

## ЛАБОРАТОРНАЯ РАБОТА № 5.

### СОЗДАНИЕ И ОБУЧЕНИЕ ПРОСТЕЙШЕЙ НЕЙРОННОЙ СЕТИ

**Цель** – освоение основных приемов работы с демонстрационной версией программного продукта *TRAJAN* в ходе создания и обучения простейшей нейронной сети.

#### Задание

1. Создать и обучить нейронную сеть, которая будет способна решать логическую задачу исключающего «ИЛИ». Таблица истинности для весьма полезной логической функции приведена в табл. 1.
2. Проверить работоспособность нейронной сети.
3. Ответить на вопросы для самопроверки № 1 – 4.

**Таблица 1. Таблица истинности для логической функции исключающего «ИЛИ»**

Вход 1	Вход 2	Истина?
0	0	1
0	1	0
1	0	0
1	1	1

#### Создание нейронной сети

Новая нейронная сеть создается в *TRAJAN* с помощью окна *Network Creating (Создание сети)*, которое доступно из меню *File/New/Network* или по нажатию соответствующей кнопки на панели инструментов.

После того, как на экране появится окно *Network Creating* для создания новой нейронной сети, следует произвести следующие действия.

#### Выбор типа нейронной сети

Демонстрационная версия *TRAJAN* предлагает два типа нейронных сетей. Для решения задач, представленных в данном лабораторном практикуме, рекомендуется использовать нейронную сеть типа *многослойный перцептрон*, которая выбрана в данном окне по умолчанию.

#### Определение количества слоев в нейронной сети и их размерностей

При задании количества слоев вашей нейронной сети, следует учитывать следующие особенности пакета *TRAJAN*:

1. Программный продукт поддерживает максимальный размер нейронные сети: 128 слоев по 128 нейронов в каждом, при этом первый слой всегда является входным и используется только для получения сетью исходных данных, а последний – выходным, и выходы его нейронов являются выходами всей сети в целом.
2. Для решения поставленной задачи рекомендуется использовать простейшую структуру нейронной сети, состоящую из трех слоев: входной

слой с двумя нейронами, скрытый слой с двумя нейронами и выходной слой с одним нейроном (2-2-1).

Для задания количества нейронов в каждом слое используется матрица, представленная в окне *Network Creating*. Она выглядит как небольшая электронная таблица.

Необходимо определить количество нейронов в каждом слое сети с помощью первой ячейки этой матрицы, при этом любые слои с нулевым количеством нейронов будут проигнорированы.

После задания количества нейронов в каждом слое нейронной сети, *TRAJAN* самостоятельно определит количество слоев путем выбора из матрицы тех слоев, у которых количество нейронов отлично от нуля.

***Примечание.** Можно заметить, что матрица содержит строку для задания «ширины» каждого слоя. Строка редко используется в *TRAJAN* для карт Кохонена, хотя с помощью нее можно задавать и ширину слоев для нейронных сетей некоторых других типов.*

## Обучение нейронной сети

Алгоритм обратного распространения обучает нейронную сеть, используя доступные ему данные, которые хранятся в наборе представительских выборок для обучения. На каждой итерации (в терминах программного продукта *TRAJAN* – «эпохе»), нейронной сети предоставляется весь подготовленный набор обучающих пар. Выходы, получаемые нейронной сетью, сравниваются с желаемыми результатами. При этом ошибка нейронной сети вычисляется как разность между желаемыми и фактическими результатами и используется для регулирования весов нейронов в сети.

Для обучения нейронной сети необходимо:

- открыть окно *Training Error Graph*, используя позицию меню *Statistics/Training Graph*;
- открыть окно *Back Propagation*, используя позицию меню *Training/Backprop*;
- расположить на экране окна так, чтобы они были оба видны и не перекрывали друг друга;
- запустить алгоритм обучения путем нажатия на кнопку *Train (Обучение)* в окне *Back Propagation*. При этом зависимость среднеквадратической ошибки обучения нейронной сети от числа используемых итераций будет вычерчиваться на графике в окне *Training Error Graph*;
- увеличить число итераций в окне *Back Propagation* и обучить нейронную сеть вновь, нажав кнопку *Train*.

Вначале моделирования при использовании небольшого числа итераций, СКЗ ошибки уменьшается, но незначительно. Это обусловлено тем, что задача «исключающего «ИЛИ» для ИНС, как не парадоксально, гораздо сложнее в решении, чем многие более сложные задачи.

Окно *Training Error Graph* отображает общую ошибку обучения нейронной сети, однако иногда бывает полезно пронаблюдать за работой

сети при использовании отдельно взятой обучающей пары. Данный режим реализуется в *TRAJAN* с помощью окна ***Pattern Error***.

### **Запуск сети**

После обучения нейронная сеть готова к запуску, причем запустить ее на выполнение можно несколькими способами.

### **Запуск, используя текущий набор представительских выборок**

Нейронная сеть может быть запущена с предъявлением полного набора представительских выборок, использованных ранее при ее обучении, или выполняемыми наборами по одиночке. При этом необходимо воспользоваться пунктом меню ***Run/Single Pattern***, чтобы получить информацию о работе нейронной сети при предъявлении одной отдельно взятой представительской выборки или целого набора представительских выборок.

### **Запуск индивидуальной представительской выборки, не входящей в набор обучающих пар**

При решении целого ряда задач необходимо проверять работу нейронной сети на представительской выборке, которая не входила в набор обучающих пар, использованных ранее при обучении. Например:

1. Прогнозирование появления новых данных с заранее неизвестными нейронной сети выходами. Если выходы заранее известны, то можно оценить качество работы подготовленной нейронной сети. В противном случае, результаты, полученные при запуске, могут быть использованы в качестве прогноза. Данный тип задач для нейронных сетей будет рассмотрен в лабораторной работе № 5.
2. Распознавание образов (задача будет рассмотрена в работе № 3). В этом случае оценивается чувствительность нейронной сети к небольшому изменению параметров исследуемого вектора, с помощью которого проводилось обучение.

### **Замечание**

Нейронная сеть, подготовленная в данной лабораторной работе, была обучена с использованием всех возможных для нее обучающих пар, поэтому она может быть запущена на выполнение с использованием каждой из четырех представительских выборок. Следовательно, можно будет оценить работу нейронной сети на каждой из них.

При запуске, также возможно единовременное использование всего набора представительских выборок для оценивания общих параметров работы нейронной сети.

## ЛАБОРАТОРНАЯ РАБОТА № 6.

### ОПРЕДЕЛЕНИЕ НАПРАВЛЕНИЯ ДВОИЧНОГО СДВИГА

**Цель** – построение, обучение и тестирование нейронной сети, предназначенной для определения направления сдвига двоичного кода.

#### **Задание**

1. Создать и обучить нейронную сеть для определения направления циклического сдвига четырехпозиционного двоичного кода.
2. Проверить работоспособность нейронной сети.
3. Ответить на вопросы для самопроверки № 5 – 8.

#### **О применении нейронных сетей для решения задачи классификации**

Типовая задача нейронных сетей – классификация того или иного исследуемого вектора (объекта). Получив в процессе обучения исходные данные об объекте, нейронная сеть определяет, к какому из множества классов принадлежат исследуемые векторы.

Проблема исключающего ИЛИ, рассмотренная в предыдущей лабораторной работе, является примером решения именно такой задачи. Если исследуемый вектор может принадлежать только к одному из двух классов, то задача называется двухклассной. Задача, поставленная в данной работе, сводится к двухклассной.

Простейший путь решения задачи двухклассной классификации при помощи нейронных сетей – формирование у сети единственного выхода, который получает значение 1 для одного класса и 0 – для другого. Значения, лежащие внутри данного диапазона, характеризуют степень принадлежности объекта к тому или иному классу.

Действительно, на том или ином выходе многоуровневого персептрона практически невозможно получить значения равные точно 0 или 1, хотя к этим значениям иногда можно подойти довольно близко.

Таким образом, для решения двухклассных задач с использованием одного выхода необходимо задаваться *уровнем доверия*, например: если значения выхода выше 0,95 – считать, что объект (исследуемый вектор) принадлежит к одному классу, а если ниже 0,05 – к другому.

#### **Решение задачи классификации в TRAJAN**

Нейронная сеть с предъявлением единственной представительской выборки запускается в окне ***Run Single Pattern*** (*Запустить единственный образец*) или в окне ***Run One-off Pattern*** (*Запустить одиночный образец не входящий в представительскую выборку*).

TRAJAN сравнивает выходную величину сети с пределами доверия и определяет:

- если выход выше установленного верхнего порога, исследуемый вектор (объект) классифицируется положительно;
- если выход ниже установленного нижнего порога, то сообщается о негативной классификации;
- если значение выхода находится между порогами, то сообщается о том, что исследуемый вектор (объект) классифицировать не удалось.

Общая статистика результатов классификации осуществляется при нажатии кнопки **Run** в окне **Statistic/Classification**, которое открывается из меню **Statistic/Classification**. Статистика в этом окне отображается в виде матрицы, содержащей один столбец для каждого класса. Каждый столбец содержит две секции: «Общая статистика» и «Статистика процесса классификации», разделенные широкой горизонтальной чертой.

Секция «Общая статистика» содержит следующую информацию:

- **Total (Всего)** – количество образцов данного класса в наборе.
- **Correct (Правильные)** – количество образцов данного класса правильно классифицированных нейронной сетью.
- **Wrong (Неправильные)** – количество образцов неправильно классифицированных сетью (как принадлежащих к другому классу).
- **Unknown (Неизвестные)** – количество образцов данного класса, которые нейронная сеть не смогла классифицировать.

Секция «Статистика процесса классификации» показывает, сколько исследуемых векторов (представительских выборок) было отнесено к каждому классу. При этом неклассифицированные векторы в данной секции не отображаются.

## Нейронная сеть для определения направления двоичного сдвига

Для решения поставленной задачи, следует построить и обучить нейронную сеть, которая должна будет определять направление двоичного сдвига.

Операция двоичного сдвига является типичной для многих языков программирования. Сущность ее заключается в том, что число представляется в двоичном коде, а затем с полученной последовательностью производится операция циклического сдвига вправо или влево. Если производится сдвиг влево, у числа самая первая (левая) цифра переставляется в конец, а если сдвиг производится вправо, то последняя (правая) цифра переставляется в начало.

Для построения нейронной сети представим в четырехпозиционном двоичном коде числа от 0 до 15. Далее, следует определить количество входов и выходов нейронной сети, необходимой для решения поставленной задачи. Очевидно, что для определения направления сдвига на входы нейронной сети необходимо представить исходную четырехпозиционную двоичную последовательность и четырехпозиционную двоичную последовательность, которая получилась в результате сдвига.

Выходной слой нейронной сети может состоять из одного нейрона. Его значение будет равно 0, если сдвиг произведен влево, и 1 – если сдвиг произведен вправо. Таким образом, для решения данной двухклассной задачи необходима нейронная сеть с *восемью* входами и *одним* выходом.

### Обучение нейронной сети

Определив количество входов в сети, приступим к созданию набора обучающих пар для обучения сети. Для этого выберем шесть четырехпозиционных двоичных кодов и выполним с ними операции сдвига вправо и влево (табл. 2).

**Таблица 2. Результаты сдвига влево и вправо четырехпозиционного кода**

Число	Двоичный код	Сдвиг влево	Сдвиг вправо
1	0001	0010	1000
2	0010	0100	0001
3	0011	0110	1001
4	0100	1000	0010
5	0101	1010	1010
6	0110	1100	0011
7	0111	1110	1011
8	1000	0001	0100
9	1001	0011	1100

Следует заметить, что в качестве представительских выборок нельзя выбирать числа с двоичным представлением **0000, 1111, 1010, 0101**, поскольку в независимости от направления сдвига (влево или вправо) для этих чисел будет получен один и тот же результат.

После подготовки набора представительских выборок (обучающих пар) и обучения, следует протестировать получившуюся нейронную сеть (проверить качество ее обучения). Тестирование проводится на оставшихся трех четырехпозиционных двоичных кодах табл. 2, которые не вошли в набор представительских выборок, использованный при обучении.

## ЛАБОРАТОРНАЯ РАБОТА № 7. РАСПОЗНАВАНИЕ СИМВОЛОВ

**Цель** – разработать и исследовать нейронную сеть обратного распространения, предназначенную для распознавания образов.

### Задание

1. Построить и обучить нейронную сеть, которая могла бы решать задачу распознавания символов.
2. Произвести тестирование нейронной сети при добавлении шума.

### Описание работы

На качество решения поставленной задачи в сильной степени влияют ограничения, которые накладываются производителями на демонстрационные версии своих программных продуктов. Так, в демо-версии программного продукта *TRAJAN* количество нейронов в слое не может превышать 9, поэтому при распознавании символов будем оперировать матрицей  $3 \times 3$ .

### Определение структуры нейронной сети

Представим в виде матрицы  $3 \times 3$  четыре латинские буквы *X*, *Y*, *L*, *I* и обучим нейронную сеть распознавать их матричное представление (см. табл. 3).

Таблица 3. Матричное представление для букв *X*, *Y*, *I* и *L*

<i>X</i>			<i>Y</i>			<i>I</i>			<i>L</i>		
1	0	1	1	0	1	0	1	0	1	0	0
0	1	0	0	1	0	0	1	0	1	0	0
1	0	1	0	1	0	0	1	0	1	1	1

В соответствии с табл. 3 входной сигнал для нейронной сети может быть представлен в виде развернутого раstra – вектора длиной 9. Например, для буквы *X* это:

**101010101**

Теперь определимся с выходами нейронной сети. Очевидно, что для распознавания образов нейронная сеть должна иметь возможность формировать столько выходных сигналов, сколько образов она должна уметь распознавать.

В нашем случае таких образов четыре, поэтому возможны два варианта представления выходных данных нейронной сети:

- выходной слой с двумя нейронами (выходами), т.е. каждому символу ставится в соответствие двухпозиционный двоичный код;
- выходной слой с четырьмя нейронами (выходами), т.е. каждому символу свой выход.

Предлагается выбрать любой вариант.

### Обучение нейронной сети

Набор обучающих пар, используемых для обучения нейронной сети, составляется с учетом того, какой вариант формирования выходного слоя выбран в предыдущем разделе. Если выбран вариант с двумя выходами – каждой букве ставится в соответствие двухпозиционный двоичный код, то выходной слой выглядит следующим образом:

$X - 00$                        $Y - 01$                        $I - 10$                        $L - 11$

Если выбран вариант с четырьмя выходами, то выходной слой такой:

$X - 0001$                        $Y - 0010$                        $I - 0100$                        $L - 1000$

После того, как набор представительских выборок (обучающих пар) создан, необходимо обучить нейронную сеть и проверить, насколько корректно она решает поставленную задачу.

### Проверка работы нейронной сети

После качественного обучения нейронной сети, следует внести в исходные данные некоторый шум (хотя это сделать непросто, так как в матрицу  $3 \times 3$  очень трудно добавлять шум).

Например, вместо раstra буквы **I** –

**010010010**

попробуйте подать

**010110010**

и посмотреть: удастся ли нейронной сети распознать символ, несмотря на внесенные в данные шум.



**Цель** – разработать и исследовать ИНС обратного распространения для искусственного носа, предназначенного для химического анализа воздушной среды .

### **Задание**

1. Исследовать и проанализировать имеющиеся экспериментальные данные (табл. 4), и определить количество вводов и выводов, требуемых для полносвязанной ИНС обратного распространения.
2. Создать и обучить нейронную сеть, которая будет способна указывать наличие определенных примесей в воздухе при анализе показаний химических датчиков.
3. Обучить нейронную сеть, расшив количество представительских выборок (обучающих пар), применяемых для обучения ИНС (табл. 5).
4. Определить оптимальную структуру нейронной сети с точки зрения минимизации среднеквадратической ошибки обучения.
5. Обучить ИНС, изменив параметры алгоритма обратного распространения.
6. Для пп. 1–3 построить графические зависимости среднеквадратической ошибки обучения от количества нейронов, используемых в скрытых слоях, и от количества итераций, используемых для обучения.
7. Сравнить результаты обучения в пп. 1–3.
8. Ответить на вопросы лабораторной работы.

### **Определение количества вводов и выводов нейронной сети**

Сформируйте представительские выборки (обучающие пары) для проведения обучения.

Для этого используйте следующие рекомендации.

1. Количество входов ИНС должно соответствовать количеству химических датчиков (рис. 1).

2. Существует два основных метода кодировки выхода нейронной сети. Первый – это использование бинарного вектора: для каждой примеси только один выход принимает значение 1. В этом случае количество выходов равно количеству примесей, определяемых системой. В другом случае все номинальные добавки пронумерованы и их числа перенесены в бинарную систему. В данной работе используйте первый метод.

3. Исходные данные:

- начальные экспериментальные данные, в виде показаний химических сенсоров, представлены в табл. 4;
- рекомендуемое число входов сети 11;
- рекомендуемое число выходов 6;

- рекомендуемое начальное число нейронов скрытого (внутреннего) слоя 4.
3. Вторичные экспериментальные данные, в виде показаний химических сенсоров, представлены в табл. 5: набор представительских выборок включает шесть обучающих пар, которые, естественно, подготовлены в формате первой обучающей пары (табл. 4).

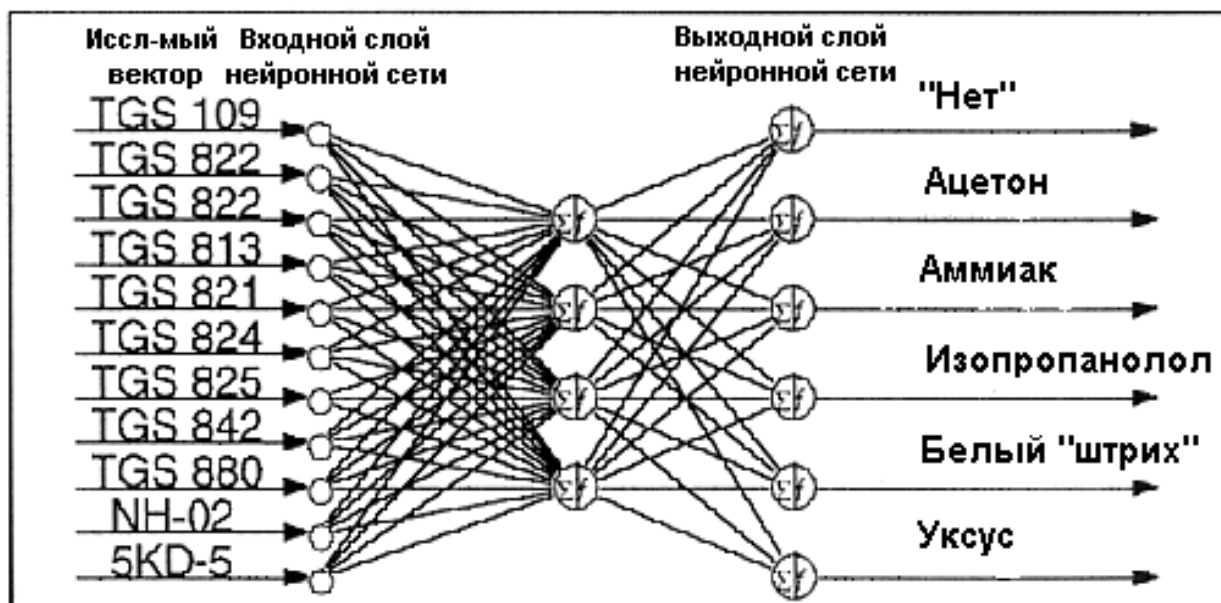


Рис. 1. Топология нейронной сети прототипа искусственного носа

Таблица 4. Первичная обучающая пара

Исследуемый вектор, составленный по показаниям сенсоров и формирующий вход нейронной сети	Выходы нейронной сети
{1; 0,05; 0,1; 0,3; 0,07; 0,08; 0,2; 0,05; 0,2; 0,6; 0,8}	{0, 0, 0, 0, 0, 1}

Таблица 5. Набор вторичных обучающих пар

Исследуемые векторы, составленные по показаниям сенсоров и формирующие вход нейронной сети	Выходной вектор нейронной сети
{1; 0,05; 0,1; 0,3; 0,07; 0,08; 0,2; 0,05; 0,2; 0,6; 0,8}	«Нет»
{0,8; 0,4; 0,7; 0,6; 0,1; 0,5; 1,0; 0,75; 0,5; 0,7; 0,8}	Ацетон
{0,9; 0,2; 0,4; 0,5; 0,1; 0,7; 0,6; 0,5; 0,5; 0,7; 0,8}	Аммиак
{0,85; 0,7; 0,8; 0,65; 0,1; 0,4; 1,0; 0,7; 0,4; 0,6; 0,7}	Изопропанол
{0,9; 0,3; 0,3; 0,4; 0,04; 0,1; 0,5; 0,3; 0,2; 0,7; 0,8}	Белый «штрих»
{0,95; 0,18; 0,21; 0,3; 0,05; 0,1; 0,3; 0,2; 0,2; 0,5; 0,7}	Уксус

## **Определение топологии нейронной сети**

Практические исследования показывают, что одного внутреннего (скрытого) слоя достаточно.

## **Обучение нейронной сети**

Следует проводить при помощи алгоритма обратного распространения. Используйте нейросетевой симулятор *TRAJAN* или любой другой доступный программный продукт для моделирования нейронных сетей, который поддерживает данный алгоритм обучения.

Как результат, получите надежную и быстродействующую нейронную сеть искусственного носа, применяемого для контроля атмосферы в воздушной среде промышленных, офисных и домашних помещениях и в других задачах.

## **Вопросы для проверки**

1. Назовите основное свойство многослойных нейронных сетей прямого распространения.
2. Какие существуют модификации алгоритма обратного распространения?
3. Назовите распространенные направления применения искусственного носа.

## ЛАБОРАТОРНАЯ РАБОТА № 8. ПРОГНОЗИРОВАНИЕ

**Цель** – разработать и исследовать нейронную сеть обратного распространения, предназначенную для прогнозирования временных серий, а также для анализа качества генератора случайных чисел.

### Задание

1. Создать и обучить нейронную сеть, предназначенную для анализа временных серий заданной размерности и отражающую структуру данных серий.
2. Осуществить прогноз значений будущих элементов временных серий.

**Таблица 5. Числа в диапазоне от 0 до 10, полученные с использованием генератора случайных чисел Турбо-паскаль**

Номера чисел	Числа, полученные с использованием функции Random(10) Турбо-паскаль									
1 – 10	0	0	8	2	2	6	3	1	3	4
11 – 20	0	4	0	8	0	2	9	3	7	3
21 – 30	6	8	7	3	1	3	4	2	8	2
31 – 40	4	1	8	2	7	9	4	8	8	0
41 – 50	1	1	5	0	5	0	7	6	7	7
51 – 60	5	2	6	5	9	6	9	2	6	2
61 – 70	0	7	4	8	5	5	9	6	3	0
71 – 80	7	9	7	7	1	1	9	7	5	8
81 – 90	6	2	0	6	2	8	1	2	5	9
91 – 100	1	1	2	1	5	4	2	1	6	7

### Исследуемые временные серии

Объектом исследования являются временные серии, полученные с помощью генератора случайных чисел, формирующего равномерно распределенные числовые значения.

За основу рекомендуется взять сто элементов (чисел) временной серии, значения которых лежат в диапазоне от 0 до 9 и сто элементов – в диапазоне от 0 до 99. Данные временные серии следует получить с помощью функции генерации случайных чисел любого программного продукта для математического моделирования (*MathCAD*, *Matlab* или др.), среды программирования на языке высокого уровня (*Delphi*, *C Builder* или др.) или взять из табл. 5 и 6.

Выясним, есть ли какая-либо закономерность в появлении элементов данных временных серий и тем самым определим качество генератора случайных чисел, который по идее должен обладать свойством некоррелированности значений числовых последовательностей.

**Таблица 6. Числа в диапазоне от 0 до 100, полученные с использованием генератора случайных чисел Турбо-паскаль**

Номера чисел	Числа, полученные с использованием функции Random(100) Турбо-паскаль									
<b>1 – 10</b>	0	3	86	20	27	67	31	16	37	42
<b>11 – 20</b>	8	47	7	84	5	29	91	36	77	32
<b>21 – 30</b>	69	84	71	30	16	32	46	24	82	27
<b>31 – 40</b>	48	14	87	28	77	97	49	88	82	2
<b>41 – 50</b>	14	14	50	2	59	0	77	65	77	70
<b>51 – 60</b>	55	20	68	59	95	64	99	24	67	29
<b>61 – 70</b>	8	77	49	88	50	57	95	68	33	0
<b>71 – 80</b>	70	98	77	74	19	14	91	78	58	86
<b>81 – 90</b>	68	28	9	62	28	87	16	27	54	96
<b>91 – 100</b>	17	15	26	17	57	49	28	15	60	73

### **Определение начальной структуры нейронной сети**

Чтобы синтезировать оптимальную структуру, необходимо подготовить и обучить несколько нейронных сетей и проверить качество выполнения требуемых операций. Ниже приведем основные этапы синтеза такой структуры.

1. В качестве нейронной сети предлагается воспользоваться сетью типа многослойный персептрон.
2. Количество входов соответствует количеству выбираемых элементов из временных серий (в нашем случае –  $N$ ).
3. Число нейронов во внутренних (скрытых) слоях и число таких слоев зависит от сложности задачи анализа или прогнозирования временных серий.
4. Выходной слой нейронной сети следует составить из одного нейрона, значение которого будет соответствовать прогнозируемому элементу временных серий.

### **Обучение нейронной сети**

Для обучения нейронной сети подготовим блок обучающих выборок следующим образом.

Выберем кадр из  $N+1$  числа элементов, идущих от конца к началу временной серии, где первые  $N$  элементов – формируют вектор входного слоя, а последний  $(N+1)$ -й – элемент выходного слоя нейронной сети. Следующую обучающую выборку получаем, передвигаясь окном на один элемент влево и т.д.

Таким образом, имея временную серию, состоящую из 100 элементов, можно подготовить  $100-N-1$  обучающих пар.

### **Оптимизация структуры нейронной сети**

Чтобы оценить качество прогнозирования, получите в соответствии с разделом «Исследуемые временные серии» еще 3 элемента, следующих за 100 элементами временной серии, которые уже использовались при обучении. Далее, запустите нейронную сеть, предъявив ей 3 новых исследуемых вектора, и сравните результаты нейронной сети с числами, синтезированными генератором.

Разницу между значениями, синтезированными генератором, и результатами работы нейронной сети используйте как качественный показатель сети. Уменьшите эту разницу, модифицируя структуру нейронной сети.

### **Замечания**

Подход, используемый в данной лабораторной работе для проверки качества генератора случайных чисел, наверное, не самый популярный и эффективный.

В то же время, разработанная в лабораторной работе нейронная сеть является аналогичной нейронным сетям, применяемым в интеллектуальных системах прогнозирования различного назначения, например систем прогноза знаков изменения биржевых индексов, систем прогноза цены, систем расчета оптимального использования ресурсов и т.п.

## ***ВОПРОСЫ ДЛЯ САМОПРОВЕРКИ***

1. На каких направлениях и для решения каких типовых задач применяются искусственные нейронные сети?
2. Дайте определение искусственного нейрона и перечислите его основные свойства.
3. Перечислите и поясните применяемые виды активационных функций.
4. Какое основное отличие искусственных нейронов, которые используются для построения нейронных сетей, получивших название персептронов?
5. К какому типу алгоритмов обучения относится алгоритм обратного распространения, и в чем отличительная черта этих алгоритмов.
6. Дайте свое определение «многослойному персептрону».
7. В чем заключается задача классификации?
8. Почему, по Вашему мнению, наиболее распространенной топологией сети является модель обратного распространения информации?