关于用于特征值问题的低秩张量模型的实验观察

胡俊*和金鹏展†

arXiv:2302.00538v1 [cs.LG] 1 Feb 2023

摘要

本文采用低秩张量模型（LTM）作为函数逼近器，并结合梯度下降法求解包括拉普拉斯算子和谐振子在内的特征值问题。实验结果表明，与张量神经网络（TNN）相比，基于多项式的低秩张量模型（PLTM）表现更优。我们还测试了此类低秩架构在 MNIST 数据集上的分类问题中的应用。

1 引言

基于神经网络的机器学习方法正迅速应用于各种数值问题，例如物理信息神经网络（PINNs）[10, 11, 12], 、深度 Ritz 方法 [2], 和深度 Galerkin 方法 [15]。这些方法的优势之一是展现了解决高维问题的可能性。在 [2], 中，研究者使用深度学习技术以及蒙特卡洛积分来求解特征值问题，为高维情形提供了一种可行策略。对于相同的特征值问题， [16] 采用基于神经网络的低秩张量模型，即张量神经网络（TNN），并结合求积方案进行高效数值积分，因此取得了远优于 [2]的结果。此外， [17] 采用 TNN 求解多体薛定谔方程，展现了此类低秩近似方法的实用价值。

记 ϕ_{ij} 为定义在紧集 Ω_j 上的函数，则关于这些函数的一个 d 阶张量可写为

$$u = \sum_{i=1}^{r} \phi_{i1} \otimes \phi_{i2} \otimes \cdots \otimes \phi_{id}, \tag{1}$$

其秩至多为 r。因此， u 定义在 $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_d$ 上。这种架构可实现高效的高维积分，如下所示

$$\int_{\Omega} u(x)dx = \sum_{i=1}^{r} \int_{\Omega_1} \phi_{i1}(x_1)dx_1 \int_{\Omega_2} \phi_{i2}(x_2)dx_2 \cdots \int_{\Omega_d} \phi_{id}(x_d)dx_d. \tag{2}$$

该特性得到了广泛应用 [1, 9], ，因为计算复杂度不会呈指数级地依赖于d。有关张量的更多讨论，请参阅 [3, 6, 7, 13] 。由于 TNN 的核心思想是利用低秩形式进行高维积分，我们重新考虑了在此类架构中采用神经网络的必要性。因此，我们倾向于提出一种更简洁的架构，仅包含解析基函数（例如多项式），从而能够对积分进行精确计算。这样的

*北京大学数学科学学院，中国北京 100871（hujun@math.pku.edu.cn）。
†北京大学数学科学学院，中国北京 100871（jpz@pku.edu.cn）。

低秩模型将大幅降低数值积分的误差，甚至使误差降为零。下面我们主要在相同的 [16]设置下，重点考察所提出模型相较于 TNN 的性能。

本文组织如下。第2节介绍低秩张量模型。随后，第3节展示相应的特征值问题求解方法。第4节给出了所提方法的若干数值实验。最后，第5节总结本文工作。

2 低秩张量模型

低秩张量模型（LTM）可简洁地写为

$$u(x)=\sum_{i=1}^{r}\prod_{j=1}^{d}\sum_{k=1}^{b}a_{ijk}\phi_k(x_j),\quad x=(x_1,\cdots,x_d)\in\Omega\subset\mathbb{R}^d, \tag{3}$$

其中， r 视为该模型的秩， d 是定义域的维数， $\{\phi_k\}_{k=1}^{b}$ 是选取的基函数，Ω 是 $\mathbb{R}^d$中的一个紧集。这里，a_{ijk} 是可学习参数。

对于多输出情形，该模型表示为

$$u_l(x)=\sum_{i=1}^{r}w_{li}\prod_{j=1}^{d}\sum_{k=1}^{b}a_{ijk}\phi_k(x_j),\quad l=1,2,\cdots,m, \tag{4}$$

其中， w_{li} 也是可学习参数。然后，$\mathbf{u}=(u_1,\cdots,u_m)^\top$ 将 $\Omega\subset\mathbb{R}^d$ 映射到 $\mathbb{R}^m$。

此类模型在 H^k-范数下的逼近性质，直接源自希尔伯特空间 $H^k(\Omega)\cong H^k(\Omega_1)\otimes\cdots\otimes H^k(\Omega_d)$ 的张量积，其中 $\Omega=\Omega_1\times\cdots\times\Omega_d$，$\Omega_i\subset\mathbb{R}$ 为紧集。

备注1. 如有需要，我们还可以令 b 和 ϕ_k 依赖于 i 和 j，形式如下：

$$u(x)=\sum_{i=1}^{r}\prod_{j=1}^{d}\sum_{k=1}^{b_{ij}}a_{ijk}\phi_{ijk}(x_j). \tag{5}$$

例如，对于两个不同维数下分别具有零边界条件和周期条件的问题，最好分别采用多项式（如第3节所示）和傅里叶基。

2.1 基于多项式的低秩张量模型

本研究考察多项式作为基函数的表现。该模型形式如下：

$$u(x)=\sum_{i=1}^{r}\prod_{j=1}^{d}\sum_{k=1}^{b}a_{ijk}P_k(x_j),\quad x=(x_1,\cdots,x_d)\in\Omega\subset\mathbb{R}^d, \tag{6}$$

其中 $\{P_k\}_{k=1}^{b}$ 是选用的多项式基，例如勒让德多项式 L_k.

为满足所需条件，我们可以像谱方法 [14] 中那样，使用常见基本多项式的线性组合。例如，我们可以令 $P_k=L_{k+1}-L_{k-1}$ 满足 $\Omega=[-1,1]^d$ 的零边界条件，这将在我们的数值算例中进行测试。对于长方体区域 $\Omega=[s_1,t_1]\times\cdots\times[s_d,t_d]$,$s_j<t_j$，我们只需进行简单的线性变换，即将 $P_k(x_j)$ 变为 $P_k(2\cdot\frac{x_j-s_j}{t_j-s_j}-1)$。

2.2 基于傅里叶的低秩张量模型

对于具有周期条件的问题，我们可以考虑傅里叶基。这种情况与（6）类似，只是我们采用傅里叶基函数而非多项式。未来将对此模型进行研究。

2.3 张量神经网络

张量神经网络（TNN）在 [4, 16], 中进行了研究，其形式为

$$u(x)=\sum_{i=1}^{r}\prod_{j=1}^{d}\phi_{ij}(x_j),\quad x=(x_1,\cdots,x_d)\in\Omega\subset\mathbb{R}^d, \tag{7}$$

其中，$\Phi_j=(\phi_{1j},\cdots,\phi_{rj})^\top$ 从 $\mathbb{R}$ 到 $\mathbb{R}^r$ 的映射由神经网络（例如前馈神经网络）建模。多输出情形可表示为

$$u_l(x)=\sum_{i=1}^{r}w_{li}\prod_{j=1}^{d}\phi_{ij}(x_j),\quad l=1,2,\cdots,m, \tag{8}$$

其中，w_{li} 也是可学习参数。然后，$\mathbf{u}=(u_1,\cdots,u_m)^\top$ 将 Ω 映射 $\subset\mathbb{R}^d$ 到 $\mathbb{R}^m$。

一般而言，TNN 将式 (3) 和 (4) 中的基函数替换为神经网络，并且这些神经网络针对维数 j不共享参数。在以下实验中，TNN 将作为基准模型。

2.4 初始化

对于基于多项式的低秩张量模型（PLTM），我们设定

$$a_{ijk}=\begin{cases}c & k=1,\\ 0 & k>1,\end{cases} \tag{9}$$

并通过均匀分布随机初始化 w_{li}，例如，采用均匀分布。这里，c 是一个需要调整的常数，若无其他说明，其默认值为 1。式 (9) 的设置非常重要且必不可少，可确保当 d 较大时，式 (3) 和式 (4) 中的乘积不会爆炸或消失。

3 特征值问题的方法

这里我们直接给出一个典型示例，展示如何使用前述模型求解特征值问题。

我们关注以下问题：

$$\begin{cases}-\Delta u+vu=\lambda u & \text{in}\quad \Omega,\\ u=0 & \text{on}\quad \partial\Omega,\end{cases} \tag{10}$$

其中 $\Omega=[s,t]^d$，$s<t$。最小特征值为

$$\lambda=\min_{0\neq u\in H_0^1(\Omega)}\frac{\int_\Omega|\nabla u|^2dx+\int_\Omega vu^2dx}{\int_\Omega u^2dx}. \tag{11}$$

将函数空间记为

$$\mathcal{F}=\left\{u(x)=\sum_{i=1}^{r}\prod_{j=1}^{d}\sum_{k=1}^{b}a_{ijk}P_k(x_j)\,\middle|\,a_{ijk}\in\mathbb{R}\right\}\subset H_0^1(\Omega), \tag{12}$$

其中 $P_k = L_{k+1} - L_{k-1}$ (L_k 是定义在 $[s,t]$ 上的勒让德多项式)， $\theta = \{a_{ijk}\}$ 是可学习参数， r 和 b 是待调节的超参数。这里， $\mathcal{F}$ 中的所有函数均满足式 (10) 的边界条件。于是，损失函数可简洁地写为

$$L(\theta) = L[\tilde{u}] = \frac{\int_\Omega |\nabla \tilde{u}|^2 dx + \int_\Omega v\tilde{u}^2 dx}{\int_\Omega \tilde{u}^2 dx}, \quad \tilde{u} \in \mathcal{F}, \tag{13}$$

其中， θ 对应于 $\tilde{u}$的可学习参数。注意，由于 $\tilde{u}$具有低秩解析表达式，对于任意低秩 v，式（13）中所有与 $\tilde{u}$ 相关的积分都可以精确计算。例如，

$$\begin{aligned}\int_\Omega \tilde{u}^2 dx &= \int_\Omega \left(\sum_{i=1}^{r} \prod_{j=1}^{d} \sum_{k=1}^{b} a_{ijk} P_k(x_j) \right)^2 dx \\ &= \int_\Omega \sum_{i_1=1}^{r} \sum_{i_2=1}^{r} \prod_{j=1}^{d} \sum_{k_1=1}^{b} \sum_{k_2=1}^{b} a_{i_1 j k_1} a_{i_2 j k_2} P_{k_1}(x_j) P_{k_2}(x_j) dx \\ &= \sum_{i_1=1}^{r} \sum_{i_2=1}^{r} \prod_{j=1}^{d} \sum_{k_1=1}^{b} \sum_{k_2=1}^{b} a_{i_1 j k_1} a_{i_2 j k_2} I_{k_1 k_2},\end{aligned} \tag{14}$$

其中

$$I_{k_1 k_2} = \int_s^t P_{k_1}(y) P_{k_2}(y) dy. \tag{15}$$

矩阵（I_{ij}） $\in \mathbb{R}^{b\times b}$ 可利用勒让德多项式的正交性轻松求得。其余积分也可以用类似的方法求得。

将 θ 初始化为 $\theta^{(0)}$后，采用基于梯度下降（GD）的方法优化损失函数 $L(\theta)$:

$$\theta^{(k+1)} = \theta^{(k)} - \eta \nabla L(\theta^{(k)}), \tag{16}$$

其中学习率 η取适当值。在以下实验中，我们选择 Adam [5] 作为优化器。经过充分训练后，我们得到近似特征函数 $\tilde{u}^{(K)}$ 以及相应的近似特征值 $\lambda^{(K)} = L(\theta^{(K)})$，此时迭代次数为 K。

4 数值结果

4.1 拉普拉斯算子

考虑第3节中的拉普拉斯算子情形，其中我们指定 $v = 0$ 和 $[s,t] = [0,1]$，则精确的最小特征值和特征函数为

$$\lambda = d\pi^2, \quad u(x) = \prod_{i=1}^{d} \sin(\pi x_i). \tag{17}$$

损失函数可写为

$$L(\theta) = L[\tilde{u}] = \frac{\int_\Omega |\nabla \tilde{u}|^2 dx}{\int_\Omega \tilde{u}^2 dx}, \quad \tilde{u} \in \mathcal{F}. \tag{18}$$

这里我们测试 $d = 10$ 和 $d = 512$。基函数数量 b 和秩 r 均设为 10，并使用 Adam优化器以 0.001 的学习率训练该模型 500 次迭代。结果如表 1 所示，几乎达到了机器精度。对于 $d = 512$，TNN 的结果相对误差为 1.6×10^{-7}，来自 [16]。

d	10	512
真实特征值	$10\pi^2 \approx 98.69604401089359$	$512\pi^2 \approx 5053.237453357751$
学习得到的特征值	98.69604401089354	5053.237453358372
相对误差	4.3×10^{-16}	1.2×10^{-13}

表 1：拉普拉斯算子的 PLTM 学习得到的特征值。结果几乎达到了机器精度，而 TNN 对于 $d = 512$ 的相对误差为 1.6×10^{-7}。

4.2 谐振子

现在我们考虑谐振子的情形，其中 $v(x) = \sum_{i=1}^{d} x_i^2$和 $[s, t] = [-5, 5]$。这里 v是一个秩为 d 的张量。精确的最小特征值和特征函数为

$$\lambda = d, \quad u(x) = \prod_{i=1}^{d} \exp(-\frac{x_i^2}{2}). \tag{19}$$

损失函数可以写为

$$L(\theta) = L[\tilde{u}] = \frac{\int_\Omega |\nabla \tilde{u}|^2 dx + \sum_{i=1}^{d} \int_\Omega x_i^2 \tilde{u}^2 dx}{\int_\Omega \tilde{u}^2 dx}, \quad \tilde{u} \in \mathcal{F}. \tag{20}$$

我们还测试了 $d = 10$ 和 $d = 512$。基函数的数量 b 和秩 r 分别设为 22 和 10。值得注意的是，对于 $d = 512$，我们将式（9）中的初始值 c 设为 0.3。我们使用 Adam优化器，以 0.001 的学习率训练该模型 1000 次迭代。结果如表 2 所示，其相对误差为 $\sim 10^{-8}$。对于 $d = 512$，TNN 的结果相对于 [16]达到了 8.5×10^{-6} 相对误差。

另一个值得注意的地方是，根据我们的观察，当基函数的数量 b 非常大时，损失 L 会降至低于理论最小值 λ 的数值。必须仔细考虑该模型中的机器误差。

d	10	512
真实特征值	10	512
学习得到的特征值	10.000000345538101	512.0000063370244
相对误差	3.5×10^{-8}	1.2×10^{-8}

表 2：谐振子的 PLTM 学习得到的特征值。其相对误差为 $\sim 10^{-8}$，而对于 $d = 512$，TNN 达到了 8.5×10^{-6} 相对误差。

4.3 分类问题

为了研究 LTM 的泛化能力，我们在手写数字 MNIST 数据集上测试模型 [8],示例见图 1。MNIST 中的每张图像都是一个 28×28实矩阵，可视为 $\mathbb{R}^{784}$中的一个向量。图像分为 10 个类别 $\{0, 1, \cdots, 9\}$，例如，如果图像 x_i表示数字“k”，则会被标记为 $y_i = k$。将该数据集记为 $\{(x_i,\ y_i)\}_{i=1}^{N}$。

分类模型可以写成

$$F(x) = \text{Softmax}(\tilde{u}(x)), \tag{21}$$

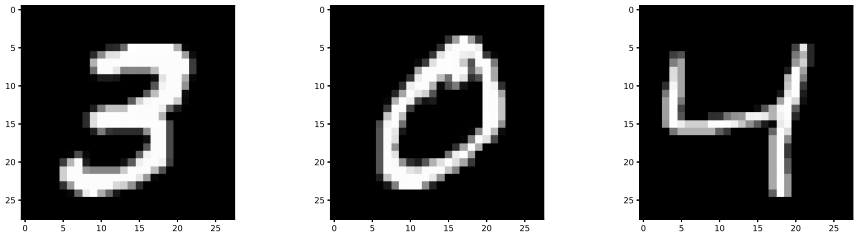

图1：手写数字。

其中

$$\mathrm{Softmax}(x) := \frac{1}{\exp(x_1)+\cdots+\exp(x_m)}\left(\exp(x_1),\cdots,\exp(x_m)\right), \quad x=(x_1,\cdots,x_m), \quad \forall m\in\mathbb{N}^*, \quad (22)$$

这实际上是对输入进行归一化。这里， $\tilde{u}$: $\mathbb{R}^{784}\to\mathbb{R}^{10}$ 是所使用的函数模型，例如前馈神经网络和 LTM。记 $F=(F_0,\ \cdots,\ F_9)$，则损失函数为

$$L(\theta)=L[\tilde{u}]=-\sum_{i=1}^{N}\ln(F_{y_i}(x_i)). \quad (23)$$

通过优化 L，我们得到作为分类器的 F。对于未知输入图像 x，我们将 $F(x)$中最大分量的索引视为其预测。这里我们测试了四种模型，即非线性前馈神经网络（多于一层）、线性前馈神经网络（一个不带激活函数的线性层）、PLTM 和 TNN。对于 PLTM，我们直接使用勒让德多项式作为基函数。

结果如表3所示。非线性前馈神经网络的准确率可达 $\approx 98\%$，而线性前馈神经网络的准确率为 $\approx 92\%$。PLTM 和 TNN 的准确率均为 $\approx 95\%$。在这种情况下，PLTM 和 TNN 之间没有明显差异；它们的表现都不如非线性前馈神经网络，但优于线性前馈神经网络。

	前馈神经网络（非线性）	前馈神经网络（线性）	PLTM	TNN
MNIST	$\approx 98\%$	$\approx 92\%$	$\approx 95\%$	$\approx 95\%$

表3： MNIST 数据集分类问题的不同函数模型 .

5 结论

我们通过低秩近似方法求解包含拉普拉斯算子和谐振子的特征值问题，其中利用了基于多项式的低秩张量模型（PLTM）和张量神经网络（TNN）。以下几点体现了 PLTM 的优势：(i) 与需要通过数值积分近似相关积分的 TNN 相比，PLTM 可以精确计算这些积分。(ii) 结果表明，PLTM 比 TNN 更准确。(iii) 此外，PLTM 的训练过程比 TNN 快得多。为考察 PLTM 和 TNN 的泛化能力，我们进一步测试了它们在 MNIST 数据集上的分类问题，发现 PLTM 和 TNN 之间没有明显差异。

未来将考虑更复杂的例子以及不同的基函数（例如傅里叶基）。

参考文献

[1] G. Beylkin 和 M. J. Mohlenkamp. 高维数值算子演算. 美国国家科学院院刊, 99(16):10246–10251, 2002.[2] W. E 和 B. Yu. 深度 Ritz 方法：一种用于求解变分问题的基于深度学习的数值算法. 《数学与统计通讯》, 6(1):1–12, 2018.[3] J. H˚astad. 张量秩是 NP 完全的. 载于国际自动机、语言与编程研讨会, 第 451–460 页. 施普林格, 1989.[4] P. Jin, S. Meng, 和 L. Lu. MIONet：通过张量积学习多输入算子. 《*SIAM* 科学计算杂志》, 44(6):A3490–A3514, 2022.[5]D. P. Kingma 和 J. Ba. Adam：一种随机优化方法. *arXiv* 预印本 *arXiv:1412.6980*, 2014.[6] T. G. Kolda 和 B. W. Bader. 张量分解及其应用. 《*SIAM* 综述》, 51(3):455–500, 2009.[7] J. B. Kruskal. 三维和 n 维数组的秩、分解与唯一性. 载于多路数据分析, 第 7–18 页. 1989.[8] Y. LeCun, L. Bottou, Y. Bengio, 和 P. Haffner. 应用于文档识别的基于梯度的学习. 《*IEEE* 会刊》, 86(11):2278–2324, 1998.[9] M. S. Litsarev 和 I. V. Oseledets. 离子-原子碰撞建模中多维积分的快速低秩近似. 《数值线性代数及其应用》, 22(6):1147–1160, 2015.[10] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, 和 S. G. Johnson. 用于逆向设计的带硬约束物理信息神经网络. 《*SIAM* 科学计算杂志》, 43(6):B1105– B1132, 2021.[11] G. Pang, L. Lu, 和 G. E. Karniadakis. fpinns：分数阶物理信息神经网络. 《*SIAM* 科学计算杂志》, 41(4):A2603–A2626, 2019.[12] M. Raissi, P. Perdikaris, 和 G. E. Karniadakis. 物理信息神经网络：一种用于求解涉及非线性偏微分方程的正问题和逆问题的深度学习框架.《计算物理学杂志》, 378:686–707, 2019.[13] R. A. Ryan 和 R. a Ryan. 巴拿赫空间张量积导论, 第 73 卷. 施普林格, 2002.[14] J. Shen, T. Tang, 和 L.-L. Wang. 谱方法：算法、分析与应用, 第 41 卷. 施普林格科学与商业媒体, 2011.[15] J. Sirignano 和 K. Spiliopoulos. DGM：一种用于求解偏微分方程的深度学习算法. 《计算物理学杂志》, 375:1339–1364, 2018.[16] Y. Wang, P. Jin, 和 H. Xie. 张量神经网络及其数值积分. *arXiv* 预印本 *arXiv:2207.02754*, 2022.[17] Y. Wang, Y. Liao, 和 H. Xie. 使用张量神经网络求解薛定谔方程. *arXiv* 预印本 *arXiv:2209.12572*, 2022.