

Министерство образования Республики Беларусь
Учреждение образования
“Брестский государственный технический университет”
Кафедра интеллектуально-информационных технологий

Лабораторная работа
“Моделирование нейронных сетей в MATLAB”

Выполнил:
студент 3 курса
группы ИИ-21
Кирилович А. А.
Проверил:
Рыжов А. С.

Брест 2023

Цели работы:

- 1) изучение средств и методов MATLAB, пакетов Neural Network Toolbox и Simulink для моделирования и исследования нейронных сетей;
- 2) получение умений и навыков:
 - в освоении базовых приемов моделирования и исследования нейронных сетей в среде MATLAB;
 - в применении нейронных сетей для аппроксимации функций;
 - в анализе полученных результатов.

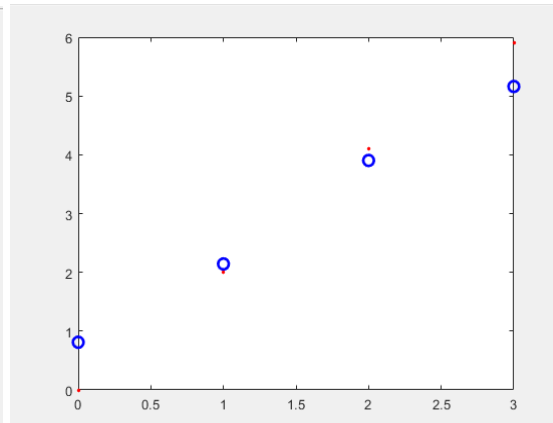
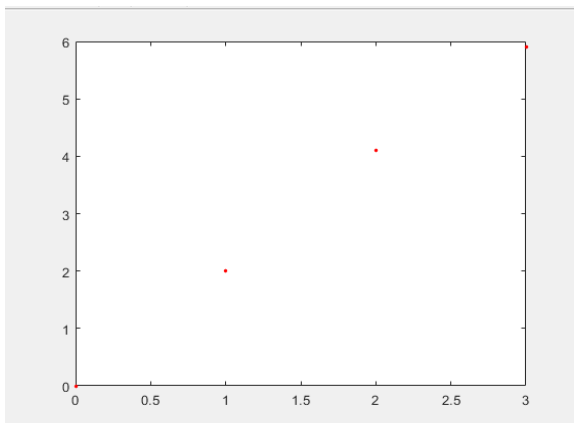
Задание для лабораторной работы

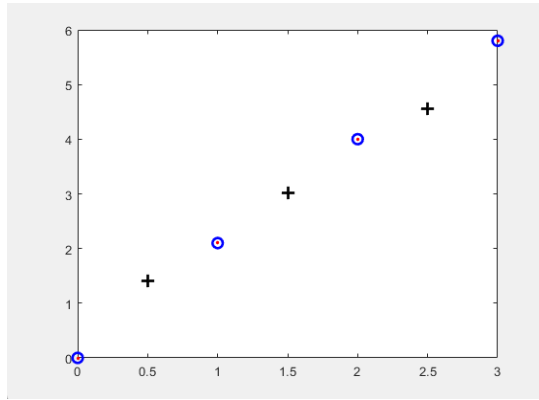
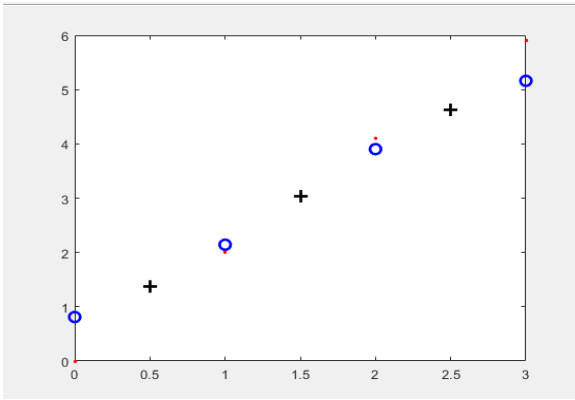
Задание 1.

1. Создать обобщенно-регрессионную НС и радиальную базисную сеть с нулевой ошибкой, для чего выполнить действия, указанные в п. 4.3 в **примерах 1, 2**.
2. Восстановить зависимость по имеющимся экспериментальным данным с использованием НС, для чего выполнить действия, указанные в п. 4.3 в **примере 3**. Результаты внести в отчет в виде картинок изображений, полученных в MATLAB. Созданную сеть сохранить.

Пример 1.

```
>> P = 0:3;
>> T = [0.0 2.0 4.1 5.9];
>> net = newgrnn(P, T);
>> gensim(net)
ans =
    'untitled'
>> plot(P,T,'*r','MarkerSize',2,'LineWidth',2)
>> hold on
>> V = sim(net,P);
>> plot(P,V,'ob','MarkerSize',8, 'LineWidth',2)
>> P1 = 0.5:2.5;
>> Y = sim(net,P1);
>> plot(P1,Y,'+k','MarkerSize',10,'LineWidth',2)
>> Y = sim(net, 0:0.5:3)
Y =
    0.4865    0.8779    1.4818    2.2900    3.2000    4.0497    4.7101
>> net = newgrnn(P,T,0.1);
>> Y = sim(net, 0:0.5:3)
Y =
    0.0000    0.5500    1.1000    2.1500    3.2000    4.4500    5.7000
>> plot(P,T,'ob','MarkerSize',8, 'LineWidth',2)
```

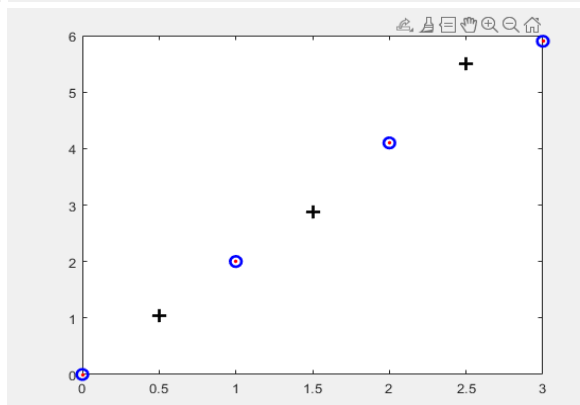
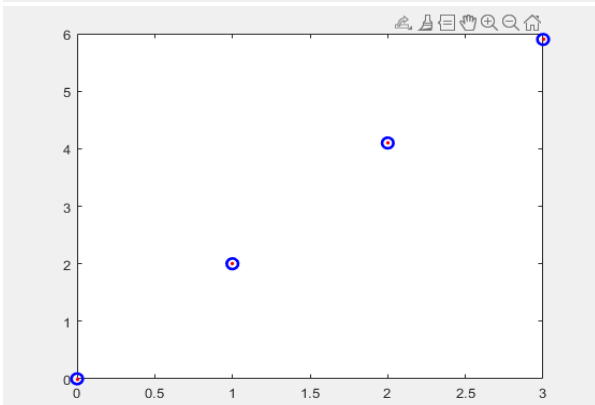
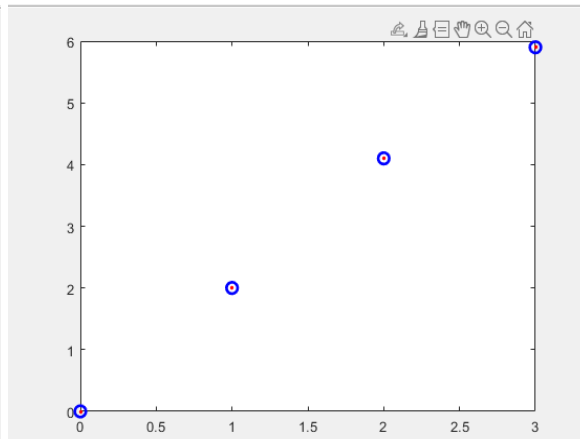
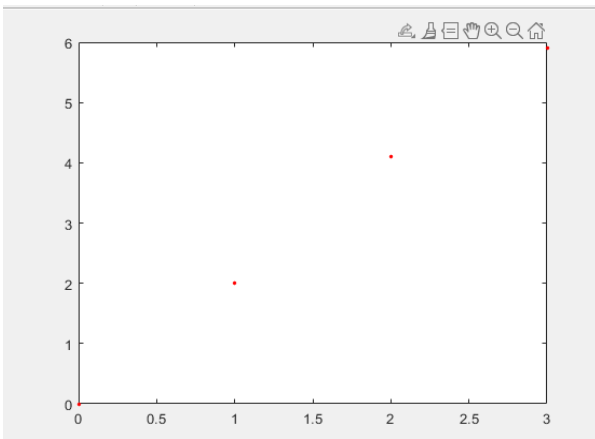




Пример 2.

```
>> P = 0:3;
>> T = [0.0 2.0 4.1 5.9];
>> net = newrbe(P,T);net.layers{1}.size
ans =
     4
>> plot(P,T,'*r','MarkerSize',2,'LineWidth',2)
hold on
>> V = sim(net,P);
>> plot(P,V,'ob','MarkerSize',8, 'LineWidth',2)

>> P1 = 0.5:2.5;
>> Y = sim(net,P1)
Y =
     1.0346     2.8817     5.5053
>> plot(P1,Y,'+k','MarkerSize',10, 'LineWidth',2)
>> Y = sim(net, 0:0.5:3)
Y =
     0.0000     1.0346     2.0000     2.8817     4.1000     5.5053     5.9000
```



Пример 3.

```
>> x = [-1 -0.8 -0.5 -0.2 0 0.1 0.3 0.6 0.9 1];
>> y = [1 0.64 0.25 0.04 0 0.01 0.09 0.36 0.81 1];
>> a=newgrnn(x,y,0.01); % Создание НС с отклонением 0.01
>> Y1 = sim(a,[-0.9 -0.7 -0.3 0.4 0.8]) % Опрос НС
```

```

Y1 =
    0.8200    0.6400    0.0400    0.0900    0.8100
>> a=newrbe(x,y);
>> Yl = sim(a,[-0.9 -0.7 -0.3 0.4 0.8])
Yl =
    0.8100    0.4900    0.0900    0.1600    0.6400
>> save('3.3');

```

Задание 2.

1. Создать НС в Fuzzy Logic Toolbox с помощью графического интерфейса пользователя (GUI, или ГИП) для выполнения операции $y = x^2$ по исходным данным и указаниям п. 4.4.1 (см. **пример 4**).
2. Провести обучение нейронной сети и проверить ее работу по п. 4.4.2 и п. 4.4.3 (см. **примеры 5, 6**).
3. Вывести структурную схему сети и получить информацию о весах и смещениях непосредственно в рабочем окне системы.

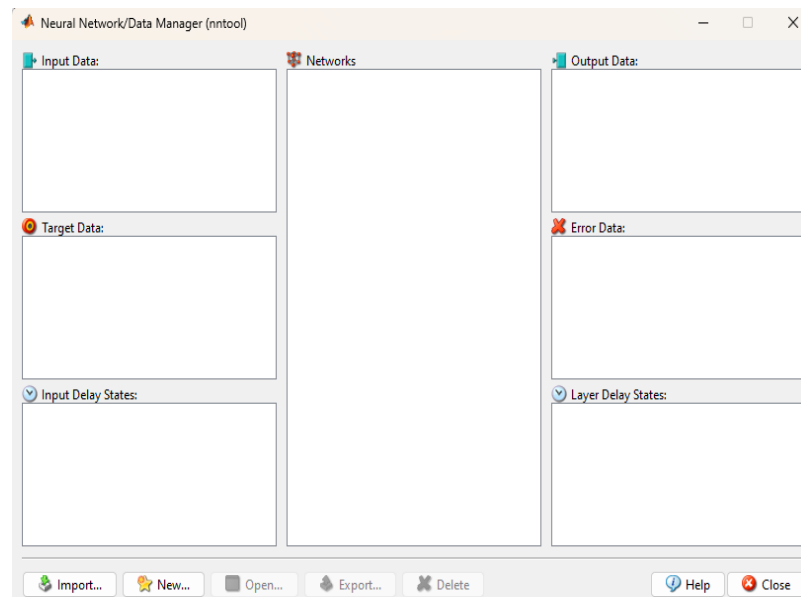
Результаты внести в отчет в виде картинок изображений, рисунков, схем, полученных в MATLAB.

Пример 4

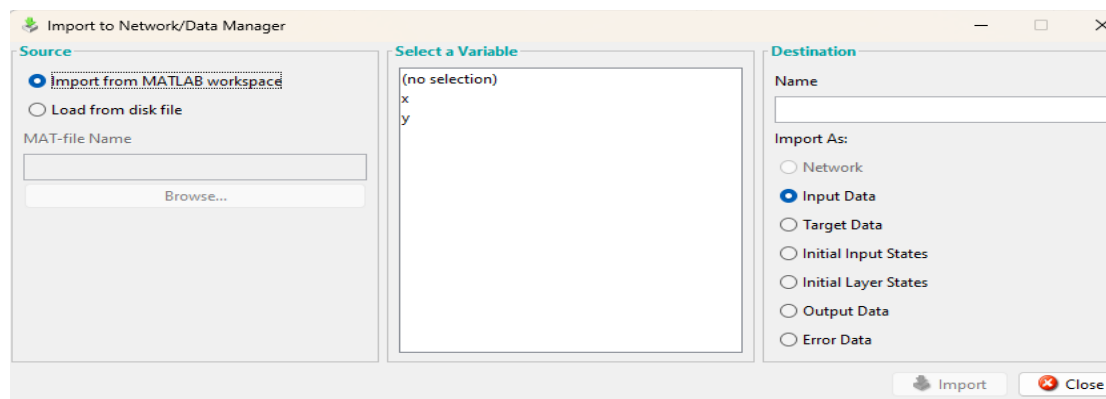
```

>> x = [-1 -0.8 -0.5 -0.2 0 0.1 0.3 0.6 0.9 1];
>> y = [1 0.64 0.25 0.04 0 0.01 0.09 0.36 0.81 1];
>> nntool

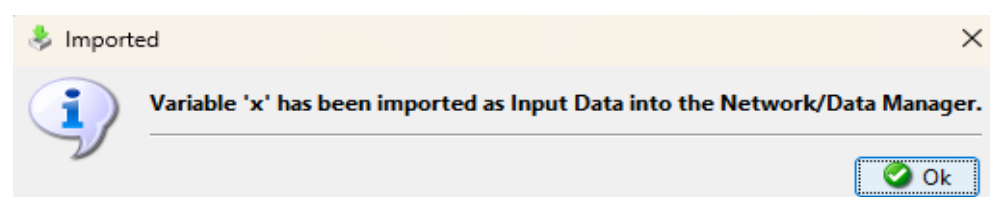
```



После выполнения команды `nntool` открываются окна создания нейронной сети **Network/Data Manager**

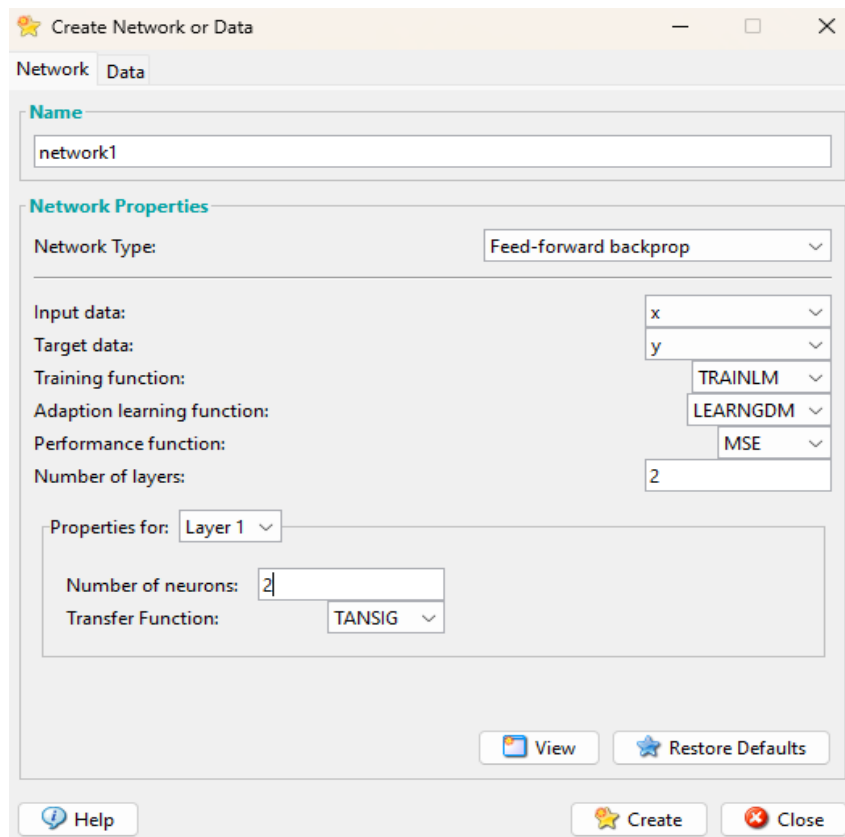


Используя кнопку **Import**, откроем окно **Import to Network/Data Manager** выберем вектор входа **x** в качестве входных данных **Input Data** и нажмем кнопку **Import**.



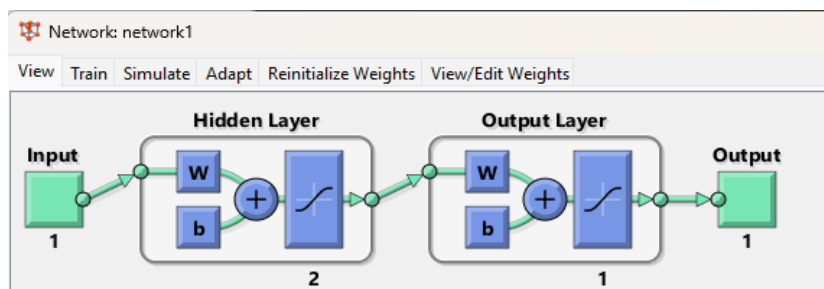
Аналогичную операцию проделаем для вектора целей **y** и целевых данных **Target Data**.

Создадим новую нейронную сеть. Для этого в окне **Network/Data Manager** нажмем кнопку **New**.



В открывшемся окне **Create Network or Data** выберем нейронную сеть типа **feed-forward backprop** с прямой передачей сигнала и с обратным распространением ошибки. По умолчанию будет имя **network1**. В качестве входных данных **Input Data** выберем **x**, а в качестве целевых данных **Target Data** — **y**. Количество нейронов первого слоя установим равным двум. Остальные установки при создании сети оставим по умолчанию.

Создание сети завершим нажатием кнопки **Create**.

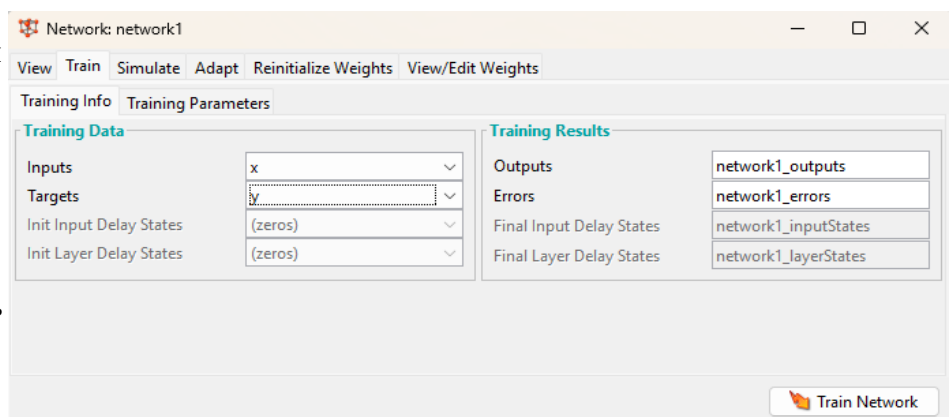


После этого в окне **Network/Data Manager**, в области **Network**, появится имя новой созданной сети — **network1**. Уже можно увидеть архитектуру нашей нейросети.

Пример 5

Для обучения созданной сети выбирается вкладка **Train** в панели **Network: network1** и открывается новая диалоговая панель. Панель имеет две вкладки: **Training Info** и **Training Parameters**

Применяя эти вкладки, можно установить имена последовательностей входа и цели (на вкладке **Training Info** — в левой ее части необходимо указать **x** и **y**), а также значения параметров процедуры обучения (на вкладке **Training Parameters**; в условиях примера сохраним значения по умолчанию).



Для обучения созданной сети нажмем кнопку **Train Network**, в результате чего откроется окно **Neural Network Training**. Качество обучения сети на выбранной обучающей последовательности отображается графиком.

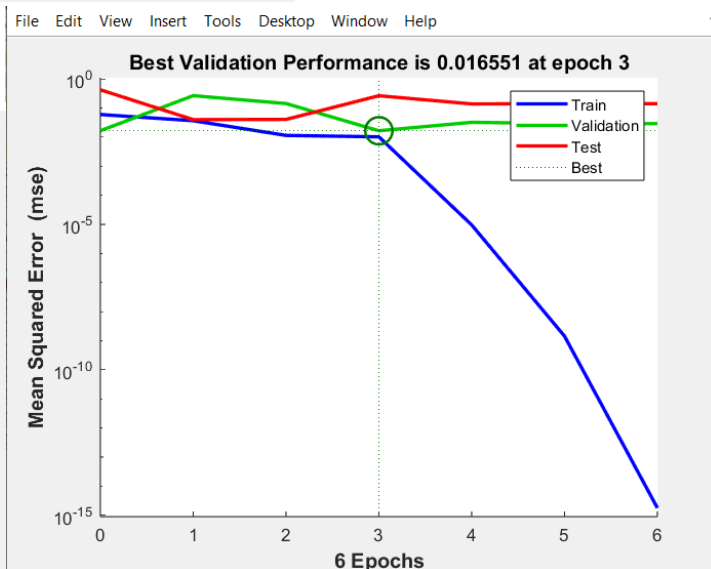
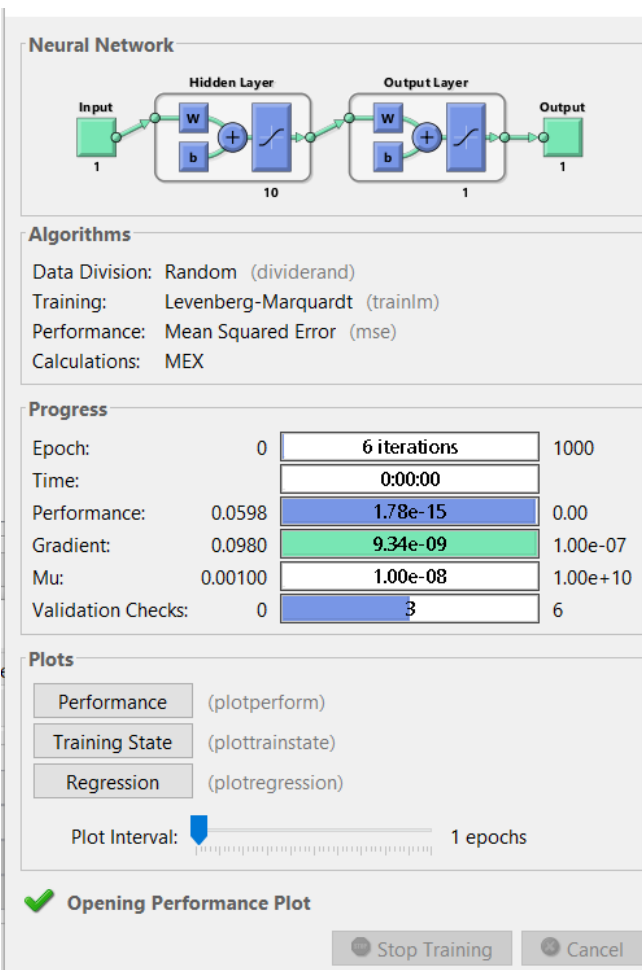
Network: network1

View Train Simulate Adapt Reinitialize Weights View/Edit Weights

Training Info Training Parameters

showWindow	true	mu	0.001
showCommandLine	false	mu_dec	0.1
show	25	mu_inc	10
epochs	1000	mu_max	10000000000
time	Inf		
goal	0		
min_grad	1e-07		
max_fail	6		

Train Network



Видно, что к концу процесса обучения ошибка становится очень малой. Результаты обучения можно просмотреть в окне **Network/Data Manager**, активизируя имена последовательностей выходов **network1_outputs** или ошибок **network1_errors** двойным щелчком левой кнопки мыши.

Пример 6

Export from Network/Data Manager

Select Variables

- x
- y
- network1
- network1_outputs
- network1_errors

Select one or more variables. Then [Export] the variables to the MATLAB workspace or [Save] them to a disk file

Select All Select None Export Save Close

При необходимости можно экспортировать созданную нейронную сеть в рабочую область системы MATLAB, нажав кнопку **Export**, и получить информацию о весах и смещениях непосредственно в рабочем окне системы, выполнив команды:

```

>> network1.IW{1,1},network1.b{1}
ans =
    3.2099
    1.9136
ans =
   -2.7287
    2.0765
>> network1.IW{2,1},network1.b{2}
ans =
     []
ans =
    3.0384

```

Основной функцией для формирования нейросетевых моделей в Simulink является функция **gensim**, записываемая в форме **gensim (net,st)**, где **net** — имя созданной НС, **st** — интервал дискретизации (если НС не имеет задержек, ассоциированных с ее входами или слоями, значение данного аргумента устанавливается равным –1).

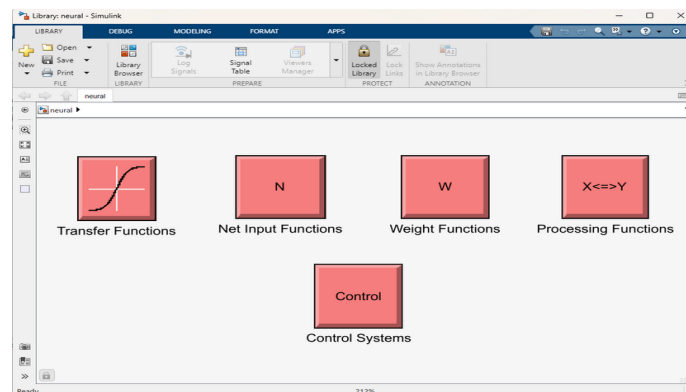
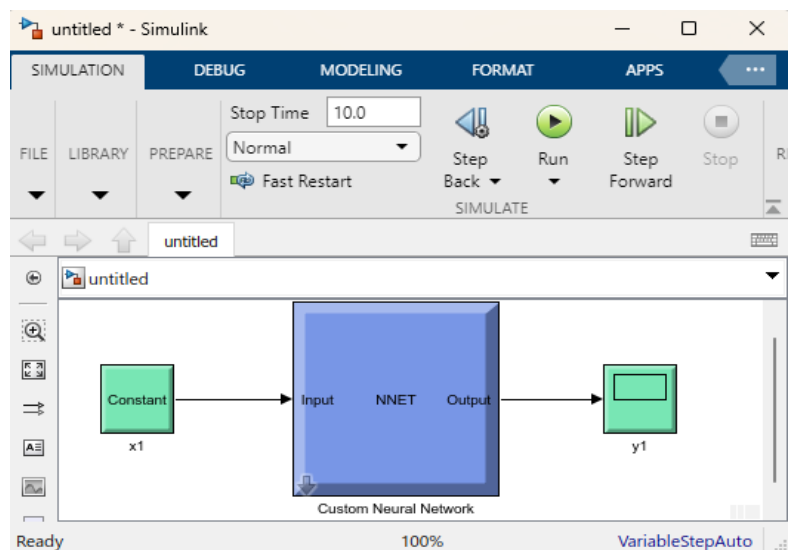
Теперь можно построить модель НС в среде Simulink и отобразить ее схему, используя команду:

```
>> gensim(network1)
```

При выполнении команды **gensim** кроме схемы НС открывается окно **Library:neural** с набором блоков, которые можно использовать для внесения изменений в схему.

Пакет **Neural Network Toolbox** содержит ряд блоков, которые могут быть либо непосредственно использованы для построения нейронных сетей в среде **Simulink**, либо применяться вместе с рассмотренной выше функцией **gensim**.

Для вызова этого набора блоков в командной строке MATLAB необходимо набрать команду **neural**, после выполнения которой появляется окно Library:neural



Пример 7.

Входной и целевой векторы имеют вид: $p = [1 \ 2 \ 3 \ 4 \ 5]$; $t = [1 \ 3 \ 5 \ 7 \ 9]$;

Задание 3.

Провести моделирование нейронных сетей при помощи Simulink по п. 4.5.5. Исходные данные взять из **примера 7**. Результаты внести в отчет в виде картинок изображений, рисунков, схем, полученных в MATLAB.

```

>> p = [1 2 3 4 5];
>> t = [1 3 5 7 9];
>> net = newlind(p,t);
>> Y = sim(net,p)
Y = 1.0000 3.0000 5.0000 7.0000 9.0000
>> gensim(net,-1)

```

Вывод: изучил средства и методы MATLAB, пакеты Neural Network Toolbox и Simulink для моделирования и исследования нейронных сетей, получил умения и навыки в освоении базовых приемов моделирования и исследования нейронных сетей в среде MATLAB, в применении нейронных сетей для аппроксимации функций, в анализе полученных результатов.