计算机科学
COMPUTER SCIENCE

一种约束验证神经网络的方法

郜玉钊，邢云汉，刘嘉祥

引用本文

郜玉钊，邢云汉，刘嘉祥. 一种约束验证神经网络的方法[J]. 计算机科学，2023，50(11A)：221000045-5.

GAO Yuzhao, XING Yunhan, LIU Jiaxiang. Constraint-based Verification Method for Neural Networks [J]. Computer Science, 2023, 50(11A): 221000045-5.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.221000045

一种约束验证神经网络的方法

郜玉钊　邢云汉　刘嘉祥
深圳大学计算机与软件学院　广东 深圳 518060
(1079330450@qq.com)

摘　要　神经网络的验证一直是人工智能领域的主要挑战之一。文中基于DeepZ方法,提出一种通过约束提升深度神经网络的局部鲁棒性验证精度的方法。在传播过程中加入约束来缩小抽象域,通过线性规划求解一个更小的神经网络输出范围,以此推断出神经网络输出节点的新的边界。应用新的边界,可以得出更精确的验证结果。基于此方法,实现了DeepZero工具,并在MNIST数据集上进行了充分的实验。实验结果表明,所提方法能有效提升DeepZ方法的验证成功率。在实验中,验证成功率平均可提升49%,说明了所提方法的有效性。

关键词:神经网络;软件验证;人工智能;机器学习;软件安全
中图法分类号　TP311

Constraint-based Verification Method for Neural Networks

GAO Yuzhao,XING Yunhan and LIU Jiaxiang
College of Computer Science and Software Engineering,Shenzhen University,Shenzhen,Guangdong 518060,China

Abstract　The verification of neural networks has always been one of the major challenges in the field of artificial intelligence. Based on the DeepZ method,this paper proposes a method to improve the accuracy of verifying local robustness of deep neural networks by constraints. During the propagation process,constraints are added to reduce the abstract domain,then a tighter neural network output range is solved by linear programming,hence deducing new bounds of neural network output node. With the new bounds,more accurate verification results can be obtained. Based on this method,the DeepZero tool is implemented in this paper,and comprehensive evaluation is carried out on the MNIST dataset. Experimental results show that our method effectively improves the verification success rate of DeepZ. Specifically,the verification success rate of DeepZ increases by 49% in average,indicating the effectiveness of the proposed method.

Keywords　Neural network,Software verification,Artificial intelligence,Machine learning,Software security

1　引言

随着大数据和人工智能时代的到来,深度神经网络技术得以迅速发展,并逐渐成为人工智能领域的关键技术。深度神经网络在许多应用中均取得了优异的性能。它们通常被应用在对基于逻辑的传统软件特别具有挑战性的任务,如自然语言处理[1]、图像分类[2]和游戏等。在神经网络得以广泛应用的同时,一系列安全问题也随之产生[4]。以汽车自动驾驶为例,2020年特斯拉电动车的自动驾驶曾出现多次事故,这表明在自动驾驶汽车正式投入市场之前,自动驾驶的安全性亟需得到保证[5]。根据兰德公司的研究报告,要足够表明自动驾驶汽车在安全方面的可靠性,需要进行数亿甚至数千亿英里的公共道路测试[6]。然而,随着驾驶自动化级别越来越高,规模越来越大,实车测试不能满足其对安全性的需求。再者,传统测试方法并不能完全说明汽车自动驾驶的安全性[7]。因此,若要推动神经网络技术的快速、安全应用,不仅要依靠传统测试方法,还需要应用形式验证方法提高神经网络的可靠性[8-9]。

与传统软件系统不同,深度学习定义了一种新的以数据驱动的编程范式,开发人员仅规定深度学习系统的网络结构,其内部逻辑则由训练过程决定,内部逻辑与传统软件逻辑不同。因此,针对传统软件的形式验证方法和度量指标无法被直接应用。相对于传统软件系统的验证,神经网络系统的验证有着以下突出的难点挑战:

挑战1　神经网络验证问题的定义。与传统软件系统的验证不同,目前对神经网络安全性没有广泛认可的一般性定义,这使得难以设计出系统性的工具以在实践中排除安全性威胁。

挑战2　状态空间爆炸。分析验证神经网络输入输出满足的性质一般需要对神经网络的内部节点进行分析,由于内部节点激活函数的存在,节点具有不同的激活状态,对每种激活状态进行分析会导致状态空间爆炸。

挑战3　精度与规模的权衡,在一定时间范围内,很难同时在验证精度和规模上都取得最好的效果。

基金项目:广东省自然科学基金(2022A1515011458,2022A1515010880);国家自然科学基金(61836005);深圳市科创委基础研究项目(JCYJ20210324094202008)
This work was supported by the Natural Science Foundation of Guangdong Province,China(2022A1515011458,2022A1515010880),National Natural Science Foundation of China(61836005) and Shenzhen Science and Technology Innovation Commission(JCYJ20210324094202008).
通信作者:刘嘉祥(jiaxiang0924@gmail.com)

近年来，越来越多的研究致力于解决深度神经网络的形式验证问题。第一个关于神经网络验证的工作是 Pulina 等在 2010 年对包含 sigmoid 激活函数的神经网络的局部鲁棒性验证[10]。其后又有许多具有创新性贡献的工作，例如基于可达性分析的 ExactReach[11] 和 MaxSens[12]，基于抽象解释的 DeepZ[13] 和 DeepPoly[14]，基于对偶优化的 Duality[15] 和 Certify[16] 等[17-19]。其中，基于抽象解释的方法更易于在大规模和结构复杂的神经网络中应用。最近，分支定界(Branch and Bound)方法在神经网络的验证上也有一定的应用，如以 CROWN[20] 方法为代表，衍生出如 α-CROWN[21]，α，β-CROWN[22]，FSB[23]，BaBSR[24] 等方法。此外，国内一些学者也在神经网络的形式验证方面做出了贡献，如使用抽象解释方法的 DeepSymbol[25]，基于符号传播(Symbolic Propagation)来提高基于抽象解释的神经网络验证的精确性的方法[26] 等。

由于验证神经网络是一个非凸优化问题，难度为 np 难(np-hard)，故前人使用一些方法来处理，如线性规划，DeepZ 便是线性规划中的一种。DeepZ 设计了一个多面体结构，用于对神经网络的激活函数进行上近似(Over-approximation)，从而验证神经网络的局部鲁棒性。本工作基于 DeepZ 的验证方法，对原有的线性近似加入新的约束，可以减小上近似的误差，进而可以验证更多的鲁棒性特征。

本文的主要贡献如下：

1)提出了一种新的基于抽象解释的约束改进方法，以验证神经网络的局部鲁棒性。

2)基于所提方法实现原型工具 DeepZero，并且在 MNIST 数据集进行实验评估。实验结果表明，该方法成功验证了 DeepZ 方法不能验证的样例且 DeepZero 的成功验证样例数量要多于 α，β-CROWN。

2 预备知识和问题描述

2.1 神经网络的局部鲁棒性和形式验证

本文研究深度神经网络(Deep Neural Networks，DNN)的局部鲁棒性，它体现了 DNN 输出结果的稳定程度。局部鲁棒性要求 DNN 的决策不受输入的微小扰动影响。也就是说，对于给定输入 x_0 和 x_0 的一个邻域，如果在该邻域内的所有输入，经过 DNN 的输出结果都与 x_0 对应的输出结果相同，则认为该 DNN 对于输入 x_0 和该邻域具有局部鲁棒性。假设有一个执行图片分类任务的神经网络，如果在输入图片的某些像素点添加人类所不能觉察的微小改变，即等价于在上述邻域内取值，直觉上这些扰动不会对人类的判断产生影响，但是这些微小改变可能会造成该神经网络输出的改变，这种情况下称这个网络不满足局部鲁棒性。

局部鲁棒性的形式描述如下，将神经网络看作将输入映射到输出的函数 $f:R^m\to R^n$，其中 R 是实数集，m，n 分别是输入和输出的维度，由具体的数据集确定。对于给定的输入 $x_0\in R^m$ 和扰动 η，若有

$$f(x)=f(x_0)，对任意\ x\in\{x\mid \|x-x_0\|_\infty\leqslant\eta\} \tag{1}$$

成立，则称神经网络 f 对输入 x_0 和扰动 η 具有局部鲁棒性。

给定神经网络 f 的输入 x_0 和扰动 η，局部鲁棒性验证问题需要判断式(1)是否成立。显然，测试与仿真技术可以证明局部鲁棒性不成立，但却无法证明其成立。形式验证技术[7,9] 可以从数学上证明局部鲁棒性成立，从而提供严格的保证。

2.2 DeepZ 简介

DeepZ 是一个验证含 ReLU 激活函数($\max(0,x)$)神经网络局部鲁棒性的方法。该方法的重要特点是，从神经网络的输入层开始，DeepZ 会逐层依次把所有节点输出值的表达式表示成以下形式：

$$\hat{y}=\alpha_0+\sum_{i>0}\alpha_i\cdot\varepsilon_i,\alpha_0,\alpha_i\in R,\varepsilon_i\in[-1,1] \tag{2}$$

其中，α_0 为该节点取值范围的中心，α_i 代表围绕中心的部分偏离，ε_i 被称作噪声符号。使用这种表达式，可以根据噪声符号的取值区间快速地计算出神经网络节点的上下界限。

对给定输入层(第 0 层)第 j 个节点的取值范围$[x_j-\eta,x_j+\eta]$，可以表示成以下形式：

$$\hat{y}_{0,j}=x_j+\eta\cdot\varepsilon_j,x_j,\eta\in R,\varepsilon_j\in[-1,1] \tag{3}$$

式(3)形式和式(2)一致。

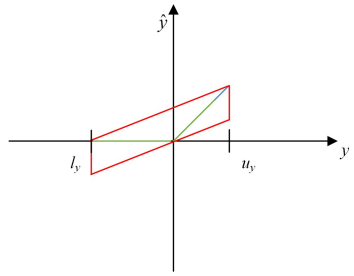
在逐层传播中，可以将节点的计算分为激活函数前的线性计算和 ReLU 激活函数的计算两个阶段。对于前者，依据前一层的输出，根据相邻节点之间的权重和节点的偏置，通过仿射变换计算中间层节点 ReLU 前的表达式，如通过第 i 层来计算第 $i+1$ 层第 j 个节点 ReLU 前的表达式如式(4)所示：

$$y_{i+1,j}=\boldsymbol{w}_{i,i+1,j}\cdot\hat{\boldsymbol{y}}_i+b_{i+1,j} \tag{4}$$

其中，$\boldsymbol{w}_{i,i+1,j}$ 表示第 i 层连接第 $i+1$ 层第 j 个节点的权重向量，$\hat{\boldsymbol{y}}_i$ 表示第 i 层的输出向量，$b_{i+1,j}$ 表示第 $i+1$ 层第 j 个节点的偏置。在式(4)中代入 $\hat{y}$ 的表达式，可以得到与式(2)一致的表达形式，求得取值范围为 $[l_y,u_y]$。对于 ReLU 激活函数，其函数图像如图 1 所示，根据 ReLU 的输入分为 3 种情况：1)激活，即确定该节点 ReLU 前的表达式大于等于 0，即 $l_y\geqslant 0$；2)不激活，即确定该节点 ReLU 前的表达式小于 0，有 $u_y\leqslant 0$；3)不确定，即 $l_y<0<u_y$。前两种激活状态是确定的，可以直接进行计算。第三种不确定的激活状态需要对 ReLU 函数上近似。3 种激活状态的 ReLU 变换后的结果如式(5)所示：

$$\hat{y}=\begin{cases}y, & l_y\geqslant 0\\ 0, & u_y\leqslant 0\\ \lambda\cdot y+\mu+\mu\cdot\varepsilon_{\text{new}}, & \text{其他}\end{cases} \tag{5}$$

其中，$\varepsilon_{\text{new}}\in[-1,1]$ 为新引入的噪声符号，$\hat{y}$ 表示 y 经过 ReLU 函数后的上近似结果。该函数表达式中，前两种为确定的激活状态，第三种为上近似结果，需要使用 y 的下界 l_y 和上界 u_y 计算 $\lambda=\dfrac{u_y}{u_y-l_y}$ 和 $\mu=-\dfrac{u_y\cdot l_y}{2\cdot(u_y-l_y)}$。ReLU 函数的不确定状态被上近似为一个平行四边形，如图 1 所示。这里可以观察到式(5)的形式仍与式(2)一致。



注：绿色折线为 ReLU 函数图像，红色多边形为 DeepZ 对 ReLU 函数的上近似。

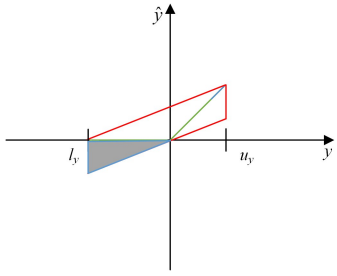
图 1 ReLU 函数图像与 DeepZ 对 ReLU 函数的上近似

Fig. 1 ReLU function and the over-approximation of ReLU function by DeepZ

依次逐层对神经网络进行向前传播直到输出层，若满足式(1)，即在给定输入的扰动内的所有输入经过神经网络得到的输出均和给定输入的输出一致，或得到的输出区间不超过特定限制，则该神经网络在给定输入和扰动下满足局部鲁棒性，否则该方法不能给出确定性结果。

3 引入约束的 ReLU 函数上近似

本文方法的核心思想是在 DeepZ 的基础上添加约束以减少上近似过程中产生的误差。ReLU 函数的定义决定了其输出总是大于等于 0，而在 DeepZ 方法中，式(5)其他情况中对 ReLU 函数的近似$\hat{y}=\lambda\cdot y+\mu+\mu\cdot\varepsilon_{new}$却违反了 ReLU 函数的这一性质，引入了不必要的误差，即图 2 中灰色部分。本文通过加入简单的约束，对 ReLU 函数的新的上近似域为图 2 原有平行四边形减去灰色三角形部分，可以减小输出范围，进而提高验证成功率。



注：灰色为相对 DeepZ 减少的误差。

图 2 加入新的约束后对 ReLU 函数的上近似图示

Fig. 2 Upper approximation of ReLU function after adding new constraints

对于输入层，在神经网络传播过程中，将所有输入数据同式(3)一样表示。定义一个约束集合 C，将噪声的取值范围$\varepsilon_i\in[-1,1](i=1,2,\cdots,n$，其中 n 为噪声个数)加入 C。

对于中间层传播，首先根据 DeepZ 方法计算得到形如式(2)的 y 的表达式，将该表达式作为线性规划的目标函数 obj，并将当前的集合 C 作为约束，求解线性规划问题(obj，C)，从而求得 y 的上下界u_y'和l_y'。注意，这里使用的上下界符号和 2.2 节中 DeepZ 求解的上下界有所不同，以示区分。此处得到的新的上下界比 DeepZ 求得的上下界更紧。接着，用新的上下界对 ReLU 激活函数进行变换。3 种激活状态的 ReLU 变换后的结果如式(6)所示：

$$\hat{y}=\begin{cases}y, & l_y'\geqslant 0\\ 0, & u_y'\leqslant 0\\ \lambda'\cdot y+\mu'+\mu'\cdot\varepsilon_{new}, & \text{其他}\end{cases}\tag{6}$$

式(6)即为引入约束的 ReLU 激活函数的上近似，其中$\lambda'=\frac{u_y'}{u_y'-l_y'}$，$\mu'=-\frac{u_y'\cdot l_y'}{2\cdot(u_y'-l_y')}$，$\varepsilon_{new}\in[-1,1]$为新的噪声符号。若为式(6)中的其他情况，则在约束集合 C 中加入新的约束$\lambda'\cdot y+\mu'+\mu'\cdot\varepsilon_{new}\geqslant 0$。依次遍历所有隐藏层节点，每经过一个节点，迭代计算得到新的约束集合 C，在 C 的约束下，得到比 DeepZ 更小的上近似域。

对于输出层的每个节点，经过仿射变换计算得到神经网络输出结果。若满足式(1)，即在给定输入的扰动内的所有输入与给定输入经过神经网络后的输出一致，则该神经网络在给定输入和扰动下满足局部鲁棒性，否则该方法不能给出确定性结果。

上述算法 DeepZero 的伪代码表述如算法 1 所示。

算法 1 DeepZero

输入：DNNf，输入 x，扰动 η

输出：鲁棒/不确定

1. function：验证(DNNf，输入 x，扰动 η)
2. 构造$\hat{y}_{0,i}=x_i+\eta\cdot\varepsilon_i$
3. C. append($\varepsilon_i\in[-1,1]$)
4. 遍历隐藏层节点：
5. 计算 $y_{i+1,j}=\alpha_0+\sum\alpha_i\cdot\varepsilon_i$
6. Obj ← $y_{i+1,j}=\alpha_0+\sum\alpha_i\cdot\varepsilon_i$
7. [l，u] ← LP 求解器(Obj，C) / * 约束集合 C 为对变量进行约束的不等式集合 * /
8. if l≥0：
9. $\hat{y}_{i+1,j}=y_{i+1,j}$
10. else if u≤0：
11. $\hat{y}_{i+1,j}=0$
12. else：
13. $\hat{y}_{i+1,j}=\lambda\cdot y_{i+1,j}+\mu+\mu\cdot\varepsilon_{new}$
14. C. append ($\hat{y}_{i+1,j}\geqslant 0,-1\leqslant\varepsilon_{new}\leqslant 1$)
15. 遍历输出层节点：
16. 计算 $\hat{y}=\alpha_0+\sum\alpha_i\cdot\varepsilon_i$
17. if 鲁棒性(f，x，$\hat{y}$)==TRUE：
18. return 鲁棒
19. else：
20. return 不确定

DeepZero 算法中，需要输入要验证的 DNN f，要验证的输入 x 和扰动 η。算法第 2 行将根据式(3)计算输入的表达式，第 3 行将ε_i的范围加入约束集合 C，第 4－14 行依次遍历所有中间层节点，得到每个节点的表达式，使用 LP 求解器求解每个节点的上下界，并根据上下界的取值将节点分为 3 种情况。第 15－16 行计算输出层的结果。最后 17－20 行根据式(1)判断局部鲁棒性结果并返回。

4 实验设计及结果分析

4.1 实验环境设置

本文将提出的 DeepZero 方法在 Python 3.8 进行实现，以 Gurobi 作为线性规划求解器。所有的实验在 Ubuntu 22.04 系统的计算机上运行，CPU 是 11th Gen Intel(R) Core (TM) i9-11900H @ 2.50 GHz，内存为 32 GB。

实验数据集和网络。MNIST 数据集是目前国际上评价神经网络鲁棒性验证技术的主流数据集之一，本文实验中使用了该数据集。为了保证实验对比的公平性，在网络上采用了文献[27]中使用的 7 个全连接神经网络作为验证目标，大小分别为 6×20，3×50，3×100，6×100，6×200，9×200 和 6×500。这里 $m\times n$ 中的 m 指隐藏层数，n 指每个隐藏层中的节点数。

为了验证改进的有效性，实验中选取的对比方法为 DeepZ。此外，为了更为全面地知晓本文方法的优缺点，本文还选取国际上知名工具 α，β-CROWN 这种前沿方法进行对比。

实验在 MNIST 数据集及网络上对比了 DeepZ，α，β-CROWN 和 DeepZero 对局部鲁棒性的验证效果，其中 DeepZ 和 DeepZero 采用本文在 DeepZero 工具中的实现，α，β-

CROWN 为文献[25]工具中的实现。由于本文对比的是非分支定界方法，在对 α,β-CROWN 的实验中，同样关闭了该方法的分支定界功能。实验验证数据为 MNIST 数据集中的前 100 张图片，实验中记录验证的成功样例和时间开销。

实验中的扰动大小分别根据不同网络的 DeepZ 方法验证成功率选取。本文认为，验证成功率代表所选取的验证问题对验证工具的难度高低。由于 DeepZero 是基于 DeepZ 的改进，因此本文希望评价 DeepZero 在不同难度下产生的改进效果，故为使实验结果更具有代表性，选取策略为在 DeepZ 方法上的验证成功率分别为高(大于 60%)、中(20%～60%)、低(小于 20%)各选一个扰动大小。

4.2 实验结果

实验中不同方法的成功数量结果如表 1 所列。

表 1 不同工具验证成功数量

Table 1 Numbers of successful cases for different tools

网络	扰动	验证成功数量		
		DeepZero	DeepZ	α,β-CROWN
6×20	0.010	**69**	67	68
	0.020	31	24	**68**
	0.030	9	6	**68**
3×50	0.010	86	86	**98**
	0.020	48	42	**79**
	0.030	9	5	**37**
3×100	0.015	68	65	**94**
	0.020	35	31	**73**
	0.025	13	8	**43**
6×100	0.010	**72**	61	38
	0.013	**47**	33	16
	0.015	**28**	16	6
6×200	0.006	**91**	87	18
	0.008	**71**	55	7
	0.010	**39**	26	3
9×200	0.006	**79**	70	62
	0.008	**54**	41	30
	0.010	**35**	11	3
6×500	0.010	**85**	70	1
	0.013	**51**	37	0
	0.020	**13**	4	0

表 1 列出了验证工具对 100 个输入的局部鲁棒性验证成功数量。

对 DeepZ，可以看到，对于同一网络，在增大扰动时，成功数量虽都有减少，但 DeepZero 相比 DeepZ 的成功数量均有增加。对于同一网络，在扰动越大时，不确定 ReLU 节点会增加[27]，而 DeepZero 相比 DeepZ 的提升为对不确定 ReLU 节点的约束，故在扰动越大时提升越明显，实验和理论一致。对于实验中所有网络下的所有扰动，DeepZero 相比 DeepZ 的验证成功数量均有增加，说明 DeepZero 在 MNIST 网络上的有效性。实验中，DeepZero 的验证成功数量比 DeepZ 的验证成功数量平均增加 49%。

对 α,β-CROWN，在实验的前 3 个网络上，该方法比其他方法验证成功数量要多，而对后面 4 个网络该方法的验证成功数量要少于 DeepZ 和 DeepZero 方法，即在实验中该方法的表现效果相比 DeepZero 方法与网络规模成反比，理论上，本文方法对当前层的处理包含之前层的优化，而 α,β-CROWN 的方法是在最后层计算结果时才进行优化，故随着网络层数加深，规模增大，DeepZero 比 α,β-CROWN 验证成功数量要多，出现表 1 所列的实验结果。

本文统计了在不同网络和扰动下，对于 100 个输入，DeepZ 和 α,β-CROWN 方法能成功验证而 DeepZero 方法不能成功验证的样例的数量，记作 N，以及 DeepZero 方法能成功验证而 DeepZ 和 α,β-CROWN 方法不能成功验证的样例的数量，记作 Y，如表 2 所列。

表 2 方法之间无法验证的验证问题数量对比

Table 2 Comparison of the numbers of cases that can be verified by one tool but not the other

网络	扰动	DeepZ		α,β-CROWN	
		Y	N	Y	N
6×20	0.010	2	0	30	29
	0.020	7	0	17	54
	0.030	3	0	7	66
3×50	0.010	0	0	6	18
	0.020	6	0	5	36
	0.030	4	0	2	30
3×100	0.015	3	0	0	26
	0.020	4	0	0	38
	0.025	5	0	5	35
6×100	0.010	11	0	40	6
	0.013	14	0	35	4
	0.015	12	0	22	0
6×200	0.006	4	0	76	3
	0.008	16	0	66	2
	0.010	13	0	36	0
9×200	0.006	8	0	27	11
	0.008	13	0	29	5
	0.010	24	0	32	0
6×500	0.010	15	0	84	0
	0.013	14	0	51	0
	0.020	9	0	13	0

因为 DeepZero 是在 DeepZ 方法的基础上，通过添加更多约束来减小上近似产生的误差，从而提高验证精度，因此在理论上，DeepZ 可以成功验证的局部鲁棒性问题，DeepZero 也必定可以成功验证；反之，DeepZero 可以成功验证的局部鲁棒性问题，DeepZ 却不一定可以成功验证。从表 2 也可以观察到，DeepZero 可以验证一些 DeepZ 不能成功验证的问题。

与 α,β-CROWN 方法相比，在实验中的难度较低的问题上，两者均可以成功验证另外一种方法不能成功验证的一些样例。而对于一些难度较高的问题，DeepZero 可以验证 α,β-CROWN 不能成功的一些问题；在验证成功的数量上，DeepZero 要比 α,β-CROWN 多。

总的来说，在实验的各种规模网络上，DeepZero 可以成功验证 DeepZ 无法验证的问题。在实验中的难度较低的问题上，α,β-CROWN 的验证成功数量最高，DeepZero 其次。在实验中难度较高的问题上，DeepZero 可以成功验证 α,β-CROWN 不能成功验证的一些问题，且 DeepZero 的验证成功数量大于 α,β-CROWN 方法。

结束语 本文提出了一种基于 DeepZ 和线性约束的方法 DeepZero，使用线性规划对 DeepZ 抽象域进行约束优化，用于解决深度神经网络的局部鲁棒性形式验证问题。实验结果表明，与原有 DeepZ 方法相比，在所有实验上 DeepZero 均能够验证更多的局部鲁棒性性质，验证成功率平均提升49%，实现更高的精度的验证。与 α,β-CROWN 方法相比，在实验中的小规模网络上 α,β-CROWN 方法的验证成功率较高，而在实验中的较大规模网络上，DeepZero 的验证成功率较高，相比该方法，DeepZero 更适用于大规模网络的鲁棒性验证。本文工作仍存在不足之处，DeepZero 方法的验证精度仍有提升空间。在未来，需要继续寻找可能存在的更优约束条件，

另外,可以结合其他求解方法共同对神经网络进行验证。

参 考 文 献

[1] HINTON G,DENG L,YU D,et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups[J]. IEEE Signal Processing Magazine, 2012,29(6):82-97.

[2] KRIZHEVSKY A,SUTSKEVER I,HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM,2017,60(6):84-90.

[3] LUBOSCH M,KUNATH M,WINKLER H. Industrial scheduling with Monte Carlo tree search and machine learning[C]// Procedia CIRP. 2018:1283-1287.

[4] HUANG X,KROENING D,RUAN W,et al. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability[J]. Computer Science Review,2020,37:100270.

[5] ZHANG M H,DU D H,ZHANG M Z,et al. Spatio-temporal trajectory data-driven autonomous driving scenario meta-modeling approach[J]. Ruan Jian Xue Bao/Journal of Software, 2021,32(4):973-987.

[6] KALRA N,PADDOCK S M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliab-ility? [J]. Transportation Research Part A: Policy and Practice,2016,94:182-193.

[7] WANG Z,YAN M,LIU S,et al. Survey on testing of deep neural networks[J]. Ruan Jian Xue Bao/Journal of Software,2020, 31(5):1255-1275.

[8] WANG J,ZHAN N J,FENG X Y,et al. Overview of formal methods[J]. Ruan Jian Xue Bao/Journal of Software, 2019, 30(1):33-61.

[9] LIU C,ARNON T,LAZARUS C,et al. Algorithms for verifying deep neural networks[J]. Foundations and Trends © in Optimization,2021,4(3/4):244-404.

[10] PULINA L,TACCHELLA A. An abstraction-refinement approach to verification of artificial neural networks[C]// International Conference on Computer Aided Verification. Berlin: Springer,2010:243-257.

[11] XIANG W,TRAN H D,JOHNSON T T. Reachable set computation and safety verification for neural networks with relu activations[J]. arXiv:1712.08163,2017.

[12] XIANG W,TRAN H D,JOHNSON T T. Output reachable set estimation and verification for multilayer neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018,29(11):5777-5783.

[13] SINGH G,GEHR T,MIRMAN M,et al. Fast and effective robustness certification[C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems(NIPS' 18). 2018:10825-10836.

[14] SINGH G,GEHR T,PÜSCHEL M,et al. An abstract domain for certifying neural networks [C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems(NIPS '18). 2018:10825-10836.

[15] DVIJOTHAM K,STANFORTH R,GOWAL S,et al. A Dual Approach to Scalable Verification of Deep Networks[C]// UAI. 2018:3.

[16] RAGHUNATHAN A,STEINHARDT J,LIANG P. Certified defenses against adversarial examples[J]. arXiv:1801.09344, 2018.

[17] ASHOK P,HASHEMI V,K ŘETÍNSK Ý J,et al. Deepabstract: Neural network abstraction for accelerating verification [C]// International Symposium on Automated Technology for Verification and Analysis. Cham:Springer,2020:92-107.

[18] GEHR T,MIRMAN M,DRACHSLER-COHEN D,et al. Ai2: Safety and robustness certification of neural networks with abstract interpretation[C]// 2018 IEEE Symposium on Security and Privacy(SP). IEEE,2018:3-18.

[19] TRAN H D,MANZANAS LOPEZ D,MUSAU P,et al. Star-based reachability analysis of deep neural networks[C]// International Symposium on Formal Methods. Cham:Springer,2019: 670-686.

[20] ZHANG H,WENG T W,CHEN P Y,et al. Efficient neural network robustness certification with general activation functions [J/OL]. Advances in Neural Information Processing Systems, 2018, 31. https://dl. acm. org/doi/abs/10. 5555/3327345. 3327402.

[21] XU K,ZHANG H,WANG S,et al. Fast and complete:Enabling complete neural network verification with rapid and massively parallel incomplete verifiers[J]. arXiv:2011.13824,2020.

[22] WANG S,ZHANG H,XU K,et al. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification[J]. Advances in Neural Information Processing Systems,2021,34:29909-29921.

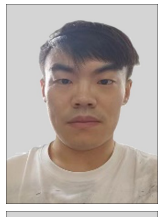
[23] DE PALMA A,BUNEL R,DESMAISON A,et al. Improved branch and bound for neural network verification via lagrangian decomposition[J]. arXiv:2104.06718,2021.

[24] BUNEL R,MUDIGONDA P,TURKASLAN I,et al. Branch and bound for piecewise linear neural network verification[J/OL]. Journal of Machine Learning Research,2020,21(2020). https://dl. acm. org/doi/10. 5555/3455716. 3455758.

[25] LI J,LIU J,YANG P,et al. Analyzing deep neural networks with symbolic propagation: Towards higher precision and faster verification[C]// International Static Analysis Symposium. Cham:Springer,2019:296-319.

[26] LIN W,YANG Z,CHEN X,et al. Robustness verification of classification deep neural networks via linear programming [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019:11418-11427.

[27] YANG P,LI R,LI J,et al. Improving neural network verification through spurious region guided refinement[C]// International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Cham:Springer,2021:389-408.



GAO Yuzhao, born in 1996, postgraduate. His main research interests include formal verification and neural network.

LIU Jiaxiang, born in 1987, Ph.D, assistant professor, is a member of China Computer Federation. His main research interests include formal verification and rewriting theory.