Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский университет

“Высшая школа экономики”»

Жалкова Наталия Евгеньевна

**Нейросетевое сложение**

Отчет студента 2 курса бакалавриата группы №НТ-6

Москва 2017

Работа Жалковой Наталии, группа НТ-6 по теме “Нейросетевое сложение”.

**Задание.**

1. Составьте программу нейросети и обучите её выполнению сложения двух одноразрядных десятичных чисел, т.е. работе по формуле С=А+В, если А и В – десятичные цифры от 0 до 9; а С изменяется от 0 до 18. Для обучения используйте следующую таблицу:

А В С

|  |  |  |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 0 | 2 | 2 |
| 0 | 3 | 3 |
| 0 | 4 | 4 |
| 0 | 5 | 5 |
| 0 | 6 | 6 |
| 0 | 7 | 7 |
| 0 | 8 | 8 |
| 0 | 9 | 9 |
| 1 | 0 | 1 |
| 1 | 1 | 2 |
| 1 | 2 | 3 |
| 1 | 3 | 4 |
| 1 | 4 | 5 |
| 1 | 5 | 6 |
| 1 | 6 | 7 |
| 1 | 7 | 8 |
| 1 | 8 | 9 |
| 1 | 9 | 10 |
| 2 | 0 | 2 |
| 2 | 1 | 3 |
| 2 | 2 | 4 |
| 2 | 3 | 5 |
| 2 | 4 | 6 |
| 2 | 5 | 7 |
| 2 | 6 | 8 |
| 2 | 7 | 9 |
| 2 | 8 | 10 |
| 2 | 9 | 11 |
| 3 | 0 | 3 |
| 3 | 1 | 4 |
| 3 | 2 | 5 |
| 3 | 3 | 6 |
| 3 | 4 | 7 |
| 3 | 5 | 8 |
| 3 | 6 | 9 |
| 3 | 7 | 10 |
| 3 | 8 | 11 |
| 3 | 9 | 12 |
| 4 | 0 | 4 |
| 4 | 1 | 5 |
| 4 | 2 | 6 |
| 4 | 3 | 7 |
| 4 | 4 | 8 |
| 4 | 5 | 9 |
| 4 | 6 | 10 |
| 4 | 7 | 11 |
| 4 | 8 | 12 |
| 4 | 9 | 13 |
| 5 | 0 | 5 |
| 5 | 1 | 6 |
| 5 | 2 | 7 |
| 5 | 3 | 8 |
| 5 | 4 | 9 |
| 5 | 5 | 10 |
| 5 | 6 | 11 |
| 5 | 7 | 12 |
| 5 | 8 | 13 |
| 5 | 9 | 14 |
| 6 | 0 | 6 |
| 6 | 1 | 7 |
| 6 | 2 | 8 |
| 6 | 3 | 9 |
| 6 | 4 | 10 |
| 6 | 5 | 11 |
| 6 | 6 | 12 |
| 6 | 7 | 13 |
| 6 | 8 | 14 |
| 6 | 9 | 15 |
| 7 | 0 | 7 |
| 7 | 1 | 8 |
| 7 | 2 | 9 |
| 7 | 3 | 10 |
| 7 | 4 | 11 |
| 7 | 5 | 12 |
| 7 | 6 | 13 |
| 7 | 7 | 14 |
| 7 | 8 | 15 |
| 7 | 9 | 16 |
| 8 | 0 | 8 |
| 8 | 1 | 9 |
| 8 | 2 | 10 |
| 8 | 3 | 11 |
| 8 | 4 | 12 |
| 8 | 5 | 13 |
| 8 | 6 | 14 |
| 8 | 7 | 15 |
| 8 | 8 | 16 |
| 8 | 9 | 17 |
| 9 | 0 | 9 |
| 9 | 1 | 10 |
| 9 | 2 | 11 |
| 9 | 3 | 12 |
| 9 | 4 | 13 |
| 9 | 5 | 14 |
| 9 | 6 | 15 |
| 9 | 7 | 16 |
| 9 | 8 | 17 |
| 9 | 9 | 18 |

2. Проверьте качество обучения нейросети (не появилось ли переобучение?). Если появилось, то уменьшите количество эпох при обучении. После оптимизации обучения сохраните в памяти полученные весовые коэффициенты (IW).

3. Создайте новую нейросеть с теми же параметрами и назовите её другим именем. Не проводя обучения, присвойте ей значения IW, полученные в предыдущем примере. В результате этого сеть должна стать обученной. Проверьте, так ли это и продемонстрируйте результат в отчёте.

4. Повторите это исследование с многослойным перцептроном, настроенным для разделения спиралей, полученных с сайта Sharky. Что нового придётся предпринять для получения успешного решения?

**Цель работы:**

Продемонстрировать основные этапы реализации нейронно-сетевого подхода для решения конкретной задачи.

**План работы:**

Можно выделить 4 основных этапа, необходимых для решения каждой конкретной задачи:

1. Подготовка данных для тренировки сети.
2. Создание сети.
3. Обучение сети.
4. Тестирование сети.
5. Моделирование сети. (Использование сети для решения поставленной задачи.)

**Выполнение задания:**

1. Рассмотрим задачу сложения чисел, где входные значения – от 0 до 9, а требуемые выходные – от 0 до 18. Буду использовать входные вектора, данные в таблице к заданию. Создам две матрицы в MatLAB. Первая будет соответствовать входным значениям, вторая – выходным. Предварительно сохраню таблицу в файл расширения .txt

>> open('mynet.txt');

>> a = load('mynet.txt');

>> for i = 1:100

P(1,i)=a(i,1);

P(2,i)=a(i,2);

T(i,1)=a(i,3);

end;

>> T=T';

С помощью этой программы формируется матрица T из M=100 входных векторов-столбцов, каждый из которых сформирован из 2 точек исходной функции с выбранными из матрицы значениями параметров A, B, и матрица P эталонов из 100 эталонных векторов-столбцов, каждый из которых сформирован из 2 соответствующих эталонных значений(взят из таблицы). Матрицы T и P будут использованы при обучении сети.

Перейдем к созданию сети.

Для решения поставленной задачи сформируем трехслойную сеть обратного распространения, включающую 1 нейрон во входном слое (по числу компонент входного вектора) с передаточной функцией logsig, 15 нейронов во втором слое с передаточной функцией logsig и 2 нейрона в выходном слое (по числу компонентов выходного вектора) с передаточной функцией purelin. При этом в качестве обучающего алгоритма выбран алгоритм Levenberg-Marquardt (trainlm). Этот алгоритм обеспечивает быстрое обучение, но требует много ресурсов.

>> net=newff(minmax(P),[100,15,2],{'logsig' 'logsig' 'purelin'},'trainlm');

Первый аргумент - матрица Mx2 минимальных и максимальных значений компонент входных векторов вычисляется с помощью процедуры minmax.

Результатом выполнения процедуры newff является объект – нейронная сеть net заданной конфигурации.

Приступим к обучению нейросети.

Задаем функцию оценки функционирования sse.

>> net.performFcn='sse';

В этом случае в качестве оценки вычисляется сумма квадратичных отклонений выходов сети от эталонов. Задаем критерий окончания обучения – значение отклонения, при котором обучение будет считаться законченным:

>> net.trainParam.goal=0.01;

Задаем максимальное количество циклов обучения. После того, как будет выполнено это количество циклов, обучение будет завершено:

>> net.trainParam.epochs=1000;

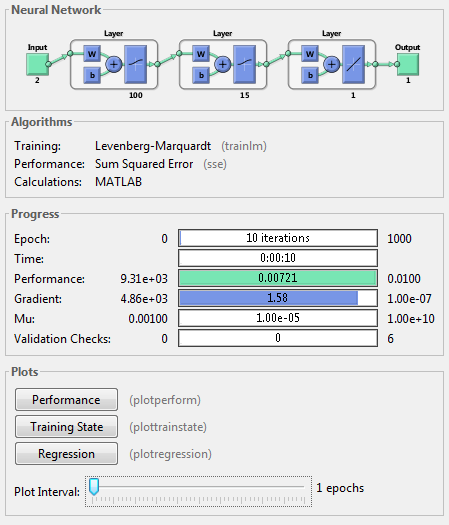
Теперь можно начинать обучение:

>> [net, tr]=train(net, P, T);

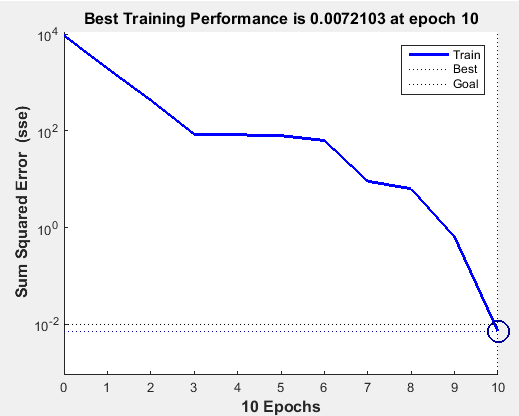
Процесс обучения иллюстрируется графиком зависимости оценки функционирования от номера цикла обучения.

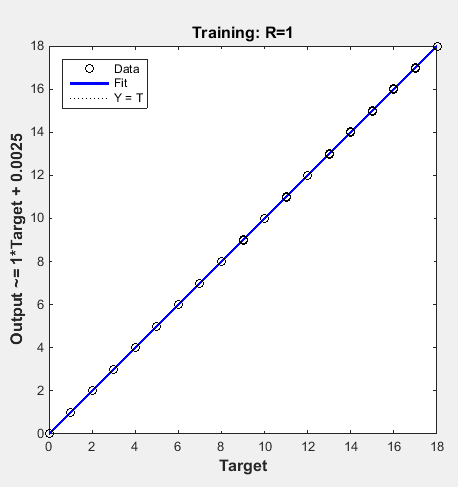
>> save nn1 net;

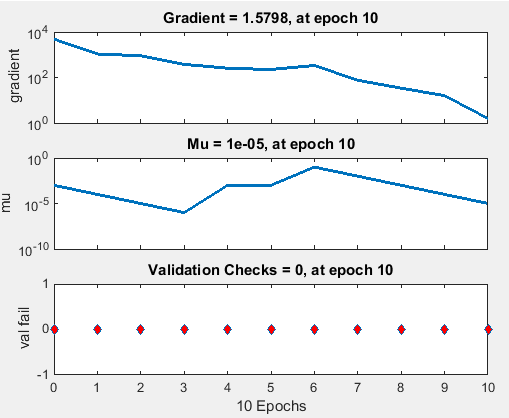
Для оценки достоверности результатов работы сети можно воспользоваться результатами регрессионного анализа, полученными при сравнении эталонных значений со значениями, полученными на выходе сети когда на вход поданы входные векторы тестового массива. В среде MATLAB для этого можно воспользоваться функцией postreg. Следующий набор команд иллюстрирует описанную процедуру:



Процесс обучения иллюстрируется графиком зависимости оценки функционирования от номера цикла обучения.







Таким образом, обучение сети окончено. Теперь эту сеть можно сохранить в файле nn1.mat:

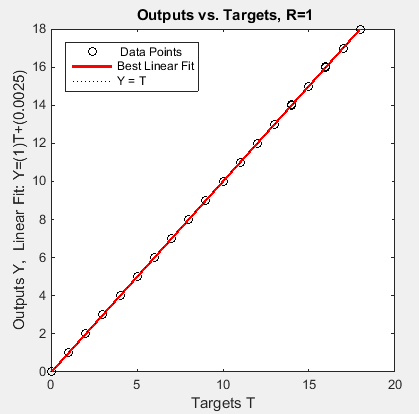
>> save nn1 net;

Теперь проведем тестирование сети:

>> mas1 - создание тестового массива P

>> y=sim(net,P); - обработка тестового массива

>> [m,b,r]=postreg(y(1,:),T(1,:)); - регрессионный анализ результатов обработки.



Как видно из рисунков, наша сеть отлично решает поставленную задачу для всех выходных параметров. Сохраним обученную сеть net на диске в файл nn1.mat

>> save nn1 net % Сохранение сети nn1.

Моделирование сети.

>> A = 1;

>> B = 6;

>> p = A+B;

>> Y=sim(net,p)

Получим:

Y =

6.9994

Что весьма близко к правильному результату = 7.

Проверка на случайных значениях

>> ex1 = randi([0, 9], 2, 10);

8 8 1 1 7 1 1 0 8 9

4 6 0 3 7 7 7 3 5 3

>> Y1 = sim(net, ex1)

11,9990472859768 13,9883294105931 1,00018996304717 4,00068330416121 13,9806662183011 7,99976972271038 7,99976972271038 2,99986437841973 12,9788466173138 11,9938564254442

Что тоже близко к необходимым результатам.

Создаётся новая сеть и без обучения ей присваиваются весовые коэффициенты, полученные при обучении старой сети:

>> new\_net=newff(minmax(P),[100,15,1],{'logsig' 'logsig' 'purelin'},'trainlm');

>> new\_net.iw{1} = net.iw{1}

>> ex2 = randi([0, 9], 2, 10);

3 9 5 9 9 3 2 2 7 4

8 6 1 8 6 7 6 5 7 1

Y2 = sim(new\_net, ex2);

-2,90285873346811 -2,41738331318904 -3,61164993183093 -2,42990415760246 -2,41738331318904 -2,84060197376762 -3,08206971874937 -3,00383816347384 -2,24150280019230 -3,53688339120421

Сеть была обучена неправильно. Необходимо найти ошибку.

4. Разберемся, какая сеть сможет разделить спирали в Sharky, чтобы затем создать аналогичную сеть в matLAB.

