

## ДОСЛІДЖЕННЯ НЕЙРОМЕРЕЖНОЇ СКЛАДОВОЇ МЕТОДУ ПРОГНОЗУВАННЯ ХАРАКТЕРИСТИК ТА ОЦІНЮВАННЯ УСПІШНОСТІ РЕАЛІЗАЦІЇ ПРОГРАМНИХ ПРОЄКТІВ

У статті описано реалізацію, архітектуру і структуру нейромережної складової методу прогнозування характеристик та оцінювання успішності реалізації програмних проєктів. Також представлено аналіз результатів навчання та функціонування штучної нейронної мережі.

Ключові слова: програмний проєкт, специфікація вимог до програмного забезпечення, показники специфікації вимог, характеристики програмного проєкту, метод прогнозування характеристик та оцінювання успішності реалізації програмних проєктів (МПХОУР), штучна нейронна мережа (ШНМ).

A. V. KRASIY

Khmelnitsky National University

### RESEARCH OF ARTIFICIAL NEURAL NETWORK'S COMPONENT OF METHOD OF CHARACTERISTICS PREDICTION AND EVALUATING THE SUCCESS OF SOFTWARE PROJECTS IMPLEMENTATION

*Abstract – The article describes the implementation, architecture and structure of neural network's component of method of characteristics prediction and evaluating the success of software projects implementation. The analysis of the artificial neural network (ANN) training and functioning is represented. During the research the ANN was built in Matlab. The ANN was trained with the training sample of 6030 vectors by different training methods and was tested with the testing sample of 610 vectors. As the research results the ANN performance function (mse<sub>reg</sub>) and ANN training methods (OSS, SCG, RPROP) were selected on the basis of epochs number and training time. The received predicted relative evaluations of the software project characteristics, provided by ANN, provides to ground the prediction about success of concrete project version implementation and to compare the different project versions on the early stages of software lifecycle, when the developers have only SRS.*

*Keywords: software project, software requirements specification (SRS), SRS indicators, software project characteristics, method of evaluating the success of software project implementation (MESSPI).*

#### Вступ

В роботах [1–3] підтверджується факт, що причини майже всіх інцидентів та катастроф, пов'язаних з ПЗ, криються у специфікації вимог до програмного забезпечення (ПЗ), тобто переважна більшість аварій, пов'язаних із ПЗ, виникли через помилкові вимоги, а не через помилки кодування. У [2] описано результати експерименту, проведеного для підтвердження або відкидання гіпотези про те, що збої та помилки ПЗ, написаного різними розробниками за однією специфікацією, статистично незалежні. Під час експерименту декілька незалежних груп розробників писали свою версію ПЗ за однією специфікацією. В результаті такого експерименту було встановлено, що версії ПЗ, написані різними розробниками за однаковими вимогами, містили ряд спільних помилок, пов'язаних із помилками або неточностями вимог (специфікації). Аналіз великої кількості програмних проєктів, проведений у [4], також виявив, що головне місце виникнення помилок у ПЗ – це етап формулювання вимог (специфікація). Робота [5] основними причинами провалу програмних проєктів називає невірні уявлення керівництва та менеджерів проєкту щодо реального часу та коштів, які необхідні для забезпечення функційних вимог користувача, тобто знов-таки підтверджує, що більшість проблем ПЗ пов'язана із проблемами специфікації. При сучасному визначенні якості ПЗ [6] – якщо встановлені цілі проєкту не відповідають потребам користувачів, то програмний продукт неможливо буде визнати якісним, навіть якщо при його розробленні були використані сучасні технології та були задіяні найкваліфікованіші розробники. Тоді якість та успішність реалізації програмного проєкту залежать від специфікації вимог. Таке експериментальне свідчення безпосередньо призводить до необхідності поглиблення аналізу специфікації.

Тоді актуальною задачею є вміння оцінити можливу успішність реалізації програмного проєкту на основі характеристик програмного проєкту (тривалість проєкту, вартість, складність, кросплатформність, зручність використання, якість) [7], прогнозовані значення яких можна отримати шляхом аналізу показників специфікації вимог до програмного забезпечення. Аналіз зазначених характеристик [8, 9] дозволив зробити висновок, що існуючі математичні інструменти та методи, а також автоматизовані засоби їх визначення не придатні для оцінки їх значень на етапі формулювання вимог, оскільки орієнтовані на готовий програмний код, а не на наявну специфікацію вимог до ПЗ. Дослідження відомих методів та автоматизованих засобів аналізу специфікацій [8–12] показало, що всі вони спрямовані на контроль за реалізацією вимог, але жоден з них не визначає прогнозованих значень характеристик за специфікацією. Отже, існуючі методи і засоби аналізу специфікацій не прийнятні для кількісного оцінювання характеристик програмного проєкту на основі аналізу специфікацій.

Тому для прогнозування успішності реалізації програмного проєкту на основі аналізу специфікації автором було розроблено метод прогнозування характеристик та оцінювання успішності реалізації програмного проєкту на основі аналізу специфікації вимог [13, 14]. Нейромережний метод прогнозування характеристик та оцінювання успішності реалізації програмних проєктів (МПХОУР) пропонує

використовувати ШНМ, котра здійснює апроксимацію показників специфікації вимог до ПЗ, в результаті чого надає прогнозовані відносні оцінки характеристик програмного проекту. МПХОУР складається з наступних етапів: 1) нейромережне прогнозування характеристик програмного проекту на основі аналізу специфікації; 2) інтерпретація отриманих відносних значень характеристик програмного проекту; 3) оцінювання ступеня успішності реалізації програмного проекту на основі інтегративного показника проекту; 4) перевірка стабільності та припустимості компенсаційних впливів характеристик програмного проекту.

#### Визначення діапазонів значень вхідних векторів нейромережної складової МПХОУР

Аналіз структури специфікації вимог до ПЗ [15, 16] показав, що вимоги специфікації дозволяють сформувати множини показників, на основі яких замовник і розробник можуть отримати прогнозовані кількісні значення характеристик програмного проекту:  $R_1$  – множина показників розділу 1 специфікації вимог до ПЗ,  $R_2$  – множина показників розділу 2,  $R_3$  – множина показників розділу 3 специфікації вимог до ПЗ.

Для встановлення зв'язків між основними характеристиками програмного проекту та показниками специфікації було проаналізовано специфікації вимог до ПЗ. Для такого аналізу розглядалися 6 типів програмних проектів (Web-додатки, мобільні додатки, програми електронного навчання, пакети прикладних програм (статистичні, бухгалтерські), автоматизовані системи, інформаційні системи). За кожним типом програмних проектів досліджувались 30–50 задач різного рівня складності. Для кожної задачі було обрано по 1-3 специфікації, запропоновані різними розробниками для вирішення однієї задачі, а також по 1–3 готових програми, написаних за аналізованими специфікаціями (розглядалися курсові проекти з навчальної дисципліни «Технологія проектування програмних систем» та дипломні роботи і проекти студентів; проекти студентського гуртка «SOFTWARE» Хмельницького національного університету; специфікації та програми софтверних компаній м. Хмельницький; специфікації та програми, знайдені у мережі Інтернет). Таким чином, було проаналізовано 200 задач, 400 специфікацій та 400 готових програм на предмет того, які показники відрізнялись у обраних специфікаціях і як залежно від цього змінювались характеристики готових програм, а також на предмет значень показників специфікацій.

Таблиця 1

#### Залежність характеристик програмного проекту від показників специфікації вимог

	Вартість	Тривалість	Складність	Зручність використання	Кросплатформність	Якість
<i>Розділ 1 специфікації вимог до ПЗ:</i>						
прогнозований термін виконання	+	+	+			+
кількість виконавців	+		+			+
розмір цільової аудиторії (прогнозована кількість користувачів)			+	+	+	
кількість компонентів ПЗ	+	+	+			+
прогнозований розмір (прогнозована LOC-оцінка) ПЗ	+	+	+			+
<i>Розділ 2 специфікації вимог до ПЗ:</i>						
вартість використовуваних у розробленні операційних систем	+		+			+
вартість використовуваних у розробленні баз даних	+		+			+
вартість використовуваних у розробленні компіляторів	+		+			+
вартість засобів розроблення	+		+		+	
розмір документації користувачів (кількість сторінок)	+	+		+		
вартість документації користувачів	+	+		+		
<i>Розділ 3 специфікації вимог до ПЗ:</i>						
кількість функційних вимог	+	+	+			+
вартість функційних вимог	+		+			+
кількість використовуваних алгоритмів		+	+			+
середня прогнозована вартість помилки	+	+				+
вартість інтерфейсів користувача	+			+	+	
кількість програмних (міжмодульних) інтерфейсів	+	+	+			+
вартість міжмодульних інтерфейсів	+		+			+
кількість апаратних інтерфейсів	+	+	+		+	
вартість апаратних інтерфейсів	+		+		+	
кількість інтерфейсів зв'язку та комунікацій	+	+	+		+	
вартість інтерфейсів зв'язку та комунікацій	+		+		+	
кількість нефункційних вимог	+	+	+	+	+	
вартість нефункційних вимог	+		+	+	+	

Проведений аналіз 400 специфікацій вимог до ПЗ та 400 готових програм дав можливість зробити наступні висновки щодо залежності основних характеристик програмного проекту від показників специфікації вимог для всіх розглядуваних типів програмних проектів – таблиця 1, а також визначити діапазони значень показників специфікації вимог до ПЗ, які прийняті як обмеження для вхідних даних розробленого методу прогнозування характеристик та оцінювання успішності реалізації програмних проектів – таблиця 2.

Таблиця 2

**Діапазон значень показників специфікації, які впливають на основні характеристики ПЗ**

Розділ специфікації вимог	Показник специфікації вимог	Діапазон значень показника
1	2	3
Розділ 1	прогнозований термін виконання $T_v$	1-24 місяці
	кількість виконавців $Q_v$	1-10 чоловік
	прогнозована кількість користувачів $S_a$	1-1000 чоловік
	кількість компонентів ПЗ $Q_{cs}$	1-50 компонентів
	прогнозований розмір (ЛОС-оцінка) $S_c$	50-50000 рядків коду
Розділ 2	вартість використовуваних операційних систем $C_{os}$	0-1250 доларів США
	вартість використовуваних баз даних $C_{db}$	0-1250 доларів США
	вартість використовуваних компіляторів $C_c$	0-1250 доларів США
	вартість засобів розроблення $C_{dt}$	0-1250 доларів США
	кількість сторінок документації користувачів $C_{ud}$	1-50 сторінок
	вартість документації користувачів $S_{ud}$	50-2500 доларів США
Розділ 3	кількість функційних вимог $Q_{fr}$	5-300 вимог
	вартість функційних вимог $C_{fr}$	50-4750 доларів США
	кількість використовуваних алгоритмів $Q_a$	1-500 алгоритмів
	середня прогнозована вартість помилки $C_b$	10-960 доларів США
	вартість інтерфейсів користувача $C_{ui}$	50-3000 доларів США
	кількість програмних (міжмодульних) інтерфейсів $Q_{mi}$	50-2450 інтерфейсів
	вартість міжмодульних інтерфейсів $C_{mi}$	25-2500 доларів США
	кількість апаратних інтерфейсів $Q_{ai}$	5-100 інтерфейсів
	вартість апаратних інтерфейсів $C_{ai}$	25-1500 доларів США
	кількість інтерфейсів зв'язку та комунікацій $Q_{ci}$	5-125 інтерфейсів
	вартість інтерфейсів зв'язку та комунікацій $C_{ci}$	25-1750 доларів США
	кількість нефункційних вимог $Q_{nfr}$	1-9 вимог
	вартість нефункційних вимог $C_{nfr}$	50-4000 доларів США

**Нейромережна складова методу прогнозування характеристик та оцінювання успішності реалізації програмних проектів**

Для оцінювання характеристик ПЗ на основі опрацювання показників специфікації вимог до ПЗ (перший етап МПХОУР) використаємо штучну нейронну мережу (ШНМ), яка здійснює апроксимацію показників специфікації та надає прогнозовані відносні оцінки характеристик ПЗ, розроблюваного за аналізованою специфікацією.

Вхідними даними для ШНМ є множини показників специфікації вимог до ПЗ:  $R1 = \{T_v, Q_v, S_a, Q_{cs}, S_c\}$  – множина показників розділу 1 специфікації вимог до ПЗ,  $R2 = \{C_{os}, C_{db}, C_c, C_{dt}, C_{ud}, S_{ud}\}$  – множина показників розділу 2,  $R3 = \{Q_{fr}, C_{fr}, Q_a, C_b, C_{ui}, Q_{mi}, C_{mi}, Q_{ai}, C_{ai}, Q_{ci}, C_{ci}, Q_{nfr}, C_{nfr}\}$  – множина показників розділу 3 специфікації вимог до ПЗ. Результатом роботи ШНМ є множина прогнозованих відносних оцінок характеристик програмного проекту:  $SCH = \{C_s, D_{sp}, C_p, C_x, U_b, Q_s\}$ , де  $C_s$  – вартість програмного проекту,  $D_{sp}$  – тривалість,  $C_p$  – кросплатформність,  $C_x$  – складність,  $U_b$  – зручність використання,  $Q_s$  – якість.

Отже, ШНМ має 5 входів  $x'$ , 6 входів  $x''$  та 13 входів  $x$ . На входи  $x'$  подаються показники 1-го розділу специфікації вимог, на входи  $x''$  – показники 2-го розділу специфікації вимог, на входи  $x$  – показники 3-го розділу специфікації вимог до ПЗ. На вхід  $x'_i$  ( $i=1..5$ ) подається значення  $i$ -го елементу множини  $R1$ , на вхід  $x''_k$  ( $k=1..6$ ) подається значення  $k$ -го елементу множини  $R2$ , на вхід  $x_j$  ( $j=1..13$ )

подається  $j$ -й елемент множини  $R3$ .

ШНМ опрацьовує набори вхідних векторів та видає 6 вихідних значень з діапазону  $[0..1]$ :

$Cs$  – відносна оцінка вартості програмного проекту, де 0 – недостатньо даних для прогнозування вартості (МПХОУР в такому випадку не працює), 0.08 – вартість проекту висока, 1 – вартість проекту низька;

$Dsp$  – відносна оцінка тривалості, де 0 – недостатньо даних для прогнозування тривалості (МПХОУР в такому випадку не працює), 0.08 – тривалість проекту висока, 1 – тривалість проекту низька;

$Cx$  – відносна оцінка складності, де 0 – недостатньо даних для прогнозування складності (МПХОУР в такому випадку не працює), 0.08 – складність проекту висока, 1 – складність проекту низька;

$Cr$  – відносна оцінка кросплатформності, де 0 – недостатньо даних для прогнозування кросплатформності (МПХОУР в такому випадку не працює), 0.08 – кросплатформність проекту низька, 1 – кросплатформність висока;

$Ub$  – відносна оцінка зручності використання, де 0 – недостатньо даних для прогнозування зручності використання (МПХОУР в такому випадку не працює), 0.08 – зручність використання проекту низька, 1 – зручність використання висока;

$Qs$  – відносна оцінка якості, де 0 – недостатньо даних для прогнозування якості (МПХОУР в такому випадку не працює), 0.08 – якість проекту низька, 1 – якість проекту висока.

На основі аналізу шести одержаних характеристик (згідно етапів 2-4 МПХОУР) робиться висновок про успішність реалізації програмного проекту.

Вибір архітектури нейромережі для реалізації нейромережної складової МПХОУР описано у [17]. Для задачі прогнозування характеристик програмного проекту на основі аналізу показників специфікації вимог використано багаточаровий перцептрон. Архітектура нейромережної складової МПХОУР представлена на рис. 1.

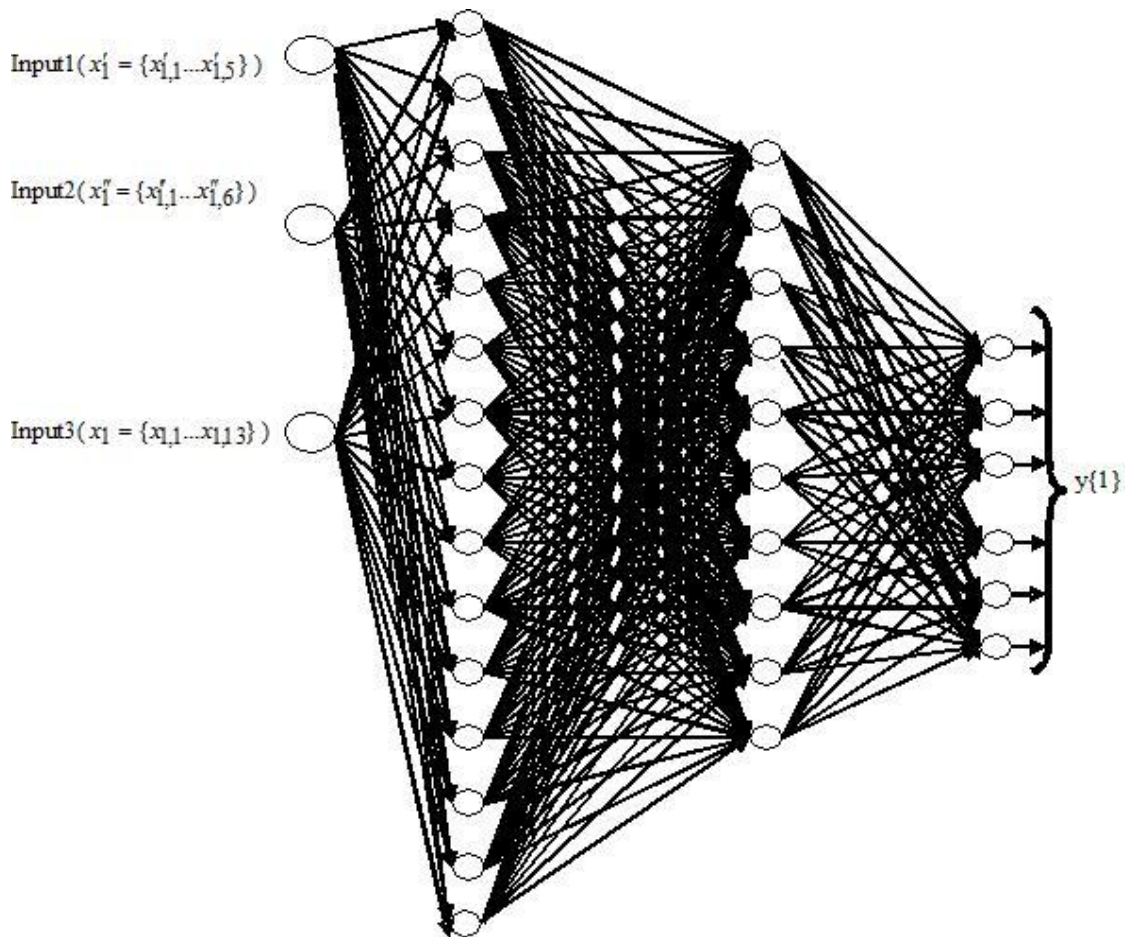


Рис. 1. Архітектура нейромережної складової МПХОУР

Вхідний вектор Input1 складається з 5 елементів (показники розділу 1 специфікації вимог); вхідний вектор Input2 – з 6 елементів (показники розділу 2 специфікації вимог); вхідний вектор Input3 – з 13 елементів (показники розділу 3 специфікації вимог). Таким чином, ШНМ має 24 нейрони вхідного шару. Також ШНМ має 15 нейронів другого, прихованого (апроксимуючого) шару та 10 нейронів третього, прихованого (коригуючого) шару. Вихідний вектор  $Y$  повинен складатися з 6 елементів: вартість,

тривалість, складність, зручність використання, кросплатформність, якість програмного проекту, тому ШНМ має 6 нейронів четвертого, вихідного шару.

Діапазони значень вхідних векторів ШНМ збігаються з діапазонами вхідних даних МПХОУР та вказані у таблиці 2. На вхід подається -1, якщо даний показник не визначено у аналізованій специфікації вимог до ПЗ.

ШНМ реалізовано у пакеті Matlab. Для створення шаблону ШНМ використовувалась функція `network`. Було визначено кількість вхідних векторів (`net.numInputs=3`), шарів (`net.numLayers=4`), елементів кожного вхідного вектора ШНМ: `net.inputs{1}.size=5`; `net.inputs{2}.size=6`; `net.inputs{3}.size=13`. Матриця зв'язності для зміщень: `net.biasConnect=[1;1;1;1]`. Матриці зв'язності для входів, шарів, виходів та цілей задано наступним чином: `net.inputConnect = [1 1 1; 0 0 0; 0 0 0; 0 0 0]`; `net.layerConnect = [0 0 0 0; 1 0 0 0; 0 1 0 0; 0 0 1 0]`; `net.outputConnect = [0 0 0 1]`; `net.targetConnect = [0 0 0 1]`. Діапазони значень вхідних векторів: `net.inputs{1}.range = [-1 24; -1 10; -1 1000; -1 50; -1 50000]`; `net.inputs{2}.range = [-1 1250; -1 1250; -1 1250; -1 1250; -1 50; -1 2500]`; `net.inputs{3}.range = [-1 300; -1 4750; -1 500; -1 960; -1 3000; -1 2450; -1 2500; -1 100; -1 1500; -1 125; -1 1750; -1 9; -1 4000]`. Кількість нейронів у шарах ШНМ: `net.layers{1}.size=24`; `net.layers{2}.size=15`; `net.layers{3}.size=10`; `net.layers{4}.size=6`. В якості функції ініціалізації для кожного шару ШНМ використано функцію Нгуена-Відрю (`initnw`). Для 1-3-го шарів у якості активаційної функції обрано гіперболічний тангенс (`tansig`), для 4-го (вихідного) шару активаційною є порогова лінійна функція (`purelin`).

Оператор `gensim(net)` дає змогу одержати модель в пакеті Simulink (рис.2-7).

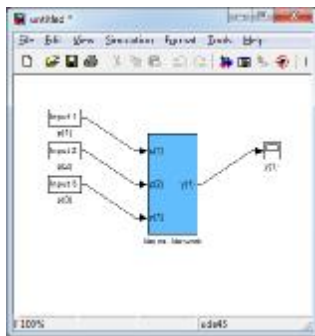


Рис. 2. Архітектура ШНМ в пакеті Simulink

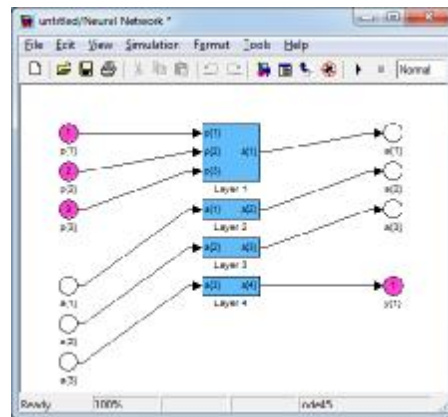


Рис. 3. Структурна схема шарів ШНМ в пакеті Simulink

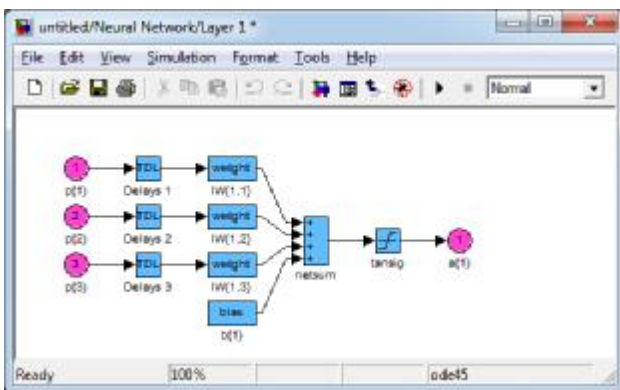


Рис. 4. Структурна схема першого шару ШНМ

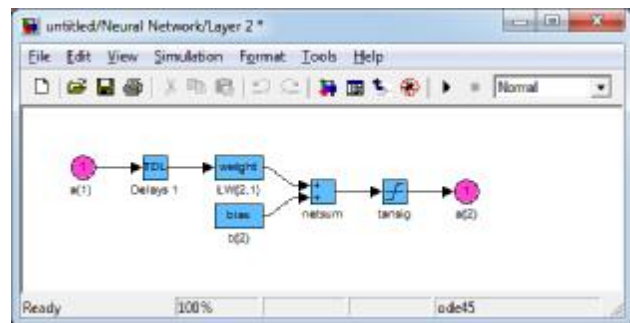


Рис. 5. Структурна схема другого шару ШНМ

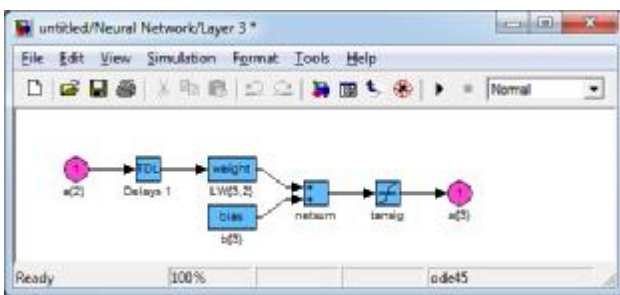


Рис. 6. Структурна схема третього шару ШНМ

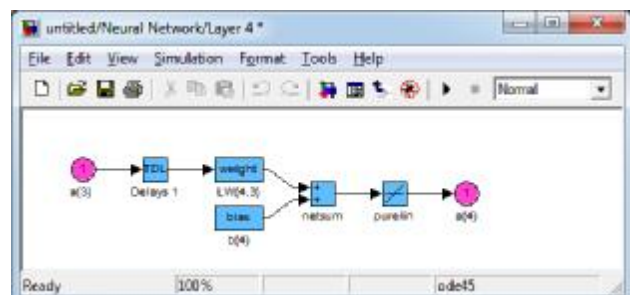


Рис. 7. Структурна схема четвертого шару ШНМ

**Навчання ШНМ та аналіз отриманих результатів**

Для навчання одержаної ШНМ послідовність навчальних векторів (навчальну вибірку) задано у вигляді:

$c = \{ [1; -1; -1; -1; -1] [6; 3; 220; 13; 10900] [\dots]; - \text{навчальні вектори для входу Input1 } (x_1^1);$

$[0; -1; -1; -1; -1; -1] [260; 324; 216; 270; 11; 690] [\dots]; - \text{навчальні вектори для входу Input2 } (x_1^2);$

$[5; -1; -1; -1; -1; -1; -1; -1; -1; -1; -1; -1; -1; -1; -1];$

$[83; 1075; 108; 216; 705; 680; 563; 29; 415; 30; 400; 3; 910] [\dots]; \} - \text{навчальні вектори для входу Input3 } (x_1^3).$

Цільовий вектор визначено як:

$m = \{ [1; 1; 1; 0; 0; 1] [0.789; 0.782; 0.792; 0.790; 0.795; 0.792] [\dots] \}.$

Для розрахунку максимального об'єму навчальної вибірки використаємо формулу комбінаторики, яка дозволяє обчислити кількість сполучень без повторень:  $C(m, n) = \frac{m!}{n!(m-n)!} = \frac{24!}{6!(24-6)!} = 134596$ , де  $m$

– кількість вхідних нейронів (входів) ШНМ ( $m=24$ );  $n$  – кількість вихідних нейронів (виходів) ШНМ ( $n=6$ ). Отже, при наявності більше 130 тисяч навчальних векторів можна застосувати статистичні методи.

ШНМ використовується в умовах неповноти вхідної інформації. Для розрахунку необхідного об'єму навчальної вибірки для ШНМ, яку потрібно навчити з похибкою порядку  $10^{-1}$  використаємо формулу

[18]:  $N > \frac{h \cdot g}{e_0} = \frac{24 \cdot 25}{10^{-1}} = 6000$ , де  $g$  – кількість вхідних нейронів (входів) ШНМ ( $g=24$ );  $h$  – кількість

нейронів прихованого шару ШНМ ( $h=15+10=25$ ),  $e_0$  – допустима похибка навчання ( $e_0=10^{-1}$ ). Отже, 6000 векторів навчальної вибірки достатньо для того, щоб навчити ШНМ розпізнавати можливі ситуації з заданою точністю.

Для вибору алгоритму навчання було проведено навчання ШНМ навчальною вибіркою з 6030 векторів за різними алгоритмами (таблиця 3).

Таблиця 3

**Аналіз результатів навчання ШНМ за різними алгоритмами навчання**

Функція навчання	Критерій якості навчання	Похибка навчання	Час навчання, секунди	Кількість епох
trainгда (Алгоритм градієнтного спуску з вибором параметра швидкості налагодження)	msereg	0.0903451	19	107
	mse	0.100291	22	105
	mae	0.255729	112	2100
trainoss (Однокроковий алгоритм методу січної)	msereg	0.0903451	16	2
	mse	0.100291	14	2
trainrp (Пороговий алгоритм оберненого поширення)	msereg	0.0903451	15	37
	mse	0.100291	18	35
	mae	0.251225	116	2100
trainlm (Алгоритм Левенберга-Марквардта)	msereg	0.0903451	170	4
	mse	0.100291	115	2
	mae	0.264611	124	2
trainscg (Алгоритм шкалованих зв'язаних градієнтів)	msereg	0.0903451	15	1
	mse	0.100291	16	1
	mae	0.395961	230	2100

Дослідження показали, що найменшу похибку навчання одержано при комбінованому критерії якості навчання (msereg). В ході дослідження виявлено, що похибка навчання суттєво не відрізняється для різних алгоритмів навчання при використанні однакового критерію якості навчання, тому вибір алгоритму навчання обумовлювався показниками "кількість епох" та "час навчання", за якими кращими є алгоритми навчання OSS, SCG та RPROP.

Тестування ШНМ здійснювалось на основі тестової вибірки з 610 векторів. Процес навчання і тестування за різними алгоритмами навчання відображено на рис. 8 – 11. На рисунках нижня крива відображає графік навчання, а верхня крива відображає графік тестування ШНМ.

Аналіз графіків навчання і тестування ШНМ дозволив зробити висновок, що мережа навчилася з високою достовірністю та точністю.

**Експерименти з прогнозування характеристик програмного проекту**

Проведемо експерименти з практичного використання нейромережної моделі прогнозування характеристик програмного проекту на основі аналізу специфікації. Для цього розглянемо чотири альтернативні програмні проекти, розроблені різними групами програмістів з різних софтверних компаній Хмельницького для вирішення однієї задачі (таблиця 4).

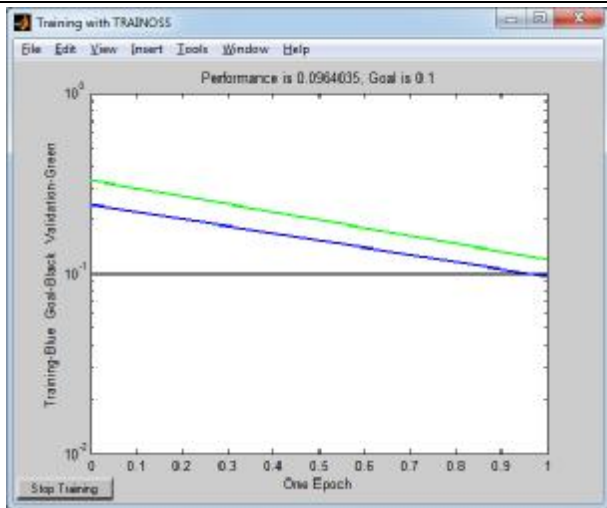


Рис. 8. Однокроковий алгоритм методу січної (OSS)

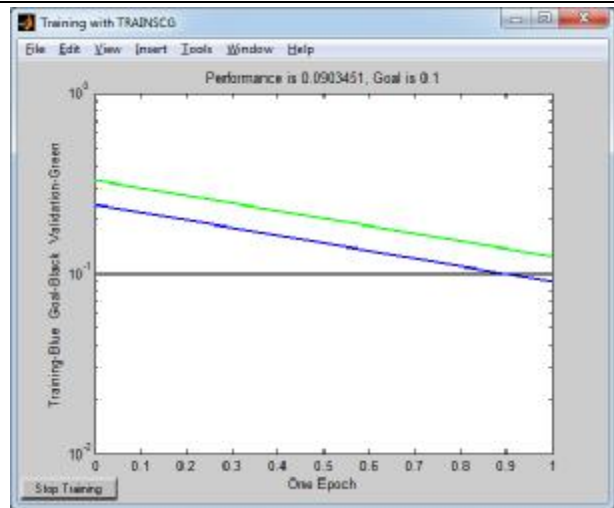


Рис. 9. Алгоритм шкалованих зв'язаних градієнтів (SCG)

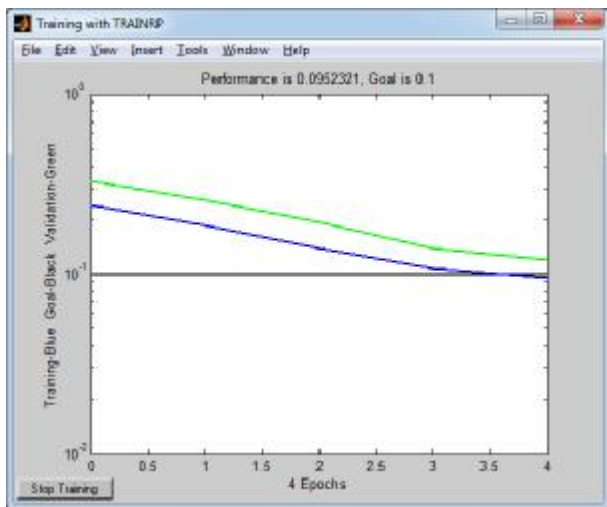


Рис. 10. Пороговий алгоритм оберненого поширення (Rprop)

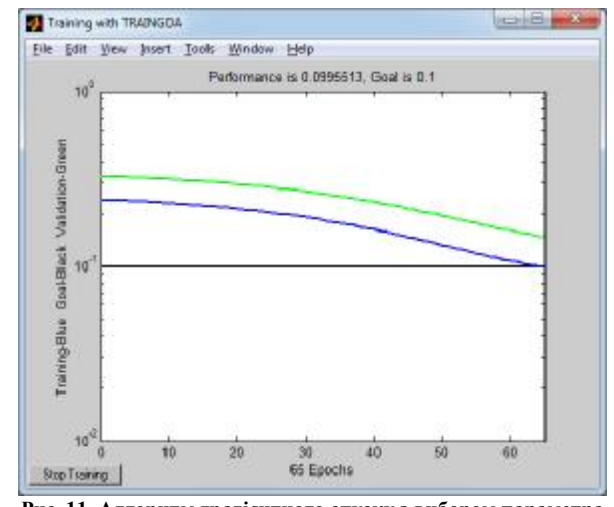


Рис. 11. Алгоритм градієнтного спуску з вибором параметра швидкості налагодження (GDA)

Таблиця 4

**Опрацювання нейромережною складовою МПХОУР показників специфікацій вимог до ПЗ**

№ п/п	Значення показників розділу 1 специфікації (Input1 ШНМ)	Значення показників розділу 2 специфікації (Input2 ШНМ)	Значення показників розділу 3 специфікації (Input3 ШНМ)	Результати ШНМ (відносні характеристики проекту)
1	[13;-1;-1;-1;-1]	[624;648;648;648;25;1329]	[160;2510;258;505;1687;1310;1315;53;805;65;925;5;2110]	[0.52;0.521;0.521;0.52;0.52;0.52]
2	[24;10;1000;50;50000]	[0;-1;-1;-1;-1;-1]	[300;4750;500;960;3000;2450;2500;53;805;65;925;5;2110]	[0.389;0.082;0.39;0.097;0.093;0.389]
3	[-1;1;-1;-1;-1]	[624;648;648;648;25;1329]	[160;2510;258;505;1687;1310;1315;53;805;65;925;5;2110]	[0.68;0.522;0.681;0.52;0.52;0.68]
4	[24;10;1000;50;50000]	[1250;1250;1250;1250;50;2500]	[-1;-1;-1;-1;-1;-1;-1;5;-1;-1;-1;-1]	[0.388;0.387;0.389;0.09;0.393;0.083]

Проаналізуємо отримані результати: найкращі характеристики має проект №3, на другому місці – проект №1, на третьому місці – проект №4, на четвертому місці – проект №2. Отже, проекти №3 та №1 імовірно будуть успішними, а проекти №2 і №4 – проблемними, а то й провальними.

Очевидно, що на основі отриманих з ШНМ відносних значень основних характеристик проекту як розробнику, так і замовнику складно комплексно оцінити успішність реалізації програмного проекту, оскільки складно вірно інтерпретувати одержані відносні значення характеристик, тому й було запропоновано наступні етапи МПХОУР, які дозволяють інтерпретувати отримані відносні значення характеристик та прогнозувати майбутню успішність реалізації програмного проекту.

**Висновки**

Необхідність поглиблення аналізу специфікації, залежність успішності реалізації програмного

проекту від специфікації вимог, неможливість успішної реалізації програмних проектів з невдалими вимогами та специфікаціями, актуальність та важливість вміння оцінити успішність реалізації програмного проекту на основі специфікації доведені у багатьох сучасних наукових дослідженнях. Раніше розроблений автором нейромережний метод прогнозування характеристик та оцінювання успішності реалізації програмних проектів (МПХОУР) пропонує використовувати ШНМ, котра здійснює апроксимацію показників специфікації вимог до ПЗ, в результаті чого надає прогнозовані відносні оцінки характеристик програмного проекту.

Дана стаття присвячена реалізації та дослідженню нейромережної складової МПХОУР. В ході дослідження було побудовано ШНМ в пакеті Matlab, після чого нейромережа навчалась вибіркою з 6030 векторів за різними алгоритмами навчання та тестувалась вибіркою з 610 векторів. В результаті дослідження обрано: критерій якості навчання ШНМ - *mse<sub>reg</sub>*; алгоритми навчання ШНМ за кількістю епох та часом навчання - *OSS*, *SCG*, *RPROP*.

Отримані прогнозовані відносні оцінки характеристик програмного проекту, надані ШНМ, дозволяють обґрунтувати прогноз щодо успішності реалізації конкретної версії проекту та порівняти між собою різні версії проектів на ранніх етапах життєвого циклу, коли наявна лише специфікація вимог до ПЗ.

### Література

1. Levenson N.G. Systemic factors in software-related spacecraft accidents / Levenson N.G. // AIAA Space 2001 Conference and Exposition, pp.1–11.
2. Levenson N.G. Software challenges in achieving space safety / Levenson N.G. // Journal of the British Interplanetary Society. – Vol. 62, July/August 2009, pp. 265–272.
3. Ishimatsu T. Hazard analysis of complex spacecraft using systems-theoretic process analysis / Ishimatsu T., Leveson N.G., Thomas J.P., Fleming C.H., Katahira M., Miyamoto Yu., Ujiie R., Nakao H., Hoshino N. // Journal of Spacecraft and Rockets – Vol. 51, No. 2 (2014), pp. 509–522.
4. Jones C. The economics of software quality / Jones C., Bonsignour O. – Boston : Pearson Education, 2012. – 588 p.
5. E.Yourdon. Death March: The complete software developer's guide to surviving "Mission impossible" projects (2nd edition) / Yourdon E.– Prentice Hall, 2003.
6. Bourque P. Guide to the software engineering body of knowledge (SWEBOOK). Version 3.0 / Bourque P., Fairley R.E. – A project of the IEEE Computer Society, 2014 – 335 p.
7. Maedche A. Software for people: fundamentals, trends and best practices (Management for professionals) / Maedche A., Botzenhardt A., Neer L. – Berlin: Springer-Verlag Berlin Heidelberg, 2012. – 293 p.
8. Красій А.В. Моделивання процесу прогнозування характеристик програмного забезпечення на основі аналізу специфікацій / А.В. Красій // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – Луцьк : Луцький національний технічний університет, 2014. – С. 66–76.
9. Fenton N. Software metrics: A rigorous approach (3rd edition) / Fenton N. – CRC Press, 2014 – 597 p.
10. Chen A. Visual models for software requirements / Chen A., Beatty J. – W.: MS Press, 2012. – 444 p.
11. Fatwanto A. Software requirements specification analysis using natural language processing technique / Fatwanto A. // International Conference on Quality in Research – 2013, pp.105–110.
12. Rehman T. Analysis of requirement engineering processes, tools/techniques and methodologies / Rehman T., Khan M.N.A., Riaz N. // I.J. Information Technology and Computer Science – 2013, pp.40–48.
13. Красій А.В. Нейромережний метод прогнозування характеристик та оцінювання успішності реалізації програмного проекту / А.В. Красій // Вісник Хмельницького національного університету – Хмельницький: ХНУ, 2015. – № 3. – С. 132–140
14. Hovorushchenko T. Method of Evaluating the Success of Software Project Implementation Based on Analysis of Specification Using Neuronet Information Technologies / Hovorushchenko T., Krasiy A. // Proceedings of the 11th International Conference on ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer – CEUR-WS, vol.1356, pp.100–107.
15. IEEE 1031-2011: IEEE Guide for the Functional Specification. URL: <https://standards.ieee.org/findstds/standard/1031-2011.html>
16. IEEE 29148-2011 – Systems and software engineering. Life cycle processes. Requirements engineering
17. Hovorushchenko T. Realization of Neural Network Model of Prediction of the Software Project Characteristics for Evaluating the Success of Its Implementation Based on Analysis of Specifications / Hovorushchenko T., Krasiy A. // Proceedings of the 2015 IEEE 8-th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS-2015) – USA, NJ 08855-1331: IEEE Operations Center, 2015 – vol.1, p. 348–353 (IEEE Catalog Number: CFP15803-PRT)
18. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / Уоссермен Ф. – М. : Мир, 1992. – 240 с.