
Classical Quantum Optimization with Neural Network Quantum States

Joseph Gomes
University of Iowa
Iowa City, IA 52246
joe-gomes@uiowa.edu

Keri A. McKiernan
Stanford University
Stanford, CA 94305
kmckiern@stanford.edu

Peter Eastman
Stanford University
Stanford, CA 94305
peastman@stanford.edu

Vijay S. Pande
Stanford University
Stanford, CA 94305
pande@stanford.edu

Abstract

The classical simulation of quantum systems typically requires exponential resources. Recently, the introduction of a machine learning-based wavefunction ansatz has led to the ability to solve the quantum many-body problem in regimes that had previously been intractable for existing exact numerical methods. Here, we demonstrate the utility of the variational representation of quantum states based on artificial neural networks for performing quantum optimization. We show empirically that this methodology achieves high approximation ratio solutions with polynomial classical computing resources for a range of instances of the Maximum Cut (MaxCut) problem whose solutions have been encoded into the ground state of quantum many-body systems up to and including 256 qubits.

1 Introduction

The key difficulty in classical quantum simulation is the exponential growth of the Hilbert space dimension with system size, leading to an intractable number of parameters needed for an exact representation of the wavefunction of modest sized many-body systems. The exponential growth can be partially avoided through the use of approximate methods such as stochastic sampling approaches (Quantum Monte Carlo) [1] or compact representations of the quantum state (tensor network states) [2–4]. The neural-network quantum state (NQS) approach proposed by Carleo and Troyer [5] demonstrates the ability of the Restricted Boltzmann Machine (RBM) to compactly represent high dimensional wavefunctions.

In contrast to previous tensor network based methods, the NQS approach has been shown to capture longer range entanglement structures leading to highly accurate representations of quantum systems. Since the introduction of NQS, there have been numerous applications in condensed matter physics [6] and strongly correlated electron systems [7], fermionic electronic structure simulation [8], simulation of general quantum circuits [9], quantum state tomography [10], as well as advances in our understanding of the entanglement properties of the neural network wavefunction ansatz [11–14]. For a comprehensive review on applications of NQS, see Ref. [15].

One area of quantum simulation where the NQS approach has yet to be applied is quantum optimization algorithms. Challenging computational problems are frequently tackled by classical heuristic algorithms chosen based on empirical performance evaluation [16]. Optimization problems can also be solved using quantum heuristic algorithms, which typically involves formulating the problem of interest into a "cost Hamiltonian" [17, 18]. The goal is then to identify the solution, or state, yielding

the lowest cost. Universal gate quantum algorithms for optimization commonly minimize this cost Hamiltonian through alternating applications of problem-specific cost and driver Hamiltonians by a series of quantum logic gates [19, 20]. Adiabatic quantum optimization algorithms minimize this cost Hamiltonian by reversible, adiabatic quantum state evolution between a Hamiltonian whose ground state is easy to prepare and the cost Hamiltonian [21].

In this work, we demonstrate the utility of the NQS ansatz to obtain the ground state of some quantum many-body systems relevant to quantum optimization. In contrast to universal gate or adiabatic quantum optimization algorithms, we obtain a variational estimate of the solution through direct optimization and subsequent sampling of the NQS wavefunction given a problem-specific cost Hamiltonian. We show empirically that this approach achieves high accuracy solutions with numerically observed polynomial time and memory complexity for a variety of instances of the MaxCut combinatorial optimization problem whose solutions have been encoded into the ground state of a quantum many-body system. The machine learning approach to many-body quantum states allows for approximate simulations of quantum optimization beyond what can be performed exactly on classical computing resources or on current quantum computing resources.

2 Methods

2.1 Neural Network Quantum States

We consider a system made of L spin-1/2 degrees of freedom denoted as $\sigma_j = \pm 1, j = 1, \dots, L$. The amplitude of the (complex) RBM wavefunction is represented by a sum of exponentials,

$$\Psi(\sigma) = \sum_{\mathbf{h}} e^{\sum_j a_j \sigma_j + \sum_i b_i h_i + \sum_{ij} h_i W_{ij} \sigma_j} \quad (2.1)$$

where the sum runs over all $\mathbf{h} = (h_1, h_2, \dots, h_M)$ with the binary variables $h_i \in \{-1, 1\}$ for $i = 1, \dots, M$. By performing the summation over the hidden variables \mathbf{h} , Eq. 2.1 reduces to $\log \Psi(\sigma) = \sum_j a_j \sigma_j + \sum_i \log[\cosh(b_i + \sum_j W_{ij} \sigma_j)]$ up to some additive factor which corresponds to the normalization constant and phase factor of the wavefunction.

2.2 Maximum Cut

Given an undirected graph $G = (V, E)$ and non-negative weights $w_{ij} = w_{ji}$ on the edges $(i, j) \in E$, the MaxCut problem seeks to find a bi-partitioning of this graph, such that the sets of vertices S and \bar{S} maximizes the weight of edges in the cut (S, \bar{S}) . In the case of MaxCut, the cost Hamiltonian minimized is of the form:

$$\hat{H} = -\frac{1}{4} \sum_{(i,j) \in E} L_{ij} \sigma_i^z \sigma_j^z, \quad (2.2)$$

where L is the graph Laplacian and σ^z is the Pauli z operator. Note that this problem has a set of solutions given by 2^N N-bit bitstrings. The MaxCut combinatorial optimization problem is known to be NP-hard [22].

2.3 Classical Quantum Optimization

The energy of a wavefunction $\Psi(\sigma)$ can be estimated as

$$\langle \hat{H} \rangle \approx \left\langle \sum_{\sigma'} \langle \sigma | \hat{H} | \sigma' \rangle \frac{\Psi(\sigma')}{\Psi(\sigma)} \right\rangle_{\sigma} \quad (2.3)$$

where $\langle \cdot \rangle_{\sigma}$ denotes a stochastic expectation value taken over a sample of configurations $\{\sigma\}$ drawn from the probability distribution corresponding to the variational wavefunction and the summation runs over all configurations σ' where the matrix element $\langle \sigma | \hat{H} | \sigma' \rangle$ is nonzero. Given a variational ansatz $\Psi(\{\alpha_k\})$ parameterized by α_k for $k = 1, \dots, N_{var}$, and target Hamiltonian \hat{H} (Eqn. 2.2), we optimize using the stochastic reconfiguration (SR) method introduced by Sorella [23]. The SR optimization method can be seen as an effective imaginary-time evolution in the variational subspace.

In this view, the optimization procedure may be interpreted as an imaginary-time evolution from a randomly initialized quantum state to one that minimizes the problem-specific cost Hamiltonian given a variational ansatz.

After optimization, we draw a set of $\max(1000, 10 * N_{var})$ sample configurations and from this set determine the bitstring that minimizes the cost Hamiltonian.

2.4 Datasets

We have created a dataset containing 90 instances of fixed edge density ($\rho = 12\%, 25\%, 50\%$), random graphs of increasing size ($|V| = 8, 16, 32, 64, 128, 256$) using `rudy`, a machine independent graph generator written by G. Rinaldi. Reference solutions of random graphs containing less than 256 vertices were generated by `BiqCrunch` [24], a semidefinite branch-and-bound method for solving binary quadratic problems. Due to size limitation of exact branch-and-bound solvers, the BURER2002 algorithm [25], a heuristic based on non-linear optimization with local search, was used, as implemented in `MQlib` [16], to estimate solutions of random graphs containing 256 vertices.

2.5 Implementation Details

All calculations were performed with the free, open source software package `Netket` [26]. The following hyperparameters were used: number of hidden variables ($M = L$ number of spins), learning rate ($\epsilon=0.05$), standard deviation of random initial NQS weights ($\sigma_{sd}=0.01$), optimization steps ($n_{opt}=200$), Monte Carlo samples ($N_{MC}=N_{var}$), SR regularization ($\lambda=0.1$). Timing benchmarks of the NQS method were obtained by averaging the wall clock time of 10 consecutive optimization steps across all MaxCut instances of a given edge density and number of vertices, except in the case of graphs containing 256 vertices where two instances of a given edge density were used. Timing benchmarks were performed on a single core of a 2.60 GHz 32-core Intel Xeon E5-2650 processor equipped with 256 GB of memory. We note that the hyperparameters used in this study were not optimized for wall clock time. The code and datasets used in this study are available at <http://www.github.com/joegomes/cqo>.

3 Results

Here we apply NQS method to a dataset of MaxCut instances described in Sec. 2.4. We first examine performance, with respect to an optimal ordering of solutions, through the progression of the training process (Fig. 1). Then, we examine quality and time benchmarks across the full `rudy` dataset (Fig. 2). For all instances studied here, ground state configurations of NQS wavefunctions were found to generate optimal or nearly optimal rewards.

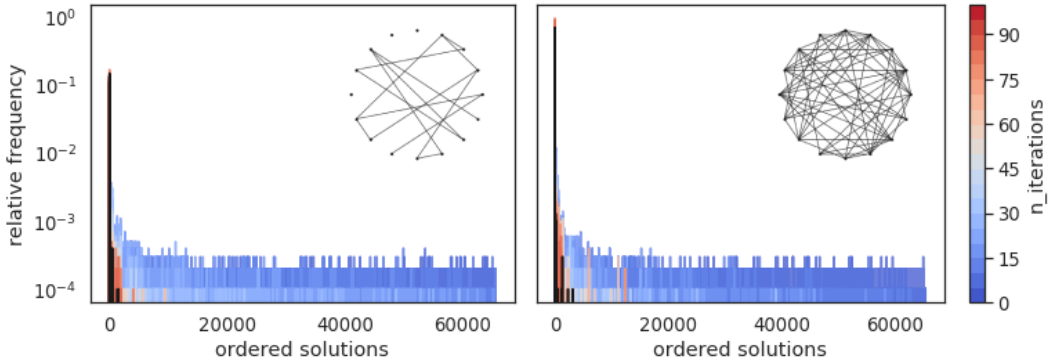


Figure 1: Solving the MaxCut problem via NQS for both sparse (left) and dense (right) 16-node `rudy` graphs. Each example graph instance is displayed as an inset. The colored distributions illustrate evolution of the reward distribution over the course of the training process. It can be seen NQS optimization procedure gradually increases the probability of sampling the best-quality solutions (indexed 0 and onward), while gradually reducing the probability of sampling the worst-quality solution.

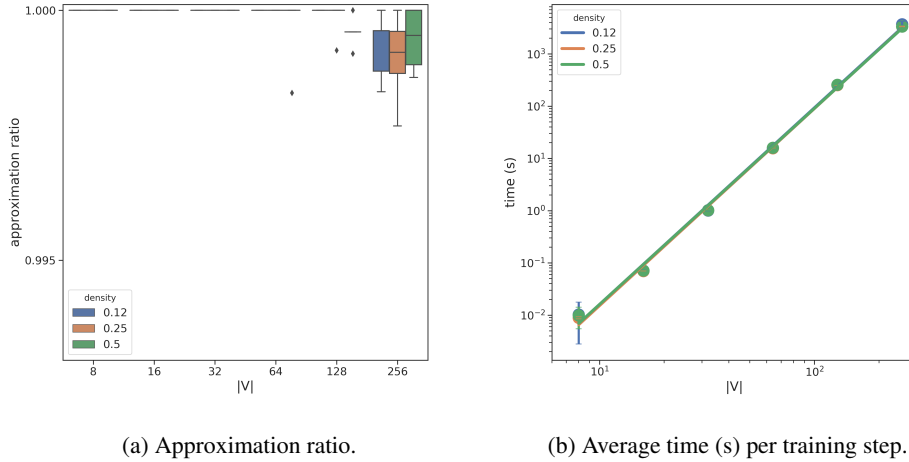


Figure 2: Approximation ratio and timing benchmarks for the NQS method on `rudy` dataset.

Fig. 1 illustrates application of the NQS method to solve the MaxCut problem on two 16-node instances. Following optimization, for a given number of training iterations (specified by the distribution color), 10,000 samples were drawn from the probability distribution corresponding to the NQS wavefunction. These samples were then used to compute an empirical probability distribution over all possible solutions. The optimal ordering of solutions was determined through exhaustive calculation of solution rewards. The final distribution of the optimized NQS wavefunction is displayed in black.

It can be seen that at the beginning of training, the NQS wavefunction yields a nearly uniform reward distribution (iteration < 20 , dark blue distributions). As training progresses, the probability of sampling high-quality solutions increases (red distributions). For both graph instances, the optimal solution is indeed the most likely solution (index 0, largest peak of the black distribution). In both cases, there remains residual probability of non-optimal solutions. However, note that although not the optimal solution, these solutions are relatively high quality solutions.

Determining the optimality of a solution to a given MaxCut problem requires exponential resources, but an (nearly) optimal solution can be obtained with classical heuristic algorithms [16]. In Fig. 2a, we compare the solutions obtained by NQS with those from the classical heuristic algorithms described in Sec. 2.4 by means of the approximation ratio, the multiplicative factor relating the returned and estimated optimal reward. We observed satisfactory performance of the NQS method with an approximation ratio > 0.995 across the set of problems studied without a strong dependence on edge density, although we do not expect this trend to hold in general. We observed that the classical heuristic algorithm BURER2002 outperforms classical quantum optimization in approximation ratio, time to solution, and efficiency.

Based on the complexity analysis presented in Ref. [5], we expect a time complexity between $\mathcal{O}(N_{var} * N_{MC})$ and $\mathcal{O}(N_{var}^2 * N_{MC})$ (or $\mathcal{O}(N^4)$ and $\mathcal{O}(N^6)$) for quadratic cost Hamiltonian optimization and observe approximately $\mathcal{O}(N^4)$ (Fig. 2b) scaling over the size of systems studied here without a noticeable dependence on graph edge density. The memory complexity for NQS Hamiltonian storage and wavefunction representation (N_{var}) grows quadratically with system size when the number of hidden RBM units is related linearly to the number of visible RBM units.

4 Outlook

We have demonstrated the application of the neural-network quantum state approach toward solving quantum optimization problems. We show for the set of instances studied here that this approach achieves accurate classical quantum simulation using polynomial time and memory resources for a variety of MaxCut combinatorial optimization problems, on systems up to 256 qubits in size. The construction of a formal proof determining theoretical complexity and performance guarantee remains an open question.

Acknowledgements

The Pande Group acknowledges the generous support of Dr. Anders G. Frøseth and Mr. Christian Sundt for our work on machine learning. The Pande Group is broadly supported by grants from the NIH (R01 GM062868 and U19 AI109662) as well as gift funds and contributions from Folding@home donors. J.G acknowledges start up funding from the University of Iowa. This research was supported in part through computational resources provided by The University of Iowa.

V.S.P is a consultant & SAB member of Schrodinger, LLC and Globavir, sits on the Board of Directors of Apeel Inc, Asimov Inc, BioAge Labs, Ciitizen, Devoted Health, Freenome Inc, Omada Health, Patient Ping, Rigetti Computing, and is a General Partner at Andreessen Horowitz.

References

- [1] Federico Becca and Sandro Sorella. *Quantum Monte Carlo approaches for correlated systems*. Cambridge University Press, 2017.
- [2] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.
- [3] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Physical review letters*, 93(4):040502, 2004.
- [4] Frank Verstraete and J Ignacio Cirac. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv preprint cond-mat/0407066*, 2004.
- [5] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.
- [6] Giuseppe Carleo, Yusuke Nomura, and Masatoshi Imada. Constructing exact representations of quantum many-body systems with deep neural networks. *Nature communications*, 9(1):5322, 2018.
- [7] Yusuke Nomura, Andrew S Darmawan, Youhei Yamaji, and Masatoshi Imada. Restricted boltzmann machine learning for solving strongly correlated quantum systems. *Physical Review B*, 96(20):205152, 2017.
- [8] Kenny Choo, Antonio Mezzacapo, and Giuseppe Carleo. Fermionic neural-network states for ab-initio electronic structure. *arXiv preprint arXiv:1909.12852*, 2019.
- [9] Bjarni Jónsson, Bela Bauer, and Giuseppe Carleo. Neural-network states for the classical simulation of quantum computing. *arXiv preprint arXiv:1808.05232*, 2018.
- [10] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447, 2018.
- [11] Dong-Ling Deng, Xiaopeng Li, and S Das Sarma. Quantum entanglement in neural network states. *Physical Review X*, 7(2):021021, 2017.
- [12] Xun Gao and Lu-Ming Duan. Efficient representation of quantum many-body states with deep neural networks. *Nature communications*, 8(1):662, 2017.
- [13] Jing Chen, Song Cheng, Haidong Xie, Lei Wang, and Tao Xiang. Equivalence of restricted boltzmann machines and tensor network states. *Physical Review B*, 97(8):085104, 2018.
- [14] Stephen R Clark. Unifying neural-network quantum states and correlator product states via tensor networks. *Journal of Physics A: Mathematical and Theoretical*, 51(13):135301, 2018.
- [15] Roger G Melko, Giuseppe Carleo, Juan Carrasquilla, and J Ignacio Cirac. Restricted boltzmann machines in quantum physics. *Nature Physics*, page 1, 2019.
- [16] Iain Dunning, Swati Gupta, and John Silberholz. What works best when? a systematic evaluation of heuristics for max-cut and qubo. *INFORMS Journal on Computing*, 30(3):608–624, 2018.

- [17] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.
- [18] Stuart Hadfield. On the representation of boolean and real functions as hamiltonians for quantum computing. *arXiv preprint arXiv:1804.09130*, 2018.
- [19] Aniruddha Bapat and Stephen Jordan. Bang-bang control as a design principle for classical and quantum optimization algorithms. *arXiv preprint arXiv:1812.02746*, 2018.
- [20] Stuart Hadfield, Zhihui Wang, Bryan O’Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019.
- [21] Tameem Albash and Daniel A Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002, 2018.
- [22] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
- [23] Sandro Sorella. Green function monte carlo with stochastic reconfiguration. *Physical review letters*, 80(20):4558, 1998.
- [24] Christoph Helmberg and Franz Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.
- [25] Samuel Burer, Renato DC Monteiro, and Yin Zhang. Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, 12(2):503–521, 2002.
- [26] Giuseppe Carleo, Kenny Choo, Damian Hofmann, James E. T. Smith, Tom Westerhout, Fabien Alet, Emily J. Davis, Stavros Efthymiou, Ivan Glasser, Sheng-Hsuan Lin, Marta Mauri, Guglielmo Mazzola, Christian B. Mendl, Evert van Nieuwenburg, Ossian O’Reilly, Hugo Théveniaut, Giacomo Torlai, Filippo Vicentini, and Alexander Wietek. Netket: A machine learning toolkit for many-body quantum systems. *SoftwareX*, page 100311, 2019.