arXiv:2303.02801v1 [cs.NE] 5 Mar 2023

基于神经元覆盖率指标驱动的半监督分类神经进化算法

RobertoSantana Ivan Hidalgo-Cenalmor Unai Garciarena

Alexander Mendiburu JoseAntonio Lozano

智能系统组 巴斯克地区大学（
UPV/EHU）圣塞巴斯蒂安，西班牙

{roberto .santana,unai.garciarena,alexander.mendiburu,ja.lozano}@ehu.eus

摘要

在一些机器学习应用中，用于监督分类的带标签实例数量有限，而未标注实例却十分丰富。半监督学习算法应对这类场景，并尝试利用无标签样本中包含的信息。本文探讨了如何针对半监督问题进化神经网络。我们提出了神经进化方法，通过使用基于每个候选解所编码的神经网络架构计算的神经元覆盖率指标，利用未标注实例。神经元覆盖率指标类似于用于测试软件的代码覆盖率指标，但旨在量化测试实例对不同神经网络组件的覆盖程度。在我们的神经进化方法中，我们定义了适应度函数，将基于有标签样本计算的分类准确率与使用无标签样本评估的神经元覆盖率指标相结合。我们在带标签实例数量不同的半监督问题上评估了这些函数的影响。结果表明，使用神经元覆盖率指标有助于降低神经进化对带标签数据稀缺性的敏感度，并且在某些情况下能够使所学习的分类器实现更稳健的泛化。

关键词：神经进化、神经元覆盖率、半监督学习、神经网络、深度学习、NAS

1 引言

神经进化方法 [1, 2] 广泛用于优化监督机器学习任务中的神经网络架构。通常，使用训练数据集来指导对最优模型的搜索。然而，在某些领域，由于人工标注成本高或数据稀缺，有标签样本数量有限。在这些情况下，未标记数据变得更加宝贵。人们提出了半监督机器学习算法，以便在模型的学习阶段利用无标签样本。在半监督学习方法中，我们可以找到基于分歧的方法，例如协同训练 [3],；基于图的方法，例如基于最小割的方法 [4], ；以及低密度分离方法，例如半监督支持向量机 [5]。

进化算法也已应用于半监督问题。在 [6], 中，提出了将文法进化 (GE) 应用于半监督分类的方法。模型的质量

通过结合有标签数据上的准确率与无标签样本上的一种聚类质量度量来评估。Kamalov 等人 [7] 将遗传编程算法与自标注算法结合起来，对时间序列进行分类。他们表明，这两类算法的结合提升了针对所研究分类问题的最先进半监督方法。Silva 等人 [8] 提出了一种将未标记数据纳入遗传编程学习的方法，以提高遗传编程分类器在带有噪声标签的问题上的准确率。

本文聚焦于用于半监督学习的神经进化方法。具体而言，我们研究通过进化神经网络来进行半监督分类的进化方法的行为。神经架构搜索（NAS）领域有一些研究考虑了半监督任务 [9, 10]。这些论文中的方法是先确定一项无监督前置任务，并据此学习一种神经架构，再将其迁移到有标签数据可用的目标监督任务中。例如，对于图像分割任务，首先将输入图像按四种预设方向旋转，然后搜索一种用于预测旋转方向这一前置任务的神经网络架构。随后，对选定的架构进行重新训练，以解决原始图像分割任务。本文采用一种截然不同的方法：利用未标记数据评估不同的度量指标，以描述候选解所编码的神经网络架构的神经元覆盖率。

神经元覆盖率指标 [11, 12, 13] 的灵感来自代码覆盖率指标，后者用于代码系统的功能测试和结构测试。通常，软件覆盖率指标用于量化一组测试样例对代码的不同组成部分（例如，代码行、程序分支、模块等）的“覆盖”程度。对于神经网络，相应指标所关注的覆盖对象是神经元、层，甚至是对网络功能更细粒度的刻画，例如神经元激活函数可能取值的范围。神经元覆盖率指标主要用于验证神经网络并检测可能存在的错误 [14, 15]。应用这些指标的依据是，能够覆盖神经网络所有可能组成部分的测试集，可以对网络行为进行详细刻画。

本文提出使用神经元覆盖率指标作为监督信号，以区分演化架构。我们的假设是，利用未标记数据计算这些指标，可以据此预测神经网络架构是否可能至少能够“正确评估”相似数据。因此，神经进化将由演化网络架构在有标签数据上的表现，以及它们被未标记数据覆盖的潜力共同驱动。

本文结构安排如下：

下一节将介绍半监督分类和神经元覆盖率指标的必要背景。第 3 节分析相关工作。第 4 节介绍神经进化方法，并说明进化算子以及基于 TensorFlow 的实现的特点。第 5 节讨论训练过程的特点，以及专为半监督场景设计的适应度函数。第 6 节介绍实验框架并讨论实验结果。最后，第 7 节总结本文并探讨未来工作。

2 半监督分类和神经元覆盖率指标

2.1 半监督分类

我们研究半监督分类任务，即从训练数据集 $D_{train} = \{(x^1, y^1) \ldots, (x^i, y^i) \ldots, (x^l, y^l)\ x^{l+1}, \ldots, x^m\}$ 中学习函数 $f\colon \mathcal{X} \mapsto \mathcal{Y}$，其中 $\mathcal{X}$ 是特征空间，$\mathcal{Y} \in \{0, 1\}, l$ 是有标签训练样本的数量，$u = m - l$ 是无标签实例的数量。此类问题可视为不完全监督的一个例子，因为训练数据中只有一部分带有标签 [16]。

2.2 神经元覆盖率度量指标

在相关文献中，神经元覆盖率指标的定义略有不同。我们主要采用了 [11, 13] 中使用的约定，并作了少许修改。

设 c 表示一个具有 A 层的多层感知机（MLP）中的一个神经元。 M_1、 M_2、.....、 M_A 分别表示每一层中的神经元数量，而 $N = \sum_{j=1}^{A} M_j$ 表示网络中的神经元总数。

我们使用 $\phi(x^i, c)$表示在给定 x^i 作为输入时返回神经元 c 输出的函数。对于给定神经元 c，若给定输入 x^i 满足 $\phi(x^i, c) > t$，则称该神经元被激活，其中 t 是给定阈值。 L_c 和 H_c 分别表示针对 $x^i \in D$的函数 $\phi(x^i, c)$的下界和上界。这些值通过分析训练数据集D中 $\phi(x^i, c)$的取值来确定。通常，集合 D 对应于训练数据集中的一组实例，即 $D = D_{train}$。

2.2.1 神经元覆盖率

给定一组实例 D 和给定阈值 t，神经网络覆盖率 [12] 衡量多层感知机中至少被 D内一个实例激活过的神经元所占比例:

$$NC = \frac{\left|\{c|\, \exists x^i \in D : \phi(x^i, c) > t\}\right|}{N} \tag{1}$$

2.2.2 Top-K 神经元覆盖率

对于给定测试输入 x^i 以及同一层中的神经元 c 和 c'，若 $\phi(x^i, c) > \phi(x^i, c')$，则 c 比 c' 更活跃。对于第 j层，位于第 j 层的 $top_K^j(x^i)$表示在给定 x^i时该层中具有最大 K个输出的神经元集合。

Top-K 神经元覆盖率（TKNC）衡量每一层中曾经位列最活跃 K 神经元之列的神经元数量。

$$TKNC(D_{test}, K) = \frac{|\bigcup_{x^i \in D_{test}} (\bigcup_{1 \le j \le A} top_K^j(x^i)))|}{N} \tag{2}$$

2.2.3 k-多区段神经元覆盖率

给定一个神经元 c，多区间神经元覆盖率衡量给定测试实例集合对范围 $[L_c, H_c]$的覆盖程度。该范围被划分为 $k > 0$ 个相等的区段，称为多区段。如果 $\phi(x^i, c) \in S_c^s$for $x^i \in D_{test}$，则认为多区段 S_c^s, $s \in \{1, \ldots, k\}$ 已被覆盖。

The 神经元 c 的k-多区段神经元覆盖率定义为 [11] 已覆盖数量与总数之比 f
sect由 D_{test} 和 k覆盖的部分,

$$\overline{KMN}(c) = \frac{\left|\{S_c^s|\, \exists x^i \in D_{test} : \phi(x^i, c) \in S_c^s\}\right|}{k} \tag{3}$$

多层感知机 [11] 的k-多区段神经元覆盖率定义如下:

$$KMN(D_{test}, k) = \frac{\sum_c \overline{KMN}(c)}{k \cdot N} \tag{4}$$

2.2.4 神经元边界覆盖率和强神经元激活覆盖率

如果存在一个神经元c，使得 $\phi(x^i, c)$ 低于 L_c 或高于 H_c，则称测试输入 $x^i \in D_{test}$ 位于多层感知机的边界情况区域。

为了覆盖多层感知机的边界情况区域，将已覆盖的边界情况区域集合定义为 :

$$LCN = \{c|\, \exists x^i \in D_{test} : \phi(x^i, c) \in (-\infty, L_c)\} \tag{5}$$

$$UCN = \{c|\, \exists x^i \in D_{test} : \phi(x^i, c) \in (H_c, +\infty)\} \tag{6}$$

神经元边界覆盖率（NBC）用于衡量给定测试输入集 D_{test} 覆盖了多少个边界情况区域。

$$NBC(D_{test}) = \frac{|LCN| + |UCN|}{2 \cdot N} \tag{7}$$

强神经元激活覆盖率（SNAC）衡量给定测试输入 D_{test} 覆盖了多少相对于上界值的边界情况。

$$SNAC(D_{test}) = \frac{|UCN|}{N} \tag{8}$$

3 相关工作

3.1 神经网络验证

在相关文献中，神经元覆盖率指标主要用于评估和创建神经网络验证测试实例 [11, 12, 17]。测试优先级排序是根据原始输入改进模型的潜力，对其进行排序。改进模型的一种典型方式是发现模型的意外行为，从而促进模型增强。通常，会针对每个测试样例计算所选的神经元指标，然后按覆盖量降序排列。排名靠前的测试实例具有更高的优先级。

Pei 等人 [12] 提出了神经元覆盖率的概念，并将其用于深度学习系统的白盒测试。他们报告称，与代码覆盖率相比，神经元覆盖率是衡量深度神经网络测试输入全面性的更好度量标准；而且，来自分类问题不同类别的输入通常比来自同一类别的输入激活更多不同神经元。Ma 等人 [11] 扩展了神经元和层覆盖率指标，并将其与对抗样本的生成结合起来，用于量化测试数据对深度神经网络的缺陷检测能力。在 [15], 中，神经元重要性分析被提出，用于识别在神经网络决策过程中发挥更重要作用的神经元。作者表明，所提出的度量标准能够检测出卷积网络中对给定输入相关像素变化更敏感的神经元。

Lee 等人 [14] 提出，将神经元覆盖率指标应用于一个与测试选择没有直接关系的问题。他们将这些指标作为选择神经元的依据，用于对神经网络进行基于梯度的白盒测试。这些白盒测试方法需要计算神经元的梯度，以量化其行为。由于这类计算可能代价高昂，一些作者提出了选择或确定神经元优先级的策略。此类策略包括随机选择未激活的神经元 [12], , 或识别接近激活阈值的神经元 [18]。将神经元覆盖率指标用于神经元优先级排序，丰富了现有方法，并表明神经元覆盖率指标还可用于区分或归类神经网络组件的不同表现或作用。

一些研究也对基于覆盖率的方法在测试优先级排序中的有效性提出了质疑；这些研究提出了其他基于统计的方法来评估神经网络 [19, 20]。其他作者报告称 [13] ，与基于梯度的方法相比，由覆盖率驱动的方法在发现缺陷和提升神经网络鲁棒性方面效果较差。

近期有不少研究探讨了神经元覆盖率指标是否适用于评估其他机器学习范式。例如，Trujillo 等人 [21] 对使用这些指标测试深度强化学习（DeepRL）系统开展了一项初步研究。他们计算了强化学习过程的覆盖率演化模式与奖励之间的相关性，并得出结论：神经元覆盖率不足以对深度强化学习网络的设计或结构作出实质性结论。

上述研究均未将神经元覆盖度量标准用于神经网络架构空间的搜索，也未用于寻找半监督问题的解决方案。

3.2 用于半监督问题的神经进化

有几篇论文采用进化优化技术解决半监督学习问题（例如，参见 [8]，了解其中一些方法的讨论）。我们仅简要介绍该领域的部分研究论文，并重点关注与我们的贡献有一定共同点的方法。

在 [6], Fitzgerald 等人使用文法进化（GE）处理半监督问题。他们采用一种对如果—那么规则进行编码的文法，并进化出能够根据实例特征将其分配到不同簇的程序。通过轮廓系数或轮廓分数（SC） [22], , 利用无标签实例来衡量聚类性能，而有标签数据则用于根据分类准确率衡量模型的性能。适应度函数计算为上述两个分数之和。尽管本文同样提出，将每个进化模型的适应度评估作为分别利用有标签数据和未标记数据计算的不同分数的组合，但神经元覆盖率指标与 SC 及其他聚类分数在本质上不同。它们并不与计算任何辅助任务或目标任务的性能相关。此外，我们用于进化神经网络的进化算法是一种遗传算法（GA），其基于神经网络架构的基于列表的表示进行操作。

半监督分类问题的另一种方法是自标注或重新训练：先在带标签实例上训练模型，再用该模型对无标签实例进行预测。这些预测作为无标签样本的伪标签，并用于重新训练分类器。 [7] 介绍了一种类似的方法，作者将 PageRank 和主成分分析算法与一种专为非线性符号回归定制的遗传编程（GP）变体相结合。该算法在三个时间序列数据集上进行了测试，结果显示，混合算法的性能优于单独使用这两种算法。另有一些论文提出将遗传编程应用于半监督问题，它们主要采用基于树的程序表示 [8]，并且也使用了自标注策略的变体 [23]。

越来越多的研究 [24, 25] 提出了适用于固定神经网络的半监督学习方法，即网络架构不会作为半监督方法的一部分而改变。例如， [24] 提出的一致性正则化方法会在添加人工噪声和不添加人工噪声的情况下分别评估每个数据点，然后计算两种预测之间的一致性代价。直到最近，才有研究探讨神经架构搜索中的半监督分类问题。 [10], 将两种半监督方法应用于语义分割。该算法联合优化神经网络的架构及其参数。这种方法通过最小化有监督损失与两种无监督损失的加权和来实现。与本节前面讨论的示例一样，该方法需要定义一项辅助任务（例如聚类），并根据模型在所有任务上的表现对其进行评估。

4 神经进化方法

我们所采用的神经进化方法基于遗传算法的应用，该遗传算法配备了专为处理 TensorFlow 程序列表表示而设计的遗传算子。本节将介绍算法的主要组成部分，下一节则重点讨论本文的主要贡献，即如何实现适应度函数以处理半监督学习场景。

4.1 神经网络表示

在这项工作中，经过进化的深度神经网络是标准的前馈、顺序式多层感知机，其特点如下：

- 隐藏层数量：由于我们考虑的是标准的前馈顺序式架构，因此只需使用一个整数即可对这一方面进行编码。
- 初始化函数：给定层中的权重可以采用不同的方式进行初始化，这会影响网络达到的局部最优解。它由一个初始化函数索引列表组成。

• 激活函数：同样，每一层应用的激活函数也不是固定的。与初始化函数类似，它由一个索引列表组成。
• 丢弃：这也是一种逐层特征，通过一个布尔元素列表实现，用于确定是否在激活函数之后应用丢弃。

• 批归一化：与前一项类似，它由一个布尔元素列表组成，用于指示每一层是否在激活函数之前执行批归一化。

可进化组件具有多种变异选项：

• 深度神经网络权重可以通过从正态分布或均匀分布中抽取数值进行初始化，也可以采用正态初始化的 Xavier [26] 变体。
• 深度神经网络的各层可以采用以下激活函数：恒等函数、ReLU、ELU、Softplus、Softsign、Sigmoid、双曲正切。

4.2 使用基于列表的编码的遗传算法

由于深度神经网络的参数采用列表编码，我们定义了一种基于列表的深度神经网络描述符，用于指定网络架构以及其他参数，例如损失函数、权重初始化函数等。这可以视为一种声明式表示，因为它仅包含网络的规格说明，而权重则不纳入进化过程。算法 1 展示了遗传算法的伪代码。

算法 1：用于进化深度神经网络的遗传算法。

1 设定 $t \Leftarrow 0$。通过生成 N 个随机深度神经网络描述，创建一个种群 D_0；2 当未满足终止条件时，执行以下操作3 使用适应度函数评估 D_t；4 根据选择方法，从 D_t 中选出由 $Q \leq N$ 个解组成的种群 D_t^S；5 以概率 $p_m = 1 - p_x$ 对 D_t^S进行变异，并创建子代集合 O_t。变异算子的选择采用均匀随机方式；6 使用选择方法从 $\{D_t, O_t\}$ 中创建 D_{t+1}；7 $t \Leftarrow t + 1$；8 结束

4.3 遗传算子

用于变异个体的算子如下：

• 层变更算子会随机重新初始化随机选定层的描述，例如其权重初始化方式、激活函数以及神经元数量。
• 添加层算子会在深度神经网络的随机位置引入一个新的（随机初始化的）层。

• 删除层算子会删除深度神经网络中随机选定的层。
• 激活函数变更算子会将随机层的激活函数更改为另一个随机选定的函数。

• 权重变更算子与激活变更类似，会改变用于获取深度神经网络层初始权重的函数。

4.4 实现

我们使用了作为 [15] 中所述工作的一部分而开发的神经元覆盖率指标的实现[1] 。为实现基于神经元覆盖率的神经进化方法，我们使用了

[1]可从 https://github.com/DeepImportance/deepimportance_代码_发布版本获取

deatf 库 [2]，它是 Evoflow 库 [3] [27], 的 TensorFlow 2 扩展。Evoflow 最初旨在对以 tensorflow[28] 实现的神经网络进行进化。
在这些库中，深度神经网络的表示分为两种类别：深度神经网络架构的表示包含在 NetworkDescriptor 中，其中涵盖了第 4.1 节提到的所有列表。架构由 TensorFlow 深度神经网络实现，并包含在 Network 类中。我们使用专为处理多层感知机而设计的 MLPDescriptor 和 MLP 类。
通过 DEAP 库，EvoFlow 提供了多种选择算子 [29]。我们采用截断选择策略。

5 网络训练与适应度评估

在将神经进化方法应用于监督分类问题时，原始数据集 D 通常会被划分为三部分 $D = D_{train} \cup D_{val} \cup D_{test}$。$D_{train}$ 用于训练网络，D_{val} 用于通过计算适应度函数来估计已训练网络的性能，而 D_{test} 仅在进化结束时用于评估算法找到的最佳网络的质量。我们考虑采用这种数据划分方式的神经进化场景。请注意，也可以使用另一个验证集来对神经网络训练过程进行早停。不过，我们不考虑这种早停策略。

在我们针对半监督问题的方法中，我们假设 D_{val} 和 D_{test} 集合将保留所有标签。它们将分别以常规方式用于在进化过程中以及进化结束时评估模型的准确率。D_{train} 中将有 $q \in [0, 1)$ 比例的无标签实例，以及 $1 - q$ 比例的带标签实例。这两个集合将分别命名为 D^l_{train} 和 D^u_{train}。

5.1 全监督情形的适应度评估

在评估过程中，可以计算一个或多个用于描述深度神经网络性能的度量指标（例如，准确率、学习时间等）。对于本文讨论的二分类问题，我们使用平衡准确率指标 [30]，它适用于处理非平衡分类问题。

$$b_acc(z, D_{val}) = \frac{1}{2}(\frac{TP}{P} + \frac{TN}{N}) \tag{9}$$

其中，z 是正在评估的神经网络，P 和 N 分别是 D_{val}中的正类实例数和负类实例数，TP和 TN 分别是模型对 D_{val}中的实例所作预测中正确的正类预测数和负类预测数。

当 $q = 0$ 时，我们处于全监督情形，所有训练数据均已标注。在这种情况下，对于候选神经网络 z，适应度函数 $f(z)$ 就是平衡准确率$b_acc(z, D_{val})$。

5.2 半监督情形的适应度评估

当 $q \in (0, 1)$ 且 $D^u_{train} \neq \emptyset$ 时，适应度评估还会考虑使用未标记样本计算的神经网络覆盖率 $NNCov(z, D^u_{train})$，其中 $NNCov$ 可以是第 2.2 节介绍的以下神经元覆盖率指标之一：NC、$TKNC$、KMN、NBC 和 $SNAC$。

最后，半监督情形的适应度函数定义如下：

$$f(z) = q \cdot NNCov(z, D^u_{train}) + (1 - q) \cdot b_acc(z, D_{val}) \tag{10}$$

[2]可从 https://github.com/IvanHCenalmor/deatf 获取[3]可从 https://github.com/unaigarciarena/EvoFlow 获取

5.3 采用神经进化方法进行半监督分类的基线方法

为了评估所提出算法的性能，我们实现了两种借鉴该领域半监督方法的方法。

第一个基线方法基于不确定性量化指标 [20]，其旨在衡量模型在预测给定实例所属类别时的不确定性。对于实例优先级排序而言，模型以低置信度分类的示例（即对于二分类问题，预测概率接近 0.5）尤其值得关注，因为它们可能代表问题案例。在 [20], 中，研究者认为，简单度量指标在测试实例优先级排序方面可能比神经元覆盖率指标更有效。因此，确定这些度量指标在神经进化优化背景下的表现是一个值得研究的问题。

在评估网络架构时，我们假设，当神经网络能够以高度确定性预测无标签实例的类别时，该网络更具潜力。我们将 $CERT$ 度量指标定义为:

$$CERT(z, D_{test}) = \frac{\sum x^i \in D_{test} max(p(x^i), 1 - p(x^i))}{|D_{test}|} \tag{11}$$

其中， $p(x^i)$表示神经网络 z所赋予的 x^i 属于类别 1 的概率 .

由于只需计算模型对 D^u_{train} 中所有实例的预测，因此该度量标准的计算成本并不高。

第二个基线采用重新训练，这是半监督学习文献中经常描述的一种方法 [23, 7]。重新训练包括使用基于带标签实例学习得到的模型，对无标签实例进行预测。随后，使用模型预测的伪标签对模型进行重新训练。在某些情况下，需要置信度值，以便判定某个伪标签是否有效并可用于重新训练。

在我们的基线中，我们将阈值设为 $p(x^i) \leq 0.4$，以判定类别 0 伪标签是否正确。同样地，我们将 $p(x^i) \geq 0.6$ 设为判定类别 1 伪标签是否正确的条件。该算法首先使用 D^l_{train}学习模型，然后对 D^u_{train}中的实例进行预测。随后，选出满足预测阈值的无标签实例，并将其与 D^l_{train} 中的实例合并，以重新训练模型。最后，适应度函数是在 D_{val}上计算的重新训练模型的平衡准确率。如果没有无标签实例满足所预测概率的约束条件而被选中，则不进行重新训练，与该网络架构对应的适应度值就是由使用 D^l_{train}训练的网络生成的、针对 D_{val} 的平衡准确率。

我们将重新训练方法记为 RET，它的计算成本可能很高，因为需要进行两次学习过程。

6 实验

实验的总体目标是确定神经元覆盖率度量指标的使用是否会影响针对半监督分类问题的神经网络架构进化搜索，以及影响程度如何。具体而言，我们探讨以下问题:

- 随着无标签实例数量的变化，进化分类器的质量会如何下降?
- 所研究的覆盖度量指标中，哪一种对演化架构的性能产生的效果更有益?
- 基于神经元覆盖率的神经进化方法与基于不确定性量化和重新训练的方法相比如何?

我们首先介绍实验设计，包括基准测试的特征。随后，我们针对上述问题展开讨论，并给出实验的数值结果。

数据集	n_个观测值	n_个特征	不平衡度
agaricus_lepiota	8145	22	0.00
analcatdata_lawsuit	264	4	0.73
澳大利亚	690	14	0.01
背痛	180	32	0.52
生物医学	209	8	0.08
乳房	699	10	0.10
乳腺癌	286	9	0.16
乳腺癌_W_	569	30	0.06
breast_w	699	9	0.10
buggyCrx	690	15	0.01
bupa	345	5	0.00
国际象棋	3196	36	0.00
客户流失	5000	20	0.51
cleve	303	13	0.01
coil2000	9822	85	0.78
疝气	368	22	0.07
信用_A_	690	15	0.01
信用_G_	1000	20	0.16
crx	690	15	0.01
糖尿病	768	8	0.09

表 1：二分类问题的数据集。

6.1 实验设计

为构建分类问题基准，我们采用 [6], 中描述的方法：使用完全标注数据集，并通过仅将训练数据中的随机子集视为有标签数据来模拟部分标注；数据集中的其余数据则视为未标注。从一个已知所有标签的二分类问题出发，我们将移除数据集中一定比例 q 的实例的标签，以模拟半监督场景。我们将使用不同的 q 值来研究缺失标签数量的影响。

我们选取了 PMLB 库 [31, 32] 中包含的 20 个二分类问题。每个问题都有一个相应的数据集，其特征见表 1。表中，不平衡度表示两个类别对应的观测值数量的不平衡程度。计算方法是测量数据集中每个类别的发生比例与完全平衡状态之间的平方距离 [31]。

每项实验都通过运行神经进化算法来解决一个特定的二分类问题。我们将种群规模设为 20 个个体，代数设为 30。神经网络架构的深度限制为最多 8 层，每层的神经元数量也最多设为 8 个。批次大小和训练轮数分别设为 10 和 50。

神经进化算法的每种可能应用均由以下参数确定：

1. 所解决的分类问题。 2. D_{train}、 $q \in \{0, 0.2, 0.4, 0.6, 0.8\}$ 中未标记数据的比例。

3. 使用的神经元覆盖率指标类型（仅当 $q > 0$ 时），即 NC、 $TKNC$、 KMN、NBC 和 $SNAC$。

对于每种可能的配置，我们都执行了 10 次神经进化搜索重复实验，总计进行了 $20 \times 10 \times (4 \times 5 + 1) = 4200$ 次实验。请注意，当 $q = 0$ 时，五种覆盖度量指标均未使用。

对于每种算法，在进化结束后，我们使用数据集 $D_{train} \cup D_{val}$ 重新训练最后一代所有个体所编码的架构。随后，我们对D_{test}进行预测，并使用这些预测计算平衡准确率。

6.2 数值结果

6.2.1 初步实验

为初步评估 q 对不同算法性能的影响，图1和图2展示了两个分类问题（analcatdata和乳腺癌）的平衡准确率分布。

该分布基于全部10次运行中最后一代的20个架构计算得出。在图中，为便于比较各算法，全监督问题（$q = 0$）的结果针对每种神经元覆盖率指标各展示一次，共展示五次。

这两个问题展示了进化分类器性能的不同情形。analcatdata问题易于解决，并且对于大多数配置，准确率结果都很高。这也是一个例子，其中当未标记数据的比例增加时，分类器的性能并不会受到太大影响。对于该问题，在神经元覆盖率指标中，当 q 变化时，NC表现出更稳定的行为。该示例表明，至少对于某些问题，将覆盖度量与大量未标记数据结合使用，有助于获得高性能分类器。

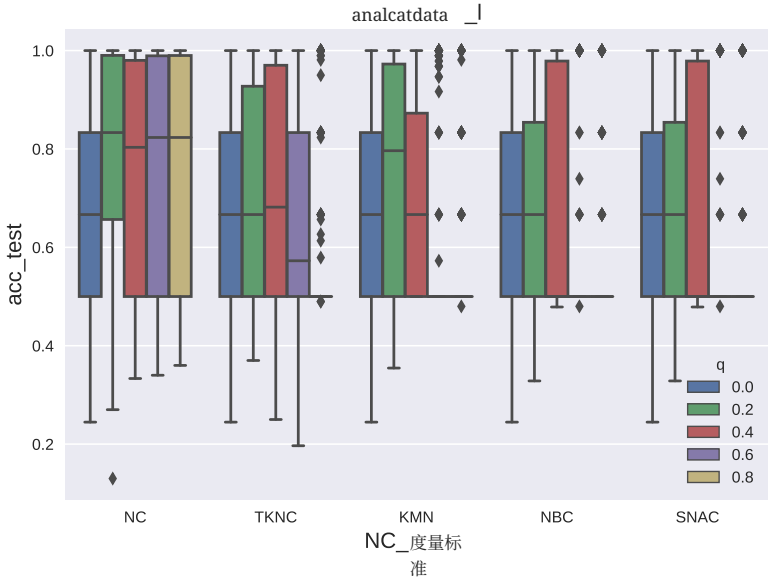


图1：analcatdata问题的结果。

图2展示了一个问题，在该问题中，进化分类器的质量相当差，所有准确率值均低于0.7。值得注意的是，在这个问题中，随着未标记数据量的增加，使用 KMN、 NBC和 $SNAC$ 覆盖度量进化得到的架构，其结果显著变差。正如接下来的实验将会展示的，这也是其他问题中普遍存在的趋势。

还值得注意的是，在某些情况下，例如与 analcatdata数据库相关的多个实例，以及乳腺癌问题中带有 $q = 20$ 和 $q = 40$ 的实例，使用神经元覆盖率度量指标演化得到的模型，其平衡准确率高于对数据库进行完整标注所得到的平衡准确率。

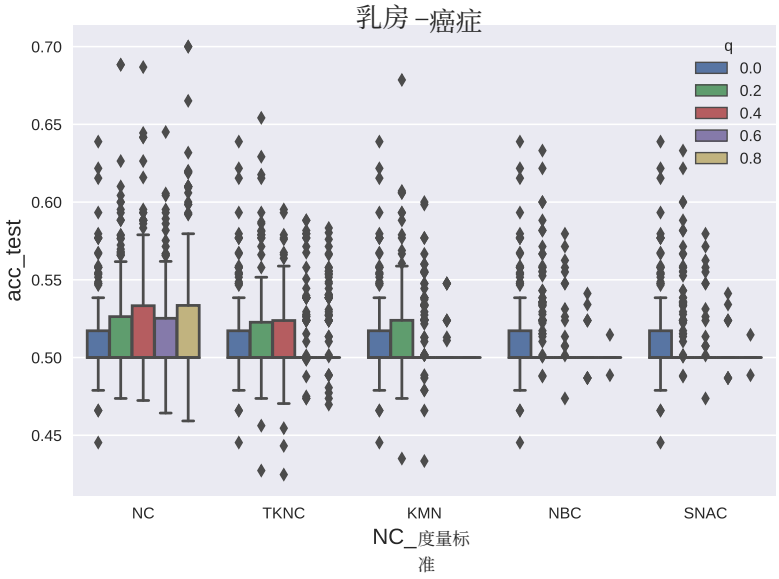


图2：乳腺癌问题的结果。

6.2.2 在所有数据集上评估算法

图3展示了针对每个数据集和配置所找到的最佳分类器在测试集上的平衡准确率。对图中内容的分析表明，就找到的最佳解而言，未标记数据的数量似乎不会对最佳分类器的准确率产生显著影响。然而，由于各数据集的问题难度存在差异，这些差别很难辨识。因此，我们通过计算20个问题的平均平衡准确率来概括图3所包含的信息。相关信息见图4。

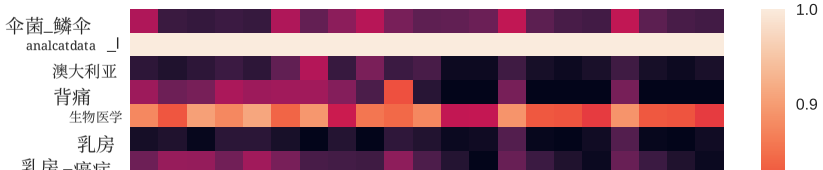
如图4所示，对于三种度量指标而言，使用较少的有标签数据训练网络会对准确率产生明显影响。对于度量指标 KMN、NBC 和 $SNAC$，随着 q 增加，平均准确率会下降。神经元覆盖率指标 NC 和 $TKNC$ 则表现得更加稳定，并证明它们能够引导搜索找到至少与使用完整有标签数据集训练出的分类器同样优秀的分类器。

6.3 与基线算法的比较

本节将所提出算法的性能与第5.3节介绍的两种方法进行比较。我们重点比较使用 NC 和 $TKNC$的算法，因为它们在之前的实验中取得了最佳结果。在比较中，所有神经网络和进化算法的设置均与之前的实验相同。算法之间唯一的区别在于适应度函数的计算方式。

表 2 总结了各算法在 20 个问题中的 10 个问题上的比较结果（受篇幅限制）。对于 agaricus lepiota 和 coil2000 数据集，重新训练变体在运行 15 小时后仍未完成。另一方面，分析中未包含 analcatdat l 和 breast cancer wand 数据集，因为在所有配置和运行中，都找到了在测试数据上准确率达到完美的分类器。

对于每个数据集、算法和未标记数据比例，我们计算了 10 次实验中最后一代种群最佳解的平均准确率。在表 2 中，每个数据集和 q 对应的最佳结果所用算法均以下划线标出。可以清楚地看出，使用 NC 和 TKNC 神经元覆盖率指标取得了最佳结果。不确定性量化证明了


伞菌_鳞伞
analcatdata _l
澳大利亚
背痛
生物医学
乳房
乳房_癌症
1.0
0.9

		NC				TKNC				CERT				RET			
工业数据库 / q	0	0.20	0.40	0.60	0.80	0.20	0.40	0.60	0.80	0.20	0.40	0.60	0.80	0.20	0.41	0.60	0.80
澳大利亚	0.56	0.54	0.56	0.58	0.56	0.63	0.72	0.57	0.65	0.57	0.62	0.54	0.54	0.57	0.57	0.73	0.57
背痛	0.70	0.64	0.65	0.71	0.70	0.70	0.70	0.67	0.60	0.59	0.66	0.60	0.65	0.67	0.59	0.58	0.67
生物医学	0.88	0.82	0.90	0.88	0.91	0.84	0.89	0.75	0.86	0.80	0.70	0.75	0.75	0.85	0.88	0.80	0.86
乳房	0.53	0.54	0.51	0.56	0.56	0.53	0.50	0.55	0.50	0.51	0.50	0.50	0.51	0.51	0.51	0.51	0.52
乳腺癌	0.64	0.69	0.69	0.65	0.70	0.65	0.60	0.59	0.58	0.67	0.63	0.60	0.55	0.69	0.60	0.68	0.68
breast w	0.81	0.87	0.89	0.84	0.84	0.96	0.86	0.89	0.82	0.85	0.78	0.78	0.70	0.81	0.87	0.86	0.90
buggyCrx	0.54	0.58	0.54	0.55	0.54	0.55	0.56	0.64	0.55	0.55	0.52	0.52	0.52	0.54	0.55	0.54	0.56
bupa	0.56	0.60	0.64	0.65	0.62	0.58	0.59	0.62	0.57	0.59	0.57	0.56	0.57	0.61	0.61	0.59	0.60
国际象棋	0.72	0.77	0.69	0.73	0.67	0.72	0.73	0.75	0.71	0.70	0.70	0.64	0.66	0.74	0.70	0.71	0.71
客户流失	0.60	0.56	0.56	0.63	0.58	0.59	0.60	0.60	0.61	0.56	0.59	0.58	0.59	0.54	0.59	0.54	0.56

表 2：基于神经元覆盖率指标的神经进化变体与采用 CERT 指标和重新训练（RET）的基线方法的比较。

并不是引导神经进化的有效竞争方案。重新训练能够取得比 CERT 更好的结果，但计算成本更高。

我们还研究了神经进化算法在进化过程中的动态变化。图 5 展示了一个示例：随着评估次数从第 1 次增加到第 620 次，最佳解的准确率（在验证数据 D_{val} 上计算）如何演变。图 5 所示的示例对应于国际象棋问题，且 $q = 0.8$。平衡准确率值是 10 次运行的平均值。

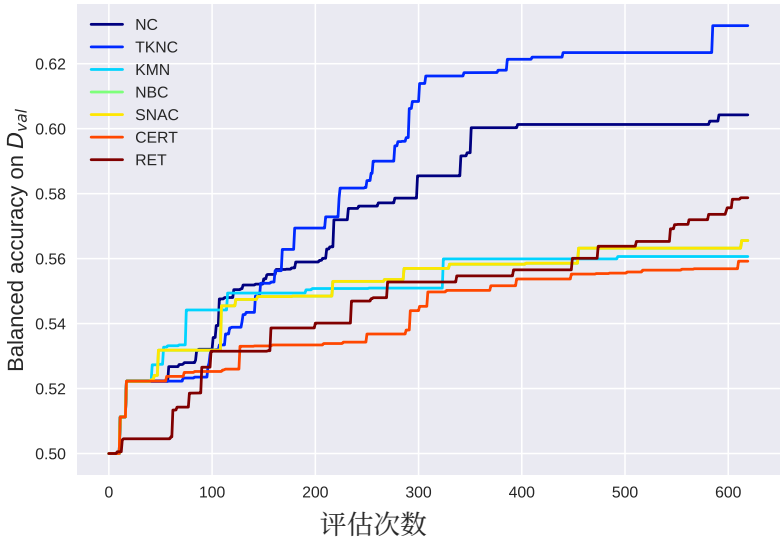


图 5：国际象棋数据集和 $q = 0.8$ 在验证数据上的准确率演变 .

图 5 中的曲线表明，所有适应度函数都能够引导搜索进入性能更优的神经网络架构区域。动态变化的差异体现在所找到解的质量和收敛速度上。对于国际象棋问题，使用 NC 和 TKNC 的神经进化算法能够更快地收敛到准确率更高的架构。请注意，最大化模型在 D_{val} 上的准确率并不能保证得到能够泛化到其他数据（例如 D_{test}）的架构，但该示例说明，除了能够生成如表 2 所示的更优架构外，收敛速度也可以更快。

7 结论

在许多领域中，难以获取或获取成本高昂的有标签实例所对应的半监督问题十分常见。当将神经进化方法应用于这些问题时，便会产生这样一个问题：如何

利用未标记数据来改进分类器的搜索。本文提出使用神经元覆盖率指标来评估每个候选架构的潜力。其隐含假设是，被无标签样本更充分覆盖的架构更具潜力。我们评估了五种不同的神经元覆盖率指标，并发现就有标签实例数量减少时结果的退化程度而言，NC 和 TKNC 指标更加稳定。我们的结果还表明，对于某些问题，使用这些指标甚至可以提升神经进化搜索的性能。

7.1 未来工作

本文提出的工作可通过多种方式扩展，以处理其他类别的半监督问题。一个必要步骤是超越二分类问题，转而解决多分类问题。本文提出的适应度函数也可用于多分类问题。另一个研究方向是制定策略，以处理具有不完整类别分配的多标签问题学习 [33]。尽管这些问题的覆盖度量计算方式不会改变，但应修改适应度函数，以考-虑多个分类问题的存在。还应对演化架构进行分析，以确定神经元覆盖率指标的使用是否会对最佳解中包含的神经网络组件类型（例如激活函数的类型）引入任何偏差。

尽管我们重点关注半监督问题，但神经元覆盖率指标也可用于其他应用神经进化的场景。它们可作为一种额外的正则-化机制，优先选择由问题输入完全覆盖的架构。在单目标或多目标场景中，对于大量候选神经网络架构具有相同被优化目-标函数值的问题，它们可被用作一种多样化机制。最后，基于神经元覆盖率指标的神经进化可与自适应实例选择相结合，用于对将部署于机器学习系统中的神经网络进行早期验证（和纠正）。

参考文献

[1] Edgar Galv´an 和 Peter Mooney。深度神经网络中的神经进化：当前趋势与未来挑战。IEEE 人工智能汇刊，2(6):476–493，2021年。[2] Kenneth O Stanley、Jeff Clune、Joel Lehman 和 Risto Miikkulainen。通过神经进化设计神经网络。《自然·机器智能》，1(1):24–35，2019。[3] Avrim Blum 和 Tom Mitchell。通过协同训练结合有标签数据与未标记数据。载于第十一届计算学习理论年会论文集，第 92–100 页。美国计算机协会，1998。[4] Avrim Blum 和 Shuchi Chawla。利用图最小割从有标签数据和未标记数据中学习。载于第十八届国际机器学习会议论文集（ICML-2001），第 19–26 页，美国加利福尼亚州旧金山，2001。Morgan Kaufmann 出版社。[5] Kristin Bennett 和 Ayhan Demiriz。半监督支持向量机。《神经信息处理系统进展》，11，1998。[6] Jeannie M Fitzgerald、R Muhammad Atif Azad 和 Conor Ryan。Geml：利用文法进化进行多类别分类的进化式无监督与半监督学习。载于2015年第七届国际计算智能联合会议（IJCCI），第 1 卷，第 83–94 页。IEEE，2015。[7] Mikhail Kamalov、Aur´elie Boisbunon、Carlo Fanara、Ingrid Grenet 和 Jonathan Daeden。Pazoe：利用少量标签对时间序列进行分类。载于2021年第29届欧洲信号处理会议（EUSIPCO），第 1561–1565 页。IEEE，2021年。[8] Sara Silva、Leonardo Vanneschi、Ana IR Cabral 和 Maria J Vasconcelos。一种用于处理噪声标签和隐性过拟合的半监督遗传编程方法。《群体与进化计算》，39:323–338，2018。[9] Chenxi Liu、Piotr Doll´ar、何恺明、Ross Girshick、Alan Yuille 和谢赛宁。神经架构搜索是否需要标签？载于《计算机视觉——ECCV 2020：第16届欧洲

会议，英国格拉斯哥，2020年8月23–28日，论文集，第 IV 部分 16，第 798–813 页。Springer，2020年。[10] Lo¨ıc Pauletto、Massih-Reza Amini 和 Nicolas Winckler。Se2nas：用于语义分割的自半监督架构优化。载于2022年第26届国际模式识别会议（ICPR），第 54–60 页。IEEE，2022年。[11] Lei Ma、Felix Juefei-Xu、Fuyuan Zhang、Jiyuan Sun、Minhui Xue、Bo Li、Chunyang Chen、Ting Su、Li Li、Yang Liu 等。DeepGauge：深度学习系统的多粒度测试标准。载于第33届美国计算机协会/IEEE自动化软件工程国际会议论文集，第 120–131 页，2018。[12] Kexin Pei、Yinzhi Cao、Junfeng Yang 和 Suman Jana。DeepXplore：深度学习系统的自动化白盒测试。载于第26届操作系统原理研讨会论文集，第 1–18 页，2017。[13] Zhou Yang、Jieke Shi、Muhammad Hilmi Asyrofi 和 David Lo。重新审视神经元覆盖率度量指标及深度神经网络质量。载于2022年IEEE软件分析、演化与再工程国际会议（SANER），第 408–419 页。IEEE，2022年。[14] Seokhyun Lee、Sooyoung Cha、Dain Lee 和 Hakjoo Oh。采用自适应神经元选择策略对深度神经网络进行有效的白盒测试。载于第29届美国计算机协会 SIGSOFT 软件测试与分析国际研讨会论文集，第 165–176 页，2020。[15] Simos Gerasimou、Hasan Ferit Eniser、Alper Sen 和 Alper Cakan。基于重要性的深度学习系统测试。载于美国计算机协会/IEEE第42届软件工程国际会议论文集，第 702–713 页，2020。[16] Jer´onimo Hern´andez-Gonz´alez、Inaki Inza 和 Jose A Lozano。弱监督及其他非标准分类问题：分类法。《模式识别快报》，69:49–55，2016。综述/弱监督及其他非标准分类问题：分类法.pdf。[17] Shenao Yan、Guanhong Tao、Xuwei Liu、Juan Zhai、Shiqing Ma、Lei Xu 和 Xiangyu Zhang。深度神经网络模型覆盖率标准与模型质量之间的相关性。载于第28届美国计算机协会欧洲软件工程会议与软件工程基础研讨会联合会议论文集，第 775–787 页，2020。[18] Jianmin Guo、Yu Jiang、Yue Zhao、Quan Chen 和 Jiaguang Sun。Dlfuzz：深度学习系统的差分模糊测试。载于2018年第26届美国计算机协会欧洲软件工程会议与软件工程基础研讨会联合会议论文集，第 739–743 页，2018。[19] Yang Feng、Qingkai Shi、Xinyu Gao、Jun Wan、Chunrong Fang 和 Zhenyu Chen。DeepGini：通过对海量测试进行优先级排序来增强深度神经网络的鲁棒性。载于第29届美国计算机协会 SIGSOFT 软件测试与分析国际研讨会论文集，第 177–188 页，2020。[20] Michael Weiss 和 Paolo Tonella。简单技术在神经网络测试优先级排序和主动学习中表现出乎意料地出色（可复现性研究）。载于第31届美国计算机协会 SIGSOFT 软件测试与分析国际研讨会论文集，第 139–150 页，2022。[21] Miller Trujillo、Mario Linares-V´asquez、Camilo Escobar-Vel´asquez、Ivana Dusparic 和 Nicol´as Cardozo。神经元覆盖率对深度强化学习是否重要？一项初步研究。载于IEEE/美国计算机协会第42届软件工程国际会议研讨会论文集，第 215–220 页，2020。[22] Peter J Rousseeuw。轮廓系数：用于解释和验证聚类分析的图形辅助工具。《计算与应用数学杂志》，20:53–65，1987。[23] Filipe de Lima Arcanjo、Gisele Lobo Pappa、Paulo Viana Bicalho、Wagner Meira Jr 和 Altigran Soares da Silva。用于分类的半监督遗传编程。载于第13届遗传与进化计算年会论文集，第 1259–1266 页，2011。[24] Antti Rasmus、Mathias Berglund、Mikko Honkala、Harri Valpola 和 Tapani Raiko。利用阶梯网络进行半监督学习。《神经信息处理系统进展》，28，2015。

[25] Antti Tarvainen 和 Harri Valpola。《平均教师是更好的榜样：权重平均一致性目标提升半监督深度学习的效果》。神经信息处理系统进展，第30卷，2017年。[26] Xavier Glorot 和 Yoshua Bengio。《理解深度前馈神经网络训练的难点》。载于第十三届人工智能与统计国际会议论文集，第249–256页，2010年。[27] Unai Garciarena、Roberto Santana 和 Alexander Mendiburu。《EvoFlow：一个用于进化 TensorFlow 深度神经网络架构的 Python 库》。载于2020年IEEE计算智能研讨会系列（SSCI-2020）论文集，第2288–2295页。IEEE，2018年。[28] Mart´ín Abadi、Paul Barham、Jianmin Chen、Zhifeng Chen、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Geoffrey Irving、Michael Isard 等。《TensorFlow：一个面向大规模机器学习的系统》。载于操作系统设计与实现研讨会，第16卷，第265–283页。美国佐治亚州萨凡纳，2016年。[29] F´élix-Antoine Fortin、De Rainville、Marc-Andr´é Gardner Gardner、Marc Parizeau、Christian Gagn´é 等。《DEAP：让进化算法变得简单》。机器学习研究杂志，13(1)：2171–2175，2012年。[30] Kay Henning Brodersen、Cheng Soon Ong、Klaas Enno Stephan 和 Joachim M Buhmann。《平衡准确率及其后验分布》。载于第二十届国际模式识别会议论文集，第3121–3124页。IEEE，2010年。[31] Randal S. Olson、William La Cava、Patryk Orzechowski、Ryan J. Urbanowicz 和 Jason H. Moore。《PMLB：用于机器学习评估与比较的大型基准测试套件》。生物数据挖掘，10(36)：1–13，2017年12月。[32] Joseph D Romano、Trang T Le、William La Cava、John T Gregg、Daniel J Goldberg、Praneel Chakraborty、Natasha L Ray、Daniel Himmelstein、Weixuan Fu 和 Jason H Moore。《PMLB v1.0：用于机器学习方法基准测试的开源数据集集合》。arXiv预印本 arXiv:2012.00058v2，2021年。[33] Serhat Selcuk Bucak、Rong Jin 和 Anil K Jain。《不完整类别分配下的多标签学习》。载于2011年IEEE计算机视觉与模式识别会议论文集，第2801–2808页。IEEE，2011年。