A Neural Network Approach to the Quantum Many-Body Problem

Emily Davis, Kevin A. Fischer, Alexander T. Hristov
Stanford University
{emilyjd,kevinf,hristov}@stanford.edu

1 Introduction

Quantum mechanics forms the theoretical framework for studying new, exotic, and potentially transformative materials, such as high temperature superconductors and topological states of matter. These materials comprise a large number of atomic systems interacting together, and hence are called many-body problems. Understanding many-body quantum systems is difficult: even their numerical solutions are intractable, owing to a host of technical challenges which may be alleviated through new computational approaches.

The underlying difficulty comes from the complexity of the large number of atomic systems involved, as well as the nature of the dynamical equations of quantum mechanics. Quantum mechanical systems are mathematically represented as elements of a Hilbert space, with a number of parameters that grows exponentially with the size of the system. Consider a chain of N interacting electron spins (binary variables), which can each point either up or down: the number of parameters potentially required to represent this simple system is 2^N. Thus, a system of just 40 spins needs 10 TB to represent each of the 2^N parameters as a complex floating point number. This specifically motivates the search for efficient representations of quantum states in which the number of parameters scales polynomially, not exponentially, with system size.

Recent applications of machine learning to this problem may help find an efficient representation of quantum states and their dynamical behavior. In particular, Restricted Boltzmann machines have been shown to represent many-body quantum states efficiently [Carleo and Troyer, 2017]. We find these results a promising direction for the field, and here provide a complete open-source python implementation of the optimization and sampling techniques necessary to train networks for the quantum many-body problem. This implementation can be used as a platform for training arbitrary network architectures. We have used our platform to verify previous state-of-the art results discussed in Carleo and Troyer [2017]. After successful training to minimize the energy of the state represented by the network, we additionally investigate statistical correlations in the resulting probability distribution.

2 Methods: Quantum mechanics as an optimization problem

Carleo [2017] outlines a theoretical formulation for the numerical simulation of quantum systems as a machine learning problem, which we summarize briefly here. The state of a quantum system can be represented by a vector ψ in a Hilbert space. Given a basis $|\mathbf{x}\rangle$ of that vector space with labels $\mathbf{x}$, the state can be expressed as

$$|\psi\rangle = \sum_{\mathbf{x}} \psi(\mathbf{x})|\mathbf{x}\rangle.$$

In this work, we take the labels $\mathbf{x}$ to be binary sequences in $\{\pm 1\}^N$. $\psi(\mathbf{x})$ can be understood as a function that provides the complex weight of a basis state, given the input label of that state. Furthermore, the square modulus of that complex weight yields a probability distribution on the set of basis vectors: $p_\psi(\mathbf{x}) = |\psi(\mathbf{x})|^2 / \left(\sum_{\mathbf{x}} |\psi(\mathbf{x})|^2\right)$.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

The interactions in any physical system are encapsulated by the Hamiltonian, a Hermitian matrix $\mathbf{H}$. The physical properties of a system can be obtained by solving the eigenvalue problem

$$\mathbf{H}\psi = \mathcal{E}\psi \tag{1}$$

for ψ and energy eigenvalue $\mathcal{E}$ pairs. Often, many of the physical properties can be determined from only the lowest energy eigenvalue $\mathcal{E}_0 \leq \mathcal{E}_i$ and its corresponding 'ground state' eigenvector ψ_0. Unfortunately, solving the eigenvalue problem directly, even for only the ground state, is intractable for all but the smallest quantum systems.

The variational Monte-Carlo method partially solves this roadblock by translating the eigenvalue problem into an optimization problem. It is known that the energy of a given state is always greater than the ground state's energy

$$\mathcal{E}[\psi] = \langle \psi, \mathbf{H}\psi \rangle \geq \mathcal{E}_0, \tag{2}$$

and thus finding the minimum energy determines the ground state. Then, through a clever algebraic transformation, the energy functional can be re-written in terms of a statistical expectation[1]:

$$\mathcal{E}[\psi] = \frac{\langle\psi|\mathbf{H}|\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\sum_{\mathbf{x},\mathbf{x}'}\langle\psi|\mathbf{x}\rangle\langle\mathbf{x}|\mathbf{H}|\mathbf{x}'\rangle\langle\mathbf{x}'|\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\sum_{\mathbf{x}}|\psi(\mathbf{x})|^2\left(\sum_{\mathbf{x}'}H_{\mathbf{x}\mathbf{x}'}\frac{\psi(\mathbf{x}')}{\psi(\mathbf{x})}\right)}{\sum_{\mathbf{x}}|\psi(\mathbf{x})|^2} \tag{3}$$

$$= E\left[\sum_{\mathbf{x}'}H_{\mathbf{x}\mathbf{x}'}\frac{\psi(\mathbf{x}')}{\psi(\mathbf{x})}\right], \tag{4}$$

where the expectation is computed over the probability distribution $p_\psi(\mathbf{x})$. The operator whose expectation value is being taken is known as the 'local-energy'

$$\mathcal{E}_{\text{loc}}(\mathbf{x}) \equiv \sum_{\mathbf{x}'}H_{\mathbf{x}\mathbf{x}'}\frac{\psi(\mathbf{x}')}{\psi(\mathbf{x})}.$$

The optimization $\psi_0 = \text{argmin}_\psi \mathcal{E}[\psi]$, yields the ground state energy and wave-function.

Given a wave function parametrized by a set of variables θ, one method for optimizing Eq. 2 is the stochastic reconfiguration method [Sorella et al., 2007]. Computing this gradient descent first requires computing the parameter derivatives

$$\mathrm{d}\psi_j(\mathbf{x};\theta) \equiv \frac{1}{\psi(\mathbf{x};\theta)}\frac{\partial\psi(\mathbf{x};\theta)}{\partial\theta_j}, \tag{5}$$

and the gradients of the cost function, which are approximated in this technique by the covariances

$$\mathrm{d}\theta_j \equiv \frac{\partial\mathcal{E}[\psi(\mathbf{x};\theta)]}{\partial\theta_j} \approx E[\mathcal{E}_{\text{loc}}(\mathbf{x})\,\mathrm{d}\psi_j(\mathbf{x};\theta)] - E[\mathcal{E}_{\text{loc}}(\mathbf{x})]E[\mathrm{d}\psi_j(\mathbf{x};\theta)]. \tag{6}$$

The gradient descent update rule is

$$\theta_i \leftarrow \theta_i - \alpha\sum_j S_{ij}\frac{\partial\mathcal{E}[\psi]}{\partial\theta_j}, \tag{7}$$

where α is the learning rate, possibly with some schedule, and $\mathbf{S}$ is the stochastic reconfiguration matrix, which plays a role similar to the $\sqrt{G_t}^{-1}$ pre-factor in the adaptive gradient method [Duchi et al., 2011]. The elements of the inverse of the stochastic reconfiguration matrix are

$$S_{ij}^{-1} = E[\mathrm{d}\psi_i^*(\mathbf{x};\theta)\,\mathrm{d}\psi_j(\mathbf{x};\theta)] - E[\mathrm{d}\psi_i^*(\mathbf{x};\theta)]E[\mathrm{d}\psi_j(\mathbf{x};\theta)]. \tag{8}$$

3 Related work

Density matrix renormalization group (DMRG) [White, 1992] is a standard numerical technique for finding quantum mechanical ground states which works well in one dimension, but scales poorly with system size in higher dimensions [Liang and Pang, 1994]. Carleo and Troyer [2017] find that

[1]Note that $\sum_{\mathbf{x}}|\mathbf{x}\rangle\langle\mathbf{x}|$ is the identity matrix and $\langle\mathbf{x}|\psi\rangle = \psi(\mathbf{x})$. The sum over $H_{\mathbf{x}\mathbf{x}'}$ is sparse which makes computing the local energy tractable.

using a restricted Boltzmann machine (RBM), generalized to complex parameters to represent the wavefunction, allows for efficient computation of ground state energies in higher dimensions—their results meet or beat standard DMRG calculations. The idea is to write

$$\psi(\mathbf{x}, \mathbf{h}; \theta) = \exp(-\mathbf{a}^{\mathrm{T}}\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{h} - \mathbf{h}^{\mathrm{T}}\mathbf{W}\mathbf{x}), \tag{9}$$

where $\mathbf{h} \in \{\pm 1\}^M$ represent M hidden units in the RBM, and the parameters are $\theta = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$. Although the biases $\mathbf{a}$ and $\mathbf{b}$ are real-valued, the weights $\mathbf{W}$ may be complex. In analogy with a standard RBM, the marginal wave-function with respect to the visible units $\mathbf{x}$ is obtained by tracing out the hidden units:

$$\psi(\mathbf{x}; \theta) \;=\; \sum_{\mathbf{h}} \psi(\mathbf{x}, \mathbf{h}; \theta) \;=\; \exp(-\mathbf{a}^{\mathrm{T}}\mathbf{x}) \prod_{i=1}^{M} 2\cosh(b_i + \sum_{j=1}^{N} W_{ij} x_j). \tag{10}$$

Furthermore, the ground state energy obtained by the RBM neural network for the 2d AFM Heisenberg model beats the energy obtained with entangled-plaquette states (EPS) [Mezzacapo et al., 2009], and projected entangled pair states (PEPS) [Lubasch et al., 2014], which have been shown to efficiently approximate ground states of local Hamiltonians [Hastings, 2007]. The RBM formulation can achieve greater accuracy than both of these for $\alpha \equiv N_{\text{hidden units}}/N_{\text{spins}} \geq 8$, although it is not obvious how the computational resources compare.

4 Dataset and Features

During training, the gradients in Eq. 7 are computed stochastically from the a set of sampled configurations $\hat{\mathbf{X}} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}\}$ drawn from $p_\psi(\mathbf{x})$ over one epoch of training [Carleo and Troyer, 2017]. The sampling is performed by well-known Markov-Chain Monte Carlo: a long sequence of samples is generated whose distribution converges to the probability distribution $p_\psi(\mathbf{x})$. This is implemented using the Metropolis–Hastings algorithm which comprises the following two steps. First, given a state $\mathbf{x}^{(k)}$ in the sequence, a candidate next state $\tilde{\mathbf{x}}^{(k)}$ is obtained by randomly flipping some number of variables in the label $\mathbf{x}^{(k)}$. Second the next state $\mathbf{x}^{(k+1)}$ in the sequence is set to either $\mathbf{x}^{(k)}$ or $\tilde{\mathbf{x}}^{(k)}$, choosing the latter with probability

$$P(\mathbf{x}^{(k)} \to \mathbf{x}^{(k+1)}) = \min\left(1, \left|\frac{\psi(\mathbf{x}^{(k+1)}; \theta)}{\psi(\mathbf{x}^{(k)}; \theta)}\right|^2\right), \tag{11}$$

which is specified by the neural network. The autocorrelation time of this Markov chain can be estimated, and sampling is carried out over a sequence much longer than the autocorrelation time to ensure convergence. This process is detailed in Figure 1.

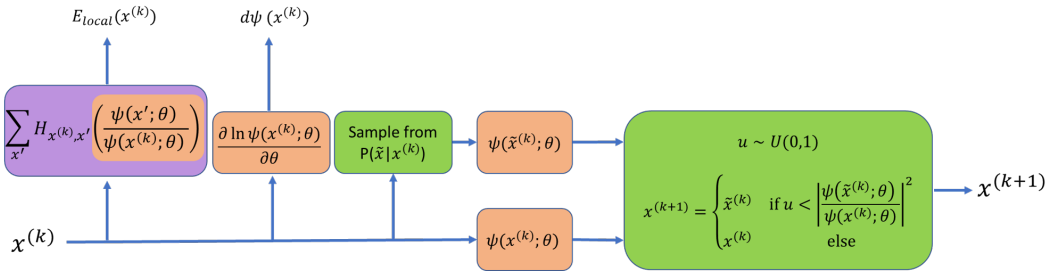

Figure 1: Schematic representing sequential process for generating samples. The above process is repeated $K = 10000$ times for each training step. All steps using the neural network are orange. A sample $\mathbf{x}^{(k)} \in \{\pm 1\}^{N_{\text{spins}}}$ is fed into the code. Its energy under the Hamiltonian is computed (purple). The derivatives of $\ln \psi(\mathbf{x}^{(k)})$ with respect to the model parameters are computed. Then, the next sample from the distribution is obtained by a two-step process from the Metropolis-Hastings algorithm. First, a candidate next state $\tilde{\mathbf{x}}^{(k)}$ is obtained by uniformly sampling sites on the lattice and flipping the binary spin variables on those sites. Second, the next state in the sequence is chosen based on the ratio of the neural network output for $\tilde{\mathbf{x}}^{(k)}$ and $\mathbf{x}^{(k)}$.

5 Methods/Approach

The goal is to optimize the parameters θ of the distribution $p_\psi(\mathbf{x})$ to minimize the energy cost functional $\mathcal{E}[\psi]$. The functional that is minimized is determined by a Hermitian matrix $H_{\mathbf{x}\mathbf{x}'}$, which is externally provided. To summarize, the algorithm is the following:

Algorithm 1 NQSlearn

procedure OPTIMIZE
  $\theta \leftarrow$ Random initialization
  while $\hat{\mathcal{E}}_0$ is not minimized **do**
    $\hat{\mathbf{X}} \leftarrow$ **MetropolisHastings**(p_ψ, K) // draw K samples from p_ψ
    $\mathrm{d}\hat{\psi} \leftarrow (K\psi(\hat{\mathbf{X}};\theta))^{-1}\partial\psi(\hat{\mathbf{X}};\theta)/\partial\theta$ // parameter derivatives
    $\mathrm{d}\hat{\theta} \leftarrow 1/K \sum_{\hat{\mathbf{X}}} \partial\mathcal{E}[\psi(\hat{\mathbf{X}};\theta)]/\partial\theta$ // gradients of cost function
    $\hat{S} \leftarrow \left(\mathrm{d}\hat{\psi}\,\mathrm{d}\hat{\psi}^\dagger - (\sum_{\hat{\mathbf{X}}} \mathrm{d}\hat{\psi})(\sum_{\hat{\mathbf{X}}} \mathrm{d}\hat{\psi})^\dagger\right)^{-1}$ // stochastic reconfiguration matrix
    $\theta \leftarrow \theta - \alpha\hat{S}\,\mathrm{d}\hat{\theta}$ // gradient descent
    $\hat{\mathcal{E}}_0 \leftarrow \sum_{\hat{\mathbf{X}}} \mathcal{E}[\psi(\hat{\mathbf{X}};\theta)]/K$ // estimate energy

6 Experiments/Results/Discussion

We have successfully replicated some of the results from Carleo and Troyer [2017]. In particular, we performed the NQSlearn optimization to find the ground state of the antiferromagnetic Heisenberg Hamiltonian and the transverse field Ising Hamiltonian:

$$H_{\text{AFH}} = \sum_{ij} \sigma_i^x\sigma_j^x + \sigma_i^y\sigma_j^y + J_z\sigma_i^z\sigma_j^z \quad \text{and} \quad H_{\text{TFI}} = \sum_{ij} \sigma_i^z\sigma_j^z + h_x \sum_i \sigma_i^z,$$

respectively, where i, j are nearest neighbors.

An example learning curve for training on the 1D transverse field Ising Hamiltonian is shown in Figure 2a. The network approximates the ground state energy to an accuracy $\pm 4e{-}6$ after 1000 epochs of 10000 samples each. Small learning rates and large sample sizes are required to avoid overflows in the feed-forward computation of the neural network because the sampling algorithm is exponentially dependent on the system parameters. We also found that sufficiently small learning rates lead to continuous variation in the network parameters, so sampling can be run with significantly smaller batches without the need for thermalization between steps, similar to the contrastive divergence technique for training an RBM [Hinton, 2002].

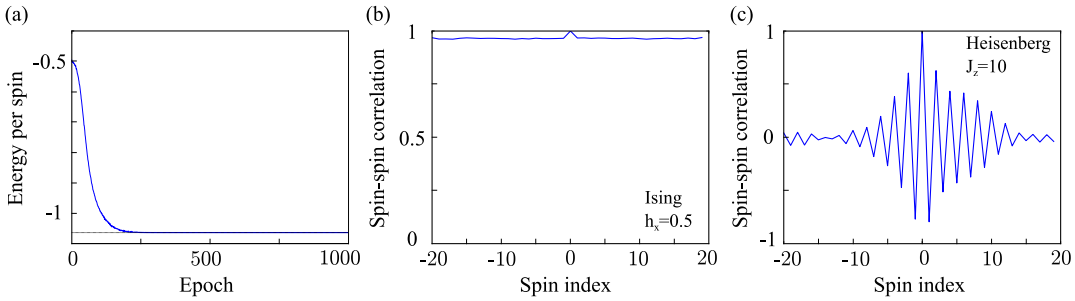

Figure 2: (a) Training curve showing average energy per spin for the 1D transverse field Ising model with $h_x = 0.5$ vs epoch. The minimized energy corresponds to the ground state for 40 spins in a 1D chain with periodic boundary conditions. The RBM has 40 visible units (spins), and 40 hidden units. (b) Spin-spin correlation $\langle \mathbf{x}^0\mathbf{x}^k \rangle$ for the network parameters at the end of training. (c) Spin-spin correlation $\langle \mathbf{x}^0\mathbf{x}^k \rangle$ for the network parameters at the end of training the 1D AFM Heisenberg Hamiltonian with $J_z = 10$ and 40 spins, 40 hidden units.

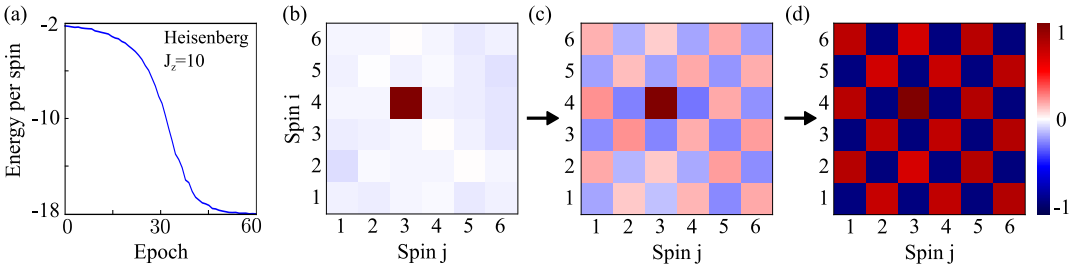

Figure 3: (a) Training curve showing average energy per spin for the 2d AFM Heisenberg model with $J_z = 10$ vs epoch. The minimized energy corresponds to the ground state for 36 spins in a 2d lattice with periodic boundary conditions. The RBM has 36 visible units (spins), 36 hidden units. Spin-spin correlations relative to the middle spin, i.e. $\langle \mathbf{x}^{43}\mathbf{x}^{ij} \rangle$, at initialization (b), at epoch 27 (c), and after training (d).

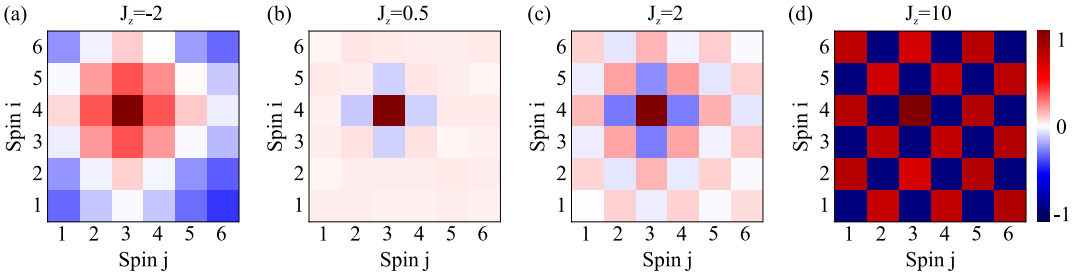

Figure 4: Spin-spin correlations relative to the middle spin $(4, 3)$, i.e. $\langle \mathbf{x}^{43}\mathbf{x}^{ij} \rangle$ as a function of J_z. (a) J_z = -2 (b) J_z = 0.5 (c) J_z = 2 (d) J_z = 10.

Confirmation that the network has obtained the ground state can be seen in the correlations $\langle \mathbf{x}^0\mathbf{x}^k \rangle$. Sites are highly correlated in the 1D TFI model with $h_x = 0.5$, where the system is known to be magnetically ordered (Fig. 2b). By contrast, the energy of the AFM Heisenberg model for $J_z = 10$ is minimized by negative correlation between neighboring sites, which is also captured by the RBM representation of the state (Fig. 2c). Figure 3 shows this neural network implementation is also suitable for 2D systems: the algorithm solves the AFM Heisenberg Hamiltonian for 36 spins arranged in a 6 x 6 lattice. While training the energy (Fig. 3a), the correct correlations also emerge between neighboring and distant sites (Fig. 3 b-d). The state is initialized randomly (Fig. 3b), but the correct correlations begin to emerge after only 27 epochs (Fig. 3c) and converge to the correct values when training is complete (Fig. 3d). The correlations of the final trained network also depend on the parameters of the given Hamiltonian (Fig. 4). For strong positive exchange J_z, anti-aligned spins are energetically preferred, while for negative exchange aligned spins are preferred. As J_z varies between positive and negative values a phase transition occurs between the expected configurations with aligned and anti-aligned spins.

7 Conclusion/Future Work

We successfully trained a restricted Boltzmann machine with stochastic reconfiguration updates to represent ground states of the 2D-AFM and 1D-Ising Hamiltonians. With this skeleton in place, we plan to evaluate other architectures which may be computationally more efficient, or may improve fidelity, depending on the system. In particular, these techniques would benefit from implementing weight sharing for systems with symmetries, particularly with convolutional architectures. Furthermore, extensions to unitary time dynamics under Hamiltonians not included in Carleo and Troyer [2017], such as non-local spin-spin coupling models, or quantum scattering experiments, would be interesting to explore. Finally, quantifying the compression and fidelity achieved by the RBM on a more extensive set of metrics would enable greater understanding of their capabilities.

8 Contributions

All authors contributed equally to writing the code and training the networks, to preparing the proposal, milestone, and presentation poster, and to writing this report.

9 Github Repo

`https://github.com/kafischer/neural-quantum-states`

References

Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017. ISSN 0036-8075. doi: 10.1126/science.aag2302. URL `http://science.sciencemag.org/content/355/6325/602`.

Giuseppe Carleo. Neural-network quantum states, June 2017. URL `http://kits.ucas.ac.cn/images/articles/2017/Machine_Learning/GiuseppeCarleo.pdf`.

Sandro Sorella, Michele Casula, and Dario Rocca. Weak binding between two aromatic rings: Feeling the van der waals attraction by quantum monte carlo methods. *The Journal of Chemical Physics*, 127(1):014105, 2007. doi: 10.1063/1.2746035. URL `https://doi.org/10.1063/1.2746035`.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992. doi: 10.1103/PhysRevLett.69.2863. URL `https://link.aps.org/doi/10.1103/PhysRevLett.69.2863`.

Shoudan Liang and Hanbin Pang. Approximate diagonalization using the density matrix renormalization-group method: A two-dimensional-systems perspective. *Phys. Rev. B*, 49:9214–9217, Apr 1994. doi: 10.1103/PhysRevB.49.9214. URL `https://link.aps.org/doi/10.1103/PhysRevB.49.9214`.

F Mezzacapo, N Schuch, M Boninsegni, and J I Cirac. Ground-state properties of quantum many-body systems: entangled-plaquette states and variational monte carlo. *New Journal of Physics*, 11 (8):083026, 2009. URL `http://stacks.iop.org/1367-2630/11/i=8/a=083026`.

Michael Lubasch, J. Ignacio Cirac, and Mari-Carmen Bañuls. Algorithms for finite projected entangled pair states. *Phys. Rev. B*, 90:064425, Aug 2014. doi: 10.1103/PhysRevB.90.064425. URL `https://link.aps.org/doi/10.1103/PhysRevB.90.064425`.

M. B. Hastings. Entropy and entanglement in quantum ground states. *Phys. Rev. B*, 76:035114, Jul 2007. doi: 10.1103/PhysRevB.76.035114. URL `https://link.aps.org/doi/10.1103/PhysRevB.76.035114`.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.