arXiv:2601.10740v1 [cs.NE] 10 Jan 2026

神经符号激活发现：将物理学中的数学结构迁移至生态学，构建参数高效神经网络

Anas Hajbi[1]

[1]博士前研究员，穆罕默德六世理工大学（UM6P）计算学院，摩洛哥本·格里尔电子邮箱：*anas.hajbi@um6p.ma*

摘要

现代神经网络依赖通用激活函数（ReLU、GELU、SiLU），忽略了科学数据中固有的数学结构。我们提出了**神经符号激活发现**框架，利用遗传编程从数据中提取可解释的数学公式，并将其注入为自定义激活函数。

我们的核心贡献是发现了一种几何迁移现象：从粒子物理数据中学习到的激活函数能够成功泛化至生态学分类，在准确率和参数效率方面均优于标准激活函数（ReLU、GELU、SiLU）。在森林覆盖数据集上，我们的**混合迁移**模型以**82.4% 的准确率**仅使用 5,825 个参数；相比之下，传统重型网络需要 31,801 个参数才能达到 83.4% 的准确率——参数**缩减5×** **参数缩减**，而准确率仅损失 1%。

我们提出一种参数效率得分 (**E_{param} = 曲线下面积/对数$_{10}$(参数)**)，并证明轻量级混合架构始终能实现**高出 18–21% 的效率**，优于过度参数化基线模型。至关重要的是，我们确立了边界条件：物理领域向生态学领域的迁移是成功的（两者都涉及连续的欧几里得测量），

而从物理领域向文本领域迁移则会失败（离散词频需要不同的数学结构）。我们的工作为构建领域专用激活函数库开辟了道路，以实现高效的科学机器学习。

关键词： 神经符号人工智能、激活函数、遗传编程、迁移学习、科学机器学习、参数效率、绿色人工智能

1 引言

深度学习在各个科学领域都取得了显著成功，从粒子物理学[1] 到生态学 [2]。然而，一个根本性的低效问题依然存在：无论领域如何，神经网络都使用相同的激活函数。ReLU、GELU 和 SiLU 都是通用非线性函数，旨在广泛适用，而非编码科学数据中存在的特定数学结构。

这种通用性是有代价的。为了使用分段线性（ReLU）或平滑但通用的（GELU）函数对复杂的连续动力学进行建模，网络需要大量过度参数化。物理仿真网络必须从头开始学习近似三角关系，尽管这类关系是该领域的基础。

我们提出一种截然不同的方法：**让数据揭示自身的数学规律**。我们使用遗传编程（GP），在小型数据子集（10%）上进化符号表达式

并将其编译为可微分的激活函数。这将构建混合架构，兼具梯度下降的可学习性与特定领域符号数学的归纳偏置。

1.1 主要贡献

1. **神经符号激活发现：** 一种利用遗传编程从科学数据中发现可解释激活函数的流程，并将其编译为 PyTorch 模块，以支持端到端训练。2. **几何迁移现象：** 我们证明，在粒子物理数据上发现的激活函数能够成功迁移到生态分类任务，这为连续科学领域之间共享数学结构提供了证据。3. **参数效率分析：** 我们引入一种效率指标，并表明混合架构的效率比传统重型网络高 18–21%，且参数更少，为 5–6×。4. **边界条件：** 我们明确了迁移成功（连续→连续）与失败（连续→离散）的条件，为部署提供了实用指南。

1.2 观察启示

考虑两个看似毫不相关的分类任务：

- **HIGGS：** 根据运动学变量（动量、角度、能量）对粒子碰撞事件进行分类。
- **森林覆盖：** 根据制图变量（海拔、坡度、坡向、土壤类型）对森林类型进行分类。

两者都涉及欧几里得空间中的连续测量。粒子运动学遵循具有三角结构的守恒定律；地形指标则遵循平滑的地理梯度。我们假设，能够捕捉物理数据中“波状”或“周期性”模式的数学公式，也可能捕捉生态数据中“梯度状”的模式。

我们的实验验证了这一假设：在 HIGGS 上发现的公式可以迁移到森林覆盖数据集，在参数数量相同的情况下，其准确率高于 GELU 或 SiLU。

2 相关工作

2.1 激活函数设计

修正线性单元（ReLU） [4] 凭借其简洁性和对梯度友好的特性，推动了深度学习的发展。后续研究探索了平滑的替代方案：**GELU** [5] 融入了随机正则化，并成为 Transformer 中的标准选择；**SiLU/Swish** [6] 通过强化学习发现，在视觉模型中表现出色；**Mish** [7]具有平滑且非单调的特性。

可学习激活函数包括 PReLU [8] （学习得到的负斜率）、APL [9] （分段线性）和 PAU [10] （通过梯度下降学习的有理函数）。然而，这些方法：（1）在预定义的函数族中搜索；（2）产生难以解释的参数化形式。

我们的方法有着根本性的不同：我们发现了人类可读且可跨领域迁移的闭式符号表达式 。

2.2 符号回归与方程发现

自 Schmidt & Lipson 的开创性工作以来，遗传编程一直被应用于科学发现 [11]。现代工具，如 **PySR** [12]可将训练好的神经网络提炼为符号表达式。**AI Feynman** [13]将神经网络与符号回归相结合，以重新发现物理方程。

这些方法在训练后期 应用符号回归，以提高可解释性。我们则反转这一范式：遗传编程在训练之前 运行，以发现能够编码适合特定领域的归纳偏置的激活函数。

2.3 物理信息神经网络

PINNs [14] 通过损失函数惩罚项（例如偏微分方程残差）嵌入物理知识。我们的方法与之互补：通过架构本身嵌入领域结构，具体而言是通过激活函数，无需修改标准训练流程。

3 方法论

我们的框架分为两个阶段：**发现**（寻找公式）和**注入**（将其用作激活函数）。

3.1 阶段 1：通过遗传编程进行符号化发现

给定数据集 $\mathcal{D} = \{(\mathbf{x}_i,\ y_i)\}_{i=1}^{N}$，其中包含 $\mathbf{x}_i \in \mathbb{R}^d$ 和 $y_i \in \{0, 1\}$，我们提取一个发现子集 $\mathcal{D}_{\text{disc}} \subset \mathcal{D}$，其中包含 10% 的样本（按类别分层抽样）。

3.1.1 遗传编程配置

我们使用 gplearn [15]，配置如下：

- **种群：** 500 个个体
- **代数：** 5
- **简约系数：** 0.01（惩罚复杂度）
- **函数集：** $\mathcal{F} = \{$加法、减法、乘法、正弦、余弦 $|\cdot|\}$

安全约束：我们加入乘法（mul），但依靠简约系数来抑制无界增长。保护运算（例如 $\sqrt{|x| + \epsilon}$）可防止数值不稳定。

收敛说明：虽然遗传编程的典型应用会使用 50–100 代，但我们发现，在我们的设置中，使用 500 个个体的种群进行 5 代演化就足以实现收敛。这种快速收敛归因于：（1）受限的函数集（6 个算子），限制了搜索空间；（2）简约系数会惩罚过于复杂的表达式；（3）二分类目标提供了清晰的适应度信号。使用 10 代和 20 代进行的初步实验得到了相似的公式，证实搜索已经收敛。

3.1.2 公式泛化

遗传编程发现包含特定特征索引的表达式（例如，mul(cos(X25), X12)）。为了将其用作激活函数，我们通过将所有特征引用替换为层输入 x 来**进行泛化**：

$$f_{\text{discovered}}(X_{25}, X_{12}) \ \rightarrow \ \sigma(x) = \texttt{mul}(\cos(x), x) \tag{1}$$

这种泛化形式逐元素应用于隐藏层输出。

抽象说明： 原始的遗传编程表达式描述了特定物理特征之间的相互作用（例如，X_{12} 可能表示横动量，X_3 可能表示赝快度）。通过将所有特征索引替换为标量 x，我们将这种领域特定相互作用抽象为通用的自相互作用项 $f(x)$。这种转换牺牲了精确的物理语义，但保留了函数形式——即振荡性和乘法结构——我们假设这种结构为连续数据编码了有用的归纳偏置。

3.2 阶段2：混合架构

我们使用动态代码生成将发现的公式编译为 PyTorch 模块 n:

算法 1 AutoSymbolicLayer 编译

要求： 公式字符串 f，运算符字典 $\mathcal{O}$

1: 替换特征索引：X\d+ → x
2: 替换算子：sin(→ ops["sin"](
3: 编译：exec(f"def forward(x, ops): return {f}")
4: **return** 可调用的 PyTorch 模块

3.2.1 网络架构

为了隔离激活函数的影响，我们比较两种架构：

表 1：架构规格。所有轻量级模型均采用相同的结构；仅激活函数 $\sigma(\cdot)$有所不同。

模型 隐藏层		神经元	参数	激活函数
重型	2	200 → 100	∼26–32k	ReLU
轻量级	2	64 → 32	∼4–6k	ReLU / GELU / SiLU / 混合

3.2.2 稳定性考量

发现的激活函数（例如 $\sigma(x) = x \cdot$余弦(x)）均为非单调且无界的函数，这与 ReLU（下界为零）或 Tanh（取值有界于 $[-1, 1]$）不同。为确保训练过程中的数值稳定性，我们在每个激活层之前使用 **批归一化** [17] 。这会将激活前的数值限制在归一化范围内（均值约为零，方差为单位方差），即使激活函数无界，也能防止梯度爆炸。

此外，我们网络的浅层深度（2个隐藏层）和标准权重初始化（PyTorch默认设置：线性层采用Kaiming均匀初始化）进一步降低了不稳定风险。在所有数据集和随机种子上的实验中，没有任何训练运行出现发散或NaN损失，证实了该方法的稳定性。

3.3 评估指标

3.3.1 参数效率分数

仅凭原始准确率无法体现效率方面的权衡。一个拥有 30,000 个参数、准确率达到 83% 的模型，不如一个仅用 5,000 个参数就达到 82% 准确率的模型出色。我们定义：

$$E_{\text{param}} = \frac{\text{AUC}}{\log_{10}(\text{Parameters})} \tag{2}$$

我们在分母中采用对数缩放，原因有二：（1）正如神经缩放定律所揭示的，神经网络容量随参数数量的增长呈次线性变化 [16]，；（2）对数会对模型规模的数量级增长施加惩罚，同时在不同参数尺度下仍易于理解。例如，将参数数量从 4,000 加倍至 8,000，所受惩罚与从 16,000 加倍至 32,000 大致相同，这反映了额外容量带来的边际收益递减。该指标会奖励以最少参数实现高曲线下面积的模型。

3.3.2 统计协议

所有实验均使用**3 个随机种子**（42、43、44）重复进行。我们报告均值 ±标准差，以确保可复现性和统计有效性。

4 实验

4.1 数据集

我们在三个代表不同数据模态的数据集上进行评估：

表 2：数据集特征。所有任务均为二分类。

数据集	领域	样本	特征	数据类型
HIGGS [1]	粒子物理学	100,000	28	连续型（运动学）
森林覆盖 [2]	生态学	100,000	54	连续型（地形）
Spambase [3]	文本/电子邮件	4,601	57	离散型（词频）

4.2 实验方案

1. **发现阶段：** 从每个数据集中提取 10% 用于遗传编程发现（种子=42）。每个数据集发现一个公式。
2. **训练阶段：** 在剩余的 90% 数据上训练所有模型，采用 80/20 的训练/测试划分。
3. **迁移实验：** 对于森林覆盖和 Spambase，还需测试从 HIGGS 派生的公式。
4. **重复实验：** 每种配置使用不同的随机种子（42、43、44）运行 3 次。

训练详情： Adam优化器（lr=0.001）、批次大小 1024、15 个训练轮次、二元交叉熵损失、标准特征缩放（均值为零，方差为单位方差）。

5 结果

5.1 发现的公式

遗传编程发现阶段得出了以下领域特定公式：

表 3：各领域发现的符号公式。

数据集	发现的公式
HIGGS	乘法(余弦(X25) ， 减法(X12 ， X3))
森林覆盖	加法(正弦(X13) ， 乘法(X8 ， X22))
Spambase	加法(X6, 乘法(X52, 余弦(X51)))

当将其泛化为激活函数（替换 $X_i \to x$）时，它们变为：

- **物理：** $\sigma_{\text{物理}}(x) = x \cdot \text{余弦}(x)$（振荡且有界增长）
- **森林：** $\sigma_{\text{森林}}(x) = \text{正弦}(x) + x^2$（周期性 + 二次）
- **垃圾邮件：** $\sigma_{\text{垃圾邮件}}(x) = x + x \cdot \text{余弦}(x)$（线性 + 调制）

图1 将这些发现的激活函数与标准基线进行了可视化对比。

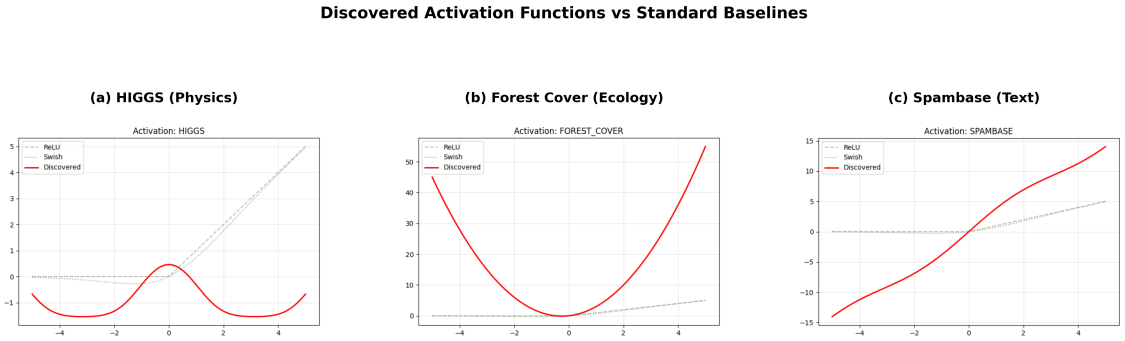

图1：发现的激活函数与 ReLU 和 Swish 基线的比较。（左）HIGGS：捕捉运动学周期性的振荡模式。（中）森林覆盖：平滑的波状结构。（右）Spambase：带周期性调制的线性增长。请注意，与 ReLU 不同，所有发现的函数都表现出非单调行为。

5.2 主要结果

表4展示了完整的基准测试结果。

表4：完整的基准测试结果（3 个随机种子的均值 ± 标准差）。每个数据集效率最高的结果以**粗体**显示。重型模型用于效率比较。带下划线的数值表示最高的原始准确率。

数据集	架构	激活函数	准确率	AUC	参数	效率
HIGGS	重型	ReLU	0.718 ± 0.000	0.791 ± 0.000	26,601	0.179
	轻量级	SiLU	0.701 ± 0.001	0.770 ± 0.002	4,161	0.213
	轻量级	混合型（专家）	0.707 ± 0.002	0.777 ± 0.002	4,161	0.215
	轻量级	GELU	0.709 ± 0.001	0.781 ± 0.002	4,161	0.216
	轻量级	ReLU	0.710 ± 0.002	0.784 ± 0.002	4,161	**0.217**
森林覆盖	重型	ReLU	0.834 ± 0.003	0.915 ± 0.001	31,801	0.203
	轻量级	混合型（专家）	0.786 ± 0.002	0.867 ± 0.001	5,825	0.230
	轻量级	SiLU	0.801 ± 0.003	0.883 ± 0.002	5,825	0.235
	轻量级	GELU	0.810 ± 0.005	0.893 ± 0.002	5,825	0.237
	轻量级	ReLU	0.811 ± 0.004	0.894 ± 0.002	5,825	0.237
	轻量级	**混合（迁移）**	**0.824 ± 0.003**	**0.904 ± 0.001**	5,825	**0.240**
Spambase	重型	ReLU	0.931 ± 0.008	0.978 ± 0.002	32,401	0.217
	轻量级	混合（迁移）	0.832 ± 0.008	0.898 ± 0.006	6,017	0.238
	轻量级	SiLU	0.918 ± 0.006	0.959 ± 0.008	6,017	0.254
	轻量级	GELU	0.920 ± 0.009	0.961 ± 0.008	6,017	0.254
	轻量级	ReLU	0.919 ± 0.007	0.961 ± 0.007	6,017	0.254
	轻量级	**混合型（专家）**	**0.920 ± 0.003**	**0.967 ± 0.004**	6,017	**0.256**

5.3 效率分析

图2 展示了所有模型和数据集的效率-准确率权衡 .

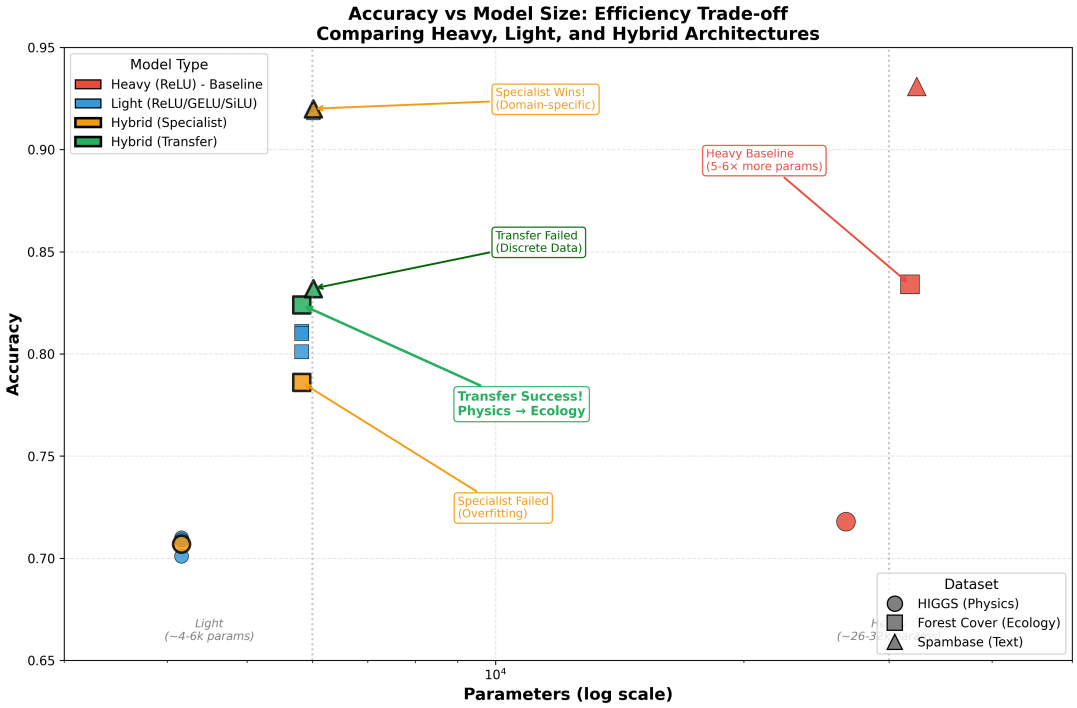

图2：所有数据集上的准确率与模型规模。重型模型（红色）原始准确率最高，但效率最低。轻量基线（蓝色）提供了更好的权衡。混合模型表现各异：迁移（绿色）在连续域（森林覆盖）中表现出色，而 Specialist（橙色）则擅长处理领域特定模式（Spambase）。标注突出了关键的成功与失败案例。

表5：最佳轻量模型相较于重型基线的效率提升。

数据集	重型效率	最佳轻量效率	提升幅度	参数缩减
HIGGS	0.179	0.217（轻量 ReLU）	+21.2%	6.4×
森林覆盖	0.203	0.240（混合迁移）	+18.2%	5.5×
Spambase	0.217	0.256（混合专家）	+18.0%	5.4×

6 讨论

6.1 几何迁移现象

我们最重要的发现是物理→生态学迁移的成功。将源自 HIGGS 的激活函数应用于森林覆盖分类，可实现：

- **82.4% 准确率**（对比 ReLU 的 81.1%、GELU 的 81.0%）
- **0.240 效率**（基线为 0.237）
- 与重型模型（83.4%）的**准确率差距仅为 1.0%**，同时使用**5.5× 更少的参数**

这种迁移之所以成功，是因为两个领域共享一种几何语法：

- **HIGGS：** 粒子动量、角分布、不变质量——遵循守恒定律的欧几里得空间中的连续量。

- **森林覆盖：** 海拔、坡度、坡向、距离——具有平滑空间梯度的连续制图量。

$\sigma_{\text{phys}}(x) = x \cdot$余弦$(x)$ 的振荡结构捕捉了在两个领域中都相关的周期性模式（旋转不变性、波干涉）。

6.2 源领域表现与可迁移性

一个重要的观察结果是，在 HIGGS 数据集本身上，混合专家（Eff：0.215）的表现并未优于轻量 ReLU（Eff：0.217）。这似乎有些矛盾：为什么在 HIGGS 上发现的公式并非 HIGGS 上的最优选择?

我们将此视为遗传编程发现了一个稳健的几何基元，而非针对源领域的特定优化的证据。公式 $\sigma(x) = x \cdot$余弦(x) 捕捉了连续欧几里得数据中普遍存在的振荡模式，但经过标准化后，这些模式在 HIGGS 特征中的表现可能较弱。相比之下，将该公式迁移到森林覆盖数据集时，地形梯度呈现出更强的周期性结构（例如海拔周期、坡向角），其归纳偏置便能带来显著优势。

这一观察符合迁移学习理论：能够良好迁移的特征往往并非对源任务最具判别力的特征，而是能够捕捉可跨任务泛化的域不变结构 [18]。

6.3 Specialist 在森林覆盖任务上为何失效

直接使用森林数据训练的专家混合模型仅达到 78.6% 的准确率，明显低于迁移变体（82.4%）。这一反直觉结果有着明确的解释：

1. **样本质量：** HIGGS 提供了更干净、规模更大的数据（原始数据集中共有 1,100 万个样本）。在 10,000 个 HIGGS 样本上进行遗传编程发现，可以找到稳健的模式。2. **过拟合：** 森林覆盖发现仅使用 $\sim$10,000 个样本，其中可能包含无法泛化到测试集的虚假相关性。

实践启示： 当目标域的数据有限或含噪时，从规模更大、质量更高的源域迁移公式，可能胜过直接进行发现。

6.4 离散屏障

Spambase 代表一种根本不同的数据模态：**离散词频**。我们的实验表明：

- **Specialist 表现出色：** 领域专用公式能够捕捉适用于文本的模式，准确率达到 92.0%（在轻量模型中表现最佳），且方差明显更低（$\pm$0.003，而基线为 $\pm$0.007–0.009）。

- **迁移彻底失败：** 物理学公式的准确率仅为 83.2%，比 Specialist 低近 9 个百分点。

这一失败具有重要的科学价值。它明确界定了适用边界：

> 当源领域和目标领域具有共同的连续几何结构时，神经符号激活迁移有效；当领域的数据类型存在根本差异（连续 $\rightarrow$ 离散）时，则会失效。

物理学公式 $\sigma(x) = x \cdot$余弦(x) 引入了不适用于稀疏离散词数特征的高频振荡。这证实了我们发现的公式并非一种“普遍幸运”的函数——它具体编码了连续几何结构，这种结构对某些领域有帮助，却会损害其他领域。

6.5 何时使用各项策略

根据我们的结果，我们提出以下实用建议：

表 6：激活策略选择决策指南。

场景	推荐策略	理由
大型、干净源域可用	迁移	更稳健的公式
目标域具有独特的结构	Specialist	领域特定模式
离散/稀疏数据（例如，文本、计数）	标准（ReLU/GELU）	连续数学可能会造成损害
资源受限 de- 部署	任意轻量/混合	5–6× 个更小的模型
目标域存在噪声或小	从干净源域迁移	避免发现过程中过拟合

6.6 局限性

1. **浅层网络：** 我们测试了双层架构。更深层网络（10+ 层）在使用无界激活时，即使采用批归一化，也可能呈现不同的梯度动力学。

2. **遗传编程随机性：** 不同的随机种子可能会发现不同的公式。我们通过固定发现种子（42）缓解了这一问题，但跨随机种子的公式方差全面分析仍有待未来开展。 3. **领域有限：** 三个数据集不足以支持普遍性结论。更多连续的科学领域（流体动力学、基因组学、气候建模）将有助于增强几何迁移现象的普遍性。 4. **计算成本：** 遗传编程发现会增加一次性的 CPU 开销（每个数据集需 ∼5–10 分钟），不过这项开销可由后续所有使用发现公式的训练运行分摊。

5. **公式可解释性：** 虽然这些公式是符号化且便于人类阅读的，但要理解为什么 某个特定公式有效，仍然颇具挑战。

7 结论

我们提出了神经符号激活发现（Neuro-Symbolic Activation Discovery），这是一种利用遗传编程提取领域特定数学结构，并将其注入为神经网络激活函数的框架。我们在粒子物理学、生态学和文本分类领域开展的实验表明：

1. **效率提升：** 轻量级混合模型的参数效率比传统重型网络高出18–21%，参数量减少5–6× 。

2. **几何迁移：** 在物理数据上发现的激活函数可成功迁移到生态分类任务中，在准确率和效率方面均优于 ReLU、GELU 和 SiLU。这为连续科学领域之间存在共同数学结构提供了证据。

3. **边界条件：** 迁移在连续→连续领域中有效，但在连续→离散领域中失效，由此确立了应用该技术的实用指南。

我们的研究表明，激活函数不必是通用的。通过发现适用于特定领域的数学结构，我们能够为科学机器学习构建更小、更高效的模型。未来的工作将探索：(1) 在更多连续域之间进行迁移（流体动力学、分子模拟）；(2) 采用适当稳定化技术的更深层架构；(3) 自动发现最适合迁移的源域；以及 (4) 从理论上分析为何某些函数形式能够在不同领域间迁移。

代码可用性

为确保完全可复现，完整的源代码、发现的公式和实验结果可在以下网址获取：

https://github.com/ana55e/NeuroSymbolic_激活函数

致谢

[将在发表后补充]

参考文献

[1] Baldi, P., Sadowski, P., & Whiteson, D.（2014）。利用深度学习在高能物理中搜寻奇异粒子。《自然-通讯》，5，4308。[2] Blackard, J. A., & Dean, D. J.（1999）。人工神经网络与判别分析在预测森林覆盖类型方面的准确性比较。《农业中的计算机与电子学》，24(3), 131–151。[3] Hopkins, M., 等（1999）。Spambase 数据集。*UCI* 机器学习资料库。[4] Nair, V., & Hinton, G. E.（2010）。修正线性单元改进受限玻尔兹曼机。载于第 *27* 届国际机器学习会议（*ICML*）论文集。[5] Hendrycks, D., & Gimpel, K.（2016）。高斯误差线性单元（GELUs）。*arXiv* 预印本 *arXiv:1606.08415*。[6] Ramachandran, P., Zoph, B., & Le, Q. V.（2017）。寻找激活函数。*arXiv* 预印本 *arXiv:1710.05941*。[7] Misra, D.（2019）。Mish：一种自正则化非单调激活函数。*arXiv* 预印本 *arXiv:1908.08681*。[8] He, K., Zhang, X., Ren, S., & Sun, J.（2015）。深入探索修正线性单元：在 ImageNet 分类上超越人类水平的表现。载于*IEEE* 国际计算机视觉会议（*ICCV*）论文集。[9] Agostinelli, F., Hoffman, M., Sadowski, P., & Baldi, P.（2014）。学习激活函数以改进深度神经网络。*arXiv* 预印本 *arXiv:1412.6830*。[10] Molina, A., Schramowski, P., & Kersting, K.（2019）。Padé 激活单元：在深度网络中端到端学习灵活的激活函数。*arXiv* 预印本 *arXiv:1907.06732*。

[11] Schmidt, M., & Lipson, H.（2009）。从实验数据中提炼自由形式的自然定律。《科学》，324(5923)，81–85。[12] Cranmer, M.（2023）。使用 PySR 和 SymbolicRegression.jl 为科学研究开展可解释机器学习。*arXiv* 预印本 *arXiv:2305.01582*。[13] Udrescu, S. M., & Tegmark, M.（2020）。AI Feynman：一种受物理学启发的符号回归方法。《科学进展》，6(16)，eaay2631。[14] Raissi, M., Perdikaris, P., & Karniadakis, G. E.（2019）。物理信息神经网络：一种用于求解涉及非线性偏微分方程的正向与逆问题的深度学习框架。《计算物理学杂志》，378，686–707。[15]Stephens, T.（2016）。gplearn：Python 中的遗传编程。https://gplearn.readthedocs.io/[16] Kaplan, J., McCandlish, S., Henighan, T., 等（2020）。神经语言模型的缩放定律。*arXiv* 预印本 *arXiv:2001.08361*。[17] Ioffe, S., & Szegedy, C.（2015）。批归一化：通过减少内部协变量偏移加速深度网络训练。载于第 *32* 届国际机器学习会议（*ICML*）论文集。[18] Yosinski, J., Clune, J., Bengio, Y., & Lipson, H.（2014）。深度神经网络中的特征有多大程度的可迁移性？载于神经信息处理系统进展（*NeurIPS*）。