

ЛАБОРАТОРНАЯ РАБОТА № 1

«РЕАЛИЗАЦИЯ НЕЙРОННЫХ СЕТЕЙ В ПАКЕТЕ NEURAL NETWORK WIZARD»

Цель: приобретение практических навыков применения нейронных сетей как инструмента искусственного интеллекта с использованием пакета Neural Network Wizard.

Теоретическое обоснование

Neural Network Wizard 1.7 – простой программный эмулятор нейронных сетей (https://freesoft.ru/windows/neural_network_wizard_v17/download). В Neural Network Wizard реализована многослойная нейронная сеть, обучаемая по алгоритму обратного распространения ошибки (*back propagation error*). Программа Neural Network Wizard предназначена для проведения исследований с целью выбора оптимальной конфигурации нейронной сети, позволяющей наилучшим образом решить поставленную задачу. Результатом работы системы является файл, который хранит в себе все параметры полученной нейронной сети. Далее на основе этого файла можно разрабатывать систему, предназначенную для решения конкретных задач.

Программа может применяться для анализа информации, построения модели процессов и прогнозирования. Для работы с системой необходимо проделать следующие операции:

- 1) собрать статистику по процессу;
- 2) обучить нейросеть на приведенных данных;
- 3) проверить полученные результаты.

В результате обучения нейросеть самостоятельно подберет значения коэффициентов и построит такую модель, которая наиболее точно описывает исследуемый процесс.

Программный пакет Neural Network Wizard может быть использован без предварительной инсталляции.

Порядок работы с программой

Прежде чем запускать программу, необходимо в любом текстовом редакторе подготовить текстовый файл с обучающей выборкой. Например, вычислим косинус. Фрагмент такого файла для функции $\text{Res} = \text{Cos}(X)$ приведен ниже.

Cos	Res
0	1
0.1	0.995
0.4	0.921
0.5	0.878
.....	
1.57	0.0

В первой строке файла указываются имена входных/выходных переменных: Cos – имена входных переменных, Res – имя выходной переменной. Далее идут их значения в колонках. Указанная структура значений сохраняется как текстовый файл с расширением **.txt** (например, Cos.txt или Косинус.txt) в директорию, где расположена программа.

Прежде чем запускать программу, необходимо визуально изучить поведение функции и для этого нарисовать, например в Excel, график функции (рис. 1.1). Это важное и обязательное действие, позволяющее понять точность работы нейронной сети.

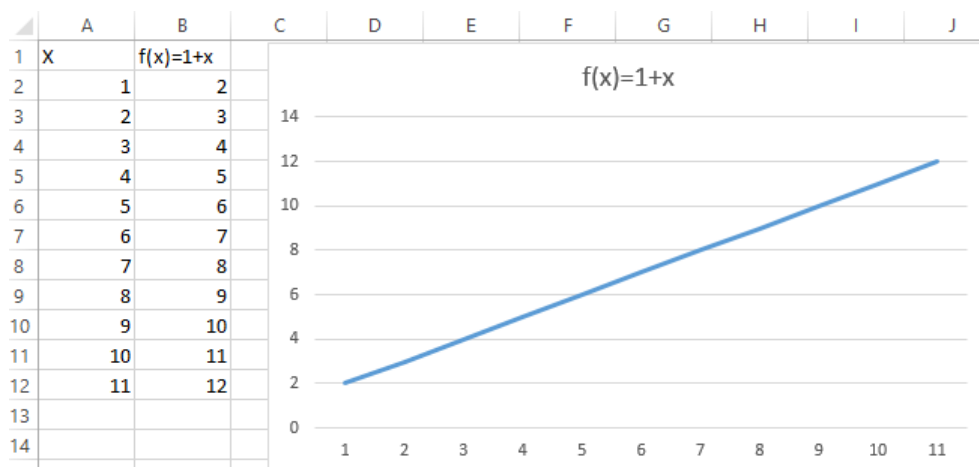


Рис. 1.1 – Графическое представление исследуемой функции

Данные для обучения нейронной сети должны быть предоставлены в текстовом файле с разделителями (Tab или пробел). Количество примеров должно быть достаточно большим. При этом необходимо обеспечить, чтобы выборка была репрезентативной, а данные были непротиворечивы. Вся информация должна быть представлена в числовом виде. Если информация дана в текстовом виде, то необходимо использовать какой-либо метод, переводящий текстовую информацию в числа.

После запуска программы в первом окне задается имя файла с обучающей выборкой (рис. 1.2).

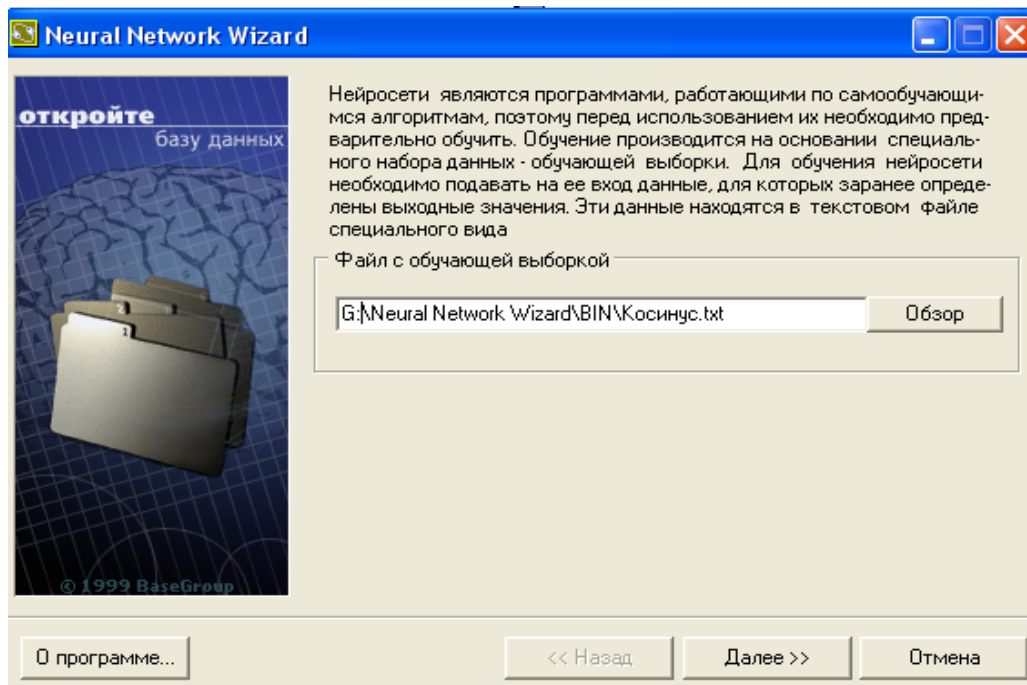


Рис. 1.2 – Выбор файла с обучающей выборкой

Кнопка **Далее** позволяет перейти к следующему окну (рис. 1.3). В новом окне перечислен список доступных полей, взятый из первой строки указанного файла. Для каждого из полей необходимо указать соответствие входной или целевой (выходной) переменной. Если данная переменная *входная*, то в группе **Использовать поле как** выбирается вариант «Входное», если *выходная* – «Целевое». Поля, отмеченные пометкой «Не использовать», в обучении и тестировании нейросети применяться не будут.

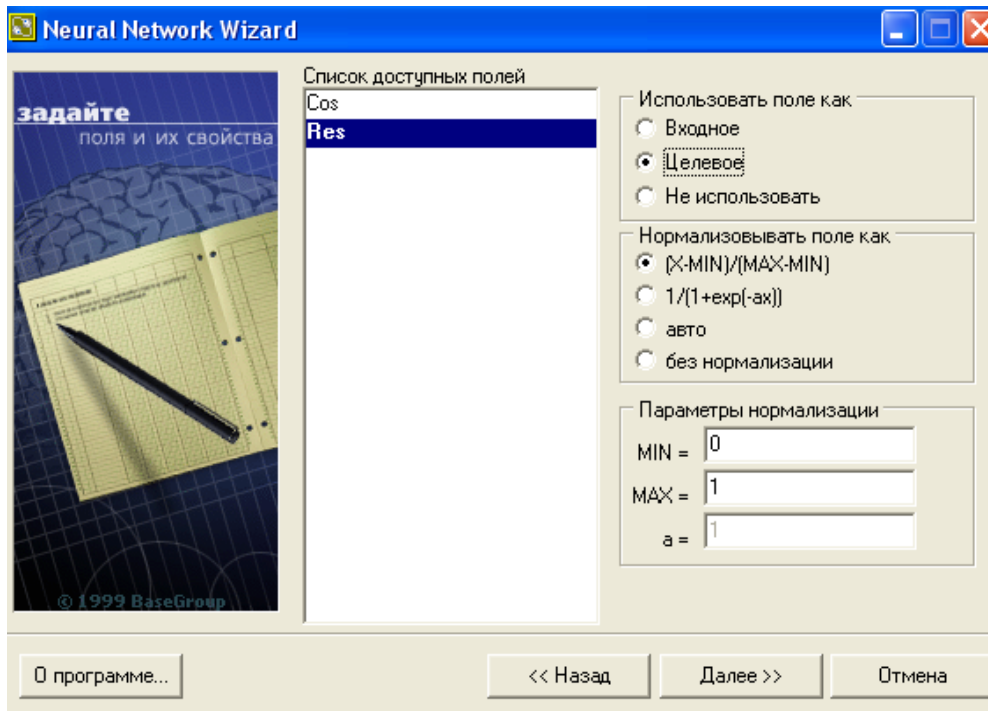


Рис. 1.3 – Определение входных и выходных данных

На вход нейросети должна подаваться информация в нормализованном виде, т. е. это числа в диапазоне от 0 до 1. Метод нормализации можно выбрать на вкладке **Нормализовать поле как:**

- **(X-MIN)/(MAX-MIN)** – линейная нормализация;
- **$1/(1+\exp(ax))$** – экспоненциальная нормализация;
- **Авто** – нормализация, основанная на статистических характеристиках выборки;
- **Без нормализации** – нормализация не производится.

На вкладке **Параметры нормализации** необходимо задать значения, используемые в формулах нормализации.

Кнопка **Далее** переводит к следующему окну (рис. 1.4), где определяются структура и параметры нейросети (в сумме они определяют архитектуру нейронной сети): количество скрытых слоев, количество нейронов в каждом слое, а также вид сигмоидальной функции.

Neural Network Wizard

задайте параметры нейросети

число нейронов во входном слое: 1

число нейронов в выходном слое: 1

число скрытых слоев нейросети: 2

#слоя	число нейронов
1	5
2	7

Параметр сигмоиды: 3

Вид сигмоиды

О программе... << Назад Далее >> Отмена

Рис. 1.4 – Определение параметров структуры нейросети

Нейронная сеть состоит из слоев: входного, выходного и скрытых. Необходимо указать *количество скрытых слоев*. Общего правила определения количества таких слоев нет, обычно задается 1–3 скрытых слоя. Считается, что чем более нелинейная задача, тем больше скрытых слоев должно быть у нейронной сети. В каждом слое должно быть более одного нейрона, их количество зависит от сложности функции и числа входных переменных.

В Neural Network Wizard все элементы предыдущего слоя связаны со всеми элементами последующего. *Количество нейронов* в первом и последнем слое зависит от того, сколько полей в обучающей выборке указаны как входные и выходные. Необходимо задать количество нейронов в каждом скрытом слое. Общих правил определения количества нейронов нет, но необходимо, чтобы число связей между нейронами было значительно меньше количества примеров в обучающей выборке. Иначе нейросеть потеряет способность к обобщению и просто «запомнит» все примеры из обучающей выборки (так называемый эффект переобучения).

Параметр сигмоиды. Сигмоида применяется для обеспечения нелинейного преобразования данных. Чем выше параметр, тем больше переходная функция походит на пороговую (крутой взлет функции активации). Параметр сигмоиды подбирается фактически эмпирически, и от него зависит точность решения задачи.

В следующем окне (рис. 1.5) задаются параметры обучения и критерии остановки обучения, если она требуется.

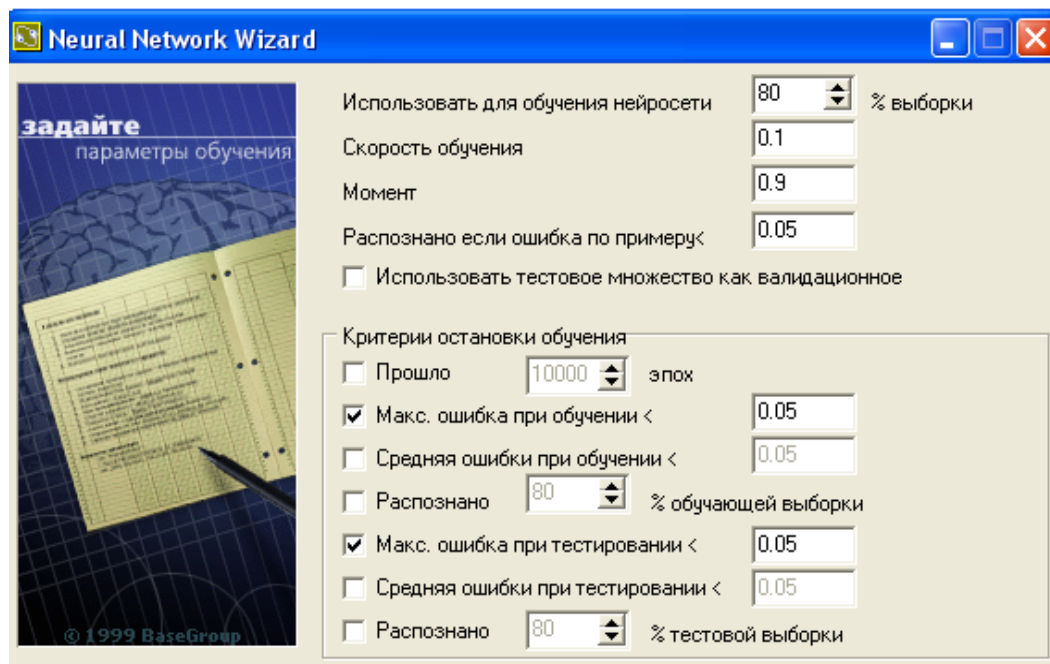


Рис. 1.5 – Параметры обучения

Использовать для обучения сети процент выборки. Все примеры, подаваемые на вход нейросети, делятся на 2 множества – обучающее и тестовое. Заданный процент примеров будет использоваться в обучающей выборке. Записи, используемые для тестирования, выбираются случайно, но пропорции сохраняются.

Скорость обучения. Параметр определяет амплитуду коррекции весов на каждом шаге обучения.

Момент. Параметр определяет степень воздействия i -й коррекции весов на $i+1$ -ю.

Распознано, если ошибка по примеру... Если результат прогнозирования отличается от значения из обучаемого множества на величину, меньшую указанной, то пример считается распознанным.

Использовать тестовое множество как валидационное. При установке этого флага обучение будет прекращено с выдачей сообщения, как только ошибка на тестовом множестве начнет увеличиваться. Это помогает избежать ситуации переобучения нейросети.

Критерии остановки обучения. Необходимо определить момент, когда обучение будет закончено.

Кнопка **Далее** показывает окно с конфигурацией нейросистемы и заданные параметры (рис. 1.6).

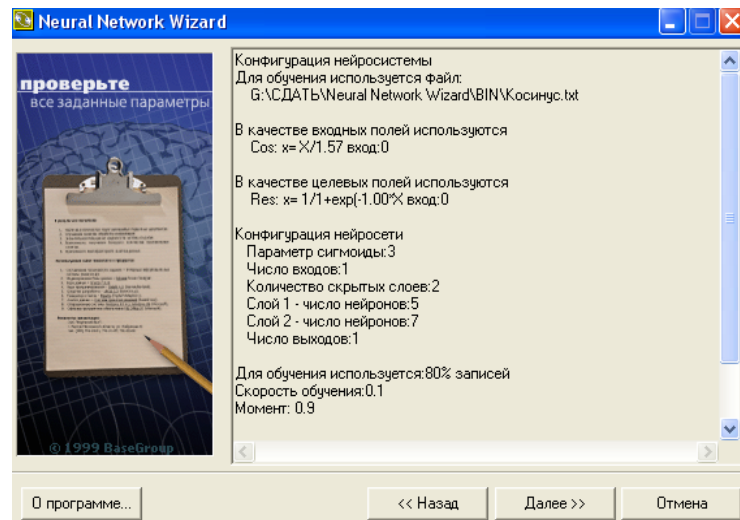


Рис. 1.6 – Заданные параметры нейросети

В следующем окне визуализирован непосредственно сам процесс обучения (рис. 1.7). Чтобы запустить обучение, необходимо нажать кнопку **Пуск обучения**. На верхней диаграмме показано распределение ошибки обучения: по горизонтали – значение ошибки (чем правее столбец, тем больше ошибка), по вертикали – количество примеров из выборки с данной ошибкой (чем выше столбец, тем больше примеров с указанной ошибкой). Зеленые столбцы – ошибка на рабочей обучающей выборке, красные – на тестовой. В процессе обучения столбцы должны стремиться в левую часть диаграммы.

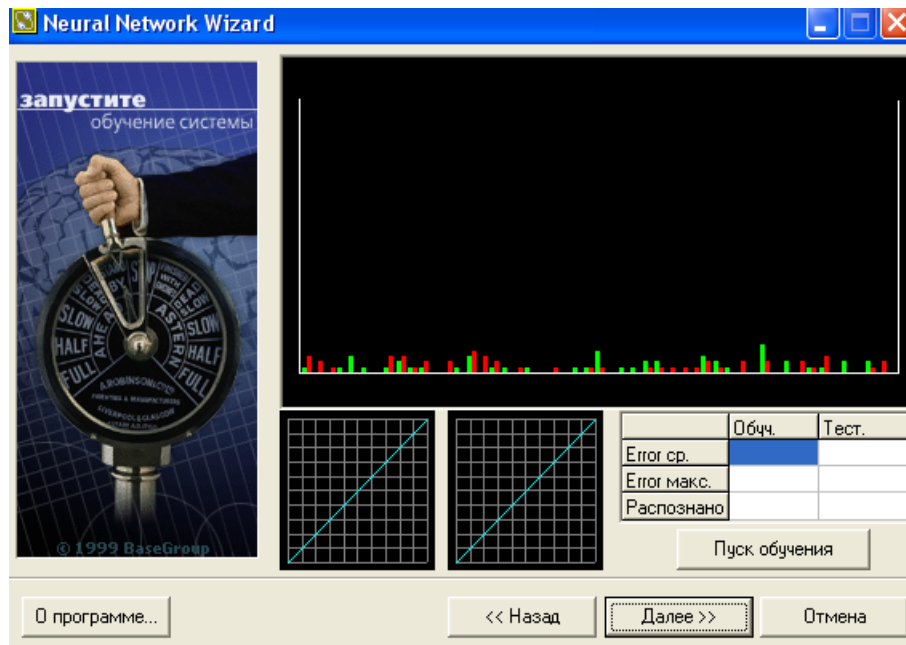


Рис. 1.7 – Обучение системы

Ниже диаграммы отображается распределение примеров на рабочей и тестовой выборках. На этих графиках можно отслеживать, насколько результаты, предсказанные нейронной сетью, совпадают со значениями в обучающей (слева) и тестовой (справа) выборке. Каждый пример обозначен на графике точкой. Если точка попадает на выделенную линию (диагональ), то нейросеть оценила результат с достаточно высокой точностью. Если точка находится выше диагонали, значит, нейросеть недооценила значения функции, ниже – переоценила. Необходимо добиваться того, чтобы точки располагались как можно ближе к диагонали.

Остановка обучения происходит либо по ранее указанному критерию, либо с помощью той же кнопки **Пуск обучения**.

В следующем окне (рис. 1.8) представляется возможность оценить точность работы обученной нейросети в эксплуатационном режиме. Для этого в левом поле указывается произвольное значение входного сигнала. После нажатия на кнопку **Расчет** в правом поле появляется рассчитанное сетью значение.

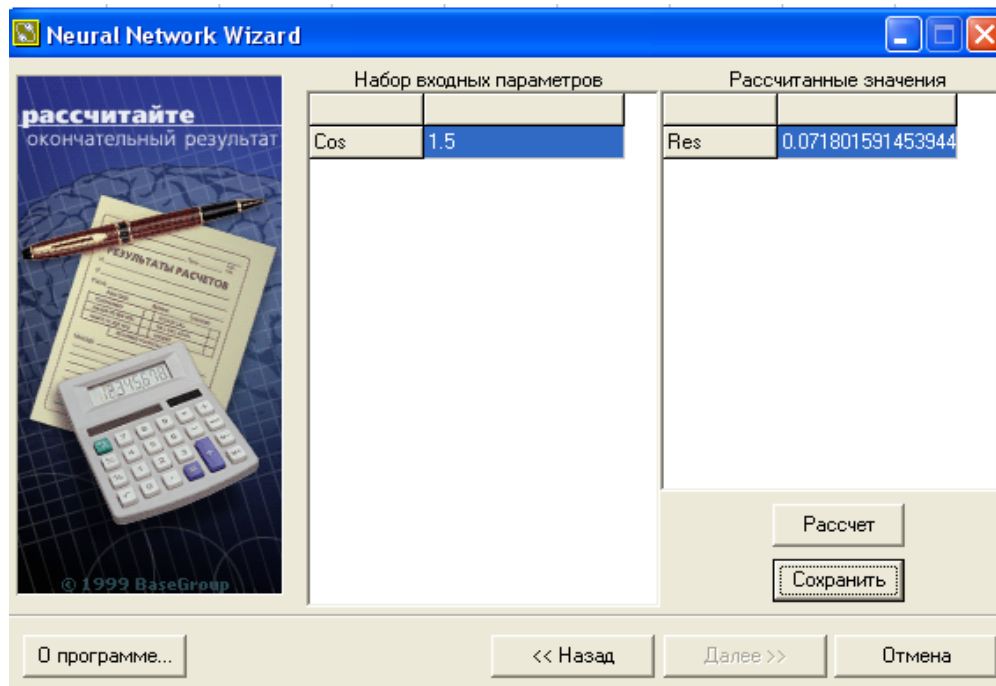


Рис. 1.8 – Работа с обученной нейронной сетью

Кнопка **Сохранить** позволяет записать параметры обученной сети в виде файла (по умолчанию расширение файла .NNW).

Порядок выполнения лабораторной работы

Шаг 1

В вариантах 1–25 необходимо с помощью нейронной сети провести аппроксимацию функции вида $Y = f(X)$ на шкале X , где X – входная переменная, а Y – выходная переменная. Диапазоны изменения входных переменных определить самостоятельно, у периодических функций они должны быть не менее периода. Значения диапазона можно оценить после расчета значений функции Y (пример порядка расчета представлен в разделе «Расчет функций в Excel»).

Основная задача аппроксимации – построение приближенной (аппроксимирующей) функции, в целом наиболее близко проходящей около данных точек исследуемой функции.

Краткое описание задачи

Пусть аналитическое выражение для функции $f(X)$ неизвестно, а известны только ее значения в некоторых точках отрезка $X \in [0; n]$, т. е. функция $f(X)$ задана табличными значениями (табл. 1.1). Эти значения представляют собой либо результаты расчетов, либо экспериментальные данные.

Таблица 1.1 – Значения выбранной функции для выполнения задачи.

Номер значения	1	2	3	4	5	...
X	0.1	0.2	0.3	0.4	0.5	...
$f(X)$	1.1	2.5	3.2	4.1	6.2	...

Значения аргумента $X(i)$ называются узлами. В общем случае узлы не являются равноотстоящими. Требуется найти приближенные значения функции $f(X)$ в любой произвольной точке отрезка $X \in [0; n]$. Приближение функции $f(X)$, заданной таблично, другой непрерывной функцией $f(X)$ называется аппроксимацией (от лат. *approximare* – приближаться). В общем смысле термином «аппроксимация» называют замену одних математических объектов другими, в том или ином смысле близкими к исходным. Чем проще аппроксимирующая функция,

тем меньше времени требуется для решения задачи аппроксимации. Эта характеристика особенно важна при большом количестве узлов. Чем больше количество узлов, тем меньше погрешность. Для каждой конкретной аппроксимируемой функции нужно стремиться выбрать такой способ аппроксимации, который обеспечивает минимальную погрешность при минимальном количестве узлов.

Шаг 2

Подготовить две обучающие выборки средствами приложения Microsoft Excel и оформить их в виде файла *.txt. *Первая* выборка рассчитывается для функции, соответствующей номеру вашего варианта (меняется от 1 до 25 и соответствует $Y = f(X)$), а *вторая* выборка связана с добавлением случайных чисел ($\text{rnd}(1)$) к функции ($Y = f(X) + \text{rnd}(1)$). Чтобы создать набор случайных чисел в Excel, воспользуйтесь функцией =СЛЧИС(). Функция СЛЧИС() возвращает равномерно распределенное случайное вещественное число, большее или равное 0, но меньшее 1. При каждом пересчете листа возвращается новое случайное вещественное число. Можно изменять величину случайного числа в зависимости от диапазона значений входной функции. Например, если $f(X) \in [0;100]$, тогда значения случайного датчика становятся СЛЧИС()*100 или СЛУЧМЕЖДУ(1;100). Сохраните файлы в папке с файлами программы, выбрав расширение .txt (для примера в директории, где расположена программа, приведены расчеты SUM, MULT, COS, SQRT, которые можно использовать для знакомства с программой).

В отчете эти данные нужно указать в виде таблицы. Также необходимо создать визуализацию точных и зашумленных (со случайными числами) значений функции.

Шаг 3

Провести обучение нескольких нейронных сетей с помощью программного нейроимитатора *Neural Network Wizard* с количеством нейронов в скрытом слое, равным 1, 3, 5, 7, и функцией активации «сигмоидная». Рассмотреть разные значения параметра сигмоиды (см. рис. 1.4) и наблюдать за точностью получаемых значений. Обучение каждой сети вести, например, в 100 шагов. Обучение

вести по алгоритму обратного распространения. Выше указано, что нужно сделать две выборки.

Сначала нужно использовать файл с первой выборкой (функция без датчика случайных чисел) для обучения нейронной сети и получения результатов, потом – со второй (функция с датчиком случайных чисел), где проделать те же манипуляции и получить новый результат, который сравнить с первым.

В отчете должны быть приведены результаты работы программы в виде скриншотов и описания полученных результатов.

Шаг 4

Проверить качество каждой обученной сети, для чего рассчитать несколько значений выходной переменной по заданной функции и по нейронной сети, затем на графике показать близость каждой обученной модели к исходной (рис. 1.7). Этот результат необходимо отразить в отчете с помощью таблицы значений, указанных вами и предсказанных нейронной сетью. Часть указанных вами входных значений должна совпадать с теми, которые были задействованы при обучении. Результат сравнения нужно проанализировать, показав отличие (в процентах) или совпадение.

Требования к структуре и содержанию отчета

Структура отчета (в основной части) должна соответствовать четырем шагам. На каждом этапе необходимо расписать, что и как сделано, подкрепив текстовое описание скриншотами и таблицами значений.

Структура отчета по лабораторной работе:

1. Титульный лист.
2. Содержание отчета по лабораторной работе.
3. Цель лабораторной работы: изучение работы программы нейронной сети в области аппроксимации заданной функции $Y(X)$ на множестве значений X . Студентом формируется два типа выборок значений: 1) исходная функций $Y(X)$ и 2) функция $Y(X)$ с добавлением случайных чисел. Для обоих вариантов решается

идентичная задача обучения нейронной сети, проверки качества обучения и выполняется сравнение результатов.

4. Описание решения индивидуального задания в соответствии с каждым из четырех шагов.

5. Анализ полученных решений с комментариями (текстовыми и графическими).

Отчет должен представлять собой документ в формате Word. Вместе с файлом Word прикладываются файлы данных, которые были рассчитаны студентом и задействованы в выполнении лабораторной работы.

Расчет функций в Excel

Расчет функции $Y(X) = \sin(X)$ в Excel.

Шаг 1. Открыть программу Excel (рис. 1.9).

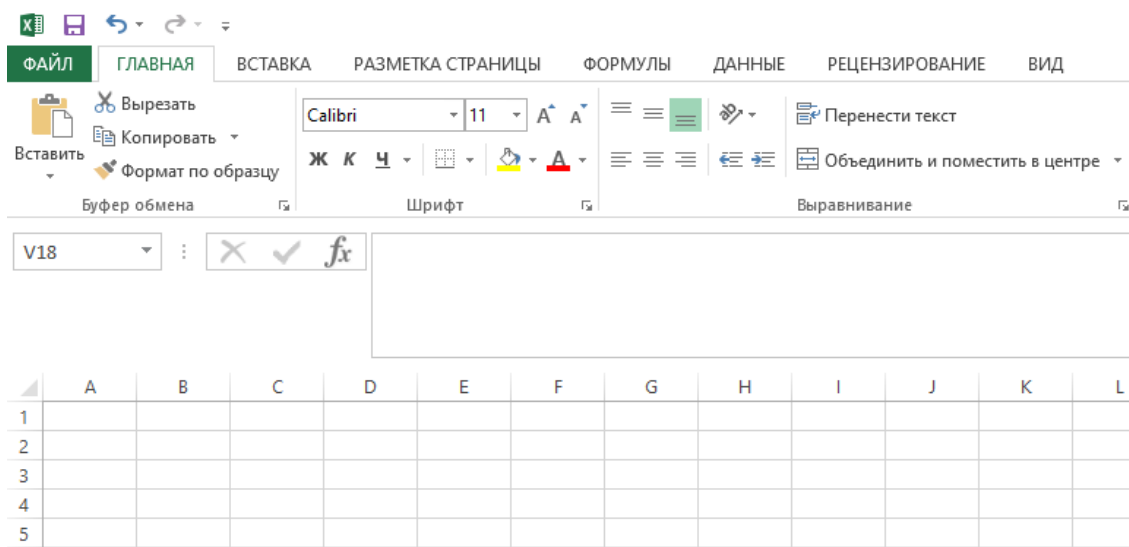


Рис. 1.9 – Шаг 1

Шаг 2. Задать наименование ячеек, как это сделано на рисунке 1.10.

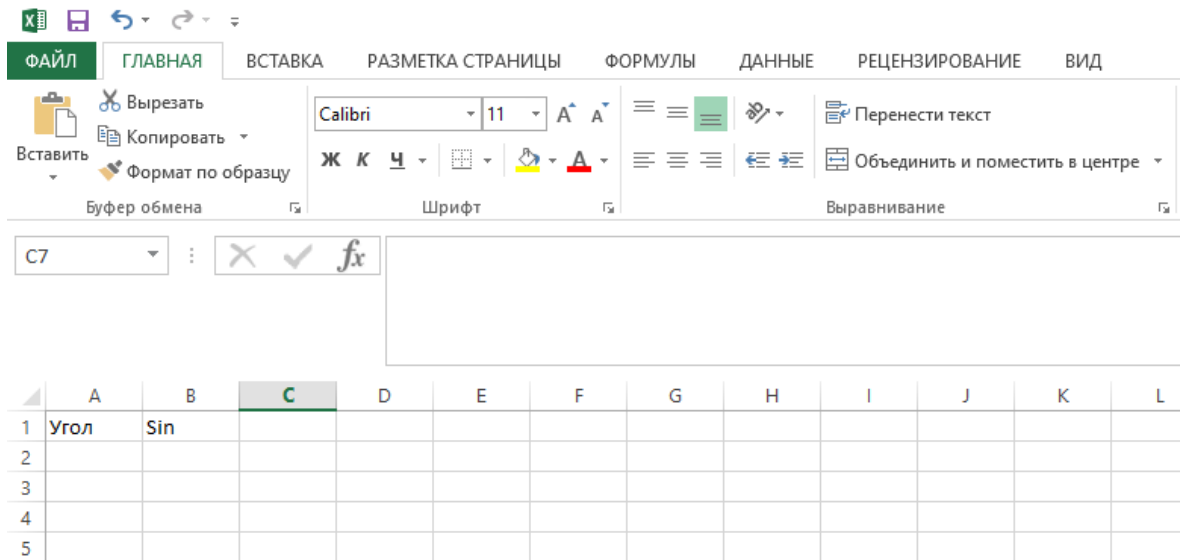


Рис. 1.10 – Шаг 2

Шаг 3. Указать значения угла. Для этого наберите значения в ячейках A2 и A3, как показано на рисунке 1.11.

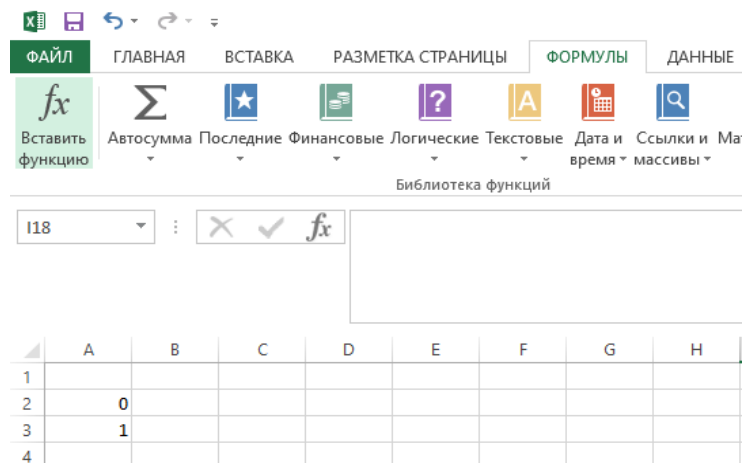


Рис. 1.11 – Шаг 3

Выделите ячейки (рис. 1.12).

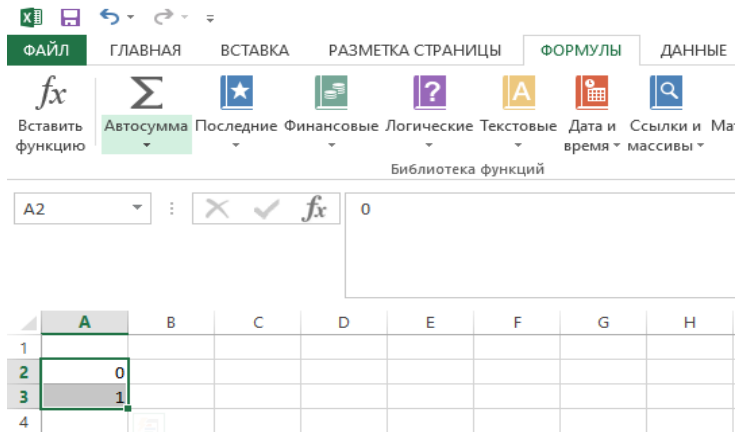


Рис. 1.12 – Шаг 3.1

Потяните за угол на несколько ячеек вниз, как показано на рисунке 1.13.

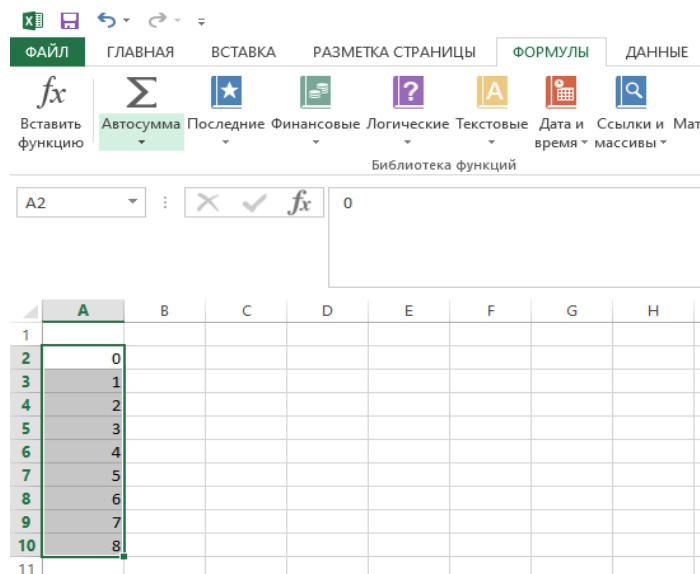


Рис. 1.13 – Шаг 3.2

Таблица чисел – значений углов будет заполняться автоматически.

Шаг 4. Выберите ячейку B2. Нажмите кнопку «Вставка функции» (слева вверху). Выберите в списке функцию SIN и получите результат, как на рисунке 1.14.

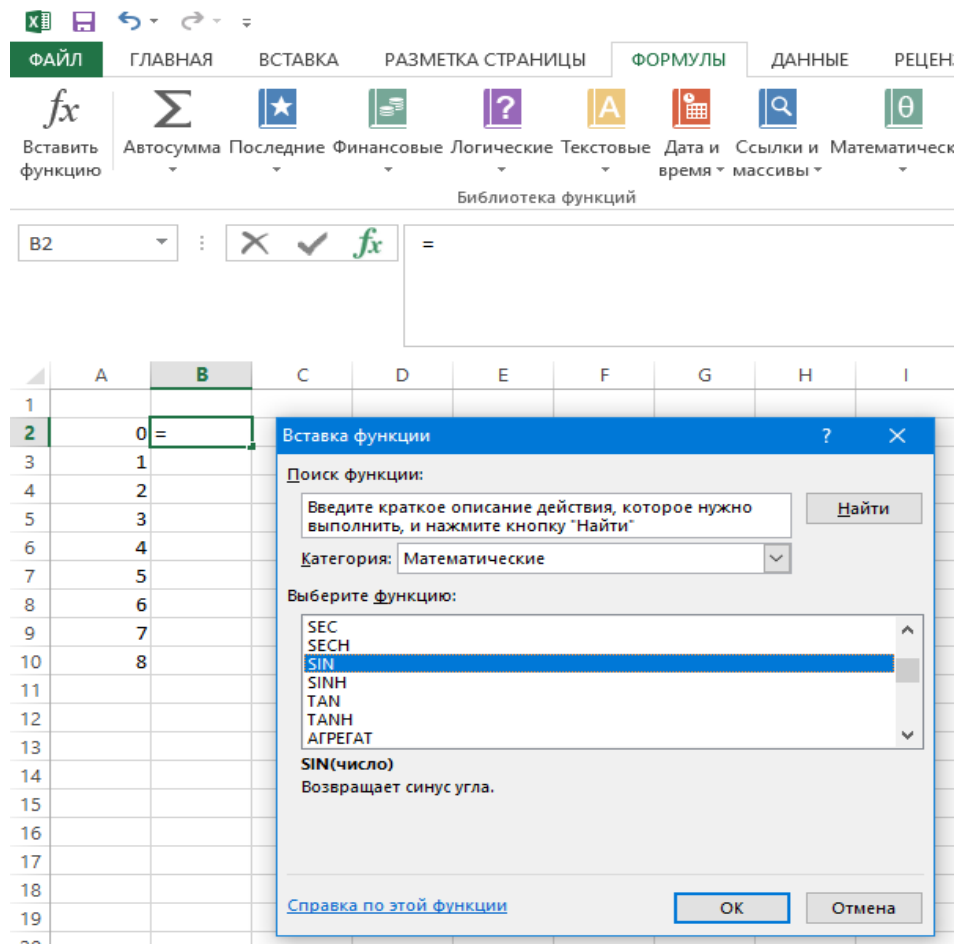


Рис. 1.14 – Шаг 4

Шаг 5. Нажимая «ОК», получим окно ввода аргументов функции (рис. 1.15).

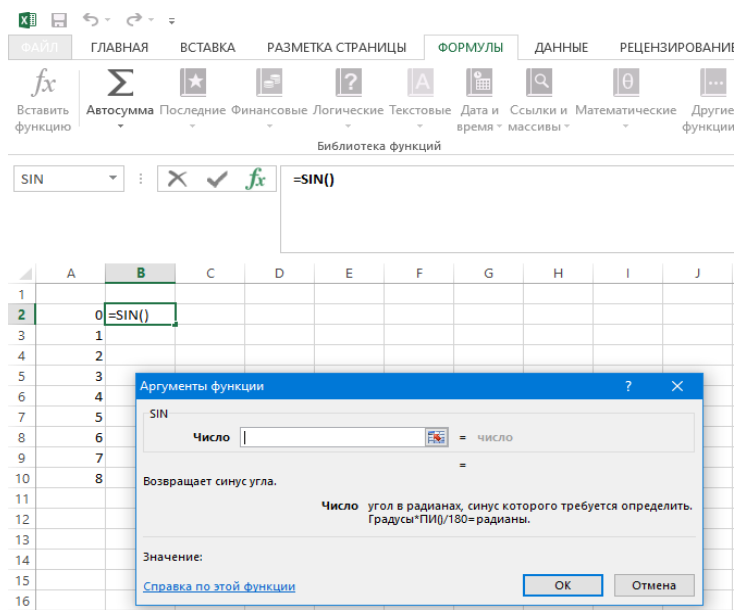


Рис. 1.15 – Шаг 5

Шаг 6. В поле под именем «Число» ввести команду.

В поле «Число» вводится аргумент функции – в рассматриваемом примере это $A2 * \text{ПИ}() / 180$. С помощью клавиатуры в это поле введите только $\text{ПИ}()$, а ссылку на ячейку A2 в формулу добавьте, щелкнув по ячейке A2 на рабочем листе. После нажатия кнопки «ОК» в ячейку B2 будет введена формула: $=\text{SIN}(\text{ПИ}() * A2 / 180)$. Далее потяните за нижний угол окна вниз, покрывая все значения аргумента, и значения $\text{SIN}(X)$ будут рассчитаны (рис. 1.16).

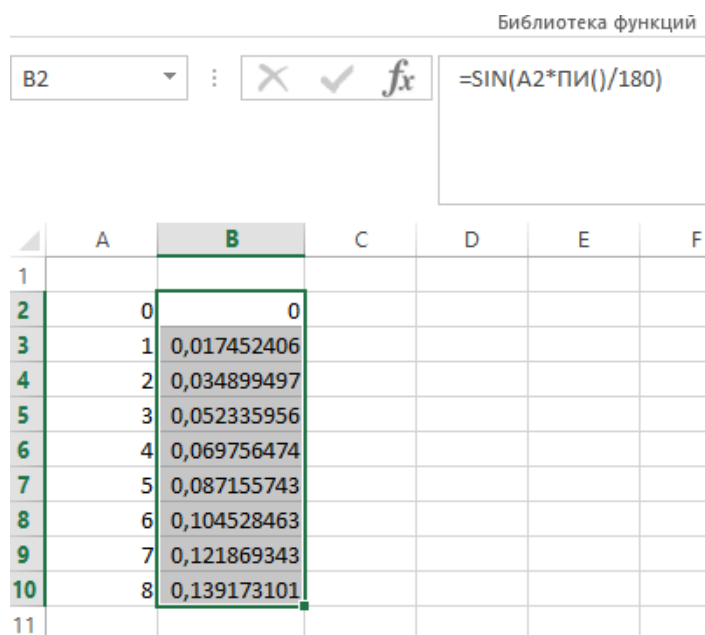


Рис. 1.16 – Шаг 6

Предлагаемый алгоритм является основой расчета всех функций задания.

Варианты заданий на лабораторную работу

Номер варианта	Задание
1	$\sin(X) * \sin(X) * \cos(X)$
2	$\cos(X) * \sin(X)$
3	$2 * \sin(X * X) * \cos(X/2)$
4	$\sin(X/2) * \sin(X) / \cos(X)$
5	$1 + 2 * X + \cos(X) * \sin(X * X)$
6	$\cos(X) + X$
7	$\sin(X) + X$
8	$\sin(X + 2)$
9	$\sin(X/2)$
10	$\sin(2 * X)$
11	$\sin(X/4) * (\cos(X/2) + 2 * X)$
12	$1 + X * X + \cos(X)$
13	$10 - X + \cos(X)$
14	$\text{Tg}(X)$
15	$\cos(X) + \sin(X)$
16	$\sin(X) - \cos(X)$
17	$\cos(X) / \sin(X)$
18	$1 + X$
19	$1 + X + X * X$
20	$(1 + X) / (1 + X)$
21	$(1 + \text{Tg}(X)) / (1 + X)$
22	$\sin(X) / (1 + X * X)$
23	$2 * \sin(X * X)$
24	$1 + X - X * X$
25	$\cos(X * X) \sin(X * X)$

Значения аргумента X должны находиться в интервале $[0;10]$ с шагом не менее 0.1. Студенты могут проводить собственные дополнительные эксперименты, уменьшая или увеличивая интервал значений и шаг (это приветствуется).

Таким образом, в ходе выполнения лабораторной задания необходимо рассмотреть точность аппроксимации функции с помощью нейронной сети. Для этого:

- 1) рассчитайте значения функции;
- 2) зашумите значения функции случайными числами;
- 3) нарисуйте графики функций из п. 1 и 2;

4) постройте архитектуру нейронной сети с разным числом скрытых слоев (1, 2 и 3) и разным количеством нейронов в них (2, 3, 4), записывая получаемую при этом точность обучения нейронной сети;

5) изучите влияние параметров нейронной сети (нормализация (два варианта), скорость обучения, процент значений функции для обучения, число итераций (эпох) обучения (10 или 1000) или максимальная точность (0.01 и 0.05)) на точность.

В ходе изменения параметров нейронной сети точность аппроксимации заданных для изучения функций должна быть высокой, при этом она не может быть более 1%.

Не используйте для сдачи лабораторных работ шаблоны, распространяемые в Сети. Работы должны быть выполнены самостоятельно. При проверке легко обнаружить несамостоятельно выполненные работы, т. к. они содержат одинаковый текст, одни и те же таблицы, идентичные число и структуру файлов и т. д. Кроме того, скачанные готовые работы часто содержат неточные решения.