arXiv:2304.02911v2 [stat.ML] 7 Apr 2023

# 深度神经网络中权重矩阵的重尾正则化

Xuanzhe Xiao[1], Zeng Li*[1]、Chuanlong Xie[2], 和 Fengwei Zhou[3]

[1] 中国深圳南方科技大学，邮编 518055 12132907@mail.sustech.edu.cn liz9@sustech.edu.cn[2] 中国珠海北京师范大学，邮编 519087 clxie@bnu.edu.cn[3] 华为诺亚方舟实验室，中国香港 fzhou@connect.ust.hk

**摘要。** 揭示深度神经网络取得非凡成功并具备卓越泛化能力背后的原因，是一项艰巨的挑战。随机矩阵理论近期的研究进展，尤其是关于深度神经网络权重矩阵谱分析的成果，为解决这一问题提供了宝贵线索。一项关键发现表明，神经网络的泛化性能与其权重矩阵谱的重尾程度相关。为利用这一发现，我们提出一种新颖的正则化技术，称为**重尾正则化**，通过正则化明确促使权重矩阵的谱呈现更重的尾部。首先，我们采用加权 alpha 和稳定秩作为惩罚项；二者均可微，因此能够直接计算其梯度。为避免过度正则化，我们提出了两种惩罚函数变体。随后，我们从贝叶斯统计的视角出发，并借鉴随机矩阵方面的知识，提出两种新颖的重尾正则化方法，分别采用幂律分布和弗雷歇分布作为整体谱和最大特征值的先验。实验结果表明，在泛化性能方面，重尾正则化优于传统正则化技术。

**关键词：** 重尾正则化 · 深度神经网络 · 随机矩阵理论。

## 1 引言

近年来，深度神经网络（DNN）表现出卓越的性能，在计算机视觉、自然语言处理和推荐系统等多个领域取得了前所未有的成功 [6,10,11,25]。然而，人们对神经网络如何实现泛化仍缺乏清晰的认识。

* 通讯作者。

为 DNN 构建泛化框架的研究借鉴了传统学习理论中的各种数学工具 [3,4,5,24]。尽管如此，人们发现这些方法大多存在一定局限性。例如，VC 维和 Rademacher 复杂度被认为不足以令人满意地解释 DNN 的泛化性能 [26]。基于一致收敛的泛化界可能无法阐明深度学习中的泛化现象，因为其泛化保证是空泛的 [20]。

出于以下原因，人们可能会认为 DNN 的权重矩阵能够体现其泛化能力：从理论角度来看，权重矩阵中的参数与模型的输出空间、输入数据、优化算法等密切相关；此外，在实际场景中，由于行业高度分工，获取训练好的模型时通常无法获得太多有关训练数据和测试数据的信息。最近，Martin 和 Mahoney [16] 提出了一种基于随机矩阵理论（RMT）的视角，用以阐释深度神经网络的泛化行为。他们研究了深度神经网络中权重矩阵的经验谱分布（ESD），并观察到一种 *5+1* 正则化阶段 ——在整个训练过程中，权重矩阵的 ESD 最初与 Marchenko-Pastur（MP）定律高度吻合，随后逐渐偏离，最终趋近于重尾分布[14,16]。这种正则化现象被称为隐式自正则化。此外，该理论指出，大型且训练良好的 DNN 架构应呈现重尾自正则化，这意味着其权重矩阵的谱可以很好地拟合重尾分布。在 Martin 和 Mahoney 的研究基础上，Meng 和 Yao [19] 发现，分类问题的复杂度会影响 DNN 权重矩阵的谱。这些理论为探索 DNN 的泛化提供了全新视角。

除了上述研究外，还有一些工作从随机梯度下降（SGD）的角度，论证了权重矩阵的重尾特性对神经网络泛化的积极影响。Zhou 等人[27] 指出，SGD 和 Adam 逃离尖锐极小值所需的时间都与梯度噪声的重尾程度负相关。他们进一步解释说，在深度学习中，SGD 的泛化能力优于 Adam，是因为 Adam 的梯度计算经过了指数移动平均的平滑处理，导致其梯度噪声尾部比 SGD 更轻。Hodgkinson 等人 [12] 提出了类似的发现，证明在随机优化问题中，与加性噪声和轻尾稳态行为相比，乘性噪声和重尾稳态行为能够增强学习探索阶段在不同吸引域之间跳转的能力。Simsekli 等人 [22] 使用 Feller 过程近似 SGD 的轨迹，并推导出一个由豪斯多夫维数控制的泛化界，而豪斯多夫维数与该过程的尾部性质相关。他们的结果表明，尾部越重的过程应当能够实现更好的泛化。Barsbey 等人 [2]

他们认为，神经网络权重矩阵中存在的重尾特性有助于网络压缩，从而增强网络的泛化能力。综合来看，这些结果表明，权重矩阵的重尾特性是随机梯度下降（SGD）训练下深度神经网络泛化能力提升的一个根本因素。

这些理论引出了一种直观的观点：在深度神经网络训练过程中，权重矩阵呈现重尾特性，是获得良好泛化性能的关键。然而，先前的研究对如何增强神经网络中的重尾行为仍缺乏充分认识。本研究聚焦于对深度神经网络进行正则化，以促使其更快、更显著地呈现重尾行为。为此，我们提出了一种名为**重尾正则化**的显式正则化技术。实证结果表明，与采用传统方法训练的模型相比，采用重尾正则化训练的模型具有更优的泛化性能。

**本文的贡献**。

1. 我们提出了一种称为重尾正则化的正则化框架。这一提议受到先前研究的启发；这些研究表明，神经网络中的权重矩阵呈现重尾行为，有助于提升其泛化能力。2. 我们开发了四种不同的重尾正则化方法，包括 (a) 加权 Alpha 正则化、(b) 稳定秩正则化、(c) 幂律先验和 (d) Fréchet 先验。前两种方法受到现有神经网络复杂度度量的启发，后两种则借鉴了随机矩阵理论（RMT）和贝叶斯统计学的见解。3. 我们在 KMNIST 和 CIFAR10 等广泛使用的数据集上，将这些方法与传统方法进行了比较。数值实验表明，重尾正则化方法高效且优于其他传统正则化方法。

## 2 重尾正则化

### 2.1 定义

考虑一个 DNN $f_{\mathbf{W}}: \mathcal{X} \to \mathcal{Y}$，它有 $L$ 层，其全连接层的权重矩阵为 $\mathrm{W} = \{\mathrm{W}_1, \mathrm{W}_2, \cdots, , \mathrm{W}_L\}$，数据样本集为 $S = \{(x_1, y_1), (x_2, y_2) \cdots, (x_N, y_N)\}$，样本量为 $N$。

将 $l(f(x), y)$ 记为模型 $f_{\mathbf{W}}$ 下样本 $(x, y) \in \mathcal{X} \times \mathcal{Y}$的损失。DNN 的优化问题可视为一个带惩罚项的经验风险最小化问题：

$$\min_{\mathbf{W}} \quad \mathcal{L}(x, y) = \frac{1}{N} \sum_{i=1}^{N} l\left(f\left(x_i\right), y_i\right) + \lambda \sum_{l=1}^{L} p_l\left(\mathbf{W}_l\right), \tag{1}$$

其中， $\lambda$ 是一个调节参数， $p_l(\cdot)$ 是作用于权重矩阵的惩罚函数。

在此，我们提出一类称为 **重尾正则化** 的正则化方法，指的是有助于使模型的权重矩阵呈现更重尾特征的正则化方法。为实现这一目标， $p_l(\cdot)$ 应作为衡量模型的复杂指标，反映模型重尾行为的程度；尾部越重，该指标越小。

为描述权重矩阵谱的重尾程度，准确估计尾指数 $\alpha$ 至关重要。在统计学中，估计尾指数是一项棘手的工作。设数据点为 $\{x_i, 1 \leq i \leq n\}$ ，并假设数据来自密度函数为 $p(x) \sim cx^{-\alpha}$ 的重尾分布，即其概率密度函数与 幂律 $x^{-\alpha}$ 相当，其中 $x \to \infty$。估计尾指数 $\alpha$ 的一种通用方法是 Hill 估计量（HE），它适用于一般的幂律情形。若将数据按升序排列，HE 可写为：

$$\hat{\alpha} = 1 + \frac{k}{\left(\sum_{i=1}^{k} \ln \frac{x_{n-i+1}}{x_{n-k}}\right)}, \tag{2}$$

其中， $k$ 是一个调节参数。根据 $k$的取值，估计量的偏差与方差之间存在权衡。本研究采用 HE，使用$k = \frac{n}{2}$ 进行尾指数估计。

### 2.2 加权 Alpha 正则化

受 Martin和 Mahoney 的工作 [15,17,18], 启发，加权 *alpha* （也称为 *AlphaHat*）被用于我们的正则化方法中。在他们的理论中，模型的测试准确率与加权alpha 之间存在很强的线性相关性。加权 alpha 定义为：

$$\text{Weighted Alpha}(\mathbf{W}) = \sum_{l=1}^{L} \alpha_l \log \lambda_{\max,l}, \tag{3}$$

其中， $\alpha_l$ 是 $\mathsf{S}_l = \mathsf{W}_l^T \mathsf{W}_l$ 的所有正特征值的尾指数， $\lambda_{\max,\ l}$ 是 $\mathsf{S}_l$ 的最大特征值。在 Martin 和 Mahoney 的理论中，只讨论了加权 alpha 在大规模、预训练、最先进模型的不同架构上的表现，而我们关注的是这一指标在 DNN 训练过程中的变化。在这里，我们进行了一些实验，并获得证据表明，加权 alpha 与测试准确率呈负相关。因此，惩罚函数 $p_l(\cdot)$ 可以写作

$$p_l\left(\mathbf{W}_l\right) = \alpha_l \cdot \log \lambda_{\max,l}. \tag{4}$$

事实上，我们不需要在整个训练过程中都对加权 alpha 施加惩罚。我们的目标是在随机优化过程中引入重尾扰动，因此只需在训练早期或间歇性地启用正则化。否则，正则化程度会过高。另一方面，

出于实际考虑，我们可以在某个时点终止正则化，以避免高昂的计算成本。因此，我们还提供了以下两种惩罚函数变体：

1. 加权 alpha 衰减：

$$p_l\left(\mathbf{W}_l\right)=d\left(\lfloor e/m\rfloor\right)\cdot\alpha_l\cdot\log\lambda_{\max,l}, \tag{5}$$

其中， $e$ 是当前训练轮次编号， $m$ 是衰减频率， $d$(·)是一个称为衰减函数的递减函数。当 $d(x)=x^{-k}I_{\{}x^{-k}>t\}$ 时，衰减函数称为幂衰减；当$d(x)=\exp($ $-kx)$ $I_{\{}\exp(-kx)>t\}$时，则称为指数衰减，其中超参数为 $k$和 $t$。采用这种惩罚函数意味着，正则化仅在训练早期启用，并会随着训练进行而减弱。

2. 加权 alpha 下限阈值：

$$p_l\left(\mathbf{W}_l\right)=\alpha_l\cdot\log\lambda_{\max,l}\cdot I\left\{\sum_{l=1}^{L}\alpha_l\cdot\log\lambda_{\max,l}\geqslant t\right\}, \tag{6}$$

其中， $t$ 是一个超参数， $I\{\cdot\}$ 是指示函数。采用这种惩罚函数意味着，只有当加权 alpha 高于预设的下限阈值 $t$时，正则化才会启用；也就是说，模型未达到我们预期的重尾程度。

### 2.3 稳定秩正则化

稳定秩是深度学习中的一种经典度量，其定义为

$$\operatorname{stable}\left(\mathbf{W}_l\right)=\frac{\|\mathbf{W}_l\|_F^2}{\|\mathbf{W}_l\|_2^2}. \tag{7}$$

经验证，随着深度神经网络（DNNs）的训练进行，稳定秩会降低[16]。近期有几项研究 [4,21] 还表明，泛化误差可以上界为 $O\left(\prod_i\|\mathrm{W}_i\|_2^2\sum_i\right.$ 稳定（$\mathrm{W}_i$），这意味着更小的$\sum_i$ 稳定（$\mathrm{W}_i$）会带来更小的泛化误差。鉴于此，稳定秩正则化的惩罚函数可以写为 $p_l$ （$\mathrm{W}_l$） ＝稳定（$\mathrm{W}_l$）

，

因此，优化问题可以写成

$$\min_{\mathbf{W}}\quad\mathcal{L}\left(x,y\right)=\sum_{i=1}^{N}l\left(f\left(x_i\right),y_i\right)+\lambda\sum_{l=1}^{L}\frac{\|\mathbf{W}_l\|_F^2}{\|\mathbf{W}_l\|_2^2} \tag{8}$$

请注意， $||\mathrm{W}||_F^2$ 是W的奇异值平方之和，而 $||\mathrm{W}||_2$ 是W的最大奇异值。回顾随机矩阵理论，当矩阵具有重尾分布时，最大特征值会远离整体谱。结合Martin的 $5+1$相变理论 [16], ，权重矩阵更小的稳定秩会带来更强的重尾自正则化。

与加权 alpha 正则化类似，为了避免过度正则化，我们也可以为稳定秩正则化添加衰减项和上限阈值，如下所示：

1. 衰减稳定秩:

$$p_l\left(\mathbf{W}_l\right)=d\left(\lfloor e/m\rfloor\right)\cdot\frac{\left\|\mathbf{W}_l\right\|_F^2}{\left\|\mathbf{W}_l\right\|_2^2}. \tag{9}$$

2. 较低阈值稳定秩:

$$p_l\left(\mathbf{W}_l\right)=\frac{\left\|\mathbf{W}_l\right\|_F^2}{\left\|\mathbf{W}_l\right\|_2^2}\cdot I\left\{\sum_{l=1}^{L}\frac{\left\|\mathbf{W}_l\right\|_F^2}{\left\|\mathbf{W}_l\right\|_2^2}\geqslant t\right\}. \tag{10}$$

### 2.4 从贝叶斯视角看重尾正则化

这里，我们从贝叶斯视角提出两种重尾正则化方法。将深度神经网络视为概率模型 $P(\mathrm{y}|\mathrm{x},\mathrm{W})$，其中 $\mathrm{x}\in\mathcal{X}=\mathbb{R}^p$ 是输入，$\mathrm{y}\in\mathcal{Y}$ 是神经网络赋予的输出概率。$\mathrm{W}=\{\mathrm{W}_1,\cdots,\mathrm{W}_L\}$ 是神经网络的权重矩阵集合。给定训练样本集 $S=\{(x_1,y_1),(x_2,y_2)\cdots,(x_N,y_N)\}$，样本量为 $N$，估计权重 W 的一种常见方法是最大似然估计（MLE）：

$$\mathbf{W}^{\mathrm{MLE}}=\arg\max_{\mathbf{W}}\sum_{i=1}^{N}\log P\left(y_i\left|x_i,\mathbf{W}\right.\right). \tag{11}$$

具体而言，对于多分类任务，概率模型通常是多项分布，此时 MLE 可写为

$$\mathbf{W}^{\mathrm{MLE}}=\arg\max_{\mathbf{W}}\sum_{i=1}^{N}y_i\log f_{\mathbf{W}}\left(x_i\right). \tag{12}$$

从贝叶斯视角来看，如果我们想在模型中引入重尾正则化，可以为概率模型指定一个重尾先验，然后求最大后验估计（MAP），而非 MLE:

$$\mathbf{W}^{\mathrm{MAP}}=\arg\max_{\mathbf{W}}\log P\left(\mathbf{y}\left|\mathbf{x},\mathbf{W}\right.\right)+\log P\left(\mathbf{W}\right). \tag{13}$$

因此，选择一个合理的权重先验 $P(\mathrm{W})$ 至关重要，它可以使权重呈现更重的尾部。

回顾重尾矩阵的随机矩阵理论，如果一个随机矩阵具有重尾特性，那么其极限谱分布（极限谱分布）应呈幂律 [7,8,9]，其最大特征值应服从弗雷歇分布 [1,23]。因此，当我们引入全局谱或最大特征值的先验知识时，很容易将先验分布设定为幂律或弗雷歇分布。

现在我们为模型引入重尾先验。首先，我们考虑权重矩阵的整体谱先验。当权重矩阵为

重尾，极限谱分布呈幂律，因此先验分布可以设为

$$P(\mathbf{W}) = \prod_{l=1}^{L} P(\mathbf{W}_l) \propto \prod_{l=1}^{L} \prod_{j=1}^{K_l} \lambda_{l,j}^{-\alpha_l}, \tag{14}$$

其中， $\alpha_l$ 是神经网络第 $l$层权重矩阵 $\mathbf{W}_l$ 的奇异值平方幂律的尾指数，$\lambda_{l,\ j}$ 是 $\mathbf{W}_l$的第 $j$个奇异值平方。 $K_l$ 是被认为服从重尾分布的 $\mathbf{W}_l$ 的奇异值数量。 $K_l$ 是一个超参数，在我们的研究中，我们将其设为 $\mathbf{W}_l$ 大小的一半。将其代入式（13），得到以下优化问题：

$$\mathbf{W}^{\mathrm{MAP}} = \arg\max_{\mathbf{W}} \sum_{i=1}^{N} y_i \log f_{\mathbf{W}}(x_i) - \sum_{l=1}^{L} \sum_{j=1}^{K_l} \alpha_l \log \lambda_{l,j} \tag{15}$$

其次，我们考虑权重矩阵最大奇异值平方的先验。当权重矩阵具有重尾特性时，最大奇异值平方服从弗雷歇分布，因此先验分布可设为

$$P(\mathbf{W}) = \prod_{l=1}^{L} P(\mathbf{W}_l) = \prod_{l=1}^{L} \exp\left(-\lambda_{\max,l}^{-\alpha_l}\right). \tag{16}$$

其中， $\alpha_l$ 是 $\mathbf{W}_l$ 的尾指数，$\lambda_{\max,\ l}$ 是最大奇异值平方。同样，将其代入式（13），得到以下优化问题：

$$\mathbf{W}^{\mathrm{MAP}} = \arg\max_{\mathbf{W}} \sum_{i=1}^{N} y_i \log f_{\mathbf{W}}(x_i) - \sum_{l=1}^{L} \lambda_{\max,l}^{-\alpha_l}. \tag{17}$$

至此，我们推导出了两种形式的最大后验估计，但还有一些问题需要解决：如何确定超参数 $\boldsymbol{\alpha} = \{\alpha_l, 1 \le l \le L\}$，以及如何求解这个最大化问题。在经验贝叶斯中，超参数通过最大化数据的边际似然来确定，即

$$\boldsymbol{\alpha} = \arg\max_{\boldsymbol{\alpha}} \log \int P(\mathbf{y}, \mathbf{W} \,|\mathbf{x}, \boldsymbol{\alpha}) \, \mathrm{d}\mathbf{W}. \tag{18}$$

显然这是不可能的，因为该积分难以计算。根据 Mandt 等人， [13], 随机梯度下降可视为一种用于贝叶斯推断的变分期望最大化（VEM）方法。(15) 和 (17) 中的最大化问题等价于以下最小化问题：

$$\min_{\mathbf{W}} \quad \mathcal{L}(x, y) = \frac{1}{N} \sum_{i=1}^{N} l(f(x_i), y_i) + \sum_{l=1}^{L} \sum_{j=1}^{K_l} \alpha_l \log \lambda_{l,j}, \tag{19}$$

$$\min_{\mathbf{W}} \quad \mathcal{L}(x, y) = \frac{1}{N} \sum_{i=1}^{N} l(f(x_i), y_i) + \sum_{l=1}^{L} \lambda_{\max,l}^{-\alpha_l}, \tag{20}$$

其中，$l(f(x)\ y) = -y \log f(x)$ 是交叉熵损失。当将随机梯度下降视为一种变分期望最大化算法时，可以优化超参数 $\boldsymbol{\alpha}$。与最大似然估计不同，Hill 估计量是估计超参数$\boldsymbol{\alpha}$的更好选择。加入调节参数后，(19) 和 (20) 可以修改为：

$$\min_{\mathbf{W}} \quad \mathcal{L}(x,y) = \frac{1}{N}\sum_{i=1}^{N} l\left(f\left(x_i\right), y_i\right) + \mu \sum_{l=1}^{L}\sum_{j=1}^{K_l} \hat{\alpha}_l \log \lambda_{l,j}, \tag{21}$$

$$\min_{\mathbf{W}} \quad \mathcal{L}(x,y) = \frac{1}{N}\sum_{i=1}^{N} l\left(f\left(x_i\right), y_i\right) + \mu \sum_{l=1}^{L} \lambda_{\max,l}^{-\hat{\alpha}_l}, \tag{22}$$

其中，$\hat{}\alpha_l$ 是第 $l$层权重矩阵的 Hill 估计量。

请注意，当 $p_l(\mathrm{W}) = \alpha_l \cdot \sum_j \log \lambda_{j,l}$ 且 $p_l(\mathrm{W}) = \lambda^{\alpha_l}{}_{\max\, l}$时，(21) 和 (22) 是 (1) 的特殊情形。由于惩罚项与加权 alpha 相似，这些正则化项可视为加权 Alpha 的变体。根据这些正则化中使用的先验，我们将 (21) 称为**幂律先验下的重尾正则化**，将 (22) 称为**Fréchet 先验下的重尾正则化**。

## 3 实验

本节通过多分类任务的实验展示重尾正则化的性能。为验证重尾正则化的有效性，我们采用了加权 Alpha 正则化和稳定秩正则化及其变体，以及幂律和 Fréchet 谱先验下的重尾正则化。此处所有尾指数均由其 Hill 估计量替代，$k = \frac{n}{2}$，其中 $n$ 是相应权重矩阵的大小。在实验中，我们采用以下训练方法与重尾正则化方法进行比较：

1. 原始问题（基准）：我们考虑了未采用任何显式正则化的原始模型。2. 权重衰减：我们考虑了式（1）中最常用的显式正则化，其中 $p_l(\mathrm{W}) = \frac{1}{2}||\mathrm{W}||_F^2$。
3. 谱范数正则化：我们考虑了另一种基于谱的显式正则化方法，其惩罚函数为 $p_l(\mathrm{W}) = \frac{1}{2}||\mathrm{W}||_2^2$。

这里的所有实验均基于小批量随机梯度下降和学习率衰减。在我们的实验中，针对模型和数据集采用了以下四种设置：

1. KMNIST 和 CIFAR10 上的三层神经网络（FC3）。2. CIFAR10 上的 LeNet5。3. CIFAR10 上的 ResNet18。

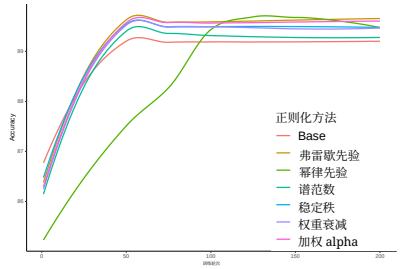


图 1：KMNIST 上的 FC3

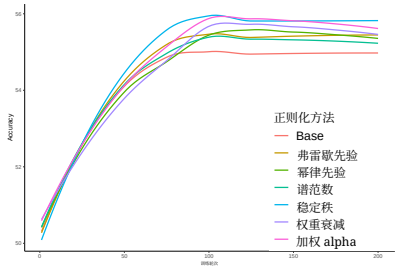


图 2：CIFAR10 上的 FC3

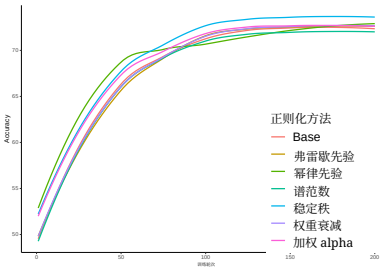


图 3：CIFAR10 上的 LeNet5

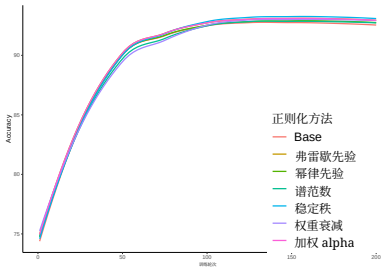


图 4：CIFAR10 上的 ResNet18

### 3.1 FC3

首先，我们在 KMNIST 和 CIFAR10 数据集上训练具有三个隐藏层的神经网络，共训练 200 轮次。KMNIST 数据集改编自 Kuzushiji 数据集，可直接替代 MNIST 数据集。KMNIST 数据集的图像大小为 $28\times28$。CIFAR10 数据集包含 60000 张彩色图像，图像大小为 $32\times32$。CIFAR10 数据集比 KMNIST 数据集更复杂，因此更难正确分类。由于这两个数据集的图像大小不同，我们针对每个数据集使用了两种不同规模的三层神经网络。对于 KMNIST 数据集，我们使用层大小为 $\mathbf{n} = [784, 128, 128, 128, 10]$ 的网络；对于 CIFAR10 数据集，我们使用层大小为 $\mathbf{n} = [3072, 512, 256, 256, 10]$ 的网络。

结果如图1、图2和表1所示。在 KMNIST 数据集和 CIFAR10 数据集上，所有重尾正则化的准确率都高于原始问题。Fréchet 先验在 KMNIST 数据集上取得了最佳测试准确率，而阈值较低（$t = 15$）的稳定秩在 CIFAR10 数据集上取得了最佳测试准确率。

[1] 我们使用带有 $k = 2$ 的幂衰减加权 alpha。[2] 我们使用较低阈值的稳定秩，并配有 $t = 1$。

表1：FC3 在 KMNIST 和 CIFAR10 数据集上采用不同正则化方法时的测试准确率平均值（± 标准误差）

| 网络 | 数据集 | 方法 | $\beta$ | 测试准确率 |
|---|---|---|---|---|
| FC3 | KMNIST | base | | 89.19±0.020 |
| | | 权重衰减 | $5.00\times10^{-4}$ | 89.44±0.037 |
| | | 谱范数 | 0.0001 | 89.27±0.013 |
| | | 加权 alpha[1] | $5.00\times10^{-5}$ | 89.60±0.011 |
| | | 稳定秩[2] | $1.00\times10^{-4}$ | 89.48±0.008 |
| | | 幂律先验 | $5.00\times10^{-4}$ | 89.58±0.175 |
| | | **Fréchet 先验** | $\mathbf{2.00\times10^{-5}}$ | **89.64±0.173** |
| | CIFAR10 | base | | 54.97±0.039 |
| | | 权重衰减 | $5.00\times10^{-4}$ | 55.56±0.092 |
| | | 谱范数 | 0.0001 | 55.27±0.003 |
| | | 加权 alpha[1] | $1.00\times10^{-4}$ | 55.72±0.053 |
| | | **稳定秩[2]** | $\mathbf{1.00\times10^{-4}}$ | **55.82±0.041** |
| | | 幂律先验 | $5.00\times10^{-5}$ | 55.44±0.038 |
| | | Fréchet 先验 | $5.00\times10^{-5}$ | 55.45±0.029 |

表 2：LeNet5 在 CIFAR10 数据集上采用不同正则化方法时的测试准确率平均值（± 标准误差）

| 网络 | 数据集 | 方法 | $\beta$ | 测试准确率 |
|---|---|---|---|---|
| LeNet5 | CIFAR10 | base | | 72.42±0.213 |
| | | 权重衰减 | $5.00\times10^{-4}$ | 72.62±0.277 |
| | | 谱范数 | 0.0001 | 71.98±0.275 |
| | | 加权 alpha[1] | 0.004 | 72.61±0.300 |
| | | **稳定秩** | **0.1** | **73.63±0.193** |
| | | 幂律先验 | $7.00\times10^{-4}$ | 72.61±1.061 |
| | | 弗雷歇先验 | $5.00\times10^{-5}$ | 72.58±0.270 |

### 3.2 LeNet5

其次，我们在 CIFAR10 数据集上对 LeNet5 训练 200 个训练轮次。LeNet5 是一种著名且经典的卷积神经网络（CNN）架构。结果如图 3 和表 2 所示。在 CIFAR10 数据集上，重尾正则化的准确率也均优于原始问题。如表中所示，使用 $\beta = 0.1$ 的稳定秩达到了最佳测试准确率。

### 3.3 ResNet18

第三，我们在 CIFAR10 数据集上训练 ResNet18，共进行 200个训练轮次。ResNet 是一种卷积神经网络架构，在各种领域将最先进水平大幅提升。

表3：CIFAR10 数据集上采用不同正则化方法的 ResNet18 的测试准确率平均值（± 标准误差）

| 网络 数据集 | 方法 | $\beta$ | 测试准确率 |
|---|---|---|---|
| ResNet18 CIFAR10 | base | | 92.65±0.066 |
| | 权重衰减 | $5.00\times10^{-4}$ | 93.15±0.087 |
| | 谱范数 | 0.0001 | 92.78±0.069 |
| | 加权 alpha | $5.00\times10^{-5}$ | 93.04±0.045 |
| | **稳定秩** | $\mathbf{5.00}\times10^{-4}$ | **93.19±0.049** |
| | 幂律先验 | $1.00\times10^{-4}$ | 92.85±0.111 |
| | 弗雷歇先验 | $3.00\times10^{-5}$ | 93.07±0.086 |

计算机视觉任务。在该实验中，我们在原始ResNet18架构中的线性层之前添加了一个大小为$512\times128$的线性层。结果如图4和表3所示。如表中所示，具有$\beta = 5 \times 10^{-4}$的稳定秩实现了最佳测试准确率。

## 参考文献

1. Auffinger, A.、Ben Arous, G.、Péché, S.：重尾随机矩阵最大特征值的泊松收敛。载于：《IHP概率与统计年鉴》，第45卷，第589–610页（2009年） 2. Barsbey, M.、Sefidgaran, M.、Erdogdu, M.A.、Richard, G.、Simsekli, U.：随机梯度下降中的重尾现象与过参数化神经网络的可压缩性。《神经信息处理系统进展》**34**，29364–29378（2021年） 3. Bartlett, P.、Maiorov, V.、Meir, R.：分段多项式网络的近线性VC维数界。《神经信息处理系统进展》**11**（1998年） 4. Bartlett, P.L.、Foster, D.J.、Telgarsky, M.J.：神经网络的谱归一化间隔界。《神经信息处理系统进展》**30**（2017年） 5. Bartlett, P.L. 、 Mendelson, S.：Rademacher 复杂度与高斯复杂度：风险界与结构性结果。《机器学习研究杂志》**3**（11月），463–482（2002年） 6. Chen, Q.、Zhao, H.、Li, W.、Huang, P.、Ou, W.：面向阿里巴巴电子商务推荐的行为序列 Transformer。载于：第一届高维稀疏数据深度学习实践国际研讨会论文集（2019年12月）。https://doi.org/10.1145/3326937.3341261，http://dx.doi.org/10 3341261 7 1145 3326937 . / . . Davis, R.A.、Heiny, J.、Mikosch, T.、Xie, X.：重尾多元时间序列样本自协方差矩阵的极值分析。《Extremes》**19**（3），517–547（2016年） 8. Davis, R.A.、Mikosch, T.、Pfaffel, O.：重尾多元时间序列样本协方差矩阵的渐近理论。《随机过程及其应用》**126**（3），767–799（2016年） 9. Davis, R.A.、Pfaffel, O.、Stelzer, R.：具有重尾的样本协方差矩阵最大特征值的极限理论。《随机过程及其应用》**124**（1），18–50（2014年）

10. Galassi, A.、Lippi, M.、Torroni, P.：自然语言处理中的注意力。《IEEE神经网络与学习系统汇刊》**32**（10），4291–4308（2020年9月）。https://doi.org/10.1109/tnnls.2020.3019893，http://dx.doi.org/10 3019893 11 1109 2020 . /tnnls. . . He, K.、Zhang, X.、Ren, S.、Sun, J.：用于图像识别的深度残差学习。载于：IEEE计算机视觉与模式识别会议论文集，第770–778页（2016年） 12. Hodgkinson, L.、Mahoney, M.：随机优化中的乘性噪声与重尾现象。载于：国际机器学习会议，第4262–4274页。机器学习研究论文集（2021年） 13. Mandt, S.、Hoffman, M.D.、Blei, D.M.：作为近似贝叶斯推断的随机梯度下降。arXiv预印本，arXiv:1704.04289（2017年） 14. Martin, C.H.、Mahoney, M.W.：神经网络模型中的传统自正则化与重尾自正则化。arXiv预印本，arXiv:1901.08276（2019年） 15. Martin, C.H.、Mahoney, M.W.：重尾普适性预测超大规模预训练深度神经网络测试准确率的趋势。载于：2020年SIAM国际数据挖掘会议论文集，第505–513页。SIAM（2020年） 16. Martin, C.H.、Mahoney, M.W.：深度神经网络中的隐式自正则化：来自随机矩阵理论的证据及其对学习的影响。《机器学习研究杂志》**22**（165），1–73（2021年） 17. Martin, C.H.、Mahoney, M.W.：一次深度学习竞赛的事后分析：辛普森悖论，以及尺度度量与形状度量的互补作用。arXiv预印本，arXiv:2106.00734（2021年） 18. Martin, C.H.、Peng, T.S.、Mahoney, M.W.：在无法获取训练数据或测试数据的情况下预测最先进神经网络的质量趋势。《自然通讯》**12**（1），1–13（2021年） 19. Meng, X.、Yao, J.：分类难度对深度学习中权重矩阵谱的影响及其在提前停止中的应用。arXiv预印本，arXiv:2111.13331（2021年） 20. Nagarajan, V.、Kolter, J.Z.：一致收敛可能无法解释深度学习中的泛化。《神经信息处理系统进展》**32**（2019年） 21. Neyshabur, B.、Bhojanapalli, S.、Srebro, N.：用于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） 22. Simsekli, U.、Sener, O.、Deligiannidis, G.、Erdogdu, M.A.：豪斯多夫维数、重尾现象与神经网络中的泛化。《神经信息处理系统进展》**33**，5138–5151（2020年） 23. Soshnikov, A.：具有重尾的Wigner随机矩阵最大特征值的泊松统计。《概率电子通讯》**9**，82–91（2004年） 24. Vapnik, V.、Levin, E.、Le Cun, Y.：测量学习机器的VC维数。《神经计算》**6**（5），851–876（1994年） 25. Vaswani, A.、Shazeer, N.M.、Parmar, N.、Uszkoreit, J.、Jones, L.、Gomez, A.N.、Kaiser, L.、Polosukhin, I.：注意力即你所需。arXiv **abs/1706.03762**（2017年） 26. Zhang, C.、Bengio, S.、Hardt, M.、Recht, B.、Vinyals, O.：理解深度学习（仍然）需要重新思考泛化。《美国计算机协会通讯》**64**（3），107–115（2021年） 27. Zhou, P.、Feng, J.、Ma, C.、Xiong, C.、Hoi, S.C.H.等：从理论上理解为什么深度学习中的随机梯度下降比Adam具有更好的泛化能力。《神经信息处理系统进展》**33**，21285–21296（2020年）