

**В.А. Панов, С.В. Семенов**

Пермский национальный исследовательский  
политехнический университет

## **ИСПОЛЬЗОВАНИЕ ПРОГРАММНОГО ПАКЕТА *NEUREX* ДЛЯ ИДЕНТИФИКАЦИИ СИГНАЛА С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ**

*Представлен учебный пример решения задачи нейросетевой идентификации для цикла лабораторных работ по системам искусственного интеллекта. Разобран процесс создания и обучения нейронных сетей в программном пакете *Neurex*.*

Сегодня искусственные нейронные сети являются одной из самых популярных тем научных и прикладных работ по самым различным дисциплинам. Причинами этого являются их универсальность, адаптивность, высокий исследовательский потенциал. Однако следует не забывать, что, несмотря на относительную новизну данного направления, уже существуют определенные общепринятые подходы и стандарты при работе с нейронными сетями. Также следует помнить про особый терминологический аппарат, сформировавшийся за годы исследований. Поэтому чрезвычайно важной представляется задача создания учебных работ, позволяющих инженеру познакомиться с базовыми принципами функционирования нейронных сетей еще на этапе обучения. В данной статье рассматривается решение задачи нейросетевой идентификации с помощью пакета *Neurex*, которое могло бы быть использовано в качестве учебного примера, входящего в цикл лабораторных работ по системам искусственного интеллекта. Кроме этого, ввиду отсутствия примеров создания нейронных сетей в вышеозначенном программном продукте, данная работа могла бы быть интересна в качестве пособия исследователям, решившим использовать *Neurex* для решения своих задач.

Сегодня существует достаточно большое количество программ, позволяющих познакомиться с работой нейронных сетей:

*NeuroFilter*, *Deductor*, *Neuro-Programmer*. Все они обладают примерно одинаковым функционалом в плане работы с нейросетями: возможность проектирования, обучения и тестирования сети. Однако данные программные продукты предоставляют доступ ко всем своим возможностям только после их приобретения. На их фоне выгодно выделяется пакет *Neurex* [1]. Он является полностью бесплатным и предназначен для решения наиболее популярной задачи, решаемой с помощью нейронных сетей, а именно для восстановления неизвестных функциональных зависимостей (задачи распознавания образов, сжатия изображений). Кроме этого, в программе имеется возможность просмотра текущего состояния обучения в текстовом виде и в графическом – в трехмерных и двумерных проекциях. Из всего вышеперечисленного следует, что *Neurex* прекрасно подходит для учебных целей. Рассмотрим пример создания нейронной сети для идентификации сигнала с помехой (рис 1).

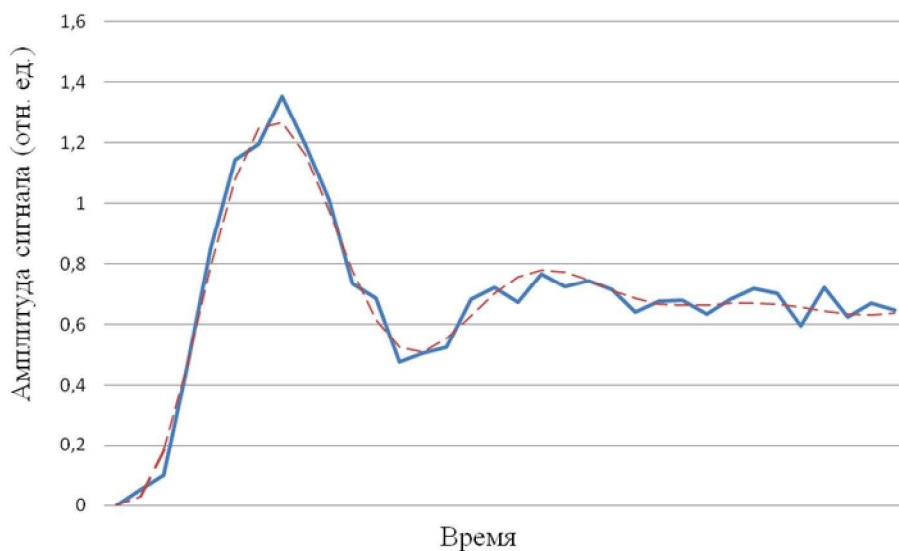


Рис. 1. Идентифицируемый сигнал: — с шумом, - - - - без шума

Задача идентификации с помощью нейронных сетей состоит в установлении ассоциативной связи (нахождении значений весовых коэффициентов) между входным и выходным векторами нейронной сети таким образом, чтобы на некотором временном участке значения на выходе сети совпадали со значениями искомой функции с заданной погрешностью.

Процесс идентификации с помощью нейронных сетей состоит из следующих этапов: подбор архитектуры нейронной сети; выбор обучающих данных и структуры входных векторов; обучение нейронной сети; тестирование сети на контрольном множестве данных и при необходимости ее дообучение.

Для решения задачи идентификации используем персептрон с одним скрытым слоем. Объем входного слоя может быть любым, но не больше количества элементов учебной последовательности. Количество выходных нейронов определяется количеством идентифицируемых параметров. Самая трудная задача – подбор количества нейронов скрытого слоя. Если их слишком мало, то погрешность обучения невозможно уменьшить до требуемого уровня. Слишком большое их количество приводит к росту погрешности обобщения. Количество скрытых нейронов можно либо подобрать экспериментально так, чтобы уменьшить до минимума погрешность обобщения, либо применить один из методов построения оптимальной структуры сети [2]. В нашем случае путем подбора число скрытых нейронов получилось равным четырем. Итоговая структура сети представлена на рис. 2.

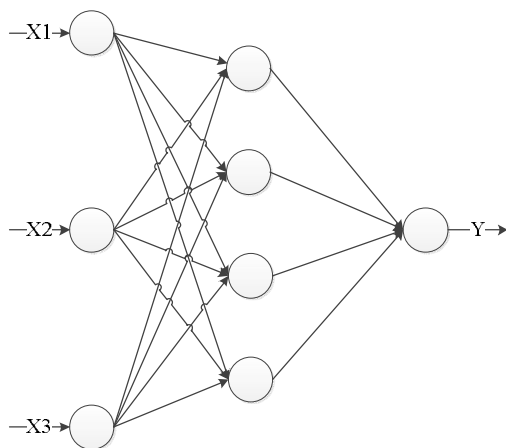


Рис. 2. Структура нейронной сети

Как видно из рис. 2, входной вектор будет состоять из трех элементов, а выходной из одного. Следовательно, обучающая последовательность должна содержать данные для входного вектора и желаемое значение выходного параметра. Стандартным подходом при формировании учебной выборки является разбиение всего временно-го ряда значений идентифицируемого сигнала на «окна» [3].

Данный подход заключается в следующем: пусть вся совокупность значений представлена следующим рядом:

$$X = (x_1, x_2, \dots, x_n), \quad (1)$$

тогда окна размером  $l$  будут формироваться следующим образом:

$$\begin{aligned} &x_1, x_2, \dots, x_l, \\ &x_2, x_3, \dots, x_{l+1}, \\ &\vdots \\ &x_k, x_{k+1}, \dots, x_{k+l-1}, x_n. \end{aligned} \quad (2)$$

На основе этих окон строятся правила обучения сети методом обратного распространения:

$$x_k, x_{k+1}, \dots, x_{l+i-1} \rightarrow x_{i+1}. \quad (3)$$

Для того чтобы создать сеть в программе *Neurex*, необходимо сделать следующие действия:

1) выделить в объекте набор его входных (независимых) параметров;

2) выделить в объекте набор выходных (предположительно зависимых от входных) параметров;

3) определить тип этих параметров – целый в некотором диапазоне, вещественный или перечислимый;

4) на специальном скрипт-языке (NNScript), интерпретатор которого реализован в *Neurex*, описать задачу и нейронную сеть для ее решения;

5) создать на основе подготовленного сценария многослойный персептрон и обучить его одним из двух методов – алгоритмом обратного распространения или стохастическим алгоритмом (обучением Коши).

В соответствии с вышеозначенной методикой скрипт, реализующий необходимый нам вариант сети, будет выглядеть следующим образом:

```
float Noise[-2, 2] //вещественный тип входных
//данных
input { //входные переменные
    Noise "X1",
    Noise "X2",
    Noise "X3"
}
output { //выходные переменные
```

```

    Noise "Y"
}
neuralnet{           //структура сети
    sigmoid: exponential, // экспоненциальный СИГМОИД
    hiddens: 4, //четыре нейрона в единственном
//скрытом слое
    algorithm(backpropagation), //обучение обратным
//распространением
    weights:auto, //доверим инициализацию начальных
//весов системе
    velocity: 0.1 // скорость обучения 0.1
}
learnseries{//набор обучающих последовательностей
(0, 0.051979, 0.102238)-> (0.458179),
(0.051979, 0.102238, 0.458179)->(0.850550),
//всего 45 элементов в обучающей выборке
(0.691518, 0.661367, 0.661367)-> (0.729672)
}

```

Далее данный скрипт обрабатывается в программе *Neurex* и получается следующее пространство решений для нейронной сети (рис. 3).

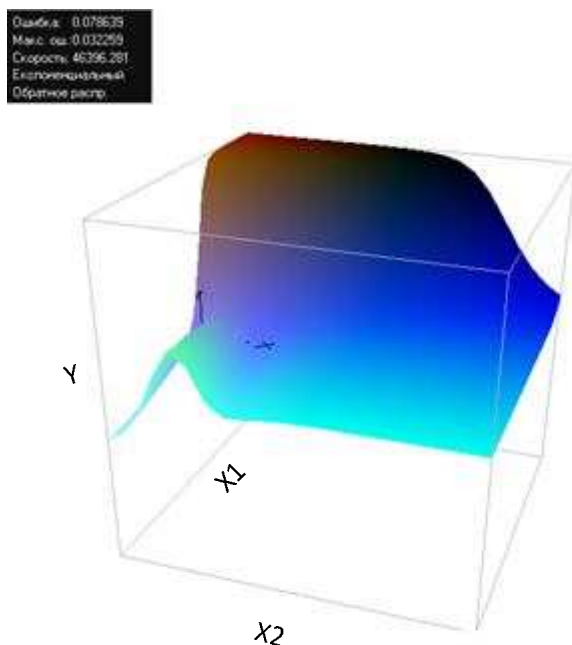


Рис. 3. Пространство решений для нейронной сети (X1, X2 – входы (X3 скрыт), Y – выход)

Вид идентифицируемого сигнала можно определить, посмотрев фронтальное сечение пространства решений (рис. 4).

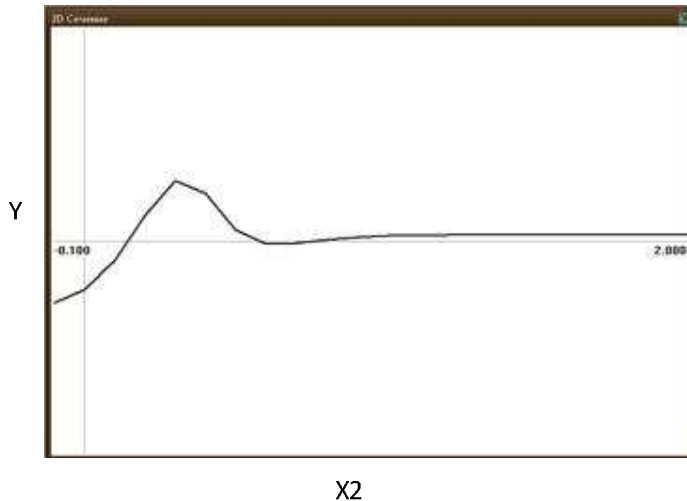


Рис. 4. Вид идентифицируемого сигнала

Из графика видно, что полученное решение достаточно сильно напоминает искомый сигнал. Ошибка при обучении составила 7,8 %.

В заключение хотелось бы отметить, что для получения достаточно точных результатов требуются тщательный подбор учебных данных и достаточно большое время обучения (до 4 часов на слабых компьютерах). Однако, несмотря на это, программа *Neurex* полностью раскрывает процесс реализации нейронных сетей и может быть использована в учебном процессе.

### Библиографический список

1. Программа для построения нейронных сетей – NeurexNeurex 1.03 [Электронный ресурс] (Сайт Олега Куртцева). – 2011. – URL: <http://nnet.chat.ru/neuroex/index.htm>
2. Оссовский С. Нейронные сети для обработки информации / пер. с пол. И.Д. Рудинского. – М.: Финансы и статистика, 2002. – 344 с.
3. Уоссерман Ф. Нейрокомпьютерная техника: теория и практика. – М.:Мир, 1992. – 256 с.

Получено 05.09.2011