

**О. Д. Шарапов, А. В. Матвійчук**

ДВНЗ «Київський національний економічний університет імені Вадима Гетьмана»

## **РОЗВИТОК АЛГОРИТМУ ЗВОТНОГО ПОШИРЕННЯ ПОМИЛКИ В ЗАДАЧАХ ОПТИМІЗАЦІЇ ПАРАМЕТРІВ НЕЙРОННИХ МЕРЕЖ**

*АНОТАЦІЯ. У статті наведено результати розрахунків з удосконалення алгоритму зворотного поширення помилки щодо налаштування нейронних мереж прямого розповсюдження шляхом реалізації градієнтного методу оптимізації для всіх параметрів нейронної мережі.*

*ANNOTATION. There are presented in the article the results of researches for enhancing of error back-propagation algorithm for forward propagation neural networks tuning by the way of realization of gradient method optimization for all neural network parameters.*

**КЛЮЧОВІ СЛОВА.** Нейронні мережі, перцептрон, алгоритм зворотного поширення помилки.

Штучні нейронні мережі — потужний математичний інструментарій, що є універсальним відтворювачем складних нелінійних функціональних залежностей. Вони знаходять застосування в таких різноманітних розділах моделювання, як аналіз часових рядів, прогнозування, обробка сигналів, розпізнавання образів та багатьох інших, завдяки такій важливій особливості, як здатність до навчання на підґрунті реальних даних за участі вчителя або без нього. Тобто, нейронні мережі здатні до виявлення закономірностей у наявній інформації навіть без відомостей щодо можливих значень вихідної величини.

Ідея створення штучних нейронних мереж була запропонована психологом та нейроанатомом Мак-Каллоком як результат двадцятирічних досліджень принципів функціонування людського мозку. Для формалізації обробки сигналів у нейронних мережах до Мак-Каллока приєднався математик Пітгс. У результаті їх співпраці з'явилась піонерська робота в області створення штучних нейромережових структур [1]. Ця робота стала проривом в області моделювання складних систем та слугувала поштовхом для зародження такого напрямку у науці, як штучний інтелект.

Найпоширенішою моделлю нейронної мережі можна вважати модель типу багатошаровий перцептрон [2]. Подібні нейронні

мережі ефективні при розв'язанні широкого спектру економічних задач та суттєво перевищують класичні економетричні аналоги, що ґрунтуються на рівняннях регресії. В попередніх авторських роботах цьому інструментарію, а також порівнянню його ефективності із альтернативними моделями, було приділено належну увагу [3—6]. Проведені дослідження показали високу залежність точності відтворення вихідної статистики від ефективності налаштування нейронних мереж. І в разі недостатньо якісної їх оптимізації, нейронні мережі могли виявитись нездатними відтворювати вихідні показники на підставі значень вхідних факторів навіть за умови, коли кількість параметрів моделі була більшою за обсяг навчальної вибірки.

Таким чином, розробка ефективного алгоритму оптимізації параметрів нейронних мереж є важливим та актуальним завданням з огляду на необхідність адекватного моделювання економічних систем, складність яких з часом все більше зростає. Визначення оптимальної конфігурації багатовимірних систем доречно здійснювати із застосуванням градієнтних методів оптимізації. Вперше алгоритм налаштування нейронних мереж, що базується на градієнтному підході, був описаний у дисертаційній роботі Пола Вербоса у 1974 році. Проте, розробниками алгоритму зворотного поширення помилки вважаються Руммельхарт, Хінтон і Вільямс за їх спільну працю [7], в якій цей алгоритм було вперше детально викладено та продемонстровано його застосування при машинній оптимізації нейронної мережі.

З моменту розробки алгоритм зазнав багато удосконалень та видозмін як у наукових працях, так і при конструюванні різноманітних промислових систем. Однак зауважимо, що в усіх удосконалених алгоритмах, проаналізованих нами, здійснювалась оптимізація тільки ваг міжнейронних зв'язків (налаштування інших параметрів нейронної мережі не відбувалось). У результаті ефективність таких нейронних мереж все ж виявляється більшою за альтернативні економетричні моделі з огляду на їх нелінійність (та, відповідно, здатність відтворювати більш складні функціональні залежності). Однак при цьому не використовуються всі переваги нелінійності. Так, з метою підвищення точності моделювання за таких умов доводиться збільшувати кількість нейронів та, відповідно, міжнейронних зв'язків, що може привести до виникнення негативного ефекту перенавчання нейронної мережі [8].

Відповідно, для ефективнішого налаштування нейронних мереж доцільно здійснювати оптимізацію ще й інших параметрів, таких як параметри функцій активації та зміщення суматорів

нейронів. Пояснимо їх сутність на прикладі звичайного нейрону (перцептронного типу) та сигмоїдної функції активації. Так, кожен нейрон у мережі типу перцептрон здійснює перетворення вхідних сигналів у вихідний сигнал згідно функції

$$y = \psi \left( \sum_{i=1}^n w_i \cdot x_i + b \right), \quad (1)$$

де  $y$  — вихідний сигнал нейрону;

$n$  — кількість входів даного нейрону;

$x_i$  —  $i$ -ий вхідний сигнал нейрону;

$w_i$  — вага  $i$ -го входу нейрону;

$b$  — параметр зміщення суматора;

$\psi(\cdot)$  — характеристична функція (функція активації) нейрону.

Функція активації є принципово важливою складовою нейрона. За її допомогою кожен нейрон здатен підсилювати чи послабляти сигнал, отриманий на вході (по аналогії з природними нейронами, що реалізують збуджуючий чи гальмуючий ефект нейроімпульсу). Здатність до якісного налаштування параметрів нейронної мережі можна забезпечити із застосуванням диференційованих функцій активації. Одною з базових функцій, що часто використовується при конструюванні штучних нейронних мереж, є сигмоїдна логістична функція активації, що графічно зображена на рис. 1, а аналітично представлена співвідношенням:

$$\psi(s) = \frac{1}{1 + e^{-ks}}, \quad (2)$$

де  $k$  — коефіцієнт стиснення-розтягування функції вздовж осі абсцис.

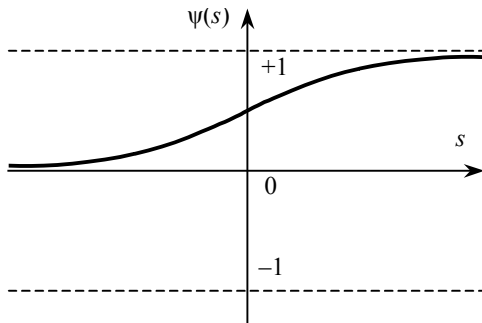


Рис. 1. Сигмоїдна логістична функція активації нейрону

Коефіцієнт стиснення-розтягування може використовуватись у якості параметра підсилення. Перевагою використання сигмоїдної функції в якості нелінійного елементу є те, що вона має обмеження подібно до порогової функції та демонструє поведінку, схожу з природним нейроном. Так, зі зменшенням значення коефіцієнту стиснення-розтягування функція активації стає більш пологою, в граничному випадку (коли  $k = 0$ ) приймаючи вигляд прямої на рівні 0,5. Зі збільшенням коефіцієнту  $k$  сигмоїдна функція все більше стискається, становлячись подібною до функції Хевісайда. При цьому залишається можливість її диференціювання, що дозволяє використовувати градієнтні методи для оптимізації параметрів моделі (зокрема, метод зворотного поширення помилки). Із застосуванням подібної функції активації нейромережа може приймати великі сигнали та залишатись чутливою до слабких змін сигналів. Подібна функція активації може застосовуватись як для нейронів проміжних шарів, так і вихідного шару.

Так, всі нейрони персептрону поєднані в шари. В таких моделях перший шар нейронів слугує для введення вхідних даних, останній — для виведення вихідних сигналів, а приховані (один або кілька) — для обробки вхідних сигналів та збереження інформації про внутрішню структуру об'єкта, що моделюється. Кількість входів та виходів мережі визначається кількістю вхідних та вихідних параметрів досліджуваного об'єкта, явища, процесу тощо. На відміну від зовнішніх шарів, число нейронів прихованих шарів встановлюється емпіричним шляхом. Міжелементні зв'язки в такій мережі утворюються лише між нейронами сусідніх шарів: окремо взятий нейрон може з'єднуватись з одним, кількома або всіма нейронами сусіднього шару (в останньому випадку така нейронна мережа називається повнозв'язною).

При цьому на входи того чи того нейрона надходять сигнали від нейронів попереднього шару (або ззовні), а вихідний сигнал нейрона передається на входи нейронів у наступному шарі (або на вихід мережі, якщо нейрон належить вихідному шару). Зважаючи на те, що зі збільшенням кількості нейронів та міжнейронних зв'язків зазвичай підвищується ефективність моделювання, необхідно відстежувати, щоби нейронна мережа не почала виявляти ефект перенавчання. На рис. 2 представлено модель багатошарового персептрону, що містить три шари.

Персептрон здійснює розрахунок вихідних показників шляхом послідовного перетворення у нейронах з першого по останній шари сигналів, поданих на входи цих нейронів ззовні або з нейронів попереднього шару, що зважені на вагові коефіцієнти між-

нейронних зв'язків. Перетворення сигналів у нейронах перцептронного типу здійснюється згідно описаної вище процедури за функцією (1). Відповідно до зазначеного вище, тришарова нейронна мережа типу перцептрон зі структурою, представленою на рис. 2, здійснюватиме розрахунок вихідних змінних за функціональним співвідношенням:

$$\widehat{y}_j = \psi_j^{(3)} \left( b_j^{(3)} + \sum_{l=1}^p \left[ w_{lj}^{(3)} \cdot \psi_l^{(2)} \left( b_l^{(2)} + \sum_{i=1}^n [w_{il}^{(2)} \cdot \psi_i^{(1)}(x_i)] \right) \right] \right), \quad j = \overline{1, m}, \quad (3)$$

де  $\psi_i^{(1)}(\cdot)$  — функція активації  $i$ -го,  $i = \overline{1, n}$ , нейрона першого (вхідного) шару нейронної мережі;

$\psi_l^{(2)}(\cdot)$ ,  $b_l^{(2)}$  — функція активації та параметр зміщення суматора  $l$ -го,  $l = \overline{1, p}$ , нейрона другого (прихованого) шару мережі;

$\psi_j^{(3)}(\cdot)$ ,  $b_j^{(3)}$  — функція активації та параметр зміщення суматора  $j$ -го,  $j = \overline{1, m}$ , нейрона третього (вихідного) шару перцептронного;

$w_{il}^{(2)}$  — вага міжнейронного зв'язку між  $i$ -им нейроном першого шару та  $l$ -им нейроном другого шару нейронної мережі;

$w_{lj}^{(3)}$  — вага міжнейронного зв'язку між  $l$ -им нейроном другого шару та  $j$ -им нейроном третього шару нейронної мережі.

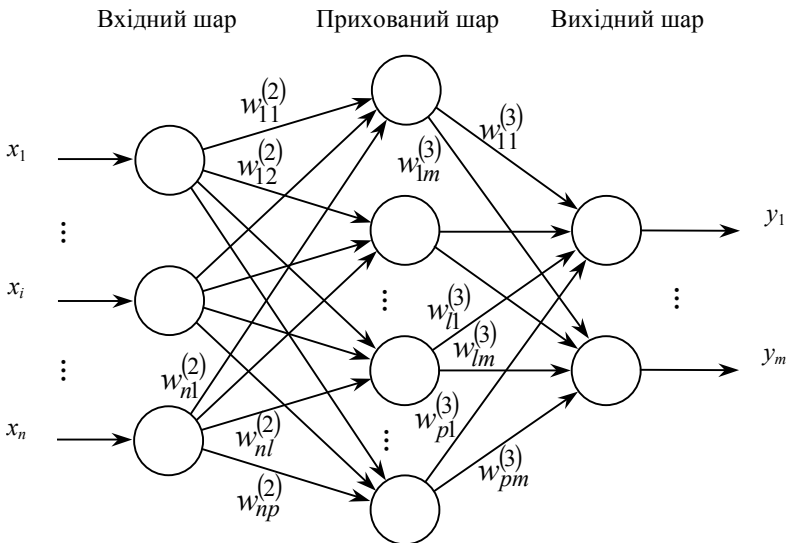


Рис. 2. Структура нейронної мережі типу перцептрон

При функціонуванні нейронної мережі в робочій фазі здійснюються перетворення інформації визначеним чином — згідно функції (3). Поведінка нейронної мережі в робочій фазі детермінована, тобто для кожної комбінації вхідних сигналів на виході завжди при даній настройці буде той самий результат. Упродовж робочої фази нейронна мережа не навчається. Це є важливим, оскільки система в такому випадку не буде схильною до екстремальної поведінки.

Налаштування параметрів нейронної мережі відбувається в іншій фазі — навчальній. При навчанні використовуються еталонні значення пар «входи — виходи», які дозволяють надати певні характеристики поведінки нейронній мережі. Навчальний алгоритм модифікує окремі нейрони мережі та ваги зв'язків між ними таким чином, щоби поведінка мережі відповідала бажаній. Так, у результаті налаштування нейронна мережа розраховуватиме вихідні сигнали на рівні близько до еталонних даних при відповідних вхідних сигналах. Оптимізацію параметрів нейронних мереж персептронного типу можна здійснити за допомогою спеціально розроблених алгоритмів, серед яких найпопулярнішим є метод, що називається узагальненим дельта-правилом або алгоритмом зворотного поширення помилки (Error Back-Propagation Algorithm).

Метод зворотного поширення помилки ґрунтується на ефективному розрахунку частинних похідних критерію оптимізації за всіма параметрами моделі, які підлягають налаштуванню, для заданого вектора вхідних даних  $\{x_1, \dots, x_i, \dots, x_n\}$ . У якості такого критерію оптимізації часто приймається величина, що є пропорційною квадрату різниці між реальними значеннями вихідних змінних та розрахунку мережі на підставі вектора вхідних даних для  $t$ -го зразка:

$$\varepsilon_t = \frac{1}{2} \sum_{j=1}^m (y_j^t - \hat{y}_j^t)^2, \quad (4)$$

де  $y_j^t$  та  $\hat{y}_j^t$  — реальне та розраховане моделлю згідно функції (3) значення  $j$ -ої вихідної змінної,  $j = \overline{1, m}$ , на  $t$ -му кроці навчання.

Така функція помилки (4) є зручною при використанні градієнтних методів оптимізації у зв'язку з її простою похідною. Сутність алгоритму зворотного поширення помилки полягає в оптимізації параметрів нейронної мережі з метою мінімізації функції похибки моделювання (4). Оптимізації підлягають вагові коефі-

цієнти міжнейронних зв'язків, параметри зміщення в суматорах нейронів та параметри функцій активації нейронів.

Для різних комбінацій параметрів нейронної мережі її погрішність можна відобразити точкою у багатовимірному просторі (кількість вимірів дорівнює кількості параметрів моделі плюс один — для відображення похибки). Всі такі точки утворюють деяку гіперповерхню — поверхню станів. Ціль навчання нейронної мережі полягає в знаходженні найнижчої точки на цій багатовимірній поверхні станів. Відштовхуючись від випадково обраної точки на гіперповерхні (від початкової конфігурації параметрів нейронної мережі), алгоритм навчання поступово відшукує найближчий мінімум шляхом розрахунку вектора градієнта поверхні похибок. Цей вектор складається з частинних похідних, котрі вказують напрям найкоротшого спуску по поверхні із заданої точки.

Частинні похідні дозволяють дослідити залежність функції похибки моделювання від одного аргументу. Якщо змінювати значення параметру налаштування в напрямі, в якому похідна є від'ємною, то, відповідно, зменшується похибка навчання (відбувається мінімізація критерію оптимізації). У цьому метод зворотного поширення помилки є своєрідною реалізацією методу градієнтного спуску в просторі параметрів нейронних мереж прямого розповсюдження, таких як на рис. 2. Зрештою, оптимізаційний алгоритм зупиняється в нижній точці поверхні станів (похідна тут дорівнюватиме нулю), що може виявитися лише локальним мінімумом.

Алгоритм працює ітеративно, його кроки називаються епохами. На кожній епосі на вхід мережі почергово подаються всі навчальні приклади, розраховані значення виходів мережі порівнюються з реальними значеннями вихідних змінних й обчислюється похибка (4). Значення похибки, а також градієнта поверхні станів використовують для корекції параметрів нейронної мережі, і дії повторюються. Модифікація всіх параметрів налаштування здійснюється шляхом зміни їх попередніх значень у напрямі зменшення відповідних частинних похідних функції похибки моделювання (4):

$$w(t+1) = w(t) - \eta \frac{\partial \varepsilon_t(w, b, k)}{\partial \omega}, \quad (5)$$

$$b(t+1) = b(t) - \eta \frac{\partial \varepsilon_t(w, b, k)}{\partial b}, \quad (6)$$

$$k(t+1) = k(t) - \eta \frac{\partial \varepsilon_t(w, b, k)}{\partial k}, \quad (7)$$

де  $\varepsilon_t(w, b, k)$  — функція похибки моделювання, що залежить від складових вектора ваг міжнейронних зв'язків мережі ( $w$ ), вектора параметрів зміщення суматорів нейронів ( $b$ ) та вектора параметрів функцій активації нейронів ( $k$ );

$\eta$  — розмір кроку наближення.

Здійснимо розрахунок функцій налаштування параметрів моделі в скалярній формі на прикладі нейронної мережі, зображеної на рис. 2. Відповідно, всі позначення у подальших перетвореннях будуть співпадати з тими, що наведені вище. Далі ми узагальнимо отримані функціональні співвідношення до нейронних мереж будь-якої структури.

Зауважимо, що оскільки процедура налаштування параметрів нейронів та міжнейронних зв'язків, які відносяться до вихідного шару, відрізняється від налаштування параметрів інших шарів, то окремо розпишемо їх функції модифікації. Розрахунок частинних похідних критерію оптимізації по вагам зв'язків, параметрам зміщення та параметрам функцій активації нейронів вихідного шару здійснюватиметься згідно співвідношень:

$$\frac{\partial \varepsilon_t}{\partial w_{lj}^{(3)}} = \frac{\partial \varepsilon_t}{\partial s_j^{(3)}} \cdot \frac{\partial s_j^{(3)}}{\partial w_{lj}^{(3)}}, \quad l = \overline{1, p}, \quad j = \overline{1, m}, \quad (8)$$

$$\frac{\partial \varepsilon_t}{\partial b_j^{(3)}} = \frac{\partial \varepsilon_t}{\partial s_j^{(3)}} \cdot \frac{\partial s_j^{(3)}}{\partial b_j^{(3)}}, \quad j = \overline{1, m}, \quad (9)$$

$$\frac{\partial \varepsilon_t}{\partial k_j^{(3)}} = \frac{\partial \varepsilon_t}{\partial y_j} \cdot \frac{\partial y_j}{\partial k_j^{(3)}}, \quad j = \overline{1, m}, \quad (10)$$

де  $k_j^{(3)}$  — параметр налаштування функції активації  $j$ -го,  $j = \overline{1, m}$ , нейрона вихідного (третього на рис. 2) шару персептрону (коефіцієнт стиснення-розтягування для сигмоїдної функції (2));

$s_j^{(3)}$  — розрахунок суматора  $j$ -го,  $j = \overline{1, m}$ , нейрона третього шару нейронної мережі:

$$s_j^{(3)} = \sum_{u=1}^p w_{uj}^{(3)} \cdot \widehat{y}_u^{(2)} + b_j^{(3)}, \quad (11)$$

де  $\widehat{y}_u^{(2)}$  — вихід  $u$ -го,  $u = \overline{1, p}$ , нейрона попереднього (другого на рис. 2) шару персептрону.

Розпишемо похідні функції суматора нейрона вихідного шару (11) по показниках ваги міжнейронних зв'язків та параметрам зміщення в суматорах, які потрібні для розрахунку відповідних частинних похідних функції помилки моделювання (8) та (9):

$$\frac{\partial s_j^{(3)}}{\partial w_{ij}^{(3)}} = \frac{\partial}{\partial w_{ij}^{(3)}} \left( \sum_{u=1}^p w_{uj}^{(3)} \cdot \hat{y}_u^{(2)} + b_j^{(3)} \right) = \hat{y}_1^{(2)}, \quad (12)$$

$$\frac{\partial s_j^{(3)}}{\partial b_j^{(3)}} = \frac{\partial}{\partial b_j^{(3)}} \left( \sum_{u=1}^p w_{uj}^{(3)} \cdot \hat{y}_u^{(2)} + b_j^{(3)} \right) = 1. \quad (13)$$

Враховуючи функції (3) та (4), запишемо частинну похідну функції помилки моделювання по аргументу (розрахунку суматора нейрона вихідного шару) у вигляді:

$$\frac{\partial \varepsilon_t}{\partial s_j^{(3)}} = \frac{\partial \varepsilon_t}{\partial y_j} \cdot \frac{\partial \hat{y}_j^t}{\partial s_j^{(3)}} = - \left( y_j^t - \hat{y}_j^t \right) \cdot \psi_j^{\prime(3)}(s). \quad (14)$$

Зазначимо, що другою складовою рівняння (14) є похідна функції активації нейрона від аргументу. Саме через цю складову для застосування алгоритму зворотного поширення помилки до нейронної мережі необхідно відбирати диференційовані функції активації нейронів. Враховуючи зазначене вище, перепишемо рівняння розрахунку частинної похідної функції помилки моделювання по параметру функції активації нейрона вихідного шару (10):

$$\frac{\partial \varepsilon_t}{\partial k_j^{(3)}} = \frac{\partial \varepsilon_t}{\partial y_j} \cdot \frac{\partial \hat{y}_j^t}{\partial k_j^{(3)}} = - \left( y_j^t - \hat{y}_j^t \right) \cdot \psi_j^{\prime(3)}(k). \quad (15)$$

Приймаючи до уваги наведені вище перетворення, корекція вхідних вагових зв'язків, параметрів зміщення суматорів та параметрів функцій активації нейронів вихідного шару буде здійснюватись відповідно до рівнянь:

$$w_{lj}^{(3)}(t+1) = w_{lj}^{(3)}(t) + \eta_{w_{lj}}^{(3)} \cdot \left( y_l^t - \hat{y}_l^t \right) \cdot \hat{y}_1^{(2)} \cdot \psi_j^{\prime(3)}(s), \quad l = \overline{1, p}, \quad j = \overline{1, m}, \quad (16)$$

$$b_j^{(3)}(t+1) = b_j^{(3)}(t) + \eta_{b_j}^{(3)} \cdot \left( y_j^t - \hat{y}_j^t \right) \cdot \psi_j^{\prime(3)}(s), \quad j = \overline{1, m}, \quad (17)$$

$$k_j^{(3)}(t+1) = k_j^{(3)}(t) + \eta_{k_j}^{(3)} \cdot \left( y_j^t - \hat{y}_j^t \right) \cdot \psi_j^{\prime(3)}(k), \quad j = \overline{1, m}, \quad (18)$$

де  $\eta_{w_{ij}}^{(3)}$ ,  $\eta_{b_j}^{(3)}$ ,  $\eta_{k_j}^{(3)}$  — розмір кроку наближення при оптимізації ваг зв'язків та відповідних параметрів  $j$ -го,  $j = \overline{1, m}$ , нейрона вихідного (третього на рис. 2) шару персептрону.

Зауважимо, що рівняння (16) — (18) придатні для корекції ваг зв'язків та параметрів нейронів лише останнього шару нейронної мережі, оскільки для них відомі еталонні значення виходів  $y_j^t$ . Однак для нейронів попередніх шарів бажані значення їх виходів невідомі. Відповідно, для модифікації значень параметрів нейронів та міжнейронних зв'язків для інших шарів необхідно володіти інформацією щодо нових значень параметрів нейронної мережі на послідовуючих шарах. Ці нові значення після їх розрахунку послідовно передаються з останніх шарів до попередніх, через що алгоритм зворотного поширення помилки й отримав свою назву.

Принцип обчислення частинних похідних для параметрів нейронів та ваг зв'язків попередніх шарів збігається з викладеним для останнього шару, тому повторюватись у дрібниціях не будемо. Принципова відмінність з'являється у співвідношенні (14) при розрахунку частинної похідної функції похибки по виходу нейрона попереднього шару:

$$\begin{aligned} \frac{\partial \varepsilon_t}{\partial \hat{y}_j^{(2)}} &= \sum_{j=1}^m \left[ \frac{\partial \varepsilon_t}{\partial s_j^{(3)}} \cdot \frac{\partial s_j^{(3)}}{\partial \hat{y}_j^{(2)}} \right] = \sum_{j=1}^m \left[ \frac{\partial \varepsilon_t}{\partial s_j^{(3)}} \cdot \frac{\partial}{\partial \hat{y}_j^{(2)}} \left( \sum_{u=1}^p w_{uj}^{(3)} \cdot \hat{y}_u^{(2)} + b_j^{(3)} \right) \right] = \\ &= - \sum_{j=1}^m \left[ (y_j^t - \hat{y}_j^t) \cdot \psi_j^{(3)}(s) \cdot w_{lj}^{(3)} \right]. \end{aligned} \quad (19)$$

Враховуючи (19), запишемо частинні похідні функції помилки моделювання по виходу суматора та параметру функції активації нейрона другого шару у вигляді:

$$\frac{\partial \varepsilon_t}{\partial s_1^{(2)}} = \frac{\partial \varepsilon_t}{\partial \hat{y}_1^{(2)}} \cdot \frac{\partial \hat{y}_1^{(2)}}{\partial s_1^{(2)}} = - \sum_{j=1}^m \left[ (y_j^t - \hat{y}_j^t) \cdot \psi_j^{(3)}(s) \cdot w_{lj}^{(3)} \right] \cdot \psi_1^{(2)}(s), \quad (20)$$

$$\frac{\partial \varepsilon_t}{\partial k_1^{(2)}} = \frac{\partial \varepsilon_t}{\partial \hat{y}_1^{(2)}} \cdot \frac{\partial \hat{y}_1^{(2)}}{\partial k_1^{(2)}} = - \sum_{j=1}^m \left[ (y_j^t - \hat{y}_j^t) \cdot \psi_j^{(3)}(s) \cdot w_{lj}^{(3)} \right] \cdot \psi_1^{(2)}(k). \quad (21)$$

У такому випадку функції корегування ваг міжнейронних зв'язків, параметрів зміщення суматорів та параметрів функцій активації нейронів другого шару персептрону набудуть вигляд:

$$w_{il}^{(2)}(t+1) = w_{il}^{(2)}(t) + \eta_{w_{il}}^{(2)} \cdot \sum_{j=1}^m \left[ (y_j^t - \bar{y}_j^t) \cdot \psi_j^{(3)}(s) \cdot w_{ij}^{(3)} \right] \cdot \bar{y}_i^{(1)} \cdot \psi_1^{(2)}(s), \quad i = \overline{1, n},$$

$$l = \overline{1, p}, \quad (22)$$

$$b_l^{(2)}(t+1) = b_l^{(2)}(t) + \eta_{b_l}^{(2)} \cdot \sum_{j=1}^m \left[ (y_j^t - \bar{y}_j^t) \cdot \psi_j^{(3)}(s) \cdot w_{ij}^{(3)} \right] \cdot \psi_l^{(2)}(s), \quad l = \overline{1, p}, \quad (23)$$

$$k_l^{(2)}(t+1) = k_l^{(2)}(t) + \eta_{k_l}^{(2)} \cdot \sum_{j=1}^m \left[ (y_j^t - \bar{y}_j^t) \cdot \psi_j^{(3)}(s) \cdot w_{ij}^{(3)} \right] \cdot \psi_l^{(2)}(k), \quad l = \overline{1, p}, \quad (24)$$

де  $\eta_{w_{il}}^{(2)}$ ,  $\eta_{b_l}^{(2)}$ ,  $\eta_{k_l}^{(2)}$  — розмір кроку наближення при оптимізації ваг зв'язків та відповідних параметрів  $l$ -го,  $l = \overline{1, p}$ , нейрона другого шару персептрону.

Подібним чином здійснюється корегування параметрів нейронів та міжнейронних зв'язків усіх шарів, крім вихідного. З наведених корегувальних функцій (16) — (18) та (22) — (24) видно, що для кожного нейрону може бути застосовано функцію активації власного виду. Однак, щоби не робити розрахунки надто громіздкими, здійснимо подальші перетворення за умови, що функції активації всіх нейронів будуть сигмоїдними. Так, похідні від сигмоїдної логістичної функції (2) по розрахунку суматора  $s$  та по параметру функції активації  $k$  обчислюються так:

$$\psi'(s) = \frac{ke^{-ks}}{(1+e^{-ks})^2} = \frac{k \cdot (1-\psi(s))}{\psi(s)} \psi^2(s) = k \cdot \psi(s) \cdot (1-\psi(s)), \quad (25)$$

$$\psi'(k) = \frac{se^{-ks}}{(1+e^{-ks})^2} = \frac{s \cdot (1-\psi(s))}{\psi(s)} \psi^2(s) = s \cdot \psi(s) \cdot (1-\psi(s)). \quad (26)$$

Отже, похідні функцій активації нейронів вихідного та другого шару нейронної мережі набудуть вигляд:

$$\psi_j^{(3)}(s) = k_j^{(3)} \cdot \psi_j^{(3)}(s) \cdot (1-\psi_j^{(3)}(s)) = k_j^{(3)} \cdot \bar{y}_j^t \cdot (1-\bar{y}_j^t), \quad j = \overline{1, m}, \quad (27)$$

$$\psi_l^{(2)}(s) = k_l^{(2)} \cdot \psi_l^{(2)}(s) \cdot (1-\psi_l^{(2)}(s)) = k_l^{(2)} \cdot \bar{y}_l^{(2)} \cdot (1-\bar{y}_l^{(2)}), \quad l = \overline{1, p}, \quad (28)$$

$$\psi_j^{(3)}(k) = s_j^{(3)} \cdot \psi_j^{(3)}(s) \cdot (1-\psi_j^{(3)}(s)) = \left( \sum_{u=1}^p w_{uj}^{(3)} \cdot \bar{y}_u^{(2)} + b_j^{(3)} \right) \cdot \bar{y}_j^t \cdot (1-\bar{y}_j^t),$$

$$j = \overline{1, m}, \quad (29)$$

$$\psi_1^{(2)}(k) = s_1^{(2)} \cdot \psi_1^{(2)}(s) \cdot (1 - \psi_1^{(2)}(s)) = \left( \sum_{u=1}^n w_{ul}^{(2)} \cdot \hat{y}_u^{(1)} + b_l^{(2)} \right) \cdot \hat{y}_1^{(2)} \cdot (1 - \hat{y}_1^{(2)}),$$

$$l = \overline{1, p}. \quad (30)$$

З рівнянь (29) та (30) видно, що корекція параметрів функції активації нейрона повинна здійснюватись після модифікації інших параметрів цього нейрона (параметрів зміщення суматора та ваг зв'язків з нейронами попереднього шару). Після модифікації всіх параметрів нейронів одного шару оптимізаційний алгоритм переходить до попереднього шару і зміна параметрів мережі здійснюється у тому ж порядку. Після того як була проведена модифікація всіх параметрів нейронної мережі у відповідності до поточного навчального прикладу, налаштування моделі здійснюється відповідно до наступного прикладу із навчальної вибірки. Процес навчання припиняється згідно з одним із критеріїв: якщо пройдено певну кількість епох, якщо похибка моделювання (4) досягає деякого рівня малості або якщо похибка перестає зменшуватись.

Отже, вище було проведено розрахунки з отримання функцій налаштування всіх параметрів перцептронну із застосуванням алгоритму зворотного поширення помилки на прикладі конкретної нейронної мережі, зображеної на рис. 2. Розглянута процедура легко може бути узагальнена для нейронної мережі типу перцептрон із довільною кількістю шарів. З цією метою функції корегування параметрів нейронної мережі перепишемо у загальному вигляді. Зауважимо, що позначення індексів випадкових величин тут не збережеться. Отже, алгоритм зворотного поширення помилки при налаштуванні параметрів нейронних мереж прямого розповсюдження складається з таких етапів:

Етап 1 (Ініціалізація параметрів мережі). Встановлюються початкові значення ваговим коефіцієнтам міжнейронних зв'язків, параметрам зміщення в суматорах та параметрам функцій активації нейронів. При ініціалізації параметрам мережі присвоюються випадкові малі величини.

Етап 2 (Формування навчальної вибірки). Налаштування параметрів перцептронну здійснюється за принципом «навчання з вчителем» — коли для кожного вектора вхідних даних  $\{x_1^t, \dots, x_i^t, \dots, x_n^t\}$  відомі відповідні значення вихідних змінних  $\{y_1^t, \dots, y_j^t, \dots, y_m^t\}$ . І навчання мережі зводиться до пошуку параметрів, які найкращим чином будуть відтворювати значення

вихідних змінних на підставі відповідних значень вхідних даних для усієї навчальної вибірки. З цією метою і виникає необхідність формування відповідної статистичної бази, що міститиме еталонні приклади векторів вхідних та вихідних даних для навчання нейронної мережі.

Етап 3 (Розрахунок виходу мережі). На підставі значень вектору вхідних даних здійснюється обчислення вихідних змінних. Приклад таких розрахунків наведено у функції (3) для персептрону, зображеного на рис. 2.

Етап 4 (Модифікація параметрів моделі). Правило корекції параметрів нейронної мережі ґрунтується на мінімізації квадратичної функції похибки (4) градієнтним методом. Згідно описаної вище процедури модифікація всіх параметрів налаштування здійснюється шляхом зміни їх попередніх значень у напрямі зменшення відповідних частинних похідних:

$$w_{ij}^{(r)}(t+1) = w_{ij}^{(r)}(t) + \eta_{w_{ij}}^{(r)} \cdot \delta_j^{(r)} \cdot k_j^{(r)} \cdot \widehat{y}_i^{(r-1)}, \quad i = \overline{1, m_{r-1}}, \quad j = \overline{1, m_r}, \quad (31)$$

$$b_j^{(r)}(t+1) = b_j^{(r)}(t) + \eta_{b_j}^{(r)} \cdot \delta_j^{(r)} \cdot k_j^{(r)}, \quad j = \overline{1, m_r}, \quad (32)$$

$$k_j^{(r)}(t+1) = k_j^{(r)}(t) + \eta_{k_j}^{(r)} \cdot \delta_j^{(r)} \cdot \left( \sum_{u=1}^{m_{r-1}} w_{uj}^{(r)} \cdot \widehat{y}_u^{(r-1)} + b_j^{(r)} \right), \quad j = \overline{1, m_r}, \quad (33)$$

де  $w_{ij}^{(r)}(t)$  — вага міжнейронного зв'язку між  $i$ -им,  $i = \overline{1, m_{r-1}}$ , нейроном  $(r-1)$ -го шару та  $j$ -им,  $j = \overline{1, m_r}$ , нейроном  $r$ -го шару нейронної мережі на  $t$ -му кроці навчання;

$b_j^{(r)}$ ,  $k_j^{(r)}$  — параметри зміщення суматора та функції активації  $j$ -го,  $j = \overline{1, m_r}$ , нейрона  $r$ -го шару персептрону;

$m_r$  — кількість нейронів  $r$ -го шару персептрону;

$\eta_{w_{ij}}^{(r)}$ ,  $\eta_{b_j}^{(r)}$ ,  $\eta_{k_j}^{(r)}$  — розмір кроку наближення при оптимізації ваг зв'язків, параметрів зміщення суматора та функції активації  $j$ -го,  $j = \overline{1, m_r}$ , нейрона  $r$ -го шару нейронної мережі;

$\widehat{y}_j^{(r)}$  — розрахунок виходу  $j$ -го,  $j = \overline{1, m_r}$ , нейрона  $r$ -го шару персептрону;

$\delta_j^{(r)}$  — швидкість зміни помилки для  $j$ -го,  $j = \overline{1, m_r}$ , нейрона  $r$ -го шару персептрону. Якщо нейрон розташований в останньому шарі мережі, то

$$\delta_j^{(r)} = (y_j^t - \widehat{y}_j^t) \cdot \widehat{y}_j^t \cdot (1 - \widehat{y}_j^t). \quad (34)$$

Для нейронів інших шарів

$$\delta_j^{(r)} = \sum_{u=1}^{m_{r+1}} [k_u^{(r+1)} \cdot \delta_u^{(r+1)} \cdot w_{ju}^{(r+1)}] \cdot \hat{y}_j^{(r)} \cdot (1 - \hat{y}_j^{(r)}). \quad (35)$$

Через використання для позначень змінних в алгоритмі зворотного поширення помилки грецької літери  $\delta$ , цю процедуру навчання ще називають узагальненим дельта-правилом.

Етап 5 (Перевірка адекватності моделі). Розраховується похибка моделювання (4) і якщо вона є меншою за мінімально-встановлений рівень, то винайдене рішення приймається за оптимальне і процес навчання завершується. Інакше алгоритм переходить до наступного навчального прикладу і навчання повторюється з третього етапу. Проте, алгоритм зворотного поширення помилки може зупинитись і за умови перевищення кількості епох або часу навчання, якщо протягом певної кількості ітерацій похибка навчання не зменшувалась тощо.

Описана вище послідовність кроків врешті приводить до знаходження оптимальних значень параметрів нейронної мережі, що мінімізують функцію помилки моделювання (4). Важливим питанням при оптимізації стає вибір довжини кроків наближення. При великій довжині кроку збіжність буде швидшою, але існує небезпека перестрибнути оптимальне рішення. При маленькому кроці правильний напрямок буде виявлено, але зростає кількість ітерацій. Удосконаленням алгоритму є корегування розміру кроку за коефіцієнтом, що визначає швидкість навчання.

При розрахунку цього коефіцієнту може враховуватись крутизна схилу (по мірі наближення до дна розмір кроку буде зменшуватись) та час (чим довше здійснюється оптимізація, тим меншим стає крок наближення). На практиці при розрахунку розміру кроку наближення іноді враховують так званий імпульс, який сприяє просуванню алгоритму у фіксованому напрямку. Так, якщо поверхня станів на деякій ділянці є надто пологою і при оптимізації було зроблено певну кількість кроків у одному напрямку, тоді швидкість навчання збільшується, що сприяє підвищенню ефективності навчального алгоритму.

Проте існують такі структури рельєфу поверхні станів, для яких зазначені удосконалення алгоритму не дадуть очікуваного позитивного ефекту. Наприклад, у разі потрапляння в локальний мінімум поверхні станів, градієнтний підхід не в змозі вийти з цієї пастки та продовжити пошук глобального оптимуму конфігурації системи. А таких локальних мінімумів на багатовимірній поверхні станів за-

звичай є безліч. Це обумовлює доцільність доповнення градієнтних методів елементами стохастичної оптимізації, щоб запобігти зупинці алгоритму навчання в таких локальних мінімумах.

У такому разі після винайдення локального мінімуму на поверхні станів із застосуванням методу зворотного поширення помилки здійснюється випадковий вибір нового початкового наближення і знову починається оптимізація на базі градієнтного підходу. Упорядкування випадкового вибору початкового наближення реалізовано, наприклад, у генетичних або мурашкових алгоритмах (цим алгоритмам буде приділено увагу в подальших наших дослідженнях). Потім серед винайдених мінімумів на поверхні станів обирається найглибший, який і приймається за глобальний. Таким чином, подібна комбінація градієнтних та стохастичних методів оптимізації дозволяє розраховувати на знаходження глобального оптимуму конфігурації мережі (або іншої конфігурації, що забезпечує максимально низьке значення функції помилки моделювання).

Отже, у статті наведено результати розрахунків з удосконалення алгоритму зворотного поширення помилки в напрямі реалізації можливості налаштування нейронних мереж прямого розповсюдження за всіма параметрами. Також надано низку пропозицій з підвищення ефективності цього алгоритму шляхом упровадження процедур корегування розміру кроку наближення та його поєднання із стохастичними методами оптимізації.

### **Література**

1. McCulloch W.S., Pitts W. A logical calculus of the ideas immanent in nervous activity // Bulletin of Mathematical Biophysics. — 1943. — Vol. 5. — P. 115—133.
2. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review. — 1958. — № 65. — P. 386—407.
3. Мороз О. В., Матвійчук А. В. Оптимальне управління економічними системами в умовах невизначеності та ризику: Монографія. — Вінниця: УНІВЕРСУМ-Вінниця, 2003. — 177 с.
4. Матвійчук А. В. Дослідження залежності якості прогнозування курсів цінних паперів нейронними мережами від форми подання вхідних даних // Збірник наукових праць Черкаського державного технологічного університету. Серія: Економічні науки. — 2003. — Вип. 8. — С. 147—156.
5. Матвійчук А. В. Прогнозирование банкротств предприятий с использованием инструментария нейронных сетей // Управление финансовыми рисками. — 2008. — № 04(16). — С. 280—288.
6. Вітлінський В. В., Матвійчук А. В., Махоткіна А. Я. Методологічний підхід до розподілу бюджетних призначень з податку на додану

вартість за регіонами // Моделювання та інформаційні системи в економіці. — К.: КНЕУ, 2008. — Вип. 78. — С. 191—208.

7. *Rummelhart D. E., Hinton G. E., Williams R. J.* Learning Representations of Back-Propagation Errors // *Nature*. — 1986. — Vol. 323. — P. 533—536.

8. *Матвійчук А. В.* Виявлення і запобігання ефекту перенавчання нейронної мережі // Збірник наукових праць Черкаського державного технологічного університету. Серія: Економічні науки. — 2002. — Вип. 3. — С. 124—131.

Стаття надійшла до редакції 20.01.09 р.

УДК 332.053

**А. П. Великий**, член-кор.НАН України,  
**О. Д. Шарапов**, канд. техн. наук, професор,  
ДВНЗ «Київський національний економічний  
університет імені Вадима Гетьмана»

## **ДИНАМІЧНІ БАЛАНСОВІ СПІВВІДНОШЕННЯ В АГРЕГОВАНІЙ МОДЕЛІ З ІНВЕСТИЦІЙНИМ ЛАГОМ**

*АНОТАЦІЯ.* Аналізуються можливі застосування динамічної моделі міжгалузевого балансу (агрегованої) для опрацювання і аналізу розрахункових темпів економічного зростання залежності від капітальних витрат, обсягів основних виробничих засобів та інших параметрів моделі.

*ANNOTATION.* The possible application of dynamic model of intersectoral balance (aggregated) for processing and analysis of calculated growth based on capital costs, the volume of basic production funds and other parameters of the model is analysed.

**КЛЮЧОВІ СЛОВА:** динамічна модель, виробничі запаси, капітальні витрати, інвестиційний лаг.

В даній роботі продовжується розгляд динамічної моделі міжгалузевого балансу (агрегованої), наведеної в [1]. Наводяться результати варіантних розрахунків і прослідковується взаємодія інвестиційних витрат на динаміку створення основних виробничих засобів (ОВЗ) відповідно до наявності інвестиційного лагу.