Machine Learning Methods for Quantum Many-body Problem

Swarma Club

Xiuzhe Luo

July 17, 2017

University of Science and Technology of China

Hello everyone, today I will report some recent progress in quantum many-body problem.

This slide is available on my blog: rogerluo.me/slides.

Table of contents

In the following part

- First I will introduce some related backgrounds in quantum physics
- Second I will show you the connection between neural networks and quantum amplitudes
- Thirdly I will introduce recent progress in this field based on neural networks
- Finally I will introduce some potential some open problems

Introduction

Quantum Many-body Problem

Quantum Many-body Problem Solving the Shordinger equation in many-body system

$$
\begin{aligned}
i\frac{d}{dt}|\Psi(t)\rangle &= (\sum_i \frac{\hat{p}_i}{2m} - \hat{v})|\Psi(t)\rangle \\
i\frac{d}{dt}|\Psi(t)\rangle &= (\sum_{ij} \vec{s}_i\vec{s}_j)|\Psi(t)\rangle
\end{aligned} \tag{1}
$$

Quantum Many-body Problem appears in many fields, eg. condensed matter physics, quantum chemistry, high-energy physics. Although, the Shordinger equation offers an elegant way to describe the quantum physics, the situation become much more different when the bodies of target system become too large, like Anderson says:

> *More is different.[1]*

The Hilbert space which contains all the quantum states increases exponentially with its dimension. Therefore, it can be hard to calculate the quantum many-body problem.

The Complexity of Hilbert Space

deterministic Turing machine: DTM
non-deterministic Turing machine: NDTM
Quantum Computer: QC

- **P**: can be solved by a DTM in polynomial time

To describe how hard is it to calculate quantities in quantum physics, we could use the computation complexity. [2] And actually you can find all these definitions on wiki.

First for deterministic Turing machine

- **P** Problem is usually what we prefer, which can be solved in polynomial
- for problems harder than **P** the explosion of complexity can make it impractical (like some encrpytion tachniques)

The Complexity of Hilbert Space

deterministic Turing machine: DTM
non-deterministic Turing machine: NDTM
Quantum Computer: QC

- **P**: can be solved by a DTM in polynomial time
- **NP**: can be verified by a DTM in polynomial time (NDTM in polynomial)

To describe how hard is it to calculate quantities in quantum physics, we could use the computation complexity. [2] And actually you can find all these definitions on wiki.

First for deterministic Turing machine

- **P** Problem is usually what we prefer, which can be solved in polynomial
- for problems harder than **P** the explosion of complexity can make it impractical (like some encrpytion tachniques)

The Complexity of Hilbert Space

deterministic Turing machine: DTM
non-deterministic Turing machine: NDTM
Quantum Computer: QC

- **P**: can be solved by a DTM in polynomial time
- **NP**: can be verified by a DTM in polynomial time (NDTM in polynomial)
- **BQP**: can be solved by a QC in polynomial time

And there is even problems that a quanutm computer can not solve. And further in these cases, it is connected with Hamiltonians. [3, 2, 4] And a popular problem is the estimation of ground state, which means the quantum state with lowest energy. The calculate of ground state is important because most physical states are in low energy.

The Complexity of Hilbert Space

deterministic Turing machine: DTM
non-deterministic Turing machine: NDTM
Quantum Computer: QC

- **P**: can be solved by a DTM in polynomial time
- **NP**: can be verified by a DTM in polynomial time (NDTM in polynomial)
- **BQP**: can be solved by a QC in polynomial time
- **NP-Complete**: all the problem in **NP** can be reduced to it in polynomial time

And there is even problems that a quanutm computer can not solve. And further in these cases, it is connected with Hamiltonians. [3, 2, 4] And a popular problem is the estimation of ground state, which means the quantum state with lowest energy. The calculate of ground state is important because most physical states are in low energy.

The Complexity of Hilbert Space

deterministic Turing machine: DTM
non-deterministic Turing machine: NDTM
Quantum Computer: QC

- **P**: can be solved by a DTM in polynomial time
- **NP**: can be verified by a DTM in polynomial time (NDTM in polynomial)
- **BQP**: can be solved by a QC in polynomial time
- **NP-Complete**: all the problem in **NP** can be reduced to it in polynomial time
- **QMA(c, s)**: can be verified by a QC in polynomial time

And there is even problems that a quanutm computer can not solve. And further in these cases, it is connected with Hamiltonians. [3, 2, 4] And a popular problem is the estimation of ground state, which means the quantum state with lowest energy. The calculate of ground state is important because most physical states are in low energy.

The Complexity of Hilbert Space

deterministic Turing machine: DTM
non-deterministic Turing machine: NDTM
Quantum Computer: QC

- **P**: can be solved by a DTM in polynomial time
- **NP**: can be verified by a DTM in polynomial time (NDTM in polynomial)
- **BQP**: can be solved by a QC in polynomial time
- **NP-Complete**: all the problem in **NP** can be reduced to it in polynomial time
- **QMA(c, s)**: can be verified by a QC in polynomial time
- **QMA-Complete**: all the problem in **QMA** can be reduced to it in polynomial time

And there is even problems that a quanutm computer can not solve. And further in these cases, it is connected with Hamiltonians. [3, 2, 4] And a popular problem is the estimation of ground state, which means the quantum state with lowest energy. The calculate of ground state is important because most physical states are in low energy.

The Complexity of Hilbert Space

deterministic Turing machine: DTM
non-deterministic Turing machine: NDTM
Quantum Computer: QC

- **P**: can be solved by a DTM in polynomial time
- **NP**: can be verified by a DTM in polynomial time (NDTM in polynomial)
- **BQP**: can be solved by a QC in polynomial time
- **NP-Complete**: all the problem in **NP** can be reduced to it in polynomial time
- **QMA(c, s)**: can be verified by a QC in polynomial time
- **QMA-Complete**: all the problem in **QMA** can be reduced to it in polynomial time
- More: Complexity Zoo

There are quite a lot complexity classes in fact, which is collected by Scott Aaronson in the Complexity Zoo.

The Complexity of Hilbert Space

It is hard to solve the many body problem in such a space, eg

k-local Hamiltonian ground-state estimation Given a k-local Hamiltonian on n qubits, $H = \sum_{i=1}^{r} H_i$, where $r = poly(n)$ and each H_i acts non-trivially on at most k qubits and has bounded operator norm $\|H_i\| \leq ploy(n)$, determine whether:

- H has an eigenvalue less than a
- all of the eigenvalues of H are larger than b

promised one of these to be the case, where $b - a \leq 1/poly(n)$

The Complexity of Hilbert Space

proved

- QMA-complete for $k \geq 2$ [5]
- QMA-complete when $k = O(log(n))$
- QMA-complete when $k = 3$ with constant norms (i.e. $\|H\| = O(1)$)
- QMA-complete when $k = 2$ on 2-D lattice (eg. the J1J2 model)
- etc.

Such problem can be really hard!

Some general techniques in quantum many-body

Variational Methods

To solve a real equation:

$$F(u) = 0, \quad F : \mathcal{C}^n \to \mathcal{R}^n \tag{2}$$

We could minimize:

$$J(u) = \min J(v), v \in \mathcal{C}^n \tag{3}$$

Approximate $J(v)$ with

$$P(v|\theta) \quad \theta \in \mathcal{D} \tag{4}$$

To solve a real equation, we could convert it to an optimization problem. However, finding a proper function J could be hard. A simple approach is to guess a solution $P(v|\theta)$ with some parameters, which is called varitional parameters. Then we adjust those parameters to approximate the real function J. In quantum physics, we call it as ansatz. $\theta \in \mathcal{D}$, $\mathcal{D}$ will be a much smaller subspace of $\mathcal{C}^n$.

Quantum Monte Carlo

In quantum many-body problem, we usually has two goals:

- find the gound state: $\min_\theta E(\Psi(\theta)) = \min_\theta \langle \Psi(\theta)|\mathcal{H}|\Psi(\theta)\rangle$
- time evolution: $\min_\theta \left\| i\frac{d}{dt}|\Psi_{\mathcal{W}}\rangle - \mathcal{H}|\Psi_{\mathcal{W}}\rangle \right\|$

To solve the above problem(which can be really hard), we usually guess a parameterized quantum state $\Psi(\theta)$ which is usually called ansatz. Then we coudl solve it in a much smaller space, but it would still be hard to calculate dot product like $\langle\Psi|\mathcal{H}|\Psi\rangle$. (eg. $O(d^n)$, d is a constant)

But wait!

$$\begin{aligned}
\frac{\partial E}{\partial W^R} &= \partial_{W^R} \frac{\langle \Psi | H | \Psi \rangle}{Z} \\
&= -\frac{1}{Z^2} \cdot \frac{\partial Z}{\partial W^R} \sum_{ss'} \overline{\Psi}_s \Psi_{s'} H_{ss'} + \frac{1}{Z} \sum_{ss'} \frac{\partial}{\partial W^R} (\overline{\Psi}_s \Psi_{s'} H_{ss'}) \\
&= -\frac{1}{Z^2} \cdot \sum_s (\frac{\partial \overline{\Psi}_s}{\partial W^R} \Psi_s + \overline{\Psi}_s \frac{\partial \Psi_s}{\partial W^R}) \sum_{ss'} \overline{\Psi}_s \Psi_{s'} H_{ss'} + \\
&\frac{1}{Z} \cdot \sum_{ss'} (\frac{\partial \overline{\Psi}_s}{\partial W^R} \Psi_{s'} H_{ss'} + \overline{\Psi}_s \frac{\partial \Psi_{s'}}{\partial W^R} H_{ss'})
\end{aligned} \tag{5}$$

But wait!

denote $O_s^R = \frac{1}{\Psi_s}\frac{\partial \Psi_s}{\partial W^R}$ and $E_s = \frac{\sum_{s'} H_{ss'}\Psi_{s'}}{\Psi_s}$ we have

$$\begin{aligned}\frac{\partial E}{\partial W^R} &= -\frac{1}{Z^2}\cdot\sum_s(|\Psi_s|^2\overline{O}_s^R + |\Psi_s|^2 O_s^R)\sum_{ss'}|\Psi_s|^2 E_s + \\ &\frac{1}{Z}\cdot\sum_s|\Psi_s|^2\overline{O}_s^R E_s + |\Psi_s| O_s^R \overline{E}_s \\ &= Re[\langle E\overline{O^R}\rangle - \langle E\rangle\langle\overline{O^R}\rangle]\end{aligned} \tag{6}$$

since the norm of each amplitude of a quantum state means the probability, we can write its energy in the form of a statistical quantity, which allows us to produce a noisy quantity of our target and its gradient at each step by sampling.

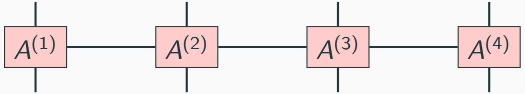
In fact, it is the quantum physics version of batched gradient descent.

Quantum configuration $\Rightarrow$ a Data sample
The target quantum state (eg. ground state) $\Psi \Rightarrow P_{real}$ real distribution function of the data set
ansatz $\Rightarrow$ P_{approx}

Tensor Networks

Tensor Network Ansatz

(a) Matrix Product State



(b) Matrix Product Operator

In fact, the nature is simple, no entire Hilbert space is needed in our calculation. Physisist only cares about a small subspace, which could be low entangled. To describe such subspace of Hilbert space, we will need to define ansatz in a smaller space with some parameters. Among all the possible solution, tensor network is one of the most sucess methods.

Tensor Network Ansatz

$$S(L) \leq 4L \log D \tag{7}$$

where $S(L) = -tr(\rho_{in} \log \rho_{in})$ is the entanglement entropy, and L is the length of the block.

Tensor network offers a way to describe target quantum state's entanglement. The entanglement entropy can be described by the legs, eg. in Projective Entangled Pair States (PEPS), which is

Why tensor networks?

- Analyse quantum states in local terms: practical complexity
- Tensor networks connects entanglement to geometry

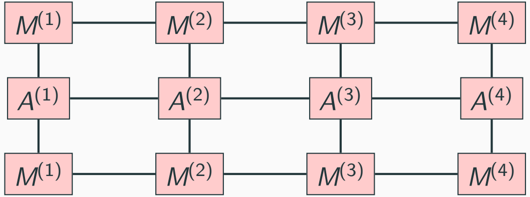


Figure 1: Calculating Energy by MPS and MPO

Tensor networks offers a practical approach to **analyse quantum states in local terms**, which reduces the calculations. eg. for Matrix Product States, calculating its energy by the contracting matrix product states and matrix product operators can reduce its complexity to polynomial.

Tensor networks connects entanglement to geometry. The tensor network diagram make it possible to connect a quantum state to a geometric structure, which leads to a novel approach to quantum gravatity. We are able to connect quantum mechanics to gravatity by constructing tensor networks on holographic planes[6, 7].

Why tensor networks?

- Analyse quantum states in local terms: practical complexity
- Tensor networks connects entanglement to geometry

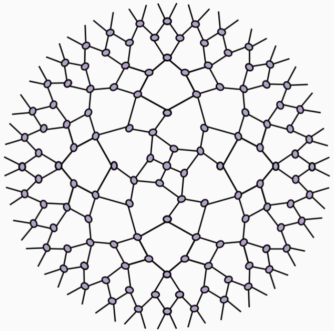


Figure 2: Holographic Tensor Networks (MERA)

Tensor networks offers a practical approach to **analyse quantum states in local terms**, which reduces the calculations. eg. for Matrix Product States, calculating its energy by the contracting matrix product states and matrix product operators can reduce its complexity to polynomial.

Tensor networks connects entanglement to geometry. The tensor network diagram make it possible to connect a quantum state to a geometric structure, which leads to a novel approach to quantum gravatity. We are able to connect quantum mechanics to gravatity by constructing tensor networks on holographic planes[6, 7].

Computer Vision

- classifying pictures/videos
- feature detection
- detecting other informations, eg. depth

Large space: A 32×32 **image space can be as large as** $2^{32\times 32}$ **for black-white images.**

Common problem:

how describe a pattern in a dimension cursed space.

One of the central problem of computer vision is how to describe the probability distribution of a picture with certain feature. eg. classifying handwrite digits. And however, the image space can also be large, and also has so called dimension-curse. A 32×32 image space can be as large as $2^{32\times 32}$ for black-white images.

If we consider the wave function as a kind of probability distribution, the problem of computer vision can be similar to quantum many-body physics: **how describe a pattern in a dimension cursed space**.

Computer Vision

- classifying pictures/videos
- feature detection
- detecting other informations, eg. depth

Large space: A 32×32 **image space can be as large as** $2^{32\times 32}$ **for black-white images.**

Common problem:

how describe a pattern in a dimension cursed space.

One of the central problem of computer vision is how to describe the probability distribution of a picture with certain feature. eg. classifying handwrite digits. And however, the image space can also be large, and also has so called dimension-curse. A 32×32 image space can be as large as $2^{32\times 32}$ for black-white images.

If we consider the wave function as a kind of probability distribution, the problem of computer vision can be similar to quantum many-body physics: **how describe a pattern in a dimension cursed space**.

Deep Learning

Figure 3: MINIST Handwrite Digits

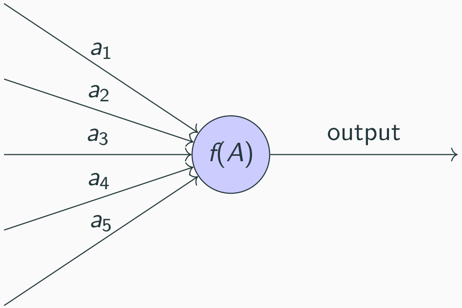
Recent progress in deep learning answers previous questions in a different manner (not precisely): by decomposing the problem to nested functions, while tensor networks decompose the space to subspaces. And most importantly, deep learning improves the precision greatly in many computer vision competitions like ImageNet[8, 9].

Deep Learning

Figure 4: ImageNet Competetion

Recent progress in deep learning answers previous questions in a different manner (not precisely): by decomposing the problem to nested functions, while tensor networks decompose the space to subspaces. And most importantly, deep learning improves the precision greatly in many computer vision competitions like ImageNet[8, 9].

Artificial Neurons



(8)

Deep Learning is actually the deep version of artificial neural networks (ANN) or sometimes just neural networks (NN). ANN is built on artificial neurons

Neural Networks

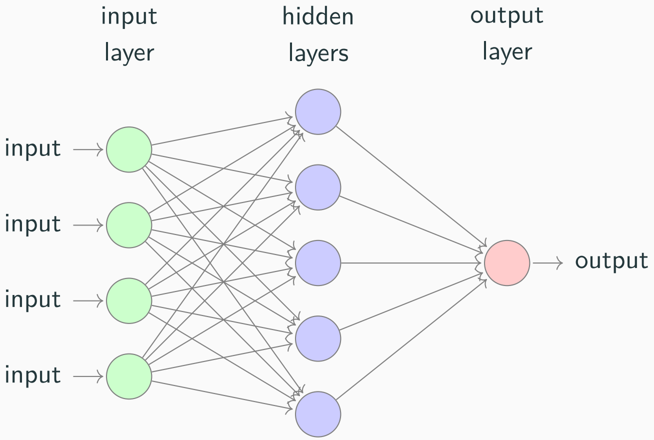


Figure 5: A feed forward neural network

Neural networks are the result of packing neurons, or it is the communication network of neurons. One specific type of neural network is the feed forward neural network, which consist of an input layer, several hidden layers, and an output layer.

Neural Networks

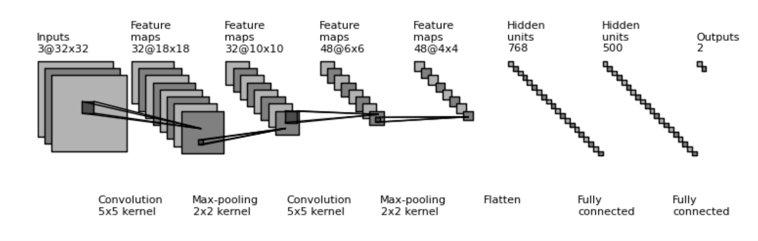


Figure 6: Convolutional Neural Network

However, simple feed forward neural networks does not work well comparing to other computer vision methods. And for video classification and other tasks which contains time series, simple feed forward cannot handle them well. New models like Convolutional Neural Networks, Recurrent Neural Networks, etc. was developed[10].

Neural Networks

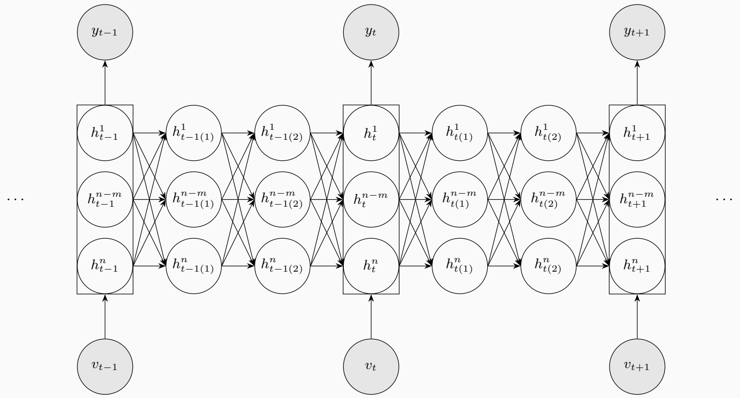


Figure 7: Recurrent Neural Network

However, simple feed forward neural networks does not work well comparing to other computer vision methods. And for video classification and other tasks which contains time series, simple feed forward cannot handle them well. New models like Convolutional Neural Networks, Recurrent Neural Networks, etc. was developed[10].

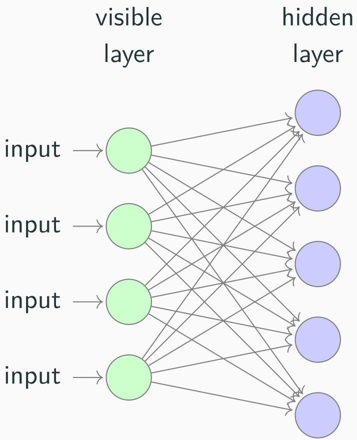


Figure 8: Restricted Boltzmann Machine

Moreover, inputs and ouputs of neurons can be stochastic. Boltzmann Machine is one of such stochastic neural networks. And a Boltzmann Machine is also called energy based model. Based on previous deterministic design of neural networks, there are also convulutional boltzmann machines.

Neural Network Ansatz

Neural Network Ansatz

- how to use neural network to represent quantum states.
- how to use neural network to represent the dynamics of quantum states. (operators)

As we have noticed in previous frames. Problems for computer vision and some other problems in artificial intelligence share something in commom with quantum many-body physics. Therefore, how about we simple use neural networks like a black box in quantum physics like they do in AI? To accomplish this goal two problems need to be done.

Neural Network Ansatz

- how to use neural network to represent quantum states.
- how to use neural network to represent the dynamics of quantum states. (operators)

to calculate physical quantities or a certain quantum state like ground state like we do previously in tensor networks, we need to decompose a quantum state into the form of neural network.

Neural Network Ansatz

- how to use neural network to represent quantum states.
- how to use neural network to represent the dynamics of quantum states. (operators)

to simulate dynamics of quantum states fully in neural networks, we need to find how to represent operators in neural networks.

Feedforward Neural Network States

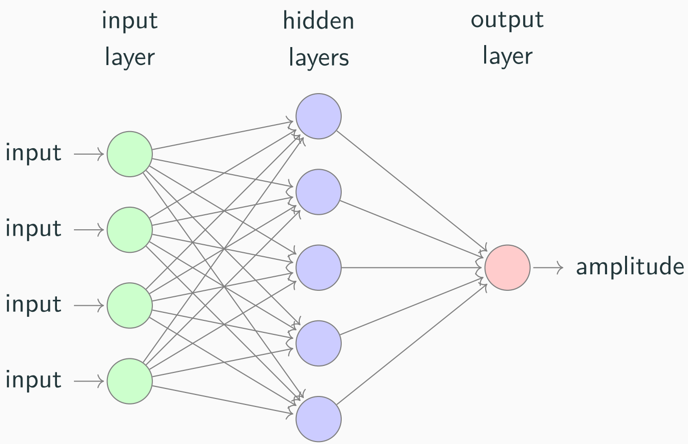


Figure 9: Multi-layer Perceptron State

Feedforward Neural Network or the Multi-layer Perceptron is the simplest model of neural networks, and it is connecting each neuron together.

Stochastic Version: Restricted Boltzmann Machine States

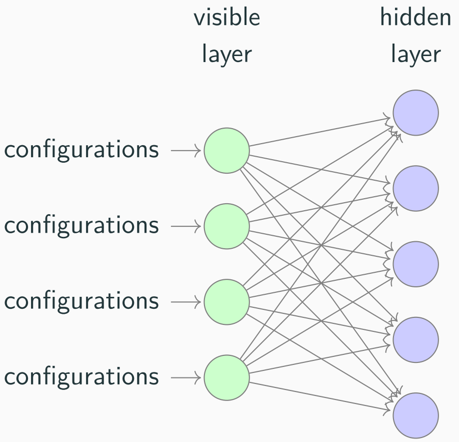


Figure 10: Restricted Boltzmann Machine

RBM quantum state is the so-called neural network quantum state in previous Science paper. [11] It is actually the string bond state or is the one layer term of multi-layer perceptron state with sigmoid activation. The disadvantage of this model is that it is hard to train and may have numerical instability.

RBM Ansatz

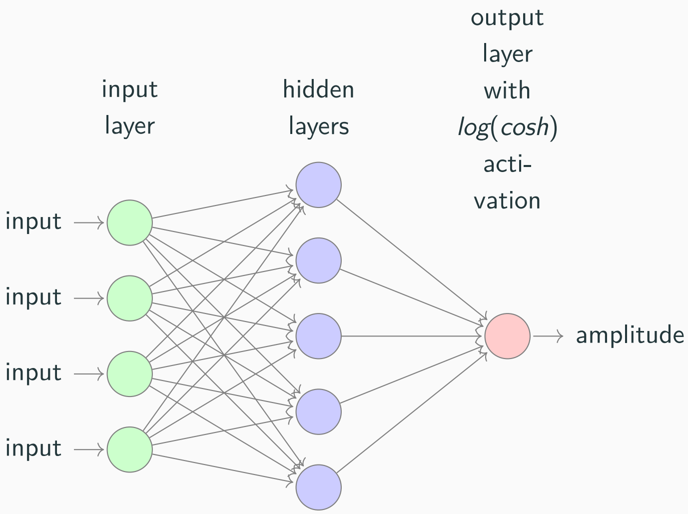


Figure 11: An equivalent MLP State

It is actually equivalent to a multi-layer perceptron state.

Preserve Symmetry: Convolutional Neural Network States

$$\Psi(S,\mathcal{W}) = \sum_{h_{i,s}} exp\{\sum_{f}^{\alpha} a^s \sum_{s}^{S}\sum_{j}^{N} \hat{\sigma}_j^z(s) + \sum_{f}^{\alpha} b^{(s)} \sum_{s}^{S} h_{f,s} + \sum_{f}^{\alpha}\sum_{s}^{S} h_{f,s} \sum_{j}^{N} W_j^{(f)} \sigma_j^z(s)\} \tag{9}$$

In previous work, sysmmetry is protected by shared weights, and in general in deep learning, this can be achieved by convolutional neural networks. The method in science paper is a kind of convolutional neural network in boltzmann machine with periodic boundary condition. The term of shared weights or convolution can be interpreted as capturing local features in the language of machine learning. Or to capture local entanglement structure.

Back to Convolutional Neural Networks

Convolutional Neural Networks usually has two components:

- Convolution layer
- Pooling Layer

Convolutional layer and pooling layer offers a kind of prior knowledge to the network, which improves the efficiency and accuracy. Moreover, symmetries like translation invarance are kept in such information processing.

CNN provides prior knowledge of images, features in images should not change when it translates. eg. A picture of an apple is a picture of an apple no matter where is the apple in the image. In quantum many-body problems, we could interpret it as the low entanglement entropy in real Hamiltonians, which could be samilar to images: An the pattern in an image usually corresponding to a small area of pixels (which in quantum physics, is the spin configurations).

Neural Tensor Network

Neural Tensor

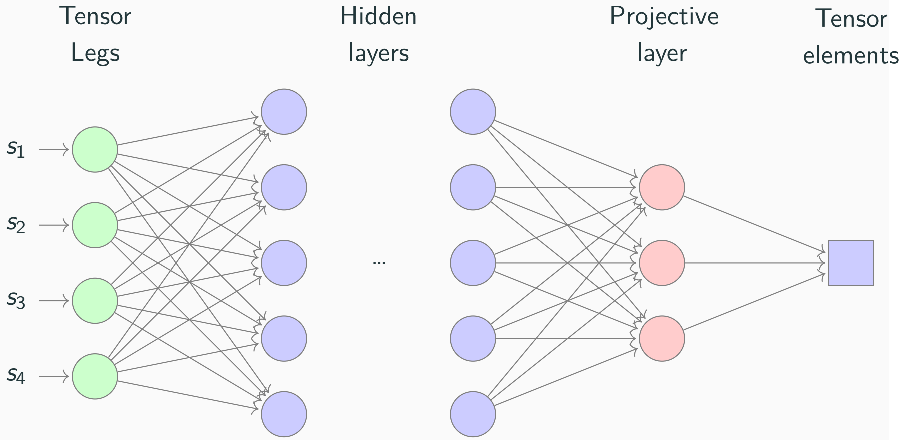


Figure 12: (Deep) Neural Tensor Network Ansatz

In fact, the study of tensor network has already offers an elegant mathematical language to describe the entanglement structure. Why not use it as a prior knowledge to construct a such neural network? neural tensor network, which parameterizes tranditional tensor network methods with neural networks. Although, this cannot solve the bound problem of tensor networks. It could help us design the structure of neural network ansatz based on quantum state's entanglement information, by sharing weights between symmetric tensors in tensor networks.

Time evolution and layer-wised SR method (Imaginary time evolution)

Unlike ground state problem, time evolution must strictly follows the Shordinger equation. The variational version of Shordinger equation can be write as follows:

$$S_H[\overline{\Psi}, \Psi] = \int_{-\infty}^{\infty} (\frac{i}{2}\langle\Psi(t)|\dot{\Psi}(t)\rangle - \frac{i}{2}\langle\dot{\Psi}(t)|\Psi(t)\rangle - \langle\Psi(t)|\hat{H}(t)|\Psi(t)\rangle)dt \tag{10}$$

for each layer we try to optimize

$$\min_{\mathcal{W}} \left\| i\frac{d}{dt}|\Psi_{\mathcal{W}}\rangle - \mathcal{H}|\Psi_{\mathcal{W}}\rangle \right\| \geq 0 \tag{11}$$

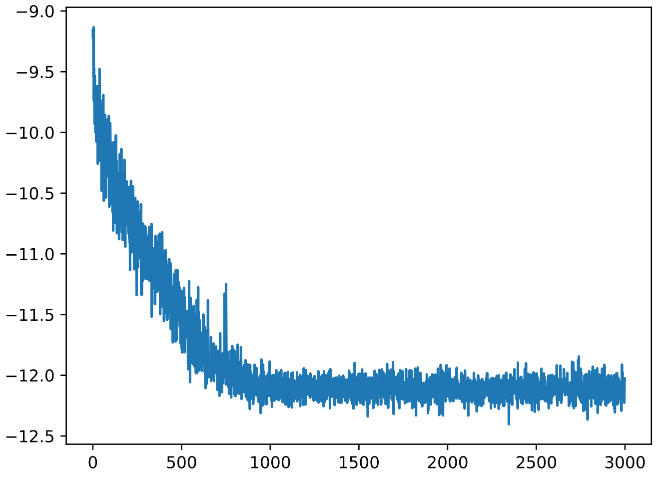


Figure 13: 1-D Traverse Ising Field (10 spins)

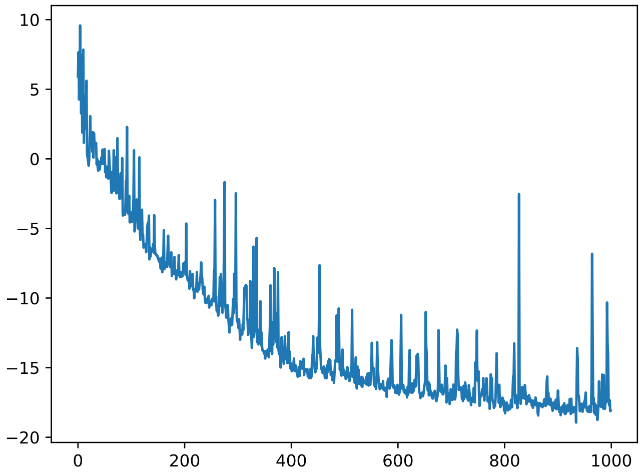


Figure 14: J1J2 Model(3x3,2D)

unfortunately, due to time limitation, we do not have much numerical results. However, for numerical evidence of the representation power of MLPS, there is a study on arxiv just before I finished my bachelor thesis. [12] Although, it shows some evidence about the representation power, it is not a practical model/algorithm.

Neural Network Ansatz: a different language

The representation power of Boltzmann machine ansatz (aka shallow feedforward neural networks) was studied by Xun Gao [13]. RBM can represent a wide class of many-body entangled states

- graph state

The complexity of calculating a RBM quantum state is mn, in computational complexity class $P/poly$.

Neural Network Ansatz: a different language

The representation power of Boltzmann machine ansatz (aka shallow feedforward neural networks) was studied by Xun Gao [13]. RBM can represent a wide class of many-body entangled states

- graph state
- toric codes

The complexity of calculating a RBM quantum state is mn, in computational complexity class $P/poly$.

Neural Network Ansatz: a different language

The representation power of Boltzmann machine ansatz (aka shallow feedforward neural networks) was studied by Xun Gao [13]. RBM can represent a wide class of many-body entangled states

- graph state
- toric codes
- violating entanglement area law (can be volume-law for long-range entanglement)[]

The complexity of calculating a RBM quantum state is mn, in computational complexity class $P/poly$.

Theorem

RBM cannot efficiently represent universal quantum computational states, PEPS, and ground states of k-local Hamiltonians unless the polynomial hierarchy collapses.

However, Xun Gao found a quantum state that is in $\#P - \textit{hard}$ and is a PEPS (a area-law tensor network state), can be generated by a polynomial size quantum circuit, the ground stateof a gapped 5-local Hamiltonian. So if RBM can efficiently represents the above states, it means polynomial hierarchy collapses(something like P=NP)

Theorem

DBM can efficiently represent all the states listed above. Any quantum state of n qubits generated by a quantum circuit of depth T can be represented exactly by a sparse DBM with O(nT) neurons

Theorem

Tensor network state with bond dimension D, maximum coordination number d, and n local tensors, can be represented efficiently by a sparse DBM with $O(nD^{2d})$ neurons.

neural network can be a detailed structure of tensor networks

Theorem

The ground state of any k-local Hamiltonian can be represented by a sparse DBM with neuron number $O(\frac{1}{\Delta}(n + log(\frac{1}{\epsilon}))m^2)$

However, this is not a practical model. Since it is not possible to calculate the sum of all hidden neurons efficiently (can be exponential, i.e $O(2^m)$, m is the number of hidden neurons)

Practically...

$$
\begin{aligned}
\Phi(S) = \sum_{h^k \in \{-1,1\}} e^{-H} &= e^{\sum_i \alpha_i v_i + \sum_k \sum_{j_k} b^k_{j_k} h^k_{j_k} + \sum_{k=1}^{l-1} \sum_{j,m} h^k_j W^{k+1}_{j,m} h^k_m + \sum_{i,j} v_i W_{ij} h^1_j} \\
&= e^{\alpha \mathbf{v}} \cdot \left(\prod_{k=1}^{l} \prod_{j=1}^{n_k} e^{b^k_j h^k_j}\right) \cdot \left(\prod_{k=1}^{l-1} \prod_{j=1}^{n_k} \prod_{m=1}^{n_{k+1}} e^{h^k_j W^{k+1}_{j,m} h^{k+1}_m}\right) \\
&\cdot \left(\prod_{i=1}^{m} \prod_{j=1}^{n_1} e^{v_i W_{ij} h^1_j}\right) \\
&= e^{\alpha v} \cdot \left(\prod_{j=1}^{n_1} e^{b^1_j h^1_j}\right) \left(\prod_{k=2}^{l} \prod_{j=1}^{n_k} e^{b^k_j h^k_j}\right) \left(\prod_{j=1}^{n_1} \prod_{m=1}^{n_2} e^{h^1_j W^2_{jm} h^2_m}\right)
\end{aligned} \tag{12}
$$

$$
\begin{aligned}
&\Big(\prod_{k=2}^{l-1}\prod_{j=1}^{n_k}\prod_{m=1}^{n_{k+1}} e^{h_j^k W_{ij}^{k+1} h_m^{k+1}}\Big)\Big(\prod_{i=1}^{m}\prod_{j=1}^{n_1} e^{v_i W_{ij} h_j^1}\Big)\\
&= e^{\alpha v}\cdot\Big(\prod_{j=1}^{n_1} e^{b_j^1 h_j^1}\Big)\Big(\prod_{j=1}^{n_1}\prod_{m=1}^{n_2} e^{h_j^1 W_{jm}^2 h_m^2}\Big)\Big(\prod_{i=1}^{m}\prod_{j=1}^{n_1} e^{v_i W_{ij} h_j^1}\Big)\\
&\Big(\prod_{k=2}^{l}\prod_{j=1}^{n_k} e^{b_j^k h_j^k}\Big)\Big(\prod_{k=2}^{l-1}\prod_{j=1}^{n_k}\prod_{m=1}^{n_{k+1}} e^{h_j^k W_{ij}^{k+1} h_m^{k+1}}\Big)\\
&= e^{\alpha v}\cdot\prod_{j=1}^{n_1} e^{b_j^1 h_j^1+\sum_{m=1}^{n_2} h_j^1 W_{jm}^2 h_m^2+\sum_{i=1}^{m} v_i W_{ij} h_j^1}\\
&\Big(\prod_{k=2}^{l}\prod_{j=1}^{n_k} e^{b_j^k h_j^k}\Big)\Big(\prod_{k=2}^{l-1}\prod_{j=1}^{n_k}\prod_{m=1}^{n_{k+1}} e^{h_j^k W_{ij}^{k+1} h_m^{k+1}}\Big)
\end{aligned}
\tag{13}
$$

Seems not possibile to use as an ansatz at least on classical computer

Open Problems

Holographic Neural Networks

Possible Advantages:

- Flexity for constucting different topological structure for different spaces
- Fixed network structure may help in time evolution of Ads/CFT

but it could be hard to find out weights

Like tensor networks, is it possible to construct a neural network on holographic plane, which connects to quantum gravity? Just before my bachelor theis was finished, there is an article about using the renormalization group in Boltzmann machine to construct such geometric sturcture[14]. However, they did not show how to calculate the Renyi entropy in such neural networks. And if the activation function is $log(cosh)$, a shallow neural network is equivallent to a RBM ansatz, which was previously proved possible to be converted to tensor networks.

Memories and Entanglement

In deep learning: **Memories could emerge from network structure**. [15] In our work: **different neural network sturcture may represent different entanglement**

Is it possible to connect them by simple numerical calculation? It is already possible to try with current utilities. (like long-range interaction)

Interacting Fermions in 2-D

In principle, application to this question can already be anticipated. (But not good at present...)

PyTorch for Quantum Physics

An extension based on pytorch is under early development. Some plans for the extension:

- complex neural network support (done for CPU version, tests needed)
- a hamiltonian gym (comparing to machine learning community's datasets, i.e MINIST, IMAGENET, OpenAI Gym)
- tensor network support

Acknowledgments

We would like to thank Zhen Wang, Shaojun Dong, Huiling Zhen, Zhe Chen, Lei Wang, Song Chen and Guiseppe Carleo for their helpful insights.

The End

The End

- P. W. Anderson *et al.*, “More is different,” *Science*, vol. 177, no. 4047, pp. 393–396, 1972.
- T. J. Osborne, “Hamiltonian complexity,” *Reports on Progress in Physics*, vol. 75, no. 2, p. 022001, 2012.
- A. D. Bookatz, “Qma-complete problems,” *arXiv preprint arXiv:1212.6312*, 2012.
- D. Aharonov and L. Eldar, “On the complexity of commuting local hamiltonians, and tight conditions for topological order in such systems,” in *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pp. 334–343, IEEE, 2011.
- J. Kempe, A. Kitaev, and O. Regev, “The complexity of the local hamiltonian problem,” *SIAM Journal on Computing*, vol. 35, no. 5, pp. 1070–1097, 2006.

References II

- B. Swingle, "Constructing holographic spacetimes using entanglement renormalization," *arXiv preprint arXiv:1209.3304*, 2012.
- G. Evenbly and G. Vidal, "Tensor network states and geometry," *Journal of Statistical Physics*, vol. 145, no. 4, pp. 891–918, 2011.
- O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

References III

- G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *arXiv preprint arXiv:1606.02318*, 2016.
- Z. Cai, "Approximating quantum many-body wave-functions using artificial neural networks," *arXiv preprint arXiv:1704.05148*, 2017.
- X. Gao and L.-M. Duan, "Efficient representation of quantum many-body states with deep neural networks," *arXiv preprint arXiv:1701.05039*, 2017.
- W.-C. Gan and F.-W. Shu, "Holography as deep learning," *arXiv preprint arXiv:1705.05750*, 2017.
- J. Ba, G. E. Hinton, V. Mnih, J. Z. Leibo, and C. Ionescu, "Using fast weights to attend to the recent past," in *Advances In Neural Information Processing Systems*, pp. 4331–4339, 2016.