

УДК 004.414.23

Л. В. Передерій

**ПРОГНОЗУВАННЯ ДЛЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ
В УПРАВЛІННІ ЕКОНОМІЧНИМИ ОБ'ЄКТАМИ
(II ЧАСТИНА)**

Анотація. Стаття присвячена огляду методу штучних нейронних мереж і застосуванню його та методу часових рядів при розробці прогнозної системи підтримки прийняття рішень, призначеної для підвищення ефективності управління економічними об'єктами.

Ключові слова: штучна нейронна мережа, прогноз, генерація прогнозу, управління прогнозом, період прогнозування, горизонт прогнозування, інтервал прогнозування, математична модель, часові послідовності.

Summary. The article is devoted the review of method of artificial neuron networks and application of him and method of sentinel rows at development of the prognosis system of support of making a decision, efficiency of management economic objects intended for an increase.

Key words: artificial neuron network, prognosis, generation of prognosis, management a prognosis, period of prognostication, horizon of prognostication, interval of prognostication, mathematical model, sentinel sequences.

Серед різних структур нейронних мереж (НМ) однією із найбільш відомих є багатошарова структура, в якій кожен нейрон довільного шару пов'язаний зі всіма аксонами нейронів попереднього шару або, у випадку першого шару, зі всіма входами НМ. Такі НМ називаються повнозв'язними.

Коли в мережі тільки один шар, алгоритм її навчання із вчителем досить очевидний, оскільки

правильні вихідні стани нейронів єдиного шару свідомо відомі, і підстроювання синаптичних зв'язків йде в напрямі, що мінімізує помилку на виході мережі. За цим принципом будується, наприклад, алгоритм навчання одношарового перцептрону. У багатошарових же мережах оптимальні вихідні значення нейронів всіх шарів, окрім останнього, як правило, не відомі, і двошаровий або більше

© Л. В. Передерій, 2014

персептрон вже неможливо навчити, керуючись тільки величинами помилок на виходах НМ.

Один із варіантів вирішення цієї проблеми — розробка наборів вихідних сигналів, відповідних вхідним, для кожного шару НМ, що, звичайно, є дуже трудомісткою операцією і не завжди може бути здійснено.

Другий варіант — динамічне підстроювання вагових коефіцієнтів синапсів, у ході якого вибираються, як правило, найбільш слабкі зв'язки і змінюються на малу величину в той або інший бік, а зберігаються тільки ті зміни, які спричинили зменшення помилки на виході всієї мережі. Третій варіант — розповсюдження сигналів помилки від виходів НМ до її входів у напрямі, зворотному прямому розповсюдженню сигналів у звичайному режимі роботи.

Що стосується інформаційної ємкості штучних нейронних мереж, то для мереж з кількістю шарів більше двох це питання залишається відкритим. Як показано в [3], для НМ з двома шарами, тобто вихідним і одним прихованим шаром, детерміністська ємкість мережі C_d оцінюється таким чином:

$$N_w/N_y < C_d < N_w/N_y \cdot \log(N_w/N_y), \quad (1)$$

де N_w — кількість підстроюваних вагів;
 N_y — кількість нейронів у вихідному шарі;
 C_d — детерміністська ємкість мережі.

Слід зазначити, що цей вираз отриманий з урахуванням деяких обмежень. По-перше, кількість входів N_x і нейронів у прихованому шарі N_h має задовольняти нерівності $N_x + N_h > N_y$. По-друге, $N_w/N_y > 1000$. Проте вищенаведена оцінка виконувалася для мереж із активаційними функціями нейронів у виді порогу, а ємкість мереж із гладкими активаційними функціями зазвичай більша [4]. Крім того, прикметник «детерміністський», що фігурує в назві ємкості означає, що отримана оцінка ємкості підходить абсолютно для всіх можливих вхідних образів, які можуть бути наведені N_x входами.

Насправді розподіл вхідних образів, як правило, володіє деякою регулярністю, що дозволяє НМ проводити узагальнення і, таким чином, збільшувати реальну ємкість. Оскільки розподіл образів у загальному випадку заздалегідь не відомий, можливо говорити про таку ємкість тільки імовірно, але зазвичай вона рази в два перевищує ємкість детерміністську.

Окрім питання про ємкість НМ, важливо також з'ясувати питання про необхідну потужність вихідного шару мережі, що виконує остаточну класифікацію образів. Річ у тому, що для розподілення множини вхідних образів, наприклад, за двома класами достатньо всього одного виходу. При цьому кожен логічний рівень — «1» і «0» — позначатиме окремий клас. На двох виходах можна закодувати вже 4 класи і так далі.

Проте результати роботи мережі, організованої таким чином, не дуже надійні. Для підвищення достовірності класифікації бажано ввести надмірність шляхом виділення кожному класу одного нейрону у вихідному шарі або, що ще краще, декілька, кожен з яких навчається визначати приналежність образу до класу зі своєю мірою достовірності, наприклад: високою, середньою і низькою. Такі НМ дозволяють проводити класифікацію вхідних образів, об'єднаних у нечіткі (розмиті або такі, що перетинаються) множини. Ця властивість наближає подібні НМ до умов реального життя.

Після того, як визначено кількість шарів і кількість елементів у кожному з них, необхідно знайти значення вагів і порогів мережі, які б мінімізували помилку прогнозу, що видається мережею. Для цього застосовуються процедури навчання штучних нейронних мереж (ШНМ).

Однією із класичних і найпоширенішою процедурою навчання ШНМ є алгоритм зворотного розповсюдження помилки (back propagation) [5]. Його суть полягає в такому.

Згідно з методом найменших квадратів цільовою функцією помилки НМ, що мінімізується, є величина:

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2, \quad (2)$$

де $y_{j,p}^{(N)}$ — реальний вихідний стан нейрону j вихідного шару N нейронної мережі при подачі на її входи p -го образу;

$d_{j,p}$ — ідеальний (бажаний) вихідний стан цього нейрону.

Підсумовування ведеться за всіма нейронами вихідного шару і за всіма оброблюваними мережею образами. Мінімізація ведеться методом градієнтного спуску, що означає підстроювання вагових коефіцієнтів таким чином:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}}, \quad (3)$$

де w_{ij} — ваговий коефіцієнт синоптичного зв'язку, що сполучає i -й нейрон шару $n - 1$ з j -м нейроном шару n ;

η — коефіцієнт швидкості навчання, $0 < \eta < 1$.

Функція $E(w)$ є складною, і за правилами обчислення похідної від складної функції маємо:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial s_j} \cdot \frac{\partial s_j}{\partial w_{ij}}, \quad (4)$$

де y_j — вихід нейрону j ;

s_j — зважена сума його вхідних сигналів, тобто аргумент активаційної функції.

Оскільки множник dy_j/ds_j є похідною цієї функції за її аргументом, із цього виходить, що похідна активаційної функції має бути визначена на всій осі абсцис. У зв'язку з цим функція одичного стрибка та інші активаційні функції з неоднорідностями не підходять для вказаних НМ. У них застосовуються такі гладкі функції, як гі-

перболічний тангенс або класична сигмоїда з експонентою. У разі гіперболічного тангенсу:

$$\frac{\partial y}{\partial s} = 1 - s^2. \quad (5)$$

Третій множник $\partial s / \partial w_{ij}$ дорівнює виходу нейрону попереднього шару $y_i^{(n-1)}$.

Перший множник у (4) легко розкладається таким чином [5]:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial s_k} \cdot \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial s_k} \cdot w_{jk}^{(n+1)}. \quad (6)$$

Тут підсумовування за k виконується серед нейронів шару $n + 1$.

Ввівши нову змінну:

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \cdot \frac{dy_j}{ds_j}, \quad (7)$$

отримаємо рекурсивну формулу для розрахунків величин $\delta_j^{(n)}$ шару n з величин $\delta_k^{(n+1)}$ більш старшого шару $n + 1$:

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} \cdot w_{jk}^{(n+1)} \right] \cdot \frac{dy_j}{ds_j}. \quad (8)$$

Для вихідного шару формула розрахунку така:

$$\delta_l^{(N)} = (y_l^{(N)} - d_l) \cdot \frac{dy_l}{ds_l}. \quad (9)$$

Запишемо (3) в розкритому виді:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}. \quad (10)$$

Іноді для додання процесу корекції ваг деякої інерційності, що згладжує різкі скачки при переміщенні по поверхні цільової функції, формула (10) доповнюється значенням зміни ваги на попередній ітерації:

$$\Delta w_{ij}^{(n)}(t) = -\eta \cdot (\mu \cdot \Delta w_{ij}^{(n)}(t-1) + (1-\mu) \cdot \delta_j^{(n)} \cdot y_i^{(n-1)}), \quad (11)$$

де μ — коефіцієнт інерційності;
 t — номер поточної ітерації.

Таким чином, повний алгоритм навчання НМ за допомогою процедури зворотного розповсюдження будується шляхом виконання таких шести кроків.

Крок 1. Подати на входи мережі один із можливих образів i в режимі звичайного функціонування НМ, коли сигнали розповсюджуються від входів до виходів, розрахувати значення останніх. Враховуючи, що

$$s_j^{(n)} = \sum_{i=0}^M y_i^{(n-1)} \cdot w_{ij}^{(n)}, \quad (12)$$

де M — кількість нейронів у шарі $n - 1$ з урахуванням нейрона з постійним вихідним станом $m + 1$, що задає зсув;

$$y_i^{(n-1)} = x_i^{(n-1)} - i\text{-й вхід нейрона } j \text{ шару } n.$$

$$y_j^{(n)} = f(s_j^{(n)}), \quad (13)$$

де $f(\dots)$ — сигмоїд.

$$y_q^{(0)} = I_q, \quad (14)$$

де I_q — q -та компонента вектора вхідного образу.

Крок 2. Розрахувати $\delta^{(N)}$ для вихідного шару за формулою (9).

Крок 3. Розрахувати за формулою (10) або (11) зміни ваг $\Delta w^{(N)}$ шару N .

Крок 4. Розрахувати за формулами (8) і (10) (або (8) і (11)) відповідно $\delta^{(n)}$ і $\Delta w^{(n)}$ для решти всіх шарів, $n = N-1, \dots, 1$.

Крок 5. Скоректувати усі ваги в НМ:

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t). \quad (15)$$

Крок 6. Якщо помилка мережі істотна, перейти на крок 1, інакше — кінець.

На першому кроці мережі пред'являються попеременно у випадковому порядку всі тренувальні образи, щоб мережа, образно кажучи, не забувала одні образи у міру запам'ятовування інших. Алгоритм ілюструється на рис. 1.

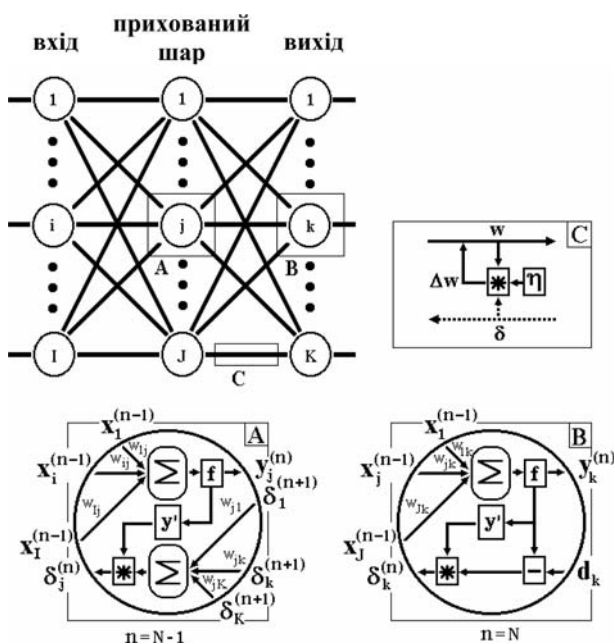


Рис. 1. Діаграма сигналів у мережі при навчанні за алгоритмом зворотного розповсюдження помилки

З виразу (10) виходить, що коли вихідне значення $y_i^{(n-1)}$ прагне до нуля, то ефективність навчання помітно знижується. При двійкових вхідних векторах в середньому половина вагових коефіцієнтів не коректуватиметься [5], тому зону можливих значень виходів нейронів $[0, 1]$ бажано зрушити в межах $[-0.5, +0.5]$, що досягається простими модифікаціями логістичних функцій. Наприклад, сигмоїда з експонентою перетвориться до виду:

$$f(x) = -0.5 + \frac{1}{1 + e^{-\alpha x}}. \quad (16)$$

У процесі навчання може виникнути ситуація, коли великі позитивні або негативні значення вагових коефіцієнтів змістять робочу точку на сигмоїді багатьох нейронів у зону насичення. Малі величини похідної від логістичної функції приведуть, відповідно до (8) і (9), до зупинки навчання, що паралізує НМ.

Окрім того, застосування методу градієнтного спуску не гарантує, що буде знайдений глобальний, а не локальний мінімум цільової функції. Ця проблема пов'язана ще з однією, а саме — з вибором величини швидкості навчання.

Доказ збіжності навчання в процесі зворотного розповсюдження заснований на похідних, тобто прирості вагів, і, отже, швидкість навчання має бути нескінченно малою, і сам процес навчання відбуватиметься неприйнятно поволі. З іншого боку, дуже великі корекції вагів можуть привести до постійної нестійкості процесу навчання. Тому у якості η зазвичай вибирається число, менше 1. Крім того, для виключення випадкових попадань в локальні мінімуми іноді, після того як значення вагових коефіцієнтів стабілізуються, короткочасно сильно їх збільшують, щоб почати градієнтний спуск з нової точки. Якщо повторення цієї процедури кілька разів приведе алгоритм в один і той же стан НМ, можна більш менш упевнено сказати, що знайдений глобальний мінімум.

Якщо прогнозування вирішується за допомогою нейронної мережі, то необхідно зібрати дані для її навчання. Навчальним набором є низка спостережень, для яких вказані значення вхідних і вихідних змінних. Вибір змінних здійснюється виходячи з аналізу наочної зони. Необхідно вибирати такі змінні, які імовірно впливають на результат.

Будь-яка ШНМ приймає на вході числові значення і видає на виході також числові значення. Тому, перед подачею на вхід мережі даних, їх необхідно нормувати. Просте нормування виконують заміною кожної компоненти вхідного вектора даних x_i величиною:

$$x_i = \frac{x_i - (\max x_i + \min x_i) / 2}{(\max x_i - \min x_i) / 2}, \quad (17)$$

де $\max x_i$ і $\min x_i$ — відповідно максимальне і мінімальне значення для даної компоненти, обчислені по всій навчальній вибірці.

За цією ж формулою перераховуються і компоненти векторів відповідей.

Якщо передбачається, що надалі поступлять дані, що сильно відрізняються, то \min і \max величини задаються користувачем за його оцінками. Ці величини повинні вводитися у момент створення мережі і надалі не залежати від навчальної вибірки.

Функції активації елементів мережі зазвичай вибираються так, щоб її вхідний аргумент міг приймати довільні значення, а вихідні значення лежали у строго обмеженому діапазоні. Вихідні сигнали мережі повинні нормуватися в діапазон дійсних значень за оберненими формулами.

Якщо вихідні значення завжди належать деякій обмеженій зоні, а вся інформація має бути наведена в числовому виді, очевидно, що при вирішенні реальних завдань методами ШНМ потрібні етапи попередньої обробки — препроцесування і завершальної обробки — постпроцесування даних [5].

Нейронні мережі можуть ефективно працювати із числовими даними, які знаходяться тільки в певному обмеженому діапазоні [6]. Це створює проблеми в тих випадках, коли дані мають нестандартний масштаб, включають пропущені значення або є номінальними або нечисловими. Для усунення цієї проблеми застосовуються такі методи обробки:

— числові дані масштабуються у відповідний для мережі діапазон;

— пропущені значення замінюються середнім значенням цієї змінної по всіх наявних навчальних вибірках або іншим статистичним значенням;

— номінальні або нечислові змінні перетворюються в числову форму.

За перетворення інформації від вихідних елементів у вихідну змінну і представлення результатів відповідає етап постпроцесування.

Весь алгоритм обробки даних в прогнозній системі підтримки прийняття рішень умовно розбитий на три етапи: препроцесування, основна обробка (прогноз) і постпроцесування з поданням результату ОПР.

Перший етап — препроцесування — передбачає такі дії:

1) вибірка необхідних даних та історичних відомостей досліджуваної наочної зони із різних джерел інформації;

2) визначення мінімального і максимального значення прогнозованої змінної;

3) визначення горизонту прогнозування;

4) нормалізація даних, необхідних для навчання мережі;

5) створення в пам'яті об'єкта обробки;

6) передача даних, необхідних для обробки, у створений об'єкт.

На другому етапі — прогнозуванні — виконуються такі дії:

1) створюється екземпляр нейронної мережі;

2) проводиться процедура навчання мережі;

3) зберігається у файлі матриця значень синоптичних вагів;

4) будується прогноз у одиничному базисі з необхідним горизонтом;

5) підсумовуються значення змінної прогнозування у одиничному базисі за весь період прогнозу;

6) результати прогнозу повертаються у процедуру, що викликає.

Третій етап — постпроцесування — передбачає такі дії:

1) проводиться зворотне шкалювання вихідних даних мережі в природний діапазон значень змінної прогнозування;

2) віднімається величина залишків із кожного об'єму майбутніх продажів товарів;

3) округляється отримана кількість в менший бік до цілого числа тарних місць мінімальної заводської упаковки товару;

4) визначається середнє значення націнки за кожним пунктом номенклатури;

5) обчислюється добуток значень, отриманих в пунктах 3 і 4;

6) сортуються за убаванням кортежі таблиці, виходячи із величини значення, отриманого в пункті 5 добутку;

7) таблиця з відсортованими кортежами зберігається в БД;

8) результати в передбаченій формі звіту надаються ОПР для подальшого прийняття рішення.

Для визначення оптимальних параметрів нейронної мережі було проведено низку експериментів. Метою експериментів було прогнозування об'єму майбутніх продажів. Для досягнення цієї мети було проведено дослідження впливу представлення історичних і прогнозованих даних на помилку прогнозування. Також були розглянуті питання впливу структури НМ на швидкість навчання мережі і помилку прогнозування. При цьому ставилися такі завдання:

- пошук критеріїв прогнозування;
- пошук оптимального представлення історичних даних;
- пошук оптимального представлення результату прогнозування;
- пошук оптимального розміру вікна;
- пошук оптимальної структури мережі.

У експерименті брали участь реальні дані щодо продажу товарів, структура яких наведена у табл. 1.

Таблиця 1

Специфікація вхідних даних для процедури навчання ШНМ

Змінна	Позначення	Тип	Формат
Ідентифікатор	ID	Number	Integer
Кількість	Quantity	Number	Long Integer
Поточний залишок	Rest	Number	Long Integer
Термін придатності	KeepingTerm	Date/Time	Day
			Month
Дата виробництва	MakingDate	Date/Time	Day
			Month
Ціна придбання	Cost	Currency	Double
Ціна продажу	Price	Currency	Double
Дата продажу	SellingDate	Date/Time	
Мінімальна заводська упаковка	PackingMin	Number	Long Integer

Кожен із експериментів проводився у чотири етапи.

Першим етапом було формування навчальної вибірки. На цьому етапі визначався вид представлення історичних і прогнозованих даних і проводилося формування наборів, що подаються на

вхідні нейрони, і відповідних їм наборів, що знімаються з виходів мережі.

Другим етапом проводилося навчання НМ на основі сформованої на першому етапі навчальної вибірки. Якість навчання характеризувалася помилкою навчання, визначуваною як сумарне квадратичне відхилення значень на виходах НМ в навчальній вибірці від реальних значень, отриманих на виходах НМ. Критерієм припинення навчання було проходження мережею 1500 ітерацій або зменшення помилки на виходах мережі на два порядки порівняно з первинною помилкою. У тому випадку, якщо при описі досвіду не вказано, що відбулося зниження помилки на два порядки, навчання має бути зупинене за першим критерієм.

На третьому етапі проводилося тестування навчання мережі. На вхід подавалося приблизно 5 % наборів з навчальної вибірки і визначалася якість розпізнавання мережі. Досвід вважався успішним, якщо відносна достовірність розпізнавання образів була не менше 80 %.

На четвертому етапі проводилася симуляція прогнозування. На вхід мережі подавалися набори, які не були внесені до навчальної вибірки, але результат за ним (прогноз) відомий.

З метою формування необхідного простору ознак номінальна змінна Ідентифікатор (ID) представлена двійковим кодом зі зміщеними, рівнями $[0, 1] \rightarrow [-0.5, +0.5]$ для управління рівнями зсуву нейронів першого шару.

Перед проведенням першої процедури навчання нейронної мережі проводилось формування початкових значень матриці синаптичних вагів мережі шляхом холостого прогону навчання за змінною ID на «нульовій навчальній вибірці», тобто всі значення змінних, за винятком ID, рівні 0.

З метою забезпечення необхідних властивостей екстраполяції ШНМ та усунення ефекту перенавчання мережі у вихідному шарі застосовується лінійна функція активації, яка не володіє зонами насичення і має невисокий коефіцієнт навчання.

На підставі формули (1) та експериментальних даних були визначені такі параметри нейронної мережі:

- кількість шарів нейронів у мережі — 3;
- розмір вікна вхідного вектора — 14;
- кількість нейронів у шарах: у вхідному — 28, у прихованому — 21, у вихідному — 14;
- функція активації сигмоїдна у вхідному та прихованому шарах і лінійна у вихідному шарі.

Архітектура нейронної мережі, побудованої за експериментальними даними, зображена на рис. 2.

Для тестових даних були обрані історичні дані, які фактично є часовими рядами динаміки продажів товарів двох видів за рік. Для прогнозування застосовувалася тришарова нейронна мережа по 28:21:14 нейронів у відповідному шарі,

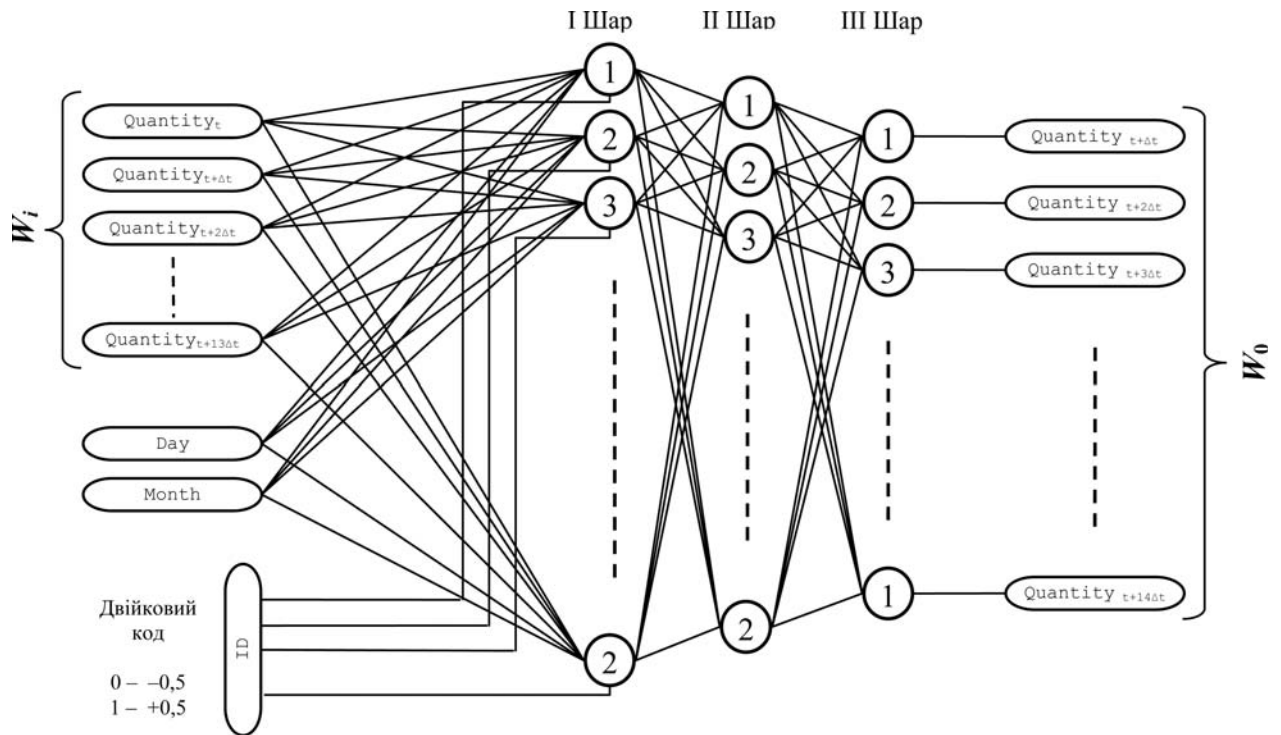
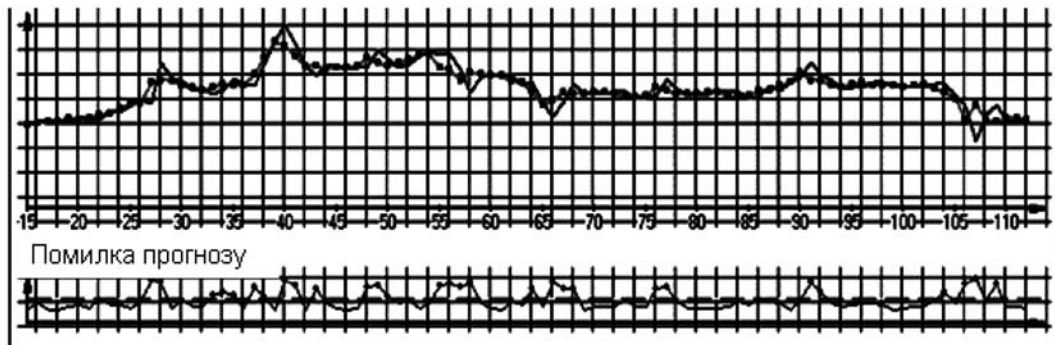


Рис. 2. Архітектура нейронної мережі

яка на основі чотирнадцяти попередніх значень часового ряду передбачала його наступне значення. У якості нелінійного перетворювача використовувалася сигмоїдна функція із характеристич-

ним коефіцієнтом. Навчання НМ здійснювалося за алгоритмом зворотного розповсюдження помилки методом ковзаючого вікна з 14 прикладів. Результати прогнозування наведені на рис. 3.

Графік руху часового ряду (приклад 1)



Графік руху часового ряду (приклад 2)

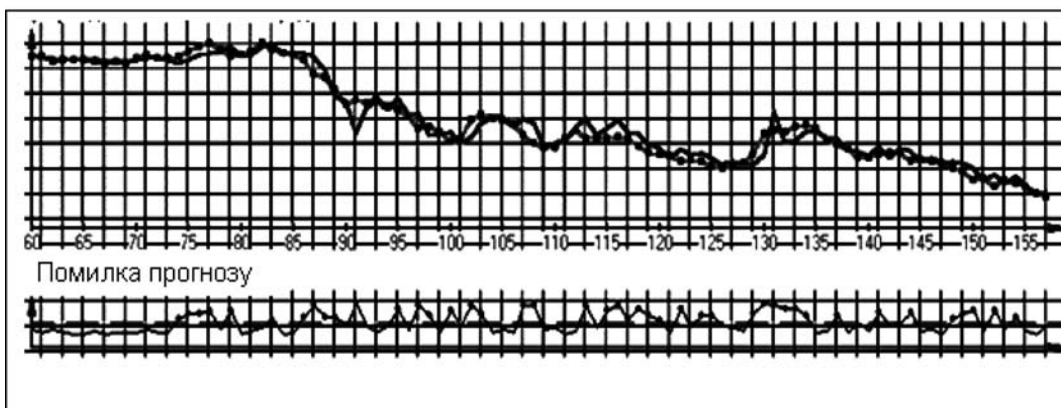


Рис. 3. Прогноз і динаміка продажу двох товарів за 1 рік

Верхні графіки описують динаміку середнього значення попиту на препарат, а також прогноз, зроблений нейронною мережею. Нижні графіки описують помилку прогнозу, поріг і самі розходження.

Ці приклади демонструють, що, не дивлячись на складну динаміку часових рядів, останні за винятком обмеженої кількості ділянок можуть успішно прогнозуватися за допомогою порівняльно нескладної моделі нейронної мережі.

Прогноз такого пристрою безпосередньо не є остаточним рішенням, тобто вихідний сигнал НМ оцінюватиме ОПР на основі своїх знань та досвіду і формуватиме власний експертний висновок. На основі складеного нейронною мережею прогнозу ОПР дістають можливість проаналізувати динаміку розвитку попиту, виявити неочевидні на перший погляд взаємозв'язки, порівняти різні альтернативи, проаналізувати аномалії і прийняти найбільш обґрунтоване рішення.

Реалізація прогновної системи підтримки прийняття рішень здійснюється шляхом інтеграції із широко використовуваним на підприємствах програмним комплексом «ІС: Предприятие», при цьому основна обробка даних (прогноз) здійснюється в зовнішньому модулі, що підключається, для чого розроблений за технологією ActiveX компонент, що містить програмний код обробки. У якості адаптивного предиктора для прогнозування використовується тришарова ШНМ. Навчання мережі проводиться за алгоритмом зворотного розповсюдження помилки; пре- і постпроцесійна обробка даних, а також представлення результатів прогнозування проводиться наявними засобами «ІС: Предприятие», для чого додатково реалізується форма зовнішнього звіту, що містить методи підготовки даних, створення об'єкту СОМ (Component Object Model) і передачі в нього даних для обробки.

Висновки. Створення прогновної системи підтримки прийняття рішень на основі ШНМ — складний, але осяжний процес, що вимагає знання бізнесу, глибоких теоретичних знань методів ШНМ, програмно-технічного інструментарію і досвіду виконання великих проектів. Разом з тим

впровадження подібних систем може дати переваги в бізнесі, які будуть більш вагомими, чим раніше організація почне створення таких систем.

Розглянутий у статті підхід до розробки систем прогнозування базується на побудові штучної нейронної мережі, її навчанні і тестуванні. Якщо навчена НМ успішно пройшла тестування, то її можна використовувати для прогнозування на нових наборах даних. У статті також проаналізовані проблеми, які потрібно вирішувати розробникам систем прогнозування, і запропонований системний підхід до вирішення цих проблем.

Розроблений у складі обліково-аналітичної системи «ІС: Предприятие» програмний модуль системи прогнозування може бути застосований, наприклад, як СППР при здійсненні господарської діяльності, пов'язаної з веденням оптово-роздрібною торгівлі. Застосування цієї системи при плануванні закупівель підвищує ефективність розподілу коштів і знижує вірогідність фінансових втрат, що, зрештою, виражається в позитивному економічному результаті.

Література

1. Паклин Н. Б. Бизнес-аналитика: от данных к знаниям / Н. Б. Паклин, В. И. Орешков. — СПб. : Питер, 2009. — 624 с.
2. Владимирова Л. П. Прогнозирование и планирование в условиях рынка : учебное пособие / Л. П. Владимирова ; рецензент: С. Р. Малютин. — 2-е изд. перераб. и доп. — М. : Издательский дом «Дашков и К^о», 2001. — 308 с.
3. Руденко О. Г. Искусственные нейронные сети : учебное пособие / О. Г. Руденко, Е. В. Бодянский. — Харьков : ООО «Компания СМИТ», 2005. — 408 с.
4. Джеффри Е. Хинтон. Как обучаются нейронные сети / Е. Хинтон Джеффри // В мире науки. — 1992. — № 11–12 — С. 103–107.
5. Барский А. Б. Нейронные сети: распознавание, управление, принятие решений / А. Б. Барский. — М. : Финансы и статистика, 2004. — 176 с.
6. Дюк В. А. Обработка данных на ПК в примерах / В. А. Дюк. — СПб. : Питер, 1997. — 240 с.