МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ІНЖЕНЕРНО-ТЕХНОЛОГІЧНИЙ ІНСТИТУТ

Затверджено
на засіданні кафедри комп'ютерних технологій
протокол № ______ від "____"____________
Тираж 100 прим.

Вимогам, що ставляться до
навчально-методичних видань,
відповідає
Зав. кафедри __________________А.А.Тимченко

Методичні вказівки

до лабораторних робіт
з курсу

ІНТЕЛЕКТУАЛЬНІ ІНФОРМАЦІЙНІ СИСТЕМИ
для студентів спеціальностей 7.080401, 7.080403
всіх форм навчання

Весь цифровий і фактичний матеріал та бібліографічні
відомості перевірено. Зауваження рецензента враховано

Зав. кафедри ________________________________
Укладач: ___________________________________
Відповідальний редактор_____________________
Рецензент __________________________________

Черкаси ЧІТІ 2001 рік

Вступ. Сьогодні технології автоматизованого інтелектуального аналізу даних переживають бурхливий розквіт. Це пов’язано головним чином з потоком нових ідей, що починаються в області комп’ютерних наук, яка утворилася на перетині штучного інтелекту, статистики і теорії баз даних. Елементи автоматичної обробки і аналізу даних стають невід’ємною частиною концепції електронних сховищ даних і часто називаються в цьому контексті data mining (одержання знань із даних).

Комп’ютерні системи підтримки прийняття рішень базуються на двох підходах. Перший, більш традиційний, полягає у тому, що в системі фіксується досвід експерта, який і використовується для одержання оптимального в даній ситуації рішення. Для другого підходу характерне знаходження розв’язку на основі аналізу історичних даних, що описують поведінку об’єкта, прийняті в минулому рішення, їх результати і т.п. Впровадження таких систем на Україні наштовхується на завади, головні з яких – порівняно невеликий строк існування підприємств та нестабільність економіки. Інформації, що накопилася за цей час, недостатньо для вироблення на її основі ефективної стратегії прийняття рішень за допомогою систем data mining. Названі фактори в значній мірі визначають тенденції до розробки та застосування інформаційних інтелектуальних систем.

Структура лабораторних робіт відображає сучасні тенденції у використанні нових методів зменшення інформаційної невизначеності у процесі прийняття рішень. Вона включає у себе такі елементи, як нейронні мережі, генетичні алгоритми, метод групового врахування алгоритмів та експертні системи.

Лабораторна робота має 10 варіантів. Номер варіанта студента співпадає із його номером в журналі групи.

ЛАБОРАТОРНА РОБОТА №1

Тема: *Бінарний перцептрон*
Мета: *Вивчити принципи функціонування і навчання перцептрона.*

Короткі теоретичні відомості. Нейронна мережа (Neural network) є сукупністю простих елементів (units), які функціонально базуються на нейронах. Біологічний нейрон зображений на рис. 1.1. Сигнали між нейронами передаються вздовж аксонів. Сигнали, які підсилюються або послаблюються синапсами, нейрон одержує через дендріти, які підсилюються, або послаблюються синапсами. Штучні нейрони, відомі як Threshold Logic Unit (TLU) і запропоновані Маккалоком і Піттсом (McCulloch and Pitts) у 1943 році. Нехай маємо вектори множину входів $X = \{x_1, x_2, ..., x_n\}$ і вагових коефіцієнтів $W = \{w_1, w_2, ..., w_n\}$. Входи – сиг-

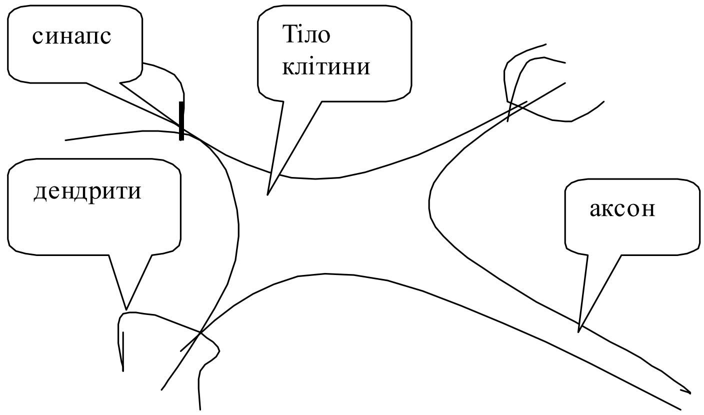


Рис.1.1. Біологічний нейрон

нали набувають значень лише “0” або “1”. Активацію a розраховуємо за формулою

$$a = \sum w_i x_i = (w, x). \qquad (1.1)$$

Вихід Y обчислюється за формулою

$$Y = \begin{cases} 1, \ \text{якщо} \ a \geq \theta, \\ 0, \ \text{якщо} \ a < \theta, \end{cases} \qquad (1.2)$$

де θ - порогове значення (threshold) найчастіше буває нулем (рис. 1.2). Більш загальний елемент нейронної мережі був введений Розенблаттом у 1962 році і названий перцептроном. Він складається із TLU, на входи яких

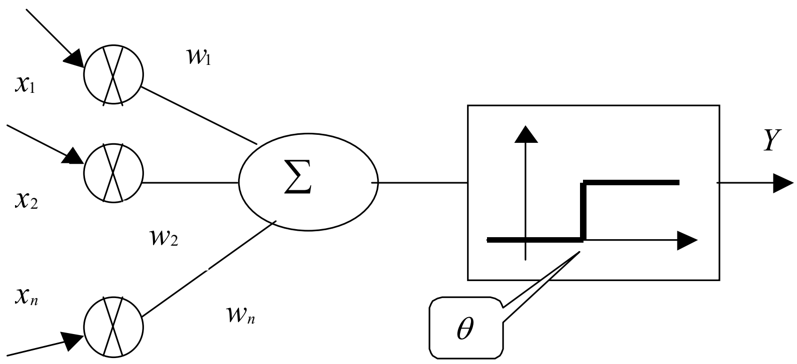


Рис. 1.2. Штучний нейрон

поступає множина входів асоціативних елементів (рис. 3).

Алгоритм навчання перцептрона.

1. Подати на вхід образ X .
2. Кожна компонента $X = \{x_1, x_2, ..., x_n\}$ множиться на відповідну компоненту

вектора вагових коефіцієнтів $W = \{w_1, w_2, ..., w_n\}$. Знаходимо суму цих добутків

$$a = XW = \sum_{i=1}^{n} x_i w_i \ . \qquad (1.3.)$$

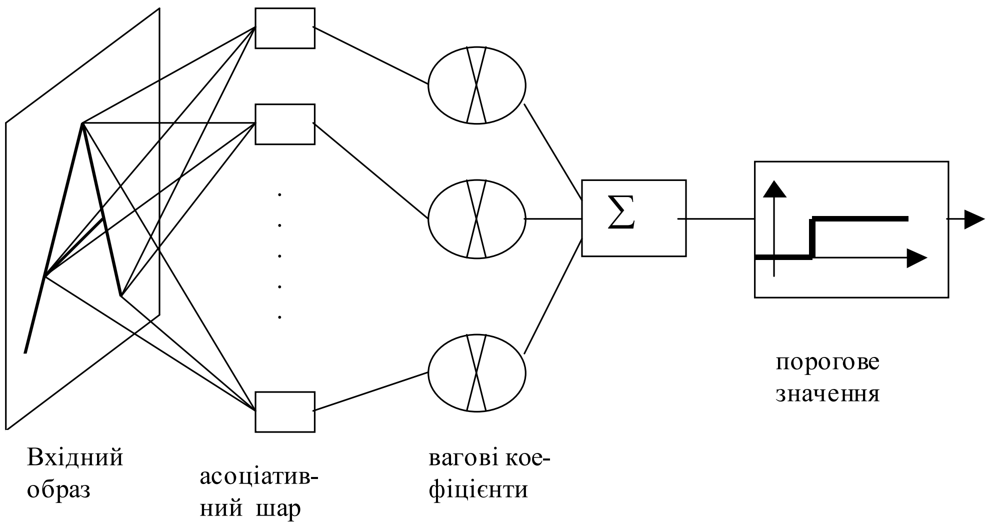


Рис. 1.3. Бінарний перцептрон

3. Якщо a перевищує порогове значення θ ($a > \theta$), то вихід нейрона Y дорівнює одиниці, в іншому випадку він – нуль. Якщо значення Y вірне, то нічого не змінюється і переходимо до кроку 1 у випадку, якщо на вході є інші образи. Якщо ні, то кінець. Якщо значення Y не вірне, то підраховуємо різницю між обчисленим значенням Y і правильним значенням T

$$\delta = T - Y. \qquad (1.4.)$$

4. Обчислюємо

$$\Delta_i = \eta \delta x_i, \qquad (1.5.)$$

де η – коефіцієнт швидкості навчання, який знаходиться, в загальному випадку, в проміжку (0;0,1).

5. Знаходимо

$$w_i(n+1) = w_i(n) + \Delta_i,\ i = \overline{1,n}, \qquad (1.6)$$

де $w_i(n+1)$ - значення i-го вагового коефіцієнта після корекції, $w_i(n)$- до корекції.

6. Якщо на вході є інші образи, то переходимо до кроку 1, якщо ні – то кінець.

Завдання до лабораторної роботи.

1. Програмно реалізувати наведений алгоритм.
2. Використовуючи дані таблиці 1.1 і результат пункту 1, побудувати таблицю 1.2.

Алгоритм реалізувати таким чином: на вхід поступово подати всі образи, коригуючи паралельно вагові коефіцієнти. Якщо для всіх образів $\delta = 0$, то кінець, якщо ні, то на вхід знову ж таки подати всі образи, але із скоригованими ваговими коефіцієнтами.

Таблиця 1.1 - Початкові дані

№ варіанта	Логічна функція	Можливі значення коефіцієнта η		
1	$x_1 \vee x_2 \wedge x_3$	0,2	0,4	0,6
2	$(x_1 \wedge x_2) \vee x_3$	0,2	0,04	0,08
3	$(x_1 \to x_2) \vee x_3$	0,05	0,1	0,3
4	$x_1 \vee x_2 \vee x_3$	0,07	0,1	0,3
5	$(\mid x_1 \vee x_2) \wedge x_3$	0,02	0,05	0,4
6	$x_1 \to (x_2 \vee x_3)$	0,1	0,5	0,7
7	$x_1 \wedge (x_2 \to x_3)$	0,2	0,02	0,002
8	$\mid x_1 \wedge (x_2 \to x_3)$	0,3	0,4	0,5
9	$x_1 \vee (\mid x_2 \to x_3)$	0,7	0,01	0,06
10	$(x_1 \wedge \mid x_2) \to x_3$	0,3	0,01	0,02

Таблиця 1.2 - Результати

w_1	w_2	w_3	θ	x_1	x_2	x_3	a	Y	T	$\eta(T-Y$	δw_1	δw_2	δw_3	$\delta\theta$

Контрольні запитання

1. Коротка історична довідка про розвиток теорії нейронних мереж.
2. Будова біологічного нейрона.
3. Штучний нейрон та активаційні функції.
4. Будова та алгоритм функціонування перцептрона.
5. Проблема лінійного поділу досліджуваної області.

ЛАБОРАТОРНА РОБОТА № 2

Тема: *Нейронна мережа із зворотнім поширенням похибки (back propagation)*

Мета: *Вивчити один із методів функціонування нейронних мереж і принципи його застосування.*

Короткі теоретичні відомості. Нейронна мережа має вигляд (рис. 2.1): Кожен шар нейронної мережі має різну кількість нейронів: **A - p, S - l, R - k.** Співвідношення між **p**, **l** і **k** визначимо пізніше. Відомо, що гіперболічний тангенс обчислюють за формулою:

$$y = thx = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{2.1}$$

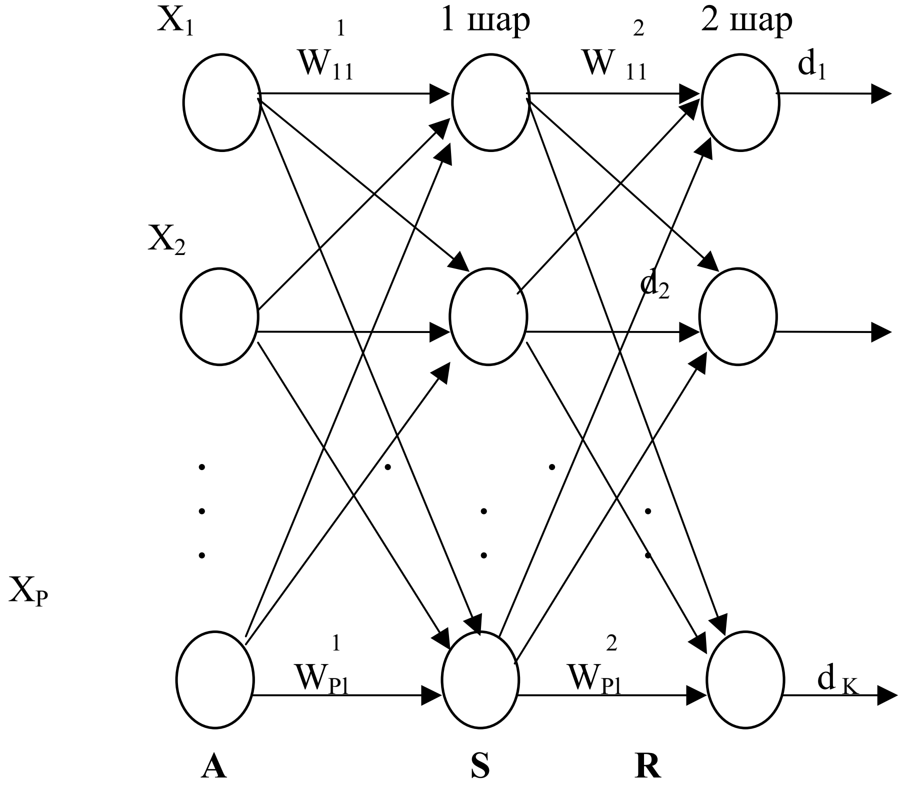


Рис. 2.1. Структура нейронної мережі

Обчислимо $th' x = \dfrac{1}{ch^2 x}$, де chx – гіперболічний косинус. Очевидно, що $(thx)' = 1 - (thx)^2$. В якості початкових даних маємо таблицю 2.1.

Таблиця 2.1 - Початкові дані для навчання нейронної мережі

X_1	x_1^1	x_1^2	x_1^3	....	x_1^N
X_2	x_2^1	x_2^2	x_2^3	....	x_2^N
. . .	. . .	. . .	. . .	....	. . .
X_p	x_p^1	x_p^2	x_p^3	....	x_p^N
d_1	d_1^1	d_1^2	d_1^3	....	d_1^N
d_2	d_2^1	d_2^2	d_2^3	....	d_2^N
. . .	. . .	. . .	. . .	....	. . .
d_k	d_k^1	d_k^2	d_k^3	....	d_k^N

N - кількість точок спостереження, одержаних в результаті експерименту або статистичних. Вектор $\vec{X}$ - входи, $\vec{D}$ - бажані або реальні виходи. $\vec{X} = (X_1, X_2, ..., X_p)$, $D = (d_1, \vec{d}_2, ..., d_k)$. Цільова функція, яку необхідно мінімізувати :

$$E(\omega) = \frac{1}{2}\left[\sum_{k=1}^{N}\sum_{j=1}^{l}(y_{kj}^1 - d_{kj}^1) + \sum_{k=1}^{N}\sum_{j=1}^{k}(y_{kj}^2 - d_{kj}^2)\right]^2 = \frac{1}{2}\sum_{k=1}^{N}\Big(\sum_{j=1}^{l}(y_{kj}^1 - d_{kj}^1) + \sum_{j=1}^{k}(y_{kj}^2 - d_{kj}^2)\Big)^2 . \quad (2.2)$$

Очевидно, що значення d^1_{kj} (задані виходи першого шару) невідомі. Тому обмежимося виходами шару R. Формула (2.2) трансформується в (2.3) :

$$E(\omega) = \frac{1}{2}\sum_{k=1}^{N}\sum_{j=1}^{k}(y_{kj}^2 - d_{kj})^2 . \qquad (2.3)$$

Маємо задачу :

$$E(\omega) \to \min . \qquad (2.4)$$

Функцію будемо мінімізувати методом градієнтного спуску, що означає настройку вагових коефіцієнтів наступним чином :

$$\Delta\omega_{ij}^{(n)} = -\eta\frac{\partial E}{\partial\omega_{ij}},\ n = \overline{1,2}, \qquad (2.5)$$

$\omega_{ij}^{(n)}$ - з'єднує і - й нейрон (n - 1) шару з j - м нейроном n - го шару, $0 < \eta < 1$ - коефіцієнт швидкості навчання. Відомо, що

$$\frac{\partial E}{\partial\omega_{ij}} = \frac{\partial E}{\partial y_j}\cdot\frac{dy_j}{dS_j}\cdot\frac{\partial S_j}{\partial\omega_{ij}}, \qquad (2.6)$$

де y_j - вихід нейрона, S_j - зважена сума його вхідних сигналів (аргумент активаційної функції). Тут підійде класичний сигмоїд ($y = \frac{1}{1+e^{\sum\omega_i x_i}}$), або гіперболічний тангенс. Третій множник $\frac{\partial S_j}{\partial\varpi_{ij}} = y_i^{(n-1)}$- виходу нейрона попереднього шару. Перший множник в (2.6) легко розкласти наступним чином :

$$\frac{\partial E}{\partial y_j} = \sum_k\frac{\partial E}{\partial y_k}\cdot\frac{dy_k}{dS_k}\cdot\frac{\partial S_k}{\partial y_j} = \sum_k\frac{\partial E}{\partial y_k}\cdot\frac{dy_k}{dS_k}. \qquad (2.7)$$

В (2.7) сума шукається серед нейронів (N + 1) шару. Введемо нову зміну :

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j}\cdot\frac{dy_j}{dS_j}, \qquad (2.8)$$

Одержимо рекурсивну формулу :

$$\delta_j^{(n)} = \left[\sum_k\delta_k^{(n+1)}\cdot\omega_{jk}^{(n+1)}\right]\cdot\frac{dy_j}{dS_j}, \qquad (2.9)$$

що дає змогу обчислити $\delta_j^{(n)}$, знаючи $\delta_j^{(n+1)}$. А для вихідного шару

$$\delta_e^{(n)} = (y_e^{(n)} - d_e)\cdot\frac{dy_e}{dS_e}, \qquad (2.10)$$

для випадку гіперболічного тангенса

$$\delta_e^{(n)} = (y_e^{(n)} - d_e)\cdot(1 - S_e^2), \qquad (2.11)$$

або у випадку сигмоїда

$$\delta_e^{(n)} = (y_e^{(n)} - d_e)\cdot(1 - S_e)\cdot S_e. \qquad (2.12)$$

Тоді (2.5) має вигляд

$$\Delta\varpi_{ij}^{(n)} = -\eta\delta_j^{(n)} y_i^{(n-1)}. \qquad (2.13)$$

Для того, щоб надати процесу корекції вагів деякої інерційності, що згладжує різкі стрибки при переміщенні по поверхні цільової функції, (2.13) доповнюється значенням зміни ваги на попередній ітерації

$$\Delta\omega_{ij}^{(n)}(t) = -\eta\cdot(\mu\cdot\Delta\omega_{ij}^{(n)}\cdot(t-1) + (1-\mu)\cdot\delta_j^{(n)}\cdot y_i^{(n-1)}), \qquad (2.14)$$

де μ - коефіцієнт інерційності, t - номер поточної ітерації.

Алгоритм навчання мережі, що зображена на рис. 2.1 із p=3, l=3, k=2.

Подаємо на вхід образ і розраховуємо значення виходів

$$S_j^{(n)} = \sum_{i=0}^{3} y_i^{(n-1)}\cdot\omega_{ij}^{(n)}, \qquad (2.15)$$

де 3 - число нейронів в шарі А, враховуючи нульовий нейрон із постійним вихідним значенням 1, що задає зміщення, $y_j^{(n-1)} = x_{ij}^{(n)}$- i - й вхід нейрона j шару n. В нашому випадку

$$\begin{aligned} S_1^1 &= w_{01}^1 + w_{11}^1 y_1^0 + w_{21}^1 y_2^0 + w_{31}^1 y_3^0, \\ S_2^1 &= w_{02}^1 + w_{12}^1 y_1^0 + w_{22}^1 y_2^0 + w_{32}^1 y_3^0, \qquad (2.16) \\ S_3^1 &= w_{03}^1 + w_{13}^1 y_1^0 + w_{23}^1 y_2^0 + w_{33}^1 y_3^0. \end{aligned}$$

Обчислити:

$$y_1^1 = \frac{1}{1+e^{-S1}},\quad y_2^1 = \frac{1}{1+e^{-S2}},\quad y_3^1 = \frac{1}{1+e^{-S3}}. \qquad (2.17)$$

Одинички вгорі замінити на двійки. Знайдемо y_i^2, $i = \overline{1,2}$. Покласти

$$y_1^0 = x_1^1,\ y_2^0 = x_2^1,\ y_3^0 = x_3^1. \qquad (2.18)$$

2. Розрахувати $\delta^{(n)}$ для вихідного шару: за формулою (2.12). В нашому випадку

$$\delta_1^{(2)} = (y_1^2 - d_1^1)(1 - S_1^2)S_1^2, \qquad (2.19)$$

$$\delta_2^{(2)} = (y_2^2 - d_2^1)(1 - S_2^2)S_2^2. \qquad (2.20)$$

Розрахувати за формулами (2.13 або 2.14) зміни вагів. В нашому випадку за (2.13):

$\Delta w_{11}^2 = -\eta\delta_1^{(2)} y_1^1,\ \Delta w_{12}^2 = -\eta\delta_{21}^{(2)} y_1^1,\ \Delta w_{21}^2 = -\eta\delta_1^{(2)} y_2^1,\ \Delta w_{22}^2 = -\eta\delta_2^{(2)} y_2^1,$

$\Delta w_{31}^2 = -\eta\delta_1^{(2)} y_3^1,\ \Delta w_{32}^2 = -\eta\delta_{21}^{(2)} y_3^1.$

Розрахувати за формулами (2.9) і (2.13), або (2.14) $\delta^{(n)}$ і $\Delta w^{(n)}$ для всіх інших шарів.

4. Скорегувати всі ваги нейронної мережі

$$w_{ij}^{(n)}(f) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t). \qquad (2.21)$$

5. Якщо помилка мережі істотна, то перейти на 1. Інакше кінець.

Завдання до лабораторної роботи.

1. Програмно реалізувати наведений алгоритм із p=3, l=3, k=2. Приклад наведено в додатку А.
2. Використовуючи дані таблиці 2.2, написати програмний фрагмент для обчислення 20 значень функції по заданих значеннях аргументів та їх варіаціях з кроком ± 1 і знайти середнє значення.
3. Подати задані значення аргументів на вхід нейронної мережі. В якості виходу d_1 вважати обчислене значення функції, в якості d_2 - 1, якщо одержане значення d_1 більше середнього значення і 0, якщо менше.
4. Задати контрольні приклади та оцінити швидкість і точність алгоритму.
5. В результаті навчання і тестування мережі необхідно побудувати графік залежності ймовірності (частотності) правильної відповіді від:

5.1. Кількості нейронів в проміжному шарі при заданій кількості вхідних нейронів.

5.2. Кількості вхідних нейронів при заданій кількості нейронів в проміжному шарі.

5.3. Кількості прикладів в навчальній вибірці при заданій кількості нейронів у вхідному і проміжному шарах.

5.4. Порога нейронів при фіксованих значеннях інших параметрів.

Таблиця 2.2 - Початкові дані

№ варіанта	x_1	x_2	x_3	Функція
1	1	2	3	$x_1^2 - x_2^2 + x_3^2$
2	5	4	3	$\sin x_1 + \sin x_2 - \sin x_3$
3	9	8	7	$tgx_1 + \sin x_2 - \sin x_3$
4	8	5	3	$\sin x_1 + \sin x_2 - \cos x_3$
5	5	6	7	$tgx_1 + \sin x_2 - tgx_3$
6	1	8	7	$\sin x_1 + tgx_2 - tgx_3$
7	2	5	4	$\cos x_1 + \cos x_2 - \sin x_3$
8	5	7	8	$\ln\cos x_1 + tgx_2 + ctgx_3$
9	2	3	6	$2^{x_1} + \cos x_2 - \sin x_3$
10	7	4	5	$\sin x_2^2 + x_1^2 - tgx_3$

Контрольні запитання

1. В чому полягає зміст алгоритму обереженого поширення похибки?
2. За яким оптимізаційним методом функціонує алгоритм?
3. Яке значення для алгоритму має функція похибки?
4. Для розв’язку яких задач застосовується даний алгоритм?
5. Які переваги та недоліки back propagation?
6. Чим відрізняється навчання з учителем від навчання без учителя?
7. Які існують методи навчання нейронних мереж?

8. До якого класу навчання відноситься навчання методом оберненого поширення похибки – до навчання без учителя чи з учителем?
9. Чому порядок пред'явлення прикладів в навчальній вибірці може впливати на якість навчання?
10. Як впливає зменшення кількості вхідних нейронів на функціонування мережі?
11. Які властивості повинна мати функція активації при використанні алгоритма оберненого поширення похибки?

ЛАБОРАТОРНА РОБОТА №3.

Тема: *Самоорганізована карта ознак Кохонена.*
Мета: *Навчитись алгоритмізувати задачі розпізнавання образів*

Короткі теоретичні відомості. Алгоритм Кохонена дає можливість будувати найронну мережу для поділу векторів вхідних сигналів на підгрупи. Мережа складається із M нейронів, що утворюють прямокутну решітку на площині (рис. 3.1). Вхідні сигнали подаються на всі нейрони мережі. В процесі функціонування налаштовуються синаптичні ваги нейронів.

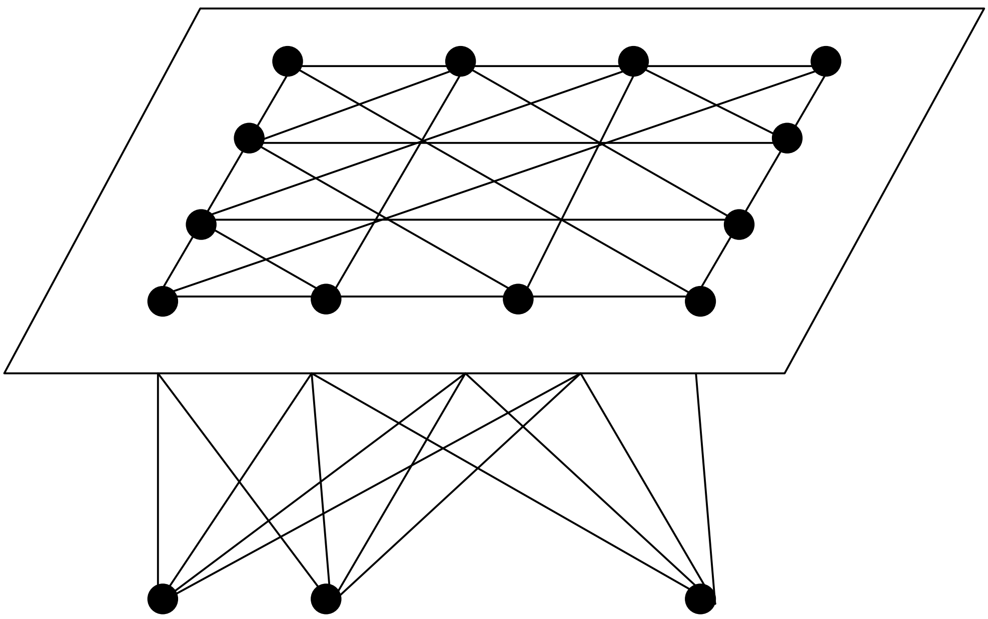


Рис. 3.1. Карта ознак Кохонена

Вхідні сигнали-вектори дійсних чисел — послідовно пред'являються мережі. Бажані входи не визначаються. Після того, як було пред'явлена достатня кількість вхідних векторів, синаптичні ваги мережі визначають кластери. Крім того, ваги організуються так, що топологічно близькі вузли чутливі до схожих вхідних сигналів.

Для реалізації алгоритму необхідно визначити міру сусідства нейронів. На рис. 3.2 показані зони топологічного сусідства нейронів на карті ознак в різні моменти часу. $NE_j(t)$ - множина нейронів, які вважаються сусідами нейрона *j* в момент часу *t*. Зони сусідства зменшуються з часом. Ваговим коефіцієнтам мережі надають малих випадкових значень. Їх загальне число MxN.

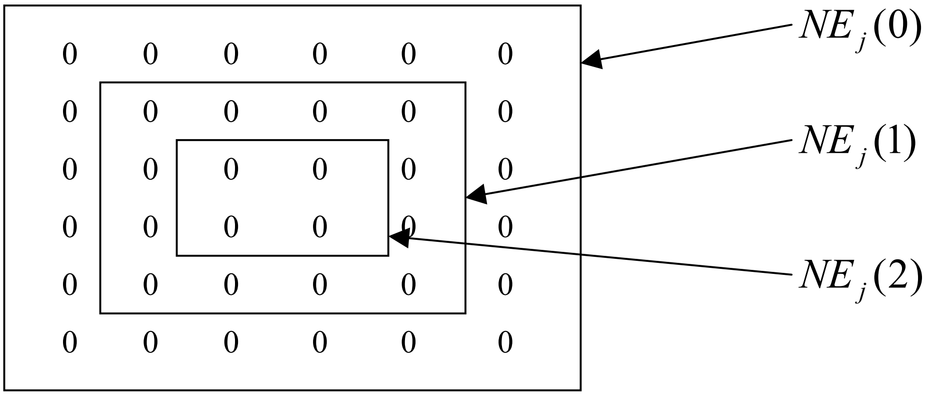


Рис. 3.2. Зони топологічного сусідства

Алгоритм Кохонена.

1. Мережі пред'являють новий вхідний сигнал.
2. Обчислюють відстань до всіх нейронів в мережі за формулою:

$$d_j = \sum_{i=0}^{N-1}(x_i(t) - w_{ij}(t))^2, \tag{3.1}$$

де $x_i^{(t)}$ - *i*-й елемент вхідного сигналу в момент часу t, $w_{ij}(t)$ – вага зв'язку від і-того елемента вхідного сигнала до нейрона j-го в момент часу t.
3. Вибирають нейрона з найменшою відстанню d_j.
4. Настроюють вагові коефіцієнти для j-го нейрона і всіх нейронів із його зони сусідства. Нові значення вагів

$$w_{ij}(t+1) = w_{ij}(t) + v(t)(x_i(t) - w_{ij}(t)), \tag{3.2}$$

де $v(t)$ – крок навчання, що змінюється із часом (додатнє число менше одиниці).
5. Повернення до кроку 2.

Завдання до лабораторної роботи

Відомо, що кількість кластерів дорівнює 3, кількість нейронів у карті ознак 9, кількість нейронів на вході 5. Необхідно сформувати таблицю 10x5, де у кожному рядку числа розміщені за алгоритмом, погодженим з викладачем.
Реалізувати алгоритм Кохонена. Провести тестування.

Контрольні запитання

1. Для розв'язку яких задач використовується карта Кохонена?

2. Зміст алгоритму Кохонена.
3. Недоліки та переваги карти Кохонена.

ЛАБОРАТОРНА РОБОТА №4.

Тема: *Машина Больцмана та мережа "INSTAR".*
Мета: *Вивчити принципи функціонування вказаних мереж та навчитись їх використовувати для розв'язку практичних задач.*

Короткі теоретичні відомості. Машина Больцмана використовується з другої половини 80-х років. Застосовується для розпізнавання образів та класифікації. Недоліком має повільний алгоритм навчання, перевагою її є те, що мережа дає можливість вибиратися із локальних мінімумів адаптивного рельєфу.

Конфігурація "INSTAR" – фрагмент нейронних мереж, що запропонований та використовувався Гроссбергом в багатьох нейромережевих моделях. Вхідна зірка, як показано на рис. 4.1 складається з нейрона, на який подається група входів через синаптичні ваги. Вхідні та вихідні зірки можуть бути взаємно з'єднані в мережі будь-якої складності. Гроссберг розглядав вхідні зірки як моделі окремих участків біологічного мозку

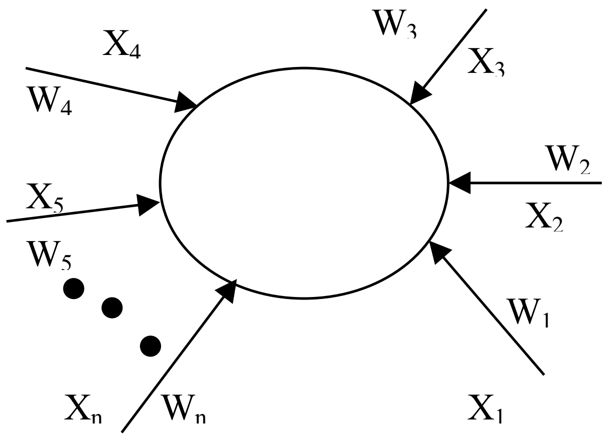


Рис. 4.1. Вхідна зірка

"INSTAR" може використовуватись в мережах розпізнавання образів. Але кожна зірка реалізує дуже просту функцію. Із зірок неможливо побудувати нейронну мережу, яка б реалізувала будь-яке задане відображення.

Алгоритм навчання машини Больцмана.

1. Визначити змінну Т, що представляє штучну температуру.
2. Пред'явити мережі множину входів, обчислити виходи і цільову функцію.
3. Дати випадкову зміну ваговому коефіцієнту і перерахувати вихід мережі і зміну цільової функції у відповідності із зробленою зміною ваги.

4. Якщо цільова функція покращилась (зменшилась), то зберегти зміну ваги. Якщо зміна ваги приводить до збільшення цільової функції, то ймовірність збереження цієї зміни обчислюється за допомогою розподілу Больцмана

$$P(c) = \exp(-\frac{c}{kT}), \quad (4.1)$$

де $P(c)$ - ймовірність зміни цільової функції, k - константа, аналогічна константі Больцмана, що вибирається в залежності від задачі, Т – штучна температура.

5. Вибрати випадкове число r із рівномірного розподілу від 0 до 1. Якщо $P(c)$ більше , ніж r, то зміна зберігається, в іншому випадку величина ваги повертається до попереднього значення (ця процедура дає можливість цільовій функції вибиратися із локальних мінімумів).

Ці кроки повторюються для кожного із вагів мережі, поступово зменшуючи Т, поки не буде досягнуто достатньо малого значення цільової функції. В цей момент пред‘являється другий вхідний вектор і процес навчання повторюється. Мережа навчається на всіх векторах навчальної множини до тих пір, поки цільова функція не стане допустимою для всіх із них. Швидкість зменшення температури повинна бути обернено пропорційною логарифму часу. При цьому мережа збігається до глобального мінімуму.

Процедура навчання вхідної зірки

Вхідна зірка навчається реагувати на визначений вхідний вектор Х і ні на який інший. Це навчання реалізується шляхом налаштування вагових коефіцієнтів таким чином, щоб вони відповідали вхідному вектору.

Вихід зірки визначається як зважена сума її входів. З іншої точки зору, вихід можна розглядати як згортку вхідного вектора з ваговим вектором. Відповідно, нейрон повинен реагувати найбільш сильно на вхідний образ, якому був навчений. Процес навчання виражається формулою

$$w_i\ (f+1) = w_i\ (f) + \alpha\ (x_i - w_i(f)), \quad (4.2)$$

де w_i - вага входу X_i, X_i – і-й вхід, α - коефіцієнт навчання, що має початкове значення і зменшується в процесі навчання.

Після завершення навчання пред‘явлення вхідного вектора Х буде активізувати навчений вхідний нейрон. Добре навчена зірка буде реагувати не тільки на визначений вектор, але і на незначні зміни цього вектора. Таким чином, зірка буде проявляти здібність до узагальнення. Це досягається поступовим налаштуванням нейронних вагів при пред‘явленні в процесі навчання векторів, що представляють варіації початкового вхідного вектора. Вагові коефіцієнти налаштовуються таким чином, щоб усереднити величини навчальних векторів, і нейрони одержують здатність реагувати на будь-який вектор цього класу.

Завдання до лабораторної роботи.

Якщо варіант студента є парним він вибирає для реалізації алгоритм "INSTAR", в іншому випадку машину Больцмана.

Для алгоритму "INSTAR" задати 20 п'ятиелементних векторів із відповідною закономірністю:

2 – сума двох перших елементів більша, ніж сума будь-яких двох перших елементів;

4 – різниця двох перших елементів є найменшою по модулю серед модулів усіх можливих різниць;

6 – перший та п'ятий елементи вхідних векторів більші за другий, третій та четвертий;

8 – частка першого і останнього елементів менша за будь-яку іншу частку, утворену із елементів вхідного вектора;

10 – добуток другого і третього елементів вхідного вектора найбільший серед усіх можливих попарних добутків.

Навчити зірку і перевірити правильність її функціонування на трьох тестових прикладах. Провести експеримент із двома з'єднаними на власний розсуд зірками. Зробити висновки.

Для моделювання машини Больцмана взяти звичайну мережу із 3-х шарів по 3 елементи у кожному шарі. Покласти T = 273 і $\kappa = 0{,}001$.

Для навчання використати 20 вхідних векторів з такими закономірностями:

1 – третій елемент менший перших двох;

3 – третій елемент є сумою перших двох;

5 – перший елемент є модулем різниці інших;

7 – другий елемент є добутком першого і третього;

9 – всі елементи кратні 3.

Навчити мережу за критерієм мінімуму середньоквадратичної похибки і перевірити правильність її функціонування на трьох тестових векторах. Зробити висновки.

Контрольні питання

1. Переваги та недоліки машини Больцмана.
2. Алгоритм навчання машини Больцмана.
3. Переваги та недоліки "INSTAR".
4. Алгоритм навчання "INSTAR".

ЛАБОРАТОРНА РОБОТА №5.

Тема: *RBF - мережа.*

Мета: *Дослідити роботу мережі, що не містить рекурсії і начитися за її допомогою апроксимувати функції.*

Короткі теоретичні відомості. RBF – мережа призначена для апроксимації функцій, яка задана в неявному вигляді набором шаблонів. Вона характеризується трьома особливостями: має єдиний прихований шар, тільки нейрони прихованого шару мають нелінійну активаційну функцію, синаптичні ваги всіх нейронів прихованого шару дорівнюють одиниці. Розглянемо наступні позначення (рис. 5.1):

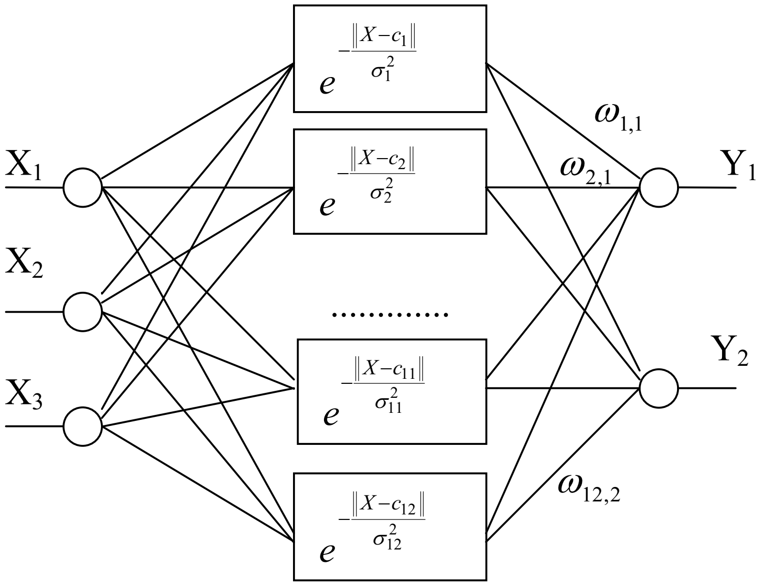


Рис. 5.1. Класична RBF – мережа

- $c = (c_1, c_2, ..., c_n)$ - вектор координат центра активаційної функції нейрона прихованого шару;
- σ_j - ширина вікна активаційної функції j- го нейрона прихованого шару;
- $f(X,c) = e^{\frac{-\sum_{j=1}^{n}(X_j - c_j)^2}{\sigma^2}}$ - радіально-симетрична активаційна функція нейрона прихованого шару;
- $\omega_{i,j}$ - вага зв'язку між i-м нейроном вихідного шару і j-м нейроном прихованого шару.

Алгоритм навчання мережі RBF

1. Вибрати розмір прихованого шару H рівним кількості тренувальних шаблонів Q. Синаптичні ваги нейронів прихованого шару прийняти рівними 1.
2. Розмістити центри активаційних функцій нейронів прихованого шару в точках простору вхідних сигналів мережі, які входять в набір тренувальних шаблонів $\Xi : c_j = \overline{X_j}, j = \overline{1,H}$.

3. Вибрати ширини вікон активаційних функцій нейронів прихованого шару σ_j, $j=\overline{1,H}$ достатньо великими, але так, щоб вони не накладались одна на іншу в просторі вхідних сигналів мережі.
4. Визначимо ваги нейронів вихідного шару мережі $w_{i,j}, i=\overline{1,Z},\ j=\overline{1,H}$. Для цього пред'явимо мережі весь набір тренувальних шаблонів. Вихід i-го нейрона вихідного шару для p-го шаблону буде рівний:

$$Y_i = w_{i1}f(\overline{X_p},c_1)+w_{i2}f(\overline{X_p},c_2)+...+w_{iH}f(\overline{X_p},c_H)= \qquad (5.1)$$
$$= w_{i1}f(\overline{X_p},\overline{X_1})+w_{i2}f(\overline{X_p},\overline{X_2})+...+w_{iH}f(\overline{X_p},\overline{X_H})=D_i\,.$$

Розписавши це рівняння для всіх виходів мережі і всіх шаблонів, одер жимо наступне рівняння в матричній формі:

$$\varPhi w^T = D\,, \qquad (5.2)$$

де $\varPhi=\begin{pmatrix} f_{11} & \cdots & f_{1H} \\ f_{21} & \cdots & f_{2H} \\ \cdots & \cdots & \cdots \\ f_{H1} & \cdots & f_{HH} \end{pmatrix}$ - інтерполяційна матриця, $f_{ij}=f(\overline{X_i},\overline{X_j})$;

$w=\begin{pmatrix} w_{11} & \cdots & w_{1Z} \\ . & .. & . \\ w_{H1} & \cdots & w_{HZ} \end{pmatrix}$ - матриця вихідних синаптичних вагів;

$D=\begin{pmatrix} D_{11} & \cdots & D_{1Z} \\ . & . & . \\ D_{H1} & \cdots & D_{HZ} \end{pmatrix}$ - матриця вихідних шаблонів.

Розв'язок

$$w^T = \varPhi^{-1}D \qquad (5.3)$$

дає нам шукані значення вихідних синаптичних вагів, що забезпечують проходження інтерполяційної поверхні через тренувальні шаблони в

<u>Завдання до лабораторної роботи.</u>

Здійснити програмну реалізацію мережі RBF. Забезпечити можливості динамічних змін кількості нейронів прихованого шару та ширини активаційних вікон, а також виведення інформації у файл та у вигляді графіків.

Для вказаного викладачем функціонального перетворення підібрати ширини вікон при яких апроксимація буде найкращою, про що пересвідчитись на контрольних шаблонах. Перевірити, як змінюється якість апроксимації при збільшенні кількості навчальних шаблонів та зміні їх дисперсії. Встановити, що RBF – мережа виконує краще: інтерполяцію, чи екстраполяцію. Пояснити результати.

Контрольні питання

1. Алгоритм обчислення оберненої матриці.

2. Для розв'язку яких задач застосовується RBF – мережа.
3. Особливості структури мережі та її функціонування. Переваги та недоліки.

ЛАБОРАТОРНА РОБОТА №6.

Тема: *Метод групового врахування аргументів.*
Мета: *Навчитись будувати математичні моделі складних об'єктів у вигляді поліноміальної функції багатьох змінних за допомогою методу самоорганізації.*

Короткі теоретичні відомості. Згідно принципа еврістичної саморганізації, математична модель оптимальної складності відповідає мінімуму деякого критерію, що називають критерієм селекції. В якості такого критерію використовуються критерії регулярності і незміщеності моделі, або балансу змінних. Встановлено, що при поступовому підвищенні складності моделі критерії проходять через мінінмальні значення. Комп'ютер знаходить глобальний мінімум і вказує на єдину модель оптимальної складності при дуже малому об'ємі апріорної інформації. Людина повинна лише вибрати критерій у відповідності до задачі. Об'єм повного перебору зменшується за допомогою алгоритмів багаторядної селекції, подібно тому, як це практикують при селекції рослин чи тварин. Метод групового врахування аргументів (МГВА) реалізує ряд алгоритмів поступового ускладнення моделі за правилами багаторядної селекції. Автором методу є відомий український вчений Олексій Григорович Івахненко. МГВА застосовується для вирішення задач зменшення інформаційної невизначеності, а саме прогнозування, розпізнавання образів, автоматичної класифікації, ідентифікації характеристик, а також для автоматичного оптимального управління.

Алгоритми МГВА відтворюють схему масової селекції. В них є генератори комбінацій, що ускладнюються із ряду в ряд і порогові самовідбори найкращих з них. Якщо об'єкт описується функцією

$$\varphi = f(x_1, x_2, ..., x_m), \tag{6.1}$$

яка, наприклад, є степеневим поліномом, то вона замінюється декількома рядами "частинних" описань:

1-й ряд селекції:

$$y_1 = f(x_1 x_2),\ y_2 = f(x_1 x_3), ..., y_s = f(x_{m-1} x_m), \tag{6.2}$$

2-й ряд селекції:

$$z_1 = f(y_1 y_2),\ z_2 = f(y_1 y_3), ..., z_p = f(y_{s-1} y_s), \tag{6.3}$$

де $s = m^2$, $p = s^2$ і так далі.

Критерії селекції представлені на рис. 6.1. Так критерій регулярності рекомендується для синтезу моделі однократного прогнозу, критерій незміщеності – для ідентифікації рівнянь об'єкта, критерій балансу змінних – для системного багатократного прогнозу і при виділенні трендів.

Критерій регулярності існує в одній із двох форм: середньоквадратична похибка, що визначається на окремій послідовності даних для перевірки (шукаємо мінімум) або коефіцієнт кореляції дійсних значень функції і вихідної величини моделі (шукаємо максимум). Значно кращі екстраполяційні якості має критерій незміщеності, при якому в процесі селекції вибирається модель, яка найменш чутлива до вибору експериментальних точок. І найкращим екстраполяційним критерієм є критерій балансу змінних.

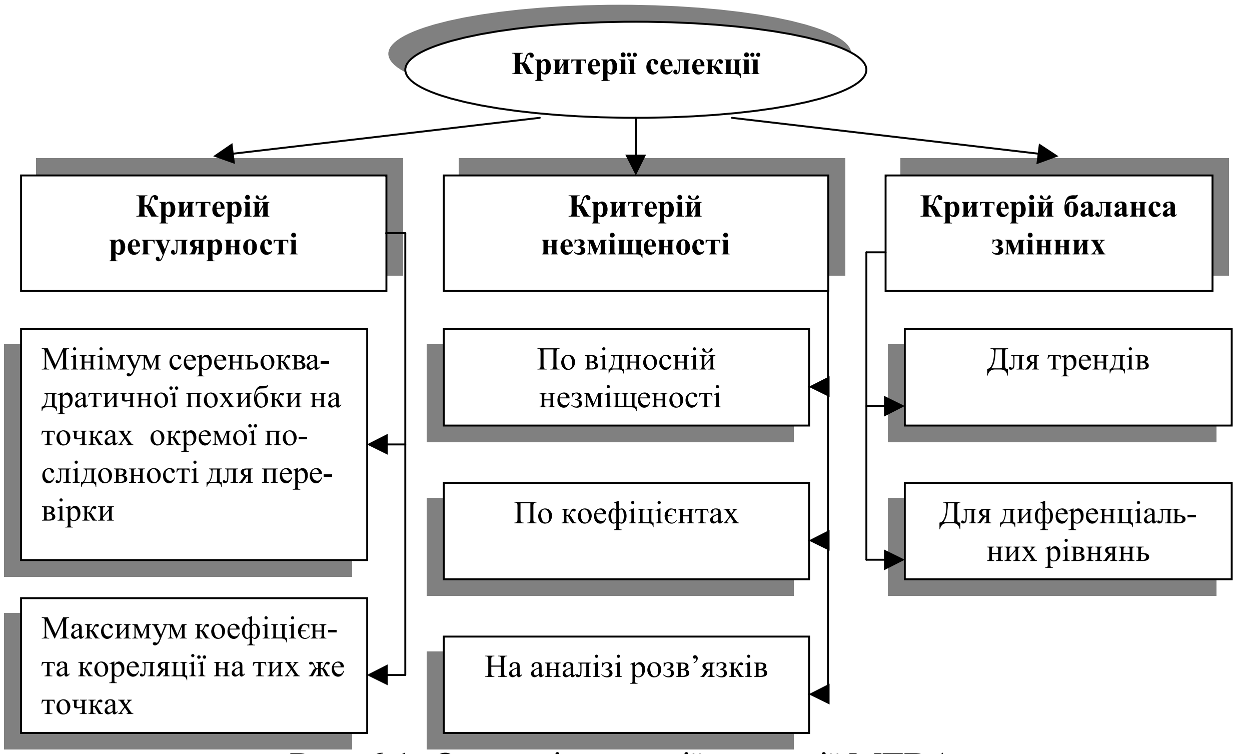


Рис. 6.1. Основні критерії селекції МГВА

<u>Основні етапи алгоритму МГВА</u>

1. Розробити програмний фрагмент, що реалізує метод найменших квадратів (МНК) для лінійної функції від n функцій.
2. Якщо згідно Вашого варіанту (див. табл. 6.1) критерієм селекції є критерій регулярності (перший варіант), то необхідно розділити набір даних на навчальну та контрольну послідовності.
3. Визначитись із своєю опорною функцією.
4. Визначити за допомогою МНК коефіцієнти опорних функцій.
5. За своїми критеріями селекції залишити певний відсоток таких функцій для подальших ітерацій.
6. При збільшенні значення критерію ітерації припинити і визначити прогнозне значення функції, порівняти його із істинним.
7. * Вивести поліном у аналітичному вигляді на друк.

Завдання до лабораторної роботи.

Одержати математичну модель за методом МГВА, здійснити прогнозування та порівняти результат МГУА із результатом, що одержаний будь-яким іншим методом, використовуючи точки відомої функції.

Таблиця 6.1

№ вар	x_1	x_2	x_3	y	Критерій селекції	Опорна функція
1	4	5	9	12	Балансу змінних	$y = a_0 + a_1 x_i x_j$
2	7	3	8	13	Регулярності (1)	$y = a_0 + a_1 x_i + a_2 x_j$
3	5	4	5	15	Незміщеності	$y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j$
4	2	2	6	18	Регулярності (2)	$y = a_0 + a_1 x_i x_j$
5	3	1	4	32	Незміщеності	$y = a_0 + a_1 x_i + a_2 x_j$
6	8	3	7	54	Балансу змінних	$y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j$
7	9	6	8	21	Незміщеності	$y = a_0 + a_1 x_i x_j$
8	2	8	2	75	Регулярності (1)	$y = a_0 + a_1 x_i + a_2 x_j$
9	1	9	1	35	Регулярності (2)	$y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j$
10	3	7	3	65	Балансу змінних	$y = a_0 + a_1 x_i + a_2 x_j$

Контрольні питання

1. Для розв’язку яких задач застосовується МГВА?
2. Критерії зовнішнього доповнення.
3. Переваги та недоліки критеріїв регулярності, балансу змінних та незміщеності.
4. Особливості реалізації алгоритму МГВУ.

ЛАБОРАТОРНА РОБОТА № 7.

Тема: *Генетичні алгоритми.*
Мета: *Навчитись визначати глобальні екстремуми складних функцій за допомогою генетичних алгоритмів.*

Короткі теоретичні відомості. Вперше ідея використання генетичних алгоритмів для навчання виникла в 1970 році. В другій половині 80-х років до цієї ідеї повернулись в зв‘язку із навчанням нейронних мереж. Генетичний алгоритм (ГА) нагадує біологічні процеси. Найбільш важливі з них випадкові мутації в хромосомах, кроссовер і рекомбінація генетичного матеріалу та міграція генів. ГА працює наступним чином. Ініціалізується популяція і всі хромосоми порівнюють у відповідності із заданою функцією оцінки. Далі виконується функція репродукції популяції хромосом. Батьки вибираються наступним чином у відповідності із значенням оцінки (ймовірність оцінки того, що дана хромосома стане батьком, пропорційна одержаній оцінці). Репродукція відбувається індивідуально для одного батька шляхом мутації хромосоми, або для двох батьків шляхом кроссовера генів. Діти, що одержалися оцінюються і розміщуються в популяції.

В результаті описаних операцій на кожному етапі еволюції одержуються популяції із більш досконалими індивідуами.

Генетичне наслідування моделюється наступним чином (табл. 7.1):

Таблиця 7.1.

Хромосома	Вектор (послідовність) із нулів та одиниць. Кожна позиція (біт) називається геном.
Індивідуум = =генетичний код	Набір хромосом = варіант розв‘язку задачі
Кроссовер	Операція, при якій хромосоми обмінюються частинами
Мутація	Випадкова зміна однієї, чи кількох позицій в хромосомі

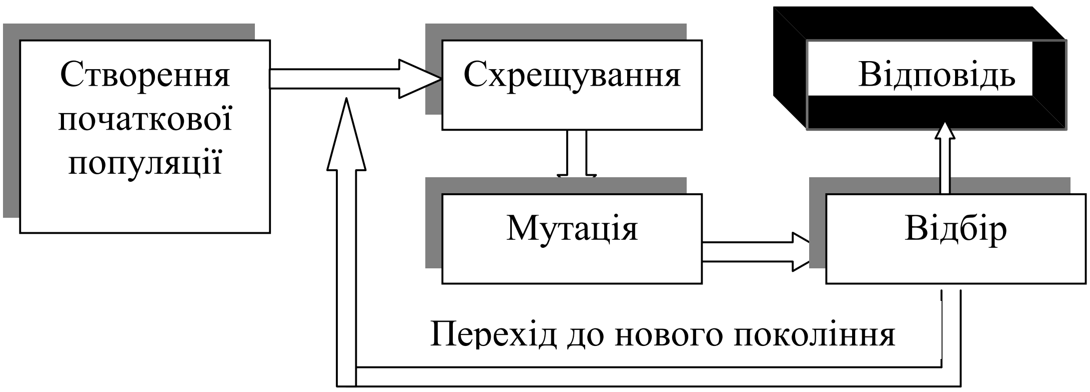


Рис.7.1. Алгоритм обчислень

Кроссовер називається одно точковим, якщо обмін частинами хромосом відбувається так

1	0	1	0	1	1	1	0	0	1
0	1	1	1	1	0	0	1	1	1

⇒

1	0	1	0	1	0	0	1	1	1
0	1	1	1	1	1	1	0	0	1

Двохточковий кроссовер

1	0	1	0	1	1	1	0	0	1
0	1	1	1	1	0	0	1	1	1

⇒

1	0	1	1	1	0	0	0	0	1
0	1	1	1	1	1	1	1	1	1

Батьківські пари можна вибирати такими способами:

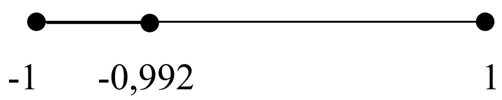
Панміксія – обидва члени популяції, що створюють батьківські пари випадковим чином вибираються із усієї популяції, при чому будь-який член популяції може стати учасником декількох пар.
Селективний – батьками можуть стати лише ті особи, значення пристосованості яких не менше середнього значення пристосованості по популяції.
Інбрідінг – перший член пари вибирається випадковим чином, а другим з більшою ймовірністю буде максимально близький до нього член популяції.
Аутбрідінг – пари формуються аналогічно, але із максимально далеких членів популяції.
Рошарняють фенотипний і генотипний інбрідінг та аутбрідінг.
Існує два механізми відбору членів нової популяції: елітний та відбір з витісненням. При *елітному відборі* нова популяція складатиметься лише із найкращих членів репродукційної групи, що об‘єднує в себе батьків, потомків і мутантів. При *відборі з витісненням* те, чи буде член репродукційної групи заноситись в нову популяцію визначається не лише величиною її пристосованості, але й тим, чи є у новій популяції член із аналогічним набором хромосом.

Постановка задачі, алгоритм її розв‘язку та початкові дані.

Знайти $extr\, F(\vec{x}), \vec{x} \in D$, де D – обмежена область.

Приклад. Необхідно знайти max $f(x_1, x_2)$, де $-1 \le x_1, x_2 \le 1$ і $f(x_1, x_2) = \sqrt{3}\cos x + \sin x$.

Розв'язок. Розіб‘ємо відрізок [-1;1] на 255 відрізків.



Будемо кодувати $-1 \to 00000000,\ -0{,}992 \to 00000001, ...,\ 1 \to 11111111$.

Таблиця 7.2. Початкові дані.

№	Функція	Кроссовер	Ймовірність мутації	Вибір батьків	Механізм відбору
1	$F(x_1, x_2) = \dfrac{100}{100(x_1^{\ 2} - x_2) + (1 - x_1)^2 + 1}$, $-1{,}28 \le x_{1,2} \le 1{,}28$.	Одноточковий	0,01	Панміксія	Елітний
2	$F(x_1, x_2, \ldots, x_5) = \sum_{i=1}^{5} [x_i]$, $-5{,}12 \le x_{1,2,3,4,5} \le 5{,}12$.	Двохточковий	0,02	Селективний	Із витісненням
3	$F(x_1, x_2) = 0{,}002 + \sum_{j=1}^{25} \dfrac{1}{j + \sum_{i=1}^{2} (x_i - a_i)^6}$,	Одноточковий	0,015	Інбрідінг	Елітний

	$a_{1j} = 16[(j \bmod 5) - 2], a_{2j} = 16[(j\%5) - 2]$.				
4	$F(x_1, x_2) = 20 + x_1{}^2 + x_2^2 - 10\cos 2\pi x_1 - 10\cos 2\pi x_2$ $-5{,}12 \le x_{1,2} \le 5{,}12$.	Двох-точко-вий	0,01	Аутбрі-дінг	Із витіс-ненням
5	$F(x_1, \dots, x_2) = \sum_{i=1}^{10}(10\cos(2\pi x_i) - x_i^2) - 100$, $-5{,}12 \le x_{1,2,\dots 10} \le 5{,}12$.	Одно-точко-вий	0,015	Панмік-сія	Елітний
6	$F(x_1, x_2, \dots, x_5) = \sum_{i=1}^{5}[x_i]$, $-5{,}12 \le x_{1,2,3,4,5} \le 5{,}12$.	Одно-точко-вий	0,01	Панмік-сія	Елітний
7	$F(x_1, x_2) = 20 + x_1{}^2 + x_2^2 - 10\cos 2\pi x_1 - 10\cos 2\pi x_2$, $-5{,}12 \le x_{1,2} \le 5{,}12$.	Одно-точко-вий	0,02	Панмік-сія	Елітний
8	$F(x_1, x_2, \dots, x_5) = \sum_{i=1}^{5}[x_i]$, $-5{,}12 \le x_{1,2,3,4,5} \le 5{,}12$.	Двох-точко-вий	0,015	Панмік-сія	Елітний
9	$F(x_1, \dots, x_2) = \sum_{i=1}^{10}(10\cos(2\pi x_i) - x_i^2) - 100$, $-5{,}12 \le x_{1,2,\dots 10} \le 5{,}12$.	Одно-точко-вий	0,02	Панмік-сія	Із витіс-ненням
10	$F(x_1, x_2) = 20 + x_1{}^2 + x_2^2 - 10\cos 2\pi x_1 - 10\cos 2\pi x_2$, $-5{,}12 \le x_{1,2} \le 5{,}12$.	Одно-точко-вий	0,01	Селек-тивний	Елітний

Оскільки змінних дві, то хромосома складатиметься із 16 генів, перша половина яких відповідає x_1, друга - x_2. Всього таких хромосом – 65536. Випадковим чином виберемо серед них 100 – початкову популяцію. Для фенотипів популяції обчислимо значення $(F_1, \dots, F_{100})$. Кожному значенню F_i співставимо ймовірність $p_i = \frac{F_i}{\sum_{i=1}^{100} F_i}$. Подальші кроки виконуються згідно таблиці 7.2.

Контрольні запитання.

1. Для розв‘язку яких задач використовують ГА?
2. Переваги та недоліки ГА.
3. Етапи функціонування ГА.
4. Генотипи та фенотипи.
5. Символьна модель ГА.
5. Геометрична інтерпретація символьної моделі.

ЛАБОРАТОРНА РОБОТА № 8.

<u>Тема</u>: *Моделі і методи визначення компетентності експертів на базі аксіоми незміщеності.*

<u>Мета</u>: *Навчитись будувати автоматизовані експертні системи на прикладі задачі визначення компетентності експертів.*

<u>Короткі теоретичні відомості і формулювання задачі</u>. Інформаційна невизначеність на початкових етапах життєвого циклу складних об'єктів і систем здійснює значний вплив на подальші процеси їх проектування і виготовлення. З огляду на високу вартість проектних і конструкторських робіт і, як наслідок, малоймовірні уточнення результатів попередніх досліджень, відзначимо, що визначення найбільш компетентних осіб, що роблять оцінки і визначають стратегію досліджень, є однією із найважливіших задач. Будемо проводити визначення компетентності експертів на базі аксіоми незміщеності, яка стверджує, що судження більшості компетентне і, як наслідок, найбільш компетентним вважати того експерта, розбіжність суджень якого із судженнями інших експертів мінімальна.

Експерти відповідають на численні різнозначні запитання, що класифікують в залежності від відповідей:
1) “Так-Ні”; 2) одна із декількох; 3) декілька з багатьох; 4) число; 5) інтервал; 6) нечіткий інтервал; 7) слово; 8) речення. Деякі з них можна було б включити в склад інших, але вважаючи, що запитання з різними варіантами відповідей знаходяться на різних рівнях ієрархії цільового дерева і мають відповідно різні рівні значимості, доцільно їх розділяти. Задача визначення рівня компетентності експертів має декілька варіантів постановки при таких початкових умовах:

- рівні компетентності експертів апріорно не відомі і особою, що приймає рішення (ОПР) не задані;
- початкові рівні компетентності задаються ОПР;
- початкові рівні компетентності є середнім арифметичним оцінки ОПР і самооцінки;
- початкові рівні компетентності є середнім арифметичним суджень інших експертів (відсутність судження - оцінка 0.5).
- при визначенні компетентності оцінки ОПР рівнозначні оцінці групи експертів;
- при визначенні компетентності враховується оцінка ОПР як експерта із заданим рівнем компетентності і т.д.

Задача визначення компетентності експертів формулюється наступним чином . Нехай n - кількість експертів, m - кількість запитань, причому $m=\sum_{i=1}^{8} m_i$, де m_i - кількість запитань i-го типу, $i=\overline{1,8}$, відповідно до вищевикладеної класифікації. Необхідно визначити рівні компетентності експертів γ_j, $j=\overline{1,n}$.

<u>Моделі і методи вирішення задачі</u>. Сутність методу полягає у визначенні матриць, що містять значення розбіжностей суджень експертів, їхньому аналізі і перетвореннях, в результаті яких будуть визначені рівні компетентності експертів. Помітимо, що метод достатньо алгоритмізований і, без обмеження загальності, в якості базового варіанта розглянемо випадок, коли рівні компетентності апріорно невідомі.

1. Методом експертного опитуванння визначимо матрицю $L=(l_{ij})_{i=1,j=1}^{n\quad m_1}$, де $l_{ij}=\begin{cases}1, \text{якщо } i-й \text{ експерт дав позитив}-\\ \text{ну відповідь на } j-е \text{ запитання,}\\ 0, \quad \text{в іншому випадку}.\end{cases}$ Формуємо послідовність трикутних матриць $\{T_1^k\}_{k=\overline{1,m_1}}$, де $T_1^k=\left(t_{ij}^k\right)_{i,j=1}^n$ і $t_{ij}^k=\chi\left(l_{ik}=l_{jk}\right)$ при $i>j$, при $i\le j$ $t_{ij}^k=0$, де $\chi(B)=\begin{cases}1, \text{ якщо } B \text{ вірно},\\ 0, \quad \text{в іншому випадку}.\end{cases}$

Обчислимо матрицю $T_1^{'}=\left(t_{ij}^{'}\right)_{i,j=1}^n$, $t_{ij}^{'}=\sum\limits_{k=1}^{m_1}\chi\left(l_{ik}=l_{jk}\right)$, $i>j$, при $i\le j$ $t_{ij}^k=0$. Нормуючи матрицю $T_1^{'}$, одержимо $T_1=\left(t_{ij}\right)_{i,j=1}^n$, $t_{ij}=\dfrac{t_{ij}^{'}}{\sum\limits_{\substack{i,j=1\\ i>j}}^{n}t_{ij}^{'}}$.

Якщо на цьому етапі є необхідність у попередньому висновку про компетентність експертів, то її обчислюють за формулою $\gamma_p=\sum\limits_{\substack{i,j=1\\ i>j\\ (j=p)\vee(i=p)}}^{n}t_{ij}\Bigg/\sum\limits_{p=1}^{n}\ \sum\limits_{\substack{i,j=1\\ i>j\\ (j=p)\vee(i=p)}}^{n}t_{ij}$, $p=\overline{1,n}$.

2. ОПР для кожного q-го запитання $q=\overline{1,m_2}$, визначає кількість відповідей k_q і надає кожній відповіді в запитанні визначений бал a_{ql} де l-номер відповіді в q-му запитанні, $a_{ql}\in[0;1]$, $\sum\limits_{l=1}^{k_q}a_{ql}=1$. (Припускаємо, що відповіді на запитання другого типу упорядковані за зростанням балів, тобто має місце кількісна або змістовна градація). Методом експертного опитуванння визначаємо матрицю відповідей експертів $A=\left(a_{ij}\right)_{i=1,\,j=1}^{n\quad m_2}$, де a_{ij} - бал i-го експерта за відповідь на j-е запитання. Формуємо послідовність трикутних матриць $\{T_2^k\}_{k=\overline{1,m_2}}$, де $T_2^k=\left(t_{ij}^k\right)_{i,j=1}^n$ і $t_{ij}^k=\left|a_{ik}-a_{jk}\right|$, $i>j$, при $i\le j$ $t_{ij}^k=0$. Обчислюємо матриці $T_2^{k'}=\left(t_{ij}^{k'}\right)_{i,j=1}^n$, $t_{ij}^{k'}=\dfrac{1}{t_{ij}^k}$ при $i>j$ і

$t_{ij}^{k} \neq 0$ якщо при $i > j$ $t_{ij}^{k} = 0$, то раціонально покласти $t_{ij}^{k'} = \dfrac{2}{\min\limits_{t_{ij}^{'} \neq 0} t_{ij}^{k}}$, інші нульові елементи залишимо без змін. Обчислимо матрицю $T_2^{'} = \left(t_{ij}^{'}\right)_{i,j=1}^{n}$, де $t_{ij}^{'} = \sum_{k=1}^{m_2} t_{ij}^{k'}$, *при* $i > j$, при $i \leq j$ $t_{ij}^{'} = 0$. Нормуючи матрицю $T_2^{'}$, одержимо матрицю $T_2 = \left(t_{ij}\right)_{i,j=1}^{n}$, $t_{ij} = \dfrac{t_{ij}^{'}}{\sum\limits_{i>j} t_{ij}^{'}}$ при $i > j$. Якщо за відповідями на запитання другого типу є необхідність у попередньому заключенні про компетентність експертів, то її обчислюють за останньою формулою.

3. ОПР для кожного p-го запитання $p = \overline{1, m_3}$, визначає кількість відповідей k_p і надає кожній відповіді визначений бал a_{pl}, де l-номер відповіді в p-му запитанні, $a_{pl} \in [0;1]$, $\sum_{l=1}^{k_p} a_{pl} = 1$. Справедливе припущення п. 2.

Методом експертного опитуванння визначаємо матрицю $L = \left(l_{ijq}\right)_{i,j=1}^{n} {}_{q=1}^{m_3} {}^{mm}$, де $mm = \max\limits_{p} k_p$ і $l_{ijq} = \begin{cases} 1, & \text{якщо } i-\text{й експерт на } j-\text{е запи}- \\ & \text{тання дав } q-\text{у відповідь,} \\ 0, & \text{в іншому випадку.} \end{cases}$ Формуємо послідовність трикутних матриць $\left\{T_3^{k}\right\}_{k=\overline{1,m_3}}$, де $T_3^{k} = \left(t_{ir}^{k}\right)_{i,r=1}^{n}$ і $t_{ir}^{k} = \sum_{q=1}^{k_p} \chi\left(l_{ikq} = l_{rkq}\right) a_{kq}$, $i > r$.. Наступні кроки аналогічні з незначними уточненнями крокам п. 2.

4. Методом експертного опитуванння формуємо числову матрицю $L = \left(l_{ij}\right)_{i=1,}^{n} {}_{j=1}^{m_4}$, де $l_{ij} -$ відповідь i-го експерта на j- запитання. Можливі випадки, коли ОПР призначає інтервали для відповідей і не призначає. Нехай у першому випадку можливі інтервали для відповідей $\lfloor a_j ; b_j \rfloor$. Позначимо $N_z = \{1, 2, ..., z\}$. Якщо $\exists i \in N_n$ *і* $\exists j \in N_{m_4} : l_{ij} \notin \lfloor a_j ; b_j \rfloor$, то $t_{ip}^{j} = 0$ при $i > p$. Якщо l_{ij} і $l_{qj} \in \lfloor a_j ; b_j \rfloor$, то $t_{iq}^{j} = \dfrac{\left|l_{ij} - l_{qj}\right|}{b_j - a_j}, i > q.$ Ця формула має місце і в другому випадку, але в ній для всіх $j \in N_{m_4}$ $a_j = \min\limits_{i} l_{ij}$, $b_j = \max\limits_{i} l_{ij}$. Подальші міркування аналогічні п. 2.

5. Методом експертного опитуванння формуємо матрицю $A = \left(a_{ij}\right)_{i=1,}^{n} {}_{j=1}^{2m_5}$, де

$$a_{ij} == \begin{cases} \textit{лівий кінець інтервала відповіді } i-\textit{го експерта на } j-\textit{е запитання}, \\ \textit{де } j = 2k-1,\ k=\overline{1,m_5}, \\ \textit{правий кінець інтервала відповіді } i-\textit{го експерта на } j-\textit{е запитання}, \\ \textit{де } j=2k,\ k=\overline{1,m_5}. \end{cases}$$

Формуємо послідовність трикутних матриць $\{T_5^k\}_{k=\overline{1,m_5}}$, де $T_5^k = (t_{ij}^k)_{i,j=1}^k$ і

$$t_{ij}^k = \frac{1}{2}\chi(\min\{a_{il},a_{jl}\} \ge \max\{a_{iq},a_{jq}\}) \cdot (\min\{a_{il},a_{jl}\} - \max\{a_{iq},a_{jq}\})\left(\frac{1}{a_{il}-a_{iq}} + \frac{1}{a_{jl}-a_{jq}}\right),$$

де $l = 2k$, $q = 2k-1$, $k=\overline{1,m_5}$, $i > j$ і $t_{ij}^k = 0$ при $j \ge i$. Наступні кроки аналогічні крокам п. 2.

6. Методом експертного опитуванння визначаємо нечіткі інтервали у вигляді п'ятірки елементів ($\alpha_i^k, \beta_i^k, \underline{m}_i^k, \overline{m}_i^k, h_i^k$), $i=\overline{1,n}$, $k=\overline{1,m_6}$. Для кожного експерта і кожного запитання визначаємо значення параметра $w_i^k = \frac{h_i^k}{2}\left(\underline{m}_i^k + \overline{m}_i^k\right) + \frac{h_i^k}{4}\left(\beta_i^k - \alpha_i^k\right)$, впевненість в одержанні якого у експерта максимальна. Зауважимо, що у кожного експерта для всіх запитань існує хоча б одне значення, впевненість в одержанні якого максимальна (дорівнює одиниці), інші випадки не розглядається. Обчислюємо елементи матриці $T_6^k = (t_{ij}^k)_{i,j=1}^n$, де $t_{ij}^k = \dfrac{|w_i^k - w_j^k|}{\max\limits_i w_i^k - \min\limits_i w_i^k}$, $i,j=\overline{1,n}$, $i > j$, $k=\overline{1,m_6}$.

Далі необхідно матриці T_6^k додати і виконати обчислення аналогічно п. 2.

7. Методом експертного опитуванння визначимо слова W_i^k, що є відповіддю i-го експерта на k-е запитання, $i=\overline{1,n}$, $k=\overline{1,m_7}$. Для кожного запитання визначимо узагальнений індикатор

$$\chi^*(W_i^k \in S_{W^k}) = \begin{cases} 0, & W_i^k \notin S_{W^k}, \\ \alpha_1, W_i^k = W_1^k \in S_{W^k}, \\ ------------ \\ \alpha_{p_k}, W_i^k = W_{p_k}^k \in S_{W^k}, \end{cases}$$

де S_{W^k} - множина синонімів, p_k - кількість синонімів для слова-відповіді на k-е запитання, $\sum\limits_{j=1}^{p_k}\alpha_j = 1$, $\forall k \in N_{m_7}$. Далі процес визначення матриці T_7 аналогічний визначенню матриці T_2.

8. Для восьмого типу запитань запропонувати адекватну алгоритмічну процедуру оцінки близькості відповідей експертів на сучасному рівні інтелектуалізації технічних засобів не уявляється можливим, тому матриця T_8 визначається ОПР.

Таким чином, додаючи вісім матриць для визначення компетентності, одержимо результуючу матрицю T, де нижче головної діагоналі розташовані оцінки компетентності експертів один іншим. По відомій процедурі (напр. п. 2) визначаємо компетентності експертів. Якщо початкові рівні компетентності задаються ОПР, то вони знаходяться на головній діагоналі матриці T і рівноправні з іншими оцінками експертів. Рівні компетентності в такому випадку обчислюються за формулою

$$\gamma_p = \sum_{\substack{i,j=1 \\ i \ge j \\ (j=p)\vee(i=p)}}^{n} t_{ij} \Bigg/ \sum_{p=1}^{n} \sum_{\substack{i,j=1 \\ i \ge j \\ (j=p)\vee(i=p)}}^{n} t_{ij}, \quad p=\overline{1,n}.$$

Якщо початкові рівні компетентності вважати середнім арифметичним суджень інших експертів (відсутність суджень - оцінка 0.5), то необхідно сформувати матрицю T_0, $T_0 = (t_{ij}^0)_{i,j=1}^n$, де на головній діагоналі знаходяться нулі, t_{ij}^0 - оцінка i-м експертом j-го експерта. Далі знаходимо суму елементів рядків і ділимо на $n-1$. Одержані числа розташовуємо на головній діагоналі T і використовуємо останню формулу. У випадку рівнозначності оцінки ОПР оцінці групи експертів, після визначення рівня компетентностей по матриці T з нульовими діагональними елементами, додаються і усереднюються отримані оцінки й оцінки, дані ОПР. Процедура визначення компетентності, при якій враховується оцінка ОПР як експерта із заданим рівнем компетентності по відношенню до групи експертів, відрізняється від попередніх тим, що при усередненні оцінка ОПР домножається на ваговий коефіцієнт.

<u>Постановка задачі</u>

Студенту необхідно вибрати предметну область, де він є досить компетентим фахівцем і перевірити компентність у цій же області своїх товаришів. Для цього необхідно сформувати базу даних запитань (8-10) і відповідей, згідно із своїм варіантом, вибрати експертів (5), створити програмний фрагмент із достатньо дружнім інтерфейсом і визначити компетентність експертів.

Контрольні запитання.

1. Яким чином класифікуються запитання в залежності від відповідей?
2. Як формулюється задача визначення компетентності експертів?
3. Дайте характеристуику кожному із алгоритмів обробки запитань.

<u>*Вимоги до виконання лабораторних робіт (ЛР)*</u>

1. ЛР виконується в зошиті, або на аркушах формату А4.
2. Перелік необхідних складових звіту до ЛР:

2.1. Титульна сторінка
2.2. Зміст
2.3. Постановка задачі із своїм варіантом
2.4. Теоретична частина (основні відомості, вирази, використані в роботі)

2.5. Алгоритм роботи
2.6. Текст програми
2.7. Одержані результати
2.7 Аналіз результатів, висновки
2.8 Література
Виконана і оформлена робота представляється до захисту не пізніше початку наступної ЛР.

Література

1. Горбань А.Н. Обучение нейронных сетей. - М.: СП Параграф, 1990.

2. Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. -Новосибирск, Наука, 1996.

3. Горбань А.Н., Дунин-Барковский В.Л., Кирдин А.Н., Миркес Е.М., Новоходько А.Ю., Россиев Д.А., Терехов С.А., Сенашова Н.Ю., Царегородцев В.Г. Нейроинформатика. – Новосибирск: Наука. Сибирское предприятие РАН, 1998.

4. Куссуль Э.М. Ассоциативные нейроподобные структуры. - Киев, Наукова думка, 1990.

5. Нейрокомпьютеры и интеллектуальные роботы. Под ред. Н.М.Амосова. – Киев, Наукова думка, 1991.

6. Уоссермен Ф.. Нейрокомпьютерная техника : Теория и практика. - М.: Мир. 1992.

7. Снитюк В.Є., Рифат Мохаммед Али. Модели и методы определения компетентности экспертов на базе аксиомы несмещенности.// Вісник ЧІТІ. - №4. – 2000. – С.121-126.

8. Ивахненко А.Г. Долгосрочное прогнозирование и управление сложными системами. – Киев, Техніка, 1975.

Додаток А

```
procedure TNejro20.BitBtn1Click(Sender: TObject);
label 100;
var w,DeW,Wmin:array[0..100,1..100,1..2] of real;
yn:array[{0}1..100,0..100] of real;
minn,maxn:array[1..100] of real;
s,Delta,Prognoz:array[1..100,1..100] of real;
d:array[1..100] of real;
```

```
kkk,i,j,ii,ij:Byte; jjj,Number,count:integer;
MinErro,Erro,ppp:real; ll:integer;
begin
AssignFile(F2,'datanet.ren'); Rewrite(F2);
AssignFile(F3,'datagraf.ren'); Rewrite(F3);
Writeln(F2,'(C) Alternative S');
Writeln(F2,'Нейронна мережа - алгоритм оберненого поширення
похибки');
Writeln(F2,'Початкові дані:');
Writeln(F2,'Вхідних нейронів - ',SpinEdit1.Value);
Writeln(F2,'Вихідних нейронів - ',SpinEdit3.Value);
Writeln(F2,'Кількість нейронів прихованого зрізу - ',
SpinEdit2.Value);
Writeln(F2,'Початковий коефіцієнт навчання - ',
StrToFloat(Edit1.Text));
Writeln(F2,'Кількість точок навчання - ',SpinEdit5.Value);
Writeln(F2,'Похибка нейронної мережі - ',
StrToFloat(Edit3.Text));
Writeln(F2,'Кількість точок прогнозу - ',SpinEdit4.Value);
Writeln(F2,'Функція активації - ',ComboBox1.Text);
Writeln(F2,'Таблиця даних:');
For i:=1 to SpinEdit5.Value do begin
For j:=1 to SpinEdit1.Value+SpinEdit3.Value do
Write(F2,StrToFloat(StringGrid1.Cells[j-1,i]):10:4,'  ');
Writeln(F2);                                end;
Writeln(F2,'Точки для прогнозу:');
For i:=1 to SpinEdit4.Value do begin
For j:=1 to SpinEdit1.Value do
Write(F2,StrToFloat(StringGrid2.Cells[j-1,i]):10:4,'  ');
Writeln(F2);                                 end;
Number:=SpinEdit4.Value {кількість точок прогнозу};
l:=SpinEdit2.Value {кількість нейронів прихованого зрізу};
ll:=1; ij:=0; randomize;
For i:=0 to n {кількість нейронів вхідного зрізу}do
For j:=1 to l {кількість нейронів прихованого зрізу}do
w[i,j,1]:=random(1000)/1000{0.5};
Wmin[i,j,1]:=random(1000)/1000;
For i:=0 to l do
For j:=1 to m {кількість нейронів вихідного зрізу}do
w[i,j,2]:=random(1000)/1000{0.5};
Wmin[i,j,1]:=random(1000)/1000;
{Зчитуємо та нормуємо початкові дані}
For j:=1 to n+m do begin
maxn[j]:=StrToFloat(StringGrid1.Cells[j-1,1]);
minn[j]:=StrToFloat(StringGrid1.Cells[j-1,1]);
For i:=2 to SpinEdit5.Value do begin
if maxn[j]<StrToFloat(StringGrid1.Cells[j-1,i])
then maxn[j]:=StrToFloat(StringGrid1.Cells[j-1,i]);
if minn[j]>StrToFloat(StringGrid1.Cells[j-1,i])
then minn[j]:=StrToFloat(StringGrid1.Cells[j-1,i]);
                            end;end;
Writeln(F2,'Таблиця нормованих даних:'); randomize;
For jjj:=1 to 90 do begin
ll:=round(random(90))+1;
For i:=1 to n do begin {зчитуємо з початкової таблиці вхідні
значення навчальної послідовності}
```

```
yn[i,0]:=(StrToFloat(StringGrid1.Cells[i-1,ll])-
minn[i])/(maxn[i]-minn[i]);  Write(F2,yn[i,0]:10:4,'  ');end;
For j:=1 to m do  begin {зчитуємо з початкової таблиці вихідні
значення навчальноїпослідовності}
d[j]:=(StrToFloat(StringGrid1.Cells[n+j-1,ll])-
minn[n+j])/(maxn[n+j]-minn[n+j]);   Write(F2,d[j]:10:4,'  ');
end; Writeln(F2);
For j:=1 to n do
s[i,1]:=s[i,1]+yn[j,0]*w[j,i,1];
If ComboBox1.Text='Гіперболічний тангенс' then
yn[i,1]:=Sd(s[i,1]); {гіп. тангенс}
If ComboBox1.Text='Класичний сигмоїд' then yn[i,1]:=Sg(s[i,1]);
{клас. сигмоїд}
If ComboBox1.Text='Сигмоїд із зміщенням' then
yn[i,1]:=Sm(s[i,1]); {сигм. із зміщ.} end;
{рахуємо значення активаційної функції для вихідного зрізу}
For i:=1 to m do begin
s[i,2]:=0;{yn[0,1]:=1;}
For j:={0}1 to l do
S[i,2]:=s[i,2]+yn[j,1]*w[j,i,2];
If ComboBox1.Text='Гіперболічний тангенс' then
yn[i,2]:=Sd(s[i,2]); {гіп. тангенс}
If ComboBox1.Text='Класичний сигмоїд' then yn[i,2]:=Sg(s[i,2]);
{клас. сигмоїд}
If ComboBox1.Text='Сигмоїд із зміщенням' then
yn[i,2]:=Sm(s[i,2]); {сигм. із зміщ.} end;
{*********************************}
Erro:=0;
For j:=1 to m do begin  {Похибка останнього зрізу}
If ComboBox1.Text='Гіперболічний тангенс' then
Delta[j,2]:=(yn[j,2]-d[j])*(1-sqr(yn[j,2]));
If ComboBox1.Text='Класичний сигмоїд' then
Delta[j,2]:=(yn[j,2]-d[j])*(yn[j,2]*(1-yn[j,2]));
If ComboBox1.Text='Сигмоїд із зміщенням' then
Delta[j,2]:=(yn[j,2]-d[j])*(0.25-yn[j,2]*yn[j,2]);
Erro:=Erro+sqr((yn[j,2]-d[j]));{середня квадратична похибка}
end;
If MinErro>Erro  then begin
MinErro:=Erro;
For i:=1 to n do
For j:=1 to l do
Wmin[i,j,1]:=w[i,j,1];
For i:=1 to l do
For j:=1 to m do
Wmin[i,j,2]:=w[i,j,2];
                       end;
For i:=1 to m do {розрахунок для останнього зрізу дельта W}
For j:=1 to l do
DeW[j,i,2]:=-StrToFloat(Edit1.Text)*Delta[i,2]*yn[j,1];
For j:=1 to l do begin{Похибка Передостаннього зрізу}
If ComboBox1.Text='Гіперболічний тангенс' then
Delta[j,1]:=(Delta[1,2]*w[j,1,2]+Delta[2,2]*w[j,2,2])*(1-
sqr(yn[j,2]));
If ComboBox1.Text='Класичний сигмоїд' then
Delta[j,1]:=(Delta[1,2]*w[j,1,2]+Delta[2,2]*w[j,2,2])*(yn[j,2]*
(1-yn[j,2]));
```

```
If ComboBox1.Text='Сигмоїд із зміщенням' then
Delta[j,1]:=(Delta[1,2]*w[j,1,2]+Delta[2,2]*w[j,2,2])*(0.25-
yn[j,2]*yn[j,2]);                          end;
For i:=1 to l do{розрахунок для передостаннього зрізу дельта W}
For j:=1 to n do
DeW[j,i,1]:=-StrToFloat(Edit1.Text)*Delta[i,1]*yn[j,0];
For i:=1 to n do
For j:=1 to l do
w[i,j,1]:=w[i,j,1]+DeW[i,j,1];
For i:=1 to l do
For j:=1 to m do
w[i,j,2]:=w[i,j,2]+Dew[i,j,2];
           ll:=ll+1;  end;
100:
   Write(F2,'Помилка:');
Writeln(F2,'   ',Erro:7:7,'    ');
For i:=1 to Number do
For j:=1 to n do       {зчитуємо вхідні значення для прогнозу}
Prognoz[i,j]:=(StrToFloat(StringGrid2.Cells[j-1,i])-
minn[j])/(maxn[j]-minn[j]);
Writeln(F2,'Значення прогнозу');
For ii:=1 to Number do begin
For j:=1 to n do begin
yn[j,0]:=Prognoz[ii,j]{Prognoz[ii,j]*(maxn[j]-
minn[j])+minn[j]};
Write(F2,(yn[j,0]*(maxn[j]-minn[j])+minn[j]):5:5,'  ');end;
For i:=1 to l do begin
s[i,1]:=0;
For j:=1 to n do
S[i,1]:=s[i,1]+yn[j,0]*Wmin[j,i,1];
If ComboBox1.Text='Гіперболічний тангенс' then
yn[i,1]:=Sd(s[i,1]); {гіп. тангенс}
If ComboBox1.Text='Класичний сигмоїд' then yn[i,1]:=Sg(s[i,1]);
{клас. сигмоїд}
If ComboBox1.Text='Сигмоїд із зміщенням' then
yn[i,1]:=Sm(s[i,1]); {сигм. із зміщ.}
end;
For i:=1 to m do begin
s[i,2]:=0;
For j:=1 to l do
s[i,2]:=s[i,2]+yn[j,1]*Wmin[j,i,2];
If ComboBox1.Text='Гіперболічний тангенс' then
yn[i,2]:=Sd(s[i,2])*(maxn[n+i]-
minn[n+i])+minn[n+i]{Sd(s[i,2])};
If ComboBox1.Text='Класичний сигмоїд' then
yn[i,2]:=Sg(s[i,2])*(maxn[n+i]-
minn[n+i])+minn[n+i]{Sg(s[i,2])};
If ComboBox1.Text='Сигмоїд із зміщенням' then
yn[i,2]:=Sm(s[i,2])*(maxn[n+i]-
minn[n+i])+minn[n+i]{Sm(s[i,2])};
Write(F2,yn[i,2]:5:5,'  ');
StringGrid3.Cells[i-1,ii{1}]:=FloatToStr(yn[i,2]);
end; Writeln(F2); end; CloseFile(F2); CloseFile(F3); end;
```