

РОЗДІЛ 4

НЕЙРОННІ МЕРЕЖІ З ЛАТЕРАЛЬНИМИ ЗВ'ЯЗКАМИ

До нейронних мереж із латеральними (боковими) зв'язками відносять шаруваті нейромережі прямого поширення сигналу, у яких допускається наявність зв'язків між нейронами усередині одного шару.

Найбільш відомими мережами із латеральними зв'язками є мережі Кохонена.

4.1 Нейронна мережа Кохонена SOM

Карта Кохонена, що самоорганізується, (Kohonen Self-organizing Map – SOM) є НМ із латеральними зв'язками і відноситься до класифікаторів, для навчання яких використовуються вибірки образів із заздалегідь не заданою класифікацією.

Схему мережі SOM зображено на рис. 4.1. Вона складається з одного шару нейронів, які використовують як дискримінантну – функцію відстані, а як активаційну – функцію winner-take-all.

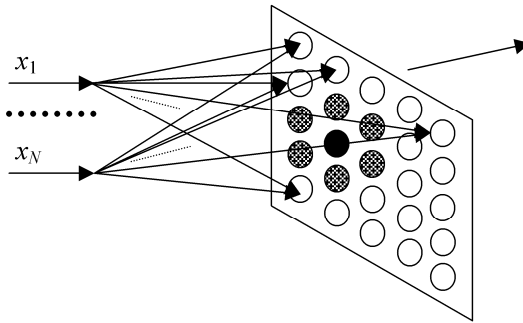


Рисунок 4.1 – Схема нейромережі SOM

Задачею мережі SOM є визначення належності вхідного вектора ознак s -го екземпляра вибірки $x^s = \{x^s_1, x^s_2, \dots, x^s_N\}^T$ до одного з L можливих кластерів, поданих векторними центрами $w_j = \{w_{j1}, w_{j2}, \dots, w_{jN}\}^T, j = 1, 2, \dots, L$, де T – символ транспонування.

Позначимо i -ту компоненту вхідного вектора x^s у момент часу t як $x^s_i(t)$, а вагу i -го входу j -го вузла у момент часу t як $w_{ij}(t)$.

Якщо вузли SOM є лінійними, а вага i -го входу j -го вузла дорівнює w_{ij} , $i = 1, 2, \dots, N$, $j = 1, 2, \dots, L$, тоді при відповідних значеннях порогів кожен i -й вихід мережі з точністю до несуттєвих постійних буде дорівнює евклідовій відстані d_j між пред'явленим вхідним вектором x_i^s і j -м центром кластера.

Вважається, що вектор x^s належить до j -го кластера, якщо відстань d_j для j -го центра кластера w_j мінімальна, тобто якщо $d_j \leq d_k$ для кожного $k \neq j$.

При навчанні НМ пред'являються вхідні вектори без указівки бажаних виходів і корегуються ваги відповідно до методу, що запропонував Т. Кохонен. *Метод Кохонена*, що формує SOM, вимагає, щоб біля кожного вузла було визначене поле NE , розмір якого з часом постійно зменшується.

Крок 1. Ініціюються ваги входів вузлів малими випадковими значеннями. Встановлюється початковий розмір поля NE .

Крок 2. Пред'являється новий вхідний вектор x^s .

Крок 3. Обчислюється відстань (метрика) d_j між вхідним вектором і кожним вихідним вузлом j :

$$d_j = \sum_{i=1}^N (x_i^s(t) - w_{ji}(t))^2.$$

Крок 4. Визначається вузол j^* з мінімальною відстанню d_j .

Крок 5. Корегуються ваги входів вузлів, що знаходяться в полі $NE_{j^*}(t)$ вузла j^* , таким чином, щоб нові значення ваг були:

$$w_{ji}(t+1) = w_{ji}(t) + \eta(t)(x_i^s - w_{ji}(t)), \quad j \in NE_{j^*}(t), \quad i = 1, 2, \dots, N.$$

При цьому коригувальний приріст $\eta(t)$ ($0 < \eta(t) < 1$) повинний спадати зі зростанням t .

Крок 6. Якщо збіжність не досягнута, то перейти до кроку 2.

Збіжність вважається досягнутою, якщо ваги стабілізувалися і коригувальний приріст η у кроці 5 знизився до нуля.

Якщо кількість вхідних векторів у навчальній множині є великою стосовно обраного числа кластерів, то після навчання ваги мережі будуть визначати центри кластерів, розподілені в просторі входів таким чином, що функція щільності цих центрів буде апроксимувати функцію щільності ймовірності вхідних

векторів. Крім того, ваги будуть організовані таким чином, що топологічно близькі вузли будуть відповідати фізично близьким (за евклідовою відстанню) вхідним векторам.

Важливо відзначити, що при класифікації за допомогою SOM, номер вузла, до якого віднесений екземпляр, і фактичний номер його класу в загальному випадку не збігаються – розділяючи екземпляри, SOM робить суб'єктивну класифікацію, що не має того реального фактичного змісту, яким ми наділяємо класи.

Результати класифікації SOM можуть бути наділені фактичним змістом шляхом постановки у відповідність номеру кожного вузла SOM номера того фактичного класу, до якого відноситься більша частина екземплярів навчальної вибірки, віднесених SOM до даного вузла.

4.2 Нейронна мережа Кохонена LVQ

Квантування навчальних векторів (Learning Vector Quantization – LVQ) є продовженням і розвитком ідеї SOM.

Нейронна мережа LVQ (рис. 4.2) складається з двох послідовно з'єднаних шарів нейронів: конкуруючого шару (SOM) і лінійного шару. Обидва шари НМ LVQ містять по одному конкуруючому та одному лінійному нейрону на кожен підклас (кластер) та цільовий клас. Позначимо S^1 – кількість підкласів (кластерів), S^2 – кількість цільових класів (S^1 завжди буде більше, ніж S^2).

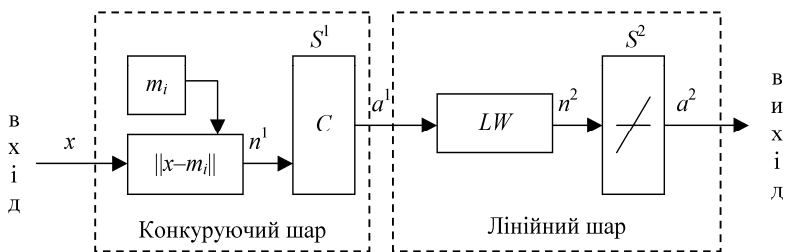


Рисунок 4.2 – Схема нейронної мережі LVQ у векторній формі

Конкуруючий шар являє собою мережу SOM і здійснює поділ вхідних векторів x на класи (кластери), виділяючи центри зосередження вхідних векторів (центри, еталони кластерів) m_i .

Для цього визначаються відстані $n^1 = \|x - m_i\|$ між вхідними векторами x і початковими значеннями центрів зосередження векторів m_i , $i = 1, 2, \dots, q$, де q – кількість вхідних векторів.

Лінійний шар містить нейрони, що використовують дискримінантну функцію зважена сума та лінійну функцію активації. Він перетворює клас вхідного вектора, визначений конкуруючим шаром – підклас a^1 , у клас, визначений користувачем – цільовий клас a^2 , шляхом множення a^1 на значення ваг LW лінійних нейронів, що встановлюються рівними 1, якщо цільовий клас і підклас збігаються і 0 – у протилежному випадку. Відповідні добутки $n^2 = a^1 LW$ подаються на виходи всіх лінійних нейронів, утворюючи бінарний вектор a^2 , всі елементи якого рівні 0, за винятком елемента, що відповідає цільовому класу (цей елемент дорівнює 1).

4.3 Методи навчання мережі LVQ

В основі методів LVQ лежить механізм навчання шару конкуруючих нейронів (конкуруючого шару), контрольований вчителем.

Конкуруючий шар може автоматично навчатися класифікувати вхідні вектори. Поділ класів, що визначає карта Кохонена, заснований тільки на відстані між вхідними векторами. Якщо два вхідних вектори дуже близькі, то конкуруючий шар, дуже ймовірно, віднесе їх до одного класу. Однак, поділ на класи, вироблений конкуруючим шаром, як правило, не збігається з тим, що визначає вчитель. Виникає задача розробки механізму, який би дозволяв карті Кохонена здійснювати класифікацію, близьку до заданої вчителем. Методом, що дозволяє вирішувати поставлену задачу, є LVQ.

Метод навчання LVQ1. Нехай деяка кількість векторів з вільними параметрами (центрів кластерів) m_i поміщена у вхідний простір для апроксимації різних областей вхідного вектора x їх квантованими значеннями. Кожному класу значень x призначається кілька векторів з вільними параметрами, і потім приймається рішення про віднесення x до того класу, до якого належить найближчий вектор m_i . Нехай індекс c визначає найближчий до x вектор m_i , позначений далі як m_c .

$$c = \arg \min_i \{ \|x - m_i\| \} .$$

Значення для m_i , що мінімізують помилку класифікації, можуть бути знайдені як асимптотичні значення в наступному процесі навчання. Нехай $x(t)$ – екземпляр, поданий на входи, $y(t)$ – номер класу, до якого відноситься екземпляр $x(t)$, $m_i(t)$ – подання послідовності m_i , дискретизованої за часом, $Y_c(t)$ – номер класу, до якого відноситься $m_i(t)$. Починаючи з правильно визначених початкових значень, основний процес методу LVQ1 визначають правилами:

$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)], y(t) = Y_c(t);$$

$$m_i(t+1) = m_i(t) - \alpha(t)[x(t) - m_i(t)], y(t) \neq Y_c(t);$$

$$m_i(t+1) = m_i(t), \forall i \neq c.$$

Тут $0 < \alpha(t) < 1$, $\alpha(t)$ може бути константою або монотонно зменшуватися з часом. Для методу LVQ1 рекомендується, щоб первісне значення α було менше 0,1.

Метод навчання LVQ2 вирішує задачу класифікації ідентично методу LVQ1. Однак у процесі навчання LVQ2 два вектори з вільними параметрами m_i і m_j , що є найближчими сусідами x , модифікуються одночасно. При цьому вони повинні належати до різних класів. Крім того, x повинний знаходитися в зоні значень, що називається «вікном», яке визначене навколо середини площини, утвореної векторами m_i і m_j . Нехай d_i і d_j – евклідові відстані x від m_i і m_j , тоді x визначено потрапить у «вікно» відносної ширини w , якщо

$$\min \left(\frac{d_i}{d_j}, \frac{d_j}{d_i} \right) > s,$$

де $s = \frac{1-w}{1+w}$.

Рекомендується, щоб значення відносної ширини «вікна» w знаходилися в межах від 0,2 до 0,3.

Метод навчання LVQ2 має вид:

$$m_i(t+1) = m_i(t) - \alpha(t)[x(t) - m_i(t)],$$

$$m_j(t+1) = m_j(t) + \alpha(t)[x(t) - m_j(t)],$$

де m_i і m_j – два найближчих до x вектора з вільними параметрами,

причому x і m_j належать до одного класу, у той час як x і m_i належать до різних класів, крім того, x повинний потрапляти в «вікно».

На рис. 4.3 квадратами позначені значення ознак екземплярів, а хрестами – значення ваг НМ LVQ. Штрихування фігур позначає їхню приналежність до класу (суцільна – клас 1, пунктирна – клас 2). Помилково класифіковані екземпляри виділені колами. На рис. 4.3, *а* показане початкове розташування ваг НМ LVQ – всі екземпляри класифіковані невірно. На рис. 4.3, *б* для тих же екземплярів показане положення ваг НМ LVQ після однієї ітерації навчання за допомогою методу LVQ2 – одночасно модифіковані значення двох векторів, кількість неправильно класифікованих екземплярів зменшилася. На рис. 4.3, *в* показане положення ваг НМ LVQ після 10 ітерацій навчання – усі екземпляри класифіковані вірно.

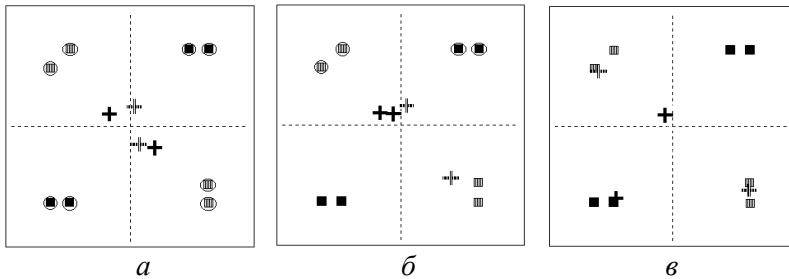


Рисунок 4.3 – Процес квантування навчальних векторів:

а – початкове положення; *б* – після першої ітерації;
в – після десяти ітерацій навчання

Метод LVQ3. Метод навчання LVQ2 заснований на ідеї диференціального зсуву меж рішення відносно Байєсовських меж, при цьому не враховується те, що може відбутися з положенням m_i у випадку досить тривалої роботи методу. Отже, необхідно внести зміни, що гарантували б, що m_i хоча б грубо продовжить апроксимацію розподілу класів. При поєднанні цих ідей, ми одержуємо поліпшений метод – LVQ3:

$$m_i(t+1) = m_i(t) - \alpha(t)[x(t) - m_i(t)],$$

$$m_j(t+1) = m_j(t) + \alpha(t)[x(t) - m_j(t)],$$

Для $k \in \{i, j\}$, якщо x , m_i і m_j належать тому самому класу:

$$m_k(t+1) = m_k(t) + \varepsilon\alpha(t)[x(t) - m_k(t)],$$

де m_i і m_j – два найближчих до x вектора з вільними параметрами, причому x і m_j належать до тому самому класу, у той час як x і m_i належать різним класам, крім того, x повинний попадати в «вікно».

У результаті ряду експериментів було встановлено, що значення ε повинні знаходитися між 0,1 і 0,5. Оптимальне значення ε , можливо, залежить від розміру найменшого вікна.

Цей метод є таким, що самостабілізується, тобто оптимальне розміщення m_i не змінюється при тривалому навчанні.

Метод навчання OLVQ1 (Optimized-learning-rate LVQ1) являє собою метод LVQ1, модифікований таким чином, щоб кожному m_i була призначена індивідуальна швидкість навчання $\alpha_i(t)$. Таким чином, ми одержуємо наступний дискретизований за часом процес навчання.

Нехай c визначається рівнянням:

$$c = \arg \min_i \|x - m_i\| .$$

Тоді встановити:

$-m_c(t + 1) = m_c(t) + \alpha_c(t)[x(t) - m_c(t)]$, якщо x класифікований правильно;

$-m_c(t + 1) = m_c(t) - \alpha_c(t)[x(t) - m_c(t)]$, якщо x класифікований неправильно;

$$-m_i(t + 1) = m_i(t) , \forall i \neq c .$$

Розглянемо спосіб визначення оптимального $\alpha_i(t)$ для найбільш швидкої збіжності OLVQ1. Виразимо розглянуті рівняння у формі:

$$m_c(t + 1) = [1 - s(t)\alpha_c(t)] m_c(t) + s(t)\alpha_c(t) x(t),$$

де $s(t) = +1$, якщо класифікація правильна і $s(t) = -1$, якщо класифікація неправильна.

Відзначено, що $m_c(t)$ статистично не залежить від $x(t)$, і статистична точність отриманих значень векторів з вільними параметрами оптимальна, якщо результати виправлень зроблені в різний час.

Помітимо, що $m_c(t + 1)$ містить слід $x(t)$ через останній член в останньому рівнянні і просліджує попередні $x(t')$, $t'=1, 2, \dots, t-1$ через $m_c(t)$.

Абсолютна величина останнього сліду $x(t)$ масштабується коефіцієнтом $\alpha_c(t)$ і, у свою чергу, слід $x(t - 1)$ масштабується коефіцієнтом $[1 - s(t) \alpha_c(t)] \alpha_c(t - 1)$.

Тепер передбачимо, щоб обоє ці масштабування були ідентичні

і застосуємо дана умова для всіх t . Тоді «сліди» усіх попередніх x , зібрані до часу t , будуть наприкінці масштабуватися одним числом, і, отже, «оптимальні» значення $\alpha_c(t)$ визначаються рекурсивно:

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t)}.$$

На практиці можна переконатися, що це правило забезпечує швидку збіжність.

Однак, помітно, що $\alpha_c(t)$ може також збільшуватися, і важливо, щоб значення $\alpha_c(t)$ не перевищувало 1. Початкові значення $\alpha_c(t)$ можуть бути обрані досить високими (наприклад, 0,3), завдяки чому навчання значне прискорюється (особливо на початку) і наближені асимптотичні значення m_i знаходяться досить швидко.

Варто звернути увагу на те, що розглянуте вираження не застосовне для LVQ2, тому що α_i , у середньому, не буде зменшуватися і процес не буде сходитися.



4.4 Приклади виконання завдань

Приклад 1. Нехай задано розпізнаваний екземпляр $x = (2,3 \ 1 \ -2 \ 0)^T$, поданий на входи мережі SOM, яка має три нейрони. Ваги нейронів дорівнюють: $w^{(1,1)} = (0 \ 3 \ -2 \ 1)^T$, $w^{(1,2)} = (-1 \ 8 \ 5 \ 3)^T$ та $w^{(1,3)} = (1 \ 1 \ -1 \ 1)^T$, а пороги вважаються відсутніми. Визначити значення на виходах нейронів SOM.

Спочатку визначимо для нейронів значення дискримінантних функцій:

$$\varphi^{(1,1)}(w^{(1,1)}, x^{(1,1)}) = (0 - 2,3)^2 + (3 - 1)^2 + (-2 - (-2))^2 + (1 - 0)^2 = 10,29;$$

$$\varphi^{(1,2)}(w^{(1,2)}, x^{(1,2)}) = (-1 - 2,3)^2 + (8 - 1)^2 + (5 - (-2))^2 + (3 - 0)^2 = 117,89;$$

$$\varphi^{(1,3)}(w^{(1,3)}, x^{(1,3)}) = (1 - 2,3)^2 + (1 - 1)^2 + (-1 - (-2))^2 + (1 - 0)^2 = 3,69.$$

Після цього для нейронів визначимо значення функції активації «переможець забирає усе»: $\psi^{(1,1)} = 0$, $\psi^{(1,2)} = 0$, $\psi^{(1,3)} = 1$.

Приклад 2. Нехай задано розпізнаваний екземпляр $x = (2 \ -1 \ 1)^T$, поданий на входи мережі LVQ, яка має два нейрони на першому

шарі та два нейрони на другому шарі. Ваги нейронів першого шару дорівнюють: $w^{(1,1)}=(3 \ -2 \ 1)^T$ та $w^{(1,2)}=(8 \ 5 \ 3)^T$, а пороги – одиниці (або вважаються відсутніми). Ваги нейронів другого шару дорівнюють: $w^{(2,1)}=(1 \ 0)^T$ та $w^{(2,2)}=(0 \ 1)^T$, а пороги – нулю. Визначити номер класу, до якого має віднести мережа розпізнаваний екземпляр x .

Спочатку визначимо для нейронів першого шару значення дискримінантних функцій:

$$\varphi^{(1,1)}(w^{(1,1)}, x^{(1,1)}) = (3 - 2)^2 + (-2 - (-1))^2 + (1 - 1)^2 = 2;$$

$$\varphi^{(1,2)}(w^{(1,2)}, x^{(1,2)}) = (8 - 2)^2 + (5 - (-1))^2 + (3 - 1)^2 = 76.$$

Після чого для нейронів першого шару визначимо значення функції активації «переможець забирає усе»: $\psi^{(1,1)}=1$, $\psi^{(1,2)}=0$.

Далі визначимо для нейронів другого шару значення дискримінантних функцій (а через те, що їхні функції є лінійними, то фактично і значення на виходах нейронів):

$$y^{(2,1)} = \varphi^{(2,1)}(w^{(2,1)}, x^{(2,1)}) = 1 \cdot 1 + 0 \cdot 0 = 1;$$

$$y^{(2,2)} = \varphi^{(2,2)}(w^{(2,2)}, x^{(2,2)}) = 0 \cdot 1 + 1 \cdot 0 = 0.$$

? 4.5 Контрольні питання

1. У чому полягає відмінність мереж Кохонена між собою та у порівнянні з іншими нейромережами?

2. Запропонуйте способи урахування апріорної інформації про значимість ознак при навчанні SOM.

3. Запропонуйте стратегію (метод) визначення необхідної кількості нейронів конкуруючого шару LVQ.

4. Обмеження мереж LVQ.

5. Опишіть структуру мережі LVQ.

6. Опишіть структуру мережі SOM.

7. Порівняйте мережі Кохонена з бінарною мережею Хопфілда.

8. Порівняйте мережі Кохонена з мережею Елмана.

9. Порівняйте мережі Кохонена з перцептронами.

10. Порівняйте можливості SOM, мереж Хопфілда та одношарового дискретного перцептрона.

11. У чому полягає метод навчання LVQ1?

12. У чому полягає метод навчання LVQ2?

13. У чому полягає метод навчання LVQ3?

14. У чому полягає метод навчання OLVQ?

15. У чому полягає навчання мережі Кохонена LVQ?


16. Чи доцільно використовувати SOM для попереднього аналізу даних у задачах діагностики і прогнозування?

17. Як впливають вид і параметри метрики на точність визначення центрів зосередження екземплярів. Відповідь поясніть і проілюструйте.


18. Які задачі можна вирішувати на основі мереж SOM і LVQ, а які не можна? Обґрунтуйте і доведіть відповідь. Наведіть приклади.


19. Які метрики можна використовувати для побудови мереж SOM?

4.6 Практичні завдання

 Завдання 1. Підготувати реферат на одну з таких тем.

1. Застосування мереж Кохонена у технічній діагностиці.
2. Застосування мереж Кохонена у медичній діагностиці.
3. Застосування мереж SOM для кластер-аналізу.
4. Застосування мереж SOM у картографії.
5. Застосування мереж SOM для створення пошукових систем.
6. Застосування мереж Кохонена для розпізнавання мови.
7. Застосування мереж SOM для планування експерименту.
8. Нейронні мережі з латеральним гальмуванням.

 Завдання 2. Нехай задано розпізнаваний екземпляр $x = (3 \ -4 \ 1 \ 2 \ 3)^T$, поданий на входи мережі SOM, яка має п'ять нейронів. Ваги нейронів дорівнюють: $w^{(1,1)} = (0 \ 3 \ -2 \ 1 \ 5)^T$, $w^{(1,2)} = (-1 \ 8 \ 5 \ 3 \ 3)^T$ та $w^{(1,3)} = (1 \ 1 \ -1 \ 1 \ 2)^T$, $w^{(1,4)} = (0 \ 7 \ -1 \ 2 \ 5)^T$ та $w^{(1,5)} = (-2 \ 9 \ 4 \ 1 \ 2)^T$, а пороги вважаються відсутніми. Визначити значення на виходах нейронів SOM.

 Завдання 3. Нехай задано розпізнаваний екземпляр $x = (1 \ 3 \ 2 \ 3)^T$, поданий на входи мережі LVQ, яка має чотири нейрони на першому шарі та два нейрони на другому шарі. Ваги нейронів першого шару дорівнюють: $w^{(1,1)} = (1 \ 2 \ 1 \ 1)^T$, $w^{(1,2)} = (2 \ 3 \ 2 \ 3)^T$, $w^{(1,3)} = (3 \ 3 \ 3 \ 3)^T$ та $w^{(1,4)} = (1 \ -1 \ 2 \ 3)^T$, а пороги вважаються відсутніми. Ваги нейронів другого шару дорівнюють: $w^{(2,1)} = (1 \ 0 \ 0 \ 1)^T$ та $w^{(2,2)} = (0 \ 1 \ 1$

0)^T, а пороги – нулю. Визначити номер класу, до якого має віднести мережа розпізнаваний екземпляр x .



Завдання 4. Емуляція та навчання мереж Кохонена SOM.

1. Задати число класів, кількість входів нейронів і розмір мережі (число нейронів), набір пар значень входів і бажаних виходів, а також вид метрики SOM.

2. Написати на алгоритмічній мові програмування високого рівня програму, що моделює SOM і реалізує алгоритми її навчання. Передбачити в програмі відображення на екран і запис у файл на диску поточного стану параметрів НМ і результатів їхньої роботи: матриці ваг, значення на входах і виходах, тривалість їхнього навчання і роботи.

3. Для заданих даних здійснити навчання SOM і зберегти у файлі на диску результати її роботи.

4. Виконати п. 3, варіюючи по черзі вид / параметри метрики SOM. Результати занести в таблицю, стовпці якої повинні мати назви: вид метрики, помилка поділу на класи (число помилкових рішень про віднесення екземплярів до вузлів, до яких вони не мають фактичної близькості). Проаналізувати отримані результати і зробити висновки про те, як впливає вид метрики на число помилкових рішень.

5. Проаналізувати результати виконання п.3–п.4.



Завдання 5. Емуляція та навчання мереж Кохонена LVQ.

1. Задати число класів, кількість входів нейронів і розмір мережі (число нейронів), набір пар значень входів і бажаних виходів, а також вид метрики.

2. Написати на алгоритмічній мові програмування високого рівня програму, що моделює LVQ і реалізує методи її навчання. Передбачити в програмі відображення на екран і запис у файл на диску поточного стану параметрів НМ і результатів їхньої роботи: матриці ваг, значення на входах і виходах, тривалість їхнього навчання і роботи.

3. Для заданих даних здійснити навчання LVQ і зберегти у файлі на диску результати її роботи.

4. Виконати п. 3, варіюючи по черзі вид / параметри метрики. Результати занести в таблицю, стовпці якої повинні мати назви: вид

метрики, помилка поділу на класи (число помилкових рішень про віднесення екземплярів до вузлів, до яких вони не мають фактичної близькості). Проаналізувати отримані результати і зробити висновки про те, як впливає вид метрики на число помилкових рішень.

5. Проаналізувати результати виконання п. 3–4.