Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

Кафедра интеллектуально-информационных технологий

Лабораторная работа №3

“Моделирование нейронных сетей в MATLAB”

Выполнил:

студент 3 курса

группы ИИ-21

Литвинюк Т. В.

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)

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

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

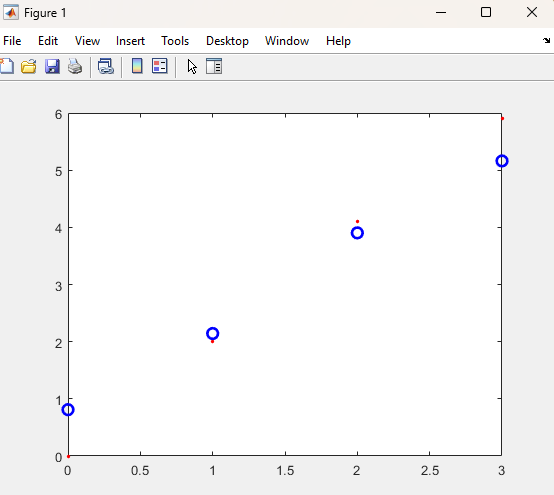
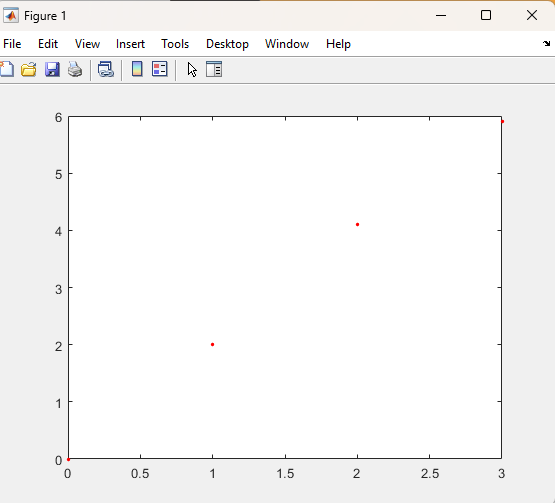
>> net = newgrnn(P,T,0.1);

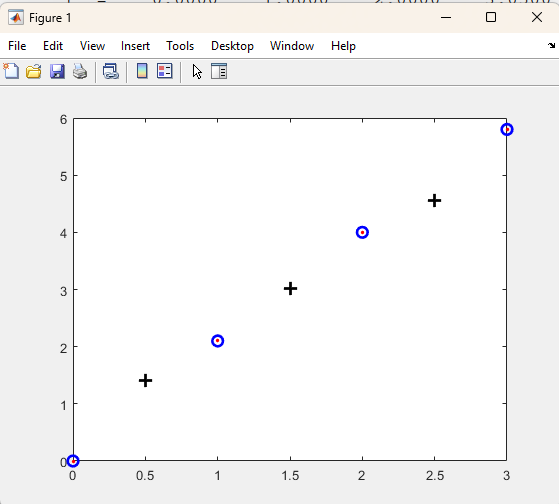
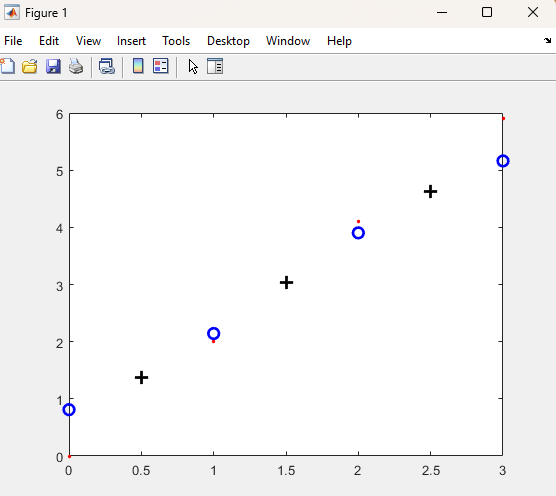
>> Y = sim(net, 0:0.5:3)

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 =

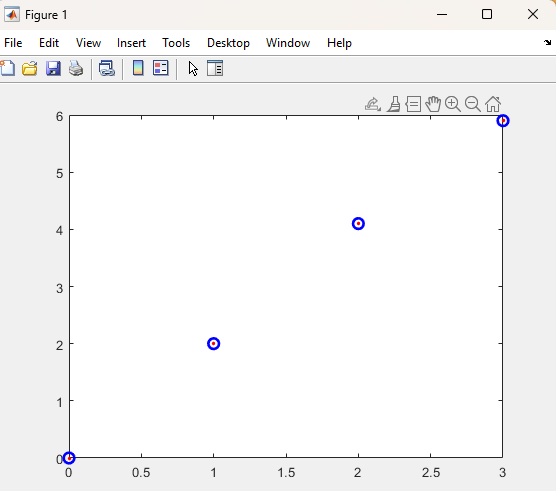
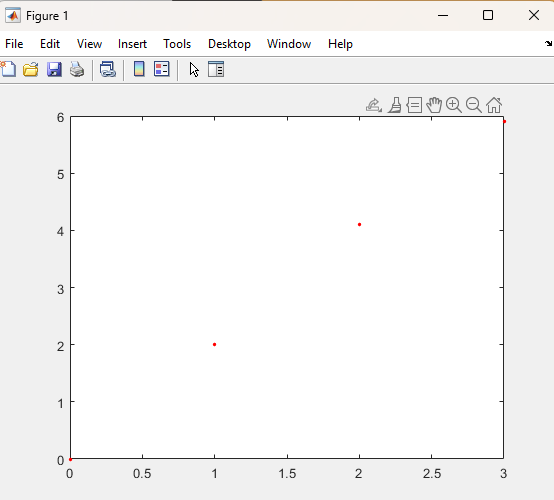
1.0346 2.8817 5.5053

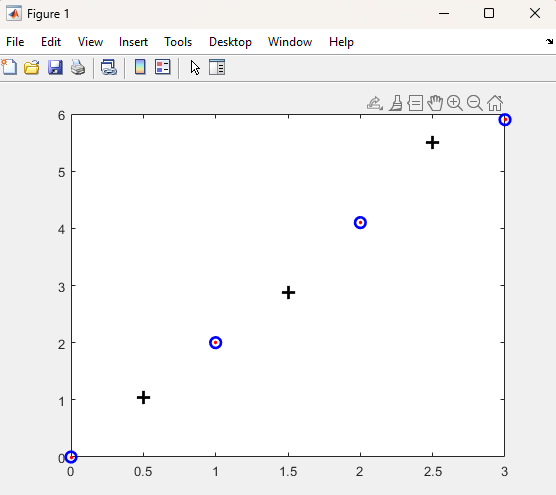
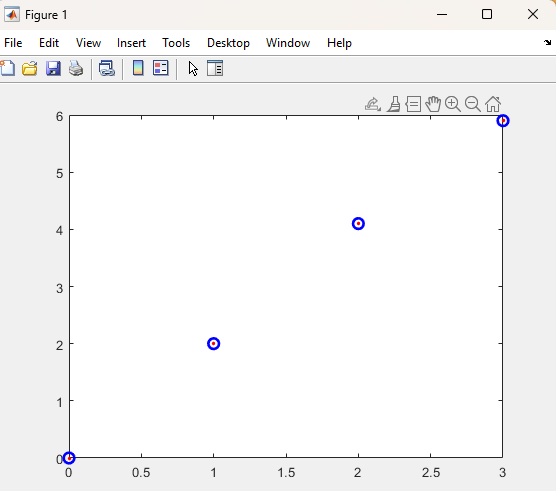
>> plot(P1,Y,'+k','MarkerSize',10, 'LineWidth',2)

>> 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');

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Задание 2.** 1. Создать НС в Fuzzy Logic Toolbox с помощью графического интерфейса пользователя (GUI, или ГИП) для выполнения операции у = 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.

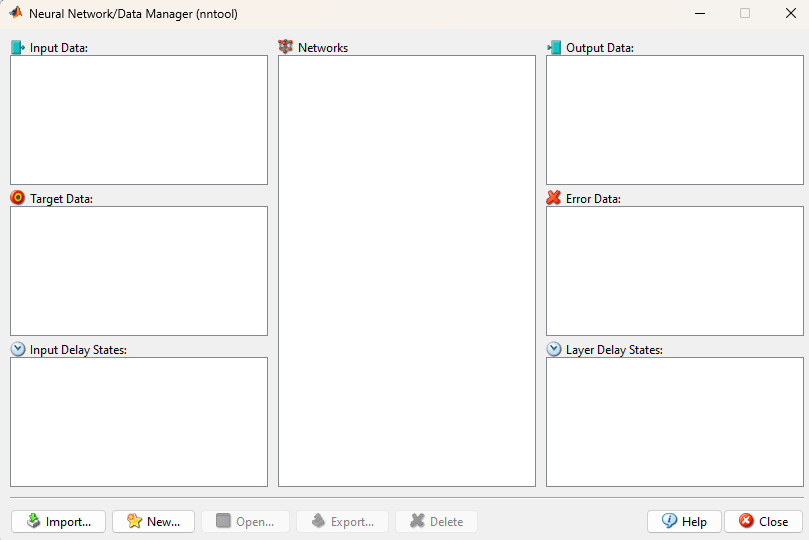
**Пример 4**

>> x = [-1 -0.8 -0.5 -0.2 0 0.1 0.3 0.6 0.9 1];

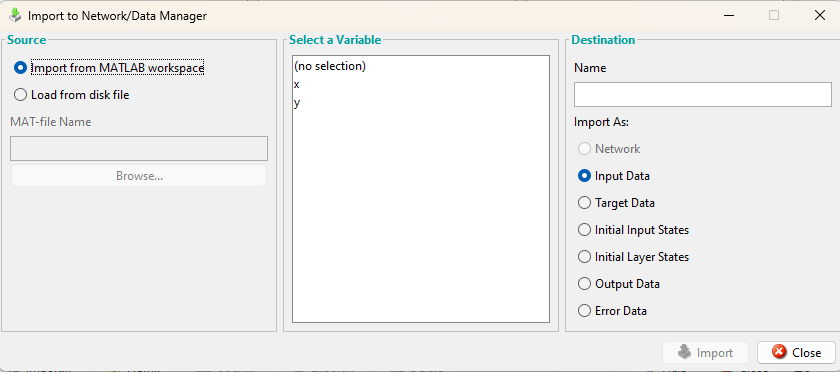
>> y = [1 0.64 0.25 0.04 0 0.01 0.09 0.36 0.81 1];

>> nntool

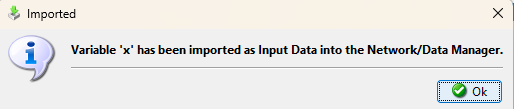
После выполнения команды nntool открываются окна создания нейронной сети **Network/Data Manager**



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



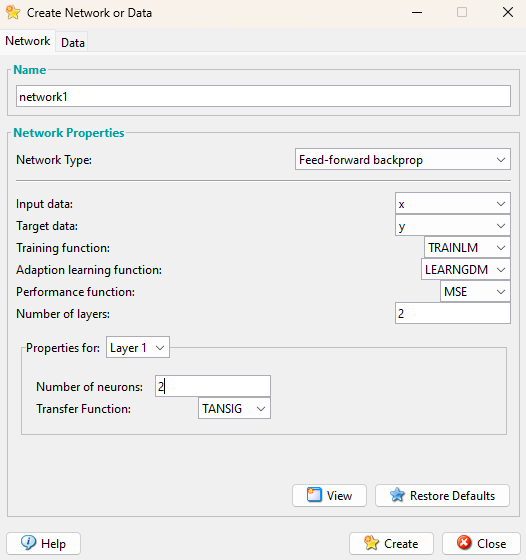
В результате появится окно **Imported**. Аналогичную операцию проделаем для вектора целей у и целевых данных **Target Data**.



Закроем окно **Import to Network/Data Manager** кнопкой **Close**.

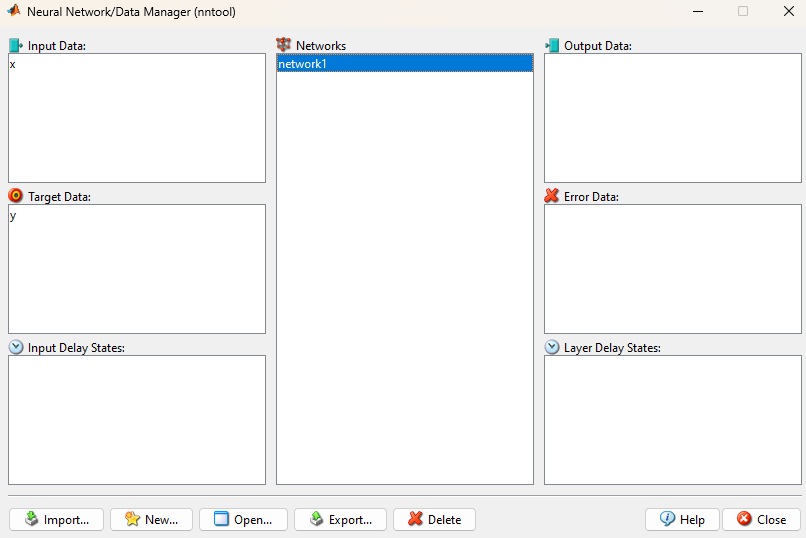
Создадим новую нейронную сеть. Для этого в окне **Network/Data Manager** нажмем кнопку **New**.

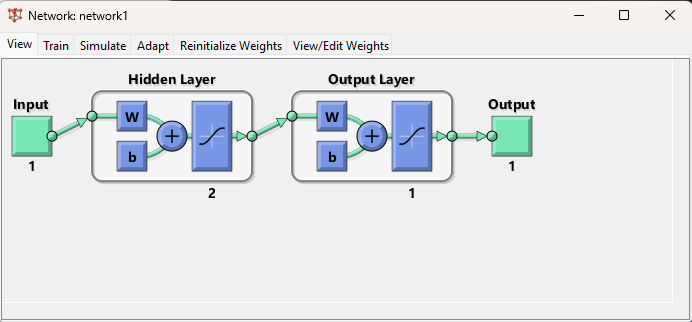
В открывшемся окне **Create Network or Data** выберем нейронную сеть типа **feed-forward backprop** с прямой передачей сигнала и с обратным распространением ошибки. При создании сети сохраним ей имя, даваемое по умолчанию **(network1)**. В качестве входных данных **Input Data** выберем **х**, а в качестве целевых данных **Target Data** — **y**. Количество нейронов (**Number of neurons**) первого слоя (**Layer 1**) установим равным двум. Остальные установки при создании сети оставим по умолчанию.



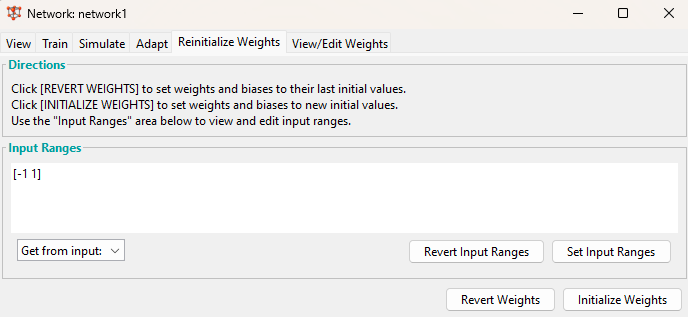
Создание сети завершим нажатием кнопки **Create**.

После этого в окне **Network/Data Manager**, в области **Network**, появится имя новой созданной сети — **network1** (рис. 4.9). Дважды щелкнем по этому имени левой кнопкой мыши, что приведет к открытию окна **Network: network1**.





Для ввода в панели **Network: network1** установленных диапазонов и инициализации весов можно воспользоваться вкладкой **Reinitialize Weights.** Если требуется вернуться к прежним диапазонам, то следует выбрать кнопки **Revert Input Ranges** (Вернуть диапазоны) и **Revert Weights** (Вернуть веса).



**Пример 5**

Для обучения созданной сети выбирается вкладка **Train** в панели **Network: network1** и открывается новая диалоговая панель.

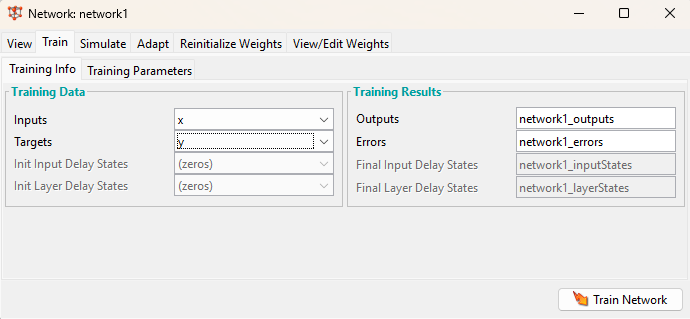
Панель имеет две вкладки:

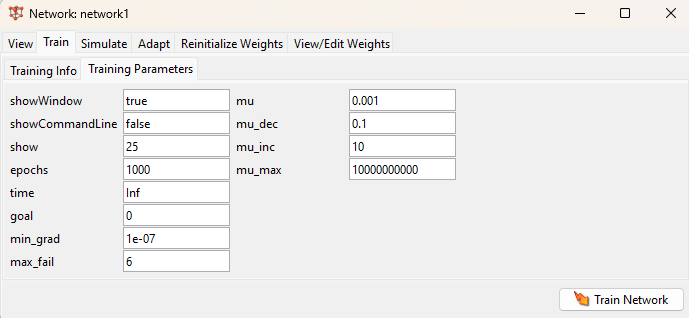
**Training Info** (Информация об обучающих последовательностях);

**Training Parameters** (Параметры обучения).

Применяя эти вкладки, можно установить имена последовательностей входа и цели (на вкладке **Training Info** — в левой ее части необходимо указать **х** и **у**), 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.





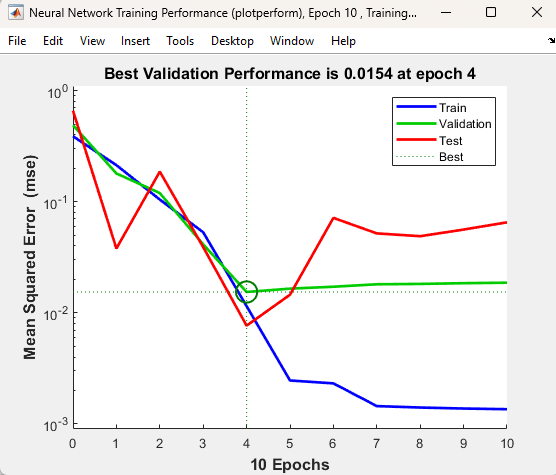
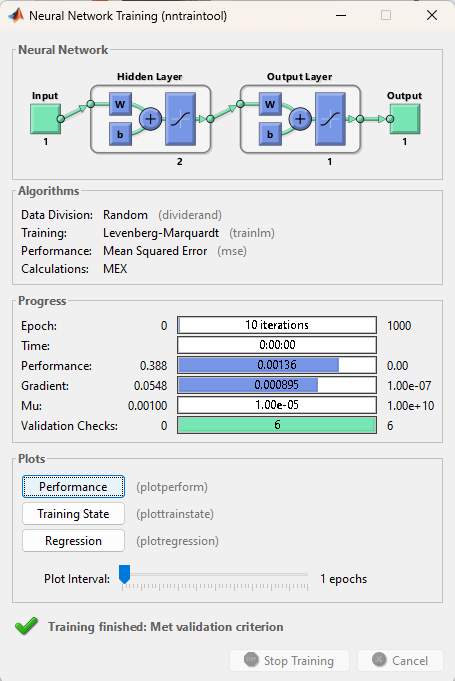
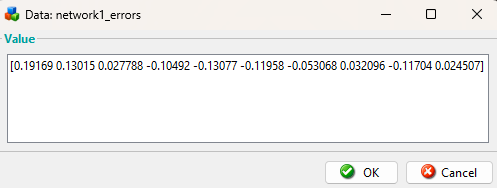
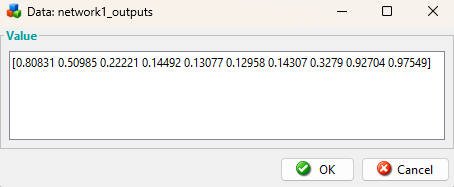
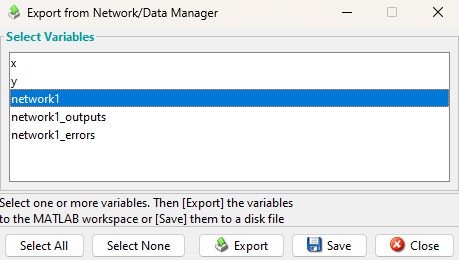


График либо выводится в виде отдельного окна самостоятельно, либо, в более поздней версии MATLAB, вызывается нажатием кнопки **Performance** в окне **Neural Network Training.** Видно, что к концу процесса обучения ошибка становится очень малой (вид данного рисунка при повторе вычислений может отличаться от приведенного). 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**

При необходимости можно экспортировать созданную нейронную сеть в рабочую область системы MATLAB, нажав кнопку **Export** (в открывшемся окне необходимо выбрать название НС, а затем нажать кнопку **Export**), и получить информацию о весах и смещениях непосредственно в рабочем окне системы, выполнив команду:



>> network1.IW{1,1},network1.b{1}

ans =

3.2099

1.9136

ans =

-2.7287

2.0765

и команду:

>> network1.IW{2,1},network1.b{2}

ans =

[]

ans =

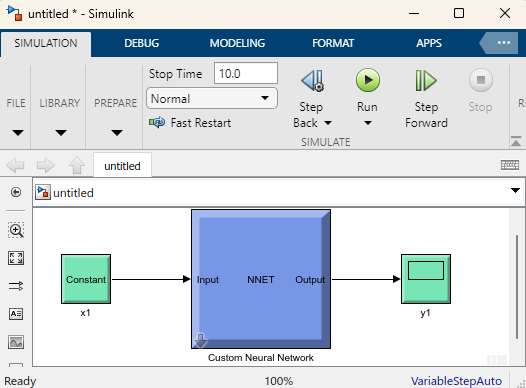
3.0384

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)

Эта схема является в полной мере функциональной схемой и может быть применена для моделирования нейронной сети.

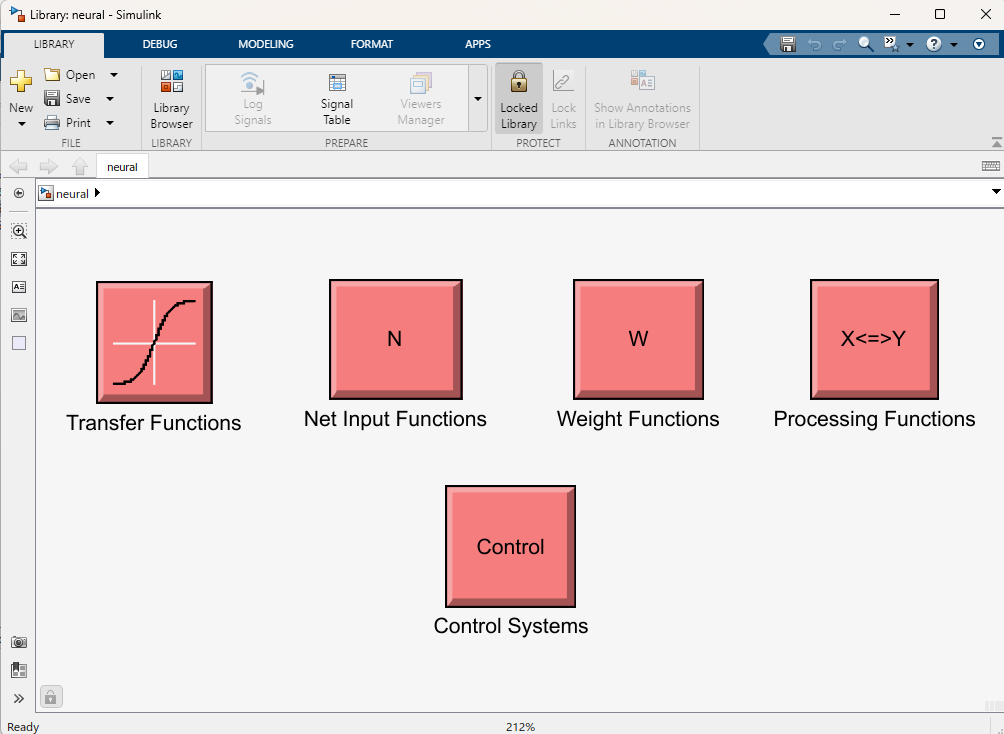


Двойной щелчок на блоке **Neural Network** раскрывает слои сети, а двойной щелчок на блоке слоя сети раскрывает его структуру.

При выполнении команды gensim кроме схемы НС открывается окно **Library:neural** c 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



**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.

**Пример 7.**

В качестве примера использования средств Simulink рассмотрим следующую задачу.

Пусть входной и целевой векторы имеют вид: р = [1 2 3 4 5] ; t=[1 3 5 7 9];

Создадим линейную НС и протестируем ее по данным обучающей выборки:

>> р = [1 2 3 4 5];

>> t = [1 3 5 7 9];

>> net = newlind(р,t);

>> Y = sim(net,p)

Y = 1.0000 3.0000 5.0000 7.0000 9.0000

Затем запустим Simulink командой:

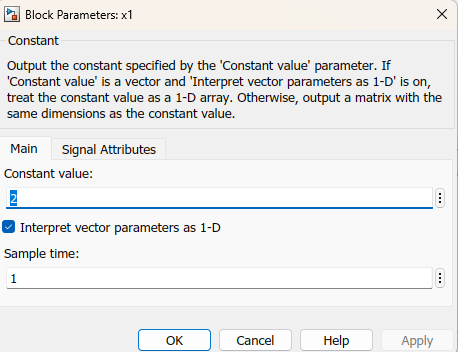
>> gensim(net,-1)

Это приведет к открытию окна с нейросетевой моделью

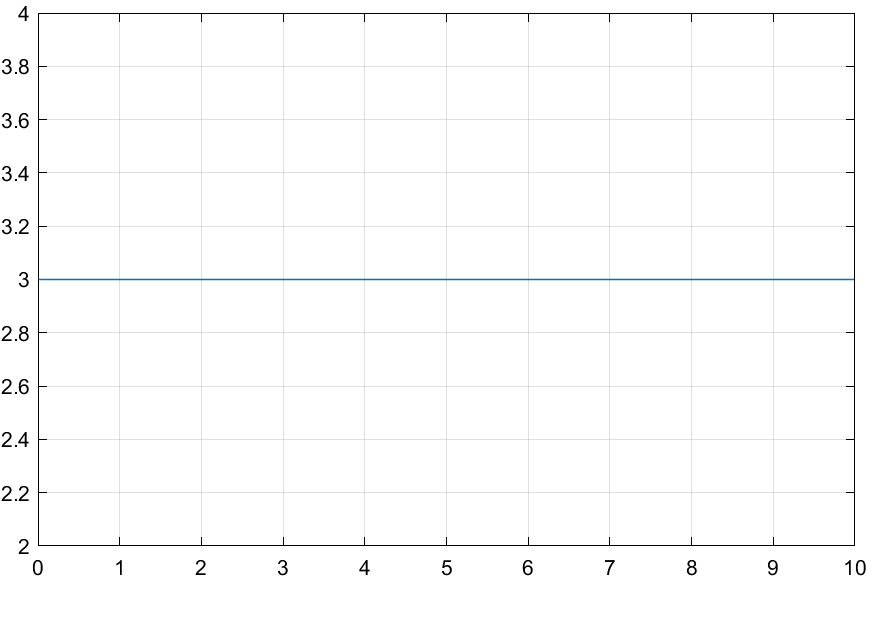
Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Для проведения тестирования модели щелкнем дважды на левом значке (с надписью **Input 1** — Вход 1), что приведет к открытию диалогового окна параметров блока. В данном случае блок **Input 1** является стандартным блоком задания константы **(Constant).** Изменим значение по умолчанию на 2 и нажмем кнопку **ОК.**



Расчет нового значения сетью производится практически мгновенно. Для его вывода необходимо дважды щелкнуть на правом значке (на блоке **у(1))**. Результат вычислений равен **3**, как и требуется, и выводится в виде графика.



**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.