神经网络谱隙正则化

Edric Tam edric.tam@duke.edu
杜克大学统计科学系，美国北卡
罗来纳州达勒姆，邮编 *27705*

David B. Dunson dunson@duke.edu
杜克大学统计科学系，美国北卡
罗来纳州达勒姆，邮编 *27705*

arXiv:2304.03096v1 [stat.ML] 6 Apr 2023

编辑： 不适用

摘要

我们提出了 Fiedler 正则化，这是一种利用谱信息/图结构信息对神经网络进行正则化的新方法。现有正则化方法通常侧重于以全局/均匀的方式惩罚权重，而忽略了神经网络的连通结构。我们提出使用神经网络底层图的费德勒值作为正则化工具，并通过谱图论为这一方法提供理论依据。我们展示了费德勒值的若干有用性质，说明它适合作为正则化工具。我们还提供了一种近似变分方法，以加快训练期间的计算。我们以结构加权 L_1 惩罚项的形式给出这一框架的另一种表述，从而将我们的方法与稀疏性诱导联系起来。通过拉德马赫复杂度分析，我们为 Fiedler 正则化提供了均匀分布的泛化误差界。我们在多个数据集上进行了实验，将 Fiedler 正则化与 Dropout、权重衰减等经典正则化方法进行比较。结果表明，Fiedler 正则化有效。本研究是 Tam 和 Dunson（2020）会议论文的期刊扩展版。

关键词： 谱间隙、神经网络、拉普拉斯矩阵、稀疏性、谱图论

1. 引言

神经网络（神经网络）是重要的工具，在计算机视觉、自然语言处理和强化学习等各种机器学习领域有着广泛的应用。在有大量带标签数据集的场景中，神经网络一直非常有效。实证和理论证据表明，近年来神经网络模型不断提升的容量，无论是在深度还是宽度方面，都是其建模灵活性和成功的重要因素。为了缓解这些灵活模型可能带来的过拟合问题，人们开发了各种各样的神经网络正则化技术。这些技术通常从全局/均匀分布的角度对网络进行正则化，例如权重衰减（KroghandHertz,1992）、对权重进行L_1/L_1 惩罚，以均匀分布的概率在各单元/层中按照伯努利方式丢弃节点/权重（Hinton et al., 2012; Srivastava et al., 2014; Wan et al., 2013），或提前停止训练。这些常用方法忽略了

神经网络的底层图结构，可为正则化提供有价值的连通性信息。

现有的前馈神经网络架构，例如多层感知器，经常采用全连接层，从而导致网络节点之间存在许多冗余路径。这些冗余连接可能通过协同适应现象导致过拟合，在该现象中，权重彼此依赖，导致不同隐藏单元之间的行为高度相关（Hinton 等，2012）。实证研究表明，在训练期间随机丢弃权重和节点，可以通过减少协同适应显著改善测试性能（Hinton 等，2012；Srivastava 等，2014；Wan 等，2013）。

这些方法的一种自然替代方案，是在正则化过程中考虑神经网络的图结构。在这项工作中，我们希望通过减少协同适应并惩罚多余连接，以一种遵循神经网络图结构/连通结构的方式对神经网络进行正则化。我们引入了 Fiedler 正则化，借鉴了谱图论的研究进展（Godsil 和 Royle，2013；Chung，1997；Spielman，2019）。连通图的费德勒值记为 λ_2，也称为代数连通度，是图的拉普拉斯矩阵的第二小特征值。 λ_2 的大小刻画了图的连通程度：如果费德勒值较小，那么该图就接近于非连通。通过在训练期间将费德勒值作为惩罚项加入损失函数，我们可以惩罚神经网络的连通性，并在考虑图的连通结构的同时减少协同适应。

我们还利用了费德勒值的若干有用特性。我们证明，费德勒值是关于神经网络权重大小的凹函数。这使我们能够将其与变量选择领域中折叠凹惩罚项的相关文献联系起来；这类惩罚项在统计学中被广泛采用，以减轻惩罚回归中的偏差。我们还证明，费德勒值关于网络权重的梯度有闭式表达，从而有助于深入了解 Fiedler 正则化的行为。

在实践中，对于较大的网络，为了加快计算，我们提出了一种变分方法，用图拉普拉斯矩阵的二次型替代原始的费德勒值惩罚项。与所谓的拉普拉斯测试向量结合使用时，这种拉普拉斯二次型可以为费德勒值提供一个紧确的上界。这种变分逼近可在训练期间显著加快计算。我们还给出了变分惩罚项的一种等价表述，即采用结构加权的 L_1 惩罚，其中权重取决于图拉普拉斯矩阵的（近似）第二特征向量。这种 L_1 表述使我们能够将 Fiedler 正则化与稀疏性诱导联系起来，这与统计学中的相关研究类似（Tibshirani，1996；Zou，2006年）。为这种方法提供理论保证，我们刻画了 Fiedler 正则化如何降低神经网络的 Rademacher 复杂度。由此，我们为该方法推导出均匀风险上界，为理解其泛化性能提供了依据。

此前已有研究利用输入数据的拉普拉斯结构对神经网络进行正则化（Kipfand Welling,2016；JiangandLin,2018；Zeng 等人，2019年）。近期也有研究从贝叶斯设定的角度理解神经网络上的正则化（Vladimirovaetal., 2018；Polson andRo˘ckov´a, 2018）。这些方法没有考虑底层神经网络的图形连通结构。这项工作的主要概念性贡献之一，是利用模型自身的图形/连通结构对其自身进行正则化。Af-

在本文会议版本发表之后，相关研究方向得到了进一步发展。（Carmichael，2021年）在统计学中的块稀疏估计场景下采用了类似的折叠凹拉普拉斯谱（FCLS）惩罚项。（Arbel等人，2021年）在贝叶斯块对角图模型场景下采用了相关的费德勒先验。在（Tuck等人，2019年）和（Carmichael，2021年）所研究的相关拉普拉斯正则化问题中，提出了极大化—最小化过程，以优化拉普拉斯正则化项。然而，在神经网络的背景下，极大化—最小化并不是一种计算上可行的优化方法，部分原因在于模型具有深层且高维的性质，同时损失函数具有高度非凸的性质。

这项工作的主要贡献包括：(1)据作者所知，这是首次通过神经网络自身的底层图结构/连通结构，将谱图论和费德勒值应用于神经网络的正则化。(2) 我们提出了实用且快速的费德勒正则化方法，并给出了强有力的理论保证和实验结果。本文是会议论文（Tam and Dunson,2020）的期刊扩展版。期刊版本在会议论文的基础上，通过拉德马赫复杂度分析提供了新的泛化误差保证，并通过折叠凹惩罚解释，与变量选择文献建立了新的联系。

2. 谱图论

2.1 设置与背景

令 W 表示前馈神经网络 f 的权重集合。我们将 f 的底层图结构记为 G。我们希望利用 G 中的结构信息，在训练过程中对 f 进行正则化。前馈神经网络不允许存在自环或循环连接，因此在此情形下，只需令 G 为有限、连通、简单、带权且无向的图。这样的图 G 可由三元组 $(V, E, |W|)$ 完全确定。这里，G 的顶点集 V 对应于神经网络 f 中的所有单元（包括输入层和输出层中的单元），而 G 的边集 E 对应于 f 中的所有边。出于正则化的目的，我们将 G 限定为具有非负权重 $|W|$，这些权重取神经网络 f 中相应权重 W 的绝对值。本文中，我们用 n 表示 G 中顶点的数量。$|W|$ 和 E 可共同表示为加权邻接矩阵 $n \times n$ $\mathrm{W}|$，其中，如果顶点 i 和 j 相连，$|\mathrm{W}|_{ij}$ 表示边 (i, j) 上的权重；否则为 0。图的度矩阵 D 是一个对角矩阵，其中 $\mathrm{D}_{ii} = \sum_{j=1}^{n} |\mathrm{W}|_{ij}$。拉普拉斯矩阵 L 是谱图论研究的核心对象，其定义为度矩阵与邻接矩阵之差，即 $\mathrm{L} = \mathrm{D} - |\mathrm{W}|$。在某些需要强调 L 对特定图 G 或权重 $|\mathrm{W}|$ 的依赖关系的情形下，我们采用记法 L_G 或 $\mathrm{L}_{|\mathbf{W}|}$。我们用 $[M]$ 表示正整数 M 对应的集合 $\{1, 2, \cdots, M\}$。本文中，所有特征值均为实数，因为所考虑的矩阵都是对称矩阵。按照惯例，所有特征向量均取为单位向量。我们按升序排列特征值，因此对于 $i < j$，有 $\lambda_i \leq \lambda_j$。若要强调 λ_i 是权重的函数，我们使用 $\lambda_i(|\mathrm{W}|)$。我们用 v_i 表示与 λ_i 对应的特征向量。我们用 $n(S)$ 表示给定集合S的基数。

2.2 图拉普拉斯矩阵

拉普拉斯矩阵编码了有关图结构的大量信息。拉普拉斯矩阵的特征值和特征向量被广泛用于谱聚类 Von Luxburg (2007)；Ng 等人 (2001) 和流形学习Belkin 和Niyogi(2003)等任务，正是因为它们能够捕获有价值的图连通性信息。我们大量使用的图拉普拉斯矩阵的一个特别有用的刻画，是下面定义的所谓拉普拉斯二次型（Batson 等人，2012；Spielman，2019年；Chung，1997）。

定义2.2.1（拉普拉斯二次型） 给定一个图 $G = (V, E, |W|)$ 及其拉普拉斯矩阵 $\mathbf{L}_G$，定义其拉普拉斯二次型 Q_G： $\mathbb{R}^{|V|} \to \mathbb{R}^+$ 为

$$Q_G(\boldsymbol{z}) := \boldsymbol{z}^T \mathbf{L}_G \boldsymbol{z} = \sum_{(i,j)\in E} |\mathbf{W}|_{ij} (\boldsymbol{z}(i) - \boldsymbol{z}(j))^2,$$

其中 $\boldsymbol{z}(k)$ 表示向量$\boldsymbol{z} \in \mathbb{R}^{|V|}$ 的第 k^{th} 个元素。

拉普拉斯二次型展示了如何从拉普拉斯矩阵中恢复图的边界信息。可以将$\mathbf{z}$ 视为一种映射，为每个顶点赋予一个值。如果我们将顶点子集 $S \subset V$ 的特征向量记为 $\mathbf{1}_S$，即 $\mathbf{1}_S(i) = 1$ 若 $i \in S$，否则 $\mathbf{1}_S(i) = 0$，并将其代入拉普拉斯二次型，就会得到 $\mathbf{1}_S^T \mathbf{L} G \mathbf{1}_S = \sum_{(i,j)\in E, i\in S, j\notin S} |\mathbf{W}|_{ij}$。该表达式刻画了图割 $(S, V - S)$ 的大小，即跨越 S 和 $V - S$ 之间边界的边的权重之和。因此，任何图割的大小都可以通过将相应的特征向量代入拉普拉斯二次型来求得。

2.3 边扩张与切格不等式

上文关于图割大小的讨论与我们对神经网络进行正则化的研究高度相关。减小神经网络中图割的大小意味着降低神经网络的连通性以及潜在的协同适应。一个相关但更为方便、能够捕捉图中边界大小这一概念的构造是图的边扩张（也称为切格常数或等周常数），可以非正式地将其理解为：顶点数不超过整个图一半的顶点子集所能达到的最小“表面积与体积之比”。

定义 2.3.1（图的边扩张） 边扩张 ϕ_G 图$G = (V, E, |W|)$ 定义为

$$\phi_G = \min_{S\subset V, n(S)\leq \frac{n(V)}{2}} \frac{\sum_{i\in S, j\notin S} |\mathbf{W}|_{ij}}{n(S)},$$

其中 $n(S)$ 表示 S中的顶点数量。

注意，分子中的项刻画了图割 $(S, V - S)$ 的大小，而分母则通过 S 中的顶点数对该表达式进行归一化。边扩张则定义为顶点集合 S that 所能达到的最小此类比值。

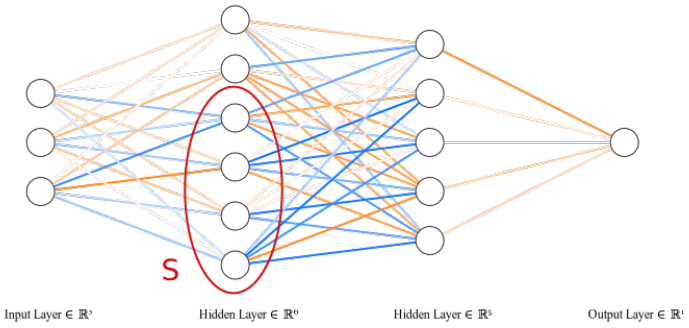

图1：神经网络上的图割图示（S， S^C）。部分顶点 S用红圈标出，图割（S， S^C）的大小为所有与红色边界相交的边的权重大小之和。权重按其符号着色（正值为橙色，负值为蓝色），不透明度则由其绝对值决定。

其基数至多为 V的一半。可以将图的边扩张理解为刻画图的连通性瓶颈的指标。它与图的稀疏程度以及图是否存在良好的平面嵌入高度相关（Hall，1970）。

我们希望通过正则化来控制神经网络底层图的边扩张。众所周知，由于组合结构的复杂性，直接计算边扩张是一个难题（Garey 等人，1974）。因此，我们通过费德勒值 λ_2 间接控制边扩张，该值是图的拉普拉斯矩阵$\mathbf{L}_G$的第二小特征值。 λ_2 与图的边扩张通过切格不等式相关联（Spielman，2019年；Chung，1997；Godsil 和 Royle，2013）。

命题 2.3.2（切格不等式）给定一个图 $G = (V, E, |W|)$，边扩张 ϕ_G 的上下界如下：

$$\sqrt{2d_{\max}(G)\lambda_2} \geq \phi_G \geq \frac{\lambda_2}{2},$$

其中， $d_{\max}(G)$ 是 G 中顶点的最大（加权）度数， λ_2是费德勒值，即 G的拉普拉斯矩阵的第二小特征值。

切格不等式最初源于对黎曼流形的研究，后来被推广到图。根据图的类型和所采用的归一化方式，切格不等式有许多不同形式。相关证明可见许多参考文献，包括 Spielman（2019年）；Chung（1997）；Godsil 和 Royle（2013）。这类切格不等式一般在渐近意义下至多相差常数因子，因而是紧的。

费德勒值也称为代数连通度，因为它包含了图连通性的大量信息。切格不等式使我们能够通过 G 费德勒值对边扩张进行精确控制。减小费德勒值，就能迫使图的边扩张变小，从而降低图的连通

性，并可能缓解神经网络中的协同适应。另一方面，较大的费德勒值必然意味着较大的边扩张。这意味着，在神经网络训练期间对费德勒值进行惩罚，是一种有望降低连通性、进而减少协同适应的正则化策略。

3. Fiedler 正则化

在本节中，我们介绍神经网络的Fiedler 正则化。我们在监督分类设置下对其进行刻画，以阐释主要思想，尽管这一通用框架也很容易扩展到回归和自动编码等其他设置。下面将介绍相关的设置和背景。

3.1 监督学习：分类设置

我们给定训练数据 $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ ，假设这些数据是从某个联合分布 $\mathcal{P}$ 中独立抽取的。这里，i 为数据点 N 编索引，$\mathbf{x} \in \mathbb{R}^d$ 表示 d维特征向量， y 表示结果；为简化说明，我们假设结果取值于二元空间 $\{0,1\}$ （即二分类）。通过简单调整，我们的许多结果都可以轻松推广到多分类和回归等场景。我们的目标是找到一个映射 f，通过 f (x) 预测 y ，使某个预先指定的损失 $\mathbb{E}_{\mathcal{P}}\mathcal{L}(f(\mathbf{x}) \ y)$ 的期望最小化。分类问题中，损失函数 $\mathcal{L}(\cdot,\cdot)$ 的典型选择包括交叉熵损失、合页损失等；而在回归问题中，均方误差很常见。训练过程中，经验逼近$\mathbb{E}_{\mathcal{P}_N}\mathcal{L}(f(\mathbf{x}) \ y) := \frac{1}{N}\sum_{i=1}^N \mathcal{L}(f(\mathbf{x_i}) \ y_i)$ 被用作期望损失的替代，其中 $\mathcal{P}_N$ 表示经验测度。假设测试集中的新观测值与训练数据一样，都是从同一分布 $\mathcal{P}$ 中独立抽样得到的。我们将 f 的估计量记作 $\hat{\mathbf{f}}$。当需要强调估计量对权重的依赖关系时，记作 $\hat{\mathbf{f}}_{\mathbf{W}}$。

3.2 神经网络设置

对于 Fiedler 正则化，我们目前只考虑前馈神经网络（潜在扩展的讨论见 {}）。对于前馈神经网络，f 被表示为一系列非线性函数 $\{\mathbf{g}^{(l)}\}_{l=1}^{\Lambda}$ 的复合，即 $\hat{\mathbf{f}} = \mathbf{g}^{(\Lambda)} \circ \mathbf{g}^{(\Lambda-1)} \circ \cdots \circ \mathbf{g}^{(1)}(\mathbf{x})$，其中 Λ 是网络中的层数。 l^{th} 层的输出形式为 $\mathbf{h}^{(l)} := \mathbf{g}^{(l)}(\mathbf{h}^{(l-1)}) := \sigma^{(l)}(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)})$，其中 $\sigma^{(l)}$、$\mathbf{W}^{(l)}$ 和 $\mathbf{b}^{(l)}$ 分别是神经网络 l^{th} 层的激活函数、权重矩阵和偏置。请注意，我们用 W 表示将整个神经网络编码为图后的加权邻接矩阵，而 $\mathbf{W}^{(l)}$ 表示编码在 l^{th} 层所应用的线性变换的矩阵。本质上，每个隐藏层首先对前一层的输出进行仿射变换，然后进行逐元素激活，该激活通常是非线性的。更多细节请参阅 Fan 等人（2019年）的精彩综述。

3.3 使用费德勒值进行惩罚

鉴于第 2 节中的动机，我们希望在训练过程中惩罚神经网络的图连通性。在 Fiedler 正则化方法中，我们添加 λ_2 作为惩罚项

将术语用于目标函数。

定义 3.2.1（Fiedler 正则化）在神经网络训练过程中，我们优化以下目标函数：

$$\min_{\mathbf{W}} \mathbb{E}_{\mathcal{P}_N} \mathcal{L}_{\mathbf{W}} + \delta \lambda_2(|\mathbf{W}|),$$

其中 $\lambda_2(|\mathbf{W}|)$ 是神经网络底层图的 *Fiedler* 值，$\mathbf{W}$ 是神经网络的权重矩阵，δ 是一个调节参数，Y 和 X 分别表示训练标签和训练特征。

ˆ我们注意到，实际的神经网络 $\mathbf{f_W}$ 的权重可能为负，但正则化项 $\lambda_2(|\mathbf{W}|)$ 仅取决于这些权重的大小，可以将其视为底层图 G 的边容量。

请注意，可以通过将可微函数 Q 应用于 λ_2 来推广惩罚项。在下面的讨论中，我们重点讨论不应用 Q 的情况，以便将分析聚焦于费德勒值。不过，纳入 Q 也很简单，而且通常可以保留一些理想性质，例如具有闭式梯度。我们还指出，不失一般性，可以将神经网络中单元的偏置视为具有常量输入的额外权重，因此在费德勒正则化中同时考虑偏置和权重也很简单。

就我们的研究目的而言，我们将神经网络的权重视为主要关注参数。在本研究中，激活函数、架构以及学习率或调节因子等超参数均视为预先设定。关于如何选择激活函数、架构和超参数，已有丰富的独立研究文献；这些内容不在本文讨论范围内。

值得注意的是，本节定义的 Fiedler 正则化旨在以理论化和理想化的方式阐释核心思想及其动机。在实践中，直接对费德勒值施加惩罚项的计算开销很大；关于具体实现，我们将采用后续章节所述的近似变分方法。

3.4 费德勒值的性质

考察费德勒值的一些性质，有助于理解它为何适合作为正则化工具。

首先，一个值得关注的问题是，在目标函数中使用费德勒值作为惩罚项是否会使优化过程更加复杂。费德勒值可视为拉普拉斯矩阵特征多项式的一个根；在高维情况下，该多项式没有闭式解，而且费德勒值对网络权重的依赖关系可能相当复杂。

为解决这一问题，以下命题表明，费德勒惩罚项是神经网络权重大小的凹函数。这说明，当我们将费德勒惩罚项加入深度学习目标函数（通常具有高度非凸性）时，并不会显著增加优化问题的难度。

命题 3.3.1（费德勒值的凹性）函数 $\lambda_2(|W|)$ 是关于神经网络权重 $|W|$大小的凹函数。

证明：由于费德勒值 λ_2 就是拉普拉斯矩阵的第二小特征值，并且我们知道拉普拉斯矩阵的第一个特征向量必为常数，因此我们可以考虑 λ_2 的 Rayleigh-Ritz 变分刻画如下：

$$\lambda_2(|\mathbf{W}|) = \inf_{||\mathbf{u}||=1, \mathbf{u}^T\mathbf{1}=0} \mathbf{u}^T\mathbf{L}\mathbf{u}$$

$$= \inf_{||\mathbf{u}||=1, \mathbf{u}^T\mathbf{1}=0} \sum_{(i,j)\in E} |\mathbf{W}|_{ij}(\mathbf{u}(i) - \mathbf{u}(j))^2$$

请注意，这是关于 $|\mathbf{W}|_{ij}$ 的线性函数的逐点下确界。由于线性函数是凹函数（也是凸函数），且逐点下确界保持凹性，因此 λ_2 是权重大小的凹函数。 ■

这与拉普拉斯矩阵特征值优化问题有关。关于更一般的讨论，请参阅 Boyd（2006年）和 S un 等人（2006年）。

命题 3.3.1 的一个直接推论是，相对于神经网络的权重 W ，费德勒值是一种折叠凹惩罚项。统计学中典型的稀疏性诱导方法采用 L_1 惩罚项，例如最小绝对收缩和选择算子（LASSO）（Tibshirani，1996年），但可能会产生较大的偏置。研究表明，采用折叠凹惩罚项的方法，例如平滑截断绝对偏差（SCAD）（Fan 和 Li，2001年）惩罚项以及极小极大凹惩罚（MCP）（Zhang，2010年），能够减少基于 L_1 的稀疏性诱导方法所产生的偏置。这类折叠凹惩罚函数的形状使较小的权重受到的惩罚相对大于较大的权重。在 Fiedler 正则化中，连通性较低的神经网络权重受到的惩罚更大。我们还可以通过分析费德勒值相对于神经网络权重的梯度，进一步理解这一现象。

3.5 梯度的闭式表达式

除最简单的情况外，优化损失函数 $\min_{\mathbf{W}} \mathcal{L}(Y, \mathbf{f}_{\mathbf{W}}(X))$ 都是一个非凸问题。针对这类目标函数，实际应用中有多种可扩展的随机算法可供选择。几乎所有广泛使用的方法，例如随机梯度下降（SGD）（Ruder，2016）、Adam（Kingma 和 Ba，2014）、Adagrad（Duchi 等，2011）、RMSProp（Graves，2013）等，都需要计算目标函数相对于参数的梯度。

我们给出了一个一般拉普拉斯矩阵特征值相对于拉普拉斯矩阵各元素的梯度闭式解析表达式。由此可直接推得费德勒值梯度的闭式解析表达式。

命题3.4.1（拉普拉斯矩阵特征值的梯度）假设拉普拉斯矩阵 $\mathbf{L}$ 的特征值不重复，则相对于$\mathbf{L}$的 $(ij)^{th}$ 元素$\mathbf{L}^{ij}$ ，最小特征值 λ_k的梯度 k^{th} 可以解析地表示为

$$\frac{d\lambda_k}{d\mathbf{L}_{ij}} = \boldsymbol{v}_k(i) \times \boldsymbol{v}_k(j),$$

其中 $\boldsymbol{v}_k(i)$ 表示拉普拉斯矩阵的 k^{th} 特征向量的第 i^{th} 个分量。我们采用的约定是，所讨论的所有特征向量均为单位向量。

证明：要计算$\frac{d\lambda_k}{d\mathbf{L}_{ij}}$，注意到由于拉普拉斯矩阵是对称的，所有特征值都为实数。根据假设，特征值互不重复。在这种情况下，已有一个闭式公式（参见（Petersen 等人，2008））：$\mathbf{v}_k^T(\partial\mathbf{L})\mathbf{v}_k = d\lambda_k$。具体到各个元素，可得 $\frac{d\lambda_k}{d\mathbf{L}_{ij}} = \mathbf{v}_k(i) \times \mathbf{v}_k(j)$。■

由上述内容，我们可以很容易地得到费德勒值相对于神经网络权重的梯度解析表达式。

推论3.4.2（费德勒值关于权重的梯度） 在命题*3.4.1*的相同假设下，作为一个直接的特殊情形，费德勒值 λ_2 可以表示为：

$$\frac{d\lambda_2}{d\mathbf{L}_{ij}} = \boldsymbol{v}_2(i) \times \boldsymbol{v}_2(j)$$

由于 $\mathbf{L}_{ij} = -|\mathbf{W}|_{ij}$ 对于 $i \neq j$，可得如下结果：

$$\frac{d\lambda_2}{d|\mathbf{W}|_{ij}} = -\boldsymbol{v}_2(i) \times \boldsymbol{v}_2(j)$$

在额外假设所考虑的权重非零的情况下，只需简单应用链式法则，即可在梯度中去掉绝对值映射。

特征值互不重复这一假设在神经网络中通常成立。神经网络中的权重通常初始化为从某些连续分布（如均匀分布或高斯分布）中独立抽取的样本。当图中存在较强的对称性时，拉普拉斯矩阵往往会出现重复特征值。在神经网络中，这类对称性通常会被打破，因为在初始化时或训练过程中，不同权重取相同非零值的概率微乎其微。

我们推导费德勒值梯度的上述解析形式，并非为了直接优化现实中的神经网络：在每次优化迭代（例如随机梯度下降）中重新计算拉普拉斯矩阵的第二特征向量，除了最小型的神经网络之外，对其他网络而言都将耗费过多计算资源。相反，我们利用梯度的解析形式，从理论上阐明费德勒正则化的作用。对于实际计算，我们将在下一节提出一种紧致的变分近似方法，该方法在实践中效果良好。

关于谱聚类的文献浩如烟海，其中有不少研究分析了拉普拉斯矩阵第二特征向量各项的含义。具体而言，如果顶点 i 和 j 之间的连接较弱（即它们属于不同的“簇”），那么 $\mathbf{v}_2(i) \times \mathbf{v}_2(j)$ 就会是一个非常小或为负的实数。这表明，对于连接较弱的顶点 i 和 j，导数 $\frac{d\lambda_2}{d|\mathbf{W}|_{ij}} = -\mathbf{v}_2(i) \times \mathbf{v}_2(j)$ 更大；这意味着，在 Fiedler 正则化下，连接较弱的顶点之间的边/权重受到的惩罚最大。

4. 用于加速计算的变分近似方法

我们给出了理论动机，并概述了将费德勒值用作神经网络正则化工具的方法。然而，计算特征值和特征向量的典型矩阵运算复杂度为 $O(n^3)$，其中 $n = |V|$。对于中型到大型网络而言，这在计算上可能难以承受。尽管存在理论上效率高得多的算法，可用于近似计算图相关矩阵的特征值/特征向量，但在训练的每次迭代中计算费德勒值，实践中仍可能代价高昂。为解决这一问题，我们提出一种近似的变分方法，将计算加速至 $O(|E|)$，其中 $|E|$ 是图中的边数。该方法可直接在 Tensorflow 和 PyTorch 等流行的深度学习软件包中实现。

我们作出以下观察。首先，为了对神经网络进行正则化，我们不需要拉普拉斯矩阵的精确费德勒值。良好的逼近就足够了。其次，在训练过程中，无需在每次迭代时都更新我们的近似 λ_2 。我们可以设置一个调度计划，定期更新我们的 λ_2 逼近，例如每100次迭代更新一次。这两点观察在实践中都能显著提速。算法1提供了这种近似变分方法的伪代码概要。

为了获得近似的费德勒值，我们使用一种涉及所谓测试向量的特殊拉普拉斯二次型。此外，我们还给出了一个扰动界，为定期更新费德勒值提供了依据。

4.1 特征值的瑞利商刻画

我们可以通过测试向量的概念对费德勒值进行紧密的上界估计（Spielman，2019年），这关键依赖于特征值的瑞利商刻画。

命题 4.1.1（测试向量界）对于任意与常向量 $\mathbf{1}$ 垂直的单位向量 $\boldsymbol{u}$ ，我们有：

$$\lambda_2 \leq \boldsymbol{u}^T \mathbf{L} \boldsymbol{u}$$

任何这样的单位向量都称为测试向量。当 $\boldsymbol{u} = \boldsymbol{v}_2$ 时，等号成立。

证明：非负权重图的拉普拉斯矩阵是对称的且半正定。因此，所有特征值都是实数且非负。特别地，拉普拉斯矩阵的最小特征值为 0，其对应的第一个特征向量是常向量。对于连通图，第二小特征值，也就是费德勒值，可根据 Courant-Fischer定理通过变分方式刻画为 $\lambda_2 = \min_{||\mathbf{u}||=1,\ \mathbf{u}^T\mathbf{1}=0} \mathbf{u}^T\mathbf{L}\mathbf{u}$。由此可得所需的上界。

■

上述描述表明，通过适当选择测试向量，我们可以有效地求出费德勒值的上界。这进而意味着，在训练过程中，我们不必使用精确的费德勒值作为惩罚项，而可以改用二次型上界作为惩罚项。换言之，我们将执行以下优化，

$$\min_{\mathbf{W}} \mathcal{L}(Y, \hat{\mathbf{f}}_{\mathbf{W}}(X)) + \delta \mathbf{u}^T \mathbf{L} \mathbf{u}$$

对于给定的$\mathbf{u}$，这能大幅加快惩罚项的计算速度，达到 $O(|E|)$，因为$\mathbf{u}^T\mathbf{L}\mathbf{u} = \sum_{(i,j)\in E} |\mathbf{W}|_{ij}(\mathbf{u}(i) - \mathbf{u}(j))^2$。现在的核心问题变成了如何选择合适的测试向量$\mathbf{u}$，使其接近$\mathbf{v}_2$。

我们建议使用精确的$\mathbf{v}_2$ 初始化训练，并且仅在训练期间定期重新计算/更新$\mathbf{v}_2$ 。对于两次精确$\mathbf{v}_2$ 更新之间的某次迭代，沿用上一次更新得到的$\mathbf{v}_2$，并将其作为当前迭代的测试向量。根据命题 4.1.1，这始终是 λ_2 的上界。

训练期间，神经网络的权重在每次迭代中都会更新。这等价于在每次迭代时向拉普拉斯矩阵$\mathbf{L}$加上一个对称矩阵$\mathbf{H}$；拉普拉斯矩阵同样是对称的。我们可以通过 Weyl 不等式（Horn and Johnson, 2012；Horn et al., 1998）来界定此类扰动对特征值的影响。

命题 4.1.2（Weyl 不等式）给定一个对称矩阵$\mathbf{L}$ 和一个对称扰动矩阵$\mathbf{H}$，二者的维度均为 $n \times n$，对于任意 $1 \leq i \leq n$，有：

$$|\lambda_i(\mathbf{L} + \mathbf{H}) - \lambda_i(\mathbf{L})| \leq ||\mathbf{H}||_{op}$$

其中，$||\cdot||_{op}$ 表示算子范数。

Weyl 不等式是矩阵分析中的经典结果，其证明可参见优秀的参考文献（Horn 和 Johnson，2012），它是线性性和 Courant-Fischer定理的直接推论。作为一个直接的特例，$|\lambda_2(\mathbf{L}+\mathbf{H}) - \lambda_2(\mathbf{L})| \leq ||\mathbf{H}||_{op}$。命题 4.1.2 告诉我们，只要扰动$\mathbf{H}$ 较小，扰动所导致的费德勒值变化也较小。事实上，这表明映射 $\mathbf{L} \to \lambda_2(\mathbf{L})$ 在对称矩阵空间上是 Lipschitz 连续的。在训练神经网络的背景下，$\mathbf{H}$ 表示网络在训练期间对权重所做的更新。这为仅定期更新 λ_2 以实现正则化提供了依据。这意味着，学习率越小，$\mathbf{H}$ 就越小，因此对 λ_2 的改变也会越小，并且测试向量的更新可以间隔更长。另一方面，对于较大的学习率，我们建议更频繁地更新测试向量。

5. 加权 $\mathbf{L}_1$ 形式与稀疏性

现在，我们进一步阐述变分 Fiedler 惩罚项的一种等价形式，即加权 $\mathbf{L}_1$ 惩罚项。我们注意到，变分 Fiedler 惩罚项中的拉普拉斯二次型可以写成 $\mathbf{u}^T\mathbf{L}\mathbf{u} = \sum_{(i,j)\in E} |\mathbf{W}|_{ij}(\mathbf{u}(i) - \mathbf{u}(j))^2$。由此得到变分目标函数：

$$\min_{\mathbf{W}} \mathcal{L}(Y, \hat{\mathbf{f}}_{\mathbf{W}}(X)) + \delta \sum_{(i,j)\in E} |\mathbf{W}|_{ij}(\mathbf{u}(i) - \mathbf{u}(j))^2$$

我们注意到，$|\mathbf{W}|_{ij}$ 和 $(\mathbf{u}(i) - \mathbf{u}(j))^2$ 均为非负数。这等价于对 $\mathbf{W}ij$ 进行 L_1 惩罚，其权重为 $(\mathbf{u}(i) - \mathbf{u}(j))^2$。

关于浅层模型中改进的 L_1 惩罚项，已有大量文献进行了研究（Zou，2006年；Candes 等人，2008年）。众所周知，在（加权） L_1约束下优化目标函数通常会得到稀疏解。因此，这将我们的 Fiedler 正则化方法与神经网络权重上的稀疏性诱导联系起来。从优化的角度来看，

算法1 使用随机梯度下降的变分费德勒正则化

输入： 训练数据 $\{\mathbf{x}_i,\ y_i\}_{i=\quad 1}^{N}$
超参数： 学习率 η、批次大小 m、惩罚参数 δ、更新周期 T
算法：
初始化神经网络的参数$\mathbf{W}$
计算神经网络的拉普拉斯矩阵$\mathbf{L}|\mathbf{W}|$
计算拉普拉斯矩阵$\mathbf{L}|\mathbf{W}|$的费德勒向量 $\mathbf{v}_2$
设置 $\mathbf{u} \leftarrow \mathbf{v}_{\quad 2}$
初始化计数器 $c = 0$
当 未满足停止准则 **执行**
  样本小批量 $\{\mathbf{x}^{(i)}, y^{(i)}\}_{i=\qquad 1}^{m}$ 来自训练集
  设置梯度 $\boldsymbol{\gamma} = \mathbf{0}$
  对于 $i = 1$ 至 m **执行**
    计算梯度 $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} + \nabla_{\mathbf{W}} \mathcal{L}(\hat{\ }\mathbf{f}_{\mathbf{W}}(\mathbf{x}^{(i)})\ y^{(i)})\ + \delta \nabla_{\mathbf{W}} \mathbf{u}^T \mathbf{L}|\mathbf{W}|\mathbf{u}$
  结束循环
  应用梯度更新 $\mathbf{W} \leftarrow \mathbf{W}\ - \eta\boldsymbol{\gamma}$
  更新拉普拉斯矩阵 $\mathbf{L}|\mathbf{W}|$
  更新计数器 $c \leftarrow c + 1$
  如果 c 模块 $T = 0$ **那么**
    重新计算 $\mathbf{v}_2$ 根据$\mathbf{L}|\mathbf{W}|$
    设置 $\mathbf{u} \leftarrow \mathbf{v}_{\quad 2}$
  结束 if
结束循环

回想一下，$\lambda_2(|\mathbf{W}|)$ 是关于 $|\mathbf{W}|$ 的凹函数，因此也是关于 $\mathbf{W}$ 的折叠凹函数。优化这种折叠凹函数的一种方法，是按照主化—最小化算法的思路，用一个主化代理函数取而代之。从我们的测试向量界很容易看出，加权 L_1 形式确实是 $\lambda_2(|\mathbf{W}|)$ 的一个主化函数，这从另一个角度支持了我们的变分近似。类似的思路见（Carmichael，2021）。

在 Fiedler 正则化中，权重 $|\mathbf{W}|_{ij}$ 会按因子 $(\mathbf{u}(i)\ -\mathbf{u}(j))^2$,进行缩放，其中 $\mathbf{u}$ 是逼近 $\mathbf{v}_2$的测试向量。从谱聚类文献（Hagen 和 Kahng，1992；Donath 和 Hoffman，1972）中，我们了解到，$\mathbf{v}_2$ 对于逼近图的最小导通率割非常有用。通常的启发式方法是，将$\mathbf{v}_2$ 的各个条目按升序排序，设定一个阈值 t，并将所有具有$\mathbf{v}_2 > t$ 的顶点 i归入一个簇，其余顶点归入另一个簇。如果阈值 t 选择得当，这种聚类将很好地逼近图的最小导通率割。因此，$\mathbf{v}_2(i)$ 与$\mathbf{v}_2(j)$相距越远，（1）节点 i 和j 之间的边（如果存在）对于图的连通结构而言就越有可能“不那么重要”；（2）节点 i 和 j 属于不同簇的可能性就越大。

这也与图的谱绘制有关（Spielman，2019年；Hall，1970），在这种情况下可以证明，如果 λ_2

很大，就不存在图的“良好”平面嵌入/绘制。从这个意义上说，Fiedler 正则化是在尊重神经网络连通结构的同时，促使神经网络变得“更平面”。

因此，采用 Fiedler 正则化时，当这条边对图的连通结构“没那么重要”时，对 $|\mathbf{W}|_{ij}$ 的惩罚最强。理论上，这会使权重较低且连接不同簇中相距较远顶点的边更加稀疏。因此，Fiedler 正则化以尊重神经网络连通结构的方式使其稀疏化。

为此，另一种保证是拉普拉斯特征值对于一系列图的排序性质（Godsil 和 Royle，2013）。

命题 5.1（拉普拉斯特征值排序）给定具有非负权重的图 G ，若移除一条边（a, b），得到 $G \setminus$（a, b），则有 λ_2（$G \setminus$（a, b）$\leq \lambda_2$（G）。

证明：该证明是 Godsil 和 Royle（2013）的简单推广。取$\mathbf{q}_2$ 和$\mathbf{r}_2$ 分别为 G 和 $G \setminus$（a, b）的第二特征向量。根据拉普拉斯二次型对特征值的刻画，我们有

$$\lambda_2(G) = \mathbf{q}_2^T \mathbf{L}_G \mathbf{q}_2 = \sum_{(c,d)\in E} |\mathbf{W}|_{cd} (\mathbf{q}_2(c) - \mathbf{q}_2(d))^2$$

$$\geq [\sum_{(c,d)\in E} |\mathbf{W}|_{cd} (\mathbf{q}_2(c) - \mathbf{q}_2(d))^2] - |\mathbf{W}|_{ab} (\mathbf{q}_2(a) - \mathbf{q}_2(b))^2$$

$$= \mathbf{q}_2^T \mathbf{L}_{G\setminus(a,b)} \mathbf{q}_2 \geq \mathbf{r}_2^T \mathbf{L}_{G\setminus(a,b)} \mathbf{r}_2 = \lambda_2(G \setminus (a,b))$$

其中，第一个不等式由所考虑权重的非负性得出，第二个不等式则由特征值的 Rayleigh-Ritz 变分刻画得出。 ■

因此，通过使边变得稀疏并降低边的权重，我们实际上是在降低神经网络的费德勒值。上述排序性质是更一般的拉普拉斯矩阵特征值交错性质的一个特例，其中 $\lambda_2(G \setminus (a,b))$ 也存在相应的下界。有关一般性处理，请参阅 Godsil 和 Royle（2013）。

我们指出，由于 Fiedler 正则化鼓励稀疏性，在训练过程中神经网络可能会变得非连通，从而使 λ_2 变为 0。实践中可以轻松避免这一问题：在训练过程中，从拉普拉斯矩阵中移除一个非连通分量（例如较小的那个），也就是删除与被移除分量中的顶点相对应的行和列。

6. Error Generalization Bounds

我们前面给出的保证侧重于 Fiedler 正则化的逼近和变分方面。本节将给出这种方法在泛化误差方面的一些理论保证。具体而言，我们运用统计学习理论中的 Rademacher 复杂度概念（Bartlett 和 Mendelson，2002），为 Fiedler 正则化提供有限样本下的均匀泛化误差上界。

这有助于了解 Fiedler 正则化如何降低所学习神经网络的表达能力，从而减少过拟合并提升泛化性能。我们聚焦于监督分类情形，并分析拉普拉斯二次型／加权 L_1惩罚形式；当使用的测试向量恰好为 $\mathbf{v}_2$ 时，该形式包含精确的 Fielder 值惩罚这一特例。据我们所知，这是针对加权-L_1 惩罚神经网络的首个 Rademacher 复杂度分析。这里的结果或许也具有独立的研究价值。

6.1 Rademacher 复杂度与误差界

刻画一类函数/分类器的泛化性能，一个关键方面是理解其表达能力。经验 Rademacher 复杂度这一概念为量化函数类的表达能力提供了一种有效方法。

定义 6.1（Rademacher 复杂度）给定一组包含 N 个点的样本 $\mathbf{z}_1, \cdots, \mathbf{z}_N$，这些点是从某个集合 $Z \subseteq \mathbb{R}^d$ 上的概率分布 $\mathcal{P}$ 中独立同分布抽取的；另给定一个函数类 $\mathcal{H}$，其中的函数将 Z 映射到 $\mathbb{R}$。我们将 $\mathcal{H}$ 的经验 *Rademacher*ˆ复杂度记为 $R_N(\mathcal{H})$，定义如下：

$$\hat{R}_N(\mathcal{H}) := \mathbb{E}_{\mathcal{R}}\left[\sup_{h\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} \Omega_i h(\mathbf{z}_i)\Big|\{\mathbf{z}_i\}_{i=1}^{N}\right]$$

其中，Ω_i 是独立同分布的随机变量，服从 *Rademacher* 分布 $\mathcal{R}$ （即在 $\{-1, 1\}$ 上的均匀分布），期望是对 *Rademacher* 随机变量 Ω_i 取的。

ˆ$R_N(\mathcal{H})$ 是一个随机量，因为它以样本 $\mathbf{z}_1, \cdots, \mathbf{z}_N$ 为条件。对 $R_N(\mathcal{H})$ 关于乘积测度 $\mathcal{P}^N$ 取期望，称为 $\mathcal{H}$ 的 *Rademacher* 复杂度，记为 $R_{N,\mathcal{P}}(\mathcal{H})$。换言之，

$$R_{N,\mathcal{P}}(\mathcal{H}) := E_{\mathcal{P}^N}[\hat{R}_N(\mathcal{H})]$$

直观来说，函数类 $\mathcal{H}$ 的 Rademacher 复杂度刻画了 $\mathcal{H}$ 的“表达能力”，具体而言，就是 $\mathcal{H}$ 匹配随机符号模式（即 Rademacher 随机变量）的能力。Rademacher 复杂度越大，函数类的表达能力就越强；这通常意味着更容易过拟合，泛化性能也更差。请注意，Rademacher 复杂度定义中的期望取决于 $\mathcal{P}$，而在许多问题中，$\mathcal{P}$ 可能复杂、维度很高，和/或未知。经验 Rademacher 复杂度的概念通过对样本进行条件化，绕过了这一问题。

Rademacher 复杂度之所以是一个有用的概念，部分原因在于它自然出现在经验过程理论和统计学习理论中许多值得关注的界中。对于满足 $Z = (X, Y)$ 的监督分类情形，可以选择 $\mathcal{H} := \{(x, y) \mapsto \mathcal{L}(f(x), y) : f \in \mathcal{F}\}$，其中 $\mathcal{L}$ 是一个固定的损失函数，而 $\mathcal{F}$ 是所考虑的分类器类。在分类问题中，函数 $\mathcal{L}$ 通常选为替代损失，例如合页损失或逻辑损失。这些常用的损失函数

是 Lipsc 是 Lipschitz 连续的，且 Lipschitz 常数为 1。下面的定理因此很有用 .

定理 6.2（基于Rademacher 复杂度的泛化界） 设 $\mathcal{H} := \{(x, y) \mapsto \mathcal{L}(f(x), y) : f \in \mathcal{F}\}$，其中 $\mathcal{L}$ 是一个 *1-Lipschitz* 连续损失函数。则对于任意 $\delta \in (0, 1)$，在乘积测度 $\mathcal{P}^N$下以至少 $1 - \delta$ 的概率，以下不等式成立：

$$\sup_{h \in \mathcal{H}} \left[\frac{1}{N} \sum_{i=1}^{N} h(\mathbf{z}_i) - \mathbb{E}_{\mathcal{P}}(h(\mathbf{z})) \right] \leq 2\hat{R}_N(\mathcal{H}) + O\left(\sqrt{\frac{\log(1/\delta)}{N}} \right),$$

其中每个$\mathbf{z}_i$ 均独立地服从 $\mathcal{P}$ 。

上述定理的证明可见于许多资料和教材中，包括 Bartlett 和 Mendelson (2002)；Shalev-Shwartz 和 Ben-David (2014)；Bousquet 等人 (2003)。上述概率界是均匀且有限样本的。将其应用于学习问题时，量 $\mathbb{E}_{\mathcal{P}}(h(\mathbf{z}))$ 对应于“真实”误差，而量 $\frac{1}{N} \sum_{i=1}^{N} h(\mathbf{z}_i)$ 对应于在训练集上得到的误差。因此，定理 6.2 提供了一种以均匀、有限样本且与分布无关的方式控制泛化误差的方法，特别适用于分析现代机器学习模型。这里使用渐近记号 $O(\cdot)$ 来省略很小的常数因子。上述引文中的资料介绍了显示常数因子的该不等式的显式形式。

6.2 神经网络的 Rademacher 复杂度界

有了上述定理，我们接下来将重点放在对采用 Fiedler 正则化惩罚的前馈神经网络的经验 Rademacher 复杂度 $R_N(\mathcal{H})$ 进行上界估计。这进而揭示了 Fiedler 正则化如何影响神经网络的泛化误差界。我们的策略是先对所考虑的前馈神经网络分类器类别的 Rademacher 复杂度 $R_N(\mathcal{F})$ 求上界。然后，我们利用关于 $R_N(\mathcal{F})$ 的结果，通过应用损失函数，推导出 $R_N(\mathcal{H})$ 的上界。

为了直观理解我们的方法，回顾一下我们所处的以下设定：固定神经网络的架构，将 σ 设为激活函数，将 Λ 设为层数。然后，我们所考虑的前馈神经网络将具有以下形式：

$$f(\mathbf{x}) := \mathbf{W}^{(\Lambda)} \sigma(\mathbf{W}^{(\Lambda-1)}(\sigma(\cdots \sigma(\mathbf{W}^{(1)}\mathbf{x}) \cdots))) \tag{1}$$

其中，激活函数 σ 按分量应用。请注意，在不失一般性的情况下，偏置被吸收到输入$\mathbf{x}$ 中。另请注意，$\mathbf{W}^{(\Lambda)}$ 表示将节点从 $l - 1^{\text{th}}$ 层映射到l^{th} 层的线性变换所对应的矩阵。

为了证明神经网络的 Rademacher 复杂度界，我们首先列出几个有用的引理。鉴于方程 1 的复合性质，下面的第一个引理有助于控制 Lipschitz 复合下函数类的 Rademacher 复杂度。

引理 6.3（Talagrand 收缩引理）（Ledoux 和 Talagrand，2013）设 $\phi_1, \cdots \phi_N$ 是 γ-*Lipschitz* 函数，从 $\mathbb{R} \to \mathbb{R}$ 映射而来，其中 $\gamma > 0$。那么

$$\hat{R}_N((\phi_1, \cdots, \phi_N) \circ \mathcal{H}) := \mathbb{E}_{\mathcal{R}}\left[\sup_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \Omega_i \phi_i(h(\mathbf{z}_i)) \Big| \{\mathbf{z}_i\}_{i=1}^{N}\right]$$

$$\leq \gamma \mathbb{E}_{\mathcal{R}}\left[\sup_{h \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \Omega_i h(\mathbf{z}_i) \Big| \{\mathbf{z}_i\}_{i=1}^{N}\right] = \gamma \hat{R}_N(\mathcal{H})$$

其中，$\circ$ 表示函数复合。

换言之，我们可以用 $\mathcal{H}$ 的（经验）Rademacher 复杂度，来控制它与 γ-Lipschitz 函数逐元素复合后的（经验）Rademacher 复杂度。Talagrand 收缩引理提供了一种实用方法，可通过分析来控制具有组合表示的函数/分类器的表达能力。

前馈神经网络天然具有组合式描述，原始输入数据经过仿射变换与激活函数的交替处理。实践中使用的几乎所有激活函数都是 Lipschitz 连续的，常见选择如 tanh、ReLU 等均为 1-Lipschitz。常用算子，尤其是计算中常用的卷积算子，也都是 Lipschitz 连续的。因此，Talagrand 收缩引理使我们能够研究广泛的前馈神经网络类别。

线性变换也是神经网络的重要组成部分。以下引理给出了权重的 L_1 范数有界时，线性变换的 Rademacher 复杂度的控制方法。

引理 6.4（线性函数类的 Rademacher 复杂度）给定一个独立同分布样本$\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$和一个函数类 $\mathcal{F}_{linear}^B := \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle \mid \mathbf{w} \in \mathbb{R}^d, ||\mathbf{w}||_1 \leq B\}$, 假设对于数据集中的所有$\mathbf{x}_i$，都有$\mathbf{x}_i||_\infty \leq C$

$$\hat{R}_N(\mathcal{F}_{linear}^B) \leq BC\sqrt{\frac{2\log(2d)}{N}}$$

这是一个标准结果，其证明基于 Massart (2000) 的一个引理。完整的证明可以在许多参考资料中找到，例如Shalev-Shwartz 和 Ben-David (2014) 的第 26.2 节。

权重 $\mathbf{w}$ 的有界 L_1 范数假设适用于我们研究神经网络稀疏性的情境，而数据集中的数据点具有均匀有界 L_∞ 范数的假设通常在实践中都会得到满足。

对上述引理稍作修改，即可将其应用于适用于 Fiedler 正则化情形的加权L_1 场景：

推论 6.5（加权 L_1约束下线性函数类的Rademacher 复杂度）给定一个独立同分布样本 $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$ 和一个函数类 $\mathcal{F}_1 := \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle \mid \mathbf{w} \in \mathbb{R}^d, \sum_{i=1}^d \mathbf{c}_i |\mathbf{w}_i| \leq B\}$，其中 $\mathbf{c} \geq \mathbf{0}$ 是一个固定的 d 维权重向量，

假设 $||\mathbf{x}_i||_\infty \leq C$ 对于数据集中的所有 $\mathbf{x}_i$ ， 我们有

$$\hat{R}_N(\mathcal{F}_1) \leq \frac{B}{\min(\mathbf{c})} C \sqrt{\frac{2\log(2d)}{N}}$$

其中，我们使用 最小值(c) 表示向量 c 中的最小元素。

证明*:* 注意，min(c) $\sum_{i=1}^{d} |\mathbf{w}_i| \leq \sum_{i=1}^{d} \mathbf{c}_i |\mathbf{w}_i| \leq B$。这意味着 $||\mathbf{w}||_1 \leq B \min(\mathbf{c})$。注意，这意味着 $\mathcal{F}_1 \subseteq \mathcal{F}_{linear}^{\min(\mathbf{c})}$。调用引理 6.4，我们得到所需的上界 $\hat{R}_N(\mathcal{F}_1) \leq \hat{R}_N(\mathcal{F}_{\min(\mathbf{c})linear}) \leq B \min(\mathbf{c}) C \sqrt{\frac{2\log(2d)}{N}}$。 ■

回顾优化中的拉格朗日对偶形式，在目标函数中添加 L_1 惩罚项对应于在目标函数的定义域上施加 L_1 约束。类似地，在 Fiedler 正则化的情况下，由于在目标函数中添加了加权 L_1 惩罚项，因此等价于对权重施加加权 L_1 约束。

现在，我们已经具备了逐层控制神经网络的 Rademacher 复杂度的工具。设 $d_0 = d$ 为输入数据的维度， d_l 为神经网络中第 l层的宽度。设 $\mathbf{c}^0, \mathbf{c}^1, \cdots, \mathbf{c}^{l-1}$ 为维度分别为$d_0, d_1, \cdots, d_{l-1}$的非负权重向量。设 $B_0, B_1, \cdots, B_{l-1}$ 为非负常数。令基类为 $\mathcal{F}_1 := \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle \mid \mathbf{w} \in \mathbb{R}^d, \sum_{i=1}^{d} \mathbf{c}_i^0 |\mathbf{w}_i| \leq B_0\}$。递归地定义 $\mathcal{F}_l$，其中 l从 2 取到 Λ:

$$\mathcal{F}_l := \{\mathbf{x} \mapsto \sum_{i=1}^{d_{l-1}} \mathbf{w}_i \sigma(f_i(\mathbf{x})) \mid f_i \in \mathcal{F}_{l-1}, \sum_{i=1}^{d_{l-1}} \mathbf{c}_i^{l-1} |\mathbf{w}_i| \leq B_{l-1}\}$$

目前，我们将 $\mathbf{c}^0, \mathbf{c}^1, \cdots, \mathbf{c}^{l-1}$ 视为一般的非负权重向量，以推导出一个通用结果；之后，我们再选择它们，使其具体对应于 Fiedler 正则化的情形。

现在我们可以陈述主要结果:

定理 6.6（加权惩罚项下神经网络的 Rademacher 复杂度界）L_1 ）令 $\mathcal{F}_\Lambda$ 表示具有固定前馈架构且逐元素应用 γ-*Lipschitz* 激活函数的 Λ 层神经网络类，其中第 l^{th} 层的权重矩阵记为 $\mathbf{W}^{(l)}$。假设输入特征 $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$ 满足 $||\mathbf{x}_i||_\infty \leq C$，对所有 $\mathbf{x}_i$均成立。还假设对于每一层 l，$\mathbf{W}^{(l)}$ 每一行的 $\mathbf{c}^{l-1}$-加权 L_1 范数均不超过 B_{l-1}。那么有

$$\hat{R}_N(\mathcal{F}_\Lambda) \leq (2\gamma)^{\Lambda-1} \left(\prod_{l=1}^{\Lambda} \frac{B_{l-1}}{\min(\mathbf{c}^{l-1})}\right) C \sqrt{\frac{2\log(2d)}{N}}$$

证明： 不难看出，权重矩阵 $\mathbf{W}^{(l)}$ 的各行恰好对应于在 $\mathcal{F}_l$ 中应用的线性变换的权重 $\mathbf{w}$ 。这意味着，假设 $\mathbf{W}^{(l)}$ 的每一行的$\mathbf{c}^{l-1}$-加权 L_1 范数至多为 B_{l-1} ，恰好对应于

常数 在 $\mathcal{F}_l$ 中的约束。因此，我们可以通过函数类 $\mathcal{F}$ 继续进行分析 l.

我们首先注意到

$$\hat{R}_N(\mathcal{F}_\Lambda) \leq 2\gamma \frac{B_{\Lambda-1}}{\min(\mathbf{c}^{\Lambda-1})} \hat{R}_N(\mathcal{F}_{\Lambda-1}) \tag{2}$$

为说明这一点，利用定义

$$\hat{R}_N(\mathcal{F}_\Lambda) = \mathbb{E}_{\mathcal{R}}\left[\sup_{f \in \mathcal{F}_\Lambda} \frac{1}{N} \sum_{i=1}^{N} \Omega_i f(\mathbf{z}_i) \middle| \{\mathbf{z}_i\}_{i=1}^N \right]$$

$$= \mathbb{E}_{\mathcal{R}}\left[\sup_{f_j \in \mathcal{F}_{\Lambda-1}, \sum_{j=1}^{d_{\Lambda-1}} \mathbf{c}_j^{\Lambda-1} \mathbf{w}_j \leq B_{\Lambda-1}} \frac{1}{N} \sum_{i=1}^{N} \Omega_i \sum_{j=1}^{d_{\Lambda-1}} \mathbf{w}_j \sigma(f_j(\mathbf{z}_i)) \middle| \{\mathbf{z}_i\}_{i=1}^N \right]$$

$$= \mathbb{E}_{\mathcal{R}}\left[\sup_{f_j \in \mathcal{F}_{\Lambda-1}, \sum_{j=1}^{d_{\Lambda-1}} \mathbf{c}_j^{\Lambda-1} \mathbf{w}_j \leq B_{\Lambda-1}} \sum_{j=1}^{d_{\Lambda-1}} \mathbf{w}_j \frac{1}{N} \sum_{i=1}^{N} \Omega_i \sigma(f_j(\mathbf{z}_i)) \middle| \{\mathbf{z}_i\}_{i=1}^N \right]$$

应用柯西-施瓦茨不等式，得到：

$$\leq \mathbb{E}_{\mathcal{R}}\left[\sup_{f_j \in \mathcal{F}_{\Lambda-1}, \sum_{j=1}^{d_{\Lambda-1}} \mathbf{c}_j^{\Lambda-1} \mathbf{w}_j \leq B_{\Lambda-1}} ||\mathbf{w}||_1 \max_j \left| \frac{1}{N} \sum_{i=1}^{N} \Omega_i \sigma(f_j(\mathbf{z}_i)) \right| \middle| \{\mathbf{z}_i\}_{i=1}^N \right]$$

利用约束条件 $\sum_{j=1}^{d_{\Lambda-1}} \mathbf{c}_j^{\Lambda-1} \mathbf{w}_j \leq B_{\Lambda-1}$，这意味着 $||\mathbf{w}||_1 \leq$ $\frac{B_{\Lambda-1}}{\mathbf{c}_{\Lambda-1}^{\Lambda-1} \min(\mathbf{c})}$，我们有：

$$\leq \frac{B_{\Lambda-1}}{\min(\mathbf{c}^{\Lambda-1})} \mathbb{E}_{\mathcal{R}}\left[\sup_{f \in \mathcal{F}_{\Lambda-1}} \left| \frac{1}{N} \sum_{i=1}^{N} \Omega_i \sigma(f(\mathbf{z}_i)) \right| \middle| \{\mathbf{z}_i\}_{i=1}^N \right]$$

$$\leq 2\frac{B_{\Lambda-1}}{\min(\mathbf{c}^{\Lambda-1})} \mathbb{E}_{\mathcal{R}}\left[\sup_{f \in \mathcal{F}_{\Lambda-1}} \frac{1}{N} \sum_{i=1}^{N} \Omega_i \sigma(f(\mathbf{z}_i)) \middle| \{\mathbf{z}_i\}_{i=1}^N \right]$$

应用引理 6.3（Talagrand 收缩引理），得到：

$$\leq 2\gamma \frac{B_{\Lambda-1}}{\min(\mathbf{c}^{\Lambda-1})} \mathbb{E}_{\mathcal{R}}\left[\sup_{f \in \mathcal{F}_{\Lambda-1}} \frac{1}{N} \sum_{i=1}^{N} \Omega_i f(\mathbf{z}_i) \middle| \{\mathbf{z}_i\}_{i=1}^N \right]$$

$$= 2\gamma \frac{B_{\Lambda-1}}{\min(\mathbf{c}^{\Lambda-1})} \hat{R}_N(\mathcal{F}_{\Lambda-1})$$

这证明了断言 2。

递归应用不等式 2 共 $\Lambda - 1$ 次，我们得到

$$\hat{R}_N(\mathcal{F}_\Lambda) \leq 2^{\Lambda-1} \gamma^{\Lambda-1} \left(\prod_{l=2}^{\Lambda} \frac{B_{l-1}}{\min(\mathbf{c}^{l-1})} \right) \hat{R}_N(\mathcal{F}_1)$$

应用推论 6.5，得到：

$$\hat{R}_N(\mathcal{F}_\Lambda) \leq 2^{\Lambda-1}\gamma^{\Lambda-1}\left(\prod_{l=2}^{\Lambda}\frac{B_{l-1}}{\min(\mathbf{c}^{l-1})}\right)\frac{B_0}{\min(\mathbf{c}^0)}C\sqrt{\frac{2\log(2d)}{N}}$$

$$= (2\gamma)^{\Lambda-1}\prod_{l=1}^{\Lambda}\frac{B_{l-1}}{\min(\mathbf{c}^{l-1})}C\sqrt{\frac{2\log(2d)}{N}}$$

这表明了所需的结果。 ■

上述结果使我们能够按层控制神经网络在加权 L_1 惩罚项下的 Rademacher 复杂度。在 Fiedler 正则化中，所施加的惩罚项为 $\sum_{(a,b)\in E}(\mathrm{u}(a)-\mathrm{u}(b))^2|\mathrm{W}_{ab}|$。回想一下，不带上标的 W 表示整个神经网络底层图的加权邻接矩阵，而 u 是一个固定的测试向量（对于精确结果，可以取第二个拉普拉斯矩阵特征向量 $\mathbf{v}_2$ ）。通过定义一个 $n\times n$ 矩阵 U，使得 $\mathrm{U}_{ab}=(\mathrm{u}(a)-\mathrm{u}(b))^2$，我们可以将其改写为矩阵形式。因此，惩罚项的等价表达式为$\sum_{(a,b)\in E}\mathrm{U}_{ab}|\mathrm{W}_{ab}|$。

这个惩罚项具有可分性，因此我们可以按层拆分。将层 l 中的节点记为 V_l ，将连接 V_{l-1} 中的节点与 V_l 中的节点的边记为 E_l。我们得到惩罚项$\sum_{l=1}^{\Lambda}\sum_{(a,b)\in E_l}\mathbf{U}_{ab}|\mathbf{W}_{ab}^{(l)}|$的等价表达式。现在，这具有加权的 L_1 形式，类似于函数类 $\mathcal{F}_l$中使用的逐层加权 L_1 约束。因此，现在可以相应地选择上文定理 6.6 中使用的权重向量$\mathbf{c}^0,\mathbf{c}^1,\cdots,\mathbf{c}^{\Lambda-1}$。

回顾一下

$$\mathcal{F}_l := \{\mathbf{x}\mapsto\sum_{i=1}^{d_{l-1}}\mathbf{w}_i\sigma(f_i(\mathbf{x}))\mid f_i\in\mathcal{F}_{l-1},\sum_{i=1}^{d_{l-1}}\mathbf{c}_i^{l-1}|\mathbf{w}_i|\leq B_{l-1}\}$$

权重$\mathbf{w}$ 将 $\mathcal{F}_l$ 中的节点映射到 V_{l-1} 中第 l 层的一个节点，并对应于矩阵$\mathbf{W}^{(l)}$的一行。因此，选取$\mathbf{c}^{l-1}$ 的一种方法是选取其 b^{th} 项，其中 b 对 V_{l-1} 中的节点进行索引，将其作为$\mathbf{U}$的 b^{th} 列中的最大项。由于网络具有前馈结构（没有任何跳跃连接），这相当于在连接 V_{l-1} 中的节点 b 与 V_l 中所有节点的边上，求 $\mathbf{U}$ 的最大值，即 $\mathbf{c}^{l-1}(b)=\max_{\{a|(a,b)\in E_l\}}\mathbf{U}_{a,b}$。沿 b^{th} 列选取最大项，可确保得到的函数类 $\mathcal{F}_\Lambda$ 足够大，能够包含 Fiedler 正则化神经网络。

于是我们得到以下推论：

推论 6.7（Fiedler 正则化神经网络的 Rademacher 复杂度界）假设定理 *6.6* 的条件成立。此外，固定一个用于 *Fiedler* 正则化的测试向量$\mathbf{u}$ ，它指定一个矩阵$\mathbf{U}$。对于从1 到 Λ 的每个 l ，按照$\mathbf{c}^{l-1}(b) = \max_{\{a|(a,b)\in E_l\}}\mathbf{U}_{a,b}$选择权重向量$\mathbf{c}^{l-1}$ ，其中 b 为 V_{l-1} 的索引。由此得到的函数类的经验 *Rademacher* 复杂度上界为

如下：

$$\hat{R}_N(\mathcal{F}_\Lambda) \leq (2\gamma)^{\Lambda-1}(\prod_{l=1}^{\Lambda} \frac{B_l}{\min_{\{b\in V_{l-1}\}}(\max_{\{a|(a,b)\in E_l\}} \mathbf{U}_{a,b})})C\sqrt{\frac{2\log(2d)}{N}}$$

证明： 只需根据$\mathbf{c}^{l-1}$ 设置向量$\mathbf{c}^{l-1}(b) = \max^{\{a|(a,b)\in E_l\}}\mathbf{U}_{a,b}$ 并应用定理 6.6。
■

近期有一些研究结果（Golowich 等人，2018）通过更精细的 Rademacher 分析，消除了 $(2\gamma)^{\Lambda-1}$ 项对深度的指数依赖。不过，基于这些结果获得更精细的界仍是一个未来方向，而且不会影响对 Fielder 正则化的解读和推导；Fielder 正则化控制的是权重矩阵的范数。

推论 6.5 中的上述界揭示了 Fiedler 正则化的作用机制。$\mathbf{U}_{ab} = (\mathbf{u}(a) - \mathbf{u}(b))^2$ 的值越大，就意味着边 (a, b) 对连通性“越不重要”。因此，$\max^{\{a|(a,b)\in E_l\}}\mathbf{U}_{a,b}$ 操作可以理解为寻找与节点 b 相关联的最不重要的边。接下来的 $\min^{\{b\in V_{l-1}\}}$操作则是在层 $l-1$ 中，找到其最不重要的边具有最佳连通性的节点。可以将其视为一种以极小极大／最坏情况方式刻画层内连通性瓶颈的方法。该值越高，惩罚越强，因此 Rademacher 复杂度越低。

我们可以结合定理 6.2 和推论 6.7，得到 Fiedler 正则化神经网络的泛化误差界。

推论 6.8（Fiedler 正则化神经网络的泛化误差界）令 $\mathcal{H} := \{(x, y) \mapsto \mathcal{L}(f(x), y) : f \in \mathcal{F}_\Lambda\}$，其中 $\mathcal{L}$ 是一个 *1-Lipschitz* 连续损失函数。那么，对于任意 $\delta \in (0, 1)$，在乘积测度$\mathcal{P}^N$下，以至少 $1-\delta$ 的概率，以下不等式成立：

$$\sup_{h\in\mathcal{H}}\left[\frac{1}{N}\sum_{i=1}^{N} h(\mathbf{z}_i) - \mathbb{E}_{\mathcal{P}}(h(\mathbf{z}))\right]$$

$$\leq 2(2\gamma)^{\Lambda-1}(\prod_{l=1}^{\Lambda} \frac{B_l}{\min_{\{b\in V_{l-1}\}}(\max_{\{a|(a,b)\in E_l\}} \mathbf{U}_{a,b})})C\sqrt{\frac{2\log(2d)}{N}} + O\left(\sqrt{\frac{\log(1/\delta)}{N}}\right),$$

其中，每个$\mathbf{z}_i$ 都独立地服从 $\mathcal{P}$ 。

^^证明： 应用 Talagrand 收缩引理，得到 $R_N(\mathcal{H}) \leq 1 \times R_N(\mathcal{F}_\Lambda)$，然后应用定理 6.2。
■

7. 监督分类：实验与结果

深度／多层前馈神经网络适用于许多分类问题，涵盖热门的图像识别任务，以及科学与生物医学问题，例如疾病分类

。我们考察了 Fiedler 正则化在标准基准图像分类数据集 MNIST 和 CIFAR10 上的表现。我们还在 UCI 机器学习知识库中的 TCGA RNA-Seq PANCAN 肿瘤分类数据集（Weinstein 等，2013）上测试了 Fiedler 正则化。我们比较了几种常见神经网络正则化方法在这些数据集上的表现，包括 Dropout、L_1 正则化和权重衰减。将此类实验扩展到其他分类任务也很直接。

以下实验的目的并不是使用最深的神经网络、最新的架构或经过最优化的超参数。目的也不是展示神经网络相对于随机森林或逻辑回归等其他分类方法的优越性。相反，我们试图将 Fiedler正则化的有效性与其他神经网络正则化技术进行比较，以作为概念验证。对于更复杂、更通用的网络架构的扩展，将在讨论部分进行探讨。

在所有实验中，我们考虑采用 ReLU 激活函数和全连接层的 5 层前馈神经网络。所有实验均使用PyTorch 1.4 和 Python3.6。我们采用动量为 0.9 的随机梯度下降进行优化，学习率设为0.001。为了选择Dropout的丢弃概率，以及L_1, Fiedler 正则化和权重衰减的正则化超参数，我们在一个小型验证数据集上进行了非常粗略的网格搜索。所有层的Dropout概率均选为 0.5，L_1, Fiedler正则化和权重衰减的正则化超参数分别为0.001、0.01 和 0.01。实验中的所有模型均采用交叉熵损失进行训练。每个实验运行 5 次，并报告性能的中位数和标准差。所有实验均在配备 Intel Core i7 处理器的 Unix机器上运行。实验所使用的代码可在第一作者的Github 仓库中找到（https://github.com/edrictam/FiedlerRegularization）。

7.1 MNIST

数据集和设置。MNIST 是一个标准的手写识别数据集，由 60,000 28×28 张单个手写数字的训练图像和 $10,000$张测试图像组成。我们将神经网络的隐藏层设置为 500 个单元宽。我们使用了 100 的批次大小，并对网络进行了 10 个训练轮次的训练。

结果。MNIST 的结果显示在表1 中。对于 MNIST 数据集，我们通过所有方法都获得了非常高的准确率，其中 Fiedler 正则化表现突出，其次是权重衰减和Dropout。使用前馈神经网络在 MNIST 数据集上获得的高准确率与先前研究的结果一致（Srivastava 等，2014）。Fiedler 正则化相较于其竞争方法取得了提升。

7.2 CIFAR10

数据集和设置。CIFAR10是一个基准目标识别数据集，由$32 \times 32 \times 3$个经过下采样的RGB彩色图像组成，涵盖10 个不同的目标类别。该数据集包含50,000张训练图像和10,000 张测试图像。我们将神经网络的隐藏层宽度设为500 个单元。我们使用了 100 的批次大小，并对网络进行了 10 个训练轮次。

结果。CIFAR10 的结果见表 2。尽管这是一个难度更高的

表 1：不同正则化方案下 MNIST 的分类准确率（单位：百分比）

Regularization	Training	Testing
L_1	90.32 ± 0.17	90.25± 0.35
Weight Decay	95.12± 0.06	94.98± 0.07
Dropout	94.52± 0.07	94.5± 0.18
Fiedler	96.54± 0.08	96.1± 0.12

表 2：各种正则化方案下 CIFAR10 的分类准确率（单位：百分比）

Regularization	Training	Testing
L_1	28.52± 0.9	28.88± 1.11
Weight Decay	52.93 ± 0.17	50.55± 0.39
Dropout	46.61± 0.35	44.63± 0.39
Fiedler	57.99 ± 0.13	52.26± 0.27

图像分类任务，但各正则化方法的表现排序与 MNIST 上相似。Fiedler 正则化表现最佳，其次是 Dropout、权重衰减和 L_1 正则化。

7.3 TCGA 癌症分类

数据集和设置。TCGA泛癌肿瘤分类数据集由800 名受试者的RNA 测序数据以及癌症分类结果组成。输入特征是20531维的基因表达水平向量，而输出是肿瘤分类标签（所考虑的肿瘤类型有 5种）。我们使用了 600 名受试者进行训练，200名进行测试。由于该分类任务具有高度过参数化的特性，我们将隐藏层的宽度设得更窄，为 50个单元。我们使用了批次大小 10，并对网络进行了 5 个训练轮次。

结果 TCGA 肿瘤分类实验的结果如表 3 所示。Fiedler 正则化和 L_1 的表现同样出色，随后是权重衰减。值得注意的是，Dropout 的准确率相对较低，仅略高于随机猜测水平。由于该数据集相对较小（测试集仅有 200 个数据点，训练集有 600 个），准确率的标准差较大。L_1 和 Fiedler 正则化能够显式地促进稀疏性，表现最佳。

表 3：TCGA 在各种正则化方案下的分类准确率（单位：百分比）

Regularization	Training	Testing
L_1	91.5 ± 13.73	94.53± 12.46
Weight Decay	57± 22.42	60.2± 20.94
Dropout	25.33± 5.9	23.88± 7.36
Fiedler	93.33± 20.31	90.55± 20.73

7.4 结果分析

我们注意到，MNIST 和 CIFAR10 的训练样本数都多于其特征维度。MNIST 和 CIFAR10 的结果基本一致，并证实了 Fiedler 正则化的有效性。在这种设置下，Dropout 和权重衰减等其他正则化方法也表现不俗。

另一方面，在 TCGA 数据集中，输入特征的维度（20531）远高于训练样本数（600），L_1 和 Fiedler 正则化能够明确地诱导稀疏性，其表现显著优于 Dropout 和权重衰减。检查 TCGA 数据集后发现，输入特征中的许多基因表达水平为 0（表明这些基因未表达），这很可能意味着网络中的许多权重，尤其是输入层的权重，并非必需。因此，在生物医学应用中常见的这类“大 p、小 n”场景下，能够明确诱导稀疏性的正则化方法表现更好，也就不足为奇了。

我们指出，Fiedler 正则化的实际运行速度较快，通常与大多数常用正则化方案（如 L_1 和 Dropout）大致相当，但稍慢一些。通过采用某些实现层面的优化来加快软件运行，Fiedler 正则化的运行时间很可能得到改善。如果采用更精细的网格搜索，或贝叶斯优化等更先进的调参方法，性能也很可能会有所提升。

8. 讨论

上述实验揭示了几个值得关注的现象。L_1 正则化在 MNIST 和 CIFAR10 上表现较差，这与 Fiedler 正则化（一种加权 L_1 惩罚项）的优异表现形成了鲜明对比。人们普遍认为，L_1 正则化在深度学习模型中的实证表现并不理想。其背后的确切原因尚不清楚。先前的研究采用了群组套索形式来对深度神经网络进行正则化，并取得了良好效果（Scardapane 等，2017）。这些结果表明，通过加权或其他类似方案对 L_1 进行改造，往往能显著改善实证表现。

在我们的实验中，我们跟踪了神经网络在训练过程中的代数连通度。一般来说，在没有任何正则化的情况下，神经网络在训练期间往往会变得更加连通，即其费德勒值会增加。在费德勒 正则化

情况下，连通性会受到惩罚，因此在训练过程中以非常渐进的方式降低。有趣的是，在L_1情况下，神经网络的代数连通度可能在训练过程中迅速降低，通常会导致网络在训练过程的早期就断开。这可能与 L_1 正则化对所有权重进行均匀分布的惩罚有关。因此，很难为L_1选择正则化超参数：如果它太小，稀疏性诱导可能发生得过慢，我们就会正则化不足；如果它太大，我们就有过度惩罚某些权重并降低模型准确率的风险。因此，诸如 Fiedler 惩罚这样的加权方案的一个优势在于，它能够在训练过程中自适应地惩罚不同的权重。

尽管我们仅考虑了相对简单的前馈全连接神经网络架构，但将其扩展到更复杂的结构是很直接的。许多卷积神经网络在初始卷积层之后包含全连接层，因此可以轻松地将 Fiedler 正则化扩展到这种情况。在 ResNet 中，网络包含跳跃连接；我们所利用的谱图性质仍然成立，因此可以直接应用 Fiedler 正则化。在非分类设置下，自动编码器以瓶颈的形式呈现出非常自然的图结构；研究基于图连通性的惩罚项（例如 Fiedler 正则化）在这类设置中的应用，是一个有待探索的方向。本文采用的谱图论框架通常适用于任意无向图。另一个有待探索的方向，是为有向图的正则化建立适当的谱图论，这将有助于循环神经网络的训练。

尽管理论上 Fiedler 正则化会使神经网络呈现稀疏连接，但在实践中，使用 Fiedler 正则化通常需要更高的惩罚项取值或更长的训练时间，才能实现稀疏性。这部分可能是所选优化方法导致的。众所周知，一般而言，随机梯度下降无法在采用 L_1 惩罚项的模型中高效地诱导稀疏性，而某些截断梯度方法（Langford 等，2009）在这种情况下可能会更加有效。

我们指出，虽然 Fiedler 正则化强调基于图结构/连通结构进行正则化，但 Dropout、L2 等全局惩罚方法仍可能有用。可以结合这两种正则化方法，实现同步正则化。

许多重要的统计模型和机器学习模型都可以视为浅层神经网络的特例，例如因子分析、线性回归等。一个自然的进一步研究方向是将 Fiedler 正则化应用于这些模型。

上文讨论的图割/边扩张视角与等周性的几何概念之间存在密切联系。上文讨论的拉普拉斯矩阵可视为拉普拉斯-贝尔特拉米算子的离散化类比。一个有待探索的方向是研究 Fiedler 正则化方法的连续松弛。

关于泛化误差界分析，存在一条独立的研究路线（Neyshabur等，2017年），该路线利用神经网络的权重矩阵的谱范数，为泛化误差提供间隔界。他们关注于利用各层的权重矩阵的谱范数，而我们则关注于通过神经网络底层图的拉普拉斯矩阵的谱来利用全局结构。

最后，我们采用了谱图论的一种形式，它考虑了未归一化（组合）拉普拉斯矩阵 $\mathbf{L} = \mathbf{D} - |\mathbf{W}|$ 以及图的边扩张。我们也可以针对归一化拉普拉斯矩阵建立类似的正则化理论

$\mathbf{L}' = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}|\mathbf{W}|\mathbf{D}^{-\frac{1}{2}}$以及图的导通率，并适当考虑神经网络的总体规模。我们所考察的组合拉普拉斯矩阵与比率切割的概念相关，而归一化拉普拉斯矩阵则与归一化切割的概念相关。理论上，这两种概念都可以用于降低神经网络的连通性和协同适应。

致谢

本研究部分由美国国立卫生研究院拨款 R01MH118927 以及 SAMSI 的资助支持。

参考文献

Julyan Arbel、Mario Beraha 和 Daria Bystrova。基于费德勒先验的贝叶斯分块对角图模型。发表于*SFdS-52 Journ´ees de Statistique de la Soci´et´e Francaise deStatistique*，第 1–6 页，2021年。Peter L Bartlett 和 Shahar Mendelson。Rademacher 和高斯复杂度：风险界与结构性结果。*Journal of Machine Learning Research*，3(Nov):463–482，2002年。Joshua Batson、Daniel A Spielman 和 Nikhil Srivastava。两倍 Ramanujan 稀疏化器。*SIAM Journal on Computing*，41(6):1704–1721，2012年。Mikhail Belkin 和 Partha Niyogi。用于降维和数据表示的拉普拉斯特征映射。*Neural computation*，15(6):1373–1396，2003年。Olivier Bousquet、St´ephane Boucheron 和 G´abor Lugosi。统计学习理论导论。发表于*Summer School on Machine Learning*，第 169–207 页。Springer，2003年。Stephen Boyd。图拉普拉斯特征值的凸优化。发表于*Proceedings of theInternational Congress of Mathematicians*，第 3 卷，第 1311–1319 页，2006年。Emmanuel J Candes、Michael B Wakin 和 Stephen P Boyd。通过重新加权 $\ell 1$ 最小化增强稀疏性。*Journal of Fourier analysis and applications*，14(5-6):877–905，2008年。Iain Carmichael。折叠凹拉普拉斯谱惩罚项通过强预言机性质学习分块对角稀疏模式。*arXiv preprint arXiv:2107.03494*，2021年。Fan RK Chung。谱图论。第 92 号。American Mathematical Soc.，1997年。William E Donath 和 Alan J Hoffman。基于连接矩阵特征向量的图划分与计算机逻辑算法。*IBM Technical Disclosure Bulletin*，15(3):938–944，1972年。John Duchi、Elad Hazan 和 Yoram Singer。用于在线学习和随机优化的自适应次梯度方法。*Journal of machine learning research*，12(Jul):2121–2159，2011年。Jianqing Fan 和 Runze Li。通过非凹惩罚似然进行变量选择及其预言机性质。*Journal of the American statistical Association*，96(456):1348–1360，2001年。Jianqing Fan、Cong Ma 和 Yiqiao Zhong。深度学习的选择性综述。*arXivpreprint arXiv:1904.05526*，2019年。Michael R Garey、David S Johnson 和 Larry Stockmeyer。一些简化的 NP 完全问题。发表于*Proceedings of the sixth annual ACM symposium on Theory of computing*，第 47–63 页，1974年。

Chris Godsil和Gordon FRoyle。代数图论，第 207 卷。SpringerScience & Business Media，2013 年。Noah Golowich、AlexanderRakhlin 和 Ohad Shamir。神经网络的规模无关样本复杂度。载于 学习理论会议，第 297–299页。PMLR，2018 年。AlexGraves。使用循环神经网络生成序列。*arXiv*预印本*arXiv:1308.0850*，2013 年。LarsHagen 和Andrew B Kahng。用于比率切割划分和聚类的新谱方法。*IEEE* 集成电路与系统计算机辅助设计汇刊，11(9)：1074–1085，1992 年。KennethM Hall。一个 r 维二次布局算法。管理科学，17(3)：219–229，1970 年。Geoffrey E Hinton、Nitish Srivastava、Alex Krizhevsky、Ilya Sutskever 和RuslanRSalakhutdinov。通过防止特征检测器的协同适应来改进神经网络。*arXiv*预印本 *arXiv:1207.0580*，2012 年。Roger A Horn 和 Charles R Johnson。矩阵分析。剑桥大学出版社，2012 年。RogerAHorn、Noah H Rhee 和 SoWasin。特征值不等式与等式。线性代数及其应用，270(1-3)：29–44，1998年。BoJiang和Doudou Lin。用于半监督学习的图拉普拉斯矩阵正则化图卷积网络。*arXiv* 预印本*arXiv:1809.09839*，2018 年。Diederik PKingma 和 Jimmy Ba。Adam：一种随机优化方法。*arXiv*预印本 *arXiv:1412.6980*，2014 年。Thomas N Kipf 和 Max Welling。使用图卷积网络进行半监督分类。*arXiv* 预印本*arXiv:1609.02907*，2016 年。Anders Krogh 和 John A Hertz。简单的权重衰减可以改善泛化能力。在神经信息处理系统进展中，第950–957 页，1992 年。JohnLangford、Lihong Li 和 Tong Zhang。通过截断梯度进行稀疏在线学习。在 神经信息处理系统进展中，第 905–912 页，2009 年。Michel Ledoux 和 Michel Talagrand。巴拿赫空间中的概率：等周不等式与过程。SpringerScience & BusinessMedia，2013 年。Pascal Massart。集中不等式在统计学中的一些应用。在 图卢兹数学科学年鉴 ´：数学 ´学中，第 9 卷，第 245–303 页，2000 年。Behnam Neyshabur、Srinadh Bhojanapalli 和Nathan Srebro。神经网络谱归一化间隔界的PAC-贝叶斯方法。*arXiv* 预印本*arXiv:1707.09564*，2017 年。

吴恩达、Michael Jordan 和 Yair Weiss。关于谱聚类：分析与一种算法。神经信息处理系统进展，14，2001年。Kaare Brandt Petersen、Michael Syskind Pedersen 等。《矩阵手册》。丹麦技术大学，7(15):510，2008年。Nicholas G Polson 和 Veronika Ro˘ckov´a。稀疏深度学习的后验集中性。载于神经信息处理系统进展，第 930–941 页，2018年。Sebastian Ruder。梯度下降优化算法综述。*arXiv* 预印本 *arXiv:1609.04747*，2016年。Simone Scardapane、Danilo Comminiello、Amir Hussain 和 Aurelio Uncini。深度神经网络的组稀疏正则化。神经计算，241:81–89，2017年。Shai Shalev-Shwartz 和 Shai Ben-David。理解机器学习：从理论到算法。剑桥大学出版社，2014。Daniel Spielman。谱图论与代数图论。2019年。Nitish Srivastava、Geoffrey Hinton、Alex Krizhevsky、Ilya Sutskever 和 Ruslan Salakhut- dinov。Dropout：一种防止神经网络过拟合的简单方法。机器学习研究期刊，15(1):1929–1958，2014。Jun Sun、Stephen Boyd、Lin Xiao 和 Persi Diaconis。图上的最快混合马尔可夫过程及其与最大方差展开问题的联系。*SIAM* 综述，48(4):681–699，2006年。Edric Tam 和 David Dunson。Fiedler 正则化：使用图稀疏性学习神经网络。*arXiv* 预印本 *arXiv:2003.00992*，2020年。Robert Tibshirani。通过套索进行回归收缩与变量选择。皇家统计学会学报：*B* 辑（方法学），58(1):267–288，1996年。Jonathan Tuck、David Hallac 和 Stephen Boyd。用于拉普拉斯矩阵正则化问题的分布式主化-最小化。*IEEE/CAA* 自动化学报，6(1):45–52，2019年。Mariia Vladimirova、Jakob Verbeek、Pablo Mesejo 和 Julyan Arbel。理解贝叶斯神经网络在单元层面的先验。*arXiv* 预印本 *arXiv:1810.05193*，2018年。Ulrike Von Luxburg。谱聚类教程。统计与计算，17:395–416，2007年。Li Wan、Matthew Zeiler、Sixin Zhang、Yann Le Cun 和 Rob Fergus。使用 DropConnect 对神经网络进行正则化。载于国际机器学习会议，第 1058–1066 页，2013。John N Weinstein、Eric A Collisson、Gordon B Mills、Kenna R Mills Shaw、Brad A Ozen- berger、Kyle Ellrott、Ilya Shmulevich、Chris Sander、Joshua M Stuart、癌症基因组图谱研究网络等。癌症基因组图谱泛癌分析项目。自然遗传学，45(10):1113，2013。

Jin Zeng、Jiahao Pang、Wenxiu Sun 和 Gene Cheung。《用于真实图像鲁棒去噪的深度图拉普拉斯正则化》。载于*IEEE*计算机视觉与模式识别会议研讨会论文集，第 0–0 页，2019年。

Cun-Hui Zh ang。《极小极大凹惩罚下近乎无偏的变量选择》。2010年。

Hui Zou。《自适应Lasso及其预言机性质》。美国统计学会会刊，101(476):1418–1429，2006年。