```
% Aditya Agre
% SYCOA06
% Input dataset
input_dataset = 7:0.2:50
```

```
input_dataset = 1×216
    7.0000    7.2000    7.4000    7.6000    7.8000    8.0000    8.2000    8.4000 ⋯
```

```
% Output dataset
output_dataset = x.^2
```

```
output_dataset = 1×216
10^3 ×
    0.0490    0.0518    0.0548    0.0578    0.0608    0.0640    0.0672    0.0706 ⋯
```

```
%Declaring network
network1 = newff(minmax(input_dataset), [20,1], {'tansig', 'poslin'})
```

```
Warning: NEWFF used in an obsolete way.
          See help for NEWFF to update calls to the new argument list.


network1 =

    Neural Network

              name: 'Custom Neural Network'
          userdata: (your custom info)

    dimensions:

         numInputs: 1
         numLayers: 2
        numOutputs: 1
    numInputDelays: 0
    numLayerDelays: 0
 numFeedbackDelays: 0
 numWeightElements: 61
        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: (none)
       divideParam: (none)
        divideMode: 'sample'
           initFcn: 'initlay'
        performFcn: 'mse'
      performParam: .regularization, .normalization
          plotFcns: {'plotperform', 'plottrainstate',
                    'plotregression'}
        plotParams: {1x3 cell array of 3 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
```

```
network1.trainParam.epochs = 1000;
network1.trainParam.goal = 1e-5;
network1.trainParam.lr=0.005;
network1=train(network1,input_dataset,output_dataset);
```