文章编号:1674-2974(2025)08-0122-08 DOI:10.16339/j.cnki.hdxbzkb.2025289

基于机器学习的功能覆盖率预测算法

刘光宇[1,2,3],王艺洋[3],林子明[3],李志强[1,2†],梁利平[4]
(1. 中国科学院 微电子研究所,北京 100029;
2. 中国科学院大学 集成电路学院,北京 101408;
3. 北京全路通信信号研究设计院集团有限公司,北京 100070;
4. 北京邮电大学 集成电路学院,北京 100876)

摘 要:芯片规模的扩大及功能的不断加强,使得芯片验证难度呈几何级数递增.对于多组合激励的功能覆盖情况,传统通用做法是依照其不同使用场景,以分片或切片形式进行统计.此类方法操作简单,但难以在随机测试下对各个配置的组合情况进行完整覆盖分析.针对该问题,提出了一种基于机器学习算法进行覆盖率快速收敛且通用性强的验证方法.该方法将各个配置激励按权重进行分解处理,对功能覆盖中的关键交叉仓进行观测,利用功能点分析不消耗仿真时间的特性,对数据集进行收集并训练,通过实际测试调整,实现了一种改进型的网络结构,可对各种激励组合情况进行覆盖率预测,并可挑选指定覆盖阈值的激励输入.仿真结果表明,与随机情况相比,该方法可显著降低仿真时间,并有效减少仿真资源占用;与其他网络结构相比,该网络收敛更为迅速,并可达到更高的预测精度.

关键词:神经网络;功能覆盖率;随机测试;训练损失
中图分类号:TN431.2 **文献标志码**:A

Functional Coverage Prediction Algorithm Based on Machine Learning

LIU Guangyu[1,2,3], WANG Yiyang[3], LIN Ziming[3], LI Zhiqiang[1,2†], LIANG Liping[4]
(1. Institute of Microelectronics of the Chinese Academy of Sciences, Beijing 100029, China;
2. School of Integrated Circuits, University of Chinese Academy of Sciences, Beijing 101408, China;
3. CRSC Research & Design Institute Group Co., Ltd., Beijing 100070, China;
4. School of Integrated Circuits, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: With the expansion of chip scale and the strengthening of function, the difficulty of verifying chips is increasing geometrically. At present, for the functional coverage of multiple combination incentive cases, the industry's common practice is to calculate it in the form of fragments or slices according to different use scenarios. This method is easy to operate, but it is difficult to perform a complete coverage analysis of the combination of various configurations under random testing. To solve this problem, a verification method based on a machine learning algorithm for fast convergence and strong universality of coverage is proposed. In this method, each configuration incentive is decomposed according to the weight, and the key cross bins in the function coverage are

* **收稿日期**:2024-08-06
基金项目:集成电路先进器件工艺仿真与智能EDA青年交叉团队项目(JCTD-2022-07), CAS Youth Interdisciplinary(JCTD-2022-07)
作者简介:刘光宇(1986—),男,河北沧州人,中国科学院大学博士研究生
† 通信联系人,E-mail: lizhiqiang@ime.ac.cn

observed. The data set is collected and trained by the feature that the function point analysis does not consume the simulation time. Through the actual test adjustment, an improved network structure is realized, which can predict the coverage rate of various incentive combinations, and also can pick an incentive input that specifies a coverage threshold. Simulation results show that compared with the random case, the proposed method can significantly reduce the simulation time and effectively reduce the simulation resource occupation. Compared with other network structures, the proposed network achieves faster convergence and higher prediction accuracy.

Key words: neural network; functional coverage; random test; training loss

随着芯片规模的不断加大,其验证难度和复杂度也在不断提高,在完整的开发周期中,验证所花费的时间已经占到50%以上,且仍有不断攀升趋势[1].在有限的时间周期和人力成本约束下,传统的验证模式逐渐不适应芯片的测试任务.当前芯片验证主要以受约束的随机测试(constrained random test, CRT)为主,以期可以发现更多的边界错误[2],而随机情况会产生大量的冗余激励[3],进一步延长验证时间.如何在更短时间内实现功能覆盖的快速收敛,已经成为验证工作的主要挑战[4].

覆盖率驱动测试[5-6],是当前普遍使用的验证方法,目标是通过观测覆盖率指导测试向量生成,其对各种待测试设计(design under test, DUT)适用性较强,但收敛速度较慢,相比之下,各种机器学习算法表现出了优秀性能.文献[7-9]采用遗传算法进行分析,利用其交叉及变异特性进行激励加速.贝叶斯网络[10]、马尔科夫链[11-14]、决策树[15]、支持向量机[16]、聚类[17]等智能化算法也表现出了优异性能.近年来,随着深度学习技术的成熟,验证模式也有了新的发展方向,文献[18]采用深度学习网络进行覆盖预测,可在更少的指令条件下达到更高的代码覆盖率.文献[19]采用深度神经网络(DNN)算法进行隐式过滤,从而实现覆盖率的加速收敛,文献[20]采用了一种自适应神经网络,以加速有限序列的生成.文献[21]提出了一种结合机器学习的硬件验证环境,可实现约束的实时更新.文献[22]对DNN、随机森林、支持向量机(SVM)、长短期记忆(LSTM)等多种网络结构进行了尝试,通过事务修剪及有向序列生成,有效提高了覆盖率的收敛速度.以上算法在不同程度上实现了对覆盖率的加速提升,但其多为针对特定待测设计开发而成,普适性较差.

目前对于专用集成电路的设计,通常设置多种配置模式,以使其满足不同场景下的功能需求.对于验证人员而言,理论上需对所有配置项进行遍历,以满足测试完备性.实际工作中,每增加一项配置,排除与其他项的互斥因素,其验证复杂度基本为倍数增长,这导致芯片验证的复杂度与配置项几乎呈指数关系、状态空间出现爆炸情况[23].在随机测试情况下,根据粗略的统计,庞大的配置集合,其所用的完整测试案例达到配置集数据的百倍以上.考虑其复杂度如此宏大,目前功能覆盖率的统计多采用切片处理,即对各个配置项单独统计,或分析特定场景部分配置项的组合情况.其统计办法较为粗糙,且不能充分覆盖不同配置的组合情况.

针对上述问题,为满足约束随机测试模式下的交叉覆盖,提出了一种改进型引擎网络(engine optimization net, EoNet),通过模型训练及参数调节,在随机测试模式下,可迅速挑选出超过阈值范围的功能覆盖生成向量,从而达到较高的覆盖要求.与传统模式及常用的神经网络相比,文中算法的预测精度更高,可在更短的测试向量或更少的资源占用下达到覆盖率的快速收敛.

1　设计及验证需求

对于采用较多配置的芯片验证,本方法具有普适性.本文以数字滤波芯片为例进行说明,其主体结构如图1所示.其中最左侧为AD接口,主要包含数据输入及相关控制信号;最右侧为CPU接口,主要进行芯片的参数配置及对滤波后结果的读取.图1中心一列为芯片的配置寄存器,其包含多种配置模式,如滤波模式可选择FIR滤波、IIR滤波、先FIR后IIR滤波;数据输入输出可采用串行或并行方式;输入输出采样率范围;AD采样位宽;AD串行采样通道数,每通道采样数据等.各个配置参数之间既有组合情

况,又有限定及互斥情况.功能点交叉情况复杂,覆盖集较为庞大.

该验证平台的功能覆盖率模型由覆盖组(cover group)实现,依据配置寄存器的特点及组合应用场景,划分为多个覆盖点(cover point).每个覆盖点依据复杂度创建相应数量的仓(bin),并在仿真中记录该覆盖点是否被覆盖及覆盖次数.针对单个配置寄存器参数,设定单点仓(single bins);针对寄存器功能点组合情况及应用场景,设置交叉仓(cross bins)和深度交叉仓.

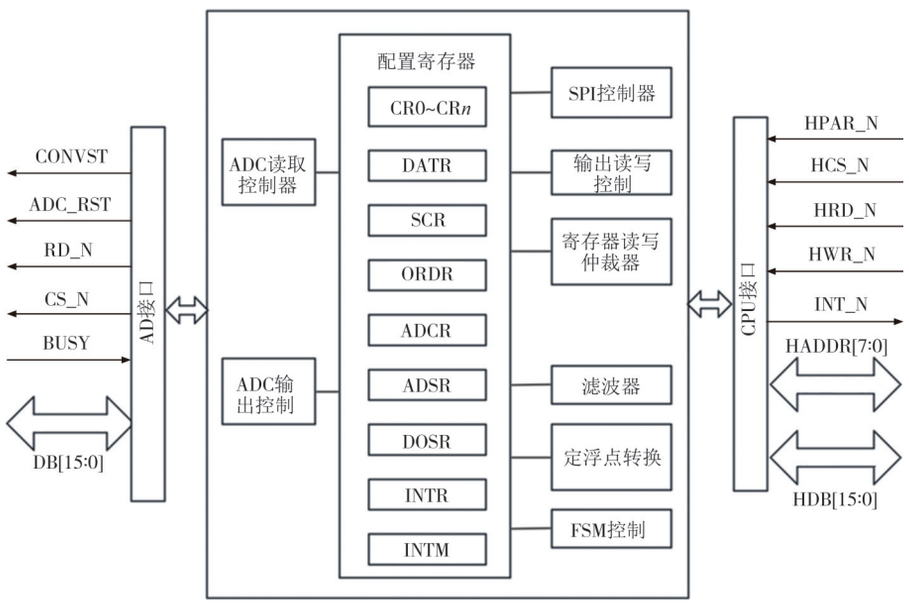


图1 数字滤波芯片主体结构

Fig.1 Digital filter chip main structure

测试的核心激励为配置寄存器的随机组合情况.除滤波器系数需根据场景使用特定的数据外,其余所有配置均可进行随机处理,并进行功能覆盖点观测.共设计单点覆盖点51个,交叉覆盖点43个,其中交叉覆盖点中深度交叉覆盖点数目为8,功能覆盖需满足上述覆盖点中所有仓均实现命中.排除参数互斥条件,交叉仓可认为是其中各个覆盖点仓数的乘积,而深度交叉仓可认为是其中交叉仓的乘积,因此深度交叉仓的覆盖是实现功能完备性的关键.在受约束随机测试条件下,要实现其完整覆盖,所需测试集极大,且包含诸多重复案例情况,因此考虑借助神经网络实现.

2 网络数据集生成

设置的8个深度交叉覆盖点共包含27 036个仓,占到所有仓的80%左右,且涵盖了所有典型测试场景.故而将其覆盖率情况作为学习网络的输出,与其有相关度的配置参数,如滤波模式、输入数据串并行情况等,作为学习网络的输入,所有配置参数的组合情况即可实现完整覆盖.

验证环节会占用大量仿真资源,如CPU使用、内存及硬盘占用等.为加速测试案例完成,实际中通常采用多进程并行仿真方式实现.考虑采用较少资源占用满足测试情况,将仿真进程并行度作为一个输入参数,设定范围为5~70.每个进程仿真事务数目直接影响仿真的时间,将其作为参数输入,设定范围为500~30 000.考虑到这两个参数数值波动较大,容易导致训练不收敛,采用min-max方法对其进行归一化处理,其余参数由于本身即为权重大小,将其转换为对应百分比即可.最终实现的神经网络输入参数见表1,其中前七个参数分别对应图1中的SCR、ADCR、ADSR、DOSR、INTR等配置寄存器,cpu_parallel对应CPU接口的HPAR_N端口.

图2所示为网络生成与使用流程图,该验证过程使用统一验证方法学[24](universal verification methodology, UVM)实现.在其随机生成组件uvm_sequence_item中,生成数据以权重体现,为简化操作,设定每个变量的权重分布总和为100,各参数可在0~100之间随机产生.考虑部分数值过低或过高会有明显不合理情况,将其数值进行进一步约束.生成数据送入测试平台中运行,并收集得到的输出

表 1　输入参数说明

Tab.1　Input parameters description

名称	参数数目	说明
filter_mode	3	FIR、IIR、FIR+IIR 情况权重
prob_im_inter	2	滤波输入级与中间级相等与否权重
prob_mo_inter	2	滤波中间级与输出级相等与否权重
sample_i_khz	3	采样率分段权重
ad_parallel	2	采样串/并行权重
ad_bitwidth	3	采样数据位宽分段权重
int_mode	2	滤波结果读取中断/查询模式权重
cpu_parallel	2	CPU 对滤波结果串/并行读取权重
multiprocess	1	仿真进程并行度
tr_times	1	每个进程仿真事务数目

功能覆盖率数据,与输入数据向量送入网络进行训练,得到可使用的网络模型.再随机生成大量输入数据,送入网络进行前向传播,然后将得到的功能覆盖率数据进行筛选,超过设定阈值的数据进行排序后,选取结果重新送入测试平台中运行,选定其中的覆盖率最高值,最终采用该组参数进行完整仿真测试.

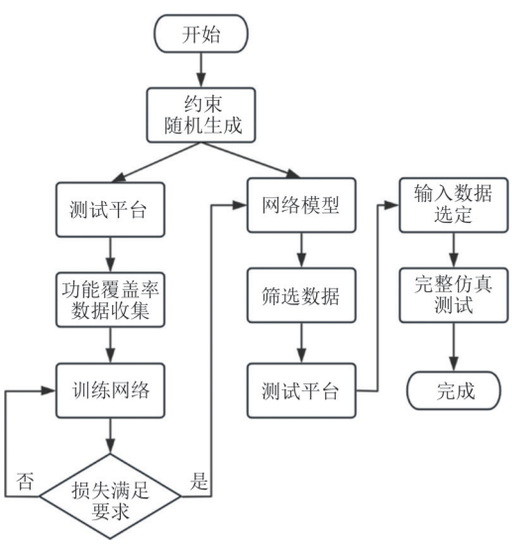


图 2　神经网络生成及使用流程

Fig.2　Neural network generation and use

神经网络的一个难点在于需要大量的训练集数据[25-26],对于代码覆盖率,需经过完整的事务仿真后得到覆盖率结果,这本身需要占用大量时间,在功能仿真中无明显优势.而对于功能覆盖点,uvm_sequence 组件在生成一笔随机事务后可通过 uvm_driver 直接传递到覆盖组件 uvm_subscriber,经分析即可产生功能覆盖率结果,此后可结束仿真进程,不必经过待测设计,而此过程不消耗仿真时间,如图 3 中实线箭头所示.

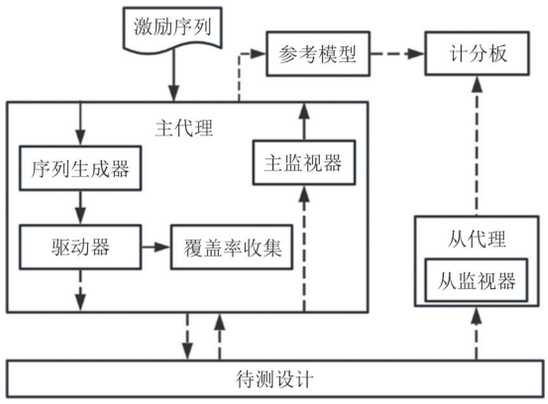


图 3　UVM 环境中训练集生成过程

Fig.3　Training set generation process in UVM

3　神经网络设计

神经网络的输入构造为 1×21 的一维向量,输出为 1×8 的一维向量,由于没有绝对正确性的判定条件,采用输出值的均方误差来判别学习的准确度.考虑网络的输入和输出呈相对线性关系,且与图像或语音相比,其输入数据量相对较小,尽量采用简单的神经网络[27].本文分别采用多层感知机(multilayer perceptron, MLP)、手写体识别卷积神经网络[28](LeNet)与改进型引擎网络 EoNet 实现并进行比较.

3.1　简单网络实现

与其他深度学习网络相比,MLP 计算量相对较小,但仍有较好的泛化能力及非线性表达,应用范围广泛.综合考虑模型复杂度及参数影响情况,经多次尝试,采用 MLP 的网络实现,设计的神经网络总体如图 4 所示.共包含四个隐层,为保持中间数据规模,设定第一隐层神经元为 800,其余各个隐层的神经元数量设定依次减少为 600、400、200.

由于输入数据向量较小,也可考虑采用经典手写识别网络 LeNet 实现,图 5 所示为参照实现的网络

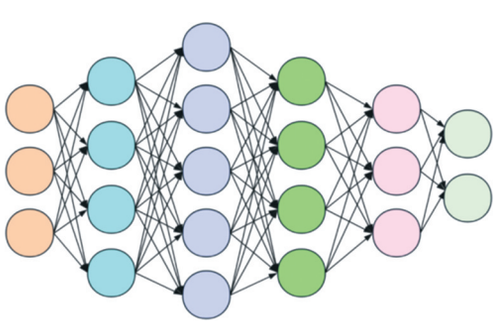


图 4　MLP 网络实现

Fig.4　MLP network implementation

结构.其输入数据设定为一维维度,经两次卷积和最大池化后进入全连接层.其中卷积核大小为1×4,池化的卷积核及步长为1×2,激活函数采用Sigmoid函数.

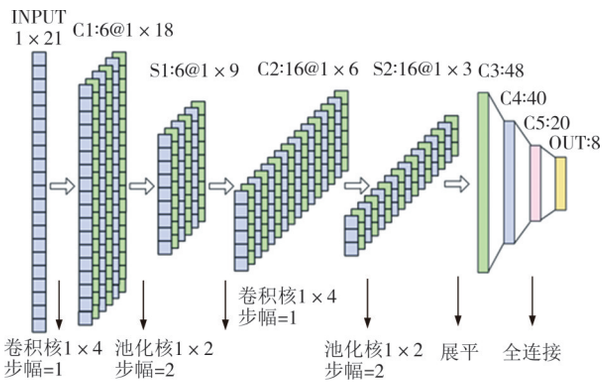


图5 LeNet网络实现

Fig.5 LeNet network implementation

3.2 EoNet网络实现

实验表明,MLP网络训练精度相对较低.LeNet虽为CNN结构,但由于数据仍为一维维度,且池化层的存在进一步减少了数据量大小,其效果并不理想,因此对其进行改进.网络设计主要基于以下几个方面的考虑:

1)考虑一维卷积核对数据量减少较为明显,为便于利用卷积的特性,将输入数据排列为二维方形结构,并在末尾补零,由1×21形状改为5×5大小.池化层的存在不利于特征向量的保存,因此网络中去掉了池化层.

2)实验表明,适当增加中间数据量可显著提高网络精度,同时网络首层需尽量保存边缘信息.综合考量,设计的卷积块如图6所示,上层1×1卷积块主要为增加通道维度,使通道数增加到16,之后3×3卷积在四周进行一层数据填充,如此保持其输出的宽和高与输入一致,并将通道数增加到32.下层进行1×1卷积仅将通道数增加到32,保持了初始数据信息.最终两个并行通路的输出在通道维度上进行连接,并经ReLU激活函数输出,最终得到64通道的5×5向量.

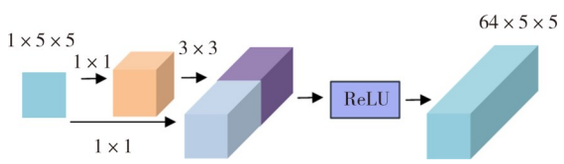


图6 首层卷积块

Fig.6 First layer convolution block

3)第二、三层设计主要参照残差网络(residual network,ResNet),其实现核心在于解决深层网络中的梯度消失和梯度爆炸问题[29],但其相加特性也有助于保留低级特性,且批量归一化的引入提高了模型的泛化能力.

综合考虑性能与复杂度,网络共设置两个残差块,每个块的结构如图7所示.共设置两级,每级均包含两组3×3卷积及批量归一化,输出与输入相加后送入下一级,最终结果经ReLU激活函数输出.其中第一残差块输入输出维持通道数不变,均为64,第二残差块将输出通道数扩至128.最终实现的网络整体结构如图8所示,残差块输出后展平进入三级全连接层,得到最后输出.

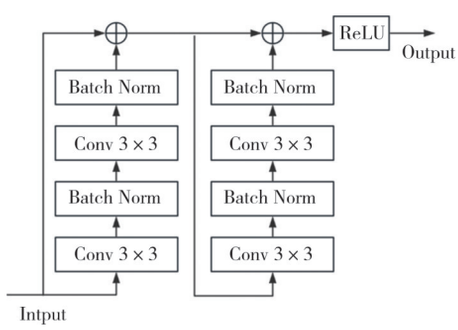


图7 残差块结构

Fig.7 Residual block structure

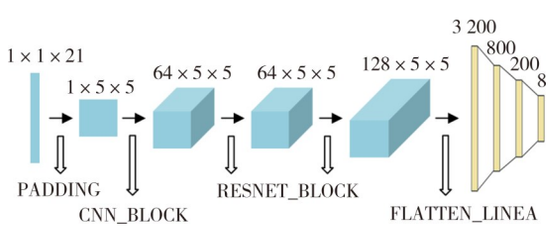


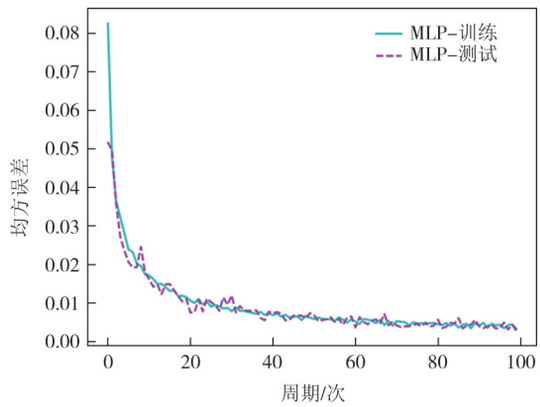
图8 EoNet网络实现

Fig.8 EoNet network implementation

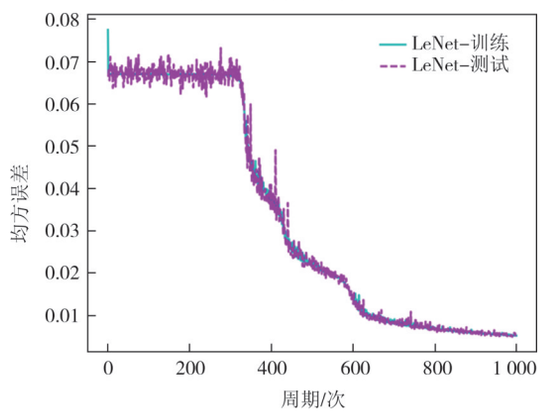
4 实验结果

4.1 神经网络训练结果对比

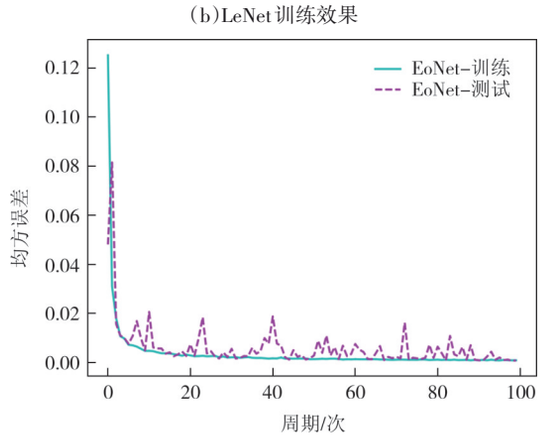
为便于直观比较,所有网络的损失函数都为均方差,训练与测试损失都为一个周期的总损失与向量数目的比值.以上三种网络的损失如图9所示.可以看出,三种神经网络的训练和测试损失的下降趋势均保持一致,证实网络得到有效训练.其中LeNet在前几百周期内损失保持不变,调节参数后效果仍不明显,考虑其陷入局部最小值.



(a)MLP 训练效果



(b)LeNet 训练效果



(c)EoNet 训练效果

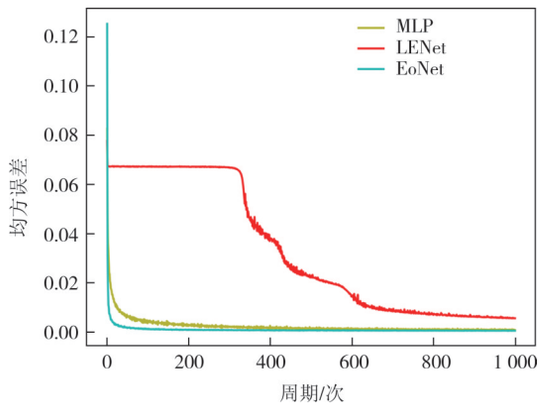
图 9　各个神经网络训练效果

Fig.9　Training effect of each neural network

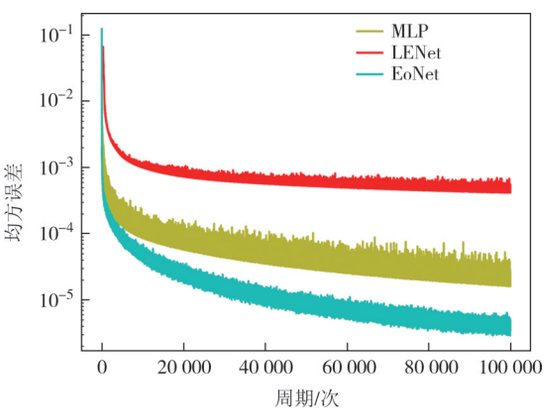
图 10 为三种网络的训练损失比较.从图 10(a)可以看出:相比于 MLP 及 LeNet,本文提出的 EoNet 收敛更为迅速.图 10(b)为 10 万次迭代后三种网络训练损失,考虑总训练周期较大,后期三种网络的损失均趋近于零,为便于观测,此处采用对数坐标形式.可以看出:MLP 最终稳定在 10^{-5} 量级,LeNet 稳定在 10^{-3} 量级,而 EoNet 可以下探到 10^{-6} 量级.

4.2　覆盖率实测结果

训练后的模型可用于对随机输入向量进行筛选,设定输出阈值为 98.5%,并对输入向量进行排



(a)三种网络在 1 000 次迭代损失对比



(b)三种网络在 10 万次迭代损失对比

图 10　三种网络的训练损失对比

Fig.10　Training loss of each neural network

序,可选择最低资源占用(并行进程数)、最快速度(进程仿真事务数)或最高覆盖情况.排序后的输入向量进入测试平台进行实测,最终各个测试情况如表 2 所示.

表 2　不同需求下关键仓覆盖情况

Tab.2　Key bins coverage under different needs

不同需求	进程数	仿真事务数	关键仓覆盖率/%	原始随机覆盖率/%	加速比
最少资源	23	28 320	99.07	73.10	1.36
最快速度	61	10 269	98.33	72.72	1.35
最高覆盖	67	28 681	99.43	85.53	1.16

对于最少资源情况,仅需 23 个并行进程即可实现 99% 以上覆盖,而原始随机覆盖率仅为 73% 左右.对于最快速度情况,虽然进程数有所增加,但仿真事务数目仅为 10 000 个左右,这会大幅减少仿真时间,且与原始随机覆盖率相比效果明显.而对于最高覆盖情况,其关键仓覆盖率与其他相比略有提高,但与原始随机覆盖率比较效果不甚明显,且资源占用相对较大,实际应用中需综合考量.

5 结束语

针对随机测试不便对功能点组合情况完整覆盖测试的问题,利用功能覆盖分析不消耗仿真时间的特性,设计了一种轻量化的卷积神经网络 EoNet.通过对输入进行权重处理,数据向量进行重新分布,并结合参数优化,实现了对任意激励覆盖情况的预测.与传统测试方法相比,该方案可在较少测试向量或较低仿真资源使用情况下达到指定覆盖要求,同其他标准神经网络相比,其表现出更快的收敛性及更高的预测精度.综上所述,文中算法对基于机器学习技术的功能覆盖方法进行了探索,为复杂模式下的芯片测试提供了一种通用性强的解决思路.

参考文献

[1] FOSTER H D. Trends in functional verification: a 2014 industry study [C]//Proceedings of the 52nd Annual Design Automation Conference. San Francisco California. ACM, 2015: 1-6.

[2] VANARAJ A T, MARSHAL R, LAKSHMINARAYANAN G, et al. Optimal test scenarios based regression suite for functional verification closure of advanced digital designs [C]//2024 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES). Tumakuru, India. IEEE, 2024: 1-6.

[3] YEHIA A. Faster coverage closure: Runtime guidance of Constrained Random stimuli by collected [C]//2013 Saudi International Electronics, Communications and Photonics Conference. Riyadh, Saudi Arabia. IEEE, 2013: 1-6.

[4] RAHUL L, NAVEEN S, SEKHAR D. Coverage Acceleration and Testcase Pruning using Smart Stimuli Generator in SOC Verification [C]// Proceedings of the Design and Verification Conference (DVCON). India: 2023: 1-10.

[5] 沈海华,卫文丽,陈云霁. 覆盖率驱动的随机测试生成技术综述[J]. 计算机辅助设计与图形学学报,2009,21(4):419-431.
SHEN H H, WEI W L, CHEN Y J. A survey on coverage directed generation technology [J]. Journal of Computer-Aided Design & Computer Graphics, 2009, 21(4): 419-431. (in Chinese)

[6] IMKOVÁ M, KOTÁSEK Z. Automation and optimization of coverage-driven verification [C]//2015 Euromicro Conference on Digital System Design. Madeira, Portugal. IEEE, 2015: 87-94.

[7] 沈海华,王朋宇,卫文丽,等. 基于遗传算法的全芯片级覆盖率驱动随机验证技术[J]. 计算机研究与发展,2009,46(10): 1612-1625.
SHEN H H, WANG P Y, WEI W L, et al. A coverage directed test generation platform for microprocessors using genetic approach [J]. Journal of Computer Research and Development, 2009, 46(10): 1612-1625. (in Chinese)

[8] 罗汉青,梁利平,叶甜春. 基于遗传算法的随机测试生成技术探究[J]. 电子测试 ,2013(13):75-77.
LUO H Q, LIANG L P, YE T C. Genetic algorithm based random test generation: a case study [J]. Electronic Test, 2013(13): 75-77. (in Chinese)

[9] DANCIU G M, DINU A. Coverage fulfillment automation in hardware functional verification using genetic algorithms [J]. Applied Sciences, 2022, 12(3): 1559.

[10] FINE S, ZIV A. Coverage directed test generation for functional verification using Bayesian networks [C]//Proceedings 2003. Design Automation Conference. Anaheim, CA, USA. IEEE, 2003: 286-291.

[11] BÖHME M, PHAM V T, ROYCHOUDHURY A. Coverage-based greybox fuzzing as Markov chain [J]. IEEE Transactions on Software Engineering, 2019, 45(5): 489-506.

[12] WAGNER I, BERTACCO V, AUSTIN T. Microprocessor verification *via* feedback-adjusted Markov models [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2007, 26(6): 1126-1138.

[13] WANG J, LI H W, LV T, et al. Abstraction-guided simulation using Markov analysis for functional verification [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2016, 35(2): 285-297.

[14] ZHANG M M, GENG S Q, WANG W S, et al. Probabilistic analysis for sequential circuits verification using Markov chains [J]. IEEE Transactions on Circuits and Systems Ⅱ: Express Briefs, 2021, 68(1): 481-485.

[15] YE F M, ZHANG Z B, CHAKRABARTY K, et al. Adaptive board-level functional fault diagnosis using decision trees [C]// 2012 IEEE 21st Asian Test Symposium. Niigata, Japan. IEEE, 2012: 202-207.

[16] ROMERO E, ACOSTA R, STRUM M, et al. Support vector machine coverage driven verification for communication cores [C]// 2009 17th IFIP International Conference on Very Large Scale Integration (VLSI-SoC). Florianopolis, Brazil. IEEE, 2009: 147-152.

[17] EL MANDOUH E, SALEM A, AMER M, et al. Cross-product functional coverage analysis using machine learning clustering techniques [C]//2018 13th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS). Taormina, Italy. IEEE, 2018: 1-2.

[18] 王培鑫. 基于机器学习的处理器验证技术研究[D]. 北京:中国科学院大学, 2020: 1-64.

WANG P X. Research on processor verification technology based on machine learning[D]. Beijing: University of Chinese Academy of Sciences, 2020: 1-64. (in Chinese)

[19] GAL R, HABER E, ZIV A. Using DNNs and smart sampling for coverage closure acceleration[C]//Proceedings of the 2020 ACM/IEEE Workshop on Machine Learning for CAD. Virtual Event Iceland. ACM, 2020: 15-20.

[20] GAD M, ABOELMAGED M, MASHALY M, et al. Efficient sequence generation for hardware verification using machine learning [C]//2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS). Dubai, United Arab Emirates. IEEE, 2021: 1-5.

[21] ABOELMAGED M, MASHALY M, ABD EL GHANY M A. Online constraints update using machine learning for accelerating hardware verification[C]//2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES). Giza, Egypt. IEEE, 2021: 113-116.

[22] GOGRI S, TYAGI A, QUINN M, et al. Transaction level stimulus optimization in functional verification using machine learning predictors [C]//2022 23rd International Symposium on Quality Electronic Design (ISQED). Santa Clara, CA, USA. IEEE, 2022: 71-76.

[23] WILE B, GOSS J C, ROESNER W. 全面的功能验证:完整的工业流程[M]. 北京: 机械工业出版社, 2010.
WILE B, GOSS J C, ROESNER W. Comprehensive Functional Verification: The Complete Industry Cycle[M]. Beijing: Chinese Machine Press, 2010.(in Chinese)

[24] HARSHITHA N B, PRAVEEN KUMAR Y G, KURIAN M Z. An Introduction to Universal Verification Methodology for the digital design of Integrated circuits (IC's): a Review [C]//2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). Coimbatore, India. IEEE, 2021: 1710-1713.

[25] CHOI J, NOH S, HONG S, el al. Finding a Needle in a Haystack: A Novel Log Analysis Method with Test Clustering in Distributed Systems [C]//Proceedings of the Design and Verification Conference (DVCON). United States, 2022: 1-11.

[26] YU D, FOSTER H, FITZPATRICK T. A Survey of Machine Learning Applications in Functional Verification [C]// Proceedings of the Design and Verification Conference (DVCON). United States, 2022: 1-9.

[27] WANG F C, ZHU H B, POPLI P, et al. Accelerating coverage directed test generation for functional verification: a neural network-based framework [C]//Proceedings of the 2018 Great Lakes Symposium on VLSI. Chicago IL USA. ACM, 2018: 207-212.

[28] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[29] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA. IEEE, 2016: 770-778.