

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»

Кафедра программного обеспечения автоматизированных систем

**Эмулятор нейронной сети для задач обучения
«Neuro-Lab»**

Методические указания
к выполнению лабораторных и практических работ для студентов
направлений подготовки 09.03.04, 09.04.04 «Программная инженерия»

Курган 2017

Кафедра: «Программное обеспечение автоматизированных систем».

Дисциплина: «Нейрокомпьютерные системы».

Составил: канд. физ.-мат. наук, профессор В.А. Симахин.

Утверждены на заседании кафедры « 20 » марта 2017 г.

Рекомендованы методическим советом университета « 12 » декабря 2016 г.

Оглавление	
Введение	4
Руководство пользователя	10
1 Установка	10
2 Использование программы.....	11
3 Типовые задачи	16
3.1 Обучающие пары	16
3.2 Распознавание образов	18
3.3 Классификация объектов.....	20
3.4 Регрессия функции одной переменной.....	22
3.5 Регрессия функции двух переменных	24
3.6 Кластер-анализ.....	26
3.7 Восстановление образов.....	30
3.8 Системы линейных уравнений.....	33
4 Теоретическая часть.....	34
4.1 Многослойный перцептрон.....	35
4.2 Сети RBF	35
4.3 Сети Кохонена	36
4.4 Сети Хопфилда	37
Список литературы.....	37

Введение

В настоящее время возрастает актуальность использования нейронных сетей в задачах анализа и обработки данных. В качестве инструментов практического изучения нейронных сетей сейчас используются такие эмуляторы, как MATLAB Neural Network Toolbox [3], STATISTICA Neural Networks [4] и т. д. Применение таких нейроэмуляторов при обучении приводит к некоторым трудностям, к которым можно отнести слишком высокий уровень сложности программных продуктов, отсутствие наглядности работы нейронных сетей, недостаток справочной информации. Все эти недостатки систем приводят к тому, что студент тратит значительное время на изучение самого программного продукта, а не на изучение основных парадигм нейронных сетей. Кроме того, для базисного обучения студентов в связи с ограниченным объемом часов обычно используется лишь небольшая часть функциональных возможностей, которые предлагают современных пакеты. Решение этих проблем лежит в написании более удобного программного комплекса для работы с нейронными сетями, который бы соответствовал представлениям студента о структуре, принципах и основных парадигмах работы нейронных сетей. Необходимо наглядно показать алгоритмы обучения с учителем и без него, последовательность действий по работе с нейронной сетью и основные задачи, которые может решать обученная сеть. Этот процесс должен происходить в интерактивном режиме с удобным интерактивным механизмом задания архитектуры сети, обучающих пар и выдачи окончательных результатов. Такой программный комплекс был реализован на кафедре ПОАС КГУ. Основной упор работы был сделан на создание удобного и понятного интерфейса работы с нейронными сетями для студентов, при этом используется ограниченный набор типов нейронных сетей и методов их обучения. Цель комплекса программ Neuro-lab – показать в доступной форме пользователям основные задачи, которые могут решать нейронные сети [1; 2].

Визуальное представление нейронной сети в программном комплексе реализовано в виде изображения нейронных элементов и связующих их синапсов (рисунок 1). При этом входной, скрытые и выходной слои разделены цветом и геометрическим изображением нейронных элементов. Разные типы нейронных сетей визуально различаются, поэтому студент всегда может понять, с какой сетью он работает. Графический интерфейс нейронной сети позволяет просматривать и легко изменять параметры нейронных элементов сети. В программу встроен редактор нейронных сетей, который позволяет визуально проектировать архитектуру нейронных сетей заданного типа.

Для решения конкретной задачи необходимо ввести исходные данные. Нередко ввод входных данных, особенно графических образов, занимает много времени, что влечет за собой снижение восприятия студентом дисциплины. Поэтому была введена специальная подсистема для подготовки данных,

которая интерпретирует человеческое представление данных во входы нейронной сети. В данной работе были реализованы несколько таких систем:

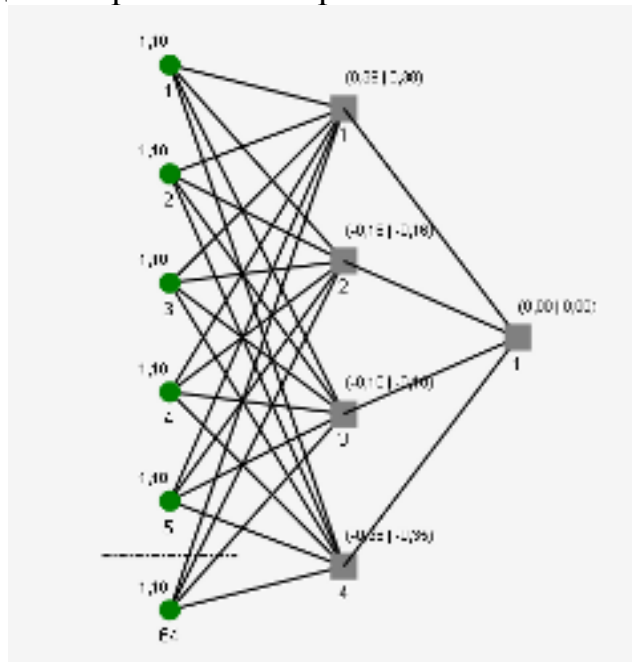


Рисунок 1 – Нейронная сеть

- система, интерпретирующая текстовые данные определенного формата;
- система, использующая табличный формат данных;
- система подготовки данных для решения задачи регрессии;
- система подготовки данных для решения задачи классификации образов;
- система подготовки данных для решения задачи кластерного анализа;
- система подготовки данных для решения задачи распознавания образов.

Каждая такая подсистема снабжена понятным интерфейсом, примерами и не вызывает сложности восприятия у студентов при подготовке данных.

Нейроэмулятор может работать с данными для обучения сети, которые находятся в базе данных под управлением Microsoft Access или Microsoft Server SQL, в табличном формате Microsoft Excel или генерируемые непосредственно при решении конкретной задачи.

В данной работе были реализованы следующие типовые задачи: задача классификации, задача кластерного анализа и задача регрессии, решение линейных уравнений, примеры ассоциативной памяти.

Рассмотрим графические интерфейсы при решении вышеупомянутых задач.

1 Задача классификации

Данная задача разбивается на два этапа, как и все задачи обучения с учителем. На первом этапе от студента требуется сформировать обучающие образы и обучить на них нейронную сеть. Формирование графических образов происходит в специальном окне редактирования образов (рисунок 2). Размерность изображений определяется размерностью матрицы, нанесенной на данное окно. По желанию пользователя размерность матрицы изображения можно изменять. На втором этапе студенту требуется распознать входящий

образ, который формируется так же, как и обучающие образы. Результатом распознавания будут вероятности отнесения входного образа к образам из обучающего множества (рисунок 2).

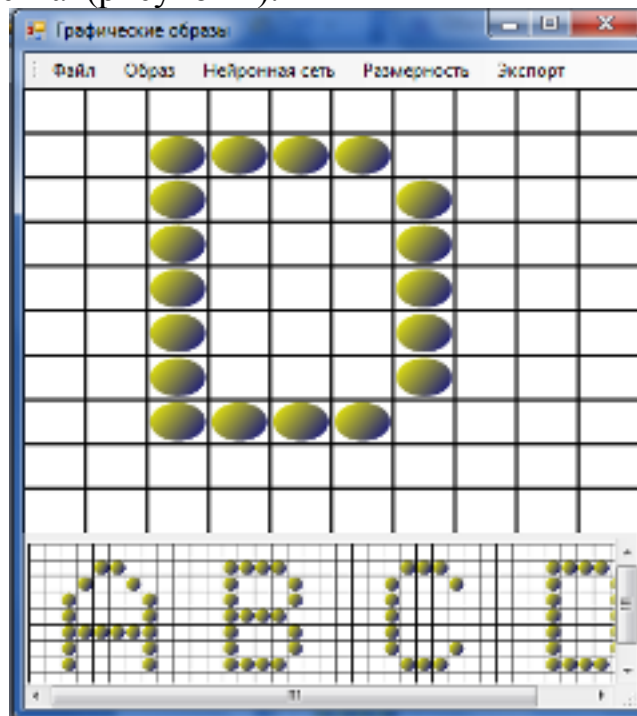


Рисунок 2 – Задача распознавания образов (исходные данные)

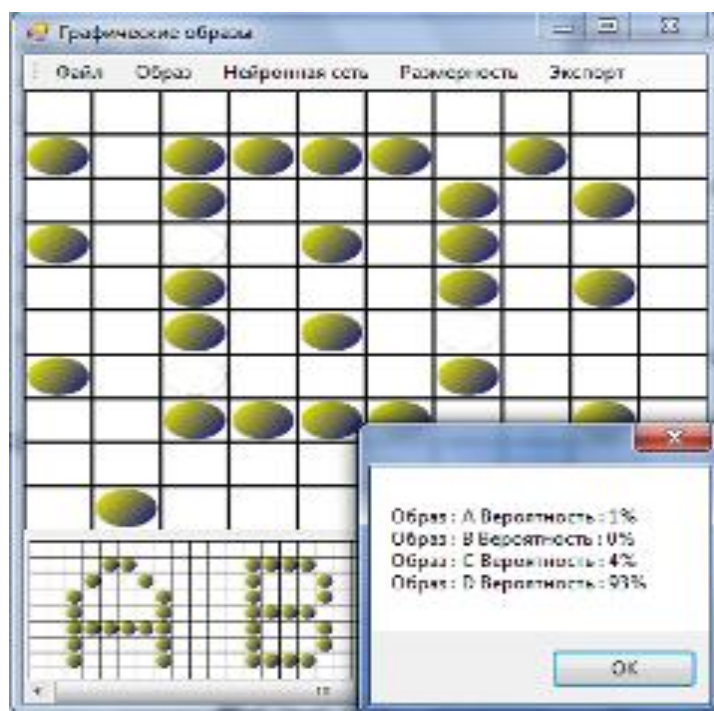


Рисунок 3 – Задача распознавания образов (результаты)

2 Задача классификации графических объектов

Графический интерфейс данной задачи представлен двумерной координатной плоскостью (рисунок 4). На нее наносятся студентом объекты в виде точек. Объекты одного класса будут иметь один и тот же цвет. После

создания и обучения нейронной сети по данным объектам можно приступить к решению задачи классификации. Этот процесс построен интерактивно. Пользователь указывает положения нового объекта, а система относит его к классу, присваивая ему соответствующий цвет (рисунок 5).

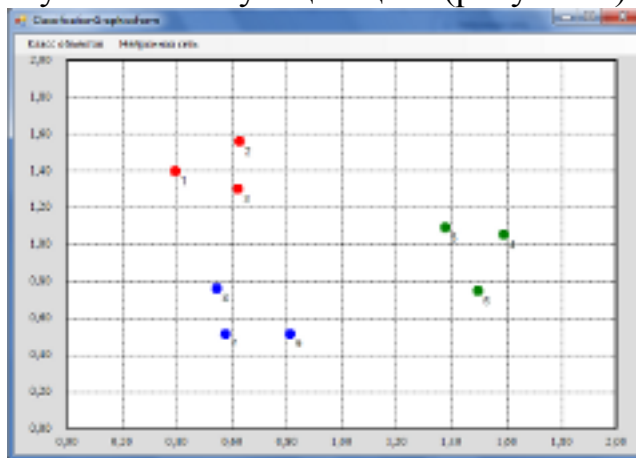


Рисунок 4 – Задача классификации (исходные данные)

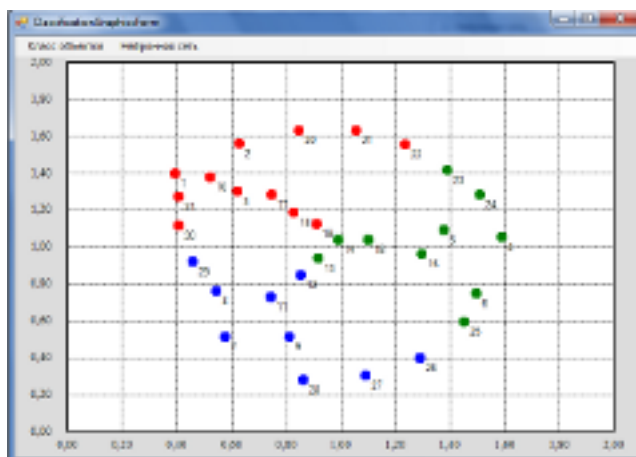


Рисунок 5 – Задача классификации (результаты)

3 Задача кластерного анализа

Графический интерфейс сходен с графическим интерфейсом предыдущей задачи. Студент наносит объекты на двумерную координатную плоскость (рисунок 6). В данной задаче визуально показан и сам процесс обучения, то есть как происходит изменение положения центров кластеров в течение обучения нейронной сети (рисунок 7). Количество кластеров студент задает путем определения архитектуры нейронной сети.

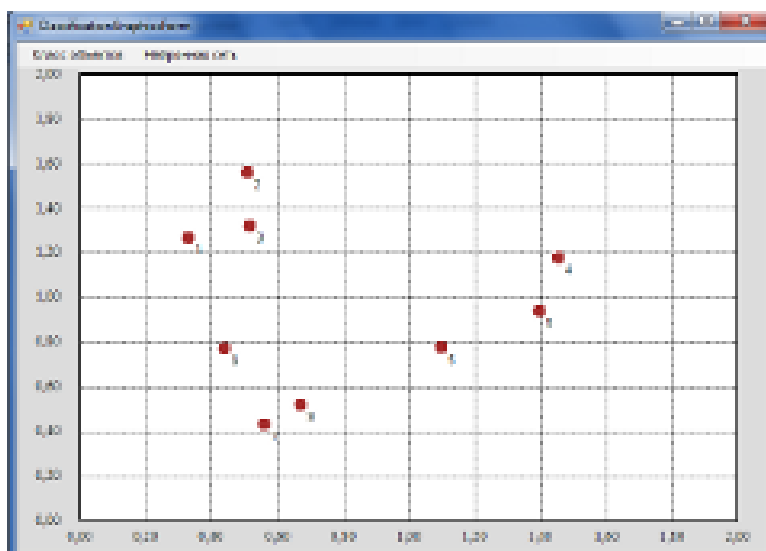


Рисунок 6 – Задача кластерного анализа (исходные данные)

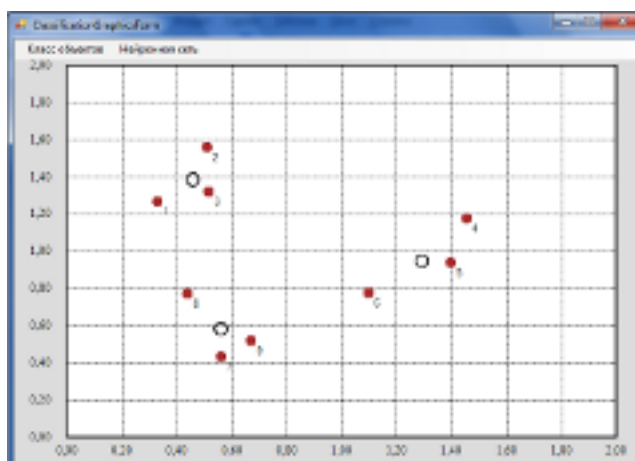


Рисунок 7 – Задача кластерного анализа (результаты)

Задача регрессии

Для решения задачи регрессии необходимо получить выборку. Для этого задается функция одного или двух переменных, интервал изменения аргументов функции, величина помех и объем выборки. В системе реализован интерпретатор математических выражений, поэтому студент может задавать различные функциональные зависимости. Графический интерфейс (рисунок 8) позволяет отображать исходную функцию, сгенерированную выборку и оценку исходной функции нейронной сетью. Процесс обучения нейронной сети визуализирован – показано, как подстраивается нейронная сеть под функциональную зависимость (рисунки 9, 10).

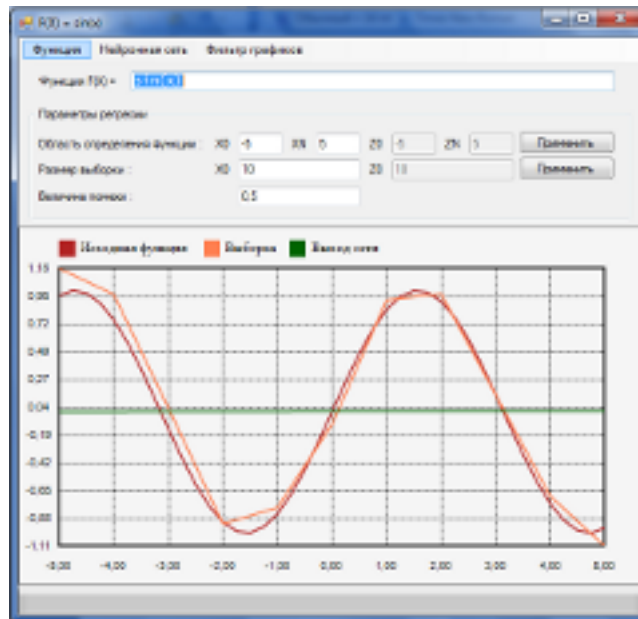


Рисунок 8 – Задача регрессии одной переменной (исходные данные)

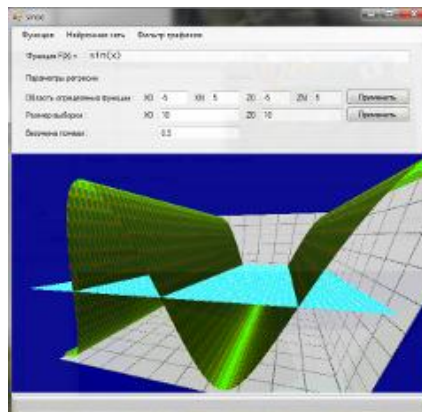


Рисунок 9 – Задача регрессии двух переменных (исходные данные)

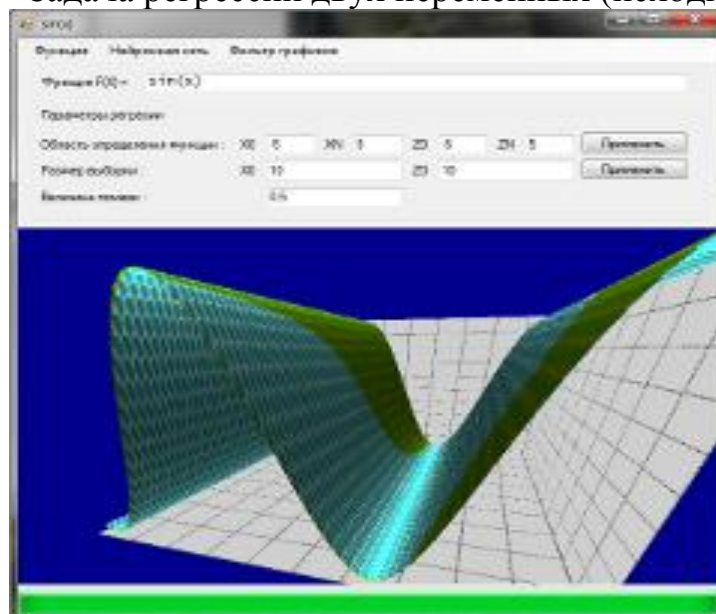


Рисунок 10 – Задача регрессии двух переменных (результаты)

Разработанный нейроэмулятор применяется для обучения студентов дисциплине «Нейронные сети» при проведении лабораторных и практических работ.

Руководство пользователя

1 Установка

Для начала установки необходимо запустить файл «SetupNeuro-lab.exe» (рисунок 11).

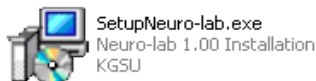


Рисунок 11 – Файл для запуска

В появившемся меню необходимо указать каталог, в котором будет установлена программа Neuro-lab. Указать каталог можно с помощью клавиши «Обзор...» или введя путь к каталогу вручную. Далее нажимаем «Установить» (рисунок 12).

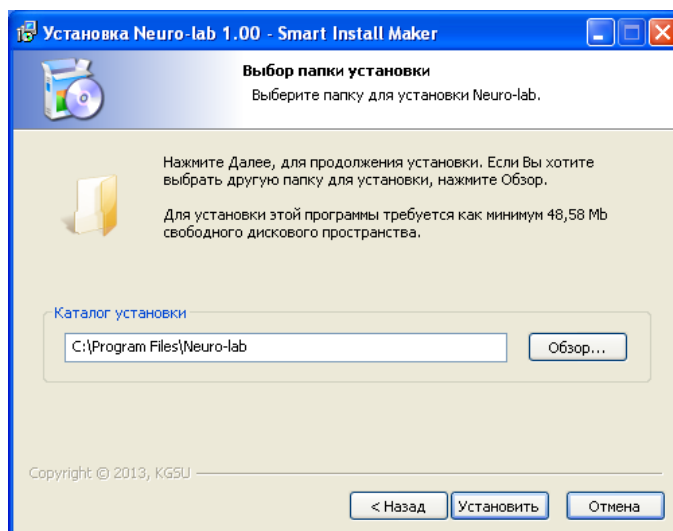


Рисунок 12 – Выбор папки установки

В процессе установки будет предложена дополнительная установка программ DirectX и .Net Framework 4. Можно пропустить их установку и сделать это позже, т. к. установочные файлы будут находиться в папке с установленным Neuro-lab. (Установщик DirectX представляет из себя on-line версию. Лучше скачать полный установочный пакет (рисунок 13).)

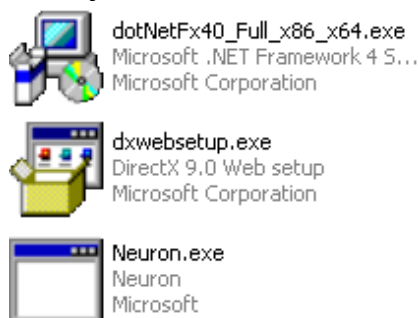


Рисунок 13 – Установочный пакет

Для завершения установки необходимо принять условия лицензионного соглашения (рисунок 14).

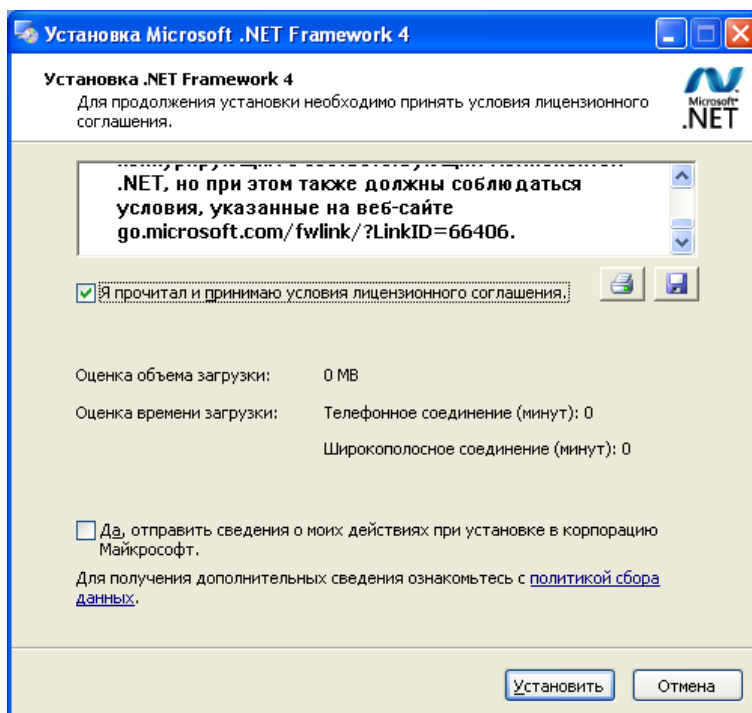


Рисунок 14 – Принятие соглашения

По завершению установки необходимо вручную создать ярлык для программы Neuro-lab. Главный файл- Neuron.exe.

2 Использование программы

Архитектура сети

Контекстное меню в верхней левой части экрана выполняет функционал, описанный пунктами ниже.

Плавающее меню «Архитектура сети» позволяет задавать параметры сети и параметры для обучения данной сети (рисунок 15).

«Количество входов» задаёт количество входов сети. «Количество слоёв» задаёт количество слоёв в сети. «Количество нейронов» позволяет задавать количество нейронов в том слое, номер которого указан в области «в слое:».

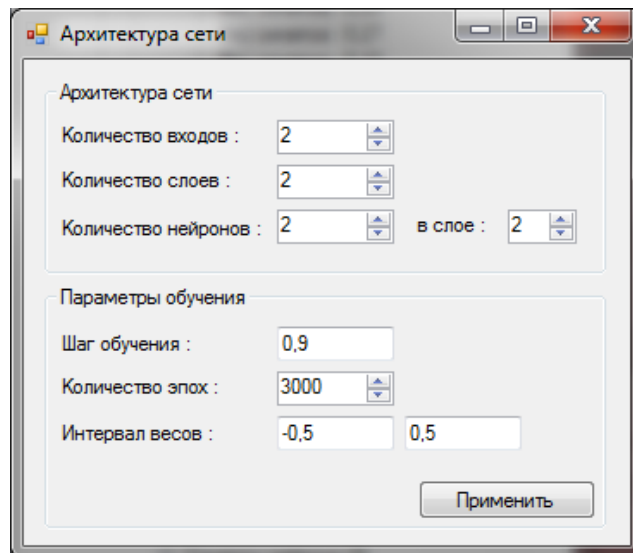


Рисунок 15 – Ввод параметров сети

«Шаг обучения» – это скорость обучения для данной сети. «Количество эпох» – количество итераций обучения. «Интервал весов» позволяет задать интервал, в котором могут быть сгенерированы случайные веса синапсов. После изменения параметров необходимо нажать «Применить».

Обновить архитектуру

«Обновить архитектуру» позволяет вывести в правую часть экрана все параметры заданной нейронной сети (рисунок 16).

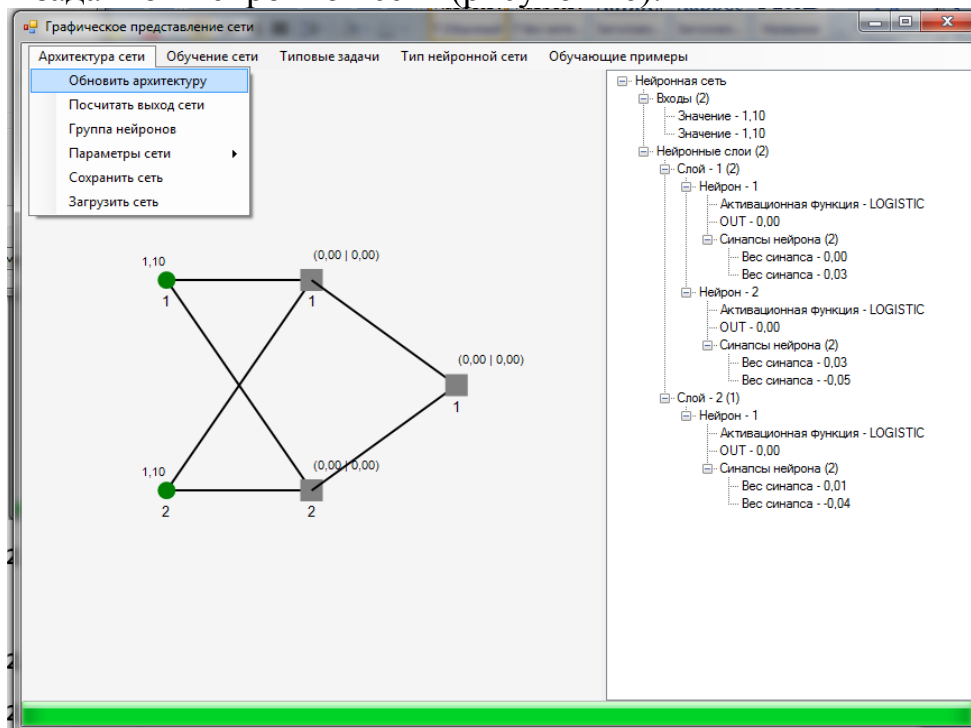


Рисунок 16 – Представление сети

Посчитать выход сети

«Посчитать выход сети» (рисунок 17) позволяет получить выходы сети при данных параметрах сети. Входные параметры можно задать в пункте «Параметры сети».

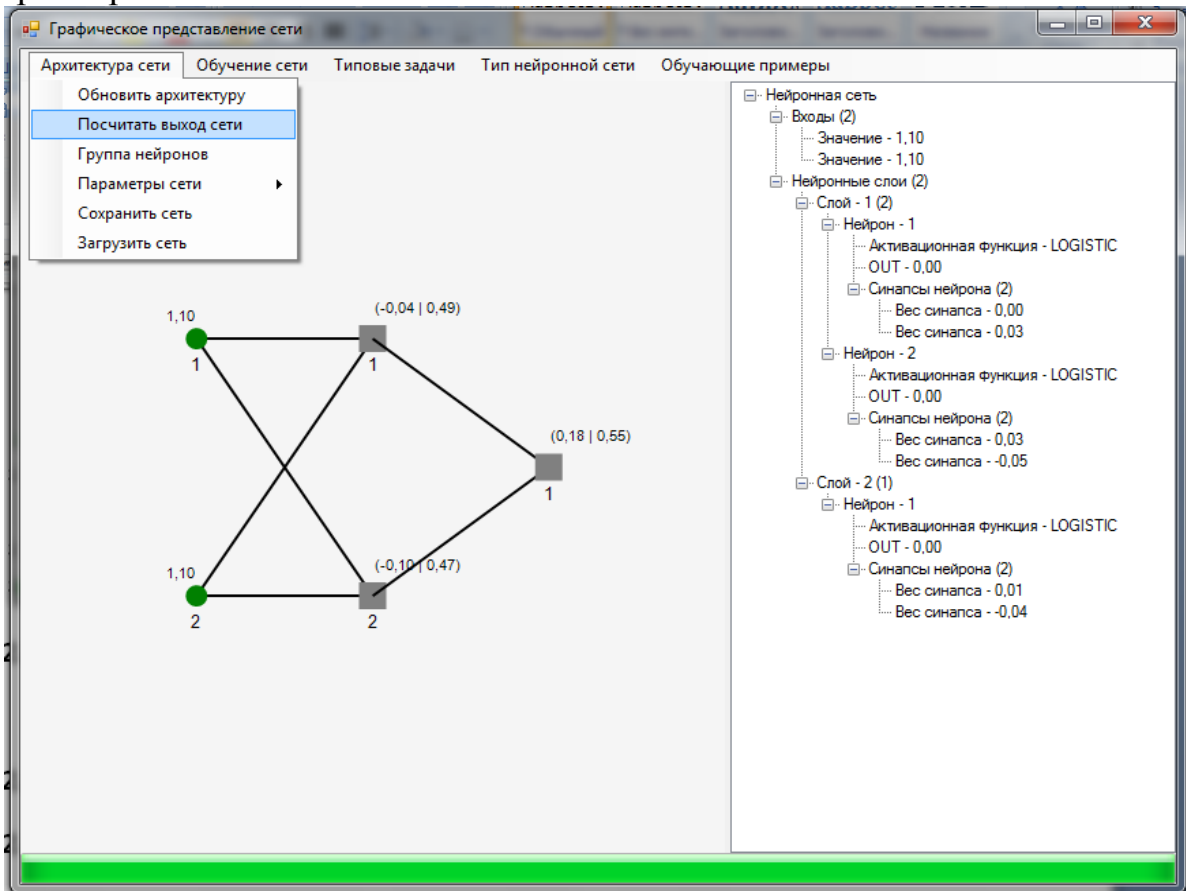


Рисунок 17 – Выбор пункта «Посчитать выход сети»

Группа нейронов

«Группа нейронов» позволяет посмотреть список возможных функций активации для нейронов (рисунок 18).

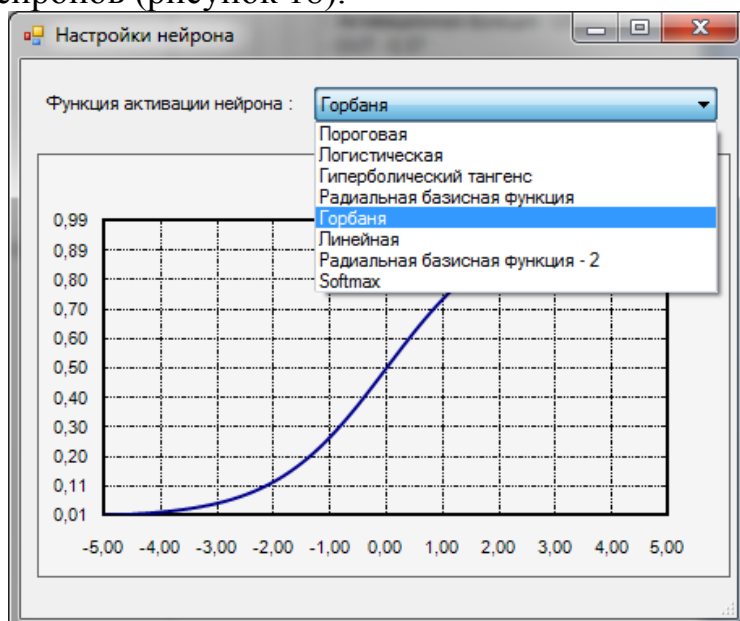
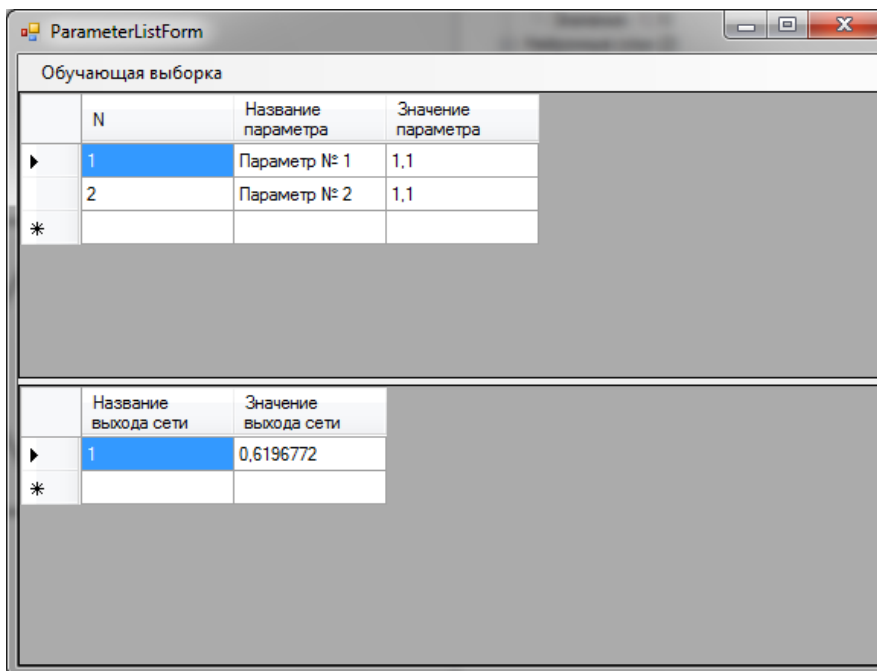


Рисунок 18 – Меню настройки нейрона

Параметры сети

«Параметры сети» (рисунок 19) позволяет задать входные данные для сети. В верхней половине окна задаются данные для входов сети. В нижней половине – видим результаты работы сети.

Примечание: результаты работы сети видны после выполнения пункта «Посчитать выход сети».



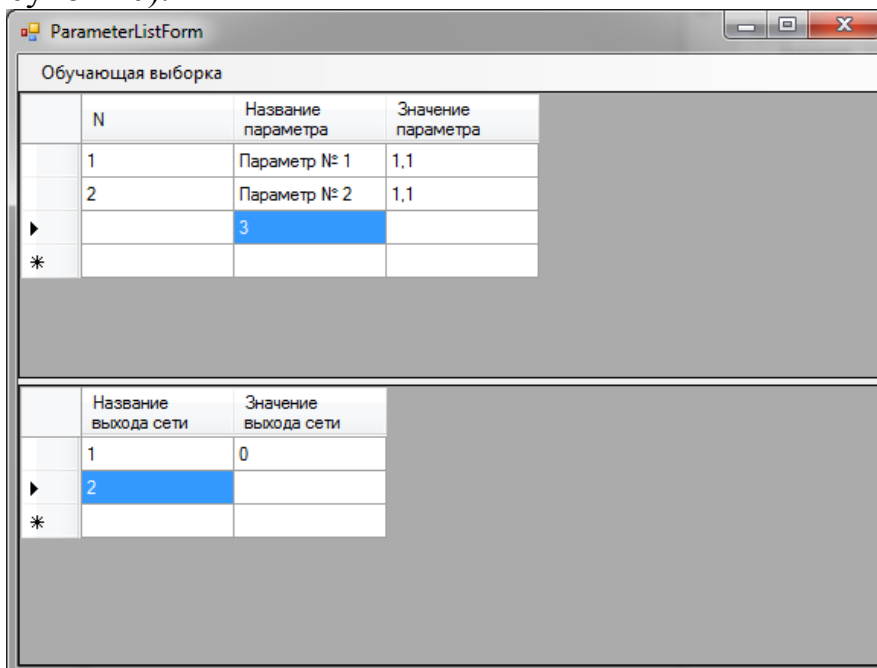
The screenshot shows a window titled "ParameterListForm" with a tab labeled "Обучающая выборка". It contains two tables. The top table lists parameters with columns "N", "Название параметра", and "Значение параметра". The bottom table lists network output results with columns "Название выхода сети" and "Значение выхода сети".

N	Название параметра	Значение параметра
1	Параметр № 1	1,1
2	Параметр № 2	1,1
*		

Название выхода сети	Значение выхода сети
1	0,6196772
*	

Рисунок 19 – Меню вывода параметров

Количество входных и выходных нейронов может быть изменено, если сделать нажатие мышью в зоне столбца с названием параметра и введением названия (рисунок 20).



The screenshot shows the same "ParameterListForm" window, but with modifications. The top table now has three parameters, and the bottom table has two output results.

N	Название параметра	Значение параметра
1	Параметр № 1	1,1
2	Параметр № 2	1,1
3		
*		

Название выхода сети	Значение выхода сети
1	0
2	
*	

Рисунок 20 – Изменение параметров

Узлы добавятся, сеть перестроится (рисунок 21).

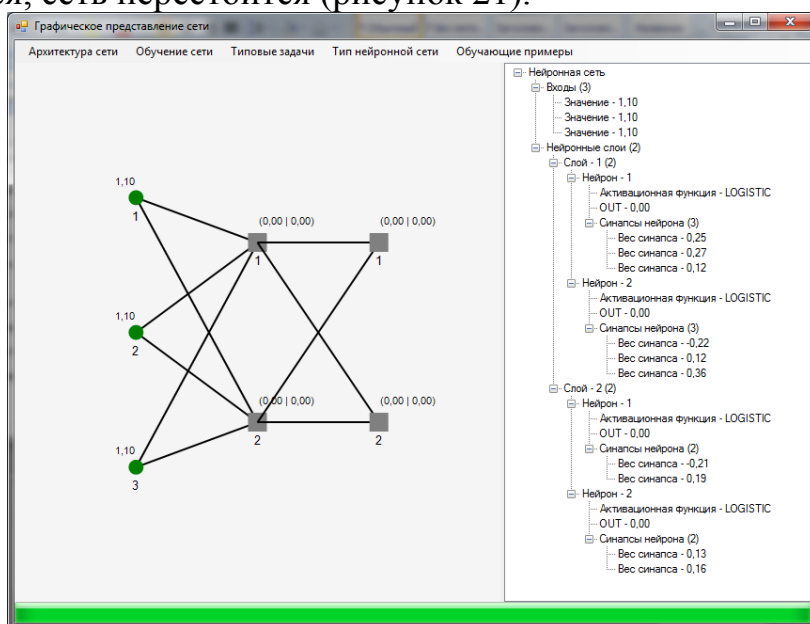


Рисунок 21 – Обновлённая сеть

Сохранить сеть

«Сохранить сеть» (рисунок 22) вызывает меню для сохранения структуры и параметров нейронной сети, с которой осуществляется работа. От пользователя требуется указать каталог и имя для сохранения файла. Формат файлов отсутствует.

Примечание: в названии лучше указывать тип нейронной сети.

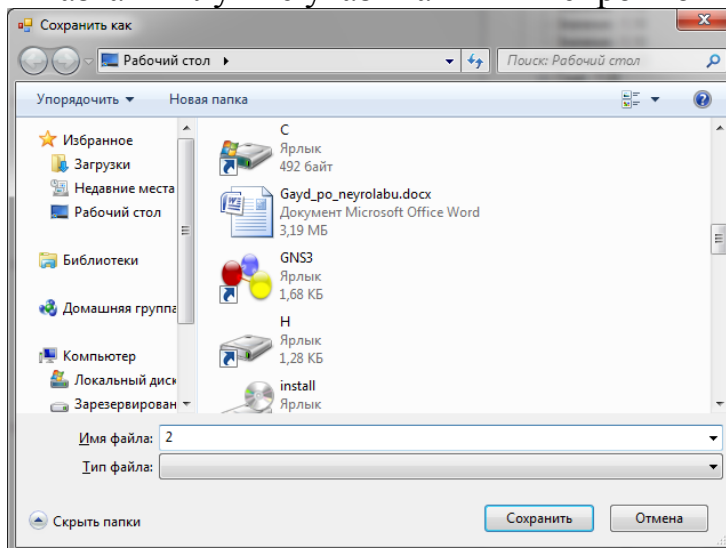


Рисунок 22 – Сохранение сети

Загрузить сеть

«Загрузить сеть» вызывает меню для загрузки структуры и параметров нейронной сети из файла. От пользователя требуется указать каталог и имя для загрузки файлов. Формат файлов отсутствует.

Примечание: перед загрузкой в Neuro-Lab необходимо выбрать тип нейронной сети, которая будет загружена, или выбрать соответствующую задачу.

Действия мыши

- 1 Кручение колёсика мыши при нахождении в меню «Графическое представление сети». При кручении колёсика мышки и наведении курсора на один из слоёв нейронной сети можно менять количество входных и нейронов среднего слоя.

Примечание: не работает для выходного слоя.

- 2 При двойном нажатии на слое входных нейронов появляется меню, с помощью которого можно задать входное значение для данного нейрона.
- 3 При двойном нажатии на нейронах среднего слоя появляется меню выбора функции активации для данного нейрона.
- 4 Если навести курсор мыши на нейрон и удерживать левую кнопку мыши, то можно изменить графическое расположение нейрона.

3 Типовые задачи

3.1 Обучающие пары

«Обучающие пары» позволяют обучать сеть с помощью заданных пользователем обучающих пар. Задаем их количество. Обучающая пара имеет следующий синтаксис:

- Все числа задаются в диапазоне от 0 до 1;
- Выход1|Выход2|...|Вход1; Вход2;...

Ниже приведён пример обучающих пар для решения задачи XOR:

Задаём 4 обучающие пары:

0,00|0,00;0,00;

1,00|0,00;1,00;

0,00|1,00;1,00;

1,00|1,00;0,00;

Для задания случайных весов синапсов нужно нажать «Обучение сети->Случайные веса синапсов» (рисунок 23).

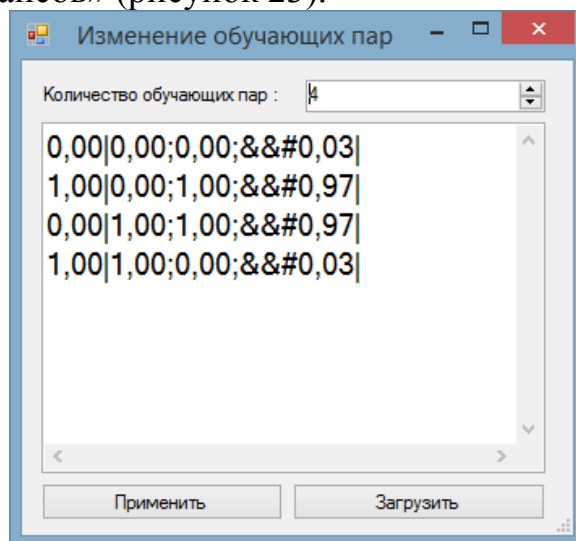


Рисунок 23 – Список обучающих пар

График обучения сети представлен на рисунке 24

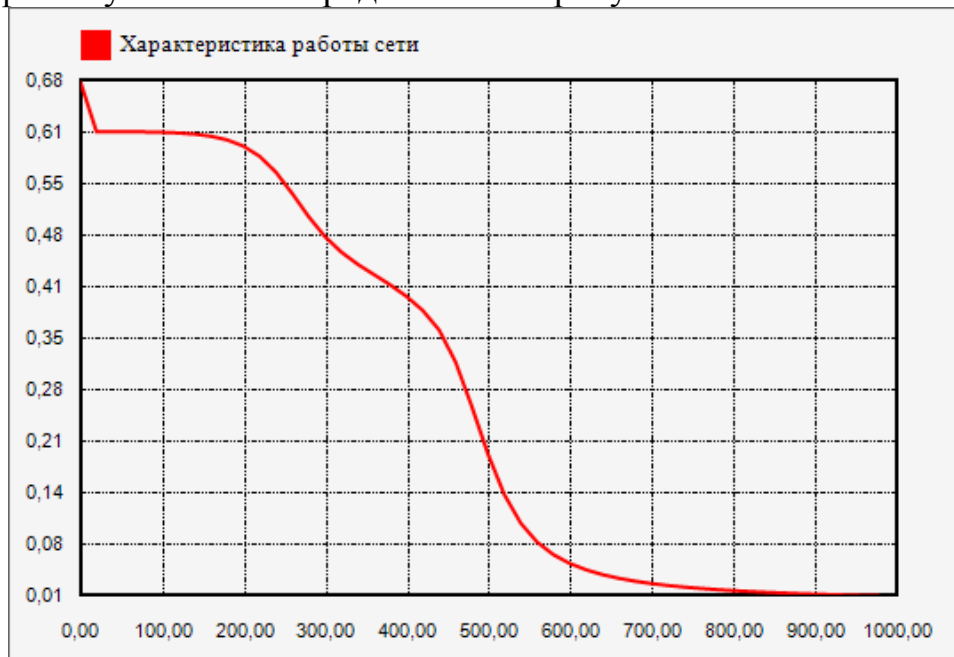


Рисунок 24 – График обучения сети

Перцептронная представляемость

«Перцептронная представляемость» – способность перцептрона (или другой сети) моделировать определенную функцию (рисунок 25)

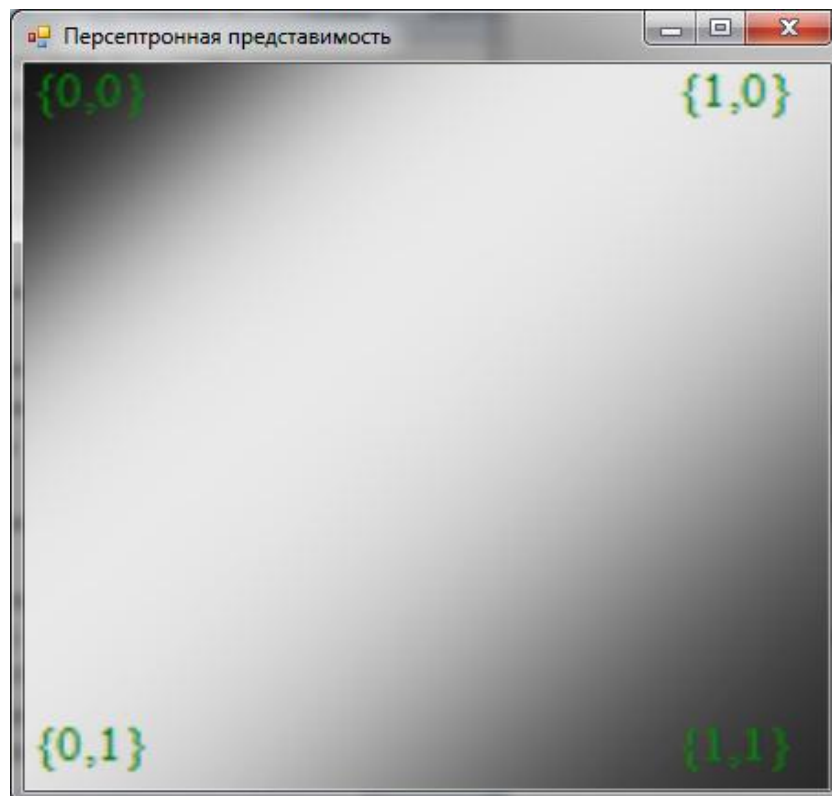


Рисунок 25 – Перцептронная представляемость

Чем темнее, тем решение ближе к 0, чем светлее – к 1.

В данном случае установлена функция активации – сигмоида, поэтому границы размыты.

Расписание обучения

Данный пункт (рисунок 26) выполнен только с ознакомительной целью. Изменить данные можно, но применить нельзя.

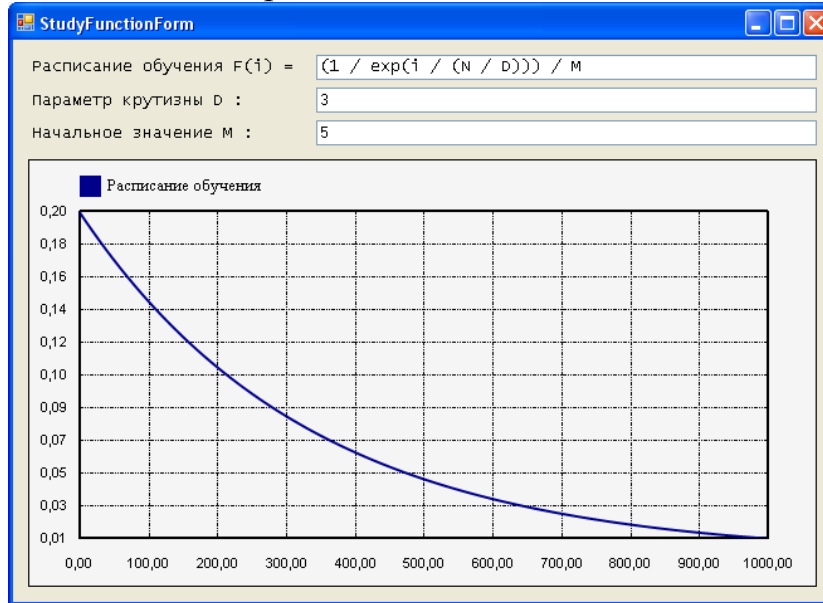


Рисунок 26 – Расписание обучения

3.2 Распознавание образов

ЛКМ – добавление точки.

ПКМ – удаление точки.

Для добавления образа нужно задать образ и нажать «Образ -> Добавить». После добавления нескольких образов (рисунок 27) можно увидеть такую картинку:

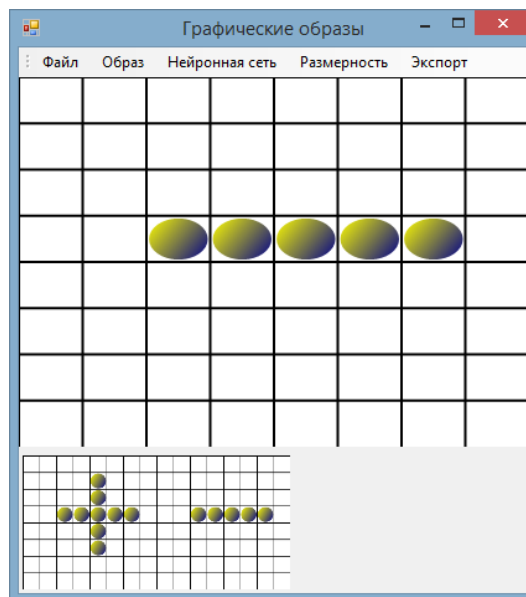


Рисунок 27 – Графическое представление образов

Далее необходимо построить и обучить сеть. Для этого нажать «Нейронная сеть -> Построить и Нейронная сеть -> Обучить» (рисунок 28). Заметим, что структура нейронной сети изменилась.

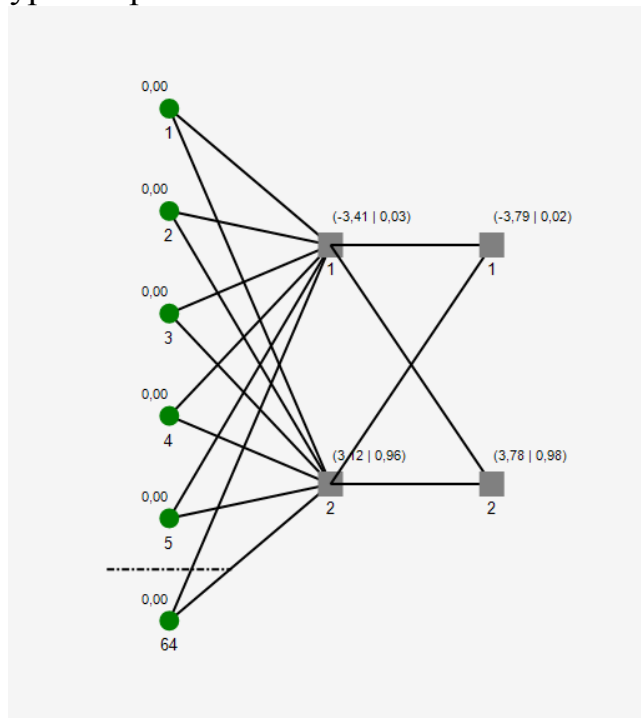


Рисунок 28 – Обучение сети

Если ошибка (рисунок 29) стремится к нулю, значит обучение прошло успешно.

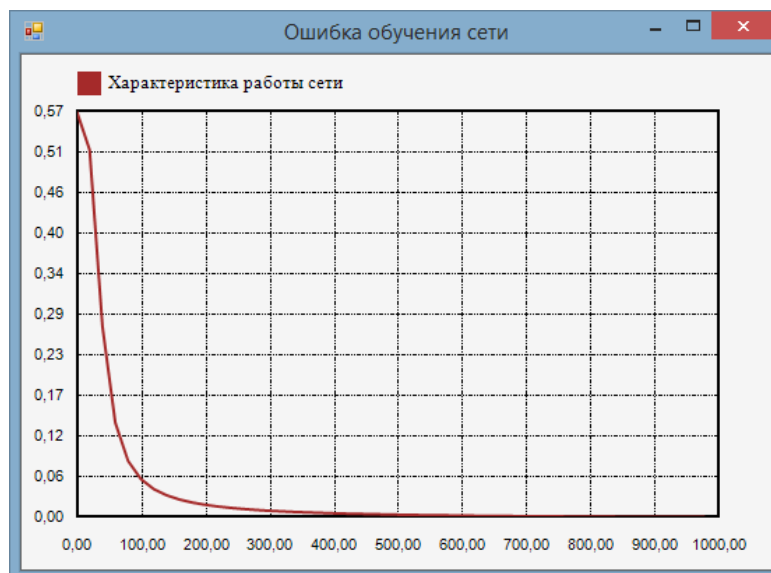


Рисунок 29 – Ошибка

Примечание: после обучения добавление новых образов невозможно. Далее задаем распознаваемый образ и нажимаем «Нейронная сеть -> Распознать образ» (рисунок 30).

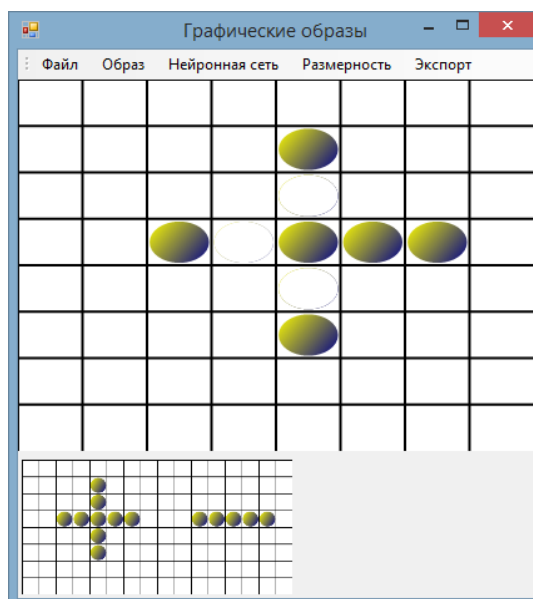


Рисунок 30 – Графическое представление образов

Получаем ответ в виде окна (рисунок 31).

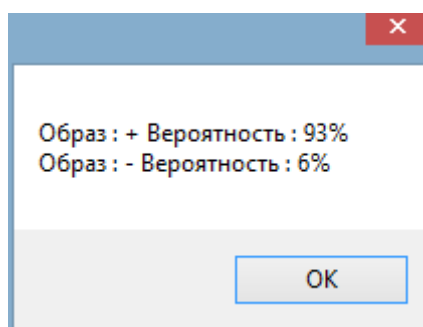


Рисунок 31 – Вероятность распознавания образа

Примечание: для корректного сохранения образов необходимо построить и обучить сеть.

3.3 Классификация объектов

ЛКМ – добавление точки.

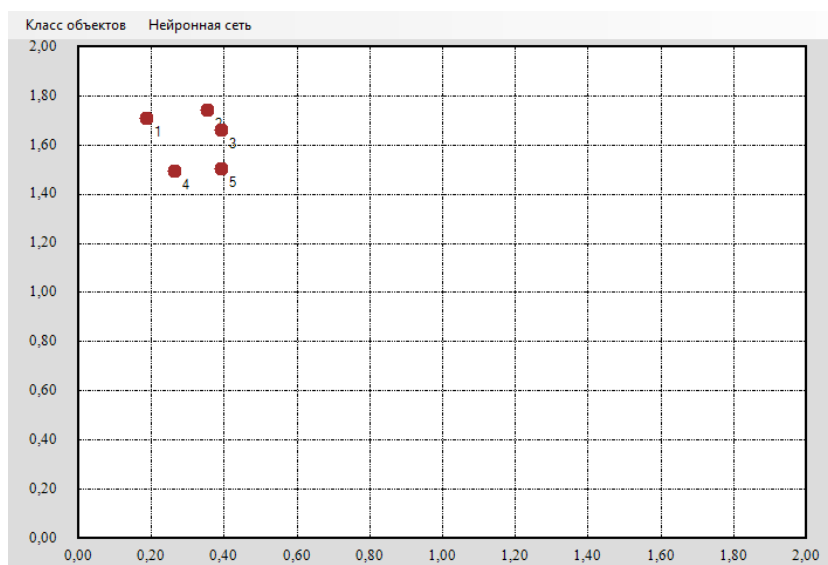


Рисунок 32 – Группа точек

Когда группа точек (рисунок 32) введена, нажать «Класс объектов -> Добавить». Введённые точки объединятся в один класс (изменяют цвет), после чего можно вводить следующий класс.

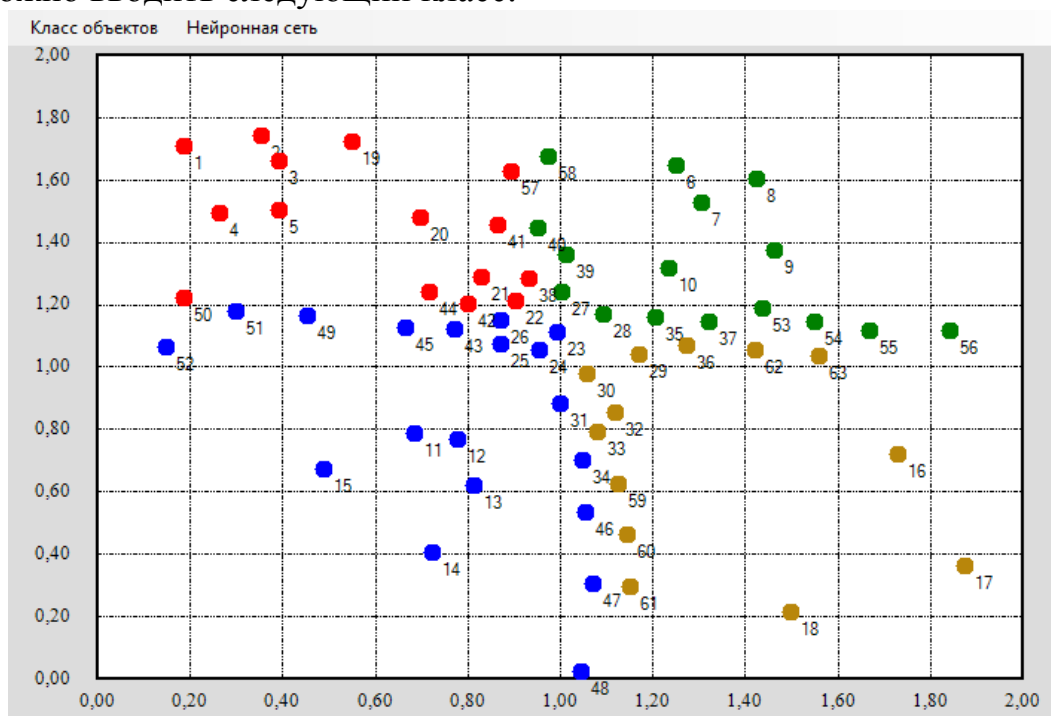


Рисунок 33 – Все группы точек

Нажимаем «Нейронная сеть -> Классифицировать» и добавляем точки. Их цвет будет соответствовать классу, в который они попадают (рисунок 33).

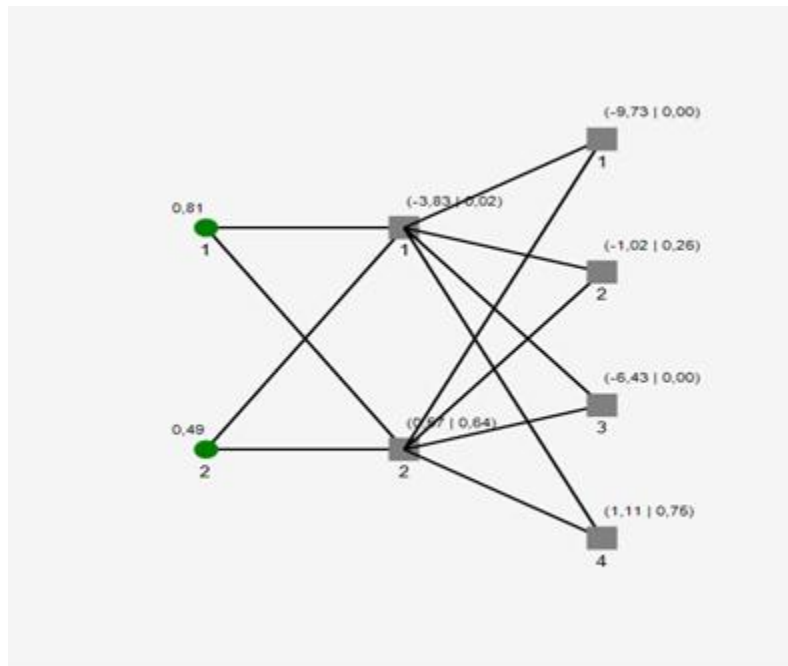


Рисунок 34 – Новая нейронная сеть

Когда все классы заданы, нужно нажать «Нейронная сеть -> Построить и Нейронная сеть -> Обучить». Структура сети – на рисунке 34.

3.4 Регрессия функции одной переменной

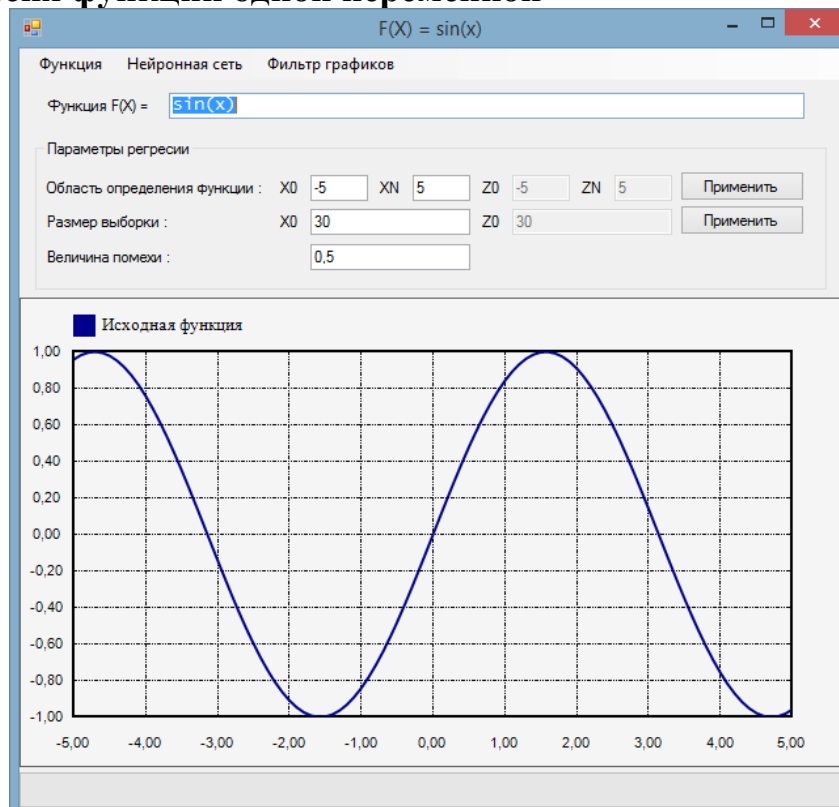


Рисунок 35 – Построение функции

Функция $F(X)$ – исходная функция. После ввода функции нужно нажать ENTER. С помощью фильтра графиков можно включить или выключить отображение графиков функции, решения или шума. Кнопки «Применить»

используются для задания области определения, размера выборки и помехи (рисунок 35).

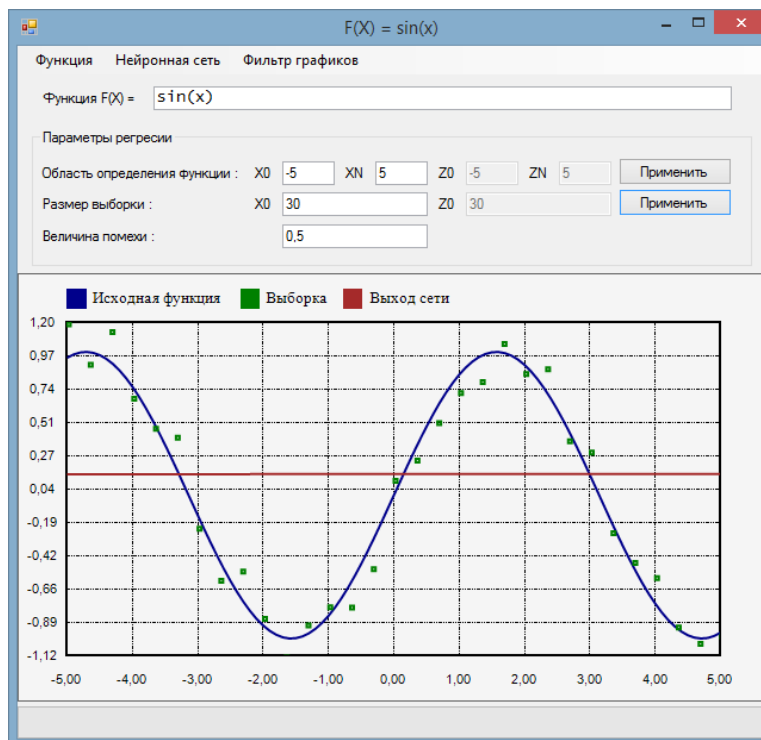


Рисунок 36 – Построение нейронной сети

После ввода исходных данных нажать «Нейронная сеть -> Построить» (рисунок 36). После нажатия «Нейронная сеть -> Обучить» — получаем результат. После ввода исходных данных нажать «Нейронная сеть -> Построить» (рисунок 37).

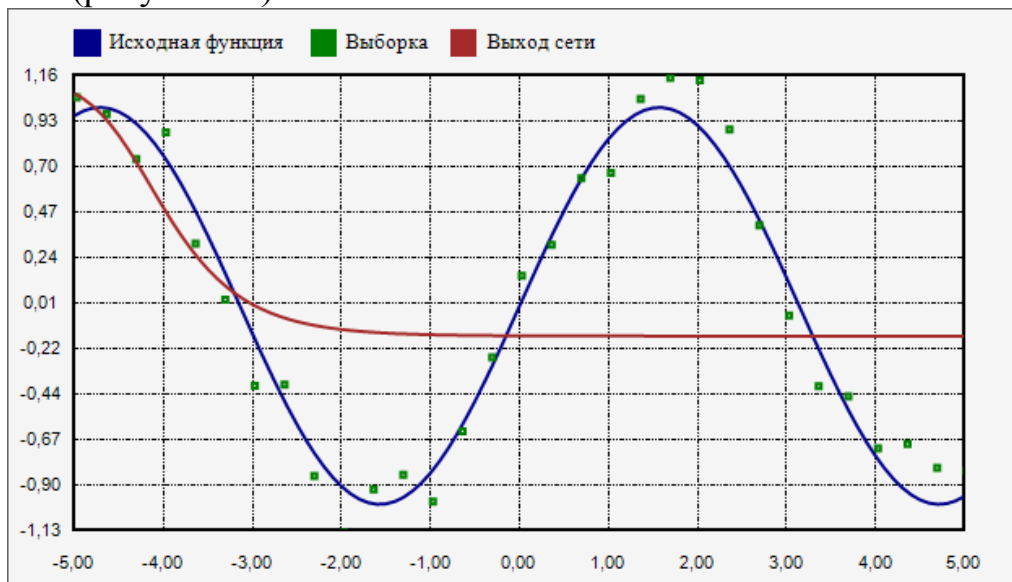


Рисунок 37 – Обученная сеть

Скорее всего, нейронная сеть не сможет решить задачу при стандартных параметрах сети. Однако если увеличить количество нейронов, сеть станет

более чувствительной и будет лучше решать задачу. После ввода исходных данных нажать «Нейронная сеть -> Построить» (рисунок 38).

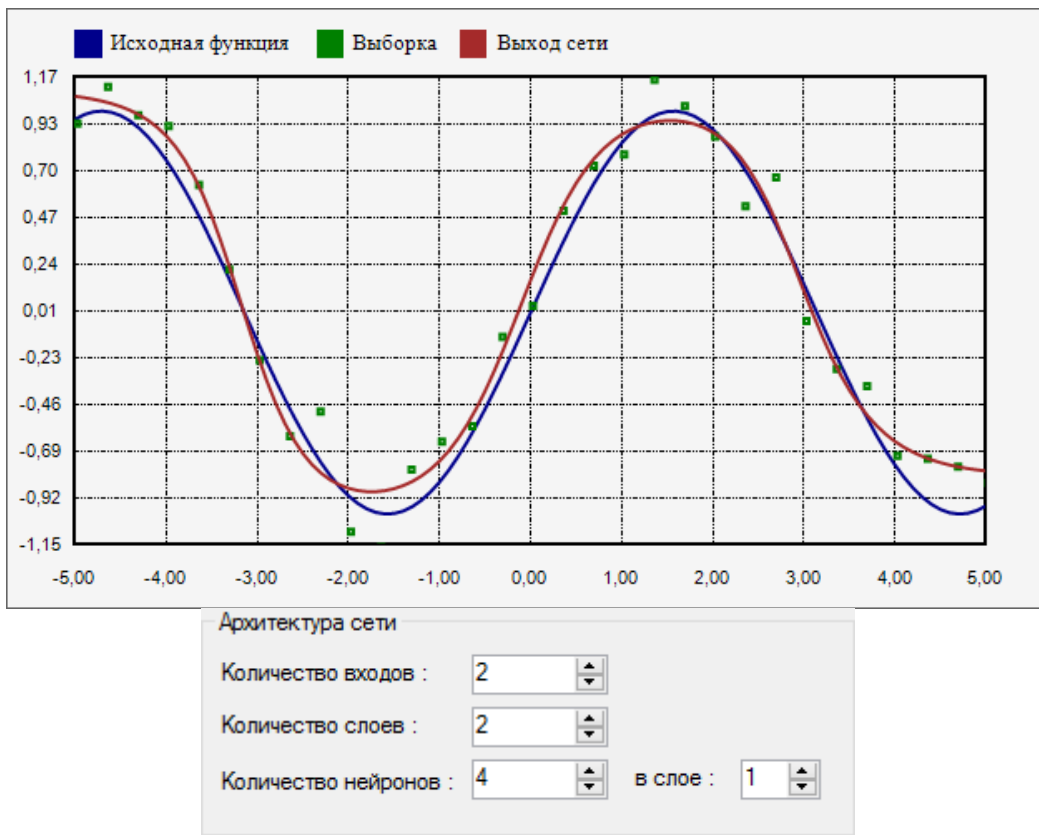


Рисунок 38 – Параметры сети

Примечание. для RBF сетей количество слоёв менять нельзя.

3.5 Регрессия функции двух переменных

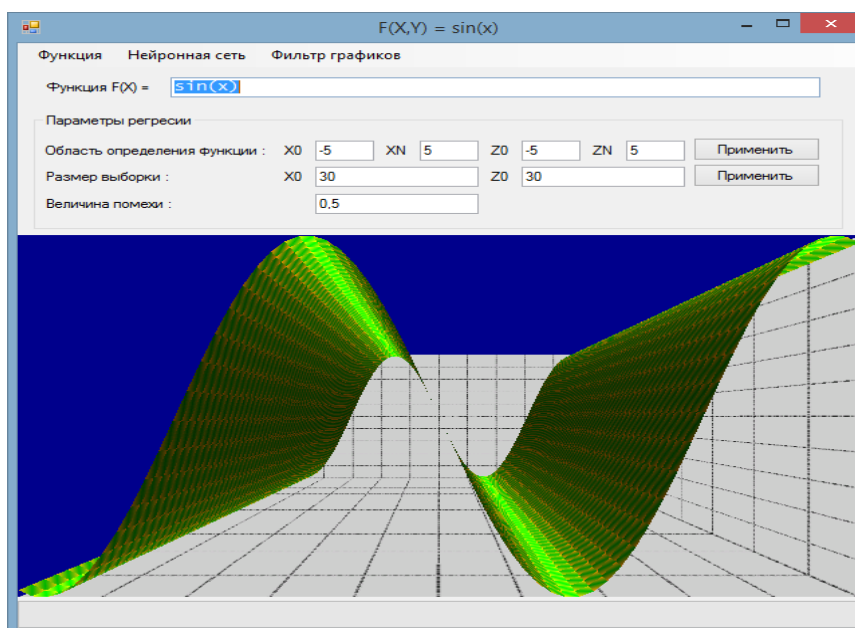


Рисунок 39 – Построение функции

Функция $F(X)$ – исходная функция. После ввода функции нужно нажать ENTER. Кнопки «Применить» используются для задания области определения размера выборки и помехи. После ввода исходных данных нажать «Нейронная сеть -> Построить» (рисунок 39).

После ввода исходных данных нажать «Нейронная сеть -> Построить». После ввода исходных данных нажать «Нейронная сеть -> Построить» (рисунок 40).

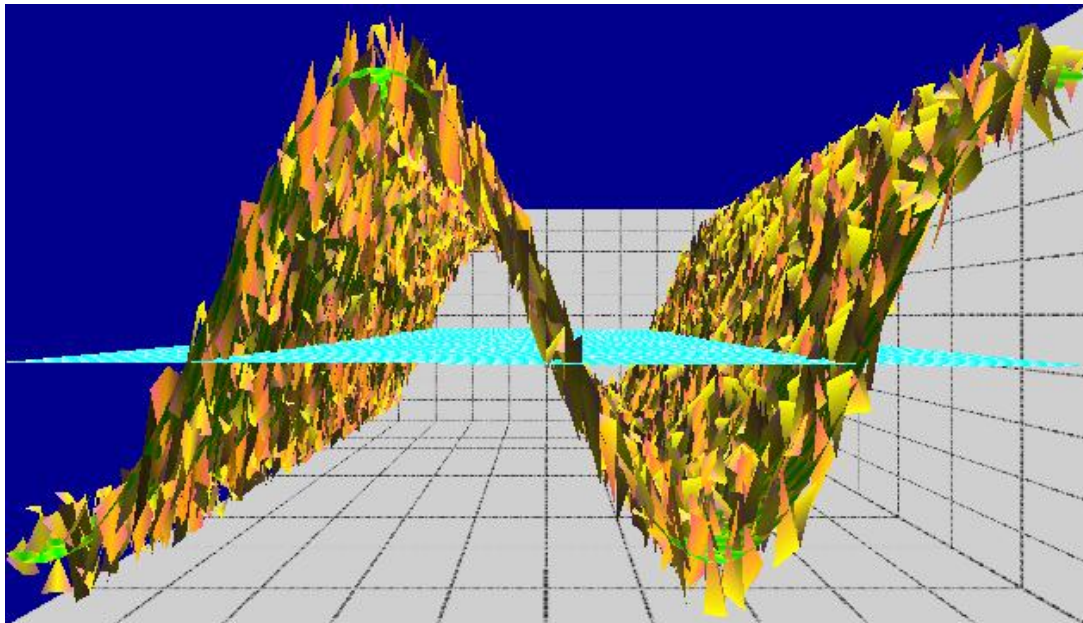


Рисунок 40 – Построение нейронной сети

После нажатия «Нейронная сеть -> Обучить» — получаем результат (рисунок 41).

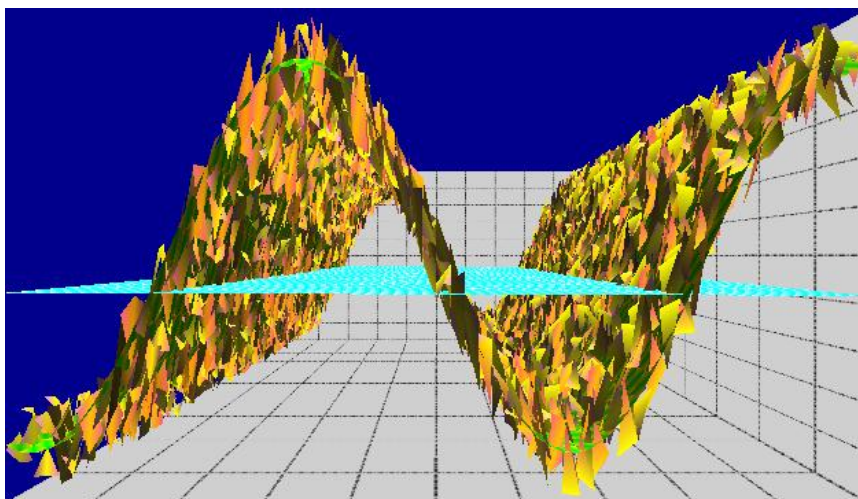


Рисунок 41 – Обученная сеть

Скорее всего, нейронная сеть не сможет решить задачу при стандартных параметрах сети. Однако если увеличить количество нейронов, сеть будет более чувствительной и станет лучше решать задачу (рисунок 42).

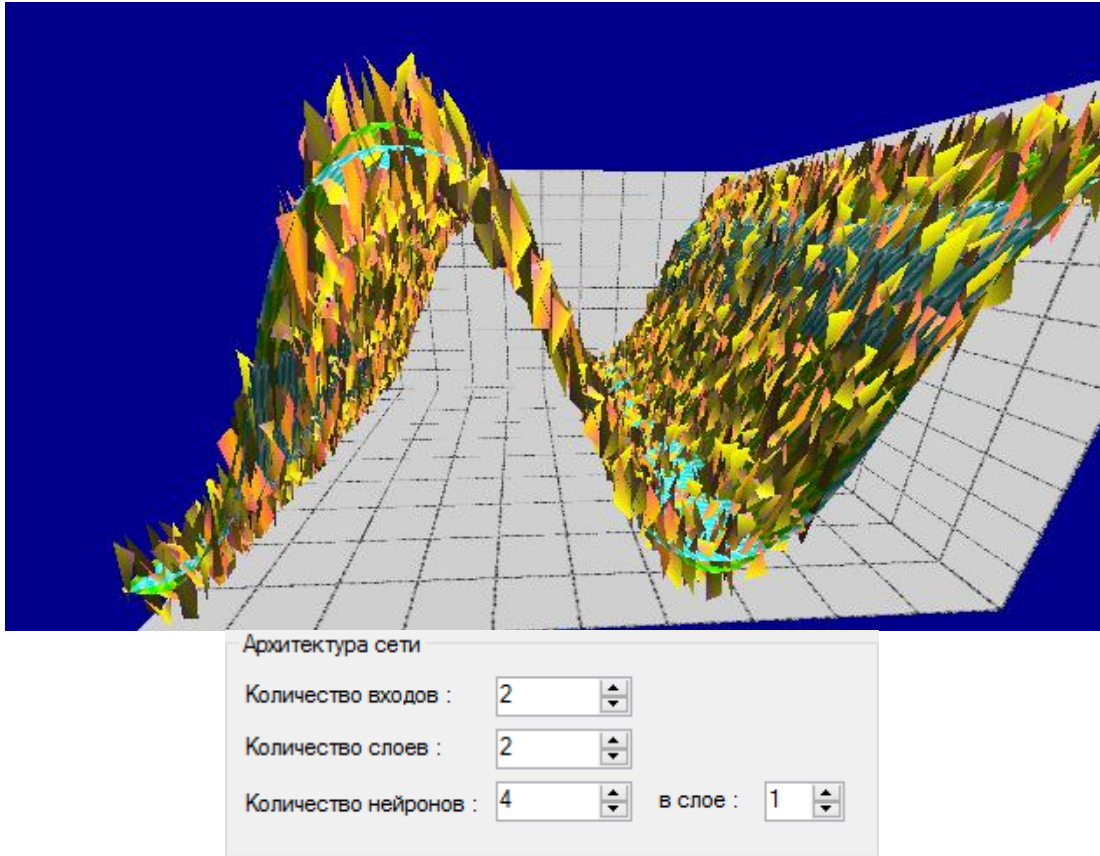


Рисунок 42 – Параметры сети

Примечание. для RBF сетей количество слоёв менять нельзя.

3.6 Кластер-анализ

ЛКМ – добавление точки.

Количество кластеров, на которые программа сможет разделить данные для кластеризации, равно количеству выходных нейронов.

Примечание: чтобы использовать вновь добавленные нейроны или найти новые кластеры, нужно сначала нажать кнопку «Применить» в окне «Архитектура сети», затем, в окне ввода данных для кластеризации – «Нейронная сеть ->Построить» и «Нейронная сеть ->Обучить».

Пример

Возьмём 3 выходных нейрона и введём данные так, что их можно разделить на 2 кластера. И нажмём «Нейронная сеть ->Построить» (рисунок 43).

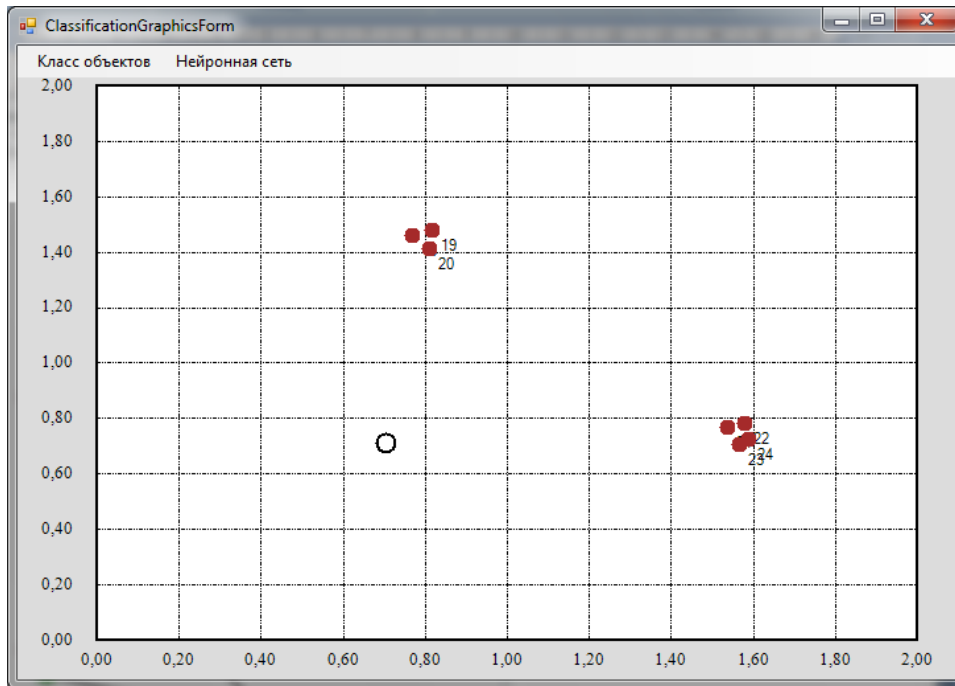


Рисунок 43 – Входные нейроны

Далее нажать «Нейронная сеть ->Обучить» (рисунок 44).

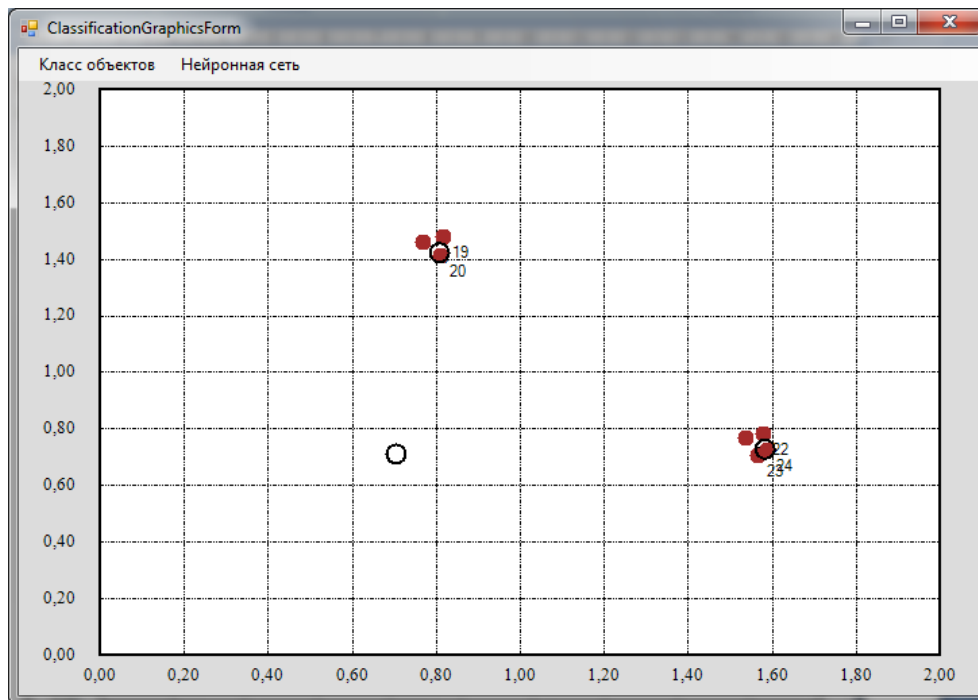


Рисунок 44 – Обученная сеть

Видим, что данные успешно разделились на 2 кластера. Добавим данные и обучим сеть (рисунок 45).

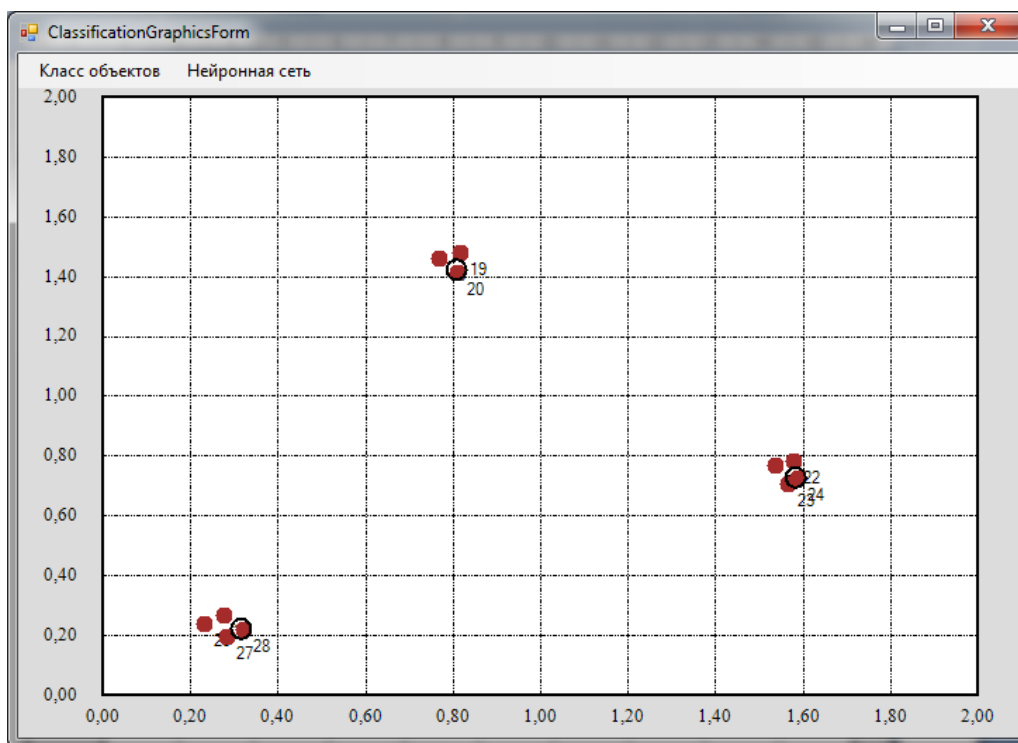


Рисунок 45 – Добавление данных

Ещё один кластер успешно найден. Добавим данные, которые можно выделить в отдельный кластер, и попробуем обучить сеть (рисунок 46).

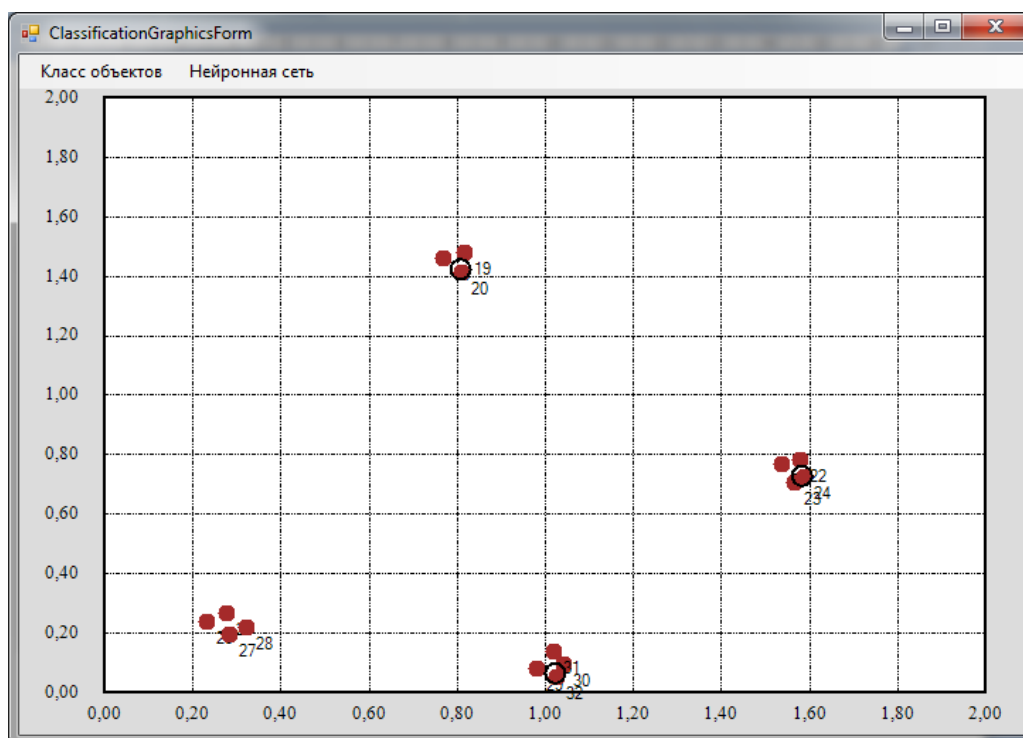


Рисунок 46 – Обученная сеть

Сеть находит только 3 кластера, т.к. она имеет 3 нейрона. Добавим ещё один выходной нейрон и попробуем обучить сеть (рисунок 47).

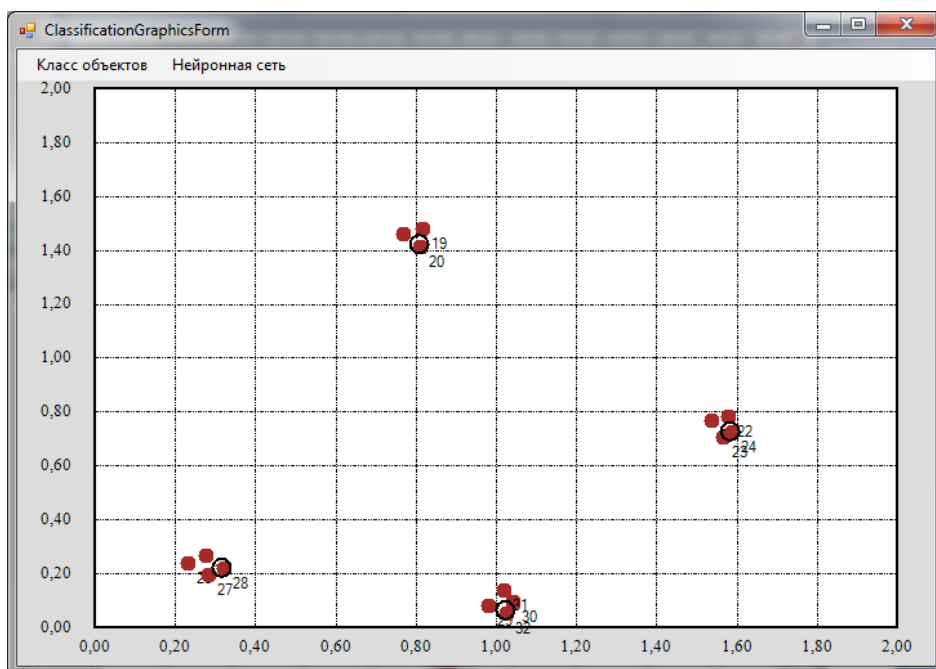


Рисунок 47 – Добавление данных

Все кластеры успешно выделены.

При больших количествах входных данных иногда требуется несколько раз выполнить нажатие «Нейронная сеть -> Обучить». Ниже приведён пример для 13 нейронов и 10 возможных кластеров (рисунок 48).

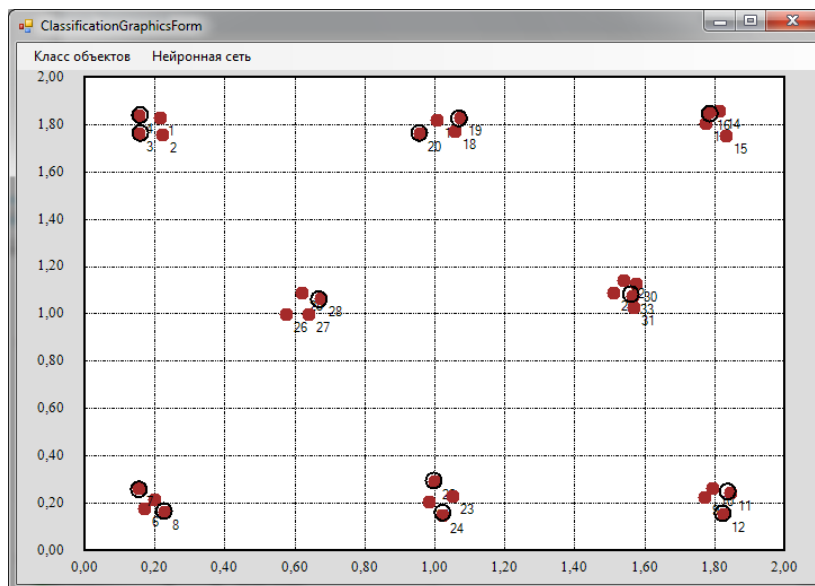


Рисунок 48 – Обученная сеть

Если ввести данные так, что их можно четко разделить на несколько кластеров, а количество нейронов задать в 2 раза больше, чем предполагаемых кластеров, то возможно выделение дополнительных кластеров. Это становится заметным после нескольких итераций обучения.

Пример

Зададим данные для 4 возможных кластеров и возьмём 8 нейронов. После первой итерации выделилось 4 кластера (рисунок 49).

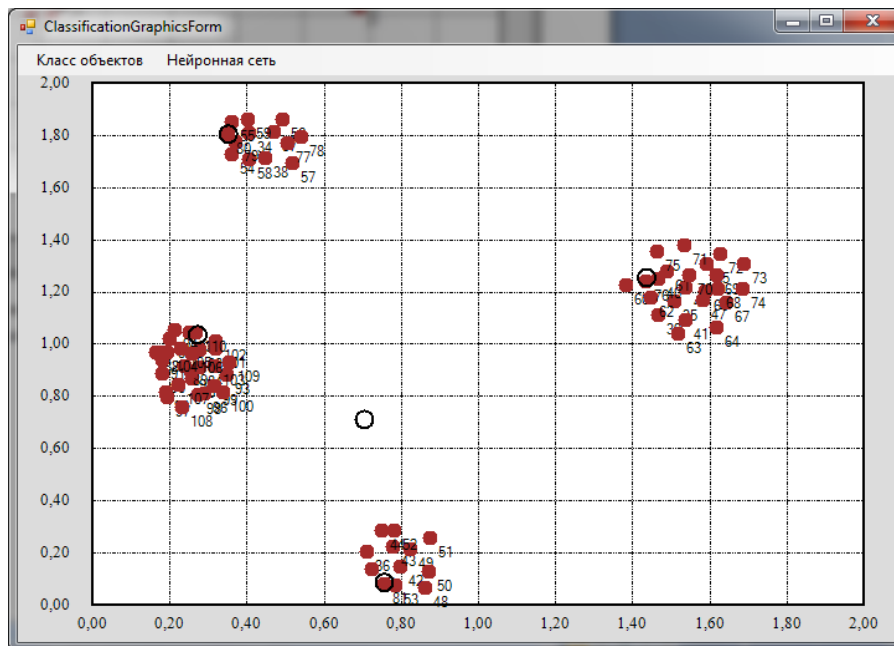


Рисунок 49 – Выделенные кластеры

После следующей итерации выделяются ещё 4 кластера (рисунок 50).

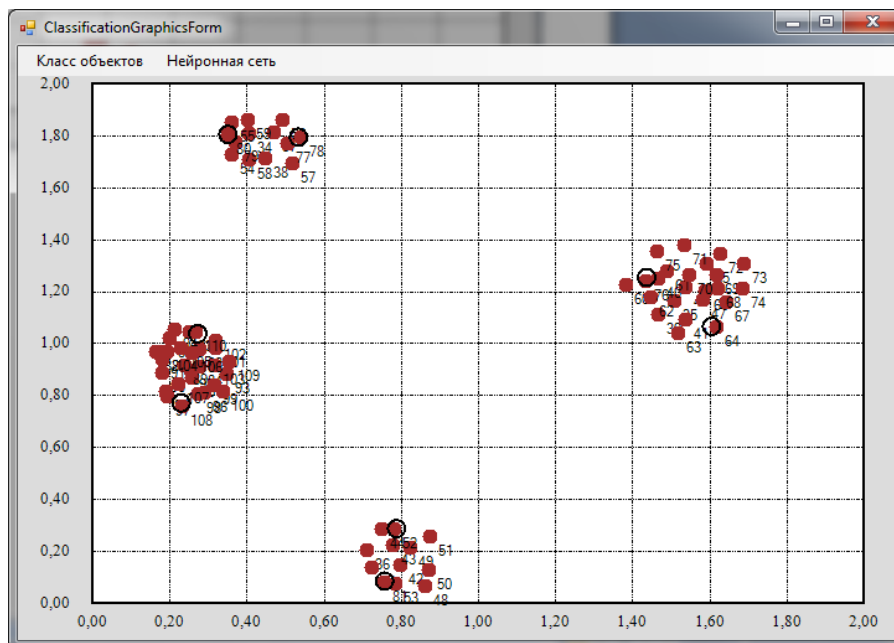


Рисунок 50 – Обновлённые данные

3.7 Восстановление образов

Сначала нужно задать размерность поля для задания образов. Это делается нажатием вкладки «Размерность» и выбором нужного действия: уменьшить или увеличить.

ЛКМ – добавление точки.

ПКМ – удаление точки.

Для добавления образа нужно задать образ с помощью мыши и нажать «Образ -> Добавить» (рисунок 51). Можно задать имена для образов.

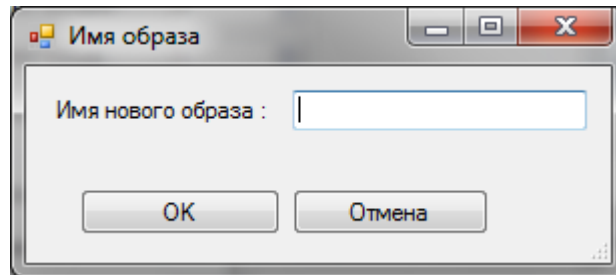


Рисунок 51 – Добавление образа

После добавления нескольких образов можно увидеть картинку (рисунок 52).

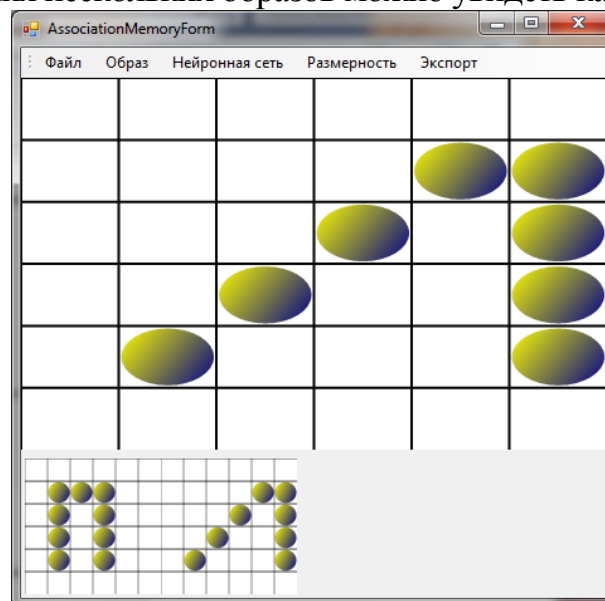


Рисунок 52 – Добавленные образы

Далее необходимо построить и обучить сеть. Для этого нажать «Нейронная сеть -> Построить» и «Нейронная сеть -> Обучить».

Заметим, что структура нейронной сети изменилась (рисунок 53).

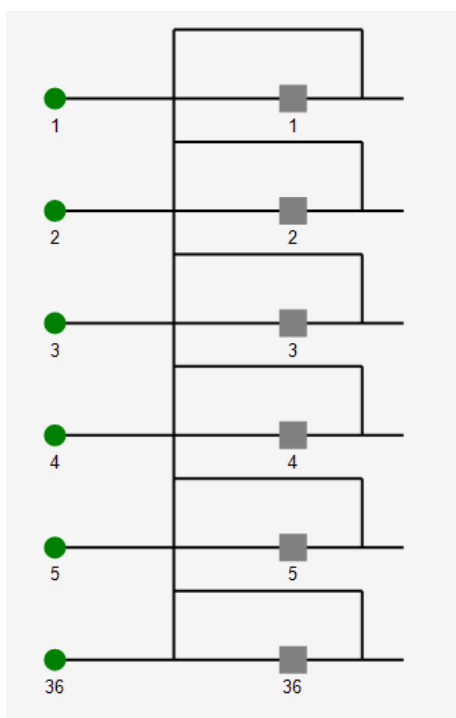


Рисунок 53 – Обновлённая структура нейронной сети

Далее задаем распознаваемый образ и нажать «Нейронная сеть -> Распознать образ» (рисунок 54).

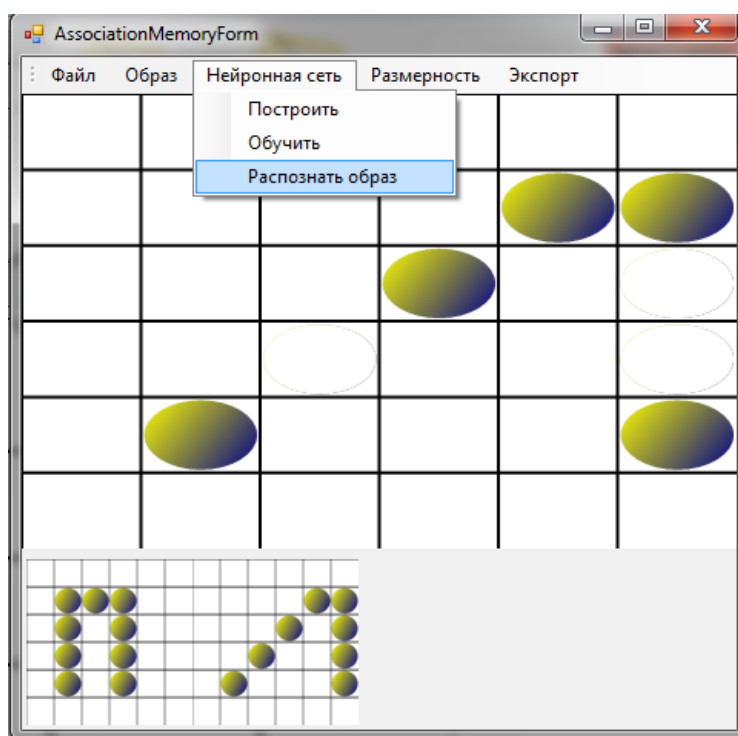


Рисунок 54 – Выполнение операции распознавания сети

Получаем ответ в виде восстановленного образа (рисунок 55).

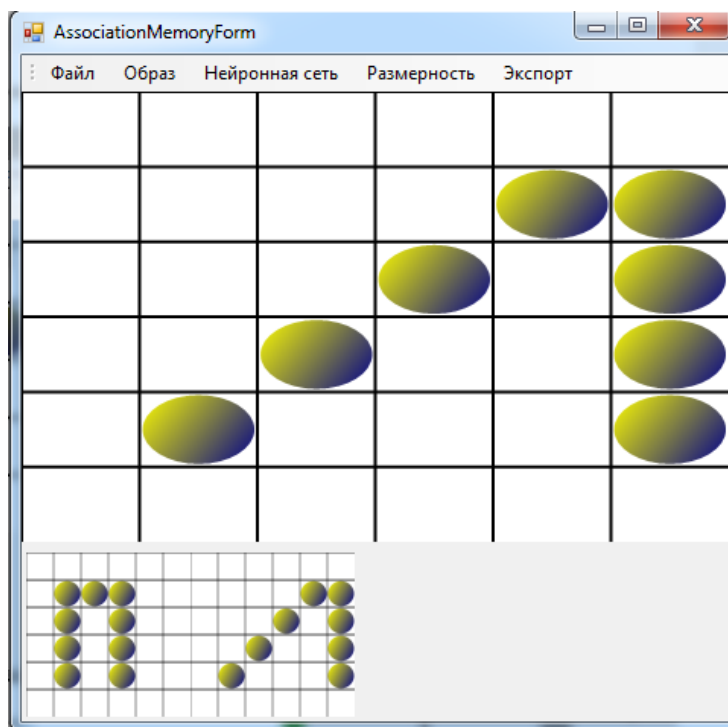


Рисунок 55 – Восстановленный образ

3.8 Системы линейных уравнений

Зададим систему уравнений (рисунок 56):

$$3x_1 + 5x_2 = 35;$$

$$5x_1 + 1x_2 = 29.$$

	№	X1	X2	=	B
	1	3	5	=	35
▶	2	5	1	=	29
*					

Рисунок 56 – Система уравнений

Нажимаем «Нейронная сеть -> Построить сеть», задаём ошибку, максимальное количество эпох и нажимаем кнопку «Решить задачу». Получаем ответ в виде графиков и таблицы (рисунки 57, 58).

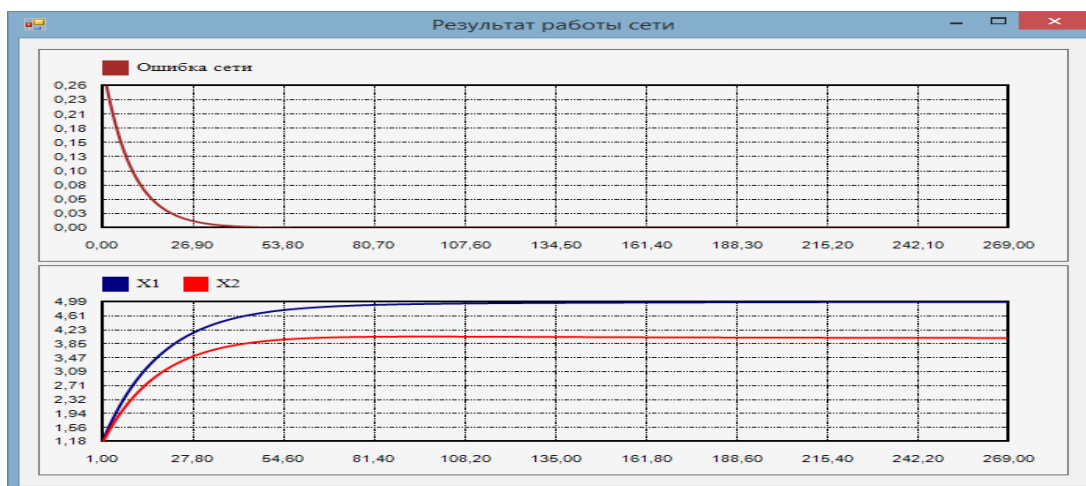


Рисунок 57 – Результат работы сети

	Тип	X1	X2
▶	Вход	1	1
	Выход	4,9929...	4,0085...
	1	1,224	1,1805...
	2	1,4351...	1,3506...

Рисунок 58 – Решение системы уравнений

4 Теоретическая часть

Перечислим некоторые понятия, решаемые в контексте ИНС и представляющие интерес для студентов.

Классификация образов. Задача состоит в указании принадлежности входного образа (например, речевого сигнала или рукописного символа), представленного вектором признаков к одному из нескольких предварительно определенных классов. К известным приложениям относятся распознавание букв, распознавание речи, классификация сигнала электрокардиограммы, классификация клеток крови.

Кластеризация/категоризация. При решении задачи кластеризации, которая известна также как классификация образов «без учителя», отсутствует обучающая выборка с метками классов. Алгоритм кластеризации основан на подобию образов и размещает близкие образы в один кластер. Известны случаи применения кластеризации для извлечения знаний, сжатия данных и исследования свойств данных.

Аппроксимация функций. Предположим, что имеется обучающая выборка $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$ (пары данных вход-выход), которая генерируется неизвестной функцией (x) , искаженной шумом. Задача аппроксимации состоит в нахождении оценки неизвестной функции (x) . Аппроксимация функций необходима при решении многочисленных инженерных и научных задач моделирования.

Предсказание/прогноз. Пусть заданы n дискретных отсчетов $\{y(t_1), y(t_2), \dots, y(t_n)\}$ в последовательные моменты времени t_1, t_2, \dots, t_n . Задача состоит в предсказании значения $y(t_{n+1})$ в некоторый будущий момент времени t_{n+1} . Предсказание/прогноз имеют значительное влияние на принятие решений в бизнесе, науке и технике. Предсказание цен на фондовой бирже и прогноз погоды являются типичными приложениями техники предсказания/прогноза.

Оптимизация. Многочисленные проблемы в математике, статистике, технике, науке, медицине и экономике могут рассматриваться как проблемы оптимизации. Задачей алгоритма оптимизации является нахождение такого решения, которое удовлетворяет системе ограничений и максимизирует или минимизирует целевую функцию. Задача коммивояжера, относящаяся к классу NP-полных, является классическим примером задачи оптимизации.

Память, адресуемая по содержанию. В модели вычислений фон Неймана обращение к памяти доступно только посредством указания адреса, который не зависит от содержания памяти. Если допущена ошибка в вычислении адреса, то может быть найдена совершенно иная информация. Ассоциативная память, или память, адресуемая по содержанию, доступна по указанию заданного содержания. Содержимое памяти может быть вызвано даже по частичному входу или искаженному содержанию. Ассоциативная память применяется, например, при создании мультимедийных информационных баз данных.

Управление. Рассмотрим динамическую систему, заданную совокупностью $\{u(t), y(t)\}$, где $u(t)$ – является входным управляющим воздействием, а $y(t)$ – выходом системы в момент времени t . В системах управления с эталонной моделью целью управления является расчет такого входного воздействия $u(t)$, при котором система следует по желаемой траектории, диктуемой эталонной моделью. Примером является оптимальное управление двигателем.

4.1 Многослойный перцептрон

Многослойная ИНС. Особенностью является наличие более чем одного обучаемого слоя. Теоретически единственного скрытого слоя достаточно, чтобы перекодировать входное представление таким образом, чтобы получить линейную разделимость для выходного представления. Существует предположение, что, используя большее число слоёв, можно уменьшить число элементов в них, то есть суммарное число элементов в слоях будет меньше, чем если использовать один скрытый слой.

4.2 Сети RBF

Сети, использующие радиальную базисную функцию типа гауссовой. Радиальная базисная функция (функция ядра) центрируется в точке, которая определяется весовым вектором, связанным с нейроном. Как позиция, так и ширина функции ядра должны быть обучены по выборочным образцам. Обычно ядер гораздо меньше, чем обучающих примеров. Каждый выходной

элемент вычисляет линейную комбинацию этих радиальных базисных функций. С точки зрения задачи аппроксимации скрытые элементы формируют совокупность функций, которые образуют базисную систему для представления входных примеров в построенном на ней пространстве.

Существуют различные алгоритмы обучения RBF-сетей. Основным алгоритмом используется двухшаговую стратегию обучения, или смешанное обучение. Он оценивает позицию и ширину ядра с использованием алгоритма кластеризации.

4.3 Сети Кохонена

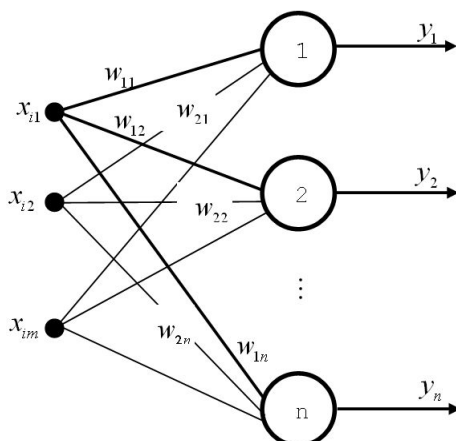


Рисунок 59 – Сеть Кохонена

Идея сети Кохонена (рисунок 59) принадлежит финскому ученому Тойво Кохонену (1982 год).

В основе идеи сети Кохонена лежит аналогия со свойствами человеческого мозга. Наиболее распространенное применение сетей Кохонена – решение задачи классификации без учителя, т. е. кластеризации.

Сеть Кохонена – это однослойная сеть, каждый нейрон которой соединен со всеми компонентами n -мерного входного вектора. Входной вектор – это описание одного из объектов, подлежащих кластеризации. Количество нейронов совпадает с количеством кластеров, которое должна выделить сеть.

Сеть Кохонена обучается методом последовательных приближений. В процессе обучения таких сетей на входы подаются данные, но сеть при этом подстраивается не под эталонное значение выхода, а под закономерности во входных данных. Начинается обучение с выбранного случайным образом выходного расположения центров.

В процессе последовательной подачи на вход сети обучающих примеров определяется наиболее схожий нейрон (тот, у которого скалярное произведение весов и поданного на вход вектора минимально). Этот нейрон объявляется победителем. Такое правило обучения является «соревновательным»; «победивший забирает всё».

Обучение при этом заключается не в минимизации ошибки, а в подстройке весов (внутренних параметров нейронной сети) для наибольшего совпадения с

входными данными. Веса организуются так, что топологически близкие узлы чувствительны к похожим входным сигналам.

В результате работы алгоритма центр кластера устанавливается в определенной позиции, удовлетворительным образом кластеризующей примеры, для которых данный нейрон является «победителем».

4.4 Сети Хопфилда

Возрождение интереса к нейронным сетям связано с работой Хопфилда (1982 г.). Эта работа пролила свет на то обстоятельство, что заимствованные из природы сети из нейроноподобных элементов с обратными связями могут быть использованы для вычислительных целей. Исследователи из многих областей знания получили стимул для дальнейших исследований этих сетей. Сети, которые работают как системы с обратными связями, называются **«рекуррентными сетями»**. Каждая прямая операция сети называется **итерацией**. Сеть с обратной связью формирует ассоциативную память. Сеть Хопфилда можно отнести к автоассоциативной памяти, то есть такой, которая может завершить или исправить образ, но не может ассоциировать полученный образ с другим образом.

Достижение устойчивого состояния не гарантирует правильный ответ сети. Это происходит из-за того, что сеть может сойтись к так называемым ложным аттракторам, иногда называемым «химерами» (как правило, химеры склеены из фрагментов различных образов).

Список литературы

- 1 Тархов Д. А. Нейронные сети. Модели и алгоритмы. Кн. 18. – Москва : Радиотехника, 2005. – 256 с. : ил.
- 2 Хайкин Саймон. Нейронные сети: полный курс, 2-е издание : пер. с англ. – Москва : Издательский дом «Вильямс», 2006. – 1104 с. : ил.
- 3 Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth. Neural Network Toolbox 7. User's Guide [Электронный ресурс].- Natick, 2010. – URL: http://www.mathworks.com/help/pdf_doc/nnet/nnet.pdf.
- 4 Нейронные сети. Statistica Neural Networks: Методология и технология современного анализа данных — Москва : Горячая линия — Телеком, 2008. – 392 с.

Симахин Валерий Ананьевич

**Эмулятор нейронной сети для задач обучения
«Neuro-Lab»**

Методические указания
к выполнению лабораторных и практических работ для студентов
направлений подготовки 09.03.04, 09.04.04 «Программная инженерия»

Редактор Г.В. Меньщикова

Подписано в печать 16.10.17	Формат 60x84 1/16	Бумага 65 г/м ²
Печать цифровая	Усл. печ. л. 2, 5	Уч.-изд. л. 2, 5
Заказ №172	Тираж 25	Не для продажи

БИЦ Курганского государственного университета.
640020, г. Курган, ул. Советская, 63/4.
Курганский государственный университет.