

МОДЕЛИРОВАНИЕ
ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ
В СИСТЕМЕ MATLAB

*Методические указания
к выполнению лабораторных работ
для студентов 3-го курса, обучающихся по специальностям
222000 «Инноватика»,
080500 «Бизнес-информатика»*

Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Воронежский государственный архитектурно-строительный университет»

Кафедра инноватики и строительной физики

**МОДЕЛИРОВАНИЕ
ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ
В СИСТЕМЕ MATLAB**

*Методические указания
к выполнению лабораторных работ
для студентов 3-го курса, обучающихся по специальностям
222000 «Инноватика»,
080500 «Бизнес-информатика»*

Воронеж 2014

УДК 004.94

Составители П.А. Головинский, Е.А. Михин

Моделирование искусственных нейронных сетей в системе MATLAB: метод. указания к выполнению лаб. работ для студ. спец. 222000, 080500 / Воронежский ГАСУ; сост.: П.А. Головинский, Е.А. Михин.–Воронеж, 2014.-45 с.

Содержат краткий теоретический материал, практические задания по вариантам, методики и рекомендации для проведения расчетов с использованием искусственных нейронных сетей.

Предназначены для выполнения лабораторных работ по дисциплинам: «Методы и программные системы анализа данных», «Анализ данных», «Нечёткая логика и нейронные сети», для студентов дневного обучения специальностей 222000 «Инноватика», 080500 «Бизнес-информатика».

Ил. 13. Табл. 1. Библиогр.: 9 назв.

УДК 004.94

*Печатается по решению научно-методического совета
Воронежского ГАСУ*

*Рецензент – С.Н. Дьяконова, к.т.н., доц. кафедры инноватики
и строительной физики Воронежского ГАСУ*

ВВЕДЕНИЕ

В ряде методов, созданных для реализации искусственного интеллекта, используются явные представления знаний и тщательно спроектированные алгоритмы перебора. Отличный от этого подход состоит в построении интеллектуальных программ с использованием моделей, имитирующих нейронные структуры в человеческом мозге или эволюцию разных альтернативных конфигураций, как это делается в генетических алгоритмах и искусственной жизни. Для того чтобы решать сложные и плохо формализуемые задачи и возникло направление, которое называется искусственные нейронные сети.

Искусственные нейронные сети состоят из нейроноподобных элементов, соединенных между собой в сеть. Существуют статические и динамические нейронные сети. В статических нейронных сетях изменение параметров системы происходит по некоторому алгоритму в процессе обучения. После обучения параметры сети не меняются. В динамических нейронных сетях отображение внешней информации, и ее обработка осуществляется в виде некоторого динамического процесса, то есть процесса, зависящего от времени.

Нейронные сети нашли применение практически во всех областях науки и техники. С использованием нейронных сетей успешно решаются многие проблемы бизнеса и финансов. Задачи управления, классификации, распознавания образов, прогнозирования, присущие практически всем прикладным областям, таким как медицина, военное дело, авиация и космос, и строительство, все чаще решаются с применением нейросетевых технологий.

В связи с этим необходимо дать студентам, обучающимся современным информационным технологиям, фундаментальное представление об основных понятиях и моделях нейронных сетей, а также научить применять эти знания на практике.

Данные лабораторные работы направлены на вырабатывание практических навыков построения искусственных нейронных сетей в среде MatLab. Особенностью выполнения работ является то, что студент приучается к самостоятельному принятию решений, развиваются его исследовательские способности, что особенно важно в динамично развивающемся мире. В настоящих методических указаниях рассматриваются нейронные сети прямой передачи сигнала, радиальные базисные сети, сеть Кохонена и сеть Хопфилда, являющиеся типовыми технологиями искусственных нейронных сетей.

Лабораторная работа № 1

Аппроксимация функции одной переменной

1.1. Цель работы

На примере задачи аппроксимации функции одной переменной научиться работать с встроенной в MATLAB функцией `newff()` [1], позволяющей формировать нейронную сеть с прямой передачей сигнала. Освоить на практике основные операции обучения нейронной сети.

1.2. Краткие теоретические сведения

В лабораторной работе рассматривается нейронная сеть с прямой передачей сигнала (с прямой связью) [2], то есть сеть, в которой сигналы передаются только в направлении от входного слоя к выходному, и элементы одного слоя связаны со всеми элементами следующего слоя. Важнейшим для реализации нейронных сетей является определение алгоритма обучения сети.

В настоящее время одним из самых эффективных и обоснованных алгоритмов обучения нейронных сетей является *алгоритм обратного распространения ошибок*, который применим к *однонаправленным многослойным сетям*. В основе данного алгоритма используется хорошо разработанный метод градиентного спуска, позволяющий эффективно отыскивать экстремумы функций многих переменных. В многослойных нейронных сетях имеется множество скрытых нейронов, входы и выходы которых не являются входами и выходами нейронной сети, а соединяют нейроны внутри сети, то есть *скрытые нейроны*. Занумеруем выходы нейронной сети индексом $j=1,2,\dots,m$ (отметим здесь, что в общем случае количество выходов нейронной сети не совпадает с количеством её входов). Пусть имеется M_0 обучающих примеров, занумеруем их индексом $M=1,2,\dots,M_0$. Тогда в качестве целевой функции можно выбрать функцию ошибки как сумму квадратов расстояний между реальными выходными состояниями y_M нейронной сети, и правильными значениями функции $d_{j,M}$, соответствующими этим примерам. Введем расстояния согласно евклидовой метрике, определив норму

$$\|y - d\| = \sqrt{(y - d, y - d)}. \quad (1)$$

Согласно указанной метрике (1), запишем целевую функцию в виде

$$E(w) = \frac{1}{2} \sum_{j,M} (y_{j,M}(w) - d_{j,M})^2. \quad (2)$$

Коэффициент $1/2$ в (2) выбран из соображений более короткой записи последующих формул. Матрица w состоит из весовых коэффициентов нейронной сети, за выяснением сути этих коэффициентов отправляем читателя в [2]. Зада-

ча обучения нейронной сети состоит в том, чтобы найти такие коэффициенты w_{jk} , при которых достигается минимум функции $E(w)$, $E \geq 0$.

На рис. 1 представлено схематическое изображение архитектуры двухслойной нейронной сети с прямой передачей сигнала, формируемое в MATLAB.

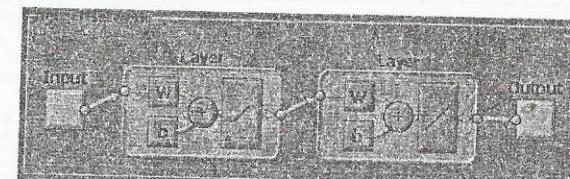


Рис. 1. Архитектура нейронной сети с прямой передачей сигнала

Словами *Input* и *Output* обозначены вход и выход нейронной сети; символы W , b обозначают матрицу весовых коэффициентов и вектор смещений для каждого слоя (*Layer*) нейронной сети. На рис. 1 представлено также схематическое изображение активационной функции нейронов в каждом слое. В этой лабораторной работе в качестве активационных функций нейронов будут использоваться гиперболическая тангенциальная (*tansig*) и линейная (*purelin*) функции.

Веса и смещения определяются с помощью алгоритма обратного распространения ошибок [3].

Обучение сети в алгоритме обратного распространения ошибок можно условно разбить на следующие операции:

1. Выбор обучающей пары (вход – соответствующий точный выход) из множества обучающих примеров.
2. Расчет выхода нейронной сети. В начале обучения выход нейронной сети сильно отличается от соответствующего точного выхода из множества обучающих примеров.
3. Вычисление разности между выходом сети и соответствующим точным выходом.
4. Корректировка весовых коэффициентов сети с целью минимизации целевой функции (1) согласно алгоритму обратного распространения ошибок.
5. Шаги 1–4 следует повторять для каждого вектора обучающего множества до тех пор, пока ошибка (1) на всем множестве не достигнет приемлемого уровня.

1.3. Пример решения типовой задачи

Пусть имеется произвольная функция $y = f(x)$, заданная в виде таблицы состоящей из координат конечного числа точек этой функции. В данном примере будем использовать функцию, изображенную на рис. 2. Необходимо по

указанной таблице сформировать и обучить нейронную сеть, наиболее точно воспроизводящую исходную функцию.

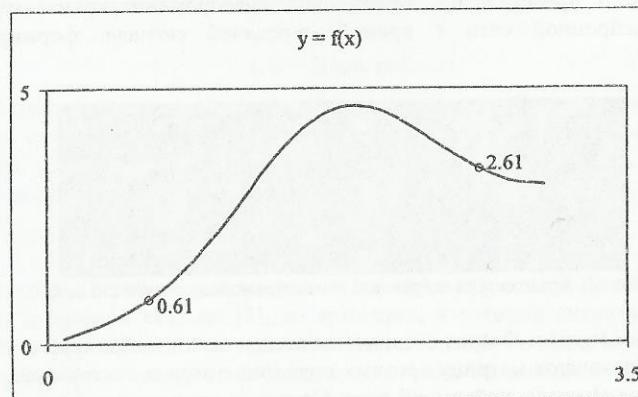


Рис. 2. Произвольная функция одной переменной.

Для решения поставленной задачи с помощью MATLAB используется код представленный в лист. 1:

Лист. 1. Апроксимация функции одной переменной с использованием нейронной сети.

```

x = [0.10 0.31 0.51 0.72 0.93 1.14 1.34 1.55 1.76 1.96 2.17 2.38 2.59 2.79
3.00];
%абсциссы аппроксимируемой функции
y = [0.1010 0.3365 0.6551 1.1159 1.7632 2.5847 3.4686 4.2115 4.6152 4.6095
4.2887 3.8349 3.4160 3.1388 3.0603];
%ординаты аппроксимируемой функции
%вектора x и у в данном примере представляют собой множество обучающих
%примеров для нейронной сети

net = newff([0 3],[5,1],{'tansig', 'purelin'},'trainbfg');
%функция формирующая нейронную сеть

%первый параметр функции задает минимальное и максимальное значение на
%выходе нейронной сети [min max]

%второй параметр ответственен за структуру сети, в данном случае первый
%слой сети имеет 5 нейронов, второй слой - 1

%в третьем параметре в фигурных скобках указываются активационные функции
%нейронов каждого слоя нейронной сети

%четвертый параметр задает алгоритм обучения нейронной сети в данном случае
%это алгоритм обратного распространения ошибок

net.trainParam.epochs = 300;
%максимальное количество эпох обучения

net.trainParam.goal = 1.37e-4;

```

%приемлемое значение ошибки (1). при достижении данного значения обучение
%сети будет остановлено

```

[net,tr] = train(net,x,y);
%функция производит обучение нейронной сети net на векторах x и y.
%в переменную tr возвращается результат обучения

x1 = 0:0.3:3.5;
%задаем новый вектор абсцисс
an = sim(net,x1);
%на векторе x1 вычисляем значения, сформированные обученной нейронной сетью
%если сеть обучена правильно, то полученные значения должны быть близки к
%значениям аппроксимируемой функции

%для визуализации расхождения аппроксимируемой функции и функции
%формируемой нейронной сетью представим их на одном графике, для этого
%запишем:
plot(x,y,'+r',x1,an,'-g');
hold on;

xx = [0.61 2.61];
%зададим еще один вектор абсцисс

v = sim(net,xx);
%вычислим значения функции формируемой нейронной сетью на векторе xx и
%выведем эти значения для их сравнения с точными значениями

%добавим к графику точки задаваемые векторами xx и v
plot(xx,v,'ob','MarkerSize',5,'LineWidth',2)

```

В языке MATLAB знаком "%" обозначаются комментарии – кусочки кода, которые не участвуют в вычислениях, а служат только для их пояснения. Результатом выполнения кода из лист. 1 будет график, изображенный на рис. 3.

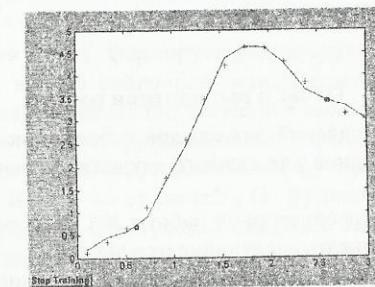


Рис. 3. Результаты аппроксимации функции нейронной сетью: "+" – аппроксимируемая функция; сплошная линия и точки "o" – результаты аппроксимации нейронной сетью на отрезке [0, 3] и в контрольных точках соответственно.

В массиве v лист. 1 содержатся приближенные значения для двух контрольных точек, указанных на рис. 2 $xx = [0.61 \ 2.61]$. В нашем примере:

$$v = 0.8417 \quad 3.3789.$$

Сравнив эти приближённые значения с точными значениями [0.85 3.37], можно сделать вывод о корректности построения нейронной сети.

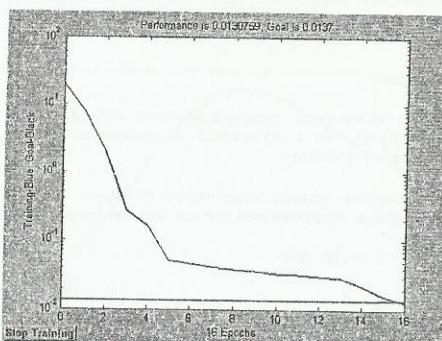


Рис. 4. Величина ошибки обучения нейронной сети (1) в зависимости от эпохи обучения.

Рис. 4 демонстрирует динамику изменения ошибки обучения нейронной сети в зависимости от эпохи обучения. Данная зависимость может быть использована для сравнения эффективностей работы различных алгоритмов обучения, а также для подбора параметров алгоритмов обучения.

Отметим также, что при повторном прогоне кода из лист 1. скорее всего мы получим несколько иной результат. Связано это с тем, что в используемом алгоритме обучения обучающие пары из множества обучающих примеров выбираются случайным образом.

1.4. Отчёт о выполнении работы

Различные варианты заданий для каждой лабораторной работы указаны в Приложении. Вариант задания для каждого студента назначается преподавателем.

Отчёт о выполнении лабораторной работы №1 должен быть выполнен на листах формата А4 и содержать следующие результаты:

1. Таблицу с точками аппроксимируемой функции, взятой из Приложения. Таблица должна содержать не менее 15 точек;
2. Код программы с подробными комментариями;
3. Результаты моделирования (рис. 3);
4. Динамику изменения величины ошибки обучения (рис. 4);
5. Краткие выводы о результатах работы. Комментарий, касающийся точности аппроксимации.

Лабораторная работа № 2

Аппроксимация функции двух переменных

1.1. Цель работы

Научиться строить радиальную базисную сеть в MATLAB, с помощью функций `newrbe()` и `newrb()`. Применить полученные знания для решения задачи аппроксимации функции двух переменных.

1.2. Краткие теоретические сведения

Радиальные базисные сети предназначены для аппроксимации функций с помощью пространственно локализованных функций, называемых в данном контексте радиально базисными функциями. Возьмем произвольную непрерывную функцию и представим ее в виде суммы колоколообразных функций, отличных от нуля лишь на небольшом отрезке. Аналитически это означает представление $f(x)$ в виде разложения

$$f(x) = \sum_i w_i \phi\|x - c_i\|, \quad (1)$$

где w_i - вес суммирования соответствующей пространственно локализованной функции, а c_i - центр отрезка на котором эта функция отлична от нуля. Формула (1) лежит в основе радиальной базисной нейронной сети. Расстояние $\|x - c_i\|$ в (1) определяется как расстояние в евклидовом пространстве:

$$\|x - c_i\| = \sqrt{(x_1 - c_{i1})^2 + (x_2 - c_{i2})^2 + \dots} \quad (2)$$

Функция `newrbe()` формирует радиальную базисную сеть с нулевой ошибкой, т.е. для входов нейронной сети, взятых из множества обучающих примеров будут получаться соответствующие выходы из этого множества. Сеть с радиальными базисными функциями как правило состоит из трех слоев: входного; скрытого с радиально базисными функциями; выходного с линейными функциями. В отличие от `newrbe()`, функция `newrb()` формирует радиальную базисную сеть с ненулевой ошибкой. На рис. 5 показана типичная архитектура радиальной базисной сети.

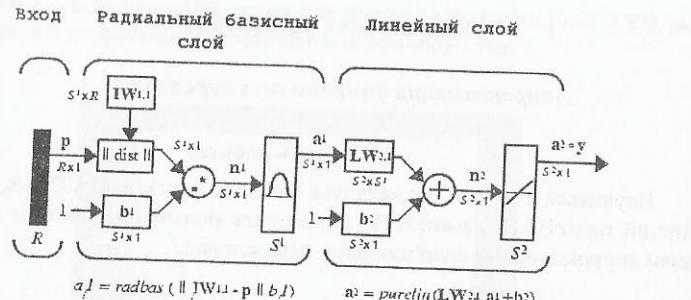


Рис. 5. Архитектура типичной радиальной базисной сети

1.3. Пример решения типовой задачи

Рассмотрим задачу аппроксимации функции двух переменных $z = e^{-x^2} \cdot e^{-y^2}$ заданной в координатной области $\{x \in [-1, 1], y \in [-1.5, 1.5]\}$, с помощью радиальной базисной нейронной сети. Для этого аппроксимируемую функцию необходимо представить в табличном виде, т.е. разбить интересующую нас координатную область на множество подобластей и в каждой из этих подобластей вычислить значение аппроксимируемой функции. Количество точек разбиений по оси x обозначим как nx , а по оси y - ny . Код из лист. 2 решает поставленную задачу.

Лист. 2. Аппроксимация функции двух переменных с помощью радиальной базисной нейронной сети

```
x1 = -1.0; x2 = +1.0; y1 = -1.5; y2 = +1.5;
%задали границы интересующей нас области
```

```
nx = 7; ny = 9;
%эти переменные определяют число разбиений заданной области по каждой из
%осей
```

```
step_x = (x2-x1)/(nx-1);
step_y = (y2-y1)/(ny-1);
%длины шагов на которые будет разбита каждая ось

step_min = min(step_x, step_y);
%минимальный шаг

[x,y] = meshgrid([x1:step_x:x2], [y1:step_y:y2]);
%сформировали координатную сетку, с минимальным шагом разбиения

z = exp(-x.^2).*exp(-y.^2);
%задали аппроксимируемую функцию, на сформированной сетке

surf(x,y,z), title('PS. Press<enter>');
pause;
```

```
%вывели график аппроксимируемой функции. после вывода этого графика
%выполнение кода приостановится до тех пор, пока пользователь не нажмет
%<enter>, о чем говорит кодовое слово "pause"
```

```
xx = reshape(x, 1, nx*ny);
yy = reshape(y, 1, nx*ny);
%превратили массивы x и y в одномерные
```

```
zz = exp(-xx.^2).*exp(-yy.^2);
p = [xx; yy];
t = zz;
goal = 0.0371;
%максимальная ошибка нейронной сети
```

```
spread = 1.0 * step_min;
%параметр влияния. чем больший диапазон входных значений должен быть принят во внимание, тем большее значение параметра влияния SPREAD
%должно быть установлено
```

```
net = newrb(p,t, goal, spread);
%генерируем нейронную сеть
net.layers(1).size
```

```
smlt = sim(net,p);
%Обучаем нейронную сеть
```

```
[zz' smlt']
%выводим в виде двух колонок точный результат и результат, полученный
%нейронной сетью (для сравнения)
```

```
smltr = reshape(smlt,ny,nx);
surf(x,y,smltr), title('AS. Press<enter>');
%выводим график аппроксимации исходной функции с помощью нейронной сети
```

В результате выполнения кода из лист. 2 будет выведен график исходной аппроксимируемой функции $z = e^{-x^2} \cdot e^{-y^2}$, см. рис. 6.

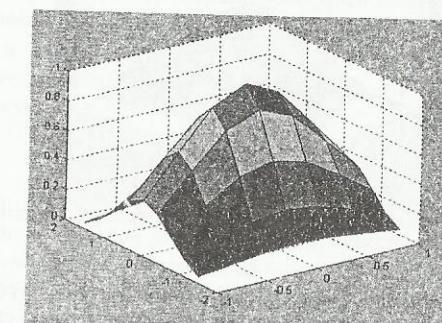


Рис. 6. График аппроксимируемой функции двух переменных.

Переменная `goal` в лист. 2 задает максимальное значение среднеквадратической ошибки, динамика изменения которой в зависимости от эпохи обучения представлена на рис. 7.

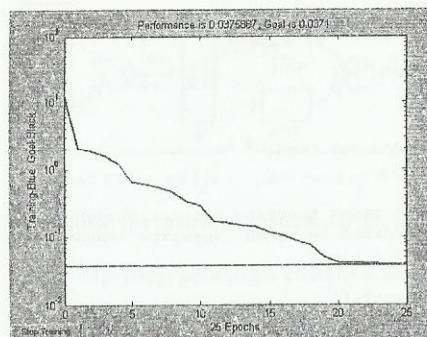


Рис. 7. Динамика изменения среднеквадратической ошибки сети на обучающем множестве в зависимости от эпохи обучения.

Варьируя параметр `goal`, а также параметр `spread` в лист. 2 можно добиться необходимой степени точности аппроксимации. Результат аппроксимации с помощью радиальной базисной нейронной сети представлен на рис. 8.

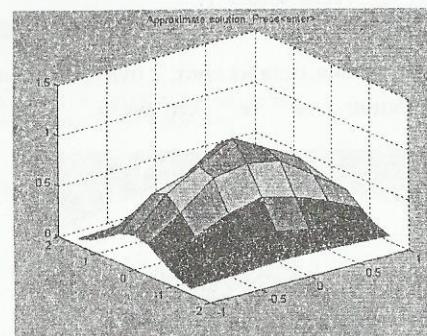


Рис. 8. Результат аппроксимации функции двух переменных радиальной базисной нейронной сетью.

Сравнение рис. 7 и рис. 8 позволяет визуально убедиться в точности результата, полученного при использовании нейронной сети. Для её количественной оценки может быть использован массив, состоящий из пар значение аппроксимируемой функции – соответствующее значение полученное с использованием нейронной сети, этот массив также формируется колом из лист .2.

1.4. Отчёт о выполнении работы

Отчёт о выполнении лабораторной работы №2 должен быть выполнен на листах формата А4 и содержать следующие результаты:

1. Аппроксимируемую функцию и координатную область, на которой необходимо произвести аппроксимацию. Эти данные выдаются преподавателем из Приложения;
2. Табличное и графическое представление аппроксимируемой функции;
3. Код программы с подробными комментариями;
4. Анализ сравнения результата, получаемого нейронной сетью с точным значением. Влияние параметров `goal` и `spread` на работу сети.
5. Выводы.

Лабораторная работа № 3

Решение задачи классификации с использованием самоорганизующейся нейронной сети Кохонена

1.1. Цель работы

Научиться строить самоорганизующуюся нейронную сеть Кохонена в MATLAB (с помощью встроенной функции `newc`). На примере решения простейшей задачи классификации, содержание которой изложено ниже ознакомиться с основными параметрами этой функции и выявить влияние этих параметров на качество получаемого результата.

1.2. Краткие теоретические сведения

Сети Кохонена, или самоорганизующиеся карты (*Kohonen maps*), предназначены для решения задач автоматической классификации, т.е. задач разбиения некоторого множества объектов на классы. При обучении сети Кохонена на её вход подается некоторый обучающий образ, однако соответствующий выход сети изначально не известен (не задано обучающее множество примеров), в этом состоит принципиальное отличие процесса обучения нейронной сети Кохонена от сетей рассмотренных в предыдущих работах. Такой механизм обучения получил название – обучение без учителя. Сеть Кохонена является двух-

слойной. Она содержит слой входных нейронов и собственно слой Кохонена. Слой Кохонена может быть одномерным, двумерным и трехмерным. В первом случае нейроны расположены в цепочку; во втором – они образуют двумерную сетку (обычно в форме квадрата или прямоугольника); в третьем – трехмерную систему. Определение весов нейронов слоя Кохонена основано на использовании алгоритмов автоматической классификации (кластеризации или самообучения).

Обозначим входные обучающие образы в виде набора векторов $x_i = (x_1, x_2, \dots, x_n)_i$, в результате решения задачи классификации необходимо этот набор векторов разбить на классы «подобных» векторов, причем число классов заранее неизвестно. На стадии обучения (точнее самообучения) сети входной вектор x_i попарно сравнивается со всеми векторами w_j всех нейронов сети Кохонена. Вводится некоторая функция близости или подобия – d (например, в виде евклидова расстояния, т.е. чем ближе пространственно вектора, тем они подобнее). Активный нейрон с номером c слоя Кохонена, для которого значение функции близости $d(x, w_c)$ минимально, объявляется «победителем». При этом образ, характеризующийся вектором x , будет отнесен к классу, который представляется нейроном-«победителем».

Рассмотрим алгоритм самообучения сетей Кохонена. Для функции близости $d = \|x - w\|$. Выигрывает нейрон с w_c определяемым из (1)

$$\|x - w_c\| = \min \|x - w_j\|. \quad (1)$$

Близость x' и w' можно переделить, пользуясь скалярным произведением, как

$$z_j = 1 - (x', w'_j) = 1 - \sum_{i=1}^n x'_i w'_{ij}. \quad (2)$$

На стадии самообучения сети Кохонена осуществляется коррекция весового вектора w_j не только нейрона-«победителя», но и весовых векторов остальных активных нейронов слоя Кохонена, однако в значительно меньшей степени. Чем больше расстояние между весовым вектором некоторого активного нейрона и весовым вектором нейрона-«победителя», тем меньше коррекция. При этом форма и величина области вокруг нейрона-«победителя», в которой происходит корректировка весовых векторов активных нейронов также корректируется в процессе обучения. Сначала начинают с очень большой области – она, в частности, может включать все нейроны слоя Кохонена, в процессе обучения эта область уменьшается. Изменение весовых векторов активных нейронов осуществляется по правилу

$$w_j(t+1) = w_j(t) + \eta(t) d_{cj}(t) [x(t) - w_j(t)], \quad j=1, 2, \dots, n, \quad (3)$$

где $w_j(t)$ – значение весового вектора на шаге t самообучения сети, $d_{cj}(t)$ – функция близости между нейронами слоя Кохонена и $\eta(t)$ – изменяемый во времени коэффициент шага коррекции. В качестве $\eta(t)$ обычно выбирается монотонно убывающая функция ($0 < \eta(t) < 1$), то есть алгоритм самообучения начи-

нается сравнительно большими шагами адаптации и заканчивается относительно малыми.

В результате n -мерное входное пространство R_n отобразится на m -мерную сетку (слой Кохонена). Это отображение реализуется в результате рекуррентной (итеративной) процедуры самообучения (*unsupervised learning*). Отличительная особенность этого отображения – формирование кластеров (*clusters*) или классов. По завершении процесса самообучения на стадии реального использования сети Кохонена неизвестные входные образы относятся к одному из выявленных кластеров (классов) по близости к некоторому весу, принадлежащему определенному кластеру, выявленному на стадии самообучения.

На рис. 9 показана архитектура слоя Кохонена.

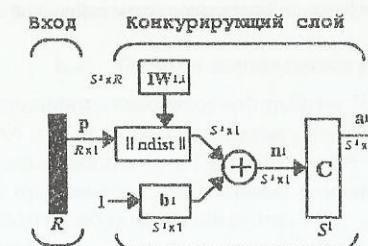


Рис. 9. Схема архитектуры слоя Кохонена

1.3. Пример решения типовой задачи

Пусть заданы 28 случайных двумерных векторов, изображенных на рис. 10 крестами. Оцифровав данный график, можно получить массив входных данных $P(1:)$, $P(2:)$ (табл.1).

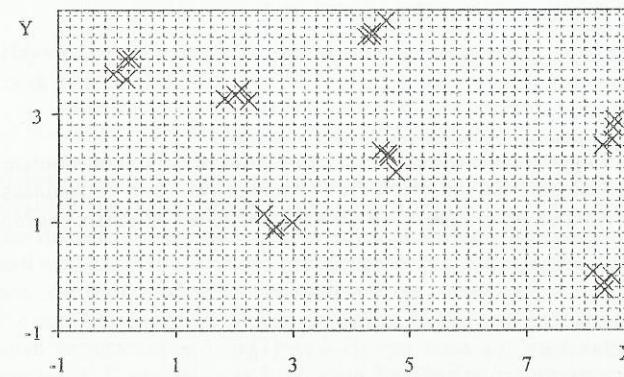


Рис. 10. Распределение входных векторов

Таблица 1

Массив входных данных

	P(1,:)	-0.1	4.3	4.6	8.5	1.8	8.3	2.5	0.2	4.6
P(2,:)	1.0	3.7	4.4	2.3	2.7	3.3	-0.1	1.2	3.6	4.7
P(1,:)	4.5	8.5	2.2	8.2	2.7	0.2	4.4	4.6	8.3	2.0
P(2,:)	2.3	2.9	3.2	0.1	0.8	4.0	4.5	2.2	2.4	3.3
P(1,:)	8.5	2.7	0.1	4.3	4.7	8.5	2.1	8.3		
P(2,:)	0.0	0.9	4.0	4.4	1.9	2.5	3.5	-0.3		

Код из лист. 3 демонстрирует процедуру обучения самоорганизующейся нейронной сети Кохонена.

Лист. 3. Решение задачи классификации с использованием нейронной сети Кохонена.

```

plot(P(1,:), P(2,:), '+m');
title('Input vectors');
xlabel('P(1,:)');
ylabel('P(2,:)');
hold on;
nclusters = 7; %число классов

a1 = -10;
a2 = +10;
b1 = -5;
b2 = +5;
%т.о. задали границы области в которой лежат входные вектора

net = newc([a1 a2; b1 b2], nclusters, 0.1, 0.0005);
%необходимо указать что означают третий и четвертый параметры функции

net.trainParam.epochs = 49;
net.trainParam.show = 7;
net = train(net,P);

w = net.IW{1}; %классы

plot(w(:,1),w(:,2), 'kp');

```

На рис. 11 представлены исходные данные (крести) и полученные центры кластеризации (звезды). Видно, что сеть Кохонена успешно справилась с задачей, центры кластеров расположены в местах скопления входных векторов.

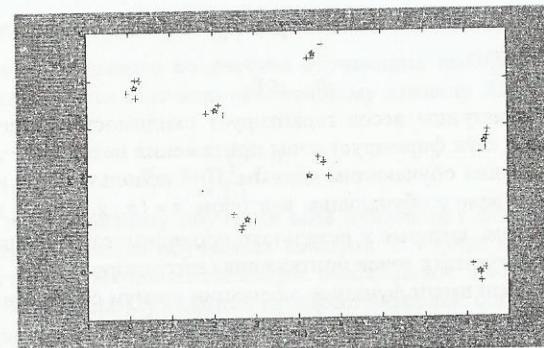


Рис. 11. Распределение входных линий (крести) и положение центров кластеризации (звезды)

1.4. Отчёт о выполнении работы

Отчёт о выполнении лабораторной работы №3 должен быть выполнен на листах формата А4 и содержать следующие результаты:

1. Исходные данные (см. Приложение);
2. Текст программы с подробными комментариями;
3. Результаты моделирования (рис. 11);
4. Выводы.

Лабораторная работа № 4

Сеть Хопфилда

1.1. Цель работы

Научиться работать с сетью Хопфилда `newhop()`, исследовать устойчивость сети и её сходимость.

1.2. Краткие теоретические сведения

Американский исследователь Хопфилд в 80-х годах 20-го века предложил специальный тип нейросетей. Названные в его честь сети Хопфилда являются рекуррентными или сетями с обратными связями и предназначены для распознавания образов. Обобщенная структура этой сети представляется, как правило, в виде системы с обратной связью выхода с входом.

В сети Хопфилда входные сигналы нейронов являются одновременно входными сигналами сети: $x_i(k)=y_i(k-1)$, при этом возбуждающий вектор особо не выделяется. В классической системе Хопфилда отсутствует связь нейрона с

собственным выходом, что соответствует $w_{ii} = 0$, а вся матрица весов является симметричной: $w_{ij} = w_{ji}$

$$\hat{W} = \hat{W}^T. \quad (1)$$

Симметричность матрицы весов гарантирует сходимость процесса обучения. Процесс обучения сети формирует зоны притяжения некоторых точек равновесия, соответствующих обучающим данным. При использовании ассоциативной памяти мы имеем дело с обучающим вектором $x = (x_1, x_2, \dots, x_n)$, либо с множеством этих векторов, которые в результате проводимого обучения определяют расположение конкретных точек притяжения (аттракторов).

Каждый нейрон имеет функцию активации сигнум со значениями ± 1 :

$$\operatorname{sgn} a = \begin{cases} 1, & a \geq 0 \\ -1, & a < 0 \end{cases}. \quad (2)$$

Это означает, что выходной сигнал i -го нейрона определяется функцией

$$y_i = \operatorname{sgn} \left(\sum_{j=1}^N w_{ij} x_j + b_i \right), \quad (3)$$

где N обозначает количество нейронов, $N=n$. Часто постоянная составляющая b_i , определяющая порог срабатывания отдельных нейронов, равна 0. Тогда циклическое прохождение сигнала в сети Хопфилда можно представить соотношением

$$y_i(k) = \operatorname{sgn} \left(\sum_{j=1}^N w_{ij} y_j(k-1) \right) \quad (4)$$

с начальным условием $y_j(0) = x_j$.

В процессе функционирования сети Хопфилда можно выделить два режима: обучения и классификации. В режиме обучения на основе известных обучающих выборок x_i подбираются весовые коэффициенты w_{ij} . В режиме классификации при зафиксированных значениях весов и вводе конкретного начального состояния нейронов $y(0) = x$ возникает переходный процесс, протекающий в соответствии с выражением (2) и заканчивающийся в одном из локальных устойчивых положений, задаваемом биполярным вектором со значениями $y_j = \pm 1$, для которого $y(k) = y(k-1)$.

Обучение не носит рекуррентного характера. Достаточно ввести значения (правило Хебба) весов, выразив их через проекции вектора точки притяжения эталонного образа:

$$w_{ij} = \frac{1}{N} x_i x_j, \quad (5)$$

В соответствии с этим правилом сеть дает правильный результат при входном примере, совпадающем с эталонным образом, поскольку

$$\sum_{j=1}^N w_{ij} x_j = \frac{1}{N} \sum_{j=1}^N x_i (x_j x_j) = x_i \sum_{j=1}^N 1 \cdot \frac{1}{N} = x_i, \quad (6)$$

так как вследствие биполярности значений элементов вектора x всегда $x_j^2 = 1$.

При вводе большого количества обучающих выборок $x^{(k)}$ для $k=1, 2, \dots, p$ веса w_{ij} подбираются согласно обобщенному правилу Хебба в соответствии с которым

$$w_{ij} = \frac{1}{N} \sum_{k=1}^p x_i^{(k)} x_j^{(k)}. \quad (7)$$

Благодаря такому режиму обучения веса принимают значения, определяемые усреднением множества обучаемых выборок. В случае множества обучаемых выборок актуальным становится вопрос о стабильности ассоциативной памяти.

Сеть Хопфилда [2] является автоассоциативной сетью (рис.12). Дискретная сеть Хопфилда имеет следующие характеристики: она содержит один слой элементов; каждый элемент связывается со всеми другими элементами, но не связан с самим собой; за один шаг работы обновляется только один элемент сети; элементы обновляются в случайном порядке; выход элемента ограничен значениями 0 или 1.

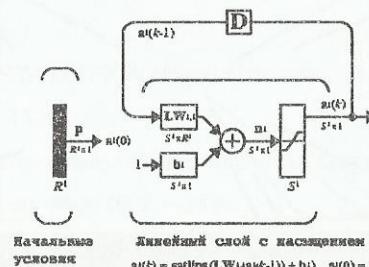


Рис. 12. Схема архитектуры модифицированной сети Хопфилда

1.3. Пример решения типовой задачи

Рассмотрим сеть Хопфилда [1] с четырьмя нейронами и определим четыре точки равновесия, см. лист 4.

Лист 4. Моделирование нейронной сети Хопфилда

```
T = [+1 -1; -1 +1; +1 +1; -1 -1];
T=T';
plot(T(1,:),T(2,:),'rh','MarkerSize',13), hold on;
axis([-1.1 1.1 -1.1 1.1]);
title('Hopfield Network State Space');
xlabel('a(1)');
ylabel('a(2)');
net = newhop(T);
[Y,Pf,Af] = sim(net, 4, [], T);
```

```

Y
Pf
Af
pause
color = 'rgbmy';
for i=1:25
    a = {rands(2,1)};
    [y,Pf,Af] = sim(net,[1 20],(),a);
    record=[cell2mat(a) cell2mat(y)];
    start=cell2mat(a);
    plot(start(1,1),start(2,1),'kx',record(1,:),
    record(2,:), color(mod(i,5)+1),'LineWidth',5)
end

```

На рис. 13 показано поведение сети при распознавании случайных начальных данных a .

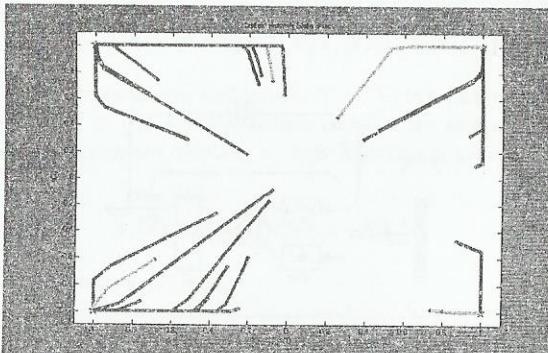


Рис. 13. Поведение сети Хопфилда при случайных начальных условиях a

1.4. Отчёт о выполнении работы

Отчёт о выполнении лабораторной работы №4 должен быть выполнен на листах формата А4 и содержать следующие результаты:

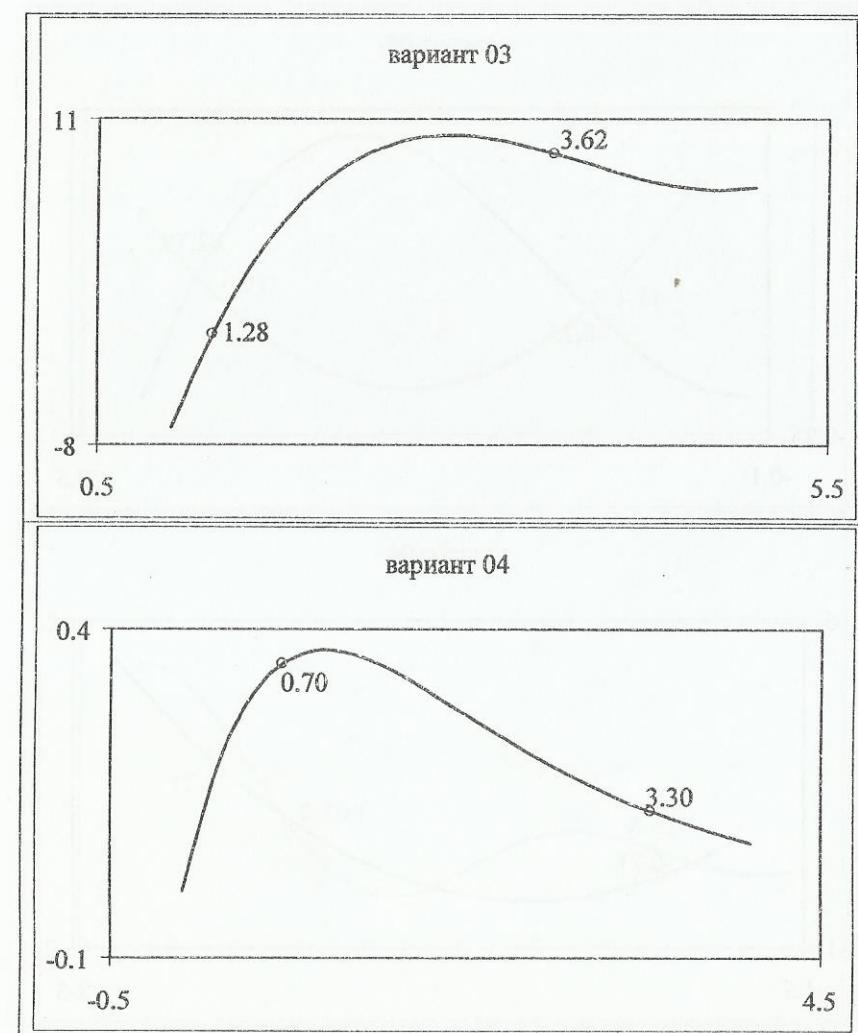
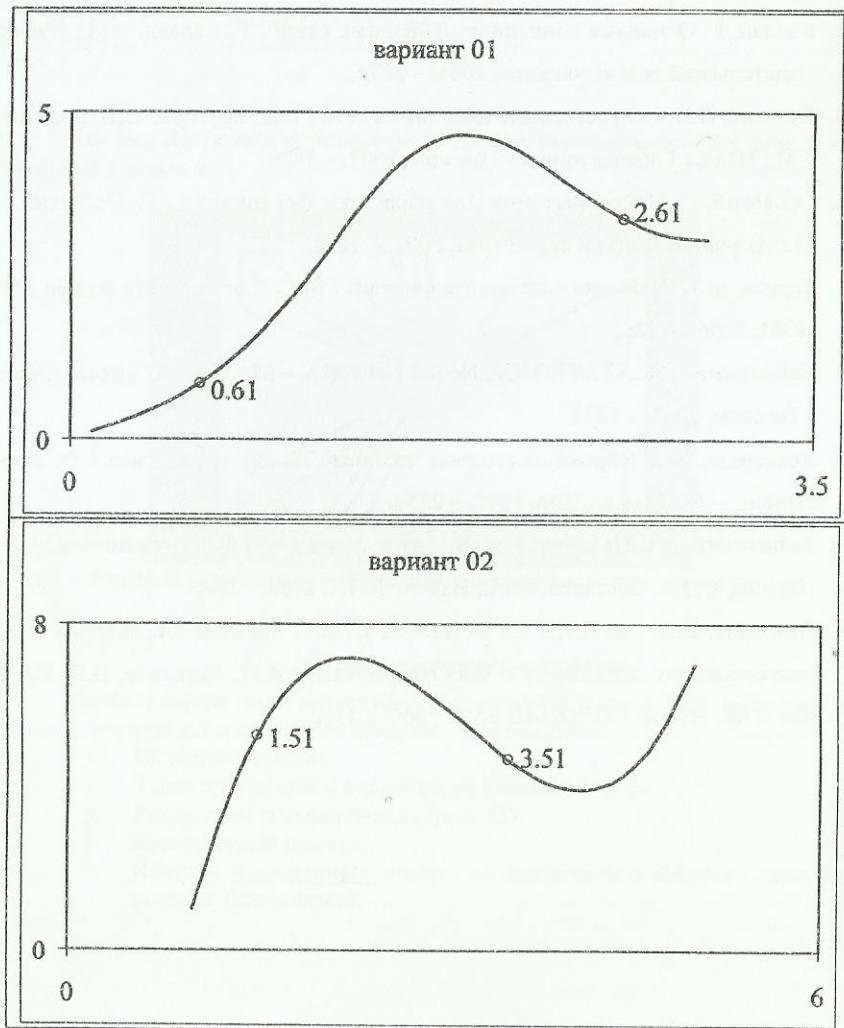
1. Исходные данные;
2. Текст программы с подробными комментариями;
3. Результаты моделирования (рис. 13);
4. Контрольный пример;
5. Краткие письменные ответы на контрольные вопросы, содержащиеся в Приложении.

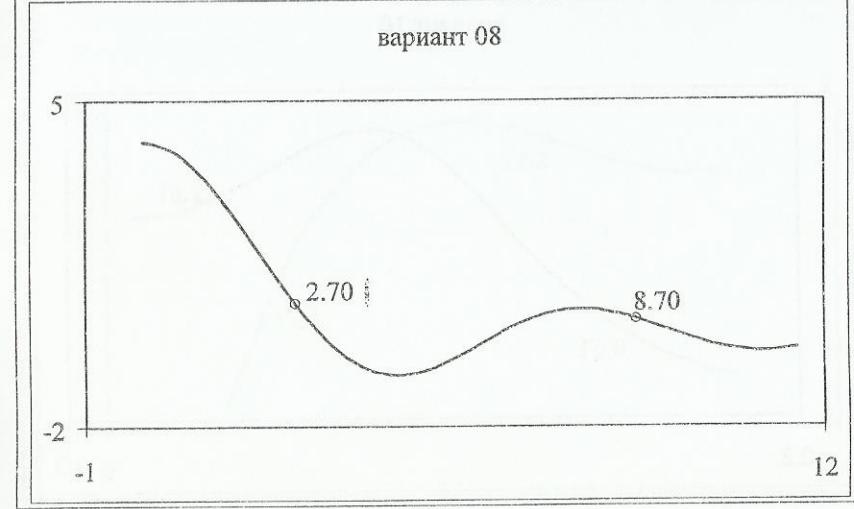
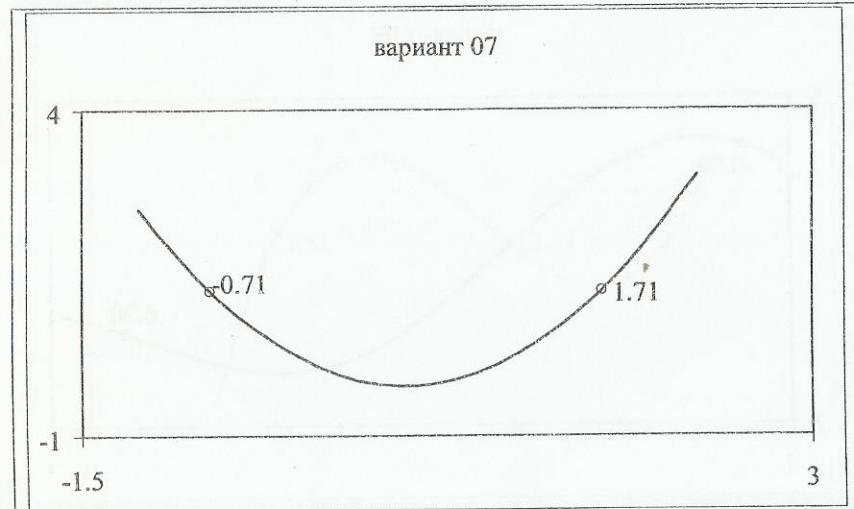
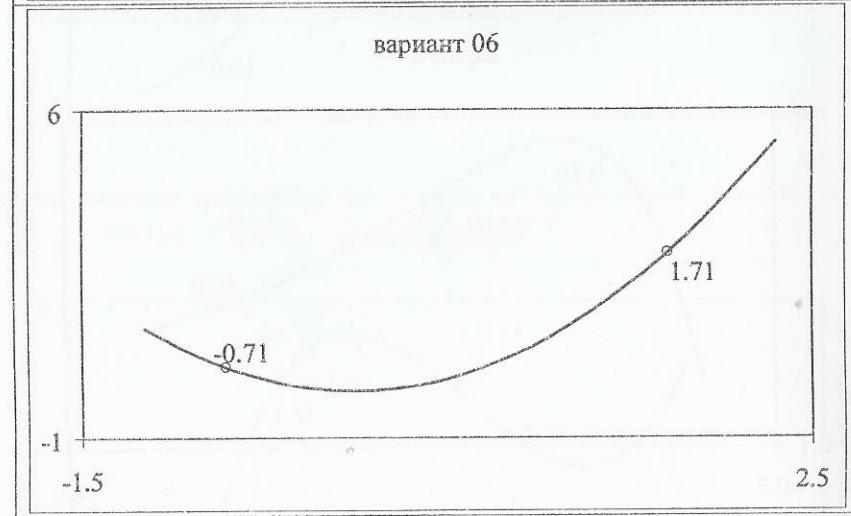
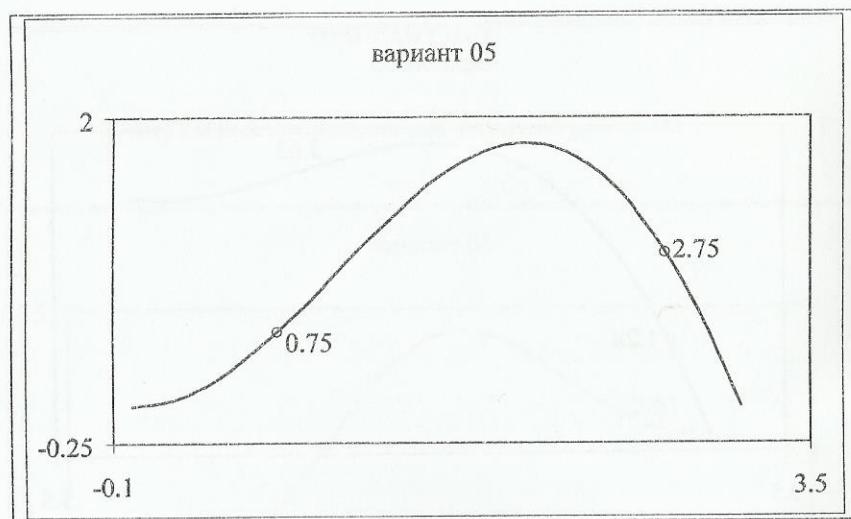
Библиографический список

1. Медведев, В.С. Нейронные сети. Матлаб 6 / В.С. Медведев, В.Г. Потемкин. -- М.: Изд-во Диалог МИФИ, 2002. -- 496с.
2. Каллан, Р. Основные концепции нейронных сетей / Р. Каллан. -- М.: Изд-во Издательский дом «Вильямс», 2001. -- 287с.
3. Круглов, В.В. Искусственные нейронные сети / В.В. Круглов, В.В. Борисов. -- М.: Изд-во Горячая линия - Телеком, 2001. -- 382с.
4. Осовский, С. Нейронные сети для обработки информации / С. Осовский. -- М.: Изд-во Финансы и статистика, 2002. -- 344с.
5. Тарков, М.С. Нейрокомпьютерные системы / М.С. Тарков. -- М.: Изд-во БИ-НОМ, 2006. -- 142с.
6. Нейронные сети. STATISTICA Neural Networks. -- М.: Изд-во Горячая линия - Телеком, 2001. -- 182с.
7. Уоссермен, Ф. Нейрокомпьютерная техника: Теория и практика / Ф. Уоссермен. -- М.: Изд-во Мир, 1992. -- 238с.
8. Зайцевский, И.В Нейронные сети и их приложения / И.В. Зайцевский, А.П. Свиридов, Д.А. Слесарев. -- М.: Изд-во МЭИ, 2002. -- 95с.
9. Дьяконов, В.П. MATLAB 6.5 SP1/7/7 SP1/7 SP2 Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики / В.П. Дьяконов, В.В. Круглов. -- М.: Изд-во СОЛОН-ПРЕСС, 2006. -- 456с.

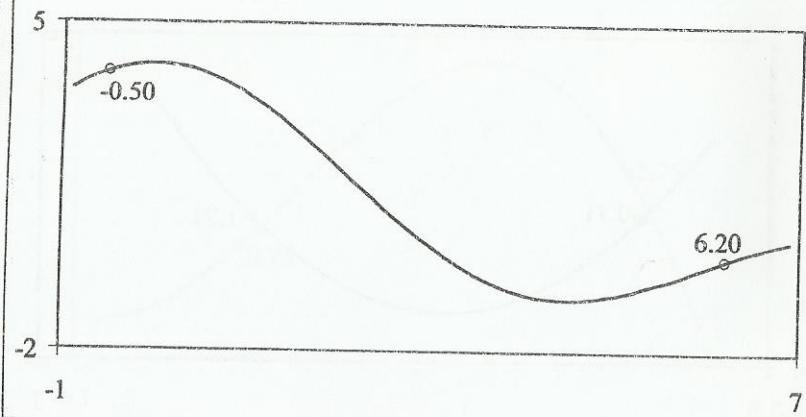
ПРИЛОЖЕНИЕ

Контрольные задания для лабораторной работы № 1

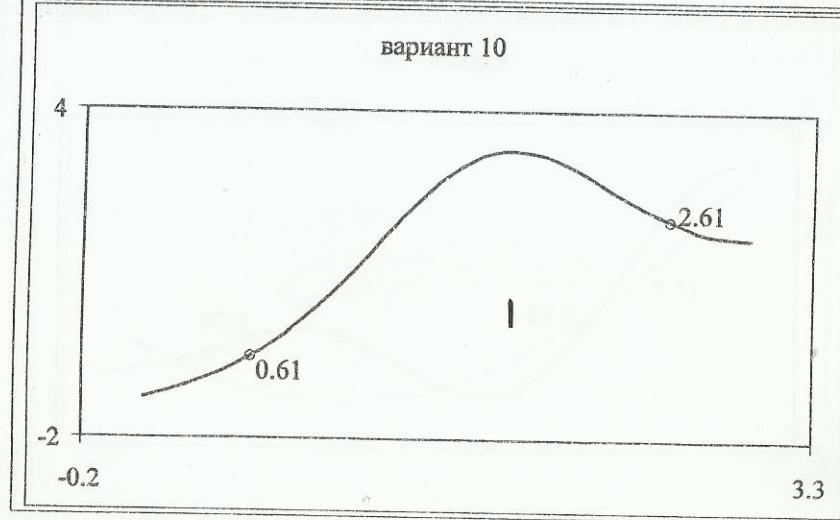




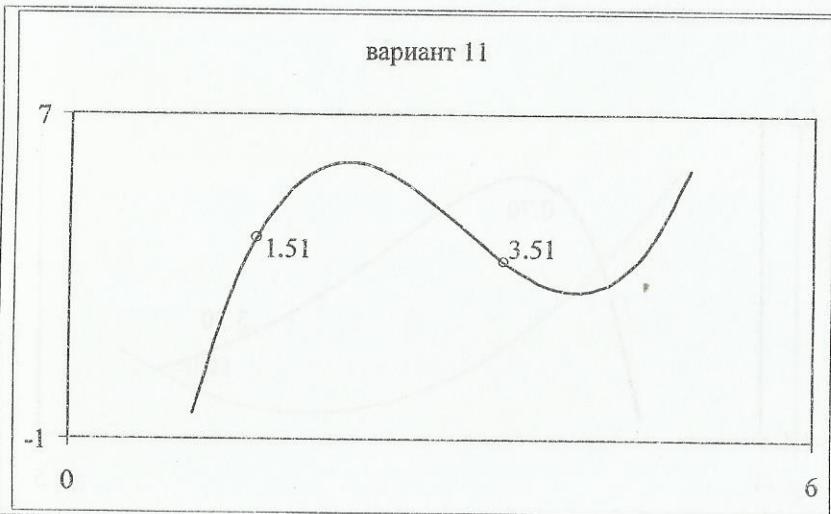
вариант 09



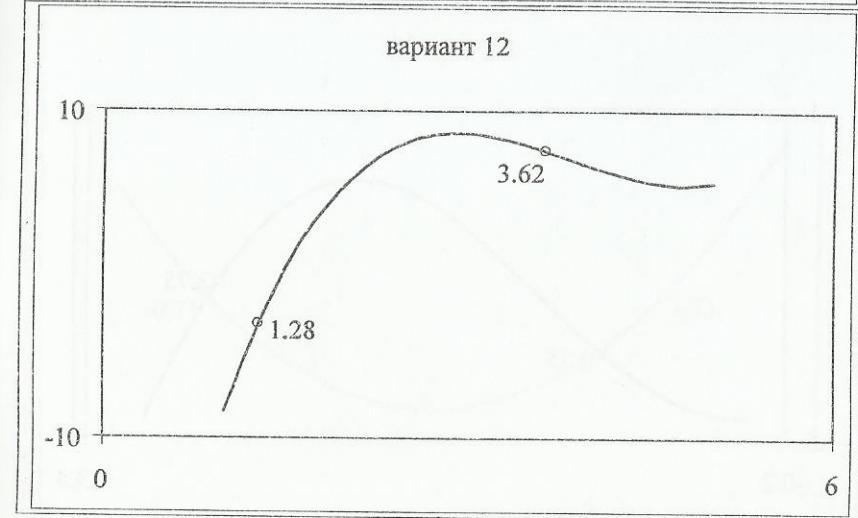
вариант 10



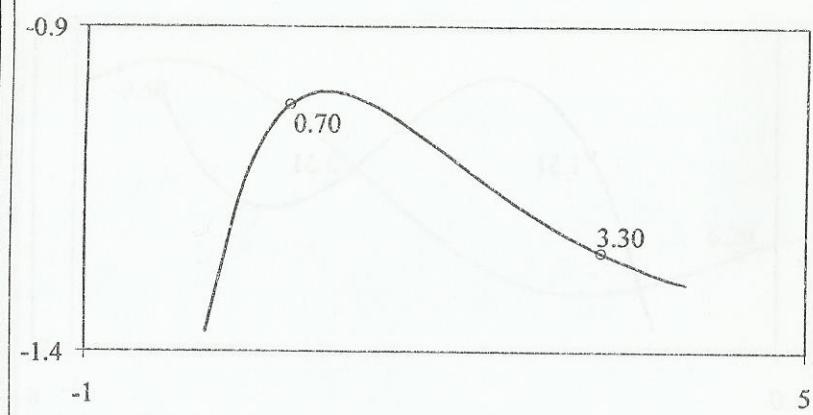
вариант 11



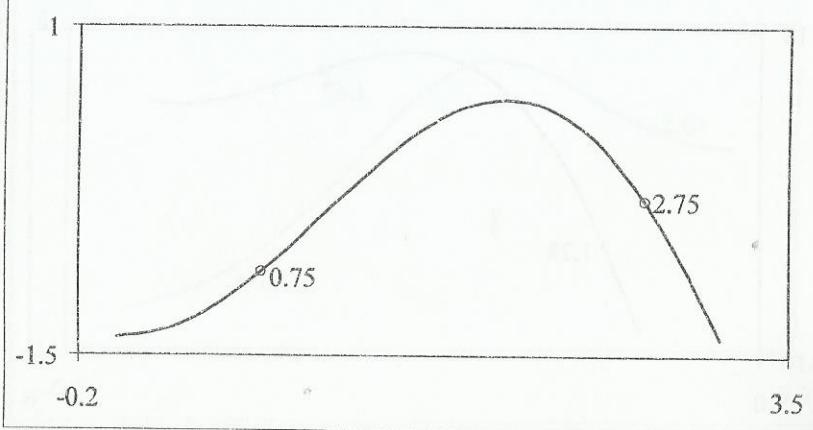
вариант 12



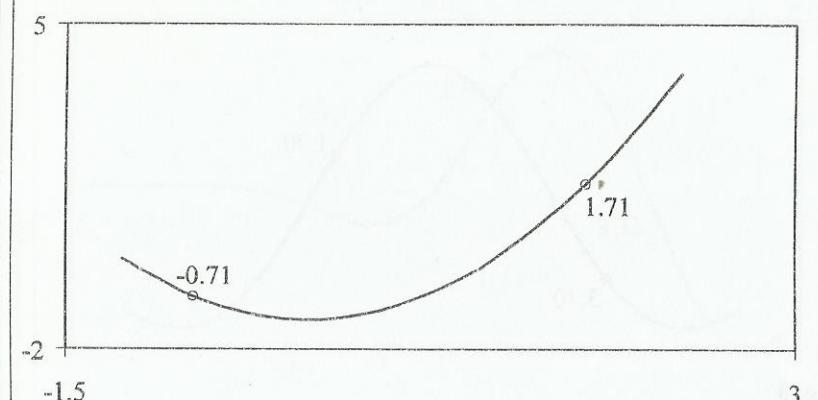
вариант 13



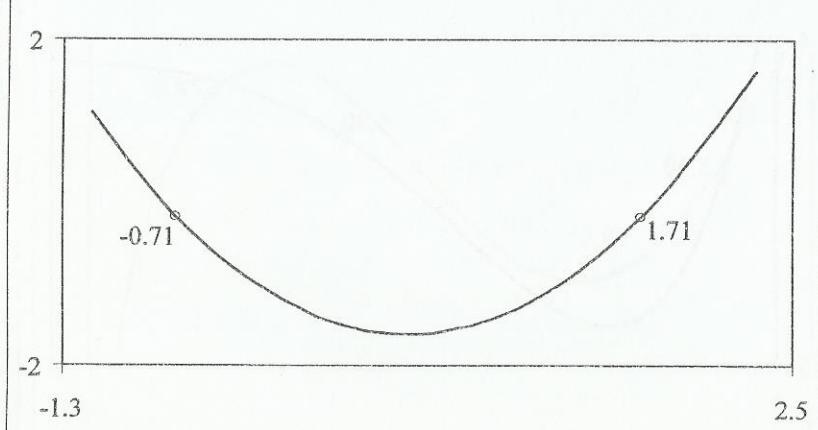
вариант 14



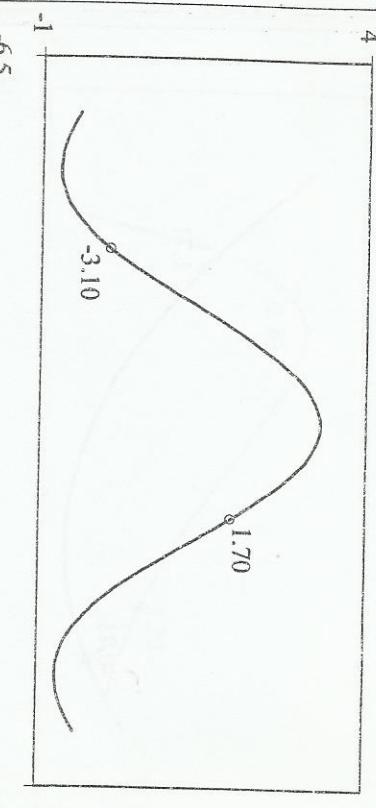
вариант 15



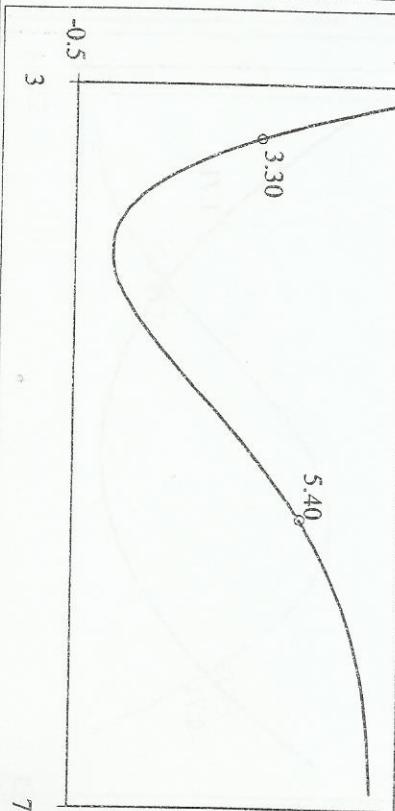
вариант 16



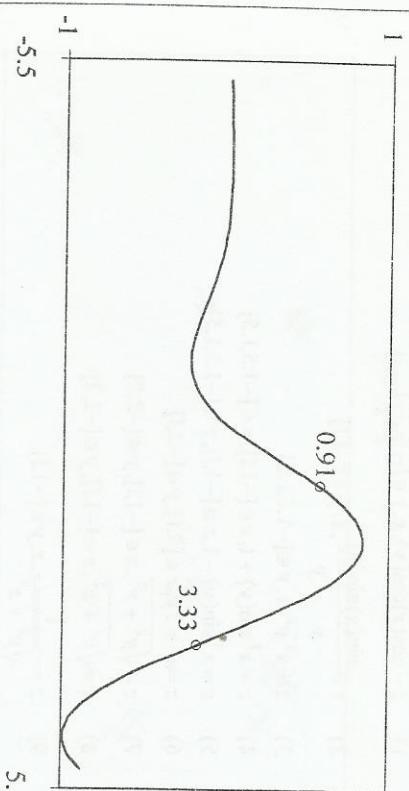
вариант 17



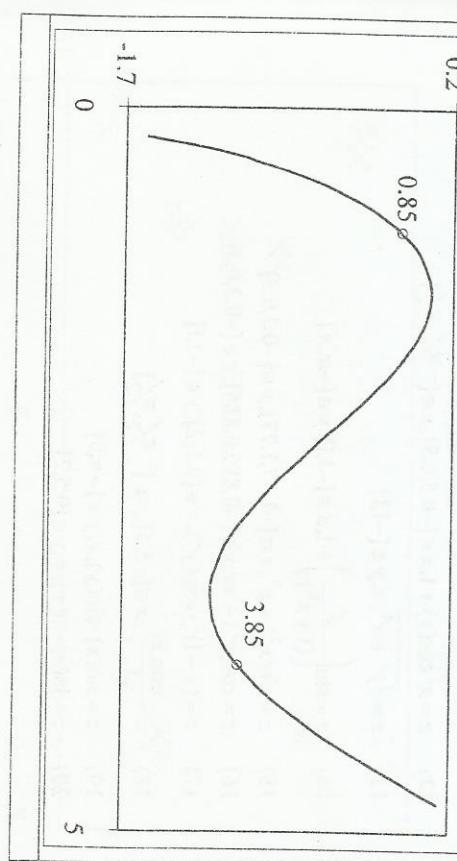
вариант 18



вариант 19



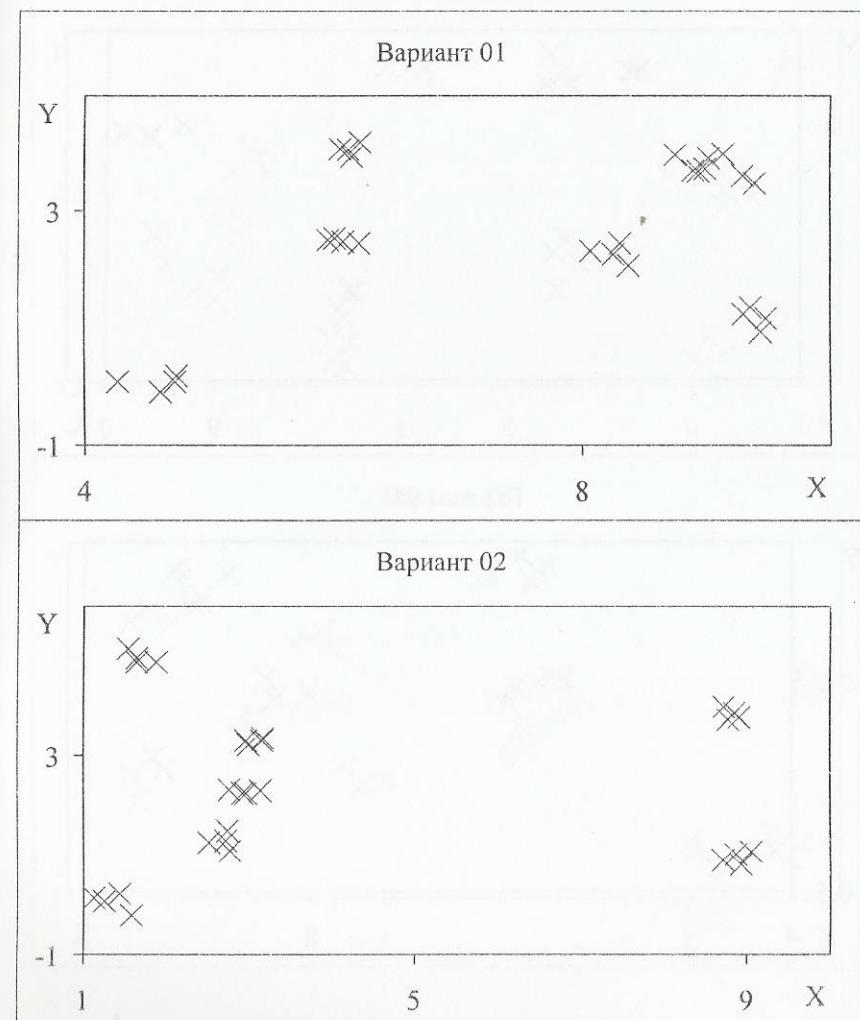
вариант 20



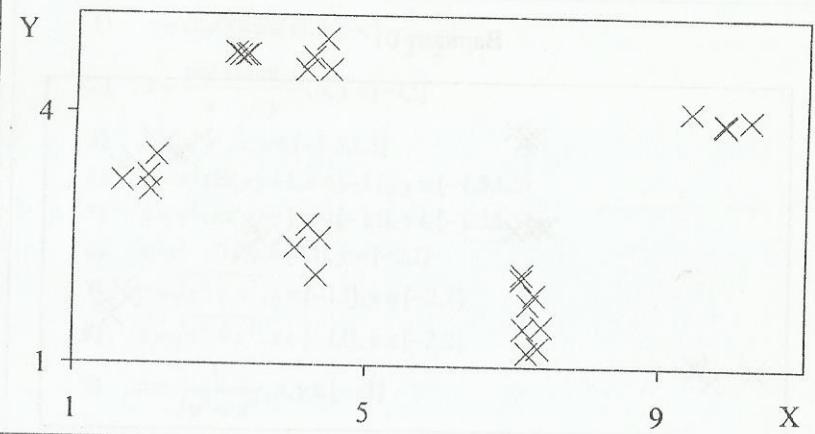
Контрольные задания для лабораторной работы № 2

- 1) $z = \cos(x)\cos(y), x, y \in [-\frac{\pi}{2}, \frac{\pi}{2}]$
- 2) $z = \frac{\sin(x)\sin(y)}{x-y}, x, y \in [-1, 1]$
- 3) $z = x^2 y^2, x, y \in [-1.5, 1.5]$
- 4) $z = x^3 \sin(y) + 1, x \in [-1, 1], y \in [-1.5, 1.5]$
- 5) $z = x^2 \sin(y) - 1, x \in [-1, 1], y \in [-1.5, 1.5]$
- 6) $z = e^x + 3y, x \in [0, 1], y \in [-2, 1]$
- 7) $z = \sqrt{y^2 + x^3}, x \in [-1, 1], y \in [-2, 2]$
- 8) $z = \sqrt{y^3 + x^2}, x \in [-1, 1], y \in [-2, 2]$
- 9) $z = \frac{1}{\sqrt{y^2 + x^3}}, x, y \in [-1, 1]$
- 10) $z = \frac{1}{\sqrt{y^3 + x^2}}, x, y \in [-1, 1]$
- 11) $z = x^2 \cos(y) + 1, x \in [-1, 1], y \in [-\pi/4, \pi/4]$
- 12) $z = x^3 \cos(y) + 1, x \in [-0.5, 0.5], y \in [-\pi/4, \pi/4]$
- 13) $z = \sqrt{y^2 + x^4}, x, y \in [-1, 1]$
- 14) $z = \sin\left(\frac{y}{(1+x^2)}\right) + 1, x \in [-1, 1], y \in [-\pi, \pi]$
- 15) $z = \sin(x^2) + y^2, x \in [-1.77, 1.77], y \in [-0.3, 0.3]$
- 16) $z = \cos(x^2) + xy, x \in [-0.886, 0.886], y \in [-0.3, 0.3]$
- 17) $z = (x-1)^2 \cdot \exp(y^2), x \in [0.5, 2], y \in [-1, 1]$
- 18) $z = \frac{\cos(x)}{x^2 + 1}, x \in [-5, 5], y \in [-\pi/2, \pi/2]$
- 19) $z = \sin(x) \cdot \sin(y), x, y \in [-\pi, 0]$
- 20) $z = \ln(x) \cdot \ln(y), x, y \in [0.5, 2]$

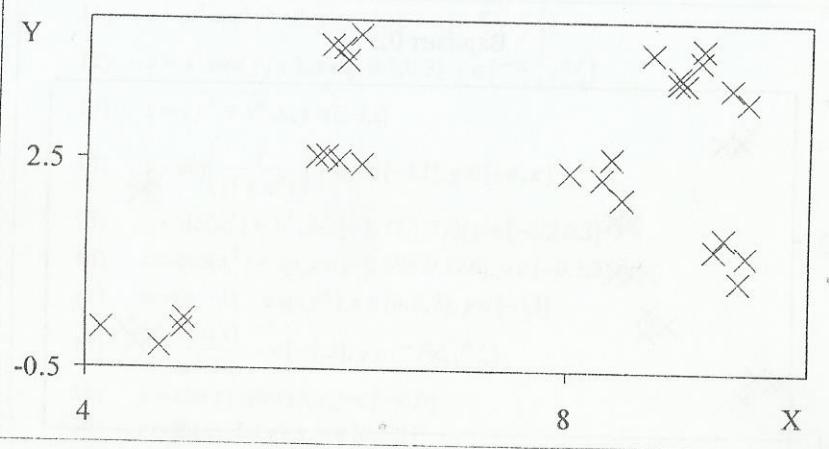
Контрольные задания для лабораторной работы № 3



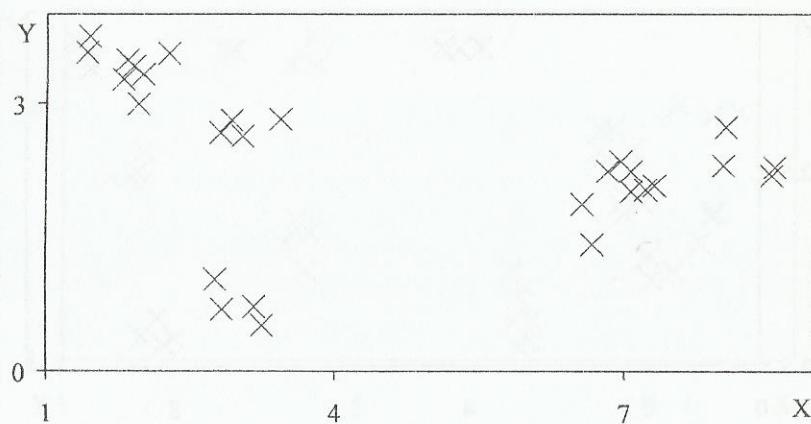
Вариант 03



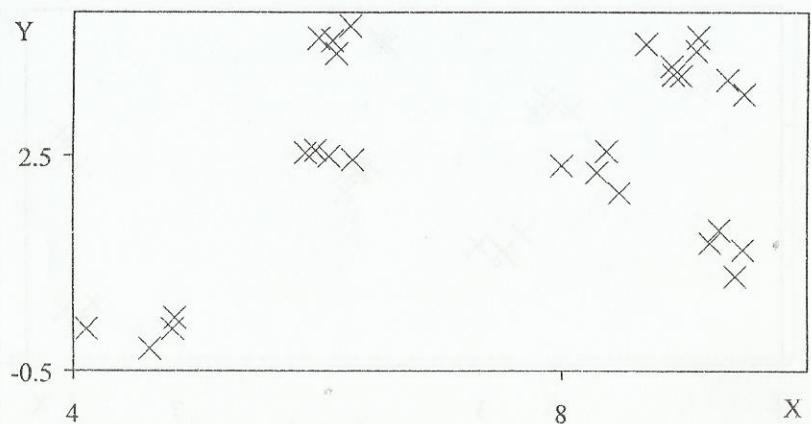
Вариант 04



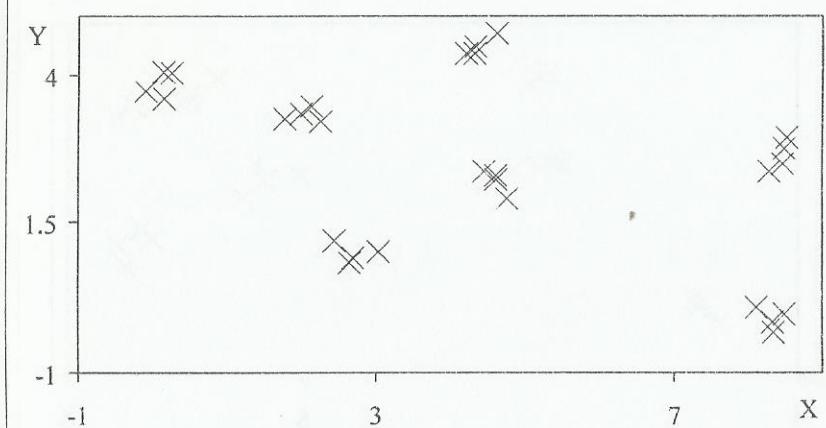
Вариант 07



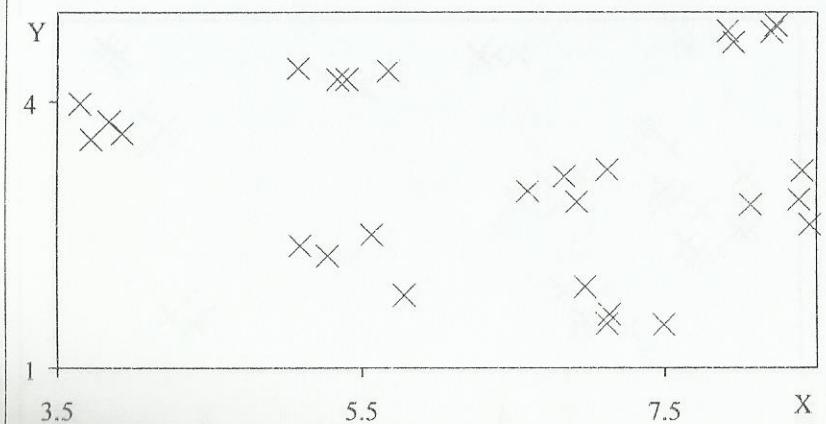
Вариант 08



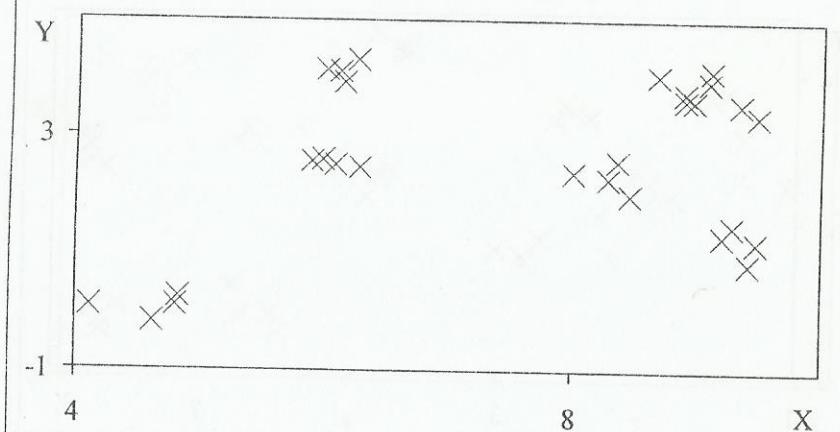
Вариант 09



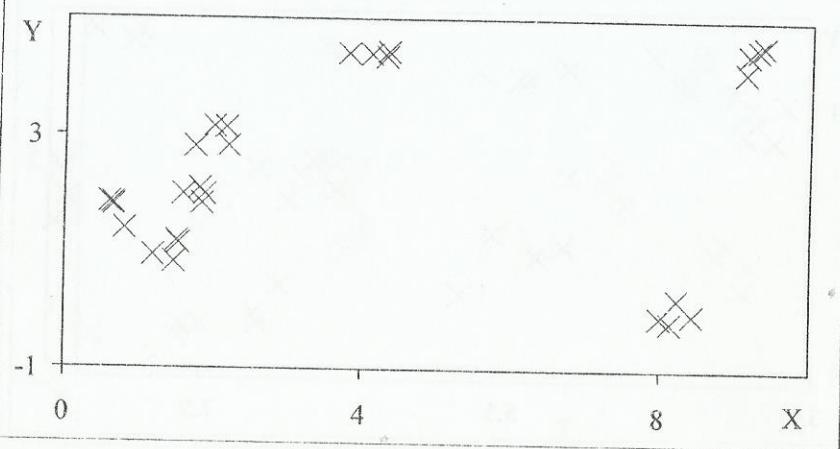
Вариант 10



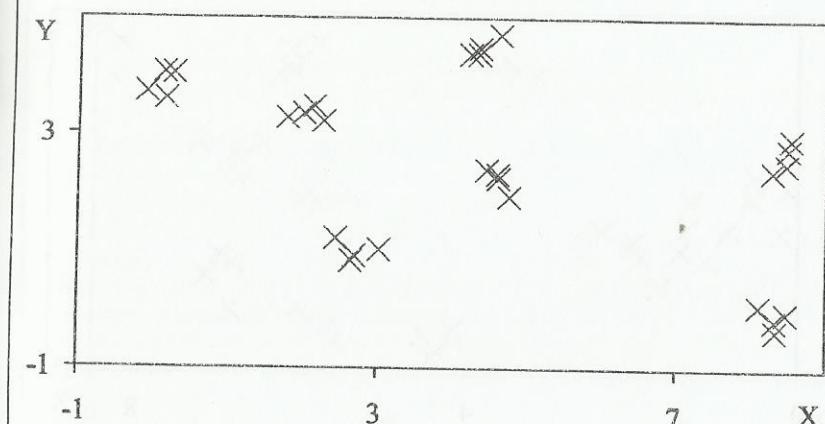
Вариант 11



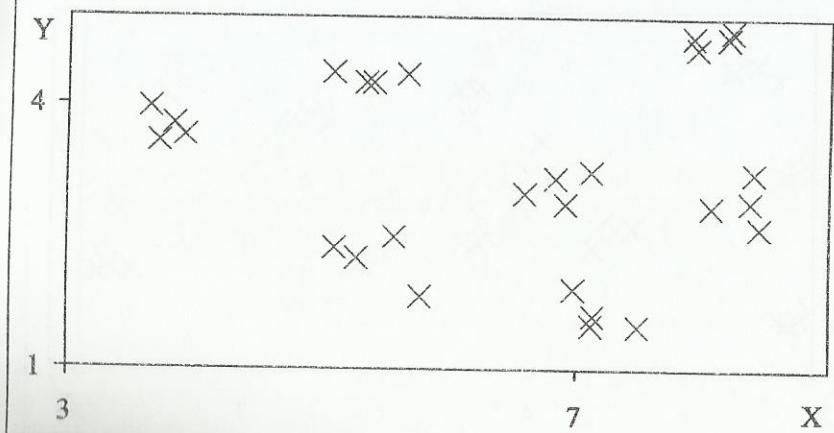
Вариант 12



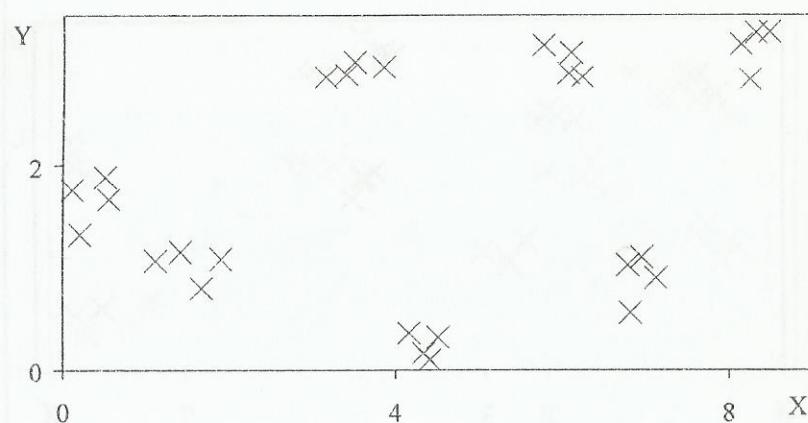
Вариант 13



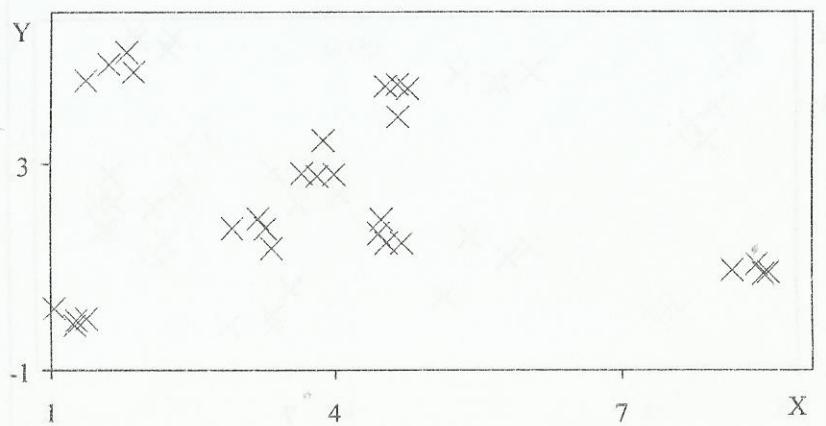
Вариант 14



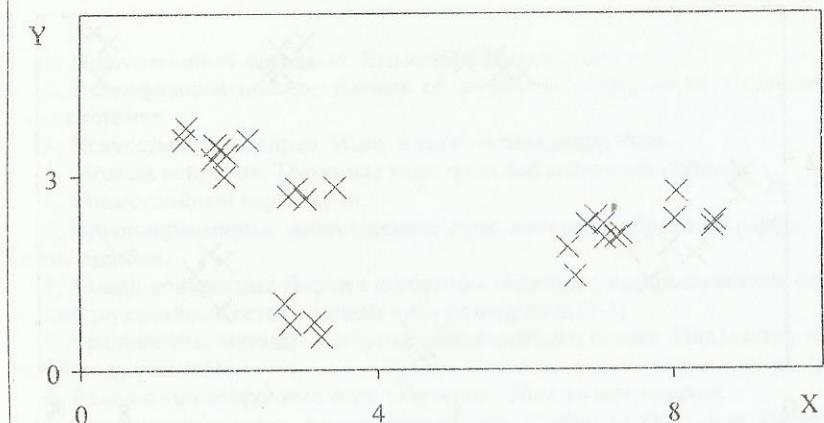
Вариант 15



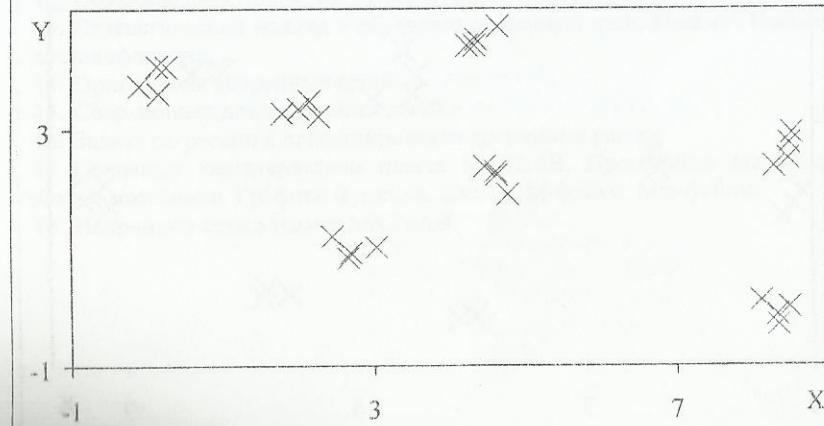
Вариант 16



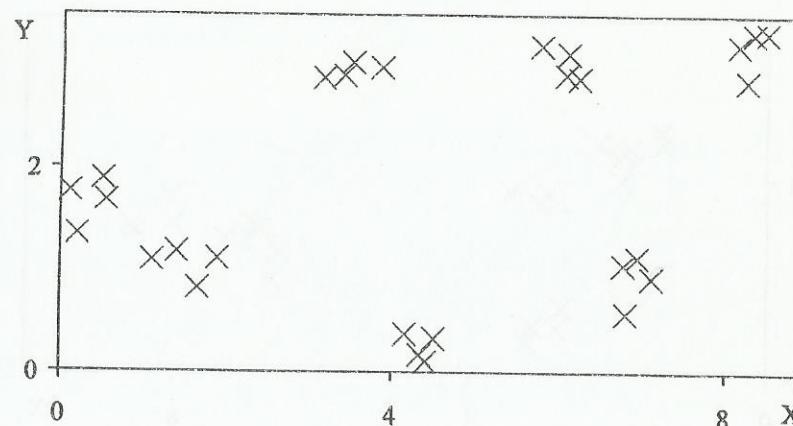
Вариант 17



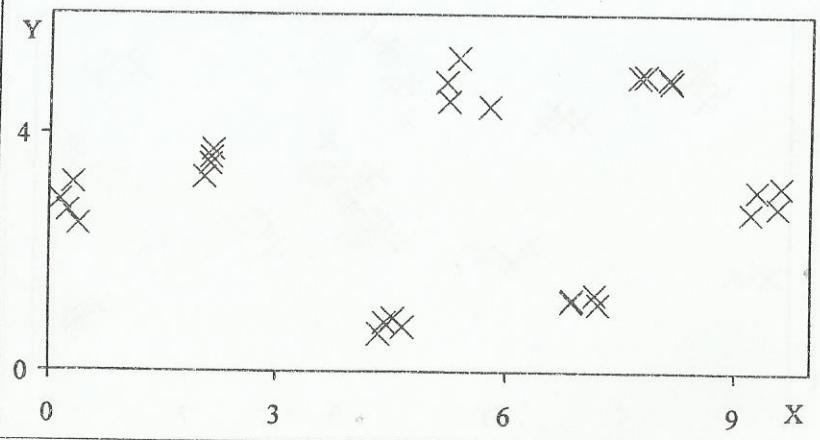
Вариант 18



Вариант 19



Вариант 20



Контрольные вопросы для лабораторной работы № 4

1. Искусственный интеллект. Его истоки и проблемы.
2. Нейрофизиологические данные об обработке информации в биологических системах.
3. Искусственный нейрон. Идея и техническая реализация.
4. Модели нейронов. Типичные виды функций активации нейрона.
5. Многослойный персептрон.
6. Однонаправленные многослойные сети. Алгоритм обратного распространения ошибки.
7. Вывод конкретных формул алгоритма обратного распространения ошибки для двухслойных сетей с малым числом нейронов (2-3).
8. Градиентные методы. Алгоритм наискорейшего спуска. Недостатки метода. Метод моментов.
9. Радиальные нейронные сети. Обучение. Область применения.
10. Рекуррентные сети. Ассоциативная сеть Хопфилда. Обучение. Распознавание образов.
11. Сеть встречного распространения.
12. Обучение слоя Кохонена. Решение задач кластеризации.
13. Статистический подход к обучению нейронной сети. Машина Больцмана и ее модификации.
14. Применение нейронных сетей.
15. Сбор данных для нейронных сетей.
16. Задача регрессии и прогнозирования временных рядов.
17. Основные характеристики пакета MATLAB. Простейшие вычисления. Работа с массивами. Графики функций. Сессия. M-файлы. Mat-файлы.
18. Нейронные сети в пакете MATLAB.

Оглавление

Введение.....	3
Аппроксимация функции одной переменной.....	4
Аппроксимация функции двух переменных	9
Решение задачи классификации с использованием самоорганизующейся нейронной сети Кохонена.....	13
Сеть Хопфилда	17
Библиографический список.....	21
Приложение	22
Контрольные задания для лабораторной работы № 1	22
Контрольные задания для лабораторной работы № 2	32
Контрольные задания для лабораторной работы № 3	33
Контрольные вопросы для лабораторной работы № 4	43

МОДЕЛИРОВАНИЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ В СИСТЕМЕ MATLAB

*Методические указания
к выполнению лабораторных работ
для студентов 3-го курса, обучающихся по специальностям
222600 «Инноватика»,
080500 «Бизнес-информатика»*

Составители: Головинский Павел Абрамович
Михин Евгений Александрович

Подписано в печать 27.10.2014. Формат 60 × 84 1/16. Уч.-изд. л. 2,0.
Усл.-печ. л.2,1. Бумага писчая. Тираж 50 экз. Заказ № 429.

Отпечатано: отдел оперативной полиграфии
издательства учебной литературы и учебно-методических пособий
Воронежского ГАСУ
394006 Воронеж, ул. 20-летия Октября, 84