

Canonical quantization of neurons

Alexander He, Nana Liu, and Mark M. Wilde

Full version available at [arXiv:2605.24386](https://arxiv.org/abs/2605.24386)

Brief summary—Canonical quantization provides a systematic procedure for constructing quantum models from classical Hamiltonians. Here, we apply this principle to a fundamental computational primitive of machine learning: the neuron. Specifically, by viewing a neuron as a composition of an energy function and an activation function, we quantize this model by replacing the energy function with a quantum Hamiltonian and applying the activation function to it through matrix functional calculus. This results in an activation observable that can be measured on an input quantum state. We investigate the use of these quantized neurons for function approximation, where the objective is to learn an unknown observable from labeled quantum data. For this purpose, we develop hybrid quantum–classical algorithms for training and evaluation, including procedures for measuring the activation observable and estimating gradients of the squared loss error. Our algorithms for gradient estimation rely on basic primitives like classical random sampling, the Hadamard test, and Hamiltonian simulation, and those for measuring an activation observable rely on quantum algorithms known as the power of one qumode and Schrödingerization. Numerical experiments demonstrate that our quantized neurons exhibit enhanced expressive capabilities relative to corresponding classical neurons on representative learning tasks. Our work establishes canonical quantization as a principled framework for constructing quantum machine learning primitives and provides a foundation for developing neural architectures tailored to quantum data.

Introduction—A central principle underlying the development of quantum mechanics is canonical quantization [1], which promotes classical variables to operators acting on a Hilbert space. For example, a classical Hamiltonian $H(x, p)$ is transformed into a quantum Hamiltonian $H(\hat{x}, \hat{p})$, where the canonical position and momentum variables x and p , respectively, are promoted to noncommuting operators satisfying $[\hat{x}, \hat{p}] = i\hbar$. This procedure provides a systematic route for constructing quantum models from classical descriptions.

Here we apply canonical quantization to the fundamental computational primitive of machine learning: the neuron. Specifically, we interpret the energy function underlying a classical neuron as a parameterized Hamiltonian, and we quantize this Hamiltonian by replacing its classical variables with noncommuting operators, yielding H_θ . The activation function φ is then promoted to a matrix function of this Hamiltonian, producing an activation observable $\varphi(H_\theta)$. For example, as identified in [2], choosing $\varphi(H_\theta) = (e^{-H_\theta/T} + I)^{-1}$, with $T > 0$ a temperature parameter, gives a quantum generalization of the sigmoid function used in classical neurons. Whereas quantum Boltzmann machines encode learning models in quantum states through the Gibbs map $H_\theta \mapsto e^{-H_\theta} / \text{Tr}[e^{-H_\theta}]$, our approach encodes neural transformations directly as quantum observables. Thus, quantized neurons provide a complementary paradigm in which the nonlinear processing element of machine learning, rather than the underlying probability distribution, is quantized.

Our key contributions include our approach for quantizing neurons, hybrid quantum–classical (HQC) algorithms for evaluating and training them, and numerical experiments demonstrating learning tasks in which quantized neurons exhibit behavior outperforming corresponding classical models on representative learning tasks. Interestingly, our HQC algorithms for training make use of basic primitives that include classical random sampling, the Hadamard test [3], and Hamiltonian simulation [4, 5], similar to recent algorithms for training quantum Boltzmann machines [6, 7].

Canonical quantization of neurons—Let us begin by recalling the perceptron model [8] of a classical neuron. For $n \in \mathbb{N}$, suppose that $z := (z_1, \dots, z_n) \in \{-1, 1\}^n$ is a vector of input spin variables. A classical neuron consists of a composition $\varphi(w^T z + b)$ of an energy function $w^T z + b$

and an activation function φ , where $w \in \mathbb{R}^n$ is a weight vector and $b \in \mathbb{R}$ is a bias.

Before quantizing the classical neuron model, let us first rewrite the energy function as follows: $w^T z + b = \sum_{i=1}^n w_i z_i + b$. Now following canonical quantization, we promote each classical variable z_i to an operator $\hat{z}_i$. Given that each input variable z_i is a classical spin variable taking values in $\{-1, 1\}$ by assumption, each operator $\hat{z}_i$ should be Hermitian with eigenvalues -1 and $+1$. As such, a natural choice for quantizing z_i is to set $\hat{z}_i = \sigma_Z^{(i)}$, where $\sigma_Z^{(i)}$ is the Pauli Z matrix acting on the i th qubit (i.e., $\sigma_Z^{(i)} := I^{\otimes i-1} \otimes \sigma_Z \otimes I^{\otimes n-i}$). In doing so, the classical energy function above is promoted to the Hamiltonian $H_C(\theta) := \sum_{i=1}^n w_i \sigma_Z^{(i)} + b I^{\otimes n}$, where $\theta \equiv (w, b)$. Now applying the activation function φ to $H_C(\theta)$, we arrive at the activation observable $\varphi(H_C(\theta))$. This observable can be measured on an input state ρ , leading to an outcome with expected value $\text{Tr}[\varphi(H_C(\theta))\rho]$.

Let $|z\rangle \equiv |z_1\rangle \otimes \cdots \otimes |z_n\rangle$ be a tensor product of ± 1 -eigenstates of σ_Z . One finds that $H_C(\theta)|z\rangle = (w^T z + b)|z\rangle$, so that $|z\rangle$ is an eigenvector of $H_C(\theta)$ with corresponding eigenvalue $w^T z + b$. By applying the functional calculus, we then conclude that the activation observable $\varphi(H_C(\theta))$ admits the spectral decomposition $\varphi(H_C(\theta)) = \sum_{z \in \{-1, 1\}^n} \varphi(w^T z + b) |z\rangle\langle z|$. Evaluating the expected value of $\varphi(H_C(\theta))$ with respect to an input state ρ then leads to $\text{Tr}[\varphi(H_C(\theta))\rho] = \sum_{z \in \{-1, 1\}^n} p(z) \varphi(w^T z + b)$, where $p(z) \equiv \langle z|\rho|z\rangle$ is the probability distribution that results from measuring ρ in the orthonormal basis $\{|z\rangle\}_{z \in \{-1, 1\}^n}$. As such, our quantization of the classical neuron model reduces exactly to the classical case, a consistency requirement that should hold for any plausible quantization of a neuron.

We can generalize the approach above to allow for non-classical neurons by taking the underlying Hamiltonian to be a quantum Hamiltonian of the following form:

$$\sum_{\alpha, \beta, i, j} \Omega_{(\alpha, i), (\beta, j)} \sigma_\alpha^{(i)} \otimes \sigma_\beta^{(j)} + \sum_{\alpha, i} \omega_{\alpha, i} \sigma_\alpha^{(i)} + b I^{\otimes n}, \quad (1)$$

where the sums over α, β are over $\{x, y, z\}$ and those over i, j are over $[n]$. Also, $\theta := (\Omega, \omega, b)$, $\Omega \in \mathbb{R}^{3n \times 3n}$, $\omega \in \mathbb{R}^{3n}$, and the interacting and noninteracting terms include all Pauli- X , Y , and Z observables indexed by x , y , and z . The Hamiltonian in (1) includes non-commuting and interaction terms, hallmarks of Hamiltonians like the transverse-field Ising [9, Chap. 5] and Heisenberg models [10, 11] that play key roles in quantum physics. We can take an even broader approach by allowing for all Hamiltonians of the form $H_\theta := \sum_{j \in [J]} \theta_j H_j$, where $\theta_j \in \mathbb{R}$ and H_j is a Hamiltonian operator acting nontrivially on k of the n qubits, where k is a constant independent of n . A distinguishing feature of such quantum Hamiltonians is that they are generally computationally difficult to diagonalize classically; however, quantum algorithms can access spectral information through procedures related to the quantum phase estimation algorithm [12].

Representative quantized neuron—A concrete example consists of setting φ to be the temperature-scaled hyperbolic tangent activation function, abbreviated as $g_T(x) := \tanh(x/T)$, where $T > 0$ is a temperature parameter that controls the sharpness of the threshold function. In doing so, we arrive at the activation observable $g_T(H(\theta))$, where $H(\theta) = \sum_{j \in [J]} \theta_j H_j$ is a parameterized Hamiltonian.

Function approximation—The goal of this task is to approximate an unknown observable O through its expectation values on a set of input states. We suppose that quantum training data is available in the form $(\rho_1, y_1), \dots, (\rho_M, y_M)$, where ρ_m is a quantum state and $y_m \in [-1, 1]$ for all $m \in [M]$. Each pair satisfies $\text{Tr}[O\rho_m] = y_m$, and the goal is to learn this mapping. We can then pick a general parameterized Hamiltonian $H_\theta = \sum_{j \in [J]} \theta_j H_j$ and use $g_T(H(\theta))$ as an approximation of O , with the goal being to minimize the loss function $\mathcal{L}(\theta) := \frac{1}{M} \sum_{m=1}^M (\text{Tr}[g_T(H(\theta))\rho_m] - y_m)^2$. Employing the shorthand $\partial_j \equiv \frac{\partial}{\partial \theta_j}$, the j th element $\partial_j \mathcal{L}(\theta)$ of the gradient $\nabla_\theta \mathcal{L}(\theta)$ is given by $\frac{2}{M} \sum_{m=1}^M (\text{Tr}[g_T(H(\theta))\rho_m] - y_m) \partial_j \text{Tr}[g_T(H(\theta))\rho_m]$. Setting $s \in \mathbb{N}$ and $\eta > 0$ a hyperparameter known as the learning rate, we can iterate the gradient-descent update rule $\theta^{(s+1)} \leftarrow \theta^{(s)} -$

$\eta \nabla_{\theta} \mathcal{L}(\theta)|_{\theta=\theta(s)}$ in order to find a local minimum of $\mathcal{L}(\theta)$. If ρ_m is a quantum state to which we have sample access and $H(\theta)$ is a parameterized local Hamiltonian, then evaluating $\text{Tr}[g_T(H(\theta))\rho_m]$ and $\partial_j \text{Tr}[g_T(H(\theta))\rho_m]$ is generally computationally intractable for classical methods, and one can instead employ a quantum computer to estimate these quantities, as delineated in what follows.

Hybrid quantum-classical algorithms for gradient estimation—Given the application mentioned above, it is important to have a method for estimating $\partial_j \text{Tr}[g_T(H(\theta))\rho]$ for all $j \in [J]$ and for every state ρ . Consider the following integral representation: $g_T(x) = \mathbb{E}_{t \sim \mu} \left[\frac{e^{itx/T}}{it} \right]$, where $\mu(t) := t/[2 \sinh(\pi t/2)]$ is a probability density function. This representation expresses the nonlinear activation function g_T as an average over phases. By employing it and Duhamel’s formula for the derivative of the matrix exponential, we find that

$$\partial_j \text{Tr}[g_T(H(\theta))\rho] = \frac{1}{T} \mathbb{E}_{t \sim \mu, s \sim \nu} \left[\text{Tr} \left[H_j e^{itH(\theta)/T} \mathcal{U}_{st/T}^{H(\theta)}(\rho) \right] \right], \quad (2)$$

where ν is a uniform probability density over $[0, 1]$ and $\mathcal{U}_{st/T}^{H(\theta)}(\rho) := e^{-istH(\theta)/T} \rho e^{istH(\theta)/T}$ is a unitary quantum channel. Recognizing that (2) takes a form similar to what the standard Hadamard test quantum circuit estimates, we can repeatedly perform the following procedure and calculate a sample average at the end, in order to estimate $\partial_j \text{Tr}[g_T(H(\theta))\rho]$: 1) sample $t \sim \mu$ and $s \sim \nu$, 2) execute the Hadamard test quantum circuit depicted in [13, Fig. 2(a)], and 3) multiply the measurement outcomes to obtain a random outcome with expected value equal to the last line in (2). By the law of large numbers (in particular, the Hoeffding inequality [14]), the resulting sample average is an unbiased estimate of (2) that converges quickly to it (see [13, Eq. (29)]).

Quantum algorithm for measuring an activation observable—The other key ingredient needed for training and testing is a quantum algorithm for measuring the activation observable $g_T(H(\theta))$. We can employ a method that builds on primitives known as the power of one qumode [15] and Schrödingerization [16, 17]. Here we employ a control qumode that interacts with the n qubits of the quantized neuron. The method begins by preparing a control qumode in the state $|\ell_{T_1}\rangle := \int_{\mathbb{R}} dp \sqrt{\ell_{T_1}(p)} |p\rangle$, where ℓ_{T_1} is the logistic probability density function $\ell_{T_1}(p) := \frac{1}{4T_1} \text{sech}^2\left(\frac{p}{2T_1}\right)$, $\{|p\rangle\}_{p \in \mathbb{R}}$ denotes the momentum quadrature eigenbasis, and $T_1, T_2 > 0$ are chosen such that $T/2 = T_1 T_2$. Then, prepare a data register of n qubits in the state ρ , interact the control qumode and the data register according to the Hamiltonian evolution $e^{i\hat{x} \otimes H(\theta)/T_2}$, and measure the momentum quadrature of the control qumode to obtain the outcome $p \in \mathbb{R}$. Finally, set the output $Y = 1$ if $p \geq 0$ and $Y = -1$ if $p < 0$. The expected value of Y is equal to $\text{Tr}[g_T(H(\theta))\rho]$, as stated in [13, Thm. 4], thus providing an effective procedure for measuring $g_T(H(\theta))$.

Outlook—Our paper [13] includes several results not mentioned here, including logistic loss minimization for binary classification of quantum data, extensions to quantized neurons for qumodes and continuous-variable quantum data, a BQP-complete computational decision problem based on our quantized neurons, and, finally, two methods for composing these quantized neurons into networks. One practical method for forming networks involves a first layer of quantized neurons for processing quantum data, while subsequent layers are purely classical.

Several directions remain open. An important question is whether quantized neurons exhibit barren-plateau behavior [18]. Existing analyses of barren plateaus, which primarily focus on parameterized quantum circuits, do not directly apply to this setting, motivating a separate investigation of trainability. Next, extending numerical simulations to larger system sizes and experimentally relevant quantum data sets is an important direction. Finally, it would be interesting to investigate the aforementioned hybrid neural network architectures in which the first layer consists of quantized neurons, while subsequent layers are classical. Such architectures combine quantum processing of quantum data with the flexibility of classical machine learning methods.

-
- [1] R. Shankar, *Principles of Quantum Mechanics* (Springer, 1994).
 - [2] N. Liu and M. M. Wilde, [Fermi–Dirac thermal measurements: A framework for quantum hypothesis testing and semidefinite optimization](#) (2026), [arXiv:2603.04061v1 \[quant-ph\]](#).
 - [3] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Quantum algorithms revisited, *Proceedings of the Royal Society A* **454**, 339 (1998).
 - [4] S. Lloyd, Universal quantum simulators, *Science* **273**, 1073 (1996).
 - [5] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, *Proceedings of the National Academy of Sciences* **115**, 9456 (2018).
 - [6] D. Patel, D. Koch, S. Patel, and M. M. Wilde, [Quantum Boltzmann machine learning of ground-state energies](#) (2025), [arXiv:2410.12935 \[quant-ph\]](#).
 - [7] D. Patel and M. M. Wilde, Natural gradient and parameter estimation for quantum Boltzmann machines, *Physical Review A* **112**, 052421 (2025).
 - [8] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review* **65**, 386 (1958).
 - [9] S. Sachdev, *Quantum Phase Transitions*, 2nd ed. (Cambridge University Press, 2011).
 - [10] D. C. Mattis, *The Theory of Magnetism Made Simple* (World Scientific, 2006).
 - [11] C. Goldschmidt, D. Ueltschi, and P. Windridge, Entropy and the quantum II, *Contemporary Mathematics* (2011) Chap. Quantum Heisenberg models and their probabilistic representations, pp. 177–224, [arXiv:1104.0983 \[math-ph\]](#).
 - [12] A. Y. Kitaev, [Quantum measurements and the abelian stabilizer problem](#) (1995), [arXiv:quant-ph/9511026 \[quant-ph\]](#).
 - [13] A. He, N. Liu, and M. M. Wilde, [Fermi-Dirac machines as quantizations of neurons](#) (2026), [arXiv:2605.24386v1 \[quant-ph\]](#).
 - [14] W. Hoeffding, Probability inequalities for sums of bounded random variables, *Journal of the American Statistical Association* **58**, 13 (1963).
 - [15] N. Liu, J. Thompson, C. Weedbrook, S. Lloyd, V. Vedral, M. Gu, and K. Modi, Power of one qumode for quantum computation, *Physical Review A* **93**, 052304 (2016).
 - [16] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations: Applications and detailed analysis, *Physical Review A* **108**, 032603 (2023).
 - [17] S. Jin, N. Liu, and Y. Yu, Quantum simulation of partial differential equations via Schrödingerization, *Physical Review Letters* **133**, 230602 (2024).
 - [18] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, Barren plateaus in variational quantum computing, *Nature Reviews Physics* **7**, 174 (2025).