黑箱，白噪声：神经函数的相似性检测

FarimaFarmahinifarahani[a] 和 Cristina V. Lopes[a]a

美国加利福尼亚大学欧文分校

摘要 相似性（或克隆）检测在侵犯版权、软件盗窃、代码搜索和恶意组件检测等方面有着重要应用。目前，已有相当数量的开源和专有克隆检测器，可用于检测传统编程语言编写的程序。然而，深度学习模型在软件中的应用日益广泛，这给这些工具带来了挑战：这类模型实现的函数是难以窥探的黑箱。随着越来越多的软件包含这些深度神经网络函数，我们需要新的技术来评估软件中深度学习组件之间的相似性。

以往研究提出了多种技术，通过向深度神经网络模型输入规范输入，比较模型各层学习到的表征。我们的目标是在规范输入不可用时也能比较深度神经网络函数——因为在许多应用场景中，规范输入可能并不存在。因此，面临的挑战是生成合适的输入，并确定一种度量指标，使其能够根据这些输入表示两个可比较的深度神经网络函数之间的功能相似性程度。

我们的方法使用取值介于 −1 和1之间、形状符合深度神经网络模型预期的随机输入。然后，我们通过进行相关性分析来比较输出。

我们的研究表明，即使缺乏有意义的规范输入，也可以进行相似度分析。两个可比较的深度神经网络函数对随机输入的响应，能够揭示它们之间的相似度，或揭示它们并不相似。在尝试的所有度量指标中，我们发现斯皮尔曼等级相关系数最为强大且用途广泛，不过在特定情况下，其他方法和度量指标能更充分地表达差异。

我们使用从 GitHub 收集的56,355 个分类器组成的数据集，系统地实证研究并比较了多种相似性度量的有效性。此外，我们还进行了敏感性分析，揭示了模型训练相关属性如何影响这些相似性度量的有效性。

据我们所知，这是首项展示如何使用随机输入检测深度神经网络函数相似度的研究。我们对相关性度量指标的研究，以及对斯皮尔曼相关系数作为其中最适用于此目的、效力最强的指标的确认，确立了一种完整且实用的深度神经网络克隆检测方法，可用于设计新工具。对于那些在遇到深度神经网络组件时方法会失效的其他程序分析任务，本研究也可带来启发。

ACM CCS 2012

- **软件及其工程 → 软件逆向工程**；**软件维护工具**；

关键词 神经网络、相似性检测、典型相关分析、斯皮尔曼

《编程的艺术、科学与工程》

投稿 2022年9月30日**发表**
2023年2月15日

数字对象标识符 10.22152/programming-journal.org/2023/7/12



© Farima Farmahinifarahani 和 Cristina V. Lopes 本作品采用“知识共享署名 4.0 国际许可协议”授权。载于《编程的艺术、科学与工程》，第7卷第3期，2023年，第12篇文章；共30页。

1 引言

越来越多的开源和专有软件系统都包含“AI 组件”，这些组件由实现特定输入/输出函数的深度神经网络（DNN）模型构成 [4, 7, 18, 30, 38, 40, 41]。这些深度神经网络函数是难以理解的黑箱，因为它们是由数字矩阵构成的：其结构无法揭示任何有关其输入/输出行为的信息（见图 1）。即使能够访问模型文件并检查其架构和学习到的权重，也无法通过检查这些信息来了解模型的函数。这给软件系统的生产和维护带来了新的挑战 [1],，因此需要扩展传统的软件工程实践和工具，采用适用于神经网络组件的新技术。

代码中的相似性识别，即代码克隆检测，在过去几十年里一直是软件工程领域关注的重点 [9, 31, 32]。然而，传统的代码克隆检测技术和工具并不适用于深度神经网络函数。深度神经网络模型的功能很大程度上取决于其训练数据 [1]；给定不同的数据，相似的配置脚本和网络架构最终可能生成截然不同的模型 [35],，进而产生截然不同的函数。因此，对深度神经网络模型进行功能比较，不能依赖模型自身的语法或结构属性。相反，必须像测试一样，比较函数在规范输入上的输出。给定一组规范测试输入，如果两个函数的输出足够相似，那么它们就是相似的。

然而，这一关于深度神经网络函数相似度的直观表述，需要从两个方面加以操作化：（1）应使用哪些输入来评估相似度；（2）如何量化“足够相似”？对于（1），测试集显然是合适的候选，但并非总是可用。事实上，在某些情况下，我们甚至可能不知道模型应该执行什么任务，更不用说它们使用了哪些数据进行训练和测试。对于（2），用于比较传统代码的方法和度量指标（例如功能等价性） [15], 不足以捕捉深度神经网络的相似度。近期的相关研究着重采用统计方法和度量指标，利用有意义的输入来比较深度神经网络在不同层中学习到的表征。一类颇有前景的方法是典型相关分析（CCA） [13], , 近期已将其应用于神经网络 [23, 28]。另一种同样有前景但适用范围更受限的度量指标是斯皮尔曼秩相关，它已在一项小规模研究中使用有意义的规范输入进行应用 [34]。

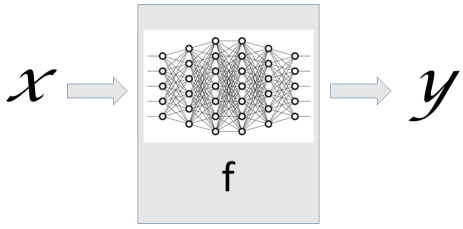


图 1 深度神经网络函数。

最后，尤其对于分类器，我们可以使用一种简单重叠度量指标，量化两个不同模型对相同输入进行分类时结果一致的次数。

本文研究了在缺乏有意义的规范输入时，如何使用从 GitHub 收集的、包含 56,355 个分类器的数据集来检测深度神经网络的相似性，并探讨了相关方法和度量指标。研究结果表明，随机输入完全适用于相似性检测。我们研究的三种度量指标效果都很好，在检测相似/不相似函数方面的准确率均超过76% 。在本研究范围内，斯皮尔曼秩相关的准确率最高，达到94% 。典型相关分析是最通用的度量指标，甚至比斯皮尔曼相关系数更通用，因为它可用于比较任意两个具有任意数量输出的函数；但在本研究中，典型相关分析导致的相似性分类错误最多。简单重叠度量的不确定性最高，但它能够检测出另外两种度量指标未能检测到的相似性。

本研究的主要贡献如下：我们阐述了寻找深度神经网络模型函数克隆这一问题，并将深度神经网络函数相似性检测付诸实践。据我们所知，这是对这一问题开展的首个大规模研究。我们表明，可使用随机输入代替规范输入来寻找相似的分类器。这一发现使模型克隆检测器的实现成为可能且更加简单，因为无需再为特定模型准备测试数据。我们揭示了三种相似性度量指标的优势与不足，其中斯皮尔曼秩相关在分类器方面表现出最高的准确率。我们整理了一个数据集，其中包含从 GitHub 收集的 56,000 多个已编译 Keras [6]分类器，并依据其输入和输出形状进行聚类。该数据集不仅可用于复现我们的研究，也可用于对深度神经网络模型开展进一步分析。

本文其余部分的结构如下：我们将在第2节讨论深度神经网络函数相似性检测问题的细节，随后在第3节详细描述本研究的设计。第4节详细介绍实验，第5节呈现敏感性分析，随后第6节介绍相关工作，第7节讨论效度威胁，第8节给出结论与未来工作。

2 问题定义

在本节中，我们首先解释在无法获取深度神经网络函数的训练/测试数据时，研究用于发现相似深度神经网络函数的过程和度量指标的必要性。然后，我们定义函数相似性的概念，以及如何对其进行量化。

2.1 动机

模型窃取。 据文献报道 [2, 39], , 软件盗窃是一个普遍存在的问题，每年造成数十亿美元的损失。软件盗窃的一种常见情形是，员工离开公司时带走专有软件代码或数据的副本；这种情形正以惊人的速度日益增多 [14]。随着人工智能

和机器学习的最新进展，以及传统软件组件被深度神经网络模型取代，软件已不再只是计算机指令。报告显示，模型窃取也开始出现，既包括直接复制模型文件（之后可能会对其稍作修改）[21]，也包括通过模型提取攻击复制模型的功能[16, 29, 36]。在这些情况下，窃取调查都需要在没有训练数据或测试数据的条件下检查并比较深度神经网络函数——训练数据不会随模型一同打包部署，并且出于各种原因（包括但不限于隐私或安全方面的顾虑），可能无法用于分析。事实上，当需要比较两个大型系统，或将一个系统与大量软件制品进行比较时，软件盗窃调查人员往往可能是第三方，甚至并不知道这些模型实现了什么样的函数。函数相似度分析对于检测模型窃取尤其有用，因为模型遭窃时，人们关注的是其函数，因此模型最终实现的函数预计仍会保持相似。

应用商店中的恶意软件。深度神经网络函数克隆检测的另一项应用，是扫描应用是否含有恶意组件的应用商店。使用深度神经网络模型支持主要功能的移动应用数量正在增加：2019 [44]中的一项研究显示，81% 包含深度神经网络模型的应用都使用这些模型来支持其核心功能。传统程序分析很快将不足以检测恶意行为，因此需要在无法访问训练数据和测试数据的情况下进行深度神经网络功能比较（应用提交时不会附带这些数据）。根据我们研究中考察的方法和度量指标，当应用包含深度神经网络函数时，可以将这些函数与先前在恶意应用中识别出的已知函数库进行自动比较。

模型搜索。深度神经网络功能比较的另一种应用场景是模型搜索。例如，搜索一个与现有模型相似、但准确率或泛化能力更高的深度神经网络函数。深度神经网络功能比较可用于将现有模型的功能与训练模型库中的模型进行比较，并为每一对模型生成相似度分数。例如，在寻找预训练模型时，可以通过进一步训练对其进行定制，以满足特定需求（类似于迁移学习）。对已训练模型进一步训练（而非从头开始）的益处包括：简化面向人工智能软件的深度神经网络函数开发，减少模型开发和训练所需的时间与计算成本，提高最终模型的性能和能力，以及克服训练数据不足的问题。

2.2 功能相似性

功能相似性衡量函数之间的相似程度。在这里，我们需要明确「函数」和「模型」这两个词的用法。函数 是一个数学对象，由其输入/输出行为描述，与其实现方式无关；模型 是函数的一种特定实现，使用神经网络实现；我们将这样的函数称为深度神经网络函数。与传统代码不同，比较

深度神经网络模型的训练脚本和测试脚本，无法用于评估其函数的相似度，因为模型的功能主要取决于用于训练它们的数据；相同的训练脚本产生完全不同的输入/输出行为，这种情况并不少见。因此，检测深度神经网络模型的功能相似性，需要进行实际的输入/输出分析。

在定义功能相似性之前，我们先定义功能等价性。在传统代码中，功能等价性此前被定义为：给定相同输入时，输出等价，同时考虑输入和输出的置换 [15]。我们将此定义推广到深度神经网络。设f_1 和f_2 为两个函数；设I 为f_1的一组输入。如果f_2 与f_1 在功能上等价，则$f_2(\phi(i)) = f_1(i), \forall i \in I$，其中 $\phi(i)$是输入i的几何变换 [3]，可以是恒等映射，也可以是缩放、反射、投影、置换及其他此类变换。对输入进行几何变换，是对 [15]中所用输入置换的推广。两者背后的直觉都是：即使输入并非完全相同，只要它们通过这些变换彼此相关，就可以搜索并找到相似性。例如，接受28×28 取值范围为 -1 到 1 的输入的数字识别函数，可以与另一个接受28×28 取值范围为 0 到 255 的输入的函数进行比较（这是缩放的一个例子）。

然而，功能等价性的意义有限。函数即使对于相同（或经过变换的）输入并未产生完全相同的输出，也可能非常相似。举例来说，考虑两个函数，它们各自只有一个数值输出；对于相同的输入，它们产生的输出满足 $o_2 = 2o_1, \forall o \in O$ ——也就是说，第二个函数的输出值始终是第一个函数输出值的两倍；显然，即便数值缩放了 2 倍，这两个函数所做的事情也非常相似。再举一例，考虑两个函数：对于相同的 10,000 个输入，它们在 9,000 种情况下产生完全相同的输出，在 1,000 种情况下产生不同的输出；同样，即便它们的输出有 10% 的时间不同，这两个函数仍然相似。

因此，衡量函数输出之间的相关性 [23, 28, 34]最能体现函数的相似度。如果两个函数对于相同（或经过变换的）输入，其输出彼此相关，那么它们就是相似的。形式化地说，f_1 和 f_2 相似，当且仅当

$$Corr(f_1(i), f_2(\phi(i))) > \theta, \forall i \in I \quad (1)$$

其中，$Corr$ 是相关性度量，ϕ 是几何变换，θ 是相似度阈值。

对于具有多个输出的函数，即使输出的数量或顺序不同，相关性度量也必须通过某种聚合方法反映多个输出的聚合结果。在这个一般性定义中，$Corr$ 表示输出的聚合相关性。

黑箱，白噪声

3 研究设计

本节将详细阐述我们工作的目标和范围，随后介绍实验所用的数据集。

3.1 研究目标与范围

方程 1 中给出的功能相似性定义要求存在用于输入/输出分析的输入。一般而言，可能无法获得有意义的规范输入，例如训练数据或测试数据。因此，本研究有两个目标：（1）考察是否可以使用随机输入代替规范输入来检测深度神经网络函数的相似性；（2）了解不同相似性度量指标之间的权衡。我们的研究结果有助于深入了解深度神经网络函数相似性检测的过程，以及各项度量指标在不同场景中的适用性。

方程 1 具有通用性，即无论两个深度神经网络函数的性质以及输入/输出形状和维度如何，都可以用它们进行比较。在本研究中，我们仅关注分类器；此外，我们将研究范围缩小至输入形状兼容、输出类别数量相同且顺序一致的单标签多分类深度神经网络分类器。此处所说的输入形状兼容，是指f_1 和f_2 的输入形状可以通过简单的重塑变换相互转换。例如，可以将28x28 2维输入转换为784 1维输入，反之亦然。本研究不包括其定义域需要更一般几何变换（例如图像缩放）的分类器（如256x256 与128x128）。此外，输出类别数量不同的分类器也不在研究范围内，例如一个分类器有10 个类别，另一个有11个类别。此处通过检查模型文件来确认输入形状和输出形状是否兼容；因此，我们假设可以访问模型文件。

遵循这些限制只是为了控制实证分析的复杂性，因为若尝试比较任意两个模型，不仅会大幅增加数据集的规模，还会使结果与分析的篇幅超出页数限制。

我们的研究分为两部分：第一部分（第 4 节）有两个目标：我们研究随机输入是否适用于深度神经网络的函数相似性检测（目标1），以及使用不同相似性度量时需要权衡哪些因素（目标2）。研究目标1很重要，因为深度神经网络模型是在规范输入上训练的，并且理应对此类输入产生输出。将规范输入替换为随机输入时，这类输入可能是白噪声，而深度神经网络生成的结果可能是随机预测，无法捕捉相似模型之间的相关性。例如，图 2 中的图像展示了一张使用随机输入生成的 28x28 图片。如果将这张图片输入一个经过训练、用于区分手写数字的深度神经网络函数，模型的输出可能是一个随机值，未必能可靠地反映相似之处和差异。因此，研究随机输入是否有助于相似性检测至关重要。我们通过比较模型在 5 个知名分类数据集上的准确率（即模型预测标签与真实标签相符的比例），以及使用随机输入计算出的模型与参考模型之间的相似度，来开展这项研究。参考模型是在目标知名数据集上训练的（由此确立真实

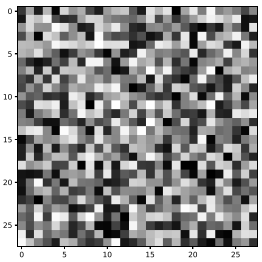


图 2 使用随机输入生成的图像示例。

值，也就是功能性的真实情况）。与目标1相关，我们还研究能否使用随机输入检测我们一无所知的模型之间的相似性。我们对此进行研究，是因为通常执行比较的人并不了解正在比较的两个模型。因此，我们使用对我们而言未知的模型运行实验，以评估在此类情形下使用随机输入进行深度神经网络相似性检测的可行性。在这里，人工检查模型的 GitHub 仓库有助于我们推理预测的相似度值。

为探索目标2，我们通过一组使用知名数据集开展的实验，比较三种相似性度量在召回率、精确率和准确率方面的表现。

- 具体而言，我们研究以下问题：研究问题 1：能否使用随机输入检测相似和不相似的深度神经网络函数？研究问题 2：各种相似性度量的表现如何？研究问题3：能否使用随机输入检测未知模型之间的相似性？
-
-

其次（第5节），我们开展敏感性分析，旨在了解输入的随机性以及一组模型训练相关特征如何影响相似性结果。因此，我们还研究以下问题：

- 研究问题4：输入生成随机化对函数响应相似性检测有何影响？研究问题5：模型训练相关特征（例如训练随机化、准确率、架构和训练数据集）对模型响应相似性检测有何影响？

3.2 评估数据集

我们实验中使用的数据集如下。

黑箱，白噪声

3.2.1 主要实验数据集(D56K)

该数据集包含**56,355** 个分类器，收集方式如下。我们使用 GitHub 代码搜索 API[1] 查找扩展名为$.h_5$ 的文件；这是保存 Keras [6] 模型时广泛使用的扩展名。[2] 我们总共获取了 340,933 个 h5 文件，其中成功下载了 335,789 个。接下来，我们尝试将这些模型加载到 Keras 环境中，以便从 h5 文件中保存的其他可能对象中筛选出深度神经网络模型，并识别出 102,602 个深度神经网络模型。加载模型后，我们根据检查模型文件所获得的输入和输出形状对其进行聚类。最后，由于我们的工作重点是分类器，因此只保留输出层具有*Sigmoid*（用于获取二分类函数）或*Softmax*（用于获取多分类函数）激活函数的模型。最终得到 56,355 个模型，分为 6,431 组。

3.2.2 敏感性分析数据集(D14)

对于与研究问题4和研究问题5相关的实验，我们使用 MNIST（10类手写黑色数字白底分类）和 FMNIST（10类服装分类）训练了 14 个模型。这两个数据集广受欢迎且公开可用，在以往研究中已得到广泛应用 [25, 27, 43]。这些模型列于表 1，并被分为若干簇；每个簇中的模型相较于第一个模型 MN Ver1 只有一个属性不同，我们将 MN Ver1 视为参考模型。“实例”组包含两个在所有方面都相同的模型，唯一的区别是分别进行训练，因此会受到训练过程随机性的影响。接下来的“训练参数”组包含训练参数与参考模型不同的模型；“架构”组包含相较于参考模型存在架构差异的模型；“数据集”组则包含使用不同数据集训练的模型，这些数据集与参考模型训练数据的相似程度各不相同。“Acc”列显示模型在各自测试数据上的准确率，“X-Acc”列显示模型在参考模型测试数据上的准确率。X-Acc 是衡量模型相对于参考模型相似度的真实值。因此，MN Rev Color（其背景与前景颜色相较于参考模型互换）和 FMN 被认为与参考模型不相似，因为它们的 X-Acc 都非常低。

3.3 功能相似性度量

为了评估两个分类器之间的功能相似性，我们分析它们对相同随机输入的预测。为了量化相似度，我们考虑三种度量指标：典型相关分析（CCA）、斯皮尔曼秩相关（ρ）和简单重叠度量。这些度量指标的选择基于以往的相似性

[1] https://docs.github.com/en/rest/reference/search#search-code。访问日期：2023-02-03[2] 此外还有其他外部存储格式，但由于 h5 格式简单，我们将搜索范围限定为 h5。

表 1 敏感性分析：训练模型（D14 数据集）

组	Name	描述	Acc	X-Acc
实例	MN Ver1（参考）	2 层全连接网络，第一层使用 ReLU，训练轮数 #epochs=10，学习率 =0.001，MNIST 训练数据缩放至 (0,1)	98.16 %	98.16 %
	MN Ver2	与参考模型采用相同的架构、训练参数和训练数据模型，分别训练	98.15 %	98.15 %
训练参数	MN Long Tr	与参考模型采用相同的架构、训练数据和学习率，训练了 50 轮	98.35 %	98.35%
	MN LR01	与参考模型采用相同的架构、训练数据和训练轮数，学习率=0.01	97.80 %	97.80%
	MN LR18	与参考模型具有相同的架构、训练数据和训练轮数，学习率=0.18	75.92 %	75.92 %
架构	MN Sig	2 层全连接网络，第 1 层使用 Sigmoid，训练参数和训练数据与参考模型相同	97.68 %	97.68%
	MN Deep NN	4 层全连接网络，前 3 层使用 ReLU，训练参数和训练数据与参考模型相同	98.21 %	98.21 %
	MN CNN	含 2 个卷积层、2 个池化层和 2 个全连接层的卷积神经网络，训练参数和训练数据与参考模型相似	99.18 %	99.18%
	MN CNN2	与 MN CNN 具有相同的架构、训练参数和训练数据，分别训练	99.06 %	99.06%
数据集	MN 第一批次	架构和训练参数与参考模型相同，训练数据为 MNIST 训练数据中前 3 万行，数值范围为 (0,1)	97.55 %	97.55 %
	MN 第二批次	采用与参考模型相同的架构和训练参数，在以下数据上训练 MNIST 训练数据中按 (0,1) 缩放的第 2 个 30K 行	98.39 %	98.39%
	MN 无缩放	采用与参考模型相同的架构和训练参数，在以下数据上训练 范围为 (0,255) 的 MNIST 训练数据	97.21 %	90.74 %
	MN Rev Color	与参考模型相同的架构和训练参数，训练数据为 黑色背景、白色数字的 MNIST，缩放至 (0,1)	95.27 %	1.67 %
	FMN 模型	与参考模型采用相同的架构和训练参数，训练所用数据为 缩放至 (0,1) 的 Fashion MNIST 训练数据	88.88 %	8.35 %

研究：典型相关分析已被用于衡量神经网络表征之间的相似度 [23, 28], , 而斯皮尔曼相关系数是一种合适的相似性度量 [5]，在存在规范输入时已被用于评估模型的功能相似性 [34]。重叠度量的灵感来自 Jiang 等人对代码功能等价性的研究 [15], , 并经过调整以衡量相似度；在这里，我们假设输出的顺序相同，只需衡量分类器在相同输入下输出的重叠程度。以下将介绍这些度量指标的具体细节。

典型相关分析（CCA）是一种统计方法，通过找出两组变量之间的线性关系，并使这些线性关系之间的相关性最大化，来推断它们之间的关系 [23, 34]。此前，研究者曾将典型数据集输入神经网络，并测量这些网络不同层的激活值上的典型相关分析，以衡量不同神经网络中各层学习到的表征之间的相似度 [23, 28]。典型相关分析具有很强的通用性，能够找出基数不同的两组变量之间的关系，因此适用于评估函数的功能相似性，即使这些函数的

输出维度不同 [34]。[3] 典型相关分析中的相关系数范围为 0 到 1 [33]。我们针对一对分类器在其接收的输入上产生的输出预测概率向量，计算了典型相关分析的相关系数，得到n 个相关值，其中n 是输出标签的数量。随后，我们按照 [23] 中所述的方法，对n 个相关值取平均 值，以汇总结果。

斯皮尔曼相关系数 [34]是一种非参数相关度量 [5]，此前的研究曾使用它，基于规范输入研究神经网络相似性 [34]。除此之外，斯皮尔曼相关系数之所以适用，还在于它通过计算秩相关，不局限于测量变量之间的线性关系。由于深度神经网络的训练数据可能经过各种数据缩放，这一特性使该度量指标即使在数据（具体而言，训练数据的输出值）发生非线性数据变换时，也能捕捉相似性。此外，一般而言，如果函数在相同输入上的单调性相似，那么这些函数也相似。斯皮尔曼相关系数的范围为 -1 至 1，其中正值表示正相关，负值表示负相关，而接近零的值（通常介于 -0.1 和 0.1 之间）表示不存在相关性。为了对分类器的输出计算斯皮尔曼相关系数，我们将每个输出标签对应的概率值视为一个变量，并计算该输出在全部 m 个样本上的相关值。换言之，若函数 f_1有 n 个输出类别 $\{c_1^1, c_1^2, .., c_1^n\}$，且函数 f_2 也有 n 个输出类别 $\{c_2^1, c_2^2, .., c_2^n\}$，我们便计算 n 个相关值 $\{\rho^1, \rho^2, .., \rho^n\}$，使得 $\rho^i = \text{Correlation}(c_1^i, c_2^i)$。为汇总结果，我们取这 n 个相关值的平均值，并将其作为最终相似度值。因此，使用该度量指标要求两个函数具有相同的输出形状和维度，并以相同顺序解释这些输出；如前所述，这是本研究的一项简化假设。

最后，我们使用的最简单的相似性度量指标是简单的重叠度，即两个函数对随机输入进行分类时结果一致的次数。该度量指标只计算两个分类器的输出值相同的百分比。它与准确率度量非常相似：如果将参考分类器的预测结果视为真实值，那么该度量指标就会计算候选模型相对于参考模型的准确程度。鉴于其分类性质，该度量指标仅适用于分类器。

总而言之：典型相关分析是最通用的度量指标，可用于比较任意两个函数；我们使用斯皮尔曼相关系数时，假设输出的数量和顺序相同；而简单重叠度量则假设函数执行的是分类任务。

3.4 相似性阈值

相似性度量需要设定一个阈值，以此将函数分类为相似或不相似。对于典型相关分析和斯皮尔曼，我们可以根据经验调整该阈值

[3]在我们的研究中，如前所述，可比较模型的输出维度从不不同。

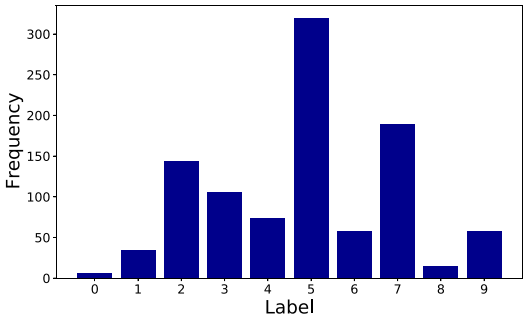


图 3 无约束随机输入的标签频率

采用关于相关性范围的通用公认准则：介于0 和0.1 之间的值通常被认为不存在相关性，介于0.1 和0.2 之间的值被认为是弱正相关，介于0.2 和0.5 之间的值被认为是中等程度的正相关，而大于0.5 的值则是高度正相关。就相似度而言，我们将中等及以上的相关值（值为>= 0.2）视为相似，将弱相关值（0.1 <值为< 0.2）视为不确定，并将值为<= 0.1 的情况视为不相似。我们在实验中进行了各种敏感性分析，并观察到这些阈值能够很好地捕捉相似函数，同时尽量减少错误预测的数量。这意味着，对于报告负值负相关的斯皮尔曼，这些值也被视为不相似，因为反向关系同样表明预测结果之间存在差异。

对于重叠度量，计算得到的相似度值表示两个分类器对相同输入预测出相同标签的比例。该度量与准确率类似，都是统计预测标签与真实标签相匹配的次数；在我们的情形中，真实值是参考模型的预测结果，我们统计候选模型的预测标签与参考模型的预测标签相匹配的次数。需要特别指出的是，研究表明，如果底层输入数据集的标签分布不平衡，准确率会受到负面影响 [22]。重叠度量也存在这一问题：当参考模型对输入的预测标签分布不平衡时，计算出的重叠度值就会变得不可靠，而且无法指定一个适用于各种分类器（无论标签数量多少）的相似度/不相似度阈值。根据我们的实验，在无约束随机输入上，模型很可能会比其他标签更频繁地预测某些标签（例如，这是由模型偏差导致的）。例如，图 3 展示了一个 10 类数字识别模型在无约束随机输入上的预测标签频率。如图所示，标签 5 的预测频率远高于其他标签，而标签 0 和 8 等则极少出现。附录（A 节）对此问题作了更深入的解释。

为了解决上述无约束随机输入带来的问题，我们使用重叠度量生成并使用平衡随机输入（此类输入的生成过程见附录）。具体而言，若输入数据集d 有m 行，且相对于参考模型m_r 是平衡的，而该模型有n_r 个输出类别

（也就是说，在 d 上，m_r 对每个输出类别各预测 m n_r 次），那么任意模型（同样有 n_r 个输出类别）仅凭随机机会，就可能与 m_r 有 $1/n_r$ 次预测一致。因此，当 m_r 与某个模型在 d 上的一致程度处于随机机会水平（$\lessapprox 1/n_r$）时，不能将这两个模型判定为相似。如果一致程度远高于 $1/n_r$，例如达到随机机会水平的两倍（$2/n_r$），那么就说明起作用的不只是随机机会，这是模型行为相似的有力迹象。然而，考虑到模型的准确率并不完美，$2/n_r$这一上限阈值可能过于严格。例如，对于二元分类器，这一阈值相当于 100% 的相似度，而这在实践中并不会发生。即使是最相似的模型，也不会在 100% 的情况下彼此预测一致，而且它们的准确率会影响一致程度。通常，我们并不知道模型的准确率，但我们知道准确率达到 100% 的情况很罕见。因此，应将相似度阈值 $2/n_r$ 乘以一个经验因子 α，使其变为 $2\alpha/n_r$。对于大多数优质模型，预期准确率会高于 90%；因此，视是否将两个模型的准确率都纳入考量而定，α 的取值可以介于 0.81 和 0.9 之间。在我们的实验中，我们发现 $\alpha = 0.9$ 的效果不错，同时仍能排除假阳性。

基于此，使用重叠度量时，我们会针对参考模型生成平衡随机输入。对于n_r 为两个模型的输出类别数的情况，相似度 $<= 1/n_r$ 时视为不相似，相似度 $>= 2\alpha/n_r$ 时视为相似，而介于两者之间时则视为不确定，此时我们无法有把握地判断模型之间的相似度。

3.5 相似度检验流程

我们为开展这项研究设计的流程由两个高层模块组成：(i) 兼容性验证模块，以及 (ii) 功能相似性模块。设计该流程时，我们考虑了将其用于比较任意两个分类器，而不仅仅是比较数据集中同一簇内的模型。在兼容性验证模块中，我们通过检查两个模型是否具有相同的输出形状，来验证它们的输出兼容性。这包括验证两个模型预测的类别数量是否相同，以及它们的输出层是否使用相同的激活函数。我们通过检查模型文件中提供的架构来完成此项工作。输入兼容性不必局限于输入形状完全相同：如果两个模型的输入形状可以相互推导，则它们也可以兼容。这是为了适应工程师在训练模型时可能进行的各种数据集重塑。因此，在验证输入形状兼容性时，给定i_r 和i_c 分别作为参考模型和候选模型的输入形状，我们将输入形状展平并进行比较：当且仅当长度(展平(i_r)) == 长度(展平(i_c))时，两个模型的输入才兼容。

最后，我们通过生成无约束随机输入（取值介于 -1 和 1 之间的浮点值）来研究典型相关分析和斯皮尔曼相关系数所衡量的函数相似度，并针对重叠度量生成平衡的随机输入（使其相对于参考模型的预测保持平衡），然后使用各度量指标测量生成输入上的相似度。此流程的实现可在此链接中查看。

表 2 相似性度量的成功与错误

查询模型	相似度范围						不相似度范围						不确定范围		
	#TP			#FP			#TN			#FN			#总计		
	典型相关分析	Spr	Ovl	典型相关分析	Spr	Ovl	典型相关分析	Spr	Ovl	典型相关分析	Spr	Ovl	典型相关分析	Spr	Ovl
MN 相同 形状	60	72	68	0	0	0	23	23	21	4	2	2	12	2	8
MN RevClr 相同 形状	1	1	1	6	0	0	40	99	97	0	0	0	53	0	2
FMN 相同形状	11	15	15	0	0	0	82	82	41	0	0	0	4	0	41
Iris	339	336	272	13	2	0	12	27	23	0	1	1	7	5	75
Sonar	60	60	25	0	0	0	1	1	1	0	0	0	1	1	36
MN 兼容形状	34	39	37	0	0	1	59	58	49	3	1	2	4	2	11
FMN 兼容形状	20	39	38	0	1	2	52	51	29	2	1	1	20	2	24
卷积神经网络 兼容形状	0	0	12	0	0	1	59	59	48	41	38	5	0	3	34
总计	**525**	**562**	**468**	**19**	**3**	**4**	**328**	**400**	**309**	**50**	**43**	**11**	**101**	**15**	**231**

4 实验

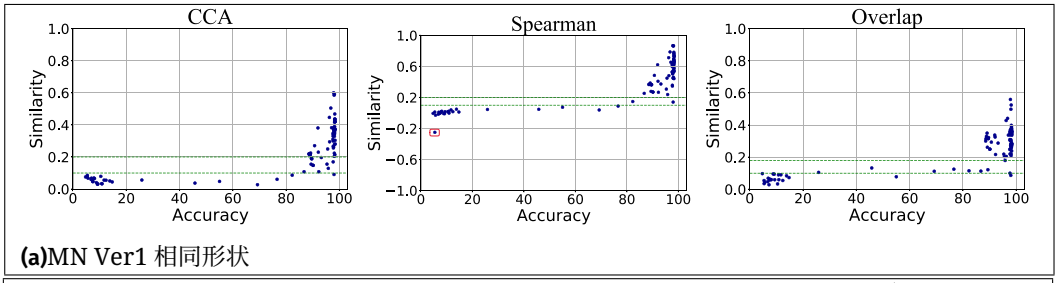
本节其余部分将详细介绍为回答研究问题而开展的实验。

4.1 准确率与功能相似性

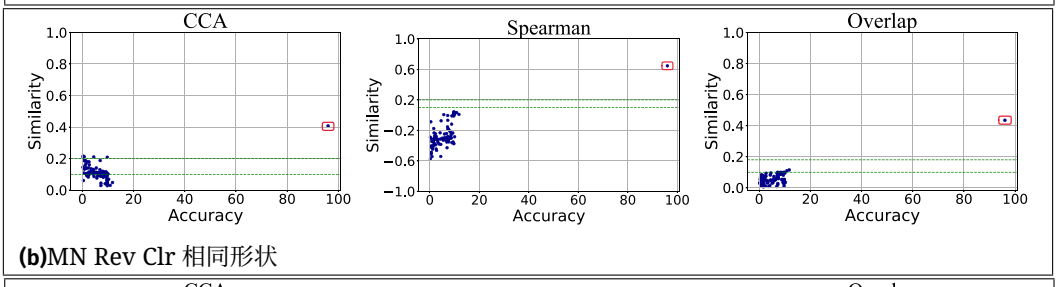
这里我们回答研究问题 1 和研究问题 2。我们选取了 5 个已知数据集——*MNIST*、*Fashion MNIST*、*MNIST* 反色、*Iris [10],* 和 *Sonar [8, 11]* ——并在这些数据集上训练分类器，将其作为参考模型（在这些数据集中，前三个用于图像分类，详见第 3.2.2 节；后两个则包含数值特征，分别描述 Iris 数据集中的鸢尾植物种类，以及 Sonar 数据集中的声纳信号）。随后，我们在 D56K 数据集中找出输入和输出形状与这些模型相匹配或输入兼容的簇，并在相应的标准测试数据集上测量从这些簇中提取的模型的准确率；也就是说，统计模型输出与标准测试数据集中的标签一致的比例。通过使用标准测试集，我们可以根据模型的准确率值获得其功能的真实值：根据经验，我们认为，当模型在某个标准测试数据集上的准确率高于 65% 时，它们对该数据集输入的分类效果足够好。当准确率低于50%时，我们认为模型分类的是其他内容；而准确率介于50%和 65％之间时，我们认为模型尚无法确定，其功能相似性较为模糊。为确保准确计算模型的准确率，我们在适用的情况下对输入采用了多种常见的特征缩放技术，测量每个模型在所有缩放和未缩放数据集上的准确率，并将其中的最高准确率作为模型的真实准确率。最终，有 16 个案例落在无法确定的区间；我们将其排除在成功和错误的计算（稍后呈现的表 2）以及召回率、精确率和准确率的计算（稍后呈现的表 3）之外。

在确定模型是否对所选测试集执行了正确分类（即真实值）之后，我们向参考模型和这些模型输入随机输入，并使用第 3.3 节中介绍的度量指标测量相似度。

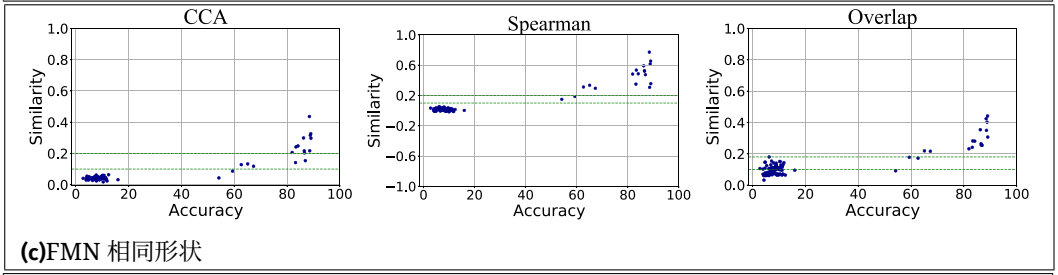
黑箱，白噪声



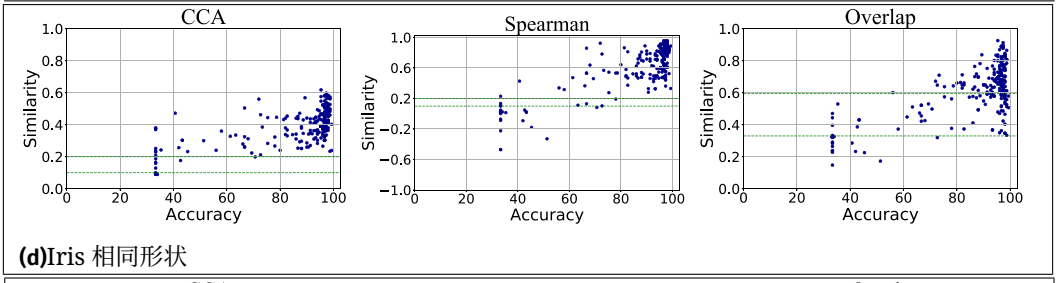
(a)MN Ver1 相同形状



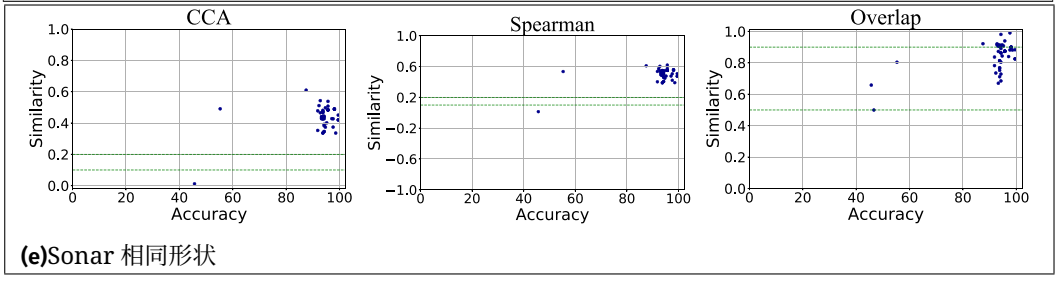
(b)MN Rev Clr 相同形状



(c)FMN 相同形状



(d) Iris 相同形状



(e) Sonar 相同形状

图4 相同形状 GitHub 模型的准确率与相似度

表 3 各相似性度量的精确率、召回率和准确率

精确率			召回率			准确率		
CCA	Spr	Ovl	CCA	Spr	Ovl	CCA	Spr	Ovl
96 %	**99 %**	**99 %**	85 %	**91 %**	76 %	83 %	**94 %**	76 %

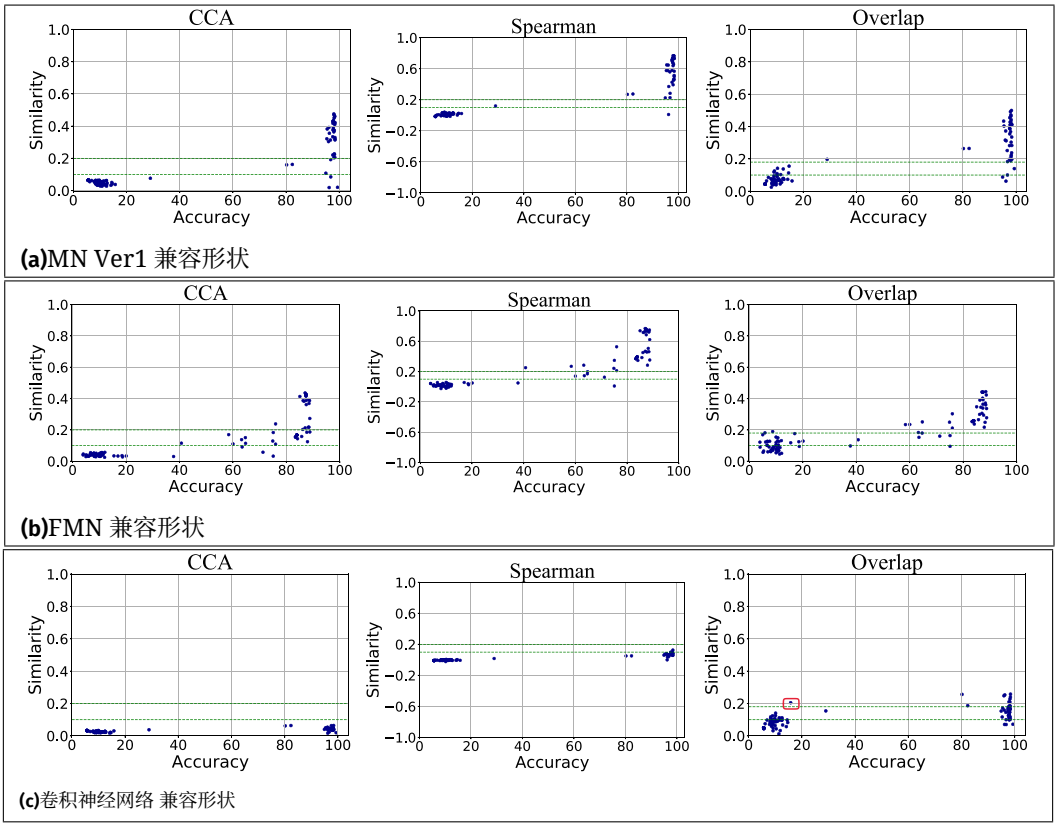


图5 GitHub 模型的兼容形状准确率与相似度

对于每个参考模型和每种度量指标，我们都展示了散点图，用于描绘候选模型（与相应参考模型进行比较的模型）的准确率（横轴，作为真实值）与其使用随机输入得到的参考模型相似度分数（纵轴）之间的关系。这些图分别展示于图4（相同形状模型）和图5（兼容形状模型）中。图中的点表示每个候选模型与图注中所述参考模型进行比较时的<样式 id='1'>相似度</样式>与<样式 id='3'>准确率</样式>。例如，在图 4(a) 中，每个点都表示与名为 <样式 id='5'>MN Ver1</样式> 的模型进行比较时，各候选模型之间的这种关系。在这些图中，“相同形状”表示参考模型及其候选模型具有完全相同的输入形状，而“兼容形状”表示输入形状并不完全相同，但彼此兼容。每个点周围的红色方框标出了有趣的案例，稍后将在本节末尾讨论。每幅图中的绿色水平虚线表示相应相似性度量的不相似阈值和相似阈值（对于重叠度量，阈值会根据输出类别数量 <样式 id='7'>n</样式> 而变化）。底部绿色虚线以下的区域为不相似区域，两条线之间的区域为不确定区域（根据这些阈值，该度量无法高置信度地判断模型的相似性），顶部绿色虚线以上的区域为相似区域。理想的度量指标应使不确定区域中的点数最少。

作为一个更详细的示例，图 4(d) 展示了我们训练的 Iris 参考模型与找到的 376 个模型之间的准确率与相似度关系；这些模型具有完全相同的输入形状（大小为 4 的向量）和输出形状（大小为 3 的向量）。准确率值来自使用标准 Iris 数据集对这些真实世界模型进行测试，而每幅图中的相似度值则来自使用相应度量指标和随机输入对这些模型进行比较。对于 CCA 和斯皮尔曼，绿色阈值线分别位于 <样式 id='1'>0.1 </样式>和<样式 id='3'>0.2</ 样式 > ；对于重叠度量，在给定 3 个输出类别的情况下，阈值线位于 $1/3$ (随机机会) 和 2×0<样式 id='7'>。</样式>$9/3$ (非随机机会)。

每种度量指标总共进行了 $1,037$ 次比较，因此，所有图表中共显示 3,111 个数据点（我们有 3 种度量指标，因此，$1,037 * 3 = 3,111$）。

此外，表 2 量化了每种度量指标在所有这些比较中的成功次数和错误次数。“相似度范围”下方的列显示了根据各度量指标的阈值被归类为相似的预测结果数量，因此也显示了每种度量指标的真正例和假阳性数量。“不相似度范围”下方的列采用相同的方法处理被预测为不相似的结果，并列出每种度量指标的真负例和假负例数量。“不确定性范围”部分显示了根据各度量指标的阈值被归类为“不确定”的预测结果。

表 3 根据表 2 的结果，展示了各度量指标的精确率、召回率和准确率（即预测的相似度/不相似度值与真实相似度/不相似度值相符的比例）。在此，我们将任何低于相似度阈值的值（包括不确定区域）都视为负面（不相似）预测。

研究问题 1：能否使用随机输入检测相似和不相似的深度神经网络函数？ 结果表明，对于这三种度量指标，相似与不相似的函数都很容易区分。从视觉上看，这一点体现在上方各图中相似度阈值以上的大簇，以及下方各图中不相似度阈值以下的大簇。表 3 中较高的精确率（超过 96%）、召回率（超过 76%）和准确率（超过 76%）也印证了这一观察。因此，我们得出结论：可以使用随机输入来检测深度神经网络函数的相似性。

研究问题 2：各种相似性度量指标的表现如何？ 我们查看表 2 的最后一行来回答这个问题。与另外两种度量指标相比，斯皮尔曼度量的真阳性和真阴性数量最多，假阳性和不确定情况的数量最少。另一方面，重叠度量的假负例数量最少。这些结果以及表 3 中的精确率、召回率和准确率表明，总体而言，斯皮尔曼度量是检测深度神经网络函数相似性的有力度量指标。

然而，观察各个具体案例可以发现，有些情况下，不同度量指标的表现各异。其中一个案例是*CNN 与兼容形状*：在此案例中，典型相关分析和斯皮尔曼度量指标未能检测到任何真正例，而重叠度量检测到了 12 个真正例。该模型的一个显著特点是，它是一个卷积神经网络模型，而与之比较的模型（除一个外）都没有采用卷积架构。这表明，参与比较的模型之间存在架构差异，可能会给相似性度量带来挑战，使其无法做出准确的预测结果。我们将在第5节全面探讨这一问题。不过目前，结果表明，

重叠度量在此类情形下仍能检测到少数真正例，而另外两个度量指标则预测几乎所有案例都不相似。

这些结果表明，总体而言，斯皮尔曼度量适用于检测深度神经网络函数的相似度。在某些情况下，例如模型架构存在显著差异时（可通过模型文件进行检查），重叠度量（使用 BRINC 输入）是更合适的选择。与斯皮尔曼度量相比，典型相关分析并未展现出更优的结果。不过，与重叠度量相比，它的召回率和准确率更高，而重叠度量的精确率更高。

这里我们还讨论一些有趣的比较案例：

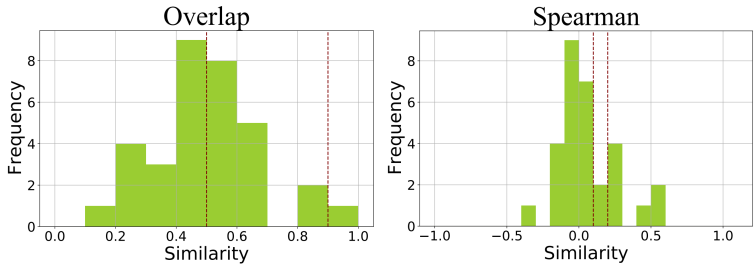
MN Rev Clr 与相同形状：这里一个有趣的发现是，斯皮尔曼度量识别出 MN Rev Clr 与多个模型存在反向关系（负相关值），这反映了训练数据集的颜色反转情况（如第 3.2.2 节所述，我们通过反转 MNIST数据集中背景和数字的颜色，创建了 MN Rev Color数据集；这样做是为了将背景颜色改为黑色，并将数字改为白色）。另一个发现是，有一个模型在 MN Rev Color数据集上的准确率为 $> 90\%$，且使用全部三种度量指标计算时，该模型的相似度值都很高。图4(b)中的所有散点图都圈出了这个案例。由于 MN Rev Color的训练数据是我们通过反转颜色整理得到的，我们原本预计不会发现任何相似函数。因此，我们调查了该案例的代码仓库及其训练代码，结果发现，我们应用于 MNIST数据集以创建 MN Rev Color数据集的同一颜色变换，也应用于该模型的数据集。图4(a)的斯皮尔曼散点图中也圈出了这个模型：斯皮尔曼度量计算得出该模型与 MN Ver1参考模型的相似度为 ≈ -0.25，表明二者之间存在中等程度的反向关系。

卷积神经网络 vs. 兼容形状：这一组中一个特殊的案例是准确率极低（低于 20%）的模型，它仅在重叠度量指标下的得分高于相似度阈值，在图5(c)的重叠度散点图中以圆圈标出。这可能是第一类错误（假阳性）的证据，因此我们分析了该模型的 GitHub代码仓库：该模型能够识别数字，且所识别的数字与 MNIST数据集中的数字相似，但训练所用的是该项目作者创建的另一数据集。因此，使用 BRINC 生成的输入时，重叠度量指标能够将其标记为与我们的卷积神经网络模型相似。另两种度量指标则未能检测到这一情况。

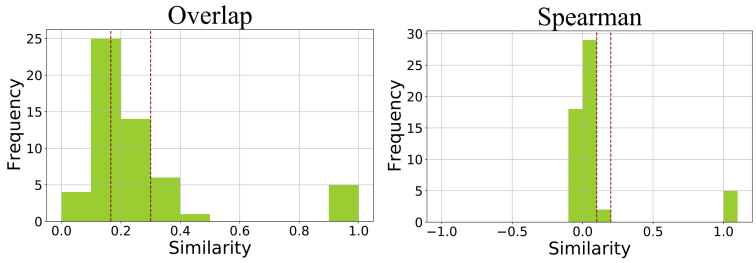
4.2 未知模型

在这里，我们回答关于能否在未知模型之间发现相似度的研究问题3。我们的目的是了解在没有任何关于参考模型或候选模型的信息时，使用随机输入的各项研究度量指标会如何表现（在上一节中，我们了解参考模型，因为这些模型是由我们训练的）。为此，我们从 D56K 数据集中随机选取两个簇，并从每个簇中选取一个分类器作为参考模型。随后，对于每个簇，我们使用第4.1节中观察到精确率最高的两种度量指标——斯皮尔曼和重叠度——计算所选参考模型与该簇中所有其他模型之间的相似度分数。

黑箱，白噪声



(a)原始数据模型



(b)无数据模型

图6 任意模型的相似度值

对于其中一个参考模型，我们在 GitHub代码仓库中找到了一个包含字符串变量和类别变量的原始数据集文件，以及一段将这些特征值转换为适用于深度神经网络模型的数值的代码。因此，尽管该模型有可用的数据集，但使用它需要执行未知的处理步骤并投入人工工作，这些都难以自动化。对于另一个参考模型，我们没有在 GitHub代码仓库中找到任何相关的数据集文件。图6按度量指标展示了各簇的参考模型与该簇中所有其他模型进行比较时得到的相似度值（横轴），以及这些相似度值的频数（纵轴）。虚线表示各度量指标的不相似性阈值和相似性阈值，这些阈值基于第3.4节讨论的阈值。

可获取原始数据的参考模型： 包含此参考模型的簇总计包含34个模型，其输入是一个长度为26的向量，输出为二元值。该参考模型的 GitHub代码仓库将其描述为旨在进行“电信客户流失检测”。根据图6(a)，斯皮尔曼检测到7个相似案例。在这些案例中，重叠度指标检测到其中1个相似案例，并将其余案例归入不确定区域：其中2个相似度较高（超过0.8），另4个的相似度为 ≈ 0.61 。对于两种度量指标判定相似度最高的3个模型，其代码仓库表明这些模型用于客户流失预测，因此我们将其视为真正的相似案例。另4个模型的代码仓库将其描述为用于人群困境识别或分析，因此，我们不认为它们是斯皮尔曼报告的真正例。

没有可用数据的参考模型： 包含此模型的簇中有 56 个模型，输入形状为 $48 \times 48 \times 1$ ，输出形状为 6。参考模型的代码仓库将其描述为一个“面部情绪识别”项目。如图6(b)所示，两种度量指标都检测到了五个完全匹配（相似度为100% ）：其中三种情况下，模型代码仓库的描述提到了面部识别或情绪识别，

另外两种情况下则没有提供任何信息。重叠度量指标还检测到了其他相似案例：一例的相似度为 ≈ 0.42，其代码仓库的描述为情感识别；另有六例的相似度为 > 0.3。在这六例中，有三例的代码仓库提到了面部识别或情绪识别，其余案例则没有找到描述。斯皮尔曼度量指标未检测到这些案例。

基于这些结果，我们可以回答**研究问题3：是否可以利用随机输入检测未知模型之间的相似性？**结果证实，借助随机输入，我们能够检测出那些我们没有（或仅有少量）信息的模型之间的相似性。

4.3 讨论

本节所呈现的结果表明，对于使用三种已研究的度量指标检测深度神经网络分类函数的相似性而言，随机输入可以替代规范测试输入。斯皮尔曼相关系数在大多数实验中表现最佳。斯皮尔曼相关系数也易于使用，因为它只需要简单的无约束随机输入，且结果限定在 $[-1,+1]$ 范围内，并具有公认的差异性/相似性阈值。该度量指标还能够检测函数之间的反向关系。典型相关分析也易于应用，在输入和阈值方面与斯皮尔曼相关系数相似，且其相似性结果的范围为 $[0,1]$。重叠度量的应用较为复杂，因为它需要生成平衡随机输入。不过，研究表明，当模型之间存在显著的架构差异时，该度量指标表现更佳。在输入数量方面，我们观察到，对于平衡随机输入，所需随机输入的数量与输出类别的数量相关；输出类别越多，为了以足够的数量覆盖所有输出标签，就需要更多输入。对于另外两种度量指标，我们观察到，输入数量超过3,000 即可获得令人满意的结果；不过，本文报告的结果基于20,000 个随机生成的输入。

总的来说，正如结果所示，这三个度量指标相互补充；有些情况下，一个度量指标检测到相似度，而其他度量指标则落在不确定性范围内。因此，为获得最准确的结果，建议结合使用这些度量指标，并根据所有指标的结果得出最终结论。

5 敏感性分析

在此，我们展示一组使用数据集 D14 的实验，以回答研究问题4和研究问题5。为回答这些研究问题，我们给出图7，其中展示了参考模型（MN Ver1）与数据集 D14 中所有其他模型之间的相似度箱线图（针对每种度量指标）。每个箱对应于 10 个不同的随机输入数据集（用于典型相关分析和斯皮尔曼指标的无限制输入，以及用于重叠度的 BRINC 输入）。生成 BRINC 输入的参数值为：$distance = 0.001$, $mutPer = 5\%, ranges = \{(-1,0),(0,1),(-1,1)\}, maxMut = 300, maxValid = 1000$.

黑箱，白噪声

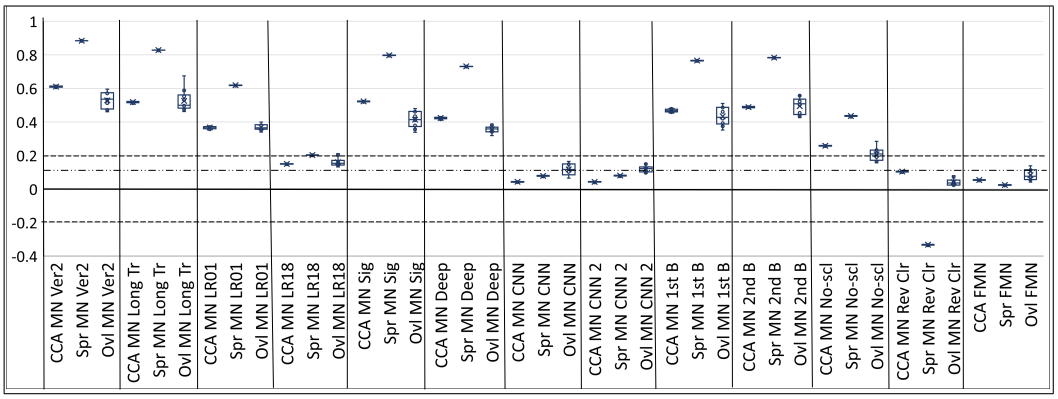


图7 与 MN Ver1 的相似度。每个箱线图显示：最小值、 1^{st} 四分位数、中位数、 3^{rd} 四分位数、最大值。

研究问题4：输入生成随机化对函数响应相似性检测有何影响？ 为回答这个问题，我们考察使用10组分别生成的随机输入计算相似性结果时的变化。图7显示，随机化对典型相关分析和斯皮尔曼指标的影响可以忽略不计。这两种度量指标对应的箱体很小，其最小值、最大值和中位数几乎相同。对于重叠度量指标，我们可以看到箱体内存在一定变化，但变化范围相对较窄。这可以从 1^{st} 和 3^{rd} 百分位数以及每个箱的最小值和最大值看出。特别是，我们注意到，相似度较低时变化较小；而相似度较高时，最小值与阈值相距甚远。这表明，通常只需进行一次输入生成，就足以确定功能相似性。当第一次测量结果接近阈值时，可能需要重复测量，并取中位数作为最佳近似值。

研究问题5：模型训练相关特征对其响应相似性检测有何影响？

训练随机化： 我们考察“实例”组中的 MN Ver1 和 MN Ver2，这两个模型几乎完全相同，准确率也几乎一致。如图7所示，斯皮尔曼和重叠度预测它们的相似度中位数最高，而典型相关分析预测的相似度中位数也位居前列，这正如我们所预期的。

训练参数： 观察图7中“训练参数”组的模型，我们发现，增加训练轮数（MN Long Tr）似乎不会对测得的相似度产生显著影响。然而，学习率（MN LR01 和 MN LR18）似乎会产生可测量的影响。对于 MN LR01，相似度值仍远高于相似度阈值；但对于 MN LR18，这些数值则低得多。原因可能是，增大学习率会使模型更快收敛，从而可能损害其泛化能力。

模型准确率： 大多数受研究的模型准确率为 $> 95\%$，有两个例外：MN LR18（准确率为 $\approx 76\%$）和 FMN（准确率为 $\approx 89\%$）。图7显示，MN LR18 与参考模型的相似度中位数要么接近（斯皮尔曼），要么位于不确定区域内（典型相关分析和重叠度）。对于准确率更高的 FMN 模型，我们看到其中位数正确地低于

不相似度阈值。这意味着，如果模型的准确率足够低，相似度值就会受到模型准确率的影响。原因可能与低准确率会损害模型的预测能力这一事实有关。

架构： 架构组包含两个与参考模型架构差异较小的模型（MN Sig 和 MN Deep）（即模型层的组织方式存在差异），以及两个与参考模型架构差异显著的模型（MN CNN 和 MN CNN2），因为它们包含卷积层。图7显示，对于差异较小的模型（MN Sig 和 MN Deep），其相似度中位数远高于相似度阈值。然而，对于重叠度量指标，CNN模型的相似度值仅略高于随机阈值；对于另外两种度量指标，其相似度值始终处于不相似区域。这表明，当架构差异显著时，使用随机输入进行相似性检测可能无法准确计算相似度测量结果。

训练数据集： 观察图7中的数据集组（从 MN 1st B 开始），可以发现，对于保持功能不变的训练数据变更（MN 1st B、MN 2nd B、MN No-scl），相似度计算相对稳健；而对于使用会产生功能不同分类器的数据训练的模型（MN Rev Clr 和 FMN），相似度计算能够正确地将其判定为不相似。

6 相关工作

神经网络表征的相似度。 有一系列工作致力于研究神经网络在不同层学习到的表征之间的相似度。Laakso等人 [19] 提出通过比较神经元激活之间的距离来比较神经网络表征。Raghu等人 [28]提出使用典型相关分析（CCA）来衡量神经网络表征之间的相似度。在此基础上，Morcos等人 [23] 提出了投影加权典型相关分析。Kornblith等人 [17]指出，CCA无法衡量维度高于数据点数量的表征之间有意义的相似度，并提出了一种名为CKA的相似度指数。Li等人 [20] 研究了分别训练的深度神经网络是否会学习收敛到相似空间的特征，Wang等人 [37] 则研究了具有相同架构但采用不同初始化进行训练的神经网络是否会学习相似的表征。与这些工作不同，我们旨在从功能视角出发，并在缺乏有意义输入的情况下，寻找深度神经网络之间的相似度。

神经网络模糊测试。我们还受到生成输入以测试深度神经网络的相关研究的启发。DeepXplore [26] 提出了神经元覆盖率，将其作为测试覆盖率指标。后续研究 [12] 发现，神经元覆盖率存在一些问题，例如生成不自然的输入，以及偏向某些标签。TensorFuzz [24]根据被激活的激活值（通常是逻辑值或前一层的值）来衡量覆盖率；如果激活向量的距离大于阈值，就会发现新的覆盖。DeepHunter [42] 提出了一种需要领域知识的蜕变变异策略。DiffChaser [43] 通过发现多个模型（主要是一个模型及其变体）之间的不一致，帮助进行模型调试。

黑箱，白噪声

7 效度威胁

在所有实验中，我们假设参考模型和候选模型的输出标签顺序相同。尽管这种顺序不同的情况不太可能出现，尤其是对于知名数据集而言，但此类情况的存在会影响我们的结果，包括计算得到的模型准确率和相似度预测。我们使用均值对模型输出的相关值进行汇总，因为它能够全面概括所有输出类别对应的相关值。为此采用其他统计指标仍需进一步研究。

8 结论与未来工作

本文提出并讨论了在缺少规范输入的情况下，如何寻找神经分类器之间的功能相似性这一问题，并研究了可用于此目的的三种度量指标。据我们所知，这是首个针对该问题的大规模研究。我们还提出了一种方法 BRINC，用于生成可与重叠度量搭配使用的分类器平衡随机输入。结果表明，使用随机输入检测深度神经网络函数的相似性是可行的，而且所研究的三种度量指标均表现出良好前景。不过，斯皮尔曼度量无论在易用性还是整体结果方面都更胜一筹。

我们计划将这项工作拓展到分类器之外，并研究适用于检测其他类型神经函数之间相似性与差异性的度量指标和输入。此外，另一个值得关注的相关工作方向是，研究在模型窃取、模型搜索和恶意软件检测场景中，使用随机输入的所提相似性度量的适用性。

A 附录

以下各节详细说明了使用非平衡输入计算重叠度量所导致的问题，以及一种生成可用于重叠度量的平衡随机输入的方法。

A.1 非平衡输入与重叠度量的问题

图8 上方的两幅图展示了使用非平衡输入和重叠度量所导致的问题。这些图显示了两个相似数字识别模型与许多其他模型之间的重叠相似度预测结果（纵轴），这些预测均使用同一组非平衡随机输入得出。此处的横轴表示候选模型在参考模型测试数据集上的准确率，因此可作为相似度的真实值。尽管这两个参考模型彼此相似，但它们与其他模型之间的相似度预测结果却大不相同，而且更糟的是，这些结果分布得很分散，难以确定阈值线应画在哪里以判断相似度。模型2的情况甚至更糟，相似度测量结果

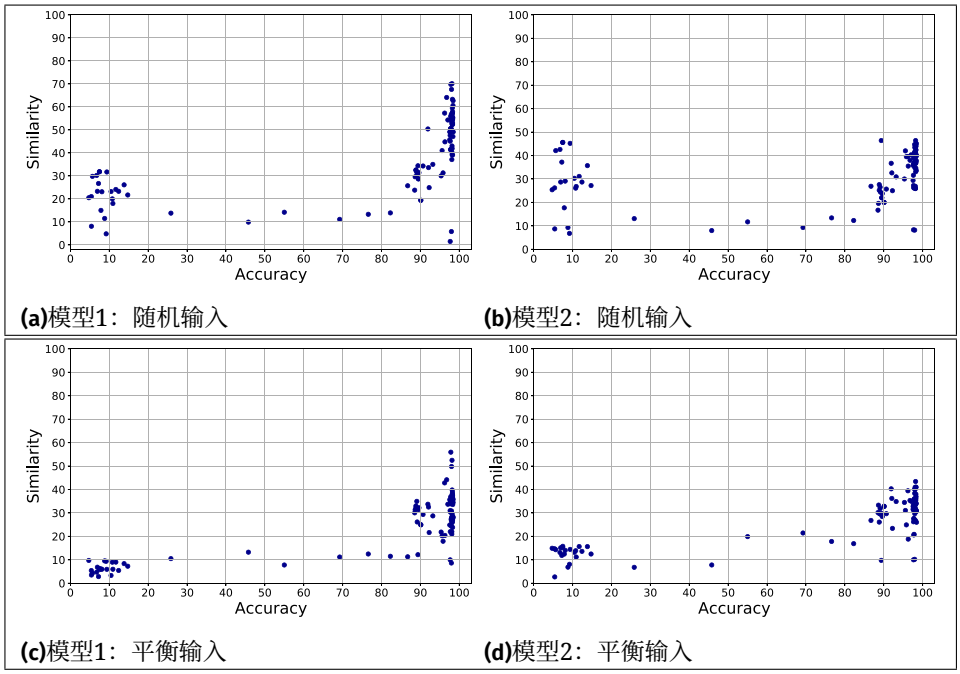


图8 重叠度量：平衡输入与非平衡输入

对于相似和不相似的模型都达到 ≈ 40%。无论选择哪条线，都可能造成多个假阳性。

在实践中，我们观察到，无约束地生成随机输入时，这些输入最终会偏向一组标签，因此会使重叠度量的相似度阈值难以设定。这促使我们考虑输入生成过程的下一种方案：施加一些约束，使参考模型对生成的随机输入的预测标签呈平衡分布。图8底部的两幅图展示了在平衡输入上，同两个模型的相似度。可以看到，相似度值聚集在一起，因此能够设定一个适用于两种情况且不会产生假阳性的阈值（相似模型的预测相似度值高于 ≈ 20%）。

A.2 平衡随机输入生成

我们使用一种称为 BRINC（神经分类器的平衡随机输入）的算法来生成平衡随机输入。生成的输入相对于参考模型的预测呈平衡分布，因此可以与重叠度量结合使用，以估计深度神经网络函数的相似度。该算法的工作方式如下：

种子输入。 BRINC 首先生成一些种子输入，以均衡的形式覆盖所有输出标签。从一组初始输入开始（我们纳入了一个值全为零的输入，以及一个值为 -1 到 1 之间随机浮点值的输入），通过不同的变异百分比反复对种子输入进行变异，从而生成一些覆盖所有输出标签的种子输入。

下一个要进行变异的输入。 生成种子输入后，在每个步骤中，系统会从当前预测最不频繁标签的输入集合中随机选择一个输入进行变异。这样可以增加变异体同样预测该最不频繁标签的随机机会，从而确保输入数据集保持均衡。

算法 1 变异输入

```
过程 变异(输入,范围, 变异百分比, 模型)
    输入平坦化 ←获取平坦形状(输入)
    随机索引 ← 随机抽样(输入平坦,变异百分比)
    对于每个 索引 ∈随机索引 执行
        索引处的值 ←范围内的随机浮点数(范围)
        输入[索引]←索引处的值
    结束循环
    返回 将输入形状转换为模型形状(输入，模型)
结束过程
```

变异。 选定的输入经过变异，以生成一个新输入。算法 1 展示了具体步骤：将 变异百分比 的输入值更改为随机浮点值。参数范围 是一个元组，包含用于生成这些值的范围。第一步，将输入展平（重塑为向量），以便对任意形状的输入进行统一的变异处理。接着，从该向量中随机选取一组索引并更改对应的值。最后，将输入转换回原始形状并返回。

变异体接受。 如果满足以下条件，便会将变异体添加到输入语料库中：(i) 通过确保该变异体的预测概率向量与所有其他输入的预测向量之间的欧几里得距离大于距离 阈值，为网络带来新的覆盖率；(ii) 预测出频率最低的标签（以确保输入均衡）。第一个条件改编自 Odena 等人提出的覆盖率度量方法 [24]，该方法使用逻辑值之间的距离来跟踪覆盖率。它确保生成的输入的预测向量彼此相距一定距离（超过某个阈值），从而避免生成大量相似的输入。

BRINC驱动程序和参数。 算法 2展示了BRINC的输入生成过程及其参数。这些参数也汇总于表 4中。我们根据模型的标签数量、希望生成的输入数量（通过最大有效变异体数控制）以及预测结果中期望的多样性（通过距离控制）来调整参数。对于范围，我们选择了列表{(−1, 0), (0, 1), (−1, 1)}。我们也尝试了其他范围，但未观察到显著差异。我们还尝试了最大变异数 的不同取值，发现取值300时效果良好。对于所有输入生成任务，我们都从表 4中建议的参数开始，并在必要时进一步调整参数。

表 4 BRINC 参数汇总。

参数	定义	建议值
变异百分比	每一步发生变化的输入值的百分比	5 %
距离	变异体之间的最小欧几里得距离阈值	0.001
范围	用于约束数字生成的范围集合	(−1, 0), (0, 1), (−1, 1)
最大变异数	每个范围内可进行的变异次数上限	300
最大有效变异体数	每个范围内生成的有效变异体数量上限	500或 1000

算法 2 生成平衡输入

```
过程 genInput(模型,距离, 变异百分比, 范围, 最大变异数, 最大有效变异体数)
    种子 ← 生成种子(模型,范围)
    对于每个 范围 ∈ 范围 执行
        生成计数 ← 0
        没有新数量 ← 0
        当 没有新数量 ≤ 最大变异数 且 已生成数量 ≤ 最大有效变异体数 时执行
            待变异对象，最低频标签 ← 查找最低频标签(种子)
            变异体 ← 执行变异(待变异对象， 范围，变异百分比)
            变异体标签 ← 获取标签(模型，变异体)
            最近邻距离 ← 按预测查找最近邻(变异体，种子)
            如果 最近邻距离 > 距离 且 变异体标签 == 最低频标签 则
                种子。添加(变异体)
                生成数量 ← 生成数量 + 1
                无新增数量 ← 0
            else
                无新增数量 ← 无新增数量 + 1
            结束如果
        结束 while 循环
    结束 for 循环
    返回 种子
结束过程
```

参考文献

[1] Anders Arpteg、Björn Brinne、Luka Crnkovic-Friis和Jan Bosch。《深度学习的软件工程挑战》。载于：*2018*年第*44*届*Euromicro*软件工程与高级应用会议（*SEAA*）。2018年，第50–59页。数字对象标识符：10.1109/SEAA.2018.00018。[2] Georg T. Becker、Daehyun Strobel、Christof Paar和Wayne Burleson。《检测嵌入式系统中的软件盗窃：一种侧信道方法》。载于：*IEEE*信息取证与安全汇刊 7.4（2012），第1144–1154页。数字对象标识符：10.1109/TIFS.2012.2191964。[3] Marcel Berger。《几何学 *I*》。Springer，1987年。ISBN：3-540-11658-3。[4] Mariusz Bojarski、Davide Del Testa、Daniel Dworakowski、Bernhard Firner、Beat Flepp、Prasoon Goyal、Lawrence D. Jackel、Mathew Monfort、Urs Muller、Jiakai Zhang、Xin Zhang、Jake Zhao和Karol Zieba。《端到端自动驾驶汽车学习》。载于：*arXiv*预印本 *arXiv:1604.07316* （2016年）。数字对象标识符：10.48550/ARXIV.1604.07316。[5] 计算相似度——最相关度量指标概览。
h
t
t
p
s://
t
o
wardsdatascience.com/calculate-similarity-the-most-relevant-metrics-in-a-nutshell-9a43564f533e。2019年。（访问日期：2022-03-07）。[6] François Chollet等。*Keras*。https://keras.io。2015年。（访问日期：2023-02-03）。[7] Li Deng、Geoffrey Hinton和Brian Kingsbury。《用于语音识别及相关应用的新型深度神经网络学习方法：综述》。载于：*2013 IEEE*国际声学、语音与信号处理会议。2013年，第8599–8603页。数字对象标识符：10.1109/ICASSP.2013.6639344。[8] Dheeru Dua和Casey Graff。*UCI*机器学习资料库。" http://archive.ics.uci.edu/ml"。2017年。（访问日期：2023-01-19）。

[9] Farima Farmahinifarahani、Vaibhav Saini、Di Yang、Hitesh Sajnani 和 Cristina V Lopes。“代码克隆检测工具的精确率”。载于：*2019*年第*26*届*IEEE*软件分析、演化与再工程国际会议（*SANER*）。2019年，第84–94页。数字对象标识符：10.1109/SANER.2019.8668015。[10] Ronald A. Fisher。“分类学问题中的多重测量之应用”。载于：《优生学年鉴》7.2（1936），第179–188页。数字对象标识符：10.1111/j.1469-1809.1936.tb02137.x。[11] R. Paul Gorman 和 Terrence J. Sejnowski。“经训练用于分类声呐目标的分层网络中隐藏单元的分析”。载于：神经网络1.1（1988），第75–89页。国际标准连续出版物号：0893-6080。数字对象标识符：10.1016/0893-6080(88)90023-8。[12] Fabrice Harel-Canada、Lingxiao Wang、Muhammad Ali Gulzar、Quanquan Gu 和 Miryung Kim。“神经元覆盖率是衡量深度神经网络测试的有效指标吗？”载于：第*28*届*ACM*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。ESEC/FSE 2020。线上活动，美国：计算机协会，2020年，第851–862页。国际标准书号：9781450370431。数字对象标识符：10.1145/3368089.3409754。[13] Harold Hotelling。“两组变量之间的关系”。载于：《生物统计学》28（1936），第321–337页。数字对象标识符：10.2307/2333955。[14] *IncydrTM* 快讯：员工纷纷离职，数据泄露激增。https://www.code42.com/blog/incydr-scoop-data-exposure-jumps-as-employees-head-for-the-doors//。2020年。（访问日期：2022-03-02）。[15] Lingxiao Jiang 和 Zhendong Su。“通过随机测试自动挖掘功能等价代码片段”。载于：第十八届国际软件测试与分析研讨会论文集。ISSTA ’09。美国伊利诺伊州芝加哥：计算机协会，2009年，第81–92页。国际标准书号：9781605583389。数字对象标识符：10.1145/1572272.1572283。[16] Mika Juuti、Sebastian Szyller、Samuel Marchal 和 N. Asokan。“PRADA：防范深度神经网络模型窃取攻击”。载于：*2019*年*IEEE*欧洲安全与隐私研讨会（*EuroS&P*）。2019年，第512–527页。数字对象标识符：10.1109/EuroSP.2019.00044。[17] Simon Kornblith、Mohammad Norouzi、Honglak Lee 和 Geoffrey Hinton。“重新审视神经网络表征的相似性”。载于：第*36*届国际机器学习会议论文集。Kamalika Chaudhuri 和 Ruslan Salakhutdinov 编。第97卷。《机器学习研究论文集》。PMLR，2019年，第3519–3529页。网址：https://proceedings.mlr.press/v97/kornblith19a.html （访问日期：2023-02-04）。[18] Utku Kose、Omer Deperlioglu、Jafar Alzubi 和 Bogdan Patrut。用于医疗决策支持系统的深度学习。Springer，2021年。国际标准书号：978-981-15-6325-6。[19] Aarre Laakso 和 Garrison Cottrell。“内容与聚类分析：评估神经系统中的表征相似性”。载于：《哲学心理学》13.1（2000），第47–76页。数字对象标识符：10.1080/09515080050002726。

[20] Yixuan Li、Jason Yosinski、Jeff Clune、Hod Lipson 和 John Hopcroft。“趋同学习：不同的神经网络会学习到相同的表征吗？”载于：*NIPS 2015*特征提取国际研讨会论文集：现代问题与挑战。PMLR，2015年，第196–212页。网址：https://proceedings.mlr.press/v44/li15convergent.html （访问日期：2023-02-04）。[21] 机器学习攻击系列：窃取模型文件。https://embracethered.com/blog/posts/2020/husky-ai-machine-learning-model-stealing/。2020年。（访问日期：2022-01-10）。[22] Giovanna Menardi 和 Nicola Torelli。“使用不平衡数据训练和评估分类规则”。载于：《数据挖掘与知识发现》28.1（2014），第92–122页。数字对象标识符：10.1007/s10618-012-0295-5。[23] Ari Morcos、Maithra Raghu 和 Samy Bengio。“利用典型相关深入探究神经网络中的表征相似性”。载于：神经信息处理系统进展。第31卷。Curran Associates, Inc.，2018年。网址：
h
t
t
p
s://
proceedings.neurips.cc/paper/2018/file/a7a3d70c6d17a73140918996d03c014f-Paper.pdf
（访问日期：2023-02-04）。[24] Augustus Odena、Catherine Olsson、David Andersen 和 Ian Goodfellow。“TensorFuzz：利用覆盖率引导模糊测试调试神经网络”。载于：第*36*届国际机器学习会议论文集。Kamalika Chaudhuri 和 Ruslan Salakhutdinov 编。第97卷。《机器学习研究论文集》。PMLR，2019年，第4901–4911页。网址：https://proceedings.mlr.press/v97/odena19a.html （访问日期：2023-02-04）。[25] Rangeet Pan 和 Hridesh Rajan。“将深度神经网络分解为模块”。载于：第*28*届*ACM*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。ESEC/FSE 2020。线上活动，美国：计算机协会，2020年，第889–900页。国际标准书号：9781450370431。数字对象标识符：10.1145/3368089.3409668。[26] 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。SOSP ’17。中国上海：计算机协会，2017年，第1–18页。国际标准书号：9781450350853。数字对象标识符：10.1145/3132747.3132785。[27] Hung Viet Pham、Shangshu Qian、Jiannan Wang、Thibaud Lutellier、Jonathan Rosenthal、Lin Tan、Yaoliang Yu 和 Nachiappan Nagappan。“深度学习软件系统训练中的问题与机遇：方差分析”。载于：第*35*届*IEEE/ACM*自动化软件工程国际会议论文集。ASE ’20。线上活动，澳大利亚：计算机协会，2021年，第771–783页。国际标准书号：9781450367684。数字对象标识符：10.1145/3324884.3416545。

[28] Maithra Raghu、Justin Gilmer、Jason Yosinski 和 Jascha Sohl-Dickstein。《SVCCA：用于深入理解和改进深度学习的奇异向量典型相关分析》。载于：*arXiv*预印本 *arXiv:1706.05806v1* (2017)。数字对象标识符：10.48550/arXiv.1706.05806。[29] Robert Nikolai Reith、Thomas Schneider 和 Oleksandr Tkachenko。《高效窃取你的机器学习模型》。载于：第*18*届*ACM*电子社会隐私研讨会论文集。WPES'19。英国伦敦：计算机协会，2019，第198–210页。ISBN：9781450368308。数字对象标识符：10.1145/3338498.3358646。[30] Abhimanyu Roy、Jingyi Sun、Robert Mahoney、Loreto Alonzi、Stephen Adams 和 Peter Beling。《深度学习检测信用卡交易欺诈》。载于：*2018*年系统与信息工程设计研讨会（*SIEDS*）。2018，第129–134页。数字对象标识符：10.1109/SIEDS.2018.8374722。[31] Chanchal K. Roy、James R. Cordy 和 Rainer Koschke。《代码克隆检测技术与工具的比较和评估：一种定性方法》。载于：计算机编程科学74.7（2009），第470–495页。ISSN：0167-6423。数字对象标识符：10.1016/j.scico.2009.02.007。[32] Abdullah Sheneamer 和 Jugal Kalita。《软件克隆检测技术综述》。载于：国际计算机应用期刊 137.10（2016），第1–21页。网址：http://cs.uccs.edu/~jkalita/papers/2016/SheneamerAbdullahIJCA2016.pdf （访问日期：2023-02-04）。[33] Alissa Sherry 和 Robin K. Henson。《人格研究中的典型相关分析：实施与解读——一本易懂的入门指南》。载于：人格评估期刊 84.1（2005）。PMID：15639766，第37–48页。数字对象标识符：10.1207/s15327752jpa8401_09。[34] Manikanta Srikar Yellapragada。《所提出的相似性度量是否也能衡量功能相似性？》硕士论文。纽约大学，2020年。[35] Yiming Tang、Raffi Khatchadourian、Mehdi Bagherzadeh、Rhia Singh、Ajani Stewart 和 Anita Raja。《机器学习系统中的重构与技术债务实证研究》。载于：*2021*年*IEEE/ACM*第*43*届国际软件工程会议（*ICSE*）。2021，第238–250页。数字对象标识符：10.1109/ICSE43902.2021.00033。[36] Florian Tramèr、Fan Zhang、Ari Juels、Michael K Reiter 和 Thomas Ristenpart。《通过预测API窃取机器学习模型》。载于：第*25*届{*USENIX*}安全研讨会（{*USENIX*} *Security 16*。2016，第601–618页。网址：https://reitermk.github.io/papers/2016/USENIX.pdf （访问日期：2023-02-04）。[37] Liwei Wang、Lunjia Hu、Jiayuan Gu、Yue Wu、Zhiqiang Hu、Kun He 和 John Hopcroft。《理解学习表征：不同神经网络在多大程度上学习相同的表征》。载于：*arXiv*预印本 *arXiv:1810.11750*（2018）。数字对象标识符：10.48550/arXiv.1810.11750。

[38] Song Wang、Taiyue Liu 和 Lin Tan。《自动学习用于缺陷预测的语义特征》。载于：*2016*年*IEEE/ACM*第*38*届国际软件工程会议（*ICSE*）。2016，第297–308页。数字对象标识符：10.1145/2884781.2884804。[39] Xinran Wang、Yoon-Chan Jhi、Sencun Zhu 和 Peng Liu。《利用基于系统调用的软件胎记检测软件盗窃》。载于：*2009*年度计算机安全应用会议。2009，第149–158页。数字对象标识符：10.1109/ACSAC.2009.24。[40] Martin White、Michele Tufano、Christopher Vendome 和 Denys Poshyvanyk。《用于代码克隆检测的深度学习代码片段》。载于：*2016*年第*31*届*IEEE/ACM*自动化软件工程国际会议（*ASE*）。IEEE。2016，第87–98页。数字对象标识符：10.1145/2970276.2970326。[41] Martin White、Christopher Vendome、Mario Linares-Vasquez 和 Denys Poshyvanyk。《迈向软件仓库的深度学习》。载于：*2015*年*IEEE/ACM*第*12*届软件仓库挖掘工作会议。2015，第334–345页。数字对象标识符：10.1109/MSR.2015.38。[42] Xiaofei Xie、Lei Ma、Felix Juefei-Xu、Minhui Xue、Hongxu Chen、Yang Liu、Jianjun Zhao、Bo Li、Jianxiong Yin 和 Simon See。《DeepHunter：面向深度神经网络的覆盖率引导模糊测试框架》。载于：第*28*届*ACM SIGSOFT*软件测试与分析国际研讨会论文集。ISSTA 2019。中国北京：计算机协会，2019，第146–157页。ISBN：9781450362245。数字对象标识符：10.1145/3293882.3330579。[43] Xiaofei Xie、Lei Ma、Haijun Wang、Yuekang Li、Yang Liu 和 Xiaohong Li。《DiffChaser：检测深度神经网络的不一致》。载于：第二十八届国际人工智能联合会议论文集，*IJCAI-19*。国际人工智能联合会议组织，2019年7月，第5772–5778页。数字对象标识符：10.24963/ijcai.2019/800。[44] Mengwei Xu、Jiawei Liu、Yuanqiang Liu、Felix Xiaozhu Lin、Yunxin Liu 和 Xuanzhe Liu。《初探智能手机上的深度学习应用》。载于：万维网会议。WWW '19。美国加利福尼亚州旧金山：计算机协会，2019，第2125–2136页。ISBN：9781450366748。数字对象标识符：10.1145/3308558.3313591。

黑箱，白噪声

关于作者

Farima Farmahinifarahani 是加利福尼亚大学尔湾分校软件工程专业的博士毕业生。请通过farimaf@uci.edu联系她。

Cristina V. Lopes是加利福尼亚大学尔湾分校的软件工程教授。请通过lopes@uci.edu联系她。