

УДК 681.325

А. С. Васюра, к. т. н, проф.; Т. Б. Мартинюк, к. т. н, доц.; Л. М. Куперштейн**МОДЕЛЮВАННЯ НЕЙРОМЕРЕЖІ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ
ІДЕНТИФІКАЦІЇ СИМВОЛІВ**

Проведено моделювання тришарової нейронної мережі прямого розповсюдження для задачі ідентифікації символів у контексті використання її в нейронній частині системи виявлення шахрайства у телекомунікаційних мережах. Моделювання виконано у середовищі MatLAB за допомогою інструментарію Neural Networks Toolbox.

Ключові слова: система виявлення шахрайства, нейронна мережа, ідентифікація символів.

Вступ

Нейронні мережі (НМ) використовуються для розв'язання різноманітних задач у багатьох галузях людської діяльності, серед яких: економіка і бізнес, медицина, авіація, зв'язок, безпека й охоронні системи, автоматизація виробництва тощо [1 – 3].

Розв'язання більшості вищезазначених задач базується на класифікації та розпізнаванні образів, прикладом яких може бути задача розпізнавання букв, розпізнавання мови, класифікація сигналу електрокардіограми, визначення кредитоспроможності клієнта банку, медичні задачі, задача керування портфелем цінних паперів, задача визначення життєздатних і схильних до банкрутства фірм тощо [4, 5].

Ще одним практичним і перспективним напрямком використання НМ як універсального класифікатора є виявлення різного роду шахрайства у телекомунікаційних системах. Це пояснюється тим, що оператори зв'язку в усьому світі саме з цієї причини втрачають до 100 мільярдів. доларів, і з кожним роком ця сума зростає [6]. Основним напрямком боротьби з такими втратами є системи виявлення шахрайства (СВШ), які є складовою систем керування та моніторингу телекомунікаційної мережі. СВШ базуються на використанні інтелектуальних технологій: нейронних мереж і нечіткої логіки [7].

Постановка задачі

Ідентифікація динамічної системи (процесу) – це одержання або уточнення за експериментальними даними математичної моделі цієї системи (процесу), яка описана за допомогою математичного апарата [8]. Якщо розглядається об'єкт у всіх його експлуатаційних режимах, то одержання або уточнення за експериментальними даними моделі цього об'єкта, дієвої для всіх експлуатаційних режимів, називається ідентифікацією об'єкта.

Задача ідентифікації, як правило, розв'язується в два етапи. На першому етапі, що називається структурною ідентифікацією, формується груба модель об'єкта, яка апроксимує взаємозв'язок вхід-вихід і містить параметри, що налаштовуються. На другому етапі, що називається параметричною ідентифікацією, підбираються такі значення параметрів, які мінімізують відстані між модельними й експериментальними виходами об'єкта [9, 10].

Крім того задачі ідентифікації можуть розглядатися з тієї точки зору, що й задачі розпізнавання (класифікації) образів [11], що полягають у віднесенні зразка до однієї з декількох множин, які попарно не перетинаються. НМ є найефективнішим способом класифікації через те, що є універсальним апроксиматором, а також вони можуть генерувати велику кількість регресійних моделей, які використовуються у розв'язанні задач класифікації статистичними методами [4, 12].

У свою чергу, головне завдання, яке ставиться перед СВШ – ідентифікація (розпізнавання) дій шахрая за нестандартною поведінкою профілю абонента, який можна охарактеризувати за такими ознаками, як частота та напрям викликів, тривалість розмови, ідентифікаційні дані абонента тощо. Таким чином, задачею СВШ є розпізнавання образу за

вхідним набором сигналів, які знімаються безпосередньо з комутатора сканером службової інформації та відомостей про здійснені дзвінки [7, 13]. Вся ця інформація надходить у цифровому вигляді для наступної її ідентифікації у СВШ нейро-нечіткою мережею, де у нейронній частині виконується попередня обробка сигнальної інформації [14], а у нечіткій – інтерпретація та логічне виведення отриманих результатів [6, 7]. Крім того початковим етапом проектування такої СВШ є побудова її імітаційної моделі.

Отже, **метою** цієї роботи є побудова і дослідження моделі НМ нейронної частини нейро-нечіткої системи виявлення шахрайства у телекомунікаційних мережах на прикладі ідентифікації символів.

Особливості ідентифікації символів

Серед великого класу програмного забезпечення для моделювання НМ використано інструментарій Neural Networks Toolbox пакету прикладних програм MATLAB 6.5 фірми MathWorks, оскільки він досить гнучкий, забезпечений широким набором команд і функцій для проектування та дослідження як статичних, так і динамічних НМ [15]. Крім того цей пакет забезпечений GUI-інтерфейсом, що дозволяє виконати дослідження НМ навіть недосвідченому користувачу.

Розглянемо варіант реалізації моделі НМ за допомогою MATLAB для розпізнавання 10 арабських цифр.

У якості датчика передбачається використовувати систему розпізнавання, що виконує відцифрування кожного символу. Наприклад, кожний символ буде представлений шаблоном розміром 3x5. Тоді символ “4” може бути представлений, як показано на рис. 1,а і 1,б.

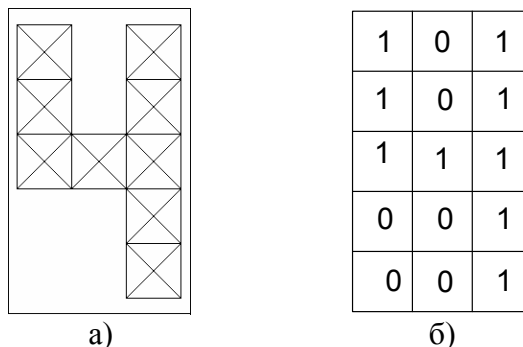


Рис. 1. Представлення символу “4”

Однак система зчитування символів зазвичай працює неідеально – і окремі елементи символів можуть виявитися перевернутими (рис. 2).

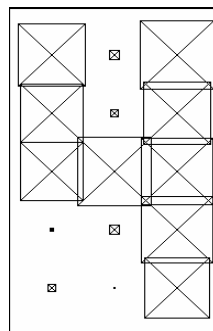


Рис. 2. Представлення зашумленого символу “4”

НМ, що моделюється, повинна точно розпізнавати ідеальні вектори входу і з максимальною точністю відтворювати зашумлені вектори. М-функція `prgrov_n` визначає 10 векторів входу, кожний з яких містить 15 елементів, цей масив називається *цифрами*. М-функція

формує вихідні змінні *numpad* і *targets*, що визначають масиви цифр і цільових векторів.

Отже, на вхід мережі надходить вектор входу з 15 елементами; вектор виходу містить 10 елементів, тільки один з яких дорівнює 1, а інші – 0. Правильно функціонуюча мережа повинна відповісти вектором зі значенням 1 для елемента, що відповідає номеру цифри від 0 до 9. Крім того мережа повинна бути здатною розпізнавати символи в умовах дії шуму. Передбачається, що шум – це випадкова величина із середнім значенням 0 і стандартним відхиленням, меншим чи рівним 0.2 [15].

Для роботи НМ необхідно 15 входів і 10 нейронів у вихідному шарі. Для розв’язання задачі задіємо тришарову НМ прямого розповсюдження з пороговою, логарифмічною сигмоїдною і лінійною функціями активації у вхідному, прихованому і вихідному шарах відповідно. Такі функції активації обрані через те, що вхідний і вихідний шари переважно здійснюють підготовку даних для подальшого використання, а також перетворення отриманих результатів, і для цього цілком придатні порогова і лінійна функції [16]. Що стосується прихованого шару, то він виконує всі основні обчислення, і його функція активації повинна бути більш “природною”, наприклад, логістичною сигмоїдною чи гіперболічним тангенсом.

Імітаційна модель нейронної мережі

Модель структури НМ для розпізнавання арабських цифр наведена на рис. 3, де $IW\{1,1\}$, $LW\{2,1\}$, $LW\{3,2\}$ – вагові матриці першого, прихованого і вихідного шарів відповідно; $b\{1\}$, $b\{2\}$, $b\{3\}$ – вектори зміщення першого, прихованого і вихідного шарів відповідно; $p1$ – вхідний інформаційний векторний масив; y – вихідний векторний масив; $a1$, $a2$, $a3$ – масиви значень функцій активації першого, прихованого і вихідного шарів відповідно.

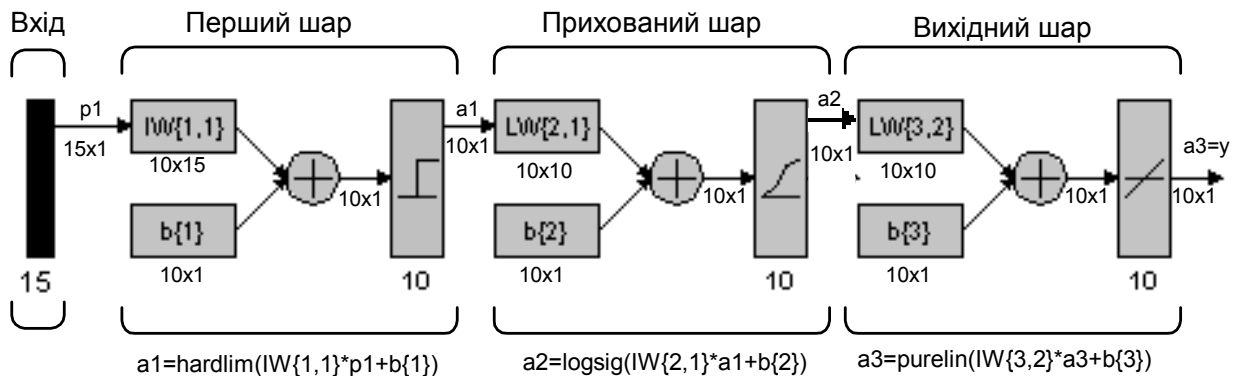


Рис. 3. Структурна схема нейронної мережі

Прихований шар має 10 нейронів, проте, якщо при навчанні мережі виникають ускладнення, можна збільшити кількість нейронів цього рівня. Мережа навчається так, щоб сформувати одиницю в єдиному елементі вектора виходу, позиція якого відповідає номеру символу, і заповнити іншу частину вектора нулями. Однак наявність шумів може призводити до того, що НМ не буде формувати вектори виходу, що складаються точно з одиниць і нулів. Тому по завершенні етапу навчання вихідний сигнал обробляється М-функцією *comet*, що присвоює значення 1 єдиному елементу вектора виходу, а всім іншим – значення 0.

Ініціалізація мережі. Викликаємо М-файл *prprob_n*, що формує масив векторів входу *numpad* розміру 15x10 із шаблонами символів цифр від 0 до 9 і масив цільових векторів *targets*:

```
[numpad,targets]=prprob_n;
[R,Q]=size(numpad);
[S3,Q]=size(targets);
```

Тришарову нейронну мережу прямого розповсюдження створюємо за допомогою команди *newff*:

```
S1=10;
S2=10;
net=newff(minmax(numpad),[S1 S2 S3],{'hardlim'... 'logsig' 'purelin'},'traingdx');
net.LW{3,1}=net.LW{3,1}*0.01;%ініціалізація ваг
net.b{3}=net.b{3}*0.01;%ініціалізація зміщення
```

Навчання. Щоб НМ могла обробляти зашумлені вектори входу, виконаємо її навчання як на ідеальних, так і на зашумлених векторах. Спочатку мережа навчається на ідеальних векторах, поки не буде забезпечена мінімальна середня квадратична похибка. Потім мережа навчається на 10 наборах ідеальних і зашумлених векторів. Навчання виконується за допомогою функції `traingdx`, що реалізує метод зворотного поширення помилки зі збуренням і адаптацією параметра швидкості настроювання.

Навчання при відсутності шуму. Мережа спочатку навчається при відсутності шуму з максимальною кількістю циклів навчання 5000 або до досягнення припустимої середньої квадратичної похибки, що дорівнює 0.11 (рис. 4):

```
P=numpad;
T=targets;
net.performFcn='mse';
net.trainParam.goal=0.11;
net.trainParam.show=20;
net.trainParam.epochs=5000;
net.trainParam.me=0.95;
[net,tr]=train(net,P,T)
```

Як видно з рис. 4, мережа навчилася вже після виконання 84 епох (циклів).

Навчання при наявності шуму. Щоб спроектувати НМ, нечутливу до впливу шуму, навчимо її із застосуванням двох ідеальних і двох зашумлених копій векторів набору цифр від 0 до 9. Цільові вектори складаються з чотирьох копій векторів.

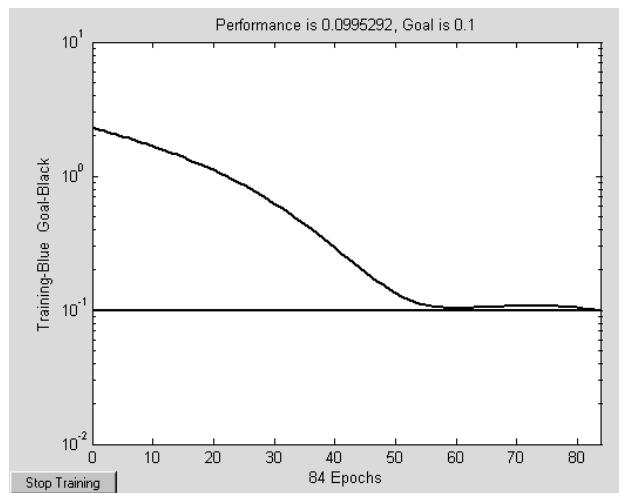


Рис. 4. Навчання мережі при відсутності шуму

Зашумлені вектори мають шум із середнім значенням 0.1 і 0.2. Це навчає НМ правильно розпізнавати зашумлені символи та ідеальні вектори. При навчанні з шумом максимальну кількість циклів навчання скоротимо до 300, а припустиму похибку збільшимо до 0.6 (рис. 5):

```
netn=net;
netn.trainParam.goal=0.6;
netn.trainParam.epochs=300;
T = [targets targets targets targets];
for pass=1:10
    P=[numpad,numpad,(numpad+randn(R,Q)*0.1),
```

```
(numpad+randn(R,Q)*0.2)];
[netn,tr]=train(netn,P,T);
end
```

Оскільки НМ навчалася за наявності шуму, то має сенс повторити її навчання без шуму, щоб гарантувати, що ідеальні вектори входу класифікуються правильно:

```
netn.trainParam.goal=0.1;%Гранична похибка
netn.trainParam.epochs=500;%Максимальна кількість циклів навчання
net.trainParam.show=5;%Частота виведення результатів на екран
[netn,tr]=train(netn,P,T)
```

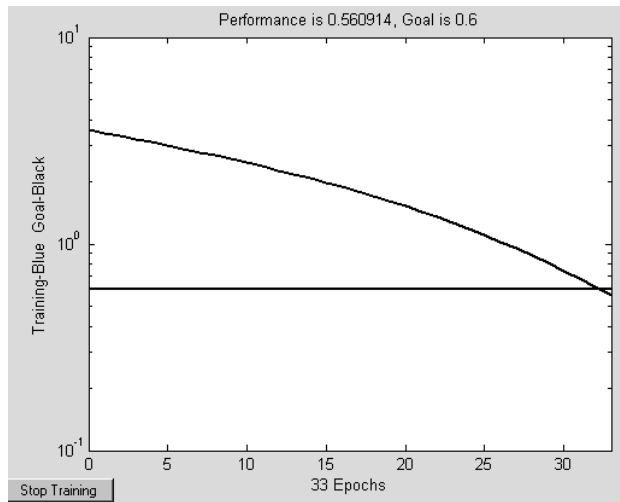


Рис. 5. Навчання мережі при наявності шуму

Якщо необхідна більш висока точність розпізнавання, мережа може бути навчена або протягом більш тривалого часу, або з використанням більшої кількості нейронів у прихованому шарі. Можна також збільшити розмір векторів, щоб скористатися шаблоном з більш дрібною сіткою, наприклад 5x7 чи 10x14 точок замість 3x5.

Перевіримо роботу НМ для розпізнавання символів. Для графічного виведення символу на екран використовується М-функція plotchar_n. Сформуємо зашумлений вектор входу для цифри 2 (рис. 6):

```
noisy_num=numpad(:,3)+randn(15,1)*0.2;
plotchar_n(noisy_num);%Зашумлена цифра 2 (рис.6)
A2=sim(net,noisy_num);
A2=compet(A2);
answer=find(compet(A2)==1)
answer = 3
plotchar_n(numpad(:,answer));%розпізнана цифра 2
НМ відновила символ “2” без помилок (рис. 7).
```

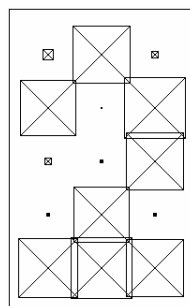


Рис. 6. Зашумлений символ

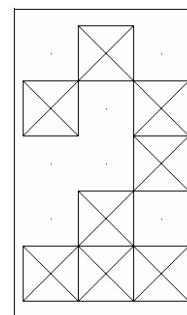


Рис. 7. Розпізнаний символ

Висновки

Отже, наведена модель НМ продемонструвала, як може функціонувати система розпізнавання образів. Мережа була навчена кілька разів при різних векторах входу. Навчання НМ на різних наборах зашумлених векторів дозволило навчити її працювати з зашумленими даними, що характерно для реальної практики. Крім того у вхідному шарі НМ використано порогові нейрони, що доводить можливість і доцільність апаратної реалізації нейроструктур з використанням простих перцептронних нейронів для проектування інтелектуальних СВШ у телекомунікаційних мережах.

СПИСОК ЛІТЕРАТУРИ

1. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. Пер. с англ. – М.: Мир, 1992. – 240 с.
2. Комашинский В.И., Смирнов Д.А. Нейронные сети в системах управления и связи. – М.: Горячая линия – Телеком, 2003. – 94 с.
3. Галушкин А.И., Иванов В.В. Некоторые концептуальные вопросы развития нейрокомпьютеров // Зарубежная радиоэлектроника. – 1997. – № 2. – С. 3 – 10.
4. Стариков А. Практическое применение нейронных сетей для задач классификации (кластеризации) // BaseGroup. – 2000 (<http://www.basegroup.ru/neural/prectice.htm>).
5. Ежов А.А., Шумский С.А. Нейрокомпьютинг и его применения в экономике и бизнесе. – М.: МИФИ, 1998. – 224 с.
6. Лезин В., Перминов К. Мошенничество в сетях операторов фиксированной связи и как с ним бороться // Информкурьерсвязь. – 2004. – № 3 (http://www.miks.ru/magazine/magazine_look.php?id=192).
7. Русеев Д. Мошенничество в мобильных сетях и средства борьбы с ним // Мобильные телекоммуникации. – 2003. – № 3 (<http://www.mobilecomm.ru/view.php?id=423>).
8. Красовский А.А. Универсальные алгоритмы оптимального управления непрерывными процессами. – М.: Наука, 1977. – 272 с.
9. Цыпкин Я.З. Основы информационной теории идентификации. – М.: Наука, 1984. – 225 с.
10. Ротштейн А.П. Интеллектуальные технологии идентификации: нечеткие множества, генетические алгоритмы, нейронные сети. – Винница: «УНІВЕРСУМ-Вінниця», 1999. – 320 с.
11. Цыпкин Я.З. Адаптация и обучение в автоматических системах. – М.: Наука, 1968. – 400 с.
12. Неймарк Ю.И. Многомерная геометрия и распознавание образов // Математика. – 1996. – №7. – С. 119 – 123.
13. Куперштейн Л.М. До проблеми виявлення шахрайства у телекомунікаційних мережах на базі інтелектуальних технологій // Матеріали III МНТК "Сучасні проблеми радіоелектроніки, телекомунікацій та приладобудування". – Вінниця. – 2007. – С. 25 – 26.
14. Борисов Е.С. Использование языка МС# для реализаций параллельного классификатора образов // Кибернетика и системный анализ. – 2005. - № 3. – С. 179 – 182.
15. Медведев В.С., Потемкин В.Г. Нейронные сети. MATLAB 6 / Под. общ. редакцией В.Г. Потемкина. – ДИАЛОГ-МИФИ, 2002. – 496 с.
16. Пономарев С. Нейронные сети // INFUSED BYTES OnLine (<http://www.enlight.ru/ib/tech/neural/index.html>).

Васюра Анатолій Степанович – директор інституту, професор кафедри автоматичної та інформаційно-вимірювальної техніки;

Мартинюк Тетяна Борисівна – доцент кафедри лазерної та оптоелектронної техніки;
Інститут автоматичної, електроніки та комп'ютерних систем управління, Вінницький національний технічний університет

Куперштейн Леонід Михайлович – старший викладач кафедри інформаційних систем і технологій

Вінницький фінансово-економічний університет