

## РОЗДІЛ 3

МЕТОД ТА АЛГОРИТМИ ІДЕНТИФІКАЦІЇ ПРИХОВАНИХ ПОМИЛОК  
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ НЕЙРОМЕРЕЖНИХ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

3.1. Метод ідентифікації прихованих помилок програмного забезпечення  
на основі нейромережних інформаційних технологій

Початковими даними для реалізації повторного тестування є інформація про типи помилок (множина  $\mathbf{P} = \{\mathbf{p}_k \mid k = 1..n\}$ ), виявлених під час основного тестування, та методи (множина  $\mathbf{M} = \{\mathbf{m}_k \mid k = 1..n\}$ ) і операції, що були застосовані для їх виявлення (множина  $\mathbf{O} = \{\mathbf{o}_k \mid k = 1..n\}$ ). Ця інформація береться із звітів про результати основного тестування, які надаються тестувальником у вигляді журналу “Метод тестування – Операція, яка виконується під час тестування – Результат операції (тип виявленої помилки)”. Оскільки основне тестування здійснює тестувальник, то на результати тестування можливий вплив суб’єктивного та людського факторів (врахування «почерку» тестувальника), що може як позитивно, так і негативно впливати на ефективність повторного тестування. Саме для зменшення зазначеного суб’єктивного фактора враховуються не тільки кількість і типи виявлених помилок, а й методи та операції тестування.

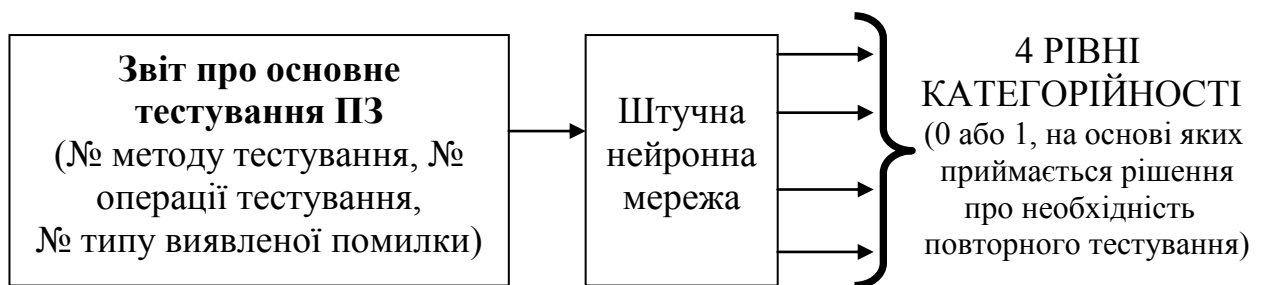


Рис. 3.1. Принцип застосування ШНМ

Вхідні дані для реалізації повторного тестування подаються у вигляді

$$\text{матриці } \mathbf{VD} = \begin{vmatrix} \mathbf{m}_1 & \mathbf{o}_1 & \mathbf{p}_1 \\ \cdot & \cdot & \cdot \\ \mathbf{m}_i & \mathbf{o}_i & \mathbf{p}_i \\ \cdot & \cdot & \cdot \\ \mathbf{m}_n & \mathbf{o}_n & \mathbf{p}_n \end{vmatrix}, \text{ де } \mathbf{m}_i, \mathbf{o}_i, \mathbf{p}_i - \text{ елементи множин } \mathbf{M}, \mathbf{O}, \mathbf{P}$$

відповідно. Кожен елемент матриці  $\mathbf{VD}$ , представлений у вигляді тексту, піддається перетворенню для представлення його у кількісному вигляді.

$$\text{Використовуючи матриці } \mathbf{MN} = \begin{vmatrix} \mathbf{1} & \mathbf{m}_1 \\ \cdot & \cdot \\ \mathbf{i} & \mathbf{m}_i \\ \cdot & \cdot \\ \mathbf{s} & \mathbf{m}_s \end{vmatrix}, \text{ де } \mathbf{m}_i - \text{ елемент множини } \mathbf{M},$$

$$\mathbf{ON} = \begin{vmatrix} \mathbf{1} & \mathbf{o}_1 \\ \cdot & \cdot \\ \mathbf{i} & \mathbf{o}_i \\ \cdot & \cdot \\ \mathbf{v} & \mathbf{o}_v \end{vmatrix}, \text{ де } \mathbf{o}_i - \text{ елемент множини } \mathbf{O}, \mathbf{PN} = \begin{vmatrix} \mathbf{1} & \mathbf{p}_1 \\ \cdot & \cdot \\ \mathbf{i} & \mathbf{p}_i \\ \cdot & \cdot \\ \mathbf{z} & \mathbf{p}_z \end{vmatrix}, \text{ де } \mathbf{p}_i - \text{ елемент}$$

множини  $\mathbf{P}$ , які представляють собою присвоєння номерів методам тестування, операціям тестування та типам виявлених помилок відповідно,  $[\mathbf{i}, \mathbf{1}]$ -й елемент матриці  $\mathbf{VD}$ , представлений у вигляді тексту, піддається перетворенню для представлення його у кількісній формі. Відбувається пошук елемента в другому стовпці матриці  $\mathbf{MN}$ , одержується порядковий номер  $j$  рядка елемента.  $[\mathbf{j}, \mathbf{1}]$ -й елемент матриці  $\mathbf{MN}$  заноситься в  $[\mathbf{i}, \mathbf{1}]$ -й

$$\text{елемент матриці } \mathbf{VDM} = \begin{vmatrix} \mathbf{mn}_1 & \mathbf{on}_1 & \mathbf{pn}_1 \\ \cdot & \cdot & \cdot \\ \mathbf{mn}_i & \mathbf{on}_i & \mathbf{pn}_i \\ \cdot & \cdot & \cdot \\ \mathbf{mn}_n & \mathbf{on}_n & \mathbf{pn}_n \end{vmatrix}, \text{ де } \mathbf{mn}_i, \mathbf{on}_i, \mathbf{pn}_i - \text{ кількісне}$$

представлення значень елементів множин  $\mathbf{M}, \mathbf{O}, \mathbf{P}$  відповідно.

Далі піддається перетворенню в кількісне представлення  $[i,2]$ -й елемент матриці  $VD$ . Відбувається пошук елемента в другому стовпці матриці  $ON$ , одержується порядковий номер  $j$  рядка елемента.  $[j,1]$ -й елемент матриці  $ON$  заноситься в  $[i,2]$ -й елемент матриці  $VDM$ .

Останнім піддається перетворенню в кількісне представлення  $[i,3]$ -й елемент матриці  $VD$ . Відбувається пошук елемента в другому стовпці матриці  $PN$ , одержується порядковий номер  $j$  рядка елемента.  $[j,1]$ -й елемент матриці  $PN$  заноситься в  $[i,3]$ -й елемент матриці  $VDM$ .

Після одержання кількісного представлення значень кожного елемента матриці  $VD$  формується набір вхідних векторів для ШНМ. На вхід  $q_i$  подається 1, якщо використовувався відповідний для  $i$ -го рівня категорійності метод основного тестування (дана відповідність наведена в

матриці  $NMRK = \begin{vmatrix} nm_1 & rk_1 \\ nm_2 & rk_2 \\ nm_3 & rk_3 \\ nm_4 & rk_4 \end{vmatrix}$ , де  $nm$  - номер методу основного тестування,

$rk$  - рівень категорійності). На вхід  $x'_i$  подається номер  $i$ -ї операції основного тестування  $on_i$  ( $[i,2]$ -й елемент матриці  $VDM$ ), на вхід  $x_i$  подається номер  $i$ -го типу виявленої під час основного тестування помилки  $pn_i$  ( $[i,3]$ -й елемент матриці  $VDM$ ). На всі інші входи подається «0».

ШНМ опрацьовує набір вхідних векторів згідно методу вирішення задачі повторного тестування та видає матрицю вихідних векторів

$VV = \begin{vmatrix} rk_{11} & rk_{12} & rk_{13} & rk_{14} \\ \cdot & \cdot & \cdot & \cdot \\ rk_{i1} & rk_{i2} & rk_{i3} & rk_{i4} \\ \cdot & \cdot & \cdot & \cdot \\ rk_{n1} & rk_{n2} & rk_{n3} & rk_{n4} \end{vmatrix}$ , де  $i$ -й рядок містить  $i$ -й вихідний вектор,

елемент  $rk_{i1}$  містить значення «нуль» або «одиниця» для рівня категорійності з номером 1  $i$ -го вихідного вектора, елемент  $rk_{i2}$  містить

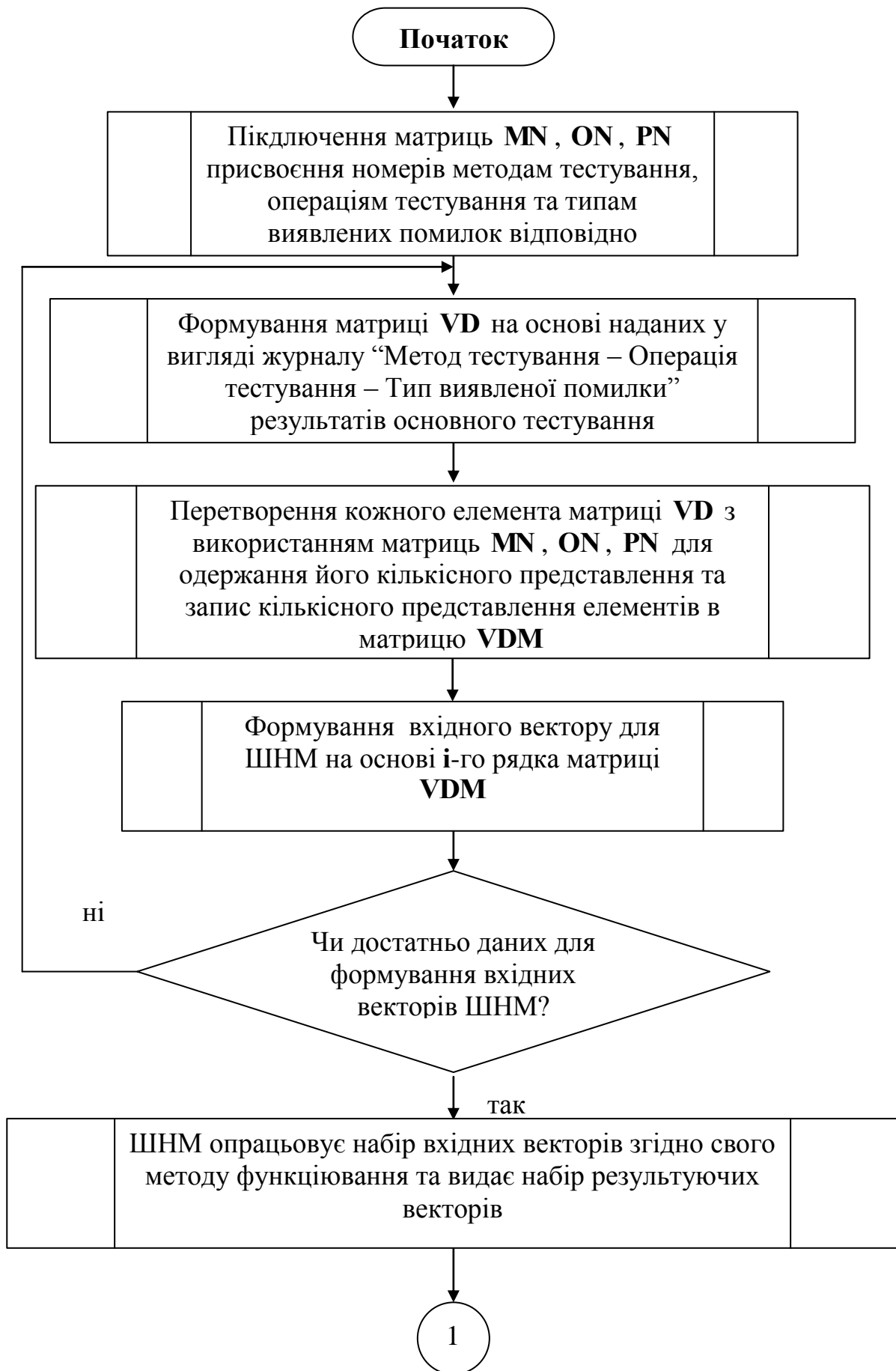
значення «нуль» або «одиниця» для рівня категорійності з номером 2  $i$ -го вихідного вектора, елемент  $rk_{i3}$  містить значення «нуль» або «одиниця» для рівня категорійності з номером 3  $i$ -го вихідного вектора,  $rk_{i4}$  містить значення «нуль» або «одиниця» для рівня категорійності з номером 4  $i$ -го вихідного вектора. Вихідні вектори потрібно піддати перетворенню для одержання результатів у лінгвістичній формі. Для цього використовується матриця присвоєння рівнів категорійності типам прихованих помилок

$$RK = \begin{vmatrix} 1 & rk_1 \\ 2 & rk_2 \\ 3 & rk_3 \\ 4 & rk_4 \end{vmatrix}, \text{ де } rk_i - \text{тип прихованих помилок. Перетворенню з}$$

кількісної в лінгвістичну форму піддається окремо кожен вихідний вектор, тобто окремо кожен рядок матриці  $VV$ . Для перетворення  $i$ -го рядка в ньому відбувається пошук «одиниці», запам'ятовується номер стовпця  $h$  та знаходиться  $[h,2]$ -й елемент матриці  $RK$ . Знайдений елемент  $rk_h$  є лінгвістичним представленням одержаного результату. Цей елемент заноситься в множину результатів  $R = \{rk_k \mid k = 1..n\}$ . На основі аналізу складу множини  $R$  робиться висновок про необхідність та тип повторного тестування.

### 3.2. Алгоритм ідентифікації прихованих помилок програмного забезпечення

На основі представленого методу ідентифікації прихованих помилок на базі НІТ побудовано алгоритм ідентифікації прихованих помилок програмного забезпечення (рис. 3.2).



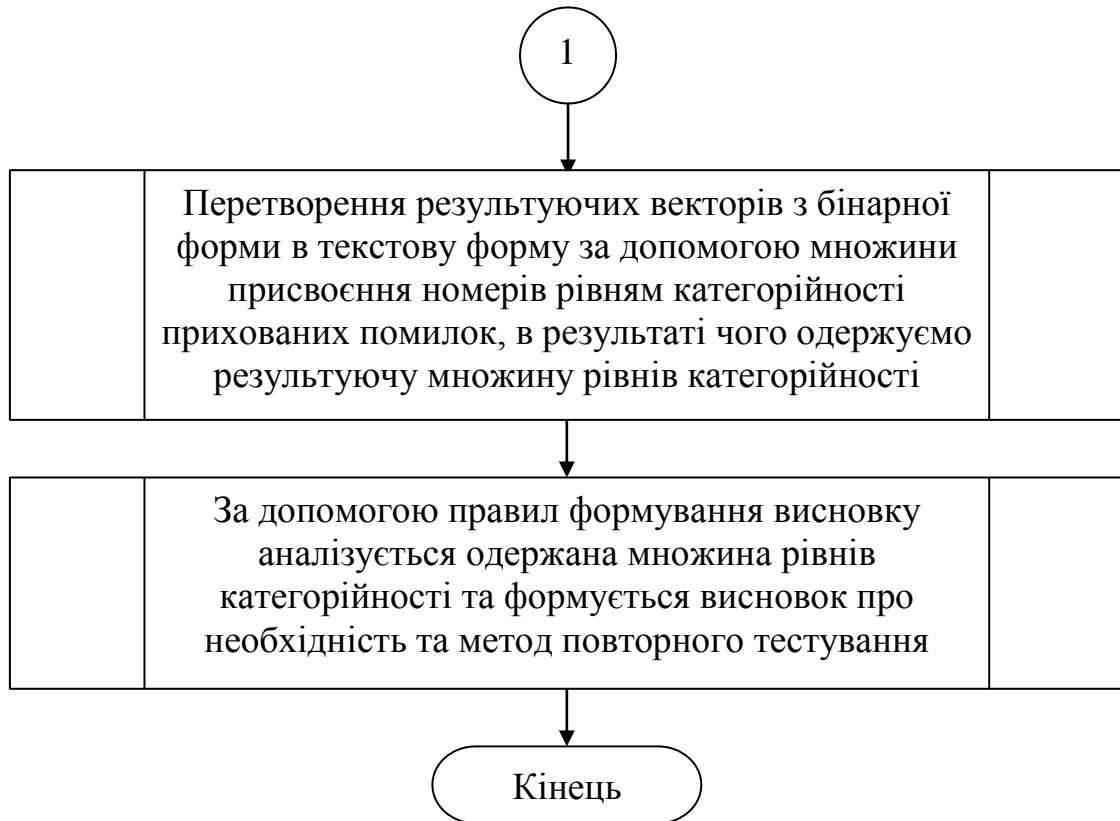


Рис. 3.2. Алгоритм повторного тестування прикладного програмного забезпечення

Отже, на першому кроці відбувається підключення матриць присвоєння номерів методам тестування, операціям тестування та типам виявлених помилок. Далі формується матриця, в якій відображаються дані вхідного звіту “Метод тестування – Операція тестування – Тип виявленої помилки” в лінгвістичній формі. На наступному кроці відбувається кодування кожного методу, операції та типа виявленої помилки шляхом заміни назви на номер, тобто відбувається формування матриці, яка відображає вхідний звіт в кількісному вигляді. На основі кожного рядка одержаної матриці відбувається формування вхідного вектора для ШНМ. Далі відбувається перевірка вхідних даних на повноту та достовірність. Якщо вхідні дані не є повними або достовірними, то відбувається повернення на етап формування матриці, в якій відображаються дані вхідного звіту, для доповнення цієї матриці необхідними достовірними даними. Якщо ж дані успішно проходять перевірку на повноту і достовірність, то ШНМ опрацьовує набір вхідних

векторів згідно свого алгоритму функціонування та видає набір результуючих векторів. На наступному кроці відбувається перетворення результуючих векторів з бінарної форми в текстову форму за допомогою множини присвоєння номерів рівням категорійності прихованих помилок, в результаті чого формується множина рівнів категорійності в лінгвістичному вигляді. Згідно методу і алгоритму формування висновку про необхідність повторного тестування та відповідний метод повторного тестування і правил формування висновку аналізується одержана множина рівнів категорійності та формується висновок про необхідність та рекомендовані методи повторного тестування.

### 3.3. Формування бази знань

База знань складається з двох частин: розділу даних і розділу правил.

*Розділ даних* містить:

- 1) таблицю присвоєння номерів методам основного тестування (таблиця 1 додатку А). Деякі методи об'єднані по два під одним номером, оскільки ними найчастіше виявляються помилки одного рівня категорійності (ця відповідність наведена в таблиці 8 додатку А);
- 2) таблицю присвоєння номерів операціям основного тестування (таблиця 2 додатку А);
- 3) таблицю присвоєння номерів типам виявлених під час основного тестування помилок (таблиця 3 додатку А);
- 4) таблицю присвоєння номерів рівням категорійності прихованих помилок (таблиця 4 додатку А);
- 5) таблицю кількісного представлення вхідних даних (таблиця 5 додатку А);
- 6) таблицю текстового представлення результуючих векторів штучної нейронної мережі (таблиця 6 додатку А);

- 7) таблицю відповідності методу основного тестування, операцій основного тестування і типів виявлених під час основного тестування помилок (таблиця 7 додатку А);
- 8) таблицю відповідності між номером методів тестування ПЗ та рівнем категорійності прихованих помилок ПЗ (таблиця 8 додатку А);
- 9) таблицю відповідності між операціями тестування ПЗ та рівнем категорійності прихованих помилок (таблиця 9 додатку А).

*Розділ правил* містить правила для формування висновку про необхідність повторного тестування та правила для формування висновку про метод повторного тестування.

Для опису правил формування висновку про необхідність повторного тестування введемо поріг  $a_i \in A$  ( $A = \{a_h \mid h = 1..4\}$ , де  $a_h$  - поріг допустимої кількості помилок і важливості помилок різних типів одного виду, при перевищенні якого необхідно здійснювати повторне тестування з метою виявлення прихованих помилок цього виду,  $h$  - кількість типів порогів, що змінюється від 1 до  $s$ ,  $s$  - кількість рівнів категорійності прихованих помилок (тобто  $s = 4$ )), при перевищенні якого необхідно здійснювати повторне тестування з метою виявлення прихованих помилок цього виду.

*Правило 1.* Якщо відношення сумарного значення помилок  $i$ -го рівня категорійності до загальної кількості виявлених під час основного тестування помилок перевищує поріг  $a_i$ , то повторне тестування здійснювати необхідно.

*Правило 2.* Якщо висновок про необхідність повторного тестування прийнятий по перевищенню порога кількістю помилок  $j$ -го рівня категорійності, то повторне тестування слід здійснювати методами і операціями, які виявляють помилки типу  $j$  (ця відповідність вказана в таблиці 7 бази знань).

3.4. Метод формування висновку про необхідність повторного тестування

На основі правил прийняття рішення про повторне тестування розроблено метод формування висновку про необхідність та рекомендований(і) метод(и) повторного тестування.

На основі результатів роботи ШНМ формуємо множину  $\mathbf{K} = \{k_i \mid i = 1..4\}$ , де  $k_i$  - сумарні значення помилок кожного з рівнів категорійності. На основі множини  $\mathbf{K}$  та множини рівнів категорійності прихованих помилок  $\mathbf{R} = \{rk_k \mid k = 1..n\}$  формуємо множину  $\mathbf{KR} = \{kr_i \mid i = 1..4\}$  відношень  $kr_i = \frac{k_i}{n}$ .

Визначаємо порядок перегляду і застосування правил на основі отриманих результатів (метод пошуку). Процедура вибору зводиться до визначення напрямку пошуку і способу його здійснення. В даному дослідженні використовується метод здійснення пошуку в ширину в прямому напрямку [117], тобто спочатку аналізуються всі правила формування висновку про необхідність повторного тестування і за відомими фактами (елементи вектора  $\mathbf{KR}$ ) відшукуються заключення, яке з цих фактів слідує, і лише потім, якщо буде сформовано висновок про необхідність повторного тестування, аналізу піддаватимуться правила формування висновку про метод повторного тестування і за відомими фактами відшукається висновок, який з цих фактів слідує.

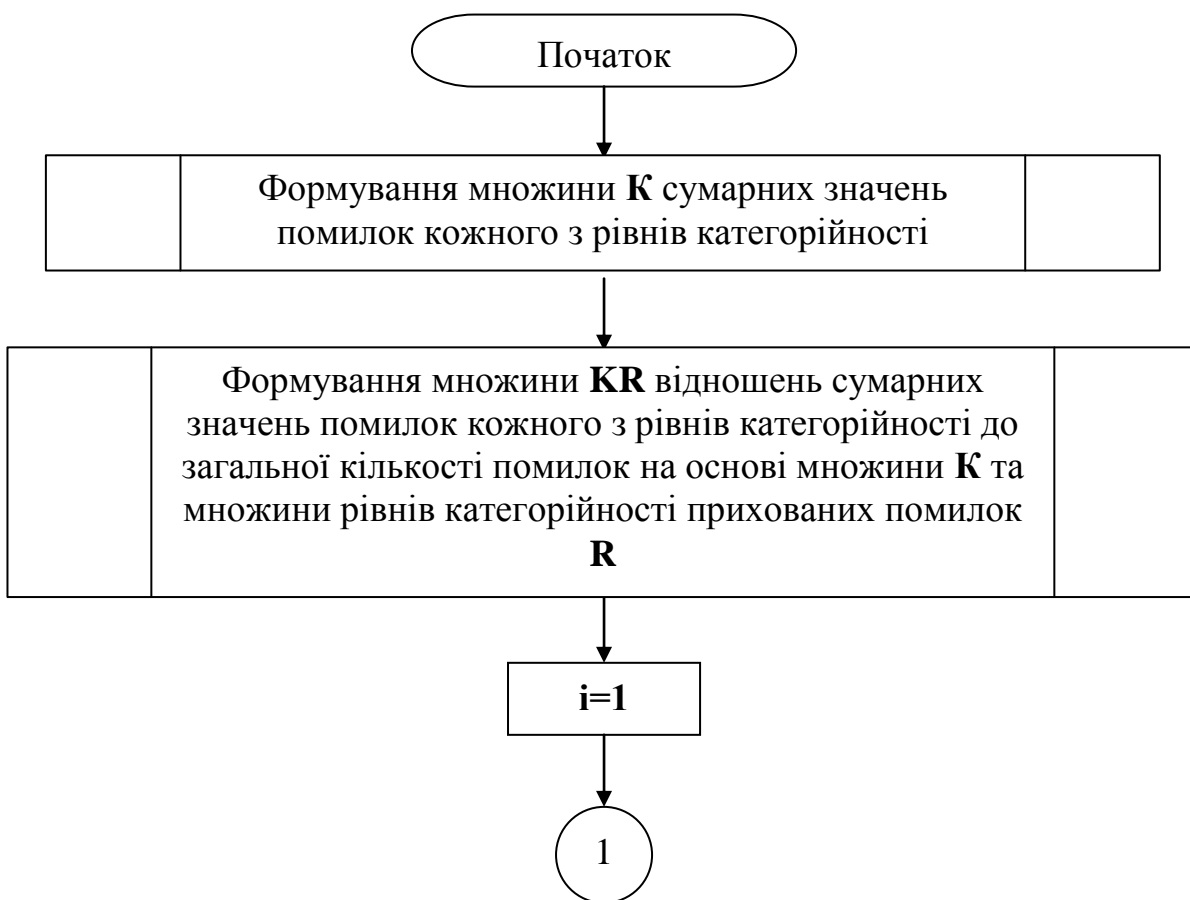
Аналіз правил формування висновку про необхідність повторного тестування відбувається наступним чином. В множині правил типу “якщо-то”  $\mathbf{PR} = \{pr_h \mid h = 1..m\}$  шукаємо правило для кожного з елементів множини  $\mathbf{KR}$ . Якщо значення елемента множини задовольняє умові лівої частини правила, то це правило заноситься в множину обраних правил  $\mathbf{OPR} = \{opr_y \mid y = 1..g\}$ . Критерій вибору єдиного правила для множини  $\mathbf{OPR}$  не актуальний, тому що у всіх правил однакова права частина (результат), в якій робиться висновок про необхідність повторного тестування. Отже, якщо

кількість відібраних правил  $g > 0$ , то робиться висновок про те, що повторне тестування необхідне.

Після формування висновку про необхідність повторного тестування аналізуємо, поріг кількості помилок якого рівня категорійності був перевищений. Тоді, за правилом 2, повторне тестування рекомендовано проводити методами, які виявляють помилки саме цього рівня категорійності.

### 3.5. Алгоритм формування висновку про необхідність повторного тестування

На основі представленого методу формування висновку про необхідність повторного тестування побудовано алгоритм формування висновку про необхідність повторного тестування (рис. 3.3).



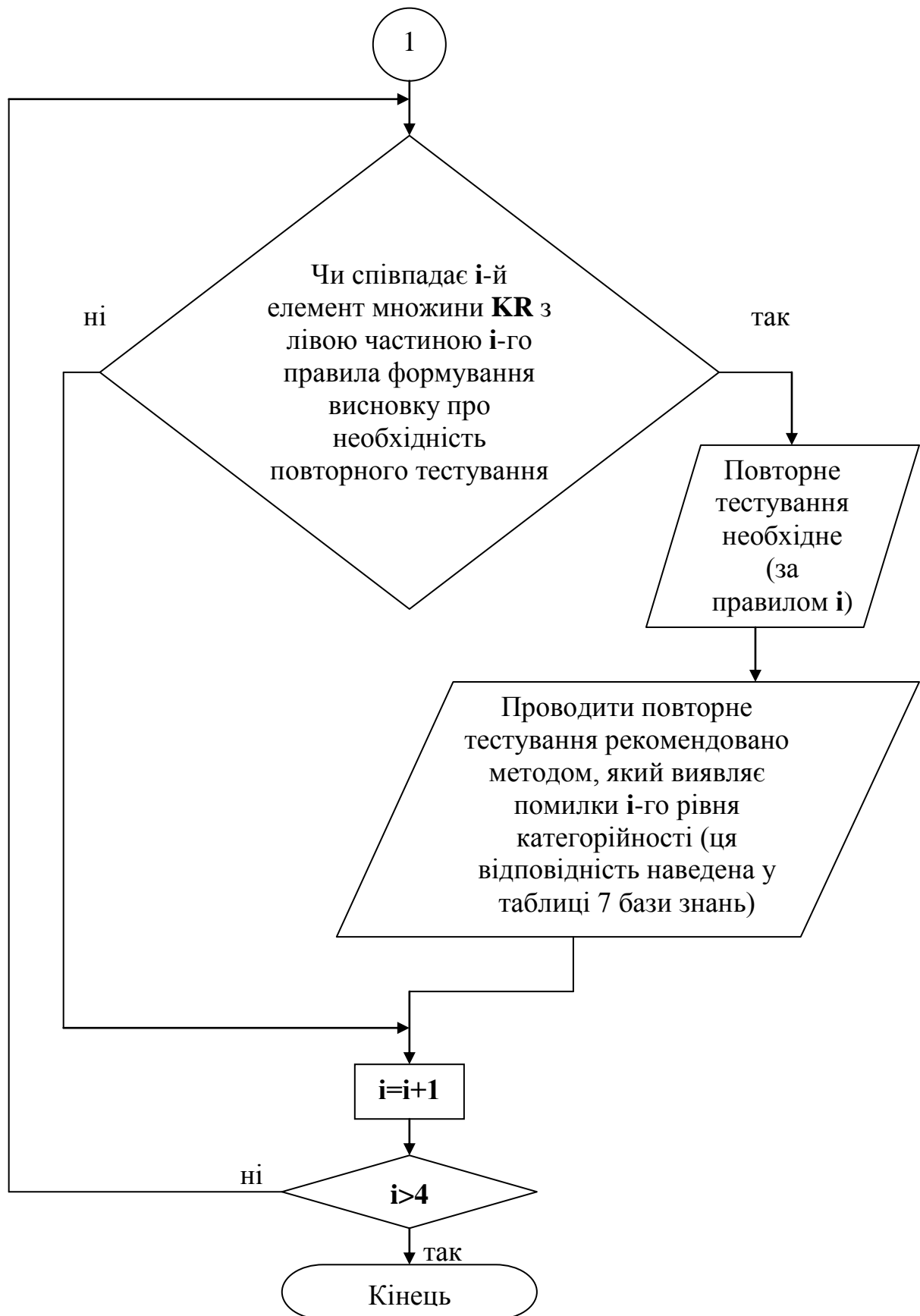


Рис.3.3. Алгоритм формування висновку про необхідність повторного тестування

На першому кроці формується множина сумарних значень помилок кожного з рівнів категорійності. На наступному кроці формується множина **KR** відношень сумарних значень помилок кожного з рівнів категорійності до загальної кількості помилок. Далі відбувається перевірка на співпадання *i*-го елемента множини **KR** з лівою частиною *i*-го правила формування висновку про необхідність повторного тестування. Якщо елемент множини **KR** співпадає з лівою частиною правила, то формується висновок, що повторне тестування необхідне і рекомендується проводити його методом, який виявляє помилки *i*-го рівня категорійності (ця відповідність наведена у таблиці 7 бази знань). Якщо ж елемент множини **KR** не співпадає з лівою частиною правила, то повторне тестування не потрібне.

Розроблені методи та алгоритми дають можливість при наявності звіту про основне тестування одержати висновок про необхідність повторного тестування і рекомендації щодо методу повторного тестування.

### 3.6. Модель ШНМ в пакеті Matlab

При імітаційному моделюванні в пакеті Matlab ШНМ, описаної в розділі 2, не використовуватимемо вихідні функціонали ефекторного шару  $Y_1, \dots, Y_i, \dots, Y_m$ , оскільки в даному випадку вони не є інформативними. Їх інформативність проявляється лише при інтеграції розробленої моделі у вже існуючі системи тестування. Активними залишаються лише виходи  $Y_1, Y_2, \dots, Y_h, \dots, Y_m$ , тобто моделюється наступна ШНМ (рис.3.4).

Структурна схема імітаційної моделі ШНМ в пакеті Matlab представлена на рис. 3.5.

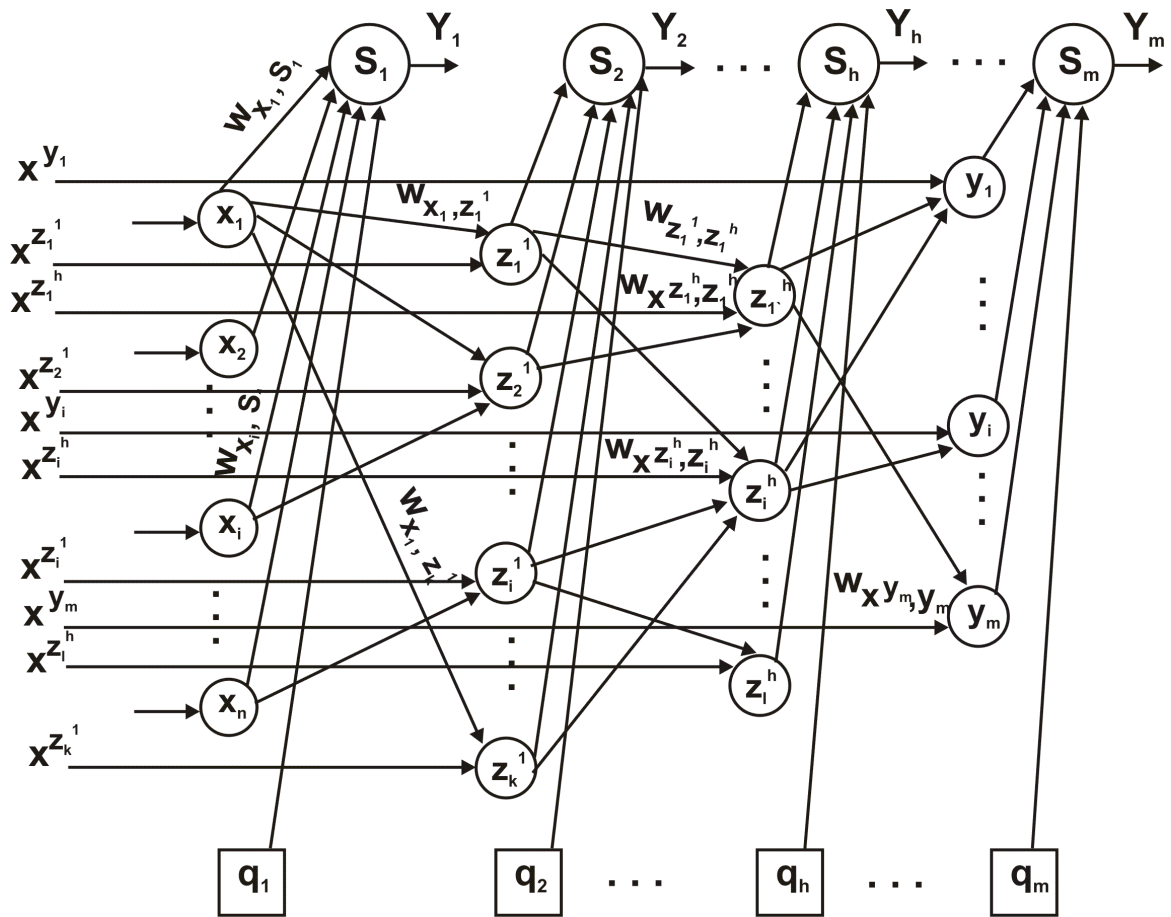


Рис. 3.4. Штучна нейронна мережа, яка моделюється в пакеті Matlab

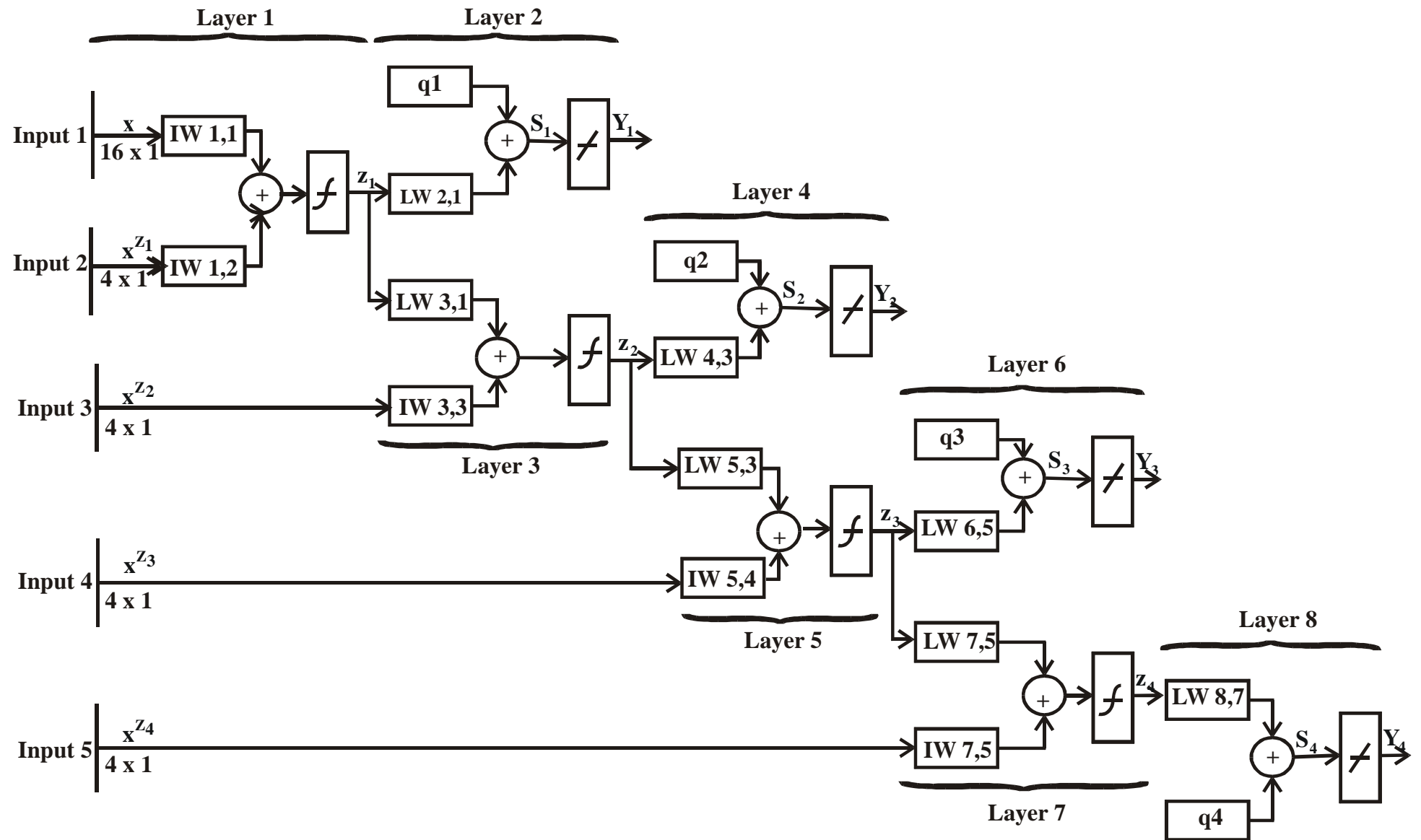


Рис. 3.5. Структурна схема імітаційної моделі ШНМ в пакеті Matlab.

На кожен з входів  $q_1 - q_4$  потрібно подати “одиницю”, тому що тестування здійснюється одним з методів тестування, які утворюються внаслідок об’єднання двох методів тестування під одним номером, що відображено в таблиці 1 бази знань (таблиця 1 додатку А).

За статистикою [58] тестувальник тестує програму не більш як чотири операціями одного методу тестування (відповідність між методами тестування та їх операціями наведена в таблиці 7 бази знань (додатку А)), тому на кожен з входів Input2 – Input5 можна подати не більше чотирьох номерів операцій тестування. На входи Input2 ( $x^{Z1}$ ), Input3 ( $x^{Z2}$ ), Input4 ( $x^{Z3}$ ), Input5 ( $x^{Z4}$ ) подаються номери операцій основного тестування ПЗ. Причому на вхід Input2 подаються номери таких операцій, при виконанні яких найчастіше виявляються помилки першого рівня категорійності (незначні); на вхід Input3 подаються номери таких операцій, виконання яких призводить до виявлення помилок другого рівня категорійності (помірні); на вхід Input4 подаються номери таких операцій, виконання яких дозволяє виявити помилки третього рівня категорійності (серйозні); на вхід Input5 подаються номери таких операцій, які дозволяють виявити помилки четвертого рівня категорійності (катастрофічні). Дана відповідність наведена в таблиці 9 бази знань повторного тестування (таблиця 9 додатку А).

На вхід Input1 ( $x$ ) подаються номери результатів операцій основного тестування ПЗ, тобто номери типів виявлених під час основного тестування помилок. Оскільки за статистикою [58] в програмі буває максимум 14-15 помилок, то на даний вхід можна подати не більше 16 типів помилок.

Значення входів ШНМ лежать в діапазонах, вказаних у таблиці 3.1 (Нумерація методів тестування ПЗ наведена в таблиці 1 бази знань системи повторного тестування (додаток А). Нумерація операцій тестування ПЗ наведена в таблиці 2 бази знань системи повторного тестування (додаток А)).

Діапазони входів ШНМ

Вхід	Діапазон
$q_1$	0..1
$q_2$	0..1
$q_3$	0..1
$q_4$	0..1
Input1 (x)	0..22
Input2 ( $x^{z1}$ )	0, 20..32, 50, 51
Input3 ( $x^{z2}$ )	0, 41..46, 52, 53
Input4 ( $x^{z3}$ )	0, 10..19, 33..40
Input5 ( $x^{z4}$ )	0, 1..9, 47..49

В першому шарі ШНМ (Layer1) значення входів  $x$  і  $x^{z1}$  множимо на відповідні вагові коефіцієнти і додаємо, від одержаної суми беремо активізаційну функцію гіперболічного тангенса ( $f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$ ) [8, 114, 116, 118], в результаті чого одержимо значення  $z_1$ . Це значення передаємо на другий (Layer2) та третій (Layer3) шари ШНМ. В другому шарі до значення  $z_1$ , помноженого на відповідний ваговий коефіцієнт, додаємо значення добутку значення  $q_1$  і його вагового коефіцієнта, від одержаної суми  $S_1$  беремо лінійну активізаційну функцію [8, 114, 116, 118], в результаті чого одержимо значення функції виходу  $Y_1$ . В третьому шарі значення  $z_1$ , помножене на ваговий коефіцієнт, сумується зі значенням добутку значення входу  $x^{z2}$  і його вагового коефіцієнта, від одержаної суми береться активізаційна функція гіперболічного тангенса, в результаті чого одержуємо значення  $z_2$ . Це значення передаємо на четвертий (Layer4) і п'ятий (Layer5) шари ШНМ, які функціонують аналогічно другому (Layer2) і третьому (Layer3) шарам відповідно. З четвертого (Layer4)

шару одержуємо значення функції виходу  $Y_2$ . З п'ятого шару (Layer5) одержуємо значення  $z_3$ , що передається на шостий (Layer6) і сьомий (Layer7) шари ШНМ, які функціонують аналогічно другому (Layer2) і третьому (Layer3) шарам відповідно. З шостого шару (Layer6) одержуємо значення функції виходу  $Y_3$ . З сьомого шару (Layer7) одержуємо значення  $z_4$ , що передається на восьмий шар (Layer8), який функціонує аналогічно другому шару (Layer2). З восьмого шару (Layer8) одержуємо значення функції виходу  $Y_4$ .

Кожен з виходів  $Y_i$  відповідає за один з чотирьох рівнів категорійності ( $Y_1$  - перший рівень категорійності,  $Y_2$  - другий рівень категорійності,  $Y_3$  - третій рівень категорійності,  $Y_4$  - четвертий рівень категорійності ) і приймає значення „1”, якщо штучною нейронною мережею спрогнозовано наявність в програмі помилок  $i$ -го рівня категорійності, в протилежному випадку значення виходу  $Y_i$  становить „0”.

Ініціалізація ваг та зміщень здійснюється за методом Нгуена-Відроу (initnw) [118, 119, 120].

### 3.7. Програмна реалізація моделі ШНМ в пакеті Matlab

Визначимо кількість входів, внутрішніх та елементів кожного вхідного вектора шарів ШНМ:

```
net.numInputs=5;
net.numLayers=8;
net.inputs{1}.size=16;
net.inputs{2}.size=4;
.....
net.inputs{5}.size=4.
```

При заданні зміщення вважаємо, що програмне забезпечення повинно було пройти основне тестування різними операціями кожного з методів. Задамо зміщення шарів Layer2, Layer4, Layer6, Layer8.

```
net.biasConnect(2)=3;
```

*net.biasConnect(4)=4;*

*net.biasConnect(6)=2;*

*net.biasConnect(8)=1*

або

*net.biasConnect=[0,3,0, 4, 0,2, 0,1].*

Матриця зв'язків ваг вхідних векторів має розмірність 8x5, оскільки кількість внутрішніх шарів – 8, а кількість вхідних векторів – 5. Якщо *i*-й шар зв'язаний з *j*-м вхідним вектором, то елемент [*i, j*] матриці зв'язків ваг вхідних векторів дорівнює 1 (*net.inputConnect(i,j)=1*), інакше цей елемент дорівнює 0. В нашому випадку одиниці дорівнюють [1,1], [1,2], [3,3], [5,4], [7,5] елементи матриці зв'язків ваг вхідних векторів:

*net.inputConnect(1,1)=1;*

*net.inputConnect(1,2)=1;*

*net.inputConnect(3,3)=1;*

*net.inputConnect(5,4)=1;*

*net.inputConnect(7,5)=1.*

Матриця зв'язків ваг внутрішніх шарів ШНМ має розмірність 8x8, оскільки кількість внутрішніх шарів складає 8. Якщо існує зв'язок, що йде від *j*-го шару до *i*-го шару, то елемент [*i, j*] матриці зв'язків ваг внутрішніх шарів дорівнює 1 (*net.layerConnect(i,j)=1*); якщо ж такий зв'язок відсутній, то цей елемент дорівнює 0. В нашому випадку одиниці дорівнюють [2,1], [3,1], [4,3], [5,3], [6,5], [7,5], [8,7] елементи матриці зв'язків ваг внутрішніх шарів:

*net.layerConnect=[0 0 0 0 0 0 0 0 ; 1 0 0 0 0 0 0 0 ; 1 0 0 0 0 0 0 0 ; 0 0 1 0 0 0 0 0 ; 0 0 1 0 0 0 0 0 ; 0 0 0 0 1 0 0 0 ; 0 0 0 0 1 0 0 0 ; 0 0 0 0 0 0 1 0].*

Матриця виходів та матриця цілей ШНМ мають однакову розмірність 1x8 елементів:

*net.outputConnect=[0 1 0 1 0 1 0 1];*

*net.targetConnect=[0 1 0 1 0 1 0 1].*

Також необхідно вказати діапазони значень вхідних векторів:

*net.inputs{1}.range = [0 2; 0 4; 0 6; 0 7; 0 8; 0 9; 0 10; 0 11; 0 12; 0 13; 0 14; 0 15; 0 16; 0 17 0 18; 0 20; 0 22];* (фактично при навчанні та роботі на ці входи будуть подаватись значення: 0, 1, 2; 0, 3, 4; 0, 5, 6; 0, 7; 0, 8; 0, 9; 0, 10; 0, 11; 0, 12; 0, 13; 0, 14; 0, 15; 0, 16; 0, 17, 18; 0, 19, 20; 0, 21, 22);

*net.inputs{2}.range=[0 26; 0 32; 0 50; 0 51];* (фактично при навчанні та роботі на ці входи будуть подаватись значення: 0, 20, 21, 22, 23, 24, 25, 26; 0, 27, 28, 29, 30, 31, 32; 0, 50; 0, 51);

*net. Inputs{3}.range=[0 43; 0 46; 0 52; 0 53];* (фактично при навчанні та роботі на ці входи будуть подаватись значення: 0, 41, 42, 43; 0, 44, 45, 46; 0, 52; 0, 53);

*net.Inputs{4}.range=[0 14; 0 19; 0 36; 0 40];* (фактично при навчанні та роботі на ці входи будуть подаватись значення: 0, 10, 11, 12, 13, 14; 0, 15, 16, 17, 18, 19; 0, 33, 34, 35, 36; 0, 37, 38, 39, 40);

*net.Inputs{5}.range=[0 9; 0 47; 0 48; 0 49];* (фактично при навчанні та роботі на ці входи будуть подаватись значення: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; 0, 47; 0, 48; 0, 49).

Для внутрішніх шарів нейромережі вкажемо такі параметри, як розмірність шару, активізаційна функція та метод ініціалізації. Для першого шару:

*net.layers{1}.size=20;*

*net.layers{1}.transferFcn='tansig';*

*net.layers{1}.initFcn='initnw';*

розмірність 1-го шару – 4, активізаційна функція – гіперболічний тангенс (*tansig*), метод ініціалізації – метод Нгуена-Відрой (*initnw*).

Для другого шару:

*net.layers{2}.size=5;*

*net.layers{2}.transferFcn='purelin';*

*net.layers{2}.initFcn='initnw';*

розмірність 2-го шару – 1, активізаційна функція – порогова лінійна функція (*purelin*), метод ініціалізації – метод Нгуена-Відрой.

Для третього шару:

$net.layers\{3\}.size=8;$

$net.layers\{3\}.transferFcn='tansig';$

$net.layer\{3\}.initFcn='initnw';$

розмірність 3-го шару – 8, активізаційна функція – гіперболічний тангенс, метод ініціалізації – метод Нгуена-Відроу.

Аналогічно можна описати всі інші внутрішні шари відповідно до рис.3.4.

Для визначення функцій, що забезпечать виконання базових нейромережових операцій, оберемо: метод ініціалізації – метод Нгуена-Відроу ( $net.initFcn='initnw'$ ). Алгоритм навчання ШНМ та критерій оцінки якості навчання потребує дослідження.

Здійснимо ініціалізацію ШНМ ( $net=init(net)$ ).

Для перегляду значень вагових коефіцієнтів нейромережі використовуємо оператори:

$net.IW\{1,1\}; net.IW\{1,2\}; net.IW\{3,3\}; net.IW\{5,4\}; net.IW\{7,5\}$  – перегляд значень вагових коефіцієнтів входів;

$net.LW\{2,1\}; net.LW\{3,1\}; net.LW\{4,3\}; net.LW\{5,3\}; net.LW\{6,5\}; net.LW\{7,5\}; net.LW\{8,7\}$  – перегляд значень вагових коефіцієнтів внутрішніх шарів;

$net.b\{1\}; net.b\{2\}; net.b\{3\}; net.b\{4\}; net.b\{5\}; net.b\{6\}; net.b\{7\}; net.b\{8\}$  – перегляд значень вагових коефіцієнтів зміщень.

Оператор  $gensim(net)$  дає змогу одержати модель в пакеті Simulink (рис.3.6-3.9).

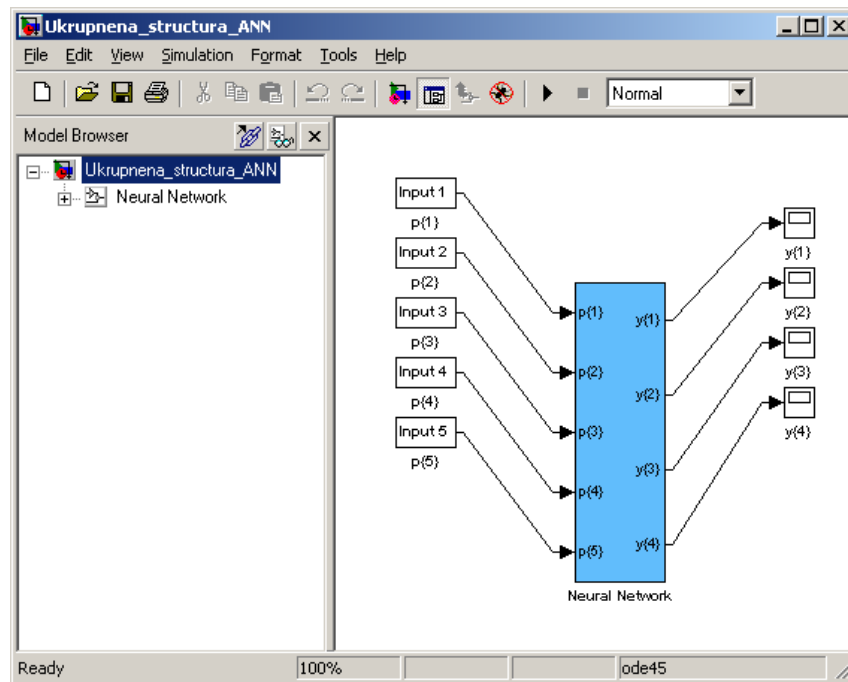


Рис. 3.6. Укрупнена структурна схема ШНМ в пакеті Simulink

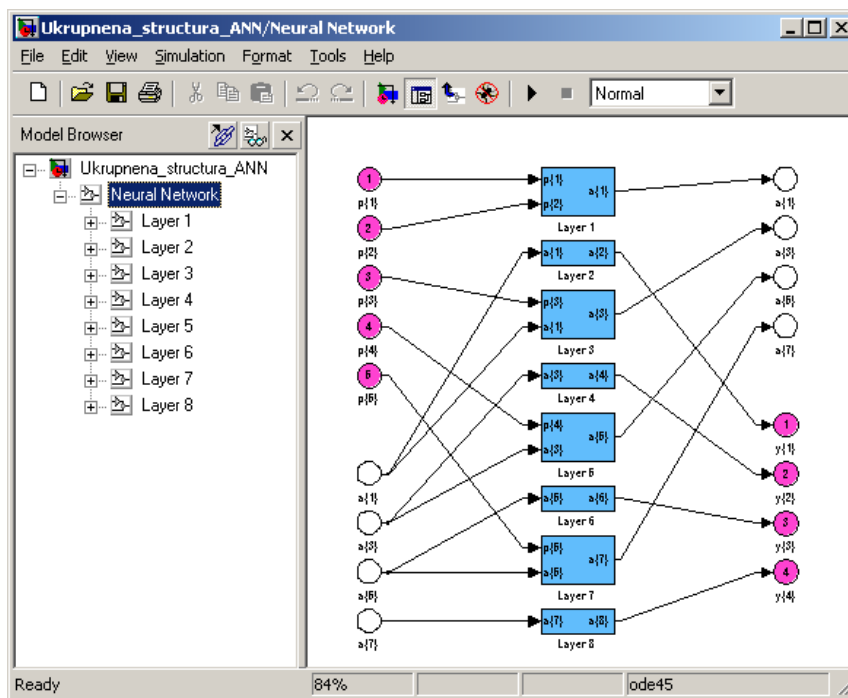


Рис. 3.7. Структурна схема шарів ШНМ в пакеті Simulink

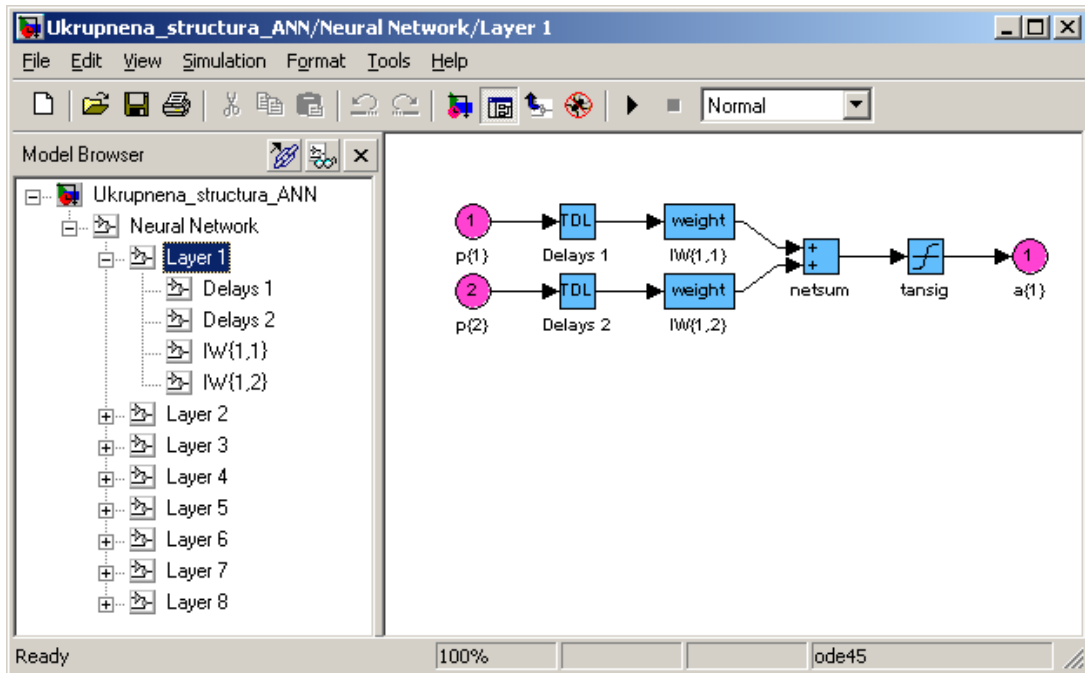


Рис. 3.8. Структурна схема першого шару ШНМ

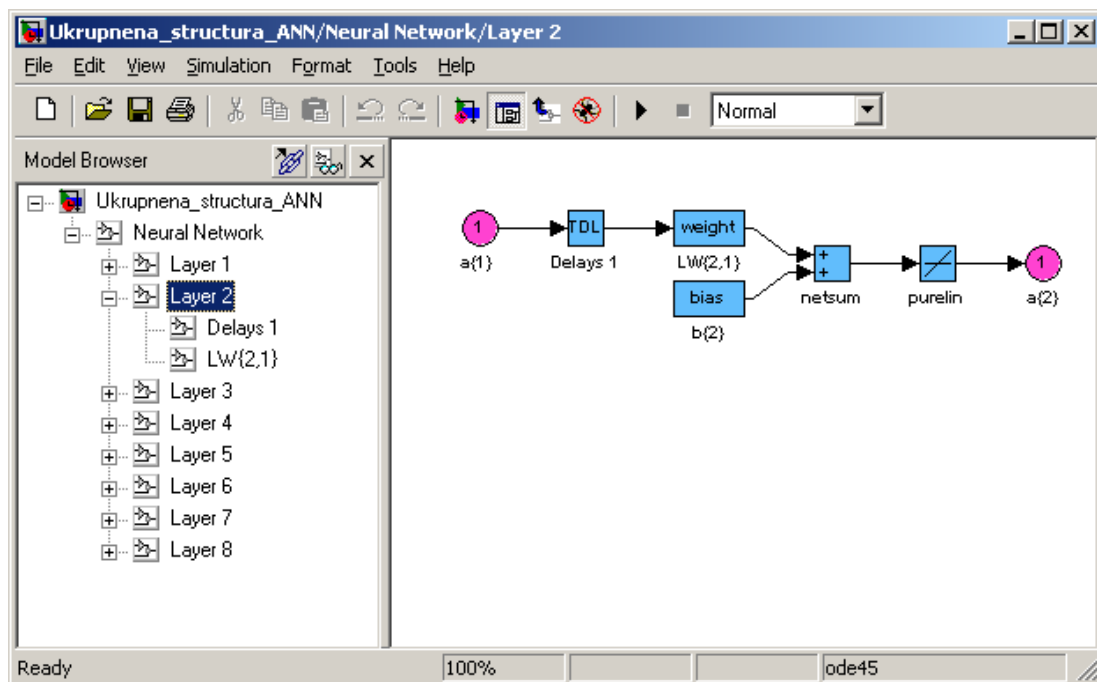


Рис.3.9. Структурна схема другого шару ШНМ

Структурні схеми третього, п'ятого та сьомого шарів аналогічні структурній схемі першого шару ШНМ, а структурні схеми четвертого, шостого та восьмого шарів аналогічні структурній схемі другого шару ШНМ.



де  $m$  – максимальна кількість помилок, як можна подати на вхід ШНМ ( $m=16$ );  
 $n$  – кількість методів основного тестування ( $n=4$ ). Тоді:

$$C(16,4) = \frac{16!}{4!(16-4)!} = \frac{12! \cdot 13 \cdot 14 \cdot 15 \cdot 16}{24 \cdot 12!} = \frac{13 \cdot 14 \cdot 15 \cdot 16}{24} = \frac{43680}{24} = 1820$$

Отже, 1820 векторів навчальної вибірки достатньо для того, щоб навчити ШНМ розпізнавати абсолютно всі вірно описані ситуації. Оскільки вибірка, на якій навчалась ШНМ, становить 2250 векторів (1820 векторів – вірні ситуації, 300 векторів – невірні ситуації, 130 векторів – граничні ситуації), то вона є достатньою, тому ШНМ працює достовірно.

В результаті аналізу таблиць 1-3 додатку Б було визначено, що за часовим показником найкраще використовувати алгоритм навчання СGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле, алгоритм Флетчера-Рівса (CGF), алгоритм Полака-Рібейри (CGP), однокроковий алгоритм методу січної (OSS), пороговий алгоритм оберненого поширення помилки (Rprop), алгоритм навчання SCG з використанням комбінованого критерію якості. За показником “кількість епох” найкращими є: алгоритм навчання СGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле, алгоритм Флетчера-Рівса (CGF), алгоритм Полака-Рібейри (CGP), алгоритм навчання SCG, алгоритм навчання Бройтона, Флетчера, Гольдфарба і Шано (BFGS), однокроковий алгоритм методу січної (OSS), алгоритм Левенберга-Марквардта (LM) з використанням комбінованого критерію якості. Процес навчання алгоритмом навчання СGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле з використанням комбінованого критерію оцінки якості навчання відображено на рис. 3.10. На рисунку верхня крива відображає похибку навчання ШНМ за заданим алгоритмом. Пряма внизу відображає цільову функцію точності.

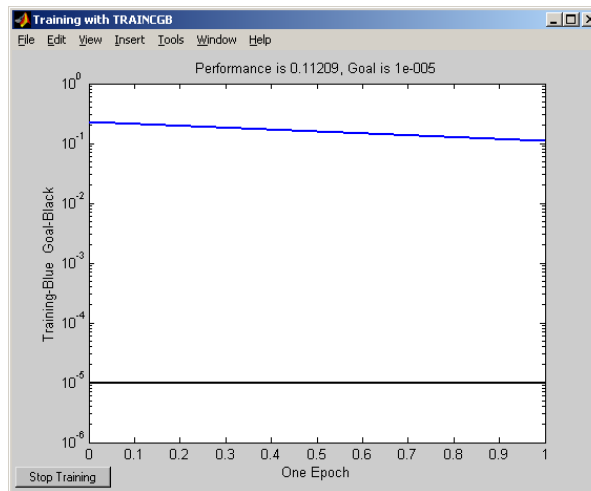


Рис.3.10. Алгоритм навчання СГВ на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле

Для розрізнення значень вибірки, які відрізняються на порядок, було вирішено масштабувати навчальну вибірку за допомогою функції `premnmx`. Діапазони входів і виходів при використанні масштабованої вибірки відображені на рис. 3.11 – 3.19.

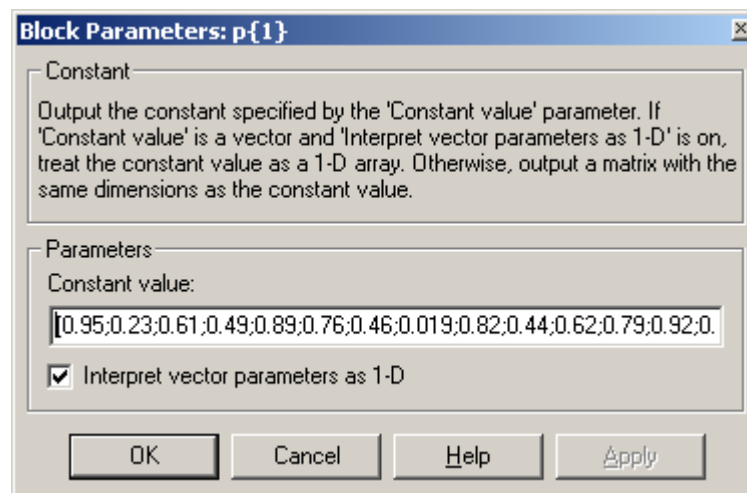


Рис. 3.11. Діапазони входу  $p_1$  ШНМ після масштабування навчальної вибірки функцією `premnmx`

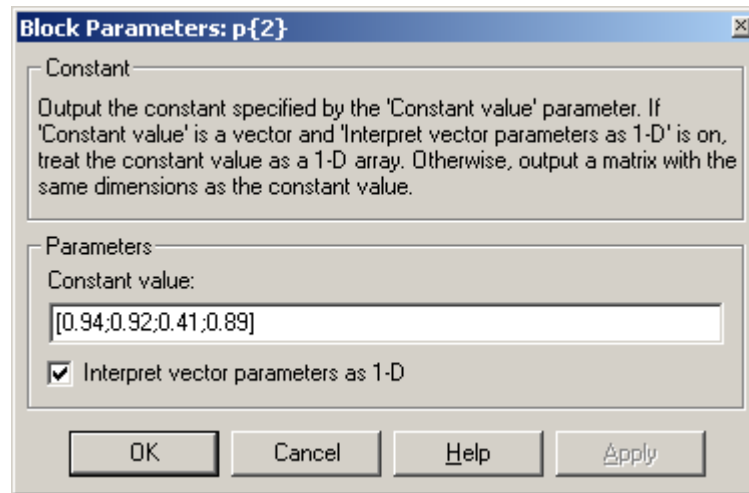


Рис. 3.12. Діапазони входу р2 ШНМ після масштабування навчальної вибірки функцією `premnmx`

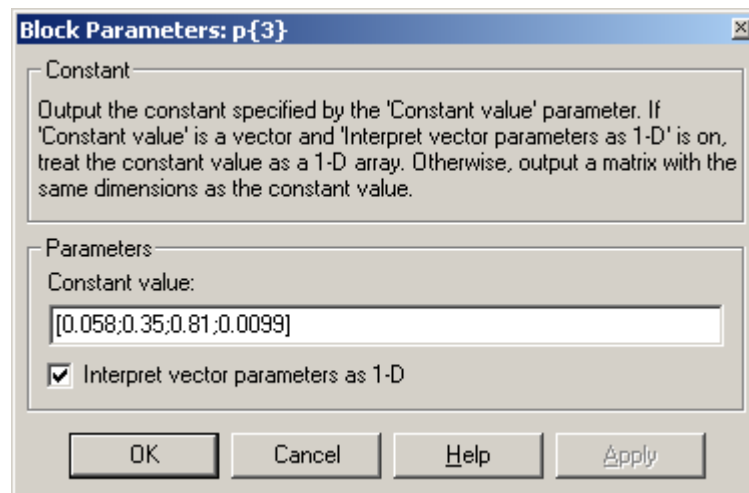


Рис. 3.13. Діапазони входу р3 ШНМ після масштабування навчальної вибірки функцією `premnmx`

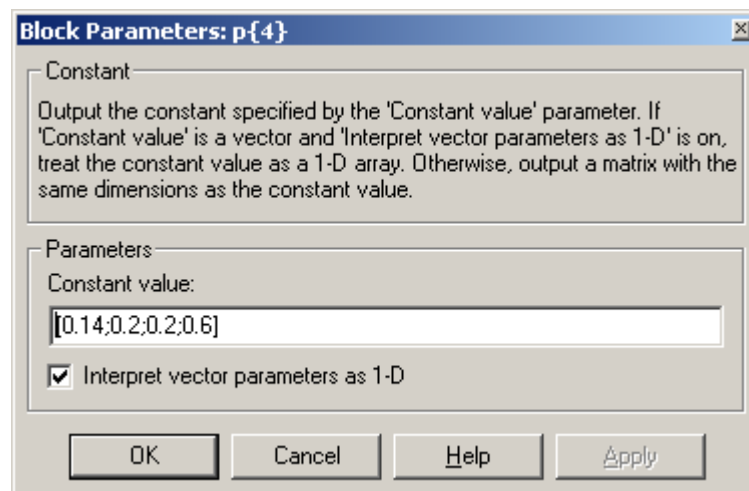


Рис. 3.14. Діапазони входу р4 ШНМ після масштабування навчальної вибірки функцією `premnmx`

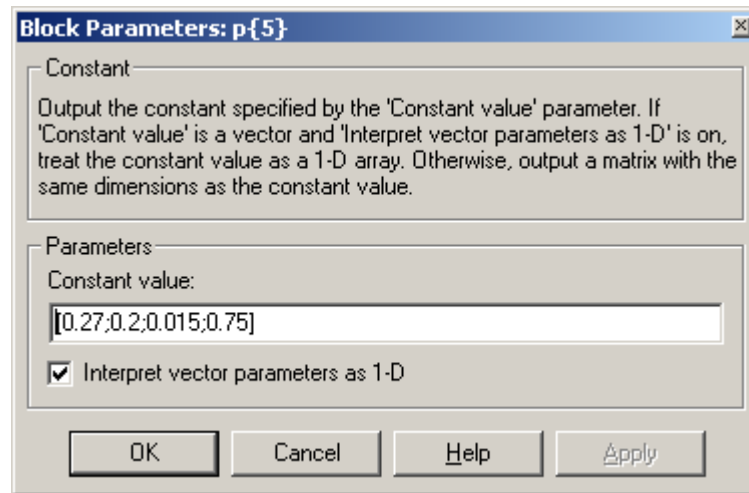


Рис. 3.15. Діапазони входу р5 ШНМ після масштабування навчальної вибірки функцією `premnmx`

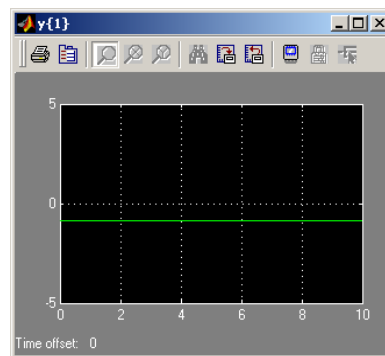


Рис. 3.16. Графік виходу  $y_1$  при навчанні мережі масштабованою вибіркою з 2250 векторів

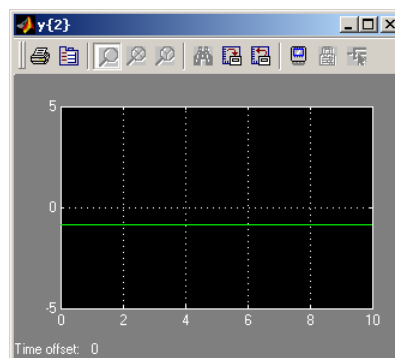


Рис. 3.17. Графік виходу  $y_2$  при навчанні мережі масштабованою вибіркою з 2250 векторів

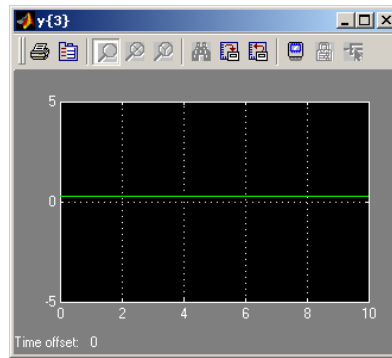


Рис. 3.18. Графік виходу  $y_3$  при навчанні мережі масштабованою вибіркою з 2250 векторів

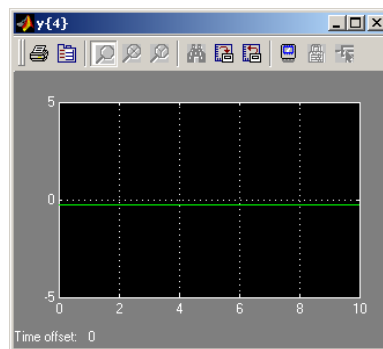


Рис. 3.19. Графік виходу  $y_4$  при навчанні мережі масштабованою вибіркою з 2250 векторів

Після масштабування було досліджено навчання мережі вибіркою з 2250 векторів різними алгоритмами навчання (таблиця 4 додатку Б).

В результаті аналізу таблиць 1-4 додатку Б було зроблено висновок, що похибка навчання змодельованої ШНМ залежить від критерію оцінки якості навчання та від форми представлення вхідних даних. Тому надалі використовується комбінований критерій якості навчання і масштабована навчальна вибірка. Мінімальна похибка, яку було досягнуто при використанні комбінованого критерію якості навчання та масштабованої навчальної вибірки з 2250 векторів, становить 0.448359. Меншої похибки навчання досягати не потрібно, оскільки виходи мережі, які знаходяться в інтервалі  $[-1; 1]$  перетворюються для представлення цілими значеннями 1 або 0 (є чи немає помилки  $i$ -го рівня категорійності відповідно):

$$Y_i = \begin{cases} 1, & \text{якщо } Y_i > 0; \\ 0, & \text{якщо } Y_i \leq 0; \end{cases}$$

В результаті аналізу таблиці 4 додатку Б за часовим показником та за показником “кількість епох” найкращими є: алгоритм навчання CGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле, алгоритм навчання SCG, алгоритм навчання Флетчера-Рівса, алгоритм навчання Полака-Рібейри, пороговий алгоритм оберненого поширення помилки Rprop. Оскільки алгоритм навчання CGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле, алгоритм навчання Флетчера-Рівса та алгоритм навчання Полака-Рібейри є модифікаціями метода спряженого градієнта, то для навчання оберемо один з них – алгоритм навчання CGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле. Графіки навчання алгоритмом навчання CGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле, порогового алгоритму оберненого поширення помилки Rprop та алгоритму навчання SCG з використанням комбінованого критерію оцінки якості навчання відображено на рис. 3.20-3.22.

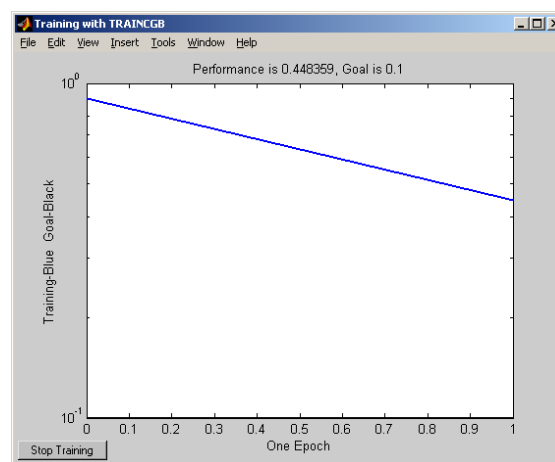


Рис. 3.20. Алгоритм навчання CGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле

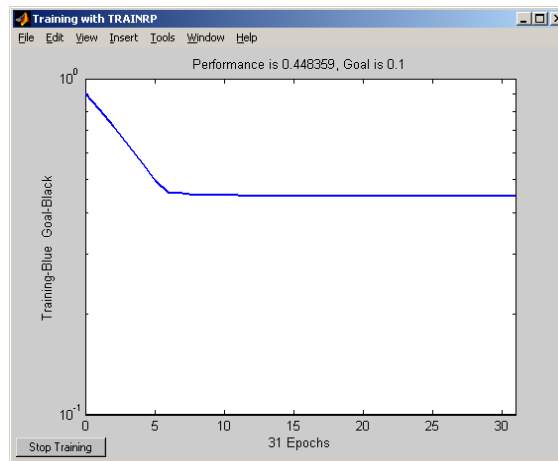


Рис. 3.21. Пороговий алгоритм оберненого поширення помилки Rprop

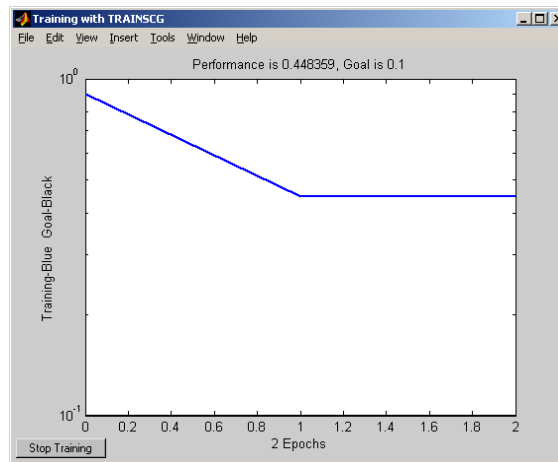


Рис. 3.22. Алгоритм навчання SCG

Для тестування ШНМ було побудовано тестову вибірку з 200 векторів, яка також підлягала масштабуванню. Процес навчання і тестування різними алгоритмами навчання з використанням комбінованого критерію якості відображається на рис. 3.23 – 3.25. На рисунках нижня крива відображає графік навчання, а верхня крива відображає графік тестування ШНМ.

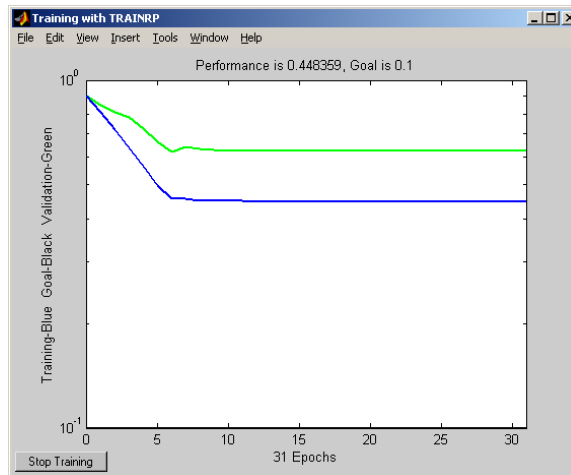


Рис. 3.23. Графіки навчання і тестування ШНМ за пороговим алгоритмом оберненого поширення помилки Rprop

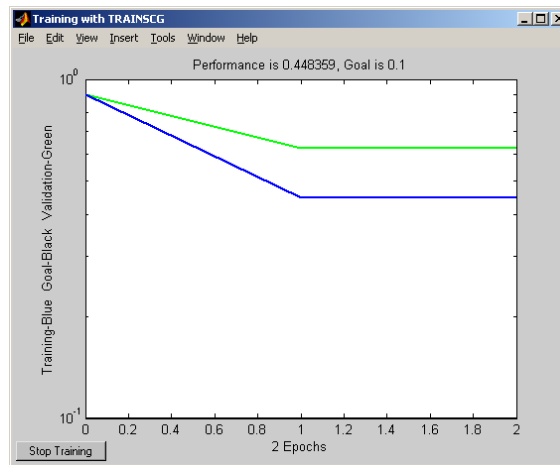


Рис. 3.24. Графіки навчання і тестування ШНМ за алгоритмом навчання SCG

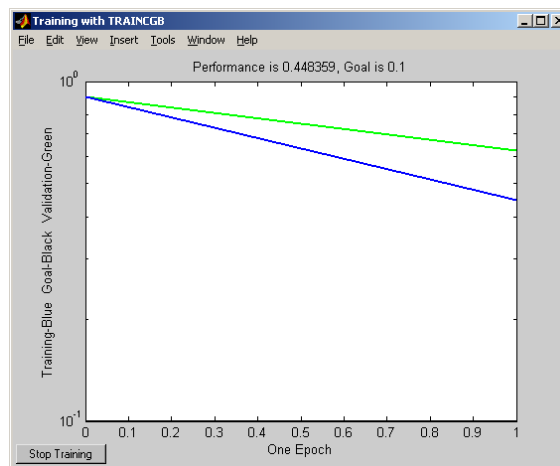


Рис. 3.25. Графіки навчання і тестування ШНМ за алгоритмом навчання CGB на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле

Отже, аналізуючи графіки навчання і тестування ШНМ, можна зробити висновок, що для навчання мережі найкраще підходить алгоритм навчання СГВ на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле та його модифікації (алгоритм навчання Флетчера-Рівса або алгоритм навчання Полака-Рібейри).

### 3.9. Висновки

1. Вперше одержано метод ідентифікації прихованих помилок ПЗ на основі нейромережних інформаційних технологій, суть якого полягає у виявленні множини типів прихованих помилок різних рівнів категорійності та аналізу цієї множини на предмет необхідності повторного тестування. Метод відрізняється від відомих тим, що вхідна інформація про результати основного тестування опрацьовується штучною нейронною мережею, яка відповідає моделі процесу повторного тестування на базі НІТ.

2. Розроблено алгоритм ідентифікації прихованих помилок програмного забезпечення, який базується на методі ідентифікації прихованих помилок ПЗ і надає висновок щодо необхідності та методу(ів) повторного тестування після опрацювання звіту про основне тестування.

3. Розроблено метод та алгоритм формування висновку про необхідність повторного тестування, які базуються на використанні експертних даних у формі продукційних правил і описують процес формування висновку про необхідність повторного тестування і рекомендації щодо методу повторного тестування.

4. Досліджено процес навчання і тестування ШНМ у пакеті Matlab, у результаті чого зроблено висновок, що похибка навчання змодельованої ШНМ залежить від критерію оцінки якості навчання та від форми представлення вхідних даних. Тому надалі використовується комбінований критерій якості навчання і масштабовану навчальну вибірку. Мінімальна похибка, яку було досягнуто при використанні комбінованого критерію якості навчання та

масштабованої навчальної вибірки з 2250 векторів, становить 0.448359. меншої похибки навчання досягати не потрібно, оскільки виходи мережі, які знаходяться в інтервалі  $[-1; 1]$  перетворюються для представлення цілими значеннями 1 або 0 (є чи немає помилки  $i$ -го рівня категорійності відповідно).

5. За часовим показником та за показником “кількість епох” найкращими для розглянутої ШНМ є алгоритм СГВ на основі метода спряженого градієнта з оберненим поширенням і рестартами в модифікації Пауела-Біеле та його модифікації (алгоритм навчання Флетчера-Рівса або алгоритм навчання Полака-Рібейри).