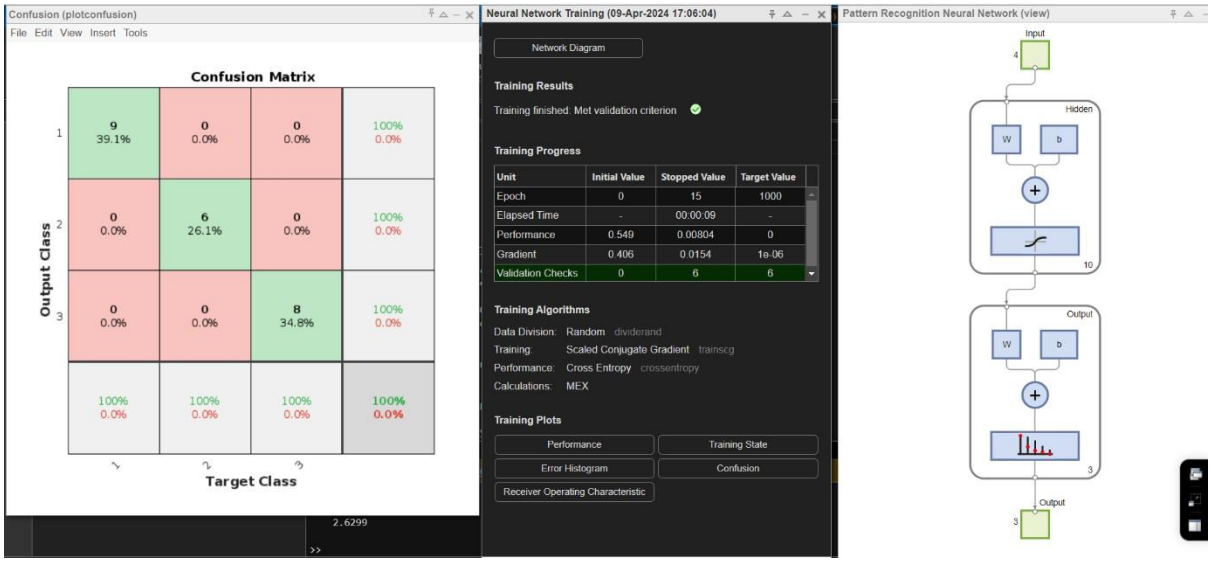
Practical no. 08

Title: Implement a program in MATLAB for classification using supervised learning algorithm.

Code:

```
% Classification
close all;
clear all;
clc;

% Load dataset
% [x, t] = simplefit_dataset;
% [x,t] = cancer_dataset;
[x,t] = iris_dataset;

net = patternnet(10); % net = feedforwardnet(10);

% Train
[net, tr] = train(net, x, t);
view(net)

% Estimate the targets
y = net(x);
classes = vec2ind(y);
perf = perform(net, y, t);
perf_test = perform(net, t(tr.testInd), net(x(:,tr.testInd)));

plotconfusion(t(:,tr.testInd), net(x(:,tr.testInd)));
```

Output:


Confusion (plotconfusion)
File Edit View Insert Tools
Confusion Matrix
Output Class
Target Class
9 39.1%
0 0.0%
0 0.0%
100% 0.0%
0 0.0%
6 26.1%
0 0.0%
100% 0.0%
0 0.0%
0 0.0%
8 34.8%
100% 0.0%
100% 0.0%
100% 0.0%
100% 0.0%
100% 0.0%
2.6299
>>
Neural Network Training (09-Apr-2024 17:06:04)
Network Diagram
Training Results
Training finished: Met validation criterion
Training Progress
Unit | Initial Value | Stopped Value | Target Value
Epoch | 0 | 15 | 1000
Elapsed Time | - | 00:00:09 | -
Performance | 0.549 | 0.00804 | 0
Gradient | 0.406 | 0.0154 | 1e-06
Validation Checks | 0 | 6 | 6
Training Algorithms
Data Division: Random dividerand
Training: Scaled Conjugate Gradient trainscg
Performance: Cross Entropy crossentropy
Calculations: MEX
Training Plots
Performance
Training State
Error Histogram
Confusion
Receiver Operating Characteristic
Pattern Recognition Neural Network (view)
Input
4
Hidden
W
b
10
Output
W
b
3
Output
3

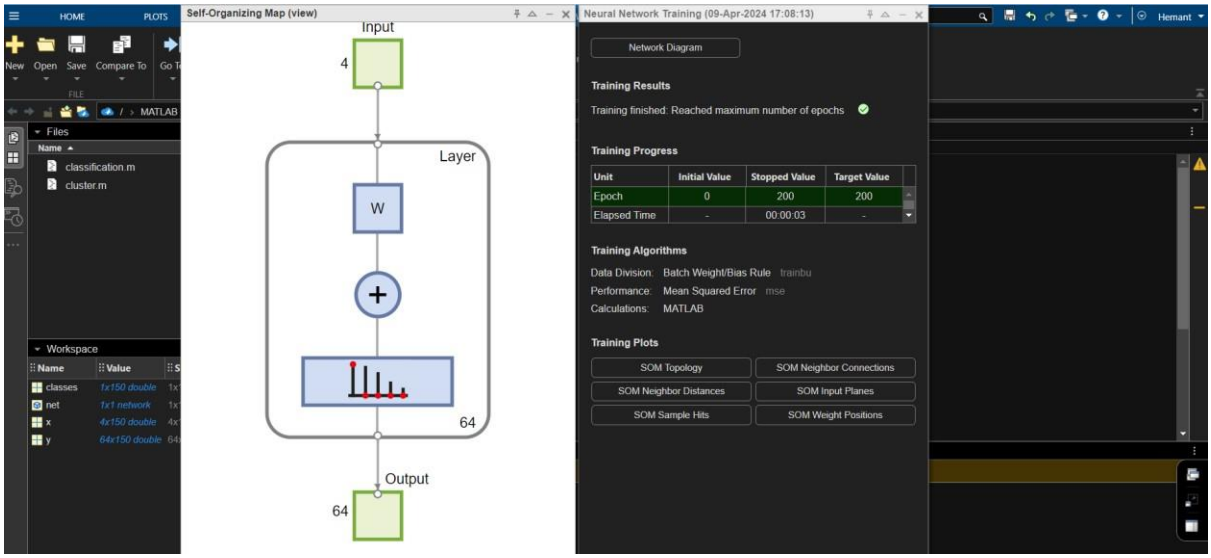
Practical No. 09

Title: Clustering of given data.

Code:

```
% Data clustering problem
clc;
close all;
clear all;

x = iris_dataset;

net = selforgmap([8 8]);
net = train(net, x);
```

Output:



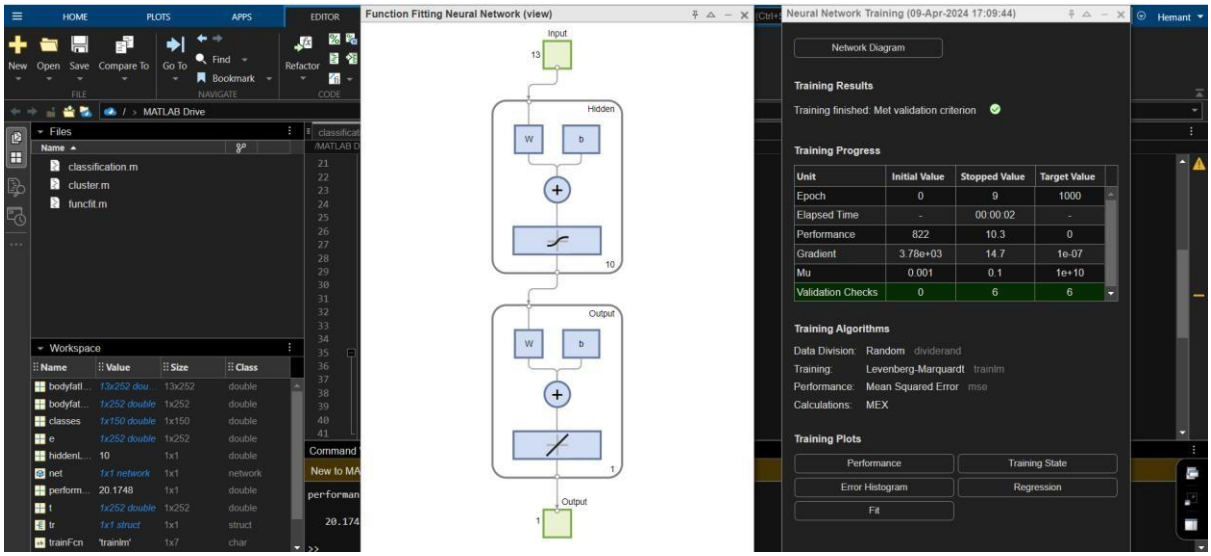
Practical No. 10

Title: Construct and train a function fitting network.

Code:

```
%  Estimation of body fat using function fitting
load bodyfat_dataset;

x = bodyfatInputs;
t = bodyfatTargets;

% Choose a Training Function
% For a list of all training functions
% 'trainbr' takes longer but may be better for
challenging problems.
% 'trainscg' uses less memory. Suitable in low memory
situations. trainFcn = 'trainlm';  % Levenberg-
Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network y =
net(x); e =
gsubtract(t,y);
performance =
perform(net,t,y)

% View the Network
view(net)

% Plots
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```

Output:


HOME
PLOTS
APPS
EDITOR
New
Open
Save
Compare To
Go To
Find
Bookmark
Refactor
FILE
NAVIGATE
CODE
MATLAB Drive
Files
Name
classification.m
cluster.m
funclit.m
Workspace
Name
Value
Size
Class
bodyfatI...
13x252 dou...
13x252
double
bodyfat...
1x252 double
1x252
double
classes
1x150 double
1x150
double
e
1x252 double
1x252
double
hiddenL...
10
1x1
double
net
1x1 network
1x1
network
perform...
20.1748
1x1
double
t
1x252 double
1x252
double
tr
1x1 struct
1x1
struct
trainFcn
'trainlm'
1x7
char
Command
New to MA
performan
20.174
Function Fitting Neural Network (view)
Input
13
Hidden
W
b
10
Output
W
b
1
Output
1
Neural Network Training (09-Apr-2024 17:09:44)
Hemant
Network Diagram
Training Results
Training finished: Met validation criterion
Training Progress
Unit
Initial Value
Stopped Value
Target Value
Epoch
0
9
1000
Elapsed Time
-
00:00:02
-
Performance
822
10.3
0
Gradient
3.78e+03
14.7
1e-07
Mu
0.001
0.1
1e+10
Validation Checks
0
6
6
Training Algorithms
Data Division: Random dividerand
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX
Training Plots
Performance
Training State
Error Histogram
Regression
Fit