

УДК 681.325

**А. С. Васюра, к. т. н, проф.; Т. Б. Мартынюк, к. т. н, доц.; Л. М. Куперштейн****МОДЕЛИРОВАНИЕ НЕЙРОСЕТИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ  
ИДЕНТИФИКАЦИИ СИМВОЛОВ**

*Проведено моделирование трёхслойной нейронной сети прямого распространения для задачи идентификации символов в контексте использования ее в нейронной части системы выявления мошенничества в телекоммуникационных сетях. Моделирование выполнено в среде MatLAB с помощью инструментария Neural Networks Toolbox.*

**Ключевые слова:** система выявления мошенничества, нейронная сеть, идентификация символов.

**Введение**

Нейронные сети (НС) используются для решения разнообразных задач во многих областях деятельности человека, среди которых: экономика и бизнес, медицина, авиация, связь, безопасность и охраняемые системы, автоматизация производства и т. п. [1 – 3].

Решение большинства вышеупомянутых задач базируется на классификации и распознавании образов, примером которых может быть задача распознавания букв, распознавание языка, классификация сигнала электрокардиограммы, определение кредитоспособности клиента банка, медицинские задачи, задача управления портфелем ценных бумаг, задача определения жизнеспособных и предрасположенных к банкротству фирм и т. п. [4, 5].

Еще одним практическим и перспективным направлением использования НС как универсального классификатора является выявление разного рода мошенничества в телекоммуникационных системах. Это объясняется тем, что операторы связи во всем мире именно по этой причине теряют до 100 миллиардов долларов, и с каждым годом эта сумма возрастает [6]. Основным направлением борьбы с такими потерями являются системы выявления мошенничества (СВМ), которые являются составной системой управления и мониторинга телекоммуникационной сети. СВМ базируются на использовании интеллектуальных технологий: нейронных сетей и нечеткой логики [7].

**Постановка задачи**

Идентификация динамической системы (процесса) – это получение или уточнение по экспериментальным данным математической модели этой системы (процесса), которая описана с помощью математического аппарата [8]. Если рассматривается объект во всех его эксплуатационных режимах, то получение или уточнение по экспериментальным данным модели этого объекта, действенной для всех эксплуатационных режимов, называется идентификацией объекта.

Задача идентификации, как правило, решается в два этапа. На первом этапе, который называется структурной идентификацией, формируется грубая модель объекта, которая аппроксимирует взаимосвязь вход-выход и содержит настраиваемые параметры. На втором этапе, который называется параметрической идентификацией, подбираются такие значения параметров, которые минимизируют расстояния между модельными и экспериментальными выходами объекта [9, 10].

Кроме того задачи идентификации могут рассматриваться с той точки зрения, что и задачи распознавания (классификации) образов [11], которые состоят в отнесении образца к одному из нескольких множеств, которые попарно не пересекаются. НС являются наиболее эффективным способом классификации потому, что они являются универсальным аппроксиматором, а также могут генерировать большое количество регрессионных моделей, используемых при решении задач классификации статистическими методами [4, 12].

В свою очередь, главная задача, которая ставится перед СВМ – идентификация

(распознавание) действий мошенника по нестандартному поведению профиля абонента, который можно охарактеризовать по таким признакам, как частота и направление вызовов, продолжительность разговора, идентификационные данные абонента и т. п. Таким образом, задачей СВМ является распознавание образа по входящему набору сигналов, которые снимаются непосредственно с коммутатора сканером служебной информации, и по сведениям об осуществленных звонках [7, 13]. Вся эта информация поступает в цифровом виде для по следующей ее идентификации в СВМ нейро-нечеткой сетью, где в нейронной части выполняется предыдущая обработка сигнальной информации [14], а в нечеткой – интерпретация и логический вывод полученных результатов [6, 7]. Кроме того начальным этапом проектирования такой СВМ является построение ее имитационной модели.

Итак, **целью** этой работы является построение и исследования модели НС нейронной части нейро-нечеткой системы выявления мошенничества в телекоммуникационных сетях на примере идентификации символов.

### Особенности идентификации символов

Среди большого класса программного обеспечения для моделирования НС использован инструментарий Neural Networks Toolbox пакета прикладных программ MATLAB 6.5 фирмы MathWorks, поскольку он довольно гибкий, обеспечен широким набором команд и функций для проектирования и исследования как статических, так и динамических НС [15]. Кроме того этот пакет обеспечен GUI-интерфейсом, который позволяет выполнить исследование НС даже неопытному пользователю.

Рассмотрим вариант реализации модели НС с помощью MATLAB для распознавания 10 арабских цифр.

В качестве датчика предполагается использовать систему распознавания, которая выполняет оцифровку каждого символа. Например, каждый символ будет представлен шаблоном размером 3x5. Тогда символ “4” может быть представлен, как показано на рис. 1,а и 1,б.

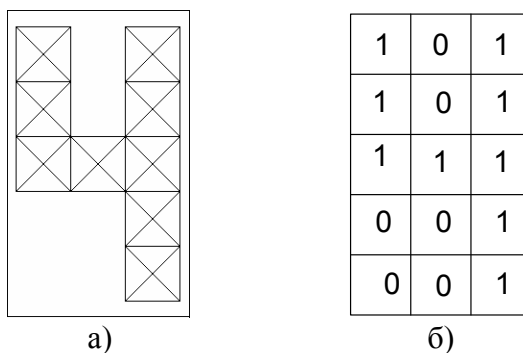


Рис. 1. Представление символа “4”

Однако система считывания символов обычно работает неидеально – и отдельные элементы символов могут оказаться искаженными (рис. 2).

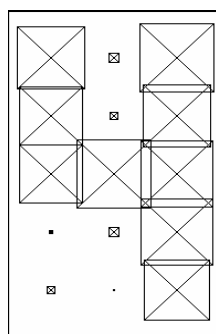


Рис. 2. Представление искаженного символа “4”

НС, которая моделируется, должна точно распознавать идеальные векторы входа и с максимальной точностью воссоздавать зашумленные векторы. М-функция `rgprob_n` определяет 10 векторов входа, каждый из которых содержит 15 элементов, этот массив называется *цифрами*. М-функция формирует исходные переменные `numrad` и `targets`, определяющие массивы цифр и целевых векторов.

Итак, на вход сети поступает вектор входа с 15 элементами; вектор выхода содержит 10 элементов, только один из которых равняется 1, а остальные – 0. Правильно функционирующая сеть должна ответить вектором со значением 1 для элемента, который отвечает номеру цифры от 0 до 9. Кроме того сеть должна быть способной распознавать символы в условиях действия шума. Предполагается, что шум – это случайная величина со средним значением 0 и стандартным отклонением, меньшим или равным 0.2 [15].

Для работы НС необходимо 15 входов и 10 нейронов в выходном слое. Для решения задачи задействуем трехслойную НС прямого распространения с пороговой, логарифмической сигмоидальной и линейной функциями активации во входном, скрытом и выходном слоях соответственно. Такие функции активации выбраны потому, что входной и выходной слою преимущественно осуществляют подготовку данных для дальнейшего использования, а также преобразования полученных результатов, и для этого вполне пригодны пороговая и линейная функции [16]. Что касается скрытого слоя, то он выполняет все основные вычисления, и его функция активации должна быть более “естественной”, например, логистической сигмоидальной или гиперболическим тангенсом.

### Имитационная модель нейронной сети

Модель структуры НС для распознавания арабских цифр приведена на рис. 3, где  $IW\{1,1\}$ ,  $LW\{2,1\}$ ,  $LW\{3,2\}$  – весовые матрицы входного, скрытого и выходного слоев соответственно;  $b\{1\}$ ,  $b\{2\}$ ,  $b\{3\}$  – векторы смещения первого, скрытого и выходного слоев соответственно;  $p1$  – входной информационный векторный массив;  $y$  – выходной векторный массив;  $a1$ ,  $a2$ ,  $a3$  – массивы значений функций активации первого, скрытого и выходного слоя соответственно.

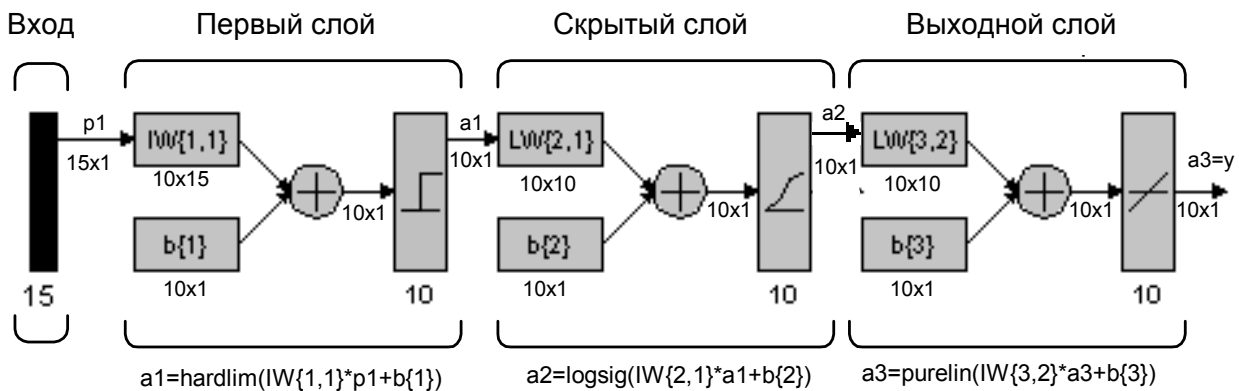


Рис. 3. Структурная схема нейронной сети

Скрытый слой имеет 10 нейронов, тем не менее, если при обучении сети возникают осложнения, можно увеличить количество нейронов этого уровня. Сеть учится так, чтобы сформировать единицу в единственном элементе вектора выхода, позиция которого соответствует номеру символа, и заполнить другую часть вектора нулями. Впрочем наличие шумов может приводить к тому, что НС не будет формировать векторы выхода, которые состоят точно из единиц и нулей. Поэтому после окончания этапа обучения выходной сигнал обрабатывается М-функцией `compnet`, которая присваивает значение 1 единственному элементу вектора выхода, а всем другим – значение 0.

*Инициализация сети.* Вызываем М-файл `rgprob_n`, формирующий массив векторов входа `numrad` размером 15x10 с шаблонами символов цифр от 0 до 9 и массив целевых векторов

targets:

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

Трехслойную нейронную сеть прямого распространения создаем с помощью команды 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;%инициализация весов
net.b{3}=net.b{3}*0.01;%инициализация смещения
```

*Обучение.* Чтобы НС могла обрабатывать искаженные векторы входа, выполним ее обучение как на идеальных, так и на искаженных векторах. Сначала сеть учится на идеальных векторах, пока не будет обеспечена минимальная средняя квадратическая погрешность. Потом сеть учится на 10 наборах идеальных и искаженных векторов. Обучение выполняется с помощью функции traingdx, которая реализует метод обратного распространения ошибки с возмущением и адаптацией параметра скорости настройки.

*Обучение при отсутствии шума.* Сеть сначала учится при отсутствии шума с максимальным количеством циклов обучения 5000 или до достижения допустимой средней квадратической погрешности, которая равняется 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)
```

Как видно из рис. 4, сеть научилась уже после выполнения 84 эпох (циклов).

*Обучение при наличии шума.* Чтобы спроектировать НС, невосприимчивую к влиянию шума, научим ее с применением двух идеальных и двух искаженных копий векторов набора цифр от 0 до 9. Целевые векторы состоят из четырех копий векторов.

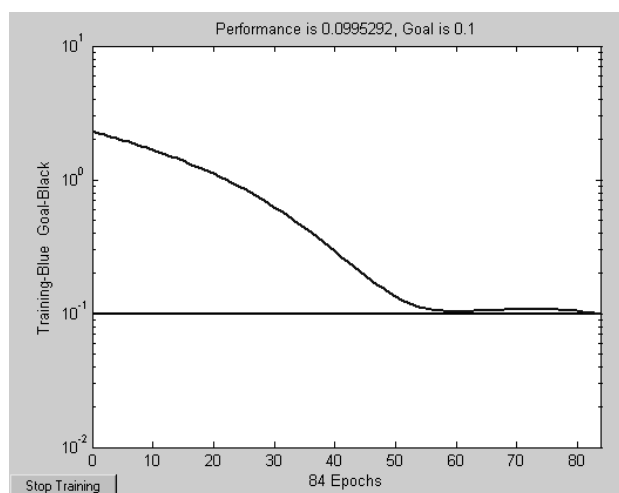


Рис. 4. Обучение сети при отсутствии шума

Искаженные векторы имеют шум со средним значением 0.1 и 0.2. Это учит НС правильно распознавать искаженные символы и идеальные векторы. При обучении с шумом

максимальное количество циклов обучения сократим до 300, а допустимую погрешность увеличим 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
```

Поскольку НС училась при наличии шума, то имеет смысл повторить ее обучение без шума, чтобы гарантировать, что идеальные векторы входа классифицируются правильно:

```
netn.trainParam.goal=0.1;%Предельная погрешность
netn.trainParam.epochs=500;%Максимальное количество циклов обучения
net.trainParam.show=5;%Частота вывода результатов на экран
[netn,tr]=train(netn,P,T)
```

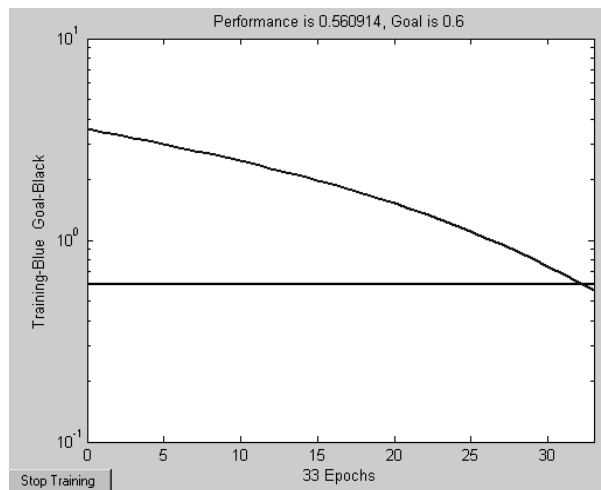


Рис. 5. Обучение сети при наличии шума

Если необходима более высокая точность распознавания, сеть может быть обучена на протяжении более продолжительного времени, или с использованием большего количества нейронов в скрытом слое. Можно также увеличить размер векторов, чтобы воспользоваться шаблоном с более мелкой сеткой, например 5x7 или 10x14 точек вместо 3x5.

Проверим работу НС для распознавания символов. Для графического вывода символа на экран используется М-функция plotchar\_n. Сформируем искаженный вектор входа для цифры 2 (рис. 6):

```
noisy_num=numpad(:,3)+randn(15,1)*0.2;
plotchar_n(noisy_num);% Искраженная цифра 2 (рис.6)
A2=sim(net,noisy_num);
A2=compet(A2);
answer=find(compet(A2)==1)
answer = 3
plotchar_n(numpad(:,answer));%распознанная цифра 2
НС восстановила символ “2” без ошибок (рис. 7).
```

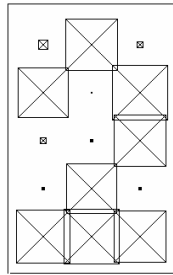


Рис. 6. Зашумленный символ

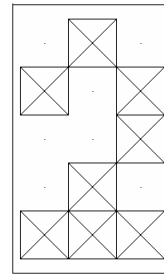


Рис. 7. Распознанный символ

### Выводы

Итак, описанная модель НМ продемонстрировала, как может функционировать система распознавания образов. Сеть была научена несколько раз при разных векторах входа. Обучение НС на разных наборах искаженных векторов позволило научить ее работать с искаженными данными, что характерно для реальной практики. Кроме того во входном слое НС использованы пороговые нейроны, что доказывает возможность и целесообразность аппаратной реализации нейроструктур с использованием простых перцептронных нейронов для проектирования интеллектуальных СВМ в телекоммуникационных сетях.

### СПИСОК ЛИТЕРАТУРЫ

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](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: «УНИВЕРСУМ-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 Анатолий Степанович** – директор института, профессор кафедры автоматизации и информационно-измерительной техники;

**Мартинюк Татьяна Борисовна** – доцент кафедры лазерной и оптоэлектронной техники;  
Институт автоматизации, электроники и компьютерных систем управления, Винницкий национальный технический университет

**Куперштейн Леонид Михайлович** – старший преподаватель кафедры информационных систем и технологий.

Винницкий финансово-экономический университет