Neural-network Quantum States

李相　2019311574

2019 年 12 月 29 日

1　Introduction

本次课程作业的主要内容是重复了一篇题为"Solving the quantum many-body problem with artificial neural networks" 的文章中的部分内容，该文章于2017年2月发表在《科学》杂志上。文章的主要工作是利用受限玻尔兹曼机来表示多体波函数，在训练的过程中利用了蒙特卡罗采样，并基于能量最低原理采用Stochastic Reconfiguration方法更新参数。文章提出的方法适用于经典的1维和2维自旋模型以及时间演化过程，我在课程作业中仅实现了1维的横场Ising模型方案，并计算了基态能量以及$<\sigma_x>$与$<\sigma_z>$的值，并与第二次作业中实现的DMRG方法的结果做对照。代码的实现过程参考了文章的Supplementary Material和Github网站上的项目"zeldredge/py-nqs"。

2　Model Details

Restricted Boltzman Mechine

受限玻尔兹曼机的每一种状态都有对应的能量$E(\mathbf{v},\mathbf{h})$，根据这个能量表达式，作者在文章中提出了利用受限玻尔兹曼机表示多体波函数的方法，

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{M}\sum_{j=1}^{N} h_i w_{ij}\sigma_j^z - \sum_{i=1}^{M} b_i h_i - \sum_{j=1}^{N} a_j \sigma_j^z \tag{1}$$

$$\Psi_M(S;W) = \sum_{\{h_i\}} e^{-E(\mathbf{v},\mathbf{h})} \tag{2}$$

其中$S = \sigma_1^z,...,\sigma_N^z$是自旋体系的一个态，$W$是受限玻尔兹曼机的所有参数。由于隐藏层参数$h_i = \{-1,1\}$，波函数的表达式可以化简为，

$$\Psi_M(S;W) = e^{\sum\limits_{j=1}^{N} a_j\sigma_j^z} \times \prod_{i=1}^{M} G_i(S) \tag{3}$$

$$G_i(S) = 2cosh[b_i + \sum_j w_{ij}\sigma_j^z] = 2cosh[\theta_i(S)] \tag{4}$$

Average of an Operator

在利用受限玻尔兹曼机表示出多体波函数后，最重要的问题就是如何求出该波函数在某个自旋态下

的能量。以1维横场Ising模型为例，

$$H=-\sum_{i=1}^{N}g\sigma_i^x-\sum_{i=1}^{N-1}\sigma_i^z\sigma_{i+1}^z \tag{5}$$

$$\sigma_i^x|...,\uparrow,...>=1\times|...,\downarrow,...>;\ \sigma_i^x|...,\downarrow,...>=1\times|...,\uparrow,...> \tag{6}$$

$$\sigma_i^z\sigma_{i+1}^z|...,\uparrow,\downarrow,...>=1\times(-1)\times|...,\downarrow,\uparrow,...> \tag{7}$$

因此，对于一个算符$\hat{o}$，当它作用于一个自旋态S，则有$\hat{o}*S=mel*S^{'}$。于是，文章中计算该算符在S态下的平均值公式为，

$$<\hat{o}>=mel\times\frac{\Psi(S^{'};W)}{\Psi(S;W)} \tag{8}$$

Training

我在训练模型时使用的初始自旋态为随机自旋态S^0，通过蒙特卡罗采样的方法来生成之后的自旋态$S^1,...,S^p$，其中每次采样的接受概率为，

$$A(S^k\to S^{k+1})=min\left(1,\left|\frac{\Psi(S^{k+1};W)}{\Psi(S^k;W)}\right|^2\right) \tag{9}$$

在多次采样生成一条自旋态的马尔可夫链后，需要对受限玻尔兹曼机的参数进行更新，更新的公式参考文章的Supplementary Material，总结如下：

$$W(p+1)=W(p)-\gamma(p)S^{-1}(p)F(p)$$

其中$\gamma(p)$是学习率，$S=S_{i,j}^{reg}=S_{i,j}+\lambda(p)\delta_{i,j}S_{i,i}$，并且

$$S_{i,j}(p)=\langle O_i^*O_j\rangle-\langle O_i^*\rangle\langle O_j\rangle \tag{10}$$

$$F_j(p)=\left\langle EO_j^*\right\rangle-\langle E\rangle\left\langle O_j^*\right\rangle \tag{11}$$

$$\lambda(p)=max(\lambda_0b^p,\lambda_{min});\lambda_0=100,b=0.9,\lambda_{min}=10^{-4} \tag{12}$$

$$\begin{aligned}
&O_k(S)=\frac{1}{\Psi(S;W)}\partial_W\Psi(S;W)\\
&\frac{1}{\Psi(S;W)}\partial_{a_i}\Psi(S;W)=\sigma_i^z\\
&\frac{1}{\Psi(S;W)}\partial_{b_j}\Psi(S;W)=tanh[\theta_j(S)]\\
&\frac{1}{\Psi(S;W)}\partial_{w_{ij}}\Psi(S;W)=\sigma_i^z tanh[\theta_j(S)]
\end{aligned} \tag{13}$$

3 Results

在本次作业中，计算的Ising模型体系均为20个自旋。由于每改变一次外加磁场g，都需要重新训练一次模型，并且每次模型训练所需时间较长，仅选取了$g=0.01,0.1,0.5,1.0$4个值来简略观察外磁场的影响。另外，受限玻尔兹曼机有一个重要的参数α，是隐藏层节点个数（$\{h_i\}$的个数）与输入层节点个数（$\{\sigma_i^z\}$的个数，即自旋粒子数，在本次作业中均为20）之比。在原文章中作者提到α的作用与DMRG方法中节点连接的维数D_{max}相同，由于α越大，模型训练所需时间越长，本次作业中仅选取了$\alpha=2,4$来评估

结果。

首先定性看一下模型训练时的收敛情况，如图1与图2。

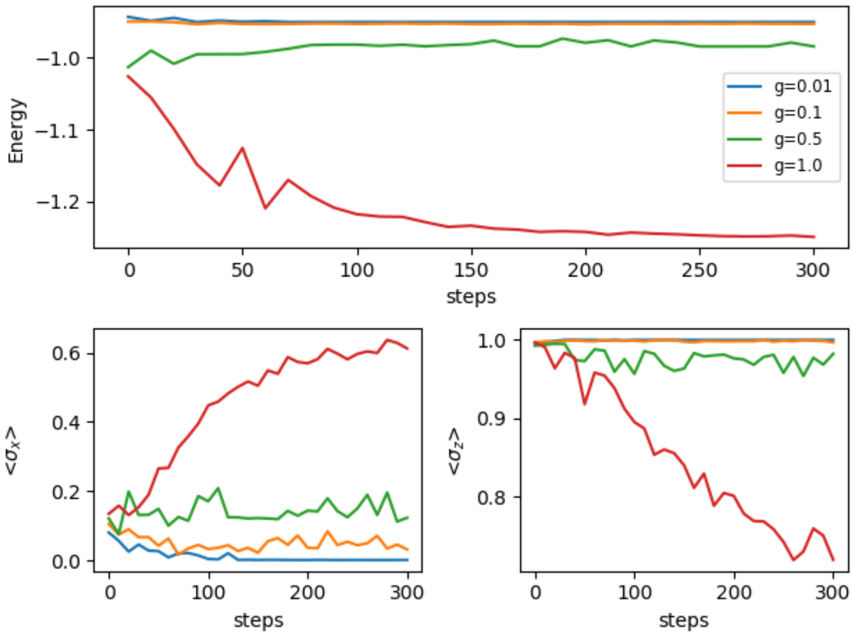

图 1: $\alpha = 2$

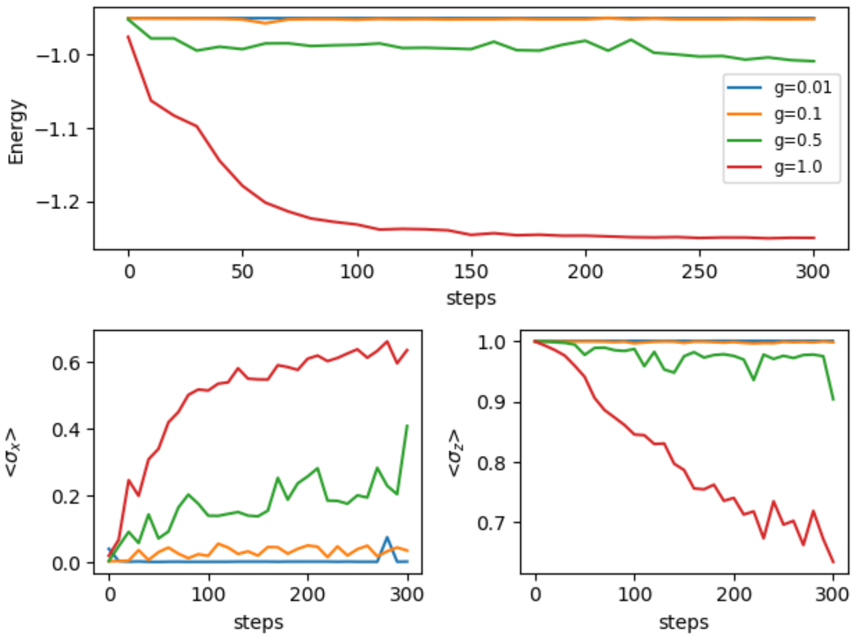

图 2: $\alpha = 4$

从图中可以看出，$\alpha = 4$时的收敛情况相比$\alpha = 2$稍显平滑，但区别不大。体系能量与$\langle\sigma_x\rangle$随外加磁场g的变化与实际相符合，但在$g = 1.0$时$\langle\sigma_z\rangle$的值仍远大于0说明了模型对$\langle\sigma_z\rangle$的预测效果较差。如果确实如作者所说α的作用与DMRG方法中节点连接的维数D_{max}相同，可以尝试使用更大的α来训练。但由

于程序的隐藏bug，当α较大时会有数据溢出错误（推测是因为随机初始态不合理，在α较大时更容易出现数据溢出），暂时无法继续验证更大的α是否能让模型对$\langle\sigma_z\rangle$的预测能力增强。

表 1: NQS模型与DMRG的计算结果对比

	Energy		$\langle\sigma_x\rangle$		$\langle\sigma_z\rangle$	
g	NQS	DMRG	NQS	DMRG	NQS	DMRG
0.01	-0.95001	-0.95003	0.00185	0.00500	1	1
0.1	-0.95118	-0.95275	0.03494	0.05037	0.99875	0.99872
0.5	-1.00863	-1.02001	0.40788	0.29318	0.90425	0
1	-1.24943	-1.25539	0.63523	0.72882	0.63450	0

接下来，定量对比一下$\alpha = 4$时受限玻尔兹曼机模型的计算结果与DMRG的计算结果。从表1中的数据可以看出，在小数点后两位的精确度上，NQS模型与DNRG的基态能量的计算结果是一致的。二者$\langle\sigma_x\rangle$的计算结果变化趋势是相同的，但具体结果有较大差异。NQS模型对$\langle\sigma_z\rangle$的计算结果不可信。

4 Conclusion

本次课程作业参考实现了文献中的受限玻尔兹曼机模型，可以通过这样的神经网络来表示1维Ising模型的多体波函数，并可以利用神经网络量子态（NQS）计算不同算符的观测值。与其他机器学习类似，NQS预测多体波函数可观测值所需的时间很短，但却需要更多的时间来训练出合适的神经网络模型参数。同样的，由于NQS在训练过程中只考虑了基态能量最低，它在预测其他物理量时的表现略有不足。虽然如此，文章提出的NQS模型仅用三层的受限玻尔兹曼机网络能够达到现在的精确度，也是启发我们深度神经网络、卷积神经网络等在解决多体问题时能够有更好的表现。