

$$R_i = \sum_{k=1}^n \text{Sgn}(x_i - x_k). \quad (8)$$

Сукупність рангів $\{R_1, \dots, R_n\}$ всіх елементів вибірки $\{x_1, \dots, x_n\}$ сполучають деяку перестановку чисел від 1 до n . Згідно гіпотези випадковості (1) всі такі перестановки рівно імовірні. Отже, незалежно від конкретного закону розподілу вихідної вибірки $\{x_1, \dots, x_n\}$ сумісний розподіл рангів $\{R_1, \dots, R_n\}$ є рівномірним:

$$P(R_1, \dots, R_n) = \frac{1}{n!}. \quad (9)$$

Аналіз методів знаходження показує, що приведений перелік інваріантних перетворень не є, безумовно, вичерпним, але він підтверджує неоднозначність в виборі перетворень масиву S . Критерієм вибору є не тільки дослідження заданих інваріантних властивостей рівня неправдоподібних тривог до виду розподілу, але і максимально можливе зберігання інформації про сигнал, що дозволяє його знайти.

Список джерел:

1. В. К. Стеклов, Л. Н. Беркман, Г. А. Возможности использования методов многоканальной модуляции для сетей доступа. Зв'язок, 2000, №4 с. 16-19.
2. В. Ю. Лапий, А. Я. Калюжный, А. Г. Красный „Устройства ранговой обработки информации”. Киев, „Техника”, 1986.
3. Э. А. Корнильев, И. Б. Прокопенко, В. М. Чуприн. Устойчивые алгоритмы в автоматизированных системах обработки информации. Киев „Техника”, 1989.

НЕЧІТКІ НЕЙРОННІ МЕРЕЖІ І ГЕНЕТИЧНІ АЛГОРИТМИ У ЗАДАЧАХ МАКРОЕКОНОМІЧНОГО ПРОГНОЗУВАННЯ

Зайченко Ю.П., Шаповаленко Н.,

Інститут Прикладного Системного Аналізу,
Національний технічний університет України “КПІ”

Метою даної роботи є дослідження й аналіз нечітких нейронних систем, розробка алгоритму навчання нейронної мережі й аналіз ефективності його застосування в задачах прогнозування в макроекономіці.

Апарат нечітких множин і нечіткої логіки вже давно з успіхом застосовується для вирішення задач, у яких вихідні дані є

ненадійними і слабо формалізованими. Сильні сторони такого підходу:

- опис умов і методу рішення задачі мовою, близькою до природної;

- універсальність: відповідно до теореми FAT (Fuzzy Approximation Theorem), доведеної Б. Коско (B. Kosko) у 1993 р., будь-яка математична система може бути апроксимована системою, заснованою на нечіткій логіці;

- ефективність (зв'язана з універсальністю), що підтверджується рядом теорем, аналогічних теоремам про повноту для штучних нейронних мереж, наприклад, теоремою вигляду: для кожної дійсної неперервної функції g , заданої на компактi U , і для довільного $\varepsilon > 0$ існує нечітка експертна система, що формує вихідну функцію $f(x)$ таку, що

$$\sup_{x \in U} \|g(x) - f(x)\| \leq \varepsilon,$$

де $\|\bullet\|$ – символ прийнятої відстані між функціями.

Разом з тим, для нечітких систем характерними є і певні недоліки:

- вихідний набір нечітких правил формулюється експертом-людиною і може виявитися неповним чи суперечливим;

- вигляд і параметри функцій належності, що описують вхідні і вихідні змінні системи, вибираються суб'єктивно, і можуть виявитися такими, що не цілком відбивають реальну дійсність.

Використовуваний у різних експертних і керуючих системах механізм нечітких висновків у своїй основі має базу знань, формовану фахівцями предметної галузі у вигляді сукупності нечітких предикатних правил вигляду:

P_1 : якщо $x \in A_1$ то $y \in B_1$

P_2 : якщо $x \in A_2$ то $y \in B_2$

...

P_n : якщо $x \in A_n$ то $y \in B_n$

де x – вхідна змінна (ім'я для відомих значень даних);

y – змінна висновку (ім'я для значення даних, що буде обчислене);

A та Y – функції належності, визначені відповідно на x та y .

Наведемо більш детальне пояснення. Знання експерта $A \rightarrow B$ відбиває нечітке причинне відношення передумови і висновку, тому його можна назвати нечітким відношенням і позначити через R :

$$R = A \rightarrow B,$$

де “ \rightarrow ” називають нечіткою імплікацією.

Відношення R можна розглядати як нечітку підмножину прямого добутку $X \times Y$ повної множини передумов X та висновків Y . Таким чином, процес одержання (нечіткого) результату висновку B' з використанням даного спостереження A' і знання $A \rightarrow B$ можна представити у вигляді композиційного правила нечіткий “modus ponens”:

$$B' = A' \bullet R = A' \bullet (A \rightarrow B),$$

де “ \bullet ” – операція згортки.

Як операцію композиції, так і операцію імплікації в алгебрі нечітких множин можна реалізовувати по-різному (при цьому буде відрізнятися й одержуваний результат), але в будь-якому випадку загальний логічний висновок здійснюється за наступні чотири етапи.

1) Введення нечіткості (фаззифікація, fuzzification)

Функції належності, визначені на вхідних змінних, застосовуються до їхніх фактичних значень для визначення ступеня істинності кожної передумови кожного правила.

2) Логічний висновок

Обчислене значення істинності для передумов кожного правила застосовується до висновків кожного правила. Це призводить до однієї нечіткої підмножини, що буде призначено кожній змінній висновку для кожного правила. Як правило, до логічного висновку звичайно використовуються тільки операції \min (МІНІМУМ) чи prod (МНОЖЕННЯ). У логічному висновку МІНІМУМУ функція належності висновку “відтинається” по висоті, що відповідає обчисленому ступеню істинності передумови правила (нечітка логіка “И”). У логічному висновку МНОЖЕННЯ функція належності висновку масштабується за допомогою обчисленого ступеня істинності передумови правила.

3) Композиція

Усі нечіткі підмножини, призначені до кожної змінної висновку (у всіх правилах), поєднуються разом, щоб сформувати одну нечітку підмножину для всіх змінних висновку. При подібному об'єднанні звичайно використовуються операції \max (МАКСИМУМ) чи sum (СУМА). При композиції МАКСИМУМУ комбінований висновок нечіткої підмножини конструюється як поточечний максимум по всіх нечітких підмножинах (нечітка логіка "АБО"). При композиції СУМИ комбінований висновок нечіткої підмножини формується як поточечна сума по всіх нечітких підмножинах, які призначені правилами логічного висновку.

4) Приведення до чіткості (дефаззифікація, defuzzification)

Використовується, якщо потрібно перетворити нечіткий набір висновків у чітке число. Існує, значна кількість методів приведення до чіткості, деякі з яких розглянуті нижче.

Алгоритм Mamdani

Даний алгоритм відповідає розглянутому прикладу на рис.1. У розглянутій ситуації він математично може бути описаний у такий спосіб [1]:

1) Введення нечіткості. Знаходяться ступені істинності для передумов кожного правила: $A_1(x_0)$, $A_2(x_0)$, $B_1(x_0)$, $B_2(x_0)$.

2) Логічний висновок. Знаходяться рівні "відсікання" для передумов кожного з правил (з використанням операції МІНІМУМ):

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0);$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0);$$

де через " \wedge " позначена операція логічного мінімуму (min).

Потім знаходяться "усічені" функції належності:

$$C_1' = (\alpha_1 \wedge C_1(z));$$

$$C_2' = (\alpha_2 \wedge C_2(z)).$$

3) Композиція. Проводиться об'єднання знайдених усічених функцій з використанням операції МАКСИМУМ (\max , позначені далі як " \vee "), що призводить до одержання підсумкової нечіткої підмножини для змінної виходу з функцією належності:

$$\mu_{\Sigma}(z) = C(z) = C_1'(z) \vee C_2'(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)).$$

4) Приведення до чіткості. Проводиться для перебування z_0 , наприклад, центроїдним методом.

Алгоритм Tsukamoto

Вихідні посилки – як у попереднього алгоритму, але тут передбачається, що функції $C_1(z)$, $C_2(z)$ є монотонними.

1) Введення нечіткості (як в алгоритмі Mamdani).

2) Нечіткий висновок. Спочатку знаходяться рівні “відсікання” α_1 і α_2 (як в алгоритмі Mamdani), а потім рішеннями рівнянь:

$$\alpha_1 = C_1(z_1) \quad \text{та} \quad \alpha_2 = C_2(z_2)$$

визначаються чіткі значення (z_1 та z_2) для кожного вихідного правила.

3) Визначається чітке значення змінної висновку (як зважене середнє z_1 та z_2):

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2}$$

У загальному випадку (дискретний варіант центроїдного методу):

$$z_0 = \frac{\sum_{i=1}^n \alpha_i z_i}{\sum_{i=1}^n \alpha_i}$$

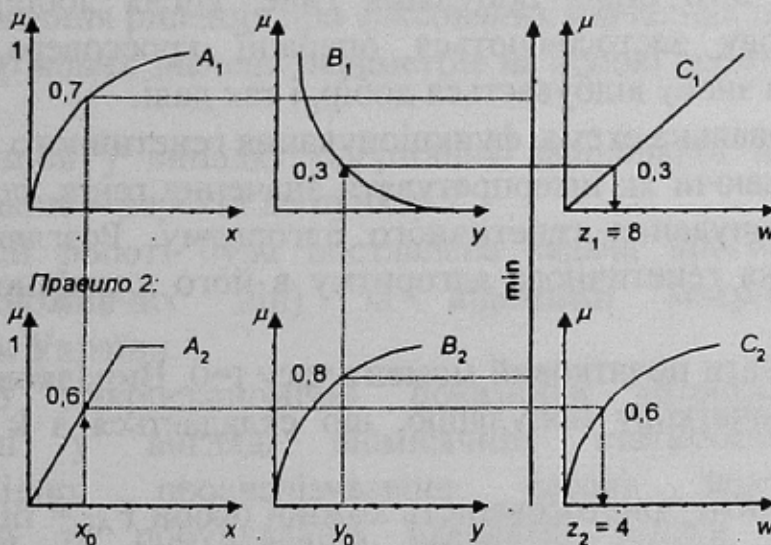


Рисунок 1 - Ілюстрація до алгоритму Tsukamoto

Генетичні алгоритми призначені для рішення задач оптимізації. Прикладом подібної задачі може служити навчання нейромережі, тобто підбора таких значень ваг, при яких досягається мінімальна помилка. При цьому в основі генетичного алгоритму лежить метод випадкового пошуку. Основним недоліком випадкового пошуку є те, що нам невідомо скільки знадобиться часу для рішення задачі. Для того, щоб уникнути таких витрат часу при рішенні задачі, застосовуються методи, що проявилися в біології. При цьому використовуються методи, відкриті при вивченні еволюції і походження видів. Як відомо, у процесі еволюції виживають найбільш пристосовані особи. Це призводить до того, що пристосованість популяції зростає, дозволяючи їй краще виживати в умовах, що змінюються.

Щоб змодельовати еволюційний процес, згенеруємо спочатку випадкову популяцію – декілька індивідуумів з випадковим набором хромосом (числових векторів). Генетичний алгоритм імітує еволюцію цієї популяції як циклічний процес схрещування індивідуумів і зміни поколінь.

Життєвий цикл популяції – це декілька випадкових схрещувань (за допомогою кроссовера, мутацій і інверсії), у результаті яких до популяції додається якась кількість нових індивідуумів. Добір у генетичному алгоритмі – це процес формування нової популяції зі старої, після чого стара популяція гине. Після добору до нової популяції знову застосовуються операції кроссовера, мутації і селекції, потім знову відбувається добір, і так далі.

Обчислювальна схема функціонування генетичного алгоритму

Тепер, знаючи як інтерпретувати значення генів, перейдемо до опису функціонування генетичного алгоритму. Розглянемо схему функціонування генетичного алгоритму в його класичному варіанті [2].

1. Ініціювати початковий момент часу $t=0$. Випадковим образом сформувати початкову популяцію, що складається з k осіб. $B_0 = \{A_1, A_2, \dots, A_k\}$.

2. Обчислити пристосованість кожної особи $F_{A_i} = \text{fit}(A_i)$, $i=1 \dots k$ та популяції в цілому $F_t = \text{fit}(B_t)$ (також іноді називану терміном

фітнес). Значення цієї функції визначає наскільки добре підходить особа, описана даною хромосомою, для рішення задачі.

3. Вибрати особу A_c з популяції. $A_c = \text{Get}(B_t)$.

4. З визначеною імовірністю (імовірністю кросовера P_c) вибрати другу особу з популяції $A_{c1} = \text{Get}(B_t)$ і зробити оператор кросовера $A_c = \text{Crossing}(A_c, A_{c1})$.

5. З визначеною імовірністю (імовірністю мутації P_m) виконати оператор мутації. $A_c = \text{mutation}(A_c)$.

6. З визначеною імовірністю (імовірністю інверсії P_i) виконати оператор інверсії $A_c = \text{inversion}(A_c)$.

7. Помістити отриману хромосому в нову популяцію $\text{insert}(B_{t+1}, A_c)$.

8. Виконати операції, починаючи з пункту 3, k раз.

9. Збільшити номер поточної епохи $t = t+1$.

10. Якщо виконалася умова останову, то завершити роботу, інакше перехід на крок 2.

Навчання нечітких нейронних мереж на основі генетичних алгоритмів

Однією з найбільш поширених областей застосування генетичних алгоритмів є задачі навчання нейронних мереж, у тому числі і нечітких, шляхом підбору адекватних параметрів. Загальними етапами такого навчання є наступні:

1) виділення керуючих параметрів задачі навчання;

2) одержання рішення при фіксованих значеннях параметрів;

3) вибір нових значень параметрів на основі роботи генетичного алгоритму;

4) останов у випадку одержання задовільної неузгодженості рішення, інакше – перехід до пункту 2.

У даній роботі була поставлена задача прогнозування ІСЦ (індекса споживчих цін) за відомими макроекономічними показниками України.

Маємо макроекономічні показники економіки України, представлені у вигляді щомісячних статистичних часових рядів. Необхідно, проаналізувавши часові залежності між показниками та побудувавши адекватні моделі спрогнозувати наступний показник: ІСЦ.

За результатами проведеного кореляційного аналізу було встановлено, що найбільш істотними перемінними, що впливають на рівень ІСЦ є:

М2 – грошовий агрегат;

КВ – кредити вкладені в економіку.

Крім того, було встановлено, що на ІСЦ впливають не тільки ці показники але і їхні часові лаги, що рівні 1 та 2. Розмір вікна склав 11 точок, тобто навчання відбувається на одинадцяти точках (місяцях), а прогнозується дванадцять. Усього було розглянуто 11 точок прогнозування (навчання проходило послідовно на 11 вікнах).

У результаті попередніх досліджень були обрані наступні оптимальні параметри алгоритму навчання:

1) Параметри генетичного алгоритму

- Розмірність популяції – 70.

- Розмірність генів – 16 біт.

- Імовірність мутації – 0,1.

- Імовірність інверсії – 0,1.

- Імовірність кроссовера – 0,6.

- Метод добору пар для кроссовера – турнірний.

- Використання стратегії елітизму – так.

- Збільшення імовірності мутації на 0,8 на 1 епоху, якщо на протязі 100 епох помилка не змінювалася.

2) Параметри нечіткого висновку правил

- Кількість правил – 3.

- Кількість вхідних змінних – 2.

- Вид правил.

П1: якщо x_1 , x_2 то y .

П2: якщо $x_1(-1)$, $x_2(-1)$ то y .

П3: якщо $x_1(-2)$, $x_2(-2)$ то y .

- Метод логічного висновку – Tsukamoto.

- Вид функції належності – трикутна.

- Вибір області значення функції при неоднозначності – 1.

3) Параметри навчання мережі

- Максимальна кількість ітерацій – 10000.

- Граничне значення похибки – 0,001.

- Вимірність навчальної вибірки – 11.

- Тривалість прогнозу – 1.

За допомогою програми були отримані наступні результати.

Таблиця 1. Експериментальні результати прогнозування

Реальні значення	Прогнозні значення	Кв. відхилення
1,4	1,45753	0,00331
1,3	1,215887	0,007075
0,2	0,282463	0,0068
0,2	0,396298	0,038533
1,3	1,458	0,024964
0	0,24653	0,060777
0	0,492283	0,242343
-0,9	-0,788988	0,012324
0,2	0,162647	0,001395
3,8	3,8	0
6,2	6,2	0
	СКВ прогнозу	0,190101

Ми одержали СКВ прогнозу 0,19 що свідчить про високу якість прогнозу і про можливість використання розробленої програми для прогнозування макроекономічних показників.

Список источников:

1. Круглов В.В. Борисов В.В. Искусственные нейронные сети. Теория и практика. М, Горячая линия Телеком, 2001
2. Г.С. Кильдишев, А.А Френкель. Анализ временных рядов и прогнозирование. М, Статистика, 1973.
3. Detlef Nauck, Rudolf Kruse. Interpreting changes in the fuzzy sets of self-adaptive neural fuzzy controller. W3300 Braunschweig, Germany.