

## ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОБОТИ НЕЧІТКИХ НЕЙРОННИХ МЕРЕЖ В ЗАДАЧІ ОЦІНКИ КРЕДИТОСПРОМОЖНОСТІ ПОЗИЧАЛЬНИКА

В статті зроблено огляд методів оцінки кредитоспроможності позичальника на основі нечіткої логіки. Зроблені експериментальні дослідження методів навчання та адаптації нечітких нейронних мереж. Запропоновано модифікований алгоритм адаптації.

The problem of assessing the creditworthiness of the borrower is considered. The application of fuzzy neural networks for this problem solution was suggested. The experimental investigations of the adaptation and learning algorithms of fuzzy neural networks were carried out. The modification of adaptation and learning algorithms of fuzzy neural networks was suggested.

### 1. Вступ

У даній статті розглянуто практичне застосування нечітких нейронних мереж (ННМ) до задачі оцінки кредитоспроможності позичальника. Досліджувалися алгоритми навчання і адаптації вибраних ННМ, особливості їх застосування та ефективність запропонованих методів до поставленої задачі.

### 2. Нечіткі нейронні мережі та їх переваги

Для зниження кредитного ризику банку традиційно виконують оцінку кредитоспроможності позичальника. Інформація, на основі якої приймається рішення про кредитування, може бути неточною, неповною, а дані про позичальника можуть бути такими, що важко формалізуються. Для врахування цих факторів доцільно використовувати нечітку логіку, де процес прийняття рішень базується на зрозумілій базі правил. В статті [1], наприклад, представлена методика оцінки кредитоспроможності фізичних осіб з використанням методу парних порівнянь та нечітких систем з логічним висновком типу Мамдані. Але недоліком систем нечіткого логічного висновку є те, що вони не здатні автоматично навчатися, і параметри та вид функцій належності, які описують фактори кредитоспроможності задаються експертом та можуть бути неадекватними. Нечіткі нейронні мережі поєднують переваги систем нечіткого логічного висновку та нейронних мереж – здатність до адаптації та автоматичного навчання, а також можливість інтерпретації процесу отримання результату.

Для аналізу кредитоспроможності юридичних осіб застосовувались нечітка нейронна мережа з виводом Мамдані [2] та нечітка нейрон-

на мережа з виводом Сугено для фізичних осіб [3].

Першим етапом навчання ННМ є початкове налаштування нечіткої бази правил: вибір не тільки параметрів функцій належності, а й структури мережі – кількості правил. Існує декілька алгоритмів адаптації ННМ:

- проста побудова повної бази правил;
- алгоритм скорочення бази правил;
- алгоритм поступового нарощування бази

правил.

Після визначення структури ННМ відбувається навчання отриманої бази правил.

Одним з найпоширеніших алгоритмів навчання нейронних мереж є градієнтний метод. Для ННМ на кожній ітерації методу уточнюються параметри функцій належності.

Наступний найпоширеніший метод навчання – генетичний алгоритм, де параметри, що настраюються складають особину популяції, до якої застосовуються механізми схрещування, мутації, селекції. В даній статті для поставленої задачі використовувались ННМ з виводом Мамдані та з виводом Сугено, а також рекурентна ННМ з виводом Сугено [4] та каскадна неофаззі мережа [5], яка має високу швидкість навчання.

### 3. Алгоритми навчання та адаптації нечітких нейронних мереж з виводом Мамдані та з виводом Сугено

На основі анкети позичальника необхідно обґрунтовано прийняти рішення – чи варто видавати кредит. Кожну таку анкету можна представити як вектор  $\{X_1, X_2, \dots, X_i, \dots, X_M\}$ , де  $X_i$  – деяким чином формалізовані дані про позичальника та параметри кредиту. Такий вектор являє собою вхідні дані мережі. Виходом

мережі буде рішення, щодо кредитування позичальника.

Перед початком роботи необхідно провести етап адаптації мережі - визначити її структуру, тобто вибрати кількість правил в базі знань, а відповідно визначити кількість нейронів другого та третього шару нечіткої нейронної мережі. Кількості правил вибирається або експертом, або за допомогою алгоритму адаптації. Серед алгоритмів адаптації ННМ найбільш ефективним виглядає алгоритм поступового нарощування правил. Так, в статті [4] пропонується алгоритм адаптації на основі активуючої сили правила. В цьому алгоритмі нове правило додається, якщо виконується умова:

$I = \arg \max_{1 \leq k \leq r} F^k(x) \leq F_{in}$ , де  $F^k(x(t)) = w_k$  - активуюча сила правила  $k$  для вхідного вектора  $t$ , а  $F_{in}$  - заздалегіть обраний поріг активуючої сили правила, який зменшується в процесі навчання. В згенерованому правилі параметри функцій належності встановлюються наступним чином:  $c_j^{(k)} = x_j(t)$ ,  $b_j^{(k)} = 1$ ,

$$\sigma_j^{(k)} = \beta \cdot \prod_{j=1}^N \frac{1}{1 + \left(\frac{x_j - c_j^{(I)}}{\sigma_j^{(I)}}\right)^{2b_j^{(I)}}}$$

мо правило, якщо жодне з вже існуючих правил не описує достатньо добре поточний вхідний вектор.

Алгоритм, запропонований в [7], базується на оцінці точності апроксимації. Він має вигляд:

1. Задаються початкові значення:  $\varepsilon$  - точність апроксимації,  $m = m_0$  - число правил в базі знань;

2. Вибирається наступна точка  $\vec{x}^i$ , для якої значення  $|d^i - y^i|$  максимальне. Якщо база знань пуста, то переходимо до кроку 4. В іншому випадку за допомогою алгоритму нечіткого логічного висновку Сугено та існуючої бази правил рахується значення  $d^i$ .

3. Далі перевіряється умова  $|d^i - y^i| \leq \varepsilon$ , тобто оцінюється точність апроксимації поточної бази правил. Якщо вона не виконується, то переходимо на крок 4, інакше на крок 5.

4. До бази знань додається правило з таким значенням параметрів функції належності (ФН):  $c_j^{(k)} = x_j^i$ ,  $b_j^{(k)} = 1$ ,

$$\sigma_j^{(k)} = \beta \cdot \prod_{j=1}^N \frac{1}{1 + \left(\frac{x_j - c_j^{(o)}}{\sigma_j^{(o)}}\right)^{2b_j^{(o)}}}$$

стає рівною  $m = m + 1$ .

5. Перевіряється умова зупинки алгоритму: чи переглянуті всі навчальні точки. Якщо ні, то вибирається наступна навчальна точка.

В цьому алгоритмі ми додаємо правило, якщо існуюча база знань дає завелику помилку для поточної точки.

Для підвищення ефективності пропонується в обох алгоритмах робити додаткову перевірку умови  $f_o = \min(f_1, f_2, \dots, f_m) > R$ , де

$f_k = \|\vec{c}^{(k)} - \vec{x}^i\|$ ,  $k = 1, m$  - відстань між центрами ФН кожного правила та поточною точкою,  $R = const$  - порогова відстань між правилами та поточною точкою. Таким чином, посилюється контроль за кількістю правил.

В ННМ результат отримується за допомогою апарату нечіткої логіки, але відповідні функції належності настроюються з використанням алгоритмів навчання ННМ, наприклад, алгоритму зворотного поширення помилки. Для функції помилок виконується крок градієнтним методом. Таким чином, мережа використовує апріорну інформацію, отримує нові знання та залишається логічно прозорою для користувача.

Так для ННМ з виводом Мамдані критерій оптимізації (функція помилок) має вигляд:

$$E(z) = \frac{1}{2}(z^* - z_0)^2 \rightarrow \min, \text{ де } z^* - \text{бажаний}$$

вихід мережі;  $z_0$  - фактичний вихід. Алгоритм навчання має вигляд:

$$\left. \begin{aligned} a_k(n+1) &= a_k(n) - \gamma_n^1 \frac{\partial E(n)}{\partial a_k} \\ \sigma_k(n+1) &= \sigma_k(n) - \gamma_n^2 \frac{\partial E(n)}{\partial \sigma_k} \\ a_{ik}(n+1) &= a_{ik}(n) - \gamma_n^3 \frac{\partial E(n)}{\partial a_{ik}} \\ \sigma_{ik}(n+1) &= \sigma_{ik}(n) - \gamma_n^4 \frac{\partial E(n)}{\partial \sigma_{ik}} \end{aligned} \right\},$$

де -  $\gamma_n^1, \gamma_n^2, \gamma_n^3, \gamma_n^4$  - розмір кроку.

В ННМ з виводом Сугено – ННМ TSK (Takagi, Sugeno, Kang'a) використовується така база правил:

$\Pi_1$  : якщо  $x_1 \in A_1^{(1)}; x_2 \in A_2^{(1)}, \dots, x_n \in A_n^{(1)}$ ,  
 то  $y_1 = p_{10} + \sum_{j=1}^N p_{1j}x_j$ ,

$\Pi_M$  : якщо  $x_1 \in A_1^{(M)}; x_2 \in A_2^{(M)}, \dots, x_n \in A_n^{(M)}$ , то  $y_M = p_{M0} + \sum_{j=1}^N p_{Mj}x_j$ ,

де  $A_i^{(k)}$  - значення лінгвістичної змінної  $x_i$  для правила  $\Pi_k$  з функцією належності

$$\mu_{A_i}^{(k)}(x_i) = \frac{1}{1 + \left( \frac{x_i - c_i^{(k)}}{\sigma_i^{(k)}} \right)^{2b_i^{(k)}}}$$

В рекурентній ННМ TSK вихід кожного правила подається на вхід мережі, таким чином враховується минулий вплив правила. Вихід кожного правила має вигляд:

$$y_k(x) = p_{k0} + \sum_{j=1}^N p_{kj}x_j + p_{kN+1}h_k$$

де  $h_k$  - вихід правила на минулому кроці.

Можна використовувати звичайний градієнтний алгоритм та налаштувати параметри  $c_j^{(k)}$ ,  $\sigma_j^{(k)}$ ,  $b_j^{(k)}$ ,  $p_{k0}$ ,  $p_{kj}$ , та  $h_k$ . Критерій оптимізації

має вигляд:  $E = \frac{1}{L} \sum_{l=1}^L (y_l - d_l)^2$ , де L – об’єм навчальної вибірки. Але найчастіше використовують гібридний алгоритм навчання, в якому параметри  $p_{k0}$ ,  $p_{kj}$  знаходяться в результаті рішення системи з L лінійних рівнянь при фіксованих інших параметрах. Детальний опис методів можна знайти в [6].

Також для знаходження параметрів функції належності використовують генетичний алгоритм. В генетичному алгоритмі навчання всі змінні, що підлягають налаштуванню включають до особини популяції:

$$W = [c_1^{(1)}, \dots, c_N^{(1)}, b_1^{(1)}, \dots, b_N^{(1)}, \sigma_1^{(1)}, \dots, \sigma_N^{(1)}, p_{10}, p_{11}, \dots, p_{1N}, c_1^{(M)}, \dots, c_N^{(M)}, b_1^{(M)}, \dots, b_N^{(M)}, \sigma_1^{(M)}, \dots, \sigma_N^{(M)}, p_{M0}, p_{M1}, \dots, p_{MN}]$$

Спершу задається початкова популяція з L особин. Один крок алгоритму містить такі етапи:

1. Знаходиться індекс придатності

$$FI(W_i) = C - \frac{1}{L} \sum_{l=1}^L (y_l - d_l)^2$$

де C – константа.

2. Вибираються особини для схрещування на основі ймовірнісного механізму. Ймовірність вибору i-го батька  $P_i = \frac{FI(W_i)}{\sum_{l=1}^L FI(W_l)}$ .

3. Далі застосовується алгоритм схрещування.

4. Отримана популяція підлягає мутації.

5. Формується нова популяція.

Оскільки в генетичному алгоритмі для різних запусків популяція може сходиться до різних рішень, доцільно використовувати острівну модель алгоритму. В цій моделі основна популяція ділиться на декілька популяцій, де для кожної виконується генетичний алгоритм, це дає можливість поєднати «досягнення» кожної підпопуляції.

Кожний з розглянутих алгоритмів має свої недоліки. Так, градієнтний алгоритм сильно залежить від початкових умов, а генетичний часто сходиться до локальних оптимумів. Автором пропонується використовувати гібридний алгоритм, в якому початкове наближення (початкові значення параметрів функцій належності) близьке до оптимального знаходиться за допомогою генетичного алгоритму, а вже далі воно розглядається як початкова точка для градієнтного алгоритму.

Для порівняння з вищеозначеними ННМ використовувалася каскадна неофаззі мережа. В цій мережі в якості вузлів використовується неофаззі нейрон, який реалізує

$$\hat{y} = \sum_{i=1}^n f_i(x_i) = \sum_{i=1}^n \sum_{j=1}^h w_{ji} \mu_{ji}(x_i)$$

де  $\mu_{ji}$  - трикутна функція належності вхідної змінної,  $w_{ji}$  - синаптична вага. Детальний опис алгоритму навчання можна знайти в [5].

#### 4. Результати досліджень

Для аналізу кредитоспроможності за допомогою запропонованих методів використовувалась вибірка даних одного з українських банків, яка складається з 998 заявок за 2010 рік. Для оцінки запропонованих методів використовувались такі показники: середньо квадратична похибка (СКП), відсоток вірних класифікацій та час роботи алгоритму. Результати досліджень, щодо розглянутих алгоритмів адаптації та їх модифіковані варіанти представлені в таблиці 2. Для дослідження алгоритмів адаптації використовувалась вибірка з 300 заявок.

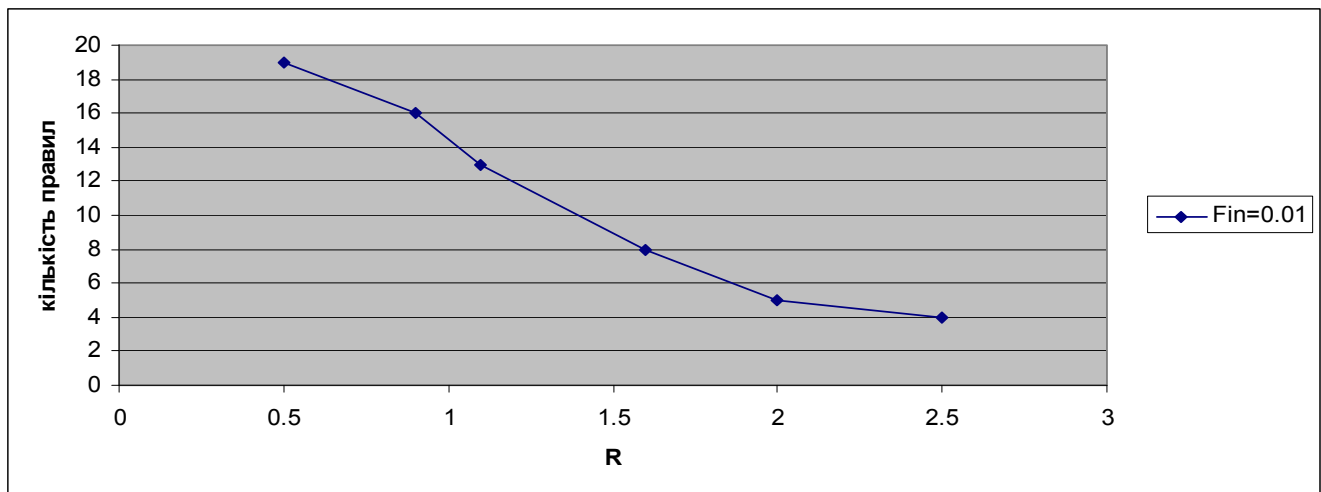
**Табл. 1. Результати оцінки кредитоспроможності позичальника для різних алгоритмів адаптації**

Метод	Кількість правил	СКП	Відсоток вірних класифікацій, %	Час роботи алгоритму, с
ННС TSK де кількість правил встановлюється експертом	8	0.1139	89	4021
ННС TSK з алгоритмом, заснованим на активуючій силі правила	6	0.1097	89.4	2281
ННС TSK з модифікованим алгоритмом, заснованим на активуючій силі правила	5	0.117	88.6	1649
ННС TSK з алгоритмом, заснованим на оцінці точності апроксимації	7	0.1103	88	3539
ННС TSK з модифікованим алгоритмом, заснованим на оцінці точності апроксимації	4	0.1139	87	1053

Як бачимо, кількість правил при модифікованих алгоритмах адаптації менше ніж при звичайних, при цьому відсоток вірних класифікацій суттєво не змінюється, а час роботи алгоритму зменшується. Експертом може бути задано надлишкове число правил в базі знань, тоді в процесі навчання, деякі правила будуть повторюватися. Але структура мережі буде відповідати запропонованим числу правил - буде надмірно складною і час навчання буде більшим. В модифікованих алгоритмах менша кількість правил генерується завдяки додатковій перевірці відстані між центрами ФН кожного правила та поточною точкою. Тобто нове правило додається якщо поточна

точка погано описується існуючою базою правил та відстань між центрами ФН кожного правила та поточною точкою достатньо велика (зادля того щоб не було однакових правил). Тож ефективніше застосовувати модифікований алгоритм.

Варто зауважити, що кількість згенерованих правил залежить від параметрів  $F_{in}$ ,  $R$  для модифікованого алгоритму адаптації, заснованому на активуючій силі правил, та від  $\varepsilon$ ,  $R$  для модифікованого алгоритму заснованому на оцінці точності апроксимації. Результати досліджень для алгоритму заснованому на активуючій силі правила представлені на графіках 1-2.



**Графік 1. Залежність згенерованої кількості правил від параметру  $R$**

Як бачимо на графіку 1, при фіксованому значенні параметру  $F_{in}$ , більша кількість правил генерується для меншого значення  $R$ .

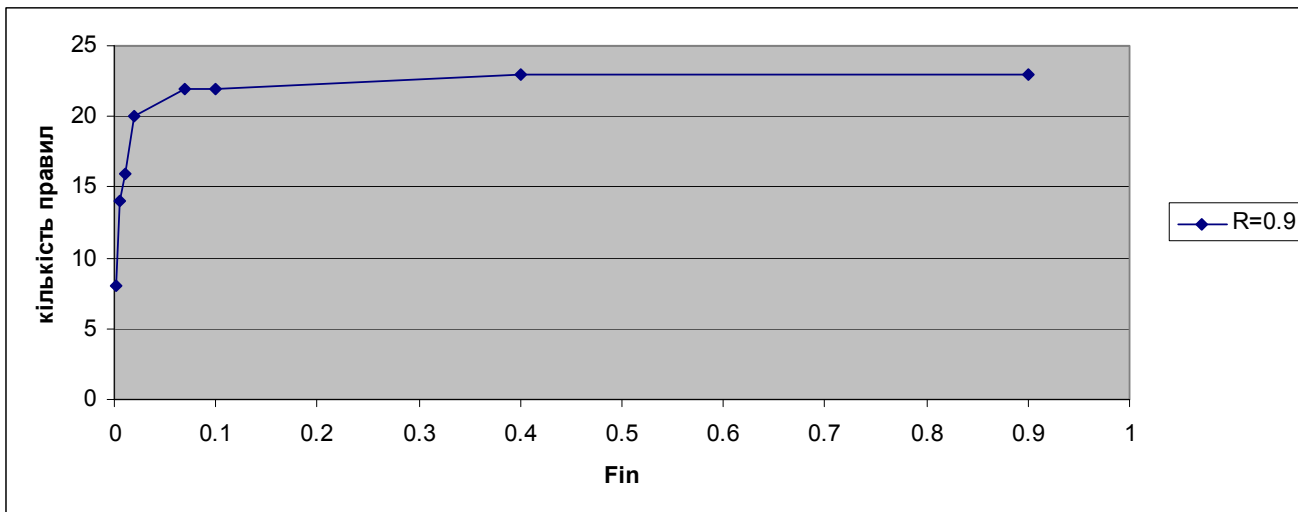
При фіксованому значенні  $R$ , кількість правил збільшується при збільшенні  $F_{in}$  - чим вище активуюча сила правила, тим краще воно описує вхідний вектор. Але збільшення правил відбувається лише до деякого

граничного значення  $F_{in}$ , після якого вже не має сенсу додавати правила, так як побудована база вже описує всі можливі вхідні вектори з навчальної вибірки. Результати досліджень для алгоритму, заснованому на оцінці точності апроксимації, представлені на графіках 3-4.

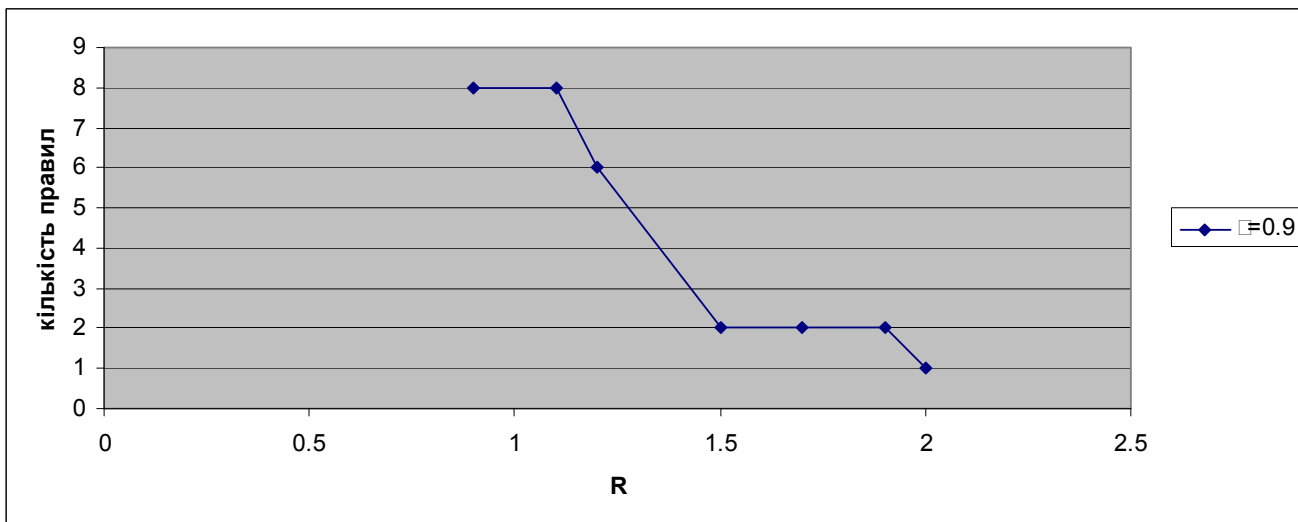
На графіку 3 бачимо, що при фіксованому значенні  $\varepsilon$  - кількість правил збільшується

при зменшенні параметра  $R$ . В обох модифікованих алгоритмах адаптації параметр  $R$  відповідає за різноманітність правил – чим

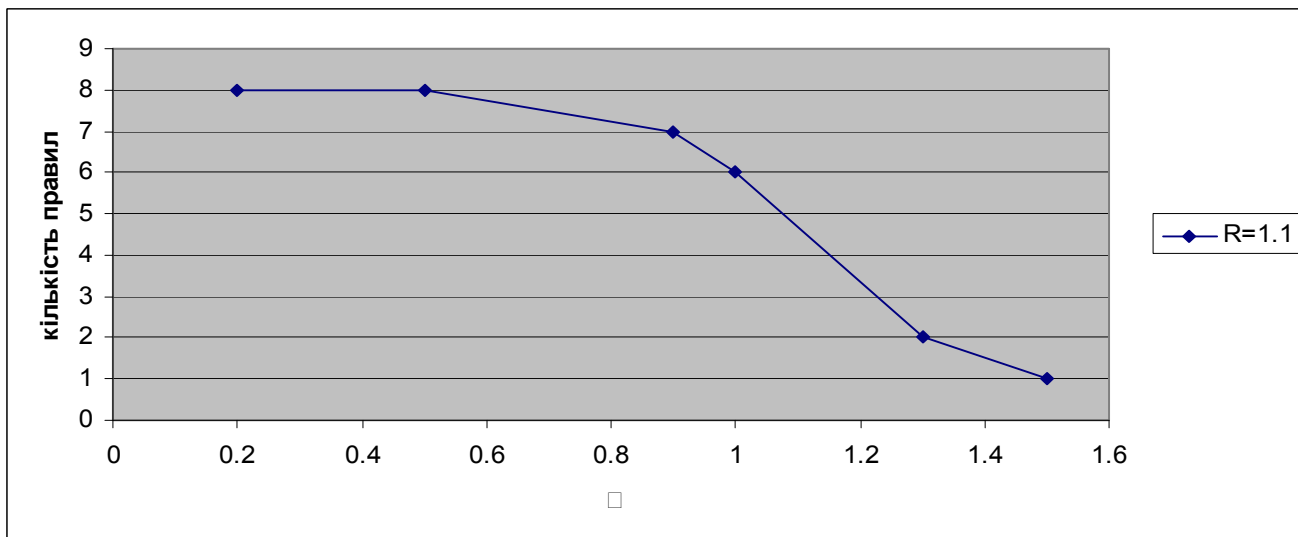
більше значення  $R$ , тим більша відстань між згенерованими правилами.



Графік 2. Залежність згенерованої кількості правил від параметру  $Fin$



Графік 3. Залежність згенерованої кількості правил від параметру  $R$



Графік 4. Залежність згенерованої кількості правил від параметру  $\epsilon$

З графіку 4 видно, що при фіксованому значенні  $R$  - кількість правил збільшується при зменшенні параметру  $\varepsilon$ . Тобто, чим більшу точність ми вимагаємо від бази правил, тим більше правил генерується.

Результати експериментів для різних нечітких нейронних мереж представлені в таблиці 1. Число правил вибрано 9.

**Табл. 2. Результати оцінки кредитоспроможності позичальника для різних ННМ**

Метод	СКП	Відсоток вірних класифікацій, %	Час роботи алгоритму, с
ННМ TSK з генетичним алгоритмом навчання	0.129	86.3	1161
ННМ TSK з градієнтним алгоритмом навчання	0.116	87.0	2024
ННМ TSK з гібридним алгоритмом навчання	0.115	87.7	3129
РННМ TSK з гібридним алгоритмом навчання	0.121	86.0	4214
Неофаззі каскадна мережа	0.102	85.3	187
ННМ з виводом Мамдани з генетичним алгоритмом навчання	0.187	85.0	1093
ННМ з виводом Мамдани з градієнтним алгоритмом навчання	0.112	85.3	2386

Як бачимо, найбільший відсоток вірних класифікацій у ННМ TSK з гібридним алгоритмом навчання. Час роботи алгоритму найменший у каскадної неофаззі мережі, але вибір параметрів трикутної ФН, що використовується в алгоритмі, лежить на експертах.

## 5. Висновки

Однією з переваг ННМ являється їх здатність до навчання, тобто відпадає необхідність залучення експертів до налаштування бази правил для поточної ситуації.

Проведено аналіз ефективності застосування нечітких нейронних мереж до задачі оцінки кредитоспроможності позичальника та прове-

дено аналіз ефективності методів їх навчання. Отримані результати показали ефективність використання при навчанні ННМ гібридного алгоритму (комбінації градієнтного та генетичного методів). Кращі результати за відсотком вірних класифікацій показала ННМ TSK з гібридним алгоритмом навчання. Запропоновано та проаналізовано новий алгоритм адаптації нечіткої нейронної мережі, в якому база правил настроюється автоматично. Як наслідок, відбувається побудова оптимальної структури мережі, для навчання якої необхідно менше часу. Реалізація запропонованих алгоритмів дає можливість побудови бази правил без допомоги експертів.

## Перелік посилань

1. Кузнецов Л.А. Оценка кредитной истории физических лиц на основе нечетких моделей / Л.А. Кузнецов, А.В. Перевозчиков // Управление большими системами. - 2007. - № 21 – с. 84-106
2. Зайченко Ю.П. Оценка кредитных банковских рисков с использованием нечеткой логики // Intelligent Information and Engineering Systems. – 2008. - №13 – с. 190-20
3. Шовгун Н.В. Аналіз кредитоспроможності позичальника за допомогою методів з нечіткою логікою // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – 2012. – № 55. – с. 169-174
4. CF Juang and CT Lin A recurrent self-organizing neural fuzzy inference network // IEEE transaction of neural networks, Volume 10, Number 4, Pages 828–845, July 1999
5. Бодянский Е.В. Каскадная эволюционная нейронная сеть с неофаззи нейронами в качестве узлов // Восточно-Европейский журнал передовых технологий – 2011. – Вып. 4/3 (52). – с. 55–58.
6. Зайченко Ю.П. Нечеткие модели и методы в интеллектуальных системах. – К.: Издательский дом «Слово», 2008. – 334 с.
7. Круглов В.В., Борисов В. В. Искусственные нейронные сети. Теория и практика. -2-е изд., стереотип.- М.: Горячая линия-Телеком, 2002. -382