МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОННОЙ ТЕХНИКИ

Институт системной и программной инженерии
и информационных технологий (Институт СПИНТех)

Лабораторная работа № 3

Создание однонаправленной нейронной сети с помощью нейронно-сетевого инструментария MATLAB

Выполнил:

Никаноров В.Д. гр. ПИН-41

Проверил преподаватель:

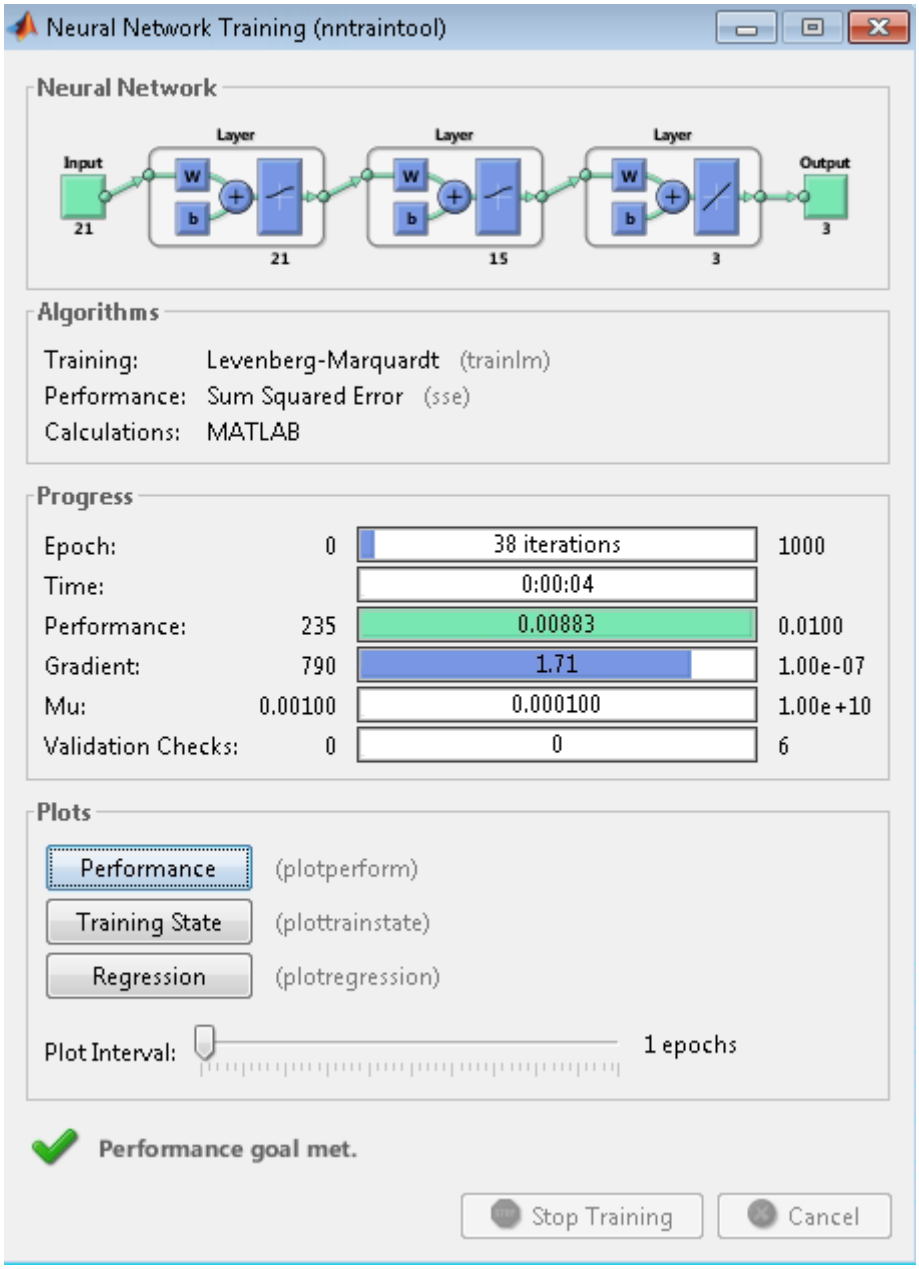
проф., д.ф.-м. н. Рычагов М.Н.

Задание №1

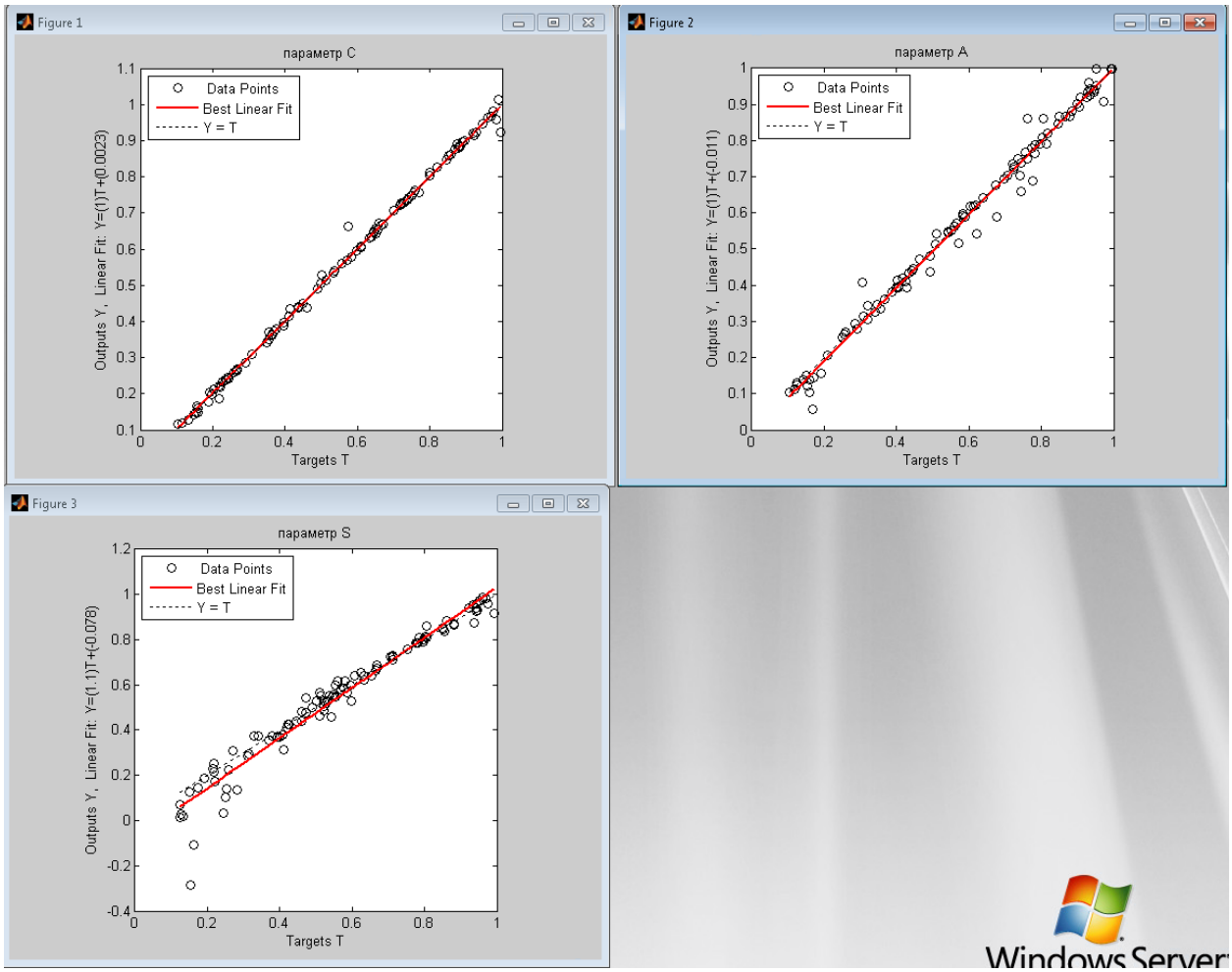
Исследовать влияние шума в исходных данных на результаты обучения нейронной сети. Для этого к исходному массиву данных прибавить случайные числа из диапазонов (0 – 0.01; 0 – 0.05; 0 – 0.1; 0 -0.2). Провести процедуру обучения и протестировать сеть.

Пример обучения и использования нейронной сети без шумов.

1.1 Обучение



1.2. Тестирование



1.3. Использование

Полученные сетью параметры для значения функции при *C=0.2, A=0.8, S=0.7*

```
Y =

    0.2064
    0.8073
    0.6704
```

1.4 Исследование влияния шумов

Пояснения к коду:

Для выполнения задания были использованы следующие функции (находящиеся в одноимённых m-файлах):

generate_data – создаёт массивы входных и выходных параметров, добавляя к выходному параметру случайные шумы (от 0 до указанной величины).

train_net – создаёт и обучает нейронную сеть на указанных данных.

test – оценивает качество работы сети с помощью регрессионного анализа. Для получения тестовой выборки, отличной от обучающей выборки, вызывается функция *generate_data* с нулевым значением шумов. Выводит коэффициенты регрессии и строит графики для каждого из параметров A C и S.

use – использует нейронную сеть для получения параметров A C и S, и сравнивает их с параметрами, фактически использованными для построения функции. (т. е. демонстрирует работу нейронной сети на одном примере)

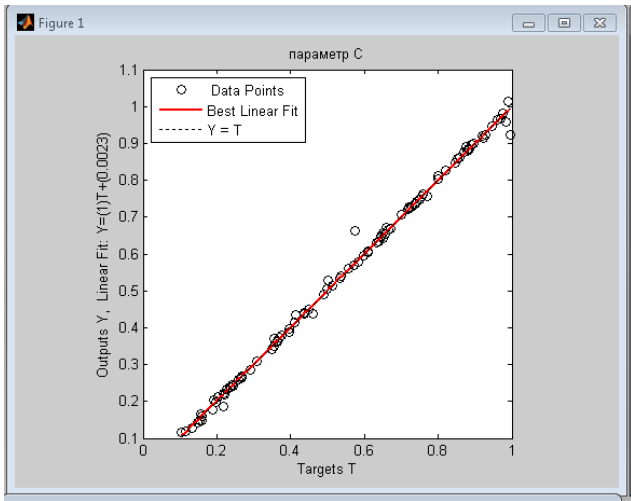
run – для удобства использования объединяет генерацию данных с указанным уровнем шумов, обучение сети, тестирование и использование обученной сети для обработки функции с параметрами *C=0.2, A=0.8, S=0.7*

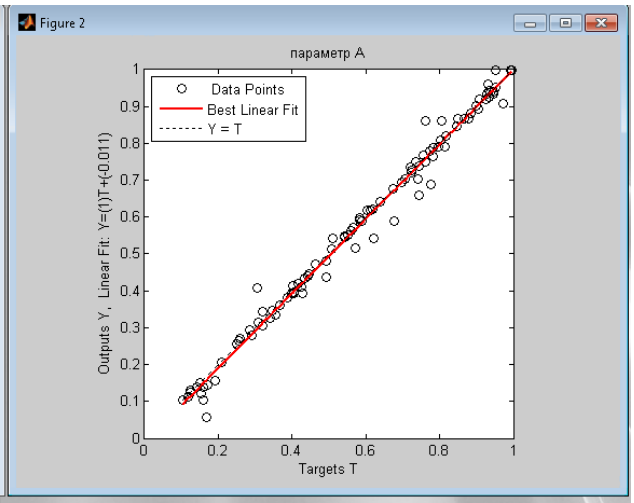
Далее приведены результаты запуска для различных уровней шумов:

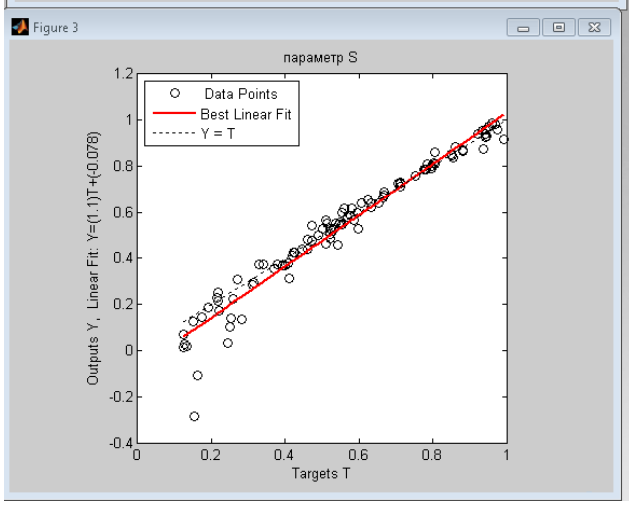
- без шумов (0)
- до 0,01
- до 0,05
- до 0,1
- до 0,2

Без шумов (0)

```
>> run(0)
коэффициенты регрессии для сети
для параметра C: 9.986098e-01
для параметра A: 9.937143e-01
для параметра S: 9.755429e-01
========================
истинные значения:          C=0.200000  A=0.800000  S=0.700000
значения полученный сетью:  C=0.203419  A=0.801199  S=0.713947
```

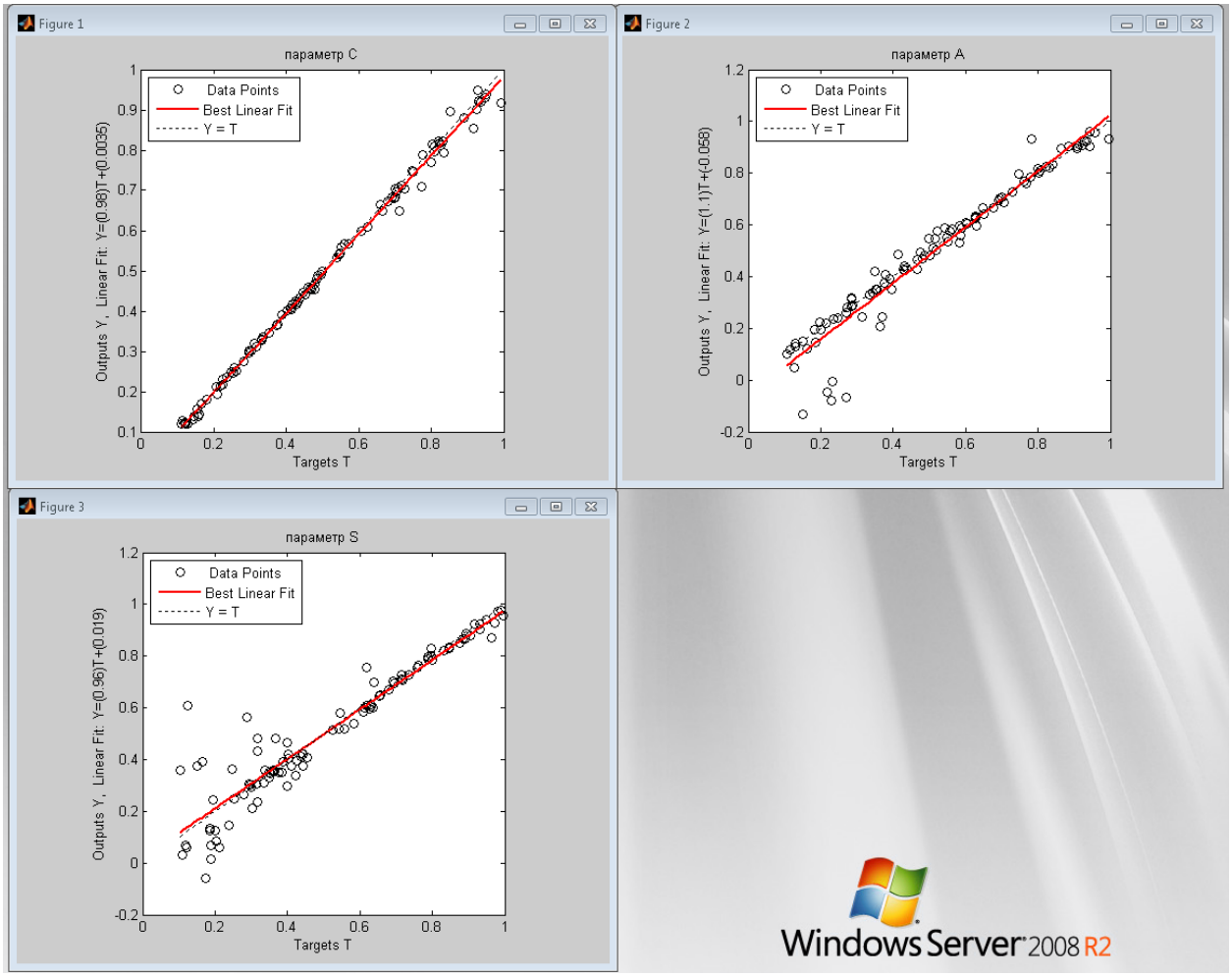






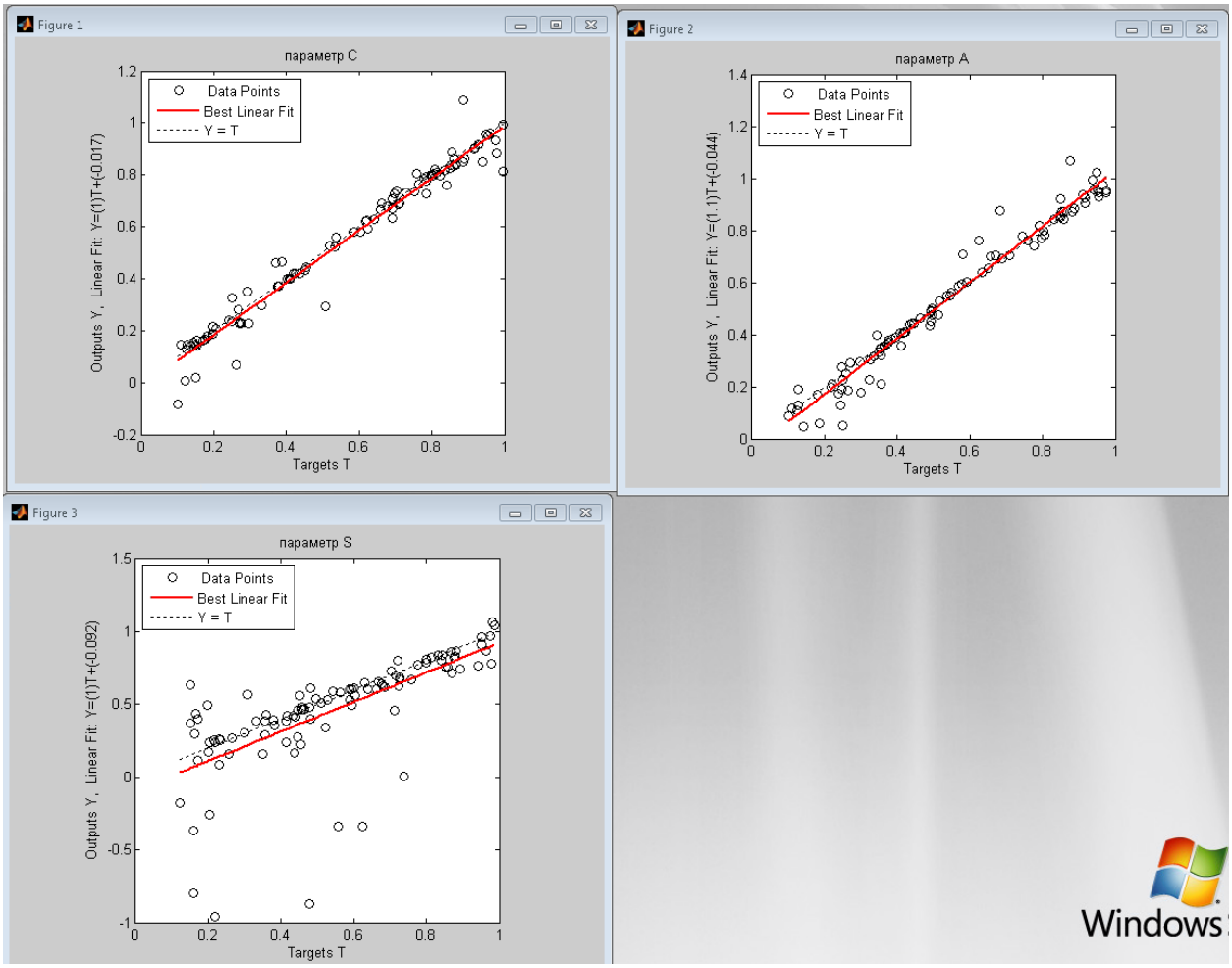
Шумы до 0,01

```
>> clear
>> run(0.01)
коэффициенты регрессии для сети
для параметра C: 9.981403e-01
для параметра A: 9.685935e-01
для параметра S: 9.409088e-01
========================
истинные значения:          C=0.200000  A=0.800000  S=0.700000
значения полученный сетью:  C=0.194096  A=0.829984  S=0.741603
```



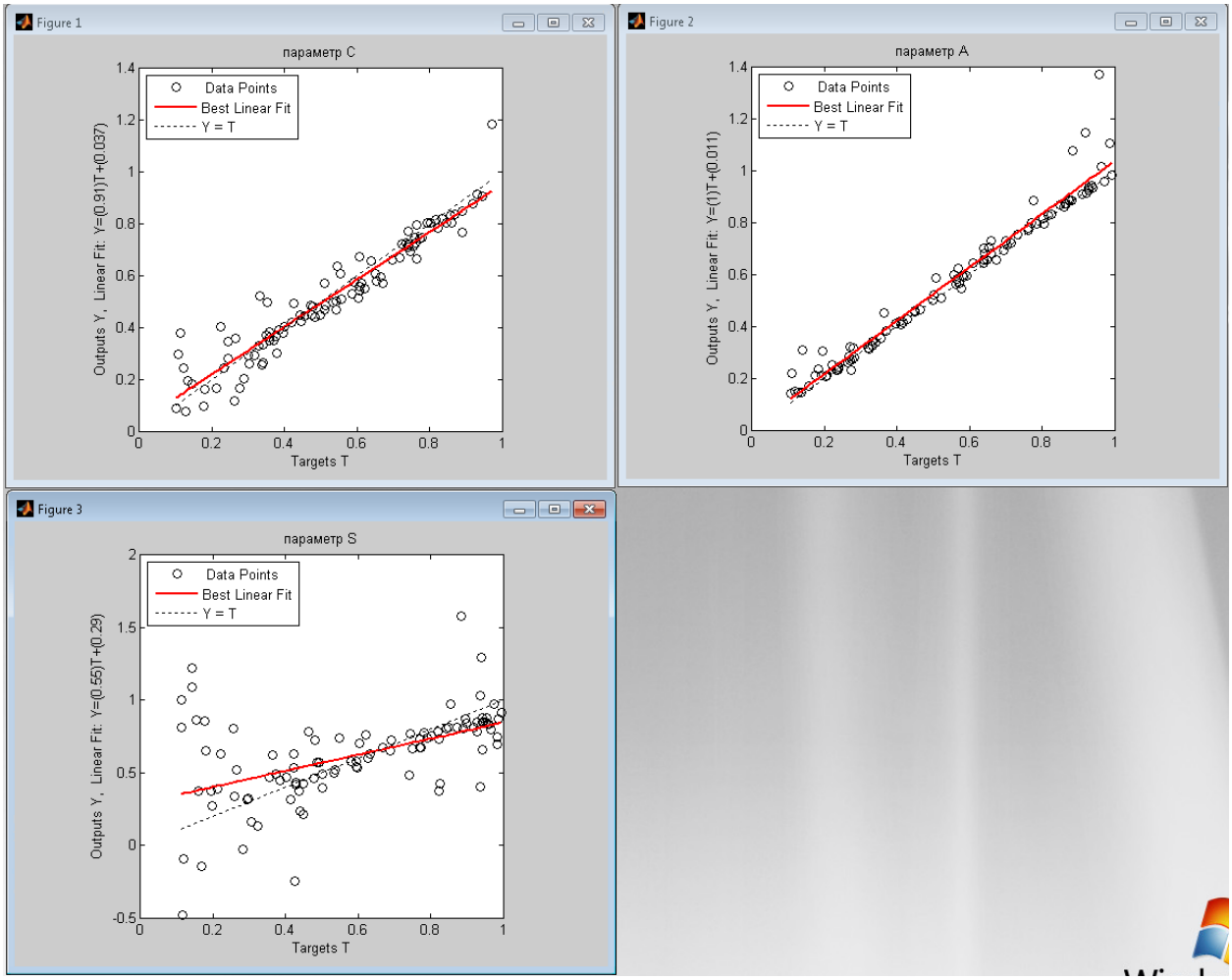
Шумы до 0,05

```
>> clear
>> run(0.05)
коэффициенты регрессии для сети
для параметра C: 9.821081e-01
для параметра A: 9.839622e-01
для параметра S: 6.823202e-01
========================
истинные значения:          C=0.200000  A=0.800000  S=0.700000
значения полученный сетью:  C=0.197138  A=0.829423  S=0.689843
```



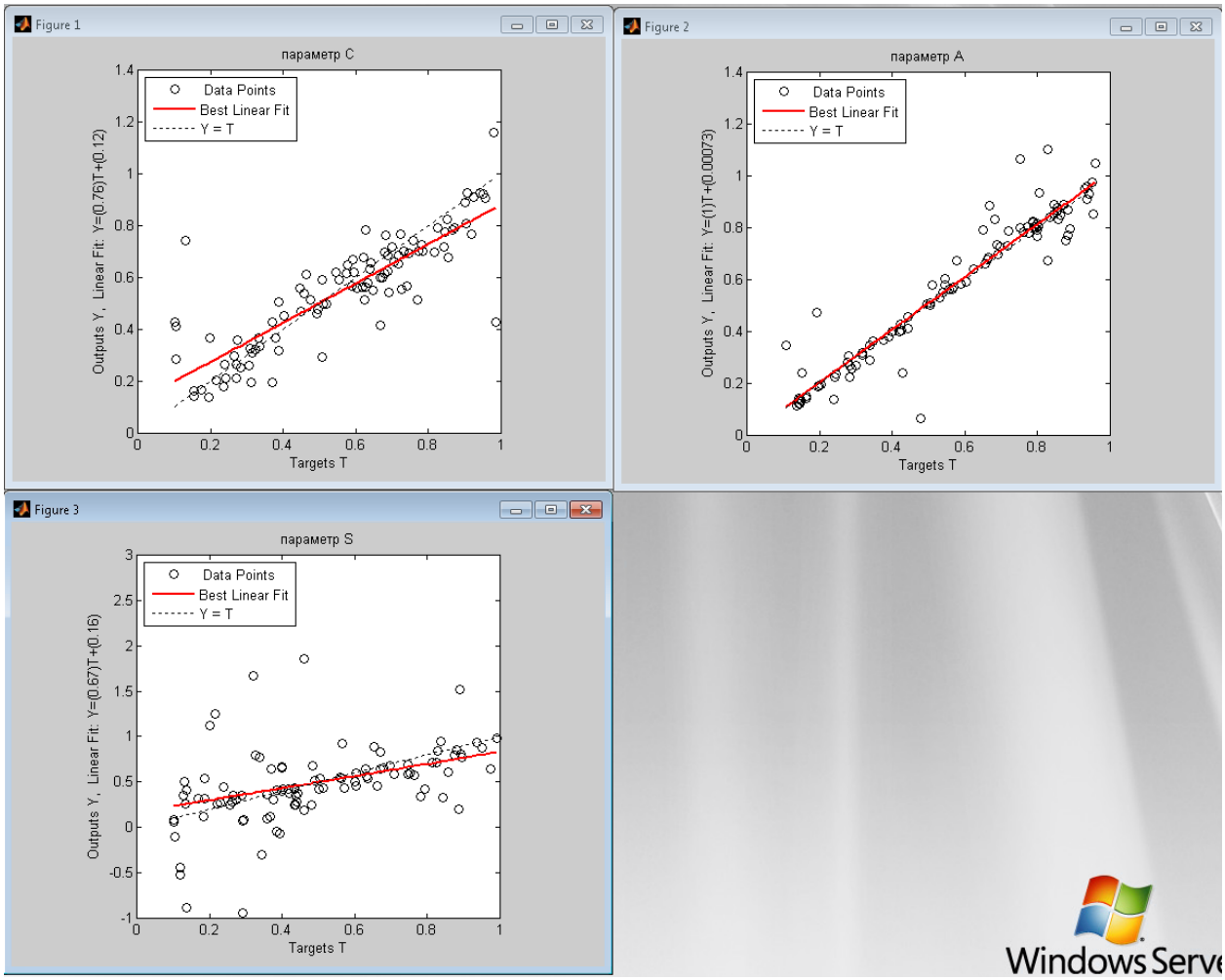
Шумы до 0,1

```
>> clear
>> run(0.1)
коэффициенты регрессии для сети
для параметра C: 9.534104e-01
для параметра A: 9.791136e-01
для параметра S: 5.096955e-01
========================
истинные значения:          C=0.200000  A=0.800000  S=0.700000
значения полученный сетью:  C=0.188815  A=0.850294  S=0.700479
```



Шумы до 0,2

```
>> clear
>> run(0.2)
коэффициенты регрессии для сети
для параметра C: 8.502798e-01
для параметра A: 9.489414e-01
для параметра S: 3.583312e-01
========================
истинные значения:          C=0.200000  A=0.800000  S=0.700000
значения полученный сетью:  C=0.240834  A=0.937684  S=1.015851
```



Вывод: с увеличением уровня шумов происходит уменьшение всех коэффициентов регрессии и, следовательно, качества работы нейронной сети. Это происходит, потому что при обучении на зашумлённых данных нейросеть "заучивает" шумы вместе с полезными сигналами, что затрудняет нахождение закономерностей, важных для решения задачи. В результате нейронная сеть хорошо работает на тренировочном наборе данных (при обучении во всех случаях была достигнута требуемая точность: *net.trainParam.goal=0.01*), однако хуже показывает себя на тестовых данных, потому что излишне адаптировалась к шуму.

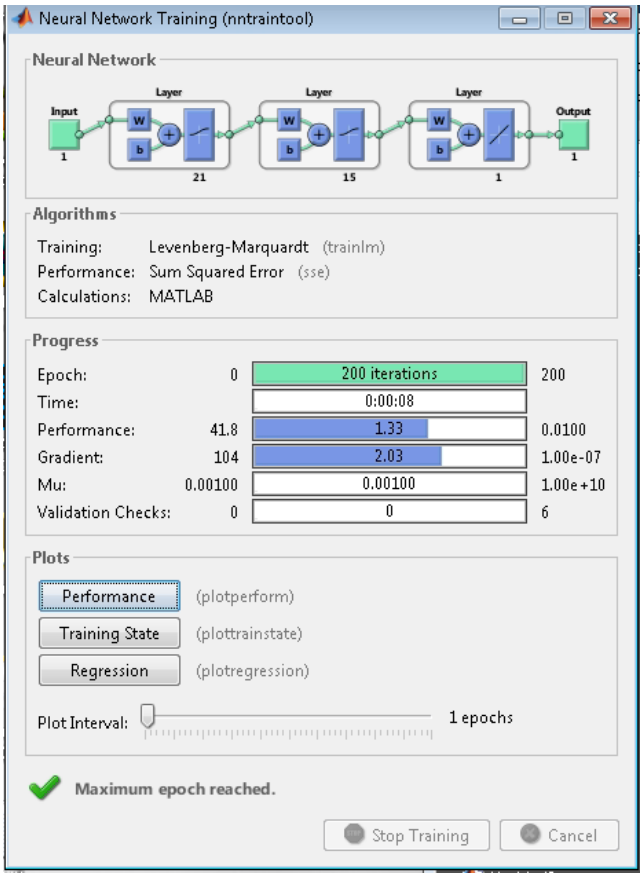
Задание №2

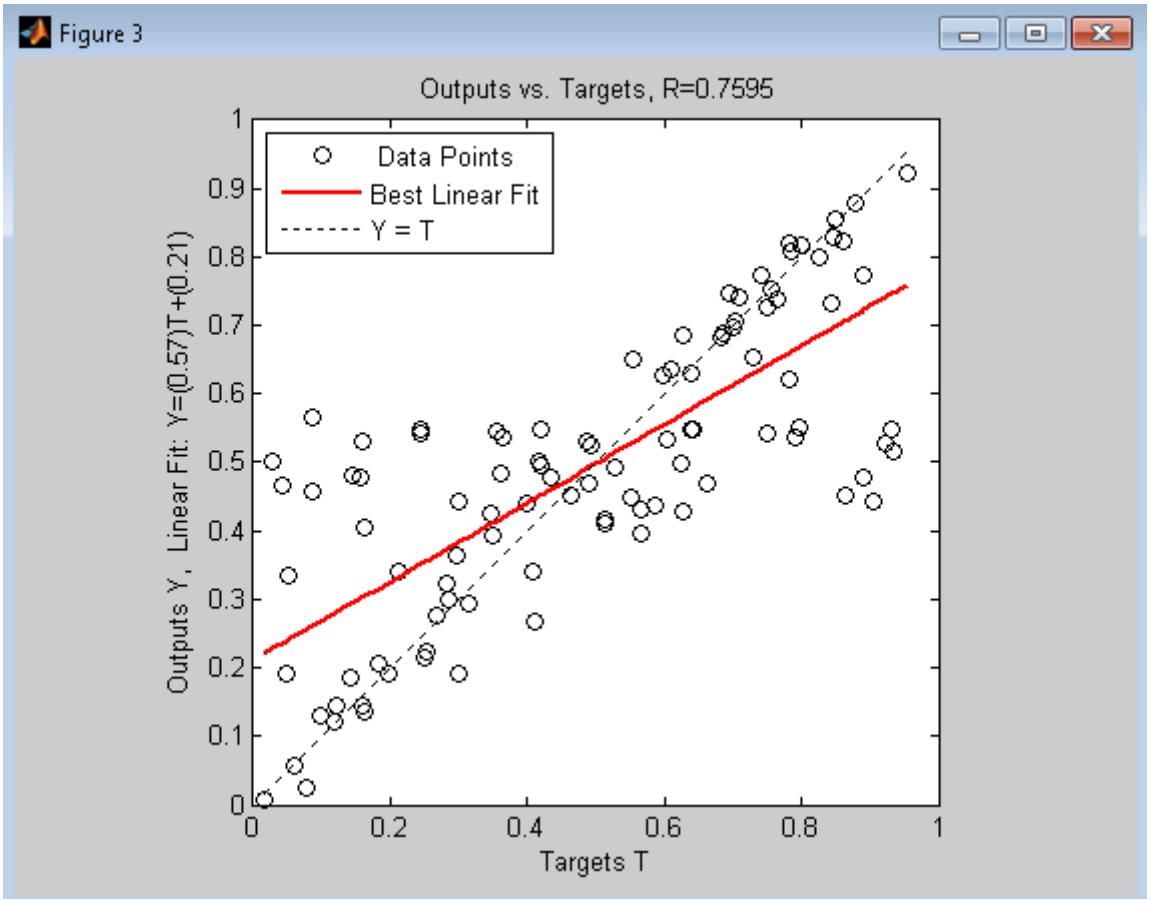
Сформировать исходный массив и массив эталонов из случайных чисел и провести обучение сети. Прокомментировать результаты

Код решения находится в файле *random.m*

Поскольку требуемая точность вероятно никогда не будет достигнута, количество итераций было сокращено до 200, с целью уменьшения времени обучения.

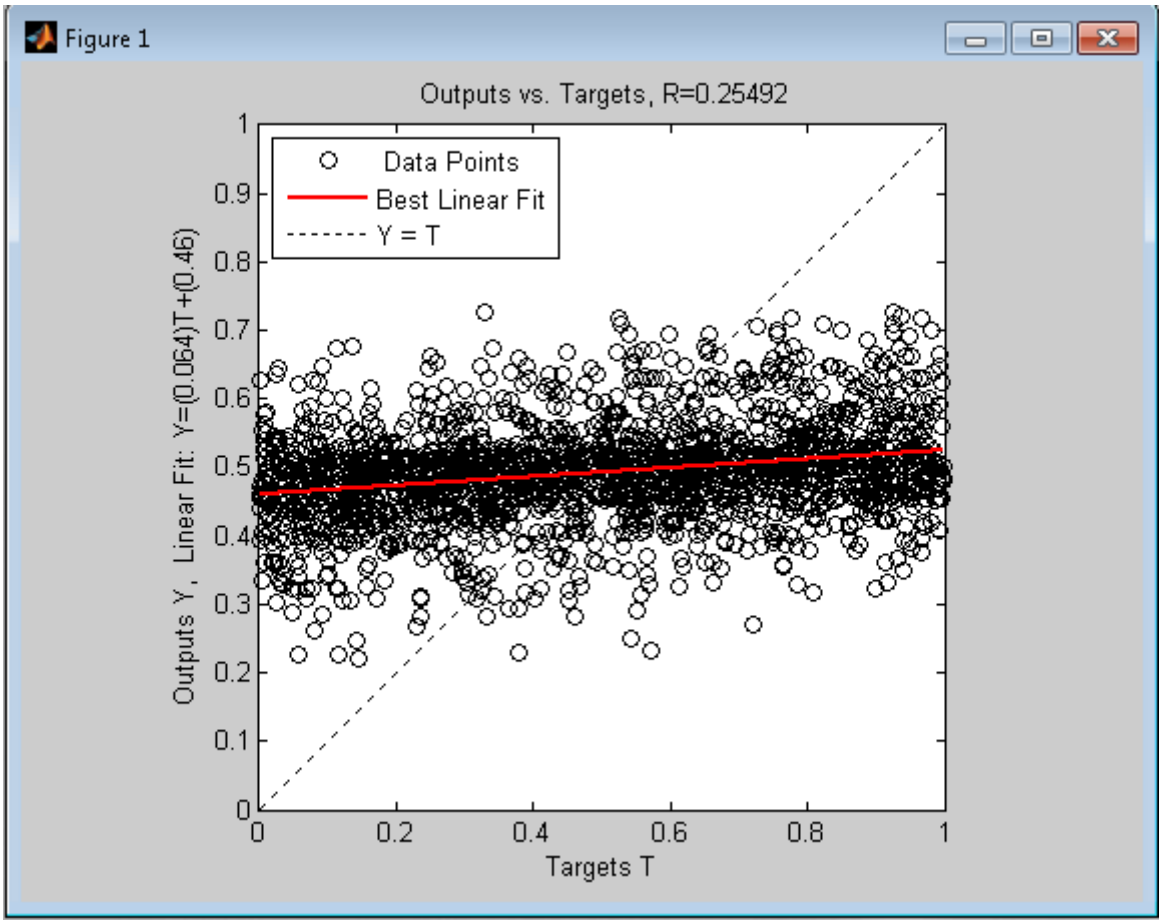
Результаты обучения:





Требуемая точность не была достигнута после максимального числа эпох обучения, а коэффициент регрессии оказался равен 0,75. Хотя это не высокий результат для тренировочной выборки, тем не менее, он является довольно большим для случайных чисел. Это является следствием переобучения на тренировочных данных, так как благодаря большому числу внутренних коэффициентов, нейросеть “выучила” случайные соответствия между случайными входными и выходными данными. Возможность переобучения обусловлена большой глубиной нейронной сети (21-15-1) относительно величины обучающей выборки (100). При увеличении обучающей выборки при неизменной архитектуре сети, способность к переобучению снижается.

Результаты обучения для выборки в 2000 элементов:



Коэффициент регрессии снизился до 0,25.