基于进化选择性微调的迁移学习优化

Jacinto Colan[1], Ana Davila[2] and Yasuhisa Hasegawa[2]

摘要—深度学习在图像分析方面取得了显著进展。然而，大型完全训练模型的计算需求仍然是需要考虑的因素。迁移学习为适应预训练模型到新任务提供了一种策略。传统的微调通常涉及更新所有模型参数，这可能会导致过拟合和更高的计算成本。本文介绍了 BioTune，这是一种进化自适应微调技术，通过选择性地对某些层进行微调以增强迁移学习的效率。BioTune 采用进化算法来识别一组需要微调的重点层，旨在优化给定目标任务上的模型性能。在来自不同领域的九个图像分类数据集上进行评估表明，与现有的微调方法如 AutoRGN 和 LoRA 相比，BioTune 实现了具有竞争力或改进的准确性和效率。通过将微调过程集中在相关的一组层上，BioTune 减少了可训练参数的数量，这可能会导致计算成本降低，并促进跨不同数据特性和分布的有效迁移学习。

***Index Terms*—迁移学习，进化优化，微调，图像分析**

I. 介绍

尽管深度学习在各种图像分析应用中取得了成功，但在需要专家标注或涉及敏感信息的许多专业领域中，对大量标注数据的需求仍然是一个重大挑战，例如医学成像。这种数据匮乏促使开发能够更有效地利用现有标注数据的技术。

迁移学习通过将预训练模型从数据丰富的领域适应到特定目标任务，为有限数据的挑战提供了实际解决方案。这种方法减少了对大量标记数据集的需求，同时提高了训练效率和分类性能。然而，当源域与目标域存在显著差异时，有效的迁移学习面临重大挑战，可能导致负面转移或灾难性遗忘，使得适应后的模型表现劣于基线训练。微调是一种常见的迁移学习方法，涉及使用任务特定的数据调整预训练模型的参数。尽管这种方法利用了有用的特征表示，但它也带来了一些技术挑战，包括可能降低预训练特征的质量和在分布外数据上的性能下降，尤其是在领域差异极大的情况下 [1]。微调中的一个常见做法是选择性地调整层。然而，确定哪些层需要进行微调增加了复杂性。传统上，人们只专注于训练终端层，但最近的研究表明，自适应的层选择策略可能更有效 [2]。此外，学习率和权重衰减等额外微调参数显著影响模型性能和稳定性 [3]，通常需要领域专业知识 [4]。

为了解决这些挑战，微调过程可以被定义为一个优化问题，从而能够应用自然启发的算法。这些算法在处理复杂的优化任务中展现了有效性，包括带约束的逆运动学 [5]、超参数调整 [6] 和特征选择 [7]。其中，进化算法（EA）和群体智能（SI）方法各自具有独特的优点。EAs 在保持种群多样性、增强解决方案稳健性方面表现出色，而 SI 算法则以效率和简洁著称 [8]。结合了 EA 和 SI 元素的混合算法旨在利用它们各自的优点，平衡新解探索与已知良好配置的开发 [9]。

本文介绍了一种使用改进的进化算法来探索微调配置的自适应微调方法。所提出的方法自动确定层冻结策略并优化活跃层的学习率，通过在每次进化生成中对目标数据集子集进行系统评估实现有效的探索。本文的主要贡献如下。

- 一种基于进化的自适应微调方法，能够动态优化学习率和层冻结策略，在不同的数据集和架构中进行调整。
- 跨越九个图像分类数据集和四种卷积神经网络架构的全面对比，展示了改进的性能和适应性。

II. 提议的方法

我们提出了一种新颖的选择性和自适应微调框架，该框架同时优化要微调的层集合及其对应的 learning rates 以最大化分类性能。我们的方法利用改进的进化优化策略来引导探索过程 [10]。所提方法的概述如图 1 所示。BioTune 方法包括以下阶段：模型选择和预训练、分层数据划分、进化搜索、适应度评估以及使用最优配置进行微调。

A. 微调优化问题

给定一个预训练模型 $M = \{m_b : b \in \{0, \ldots, B\}\}$，由基于其功能分组的 $B+1$ 个层或块集组成（例如，属于残差块或全连接分类器的层），该模型在具有丰富标

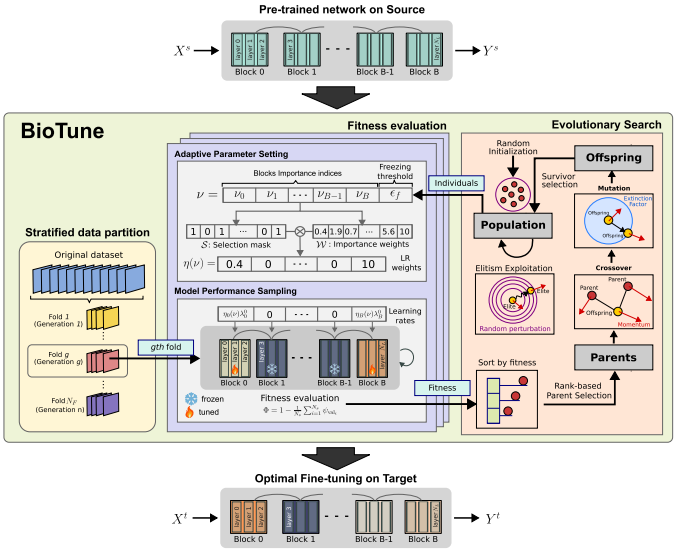

图 1. BioTune 概览

记图像的数据集 $\mathcal{X}_s = \{(x_i^s, y_i^s)\}_{i=1}^{N_s}$ 上进行预训练，我们的目标是确定一个最佳微调配置 ν^*，以最大化网络在包含不同类别的目标数据集 $\mathcal{X}_t = \{(x_i^t, y_i^t)\}_{i=1}^{N_t}$ 上的准确性。优化问题可以表述为：

$$\nu^* = \arg\max_{\nu} \mathrm{Acc}(M(\omega^0), \lambda(\nu), \mathcal{X}_t) \tag{1}$$

其中 ω^0 表示预训练模型的参数，而 $\lambda(\nu)$ 定义了微调过程中每个块的学习率，由以下给出：

$$\lambda(\nu) = \{\eta_b(\nu)\lambda_b^0 : b \in \{0, \ldots, B\}\} \tag{2}$$

其中，λ_b^0 是每个块 b 的预定义基础学习率，而 $\eta_b(\nu)$ 是一个权重，根据每个块在微调期间的重要性调整基础学习率。函数 $\eta_b(\nu)$ 通过启用块级选择和学习率分配来控制微调强度。对于关键适应层，$\eta_b(\nu) > 1$ 增加学习率以进行重大更新；对于不太重要的层，$\eta_b(\nu) < 1$ 减少速率以进行最小调整；而对于非贡献层，$\eta_b(\nu) = 0$ 冻结参数，减少计算成本并加速微调。最优配置 ν^* 是通过迭代探索配置空间的进化搜索确定的。

B. 进化搜索

微调配置 ν 被编码为基因型，其中第 s 个个体的基因型表示为 $\nu^s = [\nu_0^s, \nu_1^s, \ldots, \nu_B^s, \nu_{B+1}^s]$。每个基因 ν_b^s，其中 $b \in 0, \ldots, B$，对应于块 b 的重要索引，并且范围从 0 到 1。最终基因 $\nu_{B+1}^s = \epsilon_f^s$ 作为冻结阈值，区分要冻结的块和要微调的块。此阈值 ϵ_f^s 是优化参数，可促进种群 [15] 内的多样性。

$$\nu_b^{0,\ldots,N_p-1} = U_{[0,1]} \quad \forall b = 0, \ldots, B+1 \tag{3}$$

每个个体的适应度通过函数 Φ 进行评估，该函数衡量其在验证集上的预测性能。此评估包括两个阶段：自适应参数设置和模型性能采样。

1) 自适应参数设置: 对于被评估的个体 ν，每个块 b 的学习率权重 $\eta(\nu_b) = \mathcal{S} \cdot \mathcal{W}$ 由选择掩码 $\mathcal{S}$ 和重要性权重 $\mathcal{W}$ 向量的乘积决定：

选择掩码 $\mathcal{S}$ 控制相应的层集合是否启用微调或保持冻结，基于基因值 ν_b 和冻结阈值 ϵ_f。具体来说，掩码定义为：

$$\mathcal{S}_b = \begin{cases} 0, & \text{if } \nu_b \leq \epsilon_f \\ 1, & \text{if } \nu_b > \epsilon_f \end{cases} \tag{4}$$

这里，$\mathcal{S}_b = 0$ 表示块 b 中的层被冻结，而 $\mathcal{S}_b = 1$ 允许对该块进行微调。阈值 ϵ_f 作为截止值来确定某个块是否对微调过程有贡献。重要性权重 $\mathcal{W}$ 按照以下方式计算：

$$\mathcal{W}_b = 10^{2(\nu_b - 0.5)} \tag{5}$$

此公式允许 $\mathcal{W}$ 在 [0.1,10] 范围内指数级缩放学习率，强调重要性较高的层同时弱化重要性较低的层。最后，每个块 b 的学习率 λ_b 是通过方程 2计算得出。

2) 模型性能采样: 对于每个个体，相应的学习率配置应用于预训练模型。如果某块的 $\eta(\nu_b)$ 为 0，则其参数被冻结。否则，该块的学习率根据其对应的权重 $\eta(\nu_b)$ 进行缩放，通过基于梯度的优化方法在训练过程中实现差异化的参数更新。然后使用分类交叉熵损失，在目标训练集上以固定数量的 epoch 对模型进行微调。为了确保鲁棒性，该过程使用 N_s 个不同的随机种子重复执行。这些试验中验证准确率的平均值用作适应度指标 $\Phi(\nu) = 1 - \frac{1}{N_s}\sum_{i=1}^{N_s} \psi_{\mathrm{val}_i}(\nu)$。这里，$\psi_{val_i}$ 是第 i 次试验在验证集上获得的准确率。然后根据它们的适应度值对种群进行排序，并选择表现最好的个体作为下一代的父代。随后，对这些父代应用演化操作以生成后代：利用、交叉、变异和适应。

后代随后被评估，其中适应度最高的个体被选中以形成新的种群。这一循环在多代中重复进行，直到识别出最优的微调配置，这由验证集上的最高预测准确率指

表 I
三种运行中各种微调方法和数据集的准确性与标准误差对比。对于 BioTune，第二行显示了相对于 FT 的相对性能改进。最高得分用粗体突出显示。

	数字			对象		细粒度		专业化	
Method	MNIST	USPS	SVHN	CIFAR-10	STL-10	Flowers-102	FGVC-Aircraft	DTD	ISIC2020
FT	98.96 (0.0)	97.05 (0.1)	95.56 (0.2)	95.65 (0.1)	97.33 (0.0)	85.33 (0.5)	58.68 (1.9)	68.03 (0.1)	78.91 (0.7)
LP [1]	92.53 (0.2)	91.78 (0.0)	44.08 (0.1)	81.03 (0.0)	96.86 (0.1)	82.72 (0.6)	33.61 (0.4)	66.00 (0.1)	77.49 (0.5)
L^1-SP [11]	98.98 (0.1)	97.31 (0.0)	96.01 (0.0)	95.60 (0.0)	97.01 (0.0)	87.82 (0.5)	60.55 (1.9)	68.52 (0.1)	80.62 (1.2)
L^2-SP [11]	98.87 (0.0)	97.00 (0.1)	95.47 (0.1)	95.78 (0.1)	97.20 (0.1)	85.29 (0.5)	61.56 (1.1)	69.01 (0.2)	79.77 (2.0)
G-LF [12]	98.82 (0.1)	96.72 (0.2)	94.00 (0.0)	93.77 (0.0)	97.32 (0.0)	87.59 (0.9)	54.77 (1.3)	67.85 (0.3)	77.49 (1.2)
G-FL [13]	98.57 (0.0)	96.86 (0.1)	94.91 (0.0)	95.43 (0.0)	97.01 (0.0)	86.14 (0.2)	49.22 (0.7)	65.42 (0.3)	76.92 (1.3)
AutoRGN [2]	99.00 (0.0)	96.91 (0.2)	**96.08 (0.0)**	96.05 (0.0)	96.92 (0.1)	85.5 (0.3)	57.94 (0.8)	65.70 (0.2)	79.48 (0.4)
LoRA [14]	98.51 (0.1)	96.92 (0.0)	95.46 (0.1)	95.17 (0.1)	97.46 (0.1)	86.01 (0.2)	54.78 (1.3)	68.17 (0.4)	80.91 (1.0)
BioTune (Ours)	**99.13 (0.0)** +0.2%	**97.57 (0.1)** +0.5%	95.85 (0.0) +0.3%	**96.09 (0.1)** +0.5%	**97.50 (0.0)** +0.2%	**91.68 (0.1)** +6.7%	**64.40 (0.6)** +9.7%	**69.27 (0.6)** +1.8%	**82.90 (0.8)** +5.1%

示。然后使用这个最优配置对模型进行全面训练集的微调，并最终在目标数据集的保留测试集上进行评估。

C. 分层数据划分

为了解决进化搜索的计算需求，我们采用了分层数据划分。训练数据集被划分为 N_s 个分层折叠 $f_g = g \bmod N_s$。每个折叠保留了原始类分布，并且第 f_g 个折叠用于在第 g 代进行评估。这减少了每一代的计算成本，并通过使候选者在不同世代暴露于不同的训练样本中确保了稳健的探索。

III. 实验验证

A. 数据集

为了全面评估我们方法的性能和泛化能力，我们使用了一组多样化的图像分类数据集：Flowers-102[16], MNIST[17], USPS[18], SVHN[19], CIFAR-10[20], STL-10[21], FGVC-Aircraft[22], DTD[23], ISIC2016[24]。这些数据集涵盖了广泛的领域，包括数字分类、自然物体分类、细粒度分类和专业领域。

B. 实验设置与实现

为了评估我们的方法，我们使用了在 ImageNet 上预训练的 ResNet-50 架构 [25]，并在 NVIDIA RTX A6000 GPU 上的 PyTorch 2.0 中实现。对于 BioTune，我们使用了一个包含 10 个个体的人口，其中有 3 个精英个体，最多 3 代，每个适应度评估使用 3 个随机种子。BioTune 训练限制在 30 个周期内，并且为了提前停止设置了 3 个周期的耐心值，以及一个 0.25 的随机扰动用于探索。我们故意排除了数据增强，以专注于微调的影响，并实现了各种已建立的微调技术进行全面比较 [26]。

C. 在多样图像分类数据集上的性能

提出的 BioTune 模型在多种图像分类任务上的实验结果见表 I，其中我们使用测试集准确率作为主要性能指标。我们报告了三次独立运行的平均准确率以及标准误差（用括号表示）。对于 BioTune 而言，我们评估了所有探索生成中排名前五的表现配置，并报告最高达到的准确率。

结果显示，BioTune 在大多数数据集上与现有的微调方法相比始终能够达到具有竞争力或更优的性能，SVHN 是唯一的例外。对于与自然图像源领域紧密相关的数据集，BioTune 相对于 FT 基准展示了适度的改进（大约 0.5%）。在细粒度分类（如 FGVC-Aircraft 实现了 9.7%的提升）和远离源领域的专业数据集上观察到了更显著的增益，例如 DTD（纹理）和 ISIC2020（皮肤镜），分别显示了 1.8%和 5.1%的改进。

图 2提供了 BioTune 对每个数据集的最佳微调配置的综合可视化，冻结层用雪花符号表示，学习率权重以热图显示（值为 0.1-10），揭示了不同迁移学习场景中各层适应模式的不同。

表 II则通过展示 BioTune 使用的可训练参数占 ResNet-50 总可用参数百分比来补充这一点，显示出任务间参数使用率有显著差异，并证明了 BioTune 能够自动确定微调的最佳参数子集的能力。

我们探讨了每一代中训练数据百分比的变化如何影响 BioTune 的性能，突出了计算成本与性能提升之间

表 II
RESNET-50 在每个数据集中的可训练参数百分比

	数字			对象		专业化的			
Method	MNIST	USPS	SVHN	CIFAR-10	STL-10	Flower-102	FGVC-Aircraft	DTD	ISIC2020
BioTune	29.97	36.86	100.0	100.0	64.93	99.12	99.96	64.89	29.93

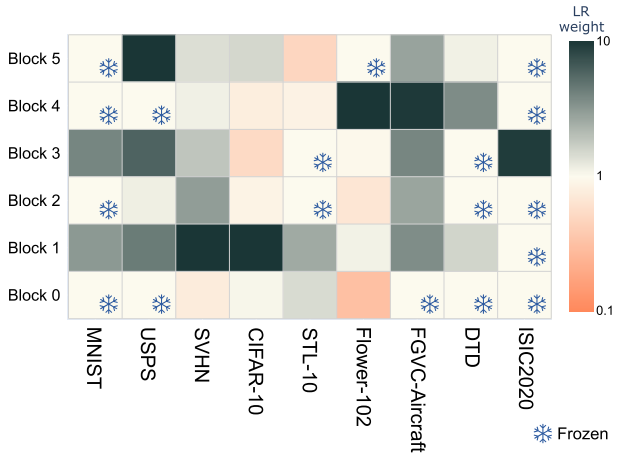

图 2. 通过 BioTune 在不同数据集上发现的最优微调配置。被冻结的层用雪花符号标记，学习率权重以热图形式显示（从 0.1 到 10）。

表 III
不同训练数据百分比下的平均测试准确率和计算时间

	训练数据集的百分比				
Dataset	10%	20%	25%	50%	100%
Mean Accuracy	90.46	90.34	90.53	90.95	91.1
Comp. time (hours)	1.6	2.2	3.7	6.0	11.4

的权衡。表 III总结了这些发现对于 Flowers-102 数据集的影响，表明仅使用 10%的训练数据并通过极短的计算时间（1.6 小时），可以达到较高的测试准确率（>90%）。虽然增加训练数据可以提高性能并减少方差，但它显著增加了计算成本——使用 100%的训练数据达到峰值准确率 91.1%需要 11.4 小时。为了我们的实验验证，我们选择了一个平衡的 50%分割来优化准确性与效率之间的权衡。

IV. 结论

我们介绍了 BioTune，这是一种基于进化的自适应微调方法，通过动态确定层冻结和学习率来优化迁移学习。我们在多样数据集上的全面基准测试表明，与现有方法相比，BioTune 表现出更优的性能，在选择性层冻结和减少可训练参数的同时实现了更高的准确性和效率提升。BioTune 为开发更加稳健的深度学习模型提供了一个有用的工具，使其能够通过有效的迁移学习在更广泛的应用领域中发挥作用。

参考文献

[1] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," 2022, arXiv:2202.10054.

[2] Y. Lee, A. S. Chen, F. Tajwar, A. Kumar, H. Yao, P. Liang, and C. Finn, "Surgical fine-tuning improves adaptation to distribution shifts," 2023, arXiv:2210.11466.

[3] T. Alshalali and D. Josyula, "Fine-tuning of pre-trained deep learning models with extreme learning machine," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 469–473.

[4] B. Nguyen and S. Ji, "Fine-tuning pretrained language models with label attention for biomedical text classification," 2022.

[5] A. Davila, J. Colan, and Y. Hasegawa, "Real-time inverse kinematics for robotic manipulation under remote center-of-motion constraint using memetic evolution," *Journal of Computational Design and Engineering*, vol. 11, no. 3, pp. 248–264, 2024.

[6] A. M. Vincent and P. Jidesh, "An improved hyperparameter optimization framework for automl systems using evolutionary algorithms," *Scientific Reports*, vol. 13, no. 1, p. 4737, 2023.

[7] M. Nssibi, G. Manita, and O. Korbaa, "Advances in nature-inspired metaheuristic optimization for feature selection problem: A comprehensive survey," *Computer Science Review*, vol. 49, p. 100559, 2023.

[8] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 2, 2004, pp. 1980–1987 Vol.2.

[9] C. Grosan and A. Abraham, "Hybrid evolutionary algorithms: methodologies, architectures, and reviews," in *Hybrid evolutionary algorithms*. Springer, 2007, pp. 1–17.

[10] S. Starke, N. Hendrich, and J. Zhang, "Memetic evolution for generic full-body inverse kinematics in robotics and animation," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 406–420, 2019.

[11] X. Li, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 10–15 Jul 2018, pp. 2825–2834.

[12] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 2018.

[13] S. Mukherjee and A. H. Awadallah, "Distilling bert into simple neural networks with unlabeled transfer data," 2020, arXiv:1910.01769.

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021, arXiv:2106.09685.

[15] D. Fister, I. Fister, T. Jagrič, I. Fister, and J. Brest, "A novel self-adaptive differential evolution for feature selection using threshold mechanism," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 17–24.

[16] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008, pp. 722–729.

[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[19] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, *et al.*, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, no. 2, 2011.

[20] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009, Technical Report.

[21] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 215–223.

[22] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013. [Online]. Available: https://arxiv.org/abs/1306.5151

[23] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[24] V. Rotemberg, N. Kurtansky, B. Betz-Stablein, L. Caffery, E. Chousakos, N. Codella, M. Combalia, S. Dusza, P. Guitera, D. Gutman, *et al.*, "A patient-centric dataset of images and metadata for identifying melanomas using clinical context," *Scientific data*, vol. 8, no. 1, p. 34, 2021.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[26] A. Davila, J. Colan, and Y. Hasegawa, "Comparison of fine-tuning strategies for transfer learning in medical image classification," *Image and Vision Computing*, vol. 146, p. 105012, 2024.