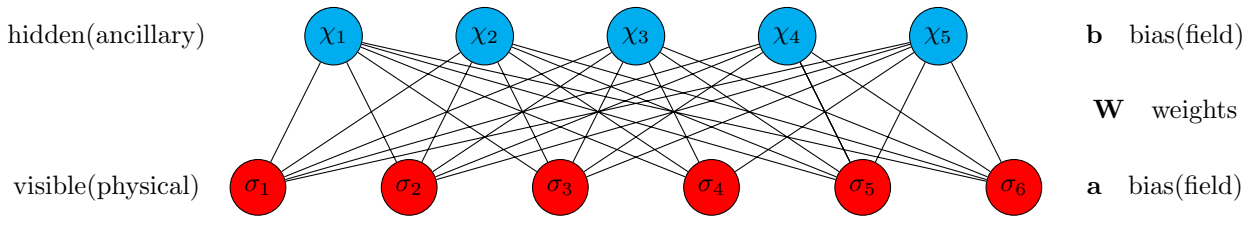
Restricted Boltzmann Machine

June 7, 2021

Contents



1 Introduction

The main task of Machine Learning(ML) is learning from data and making predictions. There are three main steps in ML:

1) Collecting the training data set, usually from experiments, our daily life, etc;
2) Finding a set of functions(also called model, which depends on a set of parameters) that is used to represent the law behind the data set;
3) Defining a loss function to measure the goodness of your function, given certain parameter value, and find the best parameter value to minimize the loss function.

Considerations on each step can lead to methods with very different models and task-oriented algorithms.

2 Basic Concepts & Application[1, 2]

2.1 Basic Concepts

Restricted Boltzmann Machine(RBM) is a natural model for studing statistical physics, which can be used to learn the probability distribution of a data set. For a particular configuration $(\vec{\sigma}, \vec{\chi}) \in \{0,1\}^{N+M}$, the RBM gives the joint distribution:

$$p(\vec{\sigma}, \vec{\chi}) = \frac{1}{Z} \exp\left(-E(\vec{\sigma}, \vec{\chi})\right), \tag{1}$$

where the corresponding energy is defined as:

$$E(\vec{\sigma}, \vec{\chi}) = -\sum_{i=1}^{M} b_i \chi_i - \sum_{j=1}^{N} a_j \sigma_j - \sum_{i,j} W_{ij} \chi_i \sigma_j, \tag{2}$$

and $Z = \sum_{\vec{\sigma},\vec{\chi}} \exp\left(-E(\vec{\sigma}, \vec{\chi})\right)$ is the so-called partition function which is essential to meet the normalization condition. Then we can get the marginal distribution for the physical configuration $\{\vec{\sigma}\}$:

$$p(\vec{\sigma}) = \sum_{\vec{\chi}} p(\vec{\sigma}, \vec{\chi}) = \frac{1}{Z} \exp\left(\sum_j a_j \sigma_j + \sum_i \ln\left(1 + \exp\left(b_i + \sum_j W_{ij}\sigma_j\right)\right)\right) = \frac{1}{Z} \exp\left(-\mathcal{E}(\vec{\sigma})\right), \tag{3}$$

and the red part is just the mean field term, while the cyan part, which is nonlinear, can capture the correlations. This makes RBM a powerful representation of a very complex probability distribution(indeed, any distribution with enough hidden units). Furthermore, we can calculate conditional distributions using Bayes's theorem as follows:

$$\begin{aligned} p(\vec{\sigma}|\vec{\chi}) &= \frac{\exp\left(-E(\vec{\sigma}, \vec{\chi})\right)/Z}{\sum_{\vec{\sigma}} \exp\left(-E(\vec{\sigma}, \vec{\chi})\right)/Z} \quad \text{insert the definition of energy } E(\vec{\sigma}, \vec{\chi}) \\ &= \frac{\prod_{j=1}^{N} \exp\left(a_j \sigma_j + \sum_i W_{ij} \chi_i \sigma_j\right)}{\prod_{j=1}^{N} (1 + \exp(a_j + \sum_i W_{ij} \chi_i))} \\ &= \prod_{j=1}^{N} \frac{\exp\left(a_j \sigma_j + \sum_i W_{ij} \chi_i \sigma_j\right)}{1 + \exp\left(a_j + \sum_i W_{ij} \chi_i\right)} . \end{aligned}$$

We can hence reformulate the conditional distributions in a 'decomposed' way(similar for $p(\vec{\chi}|\vec{\sigma})$)[1]:

$$p(\vec{\sigma}|\vec{\chi}) = \prod_{j=1}^{N} p(\sigma_j|\vec{\chi}) \quad \text{with} \quad p(\sigma_j = 1|\vec{\chi}) = \text{sigmoid}(a_j + \sum_i W_{ij}\chi_i)\,; \tag{4}$$

$$p(\vec{\chi}|\vec{\sigma}) = \prod_{i=1}^{M} p(\chi_i|\vec{\sigma}) \quad \text{with} \quad p(\chi_i = 1|\vec{\sigma}) = \text{sigmoid}(b_i + \sum_j W_{ij}\sigma_j)\,. \tag{5}$$

The sigmoid function is defined as: $\text{sigmoid}(x) = 1/(1+\exp(-x))$[2]. An attractive fact about RBM is that: given any configuration $\vec{\sigma}$ as input, we can generate a data set(obey the distribution 3) from Markov Chain Monte Carlo(MCMC) using the update rules 4 and 5.

[1] Attention: the units in the same layer are independent variables when the other layer is fixed.

[2] In practical programming, $\exp(-x)$ may cause overflow. There is another safer version for the sigmoid function as: $(1 + \tanh(x/2))/2$, which is preferred.

2.2 Application

Our aim is learning the thermodynamics of a classical spin system, and all we need to know is the partition function(Ising model for example):

$$Z(\beta) = \sum_{\vec{\sigma}} \exp\left(-\beta E(\vec{\sigma})\right) \quad \text{with} \quad E(\vec{\sigma}) = -J \sum_{<i,j>} \sigma_i \sigma_j . \tag{6}$$

However, it is not practical to calculate it for a general model. One can alternatively sample a data set from the corresponding distribution $q(\vec{\sigma}) = \exp\left(-\beta E(\vec{\sigma})\right)/Z(\beta)$ and average the quantity interested over the data set. Here, we generate the data set(training set $\mathcal{D}$) via conventional Monte Carlo Simulation and train the RBM to learn the probability distribution $q(\vec{\sigma})$ hidden in the data set, finally we sample from the trained RBM and calculate the physical quantities. Thus, we need to find parameter value(namely $\mathbf{W}$, $\mathbf{b}$, $\mathbf{c}$) which makes the RBM represent the distribution $q(\vec{\sigma})$ best. Kullback Leibler(KL) divergence is a measure of the distance between two distributions[3]:

$$\begin{aligned}
\mathbb{KL}(q||p) &= \sum_{\vec{\sigma} \text{ for all}} q(\vec{\sigma}) \ln \frac{q(\vec{\sigma})}{p(\vec{\sigma})} \quad \text{be zero iff } q(\vec{\sigma}) = p(\vec{\sigma}) \text{ for all } \vec{\sigma} \\
&= \sum_{\vec{\sigma} \text{ for all}} q(\vec{\sigma}) \ln q(\vec{\sigma}) - \sum_{\vec{\sigma} \text{ for all}} q(\vec{\sigma}) \ln p(\vec{\sigma}) \quad \text{the first term is fixed by the true distribution} \\
&\sim -\sum_{\vec{\sigma} \text{ for all}} \frac{1}{|\mathcal{D}|} \sum_{\vec{\sigma}' \in \mathcal{D}} \delta_{\vec{\sigma},\vec{\sigma}'} \ln p(\vec{\sigma}) \quad \text{note that } q(\vec{\sigma}) \text{ can be estimated from the data set } \mathcal{D} \\
&= -\frac{1}{|\mathcal{D}|} \sum_{\vec{\sigma} \in \mathcal{D}} \ln p(\vec{\sigma}) \quad \text{note that the term } \sum_{\vec{\sigma} \in \mathcal{D}} \ln p(\vec{\sigma}) = \ln \prod_{\vec{\sigma} \in \mathcal{D}} p(\vec{\sigma}) \text{ is also known as log-likelihood.}
\end{aligned}$$

The most common strategy to minimize the KL divergence(our loss function) is using Stochastic Gradient Descent(SGD). Thus we need to calculate the derivatives of KL divergence with respect to RBM parameters $\vec{\lambda}$ (attention: we write $\{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ as $\vec{\lambda}$):

$$\nabla_{\vec{\lambda}} \mathbb{KL}(q||p) = -\frac{1}{|\mathcal{D}|} \sum_{\vec{\sigma} \in \mathcal{D}} \nabla_{\vec{\lambda}} \ln p_{\vec{\lambda}}(\vec{\sigma}) \,. \tag{7}$$

Here we use $p_{\vec{\lambda}}(\vec{\sigma})$ to emphasize the parameter dependence of $p(\vec{\sigma})$. For each data $\vec{\sigma}^*$ in data set $\mathcal{D}$, we have:

$$\begin{aligned}
\nabla_{\vec{\lambda}} \ln p_{\vec{\lambda}}(\vec{\sigma}^*) &= \nabla_{\vec{\lambda}} \ln \frac{1}{Z_{\vec{\lambda}}} \sum_{\vec{\chi}} \exp\left(-E_{\vec{\lambda}}(\vec{\sigma}^*, \vec{\chi})\right) \\
&= \nabla_{\vec{\lambda}} \ln \sum_{\vec{\chi}} \exp\left(-E_{\vec{\lambda}}(\vec{\sigma}^*, \vec{\chi})\right) - \nabla_{\vec{\lambda}} \ln Z_{\vec{\lambda}} \\
&= -\frac{\sum_{\vec{\chi}} \exp\left(-E_{\vec{\lambda}}(\vec{\sigma}^*, \vec{\chi})\right) \nabla_{\vec{\lambda}} E_{\vec{\lambda}}(\vec{\sigma}^*, \vec{\chi})}{\sum_{\vec{\chi}} \exp\left(-E_{\vec{\lambda}}(\vec{\sigma}^*, \vec{\chi})\right)} + \frac{\sum_{\vec{\sigma},\vec{\chi}} \exp\left(-E_{\vec{\lambda}}(\vec{\sigma}, \vec{\chi})\right) \nabla_{\vec{\lambda}} E_{\vec{\lambda}}(\vec{\sigma}, \vec{\chi})}{\sum_{\vec{\sigma},\vec{\chi}} \exp\left(-E_{\vec{\lambda}}(\vec{\sigma}, \vec{\chi})\right)} \\
&= -\sum_{\vec{\chi}} p_{\vec{\lambda}}(\vec{\chi}|\vec{\sigma}^*) \nabla_{\vec{\lambda}} E_{\vec{\lambda}}(\vec{\sigma}^*, \vec{\chi}) + \sum_{\vec{\sigma},\vec{\chi}} p_{\vec{\lambda}}(\vec{\sigma}, \vec{\chi}) \nabla_{\vec{\lambda}} E_{\vec{\lambda}}(\vec{\sigma}, \vec{\chi}) \\
&= -\langle \nabla_{\vec{\lambda}} E_{\vec{\lambda}}(\vec{\sigma}^*, \vec{\chi}) \rangle_{p_{\vec{\lambda}}(\vec{\chi}|\vec{\sigma}^*)} + \langle \nabla_{\vec{\lambda}} E_{\vec{\lambda}}(\vec{\sigma}, \vec{\chi}) \rangle_{p_{\vec{\lambda}}(\vec{\sigma},\vec{\chi})} \,,
\end{aligned}$$

the first term is averaged over the conditional distribution for hidden layer given a certain $\vec{\sigma}^*$ which is easy to compute, while the second term is averaged over the distribution of the RBM which is difficult to calculate. Fortunately, the ML community have already proposed an efficient algorithm called contrastive divergence(CD) to estimate the second term. Here, we give more detail informations about the derivatives of KL divergence:

$$\nabla_{W_{ij}} \ln p_{\vec{\lambda}}(\vec{\sigma}^*) = \langle \chi_i \sigma_j^* \rangle_{p_{\vec{\lambda}}(\vec{\chi}|\vec{\sigma}^*)} - \langle \chi_i \sigma_j \rangle_{p_{\vec{\lambda}}(\vec{\chi},\vec{\sigma})} \tag{8}$$

$$\nabla_{a_j} \ln p_{\vec{\lambda}}(\vec{\sigma}^*) = \langle \sigma_j^* \rangle_{p_{\vec{\lambda}}(\vec{\chi}|\vec{\sigma}^*)} - \langle \sigma_j \rangle_{p_{\vec{\lambda}}(\vec{\chi},\vec{\sigma})} \tag{9}$$

$$\nabla_{b_i} \ln p_{\vec{\lambda}}(\vec{\sigma}^*) = \langle \chi_i \rangle_{p_{\vec{\lambda}}(\vec{\chi}|\vec{\sigma}^*)} - \langle \chi_i \rangle_{p_{\vec{\lambda}}(\vec{\chi},\vec{\sigma})} \,. \tag{10}$$

Once we have calculated the derivatives of KL divergence given a certain parameter value, we can update the parameters with following rule:

$$\vec{\lambda}^* = \vec{\lambda} - \eta \nabla_{\vec{\lambda}} \mathbb{KL}(\vec{\lambda}) \,,$$

[3] $-\mathbb{KL}(q||p) = \sum_{\vec{\sigma}} q(\vec{\sigma}) \ln \frac{p(\vec{\sigma})}{q(\vec{\sigma})} \leq \ln \sum_{\vec{\sigma}} q(\vec{\sigma}) \frac{p(\vec{\sigma})}{q(\vec{\sigma})} = 0$, which uses the Jensen's inequality, and the equality can be obtained via $p(\vec{\sigma})/q(\vec{\sigma}) = \text{const}$, together with the normalization condition, we finally arrive at: $\forall \vec{\sigma}, q(\vec{\sigma}) = p(\vec{\sigma})$.

where the factor η is the learning rate. The SGD will find the best parameter value automatically. Once we have trained a RBM, we can use MCMC method to sample from RBM which have learned the distribution of training data, and compute the physically relevant quantities as follows(denote sampled data set as: $\mathcal{S}$):

$$\langle \mathcal{O} \rangle = \frac{1}{|\mathcal{S}|} \sum_{\vec{\sigma} \in \mathcal{S}} \mathcal{O}(\vec{\sigma}) \,.$$

3 Training Algorithm

3.1 Gibbs Sampling and Contrastive Divergence

Now we have all the ingredients needed to train a RBM, and the main task is to calculate the values of 14, 15 and 10. Before we discuss the CD algorithm, it is essential to write these expressions more explicitly. For the first one:

$$\begin{aligned}
\nabla_{W_{ij}} \ln p_{\vec{\lambda}}(\vec{\sigma}^*) &= \sum_{\vec{\chi}} p_{\vec{\lambda}}(\vec{\chi}|\vec{\sigma}^*)\chi_i\sigma_j^* - \sum_{\vec{\sigma},\vec{\chi}} p_{\vec{\lambda}}(\vec{\sigma},\vec{\chi})\chi_i\sigma_j \quad \text{using the identity 5} \\
&= (\sum_{\chi_i} p_{\vec{\lambda}}(\chi_i|\vec{\sigma}^*)\chi_i\sigma_j^*) \prod_{k\neq i}(\sum_{\chi_k} p_{\vec{\lambda}}(\chi_k|\vec{\sigma}^*)) - \sum_{\vec{\sigma},\vec{\chi}} p_{\vec{\lambda}}(\vec{\sigma},\vec{\chi})\chi_i\sigma_j \\
&= \sum_{\chi_i} p_{\vec{\lambda}}(\chi_i|\vec{\sigma}^*)\chi_i\sigma_j^* - \sum_{\vec{\sigma}} p_{\vec{\lambda}}(\vec{\sigma}) \sum_{\vec{\chi}} p_{\vec{\lambda}}(\vec{\chi}|\vec{\sigma})\chi_i\sigma_j \\
&= p_{\vec{\lambda}}(\chi_i = 1|\vec{\sigma}^*)\sigma_j^* - \sum_{\vec{\sigma}} p_{\vec{\lambda}}(\vec{\sigma}) p_{\vec{\lambda}}(\chi_i = 1|\vec{\sigma})\sigma_j \\
&= p_{\vec{\lambda}}(\chi_i = 1|\vec{\sigma}^*)\sigma_j^* - \langle p_{\vec{\lambda}}(\chi_i = 1|\vec{\sigma})\sigma_j \rangle_{p_{\vec{\lambda}}(\vec{\sigma})} \qquad (11)
\end{aligned}$$

Similarly, we have:

$$\nabla_{a_j} \ln p_{\vec{\lambda}}(\vec{\sigma}^*) = \sigma_j^* - \langle \sigma_j \rangle_{p_{\vec{\lambda}}(\vec{\sigma})} \qquad (12)$$

$$\nabla_{b_i} \ln p_{\vec{\lambda}}(\vec{\sigma}^*) = p_{\vec{\lambda}}(\chi_i = 1|\vec{\sigma}^*) - \langle p_{\vec{\lambda}}(\chi_i = 1|\vec{\sigma}) \rangle_{p_{\vec{\lambda}}(\vec{\sigma})} \qquad (13)$$

In order to get an average over distribution $p_{\vec{\lambda}}(\vec{\sigma})$, we can perform a Gibbs sampling[4]. However, this is very time-consuming, for you have to do many steps(may be $\sim 10^3$ or larger) of Gibbs sampling before it reach the equilibrium(convergence). CD algorithm uses two tricks to circumvent this difficulty[5]: 1. start the Gibbs sampling with a $\vec{\sigma}$ from the training data set, not a random one; 2. do sampling after only a few steps of Gibbs sampling(in some cases, a single step is good enough!). In the spirit of CD algorithm, we can thus replace the average over $p_{\vec{\lambda}}(\vec{\sigma})$ in 11, 12 and 13 with $p_{\vec{\lambda}}(\chi_i = 1|\vec{\sigma}^k)\sigma_j^k$, σ_j^k and $p_{\vec{\lambda}}(\chi_i = 1|\vec{\sigma}^k)$, where $\vec{\sigma}^k$ is sampled from k-step Gibbs sampling whose starting point is $\vec{\sigma}^*$[6].

3.2 Pseudocode For Training Algorithm

Here I just give a detailed description of the whole training algorithm, see Algorithm 1.

4 Remarks

The success of RBM is impressive and it is desired to understand the root of its power. Recall the energy form defined on the RBM graph,

$$E(\vec{\sigma},\vec{\chi}) = -\sum_{i=1}^{M} b_i\chi_i - \sum_{j=1}^{N} a_j\sigma_j - \sum_{i,j} W_{ij}\chi_i\sigma_j,$$

it seems that there is no direct interaction between the physical spins. However, when we trace out the degrees of ancillary spins, the effective(or free) energy becomes,

$$\mathcal{E}(\vec{\sigma}) = -\sum_j a_j\sigma_j - \sum_i \ln\left(1 + \exp\left(b_i + \sum_j W_{ij}\sigma_j\right)\right),$$

[4]Start with a specific visible configuration $\vec{\sigma}^0$, we sample the hidden configuration $\vec{\chi}^0$ according to the probability 5 given $\vec{\sigma}^0$. Then we back to sample(reconstruct) the visible configuration $\vec{\sigma}^1$ according to the probability 4 given $\vec{\chi}^0$. If we do these sampling in order again and again, we will get a long sequence like(k-step Gibbs sampling): $\vec{\sigma}^0 \to \vec{\chi}^0 \to \vec{\sigma}^1 \to \vec{\chi}^1 \to \cdots \to \vec{\chi}^{k-1} \to \vec{\sigma}^k$. We refer to the ordered sampling procedure $\vec{\sigma} \to \vec{\chi} \to \vec{\sigma}$ as a single step of Gibbs sampling.

[5]The logic is that: our ultimate goal is getting a distribution $p_{\vec{\lambda}}(\vec{\sigma}) \sim q(\vec{\sigma})$. If we start Gibbs sampling with a $\vec{\sigma}$ sampled from $q(\vec{\sigma})$, it should already arrive at the equilibrium in some sense.

[6]You may be confused and ask the question that: how you can replace the average over a distribution $p_{\vec{\lambda}}(\vec{\sigma})$ just by one $\vec{\sigma}$ sampled from it? The answer lies in the fact that our derivative of KL divergence includes an average(or sum) over the whole data set or a batch in SGD(see expression 7). You may convince yourself that this averaging process over training data set or batch indeed mimics the averaging process over the distribution $p_{\vec{\lambda}}(\vec{\sigma})$.

and the correlations between the physical spins just appear. Thus the ancillary spins help to correlate the physical spins and make the RBM very powerful at representing the Boltzmann distribution. On the other hand, however, we can also encode the correlations of physical spins in a direct way. In fact, a direct correlation can add nothing to the representation power of the RBM, but will destroy the efficiency of the Gibbs sampling which is essential in the training algorithm. When the correlation is directly encoded, the 'decomposed' form of conditional distribution will disappear and makes the Gibb sampling resource consuming(even impossible). As a conclusion:

1) When the information of ancillary spins is unknown(marginal distribution), the physical spins are correlated which makes the RBM very powerful at representing a certain distribution;

2) When the information of ancillary spins is known(conditional distribution), the physcial spins are uncorrelated(independent) which makes the Gibbs sampling very efficient(practical),

the correlations between the physical spins must be merely introduced by the ancillary spins! This argument rules out the physical and ancillary interactions in the energy from.

Algorithm 1 Training of RBM via SGD

Require: Data Set $\mathcal{D}$, RBM's Parameters $\{\mathbf{W}, \mathbf{a}, \mathbf{b}\}$, Learning Rate η, Batch Size bs, Epoch ep, CD Order k, Visible Unit Number: N and Hidden Unit Number: M.

Ensure: $\{\mathbf{W}^*, \mathbf{a}^*, \mathbf{b}^*\} = \arg\min_{\vec{\lambda}} \mathbb{KL}(q(\vec{\sigma})||p_{\vec{\lambda}}(\vec{\sigma}))$

Initialize $\{\mathbf{W}, \mathbf{a}, \mathbf{b}\}$. Each element in $\mathbf{W}$ is sampled from a gaussian distribution: $W_{ij} \sim \mathcal{N}(0,1)/\sqrt{(N+M)}$, all the elements in $\mathbf{a}, \mathbf{b}$ are set to zero: $b_i \leftarrow 0, a_j \leftarrow 0$.
Split $\mathcal{D}$ into batches $\mathcal{B}$ randomly, each is of size bs
for *epoch* in $range(1, ep)$ **do**
    for each batch B in $\mathcal{B}$ **do**
        $\Delta W_{ij}, \Delta a_j, \Delta b_i$ are set to 0, for all i, j
        for each data $\vec{\sigma}$ in batch B **do**
            sample $\vec{\sigma}^k$ via k-step Gibbs sampling with $\vec{\sigma}$ as the starting configuration
            calculate $p(\chi_i = 1|\vec{\sigma})$ and $p(\chi_i = 1|\vec{\sigma}^k)$ for all i via 4 and 5
            $\Delta W_{ij} += p(\chi_i = 1|\vec{\sigma}) * \sigma_j - p(\chi_i = 1|\vec{\sigma}^k) * \sigma_j^k$
            $\Delta a_j += \sigma_j - \sigma_j^k$
            $\Delta b_i += p(\chi_i = 1|\vec{\sigma}) - p(\chi_i = 1|\vec{\sigma}^k)$
        end for
        $W_{ij} \leftarrow W_{ij} + \eta * \Delta W_{ij}/bs$
        $a_j \leftarrow a_j + \eta * \Delta a_j/bs$
        $b_i \leftarrow b_i + \eta * \Delta b_i/bs$
    end for
end for

5 Road to quantum many-body wave functions[3, 4]

In order to extend the application of RBM to the representation of quantum many-body wave function, we need to consider complex valued parameters:

$$\phi_{\vec{\lambda}}(\vec{\sigma}) = \frac{1}{\sqrt{Z_{\vec{\lambda}}}} \sum_{\vec{\chi}} \exp\left(\sum_i b_i \chi_i + \sum_j a_j \sigma_j + \sum_{i,j} \chi_i W_{ij} \sigma_j\right).$$

Because we do not have any data for this problem, we need to adopt variational principle:

1) Find the ground wave function, we have to minimize the energy function: $E_{\vec{\lambda}} = \langle \phi_{\vec{\lambda}} | \hat{H} | \phi_{\vec{\lambda}} \rangle / \langle \phi_{\vec{\lambda}} | \phi_{\vec{\lambda}} \rangle$;

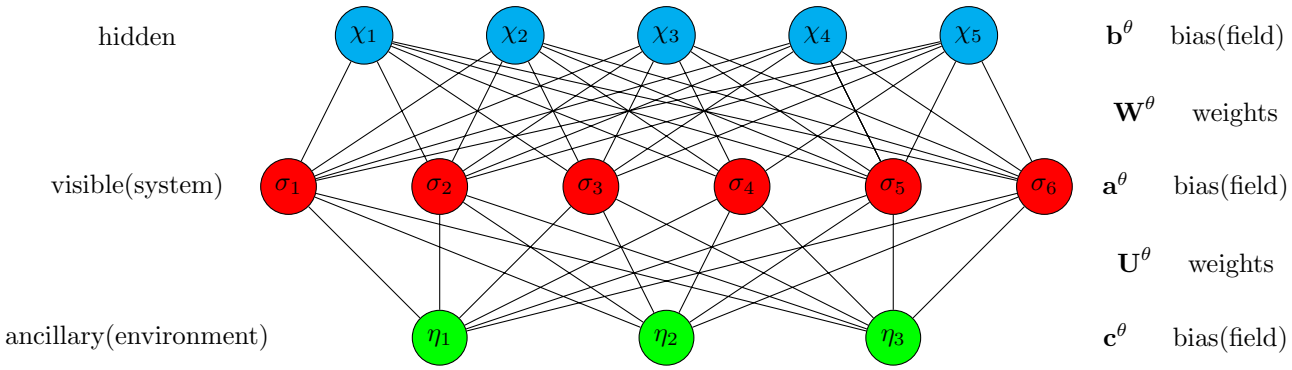
2) Simulate the dynamics of wave function, we have to minimize the distance(parameters becomes time-dependent): $\mathcal{R}(\dot{\vec{\lambda}}(t)) = \text{dist}(\partial_t |\phi(\vec{\lambda}(t))\rangle, -i\hat{H}|\phi(\vec{\lambda}(t))\rangle)$, and it is the time derivative of parameters that we have to learn.

The optimization of parameters is performed by Variational Monte Carlo(VMC) and Stochastic Reconfiguration Method(SR), which will be introduced later.

6 Road to density operator(mixed state)[5, 6, 7, 8, 9]

In the last section, we know how RBMs can be used to represent a quantum many-body wave function. However, any system in real world must experience decoherence for the coupling with the environment. Thus, it is necessary to know how RBMs can be used to represent a mixed state or density operator. Recently, people proposed three different schemes to achieve this goal.

6.1 The regular one[5, 6]



In order to represent a mixed state, the first step we may take is purifying the system with ancillary freedoms, which is a basic concept in quantum information theory. Thus the whole system(system and environment) can be written as:

$$|\phi\rangle = \sum_{\vec{\sigma},\vec{\eta}} \phi(\vec{\sigma},\vec{\eta})|\vec{\sigma}\rangle \otimes |\vec{\eta}\rangle,$$

and the state of system is easily obtained via tracing out the ancillary degrees:

$$\rho(\vec{\sigma},\vec{\sigma}') = \sum_{\vec{\eta}} \phi(\vec{\sigma},\vec{\eta})\phi^*(\vec{\sigma}',\vec{\eta}).$$

Here we use two RBMs, which share the same architecture but different parameters(denoted by $\vec{\lambda}$ and $\vec{\gamma}$), to represent the wave function of the whole system:

$$\phi(\vec{\sigma},\vec{\eta}) = \frac{1}{\sqrt{Z_{\vec{\lambda}}}}\sqrt{p_{\vec{\lambda}}(\vec{\sigma},\vec{\eta})}\exp\left(\frac{i}{2}\ln p_{\vec{\gamma}}(\vec{\sigma},\vec{\eta})\right),$$

after tracing out the ancillary degrees, we can get the expression of system's density operator.

6.2 The natural one[7, 9]

A natural way to represent a density operator is using one layer to encode the correlations within row degrees or column degrees, and another layer for the mixture of row and column degrees as[7]:

$$\rho(\vec{\sigma},\vec{\eta}) = \sum_{\vec{\chi},\vec{\gamma},\vec{\zeta}} \exp(\vec{c}\cdot\vec{\gamma}) \times \exp(\vec{a}\cdot\vec{\sigma} + \vec{b}\cdot\vec{\chi} + \vec{\chi}\cdot W\vec{\sigma} + \vec{\gamma}\cdot U\vec{\sigma}) \times \exp(\vec{a}^*\cdot\vec{\eta} + \vec{b}^*\cdot\vec{\zeta} + \vec{\zeta}\cdot W^*\vec{\eta} + \vec{\gamma}\cdot U^*\vec{\eta})$$

Obviously, this method is more natural than the first one and we don't need the domain knowledge about state purification from quantum information. However, we have to do this at the cost of using complex parameters.

6.3 The 'lazy' one[8]

Further more, there is also a 'lazy' way to represent the density matrix of the system. If we trust in the power of RBM, we can use the original structure[8]! We just equally split the visible layers into two parts, one for row and the other for column,

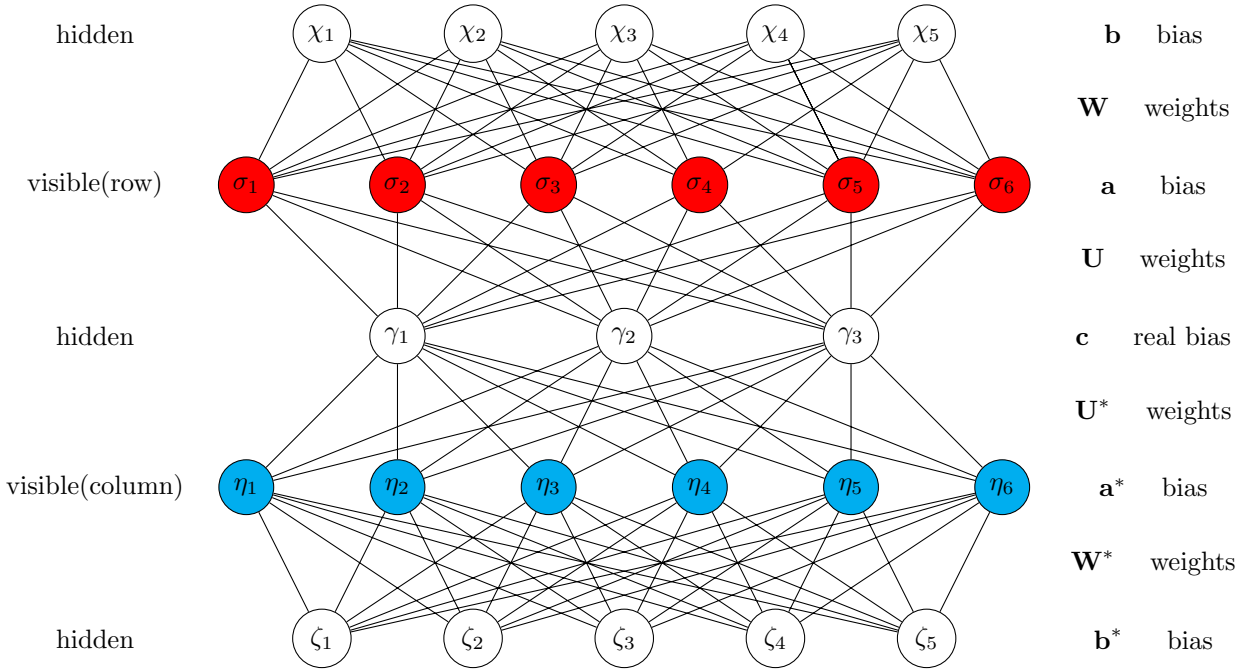
$$\rho(\vec{\sigma},\vec{\eta}) = \frac{1}{Z}\sum_{\vec{\chi}} \exp(\vec{a}\cdot\vec{\sigma} + \vec{a}^*\cdot\vec{\eta} + \vec{b}\cdot\vec{\chi} + \vec{\chi}\cdot W\vec{\sigma} + \vec{\chi}\cdot W^*\vec{\eta}).$$

Although we have applied some constraints on the RBM, it is not generally to be a legal density matrix[9]. We believe that the optimization procedure will help us find the stationary state which is legal.

[7] In order to satisfy the conditions of density operator, there must be some constraints on the RBM, like the conjugation of parameters, the symmetry of the structure, etc.

[8] With complex and conjugate parameters of course.

[9] Positive-semidefinite and hermitian. Queation: why not use real $\vec{b}$ instead of complex $\vec{b}$?



6.4 Map the optimization problem to a familiar one

The open quantum system is usually described by the so called Lindblad master equation for the reduced density matrix of the system:

$$\dot{\rho} = \mathcal{L}\rho,$$

and the stationary state is determined by the condition $\mathcal{L}\rho = 0$. As we do in HEOM, we can reshape the density matrix to be a vector and the super-operator $\mathcal{L}$ to be a matrix, then the target state is just the eigenstate of the super-operator with eigenvalue 0. Furthermore, the eigenstates other than the stationary one will have a negative value on their real part eigenvalues, which agrees with the physical picture that they will eventually decay to be zero in the dynamical evolution. Thus if we consider the super-operator $\mathcal{L}^\dagger\mathcal{L}$ which is positive-semidefinite and hermitian, we will find that the stationary state is just the ground state of the "hamiltonian" $\mathcal{L}^\dagger\mathcal{L}$ with exact zero eigenvalue!

7 Variational Monte Carlo method

7.1 Basic ideals of VMC

Assume we have a trial wave-function $|\phi\rangle$ with parameters $\vec{\alpha}$, we can expand it in the computational basis as:

$$|\phi\rangle = \sum_{\vec{\sigma}} |\vec{\sigma}\rangle\langle\vec{\sigma}|\phi\rangle = \sum_{\vec{\sigma}} \phi(\vec{\sigma})|\vec{\sigma}\rangle.$$

Thus we can compute any quantum observable $\hat{O}$ via:

$$\langle\hat{O}\rangle = \frac{\langle\phi|\hat{O}|\phi\rangle}{\langle\phi|\phi\rangle} = \frac{\sum_{\vec{\sigma}}\langle\phi|\vec{\sigma}\rangle\langle\vec{\sigma}|\hat{O}|\phi\rangle}{\sum_{\vec{\sigma}}\langle\phi|\vec{\sigma}\rangle\langle\vec{\sigma}|\phi\rangle} = \frac{\sum_{\vec{\sigma}}|\phi(\vec{\sigma})|^2 O_{\mathrm{loc}}(\vec{\sigma})}{\sum_{\vec{\sigma}}|\phi(\vec{\sigma})|^2}, \tag{14}$$

the local observable $O_{\mathrm{loc}}(\vec{\sigma})$ is defined by:

$$O_{\mathrm{loc}}(\vec{\sigma}) = \frac{\langle\vec{\sigma}|\hat{O}|\phi\rangle}{\langle\vec{\sigma}|\phi\rangle}.$$

However, it's impractical to sum over all the exponential numbered configurations. An alternative approach might be to sum over a set of configurations $\mathcal{S}$ sampled from the distribution $p(\vec{\sigma}) \propto |\phi(\vec{\sigma})|^2$, and the formula 14 can be approximated as:

$$\langle\hat{O}\rangle \approx \frac{1}{|\mathcal{S}|}\sum_{\vec{\sigma}\in\mathcal{S}} O_{\mathrm{loc}}(\vec{\sigma}). \tag{15}$$

In undergraduate quantum mechanical class, we have learned the variational principle as ($|n\rangle$ are the eigenstates of system's hamiltonian, and ordered by the eigenvalues as $E_0 \leq E_1 \leq \cdots \leq E_N$):

$$E_\phi = \frac{\langle\phi|\hat{H}|\phi\rangle}{\langle\phi|\phi\rangle} = \frac{\sum_{n=0}^{N} E_n\langle\phi|n\rangle\langle n|\phi\rangle}{\langle\phi|\phi\rangle} \geq \frac{E_0\sum_{n=0}^{N}\langle\phi|n\rangle\langle n|\phi\rangle}{\langle\phi|\phi\rangle} = E_0,$$

which states that the energy of a wave-function is bounded from below by the exact ground energy. The best choice of the parameters $\vec{\alpha}$ - the one makes the trial wave-function as close as possible to the ground state - is the one minimize the value of E_ϕ. Obviously, a good variational wave-function will be the key to the accuracy of our result. A bad variational wave-function can never be a good approximation of the ground state! In practical applications, people usually impose some constraints or forms to the variational function known from other analytic or symmetry considerations, like Hartree-Fock technique or Density functional theory. However, RBM is a very powerful representation of the wave-function, we have no need to do other complicated analysis to make the guess function have the similar form as the target one. Anyway, we have two main steps in our VMC algorithm:

1) Find the best parameters $\vec{\alpha}$ which minimize the value E_ϕ;

2) Calculate the quantities interested via formula 14 and 15.

Now, there are still two questions we have to answer:

1) How to sample the set of configuration $\mathcal{S}$ from the distribution $p(\vec{\sigma}) \propto |\phi(\vec{\sigma})|^2$;

2) How to find the best parameters $\vec{\alpha}$ for the ground state problem.

We will solve the first one in this section and leave the second one till the next section.

7.2 Importance sampling of spin configurations

The importance sampling of configurations is just similar as the Monte Carlo simulation of classical spin systems, and the only modification is the acceptance should be replaced by:

$$p(\vec{\sigma} \to \vec{\sigma}') = \min(1, \frac{|\phi(\vec{\sigma}')|^2}{|\phi(\vec{\sigma})|^2}).$$

8 Stochastic reconfiguration method

Our trial wave-function $|\phi\rangle$ is dependent on a set of parameters $\vec{\alpha} = \{\alpha_1, \alpha_2, \cdots, \alpha_p\}$, and we want to get the best $\vec{\alpha}$ to make $|\phi\rangle$ represent the ground state as close as possible. RS method tells us how to move in the parameter space to minimize the energy by an iterative way. For the trial wave-function $|\phi\rangle$ not orthogonal to the ground state, it is always possible to obtain a new state closer to the ground state by applying the operator $(\Lambda - \hat{H})$ to $|\phi\rangle$ with a sufficient large Λ at least bigger than the largest eigenvalue of hamiltonian. The basic ideal of SR is to change the parameters of $|\phi\rangle$ in order to be close as possible as $(\Lambda - \hat{H})|\phi\rangle$. We set the new state to be the following form[10]:

$$|\phi'\rangle = \delta\alpha_0|\phi\rangle + \sum_{k=1}^{p}\delta\alpha_k\frac{\partial|\phi\rangle}{\partial\alpha_k}.$$

This formula can be written in a more compact way as:

$$\begin{aligned}
|\phi'\rangle &= \delta\alpha_0|\phi\rangle + \sum_{k=1}^{p}\delta\alpha_k\frac{\partial}{\partial\alpha_k}\sum_{\vec{\sigma}}|\vec{\sigma}\rangle\langle\vec{\sigma}|\phi\rangle \\
&= \delta\alpha_0|\phi\rangle + \sum_{k=1}^{p}\delta\alpha_k\sum_{\vec{\sigma}}\frac{\partial\ln\phi(\vec{\sigma})}{\partial\alpha_k}|\vec{\sigma}\rangle\langle\vec{\sigma}|\phi\rangle \\
&= \sum_{k=0}^{p}\delta\alpha_k\hat{\Delta}_{\phi k}|\phi\rangle. \\
\hat{\Delta}_{\phi k} &:= \begin{cases} \mathbb{I} & \text{for } k=0 \\ \sum_{\vec{\sigma}}\frac{\partial\ln\phi(\vec{\sigma})}{\partial\alpha_k}|\vec{\sigma}\rangle\langle\vec{\sigma}| & \text{for } k\neq 0 \end{cases}
\end{aligned}$$

[10] In order to be close as possible as $(\Lambda - \hat{H})|\phi\rangle$, $\delta\alpha_k$ will not be a infinitesimal quantity, and a factor $\delta\alpha_0 \sim \Lambda$ is necessary before $|\phi\rangle$. However, they can point the desired trajectory in the parameter space and the updating formula needs a factor less than Λ^{-1} to ensure the convergence.

Thus, we can get the following equation:

$$\langle\phi|\hat{\Delta}_{\phi k}\sum_{k'=0}^{p}\delta\alpha_{k'}\sum_{\vec{\sigma}}\hat{\Delta}_{\phi k'}|\phi\rangle = \langle\phi|\hat{\Delta}_{\phi k}(\Lambda-\hat{H})|\phi\rangle.$$

Set the k to be zero, we have:

$$\delta\alpha_0 = \Lambda - \langle\hat{H}\rangle - \sum_{k'=1}^{p}\delta\alpha_{k'}\langle\hat{\Delta}_{\phi k'}\rangle \sim \Lambda.$$

Now we can substitute the value of $\delta\alpha_0$ for the case $k \neq 0$:

$$(\Lambda - \langle\hat{H}\rangle - \sum_{k'=1}^{p}\delta\alpha_{k'}\langle\hat{\Delta}_{\phi k'}\rangle)\langle\hat{\Delta}_{\phi k}\rangle + \sum_{k'=1}^{p}\delta\alpha_{k'}\langle\hat{\Delta}_{\phi k}\hat{\Delta}_{\phi k'}\rangle = \Lambda\langle\hat{\Delta}_{\phi k}\rangle - \langle\hat{\Delta}_{\phi k}\hat{H}\rangle$$

$$\Downarrow$$

$$\sum_{k'=1}^{p}\delta\alpha_{k'}(\langle\hat{\Delta}_{\phi k}\hat{\Delta}_{\phi k'}\rangle - \langle\hat{\Delta}_{\phi k'}\rangle\langle\hat{\Delta}_{\phi k}\rangle) = \langle\hat{H}\rangle\langle\hat{\Delta}_{\phi k}\rangle - \langle\hat{\Delta}_{\phi k}\hat{H}\rangle.$$

By defining the tensor $\overleftrightarrow{S}$ and vector $\vec{f}$ as:

$$\begin{aligned}\overleftrightarrow{S}_{kk'} &= \langle\hat{\Delta}_{\phi k}\hat{\Delta}_{\phi k'}\rangle - \langle\hat{\Delta}_{\phi k'}\rangle\langle\hat{\Delta}_{\phi k}\rangle;\\ \vec{f}_k &= \langle\hat{H}\rangle\langle\hat{\Delta}_{\phi k}\rangle - \langle\hat{\Delta}_{\phi k}\hat{H}\rangle,\end{aligned}$$

we finally come to the core formula of SR method[11]:

$$\overleftrightarrow{S}\,\delta\vec{\alpha} = \vec{f}.$$

In order to calculate $\delta\vec{\alpha}$, we need to know the expectation values in $\overleftrightarrow{S}$ and $\vec{f}$. Fortunately, they can all be approximated using the formula 15. The corresponding local observables are:

$$\begin{aligned}\frac{\langle\vec{\sigma}|\hat{\Delta}_{\phi k}|\phi\rangle}{\langle\vec{\sigma}|\phi\rangle} &= \frac{\langle\vec{\sigma}|\sum_{\vec{\sigma}'}\frac{\partial\ln\phi(\vec{\sigma}')}{\partial\alpha_k}|\vec{\sigma}'\rangle\langle\vec{\sigma}'|\phi\rangle}{\langle\vec{\sigma}|\phi\rangle} = \frac{\partial\ln\phi(\vec{\sigma})}{\partial\alpha_k};\\ \frac{\langle\vec{\sigma}|\hat{\Delta}_{\phi k}\hat{\Delta}_{\phi k'}|\phi\rangle}{\langle\vec{\sigma}|\phi\rangle} &= \frac{\langle\vec{\sigma}|\sum_{\vec{\sigma}'}\frac{\partial\ln\phi(\vec{\sigma}')}{\partial\alpha_k}|\vec{\sigma}'\rangle\langle\vec{\sigma}'|\sum_{\vec{\sigma}''}\frac{\partial\ln\phi(\vec{\sigma}'')}{\partial\alpha'_k}|\vec{\sigma}''\rangle\langle\vec{\sigma}''|\phi\rangle}{\langle\vec{\sigma}|\phi\rangle} = \frac{\partial\ln\phi(\vec{\sigma})}{\partial\alpha_k}\frac{\partial\ln\phi(\vec{\sigma})}{\partial\alpha'_k}\\ \frac{\langle\vec{\sigma}|\hat{\Delta}_{\phi k}\hat{H}|\phi\rangle}{\langle\vec{\sigma}|\phi\rangle} &= \frac{\langle\vec{\sigma}|\sum_{\vec{\sigma}'}\frac{\partial\ln\phi(\vec{\sigma}')}{\partial\alpha_k}|\vec{\sigma}'\rangle\langle\vec{\sigma}'|\hat{H}|\phi\rangle}{\langle\vec{\sigma}|\phi\rangle} = \frac{\partial\ln\phi(\vec{\sigma})}{\partial\alpha_k}E_{\rm loc}(\vec{\sigma})\end{aligned}$$

Finally, our updating formula for the parameters is:

$$\vec{\alpha}' = \vec{\alpha} + \Delta t\delta\vec{\alpha},$$

where $\Delta t \leq 1/\Lambda$ is a sufficient small factor to ensure the convergence.

9 Conclusion

So far, we have presented the basic ideal of using the model borrowed from machine learning to solve the quantum many-body physics. Besides the use of restricted Boltzmann machine, there are also many recent works which use other neural networks, like the convolutional neural network[10] and recurrent neural network[11, 12], to calculate the ground states of local Hamiltonians. It would be interesting to explore more connections between the many-body physics and machine learning in the future!

[11]The inverse of tensor $\overleftrightarrow{S}$ may be ill defined, so we often add a small identity term to ensure the legality of the inverse operation as: $\overleftrightarrow{S}+\epsilon\mathbb{I}$, and good results can be obtained with $\epsilon \leq 10^{-4}$. Further the inverse of $\overleftrightarrow{S}$ is very hard for large number of parameters, which is a constraint.

References

[1] Learning Thermodynamics with Boltzmann Machines, G. Torlai, R.G. Melko, arXiv:1606.02718v1

[2] An Introduction to Restricted Boltzmann Machines, A. Fischer, C. Igel

[3] Solving the Quantum Many-Body Problem with Artificial Neural Networks, G. Carleo, etc, Science.355.602(2017)

[4] Restricted Boltzmann machines in quantum physics, R. G. Melko, etc, Nature Physics.15.887–892(2019)

[5] Latent Space Purification via Neural Density Operators, G. Torlai, R. G. Melko, PhysRevLett.120.240503(2018)

[6] Variational neural network ansatz for steady states in open quantum systems, F. Vicentini, A. Biella, etc, PhysRevLett.122.250503(2019)

[7] Variational Quantum Monte Carlo Method with a Neural-Network Ansatz for Open Quantum Systems, A. Nagy, V. Savona, PhysRevLett.122.250501(2019)

[8] Constructing neural stationary states for open quantum many-body systems, N. Yoshioka, R. Hamazaki, PhysRevB.99.214306(2019)

[9] Neural-Network Approach to Dissipative Quantum Many-Body Dynamics, M. J. Hartmann, G. Carleo, PhysRevLett.122.250502(2019)

[10] Two-dimensional frustrated J_1-J_2 model studied with neural network quantum states, Kenny Choo, Titus Neupert, and Giuseppe Carleo, Phys.Rev.B100,125124(2019)

[11] Recurrent neural network wave functions, Mohamed Hibat-Allah, Martin Ganahl, Lauren E. Hayward, etc, Phys.Rev.Research2,023358(2020)

[12] Iterative Retraining of Quantum Spin Models Using Recurrent Neural Networks, Christopher Roth, arXiv:2003.06228