DOI 10.31891/2307-5732-2025-347-43
УДК 004.8

СОРОКОЛІТ ВІТАЛІЙ
Хмельницький національний університет
https://orcid.org/0009-0002-0182-221X
e-mail: visor@smile.fr

МЕТОД ТЕСТУВАННЯ КОДУ ІЗ ВИКОРИСТАННЯМ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ

Одним із головних аспектів, які впливають на конкурентоздатність програмного забезпечення є його якість, що є багатогранним та широким поняттям. Тестування як засіб досягнення якості програмного забезпечення є актуальною проблемою, що сприяє безкінечному пошуку розв'язання та впровадження цих рішень, а також є фундаментальним дослідженням галузі інформаційних технологій, зокрема із використанням штучного інтелекту.

Для забезпечення та підтримки якості програмного коду, який є ключовим чинником якості програмного забезпечення, використовується тестування. Тестування може проводитися вручну або автоматизовано.

В даній статті подано розгляд вдосконаленого методу тестування із використанням штучного інтелекту, а саме нейромережевих моделей для аналізу коду програмного продукту.

Ключові слова: програмне забезпечення, якість програмного забезпечення, штучний інтелект, нейронні мережі.

VITALII SOROKOLIT
Khmelnytskyi National University

A METHOD OF TESTING CODE USING NEURAL NETWORK MODELS

One of the main aspects affecting the competitiveness of software is its quality, which is a multifaceted and broad concept. Testing as a means of achieving software quality is an urgent problem that contributes to the endless search for solutions and implementation of these solutions, and is fundamental research in the field of information technology, in particular with the use of artificial intelligence.

To ensure and maintain the quality of program code, which is the main factor of software quality, testing is used, which in turn is divided into manual and automated.

Testing as the main way to control the quality of software solutions can be defined as the process of executing a program to check the correspondence between its real and expected behavior. Testing is performed on a finite set of test cases selected in a certain way. In the most general case, the testing process can be represented by a set of four main activities: work planning, test development, test execution, and analysis of software testing results.

To solve these tasks when solving automated testing problems, a method based on the use of artificial neural networks can be proposed. In practice, the use of neural networks shows more than decent results in a wide variety of applications. Since software can be viewed as a function in a multidimensional space, neural network methods can be applied to such a model.

Improving techniques and methods of testing complex software products deserves great attention from software development companies due to the undeniable importance of providing the proper level of quality of the released software.

This article presents an advanced method of maintaining the quality of program code using artificial intelligence, namely neural network models as an error analyzer.

Keywords: software, software quality, artificial intelligence, neural networks.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Оцінка якості може проводитись також і на основі вартості готового програмного продукту тощо. Низька якість програмного забезпечення (ПЗ) може бути однією з причин, що викликають порушення діяльності організації та фінансові втрати. З метою створення якісного ПЗ потрібна грамотна організація всіх етапів розробки [1], і, зокрема, етапу тестування як найважливішого засобу для визначення ступеня відповідності ПЗ заданим вимогам. Для покращення якості програмного забезпечення важливим процесом виступає автоматичне тестування. Автоматичні тести можуть допомогти переконатися, що вихідний код є надійним у багатьох умовах використання та відповідає заявленим вимогам, зокрема продуктивності та безпеки.

Аналіз останніх досліджень і публікацій

В цілому дослідженням питань, що пов'язані із якістю програмного забезпечення приділено досить значна увага. Зокрема, Лаврищева К. у своїх дослідженнях постійно звертається до питання якості програмного забезпечення. Такі дослідники як В. Юрчишин розглядав методологію оцінки якості програмного коду та програмного забезпечення в цілому для нафтогазового комплексу.

Методам обробки об'єднаних повідомлень певної довжини у програмному коді присвячені роботи М. Белларе, В. Сталлінгса, Д. Блека, Р. Мішри, С. Шарма, Б. Отмана; Фомін О. Крикун В., Орлов А. розглядали математичні моделі забезпечення якості програмного забезпечення для інтерпретації динамічних нейронних мереж. Ванмалі М., Ласт М., Кендел А. описують методологію використання нейронної мережі в процесі виконання тестування програмного забезпечення.

Таким чином актуальною проблемою залишається удосконалення методів та засобів, а також алгоритмів роботи системи підтримки якості програмного коду із використанням автоматичного тестування на основі нейромережевих моделей.

Формулювання цілей статті

Метою дослідження є удосконалення методу підтримки якості коду програмних систем із використанням автоматичного тестування на основі нейромережевих моделей.

Виклад матеріалу дослідження

Удосконалення прийомів та методів тестування складного програмного забезпечення відноситься до актуальних проблем галузі інформаційних технологій та заслуговує на велику увагу компаній-девелоперів, у зв'язку з незаперечною важливістю надання відповідного рівня якості розроблюваних програмних продуктів.

Тестування як основний [2] спосіб здійснення та підтримки якості програмних продуктів визначається як процес, в ході якого відбувається виконання програми та перевіряється чи відповідає реальна поведінка очікуваній. Щоб провести тестування необхідно мати кінцевий набір тестових випадків, що обираються певним чином. Якщо розглядати узагальнений випадок, то безпосередній процес тестування складається із сукупності таких основних складових (активностей): план потрібних робіт, розроблення тестів, виконання тестування, здійснення аналізу результатів перевірки під час виконання тестування.

Для вирішення зазначених завдань під час вирішення проблем автоматизованого тестування може бути запропонований метод, що ґрунтується на використання штучних нейронних мереж. На практиці використання нейронних мереж показує досить гарні результати у найрізноманітніших застосунках. Так як програмне забезпечення може бути розглянуте як функція в багатовимірному просторі, для такої моделі можна застосовувати й методи нейронних мереж.

Мозок людини є складною, нелінійною, паралельною системою обробки інформації, здатною до самостійної організації своїх структурних компонентів. Для досягнення високої продуктивності нейронні мережі використовують велику кількість зв’язків між базовими елементами обчислень - нейронами. У штучних нейронних мережах аналогічно працюють штучні нейрони.

До будови штучних нейронних мереж відносять прості процесори (штучні нейрони), які взаємопов’язані між та здійснюють обмін сигналами. Кожен штучний нейрон обробляє сигнали, які отримує від інших нейронів, і передає їх далі. У цьому процесі штучний нейрон імітує властивості біологічного (природного) нейрона: на його вхід надходить кілька сигналів, кожен із яких збільшується на відповідну вагу (аналог синаптичної сили). Отримані значення підсумовуються, визначаючи рівень активації нейрона.

Багаторівневу нейронну мережу можна навчати, щоб вони могла виконувати конкретні завдання, використовуючи набір вхідних даних, згенерованих у довільному порядку. У результаті навчання мережа досягає необхідного рівня точності у межах визначеної програми і відповідно стає моделлю-симулятором.

Етап вдосконалення функціоналу та створення нових версій програмного продукту є важливим та вимагає проведення так званого регресійного тестування, що полягає у перевірці вже реалізованих змін, таких як додавання нового функціоналу, виправлення помилок, міграція бази даних чи зміна операційної системи. Метою є перевірка правильності та стабільності роботи існуючого функціоналу. Якщо під час регресійного тестування використовуються тестові дані зі штучними нейронними мережами, результати порівнюються із прогнозами нейронної мережі.

Рис. 1. Пропонований метод тестування

Припустивши, що нові версії програмного продукту не змінюють наявний функціонал, досить очікуваним може бути той факт, що маючи однакові вхідні дані нейронна мережа і реальний застосунок надасть ідентичні результати. За допомогою інструменту для порівняння можна буде зробити висновок, коректні чи некоректні результати для програмного застосунку, для якого проводиться тестування. Дана методика для проведення тестування подана на рис. 1.

Використання моделі програмного забезпечення на основі штучної нейронної мережі замість оригінальної версії програмного продукту може виявитися ефективним з кількох причин.

По-перше, оригінальна версія може втратити придатність для тестування через зміни в апаратній платформі або операційній системі.

По-друге, на певному етапі тестування більшість пар вхідних і вихідних даних програми може не бути критичною. У таких випадках застосування нейронної мережі для автоматизованого моделювання вихідного застосунку дозволяє значно заощадити комп'ютерні ресурси.

По-третє, зберігання повного набору тестових сценаріїв разом із вихідними значеннями для початкового застосунку може бути практично неможливим для складних реальних програм.

Орієнтовна схема процесу навчання зображена на рис. 2.

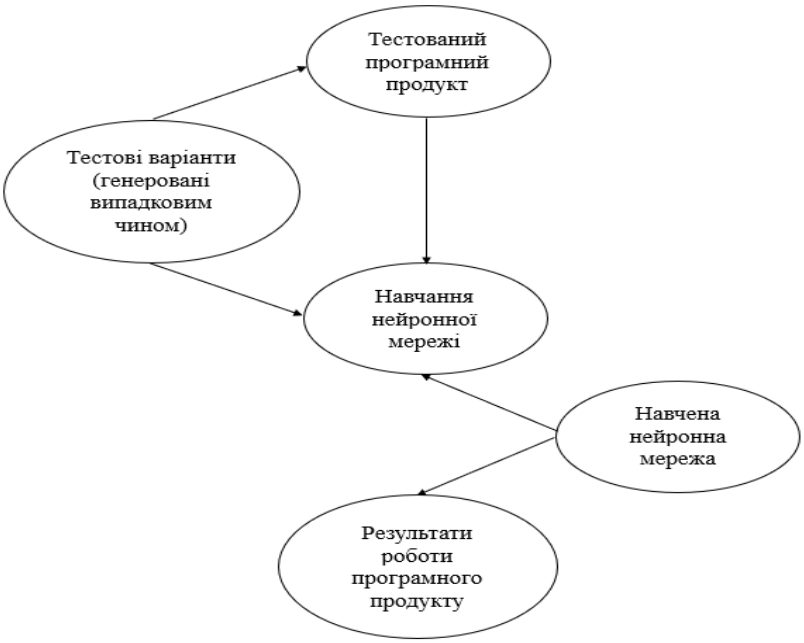

Рис. 2. Схема процесу навчання нейронної мережі

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Отже, весь описаний вище матеріал, дозволяє зробити висновки, про те, що застосовування моделі програмного забезпечення, яка побудована на основі штучної нейронної мережі замість оригінальної версії програмного продукту може бути доцільним та ефективним за цілим рядом причин.

1. Якщо відбуваються зміни у апаратній складовій або операційній системі, то первинна версія може бути погано тестованою.

2. Збереження значної кількості ресурсів комп'ютера завдяки використанню нейронної мережі для автоматизованого моделювання вихідного застосунку, оскільки більшість пар вхідних та вихідних даних застосунку може бути некритичним на даному етапі процесу тестування.

3.Збереження вичерпного набору тестових сценаріїв із вихідними значеннями початкової програми може бути неможливим для реальних застосунків.

Варто враховувати, що навіть за умови порівняння початкової та доопрацьованої версій програмного забезпечення відсутня гарантія безпомилковості початкової версії. Таке порівняння може не виявити ситуацій, коли обидві версії працюють некоректно. У цьому контексті використання нейронних мереж забезпечує додатковий параметр для підвищення надійності тестування. Оскільки програмне забезпечення, яке моделює нейронна мережа, було оновлено до нової версії, сама мережа може адаптуватися та навчатися класифікувати нові дані. Таким чином, нейронна мережа демонструє здатність до постійного навчання на основі нових версій змінюваного програмного продукту.

Окрім автоматизації тестування, штучні нейронні мережі можуть бути застосовані й для інших завдань. Наприклад, вони здатні генерувати вхідні дані для тестових сценаріїв, які з високою ймовірністю допоможуть виявити помилки в роботі програмного забезпечення.

Необхідно зазначити також, що інтеграція нейронних мереж із методами автоматизації тестування сприяє покращенню результатів у завданнях перевірки складних програмних систем. Такий підхід сприяє підвищенню ефективності процесу тестування, що в свою чергу призводить до забезпечення, підтримки та підвищення якості програмного продукту.

Література

1. Sacha Lity, Manuel Nieke, Thomas Thüm, Ina Schaefer. Retest test selection for product-line regression testing of variants and versions of variants. Journal of Systems and Software, No.147, January 2019. P. 46-63.

2. AbdulRahman A. Alsewari, Muhammad N. Kabir, Kamal Z. Zamli. Software Product Line Test List Generation based on Harmony Search Algorithm with Constraints Support. International Journal of Advanced Computer Science and Applications, 2019. Vol. 10. No. 1. P. 605-610.

3.Wu Y. and other Characterizing the Occurrence of Dockerfile Smells in Open-Source Software: An Empirical Study. IEEE Access. 2020. T. 8. P. 34127–34139.