

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

## РОЗДІЛ 1

### ГІБРИДНА КАСКАДНА НЕЙРО-ФАЗЗИ МЕРЕЖА З ОПТИМІЗАЦІЄЮ ПУЛУ НЕЙРОНІВ

Зазвичай під «навчанням» розуміють процес коригування синаптичних вагових коефіцієнтів за допомогою певної процедури оптимізації, що ґрунтується на пошуку екстремуму заданого критерію навчання. Якість процесу навчання може бути поліпшена шляхом коригування топології мережі поспіль з синаптичними вагами [11, 13]. Ця ідея лежить в основі систем обчислювального інтелекту, що еволюціонують [17, 23].

Мабуть, найбільш відомою реалізацією цього підходу є каскадно-кореляційні нейронні мережі [12, 25, 27], 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 незалежно 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 (і, 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.

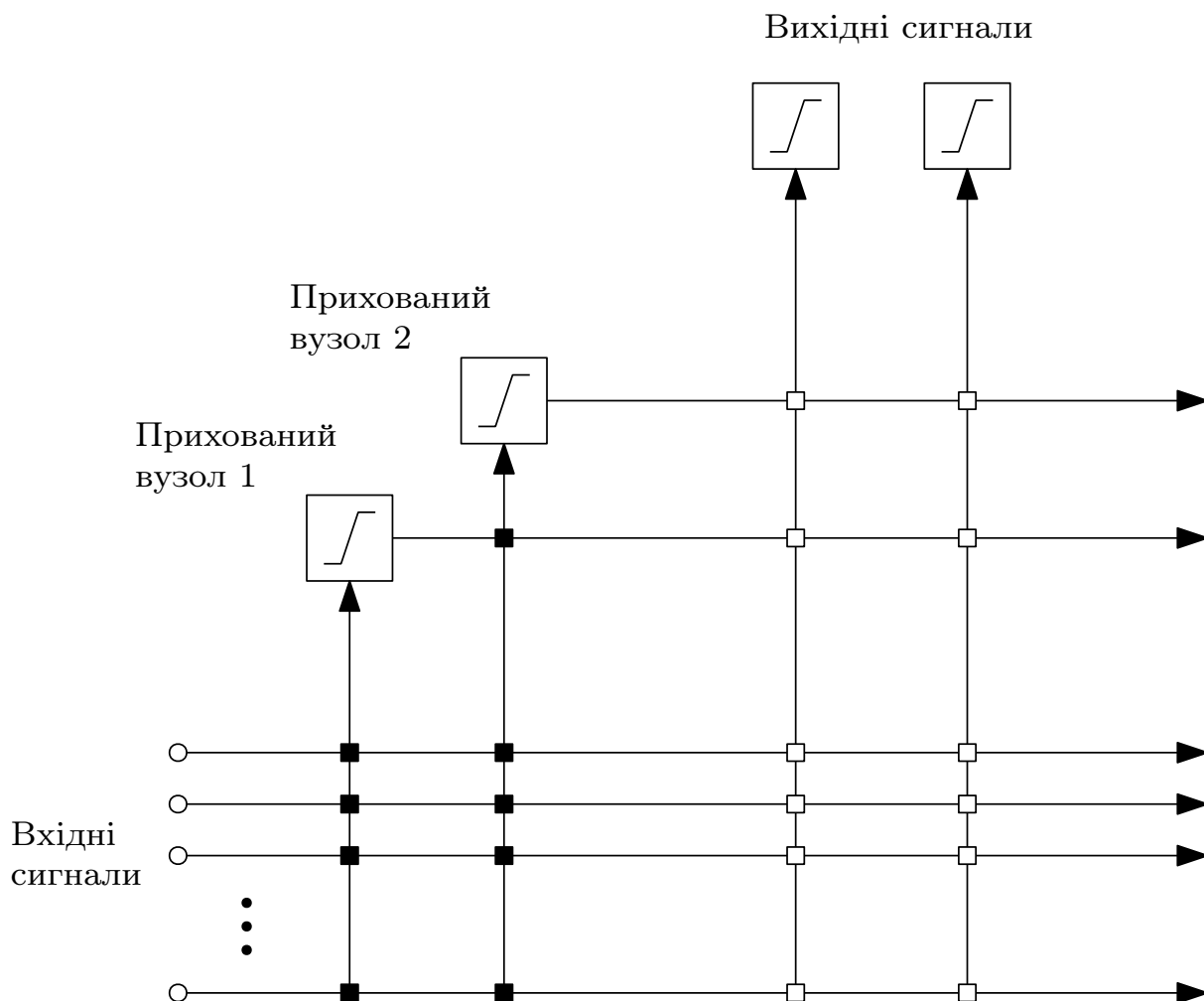


Рис. 1.1 – Архітектура каскадної системи (за Фальманом та Леб'єром) після додавання двох прихованих вузлів. Вхідні сигнали, що надходять до вертикальних ліній, сумуються; вагові коефіцієнти, позначені □, – зафіксовані, позначені ■, – налаштовуються

Автори найбільш поширеної каскадної нейронної мережі, що еволюціонує, CasCorLA (схему наведено на рис. 1.1), Фальман та Леб'єр, використовували елементарні персептрони Розенблатта з традиційними сигмоїдальними функціями активації і коригували синаптичні вагові коефіцієнти за допомогою QuickProp-алгоритму [12], 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 [3, 4, 6–8, 14, 19, 35] доцільно в якості вузлів системи використовувати такі типи нейронів, що їх виходи лінійно залежать від синаптичних ваг, що дозволить використовувати оптимальні за швидкістю 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 оброблення даних.

### 1.1. Архітектура оптимізованої каскадної нейронної мережі

Архітектура пропонованої гібридної системи з оптимізованим пулом нейронів у кожному каскаді наведена на рис. 1.2.

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

$$x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T, \quad (1.1)$$

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

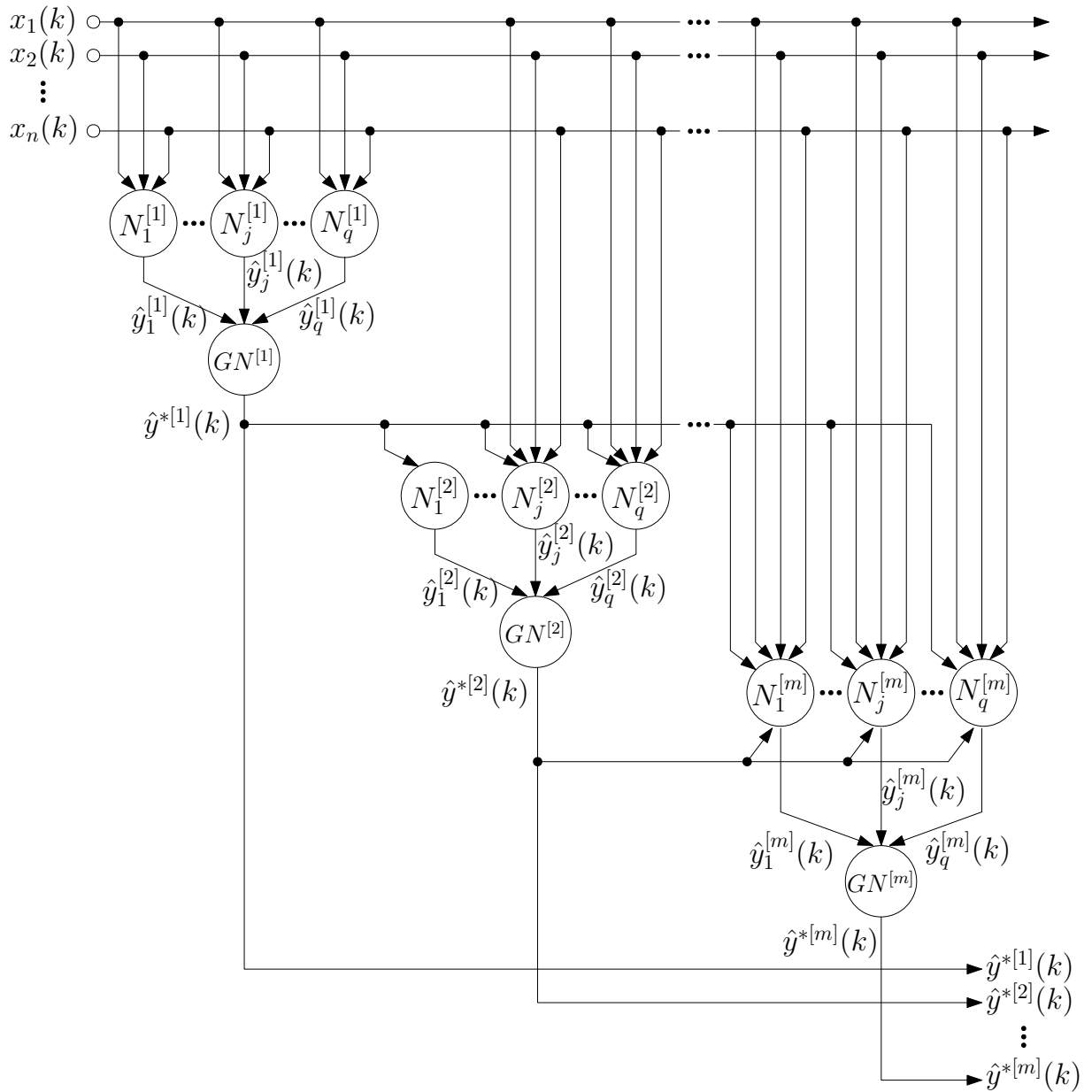


Рис. 1.2 – Архітектура гібридної системи з оптимізованим пулом нейронів

сгенерованого першим каскадом вихідного сигналу  $\hat{y}^{*[1]}(k)$ , нейрони третього каскаду оброблятимуть два додаткових сигнали  $\hat{y}^{*[1]}(k)$ ,  $\hat{y}^{*[2]}(k)$ , нейрони  $m$ -ого 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.

## 1.2. Навчання елементарних персептронів Розенблатта у каскадній оптимізованій системі

Наразі вважатимемо  $j$ -й вузол  $m$ -ого каскаду елементарним персептроном Розенблатта з активаційною функцією

$$0 < \sigma_j^m \left( \gamma_j^{[m]} u_j^{[m]} \right) = \frac{1}{1 + e^{-\gamma_j^{[m]} u_j^{[m]}}} < 1, \quad (1.2)$$

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 першого каскаду матимуть вигляд

$$\hat{y}_j^{[1]} = \sigma_J^{[1]} \left( \gamma_j^{[1]} \sum_{i=0}^n w_{ji}^{[1]} x_i \right) = \sigma_j^{[1]} \left( \gamma_j^{[1]} w_j^{[1]T} x \right), \quad (1.3)$$

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

Вихідні сигнали другого каскаду дорівнюватимуть

$$\hat{y}_j^{[2]} = \sigma_J^{[2]} \left( \gamma_j^{[2]} \left( \sum_{i=0}^n w_{ji}^{[2]} x_i + w_{j,n+1}^{[2]} \hat{y}^{*[1]} \right) \right), \quad (1.4)$$

вихідні сигнали  $m$ -ого каскаду матимуть вигляд

$$\begin{aligned} \hat{y}_j^{[m]} &= \sigma_j^{[m]} \left( \gamma_j^{[m]} \left( \sum_{i=0}^n w_{ji}^{[m]} x_i + w_{j,n+1}^{[m]} \hat{y}^{*[1]} \right. \right. \\ &\quad \left. \left. + w_{j,n+2}^{[m]} \hat{y}^{*[2]} + \dots + w_{j,n+m-1}^{[m]} \hat{y}^{*[m-1]} \right) \right) \\ &= \sigma_j^{[m]} \left( \gamma_j^{[m]} \sum_{i=0}^{n+m-1} w_{ji}^{[m]} x_j^{[m]} \right) = \sigma_j^{[m]} \left( w_j^{[m]T} x^{[m]} \right), \end{aligned} \quad (1.5)$$

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

Таким чином, нейронна мережа з персептронами 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

$$\begin{aligned} E_j^{[m]} &= \frac{1}{2} \left( e_j^{[m]}(k) \right)^2 = \\ &= \frac{1}{2} \left( y(k) - \hat{y}_j^{[m]}(k) \right)^2 = \\ &= \frac{1}{2} \left( y(k) - \sigma_j^{[m]} \left( \gamma_j^{[m]} w_j^{[m]T} x^{[m]}(k) \right) \right)^2, \end{aligned} \quad (1.6)$$

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

Градiєнтну оптимізацію критерію (1.6) відносно  $w_j^{[m]}$  можна записати у вигляді

$$\begin{aligned} w_j^{[m]}(k+1) &= w_j^{[m]} + \eta_j^{[m]}(k+1) e_j^{[m]}(k+1) \gamma_j^{[m]} \hat{y}_j^{[m]}(k+1) \\ &\times \left( 1 - \hat{y}_j^{[m]}(k+1) \right) x^{[m]}(k+1) = \\ &= w_j^{[m]}(k) + \eta_j^{[m]}(k+1) e_j^{[m]}(k+1) \gamma_j^{[m]} J_j^{[m]}(k+1), \end{aligned} \quad (1.7)$$

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

Мінімізувати критерій (1.6) відносно  $\gamma_j^{[m]}$  можна за допомогою алгоритму Крушке-Мовеланна [20]

$$\begin{aligned} \gamma_j^{[m]}(k+1) &= \gamma_j^{[m]}(k) + \eta_j^{[m]}(k+1) e_j^{[m]}(k+1) \hat{y}_j^{[m]}(k+1) \\ &\times \left( 1 - \hat{y}_j^{[m]}(k+1) \right) u_j^{[m]}(k+1). \end{aligned} \quad (1.8)$$

Поєднуючи (1.7) та (1.6), отримаємо алгоритм навчання для  $j$ -ого нейрону  $m$ -ого каскаду

$$\begin{aligned}
\begin{pmatrix} w_j^{[m]}(k+1) \\ \dots \\ \gamma_j^{[m]}(k+1) \end{pmatrix} &= \begin{pmatrix} w_j^{[m]}(k) \\ \dots \\ \gamma_j^{[m]}(k) \end{pmatrix} + \eta_j^{[m]}(k+1) e_j^{[m]}(k+1) \hat{y}_j^{[m]}(k+1) \\
&\times \left(1 - \hat{y}_j^{[m]}(k+1)\right) \begin{pmatrix} \gamma_j^{[m]} x^{[m]}(k+1) \\ \dots \\ u_j^{[m]}(k+1) \end{pmatrix},
\end{aligned} \tag{1.9}$$

або, вводячи нові змінні, у більш компактній формі

$$\begin{aligned}
\tilde{w}_j^{[m]}(k+1) &= \tilde{w}_j^{[m]}(k) + \eta_j^{[m]}(k+1) e_j^{[m]}(k+1) \hat{y}_j^{[m]}(k+1) \tilde{x}^{[m]}(k+1) \\
&= \tilde{w}_j^{[m]}(k) + \eta_j^{[m]}(k+1) e_j^{[m]}(k+1) \tilde{J}_j^{[m]}(k+1).
\end{aligned} \tag{1.10}$$

Використовуючи регуляризуючий параметр (momentum term) [10, 28, 29], можна удосконалити процес корегування синаптичних вагових коефіцієнтів під час навчання. Тоді, замість критерію (1.6) слід використовувати функцію

$$\begin{aligned}
E_j^{[m]}(k) &= \frac{\eta}{2} \left( e_j^{[m]}(k) \right)^2 \\
&+ \frac{1-\eta}{2} \left\| \tilde{w}_j^{[m]}(k) - \tilde{w}_j^{[m]}(k-1) \right\|^2, 0 < \eta \leq 1.
\end{aligned} \tag{1.11}$$

Тоді алгоритм навчання приймає вигляд

$$\begin{aligned}
\tilde{w}_j^{[m]}(k+1) &= \tilde{w}_j^{[m]}(k) \\
&+ \eta_j^{[m]}(k+1) \left( \eta e_j^{[m]}(k+1) \tilde{J}_j^{[m]}(k+1) \right. \\
&\left. + (1-\eta) \left( \tilde{w}_j^{[m]}(k) - \tilde{w}_j^{[m]}(k+1) \right) \right),
\end{aligned} \tag{1.12}$$

що є модифікацією процедури Сільви-Альмейди [28].



Доцільно вдосконалити алгоритм, використовуючи підхід, запропонований у [5], тоді алгоритм (1.12) набуває слідкуючих та фільтруючих властивостей. Таким чином, кінцева модифікація алгоритму набуває вигляду

$$\begin{cases} \tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{\eta e_j^{[m]}(k+1) \tilde{J}_j^{[m]}(k+1)}{r_j^{[m]}(k+1)} \\ \quad + \frac{(1-\eta) \left( \tilde{w}_j^{[m]}(k) - \tilde{w}_j^{[m]}(k-1) \right)}{r_j^{[m]}(k+1)}, \\ r_j^{[m]}(k+1) = r_j^{[m]}(k) + \left\| \tilde{J}_j^{[m]}(k+1) \right\|^2 - \left\| \tilde{J}_j^{[m]}(k-s) \right\|^2, \end{cases} \quad (1.13)$$

де  $s$  – розмір «ковзного» вікна.

Цікаво, що при  $s = 1$  та  $\eta = 1$  отримуємо нелінійну версію загальновідомого алгоритму Качмажа-Уїдроу-Хоффа [16, 33]:

$$\tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{e_j^{[m]}(k+1) \tilde{J}_j^{[m]}(k+1)}{\left\| \tilde{J}_j^{[m]}(k+1) \right\|^2}, \quad (1.14)$$

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

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

Низька швидкість навчання персептронів Розенблатта у поєднанні з труднощами інтерпретації результатів (властиві всім ШНС в цілому) спонукає шукати альтернативні підходи до синтезу еволюційних нейронних мереж. Як зазначається у [15], нейро-фаззі системи відомі високою інтерпретованістю і прозорістю, а також високими апроксимаційними властивостями, та є основою гібридних систем штучного інтелекту. У [6, 19] розглядаються гібридні каскадні системи штучного інтелекту побудовані на нео-фаззі нейронах [24, 34], що дозволяє їм суттєво підвищити швидкість корегування сина-

птичних вагових коефіцієнтів. Нео-фаззі нейрон (NFN), що його архітектуру наведено на рис. 1.3, – це нелінійна система, що реалізує нечітке висновування

$$\hat{y} = \sum_{i=1}^n f_i(x_i), \quad (1.15)$$

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

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

Структурними елементами нео-фаззі нейрона є нелінійні синапси  $NS_i$ , які трансформують вхідні сигнали в наступний спосіб:

$$f_i(x_i) = \sum_{l=1}^h w_{li} \mu_{li}(x_i), \quad (1.16)$$

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

$$\text{ЯКЩО } x_i \text{ У } X_{li} \text{ ТОДІ ВИХІД } w_{li}, \quad (1.17)$$

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 [15].

Запишемо вихідні сигнали для нейронів першого каскаду у наступному вигляді:

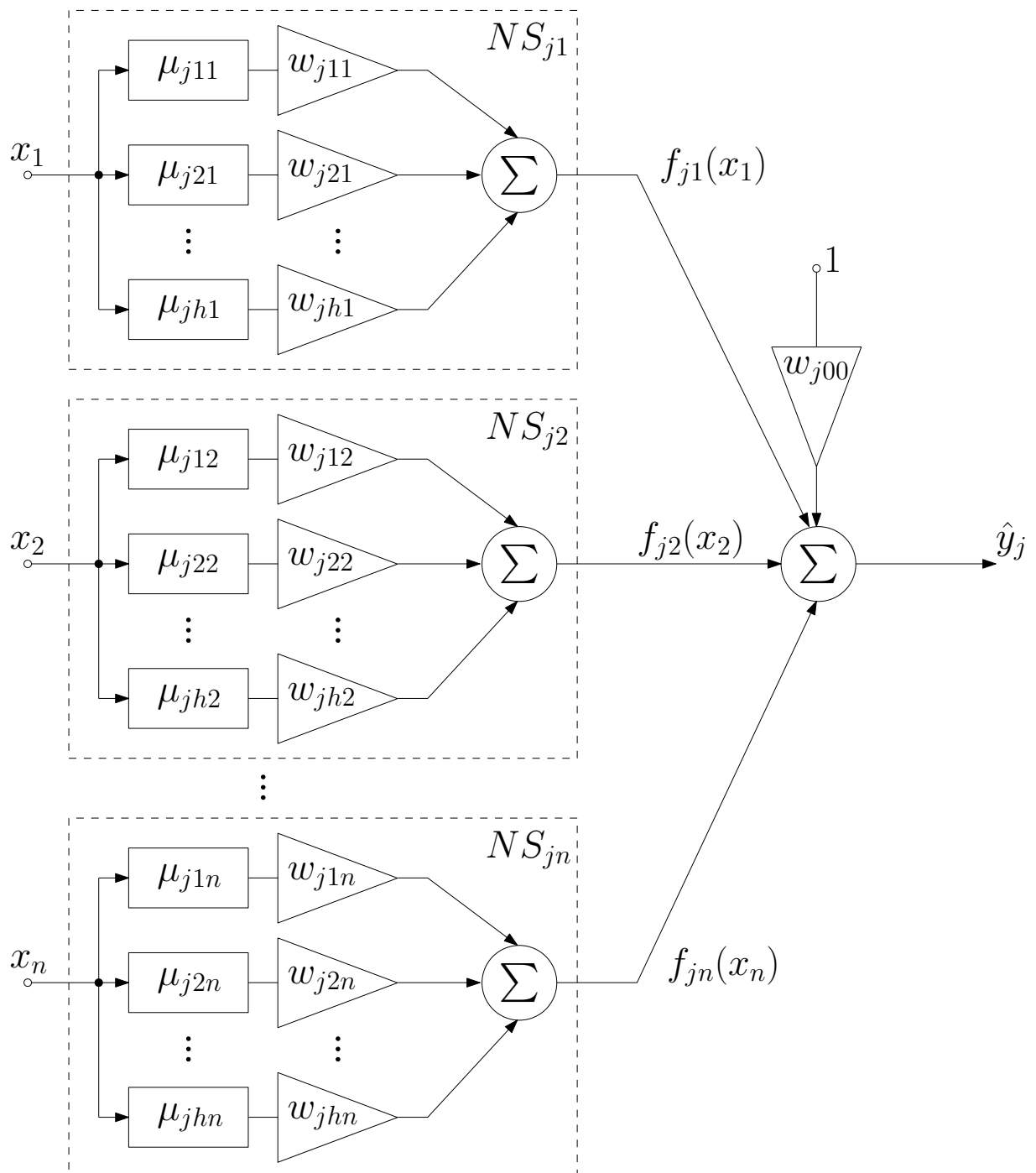


Рис. 1.3 – Архітектура нео-фаззі нейрону

$$\begin{cases} \hat{y}_j^{[1]}(k) = \sum_{i=1}^n f_{ji}^{[1]}(x(k)) = \sum_{i=1}^n \sum_{l=1}^h w_{jli}^{[1]} \mu_{jli}^{[1]}(x_i(k)), \\ \text{ЯКЩО } x_i \in X_{li} \text{ ТОДІ ВИХІД } w_{li} \end{cases} \quad (1.18)$$

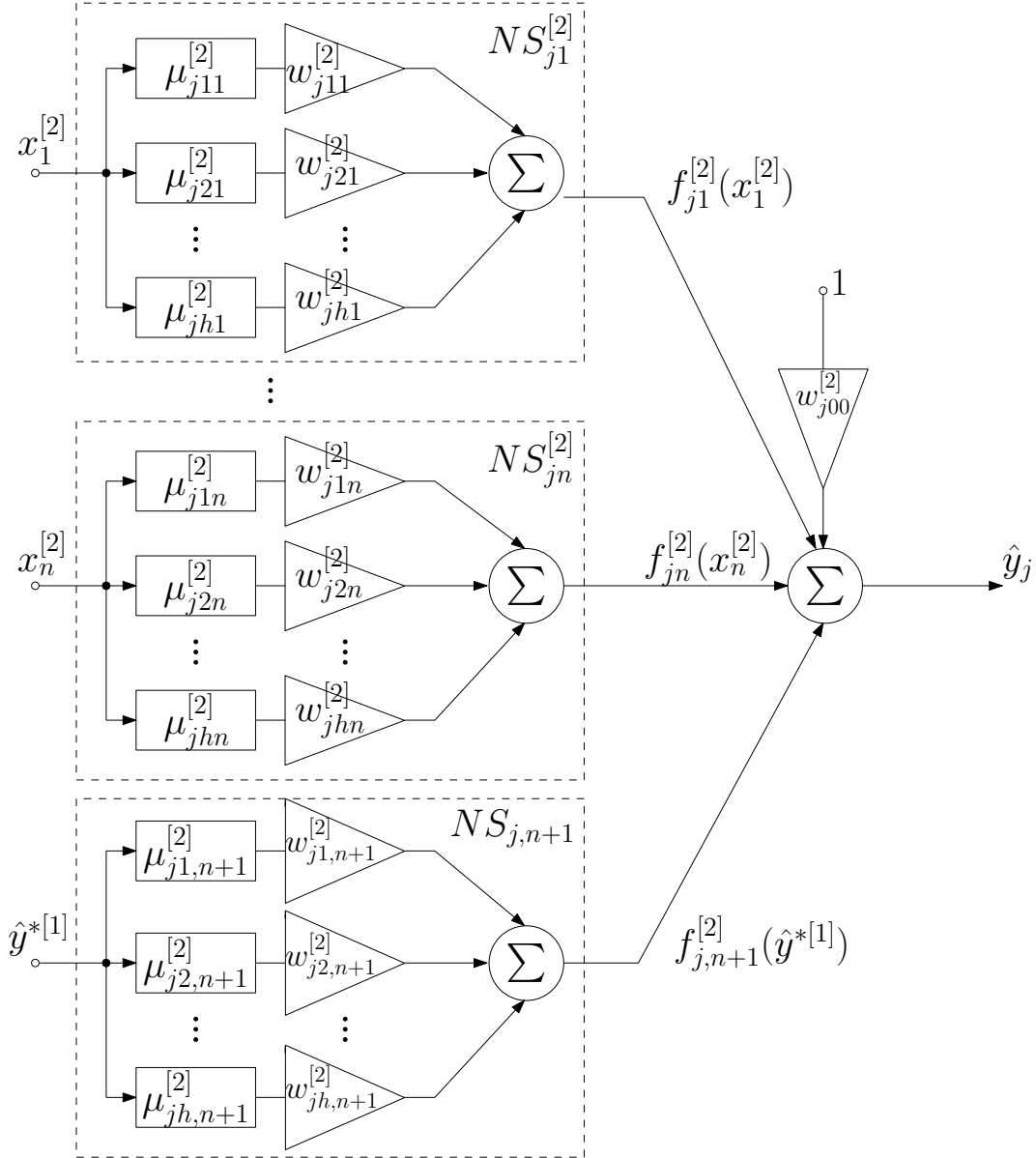


Рис. 1.4 – Нео-фаззі нейрон другого каскаду запропонованої каскадної системи

$j$ -й нео-фаззі нейрон другого каскаду зображено на рис. 1.4 згідно топології нейронної мережі, зображеної на рис. 1.1).

Автори нео-фаззі нейрона [24, 34] в якості функцій належності використовували традиційні трикутні структури, які задовільняють умові розбиття Руспіні:

$$\mu_{jli}^{[1]}(x_i) = \begin{cases} \frac{x_i - c_{j,l-1,i}^{[1]}}{c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}} & \text{if } x_i \in [c_{j,l-1,i}^{[1]}, c_{jli}^{[1]}], \\ \frac{c_{j,l+1,i}^{[1]} - x_i}{c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}} & \text{if } x_i \in [c_{jli}^{[1]}, c_{j,l+1,i}^{[1]}], \\ 0 & \text{інакше,} \end{cases} \quad (1.19)$$

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:

$$\mu_{jli}^{[1]}(x_i) + \mu_{j,l+1,i}^{[1]}(x_i) = 1, \quad (1.20)$$

$$f_{jl}^{[1]}(x_i) = w_{jli}^{[1]} \mu_{jli}^{[1]}(x_i) + w_{j,l+1,i}^{[1]} \mu_{j,l+1,i}^{[1]}(x_i). \quad (1.21)$$

Аппроксимуючі властивості системи можна поліпшити використовуючи кубічні сплайни [6] замість трикутних функцій належності:

$$\mu_{jli}^{[1]}(x_i) = \begin{cases} \frac{1}{4} \left( 2 + 3 \frac{2x_i - c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}}{c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}} - \left( \frac{2x_i - c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}}{c_{jli}^{[1]} - c_{j,l-1,i}^{[1]}} \right)^3 \right), \\ \text{якщо } x \in [c_{j,l-1,i}^{[1]}, c_{jli}^{[1]}], \\ \frac{1}{4} \left( 2 - 3 \frac{2x_i - c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}}{c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}} + \left( \frac{2x_i - c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}}{c_{j,l+1,i}^{[1]} - c_{jli}^{[1]}} \right)^3 \right), \\ \text{якщо } x \in [c_{jli}^{[1]}, c_{j,l+1,i}^{[1]}], \\ 0, & \text{інакше,} \end{cases} \quad (1.22)$$

або  $B$ -сплайни [19]:

$$\mu_{jli}^{g[1]} = \begin{cases} 1, \text{ якщо } x_i \in [c_{jli}^{[1]}, c_{j,l+1,i}^{[1]}], \\ 0, \text{ інакше} \\ \frac{x_i - c_{jli}^{[1]}}{c_{j,l+g-1,i}^{[1]} - c_{jli}^{[1]}} \mu_{jli}^{g-1,[1]}(x_i) + \frac{c_{j,l+g,i}^{[1]} - x_i}{c_{j,l+g,i}^{[1]} - c_{j,l+g,i}^{[1]}} \mu_{j,l+1,i}^{g-1,[1]}(x_i), \\ \text{для } g > 1, \end{cases} \quad (1.23)$$

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

Нескладно помітити, що при  $g = 2$  отримуємо трикутні функції належності (1.19).

$B$ -сплайни, 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 (1.18) визначаємо вихідні сигнали інших каскадів. Так, для другого каскаду можемо записати вихідні сигнали у формі

$$\hat{y}_j^{[2]} = \sum_{i=1}^n \sum_{l=1}^h w_{jli}^{[2]} \mu_{jli}^{[2]}(x_i) + \sum_{l=1}^h w_{j,l,n+1}^{[2]} \mu_{j,l,n+1}^{[2]}(\hat{y}^{*[1]}), \quad (1.24)$$

вихідні сигнали для нейронів  $m$ -ого каскаду

$$\hat{y}_j^{[m]} = \sum_{i=1}^n \sum_{l=1}^h w_{jli}^{[m]} \mu_{jli}^{[m]}(x_i) + \sum_{p=n+1}^{n+m-1} \sum_{l=1}^h w_{jlp}^{[m]} \mu_{jlp}^{[m]}(\hat{y}^{*[p-n]}). \quad (1.25)$$

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

$$\begin{aligned} \mu_j^{[m]}(k) = & \left( \mu_{j11}^{[m]}(x_1(k)), \dots, \mu_{jh1}^{[m]}(x_1(k)), \mu_{j12}^{[m]}(x_2(k)), \right. \\ & \dots, \mu_{jh2}^{[m]}(x_2(k)), \dots, \mu_{jli}^{[m]}(x_i(k)), \dots, \mu_{jhn}^{[m]}(x_n(k)), \\ & \left. \dots, \mu_{j1,n+1}^{[m]}(\hat{y}^{*[1]}(k)), \dots, \mu_{jh,n+m-1}^{[m]}(\hat{y}^{*[m-1]}(k)) \right)^T \end{aligned} \quad (1.26)$$

та відповідний вектор синаптичних вагових коефіцієнтів

$$\begin{aligned} w_j^{[m]} = & \left( w_{j11}^{[m]}, \dots, w_{jh1}^{[m]}, w_{j12}^{[m]}, \dots, w_{jh2}^{[m]}, \dots, w_{jli}^{[m]}, \right. \\ & \left. \dots, w_{jhn}^{[m]}, w_{j1,n+1}^{[m]}, \dots, w_{jh,n+m-1}^{[m]} \right)^T. \end{aligned} \quad (1.27)$$

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

$$\hat{y}_j^{[m]}(k) = w_j^{[m]T} \mu_j^{[m]}(k). \quad (1.28)$$

У такому разі критерій навчання (1.6) приймає вигляд

$$E_j^{[m]}(k) = \frac{1}{2} \left( e_j^m(k) \right)^2 = \frac{1}{2} \left( y(k) - w_j^{[m]T} \mu_j^{[m]}(k) \right), \quad (1.29)$$

а мінімізувати його можна використавши модифікацію процедури [9] для «плаваючого» вікна

$$\begin{cases} w_j^{[m]}(k+1) = w_j^{[m]}(k) + \frac{e_j^{[m]}(k+1) \mu_j^{[m]}(k+1)}{r_j^{[m]}(k+1)}, \\ r_j^{[m]}(k+1) = r_j^{[m]}(k) + \left\| \mu_j^{[m]}(k+1) \right\|^2 - \left\| \mu_j^{[m]}(k-s) \right\|^2, \end{cases} \quad (1.30)$$

або для випадку, коли  $s = 1$ ,

$$w_j^{[m]}(k+1) = w_j^{[m]}(k) + \frac{e_j^{[m]}(k+1) \mu_j^{[m]}(k+1)}{\left\| \mu_j^{[m]}(k+1) \right\|^2}, \quad (1.31)$$

що збігається з одношаговим оптимальним алгоритмом Качмажа-Уїдроу-Хоффа.

Вочевидь, замість (1.30) можна скористатися іншими алгоритмами, як-от експоненційно зважений рекурентний метод найменших квадратів (EWRLSM), що використовується у DENFIS [18], ETS [1] та FLEXFIS [2, 22]. Та варто зауважити, що EWRLSM може бути нестійким при малому коефіцієнті забування.

При використанні критерія навчання з регуляризуючим параметром (1.6) замість (1.29) отримуємо остаточний метод навчання нео-фаззі нейрона

$$\begin{cases} w_j^{[m]}(k+1) = w_j^{[m]}(k) + \frac{\eta e_j^{[m]}(k+1) \mu_j^{[m]}(k+1)}{r_j^{[m]}(k+1)} \\ \quad + \frac{(1-\eta) \left( w_j^{[m]}(k) - w_j^{[m]}(k-1) \right)}{r_j^{[m]}(k+1)}, \\ r_j^{[m]}(k+1) = r_j^{[m]}(k) + \left\| \mu_j^{[m]}(k+1) \right\|^2 - \left\| \mu_j^{[m]}(k-s) \right\|^2. \end{cases} \quad (1.32)$$

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 [21] (наприклад,



рекурентний метод найменших квадратів, робастні методи, методи, що ігнорують застарілі данні, тощо), що дозволяє обробляти нестационарні сигнали в онлайн режимі.

#### 1.4. Розширені нео-фаззі нейрони в якості елементів гібридної каскадної мережі, що еволюціонує

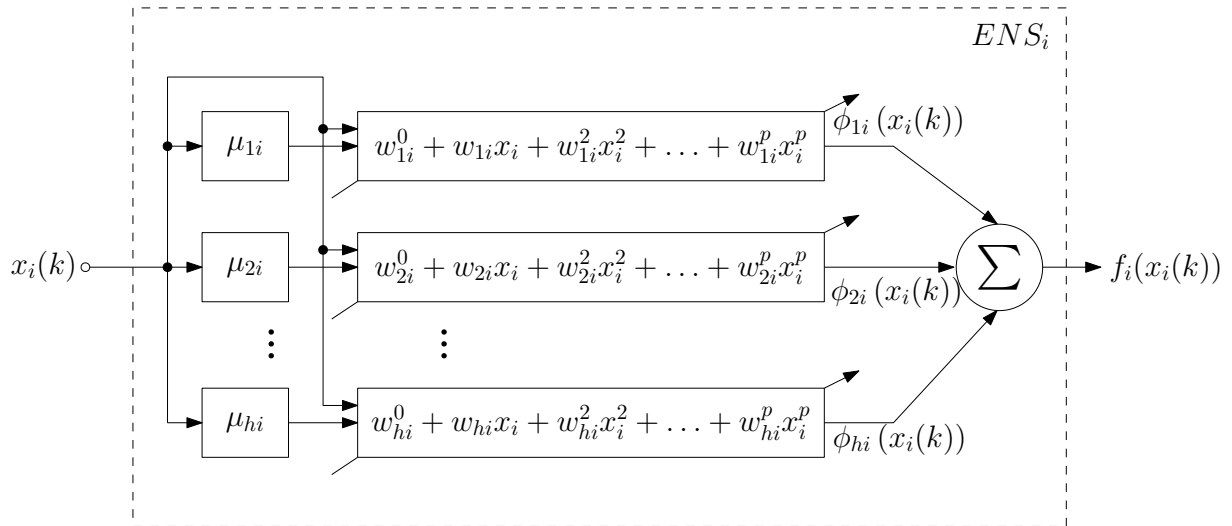


Рис. 1.5 – Синапс розширеного нео-фаззі нейрону

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-Канга, Дженга, Ванга-Менделя, і, фактично реалізує нечітке висновування Такагі-Сугено нульового порядку [31, 32]. 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. 1.5 та рис. 1.6 відповідно.

Вводячі нові змінні

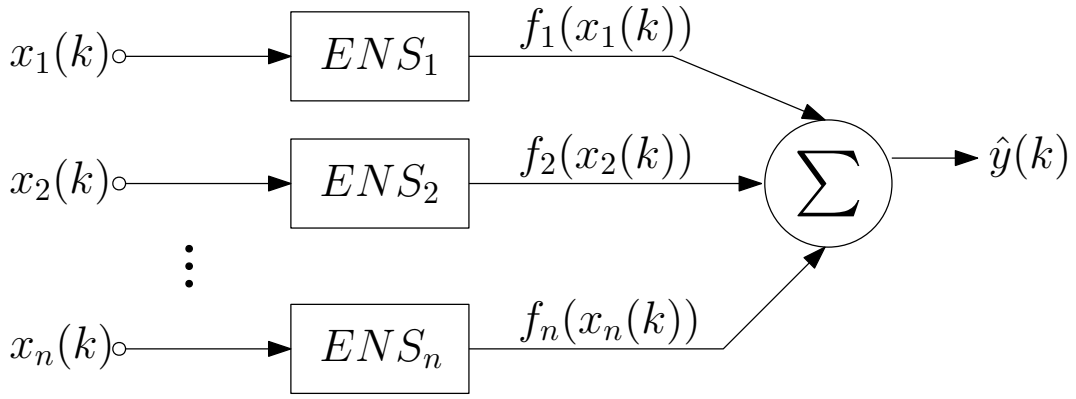


Рис. 1.6 – Розширений нео-фаззі нейрон

$$\phi_{li}(x_i) = \mu_{li}(x_i) (w_{li}^0 + w_{li}^1 x_i + w_{li}^2 x_i^2 + \dots + w_{li}^p x_i^p), \quad (1.33)$$

$$\begin{aligned} f_i(x_i) &= \sum_{l=1}^h \mu_{li}(x_i) (w_{li}^0 + w_{li}^1 x_i + w_{li}^2 x_i^2 + \dots + w_{li}^p x_i^p) \\ &= w_{li}^0 \mu_{li}(x_i) + w_{li}^1 x_i \mu_{li}(x_i) + \dots + w_{li}^p x_i^p \mu_{li}(x_i) \\ &+ w_{2i}^0 \mu_{2i}(x_i) + \dots + w_{2i}^p x_i^p \mu_{2i}(x_i) + \dots + w_{hi}^p x_i^p \mu_{hi}(x_i), \end{aligned} \quad (1.34)$$

$$w_i = (w_{1i}^0, w_{1i}^1, \dots, w_{1i}^p, w_{2i}^0, \dots, w_{2i}^p, \dots, w_{hi}^p)^T, \quad (1.35)$$

$$\begin{aligned} \tilde{\mu}_i(x_i) &= \left( \mu_{1i}(x_i), x_i(\mu_{1i}(x_i)), \dots, x_i^p(\mu_{1i}(x_i)), \right. \\ &\quad \left. \mu_{2i}(x_i), \dots, x_i^p \mu_{2i}(x_i), \dots, x_i^p \mu_{hi}(x_i) \right)^T, \end{aligned} \quad (1.36)$$

можна представити вихідні сигнали розширеного нео-фаззі нейрона у вигляді

$$f_i(x_i) = w_i^T \tilde{\mu}_i(x_i), \quad (1.37)$$

$$\begin{aligned} \hat{y} &= \sum_{i=1}^n f_i(x_i) \\ &= \sum_{i=1}^n w_i^T \tilde{\mu}_i(x_i) \\ &= \tilde{w}^T \tilde{\mu}(x). \end{aligned} \quad (1.38)$$

де

$$\tilde{w}^T = (w_1^T, \dots, w_i^T, \dots, w_n^T)^T, \quad (1.39)$$

$$\tilde{\mu}(x) = (\tilde{\mu}_1^T(x_1), \dots, \tilde{\mu}_i^T(x_i), \dots, \tilde{\mu}_n^T(x_n))^T, \quad (1.40)$$

Таким чином, ENFN містить  $(p+1)hn$  вагових коефіцієнтів та реалізує нечітке висновування Такагі-Сугено  $p$ -ого порядку, а висновування, що його реалізує кожний розширений нелінійний синапс  $\text{ENS}_i$  можна записати у формі

$$\begin{aligned} &\text{ЯКЩО } x_i \text{ У } X_{li} \text{ ТО ВИХІД} \\ &w_{li}^0 + w_{li}^1 x_i + \dots + w_{li}^p x_p, \quad l = 1, 2, \dots, h, \end{aligned} \quad (1.41)$$

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

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

$$\hat{y}^{[1]}(k) = \tilde{w}^{[1]T}(k-1) \tilde{\mu}^{[1]}(x(k)), \quad (1.42)$$

що відрізняється від виразу (1.28) для звичайних NFN тим, що містить у  $p+1$  більше параметрів, що корегуються.

Вочевидь, будь-які методи навчання нео-фаззі нейронів підійдуть і для розширених нео-фаззі нейронів. Так, вирази (1.30) та (1.31) для  $j$ -ого нейрону  $m$ -ого каскаду приймають вигляд

$$\begin{cases} \tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{e_j^{[m]}(k+1) \tilde{\mu}_j^{[m]}(k+1)}{\tilde{r}_j^{[m]}(k+1)}, \\ \tilde{r}_j^{[m]}(k+1) = \tilde{r}_j^{[m]}(k) + \left\| \tilde{\mu}_j^{[m]}(k+1) \right\|^2 - \left\| \tilde{\mu}_j^{[m]}(k-s) \right\|^2 \end{cases} \quad (1.43)$$

та

$$\tilde{w}_j^{[m]}(k+1) = \tilde{w}_j^{[m]}(k) + \frac{e_j^{[m]}(k+1) \tilde{\mu}_j^{[m]}(k+1)}{\left\| \tilde{\mu}_j^{[m]}(k+1) \right\|^2} \quad (1.44)$$

відповідно.

### 1.5. 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 [26, 30].

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

$$\hat{y}^{[m]}(k) = \left( \hat{y}_1^{[m]}(k), \hat{y}_2^{[m]}(k), \dots, \hat{y}_q^{[m]}(k) \right)^T; \quad (1.45)$$

тоді оптимальний вихідний сигнал, що його генерує нейрон  $GN^{[m]}$  (що, власне, є адаптивним лінійним асоціатором [11, 13]), можна записати у формі

$$\hat{y}^{*[m]}(k) = \sum_{j=1}^1 c_j^{[m]} \hat{y}_j^{[m]}(k) = c^{[m]T} \hat{y}^{[m]}(k) \quad (1.46)$$

з обмеженнями на незміщенність

$$\sum_{j=1}^q c_j^{[m]} = E^T c^{[m]} = 1, \quad (1.47)$$

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

$$\begin{aligned} E^{[m]}(k) &= \frac{1}{2} \sum_{\tau=k-s+1}^k \left( y(\tau) - \hat{y}^{*[m]}(\tau) \right)^2 \\ &= \frac{1}{2} \sum_{\tau=k-s+1}^k \left( y(\tau) - c^{[m]T} \hat{y}^{[m]}(\tau) \right)^2, \end{aligned} \quad (1.48)$$

зважаючи на обмеження (1.47), функція Лагранжа матиме вигляд

$$L^{[m]}(k) = E^{[m]}(k) - \lambda \left( 1 - E^T c^{[m]} \right), \quad (1.49)$$

де  $\lambda$  – невизначений Лагранжів множник.

Мінімізуючи (1.49) відносно  $c^{[m]}$ , отримуємо

$$\begin{cases} \hat{y}^{*[m]}(k+1) = \frac{\hat{y}^{[m]T}(k+1) P^{[m]}(k+1) E}{E^T P^{[m]}(k+1) E}, \\ P^{[m]}(k+1) = \left( \sum_{\tau=k-s+2}^{k+1} \hat{y}^{[m]}(\tau) \hat{y}^{[m]T}(\tau) \right)^{-1} \end{cases} \quad (1.50)$$

або у рекурентній формі

$$\begin{cases} \tilde{P}^{[m]}(k+1) = P^{[m]}(k) - \frac{P^{[m]}(k) \hat{y}^{[m]}(k+1) \hat{y}^{[m]T}(k+1) P^{[m]}(k)}{1 + \hat{y}^{[m]T}(k+1) P^{[m]}(k) \hat{y}^{[m]}(k+1)}, \\ P^{[m]}(k+1) = \tilde{P}^{[m]}(k+1) + \\ \quad + \frac{\tilde{P}^{[m]}(k+1) \hat{y}(k-s+1) \hat{y}^{[m]T}(k-s+1) \tilde{P}^{[m]}(k+1)}{1 - \hat{y}^{[m]T}(k-s+1) \tilde{P}^{[m]}(k+1) \hat{y}^{[m]}(k-s+1)}, \\ \hat{y}^{*[m]}(k+1) = \frac{\hat{y}^{[m]T}(k+1) P^{[m]}(k+1) E}{E^T P^{[m]}(k+1) E}. \end{cases} \quad (1.51)$$

У випадку, коли  $s = 1$  (1.50) та (1.51) приймають доволі простий вигляд:

$$\begin{aligned}
\hat{y}^{*[m]}(k+1) &= \frac{\hat{y}^{[m]T}(k+1) \hat{y}^{[m]}(k+1)}{E^T \hat{y}^{[m]}(k+1)} = \\
&= \frac{\|\hat{y}^{[m]}(k+1)\|^2}{E^T \hat{y}^{[m]}(k+1)} = \\
&= \frac{\sum_{j=1}^q (\hat{y}^{[m]}(k+1))^2}{\sum_{j=1}^q \hat{y}^{[m]}(k+1)}.
\end{aligned} \tag{1.52}$$

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 1

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 сформулювати 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 (інтерпритуємість 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 оптимізіція 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 оброблення сигналів нестаціонарних об'єктів.
5. Запропонований розширений нео-фаззі нейрон, який дозволяє реалізовувати нечітке висновуння за Такагі-Сугено довільного порядку, що має покращені апроксимуючі властивості. Зсинтезована архітекутра гібридної системи, що ґрунтується на розширених нео-фаззі нейронах.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] 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.
- [2] 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>.
- [3] 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.
- [4] 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.
- [5] Bodyanskiy Yevgeniy, Kokshenev Illya, Kolodyazhnyi Vitaliy. An adaptive learning algorithm for a neo fuzzy neuron. // EUSFLAT Conf. — 2003. — P. 375–379.
- [6] 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.
- [7] 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>.
- [8] 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>.
- [9] 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.
- [10] 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.
- [11] Cichocki A., Unbehauen R. Neural networks for optimization and signal processing // Journal of Signal Processing. — 1998. — Vol. 2. — P. 62–63.
- [12] Fahlman Scott E., Lebiere Christian. The cascade-correlation learning architecture // Advances in Neural Information Processing Systems 2. — Morgan Kaufmann, 1990. — P. 524–532.
- [13] Haykin S. Neural Networks: a Comprehensive Foundation. — New York, NY : Macmillan, 1994.
- [14] Bodyanskiy Yevgeniy, Dolotov Artem, Pliss Iryna, Viktorov Yevgen. International Book Series "Information Science and Computing" 13 THE CASCADE ORTHOGONAL NEURAL NETWORK. — 2008.
- [15] 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.
- [16] Kaczmarz S. Approximate solution of systems of linear equations // Int. J. Control,. — 1993. — Vol. vol. 53. — P. 1269–1271.
- [17] 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>.
- [18] 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.
- [19] Kolodyazhniy V, Bodyanskiy Ye. Cascaded multiresolution spline-based fuzzy neural network // Proc. Int. Symp. on Evolving Intelligent Systems. — 2010. — P. 26–29.
- [20] 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.
- [21] Ljung Lennart. System Identification: Theory for the User. — Prentice-Hall, Inc., 1999.
- [22] 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>.
- [23] 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>.
- [24] Miki T., (Japan) T. Yamakawa. Analog Implementation of Neo-Fuzzy Neuron and Its On-board Learning. — WSEAS, 1999. — URL: <http://www.worldses.org/online/>.
- [25] Nechyba Michael C., Xu Yangsheng. Cascade Neural Networks with Node-Decoupled Extended Kalman Filtering. — 1997.
- [26] 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>.
- [27] Prechelt L. Investigation of the CasCor Family of Learning Algorithms //

- Neural Networks. — 1997. — Vol. 10, no. 5. — P. 885–896.
- [28] Silva F. M., Almeida L. B. Speeding up back-propagation // Advanced Neural Computers / Ed. by R. Eckmiller. — Amsterdam : Elsevier, 1990. — P. 151–158.
- [29] Veitch A. C., Holmes G. A Modified Quickprop Algorithm // Neural Computation. — 1991. — Vol. 3, no. 3. — P. 310–311.
- [30] Vorobyov Sergiy A., Cichocki Andrzej, Bodyanskiy Yevgeniy V. Adaptive Noise Cancellation For Multi-Sensory Signals. — 2001. — URL: <http://citeseer.ist.psu.edu/404753.html>.
- [31] 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.
- [32] 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>.
- [33] Widrow B., Hoff M. E. Adaptive Switching Circuits // Proceedings WESCON. — 1960. — P. 96–104.
- [34] 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.
- [35] 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.