

分类号: O242.1
U.D.C.: 519.6

密级: 公开
编号: 82817

中国工程物理研究院

学位论文

基于环流的大偏差理论和涨落定理的研究

姜瑜浩

指导教师姓名 贾晨 助理教授

申请学位级别 硕士 专业名称 应用数学

论文提交日期 2022 年 3 月 论文答辩日期 2022 年 6 月

授予学位单位和日期 中国工程物理研究

答辩委员会主席

评阅人

2022 年 3 月 10 日

Classified Index: O242.1
U.D.C.: 519.6
Secret State: Public
Number: 82817

China Academy of Engineering Physics

Dissertation for the Master Degree in Engineering

Large deviations and fluctuation theorems for cycle currents

Yuhao Jiang

Supervisor: Prof. Chen Jia

Academic Degree Applied for: Master of Science

Specialty: Applied Mathematics

Date of Submitting: April, 2022 Date of Defence: June, 2022

Degree-Confering-Institution: China Academy of Engineering Physics

Chairman of defence committee

Paper Reviewers

10th 3, 2022

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中国工程物理研究院或其他教育机构的学位或证书使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

签字日期： 年 月 日

学位论文版权使用授权书

本学位论文作者完全了解并接受中国工程物理研究院研究生部有关保存、使用学位论文的规定，允许论文被查阅、借阅和送交国家有关部门或机构，同时授权中国工程物理研究院研究生部可以将学位论文全部或部分内容编入有关数据库进行检索，可以影印、缩印或扫描等复制手段保存、汇编学位论文。

学位论文作者签名：

导师签名：

签字日期： 年 月 日

签字日期： 年 月 日

摘 要

环流理论一直是随机热力学理论的关键内容。马氏系统中的环流和净环流可以由两种方式定义，即环擦除和生成树。该研究针对两种研究下的大偏差理论和涨落定理进行比较研究。首先研究环拓扑结构系统，并计算出环擦除定义下大偏差速率函数的隐式表达式，然后研究生成树定义下的大偏差速率函数，并阐明两者之间的关系。进一步，检验两个定义下的结论和涨落定理的相容性。经过严格对照，生成树定义下的环流不满足涨落定理，然而生成树的结果满足涨落定理的弱形式。

关键词：环流；大偏差；涨落定理

Abstract

The cycle current is a crucial quantity in stochastic thermodynamics. The absolute and net cycle currents of a Markovian system can be defined in the loop-erased (LE) or the spanning tree (ST) manner. Here we make a comparative study between the large deviations and fluctuation theorems for the LE and ST currents, i.e. cycle currents defined in the LE and ST manners. First, we give the explicit expression of the large deviation rate functions for the LE currents of a system with a cyclic topology and for the ST currents of a general system. The relationship between the rate functions for the LE and ST currents are clarified. Furthermore, we examine various types of fluctuation theorems satisfied by the LE and ST currents. We show that both the absolute and net LE currents satisfy the strong form of all types of fluctuation theorems. In contrast, the absolute ST currents do not satisfy fluctuation theorems, while the net ST currents only satisfy the weak form of fluctuation theorems.

Keywords: cycle currents, large deviations, fluctuation theorems

目录

第一章 绪论

1.1 问题的背景和研究现状

本文中研究的椭圆方程具有如下的一般形式。在 d 维空间中的求解区域 $\Omega \subset \mathbb{R}^d$ 上, 椭圆方程表示为

$$\begin{cases} -\nabla(a(x)\nabla u(x)) + c(x)u(x) = f(x) & \text{for } x \in \Omega \\ u(x) = u_0(x) & \text{for } x \in \partial\Omega \end{cases} \quad (1-1)$$

其中 $a(x) > a_0 > 0, c(x) \geq 0, f(x), u_0(x)$ 是已知函数, $u(x)$ 是方程的解。

椭圆特征值问题表示为

$$\begin{cases} -\nabla(a(x)\nabla u(x)) + c(x)u(x) = \lambda u(x) & \text{for } x \in \Omega \\ u(x) = 0 & \text{for } x \in \partial\Omega \end{cases} \quad (1-2)$$

其中, $u(x)$ 是待求解的特征函数, λ 是对应的特征值。

椭圆方程是应用很广泛的一类方程, 例如: 描述溶液中蛋白质电势分布的 Poisson-Boltzmann 方程 [?], 描述量子效应的 Schrodinger 方程等, 都可以看作是椭圆方程。目前已经有很多传统方法 (如: 有限差分法, 有限元方法, 有限体积方法, 谱方法等) 可以高效且高精度地求解椭圆方程。在本文中, 我们主要关注用深度学习方法求解椭圆方程。

深度学习方法是当前研究的热点问题。在许多领域内 (如: 图像处理, 语音识别等) 深度学习方法已经被广泛应用, 并且取得了超越传统方法的效果。在科学计算问题中, 深度学习方法也已经取得了一定的进展。由于神经网络的逼近性质和维数的关系不大, 深度学习方法能够一定程度上缓解维数灾难, 在求解高维方程时有很好的效果。此外, 由于深度学习方法不需要像传统的有限元或有限差分法那样使用结构化网格, 因此它可以轻易处理复杂的求解区域。文献 [? ? ?] 中讨论了深度学习方法在求解微分方程中的应用。文献 [? ?] 中讨论了如何通过深度学习方法模拟物理模型。文献 [? ?] 中讨论了深度神经网络和传统有限元方法的联系, 从数学角度解释了深度神经网络的逼近性质。

对于深度学习方法在求解椭圆方程上的具体应用, 文献 [? ?] 中提出了 Deep-Ritz 方法, 通过极小化能量泛函的方式求解方程。同时, 文献 [?] 中提出了 PINN 方法, 通过极小化残量的方式求解方程。

然而, 要将常用的深度神经网络应用于计算科学和工程问题, 我们面临着一些挑战。其中最突出的问题是, 深度学习方法通常只适合于处理低频数据。文献 [? ? ?] 中提出了频率准则。频率准则指出, 许多深度神经网络学习低频数据的速度很快, 效果很好, 但当处理高频数据时, 它们就失去了这些优势。文献 [? ? ? ?] 在理论上严格地论证了频率准则。因此, 对于解具有高震荡的问题, 我们需要采取特殊的处理。

目前用深度学习方法求解偏微分方程的工作已经有许多，但从频域的角度探索深度学习方法的研究还不完善。文献 [?] 中提出了 Phase-DNN。它通过在频域上平移的方式求解带有高震荡的方程，取得了很好的效果。但美中不足的是，Phase-DNN 在求解过程中需要事先知道精确解的频率，因此无法求解频率未知的方程。

神经网络在不同频率下的表现不同，利用这种差异设计神经网络结构可以有利于提高网络的性能。在计算机视觉领域，已经有一系列的工作，如：图像恢复 [?]、超分辨率 [?]、分类问题 [?] 等，通过利用不同图像频率的学习差异，提高了学习性能（包括泛化性能和训练速度）。在多尺度变换方面，文献 [?] 和文献 [?] 中分别基于快速多极算法和小波变换的思想设计了多尺度网络，使网络性能得到了很大提升。

1.2 本文主要研究内容

本文主要研究内容可以分为以下三个方面：

首先，在实际问题中，我们通常希望模型和输入数据的尺度无关。例如在图像处理中，我们希望输入的灰度值无论是在 $[0, 255]$ 还是 $[0, 1]$ 之间，得到的结果应该类似。很多机器学习方法（如：简单的线性模型，SVM 等）都有这种性质，但是并没有理论保证深度学习具有类似的性质。本文论述了尺度变换（在输入数据或者输出数据上）对网络的影响，得到了一些初步的结论。

其次，深度学习是基于优化算法的方法，它把方程求解的问题转化成一个优化问题。但是由于神经网络本身的复杂性，这样的优化问题通常是非凸且病态的。想要高效求解这样的优化问题并不容易，一般来说，我们只能找到问题中靠近初始点的局部极小值。因此，在这样的问題中，初值的选取十分关键，好的初值可以提高网络的精度（使训练过程收敛到误差更小的局部最优值），也可以加速训练过程（使初始值离局部最优值更近）。本文基于尺度变换，提出了一种有效的初值选取方式，同时基于这种初值选取方式提出了多尺度网络结构。

最后，本文实验了多尺度网络在求解椭圆方程和分子计算模拟问题中的效果，同时指出了深度学习方法在求解椭圆特征值问题中的潜力。经过各方面的实验，我们可以验证，相比于传统的全连接网络，多尺度网络求解椭圆方程的速度更快，误差也更小。这些实验清楚地表明，多尺度网络是一种高效且易于实现的无网格椭圆方程求解器。

第二章 模型介绍

在此考虑基于离散时间马氏链 $\xi = (\xi_n)_{n \geq 0}$ 建模的随机热力学系统，该模型的状态空间是 $S = \{1, \dots, N\}$ 转移概率矩阵是 $P = (p_{ij})_{i,j \in S}$ ，其中 p_{ij} 表示从状态 i 到状态 j 的转移概率。该马氏链的转移图表示为 $G = (S, E)$ ，其中顶点集合 S 是状态空间， E 是转移概率有向连接边的集合。在这篇论文里，用 $\langle i, j \rangle$ 表示状态 i 到状态 j 的有向边，所以有 $E = \{\langle i, j \rangle \in S \times S : p_{ij} > 0\}$ ，并且令 $M = |E|$ ，其中 $|E|$ 表示集合 E 中元素的数量。本文中假设马氏链是不可约的，也就是有向图 G 是连通的。因此对某个状态，图 G 不仅包含其他状态到其的边，还包括它到其自身的边，也就是自循环。

本文首先研究只有一个多于两状态的环的图 G ，如图 1(a) 所示，并称具有这种图结构的马氏链为单环。也就是说，如果马氏链的转移概率矩阵满足， $p_{ij} = 0$ 且 $|\text{mod}(i-j, N)| \geq 2$ ($\text{mod}(*, N)$ 表示模 N 同余)。生化反应中的很多物理过程也都用单环马氏链建模，比如酶的构象变化，磷酸化-脱磷酸化环，染色体重塑引起的启动子激活等。本文会过多关注单环系统，很多结论也会延展到一般系统中。

2.1 环擦除定义下的环流

本文研究马氏链中两种类型的环流，该章节回顾环擦除定义下的环流。马氏链中的环流是用路径定义的，比如路径 $i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_s \rightarrow i_1$ (i_1, i_2, \dots, i_s 是顶点集合 S 中不同点) 的环流为 $p_{i_1 i_2} p_{i_2 i_3} \dots p_{i_s i_1} > 0$ 。令 $j_1 \rightarrow j_2 \rightarrow \dots \rightarrow j_r \rightarrow j_1$ 为另一个环，若上述两个环满足存在一个整数 k 使得

$$j_1 = i_{k+1}, j_2 = i_{k+2}, \dots, j_n = i_{k+s},$$

且 $r = s$ 则称两个环流是等价的，其中指标 $k+1, k+2, \dots, k+s$ 被视为模 n 同余的。环流 $i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_s \rightarrow i_1$ 在上述等价关系下所属的等价类被表示为 (i_1, i_2, \dots, i_s) 。例如， $(1, 2, 3)$, $(2, 3, 1)$ 和 $(3, 1, 2)$ 表示相同的环。环 (i_1, i_2, \dots, i_s) 的反环被定义为 (i_1, i_s, \dots, i_2) 。通常称马氏链中所用环的集合为环空间 C 。

马氏链的一条轨迹会形成各种环。直观看，如果我们抛弃马氏链 ξ 中环，并且在该过程中，始终关注轨迹中剩余状态形成的轨道，那么称剩余的轨迹为导出链 $\tilde{\xi} = (\tilde{\xi}_n)_{n \geq 0}$ 。例如，如果马氏链 ξ 的轨迹为 $\{1, 2, 3, 3, 2, 3, 4, 1, 4, \dots\}$ ，那么相应的导出链 $\tilde{\xi}$ 的轨迹和环形成表 1 所示

特别地，导出链的状态用 S 的中状态组成的有限序列表示，即

2.2 生成树定义下的环流

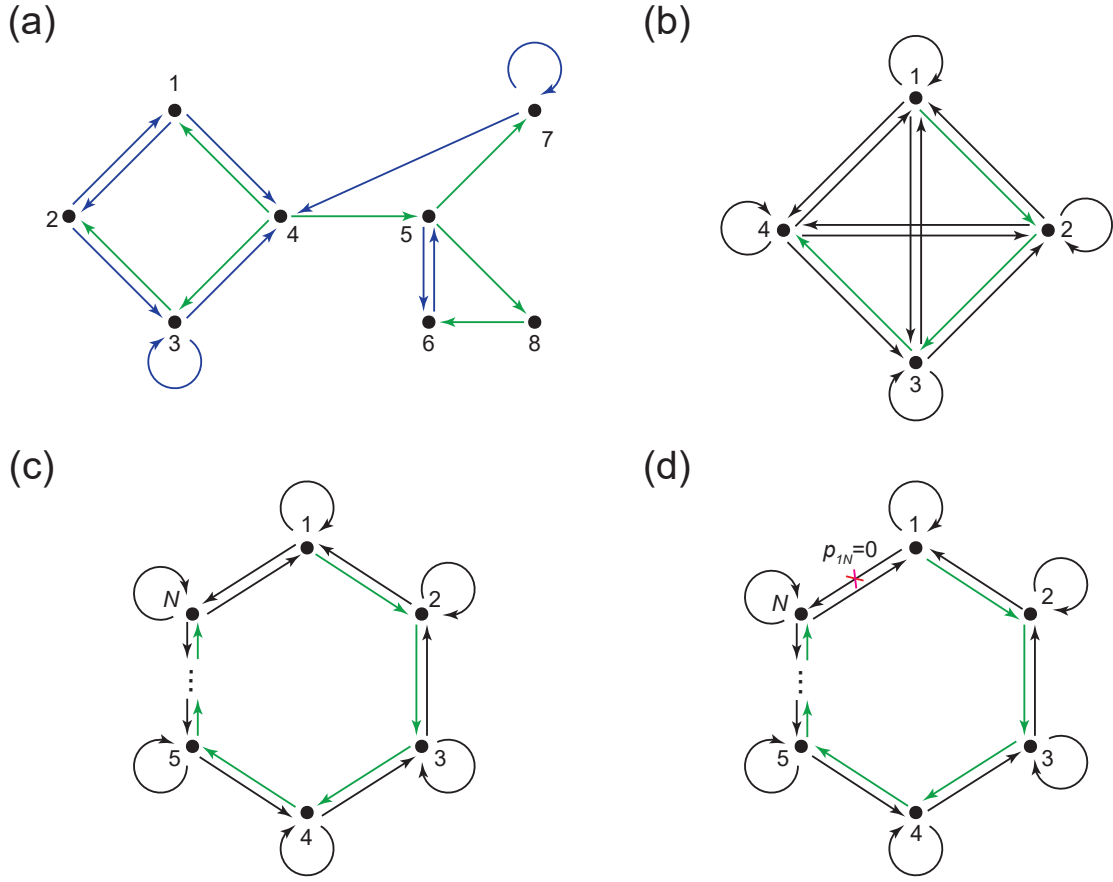


图 2.1 不同马氏链的转移图 (a) 一般马氏链的转移图，绿色线表示生成树 T ，并且蓝色线表示 T 的弦. (b) 转移概率矩阵满足 $p_{ij} > 0$ ， $\forall i, j, 1 \leq i, j \leq 4$ 的马氏链。绿色线表示生成树 $T = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$. (c) N 状态单环马氏链。绿线表示生成树 $T = 1 \rightarrow 2 \rightarrow \cdots \rightarrow N$. (d) 转移概率矩阵有 $p_{1N} = 0$ 的单环马氏链，绿色线表示生成树 $T = 1 \rightarrow 2 \rightarrow \cdots \rightarrow N$.

n	0	1	2	3	4	5	6	7	8
ξ_n	1	2	3	3	2	3	4	1	4
$\tilde{\xi}_n$	[1]	[1,2]	[1,2,3]	[1,2,3]	[1,2]	[1,2,3]	[1,2,3,4]	[1]	[1,4]
环				(3)	(2,3)			(1,2,3,4)	

表 2.1 导出链和形成的环

第三章 神经网络的仿射不变性

在实际问题中，我们通常希望模型和输入数据的尺度无关。但是在实验中，对输入或者输出数据进行线性变换，得到的结果可能完全不同。本章我们从简单的一维例子开始，论述对网络的输入和输出进行线性变换，会对网络产生什么影响。

为了简单，本章中所有的证明仅考虑了两层神经网络，同时也只考虑了在一维区间上的情形。这些证明都可以轻易地推广到高维情况和层数更多的网络。

3.1 输入变换对拟合任务的影响

已知函数 $f : \Omega = [A, B] \rightarrow \mathbb{R}$ ，用网络拟合这个函数。

这里我们考虑两层的网络，网络定义为

$$u(x; \theta) = \sum_{j=1}^H a_j \sigma(w_j x + b_j) \quad x \in \Omega \quad (3-1)$$

在拟合任务中，损失函数定义为

$$L(\theta) = \frac{1}{B-A} \int_A^B |f(x) - u(x; \theta)|^2 dx \quad (3-2)$$

这个损失函数可以看成是离散的损失函数

$$\frac{1}{N} \sum_{i=1}^N |f(x_i) - u(x_i; \theta)|^2$$

对应的连续形式。

我们通过梯度下降法来训练网络，考虑连续情形，参数 θ 随训练时间的变化为

$$\frac{d\theta}{dt} = -\eta \frac{\partial L}{\partial \theta} \quad (3-3)$$

其中 η 是学习率。

下面我们把一般区间 $\Omega = [A, B]$ 伸缩到单位区间 $\hat{\Omega} = [0, 1]$ 上。定义线性变换 $S : \hat{\Omega} = [0, 1] \rightarrow \Omega = [A, B]$ 。它就是普通的伸缩和平移 $x = S(\hat{x}) = (B-A)\hat{x} + A$ 。

把函数 f 伸缩到单位区间上，得到 \hat{f} ，它满足 $\hat{f}(\hat{x}) = f(x)$ 。

在单位区间上定义一个新的两层网络

$$\hat{u}(\hat{x}; \hat{\theta}) = \sum_{j=1}^H \hat{a}_j \sigma(\hat{w}_j \hat{x} + \hat{b}_j) \quad \hat{x} \in \hat{\Omega} \quad (3-4)$$

在相同的学习率下训练单位区间上的网络，我们希望找到单位区间上的网络 and 原区间上的网络的联系。

首先，在初始化的时候，我们把单位区间上的初值定义成由原区间上的伸缩变换而来。定义在参考单元上新的参数 $\hat{\theta}$ 为

$$\hat{a}_j = a_j \quad \hat{w}_j = (B - A)w_j \quad \hat{b}_j = Aw_j + b_j \quad (3-5)$$

这个时候，在初始 $t = 0$ 的时候，两个区间上的网络就满足 $\hat{u}(\hat{x}; \hat{\theta}) = u(x; \theta)$ 。

在初始时刻，单位区间上的损失函数为

$$\hat{L}(\hat{\theta}) = \int_0^1 |\hat{f}(\hat{x}) - u(\hat{x}; \hat{\theta})|^2 d\hat{x} = \frac{1}{B - A} \int_A^B |f(x) - u(x; \theta)|^2 dx = L(\theta) \quad (3-6)$$

也就是说， $t = 0$ 的时候，两个问题的损失函数相等。

在初始时刻，损失函数对参数 w 的导数，满足关系

$$\frac{\partial \hat{L}}{\partial \hat{w}_j} = (B - A) \frac{\partial L}{\partial w_j}$$

对参数 b 的导数，满足

$$\frac{\partial \hat{L}}{\partial \hat{b}_j} = \frac{\partial L}{\partial b_j} + A \frac{\partial L}{\partial w_j}$$

对其它参数的导数不变。

对于参数 \hat{w}_j ，它的变化为

$$\frac{d\hat{w}_j}{dt} = -\eta \frac{\partial \hat{L}}{\partial \hat{w}_j} = -\eta(B - A) \frac{\partial L}{\partial w_j} = (B - A) \frac{dw_j}{dt}$$

对于参数 \hat{b}_j ，它的变化为

$$\frac{d\hat{b}_j}{dt} = -\eta \frac{\partial \hat{L}}{\partial \hat{b}_j} = -\eta \frac{\partial L}{\partial b_j} - \eta A \frac{\partial L}{\partial w_j} = \frac{db_j}{dt} + A \frac{dw_j}{dt}$$

因此可以得到

$$\hat{a}_j = a_j \quad \hat{w}_j = (B - A)w_j \quad \hat{b}_j = Aw_j + b_j \quad \forall t \geq 0$$

这就说明：只要按照??中所示选取网络的初值，在所有的训练过程中，两个网络的参数也会一直保持??中所示的线性关系。

类似的结论可以轻易推广到更复杂的情况。这里的结果告诉我们，**对输入数据进行线性变换相当于改变网络的第一层参数的初始值**。具体来说：在区间 $[A, B]$ 上拟合函数，就相当于在区间 $[0, 1]$ 上用相同的学习率拟合函数。初始化参数时第一层的权重变成原来的 $(B - A)$ 倍，第一层的位移在原来的基础上加上 $(B - A)w + A$ ，后面神经元的所有参数不变。而且在训练的过程中，两个网络的所有权重和位移会一直保持这种线性变换的关系。

3.2 输出变换对拟合任务的影响

已知函数 $f : \Omega = [0, 1] \rightarrow \mathbb{R}$ ，我们用网络拟合这个函数。网络定义和??相同，损失函数定义和??相同，参数变化和??相同。

这里我们对目标函数的值域进行变换。定义函数 $\hat{f}(x) = Af(x)$ ，定义域和 f 相同。我们用网络??拟合函数 \hat{f} 。

首先，在初始化参数的时候，我们同样定义 \hat{u} 是由 u 伸缩变换而来的。新的参数 $\hat{\theta}$ 初始化为

$$\hat{a}_j = Aa_j \quad \hat{w}_j = w_j \quad \hat{b}_j = b_j \quad (3-7)$$

在这种初始化条件下，在未开始训练的时候，两个区间上的网络就满足 $\hat{u}(\hat{x}; \hat{\theta}) = Au(x; \theta)$ 。

在初始时刻，变换后的损失函数为

$$\hat{L}(\hat{\theta}) = \int_0^1 |\hat{f}(x) - \hat{u}(x; \hat{\theta})|^2 dx = A^2 \int_0^1 |f(x) - u(x; \theta)|^2 dx = A^2 L(\theta)$$

也就是说， $t = 0$ 的时候，两个网络的损失函数相差 A^2 倍。

在初始时刻，损失函数对参数的导数，满足关系

$$\frac{\partial \hat{L}}{\partial \hat{w}_j} = A^2 \frac{\partial L}{\partial w} \quad \frac{\partial \hat{L}}{\partial \hat{b}_j} = A^2 \frac{\partial L}{\partial b} \quad \frac{\partial \hat{L}}{\partial \hat{a}_j} = A \frac{\partial L}{\partial a}$$

这里，我们以学习率 η/A^2 训练新的网络。即 $\hat{\eta} = \eta/A^2$ 。

此时就可以满足

$$\frac{d\hat{w}_j}{dt} = \frac{dw_j}{dt} \quad \frac{d\hat{b}_j}{dt} = \frac{db_j}{dt} \quad \frac{d\hat{a}_j}{dt} = A \frac{da_j}{dt}$$

通过类似的推理可以得到

$$\hat{a}_j = Aa_j \quad \hat{w}_j = w_j \quad \hat{b}_j = b_j \quad \forall t \geq 0$$

类似的结论同样可以轻易推广到更复杂的情况。这里的结果告诉我们，**对输出数据进行线性变换相当于改变网络的最后一层参数的初始值，以及训练网络的学习率**。具体来说：用学习率 η 拟合函数 $f(x)$ ，就相当于用学习率 η/A^2 拟合函数 $Af(x) + B$ 。初始化参数时最后层的权重变成原来的 A 倍，最后一层的位移在原来的基础上加上 B ，其它神经元的所有参数不变。

3.3 输入变换对直接法解方程的影响

以求解一维区间上的常系数椭圆方程为例。方程表示为

$$\begin{cases} -u''(x) = f(x) & \text{for } x \in \Omega = (A, B) \\ u(A) = u_A \quad u(B) = u_B \end{cases} \quad (3-8)$$

两层神经网络的定义如??，求解方程的过程中，损失函数定义分为两部分。

内部的损失函数为

$$L_{ii}(\theta) = \frac{1}{B-A} \int_A^B |f(x) - \frac{d^2 u}{dx^2}(x; \theta)|^2 dx \quad (3-9)$$

边界的损失函数为

$$L_{bd}(\theta) = \frac{1}{2} |u(A; \theta) - u_A|^2 + \frac{1}{2} |u(B; \theta) - u_B|^2 \quad (3-10)$$

总体的损失函数是由两部分乘以对应的惩罚系数相加

$$L(\theta) = L_{it}(\theta) + \beta L_{bd}(\theta) \quad (3-11)$$

梯度下降的过程中，参数的变化如??所示。

把方程伸缩到单位区间 $[0, 1]$ 上，得到

$$\begin{cases} -\hat{u}''(\hat{x}) = (B-A)^2 \hat{f}(\hat{x}) & \text{for } \hat{x} \in \hat{\Omega} = (0, 1) \\ \hat{u}(0) = u_A & \hat{u}(1) = u_B \end{cases} \quad (3-12)$$

其中 \hat{u} 和 \hat{f} 满足 $\hat{f}(\hat{x}) = f(x)$ 和 $\hat{u}(\hat{x}) = u(x)$ 。

在单位区间上定义一个新的网络同??，新的参数初始值同??。这个时候，在初始时刻，两个区间上的网络就满足 $u(x; \theta) = \hat{u}(\hat{x}; \hat{\theta})$ 。

在初始时刻，参数满足??的关系，两个网络的二阶导数满足

$$\frac{d^2 \hat{u}}{d\hat{x}^2}(\hat{x}; \hat{\theta}) = (B-A)^2 \frac{d^2 u}{dx^2}(x; \theta)$$

在初始时刻，单位区间上的损失函数为

$$\hat{L}_{it}(\hat{\theta}) = \int_0^1 |(B-A)^2 \hat{f}(\hat{x}) - \frac{d^2 \hat{u}}{d\hat{x}^2}(\hat{x}; \hat{\theta})|^2 d\hat{x} = \frac{1}{B-A} \int_A^B (B-A)^2 |f(x) - \frac{d^2 u}{dx^2}(x; \theta)|^2 dx = (B-A)^2 L_{it}(\theta)$$

$$\hat{L}_{bd}(\theta) = \frac{1}{2} |\hat{u}(0; \hat{\theta}) - u_A|^2 + \frac{1}{2} |\hat{u}(B; \hat{\theta}) - u_B|^2 = \frac{1}{2} |u(A; \theta) - u_A|^2 + \frac{1}{2} |u(B; \theta) - u_B|^2 = L_{bd}(\theta)$$

值得注意的是，在伸缩变换后，内部的损失函数变成了原来的 $(B-A)^2$ 倍，而边界上的损失函数不变。为了平衡边界和内部，我们把惩罚系数改成 $\hat{\beta} = (B-A)^2 \beta$ 。

总体的损失函数就变成了

$$\hat{L}(\hat{\theta}) = \hat{L}_{it}(\hat{\theta}) + \hat{\beta} \hat{L}_{bd}(\hat{\theta}) = (B-A)^2 L(\theta)$$

我们选取学习率为 $\hat{\eta} = \eta / (B-A)^2$ 训练单位区间上的网络，可以得到对输入进行线性变换的结果。

通过类似的推导可以得到：在直接法中，对网络的输入和输出进行线性变换相当于改变网络的最后一层参数的初始值，训练网络的学习率，和边界的惩罚系数。

3.4 输入变换对 Ritz 法解方程的影响

用 Ritz 方法求解方程??，两层神经网络的定义如??，损失函数的定义同样分为两部分。

内部的损失函数：

$$L_{it}(\theta) = \frac{1}{B-A} \int_A^B \frac{1}{2} |u'(x)|^2 + f(x) u(x) dx \quad (3-13)$$

边界的损失函数：

$$L_{bd}(\theta) = \frac{1}{2} |u(A; \theta) - u_A|^2 + \frac{1}{2} |u(B; \theta) - u_B|^2 \quad (3-14)$$

总体的损失函数是由两部分乘以惩罚系数相加：

$$L(\theta) = L_{it}(\theta) + \beta L_{bd}(\theta) \quad (3-15)$$

梯度下降的过程中，参数的变化如??所示。同样地，我们单位区间 $[0, 1]$ 上求解方程??。在单位区间上定义一个新的网络同??，新的参数初始值同??。

在初始时刻，参数满足??的关系，网络的导数满足

$$\frac{d\hat{u}}{d\hat{x}}(\hat{x}; \hat{\theta}) = (B - A) \frac{du}{dx}(x; \theta)$$

在初始时刻，单位区间上的内部损失函数为

$$\hat{L}_{it}(\hat{\theta}) = \int_0^1 \frac{1}{2} |\hat{u}'(\hat{x})|^2 + \hat{f}(\hat{x}) \hat{u}(\hat{x}) d\hat{x} = \frac{1}{B - A} \int_A^B (B - A)^2 \left(\frac{1}{2} |u'(x)|^2 + f(x) u(x) \right) dx = (B - A)^2 L_{it}(\theta)$$

这里就得到了和直接法相同的结论。在伸缩变换后，内部的损失函数变成了原来的 $(B - A)^2$ 倍，而边界上的损失函数不变。我们把惩罚系数改成 $\hat{\beta} = (B - A)^2 \beta$ ，选取学习率为 $\hat{\eta} = \eta / (B - A)^2$ 训练单位区间上的网络，可以得到完全相同的结果。

3.5 小结

一系列的结论表明，无论是拟合函数还是求解方程，对网络的输入和输出进行线性变换，就相当于改变网络的最后一层参数的初始值，训练网络的学习率，和边界的惩罚系数。这些参数通常被认为对网络的性能有很大的影响。因此，调节网络的输入和输出也是提升网络性能的重要手段。

此外，本章中验证了不同输入和输出范围之间的变换关系。对于输入范围不同的问题，我们只要考虑输入范围是单位区域 $[0, 1]^d$ 上，输出范围为 $[0, 1]$ 的问题，确定最优的学习率和惩罚系数，再伸缩到目标区间就可以得到相同的结果。这可以大大减小我们调节参数过程中的工作量。

第四章 多尺度神经网络

4.1 多尺度网络结构

根据前文所说，对网络的输入进行线性变换可以改变网络的性能，但是针对某个具体的问题和网络，我们并不知道如何确定对网络进行变换的系数。这里我们用不同的系数对网络变换，之后把不同系数下的网络相加。实验表明，这样的策略具有一定的鲁棒性。基于这一思想，我们提出如下的多尺度网络。

如图??，我们对输入乘以不同的系数 a_i ，把变换后的数据输入不同的子网络，经过不同子网络的处理后，把所有子网络的输出相加得到结果。通常来说，尺度变换的系数可以选取为均匀划分 $a_i = 1, 2, 3, 4, \dots$ ，或者选取为指数划分 $a_i = 1, 2, 4, 8, \dots$ 。通常的全连接网络可以看做是只有一个尺度，且变换系数为 1 的多尺度网络。

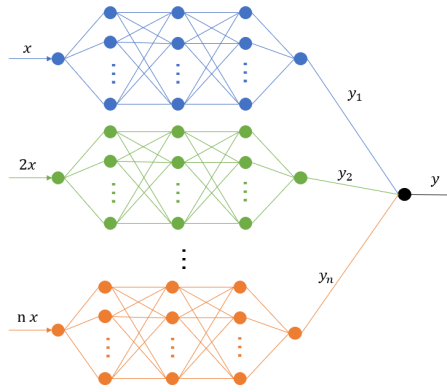


图 4.1 多尺度网络结构示意图

为了提升多尺度网络的尺度分离能力，我们借鉴了小波分析中的思想，使用具有紧支集的激活函数。与普通的 ReLU 激活函数相比，在多尺度网络中，具有紧支集的激活函数通常更有效。两种紧支集的激活函数的具体表达式为

$$\text{sReLU}(x) = (x)_+(1 - x)_+ \quad (4-1)$$

和

$$\text{Phi}(x) = (x - 0)_+^2 - 3(x - 1)_+^2 + 3(x - 2)_+^2 - (x - 3)_+^2 \quad (4-2)$$

其中 $x_+ = \max\{x, 0\} = \text{ReLU}(x)$ 。

4.2 不同参数下网络的有效性

这里，我们验证多尺度网络在不同的参数设定之间的差别。具体的实验分为以下部分。首先，我们测试不同的网络结构，展示多尺度网络和普通全连接网络之间的区别，

同时验证指数划分在多尺度网络中效果更好。随后，我们展示了 $\text{sReLU}(x)$ 和 $\text{Phi}(x)$ 这两种具有紧支集的激活函数远优于常用的 $\text{ReLU}(x)$ 激活函数。最后，我们通过实验，深入探究了多尺度网络改善网络性能的本质。

在接下来的实验中，除非特殊说明，我们用“Normal”表示规模为 1-900-900-900-1 的普通全连接网络。用“Mscale”表示多尺度网络，具体结构由 6 个规模为 1-150-150-150-1 的多尺度网络，系数为 $\{1, 2, 4, 8, 16, 32\}$ 。激活函数默认为 $\text{Phi}(x)$ 。

4.2.1 测试样例

为了测试不同参数下各种网络的表现，这里我们考虑了一维和二维情况下，拟合函数和求解椭圆方程的问题。

拟合问题：拟合目标函数 $F : [-1, 1]^d \rightarrow \mathbb{R}$ ，表达式为

$$F(x) = \sum_{j=1}^d g(x_j) \quad x_j \in [-1, 1] \quad (4-3)$$

其中 $x = (x_1, \dots, x_d)$ ，且

$$g(x) = e^{-x^2} \sin(\mu x^2)$$

在一维的情况下，我们选取 $\mu = 70$ ，在二维的情况下，选取 $\mu = 30$ 。两种情况下的目标函数图像如图??。每步中我们随机在区域 $[-1, 1]^d$ 内选取 5000 个训练数据和 500 个测试数据点。

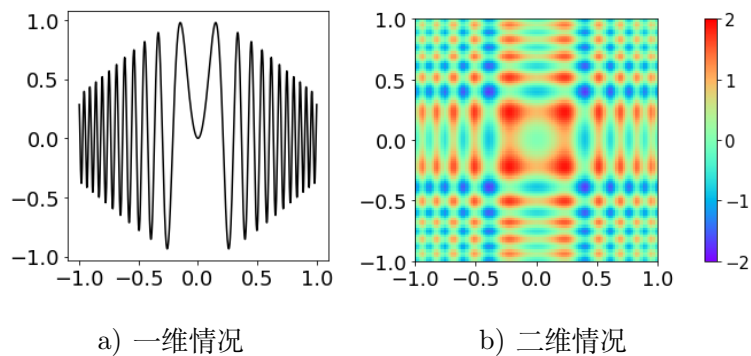


图 4.2 目标函数图像

椭圆方程：求解椭圆方程??，其中参数选取为 $a(x) = 1$ ， $c(x) = \lambda^2$ ，求解区域为 $\Omega = [-1, 1]^d$ ，右端项表达式为

$$f(x) = \sum_{i=1}^d (\lambda^2 + \mu^2) \sin(\mu x_i)$$

方程的精确解为

$$u(x) = \sum_{i=1}^d -\frac{\sin \mu}{\sinh \lambda} \sinh(\lambda x_i) + \sin(\mu x_i)$$

对应的边界条件由精确解生成。

对于一维的情况，参数选取为 $\lambda = 20, \mu = 50$ ，对于二维的情况，选取 $\lambda = 2, \mu = 30$ 。一维和二维的方程的精确解如图??。每步中我们随机在区域 $[-1, 1]^d$ 内选取 5000 个均匀分布的积分点，对于一维的情况，我们在边界上生成 400 个积分点，而二维的情况，我们生成 4000 个积分点。边界惩罚系数选取为 $\beta = 1000$ 。

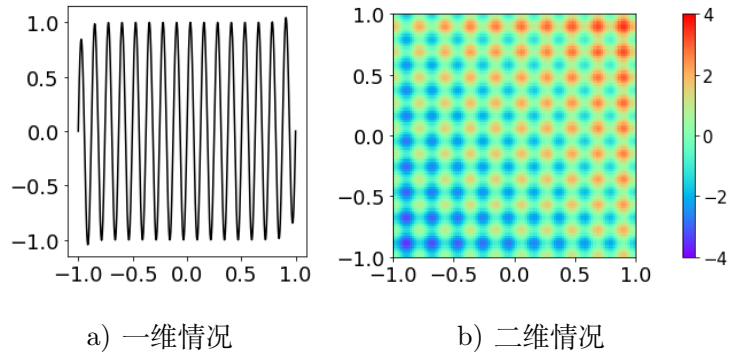


图 4.3 椭圆方程精确解的图像

4.2.2 变换系数对网络效果的影响

这里我们测试不同的尺度分割对网络性能的影响，考虑如下的系数选取方案：

1. 规模为 1-900-900-900-1 的全连接网络。(Normal)
2. 6 个规模为 1-150-150-150-1 的多尺度网络，系数为 $\{1, 1, 1, 1, 1, 1\}$ 。(Mscale(1))
3. 3 个规模为 1-300-300-300-1 的多尺度网络，系数为 $\{1, 2, 3\}$ 。(Mscale(3))
4. 3 个规模为 1-300-300-300-1 的多尺度网络，系数为 $\{1, 2, 4\}$ 。(Mscale(4))
5. 6 个规模为 1-150-150-150-1 的多尺度网络，系数为 $\{1, 2, 3, 4, 5, 6\}$ 。(Mscale(6))
6. 6 个规模为 1-150-150-150-1 的多尺度网络，系数为 $\{1, 2, 4, 8, 16, 32\}$ 。(Mscale(32))

求解结果如图??和??，图中画出了求解方程的误差随迭代步数下降的图像。由于误差的大小取决于积分点的选取，会受到随机因素的影响，我们对图像进行了平滑。图中浅色部分表示误差波动的范围，深色线是误差下降曲线平滑之后的结果。

从图中可以看出，对于拟合问题，多尺度网络的优势并不明显，但是对于求解椭圆方程，多尺度网络的性能比通常的全连接网络更好。需要注意的是，随着尺度变换范围的扩大，多尺度网络更快地解决了问题。在所有尺度均为 1 的情况下，多尺度网络和普通的全连接网络性能相似。实验表明，具有适当尺度的网络结构在求解椭圆方程中更有效，指数划分的尺度也比均匀划分的尺度更有效。

4.2.3 激活函数对网络效果的影响

我们通过一维拟合问题和一维椭圆方程问题，检验不同激活函数的有效性。

我们使用三种不同的激活函数，即 ReLU、sReLU、Phi，分别在“Normal”和“Mscale”网络结构下求解上述问题。求解结果如图??和??。图中画出了求解方程和拟合函数的误

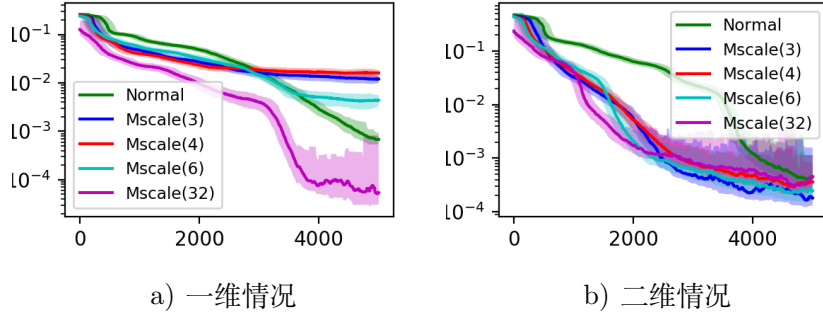


图 4.4 不同的尺度分割在一维和二维拟合任务中的表现

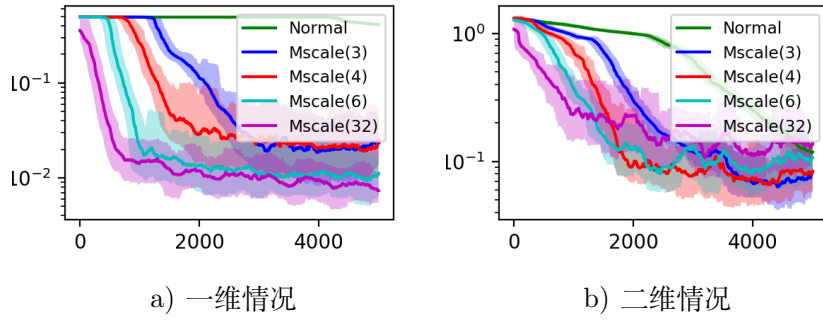


图 4.5 不同的尺度分割在一维和二维椭圆方程中的表现

差随迭代步数下降的图像。

结果表明，在一般的全连接网络结构中，改用紧支集的激活函数对网络的效果提升不大。在求解一维椭圆方程的过程中 $\Phi(x)$ 甚至没有收敛。另一方面，对于多尺度网络结构，紧支集激活函数 sReLU、 Φ 的性能都比 ReLU 好得多。这说明在多尺度网络中，采用紧支集的激活函数是十分必要的。

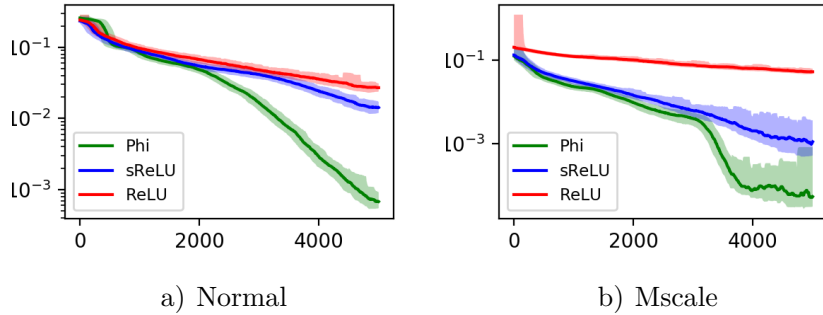


图 4.6 不同激活函数在一维拟合问题中的表现

4.2.4 提升网络效果的本质因素

为了揭示多尺度方法提升网络性能的本质因素，我们考虑不同的网络结构，用相同的系数求解前文中一维和二维的椭圆方程。

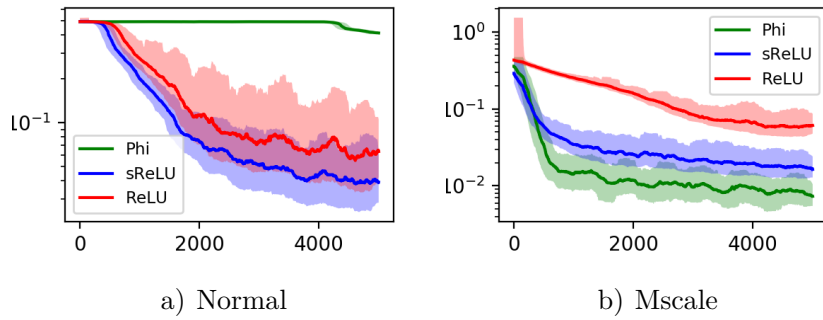


图 4.7 不同激活函数在一维椭圆方程中的表现

测试的网络结构如下：

1. 1 个规模为 1-900-900-900-1 的多尺度网络，系数为 {1}。 (Normal(1))
2. 1 个规模为 1-900-900-900-1 的多尺度网络，系数为 {16}。 (Normal(32))
3. 6 个规模为 1-150-150-150-1 的多尺度网络，系数为 {1, 1, 1, 1, 1, 1}。 (Mscale(1))
4. 6 个规模为 1-150-150-150-1 的多尺度网络，系数为 {1, 2, 4, 8, 16, 32}。 (Mscale(s))
5. 6 个规模为 1-150-150-150-1 的多尺度网络，系数为 {32, 32, 32, 32, 32, 32}。 (Mscale(32))

这里第一个网络就是没有尺度变换的普通全连接网络，第二个网络也并没有采用子网络的结构，只是把全连接网络输入的部分扩大了 32 倍。第四个网络是我们前文中实验得到的较优的网络结构。而在第三个和第五个网络中，我们对尺度没有做任何划分，所有的子网络输入都扩大了相同的倍数。

求解结果如图??，图中画出了近似解的误差随迭代步数下降的图像。从图中可以看出，影响多尺度网络性能的主要因素实际上是改变了输入层的初始化参数，只要出现了尺度变换，网络的性能总是与最大的那个尺度变换的系数有关。对尺度进行指数分割可以提升性能，本质上也是提高了尺度变换的最大系数。

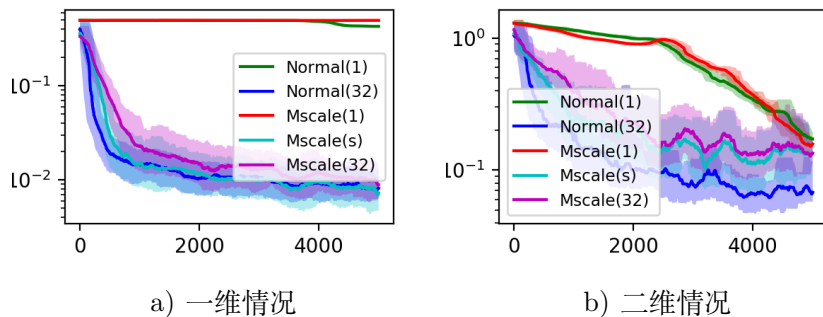


图 4.8 不同网络在一维和二维椭圆方程中的表现

第五章 多尺度网络在求解椭圆方程中的应用

这里我们应用带有激活函数为 $\text{Phi}(x)$ 的多尺度网络来求解复杂的椭圆型方程。具体实验包括求解具有宽频率范围、变系数、环形区域上和多孔的立方区域上的椭圆方程。通过这些实验，我们令人信服地证明了多尺度网络是一种有效且易于实现的求解复杂椭圆方程的无网格方法。

5.1 宽频域振荡的椭圆方程

求解椭圆方程 $\Delta u = f(x)$ ，其中参数选取为 $a(x) = 1$, $c(x) = 0$ ，求解区域为 $\Omega = [-1, 1]^d$ ，右端项表达式为

$$f(x) = \sum_{i=1}^d 4\mu^2 x_i^2 \sin(\mu x_i^2) - 2\mu \cos(\mu x_i^2) \quad (5-1)$$

方程的精确解为

$$u(x) = \sum_{i=1}^d \sin(\mu x_i^2) \quad (5-2)$$

对应的边界条件由精确解生成。

每步中我们随机在区域 $[-1, 1]^d$ 内选取 5000 个均匀分布的积分点，在边界上生成 4000 个均匀分布的积分点。这个问题的特殊之处在于，精确解虽然是高频振荡的，但是它没有固定的频率，而是有一个广泛的频率范围。对于 $\mu = 15$ ，二维情况下问题的精确解和两种网络结构下的近似解如图??所示。图中指出，多尺度网络得到的解很好地捕捉到不同尺度的振荡。例如，在红圈所示区域，普通的全连接网络没有捕捉到解的振荡，而多尺度网络的解很好地展示出了振荡的形态。在四个角处的振荡也出现了类似的行为。

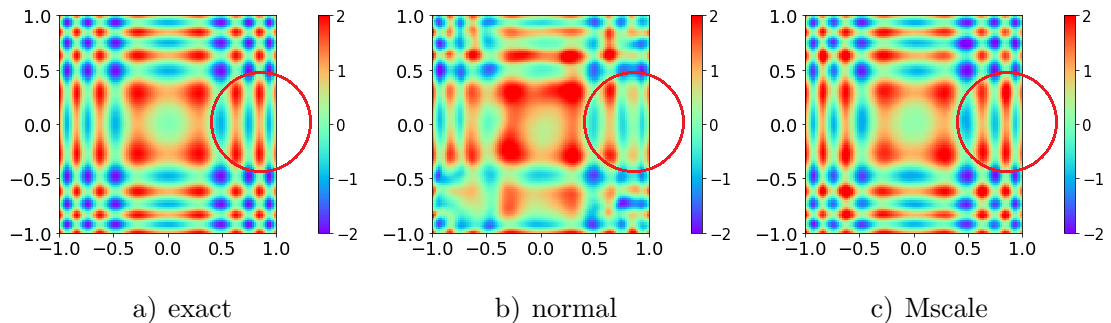


图 5.1 二维情况下椭圆方程的精确解和数值解图像

求解结果如图??，图中画出了求解方程的误差随迭代步数下降的图像。在求解二维和三维的方程中，多尺度网络都表现出了明显的优势。

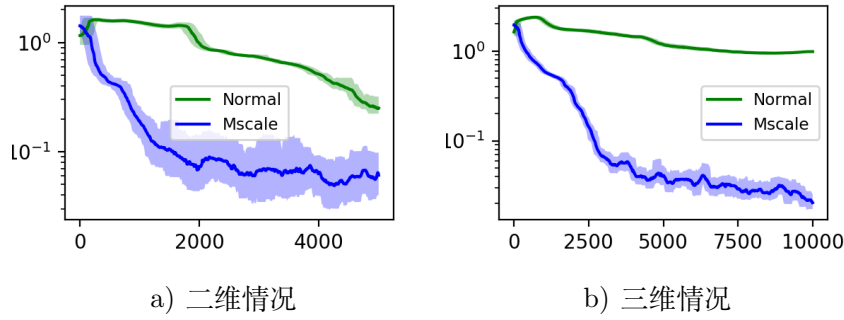


图 5.2 二维和三维情况下两种网络结构的表现

5.2 环形区域上的椭圆方程

求解椭圆方程^{??}，其中参数选取为 $a(x) = 1$ ， $c(x) = 0$ ，求解区域为一个以原点为中心的圆环，内径为 1，外径为 3。方程的右端项表达式为

$$f(x) = \mu^2 J_0(\mu|x - x_0|) \quad (5-3)$$

其中 J_0 是 Bessel 函数，方程的精确解为

$$u(x) = J_0(\mu|x - x_0|) \quad (5-4)$$

对应的边界条件由精确解生成。位移选取为 $x_0 = (0.5, 0)$ ，参数选取为 $\mu = 5$ 和 $\mu = 10$ 分别实验。

每步中我们随机在区域内选取 5000 个均匀分布的积分点，在边界上生成 4000 个均匀分布的积分点。问题的精确解和两种网络结构下的近似解如图^{??}和图^{??}所示。图中用黑色圆圈标记的区域表示解在此处有最大的振幅，正常的全连接网络完全不能刻画振荡，而多尺度网络在这两种情况下都可以得到和精确解类似的效果。

同样，如图^{??}所示，图中画出了求解方程的误差随迭代步数下降的图像。在迭代同样的步数后，多尺度网络的误差更小。

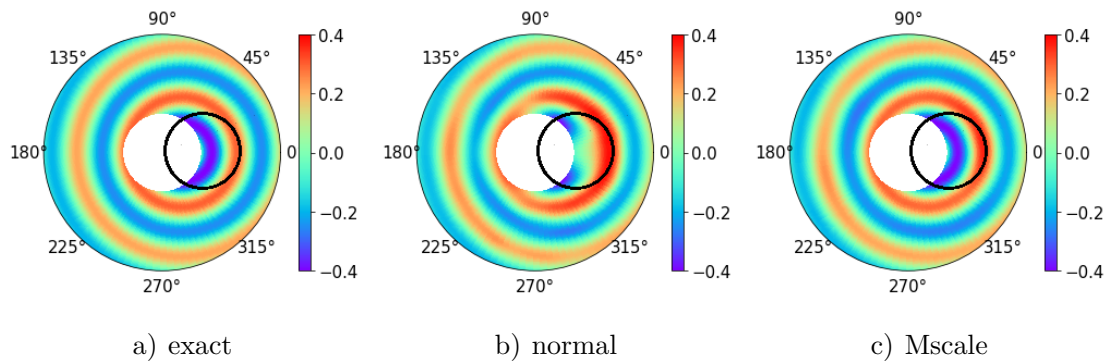


图 5.3 $\mu = 5$ 情况下椭圆方程的精确解和数值解图像

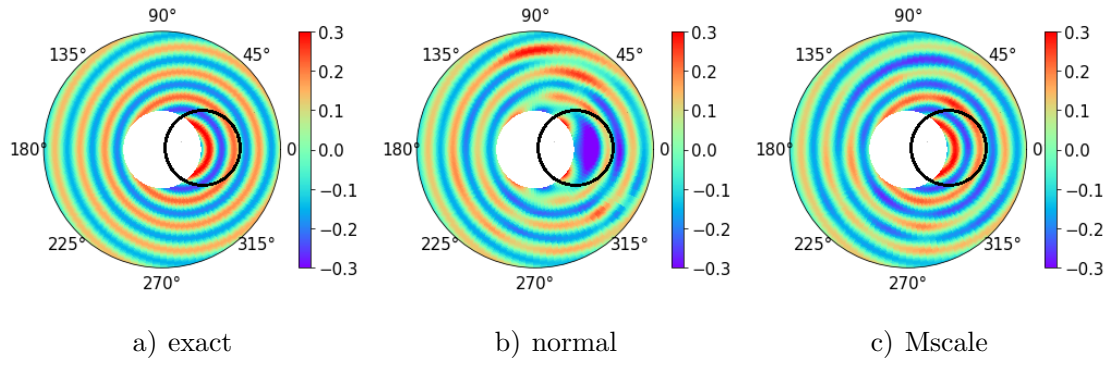


图 5.4 $\mu = 10$ 情况下椭圆方程的精确解和数值解图像

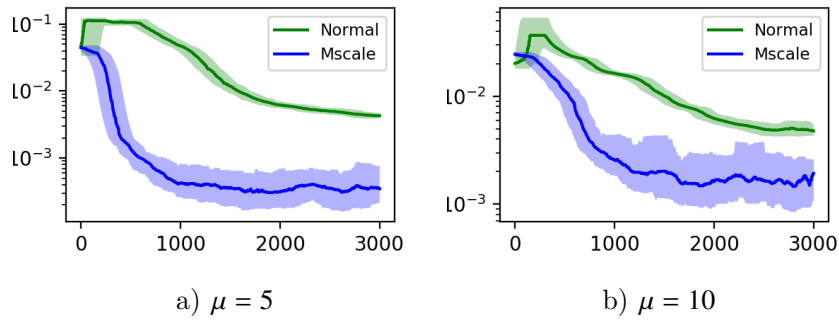


图 5.5 不同参数下两种网络结构的表现

5.3 有孔区域上的椭圆方程

这里我们考虑如下两个区域：

区域 1： 方形区域 $[-1, 1]^2$ 内有三个圆形的孔。三个圆孔的中心分别为 $(-0.5, -0.5)$ 、 $(0.5, 0.5)$ 和 $(0.5, -0.5)$ ，半径分别为 0.1、0.2 和 0.2。在训练过程中，每步我们随机在区域内选取 5000 个均匀分布的积分点，在方形区域的边界上生成 3000 个积分点，每个大圆孔的边界上生成 800 个积分点，每个小圆孔的边界上生成 400 个积分点。

区域 2 方形区域 $[-1, 1]^2$ 内有四个孔。三个圆形孔的中心分别为 $(-0.6, -0.6)$ 、 $(0.3, -0.3)$ 和 $(0.6, 0.6)$ ，半径分别为 0.3、0.6 和 0.3。椭圆形孔由方程 $16(x + 0.5)^2 + 64(y - 0.5)^2 = 1$ 表示。每步我们随机在区域内选取 5000 个均匀分布的积分点，外部边界、大圆孔边界、小圆孔边界和椭圆孔边界生成的积分点数量分别为 2400、1100、550 和 400。

求解椭圆方程??，其中参数选取为 $a(x) = 1$ ， $c(x) = 0$ 。方程的右端项表达式为

$$f(x) = 2\mu^2 \sin \mu x_1 \sin \mu x_2 \quad (5-5)$$

方程的精确解为

$$u(x) = \sin \mu x_1 \sin \mu x_2 \quad (5-6)$$

对应的边界条件由精确解生成。其中的参数选取为 $\mu = 7\pi$ 。

问题的精确解和两种网络结构下的近似解如图??和图??所示。显然，多尺度网络能更好地刻画精确解中的振荡。

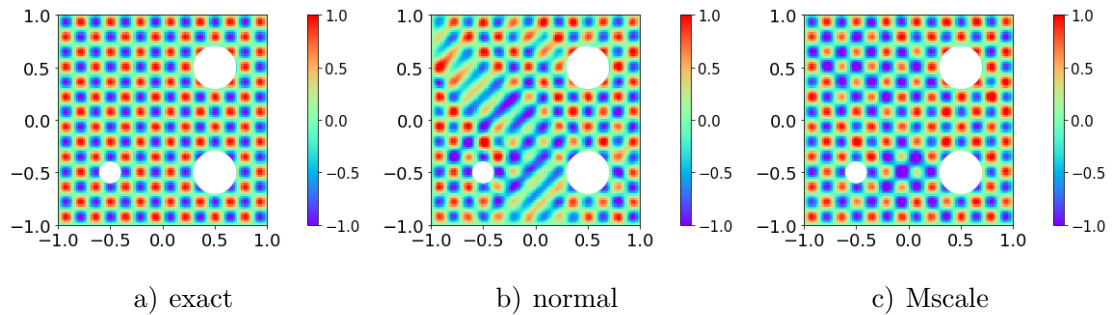


图 5.6 区域 1 上椭圆方程的精确解和数值解图像

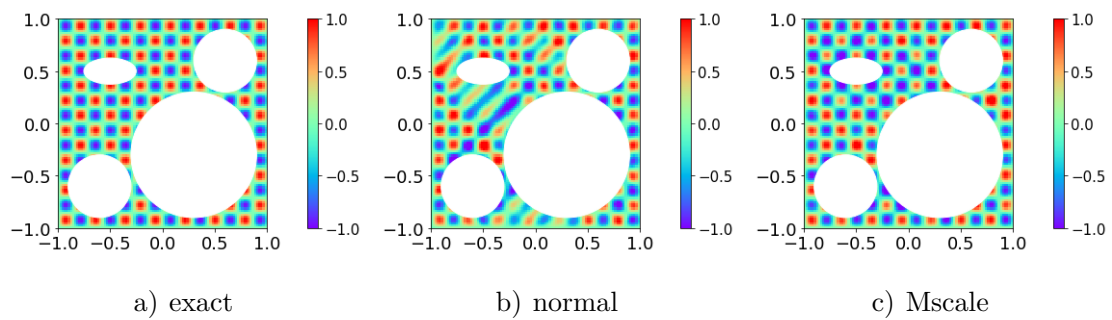


图 5.7 区域 2 上椭圆方程的精确解和数值解图像

同样，如图??所示，图中画出了求解方程的误差随迭代步数下降的图像，多尺度网络的误差下降速度更快，精度也更高。

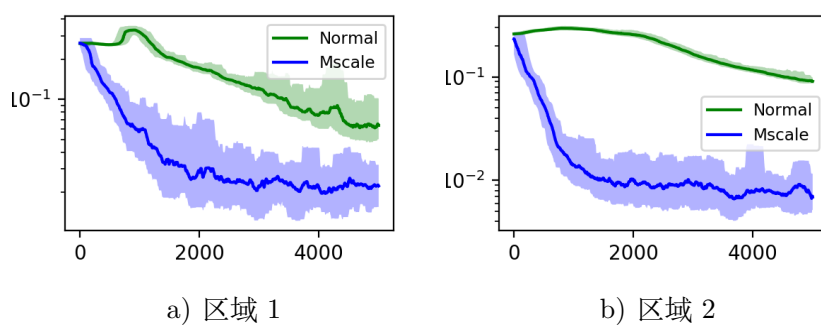


图 5.8 两种区域下不同网络结构的表现

5.4 多孔区域上的椭圆方程

为了验证多尺度网络求解复杂区域上椭圆方程的能力，我们考虑了一个三维立方体 $[-1, 1]^3$ ，内部移除了 125 个孔，如图??，这些球形孔的球心位于正方体内部的均匀网格 $\{-0.8, -0.4, 0, 0.4, 0.8\}^3$ 上。这些孔的半径是从范围在 $[0, 0.15]$ 中的均匀分布随机抽样

产生，这样可以保证各个孔之间没有相交。在训练过程中，每步我们随机在区域内选取 5000 个均匀分布的积分点，外边界生成 2500 个积分点，内部的孔上 1500 个（每个孔 12 个）积分点。

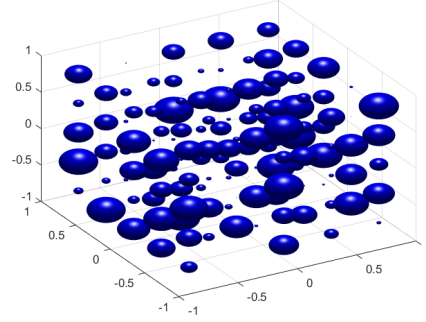


图 5.9 多孔区域示意图，蓝色表示在区域内挖去的部分

求解椭圆方程^{??}，其中参数选取为 $a(x) = 1$, $c(x) = 0$, $\mu = 7\pi$ 。我们测试如下三组示例，它们对应的精确解分别为

1. 示例 1: $u(x) = \sin \mu x_1 \sin \mu x_2 \sin \mu x_3$

2. 示例 2: $u(x) = e^{\sin \mu x_1 + \sin \mu x_2 + \sin \mu x_3}$

3. 示例 3: $u(x) = e^{\sin \mu x_1 \sin \mu x_2 \sin \mu x_3}$

对应的边界条件和右端项由精确解生成。

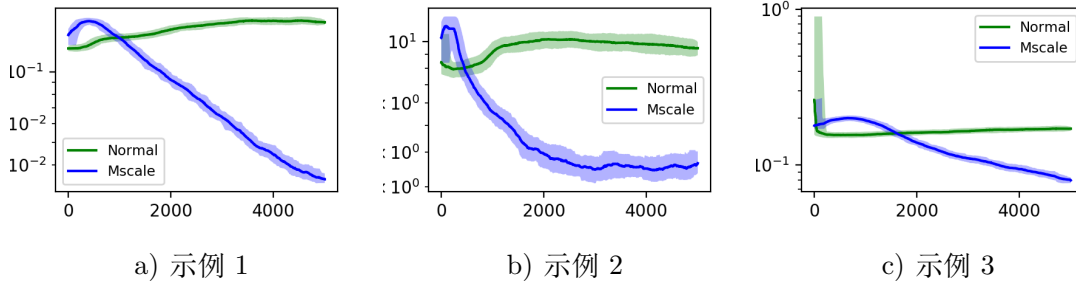


图 5.10 三个示例中不同网络结构的表现

计算结果如图^{??}所示，在三维区域上，我们难以画出精确解和近似解的图像，这里只给出了误差随迭代步数变化的图像。对于这三种情况，正常的全连通网络结构对于此类复杂问题根本不收敛，在迭代的过程中误差几乎不下降，而多尺度网络虽然在计算开始时误差上升，但是在一定的步数之后就可以明显达到更小的误差。

5.5 变系数椭圆方程

求解椭圆方程^{??}，其中参数选取为 $a(x) = 1$ ，方程的右端项表达式为

$$f(x) = (\mu_1^2 + \mu_2^2 + \mu_3^2 + x_1^2 + 2x_2^2 + 3x_3^2) \sin(\mu_1 x_1) \sin(\mu_2 x_2) \sin(\mu_3 x_3) \quad (5-7)$$

系数表达式为

$$c(x) = (x_1^2 + 2x_2^2 + 3x_3^2) \quad (5-8)$$

方程的精确解为

$$u(x) = \sin(\mu_1 x_1) \sin(\mu_2 x_2) \sin(\mu_3 x_3) \quad (5-9)$$

对应的边界条件由精确解生成。我们选取各向异性的参数，其中 $\mu_1 = 15, \mu_2 = 20, \mu_3 = 25$ 。

每步中我们随机在求解区域 $[-1, 1]^d$ 内选取 5000 个均匀分布的积分点，在边界上生成 4000 个均匀分布的积分点。

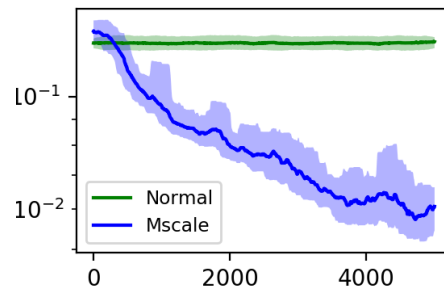


图 5.11 两种网络结构的表现

如图??所示，在训练过程中，多尺度网络的误差显著衰减，而正常的全连接网络误差几乎保持不变。多尺度网络以更高的精度更快地解决了问题。

第 六 章 多尺度网络在分子计算中的应用

这里我们展示多尺度网络在求解 Poisson-Boltzmann 方程 (PB 方程) 中的应用。PB 方程用于描述生物分子在溶液中的电势分布，在分子模拟中有重要的应用。

6.1 PB 方程简介

考虑一个有界开区域 $\Omega_1 \subset \mathbb{R}^3$ ，它将全空间 \mathbb{R}^3 分成两个不相交的子区域，曲面 $\Gamma = \partial\Omega_1$ 。在物理模型中， Ω_1 表示生物分子，而 $\Omega_2 = \mathbb{R}^3 \setminus \Omega_1$ 是溶剂区域。它们之间的边界 Γ 一般不规则。

在整个区域上，PB 方程表示为

$$-\nabla(\epsilon(x)\nabla u(x)) + \kappa(x)u(x) = f(x) \quad x \in \mathbb{R}^3 \quad (6-1)$$

其中 $\epsilon(x)$ 是离子溶剂的介电常数， $\kappa(x)$ 是离子溶剂的 Debye-Huckel 长度的倒数。在实际问题中，介电常数在界面 Γ 处一般不连续。

解在界面上的边界条件满足连续性

$$[u](x) = 0 \quad x \in \Gamma \quad (6-2)$$

和流连续性

$$\left[\epsilon \frac{\partial u}{\partial n}\right](x) = 0 \quad x \in \Gamma \quad (6-3)$$

其中 $[\cdot]$ 表示在边界上的跳跃。取无穷远点为电势零点，即

$$\lim_{|x| \rightarrow \infty} u_2(x) = 0 \quad (6-4)$$

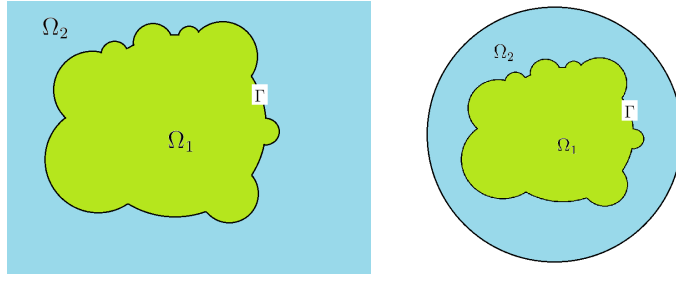
为了处理无界区域，我们将求解区域截断成一个大的球或立方体，用 Ω 表示，它满足 $\Omega_1 \subset \Omega$ 。这里我们重新定义 $\Omega_2 = \Omega \setminus \Omega_1$ ，并在 Ω 的边界上设置一个近似边界条件 $u = 0$ 。对于区域的截断，高阶的人工边界条件已经得到了广泛的研究。但是由于我们更感兴趣的是界面 Γ 附近神经网络方法的性能，这里对区域的截断比较粗糙。图??展示出了对区域的截断。

6.2 求解 PB 方程的有限差分法

这里我们介绍一种简单的有限差分方法，用于求解 PB 方程并与深度学习方法进行对比。

在 PB 方程中，我们可以定义通量为 $F(x) = \epsilon(x)\nabla u(x)$ 。尽管 $\epsilon(x)$, $\kappa(x)$ 和 $u'(x)$ 可能是不连续的，根据边界条件我们可以得到， $u(x)$ 和 $F(x)$ 连续且几乎处处可导。

考虑三维均匀网格上的数值解 $u_{i,j,k}$ ，网格步长为 h ，我们有



a) 截断前 b) 截断后

图 6.1 无界区域截断示意图

$$\frac{\partial u}{\partial x}(x_{i+\frac{1}{2}}, y_j, z_k) \approx \delta_x u_{i+\frac{1}{2}, j, k}$$

其中 $x_{i+\frac{1}{2}} = \frac{x_i + x_{i+1}}{2}$ 和 $\delta_x u_{i+\frac{1}{2}, j, k} = \frac{u_{i+1, j, k} - u_{i, j, k}}{h}$ 。这里我们可以定义数值通量为

$$(F_1)_{i+\frac{1}{2}, j, k} = \epsilon(x_{i+\frac{1}{2}}, y_j, z_k) \delta_x u_{i+\frac{1}{2}, j, k}$$

因此可以得到

$$\frac{\partial F_1}{\partial x}(x_{i+\frac{1}{2}}, y_j, z_k) \approx \delta_x (F_1)_{i, j, k}$$

这就是说

$$\frac{\partial}{\partial x}(\epsilon \frac{\partial u}{\partial x})(x_i, y_j, z_k) \approx \delta_x (\epsilon \delta_x u)_{i, j, k}$$

其中

$$\delta_x (\epsilon \delta_x u)_{i, j, k} = \epsilon(x_{i+\frac{1}{2}}, y_j, z_k) u_{i+1, j, k} - \epsilon(x_{i-\frac{1}{2}}, y_j, z_k) u_{i-1, j, k} + (\epsilon(x_{i+\frac{1}{2}}, y_j, z_k) + \epsilon(x_{i-\frac{1}{2}}, y_j, z_k)) u_{i, j, k}$$

在对其它方向进行类似的处理之后，我们可以得到最终的有限差分格式为

$$-(\delta_x (\epsilon \delta_x u)_{i, j, k} + \delta_y (\epsilon \delta_y u)_{i, j, k} + \delta_z (\epsilon \delta_z u)_{i, j, k}) + \kappa(x_i, y_j, z_k) u_{i, j, k} = f(x_i, y_j, z_k)$$

6.3 具有几何奇异性的 PB 方程

考虑到分子计算中的问题通常具有几何奇异性，我们构造具有几何奇异性的区域如下。首先选择一个中心在位于原点，半径为 0.5 的大球。在大球表面随机抽取 20 个点，以这些点为中心生成许多小球。小球的半径从 [0.1, 0.2] 中随机抽取。选取 Ω_1 是这些小球和大球的并集。 Ω_1 的形状如图??所示。各个球之间的相交会产生几何奇异性，这对传统的有限元法和边界元法的网格生成以及精确求解提出了很大的挑战。

我们在此区域上，首先考虑如下两个示例：

示例 1： 方程的精确解表示为：

$$u(x) = \frac{e^{\sin \mu x_1 + \sin \mu x_2 + \sin \mu x_3}}{|x|^2 + 1} (|x|^2 - 1) \quad (6-5)$$

方程的系数选取为

$$\mu = 15, \epsilon(x) = 1, \kappa(x) = 1 \text{ for } x \in \Omega_1, \epsilon(x) = 1, \kappa(x) = 5 \text{ for } x \in \Omega_2. \quad (6-6)$$

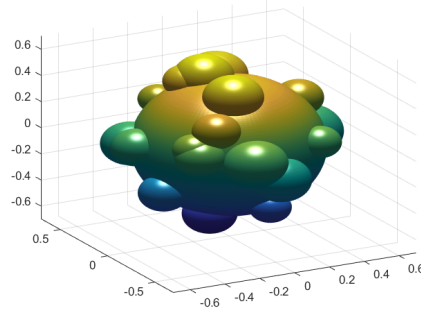


图 6.2 具有几何奇异性的区域

方程的右端项由精确解生成。

求解区域被一个球心位于原点，半径为 1 的大球截断。在球面上，精确解满足对应的边界条件。

示例 2: 方程中，右端项选取为

$$f(x) = \frac{e^{\sin \mu x_1 + \sin \mu x_2 + \sin \mu x_3}}{|x|^2 + 1} (|x|^2 - 1) \quad (6-7)$$

方程的系数选取为

$$\mu = 20, \epsilon(x) = 1 \text{ for } x \in \Omega_1, \epsilon(x) = 80 \text{ for } x \in \Omega_2, \kappa(x) = 1. \quad (6-8)$$

此时方程的精确解无法写出。在这种情况下，求解区域被立方体 $[-1, 1]^3$ 截断，参考解通过有限差分法 (FDM) 计算。在差分法中，我们选取了具有足够的精细的网格以确保精度。

在示例 1 中，在每个训练步，我们在区域 Ω 内采样 5000 个点，在边界 $\partial\Omega$ 上采样 4000 个点。在示例 2 中，我们对区域 Ω 内的 6000 个点和边界 $\partial\Omega$ 上的 3000 个点进行采样。需要注意的是，由于我们使用了 Ritz 方法，我们无需在界面 Γ 上采样，所有的跳跃边界条件都是自动满足的。

如图??所示。多尺度网络与普通的全连接网络相比，误差下降得更快。在两个示例中，多尺度网络都显示了很大的优势。

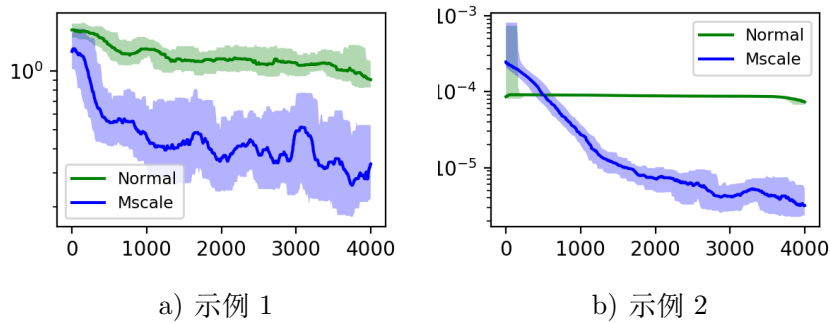


图 6.3 不同网络结构在求解 PB 方程中的表现

我们在图??展示了示例 2 中，不同方法得到的近似解在横截面 $x_1 = x_3 = 0$ 上的取值，包括 FDM ($h = 0.02$)、normal(训练 4000 步) 和 Mscale(训练 4000 步)。普通的全连通网络在区域的内部给出了错误的解，而多尺度网络的结果和参考解相近。这些实例表明，相比于普通的全连接网络，多尺度网络求解复杂区域上的 PB 方程的速度更快，精度更高。

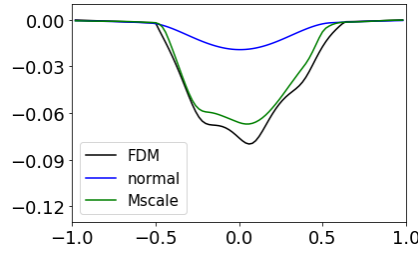


图 6.4 示例 2 中不同方法得到的近似解在截面 $x_1 = x_3 = 0$ 上的图像

6.4 具有源项奇异性的 PB 方程

在实际的生物分子中，电荷往往以点电荷的形式分布在分子内部，这就相当于方程中的右端项变为

$$f(x) = \sum_{k=1}^K q_k \delta(x - s_k) \quad (6-9)$$

其中 $\delta(x)$ 是 Dirac 函数， q_k 和 s_k 分别表示分子中点电荷的带电量 and 位置。我们假设核与分子界面之间的距离大于常量 R_0 。也就是说我们可以选取 $\Omega_0 = \{x : \exists k, |x - s_k| < R_0\} \subset \Omega_1$ 。在图??中，蓝色部分表示溶剂域 Ω_2 ，绿色部分表示生物分子区域 $\Omega_1 \setminus \Omega_0$ ，粉红色部分表示 Ω_0 。这里 Ω_0 是一些球的并集，其中已经包含了所有的点电荷。

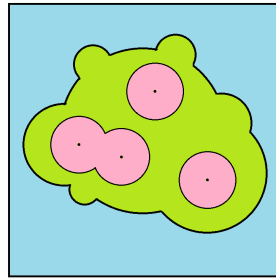


图 6.5 点电荷分布示意图

这里我们给出一种克服源项奇异性的方法。我们首先定义

$$\bar{u}(x) = \sum_{k=1}^K q_k G(x - s_k) m(x - s_k) \quad (6-10)$$

其中格林函数

$$G(x) = \frac{1}{4\pi\epsilon_1} \frac{e^{-\frac{\kappa_1}{\sqrt{\epsilon_1}}|x|}}{|x|} \quad (6-11)$$

截断函数

$$m(x) = \begin{cases} 1 - \left(\frac{|x|}{R_0}\right)^3 \left(4 - 3\frac{|x|}{R_0}\right) & |x| < R_0 \\ 0 & |x| > R_0 \end{cases} \quad (6-12)$$

通过以上定义，我们可以验证 $\bar{u}(x)$ 满足

$$\begin{aligned} -\Delta \bar{u}(x) + \kappa^2 \bar{u}(x) &= \sum_{k=1}^K q_k \delta(x - s_k) + \sum_{k=1}^K q_k F(|x - s_k|) \quad x \in \Omega_0 \\ \bar{u} = \frac{\partial \bar{u}}{\partial n} &= 0 \quad x \in \partial\Omega_0 \\ \bar{u}(x) &= 0 \quad x \in \Omega_0^c \end{aligned} \quad (6-13)$$

我们定义消除奇异性之后的解为

$$w(x) = u(x) - \bar{u}(x)\chi_{\Omega_0}(x) \quad (6-14)$$

奇异性之后的解满足这样一个没有奇异性的方程

$$-\epsilon(x)\Delta w(x) + \kappa^2(x)w(x) = f(x)\chi_{\Omega_0}(x) \quad (6-15)$$

其中

$$f(x) = -\sum_{k=1}^K q_k F(|x - s_k|) \quad (6-16)$$

和

$$F(r) = \begin{cases} \frac{3e^{-\frac{\kappa_1}{\sqrt{\epsilon_1}}r}}{\pi R_0^4} (2R_0 - 3r + 2\frac{\kappa_1}{\sqrt{\epsilon_1}}r^2 - 2R_0\frac{\kappa_1}{\sqrt{\epsilon_1}}r) & r < R_0 \\ 0 & r > R_0 \end{cases} \quad (6-17)$$

这里我们只要求解消去奇异性之后的方程，就可以得到原方程的解了。

下面我们选取示例，验证多尺度网络对消去奇异性之后方程的求解能力。

示例 1: 这里我们选取大区域 $\Omega = [-1, 1]^3$, Ω_1 是一个中心位于原点，半径为 $R = 0.7$ 的球。其它相关参数选择为

$$s = (0, 0, 0), \quad q = 1, \quad R_0 = 0.5, \quad \epsilon(x) = 1 \text{ for } x \in \Omega_1, \quad \epsilon(x) = 80 \text{ for } x \in \Omega_2, \quad \kappa(x) = 0.$$

由于此时的区域不再具有几何奇异性，我们可以算出方程的精确解为

$$u(x) = \begin{cases} \frac{1}{4\pi|x|\epsilon_1} - \left(\frac{1}{\epsilon_1} - \frac{1}{\epsilon_2}\right) \frac{1}{4\pi R} & x \in \Omega_1 \\ \frac{1}{4\pi|x|\epsilon_2} & x \in \Omega_2 \end{cases} \quad (6-18)$$

相应地，去掉奇异性的解为

$$w(x) = \begin{cases} \frac{1}{4\pi\epsilon_1|x|} \left(\frac{|x|}{R_0}\right)^3 \left(4 - 3\frac{|x|}{R_0}\right) - \left(\frac{1}{\epsilon_1} - \frac{1}{\epsilon_2}\right) \frac{1}{4\pi R} & |x| < R_0 \\ \frac{1}{4\pi|x|\epsilon_1} - \left(\frac{1}{\epsilon_1} - \frac{1}{\epsilon_2}\right) \frac{1}{4\pi R} & R_0 < |x| < R \\ \frac{1}{4\pi|x|\epsilon_2} & |x| > R \end{cases} \quad (6-19)$$

示例 2: 在第二个示例中，我们选择大区域 $\Omega = [-1, 1]^3$ 。区域 Ω_1 构造如下。我们选择一个中心为原点，半径 0.7 的大球。在大球表面随机抽取 20 个点作为小球的中心。小球的半径从 $[0.1, 0.3]$ 范围内的均匀分布中随机抽取。 Ω_1 是这些球的并集。带有奇异性的源项构造为。每个电荷的位置在一个球心位于原点，半径 0.5 的球中随机选择，电荷量从 $[-0.5, 0.5]$ 范围内的均匀分布中随机抽样。我们选择 $R_0 = 0.2$ 。如图??，图中蓝色区域表示 Ω_1 ，在蓝色区域内的红色部分表示 Ω_0 。

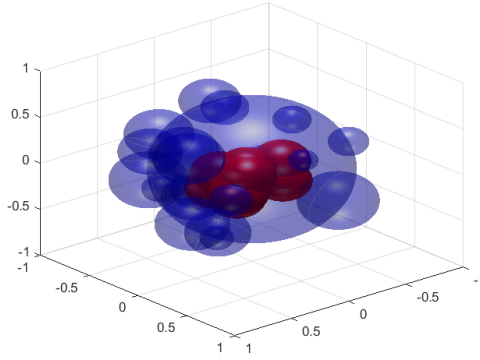


图 6.6 求解区域图像

其它参数选择如下

$$\epsilon(x) = 1 \text{ for } x \in \Omega_1 \quad \epsilon(x) = 80 \text{ for } x \in \Omega_2 \quad \kappa(x) = 0$$

需要注意的是，示例 2 中同时具有几何奇异性和源项的奇异性，这给普通的数值方法带来了很大的挑战。这里我们无法得到精确解，参考解同样由 FDM 给出。

每步中我们随机在区域 $[-1, 1]^d$ 内选取 5000 个均匀分布的积分点，在边界上生成 4000 个均匀分布的积分点。

在示例 1 中，方程具有源项奇异性，我们测试的网络结构如下：

1. 规模为 1-1000-1000-500-1 的全连接网络。(Normal)
2. 5 个规模为 1-200-200-100-1 的多尺度网络，系数为 $\{1, 2, 4, 8, 16\}$ 。(Mscale)

在示例 2 中，方程同时具有几何奇异性和源项奇异性，因此我们需要更多的神经元，使网络具有更强的逼近能力。这里测试的网络结构如下：

1. 规模为 1-1500-1000-1000-500-1 的全连接网络。(Normal)
2. 5 个规模为 1-300-200-200-100-1 的多尺度网络，系数为 $\{1, 2, 4, 8, 16\}$ 。(Mscale)

通过三种方法：FDM ($h = 0.01$)、normal (训练 10000 步) 和 Mscale (训练 10000

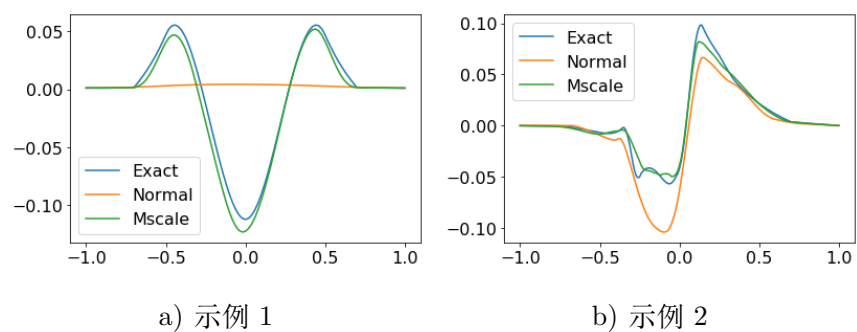


图 6.7 不同方法得到的近似解在截面 $x_2 = x_3 = 0$ 上的图像

步) 得到的截面上的数值解如图??所示。正常全连通网络的输出不能很好地捕捉精确解中的峰值，而多尺度网络得到了更加精确的结果。

第七章 深度学习方法在特征值问题中的应用

这里我们应用深度学习方法来求解椭圆特征值问题。由于特征值问题中，我们要求解的特征函数一般不具有高频震荡的性质，此外，特征值问题的结果随区间变换而改变。这里我们用普通的没有尺度变换的全连接网络求解特征值问题，目的主要是展示深度学习方法在求解特征值问题中的能力。

7.1 高维方形区域上的特征值问题

求解椭圆方程特征值问题^{??}，其中参数选取为 $a(x) = 1$ ， $c(x) = 0$ ，求解区域为 $\Omega = [0, 1]^d$ 。

问题的最小特征值为 $\lambda = d\pi^2$ ，对应的特征函数

$$u(x) = \prod_{i=1}^d \sqrt{2} \sin(\pi x_i) \quad (7-1)$$

我们用 Ritz 方法求解问题，测试规模为 1-1000-1000-1000-1000-1 的全连接网络。激活函数是 ReLU。每次训练区域内部生成 5000 个点，区域边界生成 4000 个点。选取边界惩罚系数为 1000，归一化惩罚系数为 10000。分别在 $d = 3$ ， $d = 5$ 和 $d = 10$ 的情况下求解特征值问题。实验结果如图^{??}，图^{??}，图^{??}。

在这些图中，图 (a) 画出了特征值的近似值和真实值随迭代步数变化的图像，图 (b) 画出了特征函数的近似解和真实解之间的误差。由于高维情况下的特征函数难以画图表示，我们选取两个截面作图。图中，“特征函数 1”画出的是在 $x_1 = x_2 = \dots = x_{d-1} = 0.5$ 截面上的数值解图像，“特征函数 2”画出的是在 $x_1 = x_2 = \dots = x_d$ 对角线上的数值解图像。

实验中展示了深度学习方法求解高维问题的能力，对于维数很高的问题，传统方法很难应用，但深度学习方法可以给出一些相对可以接受的近似解。但是值得注意的是，随着维度的增高，需要求解得到的目标函数也越来越复杂，这就需要更宽和更深的神经网络来逼近，这同样对计算能力提出了一定的挑战。

7.2 变系数的特征值问题

求解椭圆方程特征值问题^{??}，其中参数选取为 $a(x) = 1$ ，求解区域为 $\Omega = [0, 1]^d$ 。这里我们把 $c(x)$ 选为分片常数，测试深度学习方法对变系数特征值问题的效果。

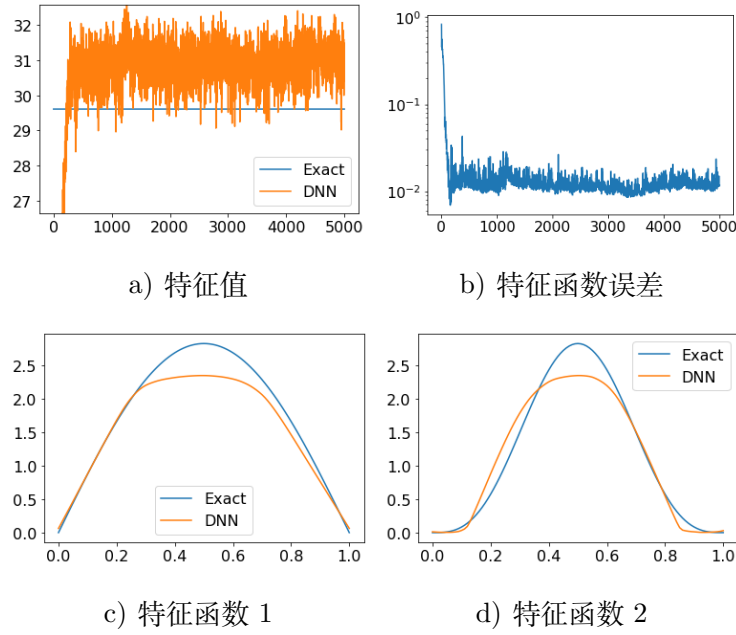


图 7.1 高维区域上的实验结果 ($d=3$)

7.2.1 一维数值实验

首先我们测试一维的情况，测试规模为 1-200-200-200-1 的全连接网络。每次训练区域内部生成 1000 个点，区域边界生成 200 个点。边界惩罚系数 β_2 为 1000，归一化惩罚系数 β_1 为 1000。其它参数相同。

计算结果如图??。图 (a) 中画出了势函数的选取，图 (b) 表示特征值的近似值和真实值随迭代步数变化的图像。图 (c) 中画出了精确解和深度学习方法给出的近似解的图像。可以看出，在一维的变系数问题上，虽然网络没有刻画出解精确的细节，但是峰值的位置已经可以确定了。

7.2.2 二维数值实验

在二维的情况下，我们测试规模为 1-800-800-800-1 的全连接网络。参数选取为每次训练区域内部生成 4000 个点，区域边界生成 1000 个点。边界惩罚系数 β_2 为 1000，归一化惩罚系数 β_1 为 10000。其它参数相同。

计算结果如图??。图 (a) 中画出了势函数的选取，图 (b) 表示特征值的近似值和真实值随迭代步数变化的图像。图 (c) 和 (d) 中画出了精确解和深度学习方法给出的近似解的图像。可以看出，二维情况下误差比一维的更大了，但还是在可以接受的范围内。

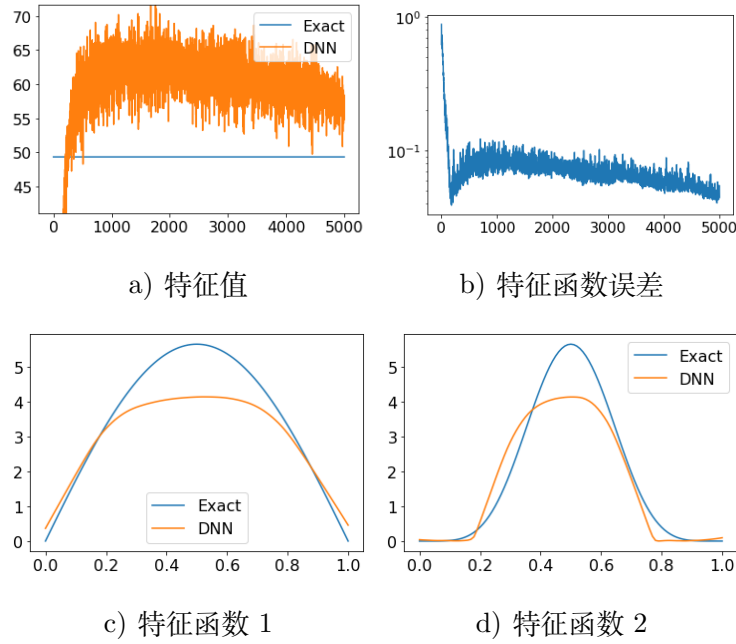


图 7.2 高维区域上的实验结果 ($d=5$)

7.3 具有几何奇异性的特征值问题

在椭圆方程特征值问题??中，参数选取为 $a(x) = 1$ ， $c(x) = 0$ ，我们选取不同的求解区域，验证几何奇异性对深度学习方法求解特征值问题的影响。

7.3.1 复杂单连通区域

这里的求解区域为一个多边形区域，多边形的顶点依次为 $(-1, 1) \rightarrow (-1, -1) \rightarrow (1, -1) \rightarrow (1, -0.6) \rightarrow (-0.3, -0.6) \rightarrow (-0.3, 0) \rightarrow (0, 0) \rightarrow (0, -0.5) \rightarrow (1, -0.5) \rightarrow (1, 1) \rightarrow (0.5, 1) \rightarrow (0.5, 0.3) \rightarrow (-1, 1)$ 。

如图??为通过 matlab 的 PDE 工具箱生成的网格和求解得到的特征函数参考解。问题的最小特征值参考解为 $\lambda = 19.2725$ 。

在深度学习方法中，我们测试规模为 1-800-800-800-1 的全连接网络。每次训练区域内部生成 8000 个点，区域边界生成 2500 个点。边界惩罚系数 β_2 为 1000，归一化惩罚系数 β_1 为 10000。计算结果如图??。图 (a) 画出了特征值的近似值和真实值随迭代步数变化的图像，图 (b) 画出了特征函数的近似解和真实解之间的误差。图 (c) 画出了迭代 5000 步后近似解的图像。可以看出，通过深度学习得到的特征函数近似解和参考解相差不大。在这样的区域上，深度学习方法效果很好。

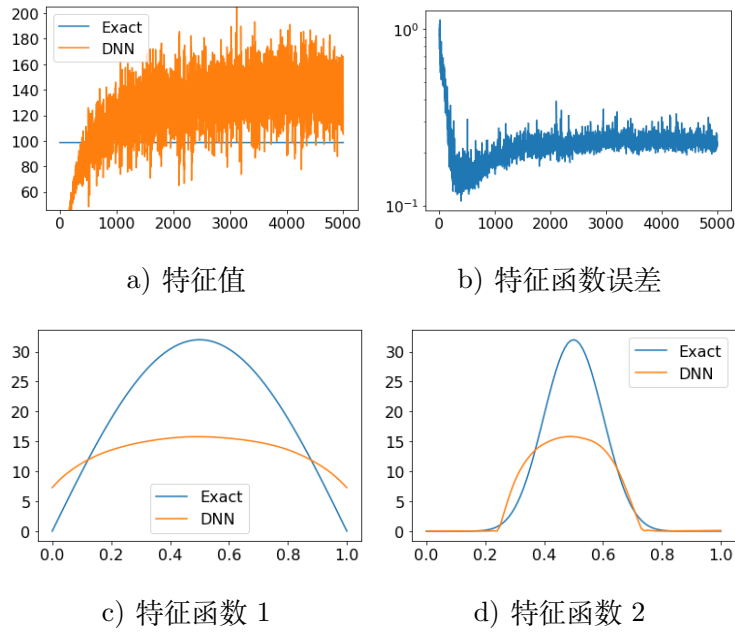


图 7.3 高维区域上的实验结果 ($d=10$)

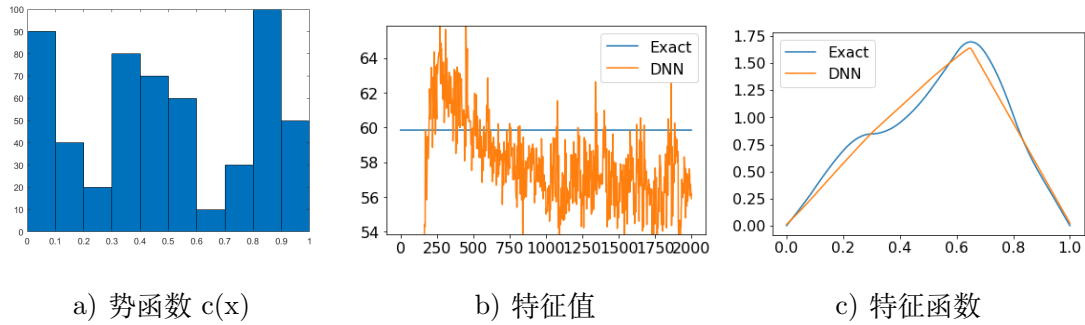


图 7.4 一维变系数方程中的实验结果

7.3.2 有孔区域

下面我们求解有孔区域上的特征值问题。求解区域为正方形 $[-1, 1]^2$ 挖去三个圆形的洞。洞的圆心和半径依次为 $c_1 = (-0.3, 0.3), r_1 = 0.5, c_2 = (0.6, 0), r_2 = 0.3, c_3 = (-0.1, -0.7), r_3 = 0.2$ 。

如图??为通过 matlab 的 PDE 工具箱生成的网格和求解得到的特征函数参考组。问题的最小特征值为 $\lambda = 28.8807$ 。

在同样的参数下用深度学习方法求解，实验结果如图??。可以看出，深度学习方法与 PDE 工具箱中给出的参考解有一定差距，但是特征函数的大致形状和峰的位置都已经被刻画出来了。

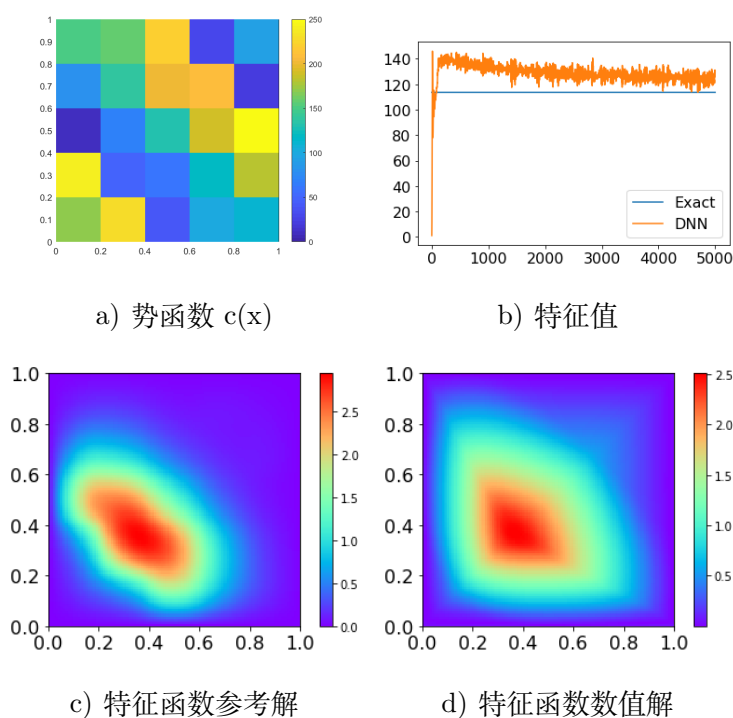


图 7.5 二维变系数方程中的实验结果

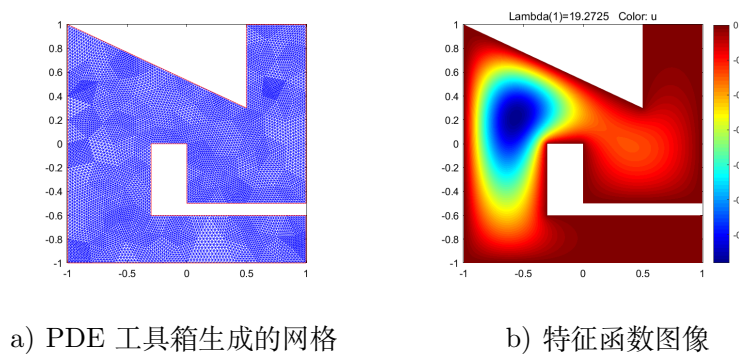


图 7.6 参考解图像

7.3.3 多孔区域

我们选取求解区域为正方形 $[-1, 1]^2$ 挖去 25 个圆形的洞。洞的圆心位于区域内的均匀网格上，半径依次在 0.05, 0.15 之间均匀划分。

如图??为通过 matlab 的 PDE 工具箱求解得到的特征函数和网格。最小特征值为 $\lambda = 68.9198$ 。

在同样的参数下用深度学习方法求解，实验结果如图??。对于这样复杂的区域，深度学习方法的效果明显变差，几乎完全没有得到正确的结果。这说明深度学习方法在处理一些复杂的特征值问题时，还有很大的改进空间。

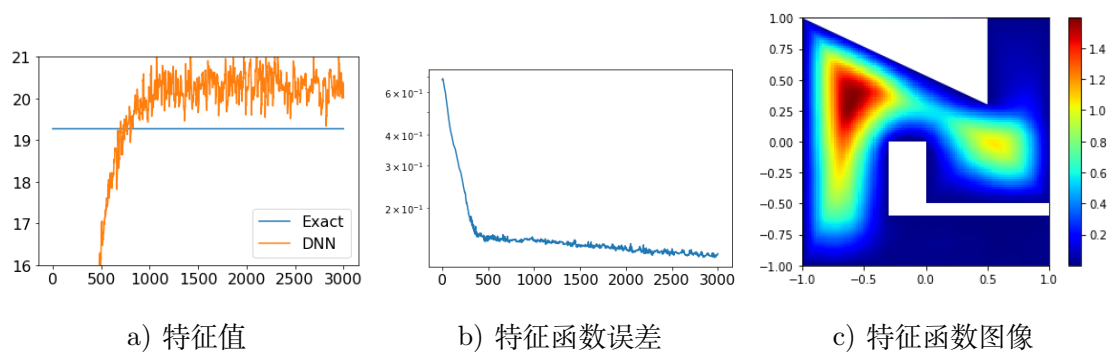


图 7.7 深度学习结果

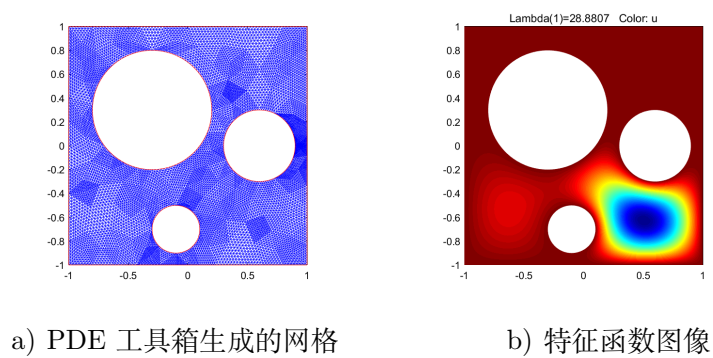


图 7.8 参考解图像

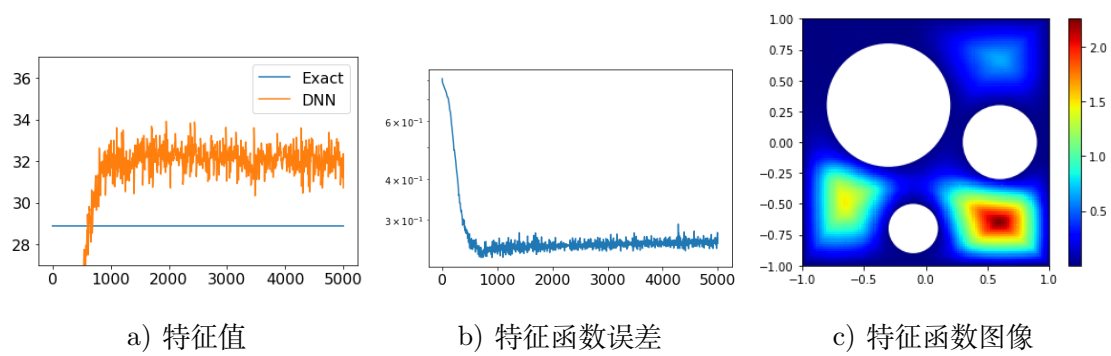


图 7.9 深度学习结果

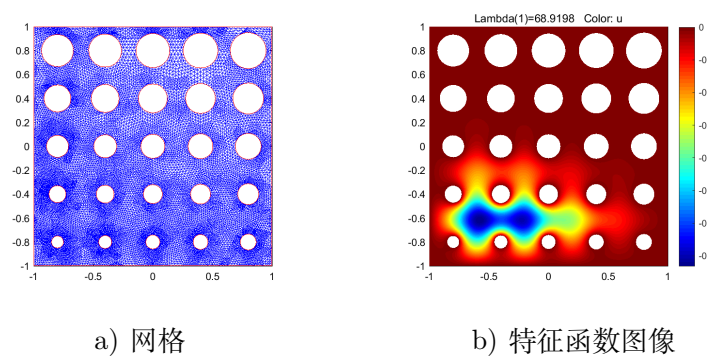
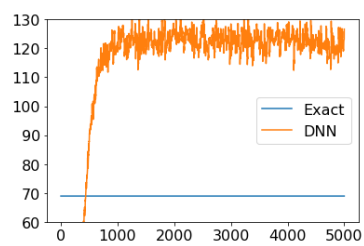
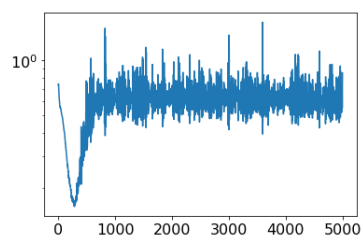


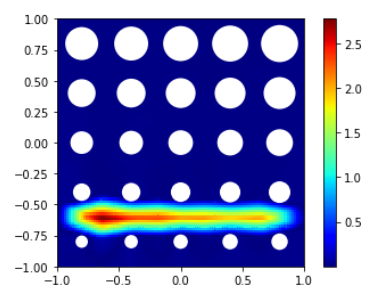
图 7.10 参考解图像



a) 特征值



b) 特征函数误差



c) 特征函数图像

图 7.11 深度学习结果

第八章 总结与展望

本文基于深度学习方法，从初始化的角度出发，介绍了一种新的多尺度网络结构。我们利用尺度变换的技术和紧支集的激活函数，构造了一种可以求解高频震荡椭圆方程的多尺度网络。本文的主要创新在于：在通常的全连接网络下，求解高频震荡的椭圆方程比较困难，而多尺度网络克服了这一缺点。大量的实验证明，多尺度网络是一种无网格、易于实现的椭圆方程求解方法。

相比于传统的有限元和有限差分方法，深度学习方法具有如下优势：首先，在复杂的区域上，由于需要生成网格和求解大型线性方程组，传统的有限元和有限差分法求解方法可能代价高昂。而深度学习方法只需要在边界和内部生成随机点，无需生成网格。此外，深度学习方法是本质非线性的，虽然本文中只讨论了线性方程的情形，这些方法都可以毫无难度地推广到非线性方程的情况。最后，深度学习方法采用了类似蒙特卡洛的抽样，因此可以克服维度灾难，更加快速地求解高维问题。

本文的末尾通过深度学习方法求解了椭圆特征值问题，指出深度学习方法在求解特征值问题中还有很大的改进空间。未来我们可以针对特征值问题设计对应的神经网络，希望可以提高求解效率。在进一步的工作中，我们还将探索在多尺度网络中，充分利用小波激活函数的思想，进一步提高多尺度网络的频率局部化和分离能力。此外，我们将尝试将多尺度网络应用于大规模计算工程问题，特别是与有限元法和有限差分法的比较。更进一步，我们将研究将多尺度网络应用于高维，更复杂的方程，如多体量子系统的薛定谔方程。在这些问题中，方程往往是非常高维的，用深度学习方法求解会为我们带来很大的便利。

致 谢

三年的硕士生涯即将画上句号。在这里，我要向这期间曾经帮助过我的人，表达最诚挚的谢意。

首先，我要感谢我的导师张智民老师，还有在学术研究中指导过我的李会元老师，许志钦老师，孙继广老师，蔡伟老师，贾晨老师（排名不分先后）。这些老师们无论是在学术方面还是生活方面，都给了我很多的指导和关心。而我也在这些老师身上学到的也不只是学术知识，更多的是严谨求实的态度和为人处世的道理。他们的言传身教使我终生受益。

同时，我也要感谢这三年来与我一起学习的各位同窗。在我面对困难灰心丧气的时候，他们总是给陪在我身边，给了我许许多多的鼓励。感谢他们陪我度过了一段永生难忘的岁月。

此外，我还要感谢我的家人和我的父母，他们是我在科研道路上的坚实后盾。感谢他们在硕士期间给了我充分的自由。我衷心希望自己能真正地成为家人眼中的骄傲。

最后，我要感谢我未来的女朋友。感谢她在我三年的硕士生涯中从来没有和我恋爱，使我可以专心地从事学术研究工作。

附录 9 攻读硕士学位期间发表的论文及其他成果

(一) 发表的学术论文

- [1] Liu Z, Cai W, Xu Z. Multi-scale Deep Neural Network (MscaleDNN) for Solving Poisson-Boltzmann Equation in Complex Domains[J]. 2020.

(二) 参加学术活动情况

- [1] 2019.06 深度学习在计算机视觉、求解微分方程和计算建模中的应用
- [2] 2019.07 第三届“偏微分方程数值方法与理论暑期学校”
- [3] 2020.07 2020 年度北京大学“应用数学专题讲习班”
- [4] 2020.07 机器学习和大数据在复杂性科学中的应用暑期培训班
- [5] 2020.12 【百旺科学论坛】认识我们的宇宙