Intro to NQS (updating)
Xingyu Ren

1. Variational Quanatum State

A quantum many-body state $|\psi\rangle$ can be represented by its wavefunction

$$|\psi\rangle = \sum_s \psi(s)|s\rangle \tag{1}$$

on a complete orthonormal basis $\left\{|s\rangle\right\}$. The essential problem of simulating a quantum many-body system is that the expentially increasing Hilbert space dimension (2^N for qubit systems) with the system size N.

Given the Hamiltonian $\hat{H}$ of the system, one of the central concern is to find the ground state $|GS\rangle$ of this system

$$\hat{H}|GS\rangle = E_{min}|GS\rangle, \tag{2}$$

where E_{min} is the minimal eigenvalue of the Hamiltonian. The reason we care about the ground state is that in the zero temperature, all quantum systems remains in the ground state, in which the quantum effect is very strong such that many exotic phenomina emerge.

The idea of variational ansatz is to assume that the wavefucntion $\psi(s)$ has some specific structure such that it can be parametrized by some parameters θ

$$(\theta, s) \mapsto \psi_\theta(s) = \langle s|\psi_\theta\rangle. \tag{3}$$

Usually, if the number of independent parameters θ is less than the Hilbert space dimesion 2^N so that this wavefunction can not represent the most general states in the Hilbert space. Here is the point: Most low-lying physical states we care about have very spacial structure such that they only occupy exponentially small parts in the whole Hilbert space. If the subspace spand by a variational state covers the low-lying physical states, then this variational ansatz can represent the physics very well in a much smaller subspace instead of the whole Hilbert space.

1.1. Estimating Observables

Recent years we have witnessed the fast development in the machine learning community, which demonstrates the power of deep neural networks. To make the neural network deep, we need a large amont of parameters θ. This will bring some difficalties in the calculation. To fix this problem, we have to make some approximation.

The quantum expectation value of an operator $\hat{A}$ on a non-normalized pure state $|\psi\rangle$ can be written as a classical expectation value $\mathbb{E}[\tilde{A}]$ over the Born distribution $\rho(s) \propto |\psi(s)|^2$

$$\langle \hat{A} \rangle = \frac{\langle \psi | \hat{A} | \psi \rangle}{\langle \psi | \psi \rangle} = \sum_s \frac{|\psi(s)|^2}{\langle \psi | \psi \rangle} \tilde{A}(s) = \sum_s p(s) \tilde{A}(s) = \mathbb{E}[\tilde{A}], \tag{4}$$

where $\tilde{A}$ is the local estimator

$$\tilde{A}(s) = \frac{\langle s | \hat{A} | \psi \rangle}{\langle s | \psi \rangle} = \sum_{s'} \frac{\psi(s')}{\psi(s)} \langle s | \hat{A} | s' \rangle. \tag{5}$$

Note that even though the sum in (5) runs over the whole Hilbert space basis, it can still be efficiently computed if the operator $\hat{A}$ is sparse enough. However, the sum in (4) is too hard to calculate such that the classical expectation value $\mathbb{E}[\tilde{A}]$ must be estimated by averaging over a sequence $\{s_i\}_{i=1}^{N_s}$ of configurations distribured according to the Born distribution $\rho(s) \propto |\psi(s)|^2$

$$\mathbb{E}[\tilde{A}] \approx \frac{1}{N_s} \sum_{i=1}^{N_s} \tilde{A}(s_i). \tag{6}$$

Given the derivatives of the log-amplitudes

$$O_i(s) = \frac{\partial \ln \psi_\theta(s)}{\partial \theta_i}, \tag{7}$$

the force vector can be defined as the covariance

$$\tilde{f}_i = \mathrm{Cov}[O_i, \tilde{A}] = \mathbb{E}\Big[O_i^* (\tilde{A} - \mathbb{E}[\tilde{A}]) \Big]. \tag{8}$$

If the parameters $\theta_i \in \mathbb{R}$ are real, the gradients of the expectation value are

$$\frac{\partial \langle \hat{A} \rangle}{\partial \theta_i} = 2\Re[\tilde{f}_i]. \tag{9}$$

If the parameters $\theta_i \in \mathbb{C}$ and the mapping $\theta_i \mapsto \psi_\theta(s)$ is complex differentiable (holomorphic), the gradients are

$$\frac{\partial \langle \hat{A} \rangle}{\partial \theta_i^*} = \tilde{f}_i. \tag{10}$$

In the case of a non-holomorphic mapping, the real part $\Re[\theta_i]$ and the imaginary part $\Im[\theta_i]$ can be treated independly as two real parameters.

1.2. Quantum Geometric Tensor

The Fubini-Study distance is defined as

$$\mathrm{d}(\psi,\phi)=\cos^{-1}\sqrt{\frac{\langle\psi|\phi\rangle\langle\phi|\psi\rangle}{\langle\psi|\psi\rangle\langle\phi|\phi\rangle}}, \tag{11}$$

which is the natral and gauge-invariant distance between two pure states $|\psi\rangle$ and $|\phi\rangle$. The Quantum Geometric Tensor (QGT) of a pure state is the metric tensor induced by the Fubini-Study distance. Assuming a infinitesimal change $\delta\theta$ of the parameters θ, then the Fubini-Study distance is $\mathrm{d}(\psi_\theta,\psi_{\theta+\delta\theta})=(\delta\theta)^\dagger G(\delta\theta)$, where G is the QGT.

For a holomorphic mapping $\theta\mapsto|\psi_\theta\rangle$, the QGT is given by

$$G_{ij}(\theta)=\frac{\langle\partial_{\theta_i}\psi_\theta|\partial_{\theta_j}\psi_\theta\rangle}{\langle\psi|\psi\rangle}-\frac{\langle\partial_{\theta_i}\psi_\theta|\psi_\theta\rangle\langle\psi_\theta|\partial_{\theta_j}\psi_\theta\rangle}{\langle\psi|\psi\rangle^2}, \tag{12}$$

where $\langle x|\partial_{\theta_i}\psi_\theta\rangle:=\partial_{\theta_i}\langle x|\psi_\theta\rangle$. The above QFT formula can also be rewritten as a classical covariance with respect to the Born distribution $\rho(s)\propto|\psi(s)|^2$

$$G_{ij}(\theta)=\mathrm{Cov}[O_i,O_j]=\mathbb{E}\Big[O_i^*(O_j-\mathbb{E}[O_j])\Big]. \tag{13}$$

The QGT or its stochastic estimate is also commonly known as the S Matrix or Quantum Fisher Matrix (QFM).

1.3. Stochastic Reconfiguration

Once the gradients are calculated, we can update the parameters using the simple Stochastic Gradient Descent (SGD) optimizer

$$\theta_i\mapsto\theta_i-\eta f_i. \tag{14}$$

However, this optimizer is not so efficient. A more efficient way is called Stochastic Reconfiguration (SR), where the gradient is preconditioned by solving the linear equations

$$\sum\nolimits_j\Re[G_{ij}]\delta_j=f_i=2\Re[\tilde{f}_i] \tag{15}$$

for real parameters, or

$$\sum\nolimits_j G_{ij}\delta_j=f_i \tag{16}$$

for complex holomorphic functions.

1.4. Zero-Variance Property

To find the ground state $\left|GS\right\rangle$ of the Hamiltonian in (2), We may naively calculate the energy expectation value $E_\theta = \left\langle\psi_\theta\middle|\hat{H}\middle|\psi_\theta\right\rangle$ of the variational state and minimize E_θ with respect to θ. However, there is no guarantee that the final state $\left|\psi_{\theta_f}\right\rangle$ such that E_f is minimal is the eigenstate of the Hamiltonian, i.e., $\hat{H}\left|\psi_{\theta_f}\right\rangle \neq E_f\left|\psi_{\theta_f}\right\rangle$.

The answer to the above problem is an important feature of the variational Monte Carlo called the zero-variance property. If a variantional state $\left|\psi_\theta\right\rangle$ is an eigenstate of the Hamiltonian $\hat{H}\left|\psi_\theta\right\rangle = E\left|\psi_\theta\right\rangle$, then the local energy (set $\hat{A} = \hat{H}$ in (5)) becomes a constant

$$\tilde{H}(s) = \frac{\left\langle s\middle|\hat{H}\middle|\psi_\theta\right\rangle}{\left\langle s\middle|\psi_\theta\right\rangle} = E, \tag{17}$$

i.e., $\tilde{H}(s) = E$ does not depend on the compurational basis $\left|s\right\rangle$. This immediately implies that its variance is zero. Clearly, this is an extreme case that is very rare for generic correlated models. However, in general, the variance of the local energy $\tilde{H}(s)$ will decrease whenever the variational state $\left|\psi_\theta\right\rangle$ approach an exact eigenstate. This fact is very important to reduce the statistical fluctuations and improve the numerical efficiency. The zero-variance property is a feature that exists only for quantum expectation values, while it is absent in classical calculations, where observables have thermal fluctuations.

The variance of the random variable $\tilde{H}(s)$ is exactly equal to the quantum variance of the Hamiltonian over the variational state

$$\sigma^2[\tilde{H}] = \frac{\left\langle\psi_\theta\middle|(\hat{H}-E)^2\middle|\psi_\theta\right\rangle}{\left\langle\psi_\theta\middle|\psi_\theta\right\rangle}. \tag{18}$$

In summary, we not only want to minimize the energy expectation value E_θ, we also have to minimize the variance $\sigma^2[\tilde{H}]$ to make sure that this state is close to an eigenstate.

2. Neural Quantum State

There are many well developed variation quantum states like Tensor Network states, Hatree-Fock states, Jastrow states and so on. The Neural Quantum State (NQS) is a variational ansatz where the input configuration s is passed into a neural network parametrized by θ resulting in the output wavefunction $\psi_\theta(s)$. Below we introduce several variational states.

2.1. PEPS

The Projected Entangled-Pair State (PEPS) is one of the most frequently used ansatz in the tensor network community. PEPS is a natrual generalization of the Matrix Product State (MPS). The advatage of these methods is that their entanglment entropies scale with the boundary of the system, which is the characteristic of most gaped ground states of local Hamlitonians. Though they are not considered as NQSs, we still introduce them here as a benchmark and competitor of NQS.

The MPS ansatz can be written as

$$\psi(s_1, s_2, \cdots, s_n) = \mathrm{Tr}\Big[A_1^{s_1} A_2^{s_2} \cdots A_n^{s_n}\Big], \tag{19}$$

where $A_i^{s_i}$ is a rank-3 tensor (in the bulk) with the physical leg labeled by s_i and the two inner legs form a matrix. The PEPS ansatz replaces the rank-3 tensor with a rank-5 tensor and forms a 2-dimensional tensor networks.

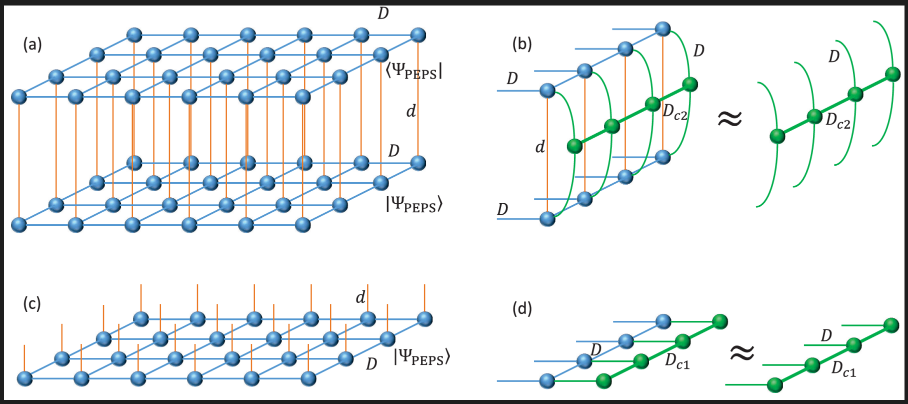

Figure 1: PEPS+VMC

(a) Exact contracion of a PEPS.
(b) Boundary-MPO method to reduce the computational complexity of (a).
(c) VMC can reduce from contracting (a) to a single layer PEPS with fixed physical legs.
(d) Boundary-MPS method to reduce the computational complexity of (c).

In princeple, we can directly contract a tensor network state to get the physical quantities. However, the size of the PEPS is usually too large to contract rigrously. Therefore, we need to introduce the VMC method and boundary-MPS method to reduce the computational complexity. Figure 1 shows how this works:

1. Introducing VMC can simplify the contraction process from (a) to (c).
2. Introducing the boundary-MPS method in (d), truncating the boud dimension of the green bold bond at a fixed value D_{c1} (ususally $D_{c1} \approx 2D$).

2.2. Transformer

The transformer architecher is the basis of Large Languiage Models (LLM). Due to the great success of ChatGPT, it has become one of the hottest neural networks recently. There are several new works utilizing the Visioin Transformer (ViT) to parametrize the wavefunctions [2–4], which will be reviewed below.

2.2.1. The origenal transformer

We first give a brief introduction to the original transformer architecher [5]. It contains an encoder block and a decoder block. The encoder processes the input sequence (e.g., a sentence in a source language) and generates a hidden representation summarizing the input information. The decoder iteratively processes both the encoder's output and the decoder's tokens, producing the final output sequence (e.g., a translation in a target language).

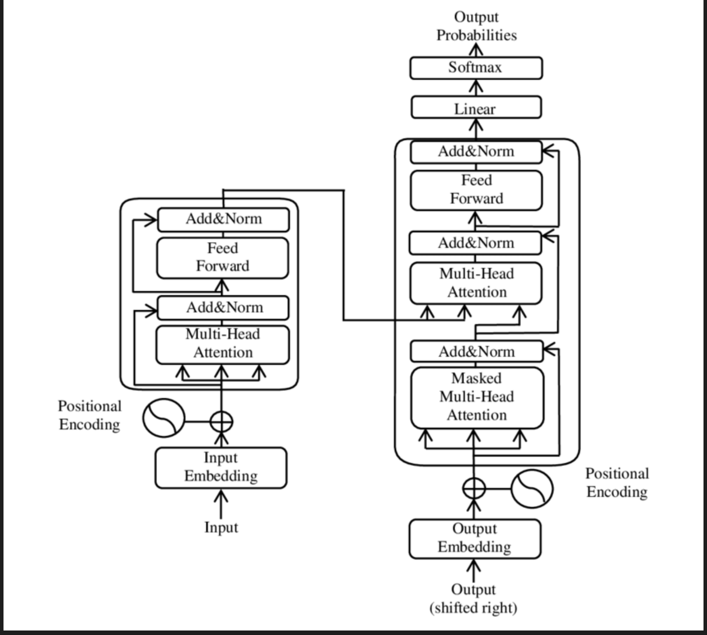

Figure 2: Encoder-decoder transoformer

The most important part of transformer is the attention mechanism[1]. The attention mechanism allows the decoder to focus on different parts of the encoder's output for each step of its own outputs. It computes a weight distribution (or attention scores) that determines the importance of each input element for each output. For each decoder step, the attention mechanism calculates a similarity score between the decoder's current hidden state and all

[1] the attention mechanism describes a weighted average of (sequence) elements with the weights dynamically computed based on an input query and elements' keys.

encoder hidden states. These similarity scores are used to compute the attention weights. The encoder's hidden states are then linearly combined using these weights to create a context vector. The context vector is used to guide the decoder's next prediction. Essentially, attention allows the model to focus on relevant parts of the input sequence, improving translation quality and handling long-range dependencies.

The JAX/FLAX code implementing transformer model can be found in [9].

2.2.2. Vision transformer wavefunction

The original transformer is aimed to deal with text. However, it can also be modified to Vision Transformer (ViT) to deal with pictures. It is important to note that the original transformer model is autoregressive, wihile the ViT is not! The wavefunction in [2–4] is parametrised using the following ViT architecher in Figure 3.

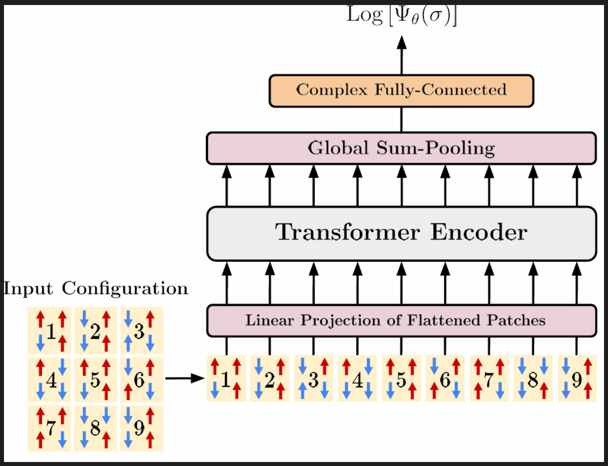

Figure 3: ViT wavefunction in 2D

As we can see, this is an encoder only model compared to the deconder only model in the mainstream LLMs. The encode block in Figure 4 is alomost the same as Figure 2 but with two main differences. The first is that now the input is patches of pictures instead of texts. The second thing to note is that the layer normalization before the attention block supports better gradient flow and removes the necessity of a warm-up stage.

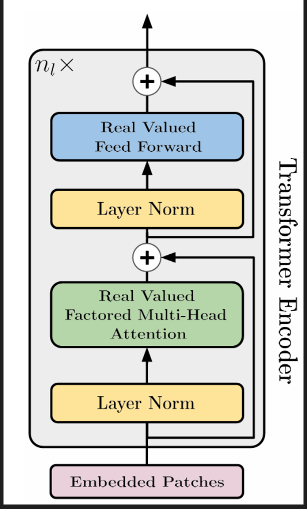

Figure 4: ViT encoder

The JAX/FLAX code implementing tViT model can be found in [10].

3. NetKet 3: A Python Package for NQS

After the seminal work of NQS [6], the community has developed a Python package called NetKet [7,8] to impliment this algorithm. The latest version of NetKet 3 [12] is based on the mechine learning package JAX/FLAX. It is easy to use. Here is a simple example of a Fully-connected Feedforward Neural Network (FFNN) for a one-dimensional J1-J2 spin chain [11] to demonstrate how to use this package.

We will make use of Google's Colab to run the code. Since for small system size, the CPU performance should be better than the GPU, we will run this code on Colab's free CPU. The code is comming soon...

[1] W.-Y. Liu, S.-J. Dong, Y.-J. Han, G.-C. Guo, and L. He, *Gradient Optimization of Finite Projected Entangled Pair States*, Phys. Rev. B **95**, 195154 (2017).

[2] R. Rende, L. L. Viteritti, L. Bardone, F. Becca, and S. Goldt, *A Simple Linear Algebra Identity to Optimize Large-Scale Neural Network Quantum States* (2023).

[3] L. L. Viteritti, R. Rende, A. Parola, S. Goldt, and F. Becca, *Transformer Wave Function for the Shastry-Sutherland Model: Emergence of a Spin-Liquid Phase* (2024).

[4] R. Rende, S. Goldt, F. Becca, and L. L. Viteritti, *Fine-Tuning Neural Network Quantum States* (2024).

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention Is All You Need* (2023).

[6] G. Carleo and M. Troyer, *Solving the Quantum Many-Body Problem with Artificial Neural Networks*, Science **355**, 602 (2017).

[7] F. Vicentini, D. Hofmann, A. Szabó, D. Wu, C. Roth, C. Giuliani, G. Pescia, J. Nys, V. Vargas-Calderón, N. Astrakhantsev, and G. Carleo, *NetKet 3: Machine Learning Toolbox for Many-Body Quantum Systems*, SciPost Phys. Codebases 7 (2022).

[8] F. Vicentini, D. Hofmann, A. Szabó, D. Wu, C. Roth, C. Giuliani, G. Pescia, J. Nys, V. Vargas-Calderón, N. Astrakhantsev, and G. Carleo, *Codebase Release 3.4 for NetKet*, SciPost Phys. Codebases 7 (2022).

[9] *Tutorial 6 (JAX): Transformers and Multi-Head Attention &#x2014; UvA DL Notebooks v1.2 Documentation — Uvadlc-Notebooks.Readthedocs.Io* (https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/JAX/tutorial6/Transformers_and_MHAttention.html, n.d.).

[10] *Tutorial 15: Vision Transformers &#x2014; UvA DL Notebooks v1.2 Documentation — Uvadlc-Notebooks.Readthedocs.Io* (https://uvadlc-notebooks.readthedocs.io/en/latest/tutorial_notebooks/tutorial15/Vision_Transformer.html, n.d.).

[11] *Ground-State: J1-J2 Model &#x2014; NetKet — Netket.Readthedocs.Io* (https://netket.readthedocs.io/en/latest/tutorials/gs-j1j2.html, n.d.).

[12] *NetKet - The Machine Learning Toolbox for Quantum Physics — Netket.Org* (https://www.netket.org/, n.d.).