

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

На правах рукопису

**Копалпані Дар'я Сергіївна**

УДК 004.032.26

**ЕВОЛЮЦІЙНІ НЕЙРО-ФАЗЗІ МЕРЕЖІ З КАСКАДНОЮ  
СТРУКТУРОЮ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАННИХ**

05.13.23 — системи та засоби штучного інтелекту

Дисертація на здобуття наукового ступеня  
кандидата технічних наук

Науковий керівник  
**Бодянський Євгеній Володимирович,**  
доктор технічних наук, професор

Харків — 2015

## ЗМІСТ

|   |           |
|---|-----------|
| <b>Розділ 1. Огляд стану проблеми та постановка задачі дослідження</b>                                    | <b>5</b>  |
| 1.1. Основні події в нейронних мережах досліджень . . . . .   | 5         |
| 1.2. Навчання та самонавчання штучних нейронних мереж . . . . .   | 8         |
| 1.3. Гібридні системи обчислювального інтелекту . . . . .   | 10        |
| 1.4. Гібридні нейро-фаззі системи та питання структурної адаптації  | 11        |
| 1.4.1. Деструктивний підхід до налаштування архітектури нейронної мережі . . . . .                        | 12        |
| 1.4.2. Конструктивний підхід до налаштування архітектури нейронної мережі . . . . .                       | 15        |
| 1.4.3. Задача нечіткого кластерування даних . . . . .   | 19        |
| <b>Розділ 2. Гібридна каскадна нейро-фаззі мережа з оптимізацією пулу нейронів</b>                        | <b>22</b> |
| 2.1. Архітектура оптимізованої каскадної нейронної мережі . . . . .                                       | 24        |
| 2.2. Навчання елементарних персепtronів Розенблатта у каскадній оптимізованій системі . . . . .           | 26        |
| 2.3. Навчання нео-фаззі нейронів у оптимізованій каскадній нейронній мережі . . . . .                     | 29        |
| 2.4. Розширенні нео-фаззі нейрони в якості елементів гібридної каскадної мережі, що еволюціонує . . . . . | 36        |
| 2.5. Оптимізація пулу нео-фаззі нейронів . . . . .  | 39        |
| Висновки до розділу 2 . . . . .   | 42        |
| <b>Розділ 3. Багатовимірна каскадна нео-фаззі система, що еволюціонує</b>                                 | <b>44</b> |

|  |           |
|--|-----------|
| 3.1. Багатовимірна каскадна система, що еволюціонує, побудована на нео-фаззі нейронах . . . . .  | 45        |
| 3.1.1. Оптимізація пулу нео-фаззі нейронів багатовимірної каскадної системи, що еволюціонує . . . . .  | 50        |
| 3.2. Багатовимірна каскадна система, що еволюціонує, побудована на багатовимірних нео-фаззі нейронах . . . . .                                     | 53        |
| 3.2.1. Багатовимірний нео-фаззі нейрон . . . . .   | 54        |
| 3.2.2. Метод визначення локально оптимального вихідного сигналу пулу багатовимірних нео-фаззі нейронів каскадної системи, що еволюціонує . . . . . | 56        |
| Висновки до розділу 3.1 . . . . .  | 59        |
| <b>Розділ 4. Каскадна нейронна мережа, що еволюціонує, для послідовного нечіткого кластерування потоків даних</b>                                  | <b>60</b> |
| 4.1. Труднощі та особливості відомих методів кластерування даних . . . . .   | 60        |
| 4.1.1. Нечітке послідовне кластерування . . . . .  | 63        |
| 4.2. Критерії дійсності нечіткого кластерування . . . . .  | 64        |
| 4.3. Архітектура каскадної мережі, що еволюціонує, для нечіткого кластерування . . . . .   | 66        |
| 4.4. Адаптивне навчання вузлів каскадної нейро-фаззі системи, що еволюціонує . . . . .   | 66        |
| 4.5. Керування каскадами самонавчаної нейро-фаззі системи, що еволюціонує . . . . .  | 71        |
| Висновки до розділу ?? . . . . .   | 73        |
| <b>Розділ 5. Моделювання та практичне застосування розроблених методів та архітектур</b>   | <b>75</b> |
| 5.1. Моделювання розширенного нейро-фаззі нейрона . . . . .  | 75        |
| 5.2. Моделювання гібридної каскадної нейро-фаззі мережі з розширенними нео-фаззі нейронами та оптимізацією пулу нейронів .                         | 83        |

|   |            |
|---|------------|
| 5.3. Моделювання багатовимірної гібридної каскадної нейро-фаззі мережі, що еволюціонує, з оптимізацією пулу нейронів . . . . .                              | 89         |
| 5.4. Моделювання самонавчаної нейро-фаззі системи, що еволюціонує . . . . .   | 93         |
| 5.5. Розв'язування практичних задач за допомогою розробленої самонавчаної гібридної каскадної системи, що еволюціонує . . . . .                             | 113        |
| 5.6. Розв'язування практичних задач за допомогою розробленої гібридної каскадної нейро-фаззі мережі, що еволюціонує, з оптимізацією пулу нейронів . . . . . | 117        |
| Висновки до розділу 5 . . . . .   | 127        |
| <b>Список використаних джерел</b>   | <b>129</b> |

## 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 конекціонистичними 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 нейромеханізмів 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 мембрани і трансмембраних 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, Розенблatt [31] запропонував 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 привело 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 цьому було використання виключно лінійної функції активації.

У 1970-ті роки, Гросберг [12, 13], фон дер Мальцбруг [37] і Фукусіма [9], натхнені результатами дослідженій зорової кори, опублікували новаторські роботи на предмет навчання і конкурентної самоорганізації. Фукусіма запропонував свою модель когнітрону і неокогнітрону, в рамках парадигми конкурентного навчання. Неокогніtron являє собою ієрархічну багатошарову нейронну мережу, спеціально призначенну 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 відображеній в топологічно впорядковано. Гросберг 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 НВІС.

Найбільш помітним орієнтиром в нейронної мережі досліджень є алгоритм зворотного поширення (BP) навчання, пропонованих для моделі багатошаровий персепtron (MLP) в 1986 році Rumelhart, Хіnton, і Вільямс [33]. Надалі, алгоритм BP був виявлений вже був винайдений в 1974 році Werbos [39]. У 1988 році, Broomhead i Lowe запропонував мережеву модель радіальної базисної функції (RBF) [3]. Обидва MLP і RBF мережі є універсальними аппроксиматорами.

У 1982 році Оя запропонував мережу аналіз 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 байесівської мережі [30]. Байесівської 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 ґрунтуються на статистичної 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 структурою, 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 учителем

лем застосовують у випадках, коли відома бажана реакція системи в кожну мить часу, себто відомий навчальний сигнал, який упливає 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 у розбігається з «правильною» реакцією вчителя, через що постає похибка  $e = y?? - y$ . 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) абопроміжкові часу (у неперервному випадку). Границним 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 систем обробляння 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 синапсовых вагах і їх не можна буде витлумачити. Однак, якщо виразити ваги нейронної мережі за допомогою нечітких правил, з'являється можливість подолати неінтерпретованість результатів роботи нейронної мережі. У такий спосіб нейрофаззі системи дають змогу створювати системи обробляння 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 обробляння 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 гіbridного підходу.

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

В наш час для розв'язання задач, які пов'язані з інтелектуальною обробкою даних в умовах апріорної 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 гіbridну систему. Такі системи відповідають усім вимогам до інтелектуальних систем і отримали назву гіbridні системи обчислювального інтелекту. Нейро-фаззі системи та м'які обчислювання є напрямами 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]:

1. обчислювальні моделі, що засновані на біологічних прототипах,
2. паралельна обробка даних у послідовному режимі,
3. в основі системи лежать експертні знання,
4. стійкість системи до зашумлення,
5. 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 зашумленні значення. Однак навчання таких систем зводиться до налаштування сінаптичних 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 видаюється очевидно корисним зсинтезувати 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  $R_n R_m$ . В якості такого відображення може виступати залежність вихідних параметрів процесу від вхідних, прогнозування від передісторії, класа об'єкту від набору його властивостей, управлюючої 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:

1. 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,
2. 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.
- 3.

**1.4.1. Деструктивний підхід до налаштування архітектури нейронної мережі.** Основна ідея деструктивних алгоритмів полягає у видаленні параметрів, що мабуть найменьший 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 могут бути цілком видалені як деякі вхіжні параметри або вузли у прихованих шарах, так і лише деякі синаптичні зв'язки між нейронами, що мають лише один параметр  $h_{\text{дефіс}}$  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 елементу мережі обчислити міру значущості:**

$$FS = \sum_{i=1}^N \sum_{l=1}^M u_{il}^\beta (\|x_i - z_l\|^2 - \|z_l - z\|^2), \quad (1.1)$$

де  $v_q$  - вихідний сигнал  $q$ -го елементу мережі, **розраховується по формулі:** де  $Vq$  - 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 мережи деяку кількість елементів, для яких міра значущості  **$s_q$**  найменьша. В цьому контексті під видаленням елементу мається 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 запропановано спосіб обійти цей недолік, а разом із тим збільшити швидкість процесу навчання. Цей подхід отримав назву Optimal Brain Surgeon(OBS) і складається він із наступних етапів:

1. вибрати мережу із достатньою надлишковою архітектурою та провести її навчання **формула**,
2. обчислити матрицю зворотню до гессіану: **формула**,
3. обчислити міру значущості для кожного елемента **формула**,
4. якщо мінімальний **sq** значно менший за поточну помилку, то **wq**
5. **додати текст**
6. **додати текст**

Зауважимо, що перший же крок цього методу може породити чимало запитань, пов'язаних з критерієм надмірності архітектури нейронної мережі. В цілому ж, запропонований в [31] метод також характеризується значною обчислювальною складністю, хоч і меншою, в порівнянні з OBD, оскільки немає необхідності перенавчати всю мережу після видалення кожного вагового коефіцієнта. Обом описаним вище методам характерний ще один істотний недолік - необхідність мати вже навчену нейронну мережу до початку роботи алгоритму. Це обмеження вдалося обійти в [36]. У запропонованому методі міра значущості для вагових коефіцієнтів визначається тестової статистикою, виходячи з припущення, що вага прирівнюється нулю в процесі навчання мережі:

В 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 обчислю-

вальна складність, а крім того орієнтація на нейронні мережі з архітектурою типу багатошарового персептрона. Використання таких алгоритмів для настройки інших архітектур, зокрема для нейро-фаззі систем, неможливо, а розробка недоцільною, оскільки в будь-якому деструктивний алгоритм буде вносити відчутно допомагає прямий вплив, як на час навчання, так і на час функціонування мережі. Як вже говорилося вище, для систем, що розробляються в рамках напряму нейро-фаззі і м'які обчислення, 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 до того моменту, поки не будуть задоволені вимоги критерію зупину. Цей підхід (як і деструктивний) допомагає уникнути перенавчання нейронної мережі, а також виробляє структурну оптимізацію її архітектури на етапі навчання.

Використовуючи конструктивний підхід, вдається повністю вирішити питання про вибір початкової архітектури мережі: в загальному випадку вона повинна бути максимально простою, що 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.

Однією з перших робіт в цьому напрямку, що отримала бурхливий розвиток, є робота Джона Платта. В [40] він описує нейронну мережу (RAN - Resource-Allocating Network), яка в процесі навчання додає в свою архітектуру нові обчислювальний елементи (штучні нейрони) кожен раз, коли на вхід мережі подається новий навчальний приклад, який в RAN має двоша-

рову 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:

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  $y$ . Вихідним сигналом мережі  $y$  є зважена сума виходів першого шару плюс незалежний вектор, що містить постійні елементи:

Також є виходом нейронної мережі в разі, якщо не активували [0] жоден з нейронів першого шару. У певному сенсі вираз  $w_j x_j$  може розглядатися як певний адитивний елемент, який може бути добавлені вичленіз для того, щоб отримати вихідний сигнал. Навчання RAN починається з нульового стану, тобто перший шар не містить жодного нейрона, а другий - рівно стільки, яка розмірність розв'язуваної задачі, проте, на цьому етапі все вони не мають вхідних параметрів (за винятком зсуву). Після подачі на вхід першого навчального прикладу у вхідній шар додається перший нейрон з центром активаційної функції (1.6) встановленим наступним чином:

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

Доданий нейрон буде реагувати на нові вхідні сигнали, якщо вони будуть перебувати на певному інтервалі, що визначаються відстанню між найближчим запомненим вектором і новим вхідним чином

де - параметр покриття, обибраний емпірично. 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 разі, якщо вхідний сигнал зна-

ходиться далеко від вже існуючих центрів функцій активації нейронів першого шару:

Якщо при подачі на вхід нового вектора на виході мережі ми отримуємо помилку більшу, ніж, то у вхідний шар мережі додається новий нейрон з центрами активаційних функцій, налаштованими на поточний вхідний образ. шляхом. Якщо згідно з умовами (1.13, 1.14) додавання нового нейрона у вхідній шар не потрібно, то проводиться підстроювання 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, 1.14), і забезпечує оптимальну складність моделі нейронної мережі поряд з хорошим рівнем узагальнюючих здібностей. У разі використання виключно (1.13) найбільш ймовірно, що ми зіткнемося з ефектом 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 можна виділити досить велику кількість емпірично обираних параметрів (, max, min,), від яких безпосередньо залежить якість роботи RAN.

Надалі підхід до конструктивної організації архітектури нейронної мережі, відомий в англомовній літературі під назвою resource allocation, породив безліч різних модифікацій, спрямованих на оптимізацію швидкості навчання і точності вихідного сигналу при вирішенні певного кола завдань [41-44]. Однак далі розглядається інший конструктивний алгоритм, що володіє тими ж перевагами, що і запропонований в [40], але в той же самий час позбавлений недоліків останнього.

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 С. Фальманом і К. Лебьєром 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 каскади,  $n$  входів і один вихід.

На початку процесу навчання формується стандартна одношарова структура з  $n$  входами і 1 єдиним виходом (рис. 1.1), яка навчається за допомогою того чи іншого нелінійного методу навчання.

Після пред'явлення всієї навчальної вибірки  $x(1), x(2), \dots, x(N)$  оцінюється точність апроксимації і в разі, якщо помилка занадто велика, формується каскад з  $n_2$  нейронів-кандидатів, паралельно підключених до входів мережі  $1, x_1, x_2, \dots, x_n$  і виходу першого каскаду о [1]. Нейрони-кандидати, як правило, відрізняються один від одного початковими значеннями синаптичних ваг  $W[2](0)$ , видом функцій активації та алгоритмами навчання. Далі проводиться навчання нейронів другого каскаду при «заморожених» синаптичних вагах  $W[1](N)$  першого. На рис. 1.1 «заморожені» ваги показані у вигляді заштрихованих сумматорів. Серед  $n_2$  нейронів-кандидатів вибирається один нейрон-переможець, у якого параметр кореляції [46]

(Тут  $oq[2]$  і  $eq[2]$  - середні значення вихідного сигналу і помилки) є максимальним. Саме цей нейрон з «замороженими» вагами  $W[2](N)$  утворює другий каскад, в той час як «ті, хто програв» нейрони вилучаються 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

У разі досягнення необхідної точності процес нарощування каскадів завершується і вихідний сигнал останнього каскаду (на рис. 1.1 - о [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 зворотного 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, синтезувавши на основі мережі, запропонованої в [45], архітектуру, яка буде відповідати критеріям, які висуваються 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 ґрунтуються 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 радше 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 велими грубою та жорсткою, особливо у випадках, коли треба розв'язати погано або слабко 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 обробляння даних, оптимуючи деякий заздалегідь заданий критерій якості. Найвідомішим представником цього класу методів є метод нечітких c-середніх [77], що його широко застосовують у задачах різноманітної складності, коли навчальний сигнал невідомий. Але хоча стандартний метод нечітких c-середніх є значно просунутіший супроти методів чіткого кластерування, але все ж таки й він має вади. Справа в тому, що однією з умов використовування цього методу є вимога, щоби сума рівнів належності будь-якого образа за всіма кластерами дорівнювала одиниці. Ця штучна вимога у випадках рівновіддаленості деякого образа 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 вадою методу, яка випливає з попередньої, є припущення, що під час обробляння даних образи, що належать новим кластерам, з'явиться не можуть. Вочевидь, в реалістичних задачах це не завжди так. До того ж обраzi, що надходять на вхід методу, можуть бути звичайним шумом, завадами. Стандартний метод нечітких c-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 с-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 с-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 обробляння 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 ієархічного 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 припущеннях, що дослідників 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 обробляння даних із мінливою (еволюційною) архітектурою [84], які здебільшого є гіbridними системами з поліпшеними архітектурами, що поєднують декілька напрямків обчислювального інтелекту.

## РОЗДІЛ 2

# ГІБРИДНА КАСКАДНА НЕЙРО-ФАЗЗІ МЕРЕЖА З ОПТИМІЗАЦІЄЮ ПУЛУ НЕЙРОНІВ

Зазвичай під «навчанням» розуміють процес коригування синаптичних вагових коефіцієнтів за допомогою певної процедури оптимізації, що ґрунтуються на пошуку екстремуму заданого критерію навчання. Якість процесу навчання може бути поліпшена шляхом коригування топології мережі поспіль з синаптичними вагами [23, 33]. Ця ідея лежить в основі систем обчислювального інтелекту, що еволюціонують [37, 47].

Мабуть, найбільш відомою реалізацією цього підходу є каскадно-кореляційні нейронні мережі [27, 49, 55], привабливі високою ефективністю та простотою налаштування як синаптичних вагових коефіцієнтів, так і топології мережі. Така мережа напочатку містить лише один пул (ансамбль) нейронів, які навчаються назалежно один від іншого (перший каскад). Кожен нейрон у пулі може мати відмінні функції активації та метод навчання. Доки навчання триває, нейрони у пулі не взаємодіють один з одним. Після того, як процес налаштування вагових коефіцієнтів завершився для всіх нейронів пулу першого каскаду, кращий нейрон відповідно до обраного критерію навчання формує перший каскад і коефіцієнти його синаптичних ваг більше не коригуються. Далі формується другий каскад зазвичай з нейронів, подібних до нейронів першого каскаду. Різниця лише в тому, що нейрони, які навчаються в пулі другого каскаду, мають додатковий вхід ( $i$ , отже, додатковий синаптичний ваговий коефіцієнт) - вихід першого каскаду. Подібно до першого каскаду, у другому каскаді залишається лише один найбільш продуктивний нейрон і його синаптичні вагові коефіцієнти фіксуються. Аналогічним чином нейрони третього каскаду матимуть два додаткових входи, а саме виходи першого та другого каскадів. Еволюційна мережа продовжуватиме розширяти свою ар-

хітектуру новими каскадами, доки вона не досягне бажаної якості вирішення завдання для заданого набору даних.

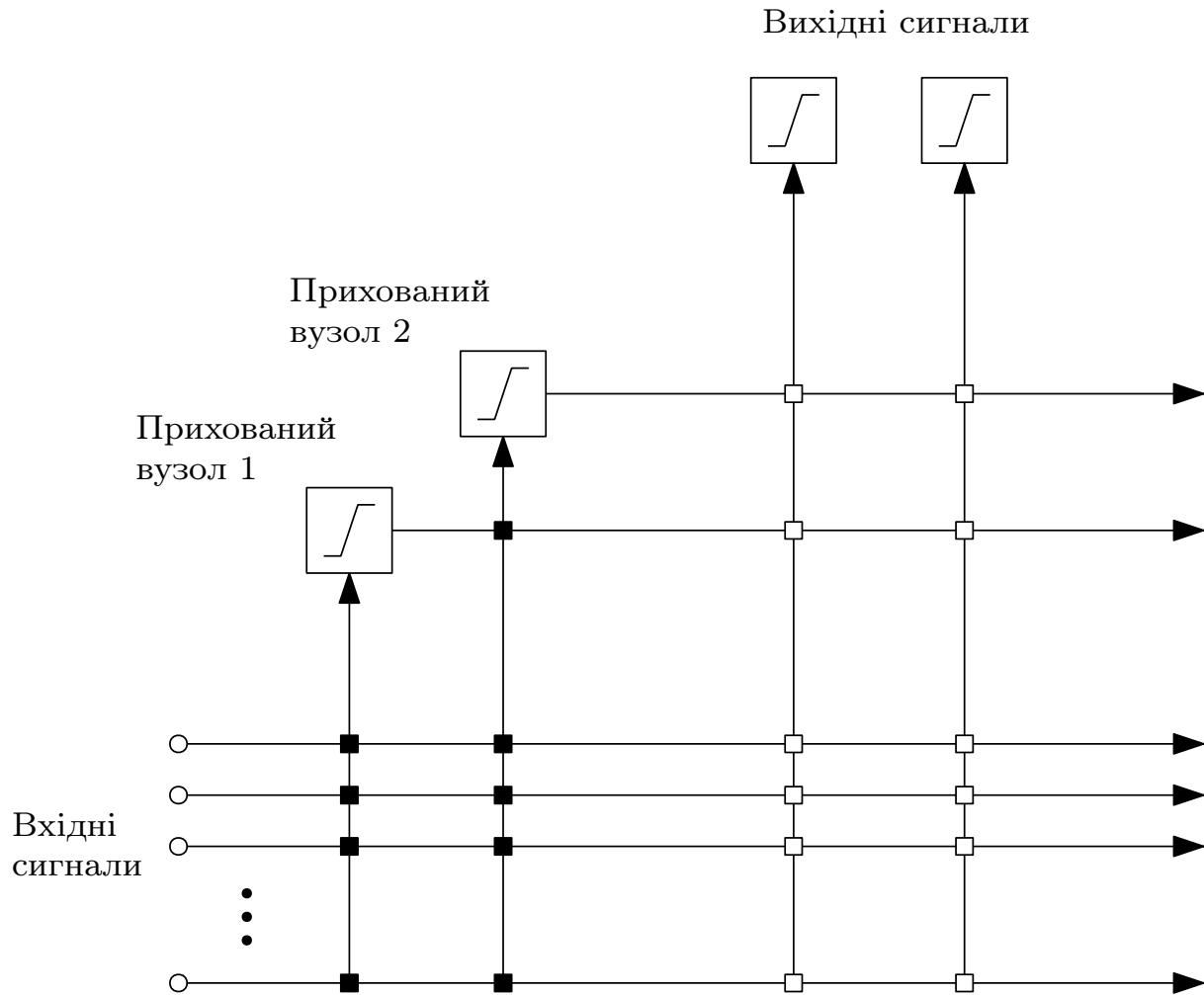


Рис. 2.1 – Архітектура каскадної системи (за Фальманом та Леб’єром) після додавання двох прихованих вузлів. Вхідні сигнали, що надходять до вертикальних ліній, сумуються; вагові коефіцієнти, позначені  $\square$ , – зафіковані, позначені  $\blacksquare$ , – налаштовуються

Автори найпопулярнішої каскадної нейронної мережі, що еволюціонує, CasCorLA (схему наведено на рис. 2.1), Фальман та Леб’єр, використовували елементарні персептрони Розенблатта з традиційними сигмоїдальними функціями активації і коригували синаптичні вагові коефіцієнти за допомогою QuickProp-алгоритму [27], що є модифікацією  $\delta$ -правила. Оскільки вихідний сигнал таких нейронів нелінійно залежить від синаптичних ваг, швидкість навчання не може бути суттєво збільшена для таких нейронів.

Для уникнення багатоепохового навчання [11, 12, 15–17, 34, 41, 74] доцільно в якості вузлів системи використовувати такі типи нейронів, що їх виходи лінійно залежать від синаптичних ваг, що дозволить використовувати оптимальні за швидкодією методи навчання та обробляти дані в онлайн режимі.

Проте варто зазначити, що у випадку послідовного навчання системи, неможливо визначити найкращий нейрон у пулі, адже при оброблянні нестационарних об'єктів певний нейрон може бути кращим для однієї частини тренувальної вибірки, проте поступатися у точності іншому нейрону на іншій частині вибірки. Отже доцільно зберегти усі нейрони пулу та використовувати певну оптимізуючу процедуру (відповідно обраному критерію якості) задля визначення нейрона-переможця на кожному кроці обробляння даних.

## 2.1. Архітектура оптимізованої каскадної нейронної мережі

Архітектура пропонованої гібридної системи з оптимізованим пулом нейронів у кожному каскаді наведена на рис. 2.2.

На вхід такої системи (так званий «рецептивний» шар) подається векторний сигнал

$$x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T, \quad (2.1)$$

де  $k = 1, 2, \dots$  – кількість образів у таблиці «об'єкт - властивість» або поточний дискретний час.

Ці сигнали подаються на входи кожного нейрона в мережі  $N_j^{[m]}$  ( $j = 1, 2, \dots, q$  – кількість нейронів у тренувальному пулі,  $m = 1, 2, \dots$  – номер каскаду) з вихідним сигналом  $\hat{y}_j^{[m]}(k)$ . Далі вихідні сигнали кожного каскаду  $\hat{y}_j^{[m]}(k)$  надходять до «узагальнюючого» вузлу  $GN^{[m]}$ , який генерує поточно-оптимальний вихідний сигнал відповідного каскаду  $\hat{y}^{*[m]}$ . Слід зауважити, що вхідними сигналами першого каскаду є вектор  $x(k)$  (що може містити опціональне порогове значення  $x_0(k) \equiv 1$ ), другий каскад має додатковий вхід для

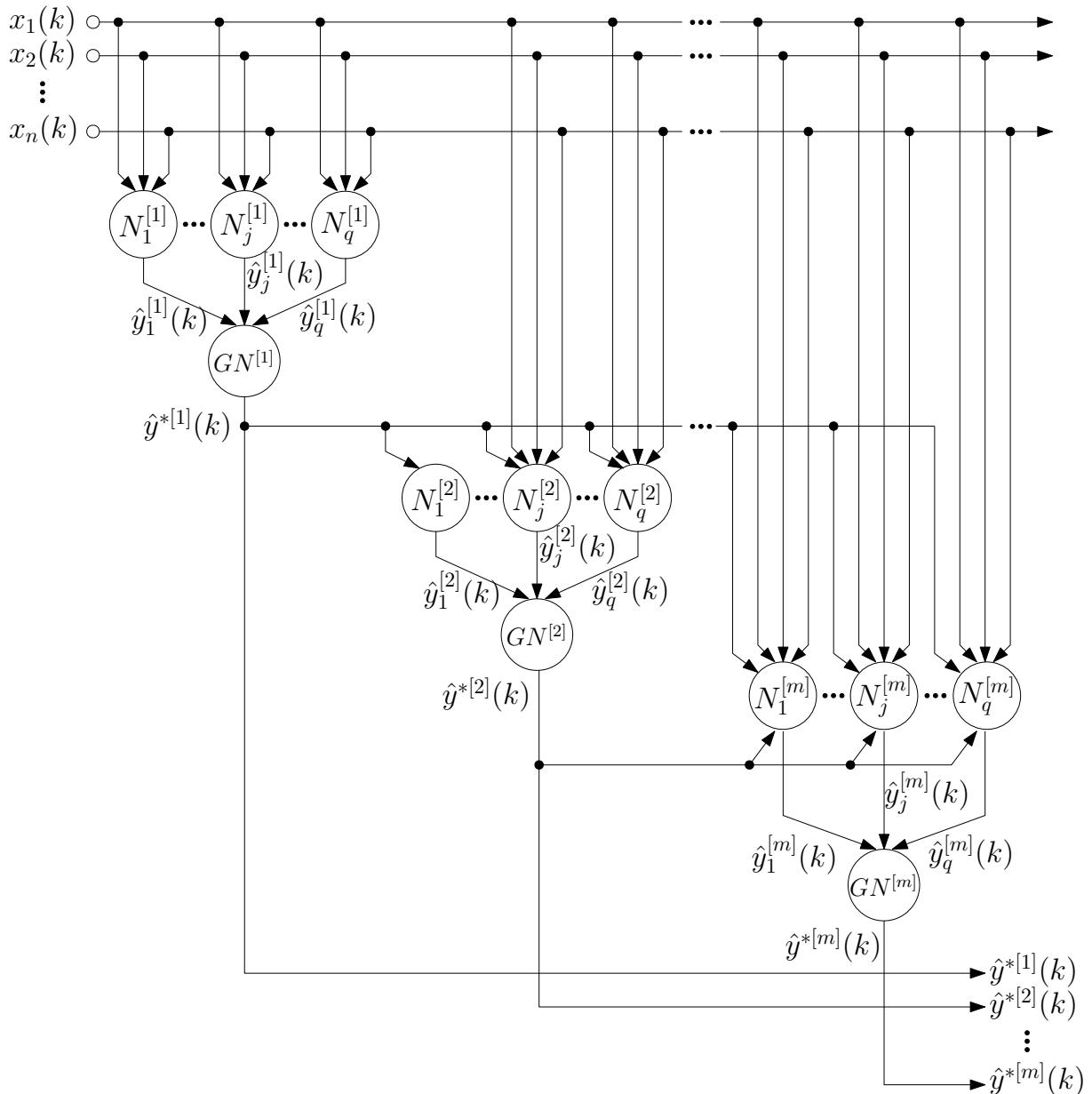


Рис. 2.2 – Архітектура гібридної системи з оптимізованим пулом нейронів

сгенерованого першим каскадом вихідного сигналу  $\hat{y}^{*[1]}(k)$ , нейрони третього каскаду оброблятимуть два додаткових сигналі  $\hat{y}^{*[1]}(k), \hat{y}^{*[2]}(k)$ , нейрони  $m$ -го каскаду матимуть  $(m - 1)$  додаткових вхідних сигналів:  $\hat{y}^{*[1]}(k), \hat{y}^{*[2]}(k), \dots, \hat{y}^{*[m-1]}(k)$ . Під час тренування системи нові каскади додаються доки не буде досягнута бажана точність.

## 2.2. Навчання елементарних персепtronів Розенблатта у каскадній оптимізованій системі

Наразі вважатимемо  $j$ -ї вузол  $m$ -ого каскаду елементарним персепtronом Розенблатта з активаційною функцією

$$0 < \sigma_j^m \left( \gamma_j^{[m]} u_j^{[m]} \right) = \frac{1}{1 + e^{-\gamma_j^{[m]} u_j^{[m]}}} < 1, \quad (2.2)$$

де  $u_j^{[m]}$  – внутрішній активаційний сигнал  $j$ -ого нейрону  $m$ -ого каскаду,  $\gamma_j^{[m]}$  – параметр посилення.

У такому випадку вихідні сигнали нейронів тренувального пулу первого каскаду матимуть вигляд

$$\hat{y}_j^{[1]} = \sigma_J^{[1]} \left( \gamma_j^{[1]} \sum_{i=0}^n w_{ji}^{[1]} x_i \right) = \sigma_j^{[1]} \left( \gamma_j^{[1]} w_j^{[1]T} x \right), \quad (2.3)$$

де  $w_{ji}^{[1]}$  –  $i$ -ї ваговий коефіцієнт  $j$ -ого нейрону первого каскаду.

Вихідні сигнали другого каскаду дорівнюватимуть

$$\hat{y}_j^{[2]} = \sigma_J^{[2]} \left( \gamma_j^{[2]} \left( \sum_{i=0}^n w_{ji}^{[2]} x_i + w_{j,n+1}^{[2]} \hat{y}^{*[1]} \right) \right), \quad (2.4)$$

вихідні сигнали  $m$ -ого каскаду матимуть вигляд

$$\begin{aligned} \hat{y}_j^{[m]} &= \sigma_j^{[m]} \left( \gamma_j^{[m]} \left( \sum_{i=0}^n w_{ji}^{[m]} x_i + w_{j,n+m-1}^{[m]} \hat{y}^{*[1]} \right. \right. \\ &\quad \left. \left. + w_{j,n+2}^{[m]} \hat{y}^{*[2]} + \cdots + w_{j,n+m-1}^{[m]} \hat{y}^{*[m-1]} \right) \right) \\ &= \sigma_j^{[m]} \left( \gamma_j^{[m]} \sum_{i=0}^{n+m-1} w_{ji}^{[m]} x_j^{[m]} \right) = \sigma_j^{[m]} \left( w_j^{[m]T} x^{[m]} \right), \end{aligned} \quad (2.5)$$

де  $x^{[m]} = (x^T, \hat{y}^{*[1]}, \hat{y}^{*[m-1]})^T$ .

Таким чином, нейронна мережа з персепtronами Розенблатта у якості вузлів, що містить  $m$  каскадів, залежить від  $\left(m(n+2) + \sum_{p=1}^{m-1} p\right)$  параметрів, у тому числі від параметрів посилення  $\gamma_j^{[p]}$ ,  $p = 1, 2, \dots, m$ .

У якості критерію навчання можна використовувати загальноприйняту квадратичну функцію

$$\begin{aligned} E_j^{[m]} &= \frac{1}{2} \left( e_j^{[m]}(k) \right)^2 = \\ &= \frac{1}{2} \left( y(k) - \hat{y}_j^{[m]}(k) \right)^2 = \\ &= \frac{1}{2} \left( y(k) - \sigma_j^{[m]} \left( \gamma_j^{[m]} w_j^{[m]T} x^{[m]}(k) \right) \right)^2, \end{aligned} \quad (2.6)$$

де  $y(k)$  – бажане значення вихідного сигналу.

Градієнтну оптимізацію критерію (2.6) відносно  $w_j^{[m]}$  можна записати у вигляді

$$\begin{aligned} w_j^{[m]}(k+1) &= w_j^{[m]} + \eta_j^{[m]}(k+1) e_j^{[m]}(k+1) \gamma_j^{[m]} \hat{y}_j^{[m]}(k+1) \\ &\quad \times \left( 1 - \hat{y}_j^{[m]}(k+1) \right) x^{[m]}(k+1) = \\ &= w_j^{[m]}(k) + \eta_j^{[m]}(k+1) e_j^{[m]}(k+1) \gamma_j^{[m]} J_j^{[m]}(k+1), \end{aligned} \quad (2.7)$$

де  $\eta_j^{[m](k+1)}$  – параметр кроku навчання.

Мінімізувати критерій (2.6) відносно  $\gamma_j^{[m]}$  можна за допомогою алгоритму Крушке-Мовеланна [42]

$$\begin{aligned} \gamma_j^{[m]}(k+1) &= \gamma_j^{[m]}(k) + \eta_j^{[m]}(k+1) e_j^{[m]}(k+1) \hat{y}_j^{[m]}(k+1) \\ &\quad \times \left( 1 - \hat{y}_j^{[m]}(k+1) \right) u_j^{[m]}(k+1). \end{aligned} \quad (2.8)$$

Поєднуючи (2.7) та (2.6), отримаємо алгоритм навчання для  $j$ -ого нейрону  $m$ -ого каскаду

$$\begin{aligned} \frac{w_j^{[m]}(k+1)}{\gamma_j^{[m]}(k+1)} &= \frac{w_j^{[m]}(k)}{\gamma_j^{[m]}(k+)} + \eta_j^{[m]}(k+1)e_j^{[m]}(k+1)\hat{y}_j^{[m]}(k+1) \\ &\times \left(1 - \hat{y}_j^{[m]}(k+1)\right) \left( \frac{\gamma_j^{[m]}x^{[m]}(k+1)}{u_j^{[m]}(k+1)} \right), \end{aligned} \quad (2.9)$$

або, вводячи нові змінні, у більш компактній формі

$$\begin{aligned} \tilde{w}_j^{[m]}(k+1) &= \tilde{w}_j^{[m]}(k) + \eta_j^{[m]}(k+1)e_j^{[m]}(k+1)\hat{y}_j^{[m]}(k+1)\tilde{x}^{[m]}(k+1) \\ &= \tilde{w}_j^{[m]}(k) + \eta_j^{[m]}(k+1)e_j^{[m]}(k+1)\tilde{J}_j^{[m]}(k+1). \end{aligned} \quad (2.10)$$

Використовуючи регуляризуючий параметр (momentum term) [19, 58, 61], можна удосконалити процес корегування синаптичних вагових коефіцієнтів під час навчання. Тоді, замість критерію (2.6) слід використовувати функцію

$$\begin{aligned} E_j^{[m]}(k) &= \frac{\eta}{2} \left( e_j^{[m]}(k) \right)^2 \\ &+ \frac{1-\eta}{2} \left\| \tilde{w}_j^{[m]}(k) - \tilde{w}_j^{[m]}(k-1) \right\|^2, \quad 0 < \eta \leq 1. \end{aligned} \quad (2.11)$$

Тоді алгоритм навчання приймає вигляд

$$\begin{aligned} \tilde{w}_j^{[m]}(k+1) &= \tilde{w}_j^{[m]}(k) \\ &+ \eta_j^{[m]}(k+1) \left( \eta e_j^{[m]}(k+1) \tilde{J}_j^{[m]}(k+1) \right. \\ &\left. + (1-\eta) \left( \tilde{w}_j^{[m]}(k) - \tilde{w}_j^{[m]}(k+1) \right) \right), \end{aligned} \quad (2.12)$$

що є модифікацією процедури Сільви-Альмейди [58].

Доцільно вдосконалити алгоритм, використовуючи підхід, запропонований у [13], тоді алгоритм (2.12) набуває слідуючих та фільтруючих властивостей. Таким чином, кінцева модифікація алгоритму набуває вигляду

$$\begin{cases} \tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{\eta e_j^{[m]}(k+1)\tilde{J}_j^{[m]}(k+1)}{r_j^{[m]}(k+1)} \\ \quad + \frac{(1-\eta)(\tilde{w}_j^{[m]}(k) - \tilde{w}_j^{[m]}(k-1))}{r_j^{[m]}(k+1)}, \\ r_j^{[m]}(k+1) = r_j^{[m]}(k) + \left\| \tilde{J}_j^{[m]}(k+1) \right\|^2 - \left\| \tilde{J}_j^{[m]}(k-s) \right\|^2, \end{cases} \quad (2.13)$$

де  $s$  – розмір «ковзного» вікна.

Цікаво, що при  $s = 1$  та  $\eta = 1$  отримуємо нелінійну версію загальновідомого алгоритму Качмажа-Уїдроу-Хоффа [36, 68]:

$$\tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{e_j^{[m]}(k+1)\tilde{J}_j^{[m]}(k+1)}{\left\| \tilde{J}_j^{[m]}(k+1) \right\|^2}, \quad (2.14)$$

який широко використовується для навчання штучних нейронних мереж і відомий високою швидкістю збіжності.

### 2.3. Навчання нео-фаззі нейронів у оптимізованій каскадній нейронній мережі

Низька швидкість навчання персепtronів Розенблатта у поєднанні з труднощами інтерпретації результатів (властиві всім ШНС в цілому) спонукає шукати альтернативні підходи до синтезу еволюційних нейронних мереж. Як зазначається у [35], нейро-фаззі системи відомі високою інтерпретовистю і прозорістю, а також високими апроксимаційними властивостями, та є основою гіbridних систем штучного інтелекту. У [15, 41] розглядаються гібридні каскадні системи штучного інтелекту побудовані на нео-фаззі нейронах [48, 73], що дозволяє їм суттєво підвищити швидкість корегування синаптичних вагових коефіцієнтів. Нео-фаззі нейрон (NFN), що його архітектуру наведено на рис. 2.3, – це нелінійна система, що реалізує нечітке висновування

$$\hat{y} = \sum_{i=1}^n f_i(x_i), \quad (2.15)$$

де  $x_i$  –  $i$ -їй вхідний сигнал ( $i = 1, 2, \dots, n$ ),

$\hat{y}$  – вихідний сигнал нео-фаззі нейрону.

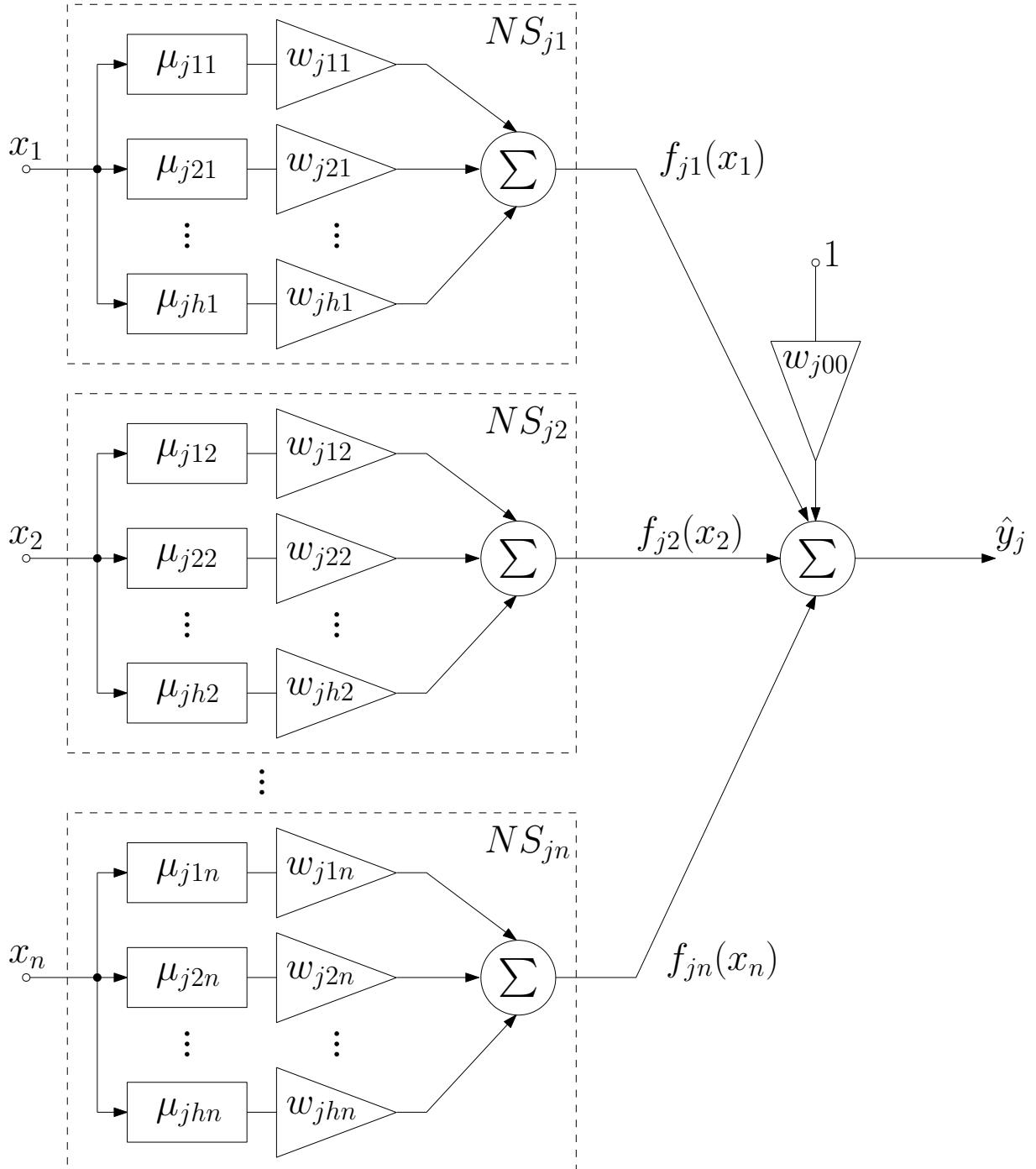


Рис. 2.3 – Архітектура нео-фаззі нейрона

Структурними елементами нео-фаззі нейрона є нелінійні синапси  $NS_i$ , які

трансформують вхідні сигнали в наступний спосіб:

$$f_i(x_i) = \sum_{l=1}^h w_{li} \mu_{li}(x_i), \quad (2.16)$$

де  $w_{li}$  –  $l$ -й ваговий коефіцієнт  $i$ -ого нелінійного синапсу,  
 $l = 1, 2, \dots, h$  – кількість синаптичних ваг, а отже і функцій належності  
 $\mu_{li}(x_i)$  у синапсі.

Таким чином, нелінійний синапс  $NS_i$  реалізує нечітке висновування

$$\text{IF } x_i \text{ IS } X_{li} \text{ THEN THE OUTPUT IS } w_{li}, \quad (2.17)$$

де  $X_{li}$  – нечітка множина з функцією належності  $\mu_{li}$ ,  
 $w_{li}$  – сінглтон (синаптичний ваговий коефіцієнт у консеквенті).  
 Тобто нелінійний синапс фактично є системою висновування Такагі-Сугено  
 нульового порядку [35].

Запишемо вихідні сигнали для нейронів першого каскаду у наступному  
 вигляді:

$$\begin{cases} \hat{y}_j^{[1]}(k) = \sum_{i=1}^n f_{ji}^{[1]}(x(k)) = \sum_{i=1}^n \sum_{l=1}^h w_{jli}^{[1]} \mu_{jli}^{[1]}(x_i(k)), \\ \text{IF } x_i \text{ IS } X_{li} \text{ THEN THE OUTPUT IS } w_{li} \end{cases} \quad (2.18)$$

$j$ -й нео-фаззі нейрон другого каскаду зображено на рис. 2.4 згідно топології нейронної мережі, зображеній на рис. 2.1).

Автори нео-фаззі нейрона [48, 73] в якості функцій належності використовували традиційні трикутні структури, які задовільняють умові розбиття Руспіні:

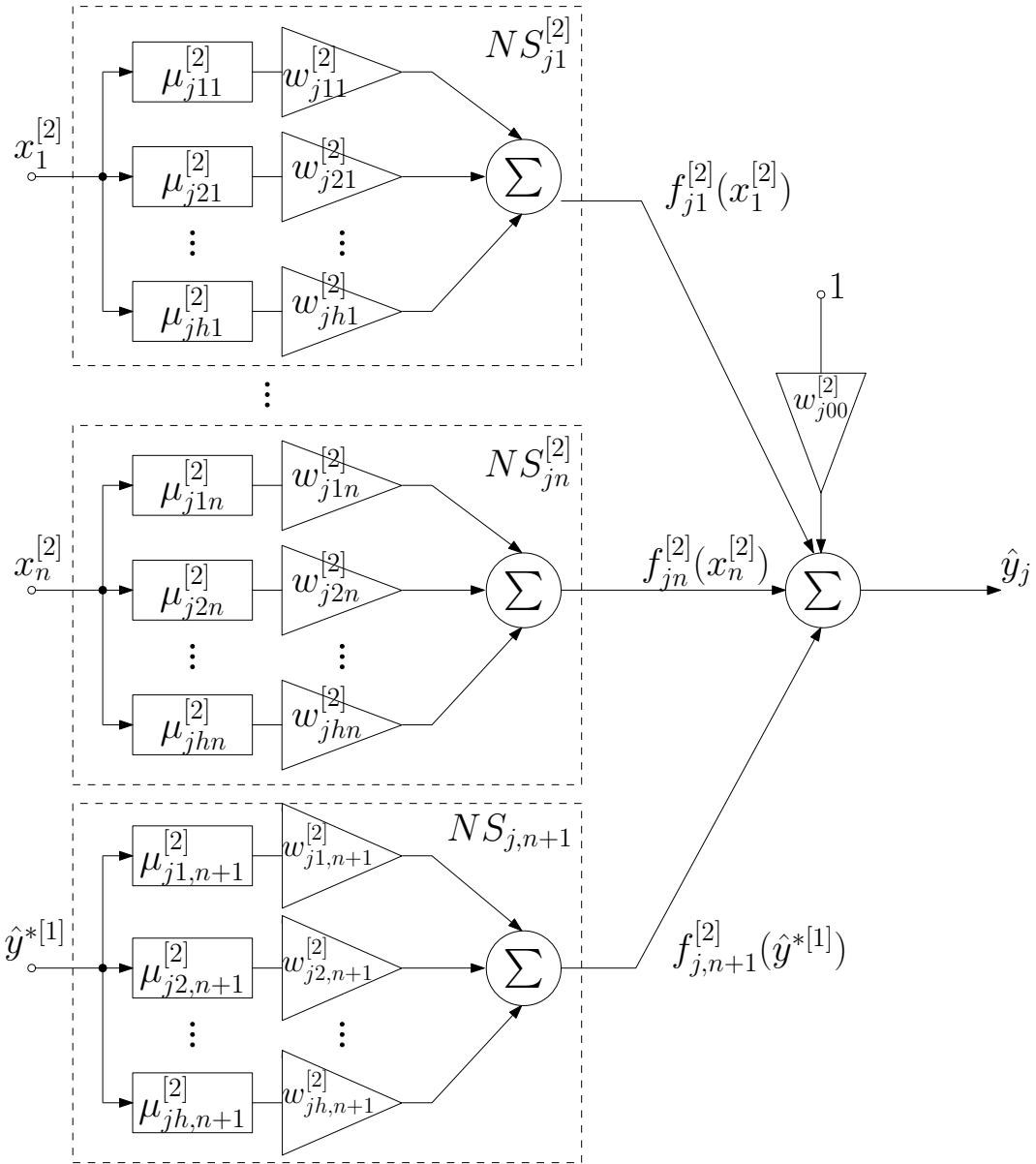


Рис. 2.4 – Нео-фаззі нейрон другого каскаду пропонованої каскадної системи

$$\mu_{jli}^{[1]}(x_i) = \begin{cases} \frac{x_i - c_{j,l-1,i}^{[1]}}{c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}} & \text{if } x_i \in [c_{j,l-1,i}^{[1]}, c_{jli}^{[1]}], \\ \frac{c_{j,l+1,i}^{[1]} - x_i}{c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}} & \text{if } x_i \in [c_{jli}^{[1]}, c_{j,l+1,i}^{[1]}], \\ 0 & \text{інакше,} \end{cases} \quad (2.19)$$

де  $c_{jli}^{[1]}$  – довільно обрані центри параметрів функцій належності на інтервалі  $[0, 1]$ , зазвичай рівномірно розподілені.

Такий вибір функцій належності гарантує, що вхідний сигнал  $x_i$  активує

лише два сусідні функції, а сума їх значень завжди дорівнюватиме 1:

$$\mu_{jli}^{[1]}(x_i) + \mu_{j,l+1,i}^{[1]}(x_i) = 1, \quad (2.20)$$

$$f_{jl}^{[1]}(x_i) = w_{jli}^{[1]} \mu_{jli}^{[1]}(x_i) + w_{j,l+1,i}^{[1]} \mu_{j,l+1,i}^{[1]}(x_i). \quad (2.21)$$

Аппроксимуючі властивості системи можна поліпшити використовуючи кубічні сплайні [15] замість трикутних функцій належності:

$$\mu_{jli}^{[1]}(x_i) = \begin{cases} \frac{1}{4} \left( 2 + 3 \frac{2x_i - c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}}{c_{jli1}^{[1]} - c_{j,l-1,i}^{[1]}} - \left( \frac{2x_i - c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}}{c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}} \right)^3 \right), \\ \text{if } x \in [c_{j,l-1,i}^{[1]}, c_{jli}^{[1]}], \\ \frac{1}{4} \left( 2 - 3 \frac{2x_i - c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}}{c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}} + \left( \frac{2x_i - c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}}{c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}} \right)^3 \right), \\ \text{if } x \in [c_{jli}^{[1]}, c_{j,l+1,i}^{[1]}], \\ 0, \text{ інакше}, \end{cases} \quad (2.22)$$

або  $B$ -сплайні [41]:

$$\mu_{jli}^{g[1]} = \begin{cases} 1, & \text{if } x_i \in [c_{jli}^{[1]}, c_{j,l+1,i}^{[1]}], \\ 0, & \text{otherwise} \end{cases} \text{ для } g = 1,$$

$$\frac{x_i - c_{jli}^{[1]}}{c_{j,l+g-1,i}^{[1]} - c_{jli}^{[1]}} \mu_{jli}^{g-1,[1]}(x_i) + \frac{c_{j,l+g,i}^{[1]} - x_i}{c_{j,l+g,i}^{[1]} - c_{j,l+g,i}^{[1]}} \mu_{j,l+1,i}^{g-1,[1]}(x_i), \quad (2.23)$$

$$\text{для } g > 1,$$

де  $\mu_{jli}^{g[1]}(x_i)$  –  $l$ -й сплайн  $g$ -ого порядку.

Нескладно помітити, що при  $g = 2$  отримуємо трикутні функції належності (2.19).

$B$ -сплайні, як і трикутні функції належності, забезпечують розбиття Руспіні, але в загальному випадку вони можуть активувати довільне число функцій належності за межами інтервалу  $[0, 1]$ , що може стати у нагоді для подальших каскадів.

Також у якості функцій належності нелінійних синапсів можна використовувати інші структури, такі, як поліноміальні, гармонійні функції, вейвлети, ортогональні функції, тощо. Проте не можна сказати наперед, які з функцій забезпечать кращі результати, тому ідея використання не одного нейрона, а пулу нейронів з різними функціями належності та активації виглядає доречною та перспективною.

За аналогією до (2.18) визначаємо вихідні сигнали інших каскадів. Так, для другого каскаду можемо записати вихідні сигнали у формі

$$\hat{y}_j^{[2]} = \sum_{i=1}^n \sum_{l=1}^h w_{jli}^{[2]} \mu_{jli}^{[2]}(x_i) + \sum_{l=1}^h w_{j,l,n+1}^{[2]} \mu_{j,l,n+1}^{[2]}(\hat{y}^{*[1]}), \quad (2.24)$$

вихідні сигнали для нейронів  $m$ -ого каскаду

$$\hat{y}_j^{[m]} = \sum_{i=1}^n \sum_{l=1}^h w_{jli}^{[m]} \mu_{jli}^{[m]}(x_i) + \sum_{p=n+1}^{n+m-1} \sum_{l=1}^h w_{jlp}^{[m]} \mu_{jlp}^{[m]}(\hat{y}^{*[p-n]}). \quad (2.25)$$

Таким чином, каскадна нейронна мережа з нео-фаззі нейронів, що сформована  $m$  каскадами, містить  $h \left( \sum_{p=1}^{m-1} p \right)$  параметрів.

Введемо вектор функцій належності для  $j$ -ого нео-фаззі нейрона  $m$ -ого каскаду

$$\begin{aligned} \mu_j^{[m]}(k) = & \left( \mu_{j11}^{[m]}(x_1(k)), \dots, \mu_{jh1}^{[m]}(x_1(k)), \mu_{j12}^{[m]}(x_2(k)), \right. \\ & \dots, \mu_{jh2}^{[m]}(x_2(k)), \dots, \mu_{jli}^{[m]}(x_i(k)), \dots, \mu_{jhn}^{[m]}(x_n(k)), \\ & \left. \dots, \mu_{j1,n+1}^{[m]}(\hat{y}^{*[1]}(k)), \dots, \mu_{jh,n+m-1}^{[m]}(\hat{y}^{*[m-1]}(k)) \right)^T \end{aligned} \quad (2.26)$$

та відповідний вектор синаптичних вагових коефіцієнтів

$$w_j^{[m]} = \begin{pmatrix} w_{j11}^{[m]}, \dots, w_{jh1}^{[m]}, w_{j12}^{[m]}, \dots, w_{jh2}^{[m]}, \dots, w_{jli}^{[m]}, \\ \dots, w_{jhn}^{[m]}, w_{j1,n+1}^{[m]}, \dots, w_{jh,n+m-1}^{[m]} \end{pmatrix}^T. \quad (2.27)$$

Тоді можемо компактно записати вихідні сигнали для  $j$ -ого нейрону  $m$ -ого каскаду

$$\hat{y}_j^{[m]}(k) = w_j^{[m]T} \mu_j^{[m]}(k). \quad (2.28)$$

У такому разі критерій навчання (2.6) приймає вигляд

$$E_j^{[m]}(k) = \frac{1}{2} (e_j^m(k))^2 = \frac{1}{2} (y(k) - w_j^{[m]T} \mu_j^{[m]}(k)), \quad (2.29)$$

а мінімізувати його можна використавши модифікацію процедури [18] для «плаваючого» вікна

$$\begin{cases} w_j^{[m]}(k+1) = w_j^{[m]}(k) + \frac{e_j^{[m]}(k+1) \mu_j^{[m]}(k+1)}{r_j^{[m]}(k+1)}, \\ r_j^{[m]}(k+1) = r_j^{[m]}(k) + \left\| \mu_j^{[m]}(k+1) \right\|^2 - \left\| \mu_j^{[m]}(k-s) \right\|^2, \end{cases} \quad (2.30)$$

або для випадку, коли  $s = 1$ ,

$$w_j^{[m]}(k+1) = w_j^{[m]}(k) + \frac{e_j^{[m]}(k+1) \mu_j^{[m]}(k+1)}{\left\| \mu_j^{[m]}(k+1) \right\|^2}, \quad (2.31)$$

що збігається з одношаговим оптимальним алгоритмом Качмажка-Уідроу-Хоффа.

Вочевидь, замість (2.30) можна скористатися іншими алгоритмами, як-от експоненційно зважений рекурентний метод найменших квадратів (EWRLSM), що використовується у DENFIS [38], ETS [3] та FLEXFIS [4, 46]. Та варто зауважити, що EWRLSM може бути нестійким при малому коефіцієнті забування.

При використанні критерія навчання з регуляризуючим параметром (momentum term) (2.6) замість (2.29) отримуємо остаточний метод навчання нео-фаззі нейрона

$$\begin{cases} w_j^{[m]}(k+1) = & w_j^{[m]}(k) + \frac{\eta e_j^{[m]}(k+1)\mu_j^{[m]}(k+1)}{r_j^{[m]}(k+1)} \\ & + \frac{(1-\eta)(w_j^{[m]}(k) - w_j^{[m]}(k-1))}{r_j^{[m]}(k+1)}, \\ r_j^{[m]}(k+1) = & r_j^{[m]}(k) + \left\| \mu_j^{[m]}(k+1) \right\|^2 - \left\| \mu_j^{[m]}(k-s) \right\|^2. \end{cases} \quad (2.32)$$

Варто зробити наголос, що оскільки вихідні сигнали нео-фаззі нейрона лінійно залежать від його синаптичних вагових коефіцієнтів, можна використовувати будь-які методи адаптивної лінійної ідентифікації [45] (наприклад, рекурентний метод найменших квадратів, робастні методи, методи, що ігнорують застарілі данні, тощо), що дозволяє обробляти нестаціонарні сигнали в онлайн режимі.

## 2.4. Розширенні нео-фаззі нейрони в якості елементів гібридної каскадної мережі, що еволюціонує

Як зазначалося вище, розглядаючи нелінійний синапс нео-фаззі нейрону з позицій нечіткої логіки, нескладно побачити, що він є вельми схожим на шар фаззіфікування таких нейро-фаззі систем як мережі Такагі-Сугено-Канґа, Дженґа, Ванг'-Менделя, і, фактично реалізує нечітке висновування

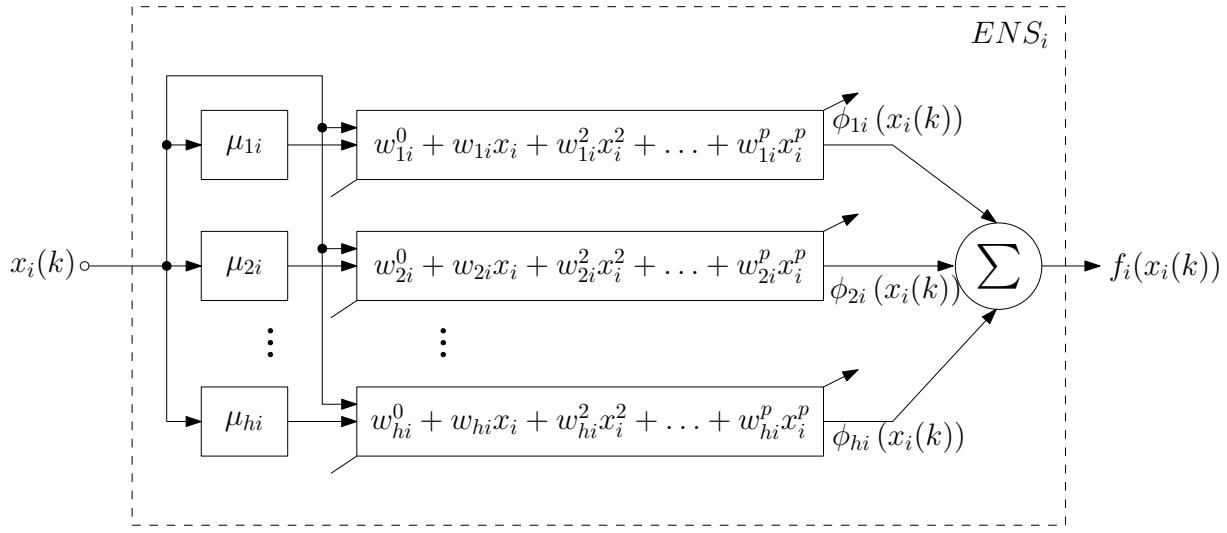


Рис. 2.5 – Синапс розширеного нео-фаззі нейрону

Такагі-Сугено нульового порядку [65, 66]. Та задля поліпшення апроксимуючих властивостей таких систем видається доцільним запропонувати удосконалений нелінійний синапс такий, що реалізує нечітке висновування довільного порядку, далі «розширений нелінійний синапс» (ENS), та зсинтезувати «розширений нео-фаззі нейрон» (ENFN), що містить такі структури замість традиційних нелінійних синапсів  $NS_i$ . Архітектури розширеного нелінійного синапсу та розширеного нео-фаззі нейрону наведено на рис. 2.5 та рис. 2.6 відповідно.

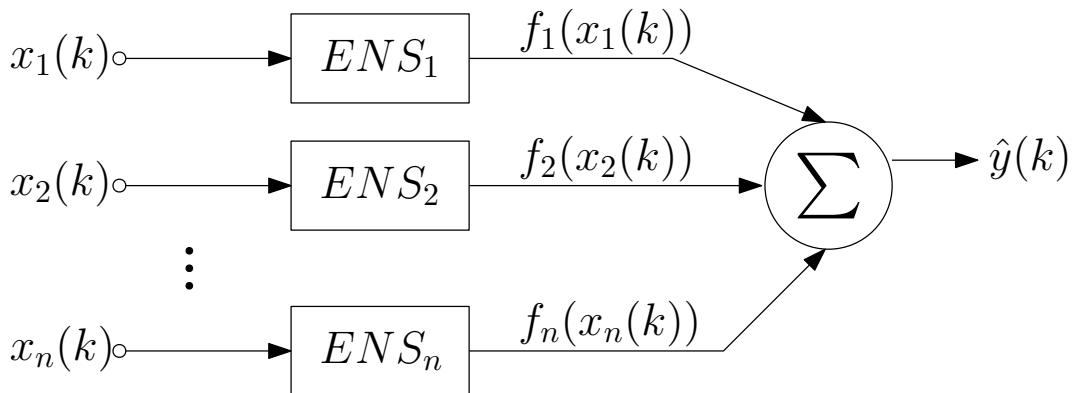


Рис. 2.6 – Розширений нео-фаззі нейрон

Вводячі нові змінні

$$\phi_{li}(x_i) = \mu_{li}(x_i)(w_{li}^0 + w_{li}^1x_i + w_{li}^2x_i^2 + \dots + w_{li}^px_i^p), \quad (2.33)$$

$$\begin{aligned}
f_i(x_i) &= \sum_{l=1}^h \mu_{li}(x_i) (w_{li}^0 + w_{li}^1 x_i + w_{li}^2 x_i^2 + \cdots + w_{li}^p x_i^p) \\
&= w_{li}^0 \mu_{li}(x_i) + w_{li}^1 x_i \mu_{1i}(x_i) + \cdots + w_{li}^p x_i^p \mu_{1i}(x_i) \\
&\quad + w_{2i}^0 \mu_{2i}(x_i) + \cdots + w_{2i}^p x_i^p \mu_{2i}(x_i) + \cdots + w_{hi}^p x_i^p \mu_{hi}(x_i),
\end{aligned} \tag{2.34}$$

$$w_i = (w_{1i}^0, w_{1i}^1, \dots, w_{1i}^p, w_{2i}^0, \dots, w_{2i}^p, \dots, w_{hi}^p)^T, \tag{2.35}$$

$$\tilde{\mu}_i(x_i) = \begin{pmatrix} \mu_{1i}(x_i), x_i(\mu_{1i}(x_i)), \dots, x_i^p(\mu_{1i}(x_i)), \\ \mu_{2i}(x_i), \dots, x_i^p \mu_{2i}(x_i), \dots, x_i^p \mu_{hi}(x_i) \end{pmatrix}^T, \tag{2.36}$$

можна представити вихідні сигнали розширеного нео-фаззі нейрона у вигляді

$$f_i(x_i) = w_i^T \tilde{\mu}_i(x_i), \tag{2.37}$$

$$\begin{aligned}
\hat{y} &= \sum_{i=1}^n f_i(x_i) \\
&= \sum_{i=1}^n w_i^T \tilde{\mu}(x_i) \\
&= \tilde{w}^T \tilde{\mu}(x).
\end{aligned} \tag{2.38}$$

де

$$\tilde{w}^T = (w_1^T, \dots, w_i^T, \dots, w_n^T)^T, \tag{2.39}$$

$$\tilde{\mu}(x) = (\tilde{\mu}_1^T(x_1), \dots, \tilde{\mu}_i^T(x_i), \dots, \tilde{\mu}_n^T(x_n))^T, \tag{2.40}$$

Таким чином, ENFN містить  $(p+1)hn$  вагових коефіцієнтів та реалізує нечітке висновування Такагі-Суг'ено  $p$ -ого порядку, а висновування, що його реалізує кожний розширений нелінійний синапс  $ENS_i$  можна записати у формі

IF  $x_i$  IS  $X_{li}$  THEN THE OUTPUT IS  
 $w_{li}^0 + w_{li}^1 x_i + \dots + w_{li}^p x_p, \quad l = 1, 2, \dots, h,$

(2.41)

що збігається з нечітким висновуванням Такагі-Сугено  $p$ -ого порядку.

Коли подати векторний сигнал  $x(k)$  на вхід  $ENFN$  першого каскаду, на виході отримуюємо скалярне значення

$$\hat{y}^{[1]}(k) = \tilde{w}^{[1]T}(k-1) \tilde{\mu}^{[1]}(x(k)), \quad (2.42)$$

що відрізняється від виразу (2.28) для звичайних  $NN$  тим, що містить у  $p+1$  більше параметрів, що корегуються.

Вочевидь, будь-які методи навчання нео-фаззі нейронів підійдуть і для розширеніх нео-фаззі нейронів. Так, вирази (2.30) та (2.31) для  $j$ -ого нейрону  $m$ -ого каскаду приймають вигляд

$$\begin{cases} \tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{e_j^{[m]}(k+1) \tilde{\mu}_j^{[m]}(k+1)}{\tilde{r}_j^{[m]}(k+1)}, \\ \tilde{r}_j^{[m]}(k+1) = \tilde{r}_j^{[m]}(k) + \left\| \tilde{\mu}_j^{[m]}(k+1) \right\|^2 - \left\| \tilde{\mu}_j^{[m]}(k-s) \right\|^2 \end{cases} \quad (2.43)$$

та

$$\tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{e_j^{[m]}(k+1) \tilde{\mu}_j^{[m]}(k+1)}{\left\| \tilde{\mu}_j^{[m]}(k+1) \right\|^2} \quad (2.44)$$

відповідно.

## 2.5. Оптимізація пулу нео-фаззі нейронів

Вихідні сигнали, згенеровані нейронами пулу кожного з каскадів, можна об'єднати у окремому вузлі-нейроні  $GN^{[m]}$ , з точністю  $\hat{y}^{*[m]}(k)$ , не меншою

від точності будь-якого нейрону пулу  $\hat{y}_j^{[m]}(k)$ . Це завдання можна вирішити за допомогою підходу ансамблей нейронних мереж. Хоча відомі алгоритми не призначені для роботи в онлайн-режимі, варто розглянути методи адаптивного узагальнюючого прогнозування [51, 62].

Введемо вектор вхідних сигналів для  $m$ -ого каскаду:

$$\hat{y}^{[m]}(k) = \left( \hat{y}_1^{[m]}(k), \hat{y}_2^{[m]}(k), \dots, \hat{y}_q^{[m]}(k) \right)^T; \quad (2.45)$$

тоді оптимальний вихідний сигнал, що його генерує нейрон  $GN^{[m]}$  (що, власне, є адаптивним лінійним асоціатором [23, 33]), можна записати у формі

$$\hat{y}^{*[m]}(k) = \sum_{j=1}^q c_j^{[m]} \hat{y}_j^{[m]}(k) = c^{[m]T} \hat{y}^{[m]}(k) \quad (2.46)$$

з обмеженнями на незміщеність

$$\sum_{j=1}^q c_j^{[m]} = E^T c^{[m]} = 1, \quad (2.47)$$

де  $c^{[m]} = \left( c_1^{[m]}, c_2^{[m]}, \dots, c_q^{[m]} \right)^T$  та  $E = (1, 1, \dots, 1)^T$  –  $(q \times 1)$ -вектори.

Введемо критерій навчання на «ковзному» вікні

$$\begin{aligned} E^{[m]}(k) &= \frac{1}{2} \sum_{\tau=k-s+1}^k (y(\tau) - \hat{y}^{*[m]}(\tau))^2 \\ &= \frac{1}{2} \sum_{\tau=k-s+1}^k (y(\tau) - c^{[m]T} \hat{y}^{[m]}(\tau))^2, \end{aligned} \quad (2.48)$$

зважаючи на обмеження (2.47), функція Лагранжа матиме вигляд

$$L^{[m]}(k) = E^{[m]}(k) - \lambda (1 - E^T c^{[m]}), \quad (2.49)$$

де  $\lambda$  – невизначений Лагранжів множник.

Мінімізуючи (2.49) відносно  $c^{[m]}$ , отримуємо

$$\begin{cases} \hat{y}^{*[m]}(k+1) = \frac{\hat{y}^{[m]T}(k+1)P^{[m]}(k+1)E}{E^TP^{[m]}(k+1)E}, \\ P^{[m]}(k+1) = \left( \sum_{\tau=k-s+2}^{k+1} \hat{y}^{[m]}(\tau)\hat{y}^{[m]T}(\tau) \right)^{-1} \end{cases} \quad (2.50)$$

або у рекурентній формі

$$\begin{cases} \tilde{P}^{[m]}(k+1) = P^{[m]}(k) - \frac{P^{[m]}(k)\hat{y}^{[m]}(k+1)\hat{y}^{[m]T}(k+1)P^{[m]}(k)}{1 + \hat{y}^{[m]T}(k+1)P^{[m]}(k)\hat{y}^{[m]}(k+1)}, \\ P^{[m]}(k+1) = \tilde{P}^{[m]}(k+1) + \\ \quad + \frac{\tilde{P}^{[m]}(k+1)\hat{y}^{[m]}(k-s+1)\hat{y}^{[m]T}(k-s+1)\tilde{P}^{[m]}(k+1)}{1 - \hat{y}^{[m]T}(k-s+1)\tilde{P}^{[m]}(k+1)\hat{y}^{[m]}(k-s+1)}, \\ \hat{y}^{*[m]}(k+1) = \frac{\hat{y}^{[m]T}(k+1)P^{[m]}(k+1)E}{E^TP^{[m]}(k+1)E}. \end{cases} \quad (2.51)$$

У випадку, коли  $s = 1$  (2.50) та (2.51) приймають доволі простий вигляд:

$$\begin{aligned} \hat{y}^{*[m]}(k+1) &= \frac{\hat{y}^{[m]T}(k+1)\hat{y}^{[m]}(k+1)}{E^T\hat{y}^{[m]}(k+1)} = \\ &= \frac{\|\hat{y}^{[m]}(k+1)\|^2}{E^T\hat{y}^{[m]}(k+1)} = \\ &= \frac{\sum_{j=1}^q (\hat{y}^{[m]}(k+1))^2}{\sum_{j=1}^q \hat{y}^{[m]}(k+1)}. \end{aligned} \quad (2.52)$$

Важливо зазначити, що навчання як нео-фаззі нейронів, так і нейронів-узагальнювачів можна організувати в онлайн-режимі. Таким чином, вагові коефіцієнти нейронів попередніх каскадів (на відміну від CasCorLA) можна не заморожувати, а постійно корегувати. Так само, число каскадів не має бути фіксованим і може змінюватись у часі, що відрізняє пропоновану нейронну мережу від інших відомих каскадних систем.

## Висновки до розділу 2

1. Розглянуті існуючі гібридні системи обчислювального інтелекту, що еволюціонують, та визначені потенційні модифікації, що їх варто привнести аби такі системи можна було застосувати у режимі послідовного надхоження даних на обробку.
2. Зсинтезована варіація каскадної системи, що еволюціонує, побудована на персептронах Розенблатта, для послідовного обробляння вхідних сигналів, що дозволило сформувати вимоги до вузлів шуканої гібридної системи.
3. Запропонована архітектура та методи навчання гібридної каскадної системи, що еволюціонує, заснованої на нео-фаззі нейронах. Пропонованій системі притаманні усі переваги нео-фаззі нейронів (інтерпритуемість та прозорість одночасно з високими апроксимаційними властивостями), а також, зрештою, вона забезпечує модель адекватної складності для кожного поставленого завдання.
4. Запропонована архітектура та методи навчання гібридної каскадної нейронної мережі, що еволюціонує, з оптимізацією пулу нейронів у кожному каскаді, що реалізують оптимальний за точністю прогноз нелінійних стохастичних і хаотичних сигналів у онлайн режимі. Варто зазначити, що оптимізація пулу нейронів дуже доречна саме у разі застосування системи для аналізу даних в онлайн режимі, адже використання узагальнюючих нейронів дозволяє визначати оптимальний нейрон на кожному етапі функціонування системи, який з високою вірогідністю може змінюватися у випадку послідовного обробляння сигналів нестаціонарних об'єктів.
5. Запропонований розширений нео-фаззі нейрон, який дозволяє реалізовувати нечітке висновування за Такагі-Сугено довільного порядку, що має покращені апроксимуючі властивості. Зсинтезована архітектура гібридної системи, що ґрунтується на розширених нео-фаззі нейро-

Hax.

## РОЗДІЛ 3

### БАГАТОВИМІРНА КАСКАДНА НЕО-ФАЗЗІ СИСТЕМА, ЩО ЕВОЛЮЦІОНУЄ

Задача апроксимації та екстраполяції багатовимірних часових рядів доволі часто виникає у багатьох технічних, медико-біологічних та інших дослідженнях, де якість прийнятих рішень істотно залежить від точності синтезованих прогнозів. У багатьох реальних задачах часові ряди характеризуються високим рівнем нелінійності та нестаціонарності своїх параметрів, наявністю аномальних викидів. Зрозуміло, що традиційні методи аналізу часових рядів, засновані на регресійному, кореляційному та інших подібних підходах, що мають на меті апріорну наявність репрезентативної вибірки спостережень, є неефективними. Альтернативою традиційним статистичним методам може слугувати математичний апарат обчислювального інтелекту, зокрема штучні нейронні мережі та нейро-фаззі-системи [23, 26, 33, 37], завдяки своїм універсальним апроксимувальним властивостям. Водночас з апроксимувальних властивостей зовсім не витікають екстраполюючі, оскільки врахування давньої передісторії для побудови прогнозувальної моделі може погіршити якість прогнозу. У зв'язку з цим під час оброблення нестаціонарних процесів треба відмовитися від процедур навчання, що базуються на зворотному поширенні помилок (багатошарові персепtronи, рекурентні нейронні мережі, адаптивні нейромережеві системи нечіткого виведення – ANFIS [69]) або методі найменших квадратів (радіально-базисні та функціонально пов'язані нейронні мережі) та скористатися процедурами на основі локальних критеріїв та «короткої» пам'яті типу алгоритма Качмажа-Уідроу-Хоффа. При цьому використані алгоритми навчання мусять забезпечувати не лише високу швидкодію, але й фільтруючі якості для придушення стохастичної «шумової» компоненти в оброблюваному сигналі. У зв'язку з цим синтез спеціалі-

зованих гібридних систем обчислювального інтелекту для розв'язання задач прогнозування істотно нестационарних часових рядів за умов невизначеності, що забезпечують разом з високою швидкістю навчання і фільтрацією завад, є досить цікавою та перспективною задачею.

Таким чином, цей розділ присвячено синтезу багатовимірної гібридної системи обчислювального інтелекту, що здатна реалізувати нелінійне відображення  $R^n \rightarrow R^g$  у режимі реального часу.

### **3.1. Багатовимірна каскадна система, що еволюціонує, побудована на нео-фаззі нейронах**

Для вирішення задачі прогнозування та ідентифікації багатовимірних даних в умовах априорної і поточної структурної та параметричної невизначеності як ніколи доречні переваги каскадно-кореляційної архітектури, адже системи з такою архітектурою успадковують всі переваги елементів, які використовуються в їх вузлах, а в процесі навчання автоматично підбирається необхідна кількість каскадів для того, щоб отримати модель адекватної складності для вирішення поставленого завдання [11, 12, 15–17, 27, 34, 41, 74]. Однак, слід зазначити, що каскадно-кореляційна мережа у формі, що її запропонували С. Фальман і К. Леб'єр [27], є системою з одним виходом, тобто не здатна реалізувати нелінійне відображення  $R^n \rightarrow R^g$ . Це досить серйозне обмеження, оскільки більшість практичних завдань містять кілька вихідних сигналів. Тож пропонуємо такі модифікації до архітектури каскадно-кореляційної мережі CasCorLA:

1. замість елементарних персепtronів Розенблата використовувати неофаззі нейрони (доцільність такого рішення було детально показано у розділі 2),
2. кількість нейронів у кожному каскаді відтепер має дорівнювати розмірності вектору вихідного сигналу системи.

Схему пропонованої архітектури наведено на рис. 3.1.

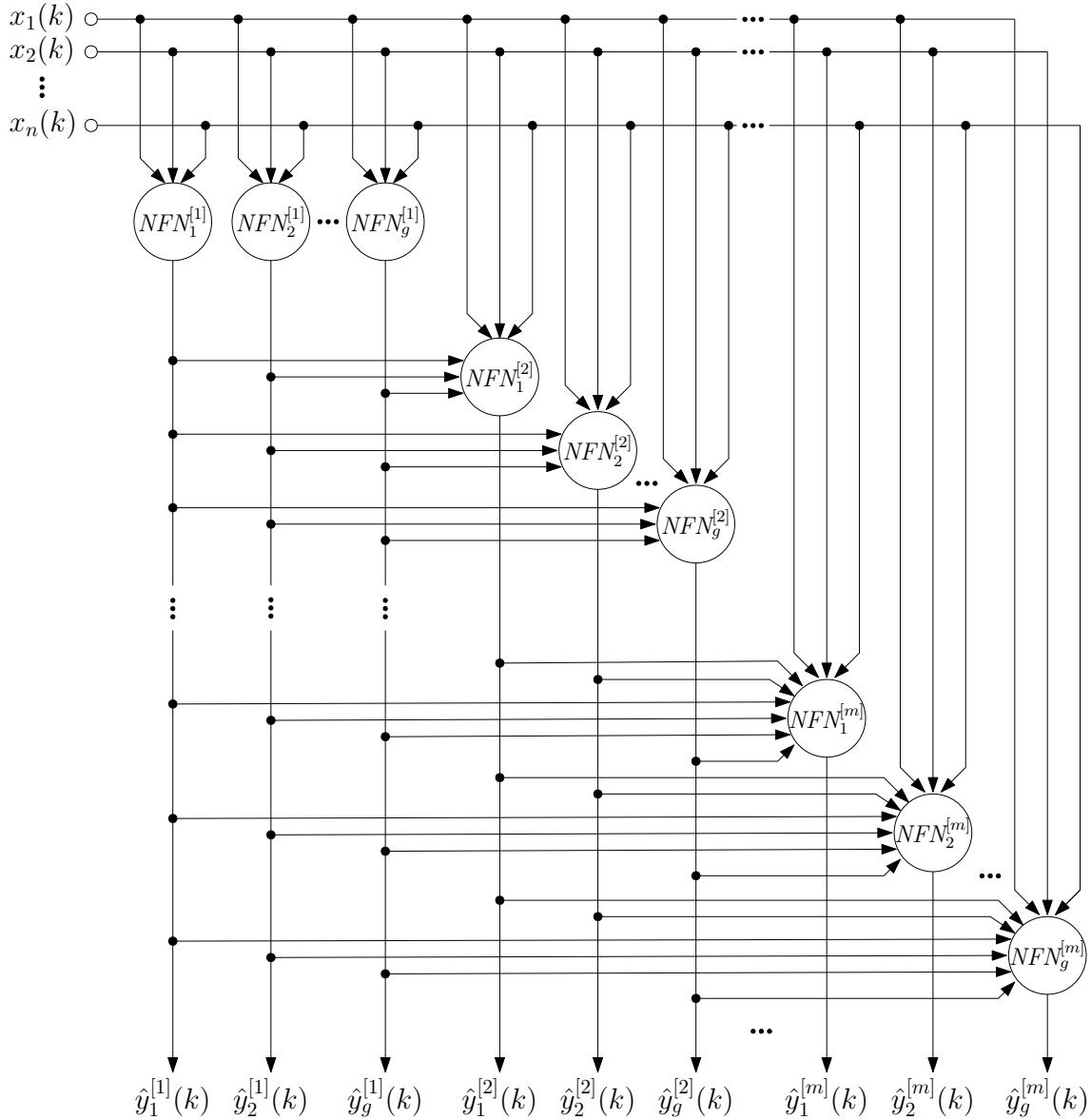


Рис. 3.1 – Архітектура гібридної МІМО системи, побудованої на нео-фаззі нейронах

Тоді вихідний сигнал системи формується з векторів, що його складають вихідні сигнали кращих нейронів останнього каскаду:

$$\hat{y}(k) = \left( \hat{y}_1^{*[m]}(k), \hat{y}_2^{*[m]}(k), \dots, \hat{y}_g^{*[m]}(k) \right)^T, \quad (3.1)$$

де  $g$  - кількість елементів вихідного вектору даних, що іх треба спрогнозувати чи ідентифікувати.

Для кожного з нео-фаззі нейронів системи в якості функцій належності мо-

жна використовувати трикутні конструкції:

$$\mu_{jli}^{[1]}(x_i) = \begin{cases} \frac{x_i - c_{d,l-1,i}^{[1]j}}{c_{dli}^{[1]j} - c_{d,l-1,i}^{[1]j}} & \text{якщо } x_i \in [c_{d,l-1,i}^{[1]j}, c_{dli}^{[1]j}], \\ \frac{c_{d,l+1,i}^{[1]j} - x_i}{c_{d,l+1,i}^{[1]j} - c_{dli}^{[1]j}} & \text{якщо } x_i \in [c_{dli}^{[1]j}, c_{d,l+1,i}^{[1]j}], \\ 0 & \text{у протилежному випадку,} \end{cases} \quad (3.2)$$

кубічні сплайни:

$$\mu_{jli}^{[1]}(x_i) = \begin{cases} \frac{1}{4} \left( 2 + 3 \frac{2x_i - c_{dli}^{[1]j} - c_{d,l-1,i}^{[1]j}}{c_{dli}^{[1]j} - c_{d,l-1,i}^{[1]j}} - \left( \frac{2x_i - c_{dli}^{[1]j} - c_{d,l-1,i}^{[1]j}}{c_{dli}^{[1]j} - c_{d,l-1,i}^{[1]j}} \right)^3 \right), \\ \text{якщо } x \in [c_{d,l-1,i}^{[1]j}, c_{dli}^{[1]j}], \\ \frac{1}{4} \left( 2 - 3 \frac{2x_i - c_{d,l+1,i}^{[1]j} - c_{dli}^{[1]j}}{c_{d,l+1,i}^{[1]j} - c_{dli}^{[1]j}} + \left( \frac{2x_i - c_{d,l+1,i}^{[1]j} - c_{dli}^{[1]j}}{c_{d,l+1,i}^{[1]j} - c_{dli}^{[1]j}} \right)^3 \right), \\ \text{якщо } x \in [c_{dli}^{[1]j}, c_{d,l+1,i}^{[1]j}], \\ 0 & \text{у протилежному випадку,} \end{cases} \quad (3.3)$$

або  $B$ -сплайни:

$$\mu_{jli}^{g[1]} = \begin{cases} 1 & \text{якщо } x_i \in [c_{dli}^{[1]j}, c_{d,l+1,i}^{[1]j}], \\ 0 & \text{у протилежному випадку} \end{cases} \quad \text{якщо } g = 1,$$

$$\frac{x_i - c_{dli}^{[1]j}}{c_{d,l+g-1,i}^{[1]j} - c_{dli}^{[1]j}} \mu_{dli}^{g-1,[1]j}(x_i) + \frac{c_{d,l+g,i}^{[1]j} - x_i}{c_{d,l+g,i}^{[1]j} - c_{d,l+g,i}^{[1]j}} \mu_{d,l+1,i}^{g-1,[1]j}(x_i), \quad \text{якщо } g > 1, \quad (3.4)$$

де  $\mu_{dli}^{g[1]j}(x_i)$  –  $l$ -й сплайн  $g$ -ого порядку. Варто зауважити, що всі ці конструкції задовільняють умовам одиничного розбиття Руспіні.

Запишемо вихідний сигнал  $j$ -ого нео-фаззі нейрону  $d$ -ого виходу першого каскаду у вигляді

$$\begin{cases} \hat{y}_d^{[1]j}(k) = \sum_{i=1}^n f_{di}^{[1]j}(x_i(k)) = \sum_{i=1}^n \sum_{l=1}^h w_{dli}^{[1]j} \mu_{dli}^{[1]j}(x_i(k)), \\ \text{ЯКШО } x_i(k) \in X_{li}^j, \text{ ТОДІ ВИХІД } w_{dli}^{[1]j}. \end{cases} \quad (3.5)$$

вихідні сигнали нео-фаззі нейронів другого каскаду:

$$\begin{aligned} \hat{y}_d^{[2]j} = & \sum_{i=1}^n \sum_{l=1}^h w_{dli}^{[2]j} \mu_{dli}^{[2]j}(x_i) + \\ & \sum_{d=1}^g \sum_{l=1}^h w_{dl,n+1}^{[2]j} \mu_{dl,n+1}^{[2]j}(\hat{y}_d^{*[1]}) \quad \forall d = 1, 2, \dots, g \end{aligned} \quad (3.6)$$

вихідні сигнали  $m$ -ого каскаду:

$$\begin{aligned} \hat{y}_d^{[2]j} = & \sum_{i=1}^n \sum_{l=1}^h w_{dli}^{[2]j} \mu_{dli}^{[2]j}(x_i) + \\ & \sum_{d=1}^g \sum_{p=n+1}^{n+m-1} \sum_{l=1}^h w_{dlp}^{[m]j} \mu_{dlp}^{[m]j}(\hat{y}_d^{*[p-n]}) \quad \forall d = 1, 2, \dots, g \end{aligned} \quad (3.7)$$

Введемо до розгляду надалі вектор функцій належності  $j$ -ого нейрону  $d$ -ого виходу  $m$ -ого каскаду:

$$\begin{aligned} \mu_d^{[m]j}(k) = & \left( \mu_{d11}^{[m]j}(x_1(k)), \dots, \mu_{dh1}^{[m]j}(x_1(k)), \mu_{d12}^{[m]j}(x_2(k)), \right. \\ & \dots, \mu_{dh2}^{[m]j}(x_2(k)), \dots, \mu_{dli}^{[m]j}(x_i(k)), \dots, \mu_{dhn}^{[m]j}(x_n(k)), \\ & \left. \dots, \mu_{d1,n+1}^{[m]j}(\hat{y}^{*[1]}(k)), \dots, \mu_{dh,n+m-1}^{[m]j}(\hat{y}^{*[m-1]}(k)) \right)^T \end{aligned} \quad (3.8)$$

та відповідний їому вектор синаптичних вагових коефіцієнтів

$$\begin{aligned} w_d^{[m]j} = & \left( w_{d11}^{[m]j}, \dots, w_{dh1}^{[m]j}, w_{d12}^{[m]j}, \dots, w_{dh2}^{[m]j}, \dots, w_{dli}^{[m]j}, \right. \\ & \left. \dots, w_{dhn}^{[m]j}, w_{d1,n+1}^{[m]j}, \dots, w_{dh,n+m-1}^{[m]j} \right)^T, \end{aligned} \quad (3.9)$$

щоб записати вихідний сигнал системи у компактній формі:

$$\hat{y}_d^{[m]j}(k) = \left( w_d^{[m]j} \right)^T \mu_d^{[m]j}(k). \quad (3.10)$$

Для навчання нео-фаззі нейронів може бути використаний будь-який з методів адаптивної ідентифікації, що ми пропонували використовувати для навчання вузлів одновимірної нео-фаззі системи у першому розділі. Так котрегулювати вагові кофіцієнти можна за допомогою експоненційно зваженого рекурентного методу найменших квадратів:

$$\begin{cases} w_d^{[m]j}(k+1) = w_d^{[m]j}(k) + \\ \frac{P_d^{[m]j}(k) \left( y^d(k+1) - \left( w_d^{[m]j}(k) \right)^T \mu_d^{[m]j}(k+1) \right)}{\alpha + \left( \mu_d^{[m]j}(k+1) \right)^T P_d^{[m]j}(k) \mu_d^{[m]j}(k+1)} \mu_d^{[m]j}(k+1), \\ P_d^{[m]j}(k+1) = \frac{1}{\alpha} \left( P_d^{[m]j}(k) - \frac{P_d^{[m]j}(k) \mu_d^{[m]j}(k+1) \left( \mu_d^{[m]j}(k+1) \right)^T P_d^{[m]j}(k)}{\alpha + \left( \mu_d^{[m]j}(k+1) \right)^T P_d^{[m]j}(k) \mu_d^{[m]j}(k+1)} \right), \end{cases} \quad (3.11)$$

де  $y^d(k+1), d = 1, 2, \dots, g$  – зовнішній навчальний сигнал,

$0 < \alpha \leq 1$  – фактор забування;

або градієнтного методу навчання, що, як зазначалося, відрізняється як згладжувальними, так і слідкуючими властивостями:

$$\begin{cases} w_d^{[m]j}(k+1) = w_d^{[m]j}(k) + \frac{y^d(k+1) - \left( w_d^{[m]j}(k) \right)^T \mu_d^{[m]j}(k+1)}{r_d^{[m]j}(k+1)} \mu_d^{[m]j}(k+1), \\ r_d^{[m]j}(k+1) = \alpha r_d^{[m]j} + \left\| \mu_d^{[m]j}(k+1) \right\|^2, 0 \leq \alpha \leq 1. \end{cases} \quad (3.12)$$

**3.1.1. Оптимізація пулу нео-фаззі нейронів багатовимірної каскадної системи, що еволюціонує.** Оскільки за мету було поставлено синтез такої багатовимірної каскадної системи, що б могла працювати саме в режимі реального часу, було б дуже доречно, якби система могла самостійно визначати найліпшу кількість функцій належності та їх форму, адже ці параметри також можуть змінюватися у часі. Тому у цьому підрозділі пропонується у кожному каскаді збільшити кількість нео-фаззі нейронів до такої, що є кратною (а не дорівнює, як пропонувалося у попередньому підрозділі) розмірності вектору вихідного сигналу та ввести узагальнюючі нейрони, що для пулу кожного каскаду визначатимуть локально оптимальні вихідні сигнали (тут під «локально оптимальним вихідними сигналами» слід розуміти сингали, оптимальні у конкретний поточний момент часу). Таким чином, коли  $g$  – розмірність вихідного векторного сигналу, а  $z$  – кількість відмінних типів нейронів (що відрізняються за кількістю чи характером функцій належності) системи, у пулі першого каскаду знаходиться  $zg$  нео-фаззі нейронів та  $g$  нейронів-узагальнювачів  $GN_d^{[1]}$ , пул другого каскаду містить  $z(g+1)$  нейронів та  $g+1$  нейронів  $GN_d^{[2]}$ , останній каскад –  $z(g+m-1)$  нейронів та  $g+m-1$  нейронів  $GN_d^{[m]}$ .

Схему такої оптимізованої MIMO (Multiple Input Multiple Output) архітектури зображено на рис. 3.1.

**3.1.1.1. Метод визначення локально оптимальних вихідних сигналів пулу нео-фаззі нейронів багатовимірної каскадної системи, що еволюціонує.** Вихідні сигнали нейронів пулу кожного каскаду пропонується об'єднати узагальнюючим нейроном  $GN^{[m]}$ , що його було введено у розділі 2.5.

Таким чином, у кожному каскаді системи маємо  $g$   $GN_d^{[m]}$  елементів, що узагальнюють вихідні сигнали нейронів пулу для кожного елементу вихідного вектору:

$$\hat{y}^{*[m]}(k) = \left( \hat{y}_1^{[m]}(k), \hat{y}_2^{[m]}(k), \dots, \hat{y}_q^{[m]}(k) \right)^T; \quad (3.13)$$

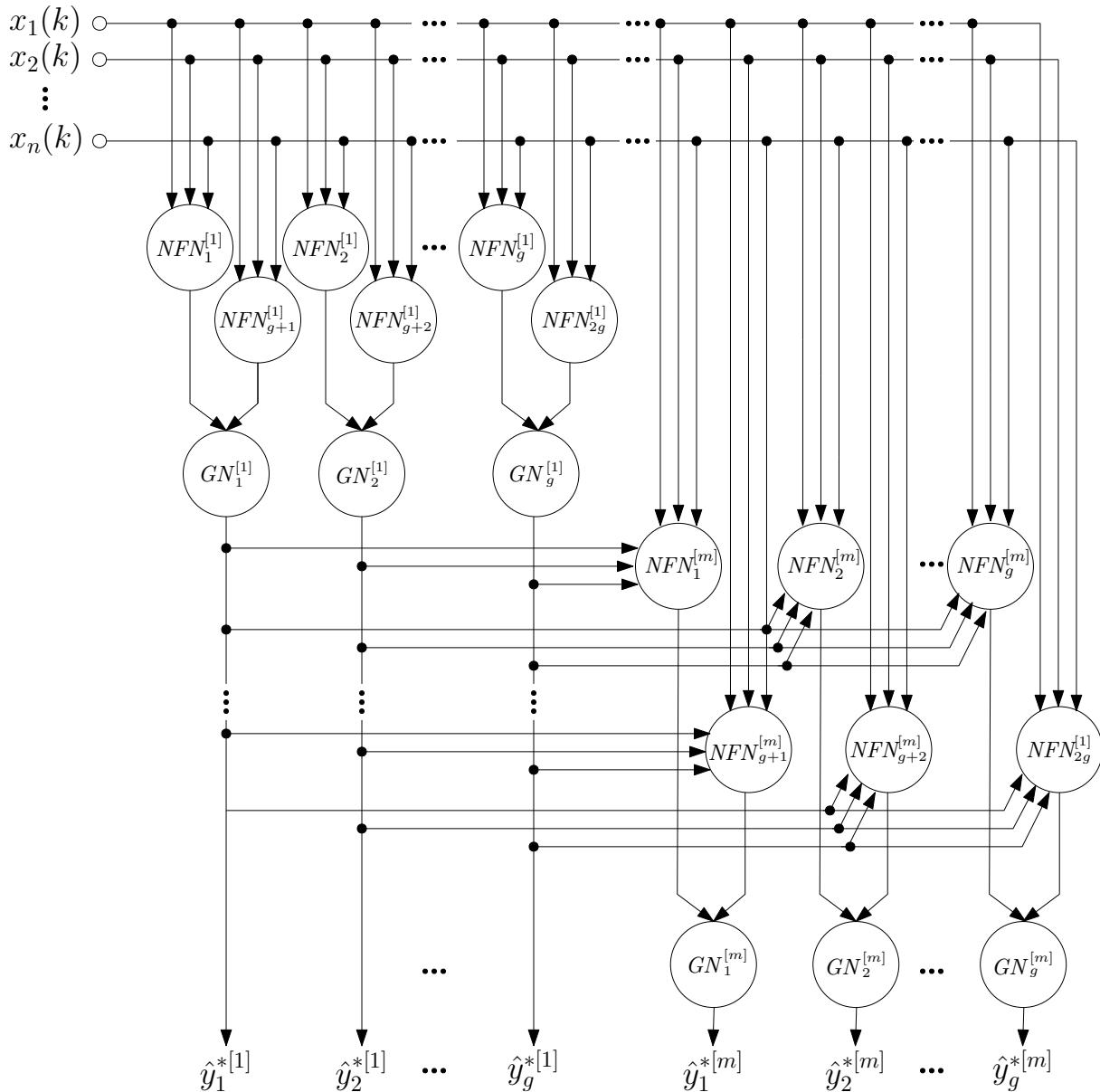


Рис. 3.2 – Архітектура гібридної оптимізованої МІМО системи, побудованої на нео-фаззі нейронах

До першого узагальнюючого елементу першого каскаду  $GN_1^{[1]}$  подаються сигнали

$$\left( \hat{y}_1^{[1]}(k), \hat{y}_{g+1}^{[1]}(k), \dots, \hat{y}_{2g+1}^{[1]}(t), \dots, \hat{y}_{(z-1)(g+1)}^{[1]}(k) \right)^T \quad (3.14)$$

до другого узагальнювача  $GN_2^{[1]}$ :

$$\left( \hat{y}_2^{[1]}(k), \hat{y}_{g+2}^{[1]}(k), \dots, \hat{y}_{2g+2}^{[1]}(t), \dots, \hat{y}_{(z-1)(g+2)}^{[1]}(k) \right)^T \quad (3.15)$$

i, нарешті, вектор вхідних сигналів останнього узагальнюючого елементу першого каскаду  $GN_g^{[1]}$ :

$$\left( \hat{y}_g^{[1]}(k), \hat{y}_{2g}^{[1]}(k), \dots, \hat{y}_{(z-1)g}^{[1]}(k) \right)^T. \quad (3.16)$$

Нагадаємо, що точність вихідного сигналу узагальнюючих елементів має бути не гіршою точності будь-якого сингналу, що узагальнюється (подається на вхід до  $GN_d^{[m]}$ ). Рекурентна форма методу навчання «на ковзному вікні» елементів  $GN_d^{[m]}$  кожного каскаду має вигляд

$$\begin{cases} \tilde{P}_d^{[m]}(k+1) = P_d^{[m]}(k) - \frac{P_d^{[m]}(k)\hat{y}_d^{[m]}(k+1)\hat{y}_d^{[m]T}(k+1)P_d^{[m]}(k)}{1 + \hat{y}_d^{[m]T}(k+1)P_d^{[m]}(k)\hat{y}_d^{[m]}(k+1)}, \\ P_d^{[m]}(k+1) = \tilde{P}_d^{[m]}(k+1) + \\ \quad \frac{\tilde{P}_d^{[m]}(k+1)\hat{y}_d(k-s+1)\hat{y}_d^{[m]T}(k-s+1)\tilde{P}_d^{[m]}(k+1)}{1 - \hat{y}_d^{[m]T}(k-s+1)\tilde{P}_d^{[m]}(k+1)\hat{y}_d^{[m]}(k-s+1)}, \\ \hat{y}_d^{*[m]}(k+1) = \frac{\hat{y}_d^{[m]T}(k+1)P_d^{[m]}(k+1)E}{E^T P_d^{[m]}(k+1)E}, \end{cases} \quad (3.17)$$

а у випадку, коли  $s = 1$ :

$$\begin{aligned} \hat{y}_d^{*[m]}(k+1) &= \frac{\hat{y}_d^{[m]T}(k+1)\hat{y}_d^{[m]}(k+1)}{E^T \hat{y}_d^{[m]}(k+1)} \\ &= \frac{\|\hat{y}_d^{[m]}(k+1)\|^2}{E^T \hat{y}_d^{[m]}(k+1)} \\ &= \frac{\sum_{j=1}^q (\hat{y}_d^{[m]}(k+1))^2}{\sum_{j=1}^q \hat{y}_d^{[m]}(k+1)}. \end{aligned} \quad (3.18)$$

### 3.2. Багатовимірна каскадна система, що еволюціонує, побудована на багатовимірних нео-фаззі нейронах

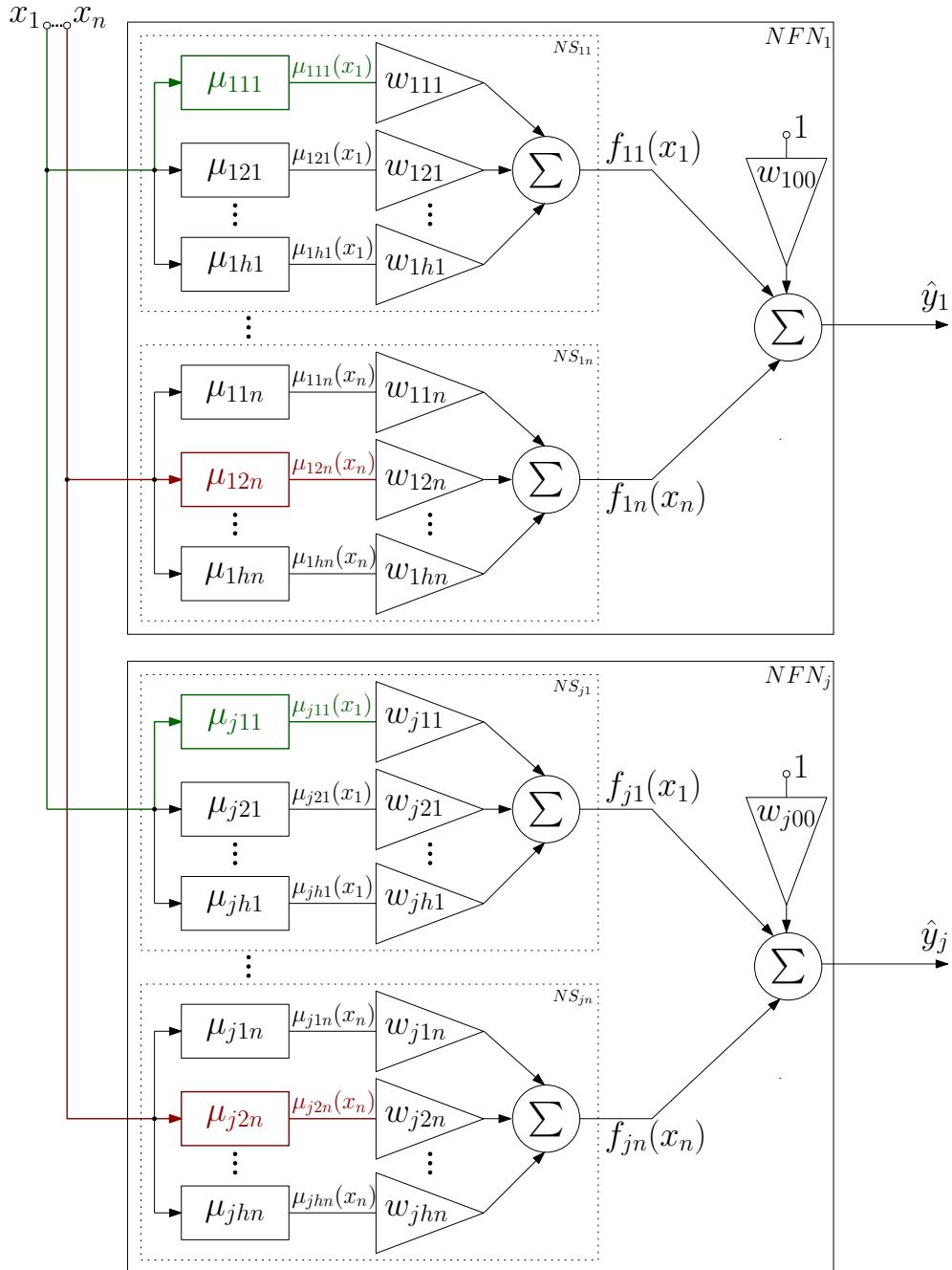


Рис. 3.3 – Ілюстрація надмірності МІМО системи, побудованої на нео-фаззі нейронах

Архітектура багатовимірної каскадної системи, яка ґрунтуються на звичайних нео-фаззі нейронах, що її описано у підрозділі 3.1, є надмірною, адже

вектор вхідних сигналів  $x(k)$  (для першого каскаду) подається на однотипні нелінійні синапси  $NS_{di}^{[1]j}$  нео-фаззі нейронів, кожен з яких на виході генерує сигнал  $\hat{y}_d^{[1]j}(k), d = 1, 2, \dots, g$ . У результаті компоненти вихідного вектора

$$\hat{y}^{[1]j}(k) = (\hat{y}_1^{[1]j}(k), \hat{y}_2^{[1]j}(k), \dots, \hat{y}_g^{[1]j}(k))^T \quad (3.19)$$

обчислюються незалежно один від одного, хоча при цьому

$$\mu_{1il}(x_i(k)) = \mu_{2il}(x_i(k)) = \mu_{jil}(x_i(k)) = \mu_{nil}(x_i(k)). \quad (3.20)$$

Надмірність архітектури, що її наведено на рис. 3.1, проілюстрована на рис. 3.3, де зеленим кольором позначені неодноразово обчислювані тотожні значення функцій належності  $\mu_{111}$  та  $\mu_{j11}$ , червоним кольором – тотожні  $\mu_{12n}$  та  $\mu_{j2n}$ . Уникнути цього можна, якщо ввести до розгляду багатовимірний нео-фаззі нейрон, що є модифікацією систем, запропонованих у [15, 48].

**3.2.1. Багатовимірний нео-фаззі нейрон.** Вузлами багатовимірного нео-фаззі нейрону MNFN (схема наведена на рис. 3.4) є складені нелінійні синапси  $MNS_i^{[1]j}$ , кожен з яких містить  $h$  функцій належності  $\mu_{li}^{[1]j}$  та  $gh$  настроюваних синаптичних вагових коефіцієнтів, але тільки  $hn$  функцій належності, що в  $g$  разів менше, ніж у випадку, коли каскад сформований із звичайних нео-фаззі нейронів.

Введемо надалі до розгляду  $(hn \times 1)$  - вектор функцій належності

$$\mu^{[1]j}(k) = \left( \mu_{11}^{[1]j}(x_1(k)), \mu_{21}^{[1]j}(x_1(k)), \dots, \mu_{h1}^{[1]j}(x_1(k)), \dots, \mu_{hn}^{[1]j}(x_n(k)) \right)^T \quad (3.21)$$

та  $(g \times hn)$  - матрицю синаптичних вагових коефіцієнтів

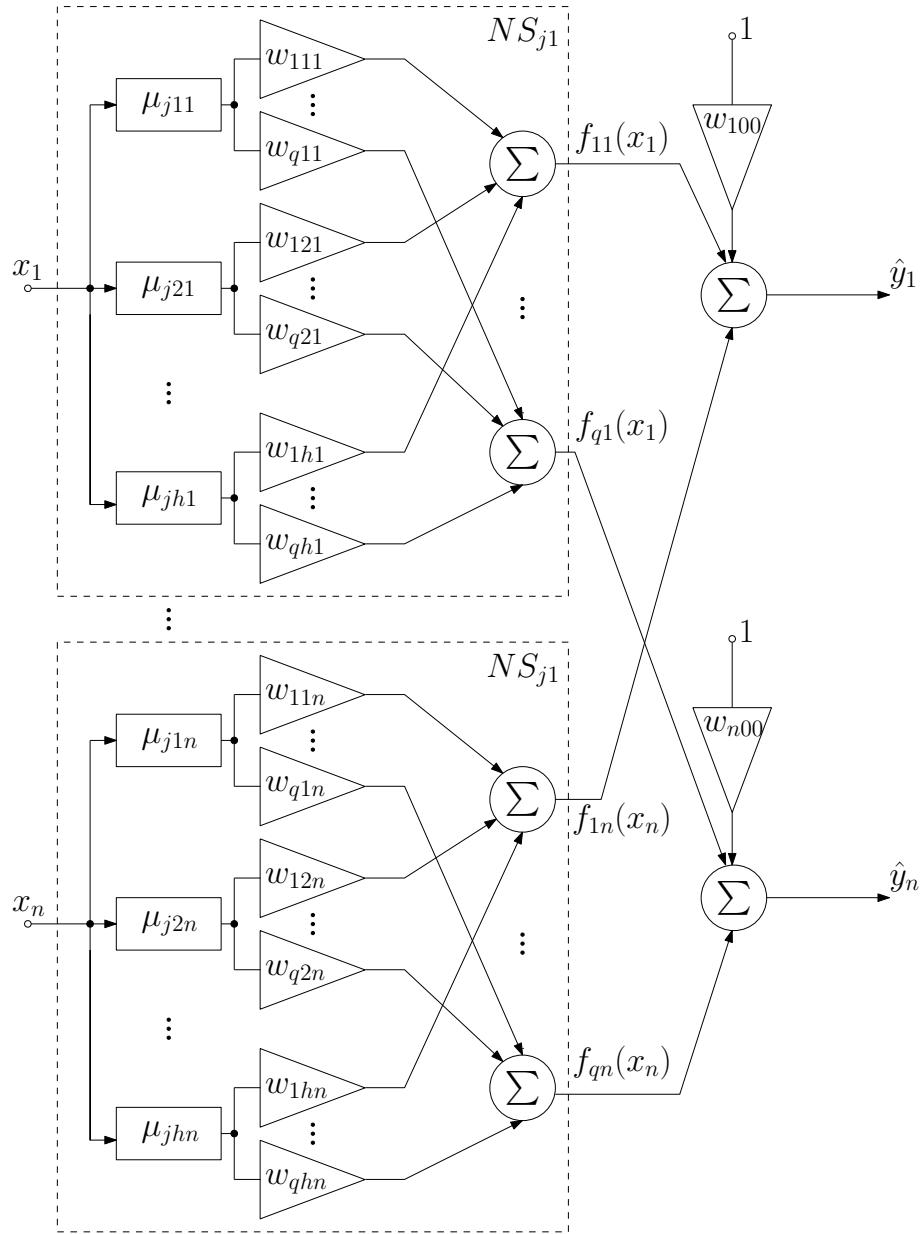


Рис. 3.4 – Багатовимірний нео-фаззі нейрон

$$W^{[1]j} = \begin{pmatrix} w_{111}^{[1]j} & w_{112}^{[1]j} & \dots & w_{1li}^{[1]j} & \dots & w_{1hn}^{[1]j} \\ w_{211}^{[1]j} & w_{212}^{[1]j} & \dots & w_{2li}^{[1]j} & \dots & w_{2hn}^{[1]j} \\ \vdots & \vdots & & \vdots & & \vdots \\ w_{g11}^{[1]j} & w_{g12}^{[1]j} & \dots & w_{gli}^{[1]j} & \dots & w_{ghn}^{[1]j} \end{pmatrix} \quad (3.22)$$

і запишемо сигнал на виході  $MN_j^{[1]}$  у  $k$ -й момент часу у вигляді

$$\hat{y}^{[1]j}(k) = W^{[1]j} \mu^{[1]j}(k). \quad (3.23)$$

Навчання багатовимірного нео-фаззі нейрону можна реалізувати за допомогою матричної модифікації експоненційно-зваженого рекурентного методу найменших квадратів (3.11) у формі

$$\begin{cases} W^{[1]j}(k+1) = W^{[1]j}(k) + \\ \quad + \frac{(y(k+1) - W^{[1]j}(k)\mu^{[1]j}(k+1))(\mu^{[1]j}(k+1))^T P^{[1]j}(k)}{\alpha + (\mu^{[1]j}(k+1))^T P^{[1]j}(k)\mu^{[1]j}(k+1)}, \\ P^{[1]j}(k+1) = \frac{1}{\alpha} \left( P^{[1]j}(k) - \frac{P^{[1]j}(k)\mu^{[1]j}(k+1)(\mu^{[1]j}(k+1))^T P^{[1]j}(k)}{\alpha + (\mu^{[1]j}(k+1))^T P^{[1]j}(k)\mu^{[1]j}(k+1)} \right), \\ 0 < \alpha \leq 1 \end{cases} \quad (3.24)$$

або багатовимірного варіанту методу (3.12)

$$\begin{cases} W^{[1]j}(k+1) = W^{[1]j}(k) + \frac{y(k+1) - W^{[1]j}(k)\mu^{[1]j}(k+1)}{r^{[1]j}(k+1)} \times \\ \quad \times (\mu^{[1]j}(k+1))^T, \\ r^{[1]j}(k+1) = \alpha r^{[1]j}(k) + \|\mu^{[1]j}(k+1)\|^2, \\ 0 \leq \alpha \leq 1, \end{cases} \quad (3.25)$$

де  $y(k+1) = (y^1(k+1), y^2(k+1), \dots, y^g(k+1))^T$ .

Аналогічним чином проводиться навчання інших каскадів, при цьому вектор функцій належності  $m$ -го каскаду  $\mu^{[m]j}(k+1)$  збільшує свою розмірність на  $(m-1)g$  компоненти, що їх утворили виходи попередніх каскадів.

**3.2.2. Метод визначення локально оптимального вихідного сигналу пулу багатовимірних нео-фаззі нейронів каскадної системи, що еволюціонує.** У цьому підрозділі запропоновано узагальнюючий ней-

рон  $GMN^{[m]}$  та рекурентний метод його навчання, щоб він об'єднував усі вихідні сигнали нейронів  $MNFN^{[m]}$  пулу каскаду у вихідний сигнал

$$\hat{y}^{*[m]}(k) = \left( \hat{y}_1^{*[m]}(k), \hat{y}_2^{*[m]}(k), \dots, \hat{y}_g^{*[m]}(k) \right)^T \quad (3.26)$$

з точністю не меншою від точності будь-якого з сигналів  $\hat{y}_j^{[m]}(k)$ .

Розв'язати це завдання можна, знову скориставшись апаратом невизначених множників Лагранжа та адаптивного багатовимірного узагальненого прогнозування [48].

Введемо до розгляду вихідний сигнал нейрону  $GMN^{[m]}$  у вигляді

$$\hat{y}^{*[m]}(k) = \sum_{j=1}^q c_j^{[m]} \hat{y}_j^{[m]}(k) = \hat{y}^{[m]}(k) c^{[m]}, \quad (3.27)$$

де  $\hat{y}^{[m]}(k) = \left( \hat{y}_1^{[m]}(k), \hat{y}_2^{[m]}(k), \dots, \hat{y}_q^{[m]}(k) \right)^T$  –  $(g \times q)$ -матриця  $c^{[m]}$  –  $(q \times 1)$ -вектор коефіцієнтів узагальнення, що відповідають умовам незміщеності

$$\sum_{j=1}^q c_j^{[m]} = E^T c^{[m]} = 1, \quad (3.28)$$

$E = (1, 1, \dots, 1)^T$  – вектор, утворений одиницями.

Введемо критерій навчання

$$\begin{aligned} E^{[m]}(k) &= \sum_{\tau=1}^k \|y(\tau) - \hat{y}^{[m]}(\tau) c^{[m]}\|^2 \\ &= Tr \left( (Y(k) - \hat{Y}^{[m]}(k) I \otimes c^{[m]})^T (Y(k) - \hat{Y}^{[m]}(k) I \otimes c^{[m]}) \right) \end{aligned} \quad (3.29)$$

де  $Y(k) = (y^T(1), y^T(2), \dots, y^T(k))^T$  –  $(k \times s)$  матриця спостережень,

$$\hat{Y}^{[m]}(k) = \begin{pmatrix} \hat{y}_1^{[m]T}(1) & \hat{y}_2^{[m]T}(1) & \dots & \hat{y}_q^{[m]T}(1) \\ \hat{y}_1^{[m]T}(2) & \hat{y}_2^{[m]T}(2) & \dots & \hat{y}_q^{[m]T}(2) \\ \vdots & \vdots & & \vdots \\ \hat{y}_1^{[m]T}(k) & \hat{y}_2^{[m]T}(k) & \dots & \hat{y}_q^{[m]T}(k) \end{pmatrix}, \quad (3.30)$$

$I$  – одинична  $(g \times g)$  матриця,

$\otimes$  – символ тензорного добутку.

З урахуванням обмежень (3.28) запишемо функцію Лагранжа

$$\begin{aligned} L^{[m]}(k) &= E^{[m]}(k) + \lambda(E^T c^{[m]} - 1) \\ &= \sum_{\tau=1}^k \|y(\tau) - \hat{y}^{[m]}(\tau)c^{[m]}\|^2 + \lambda(E^T c^{[m]} - 1) \\ &= Tr\left(\left(Y(k) - \hat{Y}^{[m]}(k)I \otimes c^{[m]}\right)^T \left(Y(k) - \hat{Y}^{[m]}(k)I \otimes c^{[m]}\right)\right) + \lambda(E^T c^{[m]} - 1) \\ &\quad + \lambda(E^T c^{[m]} - 1) \\ &= Tr\left(V^{[m]T}(k)V^{[m]}(k)\right) + \lambda(E^T c^{[m]} - 1), \end{aligned} \quad (3.31)$$

де  $V^{[m]}(k) = Y(k) - \hat{Y}^{[m]}(k)I \otimes c^{[m]}$  –  $(k \times g)$  матриця оновлень.

Розв'язання системи рівнянь Каруша-Куна-Таккера

$$\begin{cases} \nabla_{c^{[m]}} L^{[m]}(k) = \vec{0}, \\ \frac{\partial L^{[m]}(k)}{\partial \lambda} = 0 \end{cases} \quad (3.32)$$

призводить до очевидного результату

$$\begin{cases} c^{[m]} = \left(R^{[m]}(k)\right)^{-1} E \left(E^T \left(R^{[m]}(k)\right)^{-1}\right)^{-1} \\ \lambda = -2E^T \left(R^{[m]}(k)\right)^{-1} E, \end{cases} \quad (3.33)$$

де  $R^{[m]}(k) = V^{[m]T}(k)V^{[m]}(k)$ .

Таким чином, можна організувати оптимальне об'єднання виходів усіх нейронів пулу кожного каскаду. Зрозуміло, що в якості таких нейронів можуть використовуватися не тільки багатовимірні нео-фаззі нейрони, але й будь-які інші конструкції, що реалізують нелінійне відображення  $R^{n+(m-1)g} \rightarrow R^g$ .

## Висновки до розділу 3.1

1. Розглянута задача апроксимації та екстраполяції багатовимірних часових рядів за умови апріорної і поточної структурної та параметричної невизначеності; проаналізовані існуючі гібридні системи обчислювального інтелекту, що використовуються для вирішення задач прогнозування та ідентифікації багатовимірних даних у пакетному режимі; сформовані вимоги та обмеження до шуканої гібридної системи, здатної реалізувати нелінійне відображення  $R^n \rightarrow R^g$  у режимі реального часу.
2. Зсинтезовано каскадну архітектуру системи, що ґрунтуються на неофаззі нейронах, здатну реалізувати нелінійне відображення  $R^n \rightarrow R^g$  у режимі послідовного обробляння даних.
3. Запропоновано архітектуру багатовимірного нео-фаззі нейрона та метод його навчання, що забезпечують підвищену швидкість налаштування синаптичних ваг та додаткові згладжуючі властивості.
4. Запропоновано архітектуру та рекурентний метод навчання багатовимірного узагальнюючого елементу, що в режимі реального часу реалізує оптимальне об'єднання багатовимірних вихідних сигналів нейронів пулу каскаду.
5. Запропоновано МІМО архітектуру та методи навчання гібридної каскадної нейронної мережі з оптимізацією пулу багатовимірних нейронів у кожному каскаді, що реалізують оптимальний за точністю прогноз нелінійних стохастичних і хаотичних сигналів у онлайн режимі.

## РОЗДІЛ 4

# **КАСКАДНА НЕЙРОННА МЕРЕЖА, ЩО ЕВОЛЮЦІОНУЄ, ДЛЯ ПОСЛІДОВНОГО НЕЧІТКОГО КЛАСТЕРУВАННЯ ПОТОКІВ ДАНИХ**

У цьому розділі описані архітектура та методи навчання пропонованої каскадної нейро-мережі для нечіткого кластерування, зокрема потоків даних; проведено аналіз існуючих систем, що еволюціонують, для кластерування даних, зокрема нечіткого, і розглянуті особливості та труднощі послідовного кластерування, та описні два підходи, переваги яких поєднує у собі пропонована система: нечітке та ієрархічне кластерування.

### **4.1. Труднощі та особливості відомих методів кластерування даніх**

Завдання кластерування (класифікації без вчителя) досить часто зустрічається в багатьох додатках, пов'язаних з видобутком знань, де у режимі самонавчання необхідно розбити деякий вхідний нерозмічений масив даних на однорідні в прийнятому сенсі групи. Розглянемо деякі ієрархічні та розподільні методи кластерування, адже, як буде показано далі, пропована у цьому розділі самонавчанна система поєднує у собі переваги обох підходів.

Розподільні методи кластерування (чи то жорсткі, чи нечіткі) можна назвати динамічними у тому сенсі, що належність певного образу до певного кластеру (кластерів для нечіткої модифікації) не є постійною. Нездатність методів розподільного кластерування самостійно визначити кількість кластерів у певному сенсі компенсується тим, що знання форми чи розміру кластерів може стати у нагоді на етапі вибору відповідних прототипів та насамперед типу відстані (міри схожості) і суттєво поліпшити кінцеве розбиття вибірки. Але, варто зазначити чутливість таких методів до початкової ініціалізації,

шуму і викидів, їх сприйнятливість до локальних мінімумів, адже вони ґрунтуються на оптимізації певної цільової функції. Типові методи розподільного кластерування мають обчислювальну складність  $\mathcal{O}(N)$  для тренувальною вибірки розміру  $N$  [26].

Серед методів ієрархічного кластерування виділяють два основних типи: висхідні та спадні методи. Спадні методи працюють за принципом «зверху-вниз»: на початку припускається, що всі образи належать до одного кластеру, який потім розбивається на все більш дрібні кластери. Більш поширеними є висхідні алгоритми, які на початку роботи поміщають кожен об'єкт до окремого кластеру, а потім об'єднують кластери у все більш крупні, доки усі образи не матимуть свій власний кластер. Таким чином будується система вкладених розбиттів. Результати таких алгоритмів зазвичай представляють у вигляді дерева - дендрограми (тут можна провести аналогію між висхідними та спадними методами ієрархічного кластерування та конструктивними і деструктивними системами, що еволюціонують. У цій роботі здебільшого розглядається конструктивний підхід, тому пропонована самонавчанна система є у певному сенсі альтернативою системам висхідного ієрархічного кластерування, що здатна працювати у режимі реального часу).

Для обчислення відстаней між кластерами використовуються такі відстані:

- одинарний зв'язок (відстань найближчого сусіда): відстань між двома кластерами визначається відстанню між двома найбільш близькими об'єктами (найближчими сусідами) у різних кластерах. Результатуючі кластери мають тенденцію об'єднуватися в ланцюжки.
- повний зв'язок (відстань найбільш віддалених сусідів): відстані між кластерами визначаються найбільшою відстанню між будь-якими двома об'єктами різних кластерів (тобто найбільш віддаленими сусідами). Цей метод зазвичай працює дуже добре, коли об'єкти походять з окремих груп. Якщо ж кластери мають видовжену форму або їх природний тип є «ланцюжковим» цей метод непридатний.

- незважене попарне середнє: відстань між двома різними кластерами обчислюється як середня відстань між усіма парами об'єктів у них. Метод ефективний, коли об'єкти формують різні групи, проте він працює однаково добре і у випадках протяжних («ланцюжкового» типу) кластерів.
- зважене попарне середнє: метод ідентичний методу незваженого попарного середнього, за винятком того, що при обчисленнях розмір відповідних кластерів (тобто число об'єктів, що містяться в них) використовується у якості вагового коефіцієнту. Тому доцільно використовувати даний метод у випадку нерівних за розміром кластерів.
- незважений центройдний метод: у цьому методі відстань між двома кластерами визначається як відстань між їх центрами тяжкості.
- зважений центройдний метод (медіана): цей метод ідентичний попередньому, за винятком того, що при обчисленнях використовуються ваги для обліку різниці між розмірами кластерів. Тому, якщо є або підозрюються значні відмінності в розмірах кластерів, цей метод має перевагу над попереднім

Порівняно з розподільним кластеруванням, методи ієрархічного кластерування легко ідентифікують викиди, не потребують визначеної кількості кластерів та нечутливі до початкової ініціалізації чи локальних мінімумів. До недоліків варто віднести нездатність методів визначати кластери, що перекривають один інший. Крім того, ієрархічне кластерування є статичним, тобто образи віднесені до певного кластеру на ранніх стадіях не можуть бути пізніше належними іншому, що унеможливило створення модифікацій методів для послідовного кластерування, на відміну від розподільного кластерування. Методи ієрархічного кластерування здебільшого мають обчислювальну складність принаймні  $\mathcal{O}(N^2)$ , що робить їх використання недоцільним для великих наборів даних.

#### 4.1.1. Нечітке послідовне кластерування.

Традиційний підхід до завдання кластерування припускає, що кожне спостереження належить лише одному кластерові, в той час як більш природною видається ситуація, коли кожен вектор-спостереження оброблюваної вибірки можна віднести відразу декільком класам з різними рівнями належності. Така ситуація є предметом розгляду нечіткого кластерного аналізу [1, 8, 28, 30, 50, 72], а для його вирішення широко використовується апарат обчислювального інтелекту [9-12] і, насамперед, нейро-фаззі підхід [13]. При цьому більшість алгоритмів нечіткої кластеризації призначені для роботи в пакетному режимі, коли усі дані, що підлягають обробці, задані апріорно. Вихідною інформацією для такої задачі є вибірка спостережень, сформована з  $t$ -вимірних векторів ознак  $x(1), x(2), \dots, x(1), ;x(N)$ , при цьому для зручності чисельної реалізації вихідні дані попередньо деяким чином перетворюються, наприклад, так, щоб всі спостереження належали до гіперкубу  $[-1, 1]^n$  або одиничній гіперсфері  $\|x(k)\|^2$ .

Результатом такого кластерування є розбиття масиву вихідних даних на  $M$  кластерів з певним рівнем належності  $u_{Jk}(k)$   $k$ -ого вхідного образу  $x(k)$  до  $J$ -ого кластеру ( $J = 1, 2, \dots, M$ ). Передбачається, що  $N$  та  $M$ , а також параметри кластерування (в першу чергу, фаззіфікатор) задані апріорі і не змінюються під час обробки даних. Варто зауважити, що існує широкий клас задач динамічного інтелектуального аналізу даних і потоків даних (Dynamik Data Mining, Data Stream Mining) [2, 6, 20, 32, 44, 57, 70] у випадку, коли дані надходять у вигляді послідовного потоку в онлайн режимі. Отже, кількість вхідних образів  $N$  у цьому випадку не обмежується, а  $k$  набуває значення поточного дискретного часу.

Самоорганізовні мапи Кохонена [40] добре пристосовані для вирішення завдання кластерування в онлайн режимі. Ці нейронні мережі мають один шар латеральних з'єднань та навчаються за принципами «переможець отримує все» або «переможець отримує більше». Самоорганізовні мапи також відомі своєю ефективністю вирішення задачі кластерування класів, що перети-

наються. Тому, у зв'язку з дедалі більшою кількістю завдань кластерування потоків даних, з'явилися самонавчанні нейро-фазі гібридні системи, що у деякому сенсі поєднують у собі самоорганізовні мапи Кохонена (SOM) та метод нечітких  $c$ -середніх Бездека [7, 9, 22, 24, 31, 43, 53, 54, 59, 60, 63, 64]. Такі гібридні системи володіють обширною функціональністю завдяки використанню спеціальних алгоритмів налаштування, що ґрунтуються на процедурах оптимізації прийнятої цільової функції, але потребують попередньо заданої кількості класетрів та фіксованого значення фаззіфікатору.

## 4.2. Критерії дійсності нечіткого кластерування

Оскільки коефіцієнт розбиття залежить лише від значень функції належності, йому властиві деякі недоліки. Коли фаззіфікатор наближається до 1, індекс дійсності буде однаковим для усіх  $c$ , коли фаззіфікатор наближається до  $\infty$ .

Індекс розбиття ентропії PE (Partition Entropy Index) - ще один критерій дійсності нечіткого кластерування, запропонований (Bezdek, 1974a, 1981), що залежить лише від значень функції належності

$$PE = -\frac{1}{N} \sum_{l=1}^M \sum_{i=1}^N u_{li} \log_a(u_{li}). \quad (4.1)$$

Індекс ентропії розбиття набуває значень у інтервалі  $[0, \log_a M]$ . Що більше значення  $PE$  до 0, то жорсткіше розбиття вхідних даних. Значення  $PE$  близькі до верхньої межі вказують на відсутність будь-якої структури, притаманної набору вхідних даних, або на нездатність методу її виявити. Індекс ентропії розбиття має ті самі недоліки, що і коефіцієнт розбиття. Оптимальній кількості класетрів  $M^*$  відповідає мінімальне значення (4.1).

Фукуяма та Сугено запропонували індекс дійсності нечіткого кластерування, залежний як від рівнів належності так і від самих вхідних даних:

$$FS = \sum_{i=1}^N \sum_{l=1}^M u_{li}^\beta (\|x_i - z_l\|^2 - \|z_l - z\|^2), \quad (4.2)$$

де  $z$  та  $z_l$  – середнє арифметичне усієї виборки та образів віднесених до кластеру  $M_l$  відповідо. З визначення (4.2) видно, що малі значення FS говорять про компактні добре визначені кластери.

Нечітка множина  $i$ -ого образу визначається як

$$\tilde{A}_l = \sum_{i=1}^N \frac{u_{li}}{x_i}, l = 1, 2, \dots, M. \quad (4.3)$$

Ступінь, в якій  $A_l$  є підмножиною  $A_p$  визначається наступним чином

$$\begin{cases} S(\tilde{A}_l, \tilde{A}_p) = \frac{U(\tilde{A}_l \cap \tilde{A}_p)}{U(\tilde{A}_l)}, \\ U(\tilde{A}_j) = \sum_{i=1}^N u_{ji}. \end{cases} \quad (4.4)$$

Зважаючи на (4.4), можна запропонувати такі варіанти обчислення міри подібності:

$$N_1(\tilde{A}_l, \tilde{A}_p) = \frac{S(\tilde{A}_l, \tilde{A}_p) + S(\tilde{A}_p, \tilde{A}_l)}{2}, \quad (4.5a)$$

$$N_2(\tilde{A}_l, \tilde{A}_p) = \min(S(\tilde{A}_l, \tilde{A}_p), S(\tilde{A}_p, \tilde{A}_l)), \quad (4.5b)$$

$$N_3(\tilde{A}_l, \tilde{A}_p) = S(\tilde{A}_l \cup \tilde{A}_p, \tilde{A}_p \cap \tilde{A}_l). \quad (4.5c)$$

Тоді індекс дійсності кластерування, що ґрунтуються на нечіткій подібності, можна визначити як

$$FSim = \max_{1 \leq l \leq M} \max_{1 \leq p \leq M, p \neq l} N(\tilde{A}_l, \tilde{A}_p), \quad (4.6)$$

де міру нечіткої подібності  $N(\tilde{A}_l, \tilde{A}_p)$  можна знайти за будь-яким виразом (4.5).

### 4.3. Архітектура каскадної мережі, що еволюціонує, для нечіткого кластерування

До нульового шару системи послідовно передаються дані у формі векторного сигналу  $x(k) = (x_1(k), x_2(k), \dots, x_1(k))^T$ , де  $k = 1, 2, \dots, N, N+1, \dots$  — індекс поточного дискретного часу. Вхідні сигнали надходять до всіх вузлів системи  $N_j^{[m]}$ , де  $j = 1, 2, \dots, q$  — кількість вузлів у пулі-ансамблі,  $m = 1, 2, \dots$  — номер каскаду. Вузол кожного каскаду призначений для онлайн кластерування потоку даних і відрізняється від вузлів-сусідів використаним алгоритмом навчання або, у випадку спільногометоду кластерування, параметрами алгоритму. Кількість кластерів для кожного каскаду є відомою і дорівнює  $m + 1$ . Елемент  $PC_j^{[m]}$  дає оцінку якості кластерування кожного вузла у пулі, а елемент  $PC^{*[m]}$  визначає найкращий елемент у пулі кожного каскаду. Елемент системи  $XB^{[m]}$  оцінює загальну якість кластеризації пула, враховуючи прийняту кількість кластерів  $m + 1$ . Таким чином, система розв'язує задачу кластерування нестационарного потоку даних в умовах невизначеності щодо кількості кластерів, а також їх вигляду і рівню взаємного перекриття. І, нарешті, вихідний вузол системи  $XB^*$ , порівнюючи якість кластеризації кожного з каскадів, виділяє найкращий результат — кількість кластерів, їх центроїди-прототипи та рівні належності кожного спостереження до кожного з сформованих центроїдів. Незважаючи на удавану громіздкість чисельна реалізація запропонованої архітектури не викликає принципових труднощів завдяки тому, що потік даних, що надходить до системи, може оброблятися у паралельному режимі вузлами системи  $N_j^{[m]}$  [2, 6].

### 4.4. Адаптивне навчання вузлів каскадної нейро-фаззи системи, що еволюціонує

В основі алгоритмів навчання вузлів системи лежать алгоритми нечіткого кластерування, засновані на цільових функціях, такі, що вирішують задачу їх оптимізації при деяких апріорних припущеннях. Найбільш поширеним є

ймовірнісний підхід, заснований на мінімізації цільової функції

$$E\left(u_{jl}^{[m]}(k), c_{jl}^{[m]}\right) = \sum_{k=1}^N \sum_{l=1}^{m+1} \left(u_{jl}^{[m]}(k)\right)^\beta \|x(k) - c_{jl}^{[m]}\|^2 \quad (4.7)$$

при обмеженнях

$$\sum_{l=1}^{m+1}(k) = 1, \quad 0 \leq \sum_{k=1}^N u_{jl}^{[m]}(k) \leq N \quad (4.8)$$

де  $u_{ij}^{[m]}(k) \in [0, 1]$  — рівень належності спостереження  $x(k)$  до  $l$ -ого кластера у  $j$ -ому вузлі каскаду  $m$ ,

$c_{jl}^{[m]} = (n \times 1)$  - вимірний вектор-центроїд  $l$ -ого кластера у  $j$ -ому вузлі каскаду  $m$ ,

$\beta > 1$  — параметр фаззіфікації (фаззіфікатор), що визначає розмитість границь між кластерами,

$k = \overline{1, N}$  — номер образу ( $N$  — кількість образів у вхідній виборці, що, у рамках класичного подходу Бездека, вважається незмінною та такою, що задана априорі).

Вводячи функцію Лагранжа

$$L\left(u_{jl}^{[m]}(k), c_{jl}^{[m]}, \lambda_j^{[m]}(k)\right) = \sum_{k=1}^N \sum_{l=1}^{m+1} \left(u_{jl}^{[m]}(k)\right)^\beta \|x(k) - c_{jl}^{[m]}\|^2 + \sum_{k=1}^N \lambda_j^{[m]}(k) \left(\sum_{l=1}^{m+1} u_{jl}^{[m]}(k) - 1\right) \quad (4.9)$$

(тут  $\lambda_j^{[m]}(k)$  — невизначений множник Лагранжа) та вирішивши систему рівнянь Каруша-Куна-Таккера, нескладно отримати шукане рішення у вигляді

$$\begin{cases} u_{jl}^{[m]}(k) = \frac{\left(\|x(k) - c_{jl}^{[m]}\|^2\right)^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m+1} \left(\|x(k) - c_{jl}^{[m]}\|^2\right)^{\frac{1}{1-\beta}}}, \\ c_{jl}^{[m]} = \frac{\sum_{k=1}^N \left(u_{jl}^{[m]}(k)\right)^\beta x(k)}{\sum_{k=1}^N \left(u_{jl}^{[m]}(k)\right)^\beta}, \\ \lambda_j^{[m]}(k) = -\left(\left(\sum_{l=1}^{m+1} \beta \|x(k) - c_{jl}^{[m]}\|^2\right)^{\frac{1}{1-\beta}}\right)^{\frac{1}{1-\beta}}, \end{cases} \quad (4.10)$$

що при  $\beta = 2$  збігається з алгоритмом нечітких с-середніх Бездека (FCM) [29] і приймає форму

$$\begin{cases} u_{jl}^{[m]}(k) = \frac{\|x(k) - c_{jl}^{[m]}\|^{-2}}{\sum_{l=1}^{m+1} \|x(k) - c_{jl}^{[m]}\|^{-2}}, \\ c_{jl}^{[m]} = \frac{\sum_{k=1}^N \left(u_{jl}^{[m]}(k)^2 x(k)\right)}{\sum_{k=1}^N \left(u_{jl}^{[m]}(k)\right)^2}. \end{cases} \quad (4.11)$$

Тут варто відзначити, що вибір фаззіфікатора  $\beta = 2$  в (4.11) не дає жодних переваг порівняно з довільним значенням  $\beta$  у (4.10), у зв'язку з чим пропонується використовувати різні значення параметра фаззіфікації для кожного вузла пулу каскаду, після чого вибирати найкращий результат залежно від прийнятого критерію якості нечіткого кластерування [52, 56, 67].

Для послідовної обробки потоку даних, що надходять в online режимі, у [10, 14] були запропоновані рекурентні алгоритми, в основі яких лежить процедура нелінійного програмування Ерроу-Гурвіца-Удзави [5]. Так, пакетному алгоритмові (4.10) відповідає вираз

$$\begin{cases} u_{jl}^{[m]}(k+1) = \frac{\|x(k+1) - c_{jl}^{[m]}(k)\|^{\frac{1}{1-\beta}}}{\sum_{l=1}^{m+1} \|x(k+1) - c_{jl}^{[m]}(k)\|^{\frac{1}{1-\beta}}}, \\ c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \eta(k+1) \left( u_{jl}^{[m]}(k+1) \right)^{\beta_j} \left( x(k+1) - c_{jl}^{[m]}(k) \right), \end{cases} \quad (4.12)$$

(тут  $\eta(k+1)$  — параметр кроку навчання), що є узагальненням алгоритму навчання Чанга-Лі [21] і при  $\beta = 2$  близьке до градієнтної процедури Парка-Деера [53].

$$\begin{cases} u_{jl}^{[m]}(k+1) = \frac{\|x(k+1) - c_{jl}^{[m]}(k)\|^{-2}}{\sum_{l=1}^{m+1} \|x(k+1) - c_{jl}^{[m]}(k)\|^{-2}}, \\ c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \eta(k+1) \left( u_{jl}^{[m]}(k+1) \right)^2 \left( x(k+1) - c_{jl}^{[m]}(k) \right). \end{cases} \quad (4.13)$$

Варто зауважити, що, розглянувши співвідношення (4.12) з позицій навчання Кохоненової самоорганізованої мапи (SOM) [40], можна помітити, що множник  $\left( u_{jl}^{[m]} \right)^{\beta_j}$  відповідає функції сусідства в правилі навчання на основі принципу «переможецю дістается більше», маючи при цьому дзвонуватий вигляд.

Вочевидь, у випадку, коли  $\beta_j = 1$  та  $u_{jl}^{[m]}(k) \in [0, 1]$ , процедура (4.12) збігається з чітким алгоритмом  $c$ -середніх (HCM), коли ж  $\beta_j = 0$ , маємо стандартне правило навчання Кохонена «переможцю дістается все» [40]

$$c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \eta(k+1) \left( x(k+1) - c_{jl}^{[m]}(k) \right), \quad (4.14)$$

запропоноване Каш'япом та Блейдоном [39] у шістдесятих роках минулого століття. Легко побачити, що процедура (4.14) оптимізує цільову функцію

$$E\left(c_{jl}^{[m]}\right) = \sum_{k=1}^N \|x(k) - c_{jl}^{[m]}\|^2, \quad \sum_{l=1}^{m+1} N_l = N, \quad (4.15)$$

мінімум якої збігається із середнім арифметичним

$$c_{jl}^{[m]} = \frac{1}{N} \sum_{k=1}^{N_l} x(k), \quad (4.16)$$

де  $N_l$  — кількість векторів, віднесених до  $l$ -го кластеру у процесі конкуренції.

Якщо записати (4.16) у рекурентній формі, отримаємо оптимальний алгоритм самонавчання Ципкіна [25]

$$c_{jl}^{[m]}(k+1) = c_{jl}^{[m]}(k) + \frac{1}{N_l(k+1)} \left( x(k+1) - c_{jl}^{[m]}(k) \right), \quad (4.17)$$

де  $N_l(k+1)$  — число векторів, віднесених до  $l$ -го кластеру в  $k+1$ -й момент реального часу, що є стандартною процедурою стохастичної апроксимації.

У загальному випадку алгоритм навчання (4.12) вузла можна розглядати як правило самонавчання нечіткої модифікації самоорганізованої мапи Кохонена, архітектура якої наведена на рис

TODO:рис

Тут  $N_{jl}^{[m]K}$  — стандартні нейрони Кохонена, пов'язані між собою латеральними зв'язками, що налаштовуються згідно "переможцю дістається більше" правила навчання на основі другого співвідношення (4.12). Вузли  $N_{jl}^{[m]u}$  обчислюють рівні належності згідно першому співвідношенню (4.12). Вузли  $N_j^{[m]}$  кожного з каскадів відрізняються тільки фаззіфікатором алгоритму самонавчання, а вузол кожного наступного каскаду містить додатково один нейрон Кохонена і один елемент для розрахунку рівнів належності.

#### 4.5. Керування каскадами самонавчанної нейро-фаззі системи, що еволюціонує

Якість кластерування кожного вузла системи може бути оцінена за допомогою будь-якого з індексів, що використовуються у задачах нечіткого кластерування [72]. Одим за найпростіших та разом з тим найефективніших індексів є так званий «коєфіцієнт розбиття», який, власне, є середнім квадратів рівнів належності всіх спостережень до кожного кластеру і має вигляд

$$PC_j^{[m]} = \frac{1}{N} \sum_{k=1}^N \sum_{l=1}^{m+1} \left( u_{jl}^{[m]}(k) \right)^2. \quad (4.18)$$

Цей коєфіцієнт має ясний фізичний зміст: щокраще виражені кластери, то більше значення  $PC_j^{[m]}$  (верхня межа —  $PC_j^{[m]} = 1$ ), а його мінімум  $PC_j^{[m]} = (m + 1)^{-1}$  досягається, якщо дані належать усім кластерам рівномірно, що, вочевидь, є тривіальним рішенням. Для розглянутої нами системи цей коєфіцієнт зручний тим, що його легко розрахувати в online режимі

$$PC_j^{[m]}(k+1) = PC_j^{[m]}(k) + \frac{1}{k+1} \left( \sum_{l=1}^{m+1} \left( u_{jl}^{[m]}(k+1) \right)^2 - PC_j^{[m]}(k) \right). \quad (4.19)$$

Розрахунок коєфіцієнту розбиття проводиться для кожного вузла системи разом з налаштуванням їх параметрів, тобто співвідношення (4.12) та (4.19) реалізуються одночасно. На кожному такті навчання вузол  $PC^{*[m]}$  визначає найкращий елемент каскаду, що забезпечує максимальне значення коєфіцієнта розбиття у кожний поточний момент  $k$ , при цьому не виключається ситуація, коли в різні моменти обробки інформації "переможцями" виявляться різні вузли.

Кожен з каскадів розглянутої системи відрізняється від інших числом кластерів, на які розбивається оброблюваний потік даних. Тому якщо вузли  $PC_j^{[m]}$  і  $PC^{*[m]}$  оцінюють якість кластеризації без урахування кількості

сформованих класів, то вузли системи, позначені  $XB^{[m]}$  та  $XB^*$ , оцінюють результати з урахуванням числа кластерів у кожному каскаді. Одним з таких показників є індекс Ксі-Бені [71], який для фіксованої вибірки з  $N$  спостережень може бути записаний у вигляді

$$XB_j^{[m]} = \frac{\left( \sum_{k=1}^N \sum_{l=1}^{m+1} \left( u_{jl}^{[m]}(k) \right)^2 \|x(k) - c_{jl}^{[m]}\|^2 \right) / N}{\min_{l \neq q} \|c_{jl}^{[m]} - c_{jq}^{[m]}\|^2} = \frac{N XB_j^{[m]}}{DXB_j^{[m]}} \quad (4.20)$$

Вираз (4.20) також можна записати у рекурентній формі

$$\begin{aligned} XB_j^{[m]}(k+1) &= \frac{N XB_j^{[m]}(k+1)}{DXB_j^{[m]}(k+1)} = \\ &= \frac{N XB_j^{[m]}(k) + \frac{1}{k+1} \left( \sum_{l=1}^{m+1} \left( u_{jl}^{[m]}(k+1) \right)^2 \|x(k+1) - c_{jl}^{[m]}(k+1)\|^2 - N XB_j^{[m]}(k) \right)}{\min_{l \neq q} \|c_{jl}^{[m]}(k+1) - c_{jq}^{[m]}(k+1)\|^2}, \end{aligned} \quad (4.21)$$

при цьому рекурентні вирази (4.19) та (4.21) реалізуються одночасно.

Індекс Ксі-Бені є по суті співвідношенням відхилення всередині кластерів  $N XB_j^{[m]}$  до величини поділу кластерів  $DXB_j^{[m]}$ . Оптимальному числу кластерів у каскаді відповідає мінімальне значення (4.20) та (4.21). Тому процес нарощування каскадів у системі продовжується доки значення індексу не почне збільшуватися. Цей процес контролює вузол архітектури  $XB^*$ .

Варто зауважити, що оскільки вузли кожного каскаду відрізняються тільки значенням фаззифікатору, ефективність роботи кожного каскаду доцільно оцінювати за допомогою розширеного індексу Ксі-Бені  $EXB$  [72].

$$EXB_j^{[m]} = \frac{\left( \sum_{k=1}^N \sum_{l=1}^{m+1} \left( u_{jl}^{[m]}(k) \right)^{\beta[m]} \|x(k) - c_{jl}^{[m]}\|^2 \right) / N}{\min_{l \neq q} \|c_{jl}^{[m]} - c_{jq}^{[m]}\|^2} = \frac{NEXB_j^{[m]}}{DEXB_j^{[m]}} \quad (4.22)$$

або його рекурентої форми

$$\begin{aligned} XB_j^{[m]}(k+1) &= \frac{NXB_j^{[m]}(k+1)}{DXB_j^{[m]}(k+1)} = \\ &= \frac{NXB_j^{[m]}(k) + \frac{1}{k+1} \left( \sum_{l=1}^{m+1} \left( u_{jl}^{[m]}(k+1) \right)^{\beta[m]} \|x(k+1) - c_{jl}^{[m]}(k+1)\|^2 - NXB_j^{[m]}(k) \right)}{\min_{l \neq q} \|c_{jl}^{[m]}(k+1) - c_{jq}^{[m]}(k+1)\|^2}, \end{aligned} \quad (4.23)$$

де  $\beta^{[m]}$  — фаззіфікатор найкращого з вузлів  $m$ -го каскаду.

Таким чином, процес еволюції пропонованої системи зумовлений максимізуванням поточного значення показника якості кластерування потоку даних, що надходять на обробку в онлайн режимі.

## Висновки до розділу ??

1. Розглянуто завдання нечіткого кластерування у режимі послідовного надходження даних до системи.
2. Запропоновано метод визначення локально оптимальної кількості кластерів і значення параметру фаззіфікації для послідовного кластерування потоків даних.
3. Запропоновано архітектуру і метод самонавчання каскадної нейрофаззі системи, що еволюціонує, для послідовного кластерування потоків даних з автоматичним визначенням оптимальної кількості кластерів. Кожен вузол кожного каскаду системи вирішує завдання кла-

стерування незалежно від інших, що дозволяє організувати паралельну обробку інформації в каскадах, тобто підвищити швидкодію цього процесу. Система не містить жодних порогових параметрів, що за-даються суб'єктивно, а процес оцінювання якості її функціонування визначається шляхом відшукання оптимального значення певного ін-дексу дійсності розбиття даних на кластери (їх поточна оцінка також проводиться в режимі реального часу). Відмінною особливістю пропо-нованої системи є те, що вона самостійно визначає і поточне значен-ня фаззіфікатору, і оптимальну кількість кластерів на кожному етапі обробляння даних.

## РОЗДІЛ 5

### МОДЕЛЮВАННЯ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ РОЗРОБЛЕНИХ МЕТОДІВ ТА АРХІТЕКТУР

Для проведення чисельних експериментів, що наведені у підрозділах 5.1, 5.2 та 5.3 було обрано такі критерії оцінки:

- RMSE (Root Mean Squared Error, середньоквадратична похибка),
- SMAPE (Symmetric Mean Absolute Percentage Error, симетрична абсолютно процентна похибка),

що обчислюються за формулами

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(k) - \hat{y}(k))^2}, \quad (5.1)$$

та

$$\text{SMAPE} = \frac{1}{N} \sum_{k=1}^N \frac{|y(k) - \hat{y}(k)|}{|y(k)| + |\hat{y}(k)|} \quad (5.2)$$

відповідно, де  $y$  – шуканий сигнал,  $\hat{y}$  – вихідний сигнал системи.

#### **5.1. Моделювання розширенного нейро-фаззі нейрона**

Датасет для першого есперименту (фазовий портрет наведено на рис. 5.1) було сгенеровано за формулою

$$\sin(k + \sin(2k)) \text{ для } k \in [1, 600]. \quad (5.3)$$

Результати експерименту наведено у таблиці нижче, а також проілюстровано залежність точності прогнозу від порядку висновування (рис. 5.2) та кількості функцій належності (рис. 5.3).

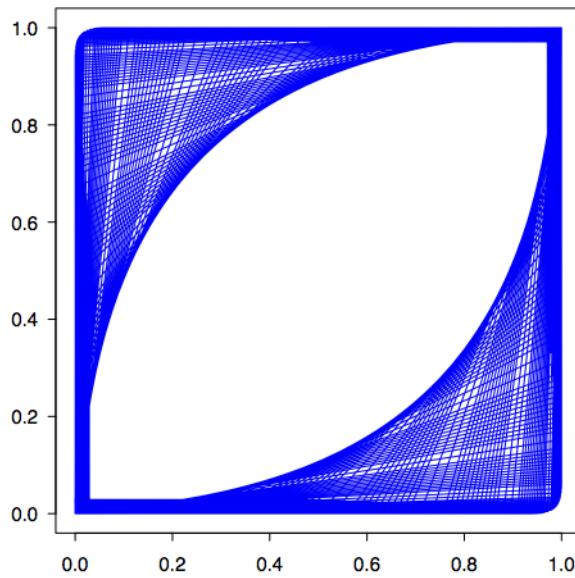


Рис. 5.1 – Фазовий портрет штучно сгенерованого датасету

Отже, як видно з таблиці 5.1 та рис. 5.4, можна зробити висновок, що точність прогнозу розширеного нео-фаззі нейрону вища від точності звичайного нео-фаззі нейрону (ENFN з нульовим порядком висновування).

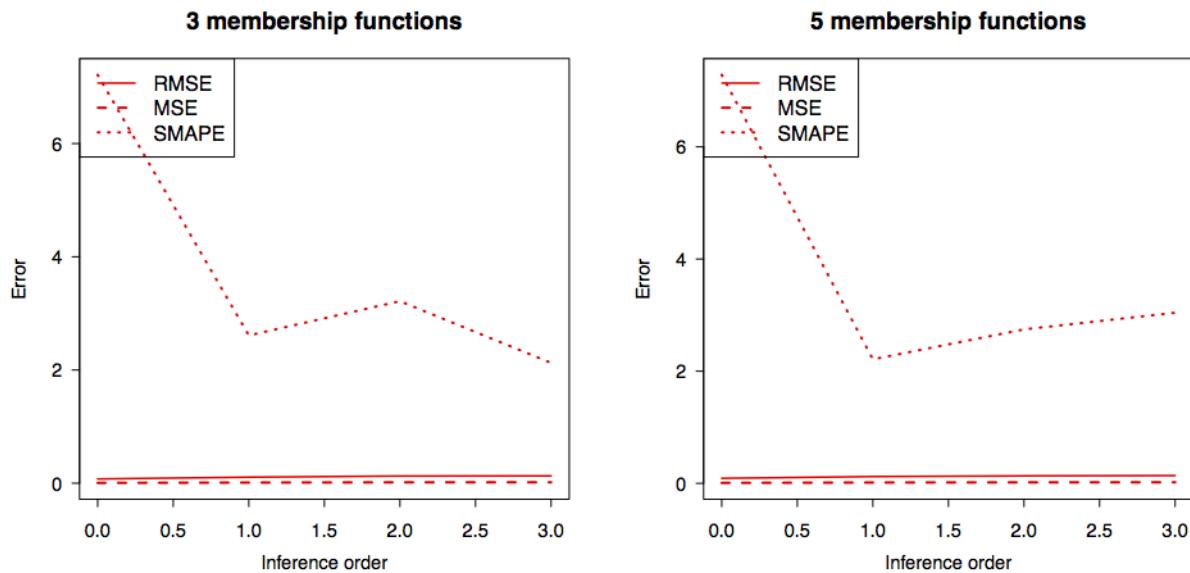


Рис. 5.2 – Похибка прогнозу розширеного нео-фаззі нейрону від порядку висновування (для трьох та п'яти функцій належності)

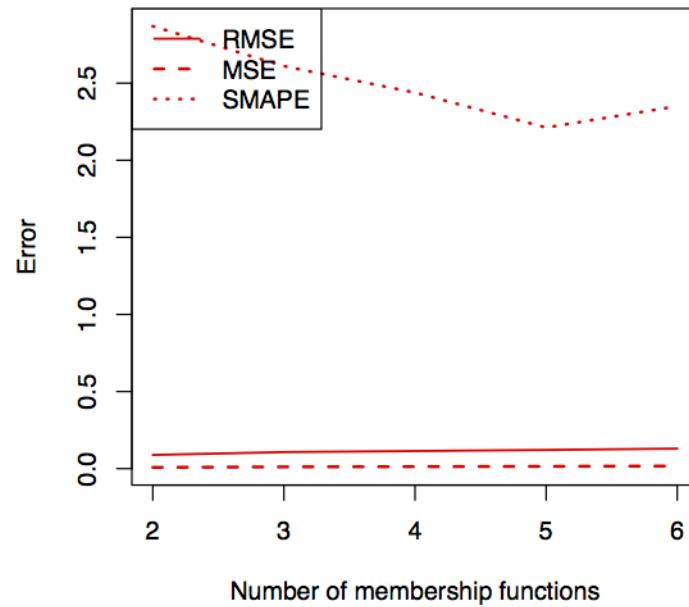


Рис. 5.3 – Похибка прогнозу розширеного нео-фаззі нейрону від кількості функцій належності (порядок висновування - 2)

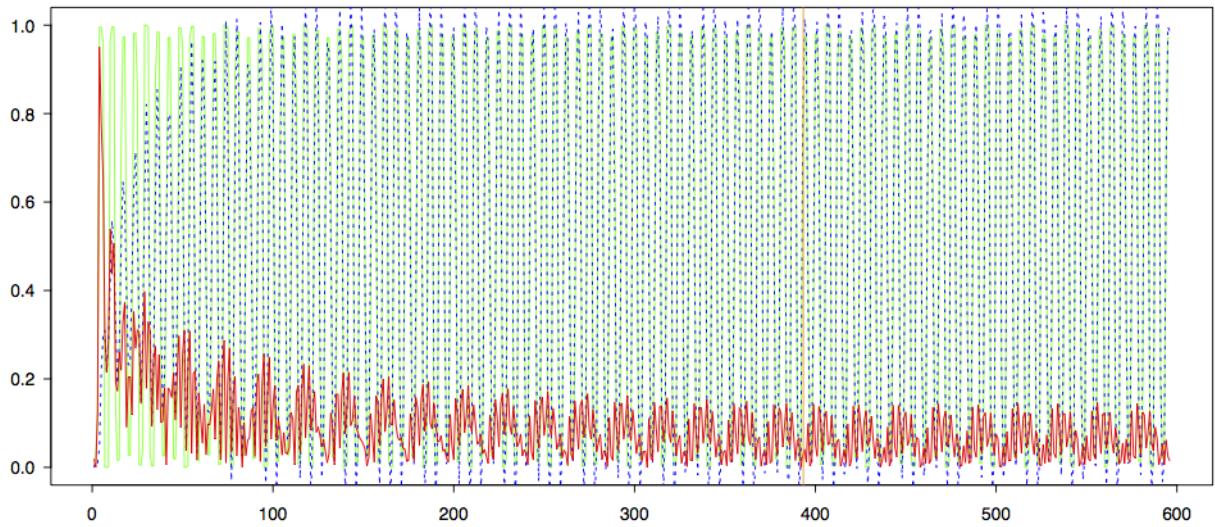


Рис. 5.4 – Прогноз розширеного нео-фаззі нейрону з 3 трикутними функціями належності, що реалізує нечітке висновування 3-ого порядку (зелена лінія – шуканий сигнал, синя пунктирна лінія – прогноз розширеного нео-фаззі нейрону, червона лінія – похибка; жовтобагряна вертикальна лінія позначає закінчення тренувальної частини датасету)

Таблиця 5.1 – Точність прогнозу розширеного нео-фаззі нейрону на штучно сгенерованому датасеті

| Нечітке висновування 0 порядку   |                    |                  |       |
|----------------------------------|--------------------|------------------|-------|
| Функції належності               |                    | RMSE             | SMAPE |
| 3                                | 0.07593154138757   | 7.2130842903092  |       |
| 4                                | 0.0816015155371995 | 3.74330006151476 |       |
| 5                                | 0.0899940897089563 | 7.28423693259539 |       |
| 6                                | 0.098548606433121  | 4.97414996889677 |       |
| Нечітке висновування I порядку   |                    |                  |       |
| Функції належності               |                    | RMSE             | SMAPE |
| 3                                | 0.107342623622113  | 2.61031587883906 |       |
| 4                                | 0.114302161682684  | 2.43845462677015 |       |
| 5                                | 0.121556611017793  | 2.21187746150696 |       |
| 6                                | 0.129922583600673  | 2.3504927587044  |       |
| Нечітке висновування II порядку  |                    |                  |       |
| Функції належності               |                    | RMSE             | SMAPE |
| 3                                | 0.127570672613453  | 3.21603084937958 |       |
| 4                                | 0.121809301731276  | 2.49191939755954 |       |
| 5                                | 0.134120568445479  | 2.74555892364861 |       |
| 6                                | 0.146724724859333  | 2.39716069975922 |       |
| Нечітке висновування III порядку |                    |                  |       |
| Функції належності               |                    | RMSE             | SMAPE |
| 3                                | 0.130173960343431  | 2.12509843555904 |       |
| 4                                | 0.154580667425542  | 2.17237774290269 |       |
| 5                                | 0.139334351692536  | 3.04236918521622 |       |
| 6                                | 0.150497843534837  | 2.43818195954604 |       |

Для подальшої апробації розширеного нео-фаззі нейрону розглянемо задачу прогнозування хаотичного ряду, що описується диференціальним рівнянням Мекі-Гласса [147 вік]:

$$y'(t) = \frac{0.2(t - \tau)}{1 + y^{10}(t - \tau)} - 0.1y(t), \quad (5.4)$$

при цьому значення часового ряду в кожній точці обчислене за допомогою методу Рунге-Кутта четвертого порядку. Часовий крок прийнятий рівним 0.1, початкові умови:  $x(0) = 1.2$ .

Традиційно завдання прогнозування полягає у визначенні  $x(t+6)$ х часового ряду (5.2) з параметром затримки 17 по значенням  $x(t+18)$ ,  $x(t+12)$ ,  $x(t+6)$  і  $x(t)$ . Перед початком обробки отриманий часовий ряд нормувався таким чином, щоб його значення лежали в інтервалі  $[0, 1]$  (область визначення трикутних функцій належності та кубічних сплайнів, що використовуються у синапсах розширеного нео-фаззі нейрону).

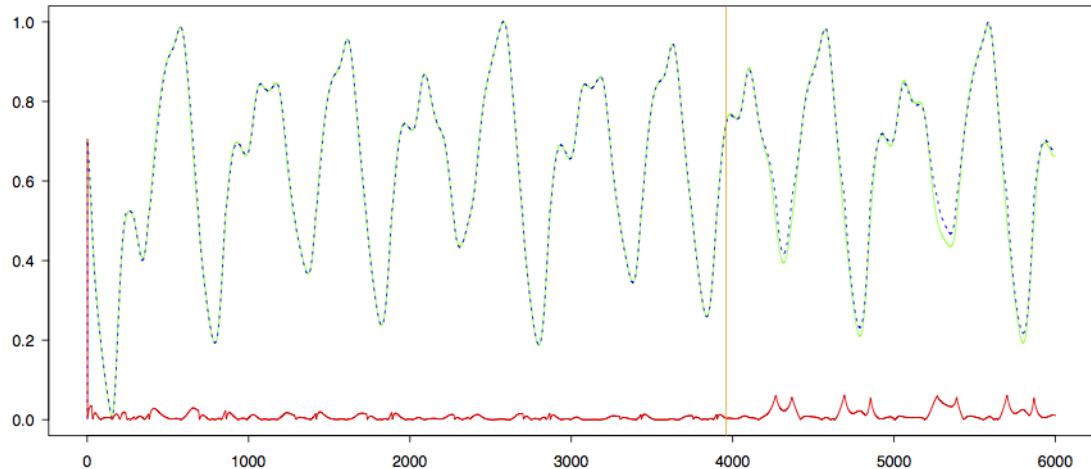


Рис. 5.5 – Прогнозування хаотичного часового ряду Макі-Гласса розширенним нео-фаззі нейроном з 3 трикутними функціями належності, що реалізує нечітке висновування 3-ого порядку (зелена лінія – шуканий сигнал, синя пунктирна лінія – сигнал на виході нейрону, червона лінія – похибка; жовто-багряна вертикальна лінія позначає закінчення тренувальної частини датасету)

На рис.5.5 зображений результат прогнозування розширеного нео-фаззі нейрона з трьома функціями належності, що реалізує нечітке висновування 3-ого порядку. Для порівняння на рис. 5.6 наведено прогнозування традиційного нео-фаззі нейрона (що реалізує нечітке висновування нульового порядку) з аналогічними функціями належності (3 трикутні функції належності).

Таблиця 5.2 – Точність прогнозу ряду Мекі-Глассу розширеним нео-фаззі нейроном від порядку висновування та кількості функцій належності

| Нечітке висновування 0 порядку   |                    |                    |                    |
|----------------------------------|--------------------|--------------------|--------------------|
|                                  | Функції належності | RMSE               | SMAPE              |
|                                  | 3                  | 0.143482904587951  | 0.325353000527264  |
|                                  | 4                  | 0.106294989490131  | 0.264501060116694  |
|                                  | 5                  | 0.094578207548207  | 0.259630642224594  |
|                                  | 6                  | 0.094578207548207  | 0.259630642224594  |
| Нечітке висновування I порядку   |                    |                    |                    |
|                                  | Функції належності | RMSE               | SMAPE              |
|                                  | 3                  | 0.0491962584140483 | 0.136034984261077  |
|                                  | 4                  | 0.0300830061654526 | 0.0978878141219238 |
|                                  | 5                  | 0.0312245619190396 | 0.10776716206662   |
|                                  | 6                  | 0.0277981639612314 | 0.0989376310206592 |
| Нечітке висновування II порядку  |                    |                    |                    |
|                                  | Функції належності | RMSE               | SMAPE              |
|                                  | 3                  | 0.0227629037042365 | 0.0658002874902031 |
|                                  | 4                  | 0.0295377560571133 | 0.108260856276106  |
|                                  | 5                  | 0.0283735498990618 | 0.0946515030316458 |
|                                  | 6                  | 0.0251818186025806 | 0.0847398989365615 |
| Нечітке висновування III порядку |                    |                    |                    |
|                                  | Функції належності | RMSE               | SMAPE              |
|                                  | 3                  | 0.0221516072948077 | 0.0622040928066835 |
|                                  | 4                  | 0.0261638552968437 | 0.0827913036508308 |
|                                  | 5                  | 0.0274818416982828 | 0.114797690969503  |
|                                  | 6                  | 0.0237394982902305 | 0.0846415828789446 |

На рис. 5.7 показана залежність похибки від порядку висновування розширенних нео-фаззі нейронів з трьома та п'ятьма дзвонуватими функціями належності.

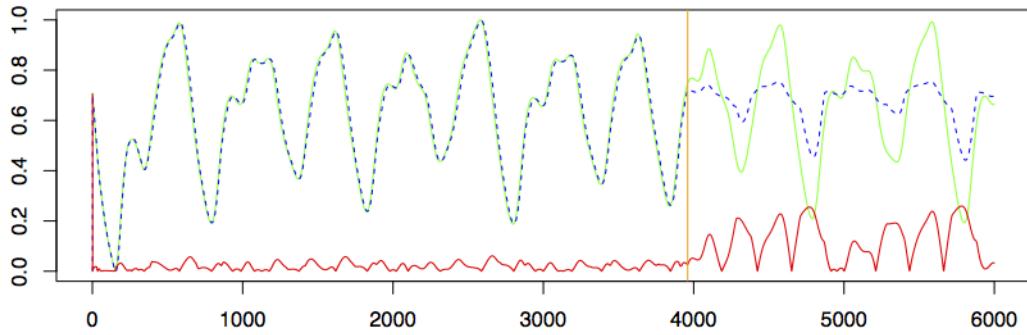


Рис. 5.6 – Прогнозування хаотичного часового ряду Макі-Гласса традиційним нео-фаззі нейроном з 3 трикутними функціями належності (зелена лінія – шуканий сигнал, синя пунктирна лінія – сигнал на виході нейрону, червона лінія – похибка; жовтобагряна вертикальна лінія позначає закінчення тренувальної частини датасету)

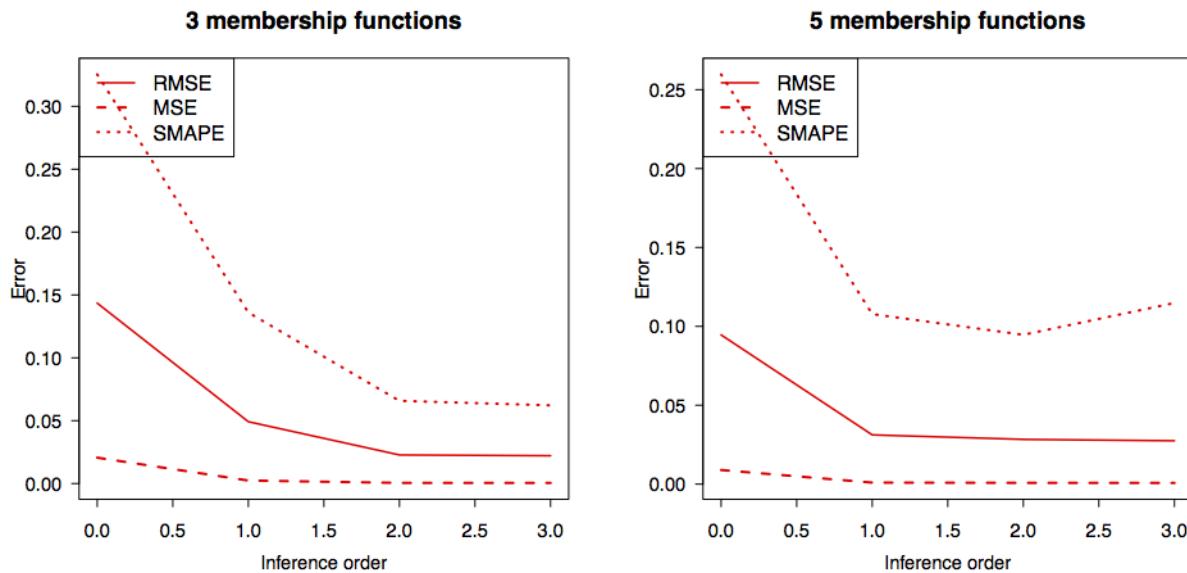


Рис. 5.7 – Похибка прогнозування хаотичного часового ряду Макі-Гласса розширенними нео-фаззі нейронами з 3 та 5 дзвонуватими функціями належності від порядку нечіткого висновування

Як видно з таблиці 5.2 та рис. 5.7, розширенний нео-фаззі нейрон, що реалізує нечітке висновування вищого від 0 порядку, прогнозує хаотичний часовий ряд за рівнянням Мекі-Гласса з суттєво вищою точністю ніж тради-

ційний нео-фаззі нейрон.

Також пропонований розширений нео-фаззі нейрон було апробовано на реальному часовому ряді «Споживання електоренергії у м. Сімферополь за 2007 рік» (фазовий портрет наведено на рис. 5.8).

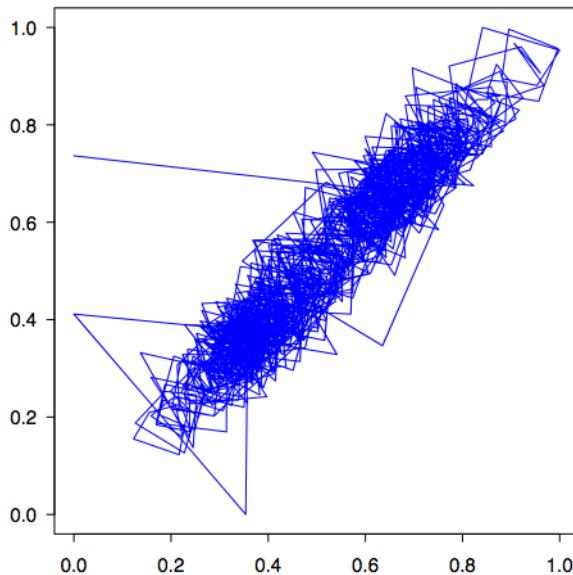


Рис. 5.8 – Фазовий портрет часовогого ряду «Споживання електоренергії у м. Сімферополь за 2007 рік»

Найліпший прогноз ( $\text{RMSE} \approx 0.14$ ,  $\text{SMAPE} \approx 0.21$ ) надав нейрон, що реалізує нечітке висновування 1-ого порядку з 6-ма функціями належності (рис. 5.9)

Експерименти, що описані у цьому підрозділі, підтверджують, що розширені нео-фаззі нейрони, які реалізують нечітке висновування довільного порядку, мають підвищенну точність прогнозування хаотичних рядів (як штучно сгенерованих, так і реальних) порівняно з традиційними нео-фаззі нейронами (які реалізують нечітке висновування нульового порядку).

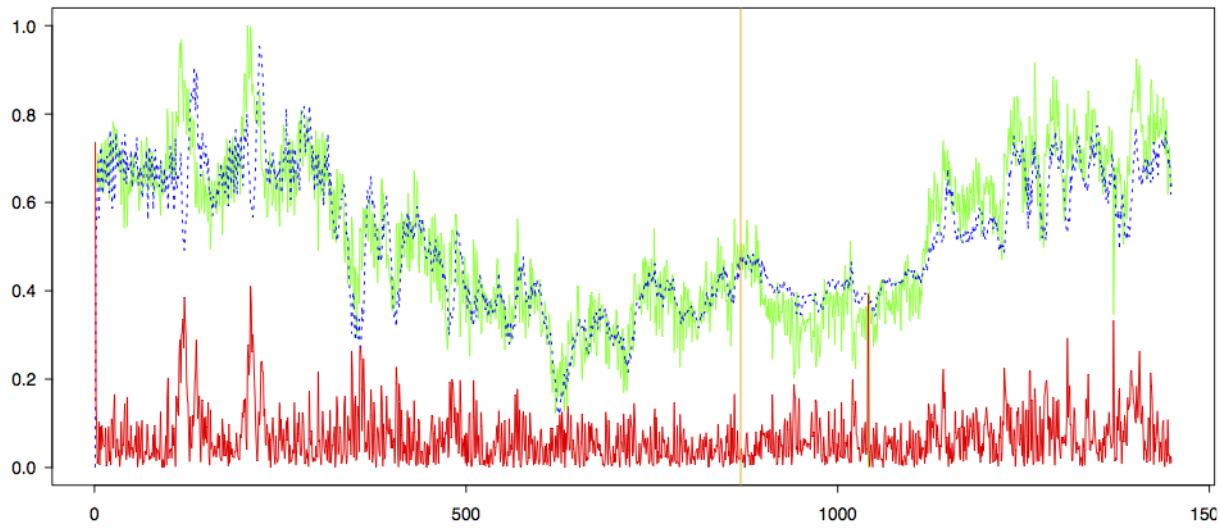


Рис. 5.9 – Прогнозування часового ряду «Споживання електроенергії у м. Сімферополь за 2007 рік» розширенним нео-фаззі нейроном з 6-ма дзвонувати функціями належності, що реалізує нечітке висновування 1-ого порядку (зелена лінія – шуканий сигнал, синя пунктирна лінія – сигнал на виході нейрона, червона лінія – похибка; жовтобагряна вертикальна лінія позначає закінчення тренувальної частини датасету)

## 5.2. Моделювання гібридної каскадної нейро-фаззі мережі з розширенними нео-фаззі нейронами та оптимізацією пулу нейронів

Низку експериментів для апробації гібридної каскадної нейро-фаззі мережі на розширенних нео-фаззі нейронах з оптимізацією пулу нейронів було проведено на датасетах, що їх надала дослідницька група «The Applications of Machine Learning (AML)» з Університету Аалто, що у Фінляндії (Aalto University School of Science, Espoo, Finland). Одним з таких датасетів є часовий ряд «Споживання електроенергії у Польщі за період з 1990-х років» (фазовий портрет наведено на рис 5.10).

Вихідний сигнал гібридної каскадної нейро-мережі наведено на рис. 5.11, а похибки розширенних нео-фаззі нейронів та нейронів-узагальнювачів для кожного каскаду системи наведено у таблиці 5.3.

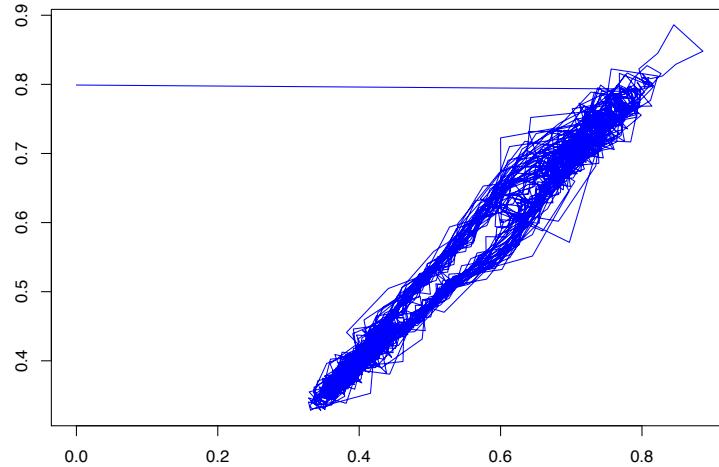


Рис. 5.10 – Фазовий портрет часового ряду Прогнозування часового ряду «Споживання електроенергії у Польщі за період з 1990-х років»

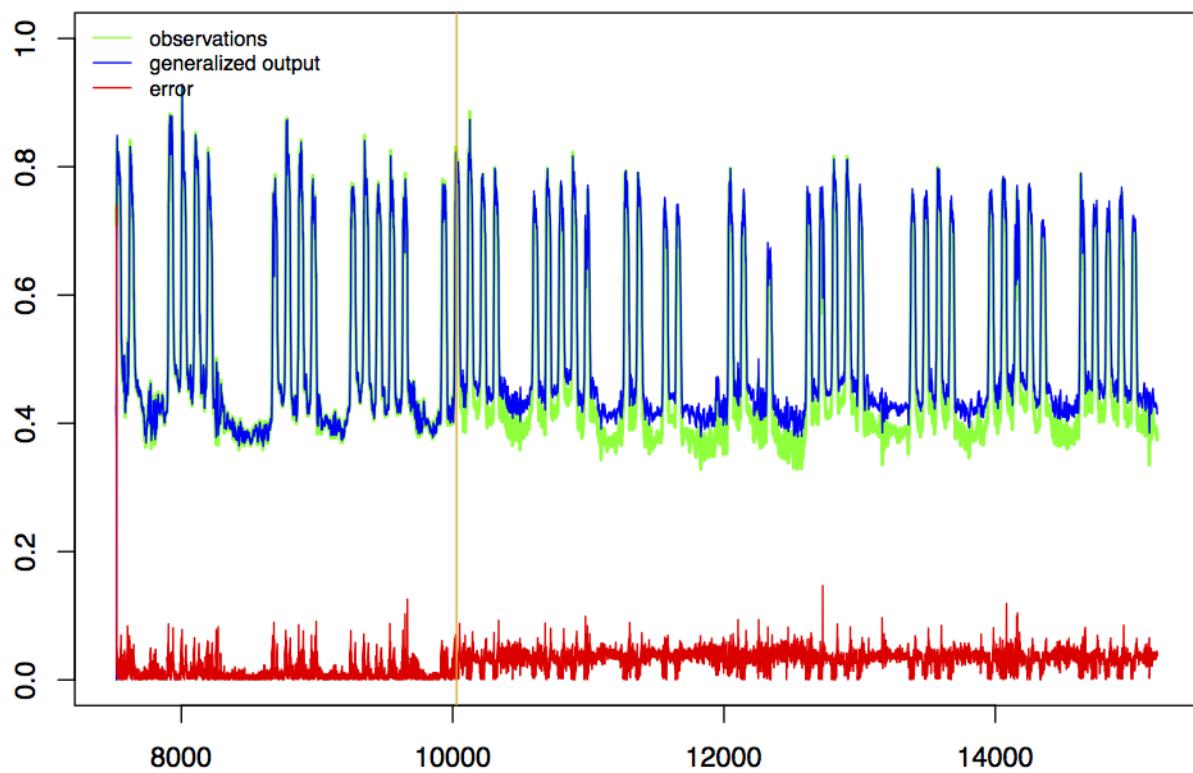


Рис. 5.11 – Прогнозування часового ряду «Споживання електоренергії у м. Сімферополь за 2007 рік» гібридної каскадної нейро-фаззі мережі з оптимізацією пулу нейронів (жовтобагряна вертикальна лінія позначає закінчення тренувальної частини датасету)

*Таблиця 5.3 – Результати прогнозування часового ряду «Споживання електротенергії у м. Сімферополь за 2007 рік» нейронів (зокрема нейронів-узагальнювачів) каскадної нейро-фаззі системи*

| Каскад I  | SMAPE    | RMSE     |
|---|----------|----------|
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.080830 | 0.39288  |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.075014 | 0.036680 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.074955 | 0.038170 |
| Нейрон-узагальнювач I каскаду                                     | 0.059835 | 0.030302 |
| Каскад II   | SMAPE    | RMSE     |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.080835 | 0.039290 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.059837 | 0.036683 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.074966 | 0.038195 |
| Нейрон-узагальнювач II каскаду                                    | 0.059821 | 0.036683 |
| Каскад III  | SMAPE    | RMSE     |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.080934 | 0.039333 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.059869 | 0.036711 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.75009  | 0.038213 |
| Нейрон-узагальнювач III каскаду                                   | 0.059869 | 0.030320 |
| Каскад IV   | SMAPE    | RMSE     |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.080892 | 0.039316 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.059869 | 0.030303 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.075034 | 0.038213 |
| Нейрон-узагальнювач IV каскаду                                    | 0.059849 | 0.030303 |
| Нейрон-узагальнювач системи                                       | 0.059821 | 0.030302 |

Фазовий портрет другого датасету «Коливання рівню приловоотливної зони» (Subtidal coastal level of fluctuations) наведено на рис. 5.11, похибки вузлів системи – у таблиці reftab:SubtidalCoastalLevelOfFluctuationsPredictionAccuracy.

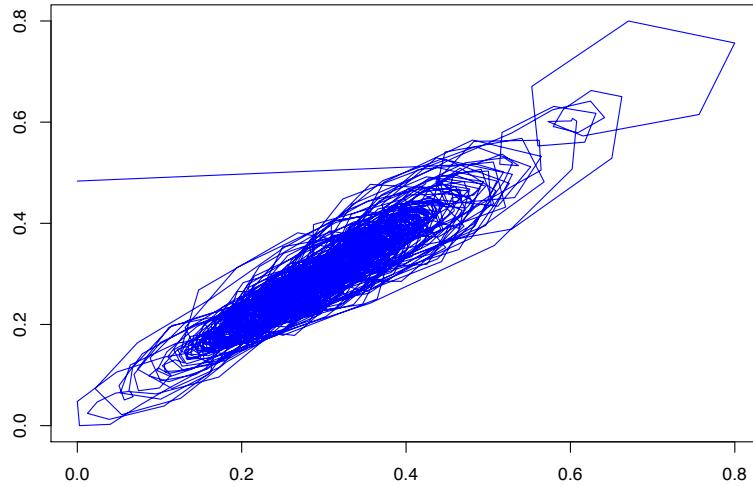
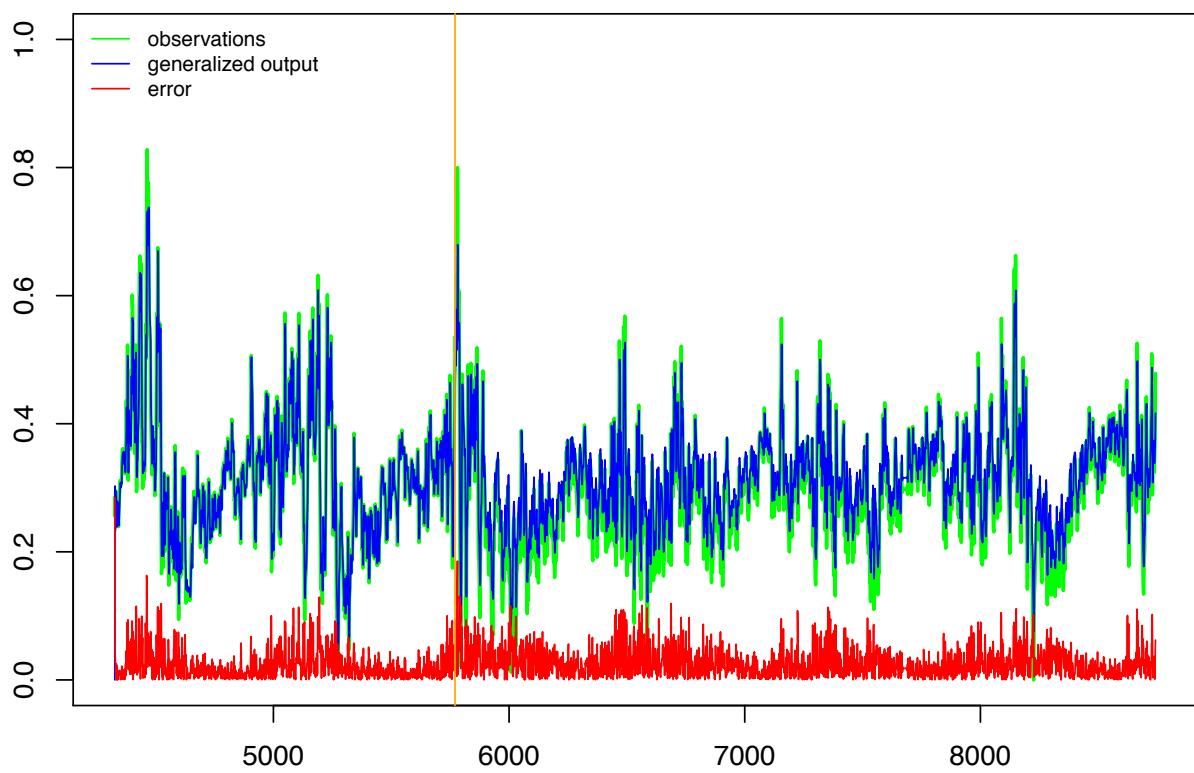


Рис. 5.12 – Фазовий портрет часового ряду Прогнозування часового ряду «Коливання рівню приловоотливної зони»



Прогнозування часового ряду «Коливання рівню приловоотливної зони» гібридної каскадної нейро-фаззі мережі з оптимізацією пулу нейронів (жовтобагряна вертикальна лінія позначає закінчення тренувальної частини датасету)

*Таблиця 5.4 – Результати прогнозування часового ряду «Коливання рівню приловоотливної зони» нейронів (зокрема нейронів-узагальнювачів) каскадної нейро-фаззі системи*

| Каскад I  | SMAPE    | RMSE     |
|---|----------|----------|
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.110067 | 0.036612 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.105192 | 0.035474 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.103129 | 0.034814 |
| Нейрон-узагальнювач I каскаду                                     | 0.105598 | 0.035301 |
| Каскад II   | SMAPE    | RMSE     |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.110023 | 0.036593 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.105118 | 0.035457 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.103148 | 0.035818 |
| Нейрон-узагальнювач II каскаду                                    | 0.102377 | 0.034698 |
| Каскад III  | SMAPE    | RMSE     |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.110013 | 0.036591 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.105126 | 0.035458 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.103155 | 0.034820 |
| Нейрон-узагальнювач III каскаду                                   | 0.103155 | 0.034820 |
| Каскад IV   | SMAPE    | RMSE     |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.110026 | 0.036594 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.105153 | 0.035464 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.103168 | 0.034822 |
| Нейрон-узагальнювач IV каскаду                                    | 0.102545 | 0.034083 |
| Нейрон-узагальнювач системи                                       | 0.102377 | 0.030302 |

Останній датасет для цієї низки експериментів було взято зі змагань у прогнозуванні часових рядів «European Symposium on Time Series Prediction 2008», фазовий портрет наведено на рис. 5.13, результат роботи пропонованої системи – на рис. 5.14.

Результати прогнозування нейронів окрім каскадів, нейронів-узагальнювачів, а також системи в цілому можна побачити у таблиці 5.5.

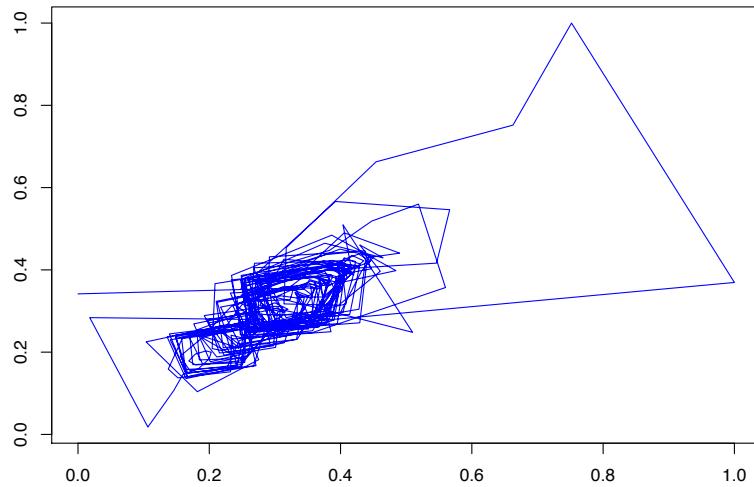


Рис. 5.13 – Фазовий портрет часового ряду Прогнозування часового ряду «ESTP Competition Time Series»

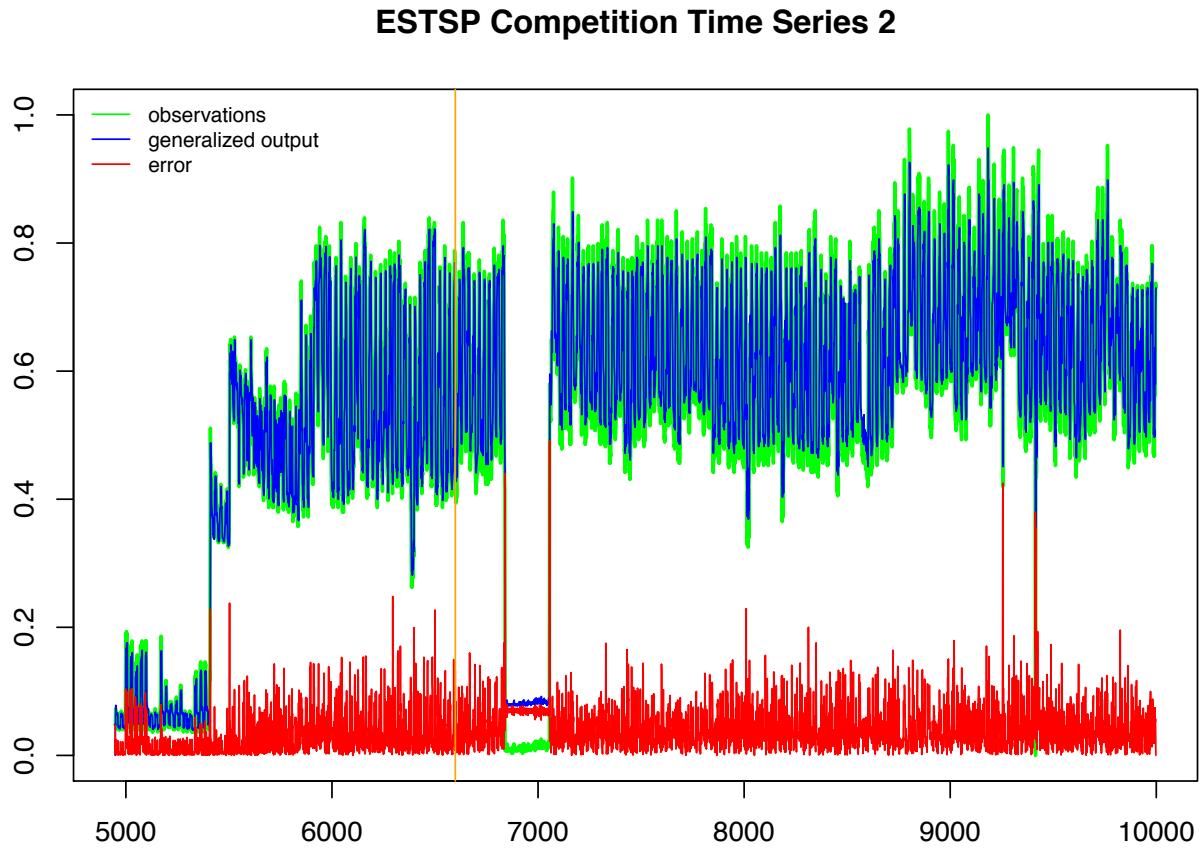


Рис. 5.14 – Прогнозування часового ряду «ESTP Competition Time Series» гібридної каскадної нейро-фаззі мережі з оптимізацією пулу нейронів (жовтобагряна вертикальна лінія позначає закінчення тренувальної частини датасету)

*Таблиця 5.5 – Результати прогнозування часового ряду «ESTP Competition Time Series» окремих нейронів (зокрема нейронів-узагальнювачів) каскадної нейро-фаззі системи*

| Каскад I  | SMAPE    | RMSE     |
|---|----------|----------|
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.175623 | 0.062985 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.149139 | 0.056551 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.146186 | 0.054414 |
| Нейрон-узагальнювач I каскаду                                     | 0.159002 | 0.055274 |
| Каскад II   | SMAPE    | RMSE     |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.175602 | 0.062942 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.149092 | 0.056487 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.146278 | 0.054415 |
| Нейрон-узагальнювач II каскаду                                    | 0.158996 | 0.055264 |
| Каскад III  | SMAPE    | RMSE     |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.175592 | 0.062933 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.149090 | 0.056487 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.146270 | 0.054415 |
| Нейрон-узагальнювач III каскаду                                   | 0.158986 | 0.055262 |
| Каскад IV   | SMAPE    | RMSE     |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.175605 | 0.062947 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.149127 | 0.056487 |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.146323 | 0.054419 |
| Нейрон-узагальнювач IV каскаду                                    | 0.159015 | 0.055268 |
| Нейрон-узагальнювач системи                                       | 0.102377 | 0.030302 |

### **5.3. Моделювання багатовимірної гібридної каскадної нейро-фаззі мережі, що еволюціонує, з оптимізацією пулу нейронів**

В якості тестового датасету для моделювання багатовимірної гібридної каскадної нейро-фаззі мережі з оптимізацією пулу нейронів (що ґрунтуються на багатовимірних нео-фаззі нейронах, як описано у [linkchapter](#)) застосовувався багатовимірний ряд, сгенерованих за домогою диференціальних рівнянь

моделі Лоренца:

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = -xz + rx - y, \\ \dot{z} = xy - bz. \end{cases} \quad (5.5)$$

У моделі Лоренца присутні три невідомих функції, а також кілька невідомих параметрів [21].

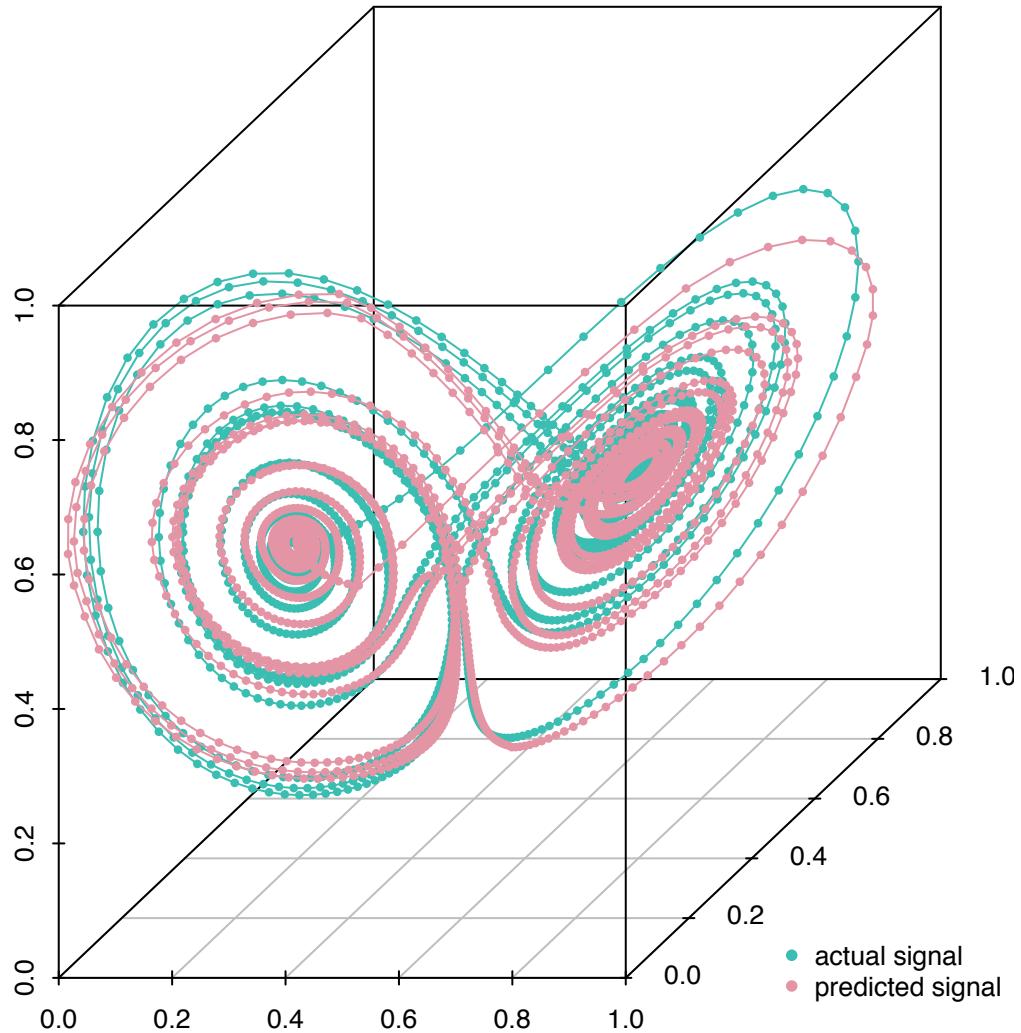


Рис. 5.15 – Прогнозування багатовимірного часового ряду гібридною каскадною нейро-фаззі мережею з оптимізацією пулу нейронів

При плавній зміні параметра динамічна система змінює тип свого аттра-

ктора. Рішення системи рівнянь Лоренца 5.5 при значенні параметра  $r$ , що перевищує біфуркаційних, виглядає майже ідентично випадковому процесу. У певному сенсі, атTRACTор Лоренца є стохастичними автоколиваннями, зо підтримуються у динамічній системі за рахунок зовнішнього джерела.

**Таблиця 5.6 – Результати прогнозування багатовимірного часового ряду нейронами (зокрема нейронами-узагальнювачами) МІМО гібридної каскадної нейро-фаззі мережі з оптимізацією пулу нейронів**

| Каскад I  | SMAPE    | RMSE      |
|---|----------|-----------|
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.132351 | 0.64333   |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.106885 | 0.055176  |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.118058 | 0.059517  |
| Нейрон-узагальнювач IV каскаду                                    | 0.106823 | 0.055111  |
| Каскад II   | SMAPE    | RMSE      |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.132370 | 0.064339  |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.106840 | 0.55165   |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.118059 | 0.059517  |
| Нейрон-узагальнювач IV каскаду                                    | 0.106840 | 0.55165   |
| Каскад III  | SMAPE    | RMSE      |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.132325 | 0.064324  |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.106858 | 0.055171  |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.118059 | 0.059513  |
| Нейрон-узагальнювач IV каскаду                                    | 0.106858 | 0.055171  |
| Каскад IV   | SMAPE    | RMSE      |
| Нейрон I (3 функції належності, нечітке висновування III порядку) | 0.132258 | 0.0642449 |
| Нейрон II (5 функцій належності, нечітке висновування IV порядку) | 0.106805 | 0.055144  |
| Нейрон III (4 функції належності, нечітке висновування V порядку) | 0.118015 | 0.059424  |
| Нейрон-узагальнювач IV каскаду                                    | 0.106805 | 0.055144  |
| Нейрон-узагальнювач системи                                       | 0.106789 | 0.55105   |

У фазовому просторі дивний атTRACTор має топологію деякого клубка траєкторій, в межах якого можна виділити дві області. У кожен момент часу рішення знаходиться в одній з цих областей, причому зміна станів системи в одну або іншу область є абсолютно непередбачуваною.

АтTRACTор Лоренца демонструє ще одну особливість, притаманну дивним атTRACTорам - чутливість до початкових умов. АтTRACTори, тобто нерухомі точки і граничні цикли, характеризуються тим, що для різних початкових умов сімейства рішень сходилися до одного асимптотичному рішення, тобто різні категорії вийшли з різних точок, які відповідають різним початковим умовам, сходилися при  $t \rightarrow \infty$  в одну точку або близькі криві. Тому поведінку звичайних систем, що мають атTRACTори поблизу нерухомих точок і граничних циклів, на великих часах добре передбачувано. З дивними атTRACTорами все зовсім інакше. Які б близькі початкові умови не вибиралися, при  $t \rightarrow \infty$  рішення будуть розходитися, віддаляючись одне від одного в фазовому просторі. Оскільки в реальних задачах початкові умови відомі з деякою погрешністю, абсолютно неможливо вказати поведінку такого атTRACTора при досить великому  $t$ , тому поведінка систем, що описуються дивними атTRACTорами, є абсолютно непередбачуваною.

Для генерування тестового датасету використовувались такі параметри:

$$r = 28,$$

$$dt = 0.001;$$

По завершенні експерименту маємо систему з чотирьох (рідше – трьох) каскадів з трьома багатовимірними нейронами MNFN та одним нейроном-узагальнювачем у кожному каскаді. Результати роботи системи зображені на рис. 5.15 та більш детально описані у таблиці 5.6.

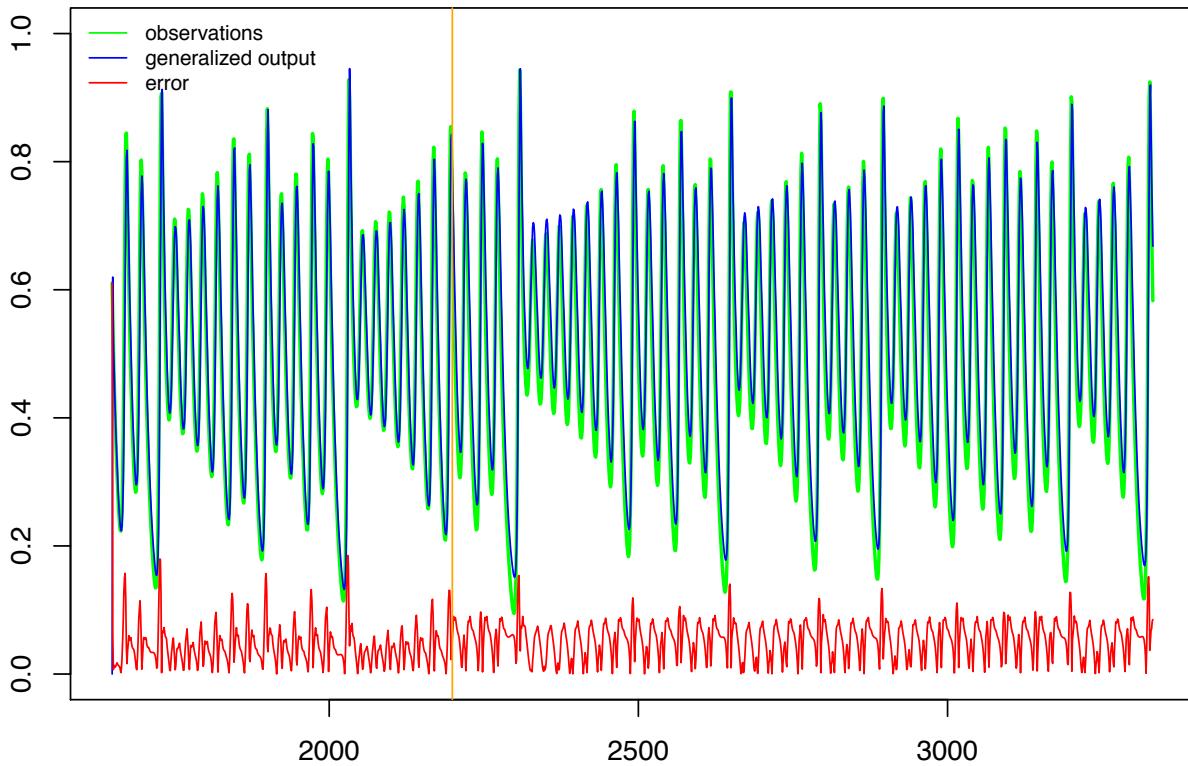


Рис. 5.16 – Прогнозування багатовимірного часового ряду гібридною каскадною нейро-фаззі мережею з оптимізацією пулу нейронів

#### 5.4. Моделювання самонавчанної нейро-фаззі системи, що еволюціонує

Одна з основних переваг, притаманних пропонованій самонавчанній нейро-фаззі системі, що еволюціонує, полягає в автоматичному визначенні оптимальної кількості кластерів та значення фаззифікатору на кожному етапі обробляння даних. Першу серію експериментів було проведено на штучно зсинтезованих наборах даних з різним ступенем розмитості та перекриття класів аби дослідити вплив значення параметру фаззіфікації на якість кластерування в режимі реального часу відповідно до обраного критерію дійсності.

Кожен з наборів даних, що їх наведено на рис. 5.17, містить 80 спостережень з 2 ознаками (для очності) у кожному спостереженні. Тестові дані були сгенеровані таким чином, аби у першому наборі класи були чітко розподі-

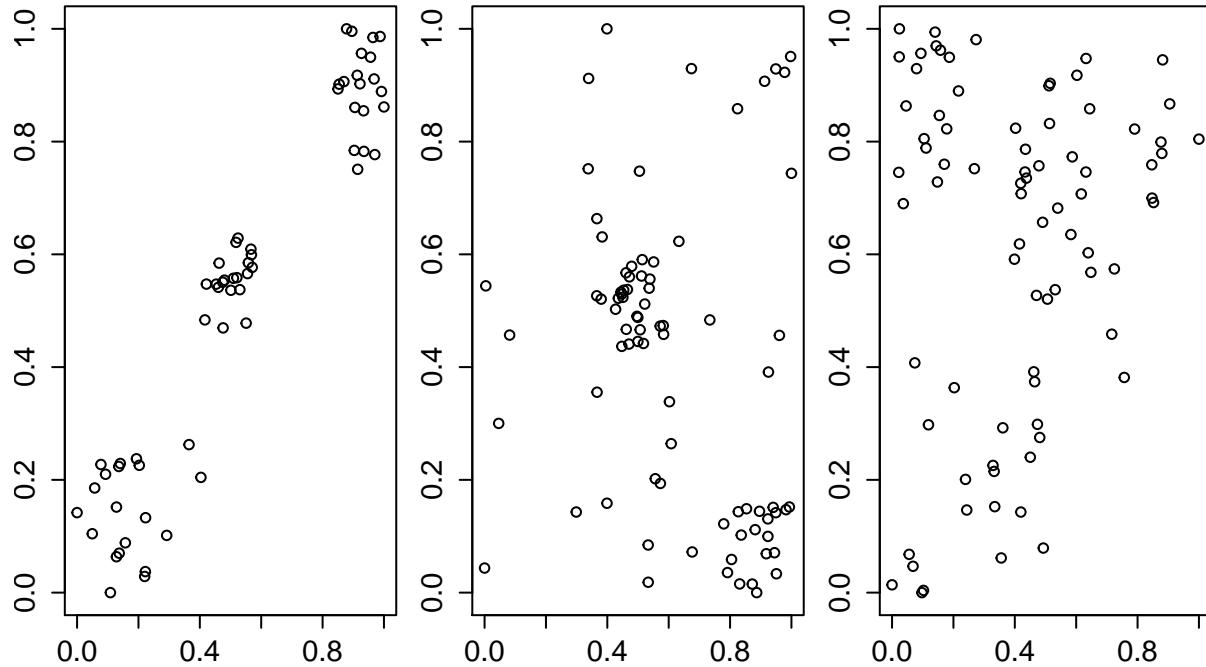


Рис. 5.17 – Штучно сгенеровані набори даних

лені (crisp dataset), у другому наборі кластерні границі були дещо розмиті (fuzzy dataset), у третьому випадку класи сильно перетиналися (extra fuzzy dataset). Логічно припустити, що система, яка тестиється, обере менше значення параметру фаззіфікації для первого датасету та більше для останнього, де границі класів спостережень є більш розмитими.

Спостереження надходили до нейро-фаззі мережі у послідовному режимі, вагові коефіцієнти нейронів були проініціалізовані, використовуючи пакетну модифікацію обраного алгоритму кластерування на датасеті з довільних двадцятьох спостережень відповідного набору даних (адже система, як і класичний fuzzy c-means, [чутлива до ініціалзації](#)). Локально оптимальні кількість кластерів та значення параметру фаззіфікації обумовлювалися максимальним середнім значенням рекурентних коефіцієнту розбиття РС (4.19) та Ксі-Бені індексу (4.21):  $\max \frac{PC_j^{[m]} + 1 - XB_j^{[m]}}{2}$  (у данному випадку використовувалося від'ємне значення Ксі-Бені індексу  $1 - XB(k)$ , оскільки щоменше  $XB_j^{[m]}$ , то ліпшим є розбиття даних на кластери).

Для первого набору даних (crisp dataset), як і передбачалося, оптималь-

Таблиця 5.7 – Індекси валідності кластерування I датасету

| Каскад 1 ( $m = 2$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
|----------------------|-------------|-------------|-------------|-------------|
| Коефіцієнт розбиття  | 0.91758     | 0.7446      | 0.64787     | 0.59236     |
| Індекс Ксі-Бені      | 0.052129    | 0.061034    | 0.092235    | 0.1294      |
| Каскад 2 ( $m = 3$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.92643     | 0.6609      | 0.50214     | 0.43305     |
| Індекс Ксі-Бені      | 0.027232    | 0.06872     | 0.17281     | 0.26914     |
| Каскад 3 ( $m = 4$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.87218     | 0.5256      | 0.37605     | 0.31993     |
| Індекс Ксі-Бені      | 0.15687     | 0.4153      | 0.84699     | 1.1765      |
| Каскад 4 ( $m = 5$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.73909     | 0.45445     | 0.32428     | 0.27063     |
| Індекс Ксі-Бені      | 0.12985     | 0.30637     | 0.68584     | 1.0551      |

ним виявився другий каскад ( $m = 3$ ) з трьома кластерами і нейроном-переможцем із найменшим значенням параметру фаззіфікації  $\beta = 2$  (рис. 5.19). Така конфігурація є оптимальною відповідно до обох використовуваних індексів валідності – найменше значення Ксі-Бені індексу  $XB_j^{[m]}$  та найбільший коефіцієнт розбиття  $PC_j^{[m]}$ :

$$PC_1^{[2]} = 0.9009951,$$

$$XB_1^{[2]} = 0.03349166.$$

Лише одне спостереження у цьому датасеті (його позначено багряним квадратом) не належить жодному кластерові з ступінем більшим від 0.6. Індекси валідності нейронів системи наведені у таблиці 5.7.

Для набору даних з середньою вираженістю класів найліпшим виявився нейрон другого каскаду ( $m = 3$ ) і фаззіфікатором  $\beta = 3$  (таблиця 5.2).

Як показано на рис. 5.18, декілька спостережень у центрі (позначені багряними квадратами) можна віднести до 2 кластерів з відносно високим ступінем належності, проте більшість спостережень можна чітко розкластеризувати, що ілюструється високим значенням коефіцієнту розбиття, та дуже низьким

Ксі-Бені індексом:

$$PC_2^{[2]} = 0.9727868,$$

$$XB_2^{[2]} = 0.087474.$$

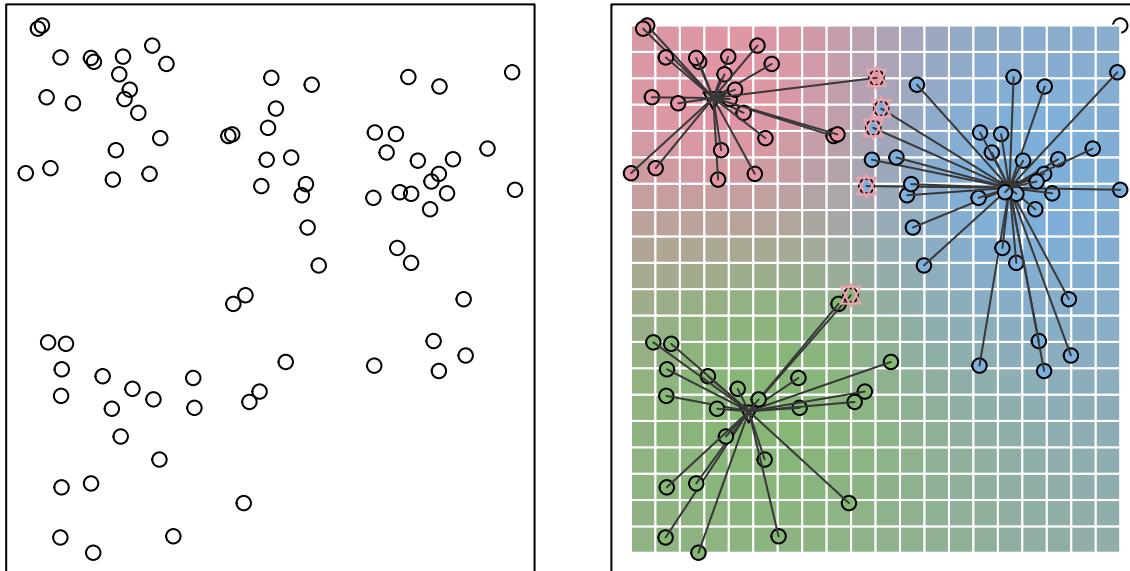


Рис. 5.18 – Набір даних з нечіткими межами класів (fuzzy dataset)

Для набору з найменш чіткими межами класів (таблиця 5.3), система обрала нейроном-переможцем вузол третього каскаду ( $m = 4$ ) з високим параметром фаззіфікації  $\beta = 4$ :

$$PC_3^{[3]} = 0.335525,$$

$$XB_3^{[3]} = 0.2128333.$$

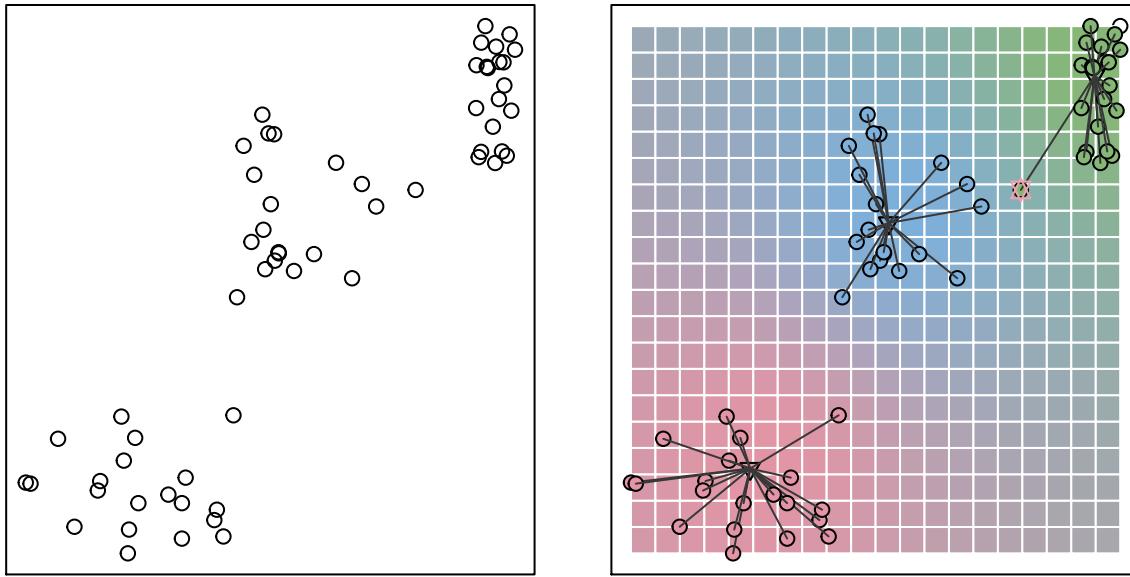


Рис. 5.19 – Набір даних з чітко вираженими класами (Crisp dataset)

Таблиця 5.8 – Індекси валідності (датасет 2)

| Каскад 1 ( $m = 2$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
|----------------------|-------------|-------------|-------------|-------------|
| Коефіцієнт розбиття  | 0.78414     | 0.58928     | 0.53853     | 0.52239     |
| Індекс Ксі-Бені      | 0.16668     | 0.30834     | 0.3745      | 0.38723     |
| Каскад 2 ( $m = 3$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 50084       | 0.71164     | 0.97275     | 0.4191      |
| Індекс Ксі-Бені      | 0.009751    | 0.031235    | 0.087474    | 0.1323      |
| Каскад 3 ( $m = 4$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.91888     | 0.47532     | 0.32777     | 0.28912     |
| Індекс Ксі-Бені      | 0.052563    | 0.1757      | 0.27516     | 0.33766     |
| Каскад 4 ( $m = 5$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.85618     | 0.34327     | 0.24778     | 0.22445     |
| Індекс Ксі-Бені      | 0.048316    | 0.19887     | 0.34307     | 0.41228     |
| Каскад 5 ( $m = 6$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.81295     | 0.30709     | 0.21636     | 0.19214     |
| Індекс Ксі-Бені      | 0.060896    | 0.19702     | 0.31393     | 0.38668     |

На рис. 5.20 спостереження, для яких ступінь належності до будь-якого кластеру не перевищує 0.6, позначені багряними квадратами. Як і очікувалося, для цього набору даних кількість таких спостережень значно вища від

попередніх датасетів з більш компактними та «чіткими» класами.

| Каскад 1 ( $m = 2$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
|----------------------|-------------|-------------|-------------|-------------|
| Коефіцієнт розбиття  | 0.85094     | 0.71415     | 0.61734     | 0.57085     |
| Індекс Ксі-Бені      | 0.10584     | 0.11462     | 0.13797     | 0.16101     |
| Каскад 2 ( $m = 3$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.61668     | 0.42848     | 0.37779     | 0.35884     |
| Індекс Ксі-Бені      | 0.1754      | 0.20364     | 0.22364     | 0.23995     |
| Каскад 3 ( $m = 4$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.33458     | 0.44082     | 0.79405     | 0.29615     |
| Індекс Ксі-Бені      | 0.20989     | 0.129       | 0.051039    | 0.26282     |
| Каскад 4 ( $m = 5$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.50244     | 0.33067     | 0.26029     | 0.23318     |
| Індекс Ксі-Бені      | 0.37268     | 0.61417     | 0.79695     | 0.93626     |
| Каскад 5 ( $m = 6$ ) | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| Коефіцієнт розбиття  | 0.53279     | 0.29731     | 0.22648     | 0.19858     |
| Індекс Ксі-Бені      | 0.27407     | 0.47298     | 0.60569     | 0.70716     |

Таблиця 5.9 – Індекси валідності (датасет 3)

Для очності у всіх наведених рисунках кольором позначені не тільки розкластеровані спостереження і центри кластерів, а й задній план (фон) малюнків, що дозволяє візуально визначити, до якого кластеру система віднесла б нові спостереження. Не дивно, що, тоді як для перших двох датасетів важко визначити домінуючий колір, оскільки кластери їх спостережень більш менш компактні та явно виражені, для останнього набору даних домінуючий колір – сірий, сформований кольорами усіх кластерів, що ілюструє великий ступінь перекриття класів і, відповідно, високе значення оптимального параметру фаззіфікації  $\beta$ , що обрала система.

Ця низка експериментів проілюструвала як важливо вірно визначати параметр фаззіфікації, оптимальне значення якого у випадку обробляння даних у послідовному режимі з високою вирогідністю змінюється у часі, а саме здатність визначати оптимальне значення цього параметру в онлайн режимі є відмінною особливістю запропонованої самонавчанної нейро-системи.

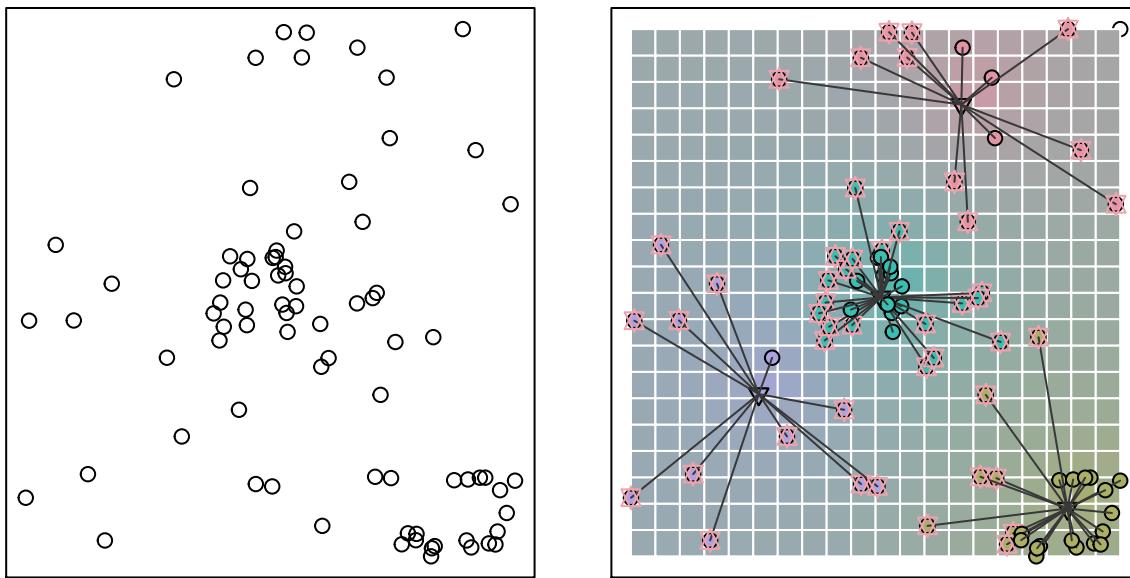


Рис. 5.20 – Набір даних з класами, що перетинаються (extra fuzzy dataset)

Наступна низка експериментів була проведена на наборі даних «Іриси Фішера» (Fisher's Iris data set).

Це багатовимірний датасет для задачі класифікації, на прикладі якого англійський статистик та біолог Рональд Фішер в 1936 році продемонстрував роботу розробленого ним методу дискримінантного аналізу. Іноді його також називають «Ірисами Андерсона» (через те, що дані були зібрані американським ботаніком Едгаром Андерсоном). Цей набір даних став класичним і часто використовується в літературі для ілюстрації роботи різних статистичних алгоритмів.

Проте цей датасет рідко використовується у кластерному аналізі, адже межі класів «*Virginica*» та «*Versicolor*» не можна чітко визначити, ґрунтуючись на даних, що їх використовував Фішер (що легко продемонструвати за допомогою ієрархічного кластерування, рис. 5.21). Саме цим і цікавий для нас цей набір даних: коли класичні методи чіткого кластерного аналізу не справляються з задачею, може стати у нагоді система, що реалізує нечітке кластерування зі змінним параметром фаззифікації та кількістю кластерів. Для більшості методів кластерного аналізу, зокрема для методу нечітких середніх (fuzzy c-means), необхідно заздалегідь задати кількість кластерів, і

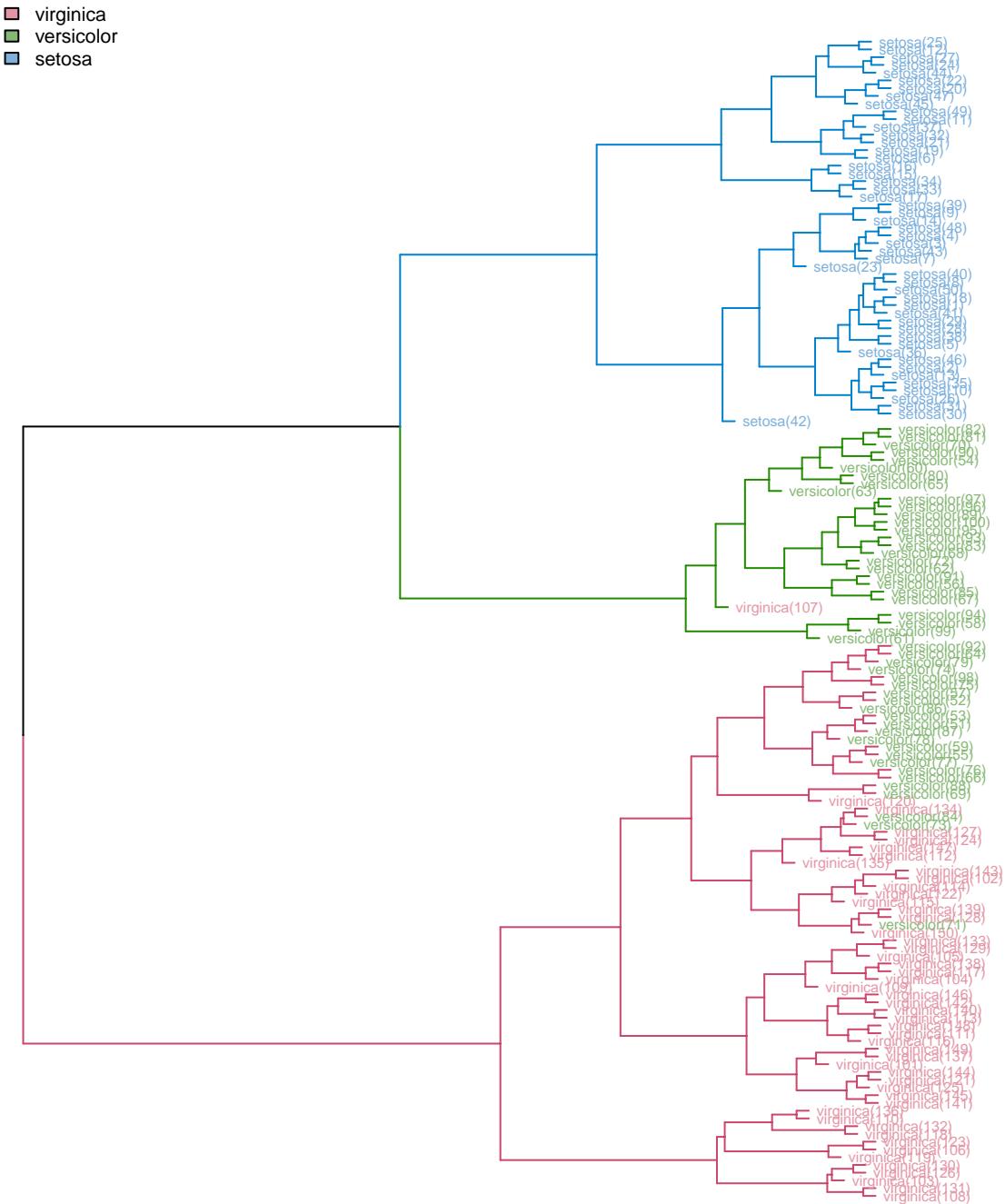


Рис. 5.21 – Ієрархічне класерування датасету «Іриси Фішера»

очевидним рішенням є прийняти  $m = 3$ , адже маємо три класи: Iris Virginica, Iris Versicolor та Iris Setosa (рис. 5.22).

Точність кластерування за допомогою методу нечітких середніх за таких умов ( $m = 3$ ,  $\beta = 2$ ) рідко перевищує 83% (таблиця 5.4). (Оскільки для обраного датасету існують мітки з вірною класифікацією, ефективність кластеризації вимірювалася у відсотках точності щодо еталонного розбиття після де-

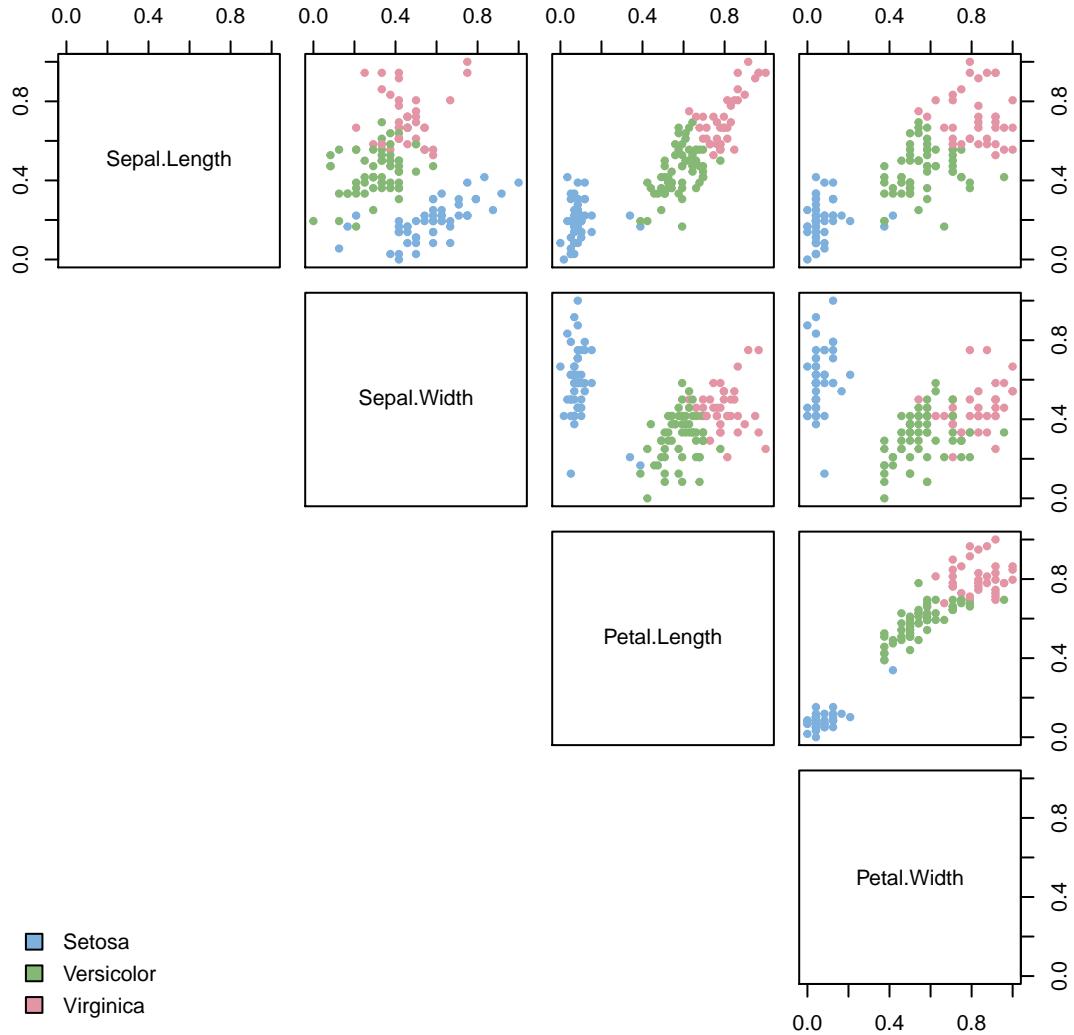


Рис. 5.22 – Розкластерований датасет «Іриси Фішера» при  $m = 3$ ,  $\beta = 2$  (Точність кластерування – 96%)

фаззіфікації.) Проте, якщо не обмежувати пропоновану систему у кількості кластерів (система ініціалізується інтервалом допустимих значень  $m$  (кількість кластерів) та параметру фаззифікації  $\beta$ ), вельми цікавими є результати кластерування нейронів кожного з каскадів.

У таблиці 5.5 наведена точність розбиття даних, коли  $m \gg 3$  кластерів відповідно. Варто зазначити, що нейрони у пулі кожного каскаду реалізують метод нечітких середніх зі змінним значенням фазифікатору, а отже є чутливими до довільно ініціалізованих цетрах кластерів, тому у таблицях наведені середня, мінімальна та максимальна точності кластерування (після дефаззі-

|     | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
|-----|-------------|-------------|-------------|-------------|
| avg | 0.8313073   | 0.8741245   | 0.8709475   | 0.8888124   |
| min | 0.7859722   | 0.7533766   | 0.6615745   | 0.7656498   |
| max | 0.8534013   | 0.9166667   | 0.935051    | 0.9604701   |

Таблиця 5.10 – Точність кластерування при  $m = 3$

фікації).

| $m = 7$  | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
|----------|-------------|-------------|-------------|-------------|
| avg      | 0.8972948   | 0.9150268   | 0.9242503   | 0.9178207   |
| min      | 0.8536056   | 0.8461905   | 0.8723182   | 0.8600289   |
| max      | 0.9621849   | 0.9736172   | 0.9810146   | 0.9663462   |
| $m = 8$  | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| avg      | 0.9065560   | 0.9296311   | 0.9243606   | 0.9248976   |
| min      | 0.8217056   | 0.8562179   | 0.8577202   | 0.8590278   |
| max      | 0.9474588   | 0.9789402   | 0.9848214   | 0.9747899   |
| $m = 9$  | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| avg      | 0.9258154   | 0.9282887   | 0.9308971   | 0.9229753   |
| min      | 0.8689921   | 0.8270525   | 0.8684641   | 0.8556390   |
| max      | 0.9849170   | 0.9806397   | 0.9664112   | 0.9748284   |
| $m = 10$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| avg      | 0.9213191   | 0.9285106   | 0.9332528   | 0.9282907   |
| min      | 0.8663370   | 0.8722271   | 0.8652272   | 0.8766667   |
| max      | 0.9663420   | 0.9838095   | 0.9723656   | 0.9756335   |
| $m = 11$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| avg      | 0.9295315   | 0.9408977   | 0.9317242   | 0.9295800   |
| min      | 0.8520268   | 0.8964924   | 0.8890781   | 0.8788656   |
| max      | 0.9716166   | 0.9848485   | 0.9704892   | 0.9798627   |
| $m = 12$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| avg      | 0.9349407   | 0.9433244   | 0.9337934   | 0.9306632   |
| min      | 0.8815133   | 0.8949802   | 0.8798160   | 0.8486111   |
| max      | 0.9795274   | 0.9783497   | 0.9630952   | 0.9772727   |
| $m = 13$ | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
| avg      | 0.9420998   | 0.9398127   | 0.9375204   | 0.9357708   |
| min      | 0.8823175   | 0.8614025   | 0.8882479   | 0.8828348   |
| max      | 0.9807518   | 0.9788034   | 0.9753452   | 0.9748873   |

Таблиця 5.11 – Точність кластерування для  $m \in [7, 13]$ ,  $\beta \in [2, 5]$

На рис 5.23 зображено залежність точності кластерування від кількості кластерів. Цікаво, що при, здавалося б, очевидному рішенні обрати кількість кластерів рівною трьом, отримуємо чиненайгіршу точність кластерування (при  $\beta = 2$ ) після дефаззіфікації щодо еталонного розбиття (Для порівняння

на рис. 5.24 та рис. 5.25 наведені розбиття, що їх запропонували нейрони-переможці деяких каскадів, де  $m \gg 3$ ).

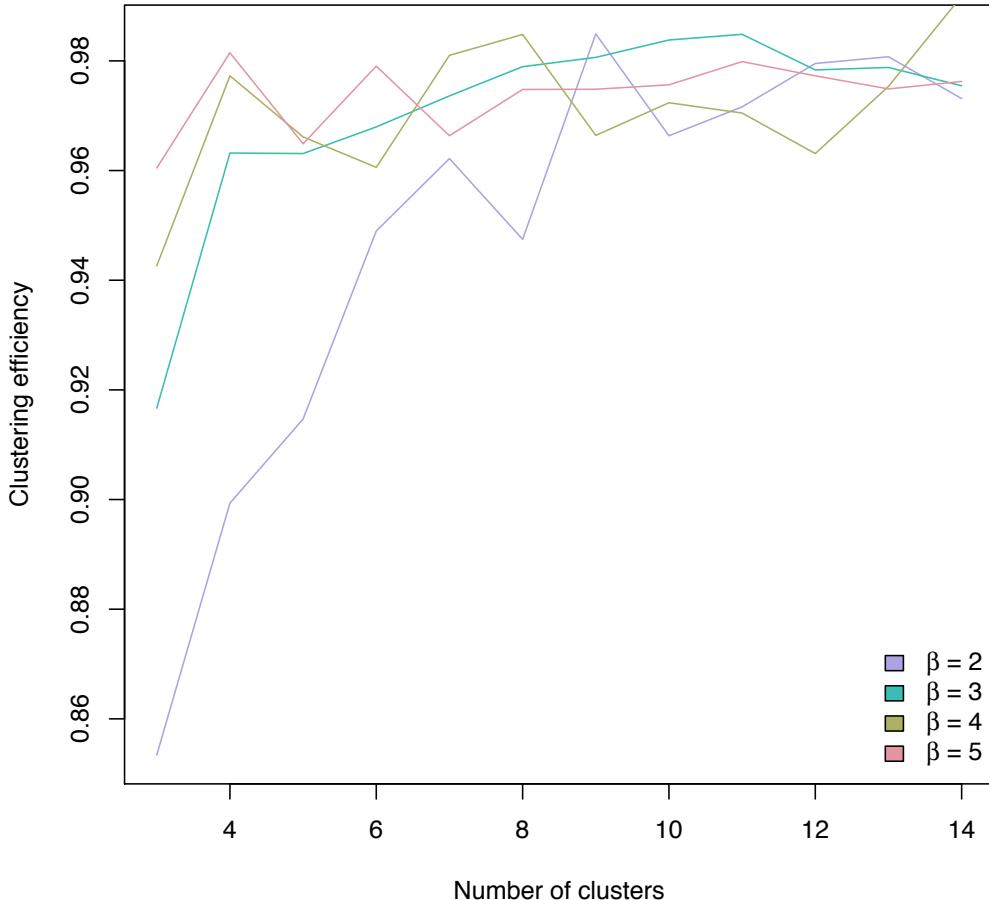


Рис. 5.23 – Точність кластерування від кількості кластерів та параметру фаз-зіфікazї

Цьому легко знайти пояснення, адже метод нечітких  $k$ -середніх (а саме цей метод у цьому експерименті реалізовують вузли пулів кожного каскаду) добре розпізнає кластери лише гіперсферичної форми. Проте кластер довільної (негіперсферичної) форми, можна розбити на декілька гіперсферичних підкластерів, що й відбувається у каскадах, де  $m > 3$ , що пропонують розбиття на дрібні кластери. На рисунках 5.26 та 5.27 наведені розбиття деяких каскадів, де кількість кластерів більша від кількості класів еталонної вибірки; тут можна побачити, що декілька кластерів, що після дефазифікації будуть віднесені до одного класу, наприклад, Iris Virginica розташовані поруч один з одним, тобто є складовими більшого кластеру негіперсферичної форми.

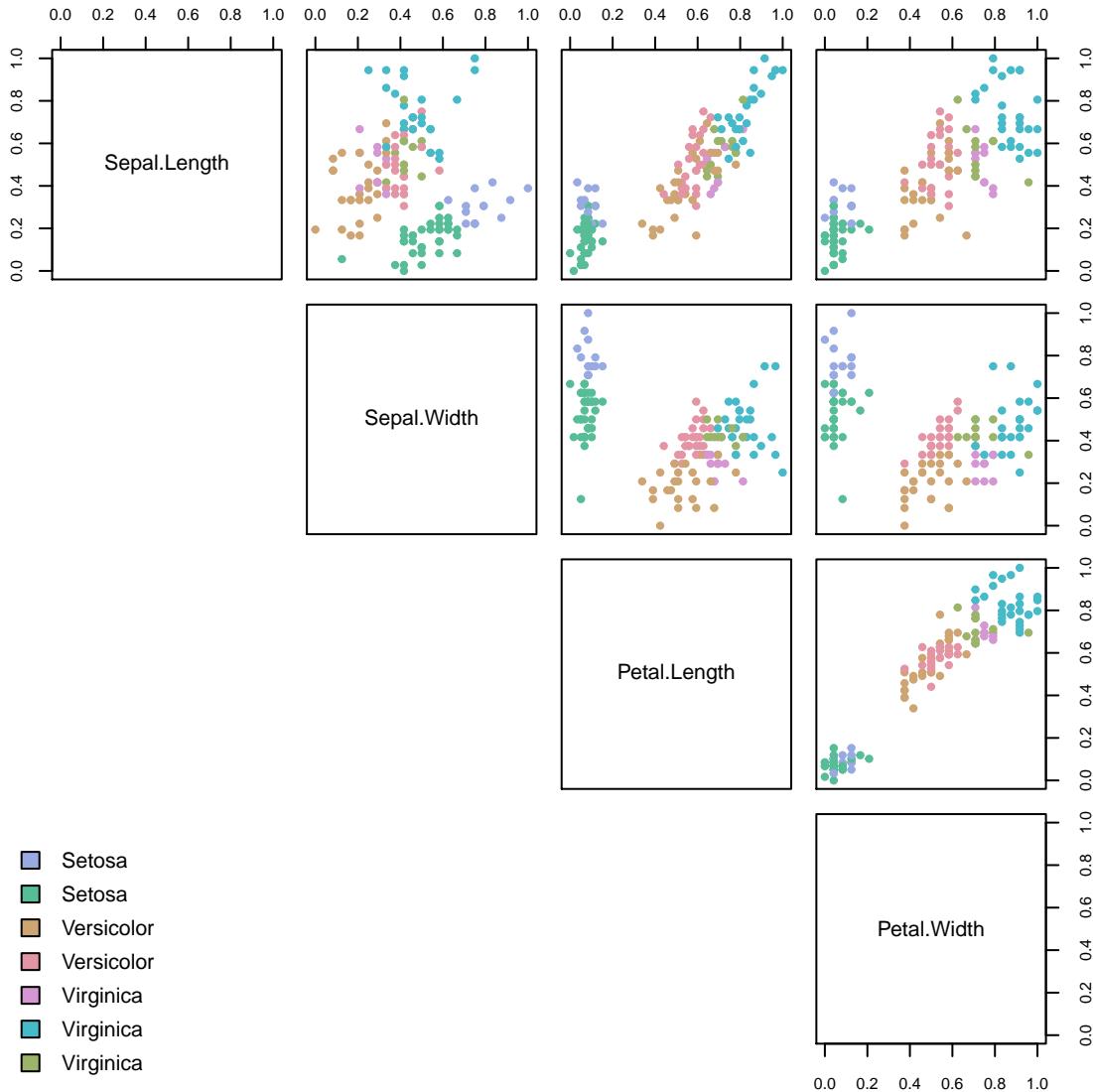


Рис. 5.24 – Розкластерований датасет «Іриси Фішера» при  $m = 7$ ,  $\beta = 5$  (Точність кластерування  $\approx 93\%$ )

Таким чином, видається доречним, навіть у випадку, коли відоме еталонне розбиття датасету, дозволити системі обрати кінцеву кількість кластерів самостійно, особливо у випадку, коли вузли системи реалізують одинаковий метод кластерування.

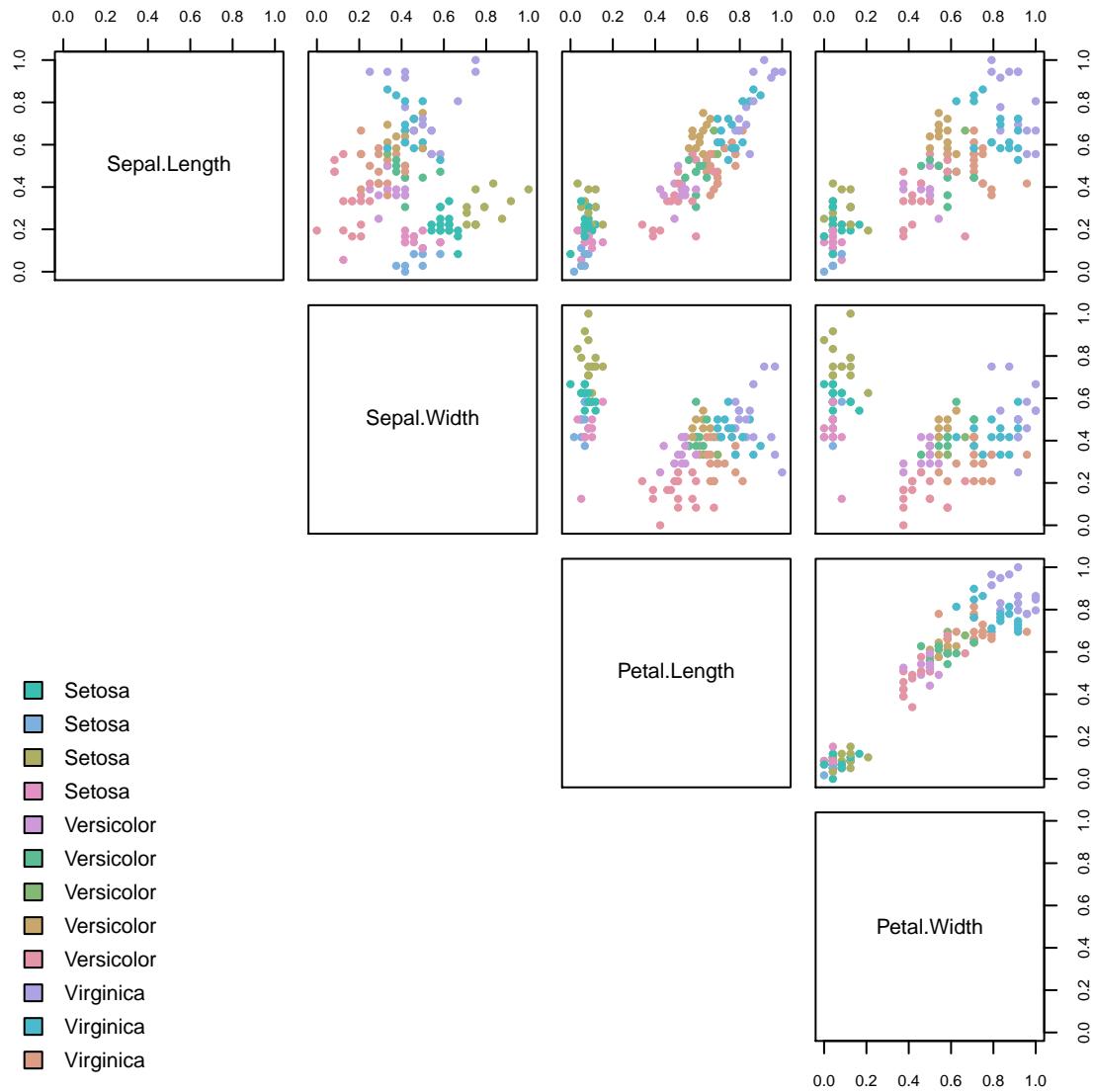
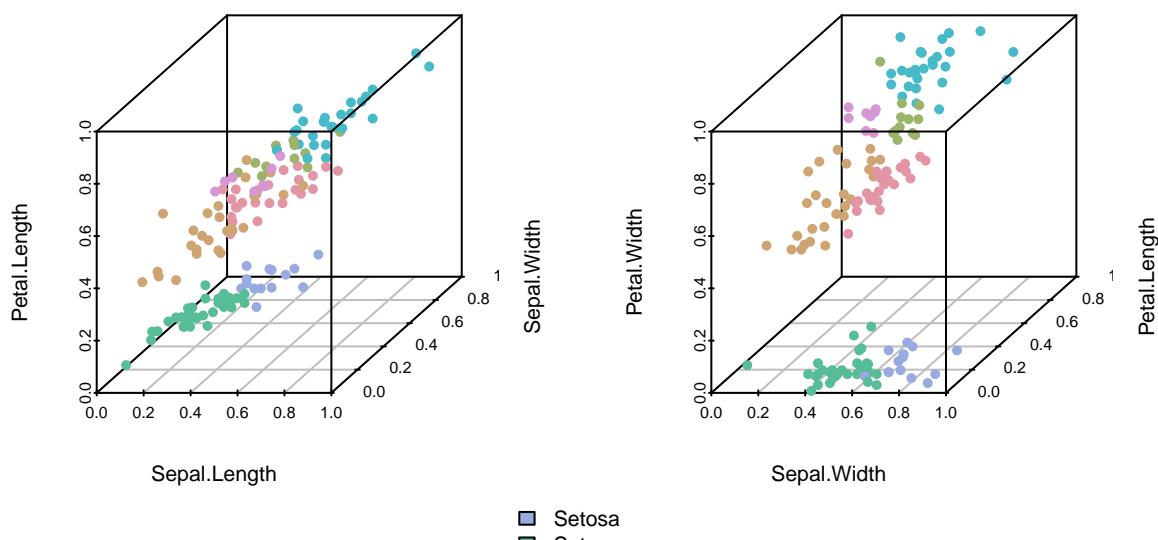


Рис. 5.25 – Розкластерований датасет «Іриси Фішера» при  $m = 12$ ,  $\beta = 4$   
(Точність кластерування  $\approx 96\%$ )



*Таблиця 5.12 – Точність кластерування при  $m = 14$*

|     | $\beta = 2$ | $\beta = 3$ | $\beta = 4$ | $\beta = 5$ |
|-----|-------------|-------------|-------------|-------------|
| avg | 0.9369168   | 0.9448829   | 0.9383179   | 0.9403416   |
| min | 0.8847819   | 0.8953380   | 0.8787879   | 0.9069805   |
| max | 0.9731262   | 0.9754579   | 0.9918301   | 0.9762515   |

Варто зауважити, що у цьому випадку для визначення локально оптимального розбиття доцільно використовувати модифіковані індекси валідності, чи такі, що не залежать від відстані центрів кластерів, наприклад ті, що ґрунтуються на щільності (density-based).

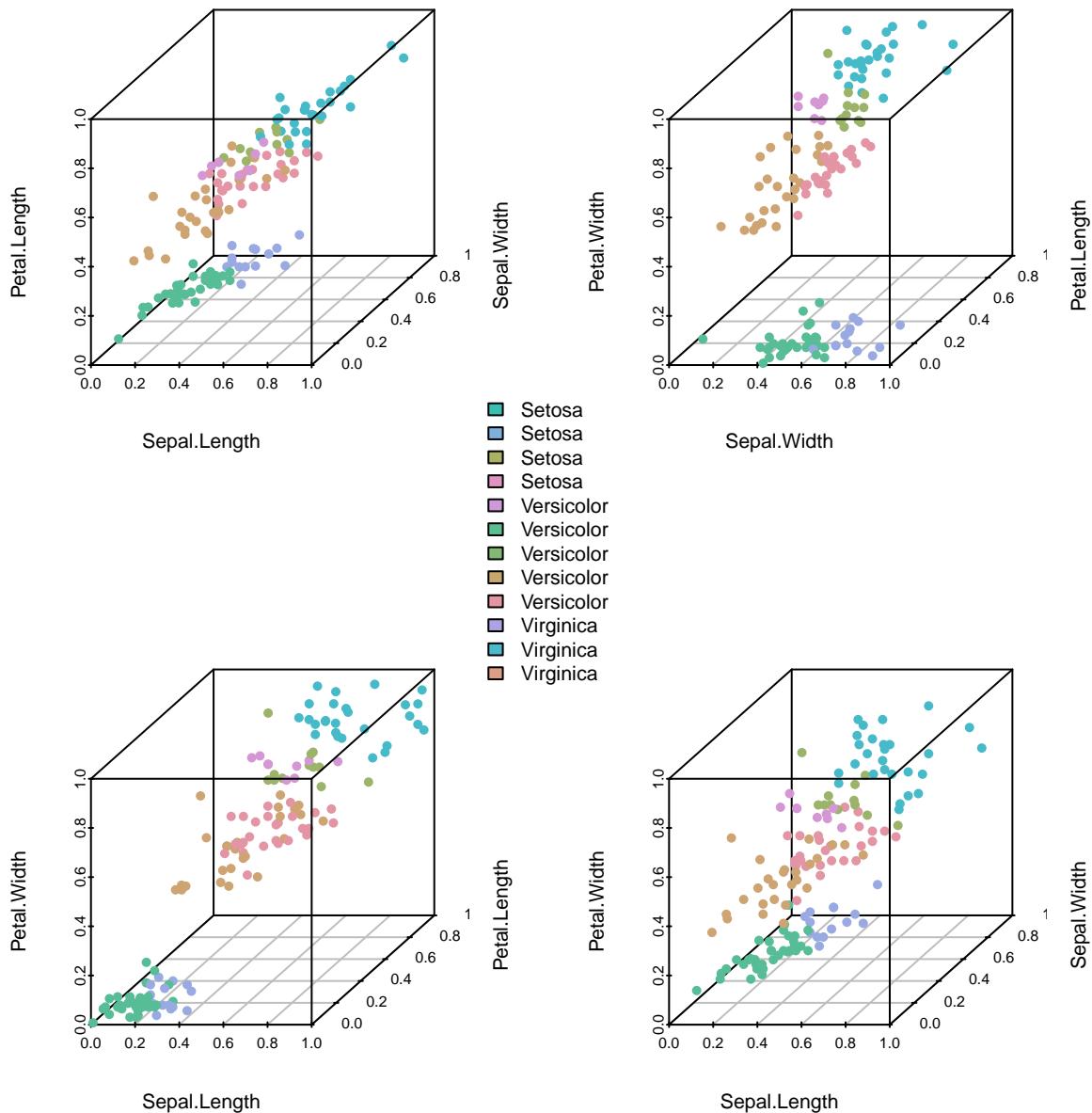


Рис. 5.27 – Розкластерований датасет «Іриси Фішера» при  $m = 12$ ,  $\beta = 4$   
(Точність кластерування  $\approx 96\%$ )

Наступну серію експериментів було проведено на датасеті «Знання студентів про електричні машини постійного струму».

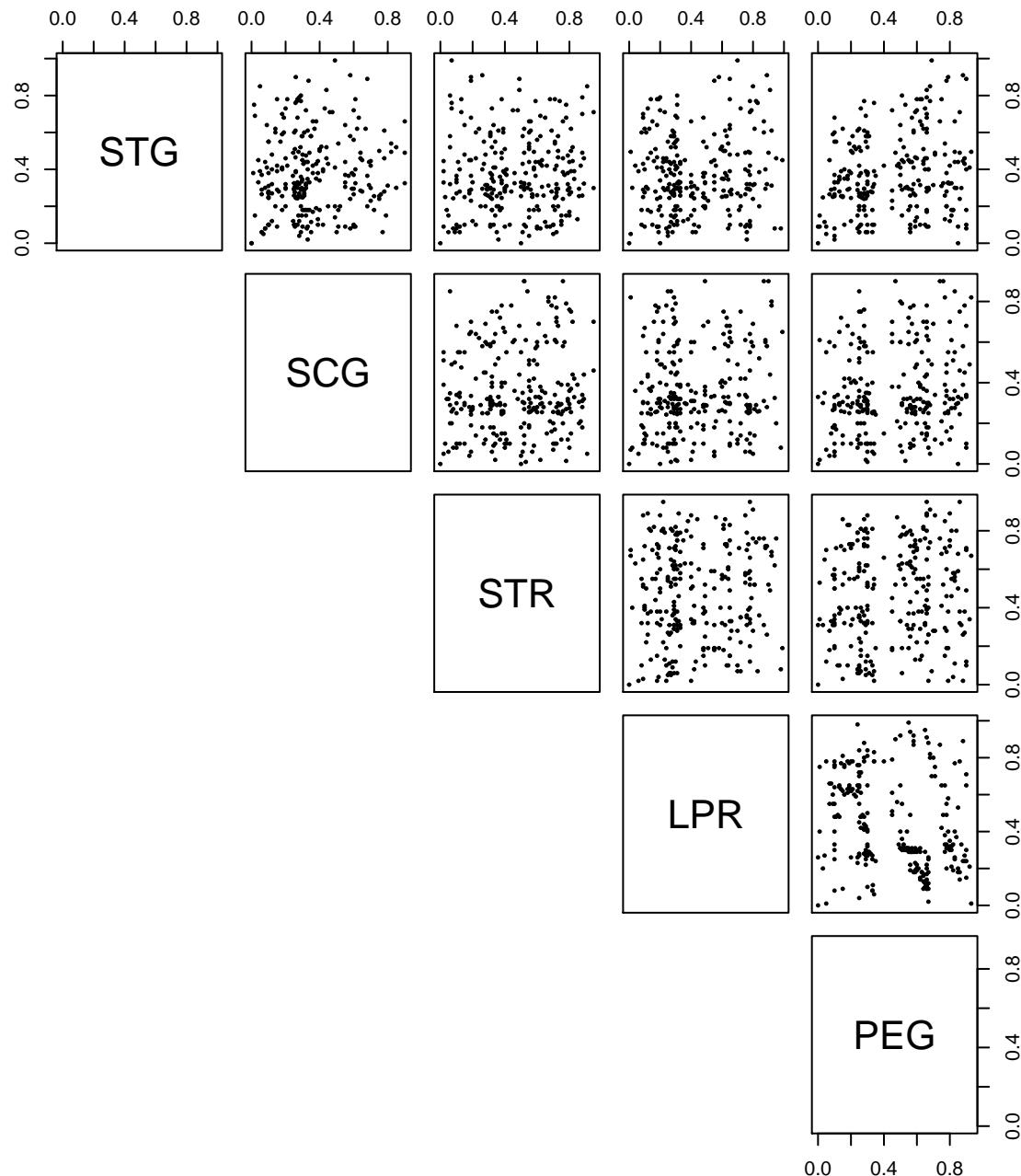


Рис. 5.28 – Датасет «Знання студентів про електричні машини постійного струму»

Цей датасет було додано до UCI репозиторію у 2013 році, він містить 403 паттерни, кожен з п'ятьма атрибутами:

1. STG: кількість часу, що його витратив(витратила) студент(ка) на навчання цільового матеріалу,
2. SCG: Кількість повторювань навчання цільового матеріалу студен-

том(студенткою),

3. STR: Кількість часу, що його використав(використала) студент(ка) на навчання матеріалу, пов'язаного з цільовим матеріалом,
4. LPR: Оцінка, що її отримав(отримала) студент(ка) на іспиті з предмету, пов'язаного з цільовим предметом,
5. PEG: Оцінка, що її отримав(отримала) студент(ка) на іспиті з цільового предмету,

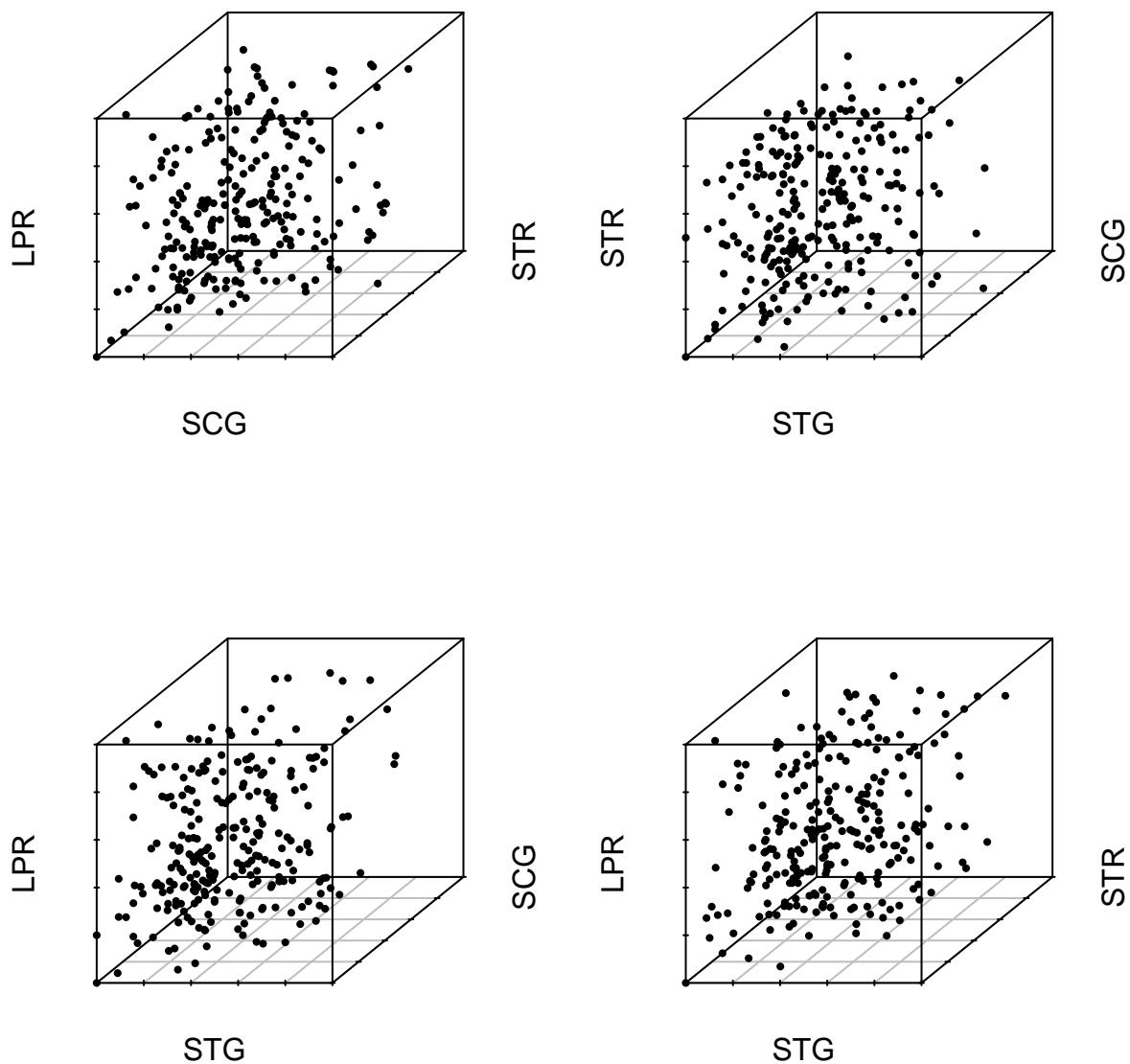


Рис. 5.29 – Датасет «Знання студентів про електричні машини постійного струму»

Попарні графіки атрибутів наведено на рис. 5.28 та у тримірному просторі на рис. 5.29.

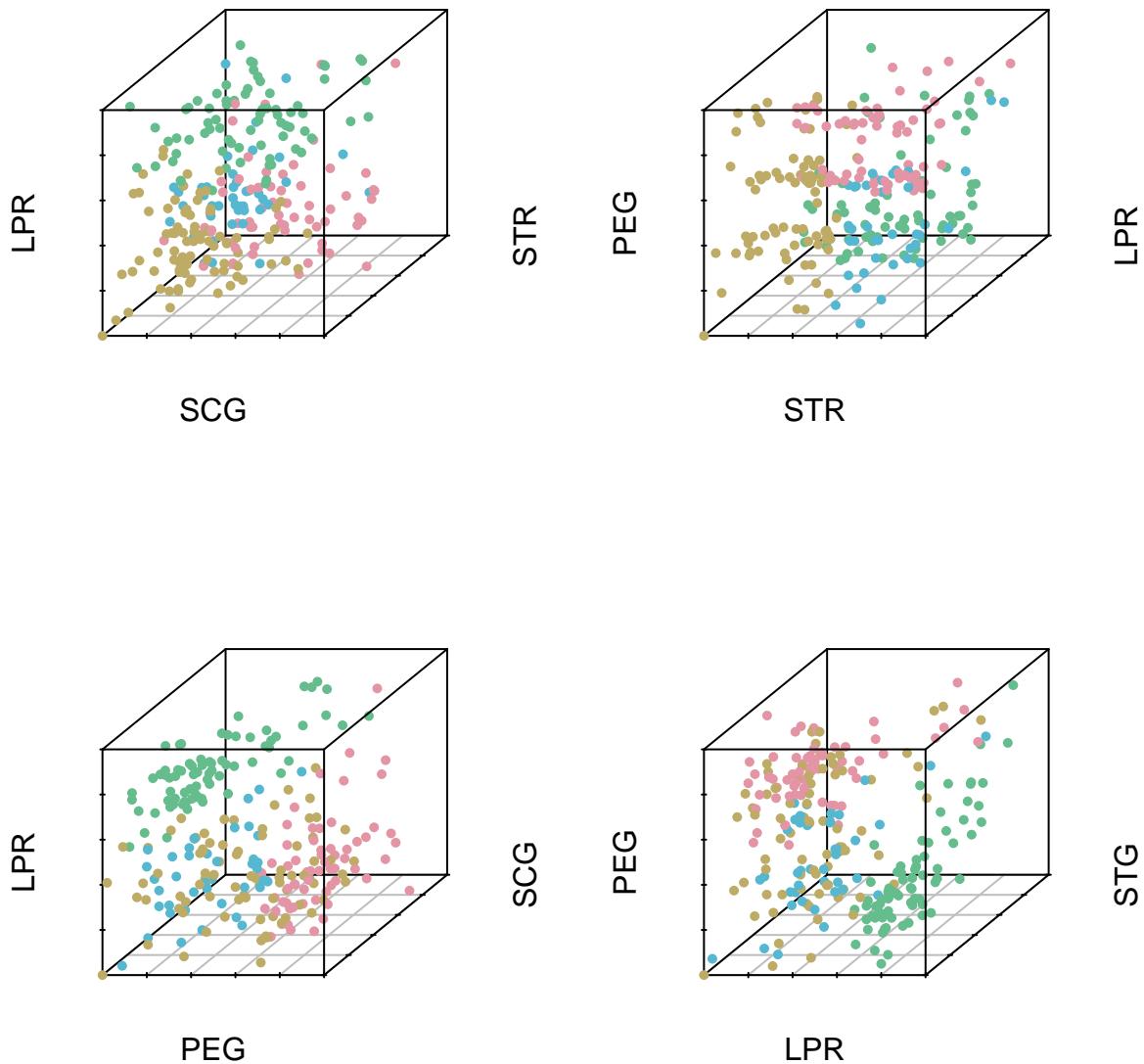


Рис. 5.30 – Датасет «Знання студентів про електричні машини постійного струму»

Для цього експерименту нейрони-узагальнювачі керувалися рекурентним Kci-Бені Індексом при визначенні локально-оптимального нейрона (з найлішшим параметром фаззіфікації) та каскаду (з оптимальною кількістю класстерів).

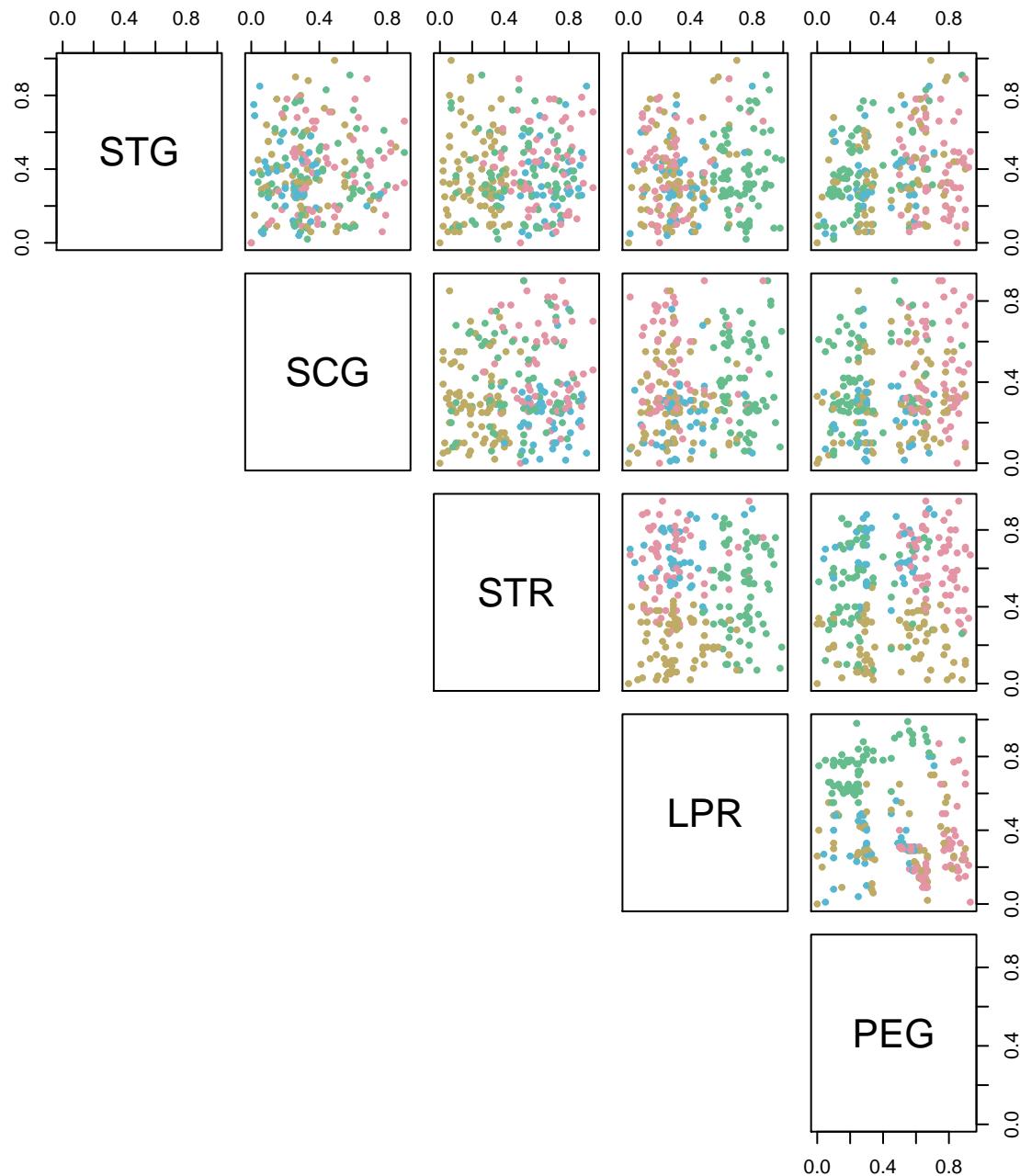


Рис. 5.31 – Датасет «Знання студентів про електричні машини постійного струму»

Оптимальне розбиття, що його наведено на рис.5.30 та на рис.5.31, надав другий нейрон третього каскаду з  $m = 4$ ,  $\beta = 2$  з коефіцієнтом Ксі-Бені 0.38155.

## 5.5. Розв'язування практичних задач за допомогою розробленої самонавчанної гібридної каскадної системи, що еволюціонує

Проблема здорового харчування - одна з найактуальніших у наші дні. Повноцінне харчування передбачає споживання достатньої кількості білків, жирів, вуглеводів, вітамінів, макро- і мікроелементів для нормального функціонування організму в цілому. Багато хвороб шлунково-кишкового тракту «молодіють» - це гастрити, виразкова хвороба шлунка і різні порушення обміну речовин. Фізичне здоров'я, стан імунітету, довголіття, психічна гармонія - все це безпосередньо пов'язано з проблемою здорового харчування людини. Для студентів проблема харчування стоїть особливо гостро. У зв'язку з браком часу у студентів немає можливості дотримуватися правильного режиму прийомів їжі в кількості 3-4 разів. Також характерний в основному сидячий спосіб життя - гіподинамія. У поєднанні з незбалансованим раціоном харчування це згубно впливає на організм і його стан. Звісно, вирішення проблеми здорового харчування потребує комплексного підходу, проте інформованість - невід'ємна складова правильного підбору раціону здорового харчування. Насьогодні нескладно знайти інформацію щодо рекомендованої денної кількості калорій, білків, жирів та вуглеводів, проте важко дати оцінку конкретному прийому їжі, наприклад, придбаному у їdalnі, де немає етикеток з такою інформацією. Мобільний додаток «Spoon app» може стати у нагоді, коли користувач прагне бути проінформованим щодо поживності конкретної трапези, роблячи аналіз світлині тарілки з їжею. Вхідними даними мобільного додатку є світлина, що її користувач має зробити таким чином, аби тарілка знаходилася у центрі, а також тип тарілки (звичайна, глибока, дуже глибока) аби на виході додаток мав обґрунтовану кількість калорій та поживність порції, що була зображена на світлині.

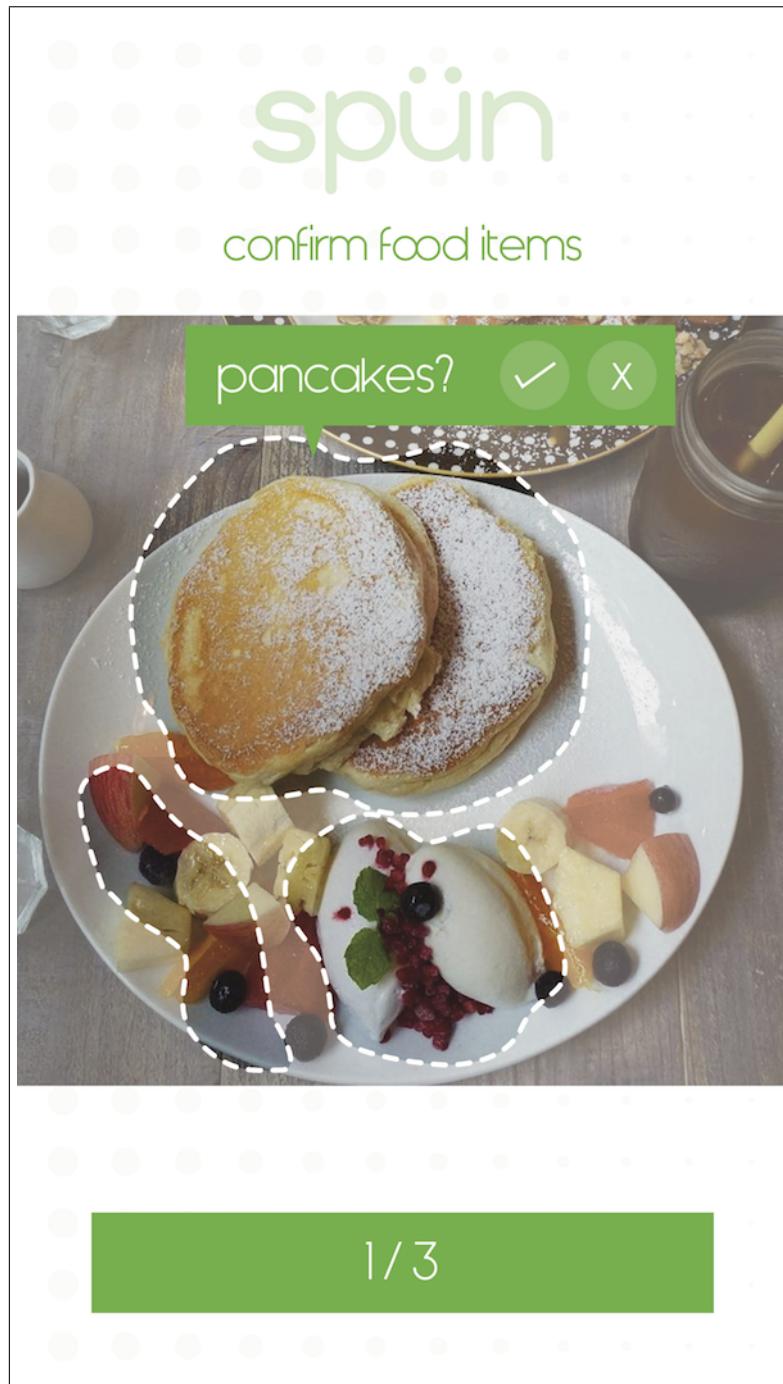


Рис. 5.32 – Другий етап аналізу світлини з прийомом їжі мобільним додатком Spoon App (навчання з підкріпленням)

Аналіз світлини можна розбити на декілька етапів:

1. кластерування даних зображених на світлині (відбувається на стороні клієнту)
2. ідентифікація окремих складових прийому їжі: класифікація кожного зображення, після розбиття світлини на кластери на першому етапі

(відбувається на серверній стороні), визначення типу продукту за допомогою бази даних, що знаходиться на сервері, та подальше визначені кількості калорій, співвідношення білкі, жирів та вуглеводів.

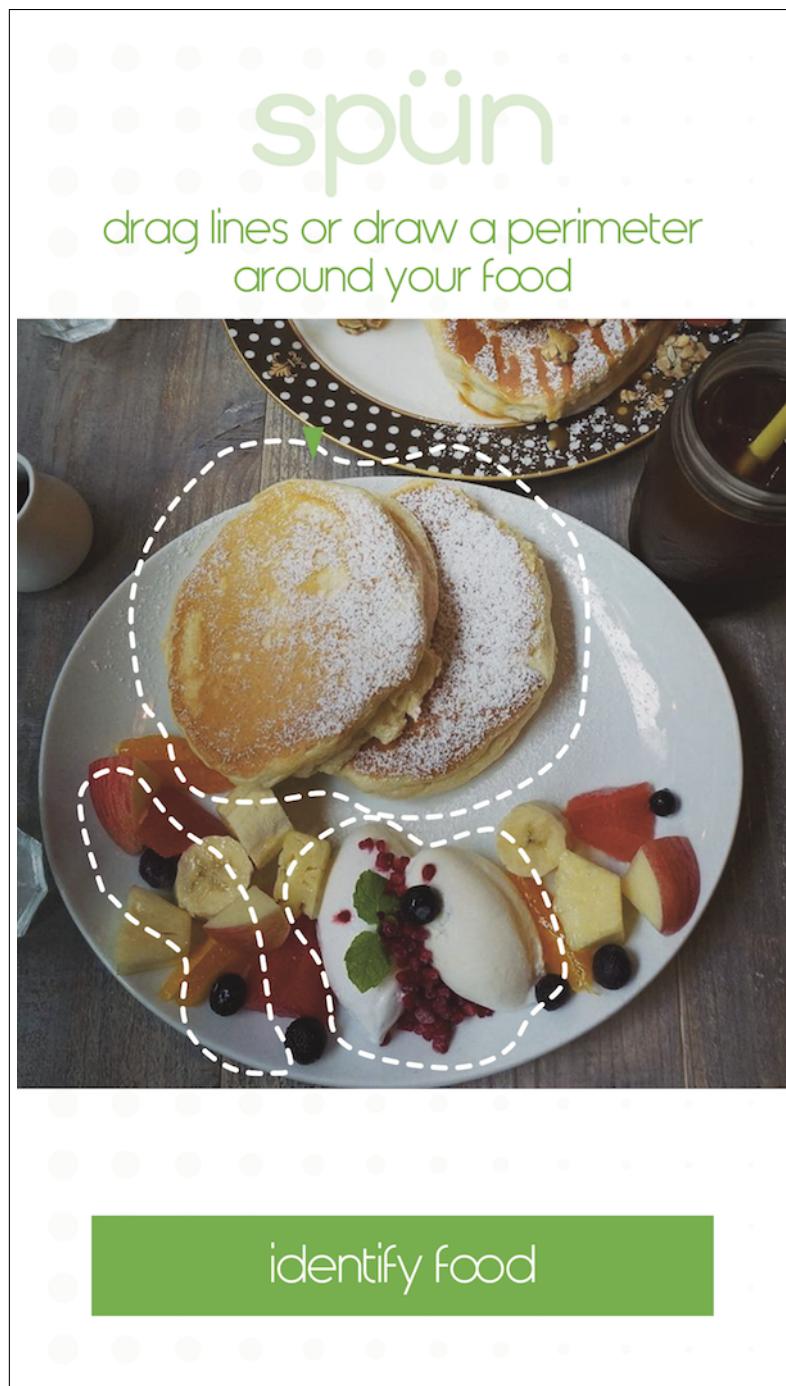


Рис. 5.33 – Перший етап аналізу світлини з прийомом їжі мобільним додатком Spoon App (кластерування за умови невизначеності щодо кількості кластерів)

Другий етап у певному сенсі є навчанням з підкріпленням, адже користу-

вачеві пропонується підтвердити чи скорегувати кінцевий результат класифікації, як зображене на рис. 5.32. Проте більш цікавим нам видається саме перший етап аналізу світлини, і пропонована гібридна самонавчана система використовується саме на цьому етапі, адже вона краще від інших існуючих систем задовольняє умовам, що їх було висунуто на етапі формування технічних вимог до програмного забезпечення (Software Requirements Specifications):

- кластерування має проходити за умови невизначеності щодо кількості кластерів,
- оскільки кластерування відбувається на стороні клієнту, важливо мінімізувати обслуговальну складність алгоритму, а отже перевага надається методам послідовного кластерування.

По завершенні аналізу на першому етапі мобільний додаток попронує розбиття світлини на  $t$  кластерів, як показано на рис. 5.33. Варто зауважити, що межі кластерів, що їх визначила самонавчана система, дещо відрізняються від тих, що зображені на рис. 5.33, для того, щоб користувачеві було зручніше візуально сприймати розбиття світлини на кластери, пунктирні лінії, що зображують межі кластерів, на декілька міліметрів віддалені від меж дійсних кластерів, проте на сервер для подальшої класифікації відправляється світлина з розбиттям, що запропонувала система. На цьому етапі користувач може скорегувати розбиття, перетягнувши пунктирну лінію меж кластерів, чи зовсім видалити пропонований кластер. Хоча навчання з підкріпленим у прямому сенсі, відбувається лише на етапі класифікації (база даних на сервері оновлюється, коли користувач корегує результат класифікації), а на цьому етапі маємо саме навчання без учителя, це все ж таки дає змогу у певному сенсі дати оцінку кластеруванню системи: вважаємо кластерування успішним, якщо користувач не робив жодних змін до пропонованого розбиття, та неуспішним, коли розбиття було скореговане. Після бета тестування мобільного додатку маємо наступні результати:

Цікавим видається перебіг подій, коли користувач видалив один кластер,

|  |     |
|--|-----|
| кількість та межі кластерів залишилися незмінними  | 608 |
| межі кластеров було дещо змінено користувачем,<br>проте кількість кластерів залишилась незмінною | 61  |
| користувач змінив кількість кластерів<br>та межі пропонованих кластерів                          | 52  |
| користувач видалив кластер(и), межі інших пропонованих<br>кластерів лишилися незмінні            | 29  |

*Таблиця 5.13 – Результати бета тестування первого етапу (кластерування за умови невизначеності щодо кількості кластерів) аналізу світлин мобільним додатком «Spoon App»*

проте залишим межі інших кластерів незмінними, у такому випадку, як зазначалося вище, вважається що кластерування не було успішним. Проте подальший аналіз показав, що у 90% таких випадків користувач залишив на тарілці неїстівний предмет (виделку, ложку тощо), і хоча система вірно відвела йому окремий кластер, не має сенсу класифікувати його та визначати калорійність цього предмету, тому логічно, що користувач видалив його на цьому етапі. У 10% випадків, як показав аналіз світлин з початковим кластеруванням, що запропонувала самонавчана система, та світлин після корегування користувачем, користувач свідомо видаляв складову прийому їжі, зазвичай найменш корисну (тітєчко, шоколад тощо). Тому видається доцільним ці 4% світлин також віднести до таких, що були вірно розкластеровані системою (яка, проте, не бере до уваги людський фактор).

## **5.6. Розв'язування практичних задач за допомогою розробленої гібридної каскадної нейро-фаззі мережі, що еволюціонує, з оптимізацією пулу нейронів**

Пропоновану гібридну каскадну нейро-фаззі мережу, що еволюціонує, було використано для вирішення задачі прогнозування витрат нормогодин на

ремонті роботи віzkів вагонів 61-425, 61-181, 47Д та 47К у ТОВ «Харківський вагонобудівний завод». Коректне формування й оптимізація виробничої програми є досить нелегким процесом, але дуже важливим. Незаплановані втрати фінансових або інших ресурсів можуть поставити під загрозу саме існування підприємства. До того ж через велику собівартість одиниці продукції будь-яки накладки у виробництві товару можуть серйозно вдарити по стабільності заводу в цілому. Все це робить оптимізацію виробничої програми першочерговим завданням планового відділу підприємства. Рома, допиши пожалуйста, про важливість ремонта тележек, и особенно важность прогнозирования этого дела

У таблиці 5.14 наведені типи та послідовність ремонтних робіт для віzkів вагонів 61-425, 61-181, 47Д та 47К. Для формування вектору вхідних даних були залучені експерти, що надали оцінку складності тому чи іншому типу ремонтних робіт, якщо це необхідно. Вочевидь, складність та тривалість деяких робіт не залежить від стану віzку на момент початку ремонтного циклу, наприклад, виконання «Сушки атмосферної відремонтованих деталей і складових одиниць віzку 0.162-04.20.00: 000 (1) після грунтування» або «Випробування гасителів коливання 45.30.045» займає приблизно однакову кількість нормогодин, а отже не потребує експертної оцінки (якщо такі типи робіт необхідні, оцінка їх складності завжди дорівнює одиниці). Проте витрата нормогодин на більшість типів ремонтних робіт (наприклад, на «Ремонт деталей підвісок 30.30.025» чи «Ремонт важелів гальмівної важільної передачі») суттєво варіюється в залежності від характеру та ступеню отриманих пошкоджень і потребує експертної оцінки (оценка надається в інтервалі  $(0, 1]$ , де 1 - найвища складність роботи).

Для вирішення поставленого завдання було обрано саме пропоновану гідбридну нейронну мережу, адже важливим аспектом є можливість системи працювати з датасетом, де кількість паттернів нижча від кількості вхідних параметрів. Пропоновану гібридну нейро-фаззі мережу було навчано на тренувальному датасеті з 62-х паттернів, кожен з яких містить 69 атрибутив

та, на момент впровадження, перевірено на тестовому датасеті, що містив 20 паттернів. Система продемонструвала високу точність прогнозу: RMSE = 0.08506766, SMAPE = 0.2545574, абсолютна похибка не перевищує 27 нормогодин (при середній витраті у 1150 нормогодин для повного циклу ремонтних робіт та 600 нормогодин для часткового).

*Таблиця 5.14 – Типи ремонтних робіт для віzkів вагонів 61-425, 61-181, 47Д та 47К*

| Код роботи | Назва роботи   | Попередні роботи         | Подальші роботи                                      |
|------------|--|--------------------------|--|
| 109.01.001 | Розбирання віzkів 0.162-04.20.00 : 000 (1)                         |                          | 109.05.046<br>103.01.002                             |
| 109.01.002 | Розбирання віzkів 0.114-04.10.00 : 000 (1)                         |                          | 109.05.047<br>103.01.002                             |
| 109.05.046 | Попередня дефектація вузлів віzку 0.162-04.20.00 : 000 (1)         | 109.01.001               | 109.05.047<br>109.05.049<br>109.01.004               |
| 109.05.047 | Попередня дефектація вузлів віzку 0.114-04.10.00 : 000 (1)         | 109.05.046<br>109.01.002 | 109.05.050<br>109.01.004                             |
| 109.05.049 | Абразивне очищення деталей і вузлів віzку 0.162-04.20.00 : 000 (1) | 109.05.046               | 109.01.022<br>109.01.035<br>109.01.036<br>109.01.003 |
| 109.05.050 | Абразивне очищення деталей і вузлів віzку 0.114-04.10.00 : 000 (1) | 109.05.047               | 109.01.022<br>109.01.035<br>109.01.036<br>109.01.003 |
| 109.01.003 | Підготовка шпіntonov для ремонту                                   | 109.05.049<br>109.05.050 | 109.05.051   |

| Код роботи | Назва роботи   | Попередні роботи                       | Подальші роботи                                      |
|------------|--|--|--|
| 109.05.051 | Ремонт шпіntonов   | 109.01.003                             | 109.01.031<br>109.01.032                             |
| 109.01.004 | Виготовлення комплекту деталей для ремонту візків                    | 109.05.046<br>109.05.047<br>109.05.048 | 109.01.005<br>109.01.006                             |
| 109.01.005 | Фарбування виготовлених деталей візку 0.162-04.20.00 : 000 (1)       | 109.01.004                             | 109.05.052   |
| 109.05.052 | Сушка атмосферна виготовлених деталей візку 0.162-04.20.00 : 000 (1) | 109.01.005                             | 109.01.044<br>109.01.007<br>109.01.009               |
| 109.01.006 | Фарбування виготовлених деталей візку 0.114-04.10.00 : 000 (1)       | 109.01.004                             | 109.05.053   |
| 109.05.053 | Сушка атмосферна виготовлених деталей візку 0.114-04.10.00 : 000 (1) | 109.01.006                             | 109.01.044<br>109.01.007<br>109.01.009<br>109.01.047 |
| 109.01.007 | Ремонт рами візку 0.162-04.20.00 : 000 (1)                           | 109.05.052<br>109.05.053<br>109.05.048 | 109.01.008   |
| 109.01.008 | Грунтuvання рами візку 0.162-04.20.00 : 000 (1)                      | 109.01.007                             | 109.05.054   |
| 109.05.054 | Сушка атмосферна рами візку 0.162-04.20.00 : 000 (1)                 | 109.01.008                             | 109.01.029<br>109.01.010                             |
| 109.01.009 | Ремонт рами візку 0.114-04.10.00 : 000 (1)                           | 109.05.052<br>109.05.053<br>109.05.048 | 109.01.010   |

| Код роботи | Назва роботи  | Попередні роботи                       | Подальші роботи          |
|------------|---|--|--------------------------|
| 109.01.010 | Грунтування рами візку 0.114-04.10.00 : 000 (1)                     | 109.01.009                             | 109.05.055               |
| 109.05.055 | Сушка атмосферна рами візку 0.114-10.00 : 000 (1) після грунтування | 109.01.010                             | 109.01.030<br>109.01.032 |
| 109.01.044 | Ремонт надресорного брусу 3751-Н (1)                                | 109.05.052<br>109.05.053<br>109.05.048 | 109.01.045               |
| 109.01.045 | Грунтування надресорного брусу 3751-Н (1)                           | 109.01.044                             | 109.05.056               |
| 109.05.056 | Сушка атмосферна надресорного брусу 3751-Н (1) після грунтування    | 109.01.045                             | 109.01.046               |
| 109.01.046 | Фарбування надресорного брусу 3751-Н (1)                            | 109.05.056                             | 109.05.057               |
| 109.05.057 | Фарбування надресорного брусу 3751-Н (1)                            | 109.01.046                             | 109.01.040<br>109.01.039 |
| 109.01.047 | Ремонт надресорного брусу 3751-Н (2)                                | 109.05.052<br>109.05.053<br>109.05.048 | 109.01.048               |
| 109.01.048 | Грунтування надресорного брусу 3751-Н (2)                           | 109.01.047                             | 109.05.058               |
| 109.05.058 | Сушка атмосферна надресорного брусу 3751-Н (2) після грунтування    | 109.01.048                             | 109.01.049               |
| 109.01.049 | Фарбування надресорного брусу 3751-Н (2)                            | 109.05.058                             | 109.05.059               |

| Код роботи | Назва роботи  | Попередні роботи         | Подальші роботи                                      |
|------------|---|--------------------------|--|
| 109.05.059 | Сушка атмосферна надресорного брусу 3751-Н (2) після фарбування | 109.01.049               | 109.01.041<br>109.01.039                             |
| 109.01.017 | Ремонт деталей підвісок<br>30.30.025                            | 109.05.052<br>109.05.053 | 109.01.041<br>109.01.023<br>109.01.024<br>109.01.040 |
| 109.01.018 | Ремонт тяги повідця   | 109.05.052<br>109.05.053 | 109.01.041<br>109.01.039<br>109.01.025               |
| 109.01.019 | Ремонту запобіжного стрижня центрального підвішування           | 109.05.052<br>109.05.053 | 109.01.041<br>109.01.039<br>109.01.025               |
| 109.01.020 | Ремонт траверсів  | 109.05.052<br>109.05.053 | 109.01.037<br>109.01.038<br>109.01.025<br>109.01.027 |
| 109.01.021 | Ремонт тяги 13.41.033   | 109.05.049<br>109.05.050 | 109.01.037<br>109.01.038<br>109.01.025<br>109.01.027 |
| 109.01.022 | Ремонт важелів гальмівної важільної передачі (візків)           | 109.05.049<br>109.05.050 | 109.01.037<br>109.01.038<br>109.01.025<br>109.01.027 |
| 109.01.023 | Випробування деталей на розтяг 0.162-04.20.00 : 000 (1)         | 109.01.017               | 109.01.025<br>109.01.027                             |

| Код роботи | Назва роботи   | Попередні роботи   | Подальші роботи          |
|------------|--|--|--------------------------|
| 109.01.024 | Випробування деталей на розтяг 0.114-04.10.00 : 000 (1)  | 109.01.017   | 109.01.025<br>109.01.027 |
| 109.01.025 | Грунтuvання відремонтованих деталей і складових одиниць візу 0.162-04.20.00 : 000 (1)                        | 109.01.023<br>109.01.017<br>109.01.018<br>109.01.019<br>109.01.020 | 109.05.060               |
| 109.05.060 | Сушка атмосферна відремонтуваних деталей і складових одиниць візу 0.162-04.20.00 : 000 (1) після грунтuvання | 109.01.025   | 109.01.026               |
| 109.01.026 | Фарбування відремонтованих деталей і складових одиниць візу 0.162-04.20.00 : 000 (1)                         | 109.05.060   | 109.05.061               |
| 109.05.061 | Сушка атмосферна відремонтуваних деталей і складових одиниць візу 0.162-04.20.00 : 000 (1) після фарбування  | 109.01.026   | 109.01.037               |
| 109.01.027 | Грунтuvання відремонтованих деталей і складових одиниць візу 0.114-10.00 : 000 (1)                           | 109.01.024<br>109.01.017<br>109.01.018<br>109.01.019<br>109.01.020 | 109.05.062               |
| 109.05.062 | Сушка атмосферна відремонтуваних деталей і складових одиниць візу 0.114-04.10.00 : 000 (1) після грунтuvання | 109.01.027   | 109.05.057               |

| Код роботи | Назва роботи  | Попередні роботи         | Подальші роботи                        |
|------------|---|--------------------------|--|
| 109.01.028 | Фарбування відремонтованих деталей і складових одиниць візу 0.114-04.10.00 : 000 (1)                        | 109.05.062               | 109.05.063                             |
| 109.05.063 | Сушка атмосферна відремонтуваних деталей і складових одиниць візу 0.114-04.10.00 : 000 (1) після фарбування | 109.01.028               | 109.01.038                             |
| 109.01.029 | Установка термодатчиків на візок 0.162-04.20.00 : 000 (1)   | 109.05.054               | 109.01.033<br>109.01.039<br>109.01.040 |
| 109.01.030 | Установка термодатчиків на візок 0.114-04.10.00 : 000 (1)   | 109.05.055               | 109.01.041<br>109.01.034<br>109.01.039 |
| 109.01.031 | Установка шпінтонов на рами візків 0.162-04.20.00 : 000 (1)   | 109.05.051<br>109.05.054 | 109.01.033<br>109.01.037               |
| 109.01.032 | Установка шпінтонов на рами візків 0.114-10.00 : 000 (1)  | 109.05.051<br>109.05.055 | 109.01.034<br>109.01.038               |
| 109.01.033 | Фарбування рами візу 0.162-04.20.00 : 000 (1)   | 109.01.029<br>109.01.031 | 109.05.064                             |
| 109.05.064 | Сушка атмосферна рами візу 0.162-04.20.00 : 000 (1) після фарбування  | 109.01.033               | 109.01.037                             |
| 109.01.034 | Фарбування рами візу 0.114-04.10.00 : 000 (1)   | 109.01.030<br>109.01.032 | 109.05.065                             |
| 109.05.065 | Сушка атмосферна рами візу 0.114-04.10.00 : 000 (1) після фарбування  | 109.01.034               | 109.01.038                             |

| Код роботи | Назва роботи  | Попередні роботи   | Подальші роботи                        |
|------------|---|--|--|
| 109.01.035 | Ремонт підвісок башмаків візків   | 109.05.049<br>109.05.050   | 109.01.037<br>109.01.038               |
| 109.01.036 | Ремонт пружин візків  | 109.05.049<br>109.05.050   | 109.01.040<br>109.01.041<br>109.01.039 |
| 109.01.037 | Збірка і монтаж гальмової важільної передачі візку 0.162-04.20.00 : 000 (1) | 109.01.021<br>109.01.022<br>109.01.020<br>109.01.035<br>109.05.061<br>109.05.064<br>109.01.031 | 109.01.040<br>109.01.039               |
| 109.01.038 | Збірка і монтаж гальмової важільної передачі візку 0.114-04.10.00 : 000 (1) | 109.01.021<br>109.01.022<br>109.01.020<br>109.01.035<br>109.05.063<br>109.05.065<br>109.01.032 | 109.01.041<br>109.01.039               |
| 109.01.039 | Випробування гасителів коливання 45.30.045                                  | 109.01.036<br>109.01.037<br>109.05.057   | 109.01.040<br>109.01.041               |

| Код роботи | Назва роботи  | Попередні роботи   | Подальші роботи  |
|------------|---|--|--|
| 109.01.040 | Збірка візку 0.162-04.20.00 : 000 (1)                   | 109.01.036<br>109.01.037<br>109.01.039<br>109.05.057<br>109.01.017<br>109.01.018 | 109.01.043   |
| 109.01.043 | Фарбування візку 0.162-04.20.00 : 000 (1) в зборі       | 109.01.040   | 109.05.066   |
| 109.05.066 | Сушка атмосферна візок 0.162-04.20.00 : 000 (1) в зборі | 109.01.043   |  |
| 109.01.041 | Збірка візку 0.114-04.10.00 : 000 (1)                   | 109.01.036<br>109.01.038<br>109.01.039<br>109.01.017<br>109.01.018               | 109.01.042   |
| 109.01.042 | Фарбування візку 0.114-10.00 : 000 (1) в зборі          | 109.01.041   | 109.05.067   |
| 109.05.067 | Сушка атмосферна візок 0.114-10.00 : 000 (1) в зборі    | 109.01.042   |  |
| 109.05.048 | Дефектація віzkів                                       | 109.05.049<br>109.05.050   | 109.01.007<br>109.01.009<br>109.01.044<br>109.01.047<br>109.01.004 |
| 103.01.002 | Нове формування пасажирської колісної пари РУ1Ш-950 Б   | 109.01.001<br>109.01.002   | 103.01.003   |

| Код роботи | Назва роботи  | Попередні роботи | Подальші роботи          |
|------------|---|------------------|--------------------------|
| 103.01.003 | Фарбування колісної пари РУ1Ш-950 ТУ 24.05.816-82 (для вагона мод. 47К) | 103.01.002       | 109.01.040<br>109.01.041 |

## Висновки до розділу 5

1. Виконано програмну реалізацію запропонованого у підрозділі 2.4 розширеного нео-фаззі нейрону, який реалізує нечітке висновування за Такагі-Суг'ено довільного порядку. Продемонстровано, що розширені нео-фаззі нейрони мають покращені апроксимуючі властивості порівняно з традиційними нео-фаззі нейронами. Досліджено залежність точності прогнозування розширеними нео-фаззі нейронами як штучно сгенерованих, так і дійсних хаотичних часових рядів від порядку нечіткого висновування та кількості функцій належності.
2. Проведено імітаційне моделювання запропонованої у розділі 2 архітектури (що ґрунтуються на розширеніх нео-фаззі нейронах) та методів навчання гібридної каскадної нейронної мережі, що еволюціонує, з оптимізацією пулу нейронів у кожному каскаді. Показано що запропоновані нейрони-узагальнювачі реалізують оптимальний за точністю прогноз нелінійних стохастичних і хаотичних сигналів у онлайн режимі.
3. Виконано програмну реалізацію запропонованого у підрозділі 3.2.1 багатовимірного нео-фаззі нейрона.
4. Змодельовано запропоновану у підрозділі 3.2 МІМО архітектуру з оптимізацією пулу багатовимірних нейронів у кожному каскаді, що реалізує нелінійне відображення  $R^n \rightarrow R^q$  у режимі послідовного оброблення даних. Показано, що запропоновані багатовимірні узагальнюючі елементи в режимі реального часу реалізують оптимальне об'єднання нейронів.

нання багатовимірних вихідних сигналів нейронів пулу каскадів.

5. Проведено імітаційне моделювання запропонованої у розділі 4 архітектури і методу самонавчання каскадної нейро-фаззі системи, що еволюціонує, для послідовного кластерування потоків даних з автоматичним визначенням оптимальної кількості кластерів.
6. Розв'язано практичну задачу нечіткого кластерування світлин для подальшої їх класифікації за умови невизначеності щодо кількості кластерів та рівня їх розмитості за допомогою самонавчанної гібридної системи, що її було запропоновано у підрозділі 4.3.
7. Розв'язано практичну задачу прогнозування витрат нормогодин для ремонтних робіт віzkів вагонів 61-425, 61-181, 47Д та 47К у ТОВ «Харківський вагонобудівний завод» за допомогою каскадної гібридної нейро-мережі, що еволюціонує, та модифікованого методу її навчання, які було запропоновано у підрозділі 2.1

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Abonyi Janos, Feil Balazs. Cluster Analysis for Data Mining and System Identification. — Birkhäuser Basel, 2007. — ISBN: 3764379871.
- [2] Adaptive Clustering for Multiple Evolving Streams / Bi-Ru Dai, Jen-Wei Huang, Mi-Yen Yeh, Ming-Syan Chen // IEEE Trans. Knowl. Data Eng. — 2006. — Vol. 18, no. 9. — P. 1166–1180. — URL: <http://dx.doi.org/10.1109/TKDE.2006.137>; <http://doi.ieeecomputersociety.org/10.1109/TKDE.2006.137>.
- [3] Angelov Plamen P, Filev Dimitar P. An approach to online identification of Takagi-Sugeno fuzzy models // Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on. — 2004. — Vol. 34, no. 1. — P. 484–498.
- [4] Angelov Plamen P., Lughofer Edwin. Data-driven evolving fuzzy systems using eTS and FLEXFIS: comparative analysis // Int. J. General Systems. — 2008. — Vol. 37, no. 1. — P. 45–67. — URL: <http://dx.doi.org/10.1080/03081070701500059>.
- [5] Arrow Kenneth Joseph, Hurwicz Leonid, Uzawa Hirofumi. Studies in linear and non-linear programming. Stanford mathematical studies in the social sciences. — Stanford,Ca : Stanford university press, 1972. — ISBN: 0-8047-0562-3. — URL: <http://opac.inria.fr/record=b1099343>.
- [6] Beringer Jürgen, Hüllermeier Eyke. Online clustering of parallel data streams // Data Knowl. Eng. — 2006. — Vol. 58, no. 2. — P. 180–204. — URL: <http://dx.doi.org/10.1016/j.datak.2005.05.009>.
- [7] Beringer Jürgen, Hüllermeier Eyke. Online clustering of parallel data streams // Data Knowl. Eng. — 2006. — Vol. 58, no. 2. — P. 180–204. — URL: <http://dx.doi.org/10.1016/j.datak.2005.05.009>.
- [8] Bezdek J.C. Pattern recognition with fuzzy objective function algorithms. — Kluwer Academic Publishers, 1981.

- [9] Bodyanskiy Yevgeniy. Computational Intelligence Techniques for Data Analysis // Leipziger Informatik-Tage / Ed. by Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig. — Vol. 72 of LNI. — GI, 2005. — P. 15–36. — URL: <http://subs.emis.de/LNI/Proceedings/Proceedings72/article3735.html>.
- [10] Bodyanskiy Ye., Gorshkov Ye., Kolodyazhniy V. New recursive learning algorithms for fuzzy Kohonen clustering network // Proc. 17th Int. Workshop on Nonlinear Dynamics of Electronic Systems. — 2009. — P. 58–61.
- [11] Bodyanskiy Yevgeniy, Grimm Paul, Teslenko Nataliya. Evolving cascaded neural network based on multidimensional Epanechnikov's kernels and its learning algorithm // Int. J. Information Technologies and Knowledge. — 2011. — Vol. 5, no. 1. — P. 25–30.
- [12] Bodyanskiy Yevgeniy, Kharchenko Oleksandra, Vynokurova Olena. Hybrid cascade neural network based on wavelet-neuron // Information Theories and Application. — 2011. — Vol. 18, no. 4. — P. 335–343.
- [13] Bodyanskiy Yevgeniy, Kokshenev Illya, Kolodyazhniy Vitaliy. An adaptive learning algorithm for a neo fuzzy neuron. // EUSFLAT Conf. — 2003. — P. 375–379.
- [14] Bodyanskiy Ye., Kolodyazhniy V., Stephan A. Recursive fuzzy clustering algorithms // Proc. 10th East West Fuzzy Colloquium. — 2002. — P. 276–283.
- [15] Bodyanskiy Yevgeniy, Viktorov Yevgen. The cascaded neo-fuzzy architecture using cubic-spline activation functions // Inf Theor Appl. — 2009. — Vol. 16, no. 3. — P. 245–259.
- [16] Bodyanskiy Yevgeniy, Viktorov Yevgen. 110 9 – Intelligent Processing THE CASCADE NEO-FUZZY ARCHITECTURE AND ITS ONLINE LEARNING ALGORITHM. — 2013. — URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.386.7492>.
- [17] Bodyanskiy Yevgeniy, Viktorov Yevgen, Pliss Iryna. International Book Series "Information Science and Computing" 25 THE CASCADE GROWING NEURAL NETWORK USING

QUADRATIC NEURONS AND ITS LEARNING ALGORITHMS FOR ON-LINE INFORMATION PROCESSING.— 2013.— URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.386.9313>.

- [18] Bodyanskiy Ye V, Pliss IP, Solovyova T. Multistep optimal predictors of multidimensional non-stationary stochastic processes // Doklady AN USSR, series A. — 1986. — Vol. 12. — P. 47–49.
- [19] Chen L H, Chang S. An adaptive learning algorithm for principal component analysis. // IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council. — 1995. — Vol. 6, no. 5. — P. 1255–63.
- [20] Chen Y., Tu L. Density-based clustering for real-time stream data // Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. — 2007. — P. 133–142.
- [21] Chung Fu Lai, Lee Tong. Fuzzy Competitive Learning // Neural Netw. — 1994. — Vol. 7, no. 3. — P. 539–551. — URL: [http://dx.doi.org/10.1016/0893-6080\(94\)90111-2](http://dx.doi.org/10.1016/0893-6080(94)90111-2).
- [22] Chung Fu-Lai, Lee Tong. Fuzzy competitive learning // Neural Networks. — 1994. — Vol. 7, no. 3. — P. 539–551. — URL: [http://dx.doi.org/10.1016/0893-6080\(94\)90111-2](http://dx.doi.org/10.1016/0893-6080(94)90111-2).
- [23] Cichocki A., Unbehauen R. Neural networks for optimization and signal processing // Journal of Signal Processing. — 1998. — Vol. 2. — P. 62–63.
- [24] Crespo Fernando, Weber Richard. A methodology for dynamic data mining based on fuzzy clustering // Fuzzy Sets and Systems. — 2005. — Vol. 150, no. 2. — P. 267–284. — URL: <http://dx.doi.org/10.1016/j.fss.2004.03.028>.
- [25] Cypkin Ja.Z., Kaplinskii A. I., Krasnenker A. S. Fundamentals of the theory of learning systems // Nauka. — 1970. — P. 252.
- [26] Du K.L., Swamy M.N.S. Neural Networks and Statistical Learning. SpringerLink : Bücher. — Springer, 2013. — ISBN: 9781447155713. — URL: <https://books.google.com.ua/books?id=wzK8BAAAQBAJ>.
- [27] Fahlman Scott E., Lebiere Christian. The cascade-correlation learning architecture // Advances in Neural Information Processing Systems 2. — Mor-

- gan Kaufmann, 1990. — P. 524–532.
- [28] Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition / Frank Höppner, Frank Klawonn, Rudolf Kruse, Thomas Runkler. — 1. Auflage edition. — John Wiley & Sons, 1999. — ISBN: 0471988642.
- [29] Fuzzy models and algorithms for pattern recognition and image processing / James C. Bezdek, James Keller, Raghu Krishnapuram, Nikhil R. Pal. The Handbooks of fuzzy sets series. — New York : Springer, 2005. — ISBN: 0-387-24515-4. — URL: <http://opac.inria.fr/record=b1102792>.
- [30] Gan Guojun, Ma Chaoqun, Wu Jianhong. Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability). — SIAM, 2007. — ISBN: 0898716233.
- [31] Han H., Qiao J. A self-organizing fuzzy neural network based on a growing-and-pruning algorithm // Fuzzy Systems, IEEE Transactions on. — 2010. — Vol. 18, no. 6. — P. 1129–1143.
- [32] Han J, Kamber M. Mining Stream, Time-Series and Sequence Data // Data Mining: Concepts and Techniques. — 2006. — P. 467–489.
- [33] Haykin S. Neural Networks: a Comprehensive Foundation. — New York, NY : Macmillan, 1994.
- [34] Bodyanskiy Yevgeniy, Dolotov Artem, Pliss Iryna, Viktorov Yevgen. International Book Series "Information Science and Computing" 13 THE CASCADE ORTHOGONAL NEURAL NETWORK. — 2008.
- [35] Jang Jyh-Shing Roger, Sun Chuen-Tsai, Mizutani Eiji. Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. MATLAB curriculum series. — pub-PH:adr : Prentice-Hall, 1997. — P. xxvi + 614. — ISBN: 0-13-261066-3.
- [36] Kaczmarz S. Approximate solution of systems of linear equations // Int. J. Control,. — 1993. — Vol. vol. 53. — P. 1269–1271.
- [37] Kasabov Nikola. Evolving connectionist systems - the knowledge engineering approach (2. ed.). — Springer, 2007. — P. I–XXI, 1–457. — URL:

[http://dx.doi.org/10.1007/978-1-84628-347-5.](http://dx.doi.org/10.1007/978-1-84628-347-5)

- [38] Kasabov Nikola K, Song Qun. DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction // Fuzzy Systems, IEEE Transactions on. — 2002. — Vol. 10, no. 2. — P. 144–154.
- [39] Kashyap R. L., Blaydon Colin C. Estimation of probability density and distribution functions // IEEE Transactions on Information Theory. — 1968. — Vol. 14, no. 4. — P. 549–556. — URL: <http://dx.doi.org/10.1109/TIT.1968.1054184>.
- [40] Kohonen Teuvo. Self-organizing maps. — Springer Science & Business Media, 2001. — Vol. 30.
- [41] Kolodyazhniy V, Bodyanskiy Ye. Cascaded multiresolution spline-based fuzzy neural network // Proc. Int. Symp. on Evolving Intelligent Systems. — 2010. — P. 26–29.
- [42] Kruschke John K, Movellan Javier R. Benefits of gain: Speeded learning and minimal hidden layers in back-propagation networks // Systems, Man and Cybernetics, IEEE Transactions on. — 1991. — Vol. 21, no. 1. — P. 273–280.
- [43] Leng Gang, Prasad Girijesh, McGinnity T. Martin. An on-line algorithm for creating self-organizing fuzzy neural networks // Neural Networks. — 2004. — Vol. 17, no. 10. — P. 1477–1493. — URL: <http://dx.doi.org/10.1016/j.neunet.2004.07.009>.
- [44] Liu Ying-Ho. Stream mining on univariate uncertain data // Appl. Intell. — 2013. — Vol. 39, no. 2. — P. 315–344. — URL: <http://dx.doi.org/10.1007/s10489-012-0415-3>.
- [45] Ljung Lennart. System Identification: Theory for the User. — Prentice-Hall, Inc., 1999.
- [46] Lughofer Edwin. FLEXFIS: A Robust Incremental Learning Approach for Evolving Takagi-Sugeno Fuzzy Models // IEEE T. Fuzzy Systems. — 2008. — Vol. 16, no. 6. — P. 1393–1410. — URL: <http://dx.doi.org/10.1109/TFUZZ.2008.925908>.
- [47] Lughofer Edwin. Evolving Fuzzy Systems - Methodologies, Advanced

Concepts and Applications. — Springer, 2011. — Vol. 266 of Studies in Fuzziness and Soft Computing. — P. 1–410. — ISBN: 978-3-642-18086-6. — URL: <http://dx.doi.org/10.1007/978-3-642-18087-3>; <http://dx.doi.org/10.1007/978-3-642-18087-3>.

- [48] Miki T., (Japan) T. Yamakawa. Analog Implementation of Neo-Fuzzy Neuron and Its On-board Learning. — WSEAS, 1999. — URL: <http://www.worldses.org/online/>.
- [49] Nechyba Michael C., Xu Yangsheng. Cascade Neural Networks with Node-Decoupled Extended Kalman Filtering. — 1997.
- [50] Oliveira Jose Valente de, Pedrycz Witold. Advances in Fuzzy Clustering and Its Applications. — New York, NY, USA : John Wiley & Sons, Inc., 2007. — ISBN: 0470027606.
- [51] Otto Peter, Bodyanskiy Yevgeniy, Kolodyazhniy Vitaliy. A new learning algorithm for a forecasting neuro-fuzzy network // Integrated Computer-Aided Engineering. — 2003. — Vol. 10, no. 4. — P. 399–409. — URL: <http://content.iospress.com/articles/integrated-computer-aided-engineering/ica00163>.
- [52] Pakhira Malay K, Bandyopadhyay Sanghamitra, Maulik Ujjwal. Validity index for crisp and fuzzy clusters // Pattern recognition. — 2004. — Vol. 37, no. 3. — P. 487–501.
- [53] Park Dong C., Dagher Issam. Gradient Based Fuzzy c-means (GBFCM) Algorithm // IEEE International Conference on Neural Networks (ICNN'94). — Vol. III. — Orlando, FL : IEEE, 1994. — jun. — P. 1626–1631. — Intelligent Computing Research Lab, Department of ECE, FL.
- [54] Pedrycz Witold, Rai Partab. Collaborative clustering with the use of Fuzzy C-Means and its quantification // Fuzzy Sets and Systems. — 2008. — Vol. 159, no. 18. — P. 2399–2427. — URL: <http://dx.doi.org/10.1016/j.fss.2007.12.030>.
- [55] Prechelt L. Investigation of the CasCor Family of Learning Algorithms // Neural Networks. — 1997. — Vol. 10, no. 5. — P. 885–896.
- [56] Roubens Marc. Fuzzy clustering algorithms and their cluster validity // European Journal of Operational Research. — 1982. — Vol. 10. — P. 294–301.

- [57] Silva Bruno, Marques Nuno. Neural Network-based Framework for Data Stream Mining. // STAIRS. — 2012. — P. 294–305.
- [58] Silva F. M., Almeida L. B. Speeding up back-propagation // Advanced Neural Computers / Ed. by R. Eckmiller. — Amsterdam : Elsevier, 1990. — P. 151–158.
- [59] Smoothly distributed fuzzy C-means: a new self-organizing map / R. D. Pascual Marqui, A. D. Pascual Montano, K. Kochi, J. M. Carazo // Pattern Recognition. — 2001. — Vol. 34, no. 12. — P. 2395–2402. — URL: <http://www.sciencedirect.com/science/article/B6V14-43W07HY-B/2/8a783959537578469a5357887f06327b>.
- [60] Tsao E. C. K., Bezdek J. C., Pal N. R. Fuzzy Kohonen clustering networks // Pattern Recognition. — 1994. — Vol. 27, no. 5. — P. 757–764. — URL: <http://www.sciencedirect.com/science/article/B6V14-48MPPXD-1YX/2/cf0e26b7498b3a4f4b7698975495f7f9>.
- [61] Veitch A. C., Holmes G. A Modified Quickprop Algorithm // Neural Computation. — 1991. — Vol. 3, no. 3. — P. 310–311.
- [62] Vorobyov Sergiy A., Cichocki Andrzej, Bodyanskiy Yevgeniy V. Adaptive Noise Cancellation For Multi-Sensory Signals. — 2001. — URL: <http://citeseer.ist.psu.edu/404753.html>.
- [63] Vuorimaa Petri. Fuzzy Self-Organizing Map // Fuzzy Sets and Systems. — 1994. — Vol. 66. — P. 223–231.
- [64] Vuorimaa P. Use of the Fuzzy Self-Organizing Map in pattern recognition // Proceedings of the Third IEEE Conference on Fuzzy Systems. IEEE World Congress on Computational Intelligence / Signal Process. Lab. , Tampere Univ. of Technol. , Finland. — Vol. 2. — New York, NY, USA : IEEE, 1994. — P. 798–801.
- [65] Wang Li-Xin. Adaptive fuzzy systems and control - design and stability analysis. — Prentice Hall, 1994. — P. I–XVII, 1–232. — ISBN: 978-0-13-099631-2.
- [66] Wang Li-Xin, Mendel Jerry M. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning // IEEE Transactions

- on Neural Networks. — 1992. — Vol. 3, no. 5. — P. 807–814. — URL: <http://dx.doi.org/10.1109/72.159070>.
- [67] Wang W, Zhang Y. On fuzzy cluster validity indices // Fuzzy Sets and Systems. — 2007. — oct. — Vol. 158, no. 19. — P. 2095–2117.
- [68] Widrow B., Hoff M. E. Adaptive Switching Circuits // Proceedings WESCON. — 1960. — P. 96–104.
- [69] Wolf M. ANFIS: Adaptive-Network-Based Fuzzy Inference System. — 1998.
- [70] Wu Yi. Network Big Data: A Literature Survey on Stream Data Mining. — 2014. — Vol. 9. — P. 2427–2434. — URL: <http://ojs.academypublisher.com/index.php/jsw/article/view/12381>.
- [71] Xie Xuanli Lisa, Beni Gerardo. A Validity Measure for Fuzzy Clustering // IEEE Trans. Pattern Anal. Mach. Intell. — 1991. — Vol. 13, no. 8. — P. 841–847. — URL: <http://dx.doi.org/10.1109/34.85677>.
- [72] Xu Rui, Wunsch Donald C. Clustering algorithms in biomedical research: a review // Biomedical Engineering, IEEE Reviews in. — 2010. — Vol. 3. — P. 120–154.
- [73] Yamakawa T., Tomoda S. A Neo Fuzzy Neuron and its Applications to System Identification and Prediction of the System Behaviour // Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks (IIZUKA'92). — Iizuka, 1992. — P. 477–483.
- [74] Ye Bodyanskiy, Vynokurova E, Teslenko N. Cascade GMDH-wavelet-neuro-fuzzy network // Proc. of the 4th Intern. Workshop on Inductive Modelling IWIM 2011. — 2011. — P. 22–30.