

структурно-чутливих параметрів некристалічних напівпровідників, яка дозволяє підвищити вірогідність контролю.

Література

1. Redaelli A. Reliability study of phase-change nonvolatile memories / A. Redaelli, A. Pellizzer, F. Ottogalli and other // IEEE Transactions on Device and Materials Reliability. – Sept. 2004. – Vol. 4, issue 3. – P. 422—427. ISSN 1530-4388.
2. Дерффель К. Статистика в аналитической химии / Дерффель К. – М. : Мир, 1969. – 248 с.

Надійшла 22.5.2012 р.
Рецензент: д.т.н. Кучерук В.Ю.

УДК 004.05

О.В. ПОМОРОВА, О.С. ЛИСАК
Хмельницький національний університет

НЕЙРОМЕРЕЖНИЙ МЕТОД ОЦІНКИ СКЛАДНОСТІ ПРОГРАМНИХ СИСТЕМ

У статті проведено аналіз методів оцінки складності систем. Представлено метод оцінки складності програмних систем з використанням штучної нейронної мережі. Описано процес формування навчальної вибірки, навчання та функціонування штучної нейронної мережі.

This paper presents the result of analysis of methods for assessing the complexity of systems. The method for estimates the complexity of software systems using artificial neural network are presented. The process of educational selection, training and operation of artificial neural network are described.

Ключові слова: штучна нейронна мережа, метод оцінки складності, програмна система, інформаційна складність, структурна складність.

Вступ. Оцінка складності програмної системи має суттєвий вплив на прийняття рішень стосовно доцільності та вартості реалізації проекту цієї системи. Від складності системи залежить трудомісткість, терміни розробки і бюджет проекту. Між складністю та надійністю системи також існує тісний зв'язок. Очевидно, що чим вища складність системи, тим вищі вимоги до засобів забезпечення її надійності. За приблизними оцінками, на сьогодні витрати на розробку програмного забезпечення складають близько 275 мільярдів доларів, 72 % проектів досягають етапу впровадження і лише 26 % усіх проектів завершуються успішно. Однією з причин таких низьких показників завершення проектів є невірна оцінка складності проекту при формуванні бюджету та команди розробників, тому задача оцінки складності програмних систем є актуальною та потребує нагального розв'язання.

Кожен розробник індивідуально підходить до оцінки складності системи, обчислюючи її за тими критеріями, які на його думку є найбільш вагомими. Р.В. Мешеряков у статті "Критерій структурної складності інформаційної системи" [1], основним параметром складності пропонує використовувати показник ентропії для оцінки трудомісткості виконуваних операцій в інформаційній системі. На прикладі оцінки структурної складності задач аналізу та синтезу мовного сигналу автор ілюструє методику розбиття блоків обробки інформації для перерозподілу ресурсів у відповідності до кількості інформації, що опрацьовується у кожному блоці. Також інформаційну ентропію як основний критерій складності виділяють К. Willcox, D. Allaire, J. Deyst у роботі [2]. Федорів Р.В [3] пропонує проводити обчислення базової оцінки складності програмних інформаційних систем на основі принципу оптимальності Белмана. Пошук оптимальної ієрархії розривів дуг виконується повним перебором всіх можливих варіантів. На кожному рівні рекурсії здійснюється пробний розрив дуг орграфа з наступним обчисленням складності системи. Таким чином забезпечується пошук оптимальної ієрархії розривів дуг, що приводить оцінку складності до глобального мінімуму. Колмогоров опирається на алгоритмічну оцінку складності програмного коду [4]. Складність Колмогорова базується на ентропії Шенона, котра відповідає інтуїтивному уявленню про міру інформації.

Обчислення складності систем потребує значних затрат часу та ресурсів. Тому для оцінки складності програмних систем, особливо на початкових етапах життєвого циклу – проектування, програмування компонентів і т.і., доцільним є визначення не точних оцінок складності, а динаміки її зростання чи спадання залежно від зміни значень параметрів, що характеризують програмну систему. У такому випадку для оцінки складності програмних систем доцільним є використання інтелектуальних методів, зокрема апарату штучних нейронних мереж (ШНМ), котрі на основі неповних та неточних даних дають можливість здійснити оцінку складності та виявити динаміку її зміни при модифікації значень параметрів програмної системи.

Постановка задачі. Розробити нейромережний метод оцінки складності програмних систем, котрий забезпечить можливість врахування різних характеристик таких систем.

Огляд відомих методів обчислення складності систем. На сьогодні науковцями розроблено ряд методів визначення складності систем. Розглянемо найбільш відомі методи.

Метод інформаційно-структурної складності [5]. За цим методом складність системи складається з двох частин: SC (структурна складність) та DC (інформаційна складність). Структурна складність окремої процедури програмної системи обчислюється за формулою:

$$SC = SFOUT^2, \quad (1)$$

де SFOUT – кількість викликаних процедур.

Інформаційна складність окремої процедури обчислюється за формулою:

$$DC = IOvars / (SFOUT + 1), \quad (2)$$

де IOvars – кількість вхідних/вихідних змінних.

На основі результатів обчислення структурної та інформаційної складності для кожної процедури, можна обчислити загальну складність всієї системи:

$$SYSC = \sum_{i=0}^n CS_i + DC_i, \quad (3)$$

де n – кількість процедур в програмній системі.

Відносна складність системи не залежить від розміру системи, а вимірює її середню складність.

Відносна складність системи:

$$RSYSC = average(CS + DC) \quad (4)$$

Метод Маккейба [6]. Маккейбом було вперше запропоновано представлення програм у графічному вигляді. Основною метрикою складності він пропонує вважати цикломатичну складність графу програми, що характеризує трудомісткість тестування програмного забезпечення. Для обчислення цикломатичного числа Маккейба Z (G) використовується формула:

$$Z(G) = e - v + 2p \quad (5)$$

де e – число дуг орієнтованого графа G, v – число вершин, p – число компонентів зв'язності графа.

Число компонентів зв'язності графа можна розглядати як кількість дуг, які необхідно додати для перетворення наявного графа в сильно зв'язний граф. Сильно зв'язним називається граф, де будь які дві вершини взаємно досяжні. Для графів коректних програм, а саме графів, що не мають недосяжних від точки входу ділянок, сильно зв'язний граф отримується шляхом замикання дугою вершини, що позначає кінець програми, з вершиною, що позначає точку входу програми.

Цикломатичне число Маккейба показує необхідну кількість проходів для перекриття усіх контурів сильно зв'язного графа або кількість тестових прогонів програми, необхідних для вичерпного тестування за критерієм "працює кожна гілка".

Метод PERT (Program Evolution and Review Technique) [7]. Дана методика призначена для аналізу часу, який потрібен для виконання кожної окремої задачі, а також визначення мінімального необхідного часу для виконання всього проекту. Виходячи з положень методу PERT тривалість кожної операції має межі, які слідує зі статистичного розподілу. У даному методі використовують 3 оцінки розрахунку часу для кожної операції:

- оптимістичну (найкращу),
- очікувану (ймовірнісну),
- песимістичну (найгіршу).

Середньозважений час операції (роботи в проекті) розраховується за формулою:

$$\text{Очікуваний час} = \left[\frac{\text{Кращий Час} + (3 \times \text{Найбільш Вірогідний Час}) + (2 \times \text{Найгірший Час})}{6} \right] \quad (6)$$

Таблиця 1

Приклад оцінки часу розробки проекту.

Оцінка часу, який необхідно затратити на розробку проекту (в людиноднях)				
Етап	Найкращий випадок	Найбільш вірогідний випадок	Найгірший випадок	Очікуваний випадок
Формування потреб	3	4,5	6	4,75
Проектування	2,25	3	4	3,21
Реалізація	4	5	7	5,50
Тестування	3,75	5,5	9	6,38
Впровадження	2	4	7,5	4,83
Експлуатація та супровід	7	14	20	14,83
Разом	22	36	53,5	39,50

Використовуючи даний метод можна обчислити час, необхідний для розробки проекту в цілому.

Нижче приведено приклад таблиці індивідуальної оцінки часу, який необхідно витратити на створення програмної інформаційної системи "Записна книжка".

Застосування **штучних нейронних мереж для обчислення складності програмних систем**. Штучні нейронні мережі побудовані за принципом функціонування біологічних нейронних мереж. Системи, архітектура і принцип їх дії базується на аналогії з мозком живих істот. Ключовим елементом цих систем виступає штучний нейрон як імітаційна модель нервової клітини мозку. ШНМ представляють собою систему з'єднаних і взаємодіючих між собою простих штучних нейронів. З точки зору машинного навчання, нейронна мережа являє собою окремий випадок методів розпізнавання образів, дискримінантного аналізу, методів кластеризації тощо. З математичної точки зору, навчання нейронних мереж – це багатопараметрична задача нелінійної оптимізації. З точки зору кібернетики, нейронна мережа використовується в задачах адаптивного управління.

Нейронні мережі не програмується в звичайному розумінні цього слова, вони навчаються. Можливість навчання – одна з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. У процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними і вихідними, а також виконувати узагальнення. У разі успішного навчання мережа зможе повернути вірний результат на підставі даних, які були відсутні в навчальній вибірці, а також неповних та "зашумлених" даних [8].

Для побудови штучної нейронної мережі необхідно визначити, які дані потрібно подавати на вхід. За методами оцінки складності, описаними вище, на вхід будуть подаватись такі змінні:

- e – кількість дуг в графі програмного коду;
- v – кількість вершин в графі програмного коду;
- p – кількість компонентів зв'язності графу;
- cs – загальна структурна складність;
- dc – загальна інформаційна складність;
- g – кількість часу, затраченого на проект в найкращому випадку;
- rg – найбільш вірогідний час, що затратиться на проект;
- b – кількість часу, що затратиться у найгіршому випадку.

Враховуючи те, що важливим фактором при обчисленні складності системи та часу, що буде витрачений на її розробку важливу роль відіграє кількість розробників проекту та їх кваліфікація, до списку необхідно додати ще дві вхідні змінні:

- n – кількість розробників в команді;
- kv – кваліфікація розробників.

Перші три змінні необхідні для обчислення складності за методом Маккейба, наступні дві змінні використовуються для обчислення складності за методом інформаційно-структурної складності, і змінні g , rg , b необхідні для обчислення складності за методом PERT. Змінні n та kv впливатимуть на час розробки проекту. У випадку використання інформаційно-структурного методу та методу PERT необхідно застосовувати додаткове програмне забезпечення, яке буде обчислювати окремо загальну структурну складність програмної системи, загальну інформаційну складність системи, час, необхідний на розробку системи у найкращому, найбільш вірогідному та найгіршому випадках.

Описані вище методи обчислення складності систем між собою ніяк не зв'язані. Але, чим вища структурна складність системи, тим більший час буде витрачено на її проектування, а чим вища кваліфікація розробників – тим швидше вони зможуть спроектувати програмну систему. Використання штучної нейронної мережі дозволить об'єднати характеристики системи та виявити залежність залежності між ними. Отже, необхідно побудувати навчальну вибірку, для цього використаємо результати експериментів та оцінок складності різних програмних систем. У таблиці 2 наведено фрагмент результатів обчислень складностей програмних систем.

Важливим етапом розробки нейронної мережі є вибір архітектури. В даному випадку використано ШНМ архітектури одношарова лінійна нейронна мережа.

Лінійна функція найкращим чином відповідає сутності даної задачі. Її областю визначення являється діапазон $(-\infty; \infty)$. Це дозволяє, використовуючи кількісні характеристики системи на вході, отримувати характерні значення величин на виході, рівні їх фактичній сумі. Структура лінійної мережі повністю визначається кількістю її входів і виходів. Лінійність мережі робить її менш чутливою до неточностей масштабування даних. Недоліком є те, що лінійність нейронної мережі обмежує її можливість вирішувати нелінійну обернену задачу.

Модель нейрона. На рисунку 1 показано лінійний нейрон з двома входами. Він має структуру, схожу зі структурою перцептрона; єдиною відмінністю є те, що використовується лінійна функція активації (purelin в системі Matlab).

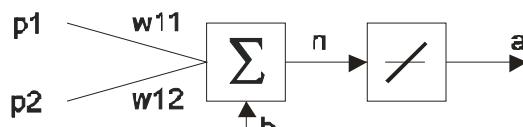


Рис. 1. Лінійний нейрон

Вагова матриця W в цьому випадку має тільки один рядок і вихід мережі визначається виразом:

$$a = \text{purelin}(n) = \text{purelin}(W_p + b) = \text{purelin}(W_{11}p_1 + W_{12}p_2 + b), \quad (7)$$

Таблиця 2

Фрагмент навчальної вибірки для ШНМ

	e	v	p	cs	dc	g	pr	b	n	kv	Z	SYSC	T
1	15	8	7	20	24,0	22,0	36,0	53,5	2	1	21	44,0	39,5
2	15	8	7	20	24,0	26,4	43,2	64,2	2	2	21	44,0	47,4
3	15	8	7	20	24,0	31,7	51,8	77,0	2	3	21	44,0	56,9
4	15	8	7	20	24,0	17,6	28,8	42,8	4	1	21	44,0	31,6
5	15	8	7	20	24,0	21,1	34,6	51,4	4	2	21	44,0	37,9
6	15	8	7	20	24,0	25,3	41,5	61,6	4	3	21	44,0	45,5
7	15	8	7	20	24,0	14,1	23,0	34,2	8	1	21	44,0	25,3
8	15	8	7	20	24,0	16,9	27,6	41,1	8	2	21	44,0	30,3
9	15	8	7	20	24,0	20,3	33,2	49,3	8	3	21	44,0	36,4
10	16	8	8	27	18,0	22,0	38,5	53,5	2	1	24	45,0	40,8
11	16	8	8	27	18,0	26,4	46,2	64,2	2	2	24	45,0	48,9
12	16	8	8	27	18,0	31,7	55,4	77,0	2	3	24	45,0	58,7
13	16	8	8	27	18,0	17,6	30,8	42,8	4	1	24	45,0	32,6
14	16	8	8	27	18,0	20,3	35,5	49,3	8	3	24	45,0	37,6
15	19	10	10	28	15,6	22,0	38,5	52,5	2	1	29	43,6	40,4
16	19	10	10	28	15,6	26,4	46,2	63,0	2	2	29	43,6	48,5
17	19	10	10	28	15,6	31,7	55,4	75,6	2	3	29	43,6	58,2
18	19	10	10	28	15,6	17,6	30,8	42,0	4	1	29	43,6	32,3

Подібно перцептрон, лінійна мережа задає в просторі входів роздільну лінію, на якій функція активації p рівна 0 (рис. 2).

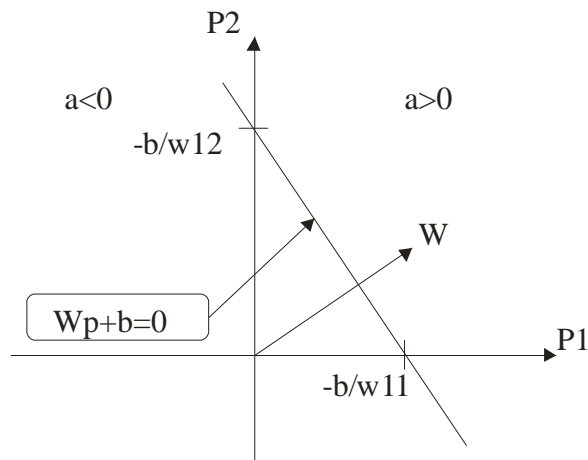


Рис. 2. Графік лінійної функції з роздільною лінією

Вектори входу, розміщені вище цієї лінії, відповідають додатнім значенням виходу, а розміщені нижче – від’ємним. Це означає, що лінійна мережа може використовуватись для розв’язання задач класифікації. Але така класифікація може бути використана тільки для класу лінійно віддільних об’єктів.

Архітектуру ШНМ для розв’язання задачі оцінки складності програмної системи показано на рис. 3.

Лінійна мережа представляється мережею без проміжних шарів, у вихідному шарі вона містить тільки елементи з лінійною функцією активації. Ваги відповідають елементам матриці, а пороги – компонентам вектора зміщення. Під час роботи мережа множить вектор входів на матрицю ваг, а далі до отриманого вектора додає вектор зміщення.

Вхідні та вихідні значення навчальної вибірки задамо, використовуючи утиліту nntool пакету MATLAB (рисунок 4).

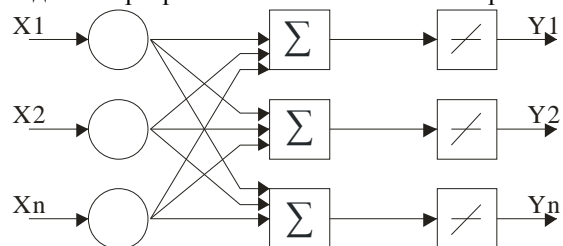


Рис. 3. Архітектура одношарової лінійної нейронної мережі

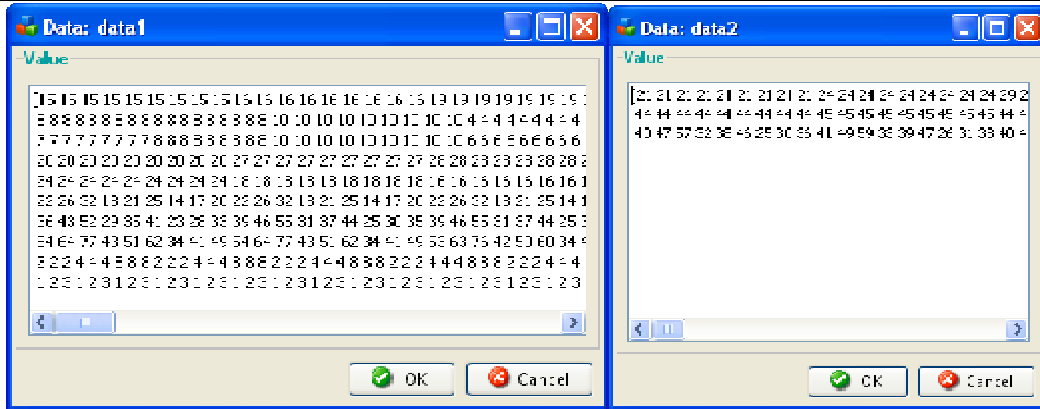


Рис. 4. Вхідні та вихідні значення навчальної вибірки

Створимо ШНМ LinearLayer у пакеті MATLAB та проведемо її навчання на основі вище заданої навчальної вибірки. Архітектуру мережі у пакеті MATLAB наведено на рис. 5.

Мережа складається з 10-и входів, 2-х виходів та одного шару.

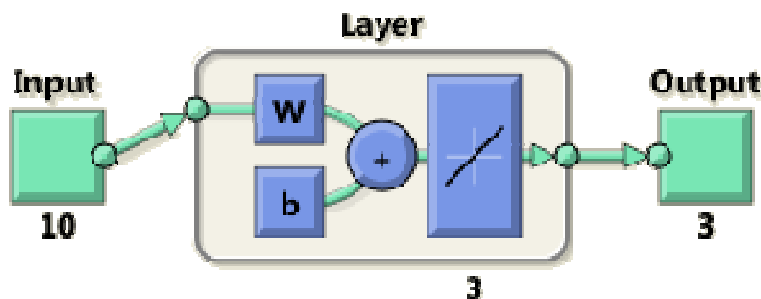


Рис. 5. Архітектура лінійної нейронної мережі

Для обчислення складності введемо у навчену ШНМ вибірку вхідних значень, що характеризують програмну систему: [12; 7; 6; 15; 21; 17; 25; 29; 4; 1].

У результаті отримаємо значення: цикломатичне число Маккейба $Z = 17$, загальна складність системи $SYSC = 36$, очікуваний час розробки програмної системи $T = 25$ людино-днів.

Висновок. Використання штучної нейронної мережі дало можливість на основі неповних та неточних даних здійснити оцінку складності системи.

Подальшим напрямком досліджень є розроблення методики верифікації результатів роботи ШНМ оцінки складності систем та порівняльний аналіз ефективності використання ШНМ інших архітектур для розв'язання задачі оцінки складності систем.

Література

1. Мещеряков Р.В. Критерий структурной сложности информационных систем / Мещеряков Роман Владимирович. – М., 2010. – 90 с.
2. K. Willcox, D. Allaire, J. Deyst, C. He, and G. Sondecker, Stochastic process decision methods for complex cyber-physical systems, Final Report Massachusetts Institute of Technology, July 2011, 28 p.
3. Федоров Р.В. Обеспечение табильного функционирования сетевых информационных систем с использованием аналитических и процедурных моднелей оценки сложности: дис. ... канд. технических наук: 05.25.05 / Федоров Роман Владимирович. – Т., 2005. – 219 с.
4. S.C. Evan, J.E. Hershe, G. Saulnie Kolmogorov Complexity Estimation and Analysis, Research Technical Report 2001CRD176. December, 2001. – 7 p.
5. Aivosto – Programming Tools for Software Developers. Електронний ресурс: <http://www.aivosto.com/project/help/pm-syscomplexity.html>.
6. Метрики [Електронний ресурс]. – Режим доступу: «<http://www.met-rix.narod.ru/page1.htm>».
7. Макконнелл С. Сколько стоит программный проект / Макконнелл С. – М.: «Русская Редакция», СПб.: Питер, 2007. – 297 с.: ил.
8. Штучна нейронна мережа [Електронний ресурс]. – Режим доступу: <http://uk.wikipedia.org/wiki/>.

Надійшла 19.5.2012 р.

Статтю представляє: д.т.н. Поморова О.В.