软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doi: 10.13328/j.cnki.jos.006863]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

前馈神经网络和循环神经网络的鲁棒性验证综述*

刘颖[1,2] , 杨鹏飞[1,4] , 张立军[1,2], 吴志林[1,2] , 冯元[3]

(1 中国科学院软件研究所 国家重点实验室, 北京 100190)

(2 中国科学院大学, 北京 100049)

(3 悉尼科技大学 悉尼 澳大利亚)

(4 人工智能与数字经济广东省实验室, 广州 510320)

通讯作者: 张立军, E-mail: zhanglj@ios.ac.cn

摘 要: 随着智能时代的到来, 部署了深度神经网络的智能系统应用已经渗透到了我们生活的各个方面. 然而, 由于神经网络具有黑盒特性和规模庞大的特点, 其预测结果难以让人完全信服, 当应用于自动驾驶等安全攸关的领域时, 如何保证其安全性仍然是学术界和工业界面临的巨大挑战. 为此, 学术界针对神经网络一种特殊的安全性——鲁棒性展开了研究, 并提出了很多鲁棒性的分析和验证方法. 目前为止, 验证前馈神经网络的方法包括精确验证方法和近似验证方法, 已经发展得比较繁荣; 而对于其他类型的网络, 如循环神经网络的鲁棒性验证研究还处于起步阶段. 本综述回顾了深度神经网络的发展以及部署到日常生活中面临的挑战; 详尽地调研了前馈神经网络和循环神经网络的鲁棒性验证方法, 并对这些验证方法间的内在联系进行了分析和比较; 我们调研了循环神经网络在现实应用场景中的安全性验证方法; 阐明了神经网络鲁棒性验证领域未来可以深入研究的方向.

关键词: 神经网络; 鲁棒性; 人工智能安全; 智能系统; 形式化方法

中图法分类号: TP301

中文引用格式: 刘颖, 杨鹏飞, 张立军, 吴志林, 冯元. 前馈神经网络和循环神经网络的鲁棒性验证综述. 软件学报 http://www.jos.org.cn/1000-9825/6863.htm

英文引用格式: Liu Y, Yang PF, Zhang LJ, Wu ZL, Feng Y. Robustness Verification of Feedforward Neural Networks and Recurrent Neural Networks: A Survey. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/6863.htm

Robustness Verification of Feedforward Neural Networks and Recurrent Neural Networks: A Survey

LIU Ying[1,2] , YANG Peng-Fei[1,4] , ZHANG Li-Jun[1,2], WU Zhi-Lin[1,2] , FENG Yuan[3]

(1 State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

(2 University of Chinese Academy of Sciences, Beijing 100049, China)

(3 University of Technology Sydney, Sydney, Australia)

(4 Pazhou Lab, Guangzhou 510320, China)

Abstract: With the development of artificial intelligence, the application of intelligent systems equipped with deep neural networks have penetrated into every aspect of our life. However, due to the black box and large-scale characteristics, the predictions of the neural networks are difficult to be completely convincing. When applied to safety-critical fields such as autonomous driving, how to ensure the safety of neural networks is still a great challenge for academia and industry. For this reason, academia carried out much research on robustness — a kind of special security of neural networks and proposed many robustness analysis and verification algorithms. The verification algorithms for feed-forward neural networks include precise

*基金项目: 中国科学院青年科学基金项目 (YSBR-040)

收稿时间: 2022-09-05; 修改时间: 2022-10-08; 采用时间: 2022-12-05; jos 在线出版时间: 2022-12-30

algorithms and approximate algorithms, which have been developed relatively prosperously; the verification algorithms for other types of networks, such as recurrent neural networks, are still in the primary stage. This survey reviews the current development of deep neural networks and the challenges of deploying them into our life; exhaustively investigate the robustness verification algorithms of feed-forward neural networks and recurrent neural networks; analyze and compare the intrinsic connection between these algorithms. The verification algorithms of recurrent neural networks in specific application scenarios are listed, and the future research directions in the field of neural network robustness verification are clarified.

Key words: neural network; robustness; artificial intelligence security; intelligent system; formal method

■ 1 引言

人工智能的概念诞生于 1956 年, 彼时人类希望利用机器来模拟人类的思考和决策, 他们发现在处理一些可以利用形式化语言描述的问题时, 电子计算机具有天然的优势. 从此, 人工智能技术开始受到人类的广泛关注, 得到了飞速发展, 其中最具影响力的研究包括 Newell 和 Simon 开发的通用问题求解器 [1], 以及 Weizenbaum 创造出的第一个聊天机器人 Eliza [2].

经过六十多年的积淀, 计算机的运算能力有了突破性的提升, 以机器学习为主的人工智能技术得到了空前的发展. 深度学习作为机器学习的一个分支, 可以利用神经网络对高维数据的特征进行建模, 学习出相应的超参数来预测新问题, 从而应对现实中的复杂问题. 目前为止, 神经网络的应用已经深入地渗透到医疗诊断 [3,4], 语音识别 [5,6], 自动驾驶 [7,8], 教育娱乐 [9,10], 计算机视觉 [11,12], 恶意软件检测 [13,14] 等多个领域. 对于自动驾驶领域而言, 自动驾驶系统的每一个环节都可能需要神经网络组件的支持. 例如在感知过程中, 神经网络会将摄像头拍摄的照片作为输入, 进一步识别出照片中包含的交通信号, 如限速或者右转弯; 基于神经网络或者强化学习的控制器再根据这些信号做出相应的决策 [7]. 在此过程中, 如果用于感知道路状况的神经网络被黑客恶意攻击, 无法正确识别道路上的交通标志, 便很可能导致意外事件的发生 [15]. 即使是训练效果良好的神经网络, 在样本经过扰动或污染后很可能会做出错误预测 [16,17]. 目前已经有许多为神经网络生成对抗性样本的研究 [18,19], 这类对抗性扰动我们人眼很难察觉到, 但是会导致神经网络产生错误的分类. 然而, 想要找到避免此类智能系统被对抗样本攻击或欺骗的方法是相当困难的, 我们很难为神经网络的行为提供一个可靠性保障. 并且由于神经网络的黑盒特性, 人类很难信任它做的决策, 这也是导致难以将神经网络应用于安全攸关系统的主要问题之一.

因此, 研究包含神经网络组件的智能系统的可靠性, 使其行为和决策能被人类理解和认可便尤为重要. 可解释的人工智能 (Explainable Artificial Intelligence, XAI) 和人工智能的安全性验证在这类安全需求的驱动下有了快速的发展. 可解释的人工智能专注于为智能算法 (神经网络等) 的决策附加合理的解释, 例如促使神经网络做出决策的原因或重要特征等等 [20,21] . 人工智能的安全性验证则着重研究神经网络在对抗恶意攻击或扰动的情况下是否仍然能够做出正确的决策. 为构建安全可靠的智能系统, 消除神经网络在安全攸关领域中的应用带来的安全隐患, 神经网络的鲁棒性验证 (Robustness Verification) 这一领域应运而生, 世界各地的学者在近年来对其进行了深入的研究.

本文将主要介绍前馈神经网络 (Feedforward Neural Network, FNN) 和循环神经网络 (Recurrent Neural Network, RNN)[1] 这两种常用神经网络的鲁棒性验证方法, 并对这些验证方法之间的内部联系进行梳理和分析. 我们于第二节形式化地定义神经网络的局部鲁棒性质, 第三节中, 我们从精确算法和近似算法两大类验证算法出发, 详细地介绍了 FNN 的鲁棒性验证方法, 其中精确的验证算法包括基于可满足性模理论、混合整数线性规划这两大类, 近似的验证算法主要包括基于抽象解释、符号传播、凸优化、反例引导的抽象-精化、Lipschitz 等验证方法. 我们从算法特性、支持的激活函数与网络结构、可验证的性质、算法精度和验证规模等方面详细地比较了各类验证算法, 具体分析可以参考表 1. 我们在第四节简要地介绍了 RNN 的结构, 并于第五节详细地分析了各类 RNN 的验证方法, 主要包括基于抽象解释、将 RNN 展开为 FNN 再进行验证、基于自动机提取的验证算法, 并于表 2 分析比较了各类验证算法的验证效果. 第六节介绍了在大型工业场景下,RNN 有关性质的验证方法, 第七节简要介绍了 FNN 局部鲁棒性之外的其他鲁棒性质, 包括全局鲁棒性和概率鲁棒性的定义及验证方法. 我们在第八节阐述了未来在神经网络验证领域可行的发展方向, 包括验证神经网络的其他性质, 研究其他网络结构的鲁棒性, 以及验证大型智能系统的安全性等等. 第九节对全文内容进行了总结.

■ 2 神经网络的鲁棒性质

深度神经网络在结构上由顺序连接的若干层 (Layer) 组成, 其中起始层称为输入层 (Input Layer), 接收神经网络的输入; 终止层称为输出层 (Output Layer), 输出神经网络的运行结果, 其他中间的层称为隐藏层 (Hidden Layer), 它们是神经网络运行的中间结果. 每一个层由若干神经元 (Neuron) 构成, 每一个神经元对应一个浮点数变量. 一般的 FNN 从输出层开始读取输入, 并按相邻两层定义的函数逐层顺序计算, 直到得到输出. 因此, 在数学上神经网络可以建模成一个多元函数 $f:\mathbb{R}^m \to \mathbb{R}^n$, 且它可以表示成神经网络中相邻层之间定义的函数的复合. 神经网络层与层之间的函数大体可分为仿射函数和非线性函数两类, 仿射函数通常在 FNN 中

[1]为了便于描述，我们在后文用 FNN 表示前馈神经网络，RNN 表示循环神经网络。

表 1: FNN 鲁棒性验证方法比较

工具	技术	特性		激活函数		网络结构		性质		精度	规模
		完备	可靠	ReLU	σ	Conv	Res	鲁棒	可达		
Never [22]	抽象解释	✗	✓	✓	✓	✗	✗	✓	✓	较低	较小
AI^2 [23]	抽象解释	✗	✓	✓	✗	✓	✗	✓	✓	中	中
DiffAI [24]	抽象解释	✗	✓	✓	✗	✓	✓	✓	✓	中	大
DeepZ [25]	抽象解释	✗	✓	✓	✓	✓	✓	✓	✓	较高	大
RefineZono [26]	抽象解释	✗	✓	✓	✗	✓	✗	✓	✓	较高	中
DeepPoly [27]	抽象解释	✗	✓	✓	✓	✓	✗	✓	✓	高	大
k-ReLU [28]	抽象解释	✗	✓	✓	✗	✓	✓	✓	✓	高	大
GPUPoly [29]	抽象解释	✗	✓	✓	✓	✓	✓	✓	✓	高	大
Reluval [30]	符号传播	✗	✓	✓	✗	✗	✗	✓	✓	中	较小
Fastlin [31]	符号传播	✗	✓	✓	✗	✗	✗	✓	✓	较高	大
Neurify [32]	符号传播	✗	✓	✓	✗	✓	✗	✓	✓	较高	大
CROWN [33]	符号传播	✗	✓	✓	✓	✗	✗	✓	✓	高	大
Gowal et al. [34]	符号传播	✗	✓	✓	✓	✓	✗	✓	✓	中	大
CNN-Cert [35]	符号传播	✗	✓	✓	✓	✓	✓	✓	✓	高	大
DeepSymbol [36]	符号传播	✗	✓	✓	✓	✓	✗	✓	✓	中	大
CROWN-IBP [37]	符号传播	✗	✓	✓	✓	✓	✗	✓	✓	较高	大
FROWN [38]	符号传播	✗	✓	✓	✓	✓	✗	✓	✓	高	大
Huang et al. [39]	SMT	✓	✓	✓	✓	✗	✓	✓	✓	精确	中
Reluplex [40]	SMT	✓	✓	✓	✗	✗	✗	✓	✓	精确	较小
Planet [41]	SMT	✓	✓	✓	✗	✓	✗	✓	✓	精确	中
DeepSafe [42]	SMT	✓	✗	✓	✗	✗	✗	✓	✓	精确	较小
Marabou [43]	SMT	✓	✓	✓	✗	✓	✗	✓	✓	精确	较小
MIPVerify [44]	MILP	✓	✓	✓	✗	✓	✓	✓	✓	精确	中
SHERLOCK [45]	MILP	✗	✓	✓	✗	✗	✗	✓	✓	高	中
Wong&Kolter [46]	凸优化	✗	✓	✓	✗	✓	✗	✓	✓	中	中
PRIMA [47]	凸优化	✗	✓	✓	✓	✓	✓	✓	✓	高	大
CHARON [48]	CEGAR	δ	✓	✓	✗	✓	✗	✓	✓	高	中
DeepSRGR [49]	CEGAR	✗	✓	✓	✓	✓	✗	✓	✓	高	中
DeepAbstract [50]	CEGAR	✗	✓	✓	✗	✗	✗	✓	✓	较高	中
Elboher et al. [51]	CEGAR	✗	✓	✓	✗	✗	✗	✓	✓	较高	较小
Hein et al. [52]	Lipschitz	✗	✗	✓	✓	✗	✗	✓	✓	较高	中
Clever [53]	Lipschitz	✗	✗	✓	✓	✓	✓	✓	✓	高	大
Fastlip [31]	Lipschitz	✗	✓	✓	✗	✗	✗	✓	✓	较高	大

注: "✓" = 满足相应的性质, "✗" = 不满足. "ReLU" 和 "σ" 分别表示神经网络的激活函数为 ReLU 或 Sigmoid 等其他类型的激活函数, "Conv" 和 "Res" 分别表示卷积神经网络和残差网络, 工具规模 "较小: 能验证神经元数小于一千" , "中: 能验证神经元数大于一千小于一万" , "大: 能验证神经元数大于一万" .

的以全连接层、卷积层或平均值池化层等形式出现, 非线性函数主要包括非线性的激活函数, 如 ReLU、Sigmoid、tanh 等, 以及最大值池化层. 我们最关注的是执行分类任务的神经网络, 此时神经网络的输出向量代表每一种分类的评分或概率, 一般我们选取评分或概率最高的类为分类的结果, 因此执行分类任务的神经网络 $f:\mathbb{R}^m\to\mathbb{R}^n$ 的分类输出可以表示为 $C_f(x)=\arg\max_{1\le i\le n} f(x)_i$, 其中 $f(x)_i$ 代表实数向量 $f(x)$ 第 i 个分量的值.

在介绍神经网络的鲁棒性之前, 我们首先梳理神经网络的安全性质. 安全性质 (Safety Property) 是指在给定的条件下, 一个系统完成规定的功能、不失效的能力, 直观上来看即为系统不会产生不安全的行为. 例如 "智能汽车永远不会撞到行人以及其他障碍物" 就是一个安全性质. 由于这种广义上的安全性质很难被形式化地定义, 本文主要关注将安全性质限制到神经网络中的研究, 即验证神经网络 f 对于给定的输入集合 X, 其对应的输出范围 $f(X)$ 是否都在一个 "安全" 的集合内.$f(X)$ 称为神经网络 f 关于输入 X 的可达集 (Reachable Set), 可达集分析相关的工作包括 [45,63] 等. 我们将神经网络的安全性质形式化地定义为 [23]:

定义 1 (安全性)**.** 给定神经网络 $f:\mathbb{R}^m\to\mathbb{R}^n$, 输入样本集合 $X\subseteq\mathbb{R}^m$ 以及安全范围 $P\subseteq\mathbb{R}^n$, 如果神经网络 f 关于 X 的可达集 $f(X)\subseteq P$, 我们称神经网络 f 在输入 X 下满足性质 P, 这个安全性质记为 (X,P).

表 2: RNN 鲁棒性验证方法比较

工具	技术	特点		激活函数		网络形式		性质		精度	规模
		完备	可靠	ReLU	σ	Van	LSTM	鲁棒	可达		
POPQORN [54]	抽象解释	✗	✓	✓	✓	✓	✓	✓	✓	中	中
Cert-RNN [55]	抽象解释	✗	✓	✓	✓	✓	✓	✓	✓	较高	中
Prover [56]	抽象解释	✗	✓	✓	✓	✓	✓	✓	✓	高	中
RNSVerify [57]	展开为 FNN	✓	✓	✓	✗	✗	✗	✓	✓	精确	较小
Zhang et al. [58]	展开为 FNN	✗	✓	✓	✗	✗	✗	✓	✓	高	中
RnnVerify [59]	展开为 FNN	✗	✓	✓	✓	✓	✗	✓	✓	较高	中
Wang et al. [60]	DFA 提取	✗	✗	✓	✓	✓	✓	✓	✓	中	中
Vengertsev et al. [59]	LTS 提取	✗	✗	✓	✓	✓	✓	✓	✓	高	中
Mayr et al. [61]	DFA 提取	✓	✗	✓	✓	✓	✓	✓	✓	高	中
Khmelnitsky et al. [62]	DFA 提取	✗	✗	✓	✓	✓	✓	✓	✓	较高	较小

注: "✓" = 满足相应的性质, "✗" = 不满足. "ReLU" 和 "σ" 分别表示神经网络的激活函数为 ReLU 或 Sigmoid 等其他类型的激活函数, "Van" 表示 Vanilla 神经网络, 工具规模 "较小: 能验证神经元数小于一千" , "中: 能验证神经元数大于一千小于一万" , "大: 能验证神经元数大于一万" .

验证此类安全问题的算法在给定神经网络 f, 输入集合 X 以及性质 P 之后, 会输出 "Yes"、"No" 或者 "Unknown" 的验证结果, 分别表示验证算法得出神经网络 f 满足、不满足以及无法确定是否满足安全性质 (X, P) 的结论. 如果一个安全性验证算法得出 "Yes" 结论时蕴含神经网络 f 一定满足安全性质 (X, P), 我们称该验证算法是*可靠的* (Sound). 直观上来看, 可靠的算法判断神经网络可达集 $f(X)$ 的*上近似* (Over Approximation) 是否满足给定的性质 P. 反之, 如果验证算法不输出 "YES" 的结论蕴含神经网络 f 一定违反安全性质 (X, P), 我们称该验证算法是*完备的* (Complete). 完备的验证算法判断神经网络可达集 $f(X)$ 的下近似 (Under Approximation) 是否存在违反性质 P 的反例. 验证算法的可靠性和完备性之间的联系可以参考图 1.

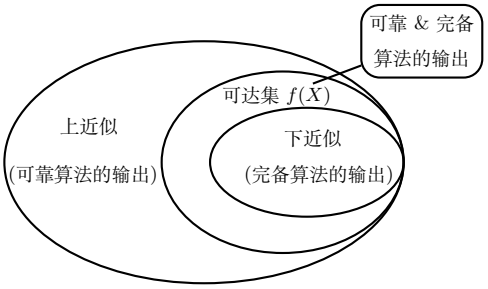

图 1: 可靠性和完备性示意图

在安全性的定义中, 将输入集 X 限制为单个样本 $x_0 \in \mathbb{R}^m$ 周围的一个扰动区域, 我们一般用 l_p 范数下半径为 r 的球来表示, 记为 $X = B_p(x_0, r) \subseteq \mathbb{R}^m$, 将安全范围 P 定义为集合 $P = \{y \in \mathbb{R}^n \mid \arg\max_i y_i = C_f(x_0)\}$, 安全性质 (X, P) 即称为*局部鲁棒性* (Local Robustness). 直观上看, 神经网络的局部鲁棒性是指当输入的信息因为被攻击而产生一个有限范围内的小扰动时, 神经网络仍然能够保持正确的输入-输出关系的特性, 即对扰动后的输入仍然可以做出正确的输出决策. 神经网络的局部鲁棒性可以形式化描述如下:

定义 2 (局部鲁棒性)**.** *给定一个神经网络 f 和输入样本 x_0, 以及样本 x_0 半径为 r 的扰动区域 $B_p(x_0, r)$, 如果性质*

$$\forall x \in B_p(x_0, r), C_f(x) = C_f(x_0)$$

成立, 我们称神经网络 f 关于样本 x_0 在扰动区域 $B_p(x_0, r)$ 内具有局部鲁棒性 (Local Robustness).

在本综述中, 我们主要关注神经网络的鲁棒性质, 如果没有特别说明, 后文关于神经网络安全性的验证方法均为局部鲁棒性的验证.

在自然语言处理和语音识别等领域, 应用最为广泛的是 RNN. 由于 RNN 比 FNN 的结构更加复杂, 定义输入序列的扰动空间也更加困难, 因此关于 RNN 鲁棒性验证的工作目前还比较少, 且多数 RNN 的验证工作借鉴了 FNN 的验证思想. 本文将首先回顾 FNN 鲁棒性验证的相关内容, 分析和比较 FNN 和 RNN 验证方法之间的内部联系, 并就 RNN 验证面临的挑战展开讨论.

■ 3 FNN 验证的研究现状

目前针对 FNN 的验证方法主要分为两大类: 一类是基于可满足性模理论, 混合整数线性规划等优化理论的精确验证方法; 另一类主要是基于凸优化, 抽象解释等近似验证方法. 在图 2 中, 我们挑选出每一类中比较有影响力的验证方法进行对比, 接下来将详细介绍具体的验证方法.

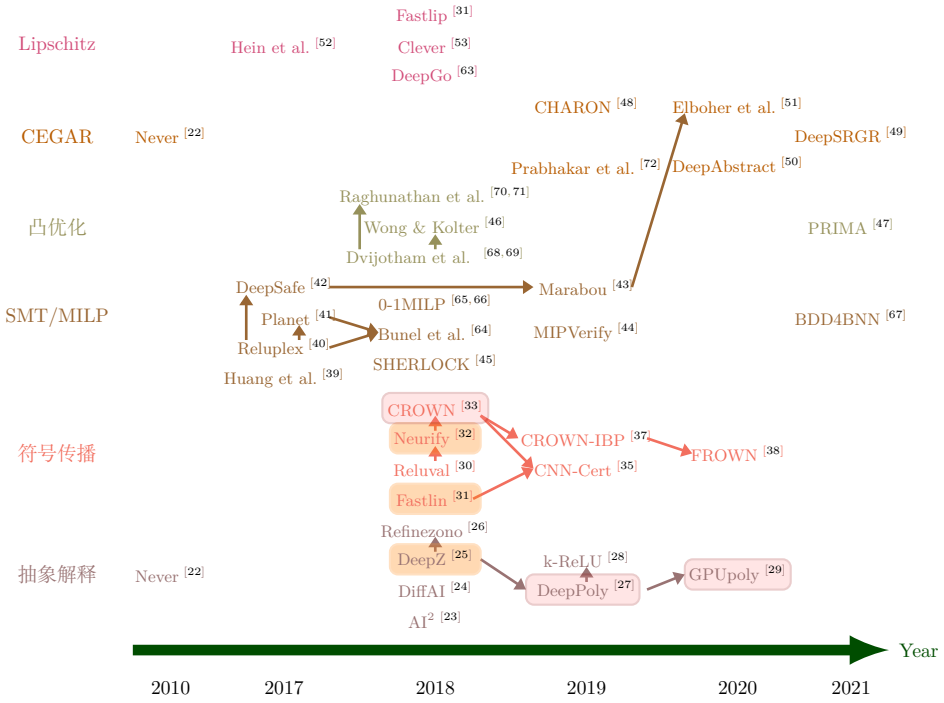

图 2: 主要的 FNN 验证方法对比, 其中不同颜色表示不同的验证方法类别, 箭头表示被指向的工具对箭头出发的工具有提升或者是由其借鉴而来, 相同颜色的矩形方框中验证工具精度相同.

■ 3.1 精确验证方法

由于神经网络往往规模很大, 并且具有非凸的结构, 所以验证神经网络的复杂度很高. 文献 [40, 73] 强调即使验证一个很简单的性质也是一个 NP 完全问题. 目前,FNN 鲁棒性的精确验证方法主要包括基于可满足性模理论 (Satisfiability Modulo Theory, SMT) 以及混合整数线性规划 (Mixed Integer Linear Programming, MILP) 这两大类. 基于 SMT 的验证方法将神经网络鲁棒性验证问题编码为可满足性问题, 然后利用 SMT 求解器进行验证. 而基于 MILP 的验证方法是将神经网络的仿射变换以及 ReLU 激活函数编码为 MILP 公式 (常用的编码方法包括大 M 法[2] [74] 等), 并精确地求解出 FNN 的可达集, 再判定该可达集是否满足给定的安全性质.

■ 3.1.1 基于 SMT 的验证方法

比较早期的验证方法包括 Bastani [75] 等人的工作, 他们为神经网络定义了两种不同的鲁棒性, 并利用线性约束来编码整个神经网络的鲁棒性验证问题, 从而将验证问题转化为可满足性问题. 他们通过将输入限制在 ReLU 函数只取正值或 0 的区域, 将原来的约束松弛为一个凸约束问题, 从而进行快速求解.Fawzi [76] 等人提出了一个理论框架来分析分类器抗扰动的鲁棒性, 并在理论上给出了鲁棒性的一般上界.Scheibler 等人 [77] 考虑了倒立摆的模型检验问题, 倒立摆的控制器是一个具有非线性激活函数的神经网络. 他们使用 SMT 求解器 iSAT3 进行求解, 实验表明对于只具有 26 个节点的神经网络, 解决这个验证问题已经非常具有挑战性.Narodytska 等人 [78] 提出二值化神经网络 (Binarized Neural Network,BNN) 可以编码为布尔公式, 并利用 SAT 求解器对其进行鲁棒性验证, 这也是首次采用精确的布尔编码来验证神经网络的方法, 实验结果显示此验证方法可以处理规模为几百个神经元的神经网络.

[2]大 M 法是一种使用单纯形算法求解线性规划问题的方法, 大 M 法引入了剩余变量和人工变量, 此处的 “大 M” 表示与人工变量相关的一个非常大的数, 用字母 M 表示.

以上几种验证神经网络的方法需要对网络结构或性质做出显著的简化, 例如 Bastani [75] 的工作中仅考虑所有 ReLU 函数都固定在激活或非激活状态的小输入区域, 只能验证所需性质的近似值. 基于 SMT 的验证方法需要求解大规模的约束问题, 因此这类将鲁棒性验证问题编码为 SMT 公式的验证方法在可拓展性上具有明显的瓶颈.

Huang [39] 等人考虑了外部条件变化 (例如划痕和不同的天气条件等) 对神经网络分类器造成的影响. 相比于前面的工作, 他们基于 SMT 提出了一种更加灵活的鲁棒性验证方法, 对于一个特定的输入点 x, 采用离散化的技术将 x 邻域的无限集简化为有限点集, 从而对扰动区域进行穷举并逐层传播. 实验表明这种离散化的验证技术甚至可以在 VGG16 等大型神经网络上进行验证实验.

Katz [40] 等人提出了 Reluplex, 将单纯形法 (Simplex Algorithm) 加以拓展得到一个能处理 ReLU 激活函数的 SMT 求解器. 具体地, 他们将 FNN 的仿射变换编码为原子命题的合取形式, 将被 ReLU 激活的神经元 v 拆分成 v_b 和 v_f, 分别连接前一层和后一层神经元. 他们将神经元分类为基变量和非基变量, 并为 SMT 求解过程定义了一系列更新规则, 直到求解算法终止, 最后即可求出违反性质的反例, 或证明神经网络的安全性.

Katz 等人在 Reluplex 工作中考虑的鲁棒性验证问题为无人机防碰撞系统 ACAS Xu [79] 中的若干安全性质,ACAS Xu [79] 系列网络包含 300 个隐藏神经元.Reluplex 仅支持 ReLU 激活函数, 并且这种完备的验证方法具有精确验证效率低下的问题. 因此, 这类基于 SMT 求解的验证方法无法直接应用到现实中包含数十万个神经元的智能系统 [80,81].Katz 等人随后 [81] 介绍了对抗鲁棒性以及其他种类的鲁棒性, 考虑了验证工业界神经网络面临的问题, 并针对自动驾驶系统中 FNN 控制器的安全性问题, 提出了结合 Reluplex 进行验证的未来规划.

为了验证含有最大池化 (Maxpool) 节点的神经网络,Ehlers 等人提出了一种基于 SMT 求解和线性规划的精确验证工具 Planet [41]. 他们通过为 ReLU 激活函数和 Maxpool 函数添加线性逼近来线性地刻画神经网络的整体行为, 并通过添加冲突字句寻找不可行子集以减少搜索空间, 从而提升搜索效率, 实验表明 Planet 可以验证具有上千个神经元的神经网络.

Xiang 等人 [82] 通过可达集分析来验证多层感知机 (Multi-Layer Perceptron, MLP) 的安全问题, 这是一种结构简单的神经网络. 他们引入了 “最大灵敏度” 的概念, 并通过求解凸优化问题来计算具有单调激活函数的 MLP 的最大灵敏度, 将神经网络的可达集估计问题转化为一系列优化问题并进行求解. 最后他们基于可达集估计的结果开发出一种可靠的自动验证算法.

Gopinath 等人提出了一种自动识别输入空间安全区域的完备验证方法 DeepSafe [42], 验证神经网络的对抗鲁棒性. 该方法通过数据引导的聚类识别出 “候选” 安全区域, 然后确认这些区域的安全性, 或者提供证明它们不安全的反例. 他们在 MNIST 数据集和 ACAS Xu 网络上评估发现 DeepSafe 可以准确地确定神经网络的安全输入区域.

Katz 等人在 Reluplex [40] 的基础之上又提出了 Marabou [43], 支持具有任意线性激活函数的 FNN 和卷积神经网络 (Convolutional Neural Network, CNN) 的验证, 在应用广泛性上相对于前身 Reluplex 有了新的突破.Marabou 是基于 SMT 的验证框架, 其内置了基于单纯形法的线性规划求解器, 并设置了一个很小的超时阈值, 当求解时间达到该阈值时, 求解器将验证过程划分为更简单的子过程, 在不同节点上并行地运行子查询. 相比于 Reluplex 在每次迭代中都要将线性规划问题翻译成外部求解器 GLPK 接受的输入, 并从中提取相应的结果,Marabou 大大提升了线性规划问题的求解效率. 在 ACAS Xu 网络上的实验表明,Marabou 相比于 Reluplex 和 Planet 等工具在验证效率上具有明显的优势.

Zhang 等人提出了 BNN 的分析框架 BDD4BNN [67]. 他们将 BNN 视为黑盒, 将 BNN 和输入区域编码为可以表示为布尔函数的二进制决策图 (Binary Decision Diagrams, BDD) 并基此设计了 BNN 的鲁棒性分析框架, 并且支持可解释性分析, 实验表明 BDD4BNN 框架可以验证上千个神经元的神经网络.

基于 SMT 的验证方法由于编码过程时间代价高昂, 在约束空间规模庞大的情形下, 通常只能处理神经元数量在几千以内的神经网络, 并且 SMT 求解器具有难以求解非线性问题的局限性, 基于 SMT 的鲁棒性验证方法往往只能处理 ReLU 或者其他具有线性激活函数的神经网络.

■ 3.1.2 基于 MILP 的验证方法

神经网络可以被编码为 MILP 公式. 在较为早期的研究中已经有了相关的工作, 例如 Fischetti 和 Jo 提出的 0-1 MILP 框架 [65,66], 用连续变量来表示每一个神经元的输出值, 离散变量表示 ReLU 激活过程; Lomuscio 和 Maganti [83] 利用大 M 法将 FNN 编码为 MILP 公式, 并且将 FNN 的可达性问题编码为 LP 问题进行求解等等.

Cheng [84] 等人开发了 MILP 的启发式算法. 在利用大 M 法将神经网络编码为 MILP 问题的基础上采用了一些优化方法, 例如在用 MILP 求解神经元范围时, 同时考虑前几层的神经元范围, 将全局 MILP 问题分割为若干个更小的子问题; 为不同隐藏层的神经元设置编码优先级等. 以上优化方法显著地减少了 MILP 求解器的运行时间, 与普通的 MILP 编码过程相比, 他们的优化策略在一些特殊的 MNIST 基准数据集上甚至可以使得运行效率提升两个数量级.

Dutta 等人提出了一种有效的范围估计算法 SHERLOCK [45], 将 MILP 全局优化和基于局部搜索的局部优化之间进行迭代, 通

过修剪次优节点, 有效地减少了搜索激活神经元的时间.SHERLOCK 很好地利用了神经网络的局部连续性和可微特性, 同时又避免了局部搜索可能陷入局部最小值的困境, 在效率上也优于传统的完全基于 SMT 或者 MILP 的验证方法. 实验显示 SHERLOCK 能求解具有 6000 个神经元的神经网络输出范围. 该算法专注于求解以 ReLU 作为激活函数的 FNN 的输出范围, 而对于其他类型的激活函数, 例如 Sigmoid 和 tanh,SHERLOCK 方法不具备求解能力.

需要注意的是,SHERLOCK 虽然利用了 MILP 技术求解神经网络的输出范围, 但是在编码 ReLU 函数的过程中添加了一个微小的松弛, 因此 SHERLOCK 得到的是神经网络输出的上近似. 为了便于归类, 我们仍于此处介绍 SHERLOCK 方法.

Bunel 等人 [64] 利用分支限界等优化方法, 将 Reluplex、Planet 等基于 SMT、MILP 的主流验证方法纳入到一个统一的验证框架中, 并强调 Reluplex、Planet 在该框架中只是一个特例. 他们另一个重要贡献在于从现有的神经网络验证文献中收集并拓展得到一个新的基准数据集 (Benchmark) 并命名为 PCAMNIST, 并在此数据集上将分支限界处理后的 Reluplex 和 Planet 等算法进行比较, 同时通过在更小的区域上精化 ReLU 的上下界, 采取分支策略分割输入域等优化方法, 对以上验证算法提出了可能的改进, 大大提高了算法效率.

利用 MILP 验证神经网络的比较有影响力的工作包括 Vincent Tjeng 等人提出的 MIPVerify [44], 他们将神经网络编码为 MILP 进行求解, 并设计了一些剪枝方法缩小搜索空间, 大大提高了求解效率, 已经可以验证中等规模的神经网络, 相对于之前的基于 SMT 的验证方法实现了不小的突破, 甚至可以拓展到带有卷积层和残差层的网络结构中.

虽然基于 SMT 和 MILP 的验证方法能够精确地确定神经网络的可达集, 并且近年来在神经网络鲁棒性验证领域中取得了很大的进展, 但是由于此类方法具有编码复杂、时间代价大的缺点, 能验证的网络规模比较小, 拓展到具有上万个神经元的神经网络仍然是一个很有挑战的问题. 同时, 在许多现实情况下, 验证神经网络的性质并不需要得出其确切的可达集, 即如果可达集的上近似满足某安全性质, 则该可达集必然也满足这一性质. 因此从 2018 年左右开始, 更高效的近似验证方法逐渐兴起.

■ 3.2 近似验证方法

FNN 的近似验证方法相比于为数不多的可靠且完备的精确验证方法来说, 正处于快速发展的状态, 主要的近似验证方法包括基于抽象解释、符号传播、Lipschitz、凸优化以及线性松弛的验证方法等.

■ 3.2.1 抽象解释

基于抽象解释验证神经网络的方法主要是利用抽象域在神经网络中逐层传播, 最后得到神经网络可达集的上近似, 并判断该上近似是否满足给定的安全性质. 最早的神经网络验证方法是由 Pulina 和 Tacchella 提出的 Never [22] 验证框架, 这也是最早的利用抽象解释验证神经网络的框架. 他们用区间定义神经网络 f 的安全性, 即对任意给定的输入 x, 分析输出 $f(x) \in [l, u]$ 是否成立, 其中 l 和 u 表示给定的 “安全阈值” . 他们将 SMT 求解器与反例引导的抽象-精化方法结合起来, 对 MLP 这种特殊的网络结构进行安全性验证. 他们利用区间表示对每一层神经元的范围进行抽象, 并为非线性的 Sigmoid 激活函数添加若干个上近似的区间约束, 从而为神经网络的每一层范围添加可靠的 Box 抽象域. 当传递到最后一层发现虚假反例时, 会触发神经网络进行自动修复, 对激活函数进行更精细的划分, 提高验证精度. 虽然只成功地验证了包含 6 个神经元的神经网络, 但该方法首次将形式化验证和抽象解释的思想应用到神经网络验证中, 具有开创性的历史意义. 他们在随后的工作中 [85,86] 比较了 SMT 求解器的性能对 MLP 验证效率的影响, 并分析发现虽然当前最先进的 SMT 求解器有能力解决 MLP 验证领域中的一些重要问题, 但是对实际规模和细粒度抽象网络的验证, 仍然是一个长期的开放性挑战.

为了将抽象解释的验证方法拓展到规模更大, 结构更一般的神经网络中,Gehr 等人提出了第一个可靠、可拓展的 ReLU 神经网络验证框架 AI^2 [23]. 他们利用区间抽象域和 Zonotope 抽象域 [87] 为扰动之后的输入添加一个可靠的边界, 并构造了基于 Zonotope 的抽象转换器来执行神经网络中的各种非线性操作, 包括 ReLU 激活函数和最大值池化层, 最后验证输出的 Zonotope 是否满足给定的性质. 由于 Zonotope 本身形状简单, 对真实可行域的松弛在有些验证场景中过于宽松, 并且在传播过程中会逐层放大误差, 其验证精度有所局限.

Mirman 等人提出了一种基于混合 Zonotope 的鲁棒性训练和验证方法 DiffAI [24]. 混合 Zonotope 是指在 Box 抽象域的基础之上添加一些形如 Zonotope 的偏移量, 作为一个表达能力等同于 Zonotope 的新抽象域. 他们在此基础上定义了一个损失函数来训练神经网络, 并通过梯度下降等技术进行优化, 通过这种方法训练得到的网络具有良好的鲁棒性和可拓展性.Mirman 等人在后续的工作 [88] 中引入了两个新概念:(i) 用于微调抽象精度和可扩展性的抽象层;(ii) 用一种灵活的领域专用语言 (Domain Specific Language,DSL) 来描述神经网络中抽象损失与具体损失相结合的目标函数, 并提出为大型残差网络 ResNet 提高鲁棒性的训练方法.

Singh 等人提出的 DeepZ [25] 框架进一步解决了 AI^2 的可拓展性问题, 适用于 ReLU,Sigmoid,tanh 等多种激活函数和前馈、卷积以及残差网络等多种神经网络结构. 类似于 AI^2,DeepZ 也采用了 Zonotope 抽象域对神经网络的扰动区域进行松弛, 但 DeepZ 利用面积最小化原理为 ReLU,Sigmoid,tanh 激活函数提供了一种更紧的可靠的抽象方式, 并利用 Zonotope 在神经网络中的传播求解

最后一层神经元的范围, 从而验证网络的鲁棒性.

以上验证方法都是利用 Zonotope 抽象域在神经网络中进行传播, 是当前神经网络鲁棒性验证领域比较主流的方法. 但是这种不完备的松弛方法会导致在每一个神经元执行激活操作时引入比较大的误差, 并且在传播过程中也会造成精度损失. 因此该抽象域在松弛精度上存在缺陷, 最后一层神经元的范围可能会被放大成百上千倍.

Singh 等人提出了 DeepPoly [27], 进一步提升了抽象域的精度.DeepPoly 基于线性不等式, 为每个神经元添加两个符号界约束 $a_i^{\leqslant}$, $a_i^{\geqslant}$ 和两个数值界约束 l_i, u_i 作为抽象域, 并提出了两种三角形松弛方法, 为 ReLU 激活函数添加一组松弛边界,DeepPoly 通过反向传播每一个神经元的符号界到输入层, 再代入输入层神经元的范围求解出神经元的数值界. 相较于前面几种验证方法,DeepPoly 在验证效率和精度上都具有明显的优势.

为了解决 Zonotope 抽象域带来的精度损失问题, 很多新的验证方法也在此基础上应运而生.Singh 等人提出了 RefineZono [26], 将 Zonotope 抽象域和 MILP、LP 结合起来验证神经网络. 他们用 Zonotope 抽象域求解得到每一层神经元的范围, 再利用 MILP 求解器对神经元的范围进行精化, 从而得到更紧的神经元边界, 进一步往前传递分析神经元的范围.Refinezono 得到了比 DeepPoly 更高的验证精度, 但是 MILP 的应用导致运行时间上存在一些劣势.

为了更进一步地提升传递效率,Müller 等人将区间分析的高效传播和 DeepPoly 的高精度这两个优势结合起来, 并利用了 GPU 的并行性提出了 GPUPoly [29] 框架来验证神经网络. 在 DeepPoly 框架中, 得到神经元的数值上下界需要反向传播到输入层, 因而效率低下,GPUPoly 利用区间分析来代替这一反向传播过程: 在求解每个神经元的数值界时, 他们为神经元的边界添加一个可靠的区间抽象, 如果当前神经元处于确定的激活或者未激活状态, 就省略反向传播的过程, 反之则利用 DeepPoly 进行反向传播得到神经元的上下界. 得益于高并行的设计,GPUPoly 的精度等同于 DeepPoly, 但是验证速度比 DeepPoly 快 35 倍到 170 倍.

此外,Singh 等人还在 DeepPoly 抽象域的基础上提出一种带参数的精化框架 k-ReLU [28], 在求解每一层神经元的符号界时, 同时考虑 k 个神经元的约束, 并为其添加一个系数为 0 或 ± 1 的八面体 (Octagon) 约束.k-ReLU 巧妙地解决了 DeepPoly 在求解神经元约束时只能考虑单个神经元, 而忽略了神经元之间的相互联系的问题. 在实验中也体现出 k-ReLU 的验证精度相比于 DeepPoly 有明显的提升. 但由于 k-ReLU 框架中新增的约束个数是 $O(2^k)$ 的, 时间复杂度随 k 呈指数的速度增长, 因此 k 的取值通常较小, 一般取为 2 或 3, 最大可达到 7.

上述基于抽象解释的验证方法以较高的效率为神经网络添加一个可靠的输出范围, 并且可以拓展到中等规模的神经网络. 神经网络真实的可达集是真包含于抽象解释得到的输出范围的, 而当抽象域过于松弛的时候, 很容易产生虚假反例, 即, 若抽象解释得到的范围不满足给定的性质, 便很难证验证真实的输出是否满足该性质, 这也是验证算法可能会得到 “Unknown” 的原因. 因此产生了反例引导的抽象-精化框架来验证神经网络.

■ 3.2.2 对网络结构抽象/反例引导的抽象-精化

当验证算法对神经网络的抽象过于松弛时, 输出范围中可能含有许多不属于安全范围内的 “反例” . 如果这些 “反例” 对应的原始输入经过神经网络传递之后得到的真实输出是属于安全范围的, 我们将这些 “反例” 称为 “假反例”; 反之, 若这些 “反例” 的原始输入对应的输出也在安全范围之外, 则称为 “真反例” , 其直观含义如下图 3 所示. 基于反例引导的抽象-精化方法 (Counterexample Guided Abstraction Refinement, CEGAR) 通过验证抽象模型得到的假反例, 将过于松弛的抽象进行精化, 从而迭代地得到一个更精确的验证结果.

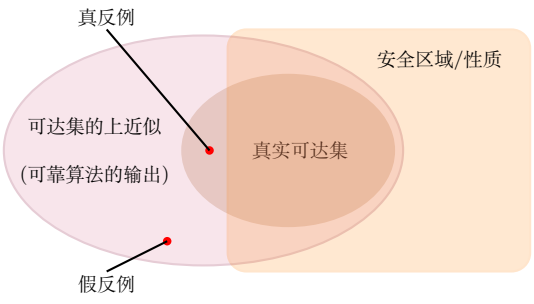

图 3: 真反例和假反例示意图

反例引导的抽象-精化方法通常会与抽象解释等松弛方法结合起来验证神经网络的鲁棒性. 其原理为: 利用抽象解释验证神经网络时, 如果因为对可达集松弛过大而无法验证给定的安全性质, 就会触发精化方法得到更紧的抽象范围或者更细的划分. 例如前面介

绍的 Never [22] 框架, 当利用抽象解释在输出层的上近似中发现反例时, 会触发神经网络的自动修复, 这是比较早期的将抽象与精化结合起来验证神经网络的方法.

除了与抽象解释相结合, 精化方法还可以与其他的松弛方式相结合验证网络, 例如对网络结构进行抽象. 对网络结构进行抽象的方法主要通过删除、合并多余的神经元来减小神经网络的规模, 并得到神经网络输出范围的上近似. 然而改变神经网络的结构很可能因为过度放大权重导致输出范围过度松弛, 此时基于反例引导的抽象-精化方法可以得到更紧、更准确的输出范围.

早在 2015 年,Srinivas 等人 [89] 提出可以通过移除神经元简化神经网络的结构, 同时保持和原始神经网络相似的性能. 他们计算传入某一层神经元的权重之间的相似性度量, 然后通过合并那些最相似的权重来移除 “多余” 的神经元, 并对移除神经元的数量和准确性之间的关系进行了分析. 类似地,Zhong [90] 等人通过聚类对权重进行合并, 从而压缩神经网络的规模.Han [91] 等人引入了 “深度压缩” 方法———一个三阶段的工作流程: 修剪、量化权重和霍夫曼编码, 在不影响神经网络准确性的条件下压缩神经网络的规模, 使得神经网络的存储需求减少数十倍. 以上通过对网络结构进行修剪、抽象来压缩神经网络的方法, 目的是减少神经网络的内存消耗, 并不对神经网络的安全性进行验证.

关于利用抽象网络结构来验证神经网络的安全性的工作,Prabhakar [72] 等人考虑了 FNN 的可达性分析问题. 他们将神经网络的权重矩阵和偏差向量分别抽象为区间, 形成区间神经网络 (Interval Neural network, INN), 示意图如图 4 所示. 具体地, 他们将同一隐藏层的两个神经元合并到一起, 与这两个神经元有关的权重和偏差将拓展为区间的形式, 该区间是合并之前权重和偏差的凸包, 从而确保合并后网络的输出为原神经网络可达集的上近似. 他们利用 MILP 对最终得到的区间神经网络进行编码并求解输出范围. 通过这种方法将神经网络抽象为一个结构更简单、神经元数量更少的区间神经网络, 实现计算时间和输出精度之间的权衡. 由于被合并神经元的选取对最终的精度影响很大, 因此如何通过选择合适的神经元提高抽象神经网络的精度, 以及当神经网络过于松弛的时候如何采取精化措施来修正抽象过程是这项工作的难点.

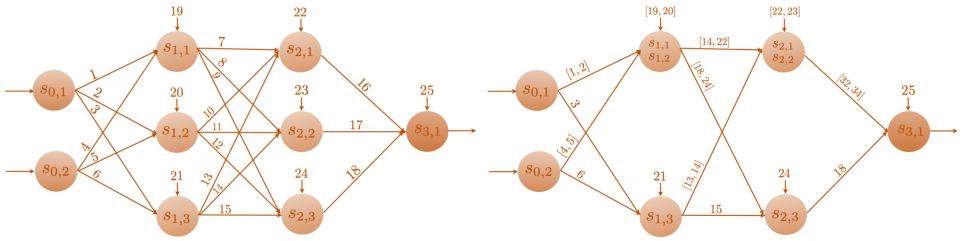

图 4: 通过将普通 FNN 的权重矩阵元素抽象为区间表示可以减少神经元数量. 左侧为抽象前的神经网络, 抽象后变为右侧区间神经网络, 神经元上方的数字表示偏差.

可靠的 Zonotope 抽象域可以确定神经网络输出范围的上近似, 但是由于抽象域过于松弛, 会经常导致假反例的产生. 另一方面, 基于优化寻找反例的过程可以高效地寻找到对抗性扰动, 但如果没有搜索到反例, 则很难证明神经网络是鲁棒的.Anderson 等人将这两种策略的优势相结合, 提出了鲁棒性验证方法 CHARON [48] , 在验证的同时寻找反例, 又利用找到的反例对抽象过程进行精化. 这是一种可靠且 δ-完备的验证方法. 其中 δ-完备是指如果一个性质 P 被反例 x_* 证伪, 则 x_* 的 δ 邻域内必然存在真正的反例. 另外他们还提出了自动学习验证策略的方法, 可以为反例搜索提供指导.

Ashok 等人提出一个适用于全连接 FNN 的抽象框架 DeepAbstract [50],DeepAbstract 基于启发式的 k-means 聚类方法, 将在输入上表现相似 (I/O 相似) 的神经元进行聚类, 然后在同一层中合并这些具有 I/O 相似性的神经元, 从而减小神经元的规模, 提高验证效率. 与纯粹的压缩技术相比, 他们的抽象提供了原始神经元和抽象神经元之间的一个映射, 便于将抽象网络的鲁棒性验证结果转移到原始神经网络上, 从而得到原始神经网络的验证结果. 在实验中他们展示了如何利用 DeepPoly 验证抽象网络, 并迁移到对原始神经网络的验证中.

Elboher 等人 [51] 通过对神经元的性质进行分类, 将每个神经元分为 pos/neg 和 inc/dec 四种类型, 并迭代地将类别相同的神经元进行合并, 减小神经网络规模. 他们利用 Marabou 验证工具来验证抽象之后的网络, 并在抽象过于松弛的情况下采用反例引导的抽象-精化的方法, 对神经网络结构的抽象进行细化. 这类方法将抽象带来的高效与反例引导的抽象-精化带来的高精度相结合, 很好的实现了神经网络验证精度和效率上的平衡.

在最近的工作中,Yang 等人在 DeepPoly 抽象域的基础上, 提出了伪区域指导的神经网络验证精化技术 DeepSRGR [49], 该方法在验证精度需求较高,DeepPoly 抽象域难以成功验证的场合, 可以利用通过性质取反构造的伪区域, 精化关键变量在伪区域语义中的上下界, 从而大幅提升 DeepPoly 抽象域的验证精度. 在该方法的框架下, 每次精化的过程并不会生成具体的反例, 用于指导精化

的是伪区域的约束和伪区域语义中关键变量更好的界.

基于反例引导的抽象-精化方法还可用于训练和验证深度强化学习 (Deep Reinforcement Learning, DRL) 系统.Peng 等人提出的 TRAINIFY 框架 [92] 可以在抽象的状态空间下训练出一个 DRL 系统, 当训练得到的 DRL 系统产生某种安全性质的反例时,TRAINIFY 根据该反例优化抽象状态空间, 并在更精细的抽象状态上再次训练 DRL 系统. 通过重复以上过程就可以训练出一个精度高、鲁棒性好的 DRL 系统.

上述基于网络结构抽象和反例引导的抽象-精化方法通过将表现相似的神经元进行合并或删除, 减小网络规模, 然后利用现有的方法 (SMT、MILP 等) 来验证神经网络的性质, 达到提高验证效率的目的, 最后得到的输出为神经网络可达集的上近似. 当对神经网络的结构抽象过于粗糙时, 此类方法会结合反例对抽象进行精化, 提高验证精度. 但是如何设定抽象神经元的标准以及如何达到精度与效率的平衡是此类方法的难点, 并且在对网络结构进行抽象的过程中需要对神经网络重新编码, 时间代价比较高.

■ 3.2.3 基于符号传播的验证方法

基于符号传播的验证方法主要通过区间表示、线性松弛等方式, 将神经网络每一层之间的迁移关系表示为一系列符号约束并逐层传递, 最后得到输出层与输入层之间的约束关系, 再将输入层的数值边界代入符号约束关系中, 得到输出范围. 由于对 ReLU 等激活函数采取了线性松弛, 通常情况下, 此类基于符号传播的方法得到的输出范围也是神经网络可达集的上近似.

Gowal [34,93] 等人提出了由区间计算衍生而来的区间边界传播 (Interval Bound Propagation,IBP), 他们展示了如何利用 IBP 来训练可证明鲁棒性的大型神经网络, 这是一种可以用于训练分类器的不完备算法, 其主要思想是为神经网络中每一层神经元的多面体约束添加一个轴平行的上近似边界框 (本质上类似于 Box 约束), 该边界框总是包含真实的多面体约束. 此方法相比于 Wong [94] 等人提出的基于对抗性训练的验证方法在扰动半径增大时有明显的竞争力.

Weng 等人提出了一种与 DeepZ 类似的验证框架 Fastlin [31], 可以求解 l_p 扰动下神经网络可达集的上近似.Fastlin 的核心思想是利用仿射变换和平行四边形松弛导出最后一层神经元的符号上下界, 逐层反向传播到第一层并将第一层神经元的范围代入, 即可得到最后一层神经元的范围. 由于 Fastlin 和 DeepZ 对于非确定神经元的 ReLU 激活都采用了相同的 Zonotope 松弛, 如果神经网络的非线性单元仅包含 ReLU, 输入是 l_∞ 范数下的一个扰动邻域, 那么由 Fastlin 和 DeepZ 得到的输出范围是相同的, 即这两种方法在 l_∞ 范数下的精度是等价的, 具体的证明如下引理 1. 而 DeepZ 采用的 Zonotope 抽象域传播相比于 Fastlin 采用的带线性约束的反向传播效率会更高, 因此实验结果显示 DeepZ 比 Fastlin 的效率高 2.5 倍以上.

引理 1 (Fastlin 和 DeepZ 的等价性)**.** *激活函数为 ReLU 的神经网络 f 的输入为 l_∞ 范数下的 Zonotope 时, 用 Fastlin 和 DeepZ 得到的输出范围是等价的.*

证明. 设神经网络 f 的权重矩阵和偏差向量分别为 $\{W_i \in \mathbb{R}^{l_{i+1}\times l_i} \mid i=1,\cdots,n\}$ 和 $\{b_i \in \mathbb{R}^{l_{i+1}} \mid i=1,\cdots,n\}$, 激活函数为 $\{\sigma_i \in \mathbb{R}^{l_{i+1}} \mid i=1,\cdots,n\}$, 以及该激活函数在 Fastlin 和 DeepZ 框架下的松弛为 $\{\sigma_i' \in \mathbb{R}^{l_{i+1}} \mid i=1,\cdots,n\}$, 输入范围为 $X_1 = B_\infty(x,r) \subseteq \mathbb{R}^{l_1}$, 我们用 $\{X_i \subseteq \mathbb{R}^{l_i} \mid i=1,\ldots,n+1\}$ 表示输入 X_1 传递到每一层神经元之后得到的像. 则神经网络在 Fastlin 和 DeepZ 框架下的输出范围即

$$f(X_1) = W_n \cdot \sigma'_{n-1}(W_{n-1} \cdot \sigma'_{n-2}(\cdots \sigma'_1(W_1 \cdot X_1 + b_1)\cdots) + b_{n-1}) + b_n.$$

其中 $\cdot$ 表示矩阵乘法, 对于 Fastlin 框架, 我们以求解输出层第 i 个神经元 $X_{n+1}^{(i)} = f(X_1)^{(i)}$ 的最大值为例, 若权重矩阵 W_n 的第 i,j 元 $W_n^{(ij)} > 0$, 第 n 层的像 X_n 的第 j 个元素 $X_n^{(j)}$ 应代入激活函数 σ'_{n-1} 平行四边形松弛的线性上界, 反之则代入平行四边形松弛的线性下界. 由此逐层反向传播, 直到代入第一层神经元的范围 X_1. 此过程等价于由第一层神经元的扰动范围 X_1 从前往后传递, 对激活函数采用相同的平行四边形松弛方式, 逐层计算得到最后一层神经元的范围. 因此我们证明下面两种情形:

1. 相同的 Zonotope 经过仿射变换之后,DeepZ 和 Fastlin 的输出是等价的 Zonotope.

2. 相同的输入范围经过 ReLU 激活[3]之后,DeepZ 和 Fastlin 的输出范围相同.

对于情形 1, 由于神经网络在传播过程中交替进行仿射变换和 ReLU 激活, 为了简化计算, 我们先忽略表示层数的下标. 设第一层仿射变换的输入为 $X = \{x \mid a^- \leqslant x \leqslant a^+\}$, 则 DeepZ 中对应的 Zonotope 形式表示为 $X = \{x \mid x = \frac{a^+ + a^-}{2} + \frac{a^+ - a^-}{2} \cdot \epsilon, \epsilon \in [-1,1]\}$.

对于 Fastlin 输入范围中的任一点 $x_0 \in X$, 都能唯一找到 DeepZ 输入范围中 $\epsilon_0 = \frac{2x_0 - (a^+ + a^-)}{a^+ - a^-} \in [-1,1]$ 对应的点 x_0', 使得经过仿射变换之后,$Wx_0 + b = W(\frac{a^+ + a^-}{2} + \frac{a^+ - a^-}{2} \cdot \epsilon_0) + b = Wx_0' + b$. 因此当 Fastlin 和 DeepZ 的输入范围是相

[3]Fastlin 只适用于 ReLU 激活函数的神经网络.

同的 Zonotope 时, 经过仿射变换之后得到 Fastlin 求解的范围包含于 DeepZ 得到的 Zonotope 范围. 反之, 对于 DeepZ 输入的 Zonotope 中任意 $\epsilon_1 \in [-1,1]$ 对应的 x_1, 都能在 Fastlin 输入中范围中找到对应的 $x_1' = \frac{a^+ + a^-}{2} + \frac{a^+ - a^-}{2} \cdot \epsilon_1 \in [a^-, a^+]$, 使得 $Wx_1 + b = W(\frac{a^+ + a^-}{2} + \frac{a^+ - a^-}{2} \cdot \epsilon_1) + b = Wx_1' + b$. 因此当 Fastlin 和 DeepZ 的输入范围为相同的 Zonotope 时, 经过仿射变换之后得到 DeepZ 求解的范围包含于的 Fastlin 得到的范围, 从而 Fastlin 和 DeepZ 在仿射变换的过程是等价的.(该情形也可以利用矩阵乘法展开来证明.)

下面证明情形 2. 情形 1 的证明可得出对于每个神经元经过 Fastlin 和 DeepZ 的仿射层传播之后会得到相同的上下界 l 和 u. 下面我们证明经过 ReLU 激活后 DeepZ 和 Fastlin 也得到相同的范围. 若 $l \geqslant 0$ 或 $u \leqslant 0$, 则 Fastlin 和 DeepZ 框架激活之后均得到确定的相同的值, 对于激活程度不确定的神经元 ($l \leqslant 0$ 且 $u \geqslant 0$),Fastlin 采用线性约束的方式为其添加平行四边形松弛. 对于激活前的神经元 $x \in [l,u]$,Fastlin 求得的 $ReLU(x)$ 的范围为:$\frac{u}{u-l} \cdot x \leqslant ReLU(x) \leqslant \frac{u}{u-l}(x-l)$. 而 DeepZ 得到的 Zonotope 松弛的中心为 $(\frac{l+u}{2}, \frac{u^2}{2(u-l)})$, 因此 DeepZ 得到的 ReLU 激活后 Zonotope 表示为:$ReLU(x) = (\frac{l+u}{2}, \frac{u^2}{2(u-l)}) + (\frac{u-l}{2}, \frac{u}{2}) \cdot \epsilon_1 + (0, \frac{-ul}{2(u-l)}) \cdot \epsilon_2$. 下面我们证明二者求得的 ReLU 激活范围 $\frac{u}{u-l} \cdot x \leqslant ReLU(x) \leqslant \frac{u}{u-l}(x-l)$ 和 $ReLU(x) = (\frac{l+u}{2}, \frac{u^2}{2(u-l)}) + (\frac{u-l}{2}, \frac{u}{2}) \cdot \epsilon_1 + (0, \frac{-ul}{2(u-l)}) \cdot \epsilon_2$ 是等价的.

对于任意 $x \in [l,u]$, 我们首先证明在 Fastlin 中经过激活后得到的范围 $ReLU(x) \in [\frac{u}{u-l} \cdot x, \frac{u}{u-l}(x-l)]$ 包含于 DeepZ 得到的 Zonotope 中. 当 $x = \frac{l+u}{2} + \frac{u-l}{2} \cdot \epsilon_1$ 时,$\frac{u^2}{2(u-l)} + \frac{u}{2} \cdot \epsilon_1 + \frac{-ul}{2(u-l)} \cdot \epsilon_2 = \frac{2ux - ul(1+\epsilon_2)}{2(u-l)}$, 由于 $\epsilon_2 \in [-1,1]$, 所以上式的范围为:$\frac{ux}{u-l} \leqslant \frac{2ux - ul(1+\epsilon_2)}{2(u-l)} \leqslant \frac{u(x-l)}{u-l}$, 显然 Fastlin 在 x 处激活后的范围包含于 DeepZ 激活后的范围. 反之我们证明对于任意的 $\epsilon_1, \epsilon_2 \in [-1,1]$,DeepZ 对应的激活后的 $ReLU(x) = \frac{u^2}{2(u-l)} + \frac{u}{2} \cdot \epsilon_1 + \frac{-ul}{2(u-l)} \cdot \epsilon_2$ 也都包含于 Fastlin 范围中. 对 $\epsilon_1, \epsilon_2 \in [-1,1]$,$x = \frac{l+u}{2} + \frac{u-l}{2} \cdot \epsilon_1$,Fastlin 得到的激活后的范围是 $\frac{u}{u-l} \cdot (\frac{l+u}{2} + \frac{u-l}{2} \cdot \epsilon_1) \leqslant ReLU(x) \leqslant \frac{u}{u-l} \cdot (\frac{l+u}{2} + \frac{u-l}{2} \cdot \epsilon_1 - l)$, 即 $\frac{u(l+u)}{2(u-l)} + \frac{u}{2} \cdot \epsilon_1 \leqslant ReLU(x) \leqslant \frac{u}{2} \cdot (\epsilon_1 + 1)$. 由于 $\frac{u^2}{2(u-l)} + \frac{u}{2} \cdot \epsilon_1 + \frac{-ul}{2(u-l)} \cdot \epsilon_2 - \frac{u}{2} \cdot (\epsilon_1 + 1) = \frac{ul(1-\epsilon_2)}{2(u-l)} \leqslant 0$, 且 $\frac{u^2}{2(u-l)} + \frac{u}{2} \cdot \epsilon_1 + \frac{-ul}{2(u-l)} \cdot \epsilon_2 - (\frac{u(l+u)}{2(u-l)} + \frac{u}{2} \cdot \epsilon_1) = \frac{-ul(\epsilon_2+1)}{2(u-l)} \geqslant 0$, 因此 DeepZ 求得的松弛范围包含于 Fastlin 求得的范围, 从而二者在 ReLU 激活函数的松弛上得到的范围也是等价的.

综上所述, 可由归纳法证明出对于任意 ReLU 激活的神经网络, 如果神经网络的输入范围是 Zonotope, 则经过 Fastlin 和 DeepZ 求出的输出范围是等价的. □

Wang 等人提出了 Reluval [30], 这是一种基于符号区间传播的验证方法.Reluval 为每一个神经元添加符号区间界, 类似于 DeepPoly 的符号界. 对于 ReLU 激活函数的模拟, 他们采用区间分割的方式, 将其中一部分神经元范围分割为若干个子区间的并集. 对于每个子区间, 该神经元很可能变成确定的激活或不激活的神经元, 从而得到更精确的输出范围. 相比于 Reluplex 等精确验证方法的高复杂度和难拓展性, 以及 Wong & Kolter [46], Dvijotham [69] 等人提出的基于多面体松弛或凸优化带来的过度松弛问题,Reluval 的区间计算能相对高效地进行传播, 同时将输入区间进行切割可以尽可能避免过度松弛导致的假反例现象.

在接下来的工作中,Wang 等人在区间分析和线性松弛分析 [30,41] 的基础之上改进了 Reluval 框架, 得到精度和效率都更高的验证工具 Neurify [32]. 相比于 Reluval, 当神经元之间的依赖关系变得太复杂而无法进一步用符号表示时,Neurify 通过添加一定的松弛来继续计算神经元的范围, 和 DeepZ、Fastlin 这两种验证框架有异曲同工之妙. 事实上,Neurify、DeepZ 以及 Fastlin 在验证精度上是等价的, 而相比于 Reluval 和 Reluplex,Neurify 在效率和精度上都有明显的优势.

以上验证方法会面临如何将 ReLU 激活函数的松弛推广到 tanh 和 Sigmoid 等其他激活函数的问题,Zhang 等人提出了通用的 CROWN [33] 框架, 通过一种自适应的松弛方式为不同的激活函数添加符号线性边界来解决可拓展性问题.CROWN 可以拓展到多种激活函数的神经网络上, 甚至可以采用二次函数为激活函数添加合适的松弛. 而对于 ReLU 神经网络来说,CROWN 实际上和 DeepPoly 的精度是等价的. 相比于之前的方法,CROWN 可以有效地验证规模超过一万个神经元、具有不同激活函数的神经网络. 随后 Salman [95] 等人提出了一个统一的分层凸松弛框架, 系统地阐述了 DeepZ, Fastlin, Neurify, DeepPoly, CROWN 等验证框架的关系, 从凸松弛和拉格朗日对偶这两个方面讨论了这些方法的验证效率, 并揭示了这类松弛方法在验证精度上具有的理论上限.

Li 等人在符号区间算术的基础上, 进一步将符号传播和抽象解释结合, 将符号传播推广到更一般的数值抽象域上, 实现了工具 DeepSymbol [36,96] 和以此工具为基础的神经网络验证平台 PRODeep [97]. 值得一提的是, 验证平台 PRODeep 是国内首个神经网

络验证平台, 它集成了约束求解、抽象解释和符号传播三类技术, 并对它们进行了深度的结合.

前述大多数工作重点关注结构简单的 FNN 的鲁棒性验证.Boopathy 等人 [35] 提出了一个通用且高效的 CNN-Cert 框架, 将深度神经网络的鲁棒性验证拓展到了 CNN, 可以处理包括卷积层、最大池化层、残差块等在内的多种神经网络结构, 并支持多种激活函数.CNN-Cert 框架通过利用卷积层的特殊结构, 使得 Fastlin 和 CROWN 只是其特例, 并且在鲁棒边界相似甚至更好的情况下, 将二者的效率分别提升了 17 倍和 11 倍.

基于 IBP 的验证方法在许多种情况下的效率明显优于基于线性松弛的方法, 但 IBP 计算得到的边界要比线性松弛宽松得多. 因此 Zhang 等人提出了一种新的神经网络对抗训练方法, 通过将前向传播中的 IBP 边界和反向传播中基于线性松弛的 CROWN 边界相结合得到 CROWN-IBP [37]. 他们构造出一个关于 IBP 边界和 CROWN-IBP 边界的损失函数, 通过最小化该损失函数求解出神经网络的参数, 使得训练之后的神经网络不仅具有良好的分类精度, 还具有较高的鲁棒性. 在训练可验证的鲁棒神经网络方面,CROWN-IBP 具有很高的计算效率, 并且得到的范围始终优于直接利用 IBP 得到的范围, 很好地实现了精度和效率之间的平衡.

Lyu 等人 [38] 在 CROWN 的基础上又提出了 Fastened CROWN (FROWN) 框架, 这是一种用于提高神经网络鲁棒性的训练方法. 他们证明如果将神经元的上下界约束个数分别只限制为一个, 则相比于线性规划,CROWN 得到的神经元的上下界是最优的. 他们在此基础上提出的 FROWN 可以进一步地缩紧 CROWN 提供的边界, 并且在理论上能拓展到其他类型神经网络的验证方法中, 例如验证 CNN 的 CNN-Cert 以及验证 RNN 的 POPQORN.

基于符号传播的验证方法将符号计算的高效、可拓展性等优势应用到神经网络验证中, 并取得了很多成果, 但是随着神经网络层数增加规模增大, 这类方法的表达式也会变得更加复杂, 并且在传递过程中会逐层放大误差.

■ 3.2.4 基于 Lipschitz 的验证方法

Szegedy [98] 等人证明带有 ReLU 激活函数或者最大池化层等结构的 FNN 和 CNN 是 Lipschitz 连续的, 所以神经网络的输出在上下界范围内的所有值都是可达的. 因此一个比较自然的想法是从神经网络自身的性质出发, 利用 Lipschitz 常数来验证有关性质. 我们称一个神经网络具有 Lipschitz 连续性, 如果这个神经网络满足以下定义 [63]:

定义 3 (Lipschitz 连续性)**.** *给定两个度量空间 (X, d_X) 和 (Y, d_Y), 其中 d_X 和 d_Y 分别是集合 X 和 Y 上的度量, 一个函数 $f : X \to Y$ 是 Lipschitz 连续的, 如果存在一个实数 $K \geq 0$ 使得对任意的 $x_1, x_2 \in X$, 下列不等式都成立:*

$$d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2). \tag{1}$$

这里的 K 称为关于函数 f 的 Lipschitz 常数, 满足该不等式的最小的 K 值称为最优的 Lipschitz 常数, 通常用符号 K_{best} 表示.

关于 Lipschitz 验证方法, 比较早期的工作是 Hull 等人提出的 [99]. 他们介绍了多项式神经网络 (Polynomial Neural Network, PNN) 验证技术, 使用 Lipschitz 常数为所有可能的输入提供神经网络输出的可保证边界, 从而避免了在所有可能的输入组合下测试网络的输出.Hein 和 Andriushchenko 等人 [52] 提出了 Cross-Lipschitz 正则化函数, 利用局部 Lipschitz 常数, 给出了只具有一个隐藏层的神经网络的鲁棒边界. 但是该方法要求网络连续可微, 因此不能直接应用于 ReLU 神经网络.

Ruan 等人利用神经网络输入和输出上的 Lipschitz 连续函数来解决神经网络的可达性问题, 提出了一种基于自适应嵌套优化的新算法 DeepGO [63], 并且证明了 ReLU 之外的其他激活函数, 例如 Softmax、Sigmoid 以及 tanh 等也具有 Lipschitz 连续性. 他们将神经网络的可达集分析转化为求解神经网络的全局最大/最小值, 进一步转化为一系列优化问题, 通过迭代的方式进行求解, 并利用 Lipschitz 常数 K 给出了收敛条件. 他们发现当 K 与最优 Lipschitz 常数 K_{best} 越接近, 收敛速度就会越快, 因此采取了一种动态更新的策略迭代地寻找更好的 K, 加速优化进程, 这项工作标志着基于 Lipschitz 常数的方法在神经网络验证领域的成功应用.

Weng 等人进一步为神经网络的鲁棒性分析问题转化为局部 Lipschitz 常数的估计问题提供了理论依据, 并通过极值理论对 Lipschitz 常数进行了有效的评估, 他们首次提出了鲁棒性度量分数 Clever [53](Cross Lipschitz Extreme Value for nEtwork Robustness), 并且可以应用到 ImageNet 等大型数据集相关的神经网络分类器. 虽然没有提供严格的保证, 但 Clever 分数确实是一个很好的鲁棒性估计, 并且在 FNN 中与 Reluplex 给出的精确鲁棒边界是比较接近的. 相比于其他的神经网络验证方法,Clever 具有以下优势:Clever 分数与攻击方法无关, 并且可以不受神经网络结构的限制, 对于规模庞大、结构复杂的神经网络来说是一个很好的衡量鲁棒性的方法.

Weng 等人在随后的工作中 [100] 对 Clever 分数进行了两个方面的拓展, 其一是利用极值理论为二次可微的分类器, 即带有 Sigmoid、tanh、Softplus 等激活函数的神经网络提供了一种鲁棒性估计, 称为二阶 Clever 分数. 其次他们阐述了如何利用 Clever 分数和反向传播可微近似 (Backward Pass Differentiable Approximation,BPDA) [101] 来处理梯度掩码[4](Gradient Masking) [101,103]

[4]梯度掩码是一种流行的防御方法, 典型的梯度掩码方法包括添加不可微层、防御性蒸馏 [102] 等.

之后的不可微变换, 然后借助 Clever 评估模型的鲁棒性. 他们发现在许多种情况下, 神经网络的一阶和二阶 Clever 分数是一致的, 因此 Clever 作为神经网络的鲁棒性衡量指标, 是比较合理且可拓展的.

前述 Clever [53] 工作中还提出局部 Lipschitz 常数可以看作是方向导数的最大范数, 因此 Weng 等人提出了 Fastlip [31], 通过分类讨论 ReLU 神经网络的三种激活模式推导出局部 Lipschitz 常数的上界, 从而获得扰动区域的鲁棒边界. 他们实验发现由 Fastlip 得到的鲁棒边界相比于 Reluplex 得到的精确边界仅仅松弛 2 ∼ 3 倍, 并且与基于线性规划的方法相比, 也具有非常接近的精度. 但是由于他们不需要求解任何的线性规划问题或者对偶问题, 因此在时间效率上要快 33 到 14000 倍. 当神经网络的规模扩大到 Resnet 时,Fastlip 仍然能给出不平凡的鲁棒边界.

此类基于 Lipschitz 常数的验证方法通常效率非常高, 与精确的验证算法相比, 效率可达到其成千上万倍. 但是由于这类方法求解得到的鲁棒半径非常依赖 Lipschitz 常数的求解精度, 因此如果 Lipschitz 常数精度不高很可能导致这些验证方法的鲁棒半径不准确, 从而难以验证一些比较强的性质. 并且由于 Lipschitz 常数仅仅是对鲁棒半径做出一个估计, 许多基于 Lipschitz 常数的方法并没有一个严格的鲁棒性保证.

■ 3.2.5 基于凸优化的验证方法

基于凸优化的神经网络验证方法主要是利用一些对偶理论、半定规划等数学原理来优化神经网络的鲁棒边界, 以及通过最小化损失函数来训练一个可证明鲁棒性的神经网络, 此类验证方法的数学理论较强.

Dvijotham 等人 [68] 基于线性规划提出了一种算法框架来训练可证明鲁棒性的神经网络, 从而验证网络是否满足给定的输入-输出特性. 该算法的关键思想是同时训练两个网络, 一个预测器网络用于当前任务的预测, 另一个验证器网络用于计算预测器网络满足被验证属性的程度. Dvijotham 等人在后续的工作中 [69] 将验证问题转化为一个凸优化问题, 寻找违反性质的最强反例, 并通过求解这个凸优化问题的拉格朗日松弛来刻画最坏情况下关于给定性质的鲁棒半径. 该框架可以应用于具有任意类型激活函数的 FNN, 并且凸优化问题的规模与神经元的个数呈线性相关, 因而复杂度较低.

Wong & Kolter [46] 将 FNN 的扰动区域定义为一个凸多面体, 通过对激活函数的可达集添加一个凸外部逼近 (Convex Outer Approximation), 并通过线性规划最小化该外部逼近在最坏情况下的损失. 他们证明出该线性规划的对偶问题可以近似地表示为反向传播的深度网络, 从而可以高效地找到一个可证明的鲁棒边界, 同时可以快速找到潜在的对抗性样本. 他们提出的方法可以训练一个鲁棒的 ReLU 深度神经网络分类器, 并且可以抵御任意类型的攻击.

在后续的工作中,Wong 等人 [94] 将上述基于凸优化的方法拓展到了其他非线性激活函数以及带卷积层或残差结构的神经网络中, 并且可验证的网络规模上也有了很大的提升. 虽然还不能达到 ImageNet 这样的验证级别, 但是在规模和可拓展性上面相比于前面的工作有了实质性的进展. 他们主要对前面的工作 [46] 进行了几点扩展:1. 通过激活函数的 Fenchel 共轭函数拓展到具有任意激活函数的神经网络结构中; 2. 使用非线性投影技术, 对网络的输入维数和隐藏神经元的个数进行线性缩放, 同时最小化精度损失. 这个工作标志着深度防御的可拓展性向前迈出了重要的一步, 但他们提到仍有一些值得改进的地方: 除了利用有界的范数攻击来描述鲁棒性之外, 还可以进一步地考虑对抗扰动的性质.

Raghunathan 等人 [70] 提出了一种基于半定规划 (Semidefinite Programming, SDP) 的验证方法来计算具有一个隐藏层的神经网络在最坏情况下的鲁棒边界, 该边界可以为网络输入的所有攻击提供鲁棒性证书. 他们利用这个鲁棒性证书的可微特性, 通过对网络参数的联合优化, 提供了一个自适应的正则化器, 可以对所有类型的攻击提供鲁棒性保证. 实验表明基于 SDP 的方法得到的鲁棒边界相比于基于 Frobenius 范数和基于谱范数得到的边界更加严格.

Raghunathan 等人在随后还提出了一种更紧的基于 SDP 的方法来证明具有 ReLU 函数的神经网络的鲁棒性 [71], 并且可以拓展到具有任意多个隐藏层的 FNN 当中. 相比于其他凸优化方法, 例如 Wong & Kolter [46]、以及 Dvijotham 等人提出的方法 [68,69], 他们提出的 SDP 松弛可以捕获到神经元之间的关联信息, 得到更紧的边界. 此外, 相比于他们之前的工作 [70], 该工作中提出的 SDP 松弛能提供更严格的鲁棒性保证. 在可拓展性方面, 新的 SDP 方法理论上可以应用于 CNN, 只需将其展开为 FNN 即可, 但是还没有进一步的实践和验证.

Müller 等人提出了一种基于凸优化的 PRIMA [47] 框架, 在基于抽象解释的 k-ReLU 的基础上进一步提升了验证精度.PRIMA 的主要亮点在于 Split-Bound-Lift Method(SBLM) 的全局验证框架, 并利用一种称为 Partial Double Description Method(PDDM) 的对偶方法求解多面体的凸包, 在最坏的情况下求凸包的过程也能达到多项式复杂度. 实验表明他们的方法相较于目前主流的验证算法, 如 DeepPoly,k-ReLU 等, 都有不错的提升. 他们在实验中利用英伟达公司自动驾驶系统装载的神经网络 Dave [7] 来测试验证算法的精度和效率, 是神经网络鲁棒性验证应用于工业系统中大型神经网络的一个进步.

以上基于凸优化的思想验证神经网络的方法, 相比于其他基于抽象解释、Lipschitz 的方法在验证规模上稍为逊色, 难以拓展到识别 ImageNet 等大型数据集的神经网络上. 但是这类方法通常可以帮助训练出一个具有良好对抗鲁棒性的神经网络, 并且部分验

证方法可以支持多种激活函数的神经网络验证, 是数学理论在形式化验证领域中的一个很好的应用.

表 3 对各类 FNN 验证方法的优缺点做出了对比, 更多关于 FNN 验证的文章可以详见文献 [104,105]. 综述 [105] 重点介绍了神经网络的验证、测试、攻击防御、可解释性等相关工作, 另外 [106] 详尽地介绍了关于 FNN 的鲁棒性分析方法, 本综述则从时间维度上展示了各类 FNN、RNN 验证方法的发展, 分析了更多关于 RNN 验证方法以及 FNN、RNN 验证方法之间的联系.

■ 4 RNN 结构

RNN 在自然语言处理和语音识别等领域应用广泛, 它以时间序列数据作为输入, 由于其内部具有循环结构,RNN 会将循环状态不断地向自身传递, 最终得到输出. 对于分类 RNN 来说, 输入可能是一系列待分类的图片序列, 输出为图片的预测标签. 而对于用作自动驾驶控制系统的 RNN, 则会以一系列连续的道路探测图像作为输入, 处理之后输出对车辆的控制信号, 例如直行, 转弯, 减速等等. 目前应用比较广泛的 RNN 包括 Vanilla RNN [107], 长短期记忆模型 [108](Long short-term memory,LSTM), 门控循环单元 [109](Gated Recurren Unit, GRU) 等等.

Vanilla RNN 是结构最简单的 RNN, 接收序列输入, 最后输出预测值, 与 FNN 结构类似, 不同之处在于每一个隐藏状态的值将由上一个隐藏状态与这一时刻的输入决定.

LSTM 首次由 Sepp Hochreiter 等人 [108] 提出并被广泛应用于不同的领域, 例如自然语言处理 [110,111], 阅读理解 [112,113], 聊天机器人 [114] 等,LSTM 提出的主要目的是用以解决在长序列训练过程中存在的梯度消失以及梯度爆炸问题. 不同于 FNN,LSTM 网络具有记忆性和遗忘性, 因而对于一些较长序列,LSTM 能表现出优异的处理能力. 如果将 LSTM 网络展开, 则其是由一系列具有重复结构的链式模块构成的.LSTM 在每个模块中都通过三个门来添加或删除信息, 分别称为忘记门、输入门以及输出门. 门的结构即为 Sigmoid 激活操作和哈达玛 (Hadamard) 积 $\otimes$ 的组合部分, 具体的结构如图 5 所示 [115].

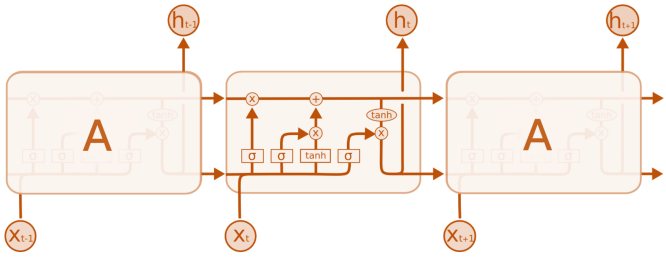

图 5: LSTM 的结构, 其中方框表示激活函数操作, 圆圈表示圈内运算符的逐点操作, 箭头表示数据的流向.

门控循环单元 (GRU) 的概念是由 Cho 等人 [109] 提出来的, 作为 LSTM 网络的变体,GRU 拥有更简单的结构以及更优越的效果, 因此在自然语言处理领域中也得到了广泛的应用.GRU 的结构包括重置门和更新门, 如下图 6 所示.

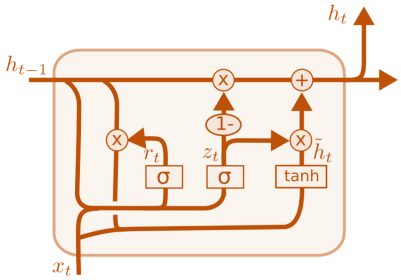

图 6: GRU 的结构, 其中方框表示激活函数操作, 圆圈表示圈内运算符的逐点操作, 箭头表示数据的流向.

由于目前 RNN 验证方法正处于起步阶段, 大多数验证方法都从 Vanilla RNN 的验证入手, 再将其拓展到目前应用最为广泛 LSTM 网络, 下一节将具体地介绍和分析 RNN 的验证方法.

■ 5 RNN 验证方法

RNN 的循环单元是为了处理序列的输入, 使得序列中的下一个元素的处理过程依赖于前面元素的处理结果, 多被用于自动驾驶中的车轨预测、语音识别、机器翻译、以及其他依赖于序列输入的领域. 由于结构上和 FNN 有显著不同,RNN 对于序列数据的扰动空间存在更严格的定义, 并缺乏适当的度量扰动程度的指标, 因此相比于 FNN, 此类序列扰动空间是更加难以描述的 [116]. 因此, 尽

管目前 FNN 的鲁棒性验证的研究已经取得了很多进展, 但 FNN 的验证方法并不能完全适用于 RNN 验证. 对 RNN 验证方法的研究目前仍处于起步阶段, 相关的工作还比较匮乏.

目前已有的文章中关于 RNN 的鲁棒性验证主要分为三大类: 一类是基于抽象解释 [117] 的方法, 将 RNN 的输入范围 X 以及 X 在 RNN 内部传播得到的像 $R_i(X)$ 抽象为一组特定的几何形状. 例如 Box 抽象域、Zonotope 抽象域、多面体、线性不等式等等, 然后验证输出层得到的抽象域是否满足相应的性质. 另一类验证方法是将 RNN 展开为 FNN, 再利用 FNN 的验证方法对其进行验证. 第三类是从 RNN 中提取出近似的自动机 (DFA) 或者有限状态机, 再利用比较成熟的自动机模型验证方法对其性质进行验证. 各种 RNN 验证方法的联系和对比由图 7 和前文表 2 所示.

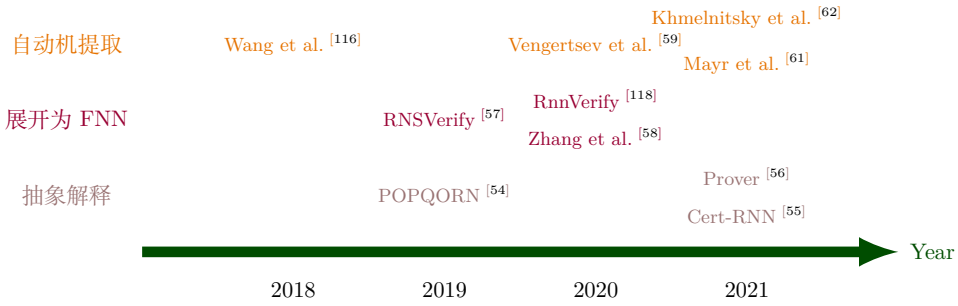

图 7: RNN 验证方法, 不同的颜色代表不同验证技术类别.

■ 5.1 基于抽象解释的验证

基于抽象解释的验证方法在 FNN 上得到了较为成熟的应用, 一个自然的想法便是将抽象解释框架应用到 RNN 验证领域中. 因为很难形式化地定义 RNN 在语言预测方面的错误, 因此对于 RNN 的验证主要被限制在基于 RNN 的分类任务中.

当前有许多通过攻击 RNN 来评估鲁棒性的工作 [119,120], 以及将 Clever 分数 [53] 直接应用于 RNN, 为 RNN 提供鲁棒性分数来衡量鲁棒性等等, 但是这些方法并没有严格的鲁棒性保证. 受到 Fastlin [31] 为神经网络的操作添加线性约束的启发,Ko 等人首次将抽象解释与 RNN 验证结合起来提出了 POPQORN [54] 验证框架, 这也是首次提出的 RNN 的验证工作. 具体地, 当输入产生一个 l_p 范数下的小扰动时, 他们用线性函数为扰动范围添加一个边界, 并为 Sigmoid 激活、tanh 激活以及哈达玛 (Hadamard) 积等非线性操作 $\sigma(v)$ 添加了一组线性的上下界 $h_U(v)=\alpha_U(v+\beta_U)$ 和 $h_L(v)=\alpha_L(v+\beta_L)$, 使得非线性操作之后的像集 $\sigma(v)$ 都落在该线性边界内, 即 $h_L(v)\leqslant\sigma(v)\leqslant h_U(v)$. 其中, 系数 α 和 β 的值取决于自变量 v 的范围, 是将系数 α 和 β 的求解过程转化为带约束的优化问题, 并利用梯度下降法求解该优化问题得到的. 由此便可从输出层满足的线性约束逐层反向传播, 将每一层非线性操作的上下界用线性函数代入, 最后代入输入层的变量范围, 得到可达集的上近似.

他们采用线性函数或线性平面逼近非线性函数, 并基于梯度优化来计算相应的线性函数的参数. 由于优化过程需要较高的计算复杂度, 因此 POPQORN 很难扩展到实际的包含成千上万个神经元的神经网络中.POPQORN 能验证的 Vanilla RNN 的神经元个数约为 896 个,LSTM 网络神经元个数约为 512 个, 离拓展到更大的网络还有一定的距离. 其次, 由于优化方法缺乏最优性保证, 线性边界的系数很可能不精确. 如果平面和真实的曲面之间间隙过大, 可能会导致给出的线性边界过于松弛, 以至于能验证的鲁棒半径比较小. 一个可能的改进方向是考虑构造一个损失函数, 尽可能减小线性平面与真实非线性函数之间的差距.

Du 等人将 DeepZ 的思想应用到了 RNN 验证中, 提出了一种比 POPQORN 更紧的验证框架 Cert-RNN [55]. 类似于 DeepZ, 他们定义了一个 Zonotope 去捕获对抗性输入的范围, 为 tanh 等非线性激活函数构造 Zonotope 上近似, 并将该 Zonotope 在神经网络中逐层传递. 而中间的仿射变换层不改变 Zonotope 精度, 最后验证输出 Zonotope 是否满足给定性质. 他们为 Sigmoid 和 tanh 等非线性激活函数采用的松弛方式比 DeepZ 更加精确: 连接自变量 x 上下界 l_x 和 u_x 对应的激活函数值 $(l_x,\sigma(l_x))$ 和 $(l_u,\sigma(l_u))$, 将该切线进行上下平移, 使其恰好与激活函数图像相切, 得到的平行四边形即为非线性激活函数添加的 Zonotope, 如图 8 所示.

在 LSTM 网络中, 如果采用 POPQORN 中的方法, 为 $Sigmoid\otimes tanh$ 函数添加关于两个变元的线性界, 其中 $\otimes$ 表示哈达玛积, 精度损失会很大. 所以他们对这个线性界做出了精化, 根据自变量的上下界讨论了九种情况, 对每种情况的抽象域都做了比较紧的约束, 从而避免精度损失过于严重. 对于 $Sigmoid\otimes Identity$ 的非线性操作, 他们也列出三种情况并为其添加 Zonotope 边界. 对比实验结果表明相比于 POPQORN,Cert-RNN 在验证神经网络的鲁棒半径上具有很大的优势.

相比于 POPQORN 验证框架,Cert-RNN 具有两个主要的优势: 首先, 由于 POPQORN 采用了不精确的区间计算, 利用体积最小化将线性平面系数求解的问题转化为带约束的优化问题, 效果可能不稳定, 并且没有最优性保证. 而 Cert-RNN 采用了 Zonotope

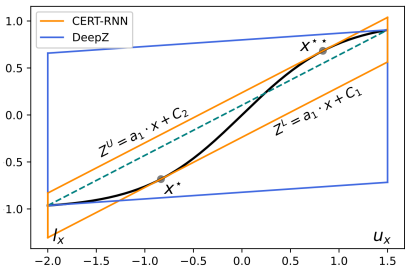

图 8: Cert-RNN 和 DeepZ 在 tanh 函数上的 Zonotope 松弛对比, 黑色曲线表示 tanh 激活函数, 绿色线段是自变量的上下界对应的激活函数值的割线, 将其上下平移至与 $tanh(x)$ 相切即可得到橙色的 Zonotope. 蓝色的 Zonotope 是 DeepZ 采用的松弛.

描述神经元之间的联系, 对非线性激活的松弛理论上会更精确. 这就是 Fastlin 和 DeepZ 在 FNN 验证上效果一致, 而它们在 RNN 上面的扩展工具 POPQORN 和 Cert-RNN 精度有差异的原因. 其次,POPQORN 采用梯度下降法逼近优化问题的解效率低下,Cert-RNN 利用了抽象变压器来对非线性激活和操作做松弛, 可以比较高效地验证 RNN 的鲁棒性. 在一些 LSTM 网络中,Cert-RNN 的验证效率比 POPQORN 高 20 倍以上.

在 FNN 的鲁棒性验证中, 以 DeepPoly 为原型的反向传播框架是一种比较主流的验证方式. 为了将反向传播的思想应用到 RNN 验证中,Ryou 等人将 DeepPoly [27] 框架与 RNN 验证相结合, 提出了新的验证理论:Prover [56]. 他们借鉴了 DeepPoly 的思想, 为 RNN 的每一个神经元添加数值界和符号界作为抽象域. 对于非线性激活函数以及非线性操作, 他们仍然采用线性函数为神经元添加上下界约束. 与 POPQORN 框架不同的是, 他们通过在自变量的可行域内采样, 将采样点与其在线性平面上投影的距离之和作为损失函数, 通过求解最小化损失函数得到的线性规划问题, 解得线性约束的系数, 如图 9 所示. 他们还提出将原始的可行域分为 4 个不同的三角形区域, 在每个区域分别采样求解出该区域对应的 “候选平面” , 并利用这些候选平面的线性组合去拟合原始的非线性函数. 在计算非线性激活函数的线性上下界时, 由于 Prover 是根据可行域中采样的点集来计算线性边界的系数, 所以求出来的线性平面可能不具有可靠性. 为了解决这个问题, 他们通过平移线性边界的系数来调整上下界, 得到一个可靠的线性边界.

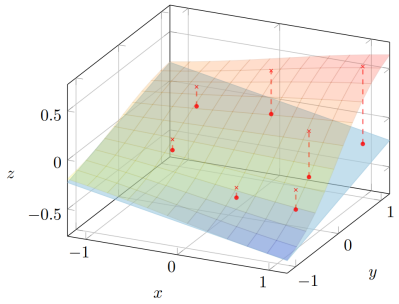

图 9: Prover 求解线性函数系数的方法示意图, 在激活函数曲面上随机采点并投影到线性平面上, 以原始采样点和投影点的距离之和作为损失函数并将其最小化求得线性平面系数.

实验部分用 LSTM 对语音分类及图像识别的数据集进行验证, 并和 POPQORN 进行了比较, 结果表明用各子区域的候选平面做线性组合的方式优于直接在整个可行域上的线性平面拟合, 并且这两种方法能验证的半径都要远远优于 POPQORN.

以上的 RNN 验证方法都是基于抽象解释, 将 RNN 的输入和中间操作都用线性约束为其添加一个上近似, 最后验证输出的线性区域是否满足性质. 基于抽象解释验证 RNN 的方法大多借鉴了 FNN 的验证方法, 关系如图 10 所示.

■ 5.2 基于 FNN 的验证

另一种常见的 RNN 验证方法是将 RNN 展开为等价的或上近似 FNN, 再验证该 FNN 的性质.

最早对此类验证方法进行研究的工作是 Akintunde 等人提出的 RNSVerify [57], 这是针对基于 RNN 的闭环系统的形式化验证方法. 基于 RNN 的闭环系统 (RNN-AES) 是基于 FNN 闭环系统 [121] 的扩展, 是由一个环境和一个代理 (Agent) 组成的闭环系统, 该环境包含若干状态, 并由该代理来更新当前的状态. 此时代理由 RNN 来充当, 并根据当前的环境选择下一个操作来更新状态. 他们形式化地定义了 RNN-AES 闭环系统, 并引入了若干支持该闭环系统的语义, 用 BNF 公式定义出需要验证的性质或规范, 便于解

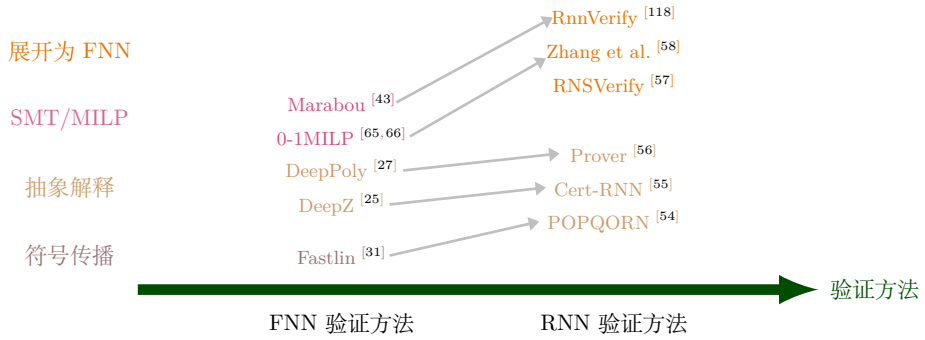

图 10: RNN 验证与 FNN 验证的关系, 不同颜色表示不同的验证方法类别, 箭头表示被指向的方法由箭头出发的方法借鉴而来.

决由此产生的验证问题.

RNSVerify 用两种方式将 RNN 展开为 FNN, 其一是 Input on Start (IOS) 展开方法. 展开之后 FNN 的输入为原 RNN 在所有时间步的输入,FNN 第一层的权重矩阵即为 RNN 每个时间步的输入到隐藏层系数矩阵 $W_{(i\to h)}$ 的拼接, 即 $W^{(1)} = I_s \odot W_{(i\to h)}$, 其中符号 $\odot$ 表示克罗内克积. 在传递进其它隐藏层时, 将每一个时间步的输入不断复制, 直到传递到该时间步对应的 FNN 的隐藏层. 第二种展开方法是 Input on Demand (IOD) 展开方法, 其主要思想是类似于 IOS 展开, 将所有时间步的输入作为 FNN 的输入, 每个时间步对应 FNN 的一个隐藏层. 通过添加权重为 1 的系数矩阵不断 “复制” 后面时刻的输入, 直到 FNN 到达的隐藏层对应于该输入的时间步, 再通过矩阵 $W_{(i\to h)}$ 将该输入传递进对应的隐藏层. 这两种展开方式如图 11 所示.

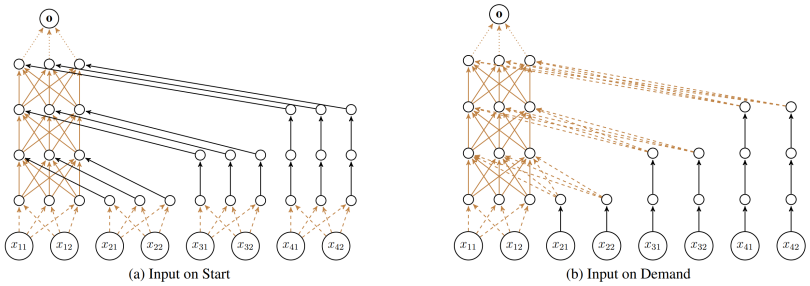

图 11: Input on Start (IOS) 展开方法和 Input on Demand (IOD) 展开方法

他们将展开得到的等价 FNN 编码为 MILP 约束, 并对 AES 系统环境中的状态和迁移规则用线性约束表示, 将问题编码为布尔可满足性问题 (Boolean Satisfiability Problem, SAT) 来判断系统是否满足给定的性质. 如果该 AES 系统满足给定的性质就输出 True, 否则算法会给出反例. 该方法对复杂程度有限的控制器是可靠且完备的. 他们利用 OpenAI 中的钟表垂直摆动作为实验, 目标是让代理学习如何通过在每个时间步施加一个小的旋转力来保持钟摆直立. 他们对 IOS 和 IOD 两种展开方法进行了比较, 发现 IOD 在 MILP 编码的过程中需要的变量和约束个数更少, 所以在验证时间上相比于 IOS 具有明显的优势.

实验中用于控制的 RNN 仅包含 16 个隐藏单元, 规模远小于其他 RNN 验证工具采取的神经网络.RNSVerify 通过展开 RNN 并利用线性规划来求解约束可满足性问题. 这种白盒验证方法非常依赖于 RNN 内部结构以及权重矩阵等参数来生成 FNN, 因此不能验证任意序列长度和更灵活的属性. 由于将 RNN 展开为 FNN 的过程中神经元的个数会扩增若干倍, 在进行 MILP 编码时会导致变量和约束个数激增, 所以将该验证方法拓展到更复杂的 AES 系统存在瓶颈.

Zhang [58] 等人在 RNSVerify 的启发下也采用了展开为 FNN 的方法来验证 RNN, 他们采用了三种已有的验证方法来验证展开之后的 RNN: 多面体传播, 并利用 MILP 编码 ReLU 激活函数 [66]、基于反例引导的抽象-精化方法 (CEGAR)、以及不变量探测. 与前面介绍的 RNN 鲁棒性验证方法不同的是, 他们介绍了一种认知任务——随机点运动任务, 目的是在有噪声的环境下使人或动物做出决策. 他们将该认知任务应用于 RNN, 并训练 RNN 使其对噪声环境下的运动点做出正确的决策. 他们提出了该任务下的三种可能满足的性质, 并对其进行验证:(1) 如果给定的刺激措施足够强烈, 网络应该会输出正确的决策; (2) 在刺激过程中出现了与其他刺激点不一致的点, 检验这个点上的特征是否会诱导 RNN 选择这个极端点的方向; (3) 检验是否存在一个输入, 导致最后几步的输出状态产生强烈的震荡, 做出相反的决策. 他们在验证以上三种性质时, 固定了 RNN 的输入长度, 从而可以将该 RNN 展开为

FNN 进行验证.

他们在固定输入长度 (110 步长) 后将 RNN 展开为 FNN, 并利用前述三种验证方式, 分别在 30 个输入区域中对上述性质进行验证, 并与参照方法 NNV [122] 进行了对比实验. 他们的创新点在于对一些 RNN 的应用场景提出了相应的性质并对其进行验证.

Jacoby 等人对展开神经网络的方法做出了改进, 提出了工具 RnnVerify [118], 其基本思想也是将 RNN 展开为 FNN. 与上述方法不同的是, 他们用不变量推理从原始的 RNN 得到 FNN, 作为原始 RNN 的上近似, 然后用 Marabou 来验证 FNN. 他们对于单个记忆单元和多个记忆单元的网络分别介绍了不变量的生成方法. 对于单个记忆单元, 采用形如 $\alpha_l \cdot (t-1) \leqslant \tilde{v}_t \leqslant \alpha_u \cdot (t-1)$ 的线性不变量模板, 其中 t 代表时间步长, 然后用二分法求出模板的上下界系数 α_l 和 α_u. 对于多个记忆单元, 仍然将每个隐藏单元的不变量形式设为 $\alpha_l \cdot (t-1) \leqslant \tilde{v}_t \leqslant \alpha_u \cdot (t-1)$. 与单个隐藏单元不同的是, 采用了 MILP, 并设置一个损失函数 $\sum \alpha_u - \sum \alpha_l$ 来考虑神经元之间的联系. 相比仅对单个神经元求解不变量的上下界系数 α_l 和 α_u, 能得到更紧的边界. 实验部分是对说话者识别任务中包含大约 220 个神经元的 Vanilla RNN 进行验证, 并和 RNSVerify 工具进行了比较. 它们在精度上都可以验证 RNN, 但是在时间上 RnnVerify 明显占优势.

以上内容都是基于将 RNN 展开为 FNN 的验证方法. 当限制输入序列长度时,RNN 中隐藏单元的循环次数也随即确定, 因此可以利用直接展开法得到等价的 FNN, 或者利用不变量探测的方式展开得到一个上近似 FNN, 再利用现有的验证 FNN 的方法进行验证. 虽然可以利用现有的工具验证展开之后的 RNN, 但由于展开为 FNN 的过程会导致参数个数扩大数倍, 将 FNN 编码为 MILP 或者利用 Marabou 验证的时间复杂度不容小觑.

■ 5.3 基于自动机的验证

RNN 可以用于处理序列输入, 所以一种自然的验证方法便是将 RNN 转化为*确定型有限自动机* (Deterministic Finite Automaton, DFA) [123,124], 并借助适当的自动机检验工具 [125] 对所提取的自动机执行验证任务. 从 RNN 中提取有限自动机可以采用白盒学习算法来实现. 例如 Weiss [126] 等人提出利用 L^* 算法, 用 RNN 来回答等价查询以学出一个近似的 DFA $\mathcal{A}$, 并不断优化 L^* 学习过程和抽象-精化过程. 从 RNN 中提取自动机已经有比较多的工作 [127–129], 也可以详见综述 [130].

在验证 RNN 的过程中会存在一些问题, 例如在 RNN 中对序列数据的扰动空间以及对扰动程度的刻画缺乏适当的度量. 为了解决这些挑战, Wang 等人 [116] 通过对字符串定义平均编辑距离来度量字符串之间的扰动程度, 并使用 DFA 来检查生成的对抗性样本是否违反了某些性质. 他们基于量化分析从给定的 RNN 中提取出一个 DFA [131,132], 在此基础上形式化地定义了一系列指标 (正确率, 成功率, 保真度等) 来刻画提取出来的 DFA 的质量, 然后利用随机采样的方法对提取出来的自动机进行性质验证. 他们在 Tomita 语法 [133] 上展示了用不同的 RNN(二阶 RNN [134],Elman-RNN,MI-RNN,LSTM 以及 GRU 等) 提取出来的 DFA 的准确率, 成功率以及保真度等性质, 并发现具有二次型形式的二阶 RNN 能更高效更精确地提取自动机. 提取出 DFA 之后, 他们利用 DFA 对 RNN 的对抗鲁棒性进行验证, 此处的鲁棒性用 γ 量化, 形式化地描述为: 给定 RNN f、根据 f 提取出来的自动机 $\mathcal{A}$、被扰动的样本 x、以及在半径为 r 的邻域 $B_r(x)$ 内随机采样的样本 $\{x_1, \ldots, x_T\}$, 输出该 RNN 在扰动范围 $B_r(x)$ 下的对抗精确度

$$\gamma = \frac{\#\{1 \leq j \leq T : f(x_j) = \mathcal{A}(x_j) = p\}}{T}.$$

其中 $f(x_j) = \mathcal{A}(x_j) = p$ 表示网络 f 和自动机 $\mathcal{A}$ 都将 x_j 分类为正, 在实验中则表示样本 x_j 属于 Tomita 语法.

这种通过在扰动区域内随机采样来估计神经网络的鲁棒性质的方法, 不能为神经网络给出一个严格的概率鲁棒保证. 如果输入空间的维数过大就很可能导致采样数不够充分, 因此进一步影响对抗鲁棒性估计的准确性.

不同于随机采样验证对抗鲁棒半径,Vengertsev [59] 等人提出了用蒙特卡洛采样法来检验他们形式化定义的六种性质, 包括四种状态安全性质: 高置信度、决断性、鲁棒性和覆盖性, 用来描述理想的 RNN 应该具有怎样的结构或参数; 以及两种时序安全性质: 长期关系和记忆性, 用以描述 RNN 在处理输入序列中的每个元素后, 之后的输入序列应满足的条件. 他们用*标签状态迁移系统* (Labeled Transition System, LTS) [135] 来描述 RNN 的行为. 在这个状态迁移系统中,RNN 模型的以上六种性质被定义为线性时态逻辑 LTL 公式, 然后对该有限状态系统的行为进行形式化推理.

Vengertsev [59] 等人假定待验证的系统有许多违反属性的路径. 为了完整刻画待验证模型的性质, 他们利用蒙特卡洛采样法来估计两个隐藏神经元个数分别为 80 和 240 的 RNN 模型 M_1 和 M_2 满足这些性质的概率, 并利用穷举法求解出模型满足相应性质的真实概率. 实验对比得出验证四个状态安全属性时, 蒙特卡洛采样法至多仅需采样穷举法 0.9% 的采样点便能以很小的误差收敛到真实概率的误差阈值内. 而对于时序安全性质的验证, 蒙特卡罗采样方法只需最多采样穷举条件下 18% 的伯努利分布点, 就可以收敛到满足这些性质的真实概率, 从而可以比较准确地估计出 RNN 满足性质的真实概率.

在实际应用中, 基于从 RNN 提取自动机的验证方法存在一些缺点: 其一是状态爆炸, 即从 RNN 中学习到的 DFA 可能太大, 无法显式构造. 另外, 由于 RNN 比自动机更有表现力, 从 RNN 提取到的自动机实际上不能完全刻画 RNN 的行为. 因此在自动机上的

反例不一定是 RNN 的真实反例, 此类提取近似自动机的验证算法是不完备的. 并且前述的文章中并没有定量评估 DFA 描述 RNN 实际行为的准确性. 此外, 这种从现有的 RNN 中提取自动机的方法通常只能验证一些正则性质, 存在一定的局限性.

相比于上述两种通过采样的方式来粗略估计 RNN 模型满足性质的概率,Mayr 等人 [61] 给出了性质的概率保证. 他们提出了一种基于黑盒模型和有界 $L^*(Bounded-L^*)$ 算法 [136] 的动态学习方法来验证 RNN 的各种性质, 并为模型提供了概率保证. 具体地, 若想验证一个 RNN $\mathcal{C}$ 是否满足性质 P, 可构造性质 $\Psi(\mathcal{C})=\mathcal{C}\cap\bar{P}$, 然后利用有界 L^* 算法学习出一个具有概率保证的 $\Psi(\mathcal{C})$ 的 (ϵ,δ)-近似自动机 $\mathcal{H}$. 若 $\mathcal{H}$ 是空自动机, 则能以至少为 $1-\delta$ 的置信度给出 RNN 模型 $\mathcal{C}$ 以不高于 ϵ 的错误率满足性质 P 的理论保证. 如果学出来的自动机 $\mathcal{H}$ 非空, 由于 L^* 算法会自动进行成员查询 (Membership Query) 查找到反例满足 $\mathcal{C}\cap\bar{P}$, 因此一定存在 RNN 模型 $\mathcal{C}$ 不满足性质 P 的反例.

虽然在运行有界 L^* 算法时, 也无法保证学出的自动机 $\mathcal{H}$ 完美地刻画性质, 但是此算法可以验证非正则性质, 例如检验两个 RNN 的等价性、验证巡航控制系统的正确性等, 并且他们给出了一定的阈值保证自动机提取算法的终止性.

Khmelnitsky [62] 等人讨论了二分类 RNN 相关的验证问题. 假定 RNN $\mathcal{C}$ 识别分类为正的字符串, 为了将 RNN 转化为有限自动机进行验证, 他们构造了一个用自动机 $\mathcal{A}$ 可以描述的性质, 即验证 $L(\mathcal{C})\subseteq L(\mathcal{A})$ 是否成立, 其中 $L(\mathcal{C})$ 和 $L(\mathcal{A})$ 分别表示网络 $\mathcal{C}$ 和自动机 $\mathcal{A}$ 接受的语言. 他们的验证方法本质上是将自动机学习 [126,136]、模型检验以及 L^* 算法结合起来的黑盒验证.L^* 算法基于给定的神经网络 $\mathcal{C}$ 进行成员查询, 产生一个假设的自动机 $\mathcal{H}$, 然后利用传统的模型检测方法 (SMC) [126] 检验学习出来的 $\mathcal{H}$ 是否满足 $L(\mathcal{H})\subseteq L(\mathcal{A})$, 再检验 $\mathcal{H}$ 是否满足性质 $L(\mathcal{C})\subseteq L(\mathcal{H})$. 如果有反例 w 使得上述性质不被满足, 算法将进一步检验该反例是真反例或假反例, 从而声明 $\mathcal{C}$ 不满足性质 $L(\mathcal{A})$ 或者将反例 w 返回 L^* 算法对自动机 $\mathcal{H}$ 进行精化 [137]. 他们的实验证明在精度和效率上相比于前人提出的 SMC [126] 和 AAMC 两种方法都具有比较明显的优势, 并且可以自动生成反例数据流.

不同于传统的模型检测, 这种将自动机学习与模型检验结合起来的方法总是可以产生一个反例数据流, 其刻画了反例的一般特征. 并且他们提出的 PDV 验证方法是黑盒验证方法, 不需要知道 RNN 的结构, 规避了 RNN 结构带来的复杂度很高的问题. 但因其结合了 L^* 算法和随机采样验证性质的 SMC 算法, 相较于简单的 SMC 算法, 复杂度往往更高.

考虑基于自动机提取来验证 RNN 是一种自然的想法, 提取自动机的过程可以利用很多现有的成熟的算法, 例如 L^* 算法以及其他量化分析的提取方法. 此类验证算法大多用采样的方式验证自动机满足某种性质的概率, 不能对性质提供一个概率保证.

下表 3 所示为 FNN 和 RNN 典型验证方法的优缺点分析和比较.

■ 6 RNN 应用场景中的其他性质验证

Mayr 等人 [61] 将黑盒验证和 L^* 算法结合起来, 通过对性质的补 $\bar{P}$ 学出一个自动机来验证 RNN 模型是否满足性质 P. 此类验证方法并不局限于正则性质, 还可以对多个领域的不同性质进行验证. 他们定义了上下文无关语言的一些相关性质, 例如验证 RNN 模型 $\mathcal{C}$ 识别的语言是否包含在上下文无关语言 S 中, 或者验证两个 RNN 模型的等价性等等. 对于实际应用中的大型控制系统, 他们也训练出了一个 RNN 来模拟巡航控制器软件模型, 并给出具体的性质 P 为 “当且仅当发生了操作 Gas 或 Acc 且其间无其他中断操作时, 执行中断操作.” 作者还将此类验证方法应用于电子商务领域以及生物信息领域, 为实际工业系统中的 RNN 模型验证奠定了基础. 此类验证方法具有很大的优越性, 不限制 RNN 模型大小以及内部参数, 对需要验证的性质也没有任何限制. 美中不足的是, 现实生活中很可能需要对时序性质进行验证, 目前还没有关于对时序逻辑如何进行验证的阐述.

目前 RNN 大多数的待验证属性都是学术界定义的, 和工程实践中所需要保证的性质可能会存在不小的差别. 所以, 为了缩短理论研究和实际应用的距离, 考虑 RNN 在实际应用中的性质很有研究价值.

近日,Bacciu [138] 等人提出了 RNN 网络在大型实际应用场景下, 特别是自动驾驶领域中相关性质的验证. 他们提出, 验证基于 RNN 的决策系统可以由以下三个步骤组成:(i) 利用 POPQORN 对 RNN 输入的扰动进行鲁棒性评估;(ii) 根据步骤 (i) 的鲁棒性验证结果, 设计一个由 RNN 和若干个冗余系统以及比较系统构成的评估系统. 冗余系统可以是与 RNN 系统输入相同的不同算法, 也可以是具有不同技术 (例如雷达或激光雷达) 的算法和传感器系统. 在基于 RNN 的系统和冗余系统计算出它们的输出数据之后, 比较系统负责对所取得的结果进行比较, 并基于 RNN 系统的输出是否可信来定义一套适当的安全措施, 即鲁棒性较差的模型需要更强的措施来避免灾难发生;(iii) 根据 “预期功能安全 (SOTIF) 指南” 对整个系统的功能进行广泛测试与验证, 并评估适当的安全性能指标, 根据确定的评估指标和相应的阈值, 以及测试长度来验证整个系统. 但他们仅在理论上提出了 RNN 在实际验证中面对的挑战的解决方法, 并未对测试用例进行实验, 可见验证实际应用中大型神经网络的安全并非易事.

RNN 在图像处理、数据分类、语音识别以及自治系统中作为控制器等方面应用广泛. 而在实际环境中, 通常需要部署在资源受限的平台 (如移动电话或嵌入式设备) 上, 这些部件的内存占用和能耗便成为了瓶颈问题, 所以人们的关注焦点转移到了利用一些启发式技术对这些 RNN f 进行压缩和优化. 然而, 这些所谓的压缩和优化技术并不能保证优化之后的网络 f' 的安全性, 包括针对对抗性输入的防御, 或者优化后的网络 f' 和原始网络 f 的等价性等等. 为了解决这个问题,Mohammadinejad 等人 [60] 另辟蹊径, 提

表 3: FNN、RNN 典型验证方法类别的优缺点比较

FNN 验证方法		
方法类别	优点	缺点
抽象解释	传播效率高, 得到可靠输出；能验证中等规模神经网络；在该框架下能同时考虑神经元之间的联系, 提高精度；支持多种激活函数和多种网络形式	抽象域在传播过程中会逐层累积误差, 容易产生虚假反例
符号传播	得到可靠输出；可拓展到非线性激活函数；能验证中等规模大小的网络	在逐层传播过程中也会逐层放大误差, 不支持 ResNet 等网络形式；部分验证算法复杂度较高
SMT/MILP	可靠且完备, 得到精确的输出	计算复杂度很高, 可验证神经网络规模很小；难以拓展到其他非线性激活函数以及多种网络形式
凸优化	得到可靠输出, 可以拓展到非线性激活函数以及卷积、残差等网络结构；能验证的网络规模大于精确验证算法	同样会逐层累积误差, 可能会产生假反例
网络结构抽象/CEGAR	得到可靠输出, 可以精化算法输出, 因此精度较高	一般不适用于非线性激活函数和卷积、残差等结构；通常需要结合其他验证方法；难以设定抽象神经元的标准；在网络结构抽象过程中需要对神经网络重新编码, 时间代价比较大, 可验证规模较小
Lipschitz	利用了神经网络的 Lipschitz 连续性, 对网络结构有良好的刻画	部分算法既不完备也不可靠, 没有严格的鲁棒保证, 得到的鲁棒边界比较宽松
RNN 验证方法		
方法类别	优点	缺点
抽象解释	传播速度高效, 可拓展性强；得到可靠的输出范围；可拓展到多种激活函数及网络结构	对于非线性激活函数的松弛可能过于宽松, 并且逐层累积误差
展开为 FNN	部分算法可靠且完备, 可以精确地求出 RNN 网络的可达集	展开过程繁琐, 时间代价大；需要利用现有的 FNN 验证方法；支持的激活函数和网络结构有限；可验证的网络规模较小
自动机提取	可以利用现有成熟方法提取自动机；有助于刻画 RNN 的可解释性以及反例数据的内在规律	自动机可能规模庞大, 难以显示构造；RNN 的表达能力比自动机更强, 提取得到的自动机通常是 RNN 的下近似；通过采样验证自动机满足某种性质的概率, 不能对性质提供严格的概率保证

出了一种对 RNN 进行微分验证的框架 DiffRNN, 这也是 RNN 的第一个微分验证方法, 目的是证明两个结构相似的神经网络的等价性, 其形式化地定义为:

定义 4 (RNN 的等价性)**.** *两个 RNN f 和 f' 是 ϵ-等价的, 如果 f 和 f' 在给定的输入集 X 上满足性质:* $\forall x \in X, |f'(x) - f(x)| < \epsilon$.

其中 ϵ 是衡量 RNN 之间差异的一个足够小的阈值. DiffRNN 也可以应对 RNN 中的非线性激活函数 Sigmoid 和 tanh 的计算、长序列的输入、以及门与状态之间的交互问题等挑战, 可以较为准确地刻画两个 RNN 之间输出的差距.

他们在实验部分与 POPQORN 验证工具对 Vanilla RNN 和 LSTM 的微分性质做了对比, 其中 LSTM 的规模最大为 3×64. 实验结果显示对于相同的阈值 ϵ,DiffRNN 的验证率以及验证时间相对于 POPQORN 都具有明显的优势. 然而更大一些规模的 4×128 的 LSTM 便超出了 DiffRNN 的验证能力, 因此, 对于结构更复杂的 RNN, 利用 DiffRNN 进行验证可能还需要更进一步的研究.

研究人员也做出了很多用于自然语言处理 (Natural Language Processing, NLP) 的 RNN 验证相关的工作. 例如 Zhang 等人 [139] 开发了一种提高 RNN 对抗任意字符串变换的鲁棒性训练方法, 产生的模型对字符串变换的攻击方式有更强的鲁棒性.Dong 等人 [140] 提出了对抗单词变换的鲁棒性训练方法, 他们将单词替换的攻击空间建模为一个凸包, 利用正则化项对实际的替换进行扰动, 并结合对抗性训练来提高鲁棒性.Xu 等人 [141] 开发了一个对任何神经网络结构进行扰动分析的自动化分析框架.Shi 等人 [142] 提出了一种结构类似于 RNN 的 transformer 的验证方法等等. 更多关于 NLP 领域模型的对抗攻防和鲁棒性分析方法可以参考文献综述 [143,144].

■ 7 神经网络的其他鲁棒性质验证

在前面的章节中, 我们详尽地介绍了 FNN 和 RNN 局部鲁棒性的验证方法, 而除了局部鲁棒性之外, 目前的研究还包括一些全局鲁棒性和概率鲁棒性验证的工作. 下式 (2) 展示了全局鲁棒性、概率鲁棒性和局部鲁棒性等安全性质之间的关系:

$$\text{安全性} = (\text{可达集} \models \text{性质 P ?}) \begin{cases} \text{鲁棒性} \begin{cases} \text{局部鲁棒性} \\ \text{全局鲁棒性} \\ \text{概率鲁棒性} \end{cases} \\ \text{其他性质} \end{cases} \tag{2}$$

局部鲁棒性仅体现出神经网络在单个样本上抗干扰的性质, 而*全局鲁棒性* (Global Robustness) 可以刻画出神经网络在整个输入集合中不同类别之间的稳定性. 我们采用文献 [145] 中定义的全局鲁棒性. 在此先定义符号 $\perp$, 它表示一个特殊的类别 (不在正常类别集合 C 中), 用于处理对分类结果置信度不足的情形. 直观上如下图 12 所示, 在输入样本中不同类别之间存在一个宽度至少为 ϵ 的决策边界, 该决策边界中的点集经过神经网络 f 预测后的结果为 $\perp$. 我们引入关系 $\overset{\perp}{=}$:$c_1 \overset{\perp}{=} c_2$ 成立, 如果 $c_1 = \perp$ 或者 $c_2 = \perp$ 或

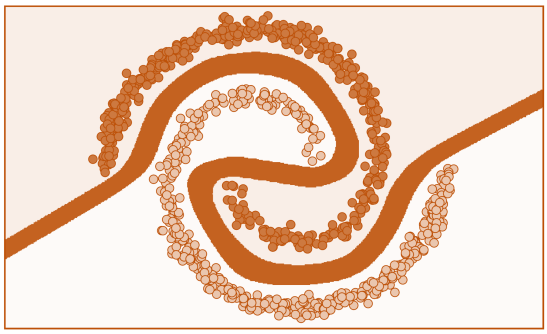

图 12: 全局鲁棒性示意图, 其中橙色的分割线即为宽度为 ϵ 的决策边界, 将输入区域分为不同的两个类别, 分别用浅橙色和深橙色表示. 不同颜色的圆点分别代表不同类别的样本点.

者 $c_1 = c_2$. 为了满足全局鲁棒性, 我们定义距离在 ϵ 范围内的两个点不会被分类为两个不同的非 $\perp$ 类, 形式化地描述全局鲁棒性即为:

定义 5 (全局鲁棒性). *给定一个神经网络 f 和输入集合 X, 我们称神经网络 f 在输入集 X 上是 ϵ-全局鲁棒的, 如果*

$$\forall x_1, x_2 \in X, \|x_1 - x_2\|_p \leqslant \epsilon \Rightarrow C_f(x_1) \overset{\perp}{=} C_f(x_2).$$

关于全局鲁棒性验证的工作目前比较少,Ruan 等人 [146] 将神经网络的全局鲁棒性定义为测试数据集上最大鲁棒半径的期望, 并开发出了高效的验证算法; Yang 等人 [96] 提出了首个具有严格可靠性保证的全局鲁棒性高效验证算法; Sun 等人也提出一种全局鲁棒性的验证方法 DeepGlobal [147], 可以展示出 FNN 的决策边界. 通过这些边界可以帮助搜索神经网络的*对抗性危险区域* (Adversarial Dangerous Regions, ADR), 即容易受到微小对抗性扰动影响的输入区域.

要保证样本 x_0 的扰动区域内所有的样本都严格满足鲁棒性的定义并不是一件容易的事情. 因此在实际应用中, 如果扰动后的样本与原样本的输出结果一致的比例相当高, 我们也可近似地认为神经网络关于原样本 x_0 是鲁棒的, 即神经网络以较高的置信度在扰动区域内满足鲁棒性质. 由此延伸出了*概率鲁棒性* (Quantitative/Probabilistic Robustness) 的定义:

定义 6 (概率鲁棒性). *给定一个神经网络 $f : \mathbb{R}^m \to \mathbb{R}^n$, 输入样本 x_0, 错误率 $\epsilon \in [0, 1]$ 以及 l_p 范数下半径为 r 的扰动空间 $B_p(x_0, r)$ 和 $B_p(x_0, r)$ 上的 Borel 概率测度 $\mathbb{P}$, 如果神经网络 f 满足性质*[5]

$$\mathbb{P}(\{x \in B_p(x_0, r) \mid C_f(x) = C_f(x_0)\}) \geq 1-\epsilon,$$

则称神经网络 f 在样本点 x_0 处关于扰动半径 r 和概率测度 $\mathbb{P}$ 具有 ϵ-概率鲁棒性.

相比于局部鲁棒性, 概率鲁棒性的验证在实际应用中也是不可或缺的, 例如在自动驾驶安全标准中通常会使用事故率来衡量自动驾驶车辆的安全性 [148]. 在神经网络的概率鲁棒性验证方面, 比较有代表性的工作为 Baluta 等人提出的 PROVERO 工具 [149],

[5]本文中讨论的神经网络 f 均是连续函数, 因此其必然是 Borel 可测函数.

将待验证的模型视为黑盒, 并基于输入空间的分布进行采样来确定神经网络的概率鲁棒性, 该算法能在一小时内验证 ImageNet 数据集相关的 VGG16、ResNet50 等大型网络的概率鲁棒性质.Li 等人指出, 概率鲁棒性对于输入维度很高的深度神经网络 (如图像分类任务的神经网络) 和严格意义上的局部鲁棒性往往差别很大, 不能真实体现神经网络在实际使用中的鲁棒性质, 并进一步提出了概率近似正确模型鲁棒性 (PAC-model Robustness) 的概念 [150], 从而通过模型学习更好地刻画高输入维度的深度神经网络的概率鲁棒性. 其他概率鲁棒性的验证工作还包括 [151,152] 等.

■ 8 未来的研究方向

本文主要介绍了 FNN 和 RNN 的鲁棒性验证相关的研究, 并探讨了二者之间的内在联系. 我们通过分析发现目前学术界对 FNN 的局部鲁棒性验证方法层出不穷, 而对于 RNN 鲁棒性质的研究和其他类型鲁棒性质的验证还处于起步阶段, 相关的验证方法目前还比较少. 然而从整体上看, 目前这些形式化验证方法能处理的神经网络规模可以拓展到 VGG、ResNet 等大型网络, 但是对工业界中的大型复杂系统的验证仍然存在瓶颈. 为了神经网络的鲁棒性验证领域的进一步发展, 我们归纳得出几个未来可行的研究方向:

- 研究局部鲁棒性之外的其他性质, 例如概率鲁棒性、几何鲁棒性、公平性、安全性、可解释性等等. 目前验证神经网络鲁棒性的研究主要集中在基于 l_p 范数的像素级别扰动上. 然而在实际应用中, 计算机视觉系统很可能会面临旋转、平移或者缩放等不同种类的攻击或扰动, 因此保证神经网络的安全性就需要对此类几何变换的鲁棒性进行研究. 例如 Balunovic 等人已经提出 DeepG [153] 用以验证神经网络的几何鲁棒性. 还有的研究集中在神经网络的概率鲁棒性, 即不要求在扰动区域下完全鲁棒, 而仅需保证足够高的概率使得神经网络在该扰动范围下仍然做出正确决策, 相关的研究包括 [154,155] 等. 在应用神经网络决策系统时, 我们希望神经网络的决策不应受到性别、年龄和种族等特征的影响, 因此需要对神经网络的公平性进行形式化规约和定量刻画; 我们还希望神经网络具有隐私性, 即攻击者无法窃取到用户的隐私; 我们还应当保证神经网络的决策足够安全并且可信赖, 因此需要对神经网络的可解释性进行研究, 目前已经有不少相关的工作 [156–158].

- 换一种的角度验证神经网络. 不管是神经网络的鲁棒性验证还是程序正确性分析, 其本质上都属于模型检验领域, 因此 Liu 等人 [159] 另辟蹊径, 从模型检验的角度出发, 对 ReLU 激活的神经网络提出了一种 ReLU 时序逻辑 (ReLU Temporal Logic, ReTL) 并定义了相关的语义, 对神经网络的可达性等性质进行验证, 是一种非常新颖的验证模型鲁棒性的方法. Guo 等人 [160] 从分治算法的角度, 提出了基于分类置信度排序的神经网络鲁棒性验证方法, 将神经网络鲁棒性验证问题转化到若干定向类型验证的子问题, 有效提升了多个开源验证工具的鲁棒性证伪性能. 因此, 从不同的角度验证神经网络的鲁棒性也会有很大的发展空间.

- 研究其他类型神经网络的鲁棒性. 在大型的工业系统包括运用比较广泛的神经网络除了深度神经网络和 RNN 之外还有 CNN 和图网络等. 深度神经网络的鲁棒性验证已经有了比较丰富的研究, 但其他种类的网络鲁棒性研究比较少, 仍然有很大的研究空间.Zhang 等人 [161] 提出了以 Swish 函数作为激活函数的神经网络的鲁棒性验证方法, 他们利用线性逼近技术将鲁棒性验证问题转化为约束求解问题, 并对非线性激活函数进行线性逼近, 进一步验证 Swish 神经网络的鲁棒性.Zhang 等人 [162] 提出了一般量化神经网络 (Quantized Neural Network, QNN) 的 MILP 建模方法, 并基此设计了一般量化神经网络鲁棒性分析框架, 实验结果显示比此前基于 SMT 的鲁棒性验证方法在效率上具有明显的优势.

- 研究大型智能系统的安全性. 目前各种各样的神经网络被广泛应用于安全攸关系统中, 例如自动驾驶, 飞机巡航控制等等, 保证神经网络组件的安全性或者鲁棒性并不足以保证整个智能系统的安全性. 对智能系统整体的安全性研究已经存在部分工作, 包括 Dreossi 等人提出的 VerifAI [163,164] 工具,Kazak 利用深度神经网络验证相关思想, 提出了一种验证深度强化学习 (Deep Reinforcement Learning, DRL) 系统的验证工具 Verily [80],Zhou 等人为多智能体学习框架 MFAC 提出了一个鲁棒性训练框架 RoMFAC [165], 使其具有良好的对抗鲁棒性, 以及其他的智能系统验证工具 NNV [122],Percemon [166] 等等, 对自动驾驶和无人机的运行场景进行验证. 因此, 对此类智能系统的鲁棒性、安全性验证也有很大的发展空间.

■ 9 结语

神经网络的鲁棒性研究在形式化验证领域正处于快速发展的时期, 随着越来越多的智能系统应用于现实生活, 神经网络的鲁棒性验证也将变得更加重要. 虽然 FNN 的鲁棒性研究工作林林总总, 但是目前的研究仍然停留在中等规模的网络上验证精度与效率的权衡, 而难以拓展到实际应用中的大型神经网络中. 并且除了局部鲁棒性, 全局鲁棒性和概率鲁棒性等其他鲁棒性质的验证工作目前还比较匮乏. 同时, 我们对神经网络的预测机制和原理仍然不甚了解, 难以将这些形式化的验证方法应用于工业界含有神经网络等智能组件的大型智能系统中. RNN 因为具有更复杂的结构, 对其局部鲁棒性的研究工作目前仍比较少, 能验证的神经网络规模仅包含几百上千个神经元. 本文分析了大量的 FNN 和 RNN 验证的相关文章, 目的是梳理 FNN 和 RNN 验证方法的发展现状, 并揭示

两种神经网络鲁棒性验证方法之间存在的内在联系. 我们在最后探讨了神经网络的一些其他鲁棒性的定义与验证方法, 以及神经网络鲁棒性验证领域在未来具有潜力的研究方向, 希望可以进一步推动神经网络鲁棒性验证的研究.

References:

[1] Allen Newell, Herbert Alexander Simon, et al. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ, 1972.

[2] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[3] Filippo Amato, Alberto López, Eladia María Peña-Méndez, Petr Vaňhara, Aleš Hampl, and Josef Havel. Artificial neural networks in medical diagnosis, 2013.

[4] Zachary Chase Lipton, David C. Kale, Charles Elkan, and Randall C. Wetzel. Learning to diagnose with LSTM recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[5] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[6] Moustapha M Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. *Advances in neural information processing systems*, 30, 2017.

[7] 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, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016.

[8] Nicolas Scheiner, Nils Appenrodt, Jürgen Dickmann, and Bernhard Sick. Radar-based road user classification and novelty detection with recurrent neural network ensembles. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 722–729. IEEE, 2019.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[10] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[12] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[13] Qinglong Wang, Wenbo Guo, Kaixuan Zhang, Alexander G Ororbia, Xinyu Xing, Xue Liu, and C Lee Giles. Adversary resistant deep neural networks with an application to malware detection. In *Proceedings of the 23rd ACM sigkdd international conference on knowledge discovery and data mining*, pages 1145–1153, 2017.

[14] Zhenlong Yuan, Yongqiang Lu, Zhaoguo Wang, and Yibo Xue. Droid-sec: deep learning in android malware detection. In *Proceedings of the 2014 ACM conference on SIGCOMM*, pages 371–372, 2014.

[15] Sanjit A. Seshia, Dorsa Sadigh, and S. Shankar Sastry. Toward verified artificial intelligence. *Commun. ACM*, 65(7):46–55, 2022.

[16] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[17] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[18] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society, 2017.

[19] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[20] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Attentive explanations: Justifying decisions and pointing to the evidence (extended abstract). *CoRR*, abs/1711.07373, 2017.

[21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[22] Luca Pulina and Armando Tacchella. An abstraction-refinement approach to verification of artificial neural networks. In *International Conference on Computer Aided Verification*, pages 243–257. Springer, 2010.

[23] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2018.

[24] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3578–3586. PMLR, 2018.

[25] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. *Advances in neural information processing systems*, 31, 2018.

[26] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. Boosting robustness certification of neural networks. In *International conference on learning representations*, 2018.

[27] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–30, 2019.

[28] Gagandeep Singh, Rupanshu Ganvir, Markus Püschel, and Martin Vechev. Beyond the single neuron convex barrier for neural network certification. 2019.

[29] Christoph Müller, François Serre, Gagandeep Singh, Markus Püschel, and Martin T. Vechev. Scaling polyhedral neural network verification on gpus. In Alex Smola, Alex Dimakis, and Ion Stoica, editors, *Proceedings of Machine Learning and Systems 2021, MLSys 2021, virtual, April 5-9, 2021*. mlsys.org, 2021.

[30] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Formal security analysis of neural networks using symbolic intervals. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1599–1614, Baltimore, MD, August 2018. USENIX Association.

[31] Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, pages 5276–5285. PMLR, 2018.

[32] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[33] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[34] Sven Gowal, Krishnamurthy Dj Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. Scalable verified training for provably robust image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4842–4851, 2019.

[35] Akhilan Boopathy, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3240–3247, 2019.

[36] Jianlin Li, Jiangchao Liu, Pengfei Yang, Liqian Chen, Xiaowei Huang, and Lijun Zhang. Analyzing deep neural networks with symbolic propagation: Towards higher precision and faster verification. In Bor-Yuh Evan Chang, editor, *Static Analysis - 26th International Symposium, SAS 2019, Porto, Portugal, October 8-11, 2019, Proceedings*, volume 11822 of *Lecture Notes in Computer Science*, pages 296–319. Springer, 2019.

[37] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019.

[38] Zhaoyang Lyu, Ching-Yun Ko, Zhifeng Kong, Ngai Wong, Dahua Lin, and Luca Daniel. Fastened crown: Tightened neural network robustness certificates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5037–5044, 2020.

[39] Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *International conference on computer aided verification*, pages 3–29. Springer, 2017.

[40] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[41] Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 269–286. Springer, 2017.

[42] Divya Gopinath, Guy Katz, Corina S. Pasareanu, and Clark Barrett. Deepsafe: A data-driven approach for checking adversarial robustness in neural networks, 2017.

[43] Guy Katz, Derek A Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, et al. The marabou framework for verification and analysis of deep neural networks. In *International Conference on Computer Aided Verification*, pages 443–452. Springer, 2019.

[44] Vincent Tjeng, Kai Yuanqing Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[45] Souradeep Dutta, Susmit Jha, Sriram Sankaranarayanan, and Ashish Tiwari. Output range analysis for deep feedforward neural networks. In *NASA Formal Methods Symposium*, pages 121–138. Springer, 2018.

[46] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

[47] Mark Niklas Müller, Gleb Makarchuk, Gagandeep Singh, Markus Püschel, and Martin T. Vechev. PRIMA: general and precise neural network certification via scalable convex hull approximations. *Proc. ACM Program. Lang.*, 6(POPL):1–33, 2022.

[48] Greg Anderson, Shankara Pailoor, Isil Dillig, and Swarat Chaudhuri. Optimization and abstraction: a synergistic approach for analyzing neural network robustness. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 731–744, 2019.

[49] Pengfei Yang, Renjue Li, Jianlin Li, Cheng-Chao Huang, Jingyi Wang, Jun Sun, Bai Xue, and Lijun Zhang. Improving neural network verification through spurious region guided refinement. In Jan Friso Groote and Kim Guldstrand Larsen, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings, Part I*, volume 12651 of *Lecture Notes in Computer Science*, pages 389–408. Springer, 2021.

[50] Pranav Ashok, Vahid Hashemi, Jan Křetínskỳ, and Stefanie Mohr. Deepabstract: neural network abstraction for accelerating verification. In *International Symposium on Automated Technology for Verification and Analysis*, pages 92–107. Springer, 2020.

[51] Yizhak Yisrael Elboher, Justin Gottschlich, and Guy Katz. An abstraction-based framework for neural network verification. In *International Conference on Computer Aided Verification*, pages 43–65. Springer, 2020.

[52] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Advances in neural information processing systems*, 30, 2017.

[53] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[54] Ching-Yun Ko, Zhaoyang Lyu, Lily Weng, Luca Daniel, Ngai Wong, and Dahua Lin. Popqorn: Quantifying robustness of recurrent neural networks. In *International Conference on Machine Learning*, pages 3468–3477. PMLR, 2019.

[55] Tianyu Du, Shouling Ji, Lujia Shen, Yao Zhang, Jinfeng Li, Jie Shi, Chengfang Fang, Jianwei Yin, Raheem Beyah, and Ting Wang. Cert-rnn: Towards certifying the robustness of recurrent neural networks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 516–534, 2021.

[56] Wonryong Ryou, Jiayu Chen, Mislav Balunovic, Gagandeep Singh, Andrei Dan, and Martin Vechev. Scalable polyhedral verification of recurrent neural networks. In *International Conference on Computer Aided Verification*, pages 225–248. Springer, 2021.

[57] Michael E Akintunde, Andreea Kevorchian, Alessio Lomuscio, and Edoardo Pirovano. Verification of rnn-based neural agent-environment systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6006–6013, 2019.

[58] Hongce Zhang, Maxwell Shinn, Aarti Gupta, Arie Gurfinkel, Nham Le, and Nina Narodytska. Verification of recurrent neural networks for cognitive tasks via reachability analysis. In *ECAI 2020*, pages 1690–1697. IOS Press, 2020.

[59] Dmitry Vengertsev and Elena Sherman. Recurrent neural network properties and their verification with monte carlo techniques. In Huáscar Espinoza, José Hernández-Orallo, Xin Cynthia Chen, Seán S. ÓhÉigeartaigh, Xiaowei Huang, Mauricio Castillo-Effen, Richard Mallah, and John Alexander McDermid, editors, *Proceedings of the Workshop on Artificial Intelligence Safety, co-located with 34th AAAI Conference on Artificial Intelligence, SafeAI@AAAI 2020, New York City, NY, USA, February 7, 2020*, volume 2560 of *CEUR Workshop Proceedings*, pages 178–185. CEUR-WS.org, 2020.

[60] Sara Mohammadinejad, Brandon Paulsen, Jyotirmoy V Deshmukh, and Chao Wang. Diffrnn: Differential verification of recurrent neural networks. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 117–134. Springer, 2021.

[61] Franz Mayr, Sergio Yovine, and Ramiro Visca. Property checking with interpretable error characterization for recurrent neural networks. *Machine Learning and Knowledge Extraction*, 3(1):205–227, 2021.

[62] Igor Khmelnitsky, Daniel Neider, Rajarshi Roy, Xuan Xie, Benoît Barbot, Benedikt Bollig, Alain Finkel, Serge Haddad, Martin Leucker, and Lina Ye. Property-directed verification and robustness certification of recurrent neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 364–380. Springer, 2021.

[63] Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. Reachability analysis of deep neural networks with provable guarantees. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 2651–2659. ijcai.org, 2018.

[64] Rudy R Bunel, Ilker Turkaslan, Philip Torr, Pushmeet Kohli, and Pawan K Mudigonda. A unified view of piecewise linear neural network verification. *Advances in Neural Information Processing Systems*, 31, 2018.

[65] Matteo Fischetti and Jason Jo. Deep neural networks as 0-1 mixed integer linear programs: A feasibility study. *CoRR*, abs/1712.06174, 2017.

[66] Matteo Fischetti and Jason Jo. Deep neural networks and mixed integer linear optimization. *Constraints*, 23(3):296–309, 2018.

[67] Yedi Zhang, Zhe Zhao, Guangke Chen, Fu Song, and Taolue Chen. Bdd4bnn: a bdd-based quantitative analysis framework for binarized neural networks. In *International Conference on Computer Aided Verification*, pages 175–200. Springer, 2021.

[68] Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. Training verified learners with learned verifiers. *CoRR*, abs/1805.10265, 2018.

[69] Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. In *UAI*, volume 1, page 3, 2018.

[70] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[71] Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in Neural Information Processing Systems*, 31, 2018.

[72] Pavithra Prabhakar and Zahra Rahimi Afzal. Abstraction based output range analysis for neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[73] Aman Sinha, Hongseok Namkoong, and John C. Duchi. Certifiable distributional robustness with principled adversarial training. *CoRR*, abs/1710.10571, 2017.

[74] Igor Griva, Stephen G Nash, and Ariela Sofer. *Linear and nonlinear optimization*, volume 108. Siam, 2009.

[75] Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. Measuring neural net robustness with constraints. *Advances in neural information processing systems*, 29:2613–2621, 2016.

[76] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *Mach. Learn.*, 107(3):481–508, 2018.

[77] Karsten Scheibler, Leonore Winterer, Ralf Wimmer, and Bernd Becker. Towards verification of artificial neural networks. In Ulrich Heinkel, Daniel Kriesten, and Marko Rößler, editors, *Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen, MBMV 2015, Chemnitz, Germany, March 3-4, 2015*, pages 30–40. Sächsische Landesbibliothek, 2015.

[78] Nina Narodytska, Shiva Kasiviswanathan, Leonid Ryzhyk, Mooly Sagiv, and Toby Walsh. Verifying properties of binarized deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[79] Kyle D Julian, Jessica Lopez, Jeffrey S Brush, Michael P Owen, and Mykel J Kochenderfer. Policy compression for aircraft collision avoidance systems. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–10. IEEE, 2016.

[80] Yafim Kazak, Clark Barrett, Guy Katz, and Michael Schapira. Verifying deep-rl-driven systems. In *Proceedings of the 2019 Workshop on Network Meets AI & ML*, pages 83–89, 2019.

[81] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Towards proving the adversarial robustness of deep neural networks. *Electronic Proceedings in Theoretical Computer Science*, 257:19–26, 2017.

[82] Weiming Xiang, Hoang-Dung Tran, and Taylor T. Johnson. Output reachable set estimation and verification for multilayer neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5777–5783, 2018.

[83] Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward relu neural networks. *CoRR*, abs/1706.07351, 2017.

[84] Chih-Hong Cheng, Georg Nührenberg, and Harald Ruess. Maximum resilience of artificial neural networks. In Deepak D'Souza and K. Narayan Kumar, editors, *Automated Technology for Verification and Analysis*, pages 251–268, Cham, 2017. Springer International Publishing.

[85] Luca Pulina and Armando Tacchella. Checking safety of neural networks with SMT solvers: A comparative evaluation. In Roberto Pirrone and Filippo Sorbello, editors, *AI*IA 2011: Artificial Intelligence Around Man and Beyond - XIIth International Conference of the Italian Association for Artificial Intelligence, Palermo, Italy, September 15-17, 2011. Proceedings*, volume 6934 of *Lecture Notes in Computer Science*, pages 127–138. Springer, 2011.

[86] Luca Pulina and Armando Tacchella. Challenging smt solvers to verify neural networks. *AI Commun.*, 25:117–135, 2012.

[87] Khalil Ghorbal, Eric Goubault, and Sylvie Putot. The zonotope abstract domain taylor1+. In Ahmed Bouajjani and Oded Maler, editors, *Computer Aided Verification*, pages 627–633, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[88] Matthew Mirman, Gagandeep Singh, and Martin T. Vechev. A provable defense for deep residual networks. *CoRR*, abs/1903.12519, 2019.

[89] Suraj Srinivas and R. Venkatesh Babu. Data-free parameter pruning for deep neural networks. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015*, pages 31.1–31.12. BMVA Press, 2015.

[90] Guoqiang Zhong, Hui Yao, and Huiyu Zhou. Merging neurons for structure compression of deep networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1462–1467. IEEE, 2018.

[91] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[92] Peng Jin, Jiaxu Tian, Dapeng Zhi, Xuejun Wen, and Min Zhang. : A cegar-driven training and verification framework for safe deep reinforcement learning. In *International Conference on Computer Aided Verification*, pages 193–218. Springer, 2022.

[93] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy A. Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018.

[94] Eric Wong, Frank R. Schmidt, Jan Hendrik Metzen, and J. Zico Kolter. Scaling provable adversarial defenses. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8410–8419, 2018.

[95] Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A convex relaxation barrier to tight robustness verification of neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9832–9842, 2019.

[96] Pengfei Yang, Jianlin Li, Jiangchao Liu, Cheng-Chao Huang, Renjue Li, Liqian Chen, Xiaowei Huang, and Lijun Zhang. Enhancing robustness verification for deep neural networks via symbolic propagation. *Formal Aspects Comput.*, 33(3):407–435, 2021.

[97] Renjue Li, Jianlin Li, Cheng-Chao Huang, Pengfei Yang, Xiaowei Huang, Lijun Zhang, Bai Xue, and Holger Hermanns. Prodeep: a platform for robustness verification of deep neural networks. In Prem Devanbu, Myra B. Cohen, and Thomas Zimmermann, editors, *ESEC/FSE '20: 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Virtual Event, USA, November 8-13, 2020*, pages 1630–1634. ACM, 2020.

[98] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[99] Jason Hull, David Ward, and Radoslaw R Zakrzewski. Verification and validation of neural networks for safety-critical applications. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, volume 6, pages 4789–4794. IEEE, 2002.

[100] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Aurelie Lozano, Cho-Jui Hsieh, and Luca Daniel. On extensions of clever: A neural network robustness evaluation algorithm. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1159–1163. IEEE, 2018.

[101] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[102] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[103] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[104] Changliu Liu, Tomer Arnon, Christopher Lazarus, Christopher Strong, Clark Barrett, Mykel J Kochenderfer, et al. Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization*, 4(3-4):244–404, 2021.

[105] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020.

[106] 纪守领, 杜天宇, 邓水光, 程鹏, 时杰, 杨珉, and 李博. 深度学习模型鲁棒性研究综述. 计算机学报, 45(1), 2022.

[107] calvin Feng. Vanilla recurrent neural network. http://calvinfeng.gitbook.io/machine-learning-notebook/supervised-learning/recurrent-neural-network/recurrent_neural_networks, June 2021.

[108] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[109] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL, 2014.

[110] Shuohang Wang and Jing Jiang. Learning natural language inference with LSTM. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1442–1451. The Association for Computational Linguistics, 2016.

[111] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.

[112] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[113] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

[114] Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 3506–3510, 2017.

[115] Olah Christopher. Understanding lstm networks. 2015.

[116] Qinglong Wang, Kaixuan Zhang, Xue Liu, and C. Lee Giles. Verification of recurrent neural networks through rule extraction. *CoRR*, abs/1811.06029, 2018.

[117] Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. pages 238–252, 01 1977.

[118] Yuval Jacoby, Clark Barrett, and Guy Katz. Verifying recurrent neural networks using invariant inference. In *International Symposium on Automated Technology for Verification and Analysis*, pages 57–74. Springer, 2020.

[119] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 49–54. IEEE, 2016.

[120] Minhao Cheng, Jinfeng Yi, Pin-Yu Chen, Huan Zhang, and Cho-Jui Hsieh. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3601–3608, 2020.

[121] Michael Akintunde, Alessio Lomuscio, Lalit Maganti, and Edoardo Pirovano. Reachability analysis for neural agent-environment systems. In *Sixteenth international conference on principles of knowledge representation and reasoning*, 2018.

[122] Hoang-Dung Tran, Xiaodong Yang, Diego Manzanas Lopez, Patrick Musau, Luan Viet Nguyen, Weiming Xiang, Stanley Bak, and Taylor T Johnson. Nnv: the neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In *International Conference on Computer Aided Verification*, pages 3–17. Springer, 2020.

[123] John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65, 2001.

[124] Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

[125] Edmund M Clarke. Model checking. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 54–56. Springer, 1997.

[126] Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. In *International Conference on Machine Learning*, pages 5247–5256. PMLR, 2018.

[127] Xiyue Zhang, Xiaoning Du, Xiaofei Xie, Lei Ma, Yang Liu, and Meng Sun. Decision-guided weighted automata extraction from recurrent neural networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 11699–11707. AAAI Press, 2021.

[128] Stéphane Ayache, Rémi Eyraud, and Noé Goudian. Explaining black boxes on sequential data using weighted automata. In *International Conference on Grammatical Inference*, pages 81–103. PMLR, 2019.

[129] Takamasa Okudono, Masaki Waga, Taro Sekiyama, and Ichiro Hasuo. Weighted automata extraction from recurrent neural networks via regression on state spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5306–5314, 2020.

[130] Henrik Jacobsson. Rule extraction from recurrent neural networks: Ataxonomy and review. *Neural Computation*, 17(6):1223–1263, 2005.

[131] Qinglong Wang, Kaixuan Zhang, Alexander G Ororbia II, Xinyu Xing, Xue Liu, and C Lee Giles. An empirical evaluation of rule extraction from recurrent neural networks. *Neural computation*, 30(9):2568–2591, 2018.

[132] C. L. Giles, C. B. Miller, D. Chen, H. H. Chen, G. Z. Sun, and Y. C. Lee. Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation*, 4(3):393–405, 1992.

[133] Masaru Tomita. Dynamic construction of finite-state automata from examples using hill-climbing. In *Proceedings of the Fourth Annual Conference of the Cognitive Science Society*, pages 105–108, 1982.

[134] C Giles, Guo-Zheng Sun, Hsing-Hen Chen, Yee-Chun Lee, and Dong Chen. Higher order recurrent networks and grammatical inference. *Advances in neural information processing systems*, 2, 1989.

[135] Jan Tretmans. Conformance testing with labelled transition systems: Implementation relations and test generation. *Computer networks and ISDN systems*, 29(1):49–79, 1996.

[136] Franz Mayr and Sergio Yovine. Regular inference on artificial neural networks. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 350–369. Springer, 2018.

[137] Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement. In *International Conference on Computer Aided Verification*, pages 154–169. Springer, 2000.

[138] Davide Bacciu, Antonio Carta, Daniele Di Sarli, Claudio Gallicchio, Vincenzo Lomonaco, and Salvatore Petroni. Towards functional safety compliance of recurrent neural networks. In *CAIP 2021: Proceedings of the 1st International Conference on AI for People: Towards Sustainable AI, CAIP 2021, 20-24 November 2021, Bologna, Italy*, page 86. European Alliance for Innovation, 2021.

[139] Yuhao Zhang, Aws Albarghouthi, and Loris D'Antoni. Certified robustness to programmable transformations in lstms. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 1068–1083. Association for Computational Linguistics, 2021.

[140] Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. Towards robustness against natural language word substitutions. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[141] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33:1129–1141, 2020.

[142] Zhouxing Shi, Huan Zhang, Kai-Wei Chang, Minlie Huang, and Cho-Jui Hsieh. Robustness verification for transformers. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[143] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41, 2020.

[144] 郑海斌, 陈晋音, 章燕, 张旭鸿, 葛春鹏, 刘哲, 欧阳亦可, and 纪守领. 面向自然语言处理的对抗攻防与鲁棒性分析综述 [j]. 计算机研究与发展, 58(8):1727–1750, 2021.

[145] Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-robust neural networks. In *International Conference on Machine Learning*, pages 6212–6222. PMLR, 2021.

[146] Wenjie Ruan, Min Wu, Youcheng Sun, Xiaowei Huang, Daniel Kroening, and Marta Kwiatkowska. Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance. International Joint Conferences on Artificial Intelligence Organization, 2019.

[147] Weidi Sun, Yuteng Lu, Xiyue Zhang, and Meng Sun. Deepglobal: A framework for global robustness verification of feedforward neural networks. *Journal of Systems Architecture*, page 102582, 2022.

[148] Nidhi Kalra and Susan M Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[149] Teodora Baluta, Zheng Leong Chua, Kuldeep S Meel, and Prateek Saxena. Scalable quantitative verification for deep neural networks. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pages 312–323. IEEE, 2021.

[150] Renjue Li, Pengfei Yang, Cheng-Chao Huang, Youcheng Sun, Bai Xue, and Lijun Zhang. Towards practical robustness analysis for dnns based on pac-model learning. In *44th IEEE/ACM 44th International Conference on Software Engineering, ICSE 2022, Pittsburgh, PA, USA, May 25-27, 2022*, pages 2189–2201. ACM, 2022.

[151] Ravi Mangal, Aditya V Nori, and Alessandro Orso. Robustness of neural networks: A probabilistic and practical approach. In *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, pages 93–96. IEEE, 2019.

[152] Stefan Webb, Tom Rainforth, Yee Whye Teh, and M. Pawan Kumar. A statistical approach to assessing neural network robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[153] Mislav Balunovic, Maximilian Baader, Gagandeep Singh, Timon Gehr, and Martin T. Vechev. Certifying geometric robustness of neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15287–15297, 2019.

[154] Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, and Andrea Patane. Robustness guarantees for bayesian inference with gaussian processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7759–7768, 2019.

[155] Renjue Li, Pengfei Yang, Cheng-Chao Huang, Youcheng Sun, Bai Xue, and Lijun Zhang. Towards practical robustness analysis for dnns based on pac-model learning. In *International Conference on Software Engineering (ICSE)*, 2021.

[156] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[157] Or Biran and Courtenay Cotton. Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, volume 8, pages 8–13, 2017.

[158] Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pages 559–560, 2018.

[159] Wan-Wei Liu, Fu Song, Tang-Hao-Ran Zhang, and Ji Wang. Verifying relu neural networks from a model checking perspective. *Journal of Computer Science and Technology*, 35(6):1365–1381, 2020.

[160] Xingwu Guo, Wenjie Wan, Zhaodi Zhang, Min Zhang, Fu Song, and Xuejun Wen. Eager falsification for accelerating robustness verification of deep neural networks. In *Proceedings of the 32nd IEEE International Symposium on Software Reliability Engineering*, pages 345–356, 2021.

[161] Zhaodi Zhang, Jing Liu, Guanjun Liu, Jiacun Wang, and John Zhang. Robustness verification of swish neural networks embedded in autonomous driving systems. *IEEE Transactions on Computational Social Systems*, 2022.

[162] Yedi Zhang, Zhe Zhao, Guangke Chen, Fu Song, Min Zhang, and Taolue Chen. Qvip: An ilp-based formal verification approach for quantized neural networks. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, 2022.

[163] Tommaso Dreossi, Daniel J. Fremont, Shromona Ghosh, Edward Kim, Hadi Ravanbakhsh, Marcell Vazquez-Chanlatte, and Sanjit A. Seshia. Verifai: A toolkit for the formal design and analysis of artificial intelligence-based systems. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I*, volume 11561 of *Lecture Notes in Computer Science*, pages 432–442. Springer, 2019.

[164] Tommaso Dreossi, Alexandre Donzé, and Sanjit A Seshia. Compositional falsification of cyber-physical systems with machine learning components. *Journal of Automated Reasoning*, 63(4):1031–1053, 2019.

[165] Ziyuan Zhou and Guanjun Liu. Romfac: A robust mean-field actor-critic reinforcement learning against adversarial perturbations on states. *CoRR*, abs/2205.07229, 2022.

[166] Anand Balakrishnan, Jyotirmoy Deshmukh, Bardh Hoxha, Tomoya Yamaguchi, and Georgios Fainekos. Percemon: Online monitoring for perception systems. In *International Conference on Runtime Verification*, pages 297–308. Springer, 2021.