文章编号:1671-4598(2024)03-0131-07 DOI:10.16526/j.cnki.11-4762/tp.2024.03.020 中图分类号:TP391 文献标识码:A

基于强化学习算法的神经网络模糊测试技术优化研究

张宇豪，关 昕
（华北计算技术研究所，北京 100083）

摘要：现有神经网络模糊测试技术在测试样本生成阶段通常对初始样本进行随机变异，导致生成样本质量不高，从而测试覆盖率不高；针对以上问题，提出一种基于强化学习算法的神经网络模糊测试技术，将模糊测试过程建模为马尔可夫决策过程，在该模型中，测试样本被看作环境状态，不同的变异方法被看作可供选择的动作空间，神经元覆盖率被看作奖励反馈，使用强化学习算法来学习最优的变异策略，指导生成最优测试样本，使其能够获得最高的神经元覆盖率；通过与现有的主流神经网络模糊测试方法的对比实验表明，基于强化学习算法的神经网络模糊测试技术，可以提升在不同粒度下的神经元覆盖。

关键词：模糊测试；神经网络；强化学习；马尔科夫决策过程；奖励函数

Research on Neural Network Fuzzy Testing Method Based on Reinforcement Learning Algorithm

ZHANG Yuhao, GUAN Xin
(North China Institute of Computing Technology, Beijing 100083, China)

Abstract: Existing neural network fuzzy testing techniques usually adopt random mutations to initial samples during the test sample generation phase, resulting in the low quality of generated samples and testing coverage. To address these issues, a neural network fuzzy testing technique based on reinforcement learning algorithm is proposed. Taking the fuzzy testing process as a Markov decision process, testing samples are regarded as environmental states in the model, different mutation methods as selected action space, and neuron coverage serves as a reward feedback, reinforcement learning algorithms are used to learn optimal mutation strategy, guiding the generation of optimal test samples to achieve the highest neuron coverage. Compared with the experiments for mainstream neural network fuzzy testing methods, the results show that the neural network fuzz testing technique based on reinforcement learning algorithm can improve the neuron coverage of different granularities.

Keywords: fuzzy test; neural network; reinforcement learning; Markov decision process; reward function

0 引言

深度神经网络在图像处理[1]、语音识别[2]、自然语言处理[3]、医疗诊断[4]等领域取得了巨大成功。但其也存在一些安全隐患，在安全性要求高的系统中，神经网络的微小错误可能造成严重后果，因此对其进行测试显得尤为重要。

神经网络是由数据驱动，大多数传统的软件测试方法不适用于神经网络测试，需要根据神经网络特性研究新的测试方法。目前针对神经网络测试的研究已经取得一定进展，相关学者提出了面向神经网络的覆盖准则以及对应的测试方法。这些方法通过最大化神经元覆盖率，找到最佳测试样本，进而发现神经网络的错误行为。文献［5］提出了一个面向深度学习系统的白盒测试框架 DeepXplore。文献［6］提出了第一个差异模糊测试框架 DLFuzz，通过最大化神经元覆盖率，并生成对抗性样本，发现神经网络中的异常行为。文献［7］提出了第一个用于神经网络的模糊测试框架 TensorFuzz，可以发现模型中的数值型错误。文献［8］提出了基于覆盖的模糊测试框架 DeepHunter，用于检测神经网络的缺陷。覆盖度量指标是用于衡量测试充分性的准则。传统软件测试在代码级别[9]和模型级别[10]分别定义了许多覆盖度量指标。而神经网络的特殊结构导致传统覆盖指标无法应用于神经网络测试中[11]，如语句覆盖在测试中可以很容易达到 100%。根据神经网络的内部逻辑，文献［5］首次提出了神经元覆盖作为覆盖指标，它指的是神经网络中激活神经元的比例。文献［12］借鉴 MC/DC 标准的思想提出了基于符号变化、值变化的覆盖指标。文献［13］在神经元覆盖的基础上提出了一组多粒度覆盖指标。文献［7］使用神经元的激活值构成的激活向量作为覆盖指标。这些指标从不同角度反应了神经

收稿日期:2023-11-07； **修回日期**:2023-12-11。

作者简介:张宇豪(1998-),男,硕士研究生。

引用格式:张宇豪,关 昕.基于强化学习算法的神经网络模糊测试技术优化研究[J].计算机测量与控制,2024,32(3):131-137.

网络的内部状态。

由于模糊测试具有自动化程度高，不依赖程序源码等特点，模糊测试是目前测试神经网络的重要方法之一。但是在现有神经网络模糊测试方法中，对测试样本进行变异往往采用随机变异策略，导致生成的测试样本质量不高，很难实现高覆盖率。针对这一问题，本文使用强化学习指导变异策略选择过程，通过设计合理的奖励规则，使得模糊器在进行样本变异时，向神经元覆盖率最大化的方向进行，找到覆盖率更高的变异策略。并以此设计了一种面向神经网络系统的模糊测试方法。

1 理论基础

1.1 基于覆盖的模糊测试

模糊测试是软件测试中最常用的方法之一。其核心思想是通过生成随机、不合法的测试用例，在有限的时间内尽可能多地覆盖目标程序的代码路径或执行路径，发现程序中的潜在漏洞。典型的模糊测试过程如图 1 所示，从种子库中根据优先级选择一个种子输入，通过变异选择器选择变异策略，由变异器执行变异生成变异样本。在此之后，使用变异样本运行待测试程序。如果变异样本产生新覆盖则将变异样本保存在最佳测试样本池中。同时，它还可以跟踪执行的细节，例如执行路径和异常报告。

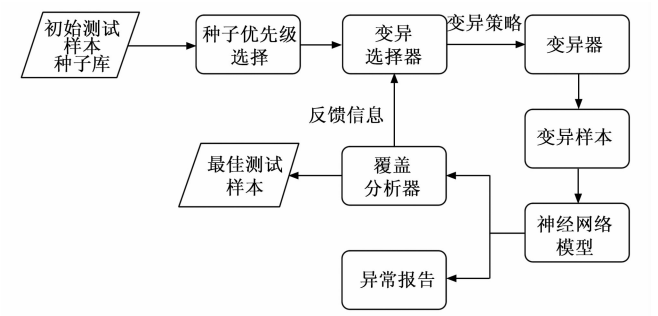

图 1 基于覆盖的模糊测试过程

目前，模糊测试技术在人工智能领域得到了广泛研究。通过对模糊测试[15]进行适应性改造，可以将其应用于神经网络测试中，将目标传统程序映射为神经网络、模糊测试的种子映射神经网络的输入、覆盖反馈映射为神经元覆盖。但目前面向神经网络的模糊测试方法仍然不够完善，制定合适的覆盖标准和有效的变异策略仍是当前研究重点。

1.2 强化学习

强化学习[16]是一种机器学习方法，旨在让智能体与环境进行交互，从中学习如何采取动作来最大化奖励。强化学习的基本架构如图 2 所示，由智能体、环境、动作、奖励组成，其基本思想是，智能体从环境中感知状态，然后采取动作来影响环境，并从环境中获得奖励或惩罚。在不断与环境交互的过程中，智能体通过学习来改善自己的决策策略，从而使它能够更好地完成任务。强化学习技术在很多领域被广泛应用，如自动驾驶[17]、多智能体系统[18]、医疗保健[19]、游戏[20]等。

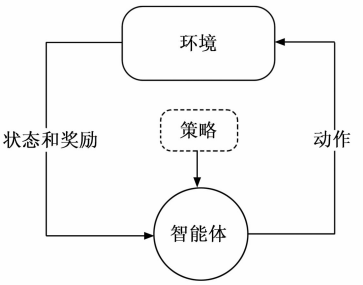

图 2 强化学习基本架构

马尔可夫决策过程是描述强化学习问题的数学框架，其为将模糊测试形式化为强化学习问题提供了基础。马尔可夫性质是指在一个随机过程中，未来的状态只依赖于当前状态，而不受过去状态的影响。在该过程中，智能体能够基于此刻的状态来选择一个动作执行，并接收来自环境的反馈，如奖励和执行动作后的下一个时刻的状态。智能体的主要目标是寻找一种最优策略，以最大化未来的期望回报，即智能体希望通过选择最佳的动作序列来实现长期的最大累积奖励。

2 基于强化学习的神经网络模糊测试技术建模

2.1 问题建模

在模糊测试中，每一次生成新样本的过程就相当于一次与环境的交互，这个过程可以看做是一个在有限状态和动作空间中的序列决策问题。通过将该过程形式化为强化学习问题，使得智能体能够不断学习并优化它的变异选择策略。经过不断地试错和更新，智能体可以逐步学习到一个最优变异策略，指导生成更高质量的新样本。

将强化学习过程形式化为一个有限马尔可夫决策过程，如公式（1）：

$$MDP=(S,A,P,R) \tag{1}$$

其中：S 表示一个有限状态集、A 表示一个有限动作集、P 表示智能体在状态 s_t 下执行动作 a_t 后转移到 s_{t+1} 的转移概率函数、R 表示在状态 s_t 下采取行动 a_t 的奖励。策略 π 是一个当前状态 s_t 要选择的动作 a_t 的映射函数。在执行动作后，状态变为 s_{t+1}，并反馈相应的奖励 r_{t+1}。强化学习的目标就是最大化累积奖励 R。

在强化学习问题中，状态、动作和奖励是 3 个关键的元素。将强化学习算法应用于神经网络模糊测试中，需要将传统的模糊测试过程抽象建模为强化学习算法可解的问题，即从中抽取动作、状态和奖励 3 个元素[21]。以下主要介绍如何定于动作、状态和奖励 3 个元素。

2.1.1 环境状态

在强化学习中，状态表示智能体可以获得的当前环境的有效特征，智能体依据当前环境状态来智能地选择下一步的动作，以达到特定的目标。而在模糊测试中，变异器以一种策略对不同的测试样本进行变异，从而生成具有高质量的新样本。所以在模糊测试中，环境状态应该定义为

测试样本，初始测试样本做为初始环境状态 s_t，在选择一个变异动作 a 后，生成的变异样本为新的环境状态 s_{t+1}。具体以测试样本图像的矩阵形式来表示对应状态 s。

2.1.2　变异动作

模糊测试的一个核心步骤就是对样本进行变异，生成能覆盖更多路径的新样本。对应到强化学习中，智能体选择的动作为变异操作，强化学习模型根据当前环境和策略选择合适的动作。因此动作空间由变异操作组成。通过对常用图像变异方法进行总结，选择如表 1 所示的 8 种方法作为变异动作空间 A，这些变异方法在计算机视觉领域应用广泛，与此同时也可以使图像语义损失最小化[22]。

表 1　变异动作空间

变异类型	具体描述
像素变换	图像对比度变换
	图像亮度变换
	图像噪声变换
	图像模糊变换
仿射变换	图像平移变换
	图像缩放变换
	图像旋转变换
	图像反射变换

需要强调的是，每次变异需要确保生成的变异样本语义不发生改变，一旦变异样本语义发生改变，即使产生了新覆盖，也没有任何意义。为了确保变异生成的样本不会偏离初始样本的语义，每个样本的变异动作只选取一次仿射变换，同时使用公式（2）约束像素变换。这种方法起到限制修改像素的数量和修改像素的绝对值的作用。如果修改的像素数量非常少，则变化程度可以很高；如果修改的像素数量非常多，则对应的变化程度应该更低[8]。

$$f(s,s')=\begin{cases} L_\infty(s,s')\leqslant 255, & L_0(s,s')<\alpha\times size(s) \\ L_\infty(s,s')\leqslant \beta\times 255, & \text{otherwise} \end{cases} \tag{2}$$

该式由 L_0 距离和 L_∞ 距离组成，L_0 表示被修改的像素数量，L_∞ 表示像素修改的最大值。其中 s 和 s' 分别表示原始输入和变异输入，α 和 β 分别表示修改像素数量的比例和限制像素修改值比例，$size$（s）表示图像 i 的像素数。如果被修改像素的数量非常少，则假设它语义不发生改变，L_∞ 的值不受限制。如果被修改的像素数量非常大，则限制最大改变值为 $\beta\times 255$。

2.1.3　奖励反馈

奖励反馈是指智能体在环境中采取特定行动后所获得的反馈信号。正面奖励信号通常表示智能体采取的行动是增益行为，而负面奖励信号表示行为有害。这些奖励反馈可以直接指导智能体选择行动，以此最大化未来奖励，进而帮助智能体采取最佳的行动策略。在传统的模糊测试中，通常将是否触发异常的程序状态来衡量测试的好坏。但触发异常状态往往需要大量时间，难以及时调整变异策略。为了解决该问题，可以用覆盖率指标衡量测试的好坏。通常具备高覆盖率的样本可以充分探索神经网络的网络空间，进而有更高的概率触发神经网络的异常行为。在神经网络模糊测试中，将测试样本输入目标网络并获取神经元覆盖率反馈，若覆盖率增加，则认为当前样本质量较高，强化学习算法根据覆盖率增益来更新变异策略。神经元覆盖率的增益作为强化学习的奖励反馈 R 计算方法，如公式（3）所示：

$$R=\frac{(C\_current-C\_previous)}{(C\_target-C\_previous)} \tag{3}$$

其中：$C\_current$ 表示当前样本的覆盖率，$C\_previous$ 表示先前样本的覆盖率，C_target 表示目标覆盖率，通常设为 1。如果当前样本覆盖率相比之前有所增加，则认为该变异策略是有效的。

2.2　D3QN 深度强化学习模型

本文使用 D3QN（Dueling Double Deep Q Network）强化学习算法来求解问题。它通过借鉴 Double DQN[23] 和 Dueling DQN[25] 两种算法的优点，在 DQN（Deep Q-Network）算法的基础上进行改进。它使用 Double DQN 算法的损失函数，减少过高估计的风险，提高学习的稳定性；使用 Dueling DQN 算法的网络结构，加速收敛。其余流程和 DQN 算法一致。D3QN 算法在解决本文定义的连续状态、离散动作的问题时，可以发挥更好的作用。

D3QN 算法是一种将深度学习和 Q 学习算法相结合的强化学习方法。它使用深度神经网络拟合动作价值函数 $Q(s,\alpha;\theta)$，从而输出当前状态下每个动作对应的 Q 值。随后使用 ε-greed 策略，在动作选择时，可以以概率 ε 选择一个随机动作，或者以 1-ε 的概率选择当前价值网络计算的最大 Q 值对应的动作，如公式（4）所示。在训练初期，往往将 ε 设置为较大的值，使智能体尽可能多的进行探索，避免陷入局部最优解。随着训练的进行，该算法会逐渐降低 ε 的值，如从 1 逐渐减小至 0，这表示当前预测结果的可靠性大大提升。随着这个过程不断进行，预测结果会逐渐收敛，形成一个最优策略。

$$\alpha_t=\begin{cases} \alpha, & \varepsilon \\ \arg max Q(s,\alpha;\theta), & 1-\varepsilon \end{cases} \tag{4}$$

同时在智能体与环境的交互过程中会产生一系列经验序列（s，α，r，s'），将其保存在经验回放池中作为训练样本，每次训练时从经验回放池中随机抽取小批量的数据进行计算。经验回放机制通过随机抽样历史经验数据，避免使用关联数据直接进行训练，降低了训练样本的相关性，提高了结果的可靠性。

D3QN 对损失函数进行了改进。在 DQN 算法中对经验回放池进行随机采样，通过损失函数 $L(\theta)$ 的最小化来更新参数 θ，从而逐步逼近最优的动作价值函数。$L(\theta)$ 如式（6）所示，而 DQN 算法往往选取最大动作价值来近似，这会导致过高估计问题。

$$y=E[r+\gamma \max_{\alpha'} Q'(s',\alpha';\theta')] \tag{5}$$

$$L(\theta) = E[(y - Q(s,\alpha,\theta))^2] \tag{6}$$

为了解决 DQN 算法的过高估计问题。D3QN 使用两个独立的 Q 网络：一个价值网络和一个目标网络。目标网络用于输出能够获得最大 Q 值的动作，而价值网络用于评估这个动作的 Q 值，生成目标值 y'，其公式见式（7）。这种分离使得更新过程更加稳定，降低了过高估计的风险。

$$y' = E[r + \gamma Q(s', \arg\max_{\alpha'} Q'(s',\alpha';\theta');\theta)] \tag{7}$$

D3QN 算法还对网络结构进行优化，提高了动作价值函数 Q（s，α；θ）的准确性。其网络结构如图 3 所示。相比于传统结构，在该网络结构中，隐藏层后多了一个新的结构。即状态经隐藏层处理后分解为两个分支。一个用于学习状态值函数 V（s），另一个用于学习优势函数 Ads（s，α），如式（8）所示。最后将状态值函数 V（s）和优势函数 A（s，α）使用式（9）进行组合以计算 Q 值。其中 V（s）用来估计在状态 s 下采取任意行动的期望回报。A（s，α）用来估计采取动作 α 相对于其他可能动作的优势。正优势函数的动作通常对应更可靠和稳定的决策，选择这样的动作有望在大多数情况下获得正奖励反馈。提高决策的稳定性，并加速训练收敛过程。

$$Adv(s,\alpha) = A(s,\alpha) - \frac{1}{N}\sum_{\alpha' \in A} A(s,\alpha') \tag{8}$$

$$Q(s,\alpha) = V(s) + Adv(s,\alpha) \tag{9}$$

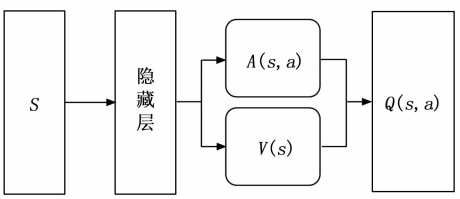

图 3 强化学习算法的网络结构

3 基于强化学习的神经网络模糊测试框架

3.1 总体框架

本文提出的面向神经网络的模糊测试方法的总体框架如图 4 所示，分为模糊测试模块和强化学习模块。其中模糊测试模块由初始测试样本种子库，种子优先级选择，变异选择器，覆盖分析器组成。在测试过程中，根据种子优先级选择从初始测试样本种子库选择合适的样本，其次变异选择器根据强化学习的训练策略对种子进行变异，生成变异样本。将变异样本输入神经网络，统计每一层神经元的输出值来计算覆盖率，若覆盖率提高，则将该样本扩充到最佳测试用例集中。该样本可以再次被选中进行变异，不断重复该过程直到满足结束条件。在过程中最关键的组件是变异选择器，通过强化学习算法训练的最佳变异策略，变异选择器可以生成高质量的变异样本，为整个测试提供支撑。

3.2 模糊测试模块

3.2.1 种子优先级选择

模糊测试需要从初始样本种子库中迭代地选择种子，

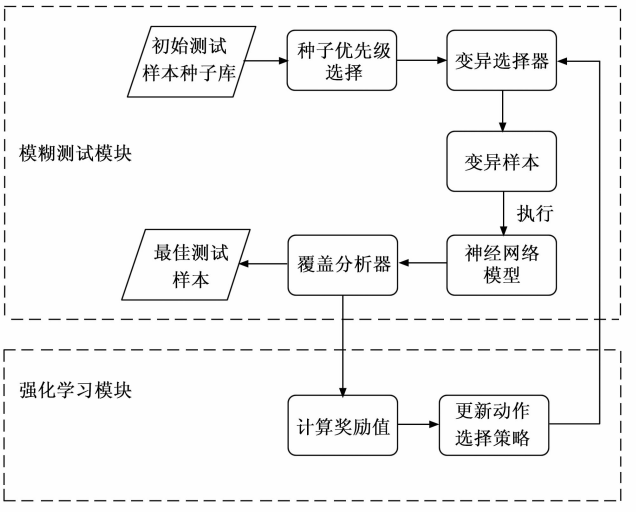

图 4 整体框架

用于生成后续的测试样本。如何选择种子直接影响了后续样本变异的效率。本文根据种子被选择的次数设置种子的选择概率，被选择的次数越少对应概率越高，同时当种子被选择的次数达到一定数量后，该种子的选择概率将被设置成一个较小值。

该策略的基本思想是优先选择较少被选择的种子。新生成的变异种子由于获得了更高的覆盖率，被视为高质量种子，因此具有更高的选择概率。同时，当某个种子被选择的次数达到一定数量后，该种子的选择概率会变为一个固定的小概率值，防止某些种子在测试过程中反复被选中，从而浪费资源。

3.2.2 变异选择器

变异选择器根据强化学习的最优策略对样本进行变异。本文采用的变异方法分别是像素变换和仿射变换。其中像素变换包括对比度变换、亮度变换、噪声变换、模糊变换。仿射变换包括平移变换、缩放变换、旋转变换、反射变换。变异选择器根据最优变异策略可以组合这些变异方法，实现不同级别的变异，以产生更多样化的测试样本。

相较于传统的变异选择器的随机变异策略，该变异选择器加入了强化学习方法，利用神经元覆盖率的增加作为反馈信息，得到最佳变异策略。不同于传统模糊器的盲目性和随机性。通过强化学习算法指导变异选择器可以更好地选择种子变异方法，使变异后的种子能最大程度地增加覆盖率或引发更多的行为错误。

3.2.3 覆盖分析器

在模糊测试中，覆盖分析器是用来分析程序执行过程中的覆盖情况。通过分析覆盖信息，可以评估测试用例的有效性和发现潜在问题的能力，有助于确定测试是否足够全面和有效。覆盖分析器的结果可以用于指导进一步的测试样本变异策略，以增加覆盖率。一种没有任何覆盖引导的模糊器会盲目地对种子进行变异操作，不知道生成的测试输入是否可用。这样的模糊器经常会保留那些不带来新

的有用信息的种子，显著降低了模糊测试的有效性。本文选择了 4 种不同的准则作为不同的反馈。这些准则可以有效表示神经网络的内部状态。

1）神经元覆盖（NC）：

神经元覆盖[5]度量神经网络中激活神经元的比例。神经元覆盖将神经元的状态分成激活和非激活两部分。给定一个输入，如果它的输出值高于预设的阈值，神经元就会被激活。如公式（10）所示：

$$NCov(T,k)=\frac{|\{n\mid\forall x\in T,out(n,x)>t\}|}{|N|} \tag{10}$$

2）K 多段神经元覆盖（KMNC）：

K 多段神经元覆盖[13]度量神经网络中一组神经元覆盖其上下界范围的程度。给定一个神经元 n 和输出的上下界，将上下界划分为 k 个部分。如果神经元输出在某一段，则认为该段被覆盖。如公式（11）所示：

$$KMNCov(T,k)=\frac{\sum_{n\in N}|\{S_i^n\mid\exists x\in T:\phi(x,n)\in S_i^n\}}{k\times|N|} \tag{11}$$

3）神经元边界覆盖（NBC）：

神经元边界覆盖[13]度量神经元主功能范围之外的边缘区域被覆盖的程度。给定一个输入 x，若 ϕ（x，n）值超过上下边界，则表明边缘区域被覆盖。其上下边界公式为：

$$UpperCornerNeuron=\{n\in N\mid\exists x\in T:\phi(x,n)\in(high_n,+\infty)\} \tag{12}$$

$$LowerCornerNeuron=\{n\in N\mid\exists X\in T:\phi(x,n)\in(-\infty,low_n)\} \tag{13}$$

神经元边界覆盖公式为：

$$NBCov(T)=\frac{|UpperCornerNeuron|+|LowerCornerNeuron|}{2\times|N|} \tag{14}$$

4）强神经元覆盖（SNAC）：

强神经元覆盖[13]度量神经元的上界边角区域的覆盖程度。

$$SNACov(T)=\frac{|UpperCornerNeuron|}{|N|} \tag{15}$$

3.3　强化学习模块

强化学习模块基于 D3QN 算法进行训练学习最佳变异策略，指导变异测试样本生成。该模块根据输入样本选择变异操作，生成新样本传入神经网络，利用奖励反馈更新变异策略，该过程如图 5 所示。

首先把神经网络、初始样本池作为输入，随机初始化价值网络的参数 θ 和目标网络的参数 $\theta=\theta'$，并根据参数初始化两个网络。在一个新回合中，从初始样本池中随机选取一个样本作为初始环境状态 s，根据价值网络选择变异动作 α，基于变异动作 α 对样本进行变异生成新样本并将其输入到神经网络中执行。在执行完毕后，返回覆盖率奖励 r 并得到新的环境状态 s'，同时将该四元组（s，α，r，s'）存储到经验重放池 D 中。在进行训练时，从经验重放池中随机

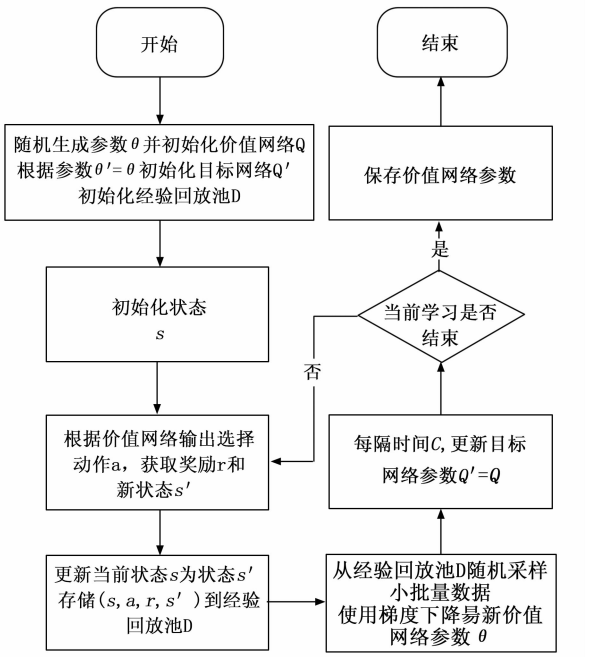

图 5　强化学习算法训练流程

采样得到 N 个四元组数据（s，α，r，s'），使用目标网络计算损失函数 L（θ'），通过最小化 L（θ'）更新参数 θ'。通过梯度下降更新价值网络参数，每隔一定步数将目标网络的参数更新为目前价值网络的参数值。一个回合的终止条件是，当生成样本出现识别错误或不满足式（2）函数关系时，则停止学习，通过不断学习最终得到一种最优策略，它能够智能地选择变异动作以最大化奖励。

4　实验与结果

4.1　数据集和模型

本文选择了两个流行的公开数据集 MINST[26]和 CIFAR-10[27]作为实验数据集。MNIST 是一个用于手写数字图像识别的数据集，其中 6 万张图像作为训练集，1 万张图像作为测试集，分为 10 类（即从 0 到 9 的手写数字）。每个 MNIST 图像是一个尺寸为 28×28×1 的单通道图像。

CIFAR-10 是一个用于图像分类的图像集合，其中 5 万张图像作为训练集，1 万张图像作为测试集，分为 10 个不同类别。每个 CIFAR-10 图像是一个尺寸为 32×32×3 的三通道彩色 RGB 图像。由于 CIFAR-10 数据集更大、复杂性更高，CIFAR-10 的分类任务通常比 MNIST 的分类任务更难。

本文将分别采用 LeNet1、LeNet4、LeNet5 在 MNIST 数据集上进行手写数字分类任务，采用 VGG16 在 CIFAR-10 上进行图像分类任务。神经网络模型的具体参数如表 2 所示。

4.2　评价指标

本实验从等价类原则出发，选取如表 3 所示的 4 种不同粒度的神经元覆盖度量指标。分别是神经元覆盖（NC）、k-

多段神经元覆盖（KMNC）、神经元边界覆盖（NBC）、强神经元激活覆盖（SNAC）。这些指标的粒度不同，反应神经网络内部状态也不同，选取这 4 个指标，可以有效检验本文方法的效果。具体描述见 3.2.3 节。

表 2 神经网络模型

数据集	神经网络模型	训练准确率/%	测试准确率/%
MINIST	LeNet1	98.2	98.0
	LeNet4	98.7	98.6
	LeNet15	99.1	98.9
CIFAR-10	VGG16	97.5	89.9

表 3 评价指标

覆盖准则	描述
神经元覆盖率	激活神经元的比例
k-多段神经元覆盖	覆盖给定上下界范围神经元的比率
神经元边界覆盖	覆盖神经元边界区的比率
强神经元覆盖	覆盖活跃边界区域的比率

4.3 对比方法

DeepHunter 是一个自动模糊测试框架，用于寻找神经网络模型的潜在缺陷。DeepHunter 提出了蜕变变异的概念，即图像语义在种子变异过程中保持不变的性质。该技术通过 8 种图像变异策略变异种子，并利用多个覆盖度量指标作为反馈，从不同角度指导测试用例生成。同时该技术提出了随机选择和概率选择两个种子选择策略，提高了模糊测试揭示故障和探索模型内部状态的有效性与效率。DeepHunter 的有效性和效率在 3 个流行的数据集和 7 个具有不同复杂性的模型上得到了验证。

Tensorfuzz 是一个基于覆盖引导的模糊测试方法。该技术提出了一种激活向量的覆盖准则，并使用最近邻算法计算测试过程中的覆盖率。在 Tensorfuzz 中，由覆盖信息指导神经网络的初始样本进行变异，进而完成神经网络测试。

4.4 参数设计

在本实验中，将神经元覆盖中神经元激活阈值设置为 0.5。对于 K 多段神经元覆盖，设 $k=1\ 000$，这表示每个神经元的激活值对应主函数区间分为 1 000 段。对于神经元边界覆盖和强神经元激活覆盖，将训练中遇到的最小激活值 l 设置为下限，最大激活值 u 设为上限，δ 是神经元输出值的标准差。具体如表 4 所示，这些是原始研究[13]中推荐的设置。

表 4 参数设计

覆盖指标	参数设计
神经元覆盖	Threshold = 0.5
K-多段神经元覆盖	$K=1\ 000$
神经元边界覆盖	$UB=u+0.5*\delta$ $LB=l+0.5*\delta$
强神经元激活覆盖	$UB=u+0.5*\delta$

4.5 实验结果分析

实验的目标是证明本文所提出的方法能够生成优秀的测试样本，提高不同粒度下的神经元覆盖率。在实验中，采用了本文提出的方法以及两个对比方法，在两个不同数据集上，分别针对四种不同的神经网络进行了测试。在 LeNet 模型中将执行 1 000 个测试样本作为终止条件，在 VGG 模型中将执行 500 个测试样本作为终止条件，得到在神经元覆盖（NC）、K-多段神经元覆盖（KMNC）、神经元边界覆盖（NBC）、强神经元激活覆盖（SNBC）这 4 种覆盖度量准则下的覆盖率。具体的测试结果如表 5 所示。

表 5 不同方法达到的覆盖率峰值

模型	方法	NC/%	KMNC/%	NBC/%	SNAC/%
LeNet1	初始样本	49.2	64.5	9.5	9.5
	DeepHunter	56.4	83.7	35.3	35.7
	Tensorfuzz	50.5	72.5	18.8	16.5
	本文方法	50.8	90.1	50.1	37.1
LeNet4	初始样本	69.3	70.5	12.3	14.8
	DeepHunter	77.3	72.3	29.5	40.1
	Tensorfuzz	74.2	72.9	19.5	22.5
	本文方法	75.0	77.4	31.1	44.6
LeNet5	初始样本	66.3	67.3	9.9	15.9
	DeepHunter	72.5	70.5	18.6	30.5
	Tensorfuzz	68.1	68.8	13.4	17.0
	本文方法	78.4	73.2	23.1	35.5
VGG16	初始样本	40.4	39.8	12.9	14.2
	DeepHunter	41.3	47.5	18.5	19.5
	Tensorfuzz	41.9	42.5	16.9	16.1
	本文方法	42.2	49.9	21.7	23.0

表 5 总结了初始样本对应的覆盖率以及 DeepHunter、Tensorfuzz 和本文方法在不同模型下达到的覆盖率。相比于初始样本的覆盖率，本文方法在不同模型下的所有覆盖率指标均得到了提升。相比于 Tensorfuzz，本文方法在 LeNet 和 VGG16 模型下覆盖率均取得了较大提升。相比于 DeepHunter，本文方法在使用 LeNet 模型的 KMNC、SNAC 和 NBC 指标中都获得了更高的覆盖率，在 VGG16 模型上同样可以获得更高的覆盖率，在 LeNet1 模型中本文方法实现的 NC 指标提升不高。

在 LeNet 1 的 NC 实验中，本文方法提升覆盖率效果不佳。这是因为神经元覆盖的阈值设计使得在小尺度模型上很难覆盖深度状态，覆盖率提升阈值出现的时间更早，从而很难找到覆盖率增加的样本，使得奖励反馈机制失效，最终变异策略变成了随机变异。

在 KMNC 上，本文方法可以稳定提高覆盖率。这是因为 KMNC 这种小粒度测试指标可以为系统提供更多的反馈，从而更好地指导学习变异策略。

在 SBNC 和 NBC 上初始覆盖率就比较低是因为 SBNC 和 NBC 主要关注边界区域的神经元，即激活值超过主边界区域的神经元。而神经元激活值超过主边界区域的情况相

对较少，因此覆盖率不高，但是经过变异，也可以提升覆盖率。

综上所述，本文方法根据强化学习算法生成最优变异策略，使其指导生成的样本可以获得更高的神经元覆盖率，验证了本文方法是一种有效的变异策略优化方法。

5　结束语

本文的主要研究内容是基于强化学习算法对神经网络模糊测试中的测试样本变异环节进行优化，将神经网络模糊测试建模为一个马尔可夫决策过程，定义了适用于神经网络模糊测试的环境状态、奖励反馈、变异动作，并通过强化学习算法学习一个最佳变异策略，指导样本变异过程，生成最佳变异样本。与随机变异相比，本文提出的方法可以更快地推动样本向覆盖率更高的方向变异。通过实验表明，本文的方法可以生成高质量测试样本，提高测试覆盖率。在未来的研究中，将利用现有的变异策略，提高测试前期生成样本的质量；设计更加有效的强化学习模型和参数，提高其性能。

参考文献：

[1] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet large scale visual recognition challenge [J]. International Journal of Computer Vision, 2015, 115 (3): 211 - 252.

[2] COLLOBERT R, WESTON J, BOTTOU L, et al. Natural language processing (almost) from scratch [J]. Journal of Machine Learning Research, 2011, 12 (ARTICLE): 2493 - 2537.

[3] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks [J]. Advances in Neural Information Processing Systems, 2014, 27.

[4] DAN C C, GIUSTI A, GAMBARDELLA L M, et al. Deep neural networks segment neuronal membranes in electron microscopy images [C]. 2012: 2852 - 2860.

[5] PEI K X, CAO Y Z, YANG J F, et al. DeepXplore: automated whitebox testing of deep learning systems [C] // 26th Symposium on Operating Systems Principles, 2017: 1 - 18.

[6] GUO J, JIANG Y, ZHAO Y, et al. Dlfuzz: Differential fuzzing testing of deep learning systems [C] //Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2018: 739 - 743.

[7] ODENA A, OLSSON C, ANDERSEN D, et al. TensorFuzz: debugging neural net-works with coverage-guided fuzz-ing [C] //2019 International Conference on Machine Learning, 2019: 4901 - 4911.

[8] XIE X, MA L, JUEFEI-XU F, et al. DeepHunter: a coverage guided fuzz testing framework for deep neural networks [C] // 28th ACM SIGSOFT Inter-national Symposium on Software Testing and Analysis, 2019: 146 - 157.

[9] YUE J, HARMAN M. An analysis and survey of the development of mutation testing [J]. IEEE Transactions on Software Engineering, 2011, 5 (37): 649 - 678.

[10] AMMANN P, OFFUTT J. Introduction to software testing [M]. Cambridge University Press, 2008.

[11] 王　赞，闫　明，刘　爽，等. 深度神经网络测试研究综述 [J]. 软件学报，2020，31 (5)：1255 - 1275.

[12] SUN Y, HUANG X, KROENING D. Testing deep neural networks [J]. arXiv: 1803.04792, 2018.

[13] MA L, XU J F, ZHANG F Y, et al. DeepGauge: multi granularity testing criteria for deep learning systems [C] // 33rd ACM/ IEEE International Conference on Automated Software Engin-eering, 2018: 120 - 131.

[14] 陈金鑫. 基于多元测试用例的神经网络模糊测试系统的设计与实现 [D]. 南京：南京大学，2020.

[15] 穆星旭. 基于启发式搜索的神经网络模糊测试技术优化研究 [D]. 西安：西安电子科技大学，2021.

[16] 杨思明，单　征，丁　煜，等. 深度强化学习研究综述 [J]. 计算机工程，2021，47 (12)：19 - 29.

[17] SALLAB A, ABDOU M, PEROT E, et al. Deep reinforcement learning framework for autonomous driving [J]. Electronic Imaging, 2017, 2017 (19): 70 - 76.

[18] NGUYEN T T, NGUTEN N D, NAHAVANDI S. Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications [J]. IEEE Transactions on Cybernetics, 2020, PP (99): 1 - 14.

[19] YU C, LIU J, NEMATI S, et al. Reinforcement learning in healthcare: a survey [J]. ACM Computing Surveys (CSUR), 2021, 55 (1): 1 - 36.

[20] SHAO K, TANG Z, ZHU Y, et al. A survey of deep reinforcement learning in video games [J]. arXiv Preprint arXiv: 1912.10944, 2019.

[21] 张　政. 基于 DDPG 强化学习算法的模糊测试技术研究 [D]. 北京：北京邮电大学，2021.

[22] YE A, WANG L, ZHAO L, et al. Ex2: Monte Carlo Tree Search-based test inputs prioritization for fuzzing deep neural networks [J]. International Journal of Intelligent Systems, 2022, 37 (12): 11966 - 11984.

[23] Human-level control through deep reinforce-ment learning [J]. Nature, 2015, 518 (7540): 529 - 533, a3.

[24] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double Q-learning [C] //Proceedings of the 30th AAAI Conference on Artificial Intelligence. 2016: 2094 - 2100.

[25] WANG Z, SCHAUL T, HESSEL M, et al. Dueling network architectures for deep reinforcement learning [C] //International Conference on Machine Learning. PMLR, 2016: 1995 - 2003.

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

[27] KRIZHEVSKY A, HINTON G. Learning multiple layers of features from tiny images [D]. Master's thesis, University of Tront, 2009.