

3.3 Методологічні засади навчання штучних нейронних мереж в інтелектуальних системах підтримки прийняття рішень

Вступ

Системи підтримки прийняття рішень (СППР) активно використовуються в усіх сферах життєдіяльності людей. Особливого поширення вони отримали при обробці великих масивів даних, забезпечення інформаційної підтримки процесу прийняття рішень особами, що приймають рішення. Проведений аналіз відомих наукових досліджень [192-201] свідчить про те, що на даний час основою існуючих СППР становлять методи штучного інтелекту.

Створення інтелектуальних СППР стало природним продовженням широкого застосування СППР класичного типу. Інтелектуальні СППР забезпечують інформаційну підтримку всіх виробничих процесів і служб підприємств (організацій, установ), включаючи проектування, виготовлення і збут продукції, фінансово-економічний аналіз, планування, управління персоналом, маркетинг, супровід створення (експлуатації, ремонту) виробів та перспективне планування. Також зазначені інтелектуальні СППР знайшли широке використання для вирішення специфічних завдань військового призначення, а саме [192, 193]: планування розгортання, експлуатації систем зв'язку та передачі даних; автоматизація управління військами та зброєю; збір, обробка та узагальнення розвідувальних відомостей про стан об'єктів розвідки та інш.

Основним інструментом для вирішення розрахункових та інших завдань в сучасних інтелектуальних СППР є штучні нейронні мережі (ШНМ), що еволюціонують. ШНМ, що еволюціонують мають як універсальні апроксимуючі властивості, так і можливості нечіткого виводу. Вони одержали широке поширення для розв'язку різних завдань інтелектуального аналізу даних, ідентифікації, емуляції, прогнозування, інтелектуального управління та т.п. ШНМ, що еволюціонують забезпечують стабільну роботу в умовах нелінійності, невизначеності, стохастичності та хаотичності, різного роду збурювань і завад.

Незважаючи на досить успішне їхнє застосування для розв'язку широкого кола завдань інтелектуального аналізу даних, ці системи мають ряд недоліків, пов'язаних з їхнім використанням.

Серед найбільш істотних недоліків можна виділити такі:

1. Складність вибору архітектури системи. Як правило, модель, заснована на принципах обчислювального інтелекту, має фіксовану архітектуру. У контексті ШНМ це означає, що нейронна мережа має фіксовану кількість нейронів і зв'язків. У зв'язку із цим, адаптація системи до нових даних, що надходять на обробку, що мають відмінну від попередніх даних природу, може виявитися проблематичним.

2. Навчання в пакетному режимі та навчання протягом декількох епох вимагає значних часових ресурсів. Такі системи не є пристосовані для роботи в online режимі з досить високим темпом надходження нових даних на обробку.

3. Багато з існуючих систем обчислювального інтелекту не можуть визначати правила, що еволюціонують, по яких відбувається розвиток системи, а також можуть представляти результати своєї роботи в термінах природньої мови. Таким чином, актуальним є завдання розробки нових методів навчання для ШНМ, які дозволять вирішити зазначені труднощі.

3.3.1 Аналіз літературних даних та постановка проблеми

В роботі [194] проведено аналіз властивостей ШНМ, які використовувалися при прогнозуванні концентрації забруднюючих речовин повітря. В роботі акцентувалася увага на те, що ШНМ мають низьку швидкість конвергенції та локальний мінімум. Запропоновано використання екстремальної навчальної машини для ШНМ, що забезпечує високу ефективність узагальнення при надзвичайно високій швидкості навчання. До недоліків зазначеного підходу слід віднести накопичення помилок ШНМ в ході проведення обчислень, неможливість вибору параметрів та типу функції належності.

В роботі [196] наведено оперативний підхід з просторового аналізу в морській галузі для кількісної оцінки та відображення супутніх екосистемних послуг. Такий підхід охоплює тривимірність морського середовища,

розглядаючи окремо всі морські області (морська поверхня, водяний стовп та морське дно). Фактично, метод будує 3-вимірні моделі моря шляхом оцінки та картографування, що пов'язані з кожним із трьох морських доменів через прийняття репрезентативних показників. До недоліків зазначеного методу слід віднести неможливість гнучкого налаштування (адаптації) оціночних моделей при додаванні (виключенні) показників і зміні їх параметрів (сумісності та значущості показників).

В роботі [197] представлена модель машинного навчання для автоматичної ідентифікації запитів та надання інформаційних служб підтримки, що обмінюються між членами інтернет-спільноти. Зазначена модель призначена для обробки великої кількості повідомлень користувачів соціальних мереж. Недоліками зазначеної моделі є відсутність механізмів оцінки адекватності рішень, що приймаються та велика обчислювальна складність.

В роботі [198] представлено використання ШНМ для виявлення аномалії серцевого ритму та інших захворювань серця. В якості методу навчання ШНМ використовується алгоритм зворотного поширення помилки. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [199] представлено використання ШНМ для виявлення сходження лавин. В якості методу навчання ШНМ використовується алгоритм зворотного поширення помилки. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [200] представлено використання ШНМ для виявлення проблем виявлення аномалій в системах домашньої авторизації. В якості методу навчання ШНМ Кохонена використовується алгоритм “переможець отримує все”. Недоліками зазначеного підходу є накопичення помилки в процесі навчання, обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [201] представлено використання ШНМ для виявлення проблем виявлення аномалій в енцефалограмі людини. В якості методу навчання ШНМ використовується метод тонкого налаштування параметрів ШНМ. Недоліками зазначеного підходу є накопичення помилки в процесі навчання, обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [203] представлено використання методів машинного навчання, а саме ШНМ та генетичних алгоритмів. В якості методу навчання ШНМ використовується генетичний алгоритм. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [204] представлено використання методів машинного навчання, а саме ШНМ та методу диференційного пошуку. В ході дослідження проведено розробку гібридного методу навчання ШНМ, що заснований на використанні алгоритму зворотного поширення помилки та диференційного пошуку. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [205] проведено розробку методів навчання ШНМ з використанням комбінованої апроксимації поверхні відгуку, який забезпечує найменші похибки навчання і прогнозування. Недоліком зазначеного методу накопичення помилки в ході навчання та неможливість зміни архітектури ШНМ в ході навчання.

В роботі [206] наведено використання ШНМ для оцінки ефективності роботи агрегату, використовуючи попередній часовий ряд його продуктивності. Для навчання ШНМ використовуються моделі SBM (Stochastic Block Model) та DEA (Data Envelopment Analysis). Недоліками зазначеного підходу є обмеженість в виборі архітектури мережі, навчання тільки синаптичних ваг.

В роботі [207] наведено використання ШНМ для оцінки геомеханічних властивостей. В якості методу навчання ШНМ використовується алгоритм зворотного поширення помилки. Покращення характеристик алгоритму зворотного поширення помилки досягається за рахунок збільшення навчальної

вибірки. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

В роботі [208] наведено використання ШНМ для оцінки інтенсивності дорожнього руху. В якості методу навчання ШНМ використовується алгоритм зворотного поширення помилки. Покращення характеристик алгоритму зворотнього поширення помилки досягається за рахунок використання пропускових з'єднань між кожним шаром, так що кожен шар викладає лише залишкову функцію щодо результатів попереднього шару. Недоліками зазначеного підходу є його обмеженість навчанням лише синаптичних ваг, без навчання виду та параметрів функції належності.

Проведення аналізу відомих наукових праць [192–208] показав, що для навчання штучних нейронних мереж використовуються загальновідомі методи навчання. Зазначені методи орієнтовані як правило на навчання синаптичних ваг або функції належності. Використання відомих алгоритмів (методів, методик) навчання штучних нейронних мереж навіть з покращеними характеристиками не задовольняє існуючим та перспективним вимогам, що висувуються до них.

Метою дослідження є розробка алгоритму навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень, яка дозволяє виконувати обробку більшої кількості інформації, при однозначності рішень, що приймаються.

Для досягнення мети були поставлені такі завдання: провести постановку завдання навчання штучних нейронних мереж; провести розробку алгоритму навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень; визначити переваги та недоліки запропонованого алгоритму.

Архітектура, багат шарової нейро-фаззі системи, що еволюціонує наведена на рис. 1. складається з п'яти послідовних шарів. На вхідний (нульовий) шар нейро-фаззі системи подається $(n \times 1)$ - мірний вектор сигналів-образів $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$ (тут $k=1, 2, \dots$ - поточний дискретний час), що підлягають обробці.

Перший прихований шар містить nh функцій належності, $\mu_{li}(x_i), i=1, 2, \dots, n; l=1, 2, \dots, h$. Таким чином, для кожного входу задається h функцій належності. Перший прихований шар виконує фазифікацію вхідного простору. При цьому мається на увазі, що в процесі навчання-еволюції повинні настроюватися як власне параметри цих функцій, так і їх число: на рис. 1 – кількість вузлів першого прихованого шару μ_h .

Другий прихований шар забезпечує агрегування рівнів належності, розрахованих у першому прихованому шарі, і складається з h блоків множення.

Третій прихований шар – це шар, що настроює синаптичні ваги $w_1, w_2, \dots, w_h$, які підлягають визначенню в процесі контрольованого навчання.

Четвертий прихований шар утворено двома суматорами, що обчислює суми вихідних сигналів другого та третього прихованого шарів.

І, нарешті, в п'ятому (вихідному) шарі проводиться дефазифікація, у результаті якої обчислюється вихідний сигнал NFS (neuro-fuzzy system) $\hat{y}(k)$.

Таким чином, якщо на вхід нейро-фазі системи поданий векторний сигнал $x(k)$, елементи першого прихованого шару роблять його фазифікацію, обчислюючи рівні належності $0 < \mu_{li}(x_i(k)) \leq 1$. Звичайно в якості функцій належності використовуються традиційні гаусіани

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma_i^2}\right), \quad (1)$$

де c_{li} – параметр центру l -ї функції належності i -го входу, σ_i – параметр ширини функції належності i -го входу.

Варто відзначити, що попереднє нормування даних на деякий інтервал, наприклад, $-1 \leq x_i(k) \leq 1$ дозволяє спростити розрахунки, оскільки параметри ширини σ_i можуть бути прийняті однаковими для всіх входів, тобто $\sigma_i = \sigma$.

Крім гаусіанів (1) можуть бути використані й інші ядерні функції, наприклад, В-сплайни, що відповідають умові одиничної розбивки, парні вейвлети, гнучкі функції активації-належності [209, 210] та інш.

У другому прихованому шарі обчислюються агреговані значення

$\prod_{i=1}^n \mu_{li}(x_i(k))$, при цьому для гаусіанів з однаковими параметрами ширини:

$$\prod_{i=1}^n \mu_{li}(x_i(k)) = \prod_{i=1}^n \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2}\right) = \exp\left(-\frac{\|x(k) - c_l\|^2}{2\sigma^2}\right),$$

де $c_l = (c_{l1}, c_{l2}, \dots, c_{ln})^T$.

Таким чином, сигнали на виходах блоків множення другого прихованого шару аналогічні сигналам на виходах нейронів першого прихованого шару звичайних радіально-базисних нейронних мереж (RBFN – Radial basis function networks) [211]. Виходами третього прихованого шару є значення $w_l \prod_{i=1}^n \mu_{li}(x_i(k))$,

четвертого – $\sum_{l=1}^h w_l \prod_{i=1}^n \mu_{li}(x_i(k))$ і $\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))$, нарешті, на виході системи (п'ятого вихідного шару) з'являється сигнал

$$\hat{y}(x(k)) = \frac{\sum_{l=1}^h w_l \prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} = \sum_{l=1}^h w_l \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} = \sum_{l=1}^h w_l \varphi_l(x(k)) = w^{hT} \varphi^h(x(k)),$$

де

$$\varphi_l(x(k)) = \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))}, \quad w^h = (w_1, w_2, \dots, w_h)^T,$$

$$\varphi^h(x(k)) = (\varphi_1(x(k)), \varphi_2(x(k)), \dots, \varphi_h(x(k)))^T.$$

Нескладно помітити, що розглянута система реалізує нелінійне відображення вхідного простору в скалярний вихідний сигнал $(R^n \rightarrow R^1)$ подібно нормалізованої RBFN [212], а по архітектурі (при фіксованому h) збігається з TSK-системою (Takagi, Sugeno, Kang'a) нульового порядку, тобто архітектурою Ванга–Менделя [213].

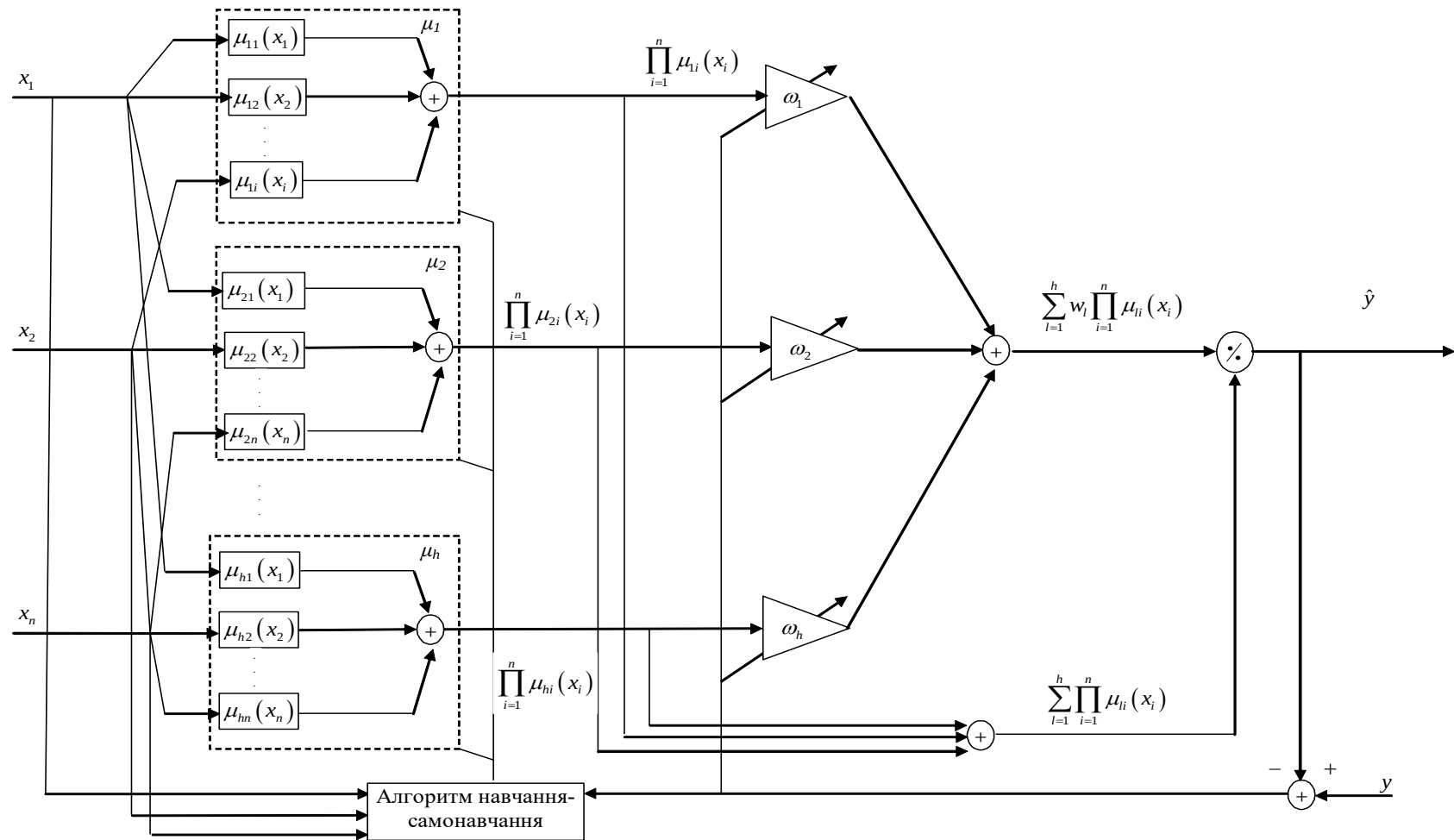


Рис. 1 Архітектура, багатошарової нейро-фаззі системи, що еволюціонує

Для того щоб зрозуміти взаємозв'язок запропонованого алгоритму в процесі функціонування ШНМ авторами було побудовано типовий алгоритм функціонування ШНМ з алгоритмом навчання ШНМ (рис. 2).

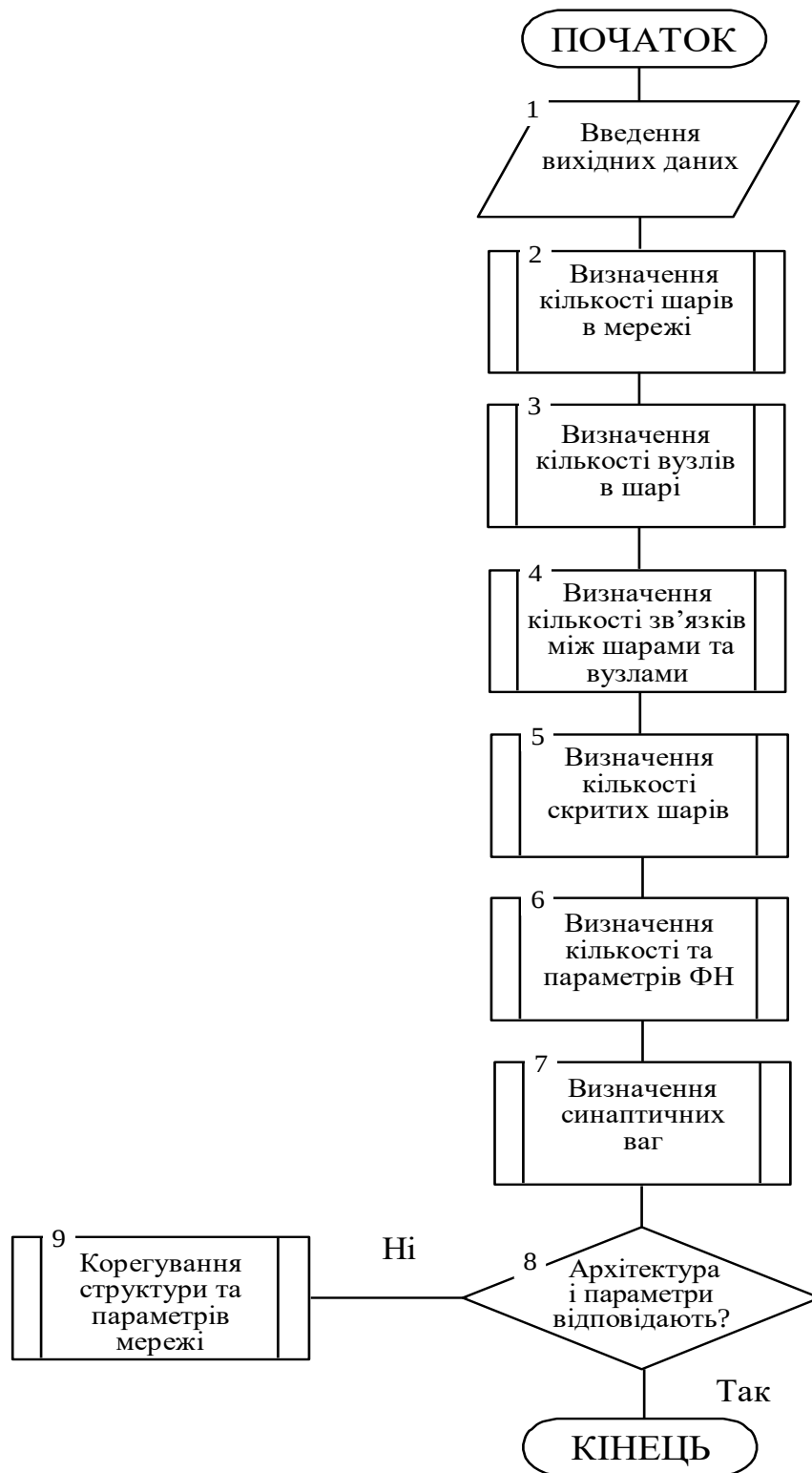


Рис. 2 Алгоритм функціонування та навчання штучної нейронної мережі, що еволюціонує

Початковим етапом є введення вихідних даних, а саме початкової архітектури та параметрів штучної нейронної мережі (етап 1-4).

Етап 1. Визначення кількості шарів в штучній нейронній мережі.

Етап 2. Визначення кількості вузлів в шарі штучної нейронної мережі.

Етап 3. Визначення кількості зв'язків між шарами та вузлами штучної нейронної мережі.

Етап 4. Визначення кількості скритих шарів штучної нейронної мережі.

Етап 5. Визначення кількості та параметрів функції належності (ФН).

Етап 6. Визначення синаптичних ваг зв'язків штучної нейронної мережі.

Етап 7. Перевірка відповідності архітектури та параметрів штучної нейронної мережі вимогам, що висуваються.

Етап 8. Прийнятті рішення на корегування архітектури та параметрів штучної нейронної мережі.

Розглянемо докладно етапи запропонованого алгоритму навчання.

Етап 5. Налаштування параметрів і кількості функцій належності.

Процес налаштування параметрів і кількості функцій належності відбувається в такий спосіб. Нехай на вхід системи, у якої в вихідному стані в першому прихованому шарі відсутні функції належності, надходить перше спостереження навчальної вибірки $x(1) = (x_1(1), x_2(1), \dots, x_n(1))^T$.

Це спостереження формує перший вузол першого прихованого шару μ_1 так, що $c_{1i}(1) = x_i(1)$. У такий спосіб створюється n функцій належності й формується єдина синаптична вага $w_1(0)$, що задається випадково в інтервалі $-1 \leq w_1(0) \leq 1$. Далі для цієї функції належності μ_1 задається радіус сусідства r , обумовлений максимально можливим числом функцій належності в NFS h . Так, якщо функції належності по осях розподілені рівномірно, то

$$r = \frac{2}{h-1}. \quad (2)$$

Далі при надходженні другого спостереження $x(2)$ проводиться перевірка умови

$$\max_i |c_{li} - x_i(2)| \leq r. \quad (3)$$

Якщо ця умова виконується, то проводиться корекція центрів функцій належності вузла μ_1 згідно із правилом

$$c_{li}(2) = c_{li}(1) + \eta(2)(x_i(2) - c_{li}(1)), \quad (4)$$

де η – параметр кроку навчання.

Таким чином, при $\eta(2) = 0,5$

$$c_{li}(2) = \frac{c_{li}(1) + x_i(2)}{2}. \quad (5)$$

У тому випадку, якщо умова (3) не виконується, то формується другий вузол μ_2 функцій належності першого прихованого шару, центри яких

$$c_{2i}(2) = x_i(2). \quad (6)$$

Одночасно з вузлом μ_2 в NFS вводиться друга синаптична вага w_2 , що також задається випадковим чином. Нехай до моменту надходження на вхід NFS спостереження $x(k)$ сформоване p вузлів функцій належності $\mu_1, \mu_2, \dots, \mu_p, p < h$, із центрами $c_{li}(k-1), l=1, 2, \dots, p; i=1, 2, \dots, n$. Із надходженням $x(k)$ проводиться перевірка умови

$$\max_i |c_{li} - x_i(k)| \leq r \forall l = 1, 2, \dots, p. \quad (7)$$

Якщо ця умова виконується, то проводиться корекція центрів функцій належності, найближчих до відповідних компонентів $x(k)$ відповідно до правила

$$c_{li}(k) = c_{li}(k-1) + \eta(k)(x_i(k) - c_{li}(k-1)). \quad (8)$$

Нескладно помітити, що (8) є не що інше, як правило самонавчання Т. Кохонена “Переможець одержує все” [214] з тою лише різницею, що самонавчання карти Кохонена реалізується на гіперсфері

$$\|x(k)\|_2 = 1, \quad (9)$$

а правило (8) – на гіперкубі

$$\|x(k)\|_\infty = 1, \quad (10)$$

У тому випадку, якщо умова (7) не виконується, у системі формується $(p+1)$ вузол $(p+1 \leq h)$ із центрами функцій належності

$$c_{p+1,i}(k) = x_i(k). \quad (11)$$

Одночасно з вузлом μ_{p+1} формується синаптична вага w_{p+1} . Як видно, зазначена процедура є гібридним алгоритмом, що еволюціонує Н. Касабова [215] та карти, що самоорганізовується Т. Кохонена [214], при цьому процес еволюції архітектури-самонавчання функцій належності може відбуватися як безперервно, так і до досягнення числа функції належності. Налаштування параметрів центру й ширини функцій належності може проводитися по алгоритму із вчителем, заснованому на мінімізації цільової функції, яка, як правило, задається з використанням евклідової норми та для однієї пари навчальних даних $(x(k), y(k))$ що має вид

$$E(k) = \frac{1}{2} (y(k) - \hat{y}(k))^2 = \frac{1}{2} (y(k) - w^T \varphi(x(k)))^2. \quad (12)$$

При застосуванні методу найшвидшого спуску відповідні формули адаптації в загальному випадку для $(n \times 1)$ -мірного вектора вхідних сигналів приймає вид

$$c_{rj}(k+1) = c_{rj}(k) - \eta_c \frac{\partial E(k)}{\partial c_{rj}}, \quad (13)$$

$$\sigma_{rj}(k+1) = \sigma_{rj}(k) - \eta_\sigma \frac{\partial E(k)}{\partial \sigma_{rj}}, \quad (14)$$

де η_c – параметр кроку навчання для параметрів центру функції належності; η_σ – параметр кроку навчання для параметра ширини функції належності; $r = 1, 2, \dots, h$,
належності; $j = 1, 2, \dots, n$.

Для спрощення розрахунків похідних і прискорення обчислення значення функції належності формула адаптації параметра ширини може бути записана у вигляді

$$-0.5\sigma_{rj}^{-2}(k+1) = \sigma_{rj}(k) - \eta_\sigma \frac{\partial E(k)}{\partial (-0.5\sigma_{rj}^{-2})}. \quad (15)$$

При використанні в якості функцій належності традиційних гаусіанів (9) відповідні формули градієнта цільової функції (12) для однієї пари навчальних даних приймає вигляд

$$\frac{\partial E(k)}{\partial c_{rj}} = (w^T \varphi(x(k)) - (x(k))) \sum_{l=1}^h w_l \frac{\partial \varphi_l(x(k))}{\varphi c_{rj}}, \quad (16)$$

$$\frac{\partial E(k)}{\partial (-0.5\sigma_{rj}^{-2})} = (w^T \varphi(x(k)) - y(x(k))) \sum_{l=1}^h w_l \frac{\partial \varphi_l(x(k))}{\partial (-0.5\sigma_{rj}^{-2})}, \quad (17)$$

де

$$\varphi_l(x(k)) = \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{p=1}^h \prod_{i=1}^n \mu_{pi}(x_i(k))}. \quad (18)$$

Похідні $\frac{\partial \varphi_l(x)}{\partial c_{rj}}$ і $\frac{\partial \varphi_l(x)}{\partial (-0.5\sigma_{rj}^{-2})}$, визначені на основі (9) і (18), можна записати

як

$$\frac{\partial \varphi_l(x)}{\partial c_{rj}} = \frac{\delta_{lr} m(x) - t_l(x)}{(m(x))^2} \prod_{i=1, i \neq j}^n \mu_{ri}(x_i) \frac{\partial \mu_{ri}(x_j)}{\partial c_{rj}}, \quad (19)$$

$$\frac{\partial \varphi_l(x)}{\partial (-0.5\sigma_{rj}^{-2})} = \frac{\delta_{lr} m(x) - t_l(x)}{(m(x))^2} \prod_{i=1, i \neq j}^n \mu_{ri}(x_i) \frac{\partial \mu_{ri}(x_j)}{\partial (-0.5\sigma_{rj}^{-2})}, \quad (20)$$

$$t_l(x) = \prod_{i=1}^n \mu_{li}(x_i),$$

де δ_{lr} – дельта Кронекера,

$$m(x) = \sum_{p=1}^h \prod_{i=1}^n \mu_{pi}(x_i).$$

Похідні $\frac{\partial \mu_{rj}(x_j)}{\partial c_{rj}}$ та $\frac{\partial \mu_{rj}(x_j)}{\partial (-0.5\sigma_{rj}^{-2})}$, визначені на основі (9), можна записати як

$$\frac{\partial \mu_{rj}(x_j)}{\partial c_{rj}} = \frac{x_j - c_{rj}}{\sigma_{rj}^2} \exp\left(-\frac{(x_j - c_{rj})^2}{2\sigma_{rj}^2}\right), \quad (21)$$

$$\frac{\partial \mu_{rj}(x_j)}{\partial (-0.5\sigma_{rj}^{-2})} = (x_j - c_{rj})^2 \exp\left(-\frac{(x_j - c_{rj})^2}{2\sigma_{rj}^2}\right). \quad (22)$$

Етап 6. Визначення синаптичних ваг зв'язків штучної нейронної мережі.

Як ми вже відзначали, для настроювання синаптичних ваг нейро-фаззі системи можуть бути використані відомі алгоритми навчання-ідентифікації [217, 218]: експоненціально-зважений рекурентний метод найменших квадратів;

$$\begin{cases} w^h(k) = w^h(k-1) + \frac{P^h(k-1)(y(k) - w^{hT}(k-1)\varphi^h(x(k)))}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \varphi^h(x(k)) = \\ = w^h(k-1) + \frac{P^h(k-1)(y(k) - \hat{y}^h(x(k)))}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \varphi^h(x(k)), \\ P^h(k) = \frac{1}{\beta} \left(P^h(k-1) - \frac{P^h(k-1)\varphi^h(x(k))\varphi^{hT}(x(k))P^h(k-1)}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \right) = \\ = \left(\sum_{\tau=1}^k \varphi^h(x(\tau))\varphi^{hT}(x(\tau)) \right)^{-1}, 0 < \beta \leq 1, \end{cases} \quad (23)$$

де $y(t)$ – зовнішній навчальний сигнал, β – параметр забування застарілої інформації;

– оптимальний по швидкодії градієнтний однокроковий алгоритм Качмажа–Уїдроза–Хоффа

$$w^h(k) = w^h(k-1) + \frac{y(k) - w^{hT}(k-1)\varphi^h(x(k))}{\|\varphi^h(x(k))\|^2} \varphi^h(x(k)); \quad (24)$$

– алгоритм навчання, що володіє як стежачими так і згладжуючими властивостями [218]

$$\begin{cases} w^h(k) = w^h(k-1) + (\beta^h(k))^{-1} (y(k) - w^{hT}(k-1)\varphi^h(x(k))) \varphi^h(x(k)), \\ \beta^h(k) = \beta \beta^h(k-1) + \|\varphi^h(x(k))\|^2, 0 \leq \beta \leq 1 \end{cases} \quad (25)$$

і тому подібні процедури.

Процедура (25) пов'язана з алгоритмом (23) співвідношенням,

$$\beta^h(k) = \text{Tr} P^h(k), \quad (26)$$

а при $\beta = 0$ отримуємо форму алгоритму (23).

Процес настроювання синаптичних ваг може відбуватися одночасно із самонавчанням-еволюцією першого прихованого шару. Нехай до моменту надходження спостереження $x(k)$ сформовано p вузлів функцій належності $\mu_1, \mu_2, \dots, \mu_p$ й розрахований вектор синаптичних ваг $w^p(k-1)$. Нехай умова (7) не

виконується, що відразу ж веде до формування вузла μ_{p+1} та завданню довільного початкового значення синаптичної ваги w_{p+1} . При цьому вихідний сигнал NFS може бути представлений у вигляді

$$\hat{y}^{p+1}(x(k)) = (w^{p+1}(k-1))^T \varphi^{p+1}(x(k)) = w^{pT}(k-1) \varphi^p(x(k)) + w_{p+1} \varphi_{p+1}(x(k)), \quad (27)$$

а алгоритм (24) –

$$\begin{cases} w^{p+1}(k) = \left(\frac{w^p(k-1)}{w_{p+1}} \right) + (\beta^{p+1}(k))^{-1} (y(k) - \hat{y}^{p+1}(x(k))) \left(\frac{\varphi(x(k))}{\varphi_{p+1}(x(k))} \right), \\ \beta^{p+1}(k) = \beta \beta^p(k-1) + \|\varphi^p(x(k))\|^2 + \varphi_{p+1}^2(x(k)). \end{cases} \quad (28)$$

Як можна бачити, процес одночасних еволюції-самонавчання-контрольованого навчання не викликає ніяких обчислювальних проблем.

Етап 7. Перевірка відповідності архітектури та параметрів штучної нейронної мережі вимогам, що висуваються.

На даному етапі відбувається перевірка відповідності архітектури та параметрів штучної нейронної мережі вимогам. Зазначені вимоги до функціонування штучної нейронної мережі висуваються на етапі проектування систем підтримки прийняття рішень та залежать від типу та обсягів завдань, що виконуються системами підтримки прийняття рішень.

Етап 8. Прийнятті рішення на корегування архітектури та параметрів штучної нейронної мережі.

На підставі порівняльної оцінки вимог, що висуваються до функціонування ШНМ та реальної їх ефективності приймається рішення на корегування параметрів ШНМ, а саме (дія 6-8 на схемі рис. 2): виду та параметрів функції належності; синаптичних ваг між зв'язками.

У разі неможливості забезпечити необхідні вимоги, що висуваються до ШНМ, приймається рішення на зміну архітектури ШНМ та визначенні початкових параметрів ШНМ (дія 1-5 на схемі рис. 2).

3.3.2 Розробка методики навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень

Метою дослідження є розробка методики навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень, яка дозволяє виконувати обробку більшої кількості інформації, при однозначності рішень, що приймаються.

Для досягнення мети були поставлені такі завдання: провести постановку завдання навчання штучних нейронних мереж; провести розробку методики навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень; визначити переваги та недоліки запропонованої методики.

Мережа Кохонена [193] відноситься до самоорганізуючих мереж. Це означає, що вони не отримують бажаний вихідний сигнал при надходженні вхідного навчального вектору, а в результаті навчання мережу розділяє вхідні сигнали на класи, таким чином формуючи топологічні карти. Одним з найважливіших властивостей навченої мережі Кохонена є її здатність до узагальнення. Суть роботи мережі Кохонена полягає в тому, що відбувається кластеризація вхідних векторів в групи схожих між собою векторів. При цьому ваги мережі настроюються так, щоб вхідні образи, які належать одному кластеру, активізували один і той ж вихідний нейрон.

Вектор кожного з нейронів мережі Кохонена може замінити групу відповідних йому класифікуються векторів. Таким чином, це властивість дозволяє використовувати даний вид мережі в галузі стиснення даних.

Варто відзначити, що самоорганізуюча карта Т. Кохонена реалізує відображення вхідного простору розмірності n в вихідна простір розмірності m .

Самоорганізована карта має дуже просту архітектуру з прямою передачею інформації. Крім нульового (рецепторного) шару, вона містить єдиний шар нейронів, який дуже часто називають шаром Кохонена [193].

Саме завдяки такій організації, кожен нейрон мережі отримує всю інформацію про аналізованому образі і генерує на свій вихід відповідний відгук. Після цього в шарі Кохонена виникає режим конкуренції, в результаті якої

визначається єдиний нейрон-переможець з максимальним вихідним сигналом. Цей сигнал по латеральним зв'язків забезпечує збудження найближчих “сусідів” переможця і придушення реакції далеко віддалених вузлів. Самоорганізуються карти можуть мати різну топологію. Однак найбільш часто рецептори і нейрони розташовуються у вузлах одно або двовимірної решітки. Розглянемо докладніше архітектуру самоорганізуючої карти. На вхід мережі надходить n -мірний вхідний сигнал. Мережа містить єдиний шар з m нейронів, які утворюють на площині прямокутні решітки.

Нейрони характеризуються своїм місцем розташування в мережі. Кожен нейрон шару Кохонена пов'язаний з кожним входом нульового (вхідного) шару прямими зв'язками, а також з усіма іншими нейронами поперечними зв'язками.

На рис. 3 представлена 1D-карта Кохонена.

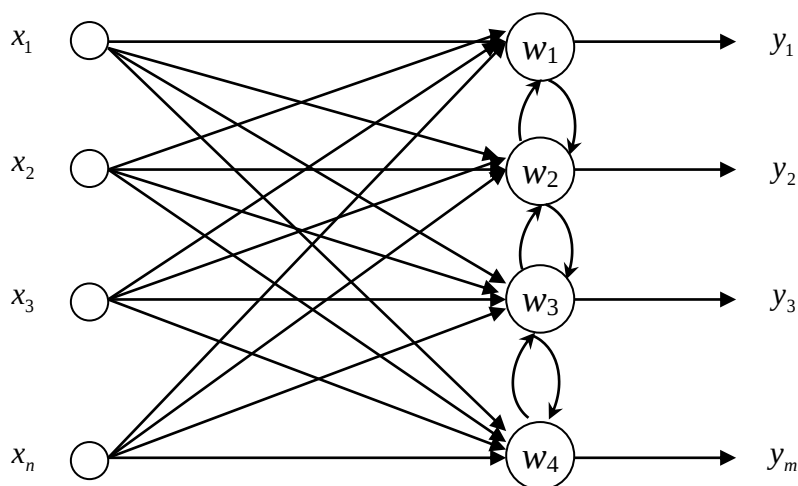


Рис. 3 – 1D-карта Кохонена

В процесі навчання сусідні нейрони впливають один на одного сильніше, ніж ті, які розташовані далі. Саме латеральні зв'язку в мережі забезпечують збудження одних нейронів і гальмування інших. Кожен нейрон з шару Кохонена формує зважену суму сигналів $f(x, w) = \sum_{i=1} w_i x_i$. При цьому, якщо синапси прискорюють, то $w_{ij} > 0$. Якщо ж синапси гальмуючі, то $w_{ij} < 0$.

На рис. 4 запропоновано методику навчання штучної нейронної мережі. Удосконалення зазначеної методики навчання полягає в додаванні дій 5-8 до відомих методів навчання штучних нейронних мереж. Тобто відбувається

додатково навчання штучних нейронних мереж, що не було враховано в роботах [192–208]:

архітектури штучних нейронних мереж в залежності від кількості вхідної інформації (кількості шарів, кількості скритих шарів, кількості зв'язків між нейронами у шарі та між шарами); кількості та параметрів функції належності; не вимагає зберігання раніше розрахованих даних.

Крок 1: Початковим етапом є ініціалізація початкових значень синаптичних ваг. *Крок 2.* Визначення ваг нейронів. *Крок 3.* Корекція ваг нейрону та визначення функції сусідства. *Крок 4.* Формування першого кластеру. *Крок 5.* Перевірка значення порогу. *Крок 6.* Перевірка спроможностей архітектури щодо обробки кількості інформації, що надходить на її вхід.

Крок 7. Еволюція архітектури системи.

Перед початком роботи алгоритму навчання мережі Кохонена вхідні вектори попередньо нормалізуються:

$$\tilde{x}_i = \frac{x_i}{\sqrt{\sum_i x_i^2}} = \frac{x_i}{\|x\|}, i = 1, 2, \dots, N \quad (29)$$

Сам алгоритм навчання мережі Кохонена може бути описаний як послідовність кроків:

Крок 1: проводиться ініціалізація початкових значень синаптичних ваг $w_{ij} = 0$.

Одним з часто застосовуваних способів ініціалізації є присвоювання синаптичним вагам значень, рівних випадково обраним векторам з безлічі спостережень. В цілому, існує три основних способу ініціалізації початкових ваг: ініціалізація випадковими значеннями, коли всім ваг даються малі випадкові величини; ініціалізація прикладами, коли в якості початкових значень задаються значення випадково вибраних прикладів з навчальної вибірки; лінійна ініціалізація.

В цьому випадку ваги ініціюються значеннями векторів, лінійно впорядкованих вздовж лінійного підпростору, що проходить між двома головними власними векторами вихідного набору даних.

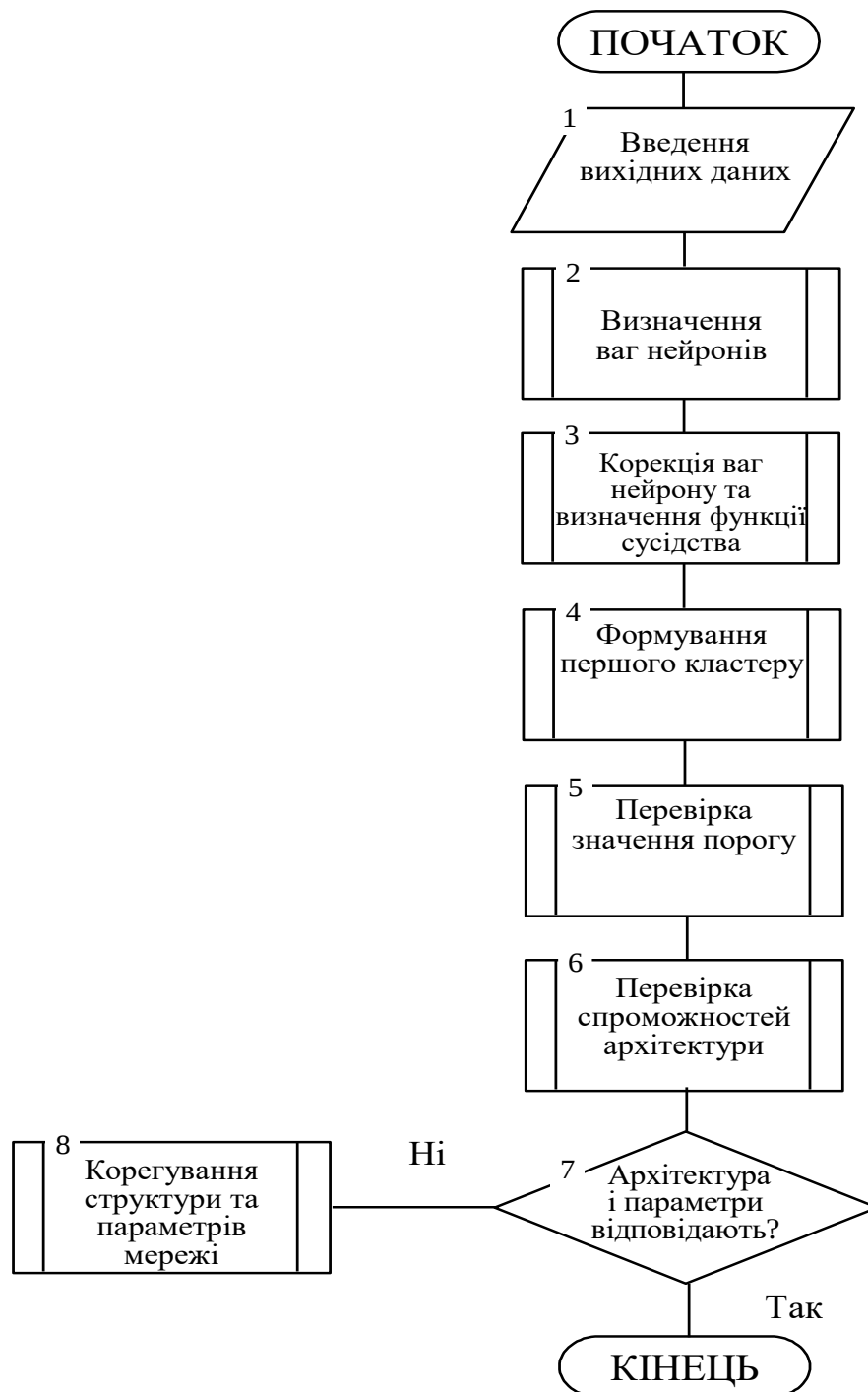


Рис. 4. – Методика навчання штучної нейронної мережі

Крок 2: на вхід системи подається нормалізований вектор сигналів $\tilde{x}$ та вибирається вектор ваг (нейрон), найближчий до $\tilde{x}$ тобто вектор, для якого Евклідова відстань до x буде найменшим:

$$\arg \min_j \|\tilde{x} - w_j\|, j=1,2,\dots,l. \quad (30)$$

Цей вираз також може бути записано в наступному вигляді:

$$c = \arg \max(\tilde{x}^T w_j), j=1,2,\dots,l. \quad (31)$$

Крок 3: проводиться корекція (настройка) вектору синаптичних ваг w_{ij} за правилом

$$w_{ij}(k+1) = w_{ij}(k) + \eta(k) f_{ij}(k)(\tilde{x}(k) - w_{ij}(k)), \quad (32)$$

де $w_{ij}(k+1)$ – нове значення ваги, $\eta(k)$ – змінюється з часом коефіцієнт посилення (на першій ітерації $\eta=1$ поступово зменшується до нуля, тобто, $\eta(k) \in (0,1]$), $f_{ij}(k)$ – монотонно спадаюча функція (функція сусідства) виду,

$$f_{ij}(k) = f(\|r_i - r_j\|, k), \quad (33)$$

де r_i – вектор, який визначає положення i -го нейрона в решітці, r_j – вектор, який визначає положення j -го нейрона в решітці. Очевидно, що для нейрона-переможця $f_c(\|r_i - r_j\|) = f_c(0) = 1$. Найчастіше в якості опції сусідства вибирають гаусіан, “Мексиканський капелюх”, косинусоїдну функцію, функції Єпанечнікова та т.п. При цьому вихідні сигнали мережі визначаються так:

$$y_j(k) = \begin{cases} 1, & \text{якщо } w_j^T(k) \tilde{x}(k) > w_p^T(k) \tilde{x}(k) \text{ для всіх } p \neq j, \\ 0, & \text{в іншому випадку} \end{cases} \quad (34)$$

Далі кроки 1 і 2 повторюються до тих пір, поки вихідні значення мережі не стабілізуються із заданою точністю. Сенс настройки за виразом (32) зводиться до того, що відбувається мінімізація різниці між вхідним вектором $\tilde{x}(k)$ і вектором синаптичних ваг нейрона-переможця.

Іншими словами можна сказати, що цей алгоритм в процесі настройки “підтягує” вектор синаптичних ваг нейрона-переможця до поточному вхідному образу $\tilde{x}(k)$. В даному випадку знаходиться вектор $\tilde{x}(k) - w_{ij}(k)$, який потім зменшується на величину $\eta(k)$, яка задає швидкість навчання. Таким чином, можна сказати, що навчання зводиться до обертання вектору ваг нейрона в напрямку вхідного вектору без істотного зміни його довжини.

В [209] запропоновано оригінальний online еволюціонуючий метод нечіткої кластеризації (evolving fuzzy clustering method, EFCM), в основі якого лежить імовірнісний підхід до вирішення завдання, при цьому основним

параметром, який в підсумку визначає кінцевий результат, є радіус сформованих кластерів, обраний з емпіричних міркувань і визначає в підсумку число можливих класів. Незважаючи на ефективність імовірнісних алгоритмів нечіткої кластеризації (FCM), їх “слабким” місцем є “жорстке” обмеження

$$\sum_{j=1}^m u_j(k) = 1 \forall k = 1, 2, \dots, N, \quad (35)$$

Цього недоліку позбавлений, так званий, можливістний (possibilistic) підхід (PCM) до нечіткої кластеризації [208], пов’язаний з оптимізацією цільової функції що веде до того, що спостереження, так само належить всім класам, тобто рівновіддалене від всіх центроїдів, може має ті ж рівні приналежності, що і спостереження, також рівновіддалене від центроїд, але не належить жодному з кластерів

$$E(u_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|x(k) - c_j\|^2 + \sum_{j=1}^m \mu_j \sum_{k=1}^N (1 - u_j(k))^\beta, \quad (36)$$

де $c_j = (c_{j1}, c_{j2}, \dots, c_{jn})^T$ – центр ваги j -го кластера, який вираховується в процесі обробки даних, $\beta > 1$ – параметр фазифікації (“фаззифікатор”), що визначає “розмитість” меж між кластерами і зазвичай покладаються $\beta = 2$, $\mu_j > 1$ – скалярний параметр, що визначає відстань, на якому приналежність приймає значення 0,5, тобто якщо:

$$\|x(k) - c_j\|^2 = \mu_j, \quad (37)$$

то $\mu_j(k) = 0,5$.

Використання можливісного підходу веде до еволюційного методу нечіткої кластеризації (EPCM), який зручно записати у формі послідовності кроків:

Крок 4: при надходженні спостереження $x(1)$ формується перший кластер з центроїдом c_1 .

Крок 5: при надходженні спостереження $x(1)$ перевіряється умова

$$\|x(2) - c_1\| \leq \Delta, \quad (38)$$

де Δ – деякий апіорі задається поріг.

Якщо ця умова виконується, то спостереження $x(1)$ не формує новий центр ваги, тобто потрібно було, що воно належить першому кластеру з рівнем належності

$$u_1(2) = \left(1 + \left(\frac{\|x(2) - c_1\|^2}{\mu_1} \right)^{\frac{1}{\beta-1}} \right)^{-1} \quad (39)$$

Якщо ж виконується умова

$$\Delta < \|x(2) - c_1\| \leq 2\Delta, \quad (40)$$

то відбувається корекція центроїда c_1 згідно WTA (winner-takes-all, WTA)-правила самонавчання Кохонена [209]:

$$c_1(2) = c_1(1) + \eta(2)(x(2) - c_1(1)), \quad (41)$$

$\eta(2)$ – параметр кроку налаштування.

При цьому центроїд c_1 підтягується до вектору спостережень $x(1)$; якщо ж для $x(1)$ виконується нерівність

$$2\Delta < \|x(2) - c_1\|, \quad (42)$$

то формується другий кластер з центроїдом

$$c_2 = x(2). \quad (43)$$

При цьому повинні бути розраховані рівні належності $u_2(1)$ та $u_1(2)$ по формулам, що наведені нижче.

Крок 6. Перевірка спроможностей архітектури щодо обробки кількості інформації, що надходить на її вхід.

Отже якщо є N спостережень та m кластерів з центроїдами c_j , розрахунки всіх належностей та скореговані координати центроїдів оцінюються згідно відношення:

$$\left\{ \begin{aligned} u_j(k) &= \left(1 + \left(\frac{\|x(k) - c\|^2}{\mu_j} \right)^{\frac{1}{\beta-1}} \right)^{-1}, \\ c_j &= \frac{\sum_{k=1}^N u_j^\beta(k) x(k)}{\sum_{k=1}^N u_j^\beta(k)}, \\ \mu_j(k) &= \frac{\sum_{k=1}^N u_j^\beta(k) \|x(k) - c_j\|^2}{\sum_{k=1}^N u_j^\beta(k)}, \end{aligned} \right. \quad (44)$$

отримаємо в результаті мінімізації (36) по всім параметрам, що оцінюються.

Система рівнянь (44) представляє собою по суті пакетний алгоритм обробки інформації так, що при надходженні спостереження $x(N+1)$ усі розрахунки повинні бути проведені заново. Зрозуміло, що при достатньо високій частоті надходження даних підхід може виявитися неефективним.

З цією метою потрібно провести розробку рекурентних процедур, що не потребують зберігання відпрацьованих раніше даних. Реалізувати зазначені рекурентні процедури можна на основі двошарової адаптивної нейро-фаззі мережі з наступною архітектурою.

Перший скритий шар мережі утворений звичайними нейронами Кохонена N_j^K , пов'язані між собою латеральними зв'язками, по яким реалізується процес конкуренції. Вихідний шар мережі, утворений вузлами N_j^u , призначений для розрахунку рівнів належності кожного спостереження $x(k)$ кожному j -му кластеру, $j = 1, 2, 3, \dots, m$. Для налаштування центроїдів кластерів використовується рекурентна процедура самонавчання, яка має вид [201]:

$$\left\{ \begin{array}{l} c_j(k+1) = c_j(k) + \frac{u_j^\beta(k)}{k+1} (x(k+1) - c_j(k)), \\ u_j(k+1) = \frac{1}{1 + \left(\frac{\|x(k+1) - c_j(k+1)\|^2}{\mu_j(k)} \right)^{\frac{1}{1-\beta}}}, \\ \mu_j(k+1) = \frac{\sum_{p=1}^{k+1} u_j^\beta(p) \|x(p) - c_j(k+1)\|^2}{\sum_{p=1}^{k+1} u_j^\beta(p)}. \end{array} \right. \quad (45)$$

Неважко помітити що перший вираз (45) і є WTM-правило самонавчання зі звужуючою функцією сусідства $(k+1)^{-1} u_j^\beta(k)$.

Крок 7. Еволюція архітектури системи.

Процес еволюції системи подібно попередньому починається з єдиного нейрона Кохонена, який уточнює координати першого центроїду c_1 . Наступний нейрон додається до мережі при виконанні умови (42), яке в даному випадку приймає форму

$$2\Delta < \|x(k) - c_1(k-1)\|. \quad (46)$$

В цей момент формується нейрон з центроїдом $c_2(k) = x(k)$. Тут необхідно помітити, що оскільки в нейронних мережах Кохонена дані попередньо нормуються на гіпершар так, що

$$\|x(k)\|^2 = \|c_j(k)\|^2 = 1, \quad (47)$$

нерівність (42), яке визначає необхідність введення до мережі нових нейронів, отримує форму

$$-1 \leq 1 - 2\Delta^2 < c_j^T(k-1)x(k) \leq 1, \forall j = 1, 2, \dots, m, \quad (48)$$

або

$$-1 \leq 1 - 2\Delta^2 < \cos(c_j(k-1)x(k)) \leq 1. \quad (49)$$

Таким чином, нарощування архітектури відбувається в результаті постійного контролю нерівностей (58) або (49) що відбувається в випадку, якщо ці нерівності порушуються.

Завдяки використанню саме можливісного підходу, доцільна реалізація ще однією “гілки еволюції”, а саме, якщо в якійсь момент часу виявиться, що належності спостереження $x(k)$ не перевищують деякого порогового значення

$$u_j(k) < \varepsilon \forall j = 1, 2, \dots, m, \quad (50)$$

тобто спостереження $x(k)$ достатньо далеко розташовано від всіх вже сформованих центроїдів, це також може служити сигналом для формування нового кластера

$$c_{m+1}(k) = x(k). \quad (51)$$

Для оцінки якості нечіткої кластеризації може бути використаний популярний індекс Ксі–Бені [201] в розширеній формі [205]. Для фіксованої вибірки, що має N спостережень цей індекс має вид

$$EXB(N) = \frac{\left(\sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|x(k) - c_j(N)\|^2 / N \right)}{\min_{j \neq l} \|c_j(N) - c_l(N)\|^2} = \frac{NEXB(N)}{DEXB(N)}. \quad (52)$$

Для online обробки цей індекс подібно центроїдам кластерів також може розраховуватися рекурентно:

$$\begin{aligned} EXB(k+1) &= \frac{NEXB(k+1)}{DEXB(k+1)} = \\ &= \frac{NEXB(k) + \frac{1}{k+1} \left(\sum_{j=1}^m u_j^\beta(k+1) \|x(k+1) - c_j(k+1)\|^2 - NEXB(k) \right)}{\min_{j \neq l} \|c_j(k+1) - c_l(k+1)\|^2}. \end{aligned} \quad (53)$$

Додавання виразу (51) в процедуру навчання (47) дозволить організувати додатковий контроль за кількістю кластерів, що формуються системою. Так, введення третього порогу δ та перевіряючи на кожному кроці умову

$$EXB(k+1) > \delta, \quad (54)$$

можна зупинити процес нарощування нейронів в випадку порушення нерівності (54).

Запропоновано методику навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень. Проведено моделювання роботи запропонованої методики в програмному середовищі MathCad 14.

Для проведення експерименту використовувався двовимірний штучно згенерований набір даних. Він містив 15 кластерів з різним рівнем перекриття. Вибірка даних містила 5000 спостережень. Дані подавалися на обробку в послідовному режимі. Для порівняння якості кластеризації використовувалися FCM і система, заснована на еволюційному методі нечіткої кластеризації (Evolving Fuzzy Clustering Method, EFCM) з різними значеннями параметра порога. Як критерій оцінки якості кластеризації був використаний індекс Ксі-Бені. У таблиці 1 представлені порівняльні результати кластеризації.

Таблиця 1 – Порівняльні результати кластеризації

Система	Кількість кластерів	Параметри алгоритму	ХВ (Індекс Ксі-Бені)	Час кластеризації, с
FCM (Fuzzy C-Means)	15	-	0.0903	2.69
EFCM	9	Dthr = 0.24	0.1136	0.14
EFCM	12	Dthr = 0.19	0.1548	0.19
Запропонована система (пакетний режим)	12	delta = 0.1	0.0978	0.37
Запропонована система (online режим)	12	delta=0.1	0.1127	0.25

Для проведення наступного експерименту була використана вибірка даних, що описує характеристики об'єкту моніторингу. Вибірка даних містила 210 спостережень. Кожне спостереження описувалося сім'ю параметрами: кількість особового складу; загальна чисельність особового складу; кількість організаційно-штатних структур; кількість зразків озброєння та військової техніки; кількість засобів зв'язку; кількість типів зразків озброєння та військової

техніки; тип засобів зв'язку. Перед кластеризацією ознаки спостережень були нормалізовані на інтервалі $[0,1]$.

Таблиця 2 – Порівняльні результати кластеризації

Система	Кількість кластерів	Параметри алгоритму	ХВ (Індекс Ксі-Бені)	Час,с
FCM (Fuzzy C-Means)	3	-	0.1963	0.81
EFCM	4	Dthr = 0.6	0.2330	0.04
Запропонована система (пакетний режим)	3	delta = 0.4	0.2078	0.45
Запропонована система (online режим)	3	delta = 0.4	0.2200	0.30

Параметри аналізованих систем, а також кількість виявлених кластерів, також наведені в табл. 2. Для оцінки якості роботи систем використовувався індекс Ксі-Бені (ХВ). Варто відзначити, що запропонована система показала кращий за критерієм РС (partition coefficient, РС – коефіцієнт розбиття) результат в порівнянні з EFCM і кращий за часом роботи результат в порівнянні з FCM. Як запропоновані системи, так і FCM виявили три нечітких кластера.

Основними перевагами запропонованої методики оцінки є: не накопичує помилки навчання в ході навчання штучних нейронних мереж за рахунок корегування параметрів та архітектури штучної нейронної мережі; однозначність отриманої результатів; широка сфера використання (системи підтримки та прийняття рішень); простота математичних розрахунків; можливість адаптації системи в ході роботи; можливість синтезу оптимальної структури системи підтримки та прийняття рішення.

До недоліків запропонованого методики слід віднести:

– втрата інформативності при оцінюванні (прогнозування) за рахунок побудови функції належності. Зазначена втрата інформативності може бути зменшена за рахунок вибору типу функції належності та її параметрів при

практичній реалізація запропонованої методики в системах підтримки та прийняття рішень. Вибір типу функції належності залежить від обчислювальних ресурсів конкретного електронно-обчислювального засобу.

- менша точність оцінювання по окремо взятому параметру оцінки стану;
- втрата точності результатів за час перестройки архітектури штучної нейронної мережі.

Зазначене дослідження є подальшим розвитком досліджень проведеними авторами, що спрямовані на розробку теоретичних основ підвищення ефективності систем штучного інтелекту, що опубліковані вже раніше [1, 2, 29–32].

Висновки

1. Розроблено алгоритм навчання методу навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень.

Підвищення оперативності обробки інформації та зменшення похибки оцінювання досягається за рахунок: навчання не тільки синаптичних ваг штучної нейронної мережі, але й виду та параметрів функції належності; навчання архітектури штучних нейронних мереж; можливості комбінації елементів штучної нейронної мережі; можливості навчання окремих елементів штучної нейронної мережі; обчислення даних за одну епоху без необхідності зберігання попередніх обчислень. Зазначене зменшує час на обробку інформації за рахунок відсутності необхідності звернення до бази даних;

- відсутності накопичення помилки навчання штучних нейронних мереж в результаті обробки інформації, що надходить на вхід штучних нейронних мереж.

Дослідження розробленого алгоритму навчання показало, що зазначений алгоритм навчання забезпечує в середньому на 10–18 % більшу високу ефективність навчання штучних нейронних мереж та не накопичує помилок в ході навчання.

2. Наукова новизна зазначеної методики навчання штучних нейронних мереж для інтелектуальних систем підтримки прийняття рішень полягає в наступному.

- проводить навчання не тільки синаптичних ваг штучної нейронної мережі, але й виду та параметрів функції належності;
- в разі неможливості забезпечити задану якість функціонування штучних нейронних мереж за рахунок навчання параметрів відбувається навчання архітектури штучних нейронних мереж;
- вибір архітектури, виду та параметрів функції належності відбувається з врахуванням обчислювальних ресурсів засобу та з врахуванням типу та кількості інформації, що надходить на вхід штучної нейронної мережі;
- не відбувається накопичення помилки навчання штучних нейронних мереж в результаті обробки інформації, що надходить на вхід штучних нейронних мереж.

Практична цінність запропонованої методики полягає в тому, що на її основі було проведено розробку практичних рекомендації щодо підвищення ефективності навчання штучних нейронних мереж.

Дослідження розробленої методики показало, що зазначена методика навчання забезпечує в середньому на 16–23 % більшу високу ефективність навчання штучних нейронних мереж та не накопичує помилок в ході навчання.