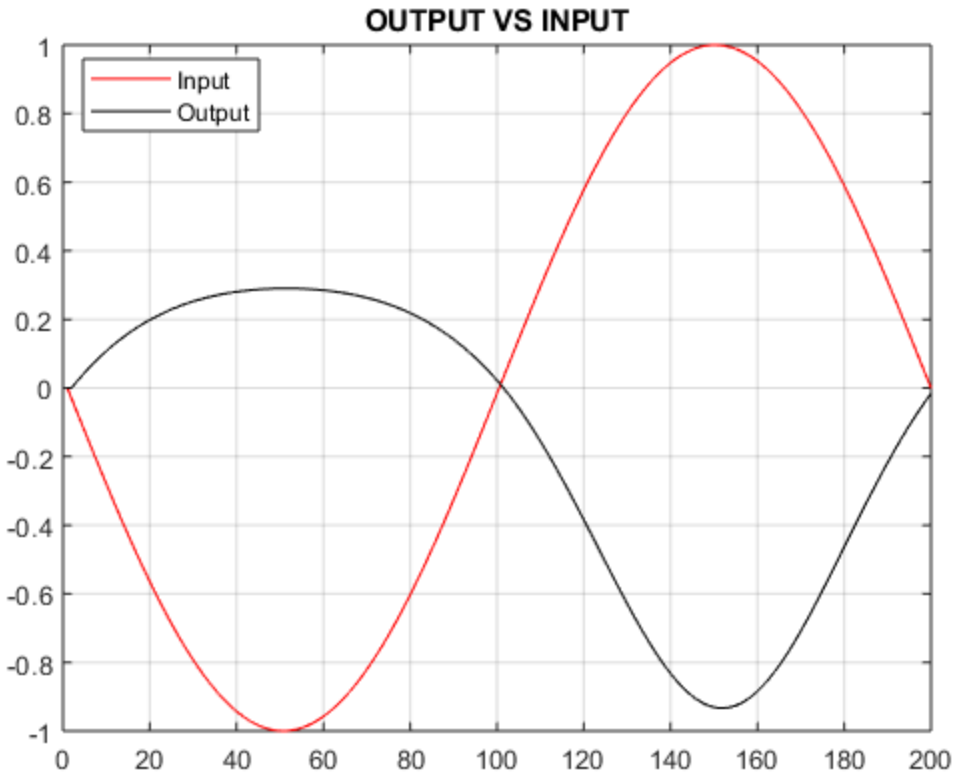
ANFIS

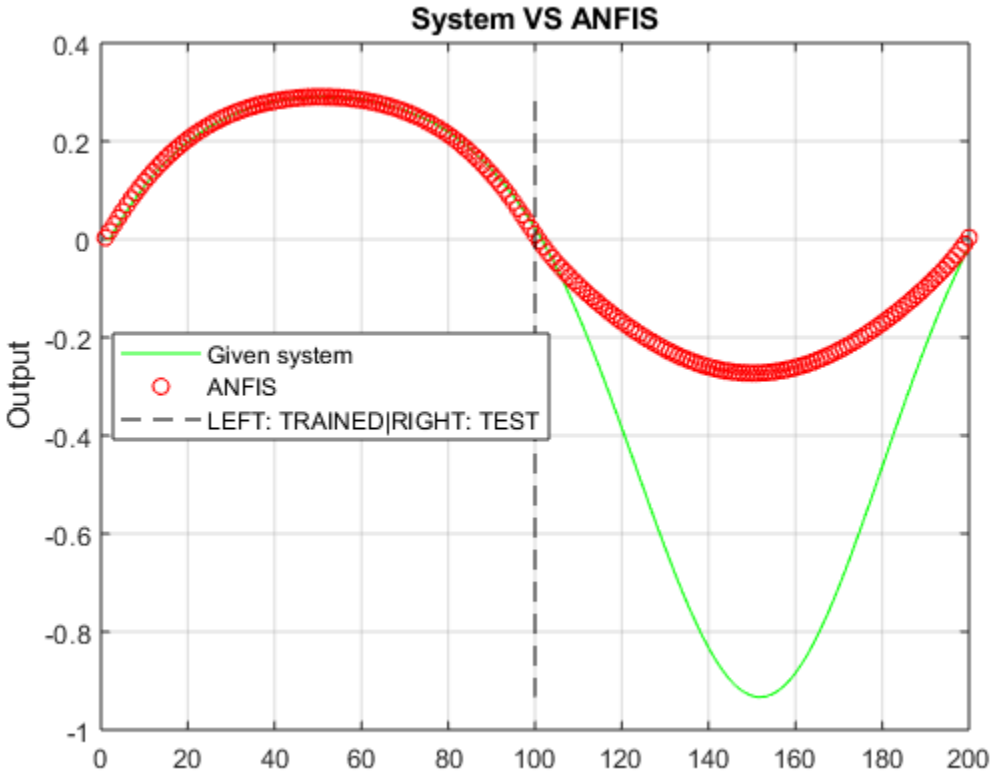
```
%We need to Make sure to include anfX.fis in directory
clear all; close all; clc; warning off;
%u=rand(200,1);
r=linspace(-pi,pi,200);
u=sin(r);
u=u';
%u=ones(200,1);
y=zeros(200,1);
% Assuming initial condition 'zero'
for k=2:length(u)-1
    y(k+1)=((y(k)*u(k))/(1+abs(y(k-1))^0.3))-(1-exp(-u(k)))/((1+exp(-
u(k))));
end

train_data=[u(1:100),y(1:100)];
test_data=[u(101:200),y(101:200)];

figure(1)
plot(u,'r');
hold on;
plot(y,'k');
title('OUTPUT VS INPUT');
legend('Input', 'Output','location','best');
grid on;
```




```
f=readfis('anfX.fis')
yf=evalfis(u, f);
figure(2)
plot(y,'g-');
hold on;
plot(yf,'ro');
hold on;
plot([100,100],[min(y),max(y)],'k--');
legend('Given system','ANFIS','LEFT: TRAINED|RIGHT:
 TEST','location','best');
grid on;
ylabel('Output');
title('System VS ANFIS');
```

```
f =

  struct with fields:

           name: 'anfX'
           type: 'sugeno'
      andMethod: 'prod'
       orMethod: 'probor'
    defuzzMethod: 'wtaver'
      impMethod: 'prod'
      aggMethod: 'sum'
```

```
        input: [1×1 struct]
       output: [1×1 struct]
         rule: [1×3 struct]
```



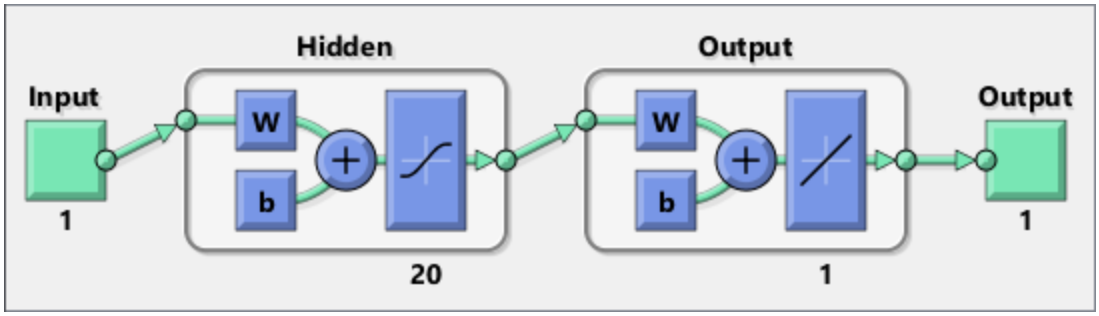
ARTIFICIAL NEURAL NETWORK

```
u1=u(1:100);
y1=y(1:100);

net = fitnet(20)
net = train(net,u1',y1');
view(net)
yn = net(u');
```

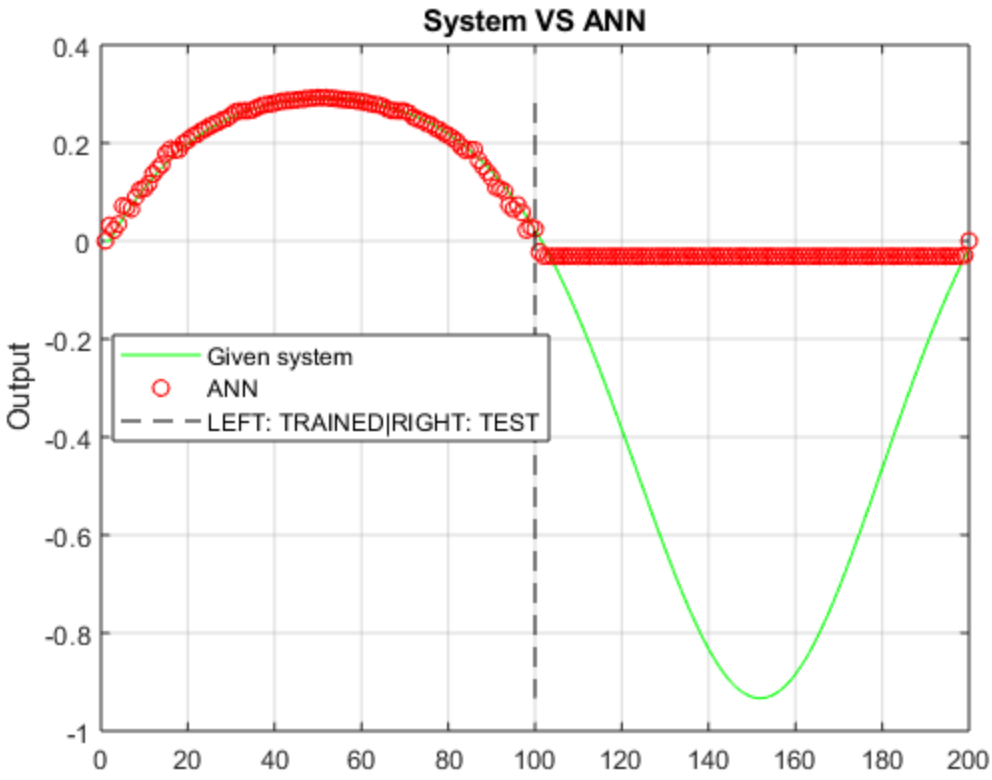
```
net =

    Neural Network

              name: 'Function Fitting Neural Network'
          userdata: (your custom info)

    dimensions:

         numInputs: 1
         numLayers: 2
```

```
        numOutputs: 1
    numInputDelays: 0
    numLayerDelays: 0
numFeedbackDelays: 0
numWeightElements: 20
        sampleTime: 1

    connections:

       biasConnect: [1; 1]
      inputConnect: [1; 0]
      layerConnect: [0 0; 1 0]
     outputConnect: [0 1]

    subobjects:

             input: Equivalent to inputs{1}
            output: Equivalent to outputs{2}

            inputs: {1x1 cell array of 1 input}
            layers: {2x1 cell array of 2 layers}
           outputs: {1x2 cell array of 1 output}
            biases: {2x1 cell array of 2 biases}
      inputWeights: {2x1 cell array of 1 weight}
      layerWeights: {2x2 cell array of 1 weight}

    functions:

          adaptFcn: 'adaptwb'
        adaptParam: (none)
          derivFcn: 'defaultderiv'
         divideFcn: 'dividerand'
       divideParam: .trainRatio, .valRatio, .testRatio
        divideMode: 'sample'
           initFcn: 'initlay'
        performFcn: 'mse'
      performParam: .regularization, .normalization
          plotFcns: {'plotperform', plottrainstate, ploterrhist,
                     plotregression, plotfit}
        plotParams: {1x5 cell array of 5 params}
          trainFcn: 'trainlm'
        trainParam: .showWindow, .showCommandLine, .show, .epochs,
                    .time, .goal, .min_grad, .max_fail, .mu, .mu_dec,
                    .mu_inc, .mu_max

    weight and bias values:

                IW: {2x1 cell} containing 1 input weight matrix
                LW: {2x2 cell} containing 1 layer weight matrix
                 b: {2x1 cell} containing 2 bias vectors

    methods:

             adapt: Learn while in continuous use
```

```
    configure: Configure inputs & outputs
       gensim: Generate Simulink model
         init: Initialize weights & biases
      perform: Calculate performance
          sim: Evaluate network outputs given inputs
        train: Train network with examples
         view: View diagram
  unconfigure: Unconfigure inputs & outputs
```




```
figure(3)
plot(y,'g-');
hold on;
plot(yn,'ro');
hold on;
plot([100,100],[min(y),max(y)],'k--');
legend('Given system','ANN','LEFT: TRAINED|RIGHT:
 TEST','location','best');
grid on;
ylabel('Output');
title('System VS ANN');
```




```
%I tested the system with ANFIS and ANN. The error was less in ANFIS
%system. Both system works well with trained input but when I used
%tesed data, It couldn't identify the system properly.
```

Published with MATLAB® R2018a