УДК 004.032.26 + 342.7 : 004.056.5 Вестник УрФО № 4(22) / 2016, с. 26–30

Трунин А. М., Рагозин А. Н.

НЕЙРОННЫЕ СЕТИ В ЗАЩИТЕ ПЕРСОНАЛЬНЫХ ДАННЫХ

Защита персональных данных является актуальной проблемой в сфере информационной безопасности. Персональные данные, такие как идентификационные коды кредитных карт, данные паспорта, номера банковских счетов, хранящиеся в электронном виде с использованием электронных каталогов или баз данных, требуют уникального подхода к защите информации и быстрой реакции на их изменение или использование. Данная работа рассматривает применение нейронных сетей: самоорганизующихся сетей Кохонена, иерархического кластер-анализа и нейронных сетей специальной архитектуры в защите и обработке большого числа персональных данных, содержащихся в каталогах или базах данных.

Ключевые слова: *защита информации, информация, персональные данные, кластер-анализ, нейронные сети.*

Trunin A. M., Ragozin A. N.

NEURAL NETWORKS IN THE PROTECTION OF PERSONAL DATA

Protection of personal data is an actual problem in the field of information security. Personal data, such as the identity of the credit card codes, passport details, bank account numbers stored in electronic form with the use of electronic catalogs or databases require a unique approach to data protection, and rapid response to changing them or use. This work considering the use of neural networks: self-organizing Kohonen networks, hierarchical cluster analysis and neural network with special architecture in protection and processing of a large number of personal data contained in directories or databases.

Keywords: *Data protection, information, personal data, cluster analysis, specific neural network.*

Нейронные сети способны обрабатывать огромные массивы данных по определенным, необходимым для решения задачи, правилам. От правильно подобранного типа нейросети зависит качество решения задачи. Далее рассмотрены основные типы нейронных сетей и иерархического кластер анализа, применяющиеся в решении задач защиты данных.

Нейронные сети Кохонена– класс нейронных сетей, основным элементом которых является слой Кохонена[3]. Слой Кохонена состоит из адаптивных линейных сумматоров («линейных формальных нейронов»). Как правило, выходные сигналы слоя Кохонена обрабатываются по правилу «победитель получает всё» - наибольший сигнал превращается в единичный, остальные обращаются в ноль[3].

Слой Кохонена состоит из некоторого количества параллельно действующих линейных элементов. Все они имеют одинаковое

число входов и получают на свои входы один и тот же вектор входных сигналов. На выходе j-го линейного элемента получаем сигнал:

$$y_j = \omega_{j0} + \sum_{i=1}^{m} \omega_{ji} x_i \qquad (1)$$

где:

- ω_{ji} – весовой коэффициент i–го входа j-го нейрона;
- i – номер входа;
- j – номер нейрона;
- ω_{j0} – пороговый коэффициент.

После прохождения слоя линейных элементов сигналы посылаются на обработку по правилу «победитель получает всё»: среди выходных сигналов выполняется поиск максимального нейрона[3]. Окончательно, на выходе сигнал с определенным номером равен единице, остальные – нулю. Если максимум одновременно достигается для нескольких сигналов, то:

- либо принимаются все соответствующие сигналы равными единице;
- либо равным единице принимают только первый сигнал в списке (по соглашению).

Кластер-анализ – это множество вычислительных процедур, которые формируют либо выявляют иерархии (разбиения), лежащие в основе тех или иных совокупностей данных. Анализ данных представляет собой множество вычислительных процедур, которые описывают, распознают или идентифицируют структуры, лежащие в основе скоплений точек, обычно принадлежащих пространству малой размерности, сконструированному по совокупности данных (многомерное шкалирование, факторный анализ и подобные методы).

Алгоритмы кластер-анализа в совокупности с самоорганизующимися сетями Кохонена представляют собой инструмент быстрой и качественной обработки информации, ее анализа и своевременной защиты[2].

Обработка персональных данных, их классификация и обеспечение их сопоставления между собой для дальнейшей защиты происходит в два этапа.

Первый этап – это анализ всех данных при помощи самоорганизующихся сетей Кохонена, классификация их и предоставление для дальнейшей обработки.

Вся представленная к обработке информация нормируется, то есть приводится к подобию на основе общих критериев для дальнейшего сопоставления данных между собой. Далее из нормированных данных формируется вектор входных данных, который подается на вход самоорганизующейся нейронной сети Кохонена.

Все данные, поданные в виде векторов, проходят процедуру «кластеризации», то есть делятся на группы «похожих» объектов, называемых кластерами. Кластеризация позволяет сгруппировать сходные данные, что облегчает их последующий анализ. Формально задача кластеризации описывается следующим образом: из множества объектов $I = \{i_1, i_2, \ldots, i_n\}$ каждый из которых характеризуется вектором $x_j = \{x_{j1}, x_{j2}, \ldots, x_{jm}\}$, j=1,2,…,n атрибутов (параметров).

Требуется построить множество кластеров C и отображение F множества I на множество C, то есть $F: I \to C$. Задача кластеризации состоит в построении множества

$$C = \{C_1, C_2, \ldots, C_k\}$$

где C_k – кластер, содержащий «похожие» объекты из множества I:

$$C_k = \{i_j, i_p | i_j \epsilon I, i_p \epsilon I \text{ и } d(i_j, i_p) < \sigma\}$$

σ – величина, определяющая меру близости для включения объектов в один кластер, $d(i_j, i_p)$ – мера близости между объектами, называемая расстоянием.

К требуемому объему информации, содержащему персональные данные, представляемые для обработки и своевременного анализа, применяется классификация[1] при помощи самоорганизующихся сетей Кохонена, которые обучаются самостоятельно на основе предоставленных данных. После прохождения данных через сеть Кохонена на выходе данные сортируются и выводятся в виде вектора $y(1)$.

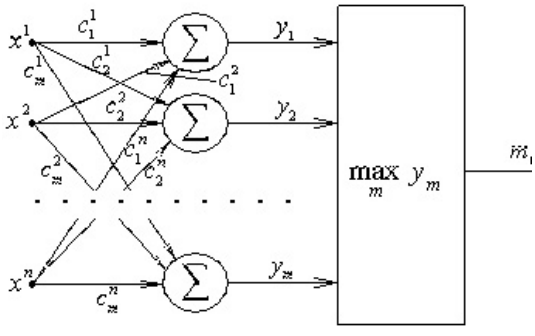

Рис. 1. Структура сети Кохонена

Второй этап – это интерпретация результатов, полученных при обработке данных нейронными сетями Кохонена, при помощи иерархического кластер-анализа.

Для этого вектор $y(1)$ проходит кодирование и описание данных (отбор и кодирование таблиц данных, первичную элементарную обработку, кодирование и выбор метрики), анализ и классификацию (структурирование и синтез, построение дендрограмм) и интерпретацию полученных результатов конкретным графическим представлением и дополнительным наглядным описанием. После проделанных процедур данные представляются в удобном и доступном для анализа виде с целью их критического исследования.

Для решения задач защиты данных, так же находят применение нейронные сети со специально-конструируемой структурой.

Общим для всех нейросетей является этап подготовки данных, а именно их нормировка, формирование входных векторов (данных для анализа) и создании целевой функции, на основе которой происходит обучение данной нейросети.

Главным отличием специальной нейронной сети от кластерного анализа и нейронных сетей Кохонена является возможность конструирования специальной структуры обработки данных для определенной задачи.

Для проведения анализа возможностей данных нейросетей и получения наглядных результатов был использован программный пакет «Matlab».

Первым этапом в формировании выбранной нейронной сети является нормирование входных и целевых данных для подачи на определенный тип нейросети.

Задание нейронной сети имеет определенный синтаксис:

net=network(numInputs,numLayers,biasConnect,inputConnect,layerConnect,outputConnect),

где:

- network – функция задания типа нейронной сети, в данном случае функция задания собственной нейросети пользователя;
- numInputs – определяет количество входов сети;
- numLayers – определяет количество слоев сети;
- biasConnect – определяет, какие слои имеют смещения;
- inputConnect - определяет, какие слои обладают связями со входами;
- layerConnect - определяет какие слои связаны с другими слоями;
- outputConnect - определяет какие слои генерируют выходы сети;

Для задания алгоритмов адаптации, инициализации, тренировки и оценки функционирования сети используются следующие функции:

- adaptFcn - определяет функцию, которая будет использована для адаптации сети (net.adaptFcn).
- initFcn - определяет функцию, используемую для инициализации матриц весов и векторов смещений сети (net.initFcn).
- performFcn – определяет функцию, используемую для оценки функционирования сети (net.performFcn).
- trainFcn - определяет функцию, используемую для тренировки сети (net. trainFcn).

Одной из самых важных задач в формировании нейронной сети является задача подбора правильной функции тренировки сети. Функции тренировки сети, доступные в программном пакете Matlab, позволяют задать требуемый для решения задачи процесс обучения и оптимизировать его выполнение. Ниже приведены некоторые функции обучения нейросети:

- trainb – пакетная тренировка с использованием правил обучения для весов и смещений;
- trainbfg – тренировка сети с использованием квази – Ньютоновского метода BFGS;
- trainc – использование приращений циклического порядка;

Листинг функции в программном пакете Matlab, задающей нейронную сеть с собственной архитектурой:

```
function net = create_network(inputs, outputs)
Создание сети
net = network();
net.numInputs = 1; Количество входов
net.numLayers = 2; Количество слоев, первый - скрытый, второй - выходной
Входы
net.inputs{1}.size = inputs;
net.inputs{1}.processFcns = {'removeconstantrows', 'mapminmax'};
```

```
    Выходы
    net.outputs{2}.processFcns = {'removeconstantrows', 'mapminmax'};
    Тренировка
    net.dividefcn = 'dividerand';
    net.divideParam.trainRatio = 70/100;     Часть выборой для тренировки
    net.divideParam.valRatio = 15/100;       Часть выборок для проверки
    net.divideParam.testRatio = 15/100;      Часть выборок для конечного
тестирования
    net.performFcn = 'mse';                  Функция оценки качества
    net.trainFcn = 'trainlm';                Функция тренировки
    net.trainParam.epochs = 500;             Максимальное количество итераций
    net.trainParam.goal = 0;                 Допустимая погрешность
    Функция адаптации
    net.adaptFcn = 'adaptwb';                Функция адаптации
    Инициализация
    net = init(net);
    end
```

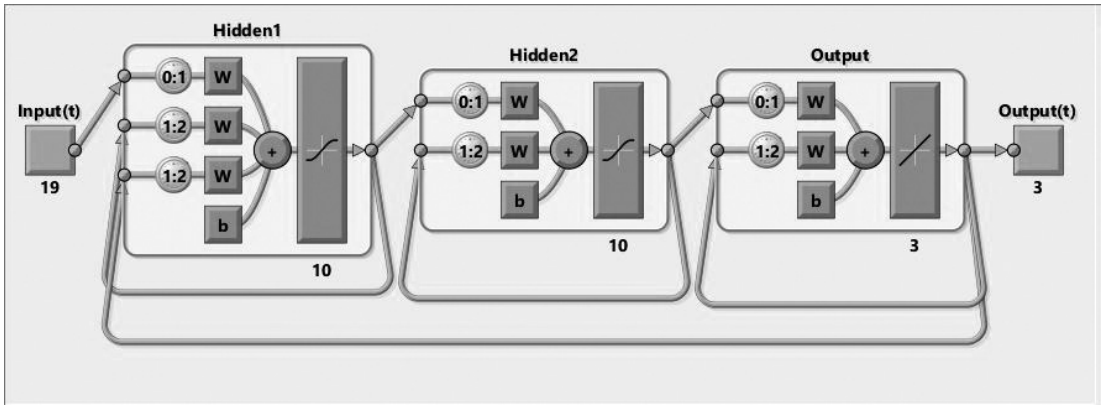

Рис. 2. Структура нейронной сети

Пример структуры нейронной сети приведен на рисунке 2.

Правильность выбора типа нейронной сети и функции обучения влияет на правильность решения и на качество защиты персональных данных пользователя. Благодаря своей гибкости, нейронные сети, сконструированные таким образом, позволяют решать задачи любой сложности и направленности, в том числе и задачи защиты данных.

Примечания

1. Мищенко Е. Ю., Соколов А. Н. Количественный анализ процедуры обезличивания персональных данных. Метод перемешивания // Вестник УрФО. Безопасность в информационной сфере /№ 3(21) / 2016, с. 30–37.

2. Жамбю М. Иерархический кластер-анализ и соответствия // Перевод с фр., 1988. С. 11–76.

3. Хайкин С. Нейронные сети: полный курс. – М.: Вильямс, 2006. – 1104 с.

ТРУНИН Андрей Михайлович, студент группы КЭ-437 высшей школы электроники и компьютерных наук ФГАОУ ВО «Южно-Уральский Государственный Университет» (Национальный исследовательский университет). Россия, 454080, г.Челябинск, проспект Ленина, д 76. E-mail: truninandrey@mail.ru.

РАГОЗИН Андрей Николаевич, научный рук., кандидат технических наук, доцент кафедры инфокоммуникационных технологий, доцент кафедры защиты информации ФГАОУ ВО «Южно-Уральский Государственный Университет» (Национальный исследовательский университет).Россия, 454080, г.Челябинск, проспект Ленина, д 76.

TRUNIN Andrey Mikhailovich, the student of higher School of Electronics and Computer Science of the FGAOU «South Ural State University» (National Research University). Russia, 454080, Chelyabinsk, Prospekt Lenina, 76. E-mail: truninandrey@mail.ru.

RAGOZIN Andrey, scientific chief., Ph.D., Candidate of Technical Sciences, assistant professor of information and communication technologies, assistant professor of information security FGAOU «South Ural State University» (National Research University). Russia, 454080, Chelyabinsk, Prospekt Lenina, 76.