

```
## Warning in grepl(db, input): input string 43 is invalid in this locale
## Warning in grepl(db, input): input string 44 is invalid in this locale
## Warning in grepl(db, input): input string 45 is invalid in this locale
## Warning in grepl(db, input): input string 48 is invalid in this locale
## Warning in grepl(db, input): input string 53 is invalid in this locale
## Warning in grep("^\\\\bibliography.", input, value = TRUE): input
string 43 is invalid in this locale
## Warning in grep("^\\\\bibliography.", input, value = TRUE): input
string 44 is invalid in this locale
## Warning in grep("^\\\\bibliography.", input, value = TRUE): input
string 45 is invalid in this locale
## Warning in grep("^\\\\bibliography.", input, value = TRUE): input
string 48 is invalid in this locale
## Warning in grep("^\\\\bibliography.", input, value = TRUE): input
string 53 is invalid in this locale
```

## РОЗДІЛ 1

### ОГЛЯД СТАНУ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

#### 1.1. Основні події в нейронних мережах досліджень

Дисципліна нейронних моделей моделює людський мозок. Середній людський мозок складається з близько 1011 нейронів різних типів, при цьому кожен нейрон поєднується до десятків тисяч синапсів. Таким чином, нейромережеві моделі також називаються конекціоністичними моделями. Інформація обробляється, в основному, в зовнігньому шарі головного мозку - його корі. Когнітивні функції, серед яких мова, абстрактне мислення, також навчання і пам'ять, є найбільш складними для визначення з точки зору нейро механізмів операціями мозку.

У 1940-х роках, Маккалох та Піттс [27] виявили, що нейрон може бути змодельовано як простий пороговий пристрій для виконання логічної функції. У 1949 році Хебб [14] запропонував правило Хебба, що описує вплив процесу навчання на синаптичний зв'язок між двома нейронами. У 1952 році на підставі фізичних властивостей клітинних мембран і трансмембранних білків Ходжкін і Хакслі [15] описали такі явища, як нейроні стрільби та поширення потенціалу дії, через набір еволюційних рівнянь. Ця робота принесла Ходжкіну та Хакслі Нобелівську премію в 1963 р. В кінці 1950-х і початку 1960-х, Розенблатт [31] запропонован модель перцептрона, а Відроу і Хофф [38] запропонували модель Adaline (адаптивний лінійний елемент), яка використовувала метод найменших квадратів (МНК) у якості алгоритму навчання.

У 1969 році Мінським і Папертом [28] було математично доведено неможливість використання перцептрона для складної логічної функції. Це призвело до істотного зменшення інтересу до нейронних мереж. В той же час,

модель Adaline, а також її багатошарова версія під назвою Madaline була успішно використана при розв'язанні багатьох завдань. Тим не менш, область застосування таких моделей все ще залишалась дуже вузькою, основною причиною цього було використання виключно лінійної функції активації.

У 1970-ті роки, Гроссберг [12, 13], фон дер Мальцбург [37] і Фукусіма [9], натхнені результатами досліджень зорової кори, опублікували новаторські роботи на предмет навчання і конкурентної самоорганізації. Фукусіма запропонував свою модель когнітрону і неокогнітрону, в рамках парадигми конкурентного навчання. Неокогнітрон являє собою ієрархічну багатошарову нейронну мережу, спеціально призначену для розпізнавання візуальних образів. Також в цей період були запропоновані кілька лінійних моделей асоціативної пам'яті. У 1982 році Кохонен запропонував самоорганізації карту (SOM) [23]. СДЛ адаптивно перетворює входять шаблони сигналів довільних розмірів на одно- або двовимірних дискретних відображень в топологічно впорядковано. Гроссберг і Карпентер [4, 13] запропонована адаптивна теорія резонансу (АРТ) моделі в середині 1980-х років. Модель ART, також заснований на конкурентному навчання, є поворотною і самоорганізації.

Модель Хопфілда введена в 1982 році [17] ввів в сучасну епоху нейронних мереж наукових досліджень. Модель працює на системному рівні, а не на одному рівні нейронів. Це повторюється нейронна мережа працює з Хебба правилом. Ця мережа може бути використана в якості асоціативної пам'яті для зберігання інформації і для вирішення завдань оптимізації. Машина Больцмана [1] був введений в 1985 році як розширення мережі Хопфілда шляхом включення стохастичних нейронів. навчання Больцмана заснований на методі, званому модельований отжиг [20]. У 1987 році Коско запропонував адаптивну двобічної асоціативної пам'яті (ВАМ) [24]. Хеммінга мережу, запропонованого Липпманом в середині 1980-х років [25] заснований на конкурентному навчання, і є найбільш простий асоціативної пам'яті. У 1988 році, Чуа і Ян [5] розширили модель Хопфілда, запропонувавши клітинну модель нейронної мережі. Мережа робота є динамічна мережева модель і особливо

підходить для двовимірної обробки сигналів і реалізації HBIC.

Найбільш помітним орієнтиром в нейронній мережі досліджень є алгоритм зворотного поширення (BP) навчання, пропонувався для моделі багатопшаровий персептрон (MLP) в 1986 році Rumelhart, Хінтон, і Вільямс [33]. Надалі, алгоритм BP був виявлений вже був винайдений в 1974 році Werbos [39]. У 1988 році, Broomhead і Lowe запропонував мережеву модель радіальної базисної функції (RBF) [3]. Обидва MLP і RBF мережі є універсальними апроксиматорами.

У 1982 році Оя запропонував мережу аналізу головних компонент (PCA) для класичного статистичного аналізу [29]. У 1994 році загальний запропонував незалежний аналіз компонент (ICA) [6]. ICA є узагальненням PCA, і це, як правило, використовується для виділення ознак і сліпого поділу джерел (BSS). З тих пір багато алгоритми нейронних мереж для класичних статистичних методів, таких як аналіз Фішера лінійний дискримінантний (LDA), канонічного кореляційного аналізу (CCA) і факторного аналізу, були запропоновані.

У 1985 році Перл представила модель байєсівської мережі [30]. Байєсівської мережі є найбільш відомим графічна модель в AI. Він володіє характеристикою буття як статистичного та подання знань формалізм. Вона закладає фундамент для виведення сучасного II.

Ще однією важливою подією в області машинного навчання і нейронних мережевих спільнот є підтримка векторної машини (SVM), запропонований Вапніка і ін. на початку 1990-х років [36]. SVM ґрунтується на статистичній теорії навчання і особливо корисно для класифікації з малими розмірами зразка. SVM використовується для класифікації, регресії і кластеризації. Завдяки успішному застосуванню в SVM, метод ядра викликав широкий інтерес.

На додаток до нейронних мереж, нечіткої логіки і еволюційні обчислення двох інших великих м'яких обчислень парадигми. Soft Computing є обчислювальним структура, яка може терпіти неточність і невизначеність, а не в залежності від точних математичних обчислень. Нечіткої логіки [40] може

включати людські знання в систему за допомогою нечітких правил.

## **1.2. Навчання та самонавчання штучних нейронних мереж**

Здатність навчатися – основна властивість біологічного мозку, а оскільки штучна нейронна мережа в деякому сенсі моделює мозок, поняття «навчання» посідає щонайперше місце в теорії штучних нейронних мереж. Математичні проблеми, що пов'язані з навчанням, вивчають у напрямі загальної теорії штучних нейронних мереж, який дістав назву «нейроматематика» [51, 52].

Із точки зору нейроматематики, навчання тлумачать як завдання адаптувати параметри, а можливо, й архітектуру мережі, щоби, оптимізуючи прийнятий критерій якості, розв'язати поставлену задачу. Таке визначення є узвичаєним та неявно припускає, що нейроматематика ґрунтується на методах оптимізації та ідентифікації

Звичайно припускають, що навчання має перманентний характер та з часом мережа покращує свої характеристики, постійно «наближаючись» до оптимального розв'язку поставленої задачі.

Тип та характер навчання обумовлені, насамперед, обсягом попередньої та поточної інформації про довкілля, в яке «занурили» мережу, а також критерієм якості (цільовою функцією), що характеризує рівень відповідності нейронної мережі до розв'язуваної нею задачі. Інформацію про довкілля здебільшого задають у вигляді навчальної вибірки образів або зразків, що їх оброблюючи мережа дістає відомості, необхідні для отримання шуканого розв'язку. Саме характер та обсяг цієї інформації визначають як тип навчання, так і конкретний метод.

З погляду математики, навчання нейронних мереж – це багатопараметрична задача нелінійного оптимування. Більшість методів навчання можна розділити на два класи: навчання з учителем (із заохоченням) та навчання без учителя (без заохочення, або самонавчання). Методи навчання з учите-

лем застосовують у випадках, коли відома бажана реакція системи в кожному мить часу, себто відомий навчальний сигнал, який впливає на налаштування параметрів системи, що навчається. Рівень «навченості» системи формально визначають за значенням цільової функції, тобто за тим станом, якого має в результаті набути коректно навчена система.

У цій схемі «учителю» відома інформація про довкілля, представлена послідовністю або пакетом ухідних векторів  $x$ , а також «правильна реакція» на ці сигнали, представлена навчальним сигналом  $y$ ?. В процесі навчання реакція нейронної мережі  $u$  розбігається з «правильною» реакцією вчителя, через що постає похибка  $e = y - u$ . Мета навчання – так налаштувати параметри ШНМ, щоби деяка скалярна функція похибки  $E(e)$  (критерій якості) досягла свого найменшого значення. Навченою вважають мережу, яка в деякому, переважно статистичному сенсі повторює реакцію вчителя. Позаяк інформація про довкілля здебільшого має нестационарний характер, навчання відбувається безперервно, для чого використовують ті чи інші рекурентні процедури.

Альтернатива навчанню «з учителем» – навчання за умов, коли правильна реакція на сигнали довкілля невідома. Цю парадигму називають навчанням «без учителя», або самонавчанням. Штучні нейронні мережі, що самонавчаються, здебільшого, призначені аналізувати внутрішню латентну структуру вхідної інформації та розв'язують задачі автоматичного класифікування, кластерування, факторного аналізу, компресування даних тощо. У теорії ШНМ самонавчання звичайно розглядають як конкурування або самоорганізування нейронів мережі, що топологічно взаємозв'язані між собою.

Щоби покращити якість навчання та прискорити збіжність, ітераційне навчання можна повторювати циклічно на так званому «вікні» – наборі послідовних значень навчального сигналу (у дискретному випадку) або проміжкові часу (у неперервному випадку). Граничним варіантом процедур з повторюванням є пакетне та послідовне (у реальному часі) модифікування. Під пакетним режимом розуміють випадок, коли всю вибірку даних задано

попередньо, а навчання відбувається «епохами». У режимі реального часу повторювання відсутнє, хоча якщо зорганізувати навчання в «прискореному» машинному часі, навчання може повторюватися. Це не суперечить концепції реального часу, бо саме поняття «реальний час» залежить од певної ЕОМ. До прикладу, якщо дані надходять щохвилини, то швидкодійна ЕОМ за 1 хвилину може обробити увесь масив даних кілька разів.

### **1.3. Гібридні системи обчислювального інтелекту**

В обчислювальному інтелекті вельми поширеним є підхід створювати системи оброблення даних на основі кількох наукових напрямів. Як засвідчують теоретичні та практичні результати, таким гібридним системам властивий синергетичний ефект, тобто вони виявляють такі властивості, яких не мають системи, що їх утворюють [85]. Одним з яскравих прикладів гібридних систем обчислювального інтелекту є нейро-фазі системи, які поєднують у собі нейронні мережі другого покоління та нечіткі системи [9, 85]. Нейронна мережа може навчатися на вхідних та вихідних даних для визначення поведінки системи, але отримані знання будуть сховані в її синапсових вагах і їх не можна буде витлумачити. Однак, якщо виразити ваги нейронної мережі за допомогою нечітких правил, з'являється можливість подолати неінтерпретовність результатів роботи нейронної мережі. У такий спосіб нейрофазі системи дають змогу створювати системи оброблення інформації та отримують більш широке застосування. Розвиваючи гібридний підхід, запропоновано й просунутіші поєднання наукових напрямків, наприклад, теорії штучних нейронних мереж та індуктивного моделювання даних [86]. Ефективність кластерування даних залежить у великій мірі від якості обраної математичної моделі розв'язуваної або досліджуваної задачі. Як уже мовилося вище, однією з проблем кластерування даних є змінна кількість кластерів оброблюваних даних. Відповідно постає складна задача обрати належну математичну модель. Індуктивне моделювання має тут ефективний розв'язок: налаштовувати не лише

параметри системи оброблення даних, але також і її структуру. Проекція такого підходу на штучні нейронні мережі веде до ідеї змінювати кількість нейронів в шарах мережі, що обробляє дані. Дієвість роботи побудованих за цим принципом систем [87, 88] засвідчує плідність гібридного підходу.

#### **1.4. Гібридні нейро-фаззм системи та питання структурної адаптації**

В наш час для розв'язання задач, які пов'язані з інтелектуальною обробкою даних в умовах апіорної та поточної невизначеності, дослідники часто не обмежуються використанням якогось одного підходу (нейронні мережі, нечітка логіка, генетичні алгоритми тощо), а задля синергії зв'язують групу методів в одну гібридну систему. Такі системи відповідають усім вимогам до інтелектуальних систем і отримали назву гібридні системи обчислювального інтелекту. Нейро-фаззі системи та м'які обчислювання є напрямками дисципліни обчислювального інтелекту, які й займаються проблемами таких систем. Серед основних характеристик систем, що розробляються в рамках нейро-фаззі систем та м'яких обчислень, можна виділити наступні [27]:

- обчислювальні моделі, що засновані на біологічних прототипах,
- паралельна обробка даних у послідовному режимі,
- в основі системи лежать експертні знання,
- стійкість системи до зашумління,
- стійкість системи до виходу із строю підсистем.

Варто зазначити, що одною із головних умов до такого типу систем, є їх орієнтованість на розв'язання практичних завдань, що означає здатність оброблювати великі масиви даних великої розмірності, які можуть мати пропущені та зашумлені значення. Однак навчання таких систем зводиться до налаштування синаптичних коефіцієнтів та/або адаптації бази нечітких правил. Тобто архітектура такої системи не зазнає жодних змін, що може в деяких випадках призвести до погіршення точності результатів. В зв'язку з



цим видається очевидно корисним зсінтезувати таку гібридну нейро-фаззі архітектуру та такі алгоритми її навчання, що здатні змінювати не тільки параметри систему, а й її архітектуру.

Як вже зазначалось, основною метою навчання є отримання нейронної мережі, яка здатна у найкращий спосіб відтворювати попередньо невідоме відображення  $R^n \rightarrow R^m$ . В якості такого відображення може виступати залежність вихідних параметрів процесу від вхідних, прогнозування від передісторії, класа об'єкту від набору його властивостей, управляючої дії від поточного стану об'єкта тощо. Коректне налаштування не тільки синаптичних коефіцієнтів, а й архітектури нейронної мережі, зокрема налаштування кількості шарів та кількості нейронів у кожному шарі, дозволяю суттєво покращити показники такої мережі. Серед підходів до налаштування архітектури нейронної мережі виділяють:

- деструктивний підхід: за основу береться заздалегіть надлишково складна модель, до неї застосовуються різні процедури, що видаляють із вихідної мережі елементи, які оказують негативний або незначний позитивний ефект на кінцевий результат,
- конструктивний підхід: за основу береться максимально проста модель (складається із одного або декількох нейронів), до неї застосовуються процедури, що додають вихідній мережі нові елементи до певного моменту, в залежності від використовуваного алгоритму. Як варіант, конструктивний алгоритм може стартувати з цілком нульової архітектури та самостійно генерувати шари мережі в процесі своєї роботи.

**1.4.1. Деструктивний підхід до налаштування архітектури нейронної мережі.** Основна ідея деструктивних алгоритмів полягає у видаленні параметрів, що мабуть найменший вплив на вихідний сигнал мережі. У ряді публікацій **30-32** був сформульований та підтвержене припущення, що використання деструктивних алгоритмів нерідко призводить до покращення

узагальнюючих властивостей мережі, допомагає нейтралізувати появу так званого ефекту перенавчання, а, крім того, після закінчення роботи такого алгоритму, архітектура мережі набуває меншого так більш простого вигляду, що очевидно позитивно відбивається на її обчислювальній складності.

У процесі функціонування деструктивних алгоритмів із мережі можуть бути цілком видалені як деякі вхідні параметри або вузли у прихованих шарах, так і лише деякі синаптичні зв'язки між нейронами, що мають лише один параметр ідефіс ваговий коефіцієнт. Вочевидь, в основі деструктивного піходу до структурної адаптації нейронної мережі повина бути закладено обчислення деякої **міри значущості**, яка буде характеризувати ступінь впливу кожного конкретного параметру на вихідний сигнал.

В одній із перших публікацій, що розглядає цю проблему, **30** автори запропонували деструктивний алгоритм структурної оптимізації під назвою Optimal Brain Damage(OBD), яких складається із наступних етапів:

1. вибрати архітектури нейронної мережі,
2. використовуючи один із методів мінімізації цільової функції якості, провести навчання мережі,
3. для кожного елемента мережі обчислити міру значущості:

$$s_q = h_q u_q^2 / 2, \quad (1.1)$$

де  $u_q$  – вихідний сигнал  $q$ -го елемента мережі,

$h_q$  розраховується по формулі:

$$h_q = \sum_{(i,j) \in V_q} \frac{\partial^2 E}{\partial w_{ij}^2}, \quad (1.2)$$

де  $V_q$  – множина пар коефіцієнтів  $i$  та  $j$  для  $q$ -го елемента мережі,

$E$  – помилка на виході нейронної мережі,

$w_{ij}$  – синаптичний ваговий коефіцієнт в  $j$ -м шарі,

4. видалити із мережі деяку кількість елементів, для яких міра значущості  $s_q$  найменша. В цьому контексті під видаленням елемента має-

ться на увазі зміна вихідного значення елемента на 0 та замороження його в такому стані,

5. повернутися на крок 2 та повторити процедуру.

Вочевидь, при використанні такого методу значно збільшується обчислювальна складність алгоритма навчання, і через необхідність розрахувати міру значущості для кожного нейрону, і через додаткові ітерації перенавчання, які необхідно виконати після видалення кожного нейрона із мережі. Також суттєвим недоліком є те, що разом з видаленням нейрона ми видаляємо одразу декілька синаптичних зв'язків, хоча деякі із них можуть бути корисними. В [31] було запропоновано спосіб обійти цей недолік, а разом із тим збільшити швидкість процесу навчання. Цей підхід отримав назву Optimal Brain Surgeon(OBS) і складається він із наступних етапів:

1. вибрати мережу із достатньо надлишковою архітектурою та провести її навчання,
2. обчислити  $H^{-1}$  - матрицю зворотно до гессіану:  $H = \frac{\partial^2 E}{\partial w^2}$ ,
3. обчислити міру значущості для кожного елемента:

$$L_q = w_q^2 / (2[H^{-1}]_{qq}), \quad (1.3)$$

де  $w_q$  –  $q$ -ий ваговий коефіцієнт в мережі,

4. якщо мінімальний  $S_q$  значно менший за поточну помилку, то  $w_q$  має бути видалений, після чого перейти до кроку номер 5, в іншому разі перейти до кроку номер 6,
5. оновити вектор вагових коефіцієнтів мережі, використовуючи наступний вираз:

$$\delta W = -\frac{w_q}{[H^{-1}]_{qq}} H^{-1} \zeta_q, \quad (1.4)$$

де  $\zeta_q$  – орт-вектор в площині вагових коефіцієнтів мережі, який відповідає  $q$ -й синаптичній вазі,

6. усі незначущі вагові коефіцієнти видаляються, після чого бажано перенавчити мережу.

Слід зауважити, що перший же крок цього методу, а саме вибір критерію надмірності архітектури, може викликати багато запитань. В цілому ж, запропонований в [31] метод також характеризується значною обчислювальною складністю, хоч і меншою, в порівнянні з OBD, оскільки він не потребує перенавчання всю мережу після видалення кожного вагового коефіцієнта. Але описаним вище методам притаманний ще один істотний недолік - необхідність мати вже навчену нейронну мережу до початку роботи алгоритму. Це обмеження вдалося обійти завдяки методу, що був запропонований у [36]. Міра значущості для вагових коефіцієнтів визначається через тестову статистику  $T$ , опираючись на умову того, що вага обнуляється у процесі навчання мережі:

$$T_{(w_q)} = \log \left( \frac{\left| \sum_{k=1}^N \left( w_q - \eta \left( \frac{\partial E}{\partial w_q} \right)_k \right) \right|}{\eta \sqrt{\sum_{k=1}^N \left( \left( \frac{\partial E}{\partial w_q} \right)_k - \left( \frac{\partial E}{\partial w_q} \right) \right)^2}} \right), \quad (1.5)$$

де  $N$  – кількість прикладів у виборці.

У рамках деструктивного підходу відомі також і багато інших методів оптимізації архітектури нейронної мережі [33-35,37-39], проте в силу специфіки цього підходу всім їм в тій чи іншій мірі властива додаткова обчислювальна складність, а крім того орієнтація на нейронні мережі з архітектурою типу багат шарового персептрона. Використання таких алгоритмів для налаштування інших архітектур, зокрема для нейро-фаззі систем, неможливо, а розробка видається недоцільною, оскільки в будь-якому разі використання деструктивного алгоритму буде мати прямий негативний вплив як на час навчання, так і на час функціонування мережі. Для систем, що розробляються в рамках напряму нейро-фаззі і м'яких обчислень, час є досить критичним параметром, оскільки ці системи орієнтуються на рішення практичних задач. У зв'язку з цим вельми привабливо виглядає використання конструктивного підходу для синтезу архітектури нейронної мережі, про що йдеться далі.

**1.4.2. Конструктивний підхід до налаштування архітектури нейронної мережі.** Суть конструктивного підходу полягає в нарощуванні архітектури нейронної мережі і налаштування її вагових коефіцієнтів паралельно доки не будуть задоволені вимоги критерію зупинки. Тобто це підхід дозволяє не тільки уникнути перенавчання нейронної мережі, а й оптимізацію структуру й архітектури мережі на етапі навчання.

Таким чином, використовуючи конструктивний підхід, можна повністю вирішити питання про вибір початкової архітектури мережі: в загальному випадку вона повинна бути максимально простою, складається з одного або декількох нейронів (залежить від методу). Слід зауважити, що, як правило, в результаті роботи конкретного конструктивного алгоритму на виході виходить нейронна мережа з нетрадиційною архітектурою.

В [40] Джон Платт описує нейронну мережу (RAN - Resource-Allocating Network), яка в процесі навчання додає в свою архітектуру нові обчислювальні елементи (штучні нейрони) кожен раз, коли на вхід мережі подається новий навчальний приклад, який в RAN має двошарову архітектуру. Перший шар складається з нейронів, які відповідають за локальну область з простору вхідних сигналів. У разі якщо вхідний сигнал віддаляється від області конкретного нейрона, то значення сигналу на його виході буде зменшуватися відповідно до співвідношення:

$$x_j = \begin{cases} (1 - (z_j / \chi w_j^2))^2, & \text{ЯКЩО } z_j < \chi w_j^2 \\ 0, & \text{ІНАШКЕ} \end{cases} \quad (1.6)$$

$$z_j = \sum_i (c_{ij} - X_i)^2, \quad (1.7)$$

де  $X_i$  –  $i$ -ий вхід нейронної мережі,

$c_{ij}$  –  $i$ -ий ваговий коефіцієнт  $j$ -го нейрона першого шару,

$\chi$  – параметр налаштування, який підбирається емпіричним шляхом.

Виходи першого шару  $x_j$  подаються на другий шар, який агрегує ці значення і генерує вихідний сигнал. Метою кожного синапсу другого шару є визначити, який вплив надає кожен нейрон першого шару на формування конкретного цільового вектора  $\vec{y}$ . Вихідним сигналом мережі  $\vec{y}$  є зважена сума виходів першого шару плюс незалежний вектор  $\vec{\gamma}$ , що містить постійні елементи:

$$\vec{y} = \sum_j \vec{w}^{[o]} x_j + \vec{\gamma}, \quad (1.8)$$

де  $\vec{w}^{[o]}$ —вектор синаптичних ваг вихідного шару, або в скалярній формі:

$$y_i = \sum_j w_{ji}^{[o]} x_j + \gamma. \quad (1.9)$$

Також  $\vec{\gamma}$  є виходом нейронної мережі у випадку, якщо не активувався жоден з нейронів першого шару. У певному сенсі вираз  $\vec{w}_j^{[o]} x_j$  може розглядатися як певний адитивний елемент, який може бути використаний для того, щоб отримати бажаний вихідний сигнал. Навчання RAN починається з нульового стану, тобто в першому шарі не міститься жодного нейрона, а в другому кількість нейронів дорівнює розмірності задачі, проте, на цьому етапі всі вони, за винятком зсуву  $\gamma$ , не мають вхідних параметрів. Після подачі на вхід першого навчального прикладу у вхідний шар додається перший нейрон, центр функції активації (1.6) якого встановлено наступним чином:

$$\vec{c}_i = \vec{X}, \quad (1.10)$$

де  $k$ — номер прикладу у виборці.

Вихід першого шару автоматично передається на всі нейрони другого шару, а його лінійні синапси налаштовуються таким чином, щоб різниця між виходом мережі і навчальним сигналом була мінімальна:

$$\vec{w}_j^{[o]} = \vec{Y} - \vec{y}, \quad (1.11)$$

де  $\vec{Y}$  — бажаний вихід мережі.

Доданий нейрон буде реагувати на нові вхідні сигнали, якщо вони будуть перебувати в певному інтервалі, який визначаються відстанню між найближчим вектором і новим вхідним образом

$$w_i = \omega \left\| \vec{X} - \vec{c}_{nearest} \right\|, \quad (1.12)$$

де  $\omega$  – параметр покриття, підібраний емпіричним шляхом. Чим більше значення цього параметра, тим на більшу кількість вхідних сигналів будуть реагувати вже існуючі нейрони першого шару.

У RAN використовуються дві умови для додавання нового нейрона в перший шар мережі. По-перше, це відбувається в тому випадку, якщо вхідний сигнал знаходиться далеко від вже існуючих центрів функцій активації нейронів першого шару:

$$\left\| \vec{X} - \vec{c}_{nearest} \right\| > \delta(t), \quad (1.13)$$

а також у випадку, коли із допомогою поточного набору елементів не вдається забезпечити необхідну точність вихідного сигналу:

$$\left\| \vec{Y} - \vec{y}(\vec{X}) \right\| > \varepsilon, \quad (1.14)$$

де  $\varepsilon$  – необхідна точність вихідного сигналу. Якщо при подачі на вхід нового вектора на виході мережі ми отримуємо помилку більшу, ніж  $\varepsilon$ , то у вхідний шар мережі додається новий нейрон з центрами активаційних функцій, налаштованими на поточний вхідний образ. Відстань  $\delta(k)$  – динамічний параметр, який змінює своє значення протягом процесу навчання. Для його калькуляції використовується наступний вираз:

$$\delta(k) = \max \left( \delta_{max} e^{-i/\tau}, \delta_{min} \right), \quad (1.15)$$

де  $\delta_{max}$ ,  $\delta_{min}$ ,  $\tau$  – параметри, що вибираються емпірично.

Якщо згідно з умовами (1.13) і (1.14) не потребується додавання нового нейрона у вхідний шар, то проводиться підналаштування вагових коефіцієнтів вихідного шару. Для цього можуть використовуватися градієнтні методи мінімізації або ж метод найменших квадратів.

На перших етапах навчання в мережу переважно додаються нові елементи, проте через деякий час цей процес сповільнюється і замість додавання нових нейронів у вхідному шар відбувається налаштування синаптичних вагових коефіцієнтів вихідного шару. Такий порядок роботи конструктивного алгоритму стає можливим завдяки використанню двох умов додавання нового нейрона (1.13) та (1.14), забезпечує оптимальну складність моделі нейронної мережі поряд з хорошим рівнем узагальнюючих здібностей. У разі використання виключно (1.13) найбільш ймовірно, що це призведе до перенавчання, а в разі – (1.14) можуть бути пропущені деякі нейрони вхідного шару, що вплине на точність вихідного сигналу мережі.

Серед недоліків запропонованого Джоном Платтом методу має сенс назвати досить велику кількість емпірично підбираємих параметрів, від яких безпосередньо залежить якість роботи RAN.

Надалі підхід до конструктивної організації архітектури нейронної мережі, відомий в англomовній літературі під назвою resource allocation, породив безліч різних модифікацій, які спрямовані на оптимізацію швидкості навчання і точності вихідного сигналу при вирішенні певного кола завдань [41-44].

В рамках конструктивного підходу можна виділити такий напрямок, як каскадні нейронні мережі [45-50], найбільш характерним і ефективним представником яких є каскадно-кореляційний архітектура, запропонована Фальманом і Лебьером в роботі [45]. Основна особливість мережі цього типу полягає в можливості додавання нових вузлів в процесі навчання. На рис. 1.1 приведена схема подібної мережі, яка містить три каскаду,  $n$  входів і один вихід.

На початку процесу навчання формується стандартна одношарова структура з  $n$  входами і єдиним виходом (рис. 1.1), яка навчається за допомогою того чи іншого нелінійного методу навчання.

Після проходження всієї навчальної вибірки  $x(1), x(2), \dots, x(N)$  оцінюється точність апроксимації і в тому випадку, якщо помилка занадто велика, формується каскад з  $n_2$  нейронів-кандидатів, паралельно підключених до вхо-



дів мережі  $1, x_1, x_2, \dots, x_n$  і виходу першого каскаду  $o^{[1]}$ . Нейрони-кандидати, як правило, відрізняються один від одного початковими значеннями синаптичних вагових коефіцієнтів  $W^{[2]}(0)$ , видом функцій активації та алгоритмами навчання. На наступному етапі проводиться навчання нейронів другого каскаду при «заморожених» синаптичних коефіцієнтах  $W^{[1]}(N)$  першого каскаду. На **рис. 1.1** «заморожені» коефіцієнти показані у вигляді заштрихованих суматорів. Серед  $n_2$  нейронів-кандидатів вибирається один нейрон-переможець, параметр кореляції якого **[46]**

$$R_{[q]}^2 = \left| \sum_{k=1}^N \left( o_q^{[2]}(k) - \bar{o}_q^{[2]} \right) \left( e_q^{[2]}(k) - \bar{e}_q^{[2]} \right) \right|, \quad q = 1, 2, \dots, n_2 \quad (1.16)$$

(тут  $\bar{o}_q^{[2]}$  і  $\bar{e}_q^{[2]}$  - середні значення вихідного сигналу і помилки) є максимальним. Саме цей нейрон з «замороженими» вагами  $W^{[2]}(N)$  утворює другий каскад, в той час як нейрони, які програли, видаляються з мережі.

Далі оцінюється точність апроксимації, що забезпечується другим каскадом, і в разі потреби формується набір з  $n_3$  кандидатів третього каскаду, серед яких вибирається переможець з максимальним значенням

$$R_{[q]}^3 = \left| \sum_{k=1}^N \left( o_q^{[3]}(k) - \bar{o}_q^{[3]} \right) \left( e_q^{[3]}(k) - \bar{e}_q^{[3]} \right) \right|, \quad q = 1, 2, \dots, n_3 \quad (1.17)$$

У разі досягнення необхідної точності процес нарощування каскадів завершується і вихідний сигнал останнього каскаду (на рис. 1.1 –  $\bar{o}_q^{[3]}$ ) приймається в якості вихідного сигналу мережі в цілому. В якості основних відмінних рис каскадно-кореляційних мереж слід зазначити наступні:

- ці мережі не вимагають попереднього завдання ні архітектури, ні кількості нейронів в каскадах,
- нейрони в мережу додаються в міру необхідності, утворюючи не приховати шари, а каскади, кожен з яких в якості вхідних сигналів використовує входи мережі і виходи попереднього каскаду,
- навчання не пов'язане з концепцією зворотнього поширення помилок, що дозволяє істотно скоротити час налаштування

— за рахунок «заморожування» синаптичних ваг сформованих раніше каскадів скорочуються обчислювальні витрати на навчання.

Головним недоліком даних мереж прийнято вважати неможливість їх навчання в режимі послідовної обробки інформації [51]. Далі буде показано, як можна подолати це обмеження, синтезуючи на основі мережі, запропонованої в [45], архітектуру, яка буде відповідати критеріям, які висувуються до нейро-фаззі систем.

**1.4.3. Задача нечіткого кластерування даних.** Кластеризація є основним інструментом для аналізу даних. Він знаходить широке застосування в багатьох інженерних і наукових областях, включаючи розпізнавання образів, виділення ознак, вектор квантування, сегментації зображень, біоінформатики і інтелектуального аналізу даних. Кластеризація є класичним методом вибору прототипу ядра на основі нейронних мереж, таких як RBF мережі, і є найбільш корисним для нейро нечітких систем. Кластеризація є неконтрольований метод класифікації, яка ідентифікує деяку неіз структуру, присутню в ЛОР безлічі об'єктів на основі показника подібності. Методи кластеризації можуть бути отримані зі статистичних моделей або конкурентного навчання, і, відповідно, вони можуть бути класифіковані в генеративної (або на основі моделі) і дискриміна- TIVE (або подібності на основі) підходів. Проблема кластеризація також може бути змодельована як КС. Кластеризація нейронні мережі є статистичні моделі, де функція щільності ймовірності (PDF) для даних оцінюється шляхом вивчення його параметрів.

Нечітке кластерування даних – один з напрямів кластерного аналізу, що використовує для обробляння даних деякі принципи та елементи нечіткої логіки [77, 78]. Концептуальний взаємозв'язок між кластерним аналізом та нечіткою логікою ґрунтується на тій обставині, що розв'язання задач структурування складних систем формує здебільшого кластери об'єктів, що є розмиті, нечіткі за своєю природою. Така нечіткість може полягати в тому, що перехід од належності до неналежності образів щодо певних кластерів рад-

ше поступовий, аніж стрибкуватий. Тому адекватнішою в таких випадках є не однозначна належність до певного кластеру, а низка рівнів належності до кількох кластерів. Вимога однозначно розкластерувати елементи досліджуваної проблемної області є вельми грубою та жорсткою, особливо у випадках, коли треба розв'язати погано або слабо структуровані задачі інтелектуального аналізу даних. Засоби нечіткого кластерування послаблюють цю вимогу введенням до розгляду нечітких кластерів та їхніх функцій належності, які приймають значення на інтервалі  $[0, 1]$ .

З-поміж цілої низки методів та підходів нечіткого кластерування особливе місце займають методи, що ґрунтуються на цільових функціях [79]. Такі методи розв'язують задачу оброблення даних, оптимуючи деякий заздалегідь заданий критерій якості. Найвідомішим представником цього класу методів є метод нечітких  $s$ -середніх [77], що його широко застосовують у задачах різноманітної складності, коли навчальний сигнал невідомий. Але хоча стандартний метод нечітких  $s$ -середніх є значно просунутіший супроти методів чіткого кластерування, але все ж таки й він має вади. Справа в тому, що однією з умов використання цього методу є вимога, щоби сума рівнів належності будь-якого образу за всіма кластерами дорівнювала одиниці. Ця штучна вимога у випадках рівновіддаленості деякого образу від усіх кластерів спричиняє те, що такий образ отримує рівень належності до кожного з кластерів, який не залежить од відстані між образом та центром відповідного кластеру. Іще однією вагою методу, яка випливає з попередньої, є припущення, що під час оброблення даних образи, що належать новим кластерам, з'явитися не можуть. Вочевидь, в реалістичних задачах це не завжди так. До того ж образи, що надходять на вхід методу, можуть бути звичайним шумом, завадами. Стандартний метод нечітких  $s$ -середніх не впорається з такою ситуацією, що відповідно позначиться на ефективності кінцевого нечіткого розбиття даних.

Зазначену ваду долає метод можливісного нечіткого кластерування даних – метод можливісних  $s$ -середніх [77, 80, 81]. Він не має вимог щодо значення суми рівнів належності образів за всіма кластерами, що, відповідно, покращує

його дієвість за умов наявності шуму в ухідному сигналі. Проте цей метод має певні труднощі з його початковим ініціалізуванням.

Беручи до уваги те, що обидва розглянуті методи належать до методів нечіткого кластерування даних, тут годилося б зазначити, що коли виникає потреба виокремити стандартний метод нечітких  $s$ -середніх, його називають імовірнісним.

Два розглянуті методи нечіткого кластерування даних – імовірнісний та можливісний – є базові методи, які утворюють цілі сім'ї похідних від них і пристосованих до певних специфічних задач оброблення даних [77]. Зазвичай, кожен зі згаданих методів обробляє дані в пакетному режимі. У ситуаціях, коли дані надходять окремо й їх потрібно обробляти он-лайн, для методів нечіткого кластерування запропоновано послідовні модифікації [77, 82].

Іще одна вада стандартних методів нечіткого кластерування даних – нездатність виявляти кластери складної, несферичної форми. Цю ваду, oprіч уже згаданого ієрархічного кластерування, долає метод нечіткого кластерування Густафсона-Кесселя, який, замість Евклідової, використовує Магаланобісову метрику [83].

Як видно, існує розмаїття методів нечіткого кластерування, зоснованих на оптимуванні цільової функції. Але усі вони ґрунтуються на припущенні, що дослідникові попередньо відома кількість кластерів, що їх треба виявити. Допевне, таке припущення слушне не для всіх задач, адже інколи кількість кластерів у вхідних даних може бути невідомою або вона може змінюватися в часі. Цей випадок є предметом розгляду систем оброблення даних із мінливою (еволюційною) архітектурою [84], які здебільшого є гібридними системами з поліпшеними архітектурами, що поєднують декілька напрямків обчислювального інтелекту.