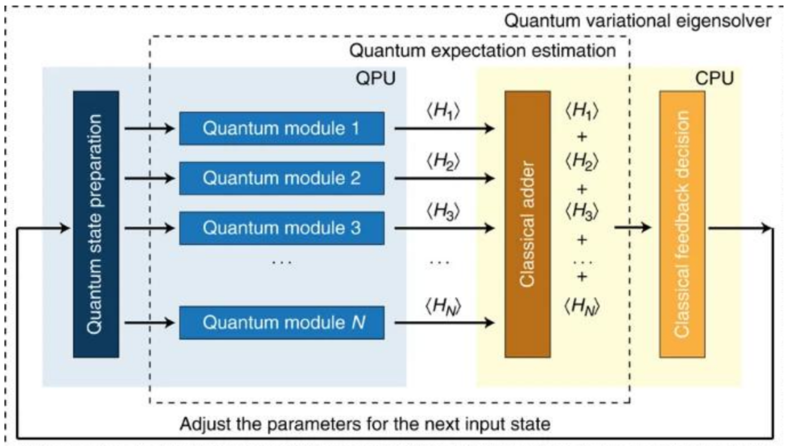
Solving the Ground State of the Classical Spin Model Using a Single-Layer Neural Network

Zhongling Lu, Shanghai Jiao Tong University, Department of Physics & Zhiyuan College

(Notice: the following content is auto-translated by Google, it could have several errors.)

I. Introduction

Numerical methods (i.e. non-analytical solutions) to solve physical models by minimizing/optimizing some value, the Variational Quantum Eigensolver VQE as far as I know.



Taking the spin model as an example, on the Bloch sphere, each spin has 2 degrees of freedom and is controlled by a unitary gate of two parameters (if there are many-body operators in the Hamiltonian, it is controlled by a multi-quantum gate); After the gate operation, a random set of spin quantum configurations is obtained, and its "energy" (which is not actually the real energy) is measured many times, and the expected value of the Hamiltonian is obtained. Then, the expected value is optimized to the minimum through optimization algorithms such as gradient descent, and the ground state of the spin Hamiltonian is reached. Essentially, Variation Minimization$\langle\psi|H|\psi\rangle$the process of.

Such a method is physically explicit, and if the Hamiltonian is well behaved, the ground state and energy can usually be found. But for quantum models, there is no doubt that quantum computers or quantum simulators are needed, but the calculation speed and accuracy of the two have so far been unsatisfactory. Second, if this idea is applied to the classical spin model, the Hamiltonian that needs to be optimized should contain$m \times 2$parameters,$m$is the number of spins. This is relatively simple for a small number of spins, but for a 2D model, there are many parameters that need to be optimized, and the computational complexity rises sharply.

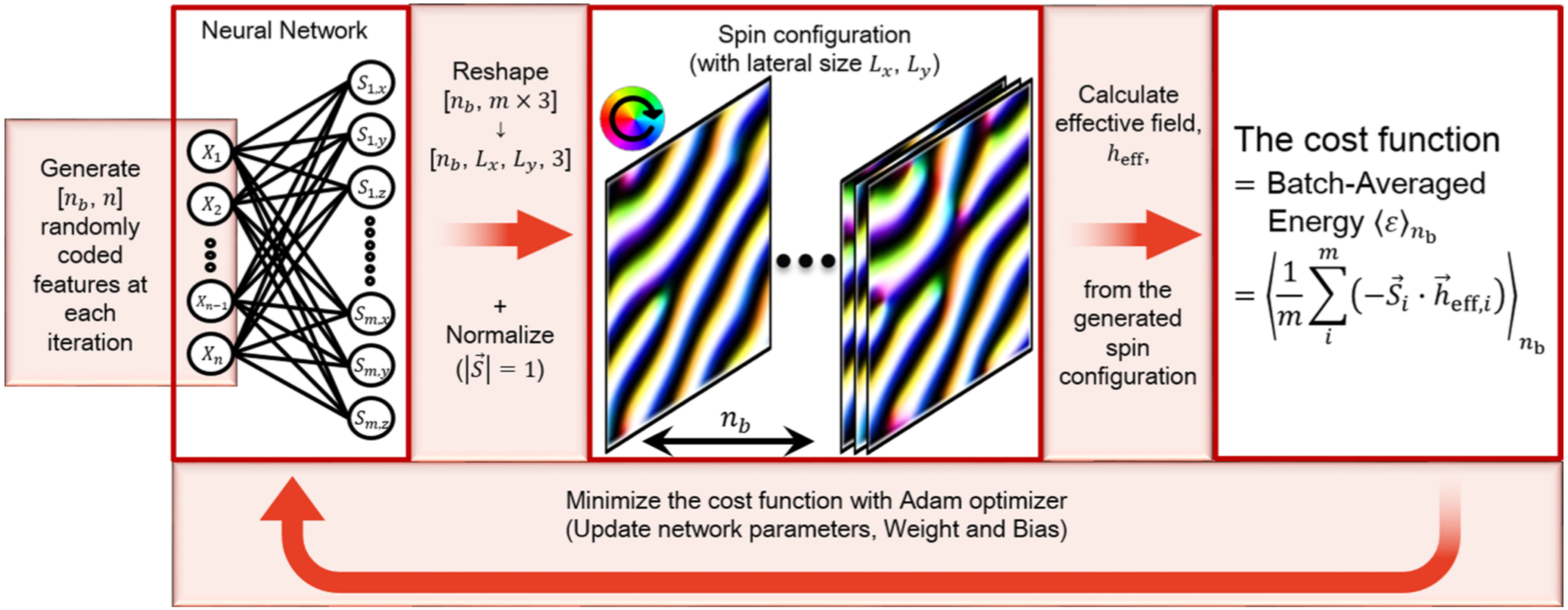
However, in this assignment, I refer to a paper ([*Kwon HY, et al. Physical Review B, 2019*] [1]) which attempts to construct a classical spin configuration using a neural network and solve the ground state by training the network. This gives us another way of thinking. Although the computational complexity of the model designed in this paper is not much different from the above methods, the characteristics of the neural network are its flexibility and extensibility. By designing different model architectures, we may be able to reduce the complexity very low, such as reducing the width and increasing the depth; in addition, this method may even be portable to the solution of quantum models.

I took the idea of this work and tried to deal with some simple classical spin models. I built a single-layer FNN network and tried to encode the classical spin model into this network and optimize to get its ground state. In the following example, we demonstrate a 1D classical AFM Heisenberg chain, add an external magnetic field, and observe how the <Sz> of the ground state changes with the magnetic field B. The implementation is based on the deep learning framework pytorch.

Second, the experimental idea

As shown in the figure below (source [1]), the author first builds a fully connected network without hidden layers, whose input is a random feature (my own understanding is a random array), and the output is the length$m \times 3$an array of$m$three components of a spin. The output is then normalized so that it meets the requirements of classical spin, which also acts as an activation function for the network, adding nonlinearity. Subsequently, the average energy of each batch is used as the loss function for training.

In a physical sense, if the training results, the output layer should have all the information about the spin configuration. In the cited paper, the authors compute a 2D DM interaction model and observe the spin wave structure in the output layer. (middle part of the picture below)



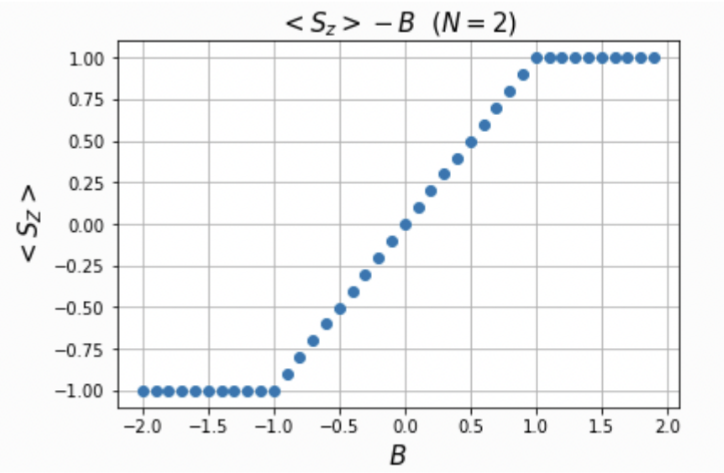
3. Experimental results

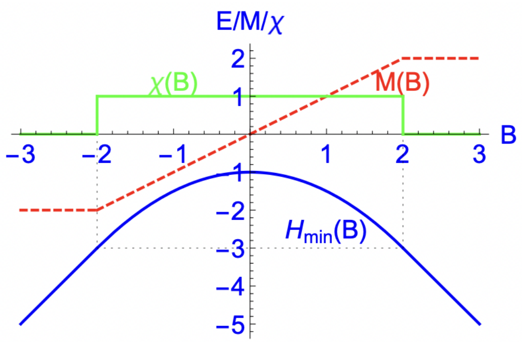
In the reproduction, for simplicity, I constructed a 1D classical AFM Heisenberg chain, added an external magnetic field, and trained it. The convergence is found to be good:

```
epoch 8, loss -201.000000
net deleted
epoch 8, loss -176.000000
net deleted
epoch 8, loss -151.000000
net deleted
epoch 8, loss -126.000000
net deleted
```

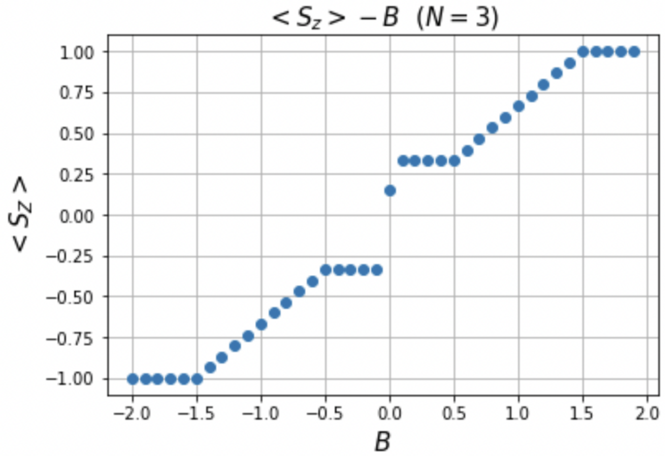
Then observe how the $<S_z>$ of the ground state changes with the magnetic field B. We trained spin numbers separately$N = 2, 3, 50$Three cases, and compared the first two cases with the theory (***Schmidt H J. , arXiv:1710.00318, 2017.*** [2]), and found that they are in good agreement (the **red line on the right is the theoretical value**):

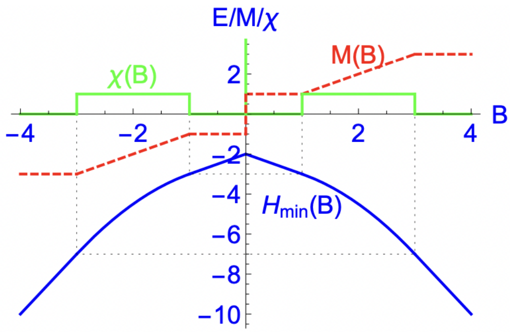
$N = 2$:



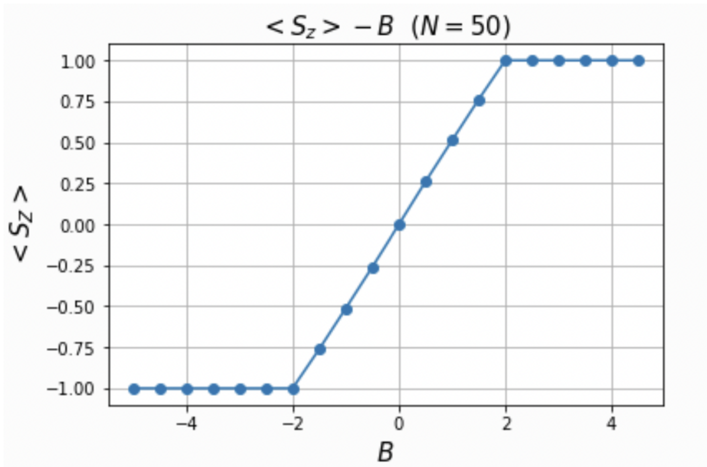


$N = 3$: (note the step near 0 in the red part of the right picture, which is consistent with the experimental results; when B=0, it cannot converge due to extreme energy oscillation)





$N = 50$: **(After trying, not only N=50, but when the number of spins is large, the order will also disappear and become the following curve; no theoretical curve was found)**



4. Analysis and discussion of results

Through the above experiments, we show some preliminary results, which are in good agreement with the analytical solution to a certain extent. This kind of agreement not only refers to the agreement of the calculation results, but also reveals the physical meaning - the output layer can accurately reflect the change of the order parameter, indicating that the network has indeed learned the physical image of the configuration of the spin model. This is completely different from the "black box mechanism" of traditional machine learning, especially neural networks.

It should be pointed out, however, that I didn't fully understand why this approach worked, and one of the things that puzzled me in particular was the input of random features. I've tried inputting random features with different distributions, but it doesn't seem to have an effect on the results, and the robustness is very strange. In the text, it is only mentioned that this method is more commonly used in ML. In addition, I also roughly tried the Hamiltonian of the XXZ model without an external field, but the training results seem to be a little wrong. It may be that the depth of the model is not enough or the loss is too oscillating, or it may be that my code is wrong. The above two points have not been further explored due to time reasons.

However, through the above experiments, we found that this idea is indeed valuable, and it may be extended to various physical models through continuous improvement. As far as I know, the d-dimensional quantum model is equivalent to the d+1-dimensional classical model in the sense of the partition function (I don't know if the expression is accurate or not). From the model that has been solved, can we try to solve it in this way$d = 2$The classical XXZ model of , and contrast it with the 1D quantum XXZ chain solved rigorously by Bethe Ansatz? Can solving a 2D XY model observe the topology, or even the BKT phase transition? Furthermore, we may even be able to solve some models that are not yet well explained, and obtain insights from the results to further improve the existing theory. But as far as my personal opinion is concerned, I am not very optimistic about the application of deep learning in this area. After all, we cannot guarantee that a black box that has not been well explained can accurately show us the physical image in it. We should be careful not to abuse this technology, let alone forcefully combine deep learning with physics for the sake of water articles.

(Code and implementation are on the next page)