

ВСТУП

Останнім часом відбувається активний розвиток інформаційних технологій на базі теорії штучного інтелекту. Зокрема на напрямках, які імітують природні процеси, такі як діяльність нейронів мозку або процес природного добору. Найбільш популярними серед них є нейронні мережі і генетичні алгоритми. Ці технології дозволяють розв'язувати задачі прогнозування, класифікації, керування, пошуку оптимальних варіантів і зовсім незамінні під час розв'язання задач, що не мають простого алгоритмічного розв'язання. Це обіцяє перспективність використання нейронних мереж і генетичних алгоритмів у системах автоматизованого проектування, а також у системах управління й автоматики.

Виконання лабораторних робіт спрямовано на набуття студентами практичних навичок побудови, дослідження і використання штучних нейронних мереж і генетичних алгоритмів для розв'язання задач проектування і побудови автоматизованих систем управління. У роботах вивчаються основні поняття, пов'язані зі штучними нейронними мережами і генетичними алгоритмами: основні генетичні оператори, їхні типи, параметри генетичних алгоритмів, способи кодування хромосом, стратегії вибору батьківських пар та осіб у нову популяцію, елементи нейронних мереж, архітектури нейронних мереж, методи навчання нейронних мереж, розв'язання задач за допомогою нейронних мереж і генетичних алгоритмів.

Практикум орієнтований на розв'язання студентами задач із застосуванням комп'ютерних засобів і пакетів моделювання. Тому під час підготовки до виконання робіт треба вивчити особливості режимів роботи і застосування використовуваних програмно-технічних засобів.

Перед виконанням лабораторних робіт студенти повинні пройти інструктаж з техніки безпеки. Її вимог, як і інструкцій з поведінки студентів у лабораторії необхідно строго дотримуватись протягом усього лабораторного практикуму.

1 ОПТИМІЗАЦІЯ СКЛАДНИХ ФУНКЦІЙ ЗА ДОПОМОГОЮ ГЕНЕТИЧНИХ АЛГОРИТМІВ

1.1 Мета роботи

Вивчення принципів роботи генетичних алгоритмів. Набуття навичок реалізації генетичних алгоритмів для оптимізації складних багатопараметричних функцій.

1.2 Вказівки з організації самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно: ознайомитися з принципами роботи генетичних алгоритмів (ГА); з основними видами генетичних операторів, способами кодування рішень у ГА, стратегіями вибору батьківських пар для схрещування і способами добору індивідуумів у нову популяцію.

З цією метою може бути використаний лекційний матеріал з відповідних тем, матеріал, викладений у рекомендованій літературі [1, с. 10 - 25], а також матеріал цих методичних вказівок.

Основні принципи ГА були сформульовані Холландом у 1975 році. ГА використовують пряму аналогію з біологічними процесами.

Основними генетичними операторами є схрещування, мутація та інверсія.

ГА працює в такий спосіб. Випадково генерується початкова популяція індивідуумів, що являють собою набір вихідних розв'язків. Робота ГА є ітераційним процесом, який продовжується поки не виконається задане число поколінь (генерацій) або який-небудь інший критерій зупинки. У ході кожної ітерації кожний розв'язок оцінюється з використанням функції відповідності (fitness function). Для створення наступного покоління нові індивідууми, що мають назву нащадків, створюються або шляхом схрещування, або мутації. Нова популяція формується шляхом вибору пропорційно функції відповідності деяких батьків і нащадків і вилучення тих індивідуумів, що залишилися, для того, щоб зберігати постійний розмір популяції. У результаті наведених операцій на кожному етапі еволюції створюються популяції з більш досконалішими індивідуумами.

Структурна схема генетичного алгоритму зображена на рис. 1.1.

Генетичне наслідування моделюється таким чином (табл. 1.1)

Таблиця 1.1 – Модель генетичного наслідування

Хромосома	Вектор (послідовність) із нулів та одиниць. Кожна позиція (біт) називається геном.
Індивідуум	Набір хромосом - варіант розв'язку задачі
Схрещування	Операція, при якій хромосоми обмінюються частинами
Мутація	Випадкова зміна однієї або кількох позицій в хромосомі

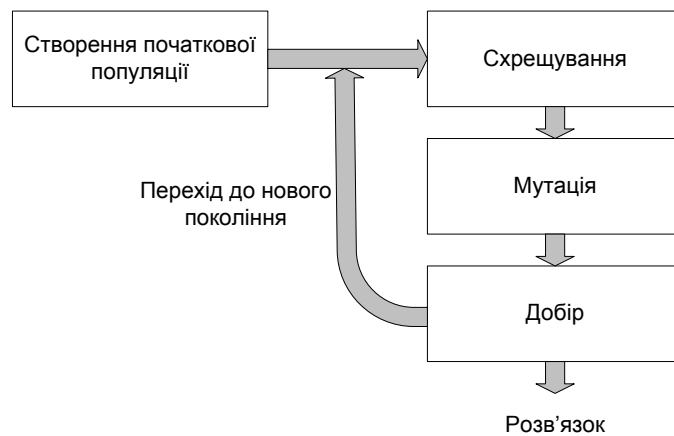


Рисунок 1.1 - Структурна схема генетичного алгоритму

Стратегії вибору батьківських пар для схрещування.

Панміксія - обидва члени популяції, що створюють батьківську пару випадковим чином вибираються із усієї популяції, причому будь-який член популяції може стати учасником декількох пар.

Селективний - батьками можуть стати лише ті індивідууми, значення пристосованості яких не менше середнього значення пристосованості за популяцією.

Інбридинг - перший член пари вибирається випадково, а другим з більшою імовірністю буде максимально близький до нього член популяції.

Аутбридинг - пари формуються аналогічно, але з максимально далеких членів популяції.

Розрізняють фенотипний та генотипний інбридинг і аутбридинг.

Існують такі механізми добору членів нової популяції: елітний, добір з витисненням, "колесо рулетки".

При *елітному* доборі нова популяція складатиметься лише з найкращих членів репродукційної групи, яка поєднує батьків, нащадків і мутантів.

При доборі з *витисненням*, чи буде член репродукційної групи заноситися в нову популяцію визначається не тільки величиною його пристосованості, але і тим, чи є в новій популяції член з аналогічним набором хромосом.

Відповідно до підходу "*колесо рулетки*", добір здійснюється на основі деякої функції розподілу, що будується пропорційно обчисленим функціям відповідності створених варіантів-хромосом.

1.3 Опис лабораторної установки

Як лабораторна установка використовується персональна ЕОМ типу IBM PC. Для оптимізації заданої функції та визначення параметрів генетичного алгоритму використовується програма, підготовлена під час самостійної роботи.

1.4 Порядок виконання роботи

Ознайомитися за допомогою викладача з особливостями і режимами роботи застосовуваних комп'ютерних засобів.

Отримати завдання відповідно до номера варіанта. Варіанти завдань наведені у табл. 1.2 (необхідно знайти максимум функції).

Скласти та відлагодити програму оптимізації заданої функції за допомогою ГА.

Підібрати параметри ГА і знайти найкращий розв'язок задачі за допомогою складеної програми. Необхідна точність - 2 знаки після коми.

Зробити висновки, оформити і захистити звіт про виконану роботу.

Таблиця 1.2 – Варіанти завдань

Но- мер	Функція	Крос- совер	Вибір батьків	Механізм добору
1	2	3	4	5
1	$F(x_1, x_2) = \frac{100}{100(x_1^2 - x_2) + (1 - x_1)^2 + 1},$ $-1,28 \leq x_{1,2} \leq 1,28.$	Одно- точковий	Панміксія	Елітний
2	$F(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10\cos(2\pi x_1) - 10\cos(2\pi x_2),$ $-5,12 \leq x_{1,2} \leq 5,12.$	Двох- точковий	Селектив- ний	З витис- ненням
3	$F(x_1, x_2) = (-2x_2^3 + 6x_2^2 + 6x_2 + 10) \cdot \sin(\ln(x_1) \cdot e^{x_2}),$ $0,5 \leq x_1 \leq 1,1 \quad 1,0 \leq x_2 \leq 4,6.$	Рівно- мірний	Інбридинг	Колесо рулетки
4	$F(z) = \frac{1}{40}(2z - 0,9)(7z - 1)(17z - 19)(15z - 2),$ $z = \frac{(x_1 - 3)^2 + (x_2 - 3)^2}{18}, \quad 0 \leq x_{1,2} \leq 6.$	Одно- точковий	Аутбри- динг	Елітний
5	$F(x_1, x_2) = \frac{1}{\frac{x_1^2 + x_2^2}{200} - \cos(x_1) \cos\left(\frac{x_2}{\sqrt{2}}\right) + 2},$ $-20 \leq x_{1,2} \leq 20.$	Двох- точковий	Панміксія	З витис- ненням
6	$F(x_1, x_2, \dots, x_5) = \sum_{i=1}^5 [x_i],$ $-5,12 \leq x_{1,2,3,4,5} \leq 5,12.$	Рівно- мірний	Селектив- ний	Колесо рулетки
7	$F(x_1, x_2) = 0.5 + \frac{\sqrt{\sum x_i^2} - 0.5}{(1 + 0.001 \cdot \sum x_i^2)^2}$ $-100 \leq x_{1,2} \leq 100$	Одно- точковий	Інбридинг	Елітний
8	$F(x_1, x_2) = 1 + (\sum x_i^2)^{0.25} \cdot [\sin(50 \cdot (\sum x_i^2)^{0.1}) + 1]$ $-100 \leq x_{1,2} \leq 100$	Двох- точковий	Аутбри- динг	З витис- ненням

Продовження таблиці 1.2

1	2	3	4	5
9	$F(x_1, x_2) = \frac{100}{100(x_1^2 - x_2^2) + (1 - x_1)^2 + 1},$ $-1,28 \leq x_{1,2} \leq 1,28.$	Рівно- мірний	Панміксія	Колесо рулетки
10	$F(x_1, x_2) = 20 + x_1^2 + x_2^2 - 10\cos(2\pi x_1) - 10\cos(2\pi x_2),$ $-5,12 \leq x_{1,2} \leq 5,12.$	Одно- точковий	Селектив- ний	Елітний
11	$F(x_1, x_2) = (-2x_2^3 + 6x_2^2 + 6x_2 + 10) \cdot \sin(\ln(x_1)) \cdot e^{x_2},$ $0,5 \leq x_1 \leq 1,1 \quad 1,0 \leq x_2 \leq 4,6.$	Двох- точковий	Інбридинг	3 витис- ненням
12	$F(z) = \frac{1}{40}(2z - 0,9)(7z - 1)(17z - 19)(15z - 2),$ $z = \frac{(x_1 - 3)^2 + (x_2 - 3)^2}{18}, \quad 0 \leq x_{1,2} \leq 6.$	Рівно- мірний	Аутбри- динг	Колесо рулетки
13	$F(x_1, x_2) = \frac{1}{\frac{x_1^2 + x_2^2}{200} - \cos(x_1)\cos\left(\frac{x_2}{\sqrt{2}}\right) + 2},$ $-20 \leq x_{1,2} \leq 20.$	Одно- точковий	Панміксія	Елітний
14	$F(x_1, x_2, \dots, x_5) = \sum_{i=1}^5 [x_i],$ $-5,12 \leq x_{1,2,3,4,5} \leq 5,12.$	Двох- точковий	Селектив- ний	3 витис- ненням
15	$F(x_1, x_2) = \sum (x_i^2 - 10\cos(2\pi \cdot x_i)),$ $-100 \leq x_{1,2} \leq 100$	Рівно- мірний	Інбридинг	Колесо рулетки
16	$F(x_1, x_2) = \sum (10\cos(2\pi x_i) - x_i^2) - 100,$ $-5.12 \leq x_{1,2} \leq 5.12$	Одно- точковий	Аутбри- динг	Елітний
17	$F(x_1, x_2) = 0.5 + \frac{\sqrt{\sum x_i^2} - 0.5}{(1 + 0.001 \cdot \sum x_i^2)^2}$ $-100 \leq x_{1,2} \leq 100$	Двох- точковий	Панміксія	3 витис- ненням

1.5 Зміст звіту

Звіт має містити:

- титульний аркуш;
- мету роботи;
- постановку та вихідні дані задачі;
- результати оптимізації функції;
- значення параметрів ГА, при яких знайдено найкращий розв'язок задачі оптимізації;
- кількість ітерацій алгоритму, що були виконані для знаходження найкращого розв'язку;
- оцінку витрат часу для знаходження розв'язку;
- аналіз отриманих результатів та висновки по роботі.

1.6 Контрольні запитання та завдання

1. Для розв'язання яких задач використовують ГА?
2. Назвіть переваги та недоліки ГА.
3. Назвіть основні етапи функціонування ГА.
4. Що являють собою генотип і фенотип хромосоми?
5. Опишіть символічну модель ГА.
6. Перелічіть відомі вам типи оператора кроссовера.
7. Які стратегії добору батьківських пар для схрещування ви знаєте?
8. Назвіть основні механізми добору індивідів у нову популяцію.

2 СИНТЕЗ ТОПОЛОГІЧНИХ СТРУКТУР ТРСО ЗА ДОПОМОГОЮ ГЕНЕТИЧНИХ АЛГОРИТМІВ

2.1 Мета роботи

Вивчення методики проектування топологічних структур територіально розподілених систем обслуговування (ТРСО). Вивчення особливостей реалізації генетичних алгоритмів (ГА) для синтезу топологічних структур. Набуття навичок розв'язання задачі структурно - топологічної оптимізації ТРСО за допомогою ГА.

2.2 Вказівки з організації самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно ознайомитися: з постановкою задачі структурно-топологічного синтезу ТРСО; способами подання даних у ГА; з основними генетичними операторами і їхніми різновидами; з'ясувати принципи роботи ГА у процесі пошуку оптимуму функції.

З цією метою може бути використаний лекційний матеріал з відповідних тем, матеріал, викладений у рекомендованій літературі [1, с. 10 - 25; 2, с. 57 - 62; 3, с. 131 - 145], а також матеріал цих методичних вказівок.

Об'єктом дослідження є кільцева і деревоподібна топологічні структури ТРСО.

Загальна постановка задачі структурно-топологічної оптимізації ТРСО формулюється так. Задані: множина об'єктів, що потребують обслуговування $I = \{i : i = \overline{1, n}\}$, їх територіальне розміщення; вартості типового вузла обслуговуючої підсистеми та одиниці довжини зв'язку; відомі параметри елементів обслуговуючої системи (центру, вузлів, зв'язків).

Прийнято такі припущення: вузли можуть розміщуватися тільки в місцях розташування об'єктів; центральна підсистема має фіксовану вартість та продуктивність, достатню для обробки запитів усіх об'єктів.

Необхідно визначити: кількість вузлів m^o , місця їхнього розміщення $B^o = \{b_j^v\}, j = 1, m^o, v \in H$ (де $H = \{j: j = \overline{1, n}\}$ – множина місць можливого розміщення вузлів), перелік об'єктів, що обслуговуються кожним із вузлів $I_j^o = \{w_{ij}\}$.

При цьому сумарна вартість системи має бути мінімальною:

$$Z = \sum_{j=0}^n f_j b_j + \sum_{i,j=0}^n c(l_{ij}) w_{ij} \rightarrow \min, \quad (2.1)$$

де n – кількість об'єктів системи (місць можливого розміщення вузлів); f_j – фіксована вартість вузла системи; $c(l_{ij})$ – вартість зв'язку об'єкта i системи з об'єктом j (є функцією відстані між об'єктами - l_{ij}); w_{ij} – шукана змінна, яка відображає схему взаємозв'язків елементів системи ($w_{ij} = 1$, якщо пункт i з'єднано з пунктом j , та $i \neq j$, $w_{ij} = 0$ у протилежному випадку); b_j – шукана змінна, яка відображає розміщення вузлів системи ($b_j = 1$, якщо $\sum_{i=0}^n w_{ij} > 1$, $b_j = 0$ у протилежному випадку).

Структурні обмеження:

- системою обслуговуються всі об'єкти, а кожен з них тільки однією підсистемою;
- загальна кількість зв'язків в системі

$$\sum_{i=0}^n \sum_{j=0}^n w_{ij} = 2n; \quad (2.2)$$

- в системі використовуються однотипні підсистеми (елементи, вузли та зв'язки), потужності яких достатні для обслуговування закріплених об'єктів.

Синтез кільцевих структур здійснюється за критерієм мінімуму довжини каналів зв'язку:

$$Z = \sum_{i=0}^n \sum_{j=i+1}^n l_{ij} w_{ij} \rightarrow \min. \quad (2.3)$$

Для синтезу кільцевих структур можна використовувати подання хромосоми у вигляді масиву цілих чисел. При такому кодуванні в позиції i хромосоми знаходиться j -й пункт ТРСО. Як функція відповідності при синтезі кільцевих структур використовується співвідношення (2.3).

Хромосома при синтезі деревоподібних структур подається у вигляді вектора попередніх елементів. Деревоподібна структура може бути подана вектором з n елементів шляхом визначення довільної вершини як кореня та

зазначення в позиції i даного вектора пункту, що передуює пунктові i стосовно кореня. Наприклад, вектор $(0,1,2,3,2)$ - це дерево з ребрами $(1,2)$, $(2,3)$, $(3,4)$, $(2,5)$. Елемент "0" вектора визначає корінь дерева (пункт 1). Елементами вектора є гени, що є зв'язки дерева. Як функція відповідності при синтезі деревоподібних структур використовується співвідношення (2.1).

Загальна схема генетичного алгоритму може бути подана в такий спосіб.

1. Завдання розміру популяції M , числа генерацій N_{gen} , ймовірності кросовера p_{cros} , ймовірності мутації p_{mut} , кількості операцій кросовера N_{cros} .

2. $v := 0$ ($v = 1, 2, \dots, N_{gen}$).

3. Синтез випадкової початкової популяції $P(0)$ розміром M .

4. Випадковий вибір N_{cros} пар хромосом з популяції $P(v)$ і застосування операції кросовера до кожної пари з заданою ймовірністю.

5. Застосування операції мутації до кожної хромосоми популяції $P(v)$ із заданою ймовірністю.

6. Сортування хромосом у порядку погіршення значення функції відповідності.

7. Добір M хромосом з найкращими значеннями функції відповідності з популяції $P(v)$ у нову популяцію $P(v+1)$.

8. $v := v + 1$.

19. Якщо $v \leq N_{gen}$, то перейти до п.4.


10. Виведення хромосоми з найкращим значенням функції відповідності.

2.3 Опис лабораторної установки


Як лабораторна установка використовується персональна ЕОМ типу IBM PC. Структурно-топологічна оптимізація TPCO здійснюється за допомогою програми NWDesigner.

Після запуску програми (TopProg.exe) на екран виводиться форма "Синтез системи". Для початку роботи з програмою необхідно задати місце розташування об'єктів TPCO. Існує три варіанти введення даних. Один з них, найбільш зручний для користувачів - це введення координат пунктів за допомогою маніпулятора "миша". Для цього необхідно:

1. Вибрати в головному меню пункт "Файл", а в ньому - команду "Створити".

2. Увімкнути режим введення пунктів. Кнопка  на панелі інструментів.

3. Натисканням лівої кнопки миші на робочій області додавати пункти.

Якщо необхідно змінити місце розташування пункту, то необхідно натиснути кнопку  на панелі інструментів і, натиснувши ліву кнопку миші, перетягнути пункт на нове місце.

Оскільки оптимізація топологічних структур ведеться за критеріями сумарної вартості та сумарної довжини ліній зв'язку, то необхідно в головному меню вибрати пункт "Вид", команду "Инспектор структуры", закладку - "Весовые коэффициенты". Вибрати критерій вартості або довжини.

Для синтезу топологічної структури ТРСО необхідно в головному меню вибрати пункт "Синтез сети", у ньому - пункт меню "Автоматический", потім тип синтезованої структури (деревоподібна або кільцева), після чого вибрати один з алгоритмів синтезу.

Процес синтезу продовжується поки курсор миші зображений у вигляді пісочного годинника. Якщо обрано генетичний алгоритм, то процес синтезу можна або призупинити, натиснувши на кнопку "Пауза", або зупинити - "Стоп".

На закладці "Параметры генетического алгоритма" інспектора структури вводяться параметри генетичного алгоритму: кількість генерацій, кількість операцій кроссовера, ймовірність кроссовера, ймовірність мутації, розмір популяції. Тут також задається максимальна кількість рівнів для синтезу деревоподібних структур, що є обмеженням при використанні генетичного алгоритму.

2.4 Порядок виконання роботи

Ознайомитися за допомогою викладача з особливостями і режимами роботи використовуваних комп'ютерних засобів та програм.

Ввести координати пунктів ТРСО (не менш 20).

Створити варіанти кільцевої топології за критерієм сумарної довжини зв'язків та деревоподібної - за критерієм сумарної вартості системи за допомогою генетичного алгоритму та ще одного з запропонованих алгоритмів.

Порівняти результати оптимізації за точністю розв'язків і за часом, який було витрачено на їхнє отримання.

Провести оптимізацію деревоподібної структури при різних комбінаціях критеріїв за допомогою генетичного алгоритму й одного з запропонованих алгоритмів. Порівняти одержані результати.

Побудувати графіки залежності обчислювальної складності генетичного алгоритму від ймовірностей кроссовера і мутації, подивитися, як при цьому змінюється якість розв'язків.

Підібрати такі значення керуючих параметрів ГА (розмір популяції, ймовірність мутації, ймовірність кроссовера), які забезпечать найкраще відношення параметрів витрати часу - точність розв'язків для задач, що розглядаються.

Зробити висновки, оформити і захистити звіт про виконану роботу.

2.5 Зміст звіту

Звіт має містити:

- титульний аркуш;
- мету роботи;
- постановку і вхідні дані задачі;
- варіанти топології ТРСО, отримані за допомогою різних алгоритмів оптимізації;
- значення керуючих параметрів ГА, при яких досягається найкраще поєднання параметрів витрати часу - точність рішень;
- графіки залежностей обчислювальної складності генетичного алгоритму від ймовірностей кроссовера та мутації;
- аналіз отриманих результатів та висновки по роботі.

2.6 Контрольні запитання та завдання

1. Опишіть структуру даних ГА при оптимізації кільцевих структур ТРСО.
2. Опишіть структуру даних ГА при оптимізації деревоподібних структур ТРСО.
3. Які типи оператора кроссовера застосовуються при синтезі кільцевих структур?
4. Які типи оператора мутації застосовуються при синтезі кільцевих структур?
5. Назвіть стратегії, що використовуються для запобігання передчасної збіжності ГА.
6. Опишіть загальну схему ГА, який застосовується при синтезі топологічних структур ТРСО.
7. Як виконується операція кроссовера при синтезі деревоподібних структур ТРСО?
8. Які типи оператора мутації застосовуються при синтезі деревоподібних структур?

3 НЕЙРОННА МЕРЕЖА ЗІ ЗВОРОТНИМ ПОШИРЕННЯМ ПОХИБКИ (BACK PROPAGATION)

3.1 Мета роботи

Вивчення принципів функціонування нейронних мереж з прямим поширенням сигналу. Набуття практичних навичок навчання багатошарової нейронної мережі методом зворотного поширення похибки.

3.2 Вказівки з організації самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно: ознайомитися з особливостями функціонування і навчання багатосарових персептронів. Ознайомитися з методом зворотного поширення похибки й особливостями його реалізації. З цією метою може бути використаний лекційний матеріал з відповідних тем, матеріал, викладений у рекомендованій літературі [4, с. 26 – 50, 5 с. 29 – 47, 80 – 87, 6 с. 62-80], а також матеріал цих методичних вказівок.

Штучна нейронна мережа (artificial neural network) - це рівнобіжна система обробки інформації, яка складається з обробних елементів (нейронів), що локально виконують операції над сигналами і можуть мати локальну пам'ять. Кожний нейрон характеризується своїм поточним станом за аналогією з нервовими клітками головного мозку, що можуть бути збудженими або загальмованими. Він має групу синапсів - односпрямованих вхідних зв'язків, з'єднаних з виходами інших нейронів, а також має аксон - вихідний зв'язок даного нейрона, з якого сигнал надходить на синапси наступних нейронів. Схема формального штучного нейрона наведена на рис. 3.1. Кожний синапс характеризується величиною синаптичного зв'язку або його вагою w_i .

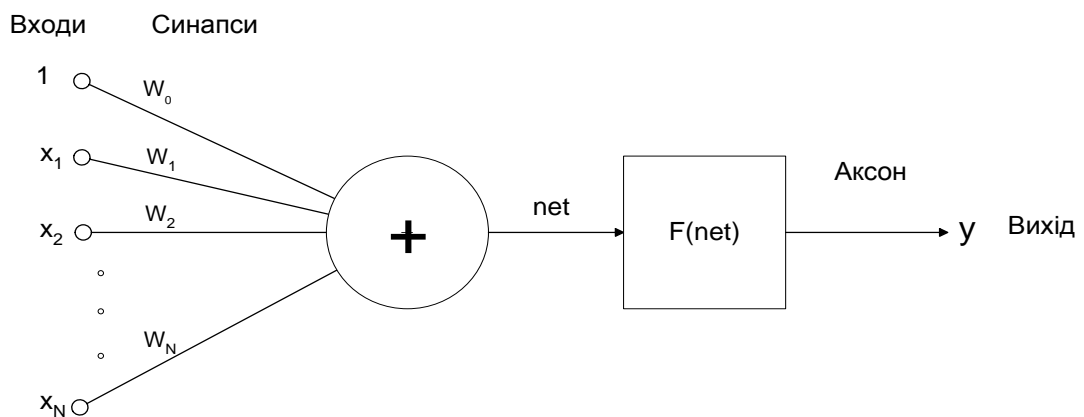


Рисунок 3.1 – Схема формального штучного нейрона

Поточний стан нейрона визначається, як зважена сума його входів:

$$\text{net} = \sum_{i=0}^N w_i x_i. \quad (3.1)$$

Вихід нейрона є функцією його стану:

$$y = f(\text{net}). \quad (3.2)$$

Нелінійна функція f називається активаційною і може мати різний вигляд. Однією з найбільш поширених є нелінійна функція з насиченням - логістична функція, або сигмоїд:

$$f(\text{net}) = \frac{1}{1 + e^{-\alpha \text{net}}} . \quad (3.3)$$

При зменшенні α сигмоїд стає більш пологим, при $\alpha=0$ вироджуючись у горизонтальну лінію на рівні 0.5, при збільшенні α сигмоїд наближається за зовнішнім виглядом до порогової функції. Логістична функція дуже зручна, тому що має просту похідну, що використовується при реалізації алгоритму зворотного поширення:

$$\frac{\partial f(\text{net})}{\partial \text{net}} = \alpha f(\text{net})(1 - f(\text{net})) .$$

З'єднані між собою нейрони утворюють штучну нейронну мережу (ШНМ), отже ШНМ - це пара (M, R) , де M - множина нейронів, R - множина зв'язків. Структура мережі задається у вигляді графа, у якому вершини є нейронами, а ребра - зв'язками.

У загальному випадку ШНМ складається з декількох шарів, серед яких обов'язково є вхідний, що одержує зовнішні сигнали, вихідний, що відображає реакцію нейронів на комбінації вхідних сигналів, і в багатошарових мережах - приховані. Структура багатошарової нейронної мережі наведена на рис. 3.2.

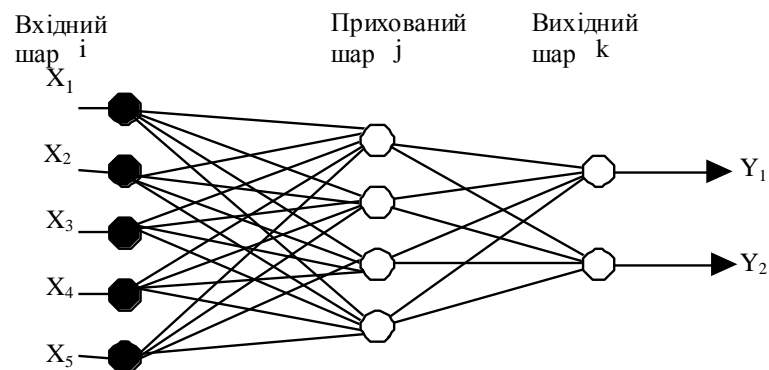


Рисунок 3.2 - Структура багатошарової нейронної мережі

Зв'язки між нейронами задаються у вигляді векторів і матриць. Ваги зручно подавати елементами матриці $W = [w_{ij}]$ розмірності $N \times M$, де N - кількість входів, M - кількість нейронів. Елемент w_{ij} відображає зв'язок між i -м і j -м нейронами, при цьому, якщо $w_{ij} = 0$, то зв'язок між i -м та j -м нейронами відсутній; $w_{ij} < 0$ - зв'язок, що гальмує сигнал; $w_{ij} > 0$ - зв'язок, що прискорює (збуджує) сигнал.

Відмітною властивістю ШНМ є можливість навчатися. Навченість нейронної мережі полягає у виробленні правильних реакцій на різні вхідні сигнали, що їй подаються. Існує багато можливостей навчання ШНМ. Найширше - отримав підхід, при якому структура мережі задається апріорно, а мережа навчається шляхом підстроювання вагових коефіцієнтів кожного нейрона. Від того, наскільки вдало підібрані вагові коефіцієнти, залежить ефективність даної мережі. У цьому випадку навчання полягає в зміні за визначеною процедурою елементів матриці W при послідовному поданні мережі деяких навчальних векторів.

У даний час існує велика кількість алгоритмів навчання. Для навчання багатошарових нейронних мереж застосовується алгоритм зворотного поширення помилки.

Основна ідея зворотного поширення полягає в тому, як одержати оцінку похибки для нейронів прихованих шарів. Відомі помилки, що робляться нейронами вихідного шару, виникають унаслідок невідомих помилок нейронів прихованих шарів. Чим більше значення синаптичного зв'язку між нейроном прихованого шару і вихідним нейроном, тим сильніше помилка першого впливає на помилку другого. Отже, оцінку помилки елементів прихованих шарів можна одержати, як зважену суму помилок наступних шарів.

Розглянемо алгоритм навчання мережі, що зображена на рис 3.2. Ця мережа має тільки один прихований шар. Матрицю вагових коефіцієнтів від входів до прихованого шару позначимо W , а матрицю вагових коефіцієнтів, що з'єднують прихований і вихідний шар - як V . Для індексів прийемо такі позначення: входи нумеруватимемо тільки індексом i , елементи прихованого шару - індексом j , а виходи, відповідно, індексом k .

Нехай мережа навчається на вибірці (X^a, Y^a) , $a = 1..p$ (a - образи, що подаються на вхід мережі). Активності нейронів позначатимемо малими буквами y (виходи) з відповідним індексом, а сумарні зважені входи нейронів - net .

Розглянемо алгоритм навчання більш докладно.

Крок 0. Початкові значення ваг усіх нейронів усіх шарів $V(t=0)$ і $W(t=0)$ покладаються випадковими числами.

Крок 1. Мережі подається вхідний образ X^a , у результаті формується вихідний образ $y \neq Y^a$. При цьому нейрони послідовно від шару до шару функціонують за такими формулами:

прихований шар

$$net_j = \sum_i W_{ij} X_i^a; y_j = f(net_j), \quad (3.4)$$

де $f(net) = \frac{1}{1 + e^{-net}}$ - активаційна функція нейронної мережі;

вихідний шар

$$\text{net}_k = \sum_j V_{jk} y_j; y_k = f(\text{net}_k). \quad (3.5)$$

Крок 2. Відповідно до методу найменших квадратів, цільовою функцією помилки нейронної мережі, яку треба мінімізувати, є величина:

$$E = \frac{1}{2} \sum_a \sum_k (y_k - Y_k^a)^2. \quad (3.6)$$

Функцію помилки мінімізуватимемо методом градієнтного спуску, що означає настройку вагових коефіцієнтів за формулою:

$$V_{jk}(t+1) = V_{jk}(t) - h \frac{\partial E}{\partial V_{jk}}, \quad (3.7)$$

де $V_{jk}(t+1)$ - ваговий коефіцієнт зв'язку, що з'єднує j -й нейрон прихованого шару $(n-1)$ з k -м нейроном вихідного шару (n) ; h - коефіцієнт швидкості навчання, $0 < h < 1$ (звичайно від 0,01 до 1,0).

Функція помилки в явному вигляді не містить залежності від ваги V_{jk} , тому скористаємося формулами неявного диференціювання складної функції:

$$\begin{aligned} \frac{\partial E}{\partial y_k} &= \delta_k = (y_k - Y_k^a); \\ \frac{\partial E}{\partial \text{net}_k} &= \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial \text{net}_k} = \delta_k \cdot y_k (1 - y_k); \\ \frac{\partial E}{\partial V_{jk}} &= \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial \text{net}_k} \cdot \frac{\partial \text{net}_k}{\partial V_{jk}} = \delta_k \cdot y_k (1 - y_k) \cdot y_j, \end{aligned} \quad (3.8)$$

де $\frac{\partial y}{\partial \text{net}} = y(1 - y)$ - похідна активаційної функції $f(\text{net})$, яка виражається тільки через значення функції.

Крок 3. На цьому кроці виконується підстроювання ваг прихованого шару. Градієнтний метод як і раніше дає:

$$W_{ij}(t+1) = W_{ij}(t) - h \frac{\partial E}{\partial W_{ij}}. \quad (3.9)$$

Обчислення похідних виконується за тими ж формулам, за винятком деякого ускладнення формули для помилки δ_j :

$$\begin{aligned}\frac{\partial E}{\partial \text{net}_k} &= \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial \text{net}_k} = \delta_k \cdot y_k(1 - y_k) \\ \frac{\partial E}{\partial y_j} &= \delta_j = \sum_k \frac{\partial E}{\partial \text{net}_k} \cdot \frac{\partial \text{net}_k}{\partial y_j} = \sum_k \delta_k \cdot y_k(1 - y_k) \cdot V_{jk}; \\ \frac{\partial E}{\partial W_{ij}} &= \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial \text{net}_j} \cdot \frac{\partial \text{net}_j}{\partial W_{ij}} = \delta_j \cdot y_j(1 - y_j) \cdot X_i^a.\end{aligned}\tag{3.10}$$

При обчисленні δ_j був застосований принцип зворотного поширення похибки: часткові похідні беруться тільки за перемінними наступного шару. За допомогою отриманих формул модифікуються вагові коефіцієнти нейронів прихованого шару. Якщо в мережі є кілька прихованих шарів, процедура зворотного поширення застосовується послідовно для кожного з них, починаючи із шару, що передуює вихідному, і далі до шару, що йде за вхідним. При цьому формули зберігають свій вигляд із заміною елементів вихідного шару на елементи відповідного прихованого шару.

Крок 4. Кроки 1-3 повторюються для всіх навчальних векторів. Навчання завершується після досягнення малої повної помилки або максимально припустимої кількості ітерацій.

3.3 Опис лабораторної установки

Як лабораторна установка використовується ПЕОМ типу IBM PC. Моделювання нейронної мережі здійснюється з використанням пакета ST Neural Networks системи STATISTICA.

У пакеті ST Neural Networks навчальна вибірка зберігається у вигляді набору даних (Data Set), що містить деяку кількість спостережень, для кожного з яких задані значення вхідних і вихідних перемінних. Як правило, дані беруться з якогось зовнішнього джерела (наприклад, системи STATISTICA або електронної таблиці). Однак новий набір даних можна створити прямо в пакеті ST Neural Networks. Для цього потрібно виконати такі дії.

1. Ввійти в діалогове вікно *Створити набір даних - Create Data Set* за допомогою команди *Набір даних - Data Set...* з меню *Файл - Створити - File - New*.

2. Ввести значення кількості вхідних (*Inputs*) і вихідних (*Outputs*) змінних у майбутньому наборі даних.

3. Натиснути кнопку *Створити - Create*.

При створенні нового набору даних програма ST Neural Networks відкриває вікно *Редактор даних - Data Set Editor*.

Основний елемент вікна *Редактор даних - Data Set Editor* - таблиця, що містить усі записи (спостереження) набору даних. Кожному спостереженню відповідає один рядок таблиці. У початковий момент таблиця міститиме один рядок, а значення усіх змінних будуть "невідомі" (позначені знаком питання).

У вхідних змінних заголовків стовпця чорного кольору, у вихідних - блакитного; входи від виходів відокремлюються темною вертикальною лінією.

Редагування даних здійснюється звичайним редагуванням цієї таблиці. Додавання нових спостережень здійснюється натисканням клавіші *СТРІЛКА ВНИЗ*. При спробі вийти вниз за межі таблиці, програма ST Neural Networks створює нове спостереження. Вилучення спостережень здійснюється натисканням клавіш *CTRL+X*.

У пакеті ST Neural Networks є можливість давати імена окремим спостереженням і змінним. Для цього в позначку відповідного стовпця або рядка необхідно ввести ім'я змінної або спостереження. Як позначення рядка за замовчуванням береться його номер.

У будь-якому місці таблиці можна вставити новий рядок або стовпець. Для цього необхідно помістити курсор миші на лінію, що розділяє позначки двох сусідніх рядків або стовпців (при цьому курсор перетвориться у двосторонню стрілку), і клацнути лівою кнопкою - відкриється смуга вставки. Після натискання клавіші *INSERT* буде вставлений новий рядок/стовпець.

Щоб призначити тип змінної - *Вхідна - Input*, *Вихідна - Output*, *Вхідна/Вихідна - Input/Output* або *Яка не враховується - Ignored*, виберіть змінну, потім натиснувши праву кнопку миші виберіть потрібний тип з контекстного меню.

Щоб задати тип підмножини - *Навчальне - Training*, *Контрольне - Verification*, *Тестове - Test* або *Що не враховується - Ignored*, виберіть спостереження, клацнувши на позначці відповідного рядка, і натиснувши праву кнопку миші вибирайте потрібний тип з контекстного меню.

Створити нову мережу в пакеті ST Neural Networks можна засобами діалогового вікна *Створити мережу - Create Network* (рис. 3.3), доступ до якого здійснюється через команду *Мережа... - Network...* меню *Файл - Створити - File - New*.

Для створення мережі необхідно.

1. Вибрати тип мережі зі списку *Type*.
2. Натиснути кнопку *Порада - Advise*. Програма ST Neural Networks встановить параметри для пре/пост-процесування і конфігурації мережі, виходячи з типу змінних, які складають вихідні дані.
3. Увести необхідні поправки у визначення змінних і специфікації шарів мережі.
4. Натиснути кнопку *Створити - Create*.

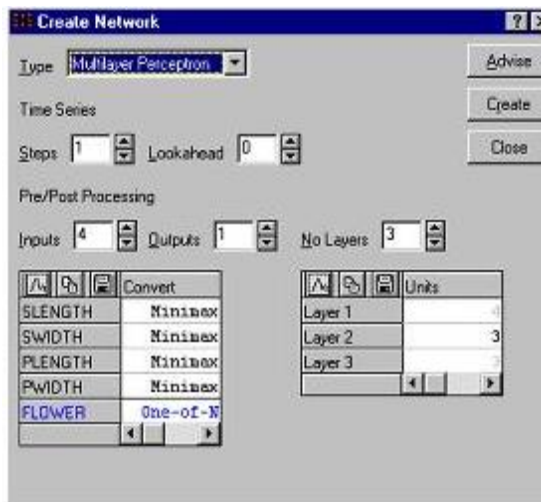


Рисунок 3.3 - Діалогове вікно «Створити мережу»

Навчання мережі.

У пакеті ST Neural Networks реалізовані основні алгоритми навчання багатоваріантних перцептронів: методи зворотного поширення похибки, сполучених градієнтів і Левенберга-Марквардта.

Для навчання мережі необхідно в меню *Train* вибрати тип мережі й у меню, що випадає, вибрати алгоритм навчання, наприклад, *Зворотне поширення - Back Propagation*. На екрані з'явиться діалогове вікно *Back Propagation* (рис. 3.4). При натисканні кнопки *Train* у правому верхньому куті вікна буде запущений алгоритм навчання.

Режим роботи алгоритму зворотного поширення залежить від ряду параметрів, і більшість з них зібрано в діалоговому вікні *Зворотне поширення*. Опишемо коротко найбільш важливі параметри.

1. *Epochs*. Задає кількість епох навчання, що проходять при одному натисканні клавіші *Train*.

2. *Learning rate* (швидкість навчання). При збільшенні швидкості навчання алгоритм працює швидше, але в деяких задачах це може призвести до нестійкості (особливо, якщо дані з шумами).

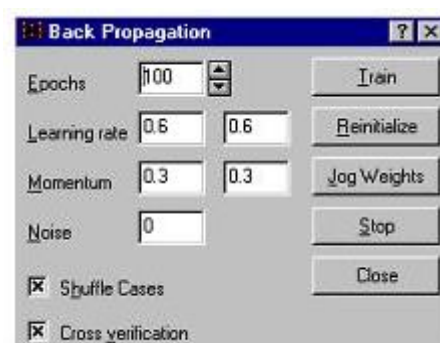


Рисунок 3.4 - Діалогове вікно «Зворотне поширення похибки»

3. *Momentum* (інерція). Цей параметр поліпшує (прискорює) навчання в ситуаціях, коли помилка мало змінюється, а також додає алгоритмові додаткову стійкість.

4. *Shuffle Cases* (перемішувати спостереження). При використанні цієї функції порядок, у якому спостереження подаються на вхід мережі, змінюється в кожній новій епосі.

5. *Cross Verification* (крос-перевірка). При використанні крос-перевірки мережа навчатиметься на множині навчальних спостережень і після кожної епохи перевірятиметься на множині контрольних спостережень. Цю функцію можна відключити, тоді програма ST Neural Networks просто навчатиме мережу на навчальній множині, цілком ігноруючи контрольну множину. Остаточну контрольну помилку можна буде обчислити засобами вікна *Прогнати набір даних - Run Data Set* (у ньому обчислюється також і помилка на тестовій множині).

Крім того, у пакеті ST Neural Networks є можливість змінювати швидкість навчання і/або коефіцієнт інерції від епохи до епохи, поступово зрушуючи їх від початкових значень, заданих у полях у лівій частині діалогового вікна *Зворотне поширення*, до їхніх кінцевих значень, заданих у правій частині вікна.

Для відображення на графіку помилки навчання необхідно відкрити вікно *Графік помилки навчання - Training Error Graph* за допомогою команди *Графік навчання... - Training Graph...* меню *Статистики - Statistics* (рис. 3.5).

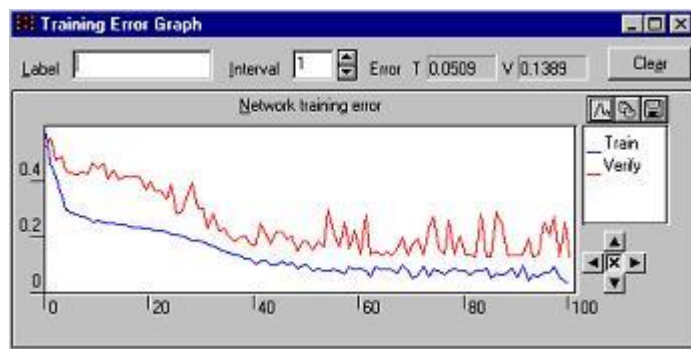


Рисунок 3.5 – Графік помилки навчання

Щоб порівняти результати роботи алгоритму в різних варіантах, необхідно скористатися кнопкою *Переустановити - Reinitialize* діалогового вікна *Зворотне поширення - Back Propagation*. У результаті вагові коефіцієнти мережі знову будуть установлені випадковим чином для початку наступного сеансу навчання. Якщо тепер після кнопки *Переустановити - Reinitialize* натиснути кнопку *Навчити - Train*, на графіку почне рисуватися нова лінія. Якщо в результаті таких дій графік стане занадто "засміченим", його можна очистити за допомогою кнопки *Очистити - Clear* вікна *Графік навчання - Training Graph*.

Умови припинення

Під час навчання мережі можна як умову припинення задавати число епох. Це цілком розумно, особливо при інтерактивному навчанні мережі, коли в будь-який момент можна натиснути на кнопку *Смин - Stop*, щоб перервати навчання, якщо виявиться, що щось йде не так, як треба.

Однак існують і більш зручні способи (особливо при тривалому навчанні) завдання умови припинення. У пакеті ST Neural Networks це робиться у вікні *Умови припинення - Stopping Conditions*, яке зображено на рисунку 3.6 (доступ через пункт *Умови припинення - Stopping Conditions...* меню *Навчання Додатково - Train - Auxiliary*). Тут крім максимального числа епох (*Epochs*), можна задати рівень припустимої помилки (*Target Error*), при досягненні якого навчання має припинятися, або величину мінімального поліпшення (*Minimum Improvement*) помилки за фіксовану кількість епох.



Рисунок 3.6 – Вікно «Умови припинення»

Можливо, найкорисніший з цих параметрів - величина мінімального поліпшення (*Minimum Improvement*). Зміст її такий: якщо протягом заданої кількості епох (параметр *Вікно - Window*) помилка навчання і контрольна помилка не покращилися на цю величину, навчання припиняється.

Наприклад, якщо задати параметр *Вікно - Window* рівним одній епосі, повторно ініціалізувати і навчити мережу і узяти величини мінімального поліпшення рівними нулеві, програма ST Neural Networks зупинить навчання, як тільки помилка навчання або контрольна помилка почнуть зростати.

Одна з труднощів, що зустрічається при такому підході, полягає в тому, що в процесі навчання помилка може плавати вверх-вниз. Це можна врахувати за допомогою параметра *Вікно - Window*, так щоб навчання зупинялося тільки в тому випадку, якщо результати незадовільні на декількох епохах підряд. Наприклад, якщо задати "вікно" у п'ять епох, то навчання припиниться тільки в тому випадку, коли помилка погіршилася, а потім протягом п'яти епох не досягла свого попереднього найкращого значення.

Запуск нейронної мережі

Після того, як мережа навчена, її можна запустити на виконання. У пакеті ST Neural Networks це можна зробити в декількох варіантах:

- на поточному наборі даних - у цілому або на окремих спостереженнях;
- на іншому наборі даних - у цілому або на окремих спостереженнях (такий набір даних уже може не містити вихідних значень і призначатися винятково для тестування);

- на одному конкретному спостереженні, для якого значення змінних уведені користувачем, а не узяті з якогось файлу даних;
- з іншого програмного засобу за допомогою інтерфейсу прикладного програмування SNN API.

Під час запуску мережі на поточному наборі даних можливі два варіанти: або обробляти окремі спостереження, або всю множину цілком. У другому варіанті підраховуються сумарні статистики.

Для обробки окремих спостережень з набору даних служить вікно *Прогнати одне спостереження - Run Single Case* (рис. 3.7), доступ до якого здійснюється через пункт *Одне спостереження - Single Case ..* меню *Запуск - Run*.

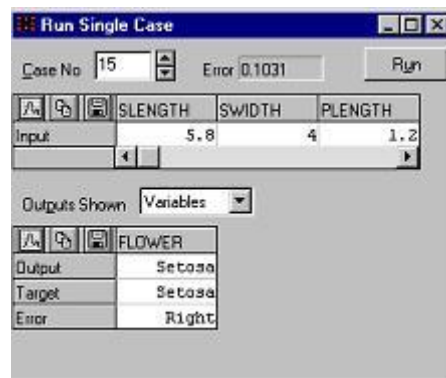


Рисунок 3.7 - Вікно «Прогнати одне спостереження»

В полі *Номер спостереження - Case No* задається номер спостереження, що підлягає обробці. Щоб обробити поточне спостереження, необхідно натиснути кнопку *Запуск - Run*, а для обробки будь-якого іншого спостереження - увести відповідний номер у поле *Номер спостереження - Case No* і натиснути клавішу *ВВЕДЕННЯ*.

Значення вхідних змінних для поточного спостереження відображаються у таблиці, що розташована у верхній частині вікна, а вихідні значення - у нижній таблиці.

Крім фактичного вихідного значення, що видає мережа, виводиться також цільове значення і помилка, тобто різниця між першим і другим.

Для тестування мережі на всьому наборі даних служить вікно *Прогнати набір даних - Run Data Set*, доступ до якого здійснюється через пункт *Набір даних - Data Set...* меню *Запуск - Run*. У таблиці вікна *Прогнати набір даних - Run Data Set* утримуються такі значення: фактичні виходи мережі, цільові вихідні значення, помилки, тобто різниці між першим і другими, і сумарна помилка за кожним спостереженням.

Іноді необхідно протестувати мережу на окремому спостереженні, що не належить ніякому набору даних.

Тестування заданих користувачем спостережень проводиться з вікна *Прогнати окреме спостереження - Run One-off Case* (рис.3.8), доступ до якого здійснюється через пункт *Окреме - One-off...* меню *Запуск -Run*.

Для цього потрібно увести вхідні значення в таблицю, розташовану у верхній частині вікна, і натиснути кнопку *Запуск - Run*, результати будуть виведені в нижню таблицю.



Рисунок 3.8 - Вікно «Прогнати окреме спостереження»

Відновлення найкращої мережі

Навіть у тому випадку, коли використовуються умови припинення, що переривають навчання кожний раз, як тільки результати починають погіршуватися, залишається проблема того, що найкраще настроювання мережі, виявлене програмою ST Neural Networks, буде затерто при наступному навчанні.

У пакеті ST Neural Networks є вікно *Зберегти кращу мережу - Retain Best Network*, за допомогою якого можна зберегти кращу з мереж, отриманих у процесі навчання. Доступ до вікна здійснюється за допомогою команди *Краща мережа - Best Network...* меню *Навчання - Додатково - Tram - Auxiliary*.

Програма ST Neural Networks автоматично запам'ятовує кращу з отриманих мереж, причому не тільки на одному прогоні навчання, але і на декількох прогонах однієї і тієї ж або різних мереж. Завдяки цьому можна експериментувати з мережами, будучи упевненим, що найкращий досягнутий результат буде завжди доступний. Для того, щоб відновити найкращу мережу з отриманих на даний момент, потрібно натиснути кнопку *Відновити - Restore*.

3.4 Порядок виконання роботи

Ознайомитися за допомогою викладача з особливостями і режимами роботи використовуваних комп'ютерних засобів та прикладних програм.

Отримати у викладача варіант завдання та додаткові дані. Варіанти завдань наведені в табл. 3.1.

Створити набір даних. Розбити набір даних на навчальну множину, яка використовуватиметься при навчанні мережі і контрольну множину - для перевірки якості її роботи. Перемішати навчальні і контрольні спостереження (меню: *Edit - Cases - Shuffle*).

Створити нову мережу. Тип мережі - багатошаровий перцептрон. Навчити нейронну мережу методом зворотного поширення похибки.

Запустити нейронну мережу на поточному наборі даних і на іншому наборі даних (цей набір даних може не містити вихідних значень, і призначений

винятково для тестування). Проаналізувати результати роботи мережі.

Таблиця 3.1 – Варіанти завдань

Но- мер	Функція	Но- мер	Функція
1	$f(x_1, x_2) = x_1^2 - x_1x_2 + x_2^2$	14	$f(x_1, x_2) = x_1^2 + 2x_2^2 - 2x_1 + 4x_2 + 1$
2	$f(x_1, x_2) = x_1^2 - \cos x_1x_2 - x_2^2$	15	$f(x_1, x_2) = 2x_1^2 + x_2^2 + 2x_1x_2 - x_1$
3	$f(x_1, x_2) = x_1^2 - 2x_1x_2 + 2x_2^2 + 2x_1$	16	$f(x_1, x_2) = x_1x_2 \cdot (1 - x_1 - x_2)$
4	$f(x_1, x_2) = x_1^3 + x_2^3 - x_1^2 - 2x_1x_2$	17	$f(x_1, x_2) = 2x_1^2/x_2 - 4x_1 + 2x_2^2$
5	$f(x_1, x_2) = x_1^3 - 2x_2^3 - 3x_1 + 6x_2$	18	$f(x_1, x_2) = \exp(-x_1^2 - x_2) \cdot (x_1 + x_2)$
6	$f(x_1, x_2) = x_1^3 - 2x_1^2x_2^2 + x_2^2$	19	$f(x_1, x_2) = 100\sqrt{x_2 - x_1^2} + \sqrt{1 - x_1}$
7	$f(x_1, x_2) = x_1x_2 + \frac{1}{2(x_1 + x_2)}$	20	$f(x_1, x_2) = 2x_1 + x_1^2 - x_2^2$
8	$f(x_1, x_2) = \exp(x_1 + 2x_2) \cdot x_1^2 - x_2^2$	21	$f(x_1, x_2) = \sin x_1 + \sin x_2$
9	$f(x_1, x_2) = \exp(x_1 - x_2) \cdot (x_1^2 - 2x_1x_2 - 2x_2^2)$	22	$f(x_1, x_2) = e^{x_1} + \sin x_2$
10	$f(x_1, x_2) = \exp(-x_1^2 - x_2^2) \cdot (x_1^2 + 2x_2^2)$	23	$f(x_1, x_2) = \cos x_1 + \cos x_2$
11	$f(x_1, x_2) = \exp(-x_1^2 - x_2^2) \cdot (x_1 - 2x_2)$	24	$f(x_1, x_2) = 2^{x_1} + \cos x_2$
12	$f(x_1, x_2) = x_1x_2 \cdot \ln(x_1^2 + x_2^2)$	25	$f(x_1, x_2) = \sin x_2^2 + x_1^2$
13	$f(x_1, x_2) = \frac{x_1}{x_2} + \frac{1}{x_1 + x_2}$	26	$f(x_1, x_2) = \cos x_1x_2 + \frac{2x_1}{x_2}$

Підібрати архітектуру мережі і параметри алгоритму навчання так, щоб контрольна помилка мережі не перевищувала 0,2.

3.5 Зміст звіту

Звіт має містити:

- титульний аркуш;
- мету роботи;
- постановку і вихідні дані задачі;
- стислий опис алгоритму навчання мережі;
- значення вагових коефіцієнтів та порогових значень мережі, які установилися після навчання;
- графіки залежності погрішності навчання і погрішності тестування від кількості циклів навчання;
- висновки по роботі.

3.6 Контрольні запитання та завдання

1. Опишіть будову та алгоритм функціонування перцептрона.
2. Назвіть найпоширеніші види активаційних функцій.
3. У чому складається суть методу зворотного поширення похибки?

4. За яким оптимізаційним методом функціонує алгоритм зворотного поширення похибки?
5. Для розв'язання яких задач використовується метод зворотного поширення похибки?
6. Які переваги та недоліки має back propagation?
7. Чим відрізняється навчання з учителем від навчання без учителя?
8. Які ще методи навчання нейронних мереж ви знаєте?
9. Які властивості повинна мати функція активації при використанні алгоритму зворотного поширення похибки?
10. Чи може порядок подання образів у навчальній вибірці впливати на якість навчання?

4 МОДЕЛІ НЕЙРОННИХ МЕРЕЖ ДЛЯ АВТОАСОЦІАЦІЇ ОБРАЗІВ

4.1 Мета роботи

Вивчення особливостей структурної організації мереж Хопфілда і Хемминга та алгоритмів їхнього навчання. Набуття навичок розв'язання задачі розпізнавання образів за допомогою мереж Хопфілда і Хемминга.

4.2 Вказівки з організації самостійної роботи

Під час підготовки до виконання лабораторної роботи необхідно: ознайомитися з особливостями структурної організації моделей нейронних мереж Хопфілда і Хемминга, алгоритмами їхнього навчання, з'ясувати принципи функціонування цих моделей у процесі автоасоціації образів.

З цією метою може бути використаний лекційний матеріал з відповідних тем, матеріал, викладений у рекомендованій літературі [5 с. 116-122, 133-137], а також матеріал цих методичних вказівок.

Структурна схема мережі Хопфілда зображена на рис. 4.1. Вона складається з єдиного шару нейронів, кількість яких є одночасно кількістю входів та виходів мережі. Кожний нейрон зв'язаний синапсами з усіма іншими нейронами, а також має один вхідний синапс, через який здійснюється введення сигналу. Вихідні сигнали утворюються на аксонах. Силу зв'язку від i -го до j -го нейрона позначають як w_{ij} . У моделі Хопфілда передбачається умова симетричності зв'язків $w_{ij} = w_{ji}$ з нульовими діагональними елементами $w_{ij} = 0$.

Мережа Хопфілда є рекурентною у тому розумінні, що для кожного вхідного зразка вихід мережі повторно використовується як вхід доти, поки не буде досягнуто стану стійкості.

Задача, розв'язувана даною мережею у якості асоціативної пам'яті, як правило, формулюється таким чином. Відомий деякий набір двоїчних сигналів (зображень), що вважаються зразковими. Мережа повинна вміти з довільного неідеального сигналу, поданого на її вхід, виділити ("згадати" за

частковою інформацією) відповідний зразок (якщо такий є) або "дати висновок" про те, що входні дані не відповідають жодному зі зразків.

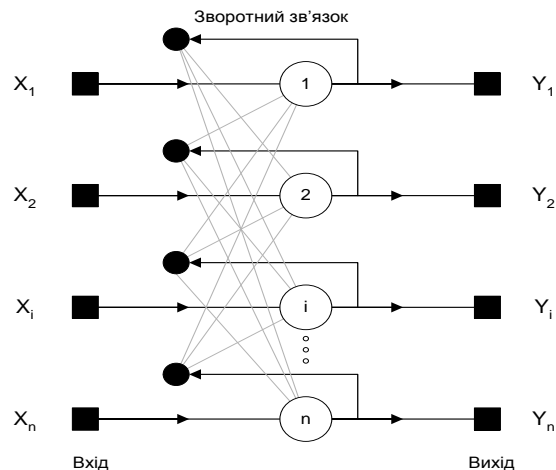


Рисунок 4.1 - Структурна схема мережі Хопфілда

Мережа функціонує в такий спосіб. Кожний нейрон першого шару обчислює зважену суму своїх входів, утворюючи сигнал net , який потім за допомогою нелінійної функції F перетворюється у сигнал Y . Як активаційна функція використовується гранична функція.

Зміна стану кожного нейрона s_j у моделі Хопфілда відбувається за відомим правилом для формальних нейронів. Сигнали s_i , що надходять на входи нейронів, в момент часу t зважуються з ваговими коефіцієнтами w_{ij} і підсумовуються, визначаючи повний рівень сили вхідного сигналу:

$$net_j = \sum_{i \neq j} w_{ij} s_i + T_j. \quad (4.1)$$

Далі в момент $t+1$ нейрон змінює свій стан в залежності від рівня сигналу net та індивідуального порога кожного нейрона T :

$$\begin{cases} s_j(t+1) = -1, net_j(t) < T_j; \\ s_j(t+1) = 1, net_j(t) > T_j; \\ s_j(t+1) = s_j(t), net_j = T_j. \end{cases} \quad (4.2)$$

Вхідний вектор задає початкові стани всіх елементів. Елемент для відновлення вибирається випадково. Обраний елемент одержує зважені сигнали від усіх інших елементів і змінює свій стан. Вибирається інший елемент, і процес повторюється. Мережа досягає граничного стану, коли жоден з її елементів, який обрано для відновлення, не змінює свого стану, тобто досягається стійкий стан.

Навчання НС Хопфілда відбувається відповідно до правила Хебба:

$$w_{ij} = \sum_{k=1}^m x_i^k \cdot x_j^k, \text{ якщо } i \neq j \text{ інакше } w_{ij} = 0, \quad (4.3)$$

де k – кількість образів навчальної вибірки.

Коли немає необхідності, щоб мережа в явному вигляді видавала зразок, тобто досить, скажемо, одержувати номер зразка, асоціативну пам'ять успішно реалізує мережа Хемминга. Ця мережа характеризується, у порівнянні з мережею Хопфілда, меншими витратами на пам'ять і обсягом обчислень, що стає очевидним з її структури (рис. 4.2).

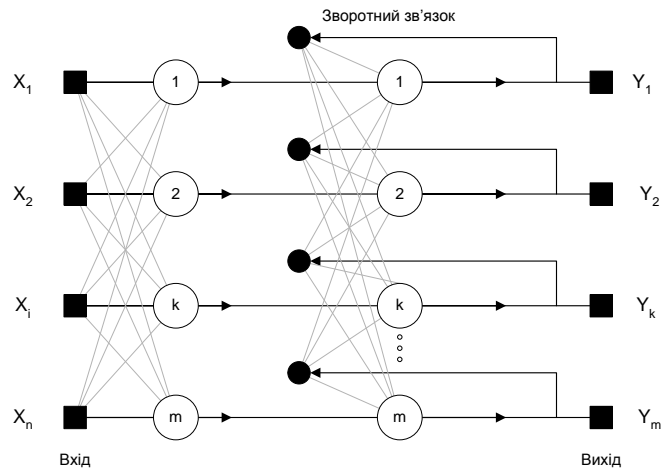


Рисунок 4.2 - Структурна схема мережі Хемминга

Мережа складається з двох шарів. Перший і другий шари мають по m нейронів, де m - число зразків. Нейрони першого шару мають по n синапсів, з'єднаних із входами мережі. Нейрони другого шару зв'язані між собою негативними зворотними синаптичними зв'язками. Єдиний синапс із позитивним зворотним зв'язком для кожного нейрона з'єднаний з його ж аксоном.

Ідея роботи мережі полягає у визначенні відстані Хемминга від зразка, що тестується до всіх зразків. Відстанню Хемминга називається кількість бітів якою відрізняються два бінарних вектори. Мережа має вибрати зразок з мінімальною відстанню Хемминга до невідомого вхідного сигналу, у результаті чого буде активізований тільки один вихід мережі, який відповідає цьому зразкові.

На стадії ініціалізації ваговим коефіцієнтам першого шару і порогові активаційної функції привласнюються такі значення:

$$w_{ik} = \frac{x_i^k}{2}, i = 1 \dots n, k = 1 \dots m, \quad (4.4)$$

де x_i^k - i -й елемент k -го зразка.

$$T_k = n/2, k = 1 \dots m. \quad (4.5)$$

Вагові коефіцієнти гальмуючих синапсів у другому шарі беруть рівними деякій величині $0 < \varepsilon < 1/m$. Синапс нейрона, зв'язаний з його ж аксоном має вагу +1.

Алгоритм функціонування мережі Хемминга:

1. На входи мережі подається невідомий вектор $X = \{x_i : i = 1 \dots n\}$, виходячи з якого розраховуються стани нейронів першого шару (верхній індекс у дужках указує номер шару):

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=1}^n w_{ij} x_i + T_j, j = 1 \dots m. \quad (4.6)$$

Після цього отриманими значеннями ініціалізуються значення аксонів другого шару:

$$y_j^{(2)} = y_j^{(1)}, j = 1 \dots m. \quad (4.7)$$

2. Обчислити нові стани нейронів другого шару:

$$s_j^{(2)}(t+1) = y_j(t) - \varepsilon \sum_{k=1}^m y_k^{(2)}(t), k \neq j, j = 1 \dots m, \quad (4.8)$$

і значення їхніх аксонів:

$$y_j^{(2)}(t+1) = f(s_j^{(2)}(t+1)), j = 1 \dots m. \quad (4.9)$$

Активаційна функція f має вигляд, що наведений на рис 4.3, причому величина F має бути досить великою, щоб будь-які можливі значення аргументу не приводили до насичення.

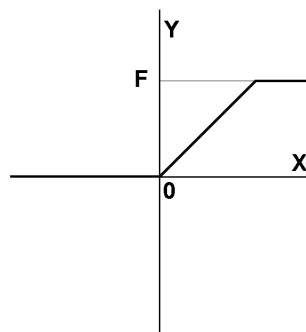


Рисунок 4.3 - Активаційна функція мережі Хемминга

3. Перевірити, чи змінилися виходи нейронів другого шару за останню ітерацію. Якщо так - перейти до кроку 2. Інакше – кінець обчислень.

З оцінки алгоритму видно, що роль першого шару досить умовна: скориставшись один раз на кроці 1 значеннями його вагових коефіцієнтів, мережа більше не звертається до нього.

4.3 Опис лабораторної установки

Як лабораторна установка використовується ПЕОМ типу IBM PC. Задача розпізнавання образів вирішується за допомогою програми NeuralNets.

4.4 Порядок виконання роботи

Ознайомитися за допомогою викладача з особливостями і режимами роботи використовуваних комп'ютерних засобів і програм.

Створити образи навчальної вибірки.

Протестувати мережу на прикладі образів навчальної вибірки.

Увести перекручені образи і проаналізувати можливості мереж за їхнім розпізнаванням.

Зобразити структуру мереж Хопфілда і Хемминга для розглянутої навчальної вибірки.

Зробити висновки, оформити і здати звіт про виконану роботу.

4.5 Зміст звіту

Звіт має містити:

- титульний аркуш;
- мету роботи;
- постановку і вихідні дані задачі;
- результати розв'язання задачі розпізнавання образів за допомогою нейронних мереж Хопфілда та Хемминга;
- структуру мереж Хопфілда і Хемминга для розглянутої навчальної вибірки;
- аналіз отриманих результатів та висновки по роботі.

4.6 Контрольні запитання та завдання

1. Для розв'язання яких задач використовуються мережі Хопфілда і Хемминга?
2. Які ще моделі нейронних мереж ви знаєте?
3. Опишіть принцип функціонування мережі Хопфілда.
4. Опишіть принцип функціонування мережі Хемминга.
5. Опишіть алгоритм навчання мережі Хопфілда.

6. Опишіть алгоритм навчання мережі Хеммінга.
7. Які недоліки мають моделі Хопфілда і Хеммінга?

ПЕРЕЛІК ПОСИЛАНЬ

1. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности / Г.К. Вороновский, К.В. Махотило, С.Н. Петрашев, С.А. Сергеев. – Х.: ОСНОВА, 1997. – 112 с.
2. Петров Э. Г., Писклакова В. П., Бескоровайный В. В. Территориально распределенные системы обслуживания. – К. Техніка, 1992. – 208с.
3. Автоматизация управления транспортными системами / А.П.Артынов, В.Н.Ембулаев, А.В.Пупышев и др.; Под ред. С.В.Емельянова. – М.: Наука, 1984.- 272 с.
4. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. – М.: Мир, 1992. – 184 с.
5. Руденко О.Г., Бодянский Е.В. Основы теории искусственных нейронных сетей. – Харьков: ТЕЛІЕТЕХ, 2002. – 317 с.
6. Нейронные сети. Statistica Neural Networks: Пер. с англ. – М.: Горячая линия – Телеком, 2001. – 182 с.

ЗМІСТ

Вступ	3
1 Оптимізація складних функцій за допомогою генетичних алгоритмів	4
2 Синтез топологічних структур ТРСО за допомогою генетичних алгоритмів	8
3 Нейронна мережа зі зворотним поширенням похибки	12
4 Моделі нейронних мереж для автоасоціації образів	25
Перелік посилань	30