

Лабораторная работа №2

«Исследование многослойного персептрона с обучением по методу с обратным распространением ошибки»

Целью выполнения лабораторной работы является изучение алгоритма обратного распространения ошибки в процессе обучения нейронной сети. Работа выполняется с помощью программы br.exe.

1.1 Порядок выполнения лабораторной работы

1.1 Вариант задания выдается преподавателем и определяется файлом обучающей выборки и ознакомиться с теоретической частью лабораторной работы, приведенной в подразделе 4.2 и 4.3.

Запустить программу br.exe и установить параметры нейронной сети и режимов работы:

1.2 В главном меню (рис.1) выбрать в меню НЕЙРОСЕТЬ/СОЗДАТЬ НЕЙРОСЕТЬ и щелкнуть по кнопке ЗАГРУЗИТЬ ТАБЛИЦУ (файл в формате *.dbf). Установить флаг СОХРАНЯТЬ ПРОТОКОЛ.

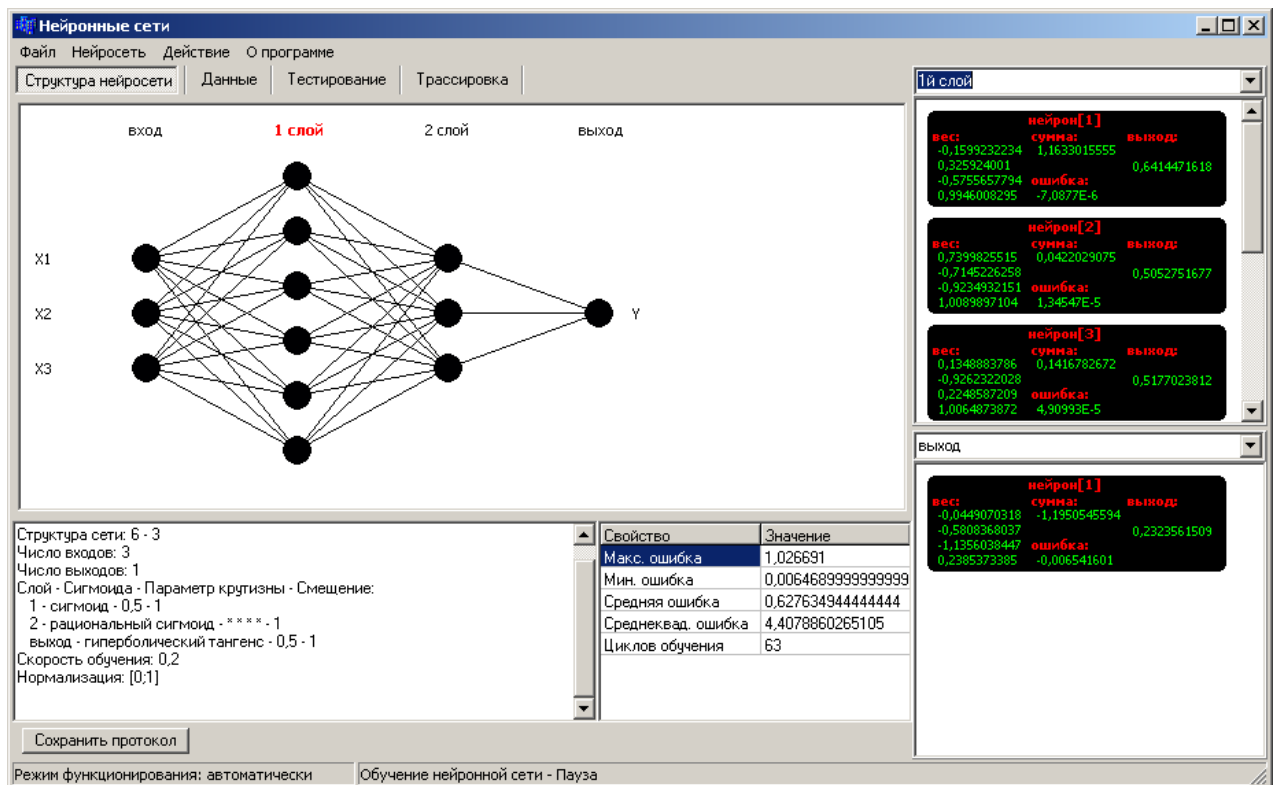
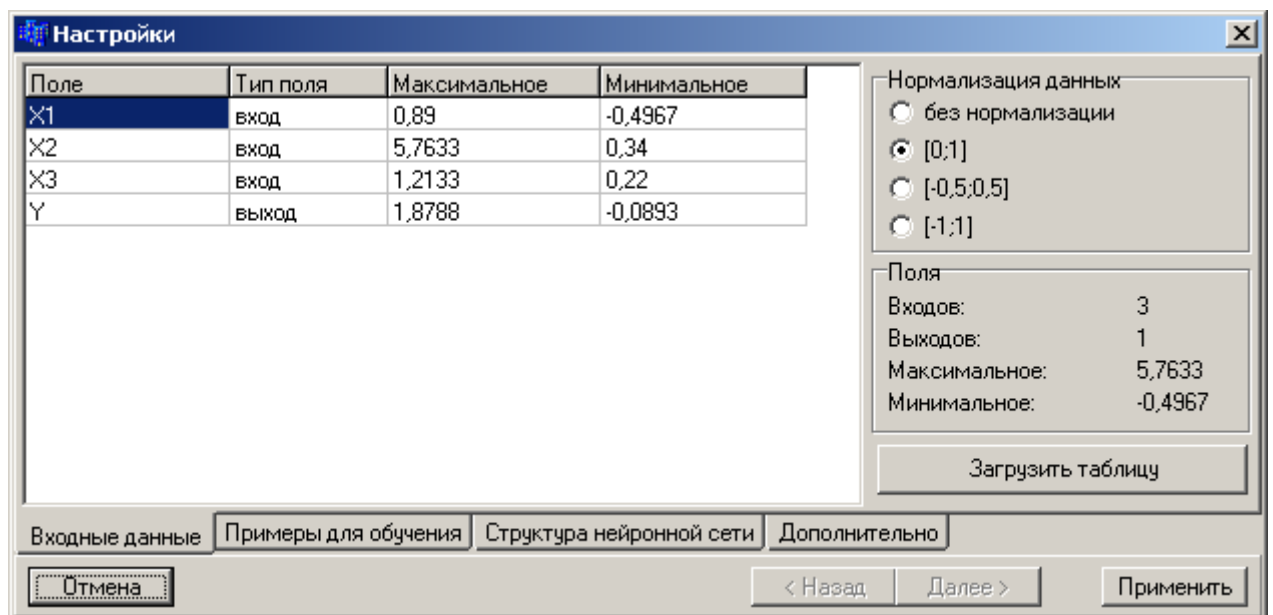


Рис.1 Главное меню программы

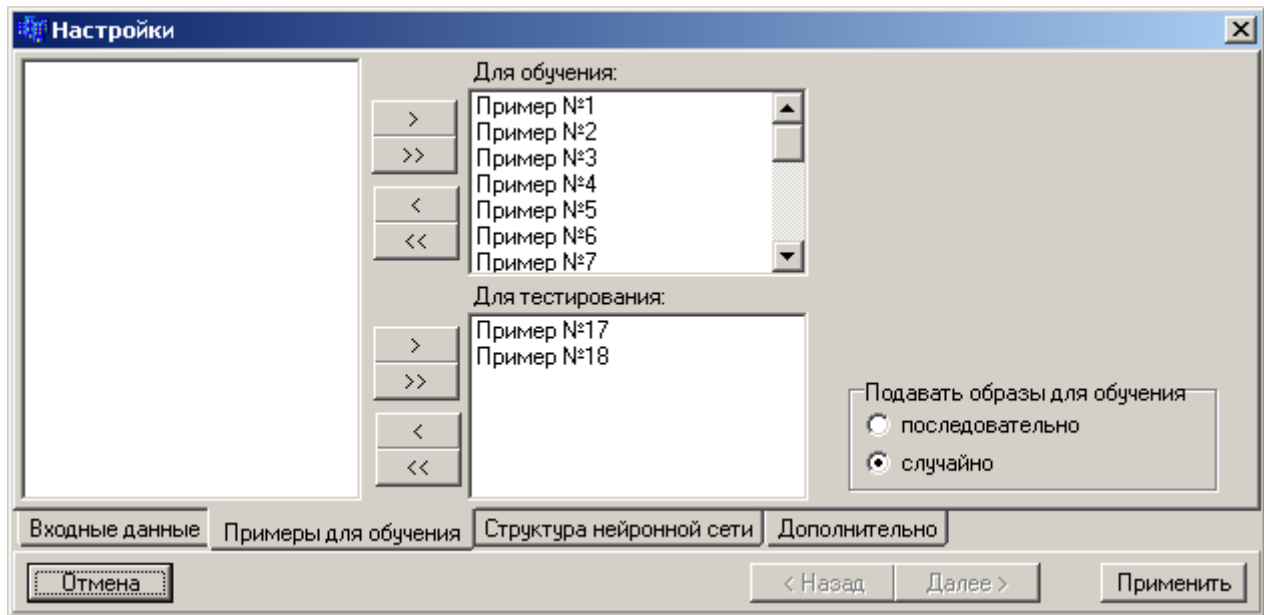
1.3 Выбрать вкладку **ВХОДНЫЕ ДАННЫЕ** и установить режим нормализации (рис.2).

Рис.2 Входные данные



1.4 Выбрать вкладку ПРИМЕРЫ ДЛЯ ОБУЧЕНИЯ и перенести в «поле обучения» все примеры кроме двух-трёх, которые перенести в «поле тестирования». Установить параметр обучения: «Подавать образы для обучения последовательно», или «Подавать образы для обучения случайно» (рис. 3).

Рис.3 Примеры для обучения



1.5 Выбрать вкладку СТРУКТУРА НЕЙРОСЕТИ и установить параметры: количество слоёв и нейронов в слое, параметры крутизны функции, смещения и его веса (рис. 4).

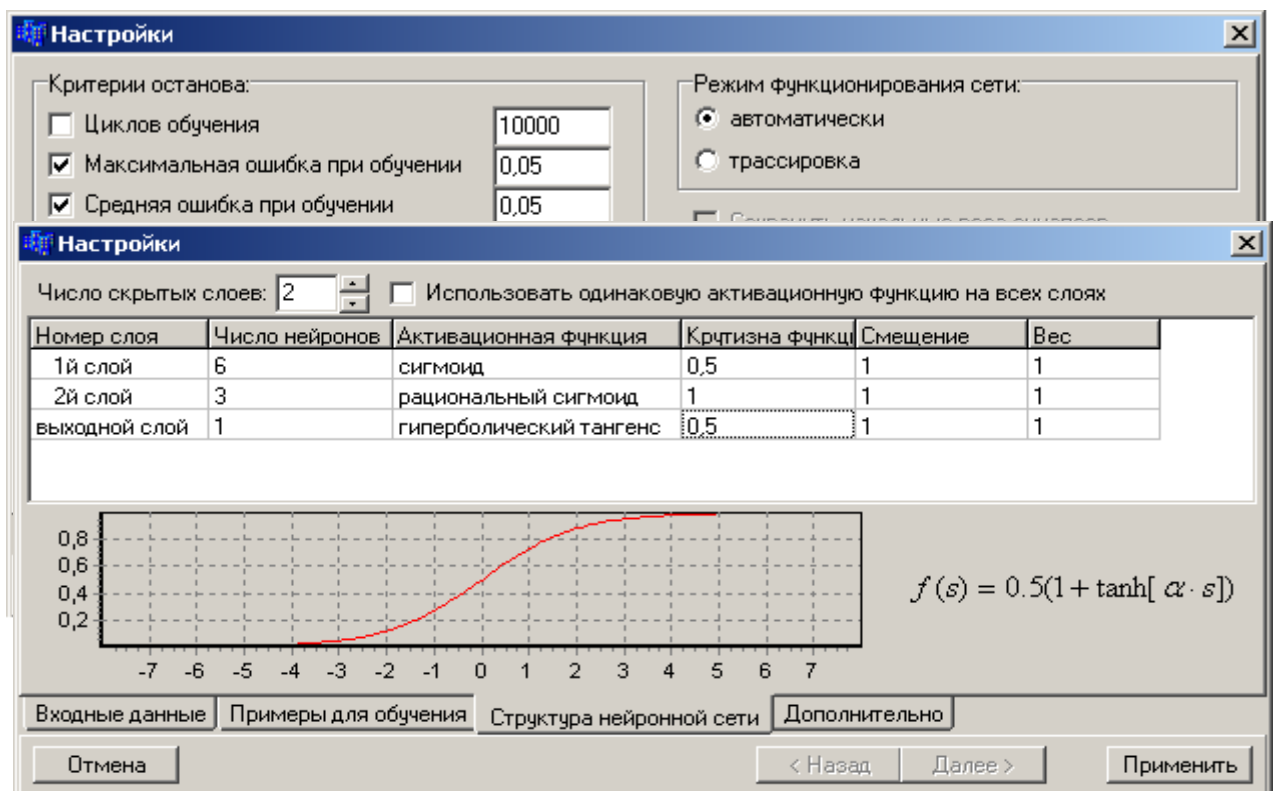


Рис.4 Настройки

- 1.5.1 Выбрать вкладку ДОПОЛНИТЕЛЬНО и установить критерии останова, режим функционирования (автоматически или трассировка) с сохранением начальных весов синапсов, параметр скорости обучения (рис.5).

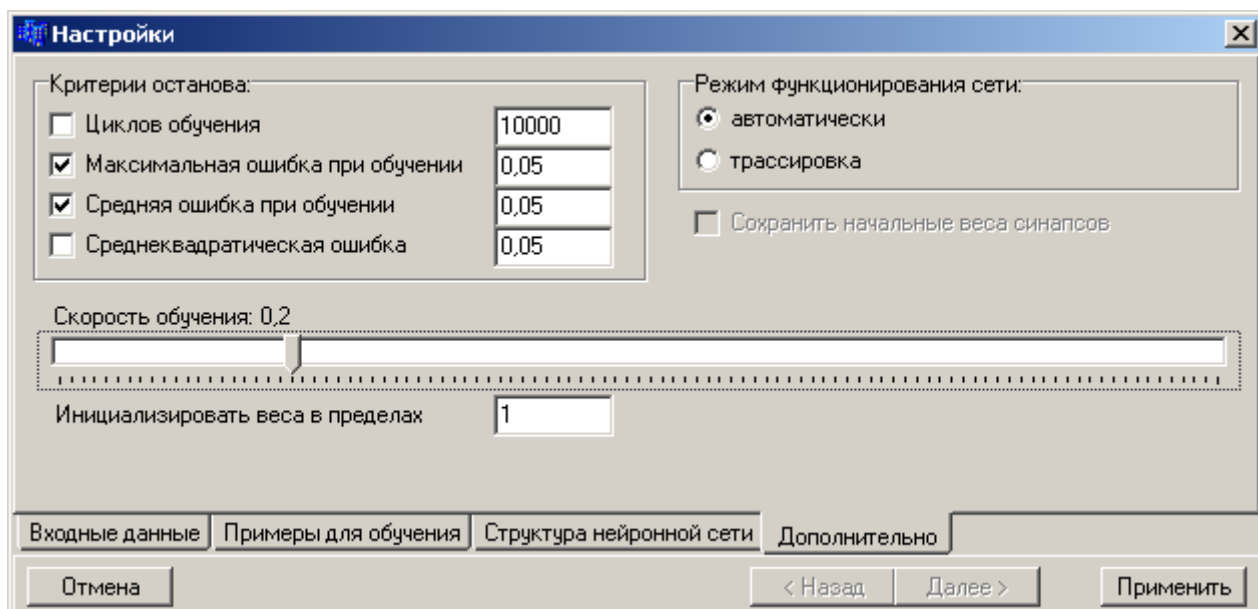


Рис.5 Настройки –вкладка ДОПОЛНИТЕЛЬНО

1.5.2 Подготовить отчет по результатам выполнения лабораторной работы, отразив в нем следующее:

- Задание с обучающей выборкой;
- Протокол выполнения лабораторной работы;
- Расчёт весов синапсов при выполнении одного прохода вперёд и назад; сравнение полученных расчётных данных весов с результатами работы сети в пошаговом режиме;
- Результаты вычисления значений нейронов и синапсов сети по шагам алгоритма;
- Влияние следующих параметров на эффективность алгоритма обратного распространения ошибки: без нормализации; с нормализацией $[0,1]$, $[-0.5; +0.5]$, $[-1;+1]$;
- Выбор примеров из выборки случайный или последовательный;
- Влияние количества слоев и нейронов в слое, параметра крутизны, коэффициента скорости обучения, смещения и его веса на качество и время обучения;

2 Общие сведения по методам обучения

2.1 Описание алгоритма обратного распространения ошибки

Алгоритм обратного распространения ошибки (ОРО) — итеративный градиентный алгоритм обучения, который используется с целью минимизации среднеквадратичного отклонения текущего выхода и желаемого выхода многослойных НС. На каждый нейрон первого слоя подаются все элементы внешнего входного сигнала. Нейроны выполняют взвешенное суммирование элементов входных сигналов, прибавляя к сумме смещение нейрона. Над полученной суммой выполняется нелинейное преобразование активационной функцией. Значение функции активации и есть выход нейрона.

В многослойных НС оптимальные выходные значения нейронов скрытых слоев неизвестны. В таком случае, наиболее приемлемым оказывается использование градиентного метода поиска минимума функции ошибки от выходов сети к ее входам.

Согласно методу наименьших квадратов, минимизирующей целевую функцию ошибки НС является величина:

$E = \frac{1}{2} \sum_j (y_j - t_j)^2,$	(1)
---	-----

где y_j — реальное выходное состояние нейрона j выходного слоя;

d_j — идеальное (желаемое) выходное состояние этого нейрона.

Выходное значение j -го нейрона y_j определяется по формуле

$y_j = F(S_j),$	(2)
-----------------	-----

где F — активационная функция

S_j — взвешенная сумма входов j -го нейрона

$S_j = \sum_i w_{ij} y_i - T_j,$	(3)
----------------------------------	-----

где y_i — выходное значение i -го нейрона предыдущего слоя

w_{ij}, T_j — соответственно весовой коэффициент и порог j -го нейрона.

Согласно методу градиентного спуска изменение весовых коэффициентов и порогов НС происходит по следующему правилу:

$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{\partial E}{\partial w_{ij}(t)},$	(4)
$T_j(t+1) = T_j - \alpha \frac{\partial E}{\partial T_j(t)},$	(5)

Ошибка j -го нейрона γ_j выходного слоя

$\gamma_j = y_j - t_j,$	(6)
-------------------------	-----

Для нейронов же скрытых слоев формула, как показано в /1/, имеет вид

$\gamma_j = \sum_{i=1}^m \gamma_i F'(S_i) w_{ji},$	(7)
--	-----

где m — число нейронов следующего слоя по отношению к скрытому слою.

Также в /1/ показано, что

$\frac{\partial E}{\partial w_{ij}} = -\gamma_j F'(S_j) y_i,$	(8)
$\frac{\partial E}{\partial T_j} = \gamma_j F'(S_j)$	(9)

С учетом полученных формул, весовые коэффициенты и пороги нейронных элементов должны изменяться с течением времени следующим образом:

$w_{ij}(t+1) = w_{ij}(t) - \alpha \gamma_j F'(S_j) y_j,$	(10)
$T_j(t+1) = T_j(t) + \alpha \gamma_j F'(S_j).$	(11)

В качестве активационной функции F может выступать сигмоид, гиперболический тангенс или иная функция.

Полностью алгоритм обучения методом ОРО выглядит следующим образом:

На первом шаге на все входы сети подается один из возможных образов из обучающей выборки. В режиме нормального функционирования сети по формулам (2) и (3) вычисляются выходные значения.

На втором шаге рассчитывается по формуле (6) ошибка нейронов и по формуле (10) ошибка синаптических весов выходного слоя.

На третьем шаге вычисляются ошибки нейронов и синаптических весов для скрытых слоев в порядке, обратном нормальному выполнению суммирования.

На четвертом этапе происходит корректировка по формуле (10) синаптических весов всех слоев.

На пятом шаге полученная ошибка обучения сравнивается с максимально допустимой и принимается решение об окончании или продолжении процесса обучения.

К описанному методу есть ряд дополнений.

Во-первых, представляться на вход сети элементы обучающей выборки должны в случайном порядке, чтобы не вызвать привыкания НС.

Во-вторых, предпочтительно нормализовать входные сигналы, используя одну из формул нормализации. Наиболее часто используется такая формула нормализации:

$x'_i = \frac{x_i - \min}{\max - \min},$	(12)
--	------

где \min , \max — соответственно минимальное и максимальное значение по выборке. Разработанная программа производит нормализацию данных автоматически, на различных интервалах оптимизации.

Вопросы для проверки

1. Каким образом влияют начальные веса синапсов на качество обучения?
2. Почему неприменим к многослойным НС простой алгоритм обучения перцептронов?
3. Как влияет коэффициент скорости обучения на её качество?
4. Как влияет нормализация обучающей выборки на результаты обучения?
5. Какие методы используются для выхода из локального экстремума при обучении НС?
6. Каким образом эффективнее выбирать примеры из обучающей выборки?

Библиографический список

1. Головкин В. А. Нейронные сети: обучение, организация и применение. Кн. 4: Учеб. Пособие для вузов М.: ИПРЖР, 2001.-256с.

2. Круглов В. В. Нечеткая логика и искусственные нейронные сети / В. В. Круглов, М. И. Дли, Р. Ю. Голунов, Физматлит 2001 г. — 224 с.
3. Короткий С. Нейронные сети: алгоритм обратного распространения 1996 г. — 15 с.
4. Almeida L. B. 1987. Neural computaters. Proceedings of NATO ARW on Neural Computers, Dusseldorf. Heidelberg: Springer-Verlag.
5. Parker D. B. 1987. Second order back propagation: Implementing an optimal $O(n)$ approximation to Newton's method as an artificial newral network. Manuscript submitted for publication.
6. Pineda F. J. 1988. Generalization of backpropagation to recurrent and higher order networks. In Newral information processing systems, ed. Dana Z. Anderson, pp. 602-11. New York: American Institute of Phisycs.
7. Stornetta W. S., Huberman B. A. 1987. An improwed three-layer, backpropagation algorithm. In Proceedings of the IEEE First International Conference on Newral Networks, eds. M. Caudill and C. Butler. San Diego, CA: SOS Printing.
8. Sejnowski T. J., Rosenberg C. R. 1987. Parallel networks that learn to pronounce English text. Complex Systems 1:145-68.