Neural Quantum States

Background Summary

Lucas Z. Brito
CSCI2470

April 20, 2024

Contents

1 Physics Background

Quantum many-body theory (QMBT) is the study of many interacting quantum degrees of freedom representing, say, atoms or electrons in a material. QMBT is intimately related to, and can be seen as an application of, quantum field theory (QFT), a more general framework used to describe a quantum system whose degrees of freedom are situated at every point on some sort of space, analogously to a vector field from multivariable calculus. Quantum field theory is tremendously important—it is not only used to describe quantum matter, but also the most fundamental confirmed physical theory, being that the standard model is a quantum field theory—but it is very mathematically difficult to make headway in understanding it or computing its predictions.

In the case of QMBT, one would like to understanding what phases of matter arise from strongly interacting quantum degrees of freedom. This is an interesting question because those phases display exotic properties desirable for, say, quantum computing applications, and because theses phases afford us insight into deep aspects of quantum field theory generally. The fact that the degrees of freedom are strongly interacting makes this in general a challenging task. Typically these systems are situated on a **lattice**—a grid[1] of points which may represent, say, the atoms of a crystal. We can work with lattices in any dimension, but for simplicity this project will focus on one-dimensional lattices (referred to as chains). We refer to the points on the grid as **sites**, and of things occurring on a particular site or on a group of nearby sites as **local**.

What are the degrees of freedom on the lattice? In principle can be anything: discrete, continuous, multivalued, etc. In the case of (strongly interacting) quantum matter usually it

[1] Technically it does not have to be a square grid, it can be for instance a tiling of triangles with degrees of freedom at every vertex.

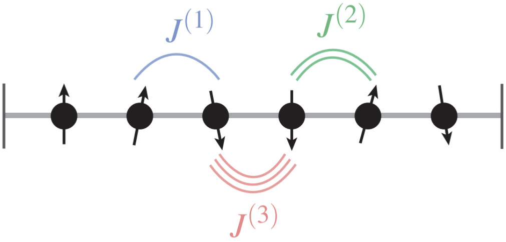


Figure 1

is something called the **spin**. Every electron comes with a small magnetic moment which, when measured, points in one of two directions (we can think of them as up or down). Of course, since this is quantum physics, the electrons can also be in a superposition of the up and down state; nonetheless, when we make a measurement we only find them in one of these two configurations. Electrons' spins might interact with one another such that the spins tend to align, or perhaps anti-align; thus, we have a strongly interacting quantum system on a chain, see fig. 1.

We represent the state that the system can be found in as a complex-valued vector called the **wavefunction**. For instance, for the spin of one electron, we might have a wavefunction

$$|\psi\rangle = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}$$

The notation on the left is called bra-ket notation. In that notation vectors are represented by $|\rangle$ and transposed vectors by $\langle|$ (note this is the conjugate transpose, so we also take the complex conjugate). An inner product is then $\langle\psi|\phi\rangle$. The wavefunction represents, roughly, the probability that the system is found in a particular state upon observation. More precisely, the absolute value squared represents the probability. For instance, consider the up state:

$$|\uparrow\rangle = \begin{bmatrix}1 \\ 0\end{bmatrix}, \qquad |\langle\uparrow|\,\psi\rangle\,|^2 = |\alpha|^2$$

A down state can be written similarly as $|\downarrow\rangle = (0, 1)$.

Observable quantities of a quantum-mechanical system are referred to as **observables**; they are represented by matrices. The expectation value of an observable is calculated by the inner product $\langle\psi|\, A\,|\psi\rangle$ and is denoted $\langle A\rangle$. One very important observable is the **Hamiltonian**, denoted H, which is nothing more than the energy of the system. It contains the particular interactions the system experiences—for instance, whether the spins tend to align, or whether they are reacting to magnetic field, etc. It is an important observable because how the system changes over time turns out to be entirely dependent on the Hamiltonian. In fact, the states of the system that *don't change* in time are the eigenvectors of the Hamiltonian (called eigenstates); their energies are the eigenvalues. The lowest-energy eigenstate, termed the **ground state**, is taken to be the most important state, as we find that it describes the state the system will tend to settle in under typical circumstances, and thus captures the properties of that particular phase of matter. Generally, we know the Hamiltonian for the system but not the ground state; thus, finding the ground state wavefunction is more or less the main aim of quantum mechanics problems.

We have considered a wavefunction for one spin. For many spins, one must combine the wavefunctions by taking tensor products. The wavefunction will then be a vector of size 2^N, where N is the number of electrons/lattice sites. This exponential growth makes it very difficult to find the eigenvectors of the Hamiltonian (and thus the ground state) by brute force algorithms as one runs out of memory quickly. It is referred to as the curse of dimensionality. The bulk of computational QMBT is dedicated to circumventing this bottleneck: in order to solve a Hamiltonian for a significant number of particles, one needs to be cleverer.

The many-body wavefunction can be written as

$$|\psi\rangle = \sum_{\{\sigma_i\}} \psi(\sigma_1, \ldots, \sigma_N) \, |\sigma_1, \ldots, \sigma_N\rangle$$

here σ_i represents the spin of the electron on the i-th site, and $|\sigma_1, \ldots, \sigma_N\rangle$ is a basis vector for a particular spin configuration (e.g., $|\uparrow, \downarrow, \ldots \downarrow\rangle$). $\{\sigma_i\}$ denotes a sum over all possible configurations of spins. $\psi(\sigma_1, \ldots, \sigma_N)$ is the complex number representing the value of that entry of the wavefunction vector. Since figuring out a basis is easy (we just need to list out every combination of up and down spins), the task lies in finding $\psi(\sigma_1, \ldots, \sigma_N)$.

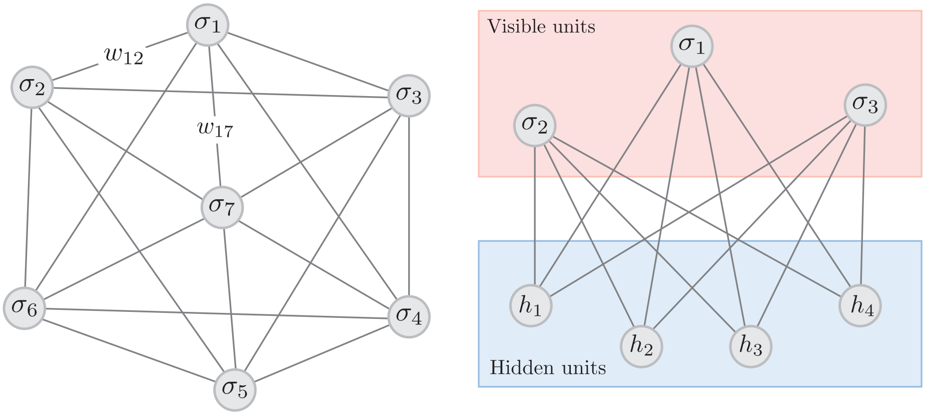
2 (Restricted) Boltzmann Machines

Boltzmann machines are a type of neural network which leverage insights from statistical physics to perform unsupervised learning of probabilistic models. That is, the goal is to use the Boltzmann machine to learn some target probability distribution $p_{\mathrm{T}}(X)$ given samples drawn from that probability distribution.

The machine itself consists of a set of units σ_i and weights w_{ij} connecting each of those units (fig. 2a). One defines an energy function

$$E(\sigma; w_{ij}, a_i) = \sum_i a_i \sigma_i + \sum_{ij} \sigma_i w_{ij} \sigma_j \tag{1}$$

with biases a_i and weights w_{ij} connecting each unit to every other unit. The units themselves are binary with ±1 (some other authors choose $\{0, 1\}$). This model belongs to the class of **energy-based models**; such models leverage the fact that for an appropriately chosen $E(v)$



(a) Boltzmann machine.

(b) Restricted Boltzmann machine. h_i do not connect to other h_i and likewise for σ_i.

Figure 2

(in the Boltzmann machine case, appropriately chosen w_{ij}) the target probability distribution—call this $p_{\mathrm{T}}(\sigma)$—can be sufficiently approximated by the Boltzmann distribution

$$p(\sigma) = \frac{1}{Z} e^{-E(\sigma; w_{ij}, a_i)}$$

where

$$Z = \sum_{\{\sigma_i\}} \exp(-E(\sigma_i; w_{ij}, a_i))$$

is the partition function, i.e., the sum over all possible σ_i configurations that functions as a normalizing constant.

The model then trains w_{ij} by minizing some sort of measure of the divergence of $p(\sigma)$ from $p_{\mathrm{T}}(\sigma)$—for instance maximum likelihood estimation or the Kullbach-Leibler divergence. Thus, given an unlabelled dataset $\{\boldsymbol{\sigma}_n\}$ which is drawn from the unknown distribution p_{T}, we may find $E(\sigma; w_{ij}, a_i)$ such that $p(\sigma)$ approximates p_{T}.

In practice, however, it is quite difficult to sample $p(\sigma)$ for training purposes. The culprit is Z, whose difficulty to evaluate is well-known to physicists. One then resorts to the usual techniques employed to evaluate complex sums or integrals of this kind—chiefly, Monte Carlo methods. We will see that the training of neural quantum states is no different and we will need to employ Monte Carlo sampling of the wavefunction.

A special case of the Boltzmann machine that was designed with training convenience in mind is the **restricted Boltzmann machine**. A restricted Boltzmann machine demotes a subset of σ_i to hidden units denoted h_i—these are units which do not appear as arguments to the probability distribution $p(\sigma_i)$ and whose purpose is analogous to a hidden layer in a feedforward network. Further, and crucially, we stipulate that the hidden units do not connect to themselves, only to visible units, and likewise for visible units (fig. 2b). This amounts to writing $\sigma_i w_{ij} \sigma_j$ as $h_i w_{ij} \sigma_j$, so that

$$E(\sigma_i; w_{ij}, a_i, b_i) = \sum_i a_i \sigma_i + \sum_i b_i h_i + \sum_{ij} h_i w_{ij} \sigma_j$$

(we have separated the bias into a visible bias a_i and a hidden bias b_i). The target probability distribution is approximated by the marginalization over h_i—the trace, in physics parlance—so we must sum over h_i

$$p(\sigma_i) = \sum_{\{h_i\}} e^{-E(\sigma,h)}$$

The gradient (of the log) for the n-th vector $\boldsymbol{\sigma}_n$

$$\begin{aligned}\frac{\partial \log p(\hat{\boldsymbol{\sigma}}_n)}{\partial w_{ij}} = \frac{1}{p(\hat{\boldsymbol{\sigma}}_n)} \frac{\partial p(\hat{\boldsymbol{\sigma}}_n)}{\partial w_{ij}} &= \frac{1}{p(\hat{\boldsymbol{\sigma}}_n)} \left[\sum_{\{h_i\}} e^{-E(v,\hat{\boldsymbol{\sigma}}_n)} \frac{\partial}{\partial w_{ij}} \left(\frac{1}{Z} \right) + \frac{1}{Z} \sum_{\{h_i\}} \frac{\partial}{\partial w_{ij}} e^{-E(h,\hat{\boldsymbol{\sigma}}_n)} \right] \\ &= \frac{1}{Z} \frac{1}{p(\hat{\boldsymbol{\sigma}}_n)} \left[- \sum_{\{h_i\}} e^{-E(v,\hat{\boldsymbol{\sigma}}_n)} \frac{1}{Z^2} \frac{\partial Z}{\partial w_{ij}} - \frac{1}{Z} \sum_{\{h_i\}} e^{-E(h,\hat{\boldsymbol{\sigma}}_n)} \hat{\sigma}_{n,i} h_j \right]\end{aligned}$$

The derivative of the partition function is

$$\frac{\partial Z}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_{\{h_i\}} \sum_{\{\sigma_i\}} e^{-E(h,\boldsymbol{\sigma}_n)} = \sum_{\{h_i\}} \sum_{\{\sigma_i\}} e^{-E(h,\boldsymbol{\sigma}_n)} (-\sigma_i, h_j) = -Z \langle \sigma_i v_j \rangle_{\text{model}}$$

The subscript "model" says that this is the expectation value of the current forward pass Boltzmann machine distribution as opposed to the expectation value with respect to the data, which will show up shortly. Then we have

$$\begin{aligned}\frac{\partial \log p(\hat{\boldsymbol{\sigma}}_n)}{\partial w_{ij}} &= \frac{1}{p(\hat{\boldsymbol{\sigma}}_n)} \left[\overbrace{\frac{1}{Z} \sum_{\{h_i\}} e^{-E(v,\hat{\boldsymbol{\sigma}}_n)}}^{p(\hat{\boldsymbol{\sigma}})} \frac{Z}{Z} \langle \sigma_i h_j \rangle_{\text{model}} - \overbrace{\frac{1}{Z} \sum_{\{h_i\}} e^{-E(h,\hat{\boldsymbol{\sigma}}_n)}}^{p(\hat{\boldsymbol{\sigma}})} \hat{\sigma}_{n,i} h_j \right] \\ &= \langle \sigma_i h_j \rangle_{\text{model}} - \hat{\sigma}_{n,i} h_j\end{aligned}$$

Now, the gradient is the expected value over the training vectors $\hat{\boldsymbol{\sigma}}_n$, so the derivative is

$$\frac{\partial \log p(\hat{\boldsymbol{\sigma}}_n)}{\partial w_{ij}} = \langle \sigma_i h_j \rangle_{\text{model}} - \langle \hat{\sigma}_{n,i} h_j \rangle_{\text{data}}.$$

The benefit of the restricted Boltzmann machine is that there is an exact distribution that h_j follows given some $\hat{\sigma}_n$, so we can evaluate the second term. Let's say that $h_k \in \{0, 1\}$; the probability that $h_k = 1$ given a visible unit can be calculated as follows. We have

$$\frac{p(h_k = 1 \mid \sigma)}{p(h_k = 0 \mid \sigma)} = \frac{\exp\left[b_k + \sum_{i\neq k h_i b_i} + \sum_i v_i a_i + \sum_{i\neq k,j} h_i w_{ij} v_j + \sum_j w_{kj} v_j\right]}{\exp\left[0 \cdot b_k + \sum_{i\neq k h_i b_i} + \sum_i v_i a_i + \sum_{i\neq k,j} h_i w_{ij} v_j + 0 \cdot \sum_j w_{kj} v_j\right]}$$

Now, since h_k can only be in two states, $p(h_k = 0 \mid \sigma) = 1 - p(h_k 0 \mid \sigma)$, and

$$\begin{aligned}\frac{p(h_k = 1 \mid \sigma)}{1 - p(h_k = 1 \mid \sigma)} &= \frac{\exp\left[b_k + \sum_{i\neq k h_i b_i} + \sum_i v_i a_i + \sum_{i\neq k,j} h_i w_{ij} v_j + \sum_j w_{kj} v_j\right]}{\exp\left[\sum_{i\neq k h_i b_i} + \sum_i v_i a_i + \sum_{i\neq k,j} h_i w_{ij} v_j\right]} \\ &= \exp\left(b_k + w_{kj} v_j\right)\end{aligned}$$

We can work out

$$p = e^W(1-p) \Longrightarrow (1 + e^W) = e \Longrightarrow p = \frac{e^W}{1 + e^W} \Longrightarrow \frac{1}{1 + e^{-W}} = \sigma(-W)$$

So that the above implies

$$p(h_k = 1 \mid \sigma) = \sigma(b_k + \sum_j w_{kj} v_j).$$

Notice that the argument relies on the fact that h_i does not connect to other hidden units. An identical argument finds

$$p(\sigma_k = 1 \mid h) = \sigma(a_k + \sum_j w_{jk})$$

Sampling $\langle \sigma_i h_j \rangle_{\text{model}}$, however, remains difficult. Strategies for doing so that leverage $p(h_k = 1 \mid \sigma)$ and $p(\sigma_k = 1 \mid h)$ are described in Hinton. In the case of neural quantum states, however, the natural loss function is the energy of the system, and the gradients we must calculate are not the above. The evaluation will still be difficult, but in this case only because we must obtain the energy and its gradients indirectly in order to circumvent the curse of dimensionality.

3 Neural Quantum States

One can view the many-body wavefunction $\psi(\sigma_1, \ldots, \sigma_N)$ appearing in

$$|\psi\rangle = \sum_{\{\sigma_i\}} \psi(\sigma_1, \ldots, \sigma_N) \left|\sigma_1, \cdots, \sigma_N\right\rangle$$

as a function from local degrees of freedom to a complex number, the amplitude, $\psi : \{\sigma_i\} \to \mathbb{C}$. Thus we may treat ψ as a a Boltzmann machine with visible units $\{\sigma_i\}$, albeit one that produces a complex number corresponding to the wavefunction as opposed to the probability amplitude. We thus provide ψ with a set of weights $W = (w_{ij}, a_i, b_i)$ and hidden units h_i such that

$$\psi(\sigma_1, \ldots, \sigma_N; W) = \sum_{\{h_i\}} \exp\left(\sum_j a_j \sigma_j^z + \sum_i b_i h_i + \sum_{ij} w_{ij} h_i \sigma_j^z\right)$$

Because ψ is complex-valued, w_{ij}, a_i, b_i are likewise taken to be complex valued. This is a restricted Boltzmann machine; we thus can explicitly marginalize ψ:

$$\begin{aligned}
\psi &= \sum_{\{h_i\}} \exp\left(\sum_j a_j \sigma_j^z + \sum_i b_i h_i + \sum_{ij} w_{ij} h_i \sigma_j^z\right) \\
&= \sum_{\{h_i\}} \exp\left(\sum_j a_j \sigma_j^z\right) \prod_i^M \exp\left(\sum_i b_i h_i\right) \exp\left(\sum_j w_{ij} h_i \sigma_j^z\right)
\end{aligned}$$

where we move the summation into the product as follows:

$$\begin{aligned}
&= \exp\left(\sum_j a_j \sigma_j^z\right) \prod_i^M \sum_{h_i = \{\pm 1\}} \exp\left(\sum_i b_i h_i\right) \exp\left(\sum_{ij} w_{ij} h_i \sigma_j^z\right) \\
&= \exp\left(\sum_j a_j \sigma_j^z\right) \prod_i^M \left[\exp\left(b_i + \sum_j w_{ij} \sigma_j^z\right) + \exp\left(-b_i - \sum_j w_{ij} \sigma_j^z\right)\right] \\
&= \exp\left(\sum_j a_j \sigma_j^z\right) \prod_i^M 2\cosh\left(b_i + \sum_j w_{ij} \sigma_j^z\right)
\end{aligned}$$

Note

For some reason I always have to convince myself that the above move works, so for future reference here's the argument: we have a product indexed by i and a sum indexed by ℓ, which expanded looks like

$$\begin{aligned}\prod_i^I \sum_\ell^L a_{i\ell} = (a_{11} + a_{12} + \cdots + a_{1L}) \\ \times (a_{21} + \cdots + a_{2L}) \\ \cdots \times (a_{I1} + \cdots + a_{IL})\end{aligned}$$

If we expand the products of sums, we obtain

$$= a_{11}a_{21}\cdots a_{I1} + \cdots + a_{1L}a_{2L}\cdots a_{IL} = \sum_{a_{1i}}^L \sum_{a_{2i}}^L \cdots \sum_{a_{Li}}^L \prod_i^I a_{i\ell}.$$

Thus

$$\psi(\sigma_1, \ldots, \sigma_N; W) = \exp\left(\sum_j a_j \sigma_j^z\right) \prod_i^M 2\cosh\left(b_i + \sum_j w_{ij}\sigma_j^z\right)$$

Given a Hamiltonian H, how do we update w_{ij}, a_i, b_i to minimize the energy $H = \langle\psi| H |\psi\rangle$? As is typical with this type of problem, one resorts to a Monte Carlo method. Carleo and Troyer use a technique called stochastic reconfiguration wherein one updates the weights as

$$W_{k+1,i} = W_{k,i} - \gamma_k S_{k,ij}^{-1} F_{k,j} \qquad (k = \text{iteration step})$$

where

$$\begin{gathered} S_{k,ij} = \langle O_i^\dagger O_j \rangle - \langle O_i^\dagger \rangle \langle O_j \rangle, \qquad F_i = \langle E_{\text{loc}} O_i^\dagger \rangle - \langle E_{\text{loc}} \rangle \langle O_i^\dagger \rangle \\ O_i = \frac{1}{\psi} \partial_{W_i} \psi, \qquad E_{\text{loc}} = \frac{1}{\psi} \langle \{\sigma_i\}| H |\psi\rangle \end{gathered}$$

Here the expectation values denote $\langle A \rangle = \sum_{\{\sigma_i\}} A(\{\sigma_i\}) |\psi(\{\sigma_i\})|^2$. They are evaluted with a Markov chain sampled with a Metropolis-Hastings procedure: as is common with these VMC techniques, we flip a random spin and accept the new configuration with probability

$$P(\{\sigma_i\}_k \to \{\sigma_i\}_{k+1}) = \min\left(1, \left|\frac{\psi(\{\sigma_i\}_{k+1})}{\psi(\{\sigma_i\}_k)}\right|^2\right).$$

References

- Giuseppe Carleo, Matthias Troyer, *Solving the quantum many-body problem with artificial neural networks*. Science 355, 602-606 (2017). DOI:10.1126/science.aag2302. See the supplemental materials for information on their Monte Carlo sampling.

- Torlai, Giacomo, and Roger G. Melko. *"Learning thermodynamics with Boltzmann machines."* Physical Review B 94.16 (2016): 165134. DOI:10.1103/PhysRevB.94.165134.
- Grathwolh, et al., *Your Classifier is Secretly an Energy Based Model and You Should Treat it Like One* (2019). arXiv:1912.03263.
- Hinton, *A Practical Guide to Training Restricted Boltzmann Machines*. `https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf`.
- Sorella, *Green Function Monte Carlo with Stochastic Reconfiguration*, DOI:10.1103/PhysRevLetter.80.5668.