta)\rangle$ and the parameters θ are then updated using a classical algorithm.

In QAOA, a special set of unitary operators are used to alter the ansatz. First, a quantum state $|\psi(\gamma,\beta)\rangle$ is prepared using unitary operators of the form $U(\beta) = e^{-i\beta H_0}$ and $U(\gamma) = e^{-i\gamma H}$ with the problem Hamiltonian H and a mixing Hamiltonian H_0 such as the bit-mixer described in the quantum annealing section above. These unitaries are applied repeatedly in blocks p times to some initial state $|\psi_0\rangle$.

$$|\psi(\beta,\gamma)\rangle = U_0(\beta)U_0(\gamma)...U_p(\beta)U_p(\gamma)\,|\psi_0\rangle$$

Then, an expectation is computed using the given parameters γ, β with respect to the problem Hamiltonian H.

$$F_p(\gamma,\beta) = \langle\psi_p(\gamma,\beta)| H |\psi_p(\gamma,\beta)\rangle$$

Finally, a classical optimizer is used to optimize the parameters (γ,β). This process is repeated until some threshold or convergence criteria is met.

In general, variational and annealing algorithms do not have computational complexity guarantees over classical methods [1].

3.2 Layerwise Quantum Neuron Selection With Knowledge Distillation

In the previous section, we approach quantum neuron selection by implementing unstructured search and combinatorial optimization over a bit mask for a fixed, large, and randomly initialized neural network. However, there is a serious challenge with scaling quantum computers that is related to the fragility of quantum states [13]. The current state of quantum hardware and software are quite limited in the potential for implementing large scale quantum neuron selection for neural networks of moderate size.

Divide and Conquer methods are one such solution where large scale optimization is broken down into manageable subproblems [24]. Here we introduce a method for local neuron selection built on the popular practice of knowledge distillation for supervised learning problems [21]. This allows us to apply neuron selection layer by layer, using a small, trained teacher network as a guide. This hybrid approach can enable the discovery of effective subnetworks in large neural networks without needing to place the entire network into a quantum state.

Consider two neural networks, a small fully trained teacher F_t and a large randomly initialized student network F_s. Assuming the student network is sufficiently wide and twice as deep as the target network, it is reductively possible that every two layers in the student network should contain a subnetwork that can approximate a single corresponding layer in the fully trained teacher network.

We start by running a validation set through the teacher network, where the outputs at each layer are recorded for all the training samples. We then perform quantum neuron selection on the student network F_s two layers at a time, where the goal is to find the subnetwork for those two layers that approximates the target layer in the teacher network F_t. We can use several of the quantum optimization methods described in the previous sections to find the optimal bit string that maximizes similarity between the teacher network's layer output and the student network's layer output.

The loss for a given block i for input x is the L2 norm of the layer outputs between the teacher network F_{t_i} and the student network

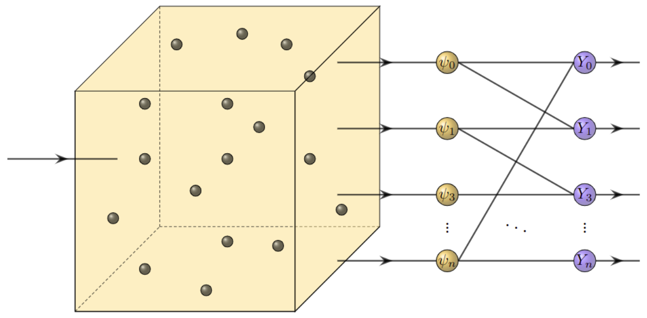

Figure 3: A network architceture diagram of an NK Echo State Network. A recurrent reservoir is connected to a probe filter layer which is then connected to an ensemble of output nodes Y. The probe filter layer ψ is able to be optimized with an efficient quantum neuron selection algorithm.

F_{s_i} after a bit mask M is applied.

$$\mathcal{L}_i(x) = ||F_{t_i}(x) - ((M_i \circ F_{s_i})(x))||_2$$

We also need to compress the student networks outputs such that the vectors are of the same dimensionality. As the student network's outputs are sparse and contain approximately the same number of active outputs as the teacher, it is trivial to compress this matrix with minimal loss of information with either magnitude pruning or average pooling.

Knowledge distillation allows us to tackle the problem of neuron selection in an iterative manner, regardless of network depth. While this approach would necessitate a fully trained teacher network, the computational advantages that local quantum neuron selection provides would enable applications on near-term quantum hardware for deep neural networks.

3.3 Reducing Entanglement Complexity With NK Landscapes

Along with applying neuron selection to localized network structures, it's possible to impose constraints on neural network connectivity in order to reduce the entanglement complexity of quantum systems.

Below we illustrate how NK Echo State Networks provide an elegant framework for implementing localized neuron selection with a quantum classification layer.

3.3.1 NK Echo State Networks. Echo State Networks belong to a class of recurrent networks called reservoir computing models. In the simplest case, an echo state network consists of an input layer, a reservoir, and an output layer. During training, only the weights associated with the output layer are updated [22].

The reservoir contains a moderate amount of sparse and randomly connected neurons. At each iteration, the output of the reservoir is fed back in to the reservoir in a recurrent fashion. The connectivity of the neurons and the weights associated are initialized such that the spectral radius of the weight matrix is less than 1. This property ensures that the recurrent state that is passed back into the reservoir slowly decays over time, like an echo.

Echo State Networks can be succinctly described with the following system equations:

$$z_{res}(x_t) = W_{res} z_{res}(x_{t-1}) + W_{in} x \tag{1}$$
$$y(x_t) = \sigma(W_{out} z_{res}(x_t)) \tag{2}$$

where $z_{res}(x_t)$ is the reservoir state at iteration t for a given input x. The input, reservoir, and output weights are W_{in}, W_{res}, and W_{out} respectively. The output, $y(x_t)$, is gotten by multiplying the reservoir state with the output weights and applying a nonlinear activation function σ.

Echo State Networks are fast, simple to implement and well suited for dynamic time series data, thanks to the chaotic and cyclical behavior of the reservoir.

NK Echo State Networks are a variation on echo state networks that utilize neuron selection by incorporating a probe filter layer between the output layer and the reservoir [34]. The probe filter layer contains N neurons that each connect randomly to neurons within the reservoir. Turning off and on different combinations of neurons in the probe filter layer results in extremely diverse reservoir dynamics as the neural circuitry is altered.

The network is trained only by optimizing a bitmask M which describes which probe filter neurons are turned on or off. The system dynamics are then described as:

$$z_{res}(x_t) = W_{res} z_{res}(x_{t-1}) + W_{in} x \tag{3}$$
$$z_{pf}(x_t) = M \circ W_{pf} z_{res}(x_t) \tag{4}$$
$$y(x_t) = \sigma(W_{out} z_{pf}(x_t)) \tag{5}$$

where z_{pf} is the output of the probe filter layer, W_{pf} are the weights of the probe filter layer, $M \in \{0,1\}^n$ is an n-dimensional bit vector that describes which neurons to turn on and off, and $\circ$ is the Hadamard product.

The probe filter layer is connected to the output layer according to an NK-landscape, where N neurons in the probe filter layer are connected to exactly K neurons in the output layer. This formulation allows each seperate output neuron to act as a k-bounded pseudo-boolean subfunction $f_i(x)$ in which the search space is restricted to the size of K.

For each output neuron, performance is recorded for every possible on/off pattern for the K neurons in the probe filter layer that connect to it. This results in a look-up table for performance evaluations with $2^K N$ entries. The bitstring is then optimized using dynamic programming in order to maximize the performance when all outputs are averaged together.

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

3.3.2 NK Quantum Neuron Selection. NK Echo State Networks make use of clever ensembling scheme in which output neurons are duplicated and connected to different subsets of neurons in the probe filter layer according to an NK-landscape. This set up allows each output neuron to be evaluated independently, in which combinations of only the K neurons that connect to it need to be exhausted.

We note that this connective scheme can significantly reduce the entanglement complexity that quantum neuron selection would normally entail. Consider an output and probe filter layer that contains N neurons. The output for the ith neuron can

$$y_i = \phi(\sum_{j}^{K} w_{mask(i,j),i} S(mask(i,j)) x(mask(i,j)))$$

where ϕ is an activation function, $mask(i,j)$ is a lookup table that returns the index of the jth neuron that connects to the ith output, $w_{i,j}$ is the weight between the ith and jth neuron, and $S(mask(i,j))$ is the output of the jth neuron in the probe filter layer, and $x(mask(i,j))$ indicates whether the neuron in the probe filter layer is turned on or off [34].

Quantum Neuron Selection then only needs to be formulated for optimizing a quantum state consisting of K qubits. Quantum neuron selection can then be applied to each output neuron independently. Assuming we implement Grover's algorithm, as described in section 3, we can reduce the computational complexity for finding the optimal bitmask for the probe filter layer from $O(2^K N)$ to $O(\sqrt{2^K} N)$.

While neuron selection works particularly well for echo state networks, due to the complex dynamics of the reservoir state, we note that an quantum probe filter and ensemble output can be embedded in network architectures of any type. NK connectivity enables computationally efficient pseudo-ensembles where layers of any width can be optimized with quantum neuron selection.

4 CONCLUSIONS

Several works have demonstrated that large neural networks contain subnetworks that perform as well as fully trained networks [25, 27, 28, 30]. However, there is no efficient classical algorithm for finding the best subnetwork in a given network. We discuss several prominent quantum optimization algorithms, including Grover's Algorithm, Quantum Annealing, Variational Quantum Eigensolver, and Quantum Approximate Optimization Algorithm, and we explore how quantum neuron selection could be formulated to be solved with these algorithms.

Due to the vast scale of modern neural networks and the fragility of quantum systems with many qubits, full quantum neuron selection will require significant advances in both quantum hardware and software. Nevertheless, we introduce two methods for performing local quantum neuron selection that are tractable for current quantum systems. The first is a layerwise knowledge distillation algorithm which finds subnetworks in a random student subnetwork by learning to approximate the representations of a smaller trained teacher network. This enables applications of quantum neuron selection to neual networks of arbitrary depth. The second is an exploration of NK Echo State Networks, where we discuss how k-bounded connectivity patterns can significantly reduce the entanglement complexity required for quantum systems. This approach enables quantum neuron selection to be applied to probe filter layers of arbitrary width.

Quantum Neuron Selection is an exciting new paradigm for constructing deep neural networks. There is strong biological motivation for exploring neuron selection as a training mechanism [10] and we think this approach could potentially enable new methods for building and designing neural networks of the future.

REFERENCES

[1] Scott Aaronson. 2018. Introduction to Quantum Information Science. https://www.scottaaronson.com/qclec.pdf

[2] MD SAJID ANIS et al. 2021. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2573505

[3] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the State of Neural Network Pruning?. In *Proceeding of theMachine Learning and Systems Conference.*

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. https://doi.org/10.48550/ARXIV.2005.14165

[5] Stephen G. Brush. 1967. History of the Lenz-Ising Model. *Reviews of Modern Physics* 39, 4 (Oct. 1967), 883–893. https://doi.org/10.1103/RevModPhys.39.883

[6] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3, 9 (aug 2021), 625–644. https://doi.org/10.1038/s42254-021-00348-9

[7] Hugh Collins and Kortney Easterly. 2021. IBM Unveils Breakthrough 127-Qubit Quantum Processor. https://newsroom.ibm.com/2021-11-16-IBM-Unveils-Breakthrough-127-Qubit-Quantum-Processor

[8] Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. 2021. CoAtNet: Marrying Convolution and Attention for All Data Sizes. arXiv:2106.04803 [cs.CV]

[9] James Diffenderfer and Bhavya Kailkhura. 2021. Multi-Prize Lottery Ticket Hypothesis: Finding Accurate Binary Neural Networks by Pruning A Randomly Weighted Network. https://doi.org/10.48550/ARXIV.2103.09377

[10] Gerald M. Edelman. 1987. *Neural Darwinism : the theory of neuronal group selection.* Basic Books. http://www.worldcat.org/isbn/9780465049349

[11] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A Quantum Approximate Optimization Algorithm. https://doi.org/10.48550/ARXIV.1411.4028

[12] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. 2000. Quantum Computation by Adiabatic Evolution. arXiv:quant-ph/0001106 [quant-ph]

[13] Marco Fellous-Asiani, Jing Hao Chai, Robert S. Whitney, Alexia Auffèves, and Hui Khoon Ng. 2021. Limitations in Quantum Computing from Resource Constraints. *PRX Quantum* 2 (Nov 2021), 040335. Issue 4. https://doi.org/10.1103/PRXQuantum.2.040335

[14] Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. arXiv:1803.03635 [cs.LG]

[15] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. 2019. Linear Mode Connectivity and the Lottery Ticket Hypothesis. https://doi.org/10.48550/ARXIV.1912.05671

[16] Adam Gaier and David Ha. 2019. Weight Agnostic Neural Networks. arXiv:1906.04358 [cs.LG]

[17] Fred Glover, Gary Kochenberger, and Yu Du. 2018. A Tutorial on Formulating and Using QUBO Models. https://doi.org/10.48550/ARXIV.1811.11538

[18] Lov K. Grover. 1996. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Philadelphia, Pennsylvania, USA) *(STOC '96)*. Association for Computing Machinery, New York, NY, USA, 212–219. https://doi.org/10.1145/237814.237866

[19] Laszlo Gyongyosi and Sandor Imre. 2019. A Survey on quantum computing technology. *Computer Science Review* 31 (2019), 51–71. https://doi.org/10.1016/j.cosrev.2018.11.002

[20] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor Rieffel, Davide Venturelli, and Rupak Biswas. 2019. From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. *Algorithms* 12, 2 (feb 2019), 34. https://doi.org/10.3390/a12020034

[21] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. https://doi.org/10.48550/ARXIV.1503.02531

[22] Herbert Jaeger and Harald Haas. 2004. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science* 304, 5667 (2004), 78–80. https://doi.org/10.1126/science.1091277 arXiv:https://www.science.org/doi/pdf/10.1126/science.1091277

[23] Yann LeCun, John Denker, and Sara Solla. 1989. Optimal Brain Damage. In *Advances in Neural Information Processing Systems*, D. Touretzky (Ed.), Vol. 2. Morgan-Kaufmann. https://proceedings.neurips.cc/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf

[24] Junde Li, Mahabubul Alam, and Swaroop Ghosh. 2021. Large-scale Quantum Approximate Optimization via Divide-and-Conquer. https://doi.org/10.48550/

ARXIV.2102.13288

[25] Eran Malach, Gilad Yehudai, Shai Shalev-Shwartz, and Ohad Shamir. 2020. Proving the Lottery Ticket Hypothesis: Pruning is All You Need. https://doi.org/10.48550/ARXIV.2002.00585

[26] Catherine McGeoch and Pau Farré. 2022. Advantage Processor Overview. https://www.dwavesys.com/media/3xvdipcn/14-1048a-a_advantage_processor_overview.pdf

[27] Laurent Orseau, Marcus Hutter, and Omar Rivasplata. 2020. Logarithmic Pruning is All You Need. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 2925–2934. https://proceedings.neurips.cc/paper/2020/file/1e9491470749d5b0e361ce4f0b24d037-Paper.pdf

[28] Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris Papailiopoulos. 2020. Optimal Lottery Tickets via SubsetSum: Logarithmic Over-Parameterization is Sufficient. https://doi.org/10.48550/ARXIV.2006.07990

[29] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications* 5, 1 (jul 2014). https://doi.org/10.1038/ncomms5213

[30] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. 2020. What's Hidden in a Randomly Weighted Neural Network?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[31] Michele Sasdelli and Tat-Jun Chin. 2021. Quantum Annealing Formulation for Binary Neural Networks. https://doi.org/10.48550/ARXIV.2107.02751

[32] Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. 2017. Failures of Gradient-Based Deep Learning. https://doi.org/10.48550/ARXIV.1703.07950

[33] Michael Streif and Martin Leib. 2019. Comparison of QAOA with Quantum and Simulated Annealing. https://doi.org/10.48550/ARXIV.1901.01903

[34] Darrell Whitley, Renato Tinós, and Francisco Chicano. 2015. Optimal Neuron Selection: NK Echo State Networks for Reinforcement Learning. https://doi.org/10.48550/ARXIV.1505.01887

[35] Mitchell Wortsman, Ali Farhadi, and Mohammad Rastegari. 2019. Discovering Neural Wirings. https://doi.org/10.48550/ARXIV.1906.00586

[36] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. 2021. Scaling Vision Transformers. arXiv:2106.04560 [cs.CV]

[37] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. 2019. Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask. https://doi.org/10.48550/ARXIV.1905.01067