

用神经网络量子态来求解多体哈密顿量的基态物理

幾 (Qamber)

2021 年 10 月 16 日

目录

1	神经网络量子态	2
2	物理量的计算: 变分蒙特卡洛采样	2
3	参数的更新规则: 随机重构方法	3
4	时间演化	4
5	示范代码	5

1 神经网络量子态

在这一节, 我们将介绍神经网络量子态 (Neural Network Quantum State) [Carleo and Troyer(2017)] 的概念, 并且详细说明如何用其来求解量子多体哈密顿量的基态问题.

对于一个局域 (local) 的多体哈密顿量 H , 我们一般可以将其分解为求和的形式:

$$H = \sum_i h_i + \sum_{i,j} h_{ij} + \cdots,$$

其中, h_i 仅涉及第 i 个自由度 (单体项), h_{ij} 仅涉及第 i, j 个自由度 (两体项), 其它项依此类推. 为了简单起见, 我们考虑一个有 N 个 $1/2$ 自旋的量子系统, 其量子态在计算基矢 (σ_i^z 表象) 下可以写为:

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

为了求解体系的基态, 我们可以采用变分原理, 即先提出波函数的一个拟设, 再通过调节其参数来寻找基态. 在这里, 我们采用神经网络来编码一个量子态的波函数, 为了简单起见, 我们仅考虑用限制玻尔兹曼机 (RBM) 来完成这个任务,

$$\begin{aligned} \Psi(\vec{\sigma}) &= \sum_{\vec{\chi}} \exp\left(\sum_i b_i \chi_i + \sum_j a_j \sigma_j + \sum_{i,j} W_{ij} \chi_i \sigma_j\right) \\ &= \exp\left(\sum_j a_j \sigma_j\right) \prod_i 2 \cosh\left(b_i + \sum_{j'} W_{ij'} \sigma_{j'}\right), \end{aligned} \quad (1)$$

为了能够编码波函数的振幅与幅角, 此处的网络参数 $\{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$ 均为复数并且为了准确地编码半整数的自旋, 我们选取 σ_j, χ_i 的取值为 $\{1, -1\}$, 分别对应了在 z 方向朝上与朝下的自旋. 需要注意的是, 在这里我们还省略了一个归一化的系数, 事实上, 这个归一化的系数并不会参与到后续的操作中, 物理量的计算和参数的更新所涉及到的采样只需要知道不同自旋构型间的相对权重, 归一化系数是自然会被消掉的.

2 物理量的计算: 变分蒙特卡洛采样

首先, 我们需要知道如何从神经网络量子态中提取我们需要的物理量. 考虑一个算符 \hat{O} (不一定要厄米性), 我们计算其在量子态 $|\Psi\rangle$ 下的期望值 (不一定为实数):

$$\langle \hat{O} \rangle = \frac{\langle \Psi | \hat{O} | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \sum_{\vec{\sigma}} \frac{|\Psi(\vec{\sigma})|^2}{\sum_{\vec{\sigma}''} |\Psi(\vec{\sigma}'')|^2} \sum_{\vec{\sigma}'} \frac{\Psi(\vec{\sigma}')}{\Psi(\vec{\sigma})} \langle \vec{\sigma} | \hat{O} | \vec{\sigma}' \rangle.$$

我们引入标记

$$\mathcal{O}_{\text{loc}}(\vec{\sigma}) = \sum_{\vec{\sigma}'} \frac{\Psi(\vec{\sigma}')}{\Psi(\vec{\sigma})} \langle \vec{\sigma} | \hat{O} | \vec{\sigma}' \rangle = \frac{\langle \vec{\sigma} | \hat{O} | \Psi \rangle}{\langle \vec{\sigma} | \Psi \rangle},$$

则上述的期望值表达式可以化简为:

$$\langle \hat{O} \rangle = \langle \mathcal{O}_{\text{loc}}(\vec{\sigma}) \rangle,$$

式中, $\langle \cdots \rangle$ 代表按相对概率分布 $|\Psi(\vec{\sigma})|^2$ 取平均. 这样一来, 我们就将物理量的计算 (在 N 比较大的时候, 直接的计算是不可能的) 转换成了一个依概率采样的问题. 显然, 这样的采样问题是我们所熟悉的, 我们从某个自旋构型 ($\vec{\sigma}_{\text{old}}$) 出发, 随机选取一个自旋将其反转, 从而得到一个新的自旋构型 ($\vec{\sigma}_{\text{new}}$), 接着以概率 $\min(1, |\Psi(\vec{\sigma}_{\text{new}})|^2 / |\Psi(\vec{\sigma}_{\text{old}})|^2)$ 来接受这个新构型. 通过这种采样方法, 我们可以得到一个满足相对概率分布 $|\Psi(\vec{\sigma})|^2$ 的自旋构型集 $\mathcal{S} = \{\vec{\sigma}\}$, 而算符 \hat{O} 对量子态的期望, 就转化为了对采样得到的自旋构型集的平均,

$$\langle \hat{O} \rangle \approx \frac{1}{|\mathcal{S}|} \sum_{\vec{\sigma} \in \mathcal{S}} \mathcal{O}_{\text{loc}}(\vec{\sigma}).$$

现在的关键是对其中的每个 $\vec{\sigma}$ 计算对应的局部量 $\mathcal{O}_{\text{loc}}(\vec{\sigma})$. 单纯看 $\mathcal{O}_{\text{loc}}(\vec{\sigma})$ 的定义, 我们会以为它将涉及非常巨大的计算 (毕竟需要遍历所有的构型 $\vec{\sigma}'$). 虽然在理论上, 这是一个不可能完成的任务, 但是我们实际所考虑的多体物理问题往往会大大简化这里的

计算, 从而保证这个计算的可行性. 一般而言, 我们所考虑的算符 $\hat{\mathcal{O}}$ 与我们的哈密顿量有类似的局域性, 即可以按照所作用的自旋自由度进行分解:

$$\hat{\mathcal{O}} = \sum_i \hat{o}_i + \sum_{i,j} \hat{o}_{ij} + \dots.$$

这样一来, 局部量的计算可以分解为:

$$\mathcal{O}_{\text{loc}}(\vec{\sigma}) = \sum_{\vec{\sigma}'} \frac{\Psi(\vec{\sigma}')}{\Psi(\vec{\sigma})} \langle \vec{\sigma} | \left(\sum_i \hat{o}_i + \sum_{i,j} \hat{o}_{ij} + \dots \right) | \vec{\sigma}' \rangle.$$

显然, 如果我们所考虑的算符 $\hat{\mathcal{O}}$ 仅涉及少体的相互作用 (对相互作用的短程与否没有要求), 那么局部量 $\mathcal{O}_{\text{loc}}(\vec{\sigma})$ 的计算就仍然是一个资源消耗不大的问题.

接着, 我们需要来考虑如何优化变分波函数 $\Psi(\vec{\sigma})$ 的问题. 为此, 我们采用随机重构 (Stochastic Reconfiguration) 的方法来确定变分波函数的参数更新规则.

3 参数的更新规则: 随机重构方法

假设我们的试探量子态 $|\phi\rangle$ 依赖于一组参数 $\vec{\alpha} = \{\alpha_1, \dots, \alpha_p\}$, 我们的目标是得到合适的参数取值使得量子态 $|\phi\rangle$ 尽可能地逼近哈密顿量 \hat{H} 的基态. 随机重构方法可以告诉我们应该如何逐步地更新我们的参数来达到我们的目标. 实际上, 随机重构方法不过是应用了无穷小虚时演化. 如果我们对一个与基态不正交的量子态 $|\phi\rangle$ 施加一个较长时间的虚时演化 ($t = -i\tau$), 我们可以预期得到的末态会很接近基态,

$$\lim_{\tau \rightarrow \infty} e^{-\hat{H}\tau} |\phi\rangle \propto |g\rangle.$$

随机重构方法不过是将这个较长的虚时演化拆解成了很多的短时演化 (优势在于能做线性化近似处理, 而不用直接处理算符的指数运算). 为此, 我们考虑将 $\exp(-\epsilon\hat{H}) \approx \mathbb{I} - \epsilon\hat{H}$ 作用在量子态 $|\phi\rangle$ 上, 并且相应地改变参数 $\vec{\alpha}$ 使得更新后的量子态尽可能地符合虚时演化. 因此, 我们有 $(|\phi\rangle_{\vec{\alpha}+\delta\vec{\alpha}} \approx |\phi\rangle_{\vec{\alpha}} + \delta\vec{\alpha} \cdot \nabla_{\vec{\alpha}} |\phi\rangle)$

$$\begin{aligned} (\mathbb{I} - \epsilon\hat{H})|\phi\rangle &= |\phi'\rangle = \delta\alpha_0|\phi\rangle + \sum_{k=1}^p \delta\alpha_k \frac{\partial|\phi\rangle}{\partial\alpha_k} \quad \text{其中, } \delta\alpha_0 \sim 1, \delta\alpha_k \sim \epsilon \\ &= \delta\alpha_0|\phi\rangle + \sum_{k=1}^p \delta\alpha_k \sum_{\vec{\sigma}} \frac{\partial\phi(\vec{\sigma})}{\partial\alpha_k} |\vec{\sigma}\rangle \\ &= \delta\alpha_0|\phi\rangle + \sum_{k=1}^p \delta\alpha_k \sum_{\vec{\sigma}} \frac{\partial \log \phi(\vec{\sigma})}{\partial\alpha_k} |\vec{\sigma}\rangle \langle \vec{\sigma} | \phi \rangle \\ &= \sum_{k=0}^p \delta\alpha_k \Delta_{\phi k} |\phi\rangle, \end{aligned} \tag{2}$$

式中

$$\Delta_{\phi k} := \begin{cases} \mathbb{I} & \text{若 } k = 0 \\ \sum_{\vec{\sigma}} \frac{\partial \log \phi(\vec{\sigma})}{\partial\alpha_k} |\vec{\sigma}\rangle \langle \vec{\sigma} | & \text{若 } k \neq 0 \end{cases}.$$

我们对上式两侧的左边共同作用 $\langle \phi | \Delta_{\phi k}$

$$\langle \phi | \Delta_{\phi k} \sum_{k'=0}^p \delta\alpha_{k'} \Delta_{\phi k'} |\phi\rangle = \langle \phi | \Delta_{\phi k} (\mathbb{I} - \epsilon\hat{H}) |\phi\rangle.$$

先考虑 $k = 0$ 的情形, 我们有

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

接着我们将 $\delta\alpha_0$ 的值代入 $k \neq 0$ 的情形

$$\langle \phi | \Delta_{\phi k} \left(1 - \epsilon \langle \hat{H} \rangle - \sum_{k'=1}^p \delta\alpha_{k'} \langle \Delta_{\phi k'} \rangle + \sum_{k'=1}^p \delta\alpha_{k'} \Delta_{\phi k'} \right) |\phi\rangle = \langle \phi | \Delta_{\phi k} (\mathbb{I} - \epsilon\hat{H}) |\phi\rangle$$

\Downarrow

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

通过定义矩阵 S 与向量 f ,

$$S_{kk'} \equiv \langle \Delta_{\phi k} \Delta_{\phi k'} \rangle - \langle \Delta_{\phi k} \rangle \langle \Delta_{\phi k'} \rangle \quad \text{及} \quad f_k \equiv \langle \hat{H} \rangle \langle \Delta_{\phi k} \rangle - \langle \Delta_{\phi k} \hat{H} \rangle,$$

参数的变化 $\delta \alpha_k$ 可以写成一个紧凑的形式

$$\sum_{k'=1}^p S_{kk'} \delta \alpha_{k'} = \epsilon f_k \quad \Rightarrow \quad \delta \vec{\alpha} = \epsilon S^{-1} f.$$

因此, 参数的更新规则可以写为:

$$\vec{\alpha}(n+1) \leftarrow \vec{\alpha}(n) + \eta S^{-1}(n) f(n), \quad (3)$$

式中的 η 是学习率, S^{-1} 是 S 的伪逆, 并且我们在 S 的基础上再加入一个微小的单位矩阵来正规化 (regularization) 它. 此处, 我们采用文献 [Carleo and Troyer(2017)] 的策略,

$$S(n) \leftarrow S(n) + \lambda(n) \mathcal{I} \quad \text{with } \lambda(n) = \max(100 \times 0.9^n, 10^{-4}),$$

$\lambda(n)$ 所涉及的参数 $(100, 0.9, 10^{-4})$ 皆可随需要进行调整. 显然, 随机重构的更新规则需要我们计算 S 与 f , 而这两个量的计算可以利用上一小节中的采样方案来实现. 为此, 我们给出相关局部量的表达式,

$$\begin{aligned} H_{\text{loc}}(\vec{\sigma}) &= \frac{\langle \vec{\sigma} | \hat{H} | \phi \rangle}{\langle \vec{\sigma} | \phi \rangle} = \sum_{\vec{\sigma}'} \frac{\phi(\vec{\sigma}')}{\phi(\vec{\sigma})} \langle \vec{\sigma} | \hat{H} | \vec{\sigma}' \rangle \\ (\Delta_{\phi k})_{\text{loc}}(\vec{\sigma}) &= \sum_{\vec{\sigma}'} \frac{\phi(\vec{\sigma}')}{\phi(\vec{\sigma})} \langle \vec{\sigma} | \sum_{\vec{\sigma}''} \frac{\partial \log \phi(\vec{\sigma}'')}{\partial \alpha_k} | \vec{\sigma}'' \rangle \langle \vec{\sigma}' | \vec{\sigma} \rangle = \frac{\partial \log \phi(\vec{\sigma})}{\partial \alpha_k} \\ (\Delta_{\phi k} \Delta_{\phi k'})_{\text{loc}}(\vec{\sigma}) &= \frac{\partial \log \phi(\vec{\sigma})}{\partial \alpha_k} \frac{\partial \log \phi(\vec{\sigma})}{\partial \alpha_{k'}} \\ (\Delta_{\phi k} \hat{H})_{\text{loc}}(\vec{\sigma}) &= \frac{\partial \log \phi(\vec{\sigma})}{\partial \alpha_k} H_{\text{loc}}(\vec{\sigma}), \end{aligned} \quad (4)$$

其中,

$$\begin{aligned} \frac{\partial \log \phi(\vec{\sigma})}{\partial a_j} &= \sigma_j \\ \frac{\partial \log \phi(\vec{\sigma})}{\partial b_i} &= \tanh \left(b_i + \sum_j W_{ij} \sigma_j \right) \\ \frac{\partial \log \phi(\vec{\sigma})}{\partial W_{ij}} &= \sigma_j \tanh \left(b_i + \sum_j W_{ij} \sigma_j \right). \end{aligned} \quad (5)$$

至此, 我们就集齐了拼图所需要的所有元素!

4 时间演化

除了求解多体哈密顿量的基态以外, 借助于随机重构的方法, 我们还能求解多体量子态的时间演化问题. 考虑到随机重构方法实际上是模拟了量子态的虚时演化, 我们只需要稍微更改一下上一节的参数更新规则就能实现量子态的实时演化,

$$\vec{\alpha}(t + \Delta t) \leftarrow \vec{\alpha}(t) + i \Delta t S^{-1}(t) f(t). \quad (6)$$

注意, 与上一节不同的是, 对于这里的矩阵 S , 我们不能再用任何的正规化程序, 并且每一次的参数更新的时间步长 Δt 要足够得短.

5 示范代码

在托管网站 GitHub 上, 我们提供了用神经网络量子态求解多体哈密顿量基态的示范代码: <https://github.com/Qqamber/RBM-GroundStateSolver>.

参考文献

[Carleo and Troyer(2017)] G. Carleo and M. Troyer, Science **355**, 602–606 (2017).