

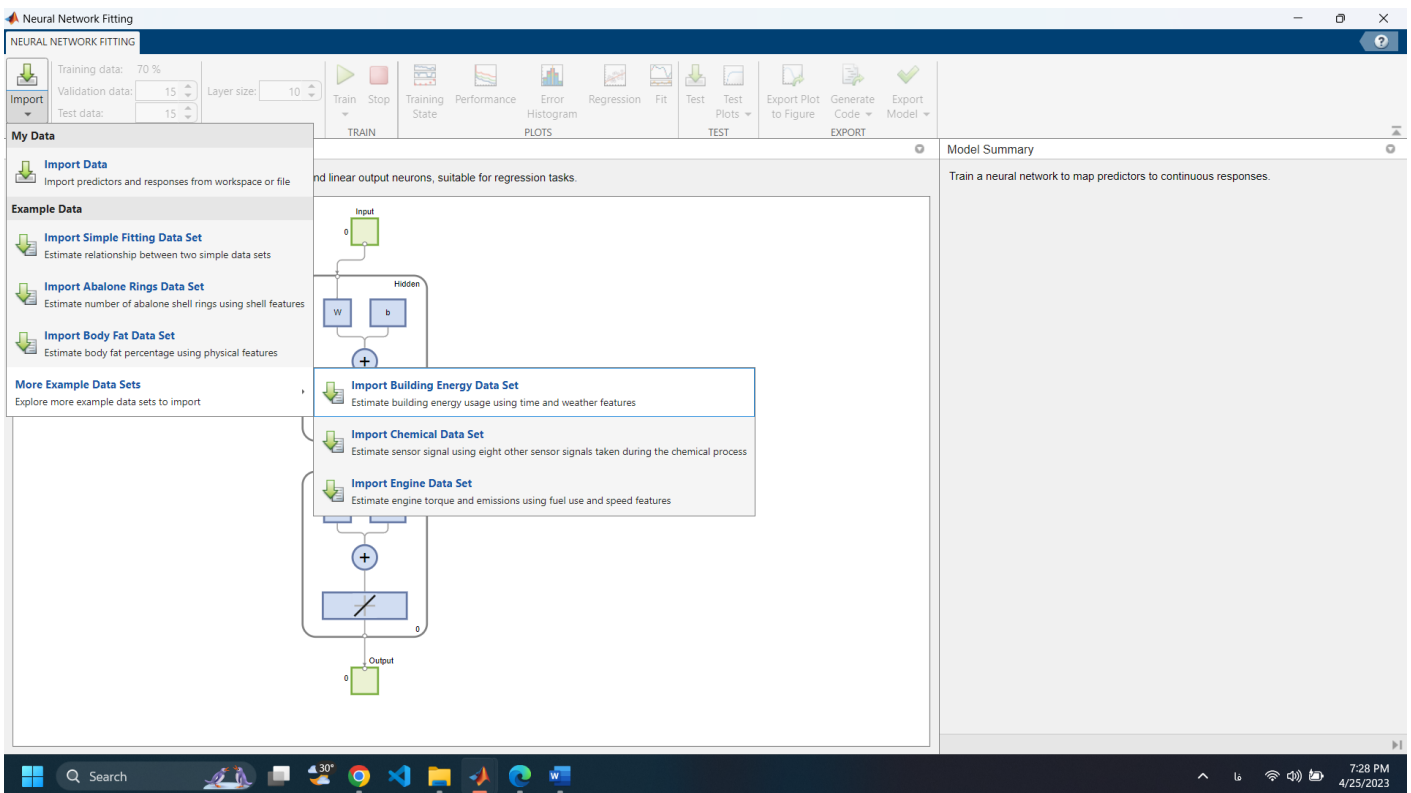
Date: 1402-01-29

نام و نام خانوادگی دانشجو: رحمت اله انصاری

شماره دانشجویی: ۹۹۱۲۳۷۷۳۳۱

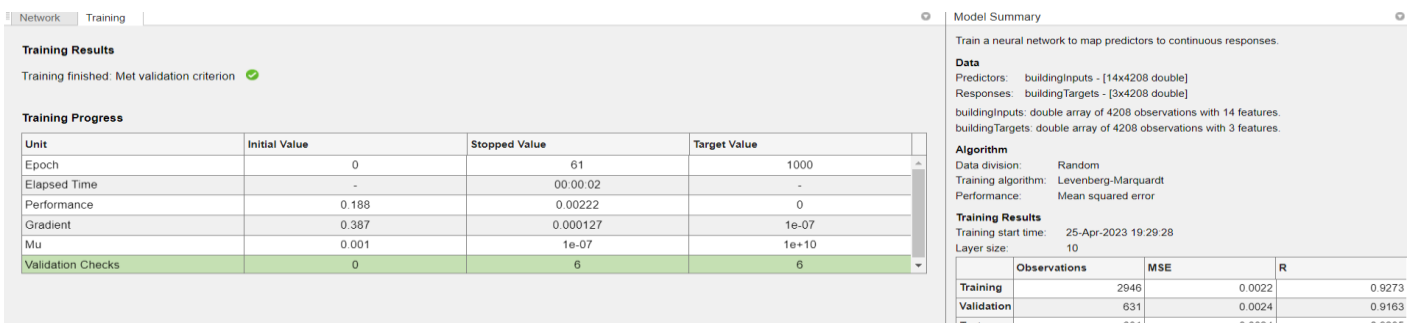
تمرین کار با شبکه عصبی در متلب

ابتدا یک شبکه عصبی میسازیم. Neural network fitting



سپس نمونه را ایمپورت میکنیم.

نتیجه به صورت زیر است:



Training Results

Training finished: Met validation criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	61	1000
Elapsed Time	-	00:00:02	-
Performance	0.188	0.00222	0
Gradient	0.387	0.000127	1e-07
Mu	0.001	1e-07	1e+10
Validation Checks	0	6	6

Model Summary

Train a neural network to map predictors to continuous responses.

Data

Predictors: buildingInputs - [14x4208 double]
 Responses: buildingTargets - [3x4208 double]
 buildingInputs: double array of 4208 observations with 14 features.
 buildingTargets: double array of 4208 observations with 3 features.

Algorithm

Data division: Random
 Training algorithm: Levenberg-Marquardt
 Performance: Mean squared error

Training Results

Training start time: 25-Apr-2023 19:29:28
 Layer size: 10

	Observations	MSE	R
Training	2946	0.0022	0.9273
Validation	631	0.0024	0.9163

با زدن گزینه زیر نتیجه را در ورک اسپیس وارد میکنیم.

Neural Network Fitting

Training data: 70 %
Validation data: 15 %
Test data: 15 %
Layer size: 10

Export

- Export to Workspace: Export structure array containing trained network and results to the workspace
- Export to Simulink: Export network as Simulink block
- Export Network Function for MATLAB Compiler: Export network as MATLAB function with matrix and cell array arguments
- Export Network Function for MATLAB Coder: Export network as MATLAB function with matrix-only arguments

Training Results

Training finished: Met validation criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	61	1000
Elapsed Time	-	00 00 02	-
Performance	0.188	0.00222	0
Gradient	0.387	0.000127	1e-07
Mu	0.001	1e-07	1e+10
Validation Checks	0	6	6

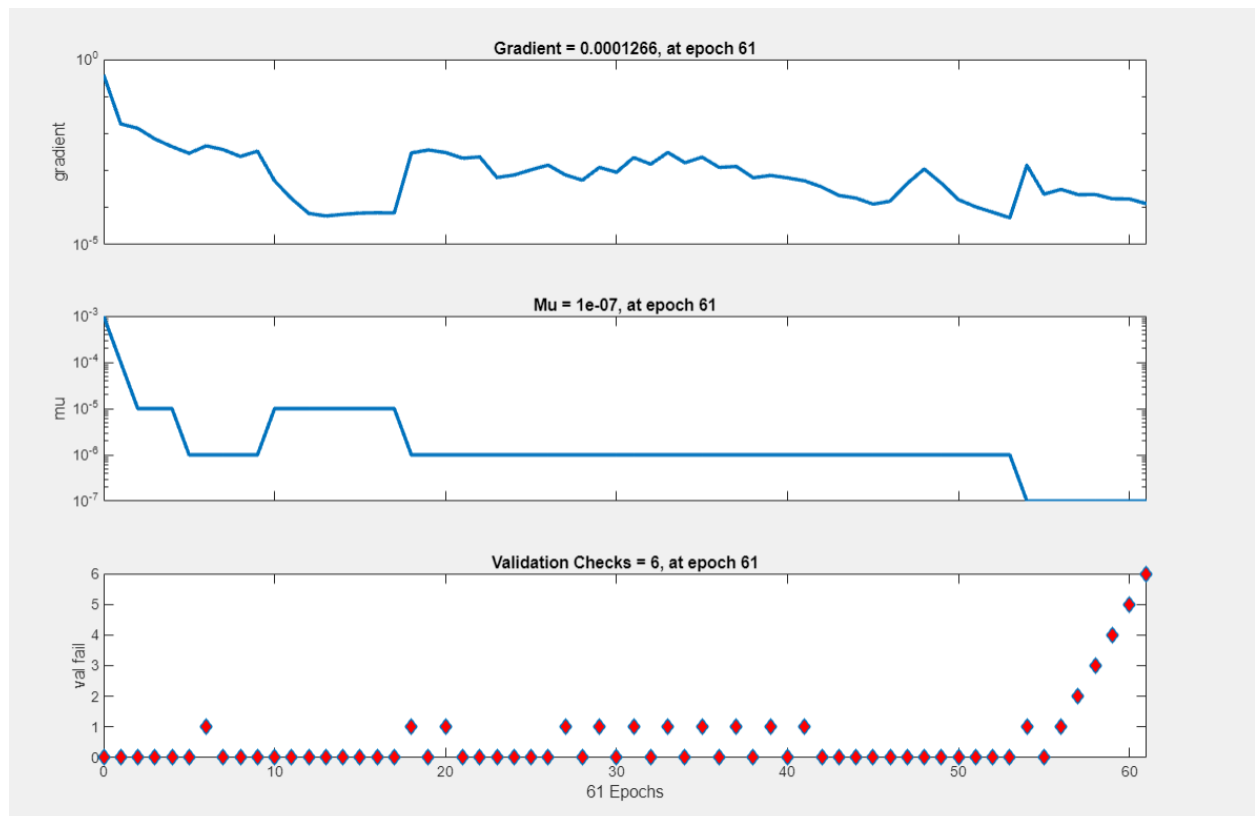
Training algorithm: Levenberg-Marquardt
Performance: Mean squared error

Training Results

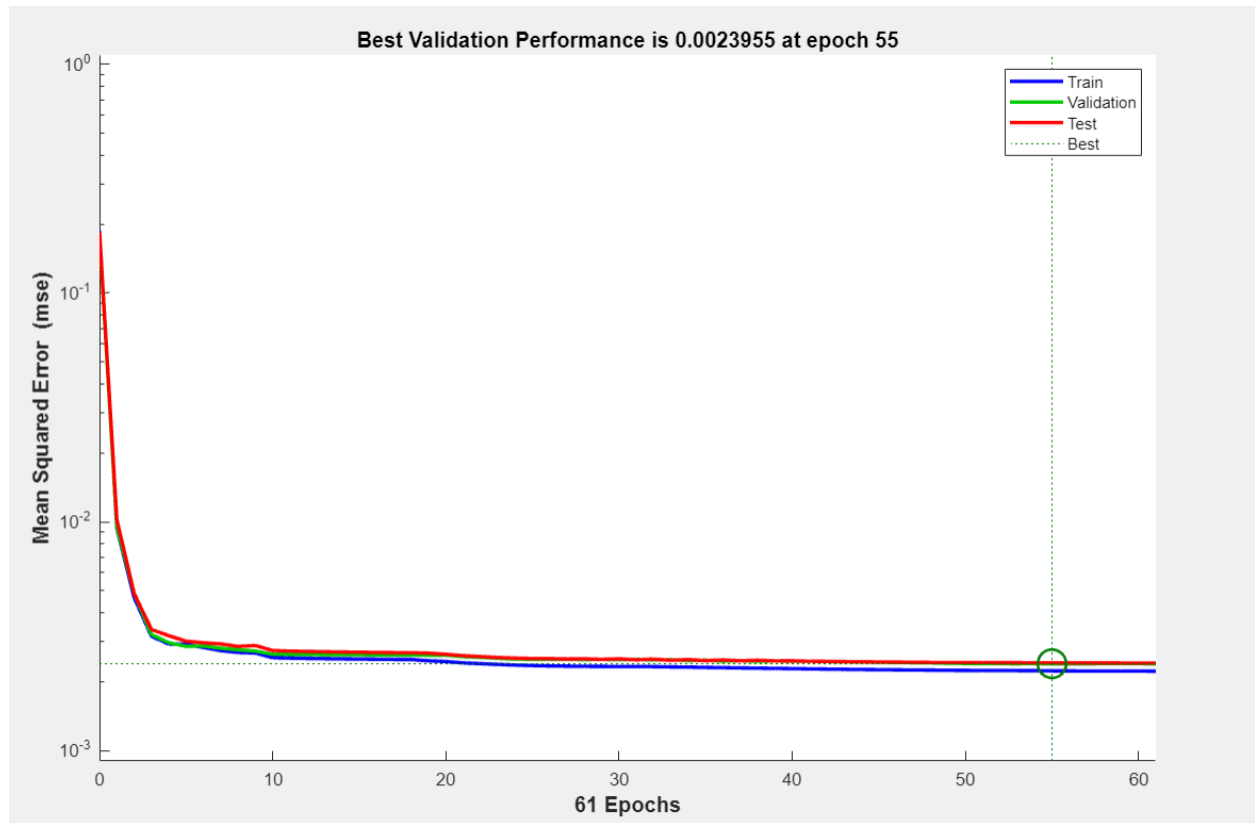
Training start time: 25-Apr-2023 19:29:28
Layer size: 10

	Observations	MSE	R
Training	2946	0.0022	0.9273
Validation	631	0.0024	0.9163
Test	631	0.0024	0.9235

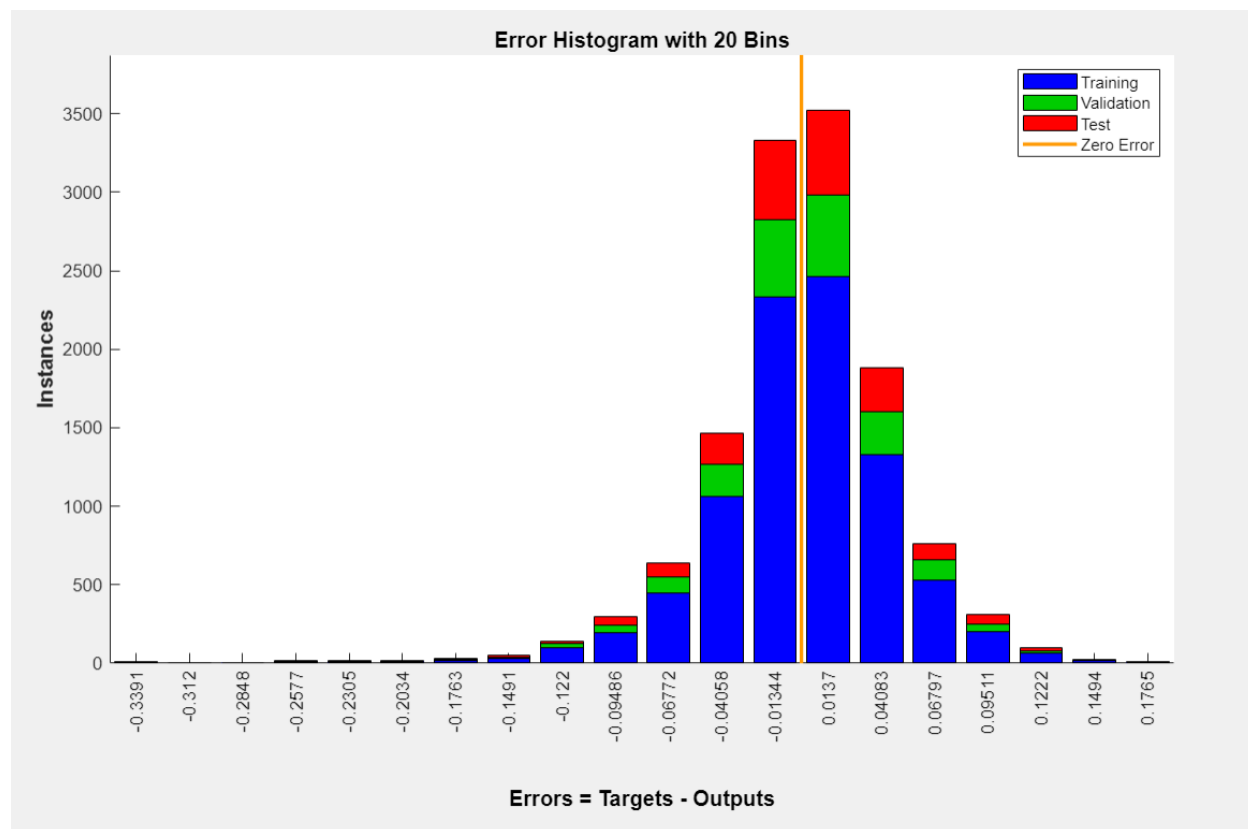
وضعیت آموزش به صورت زیر است:



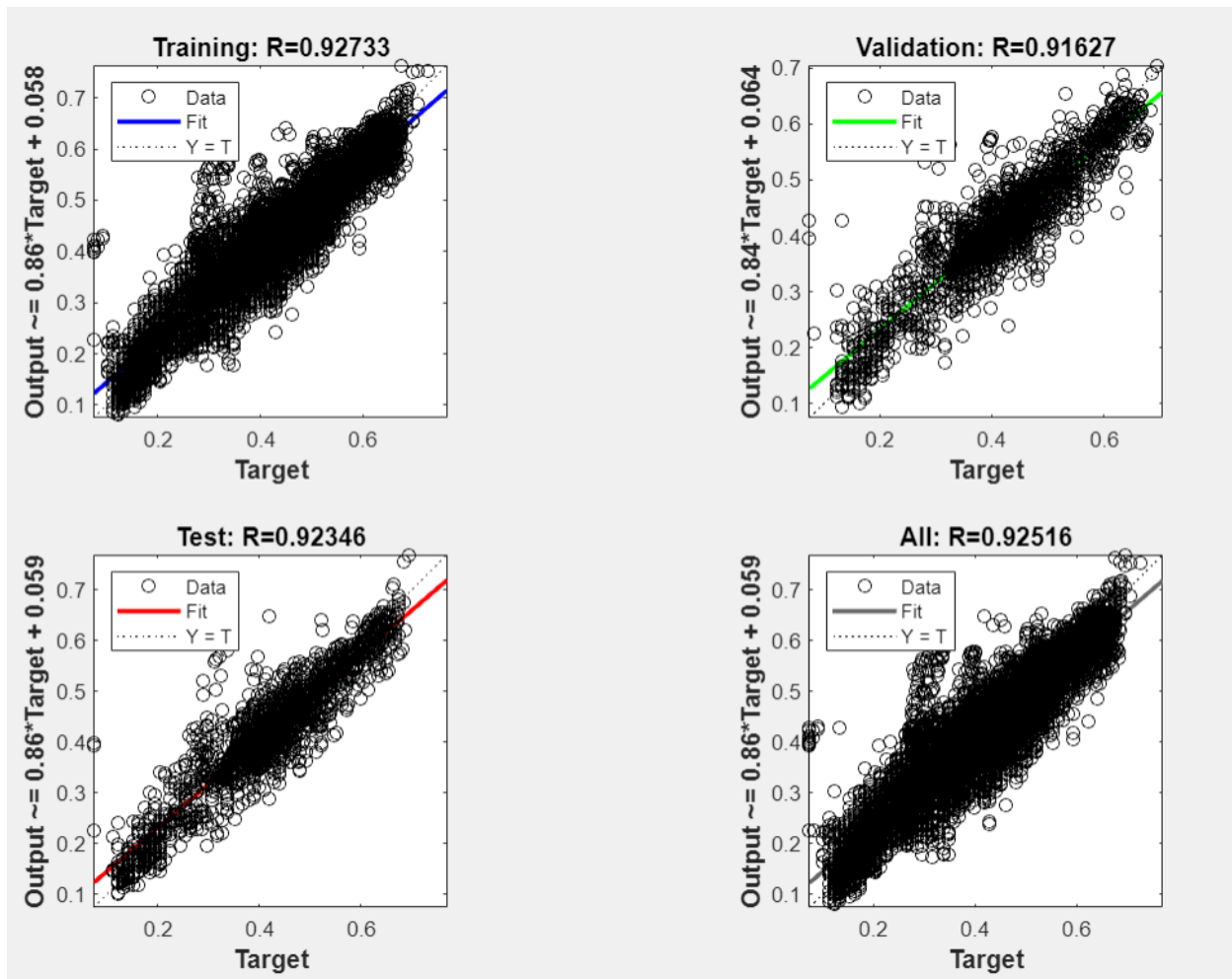
پرفورمنس به صورت زیر است:



هیستوگرام ارور هم به صورت زیر است:



نمودار رگرسیون آن هم به صورت زیر است:



اسکرپت ساخته شده هم به صورت زیر است:

```
function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Auto-generated by MATLAB, 25-Apr-2023 19:35:31.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 14xQ matrix, input #1
% and returns:
%   y = 3xQ matrix, output #1
% where Q is the number of samples.

%#ok<*RPMT0>
```

```
% ===== NEURAL NETWORK CONSTANTS =====
```

```
% Input 1
```

```
x1_step1.xoffset = [0;0;0;0;0;0;0;0;-1;-1;0.026;0;0.00877358;0];  
x1_step1.gain =  
[2;2;2;2;2;2;2;2;1;1;2.0639834881321;2.25225225225225;2.06688134536448;2.130898994322  
22];  
x1_step1.ymin = -1;
```

```
% Layer 1
```

```
b1 = [2.4870705358876583624;-1.8705052363040433061;-0.84976462985855527599;-  
0.31465903334728767549;0.17005715827828116193;-0.24068296915264014713;-  
0.25044295356029949717;-1.3084964623007431239;-  
1.2196557237584846334;1.7174269613099353293];  
IW1_1 = [-0.71463299315271933132 -1.1651199952137885063 -0.76876393207532334184 -  
0.53892151588759451553 -0.46043970838093600806 -0.41914730342237077609 -  
0.44719594147801061679 -1.0842204660817826856 -2.4311943133243185677 -  
2.5550631889457724455 -4.308144701545219668 1.049102573433542851  
2.3073812927807080264 1.0101075421685963285;1.1775616965175303985 -  
1.672749615609507412 -2.4201668675162983924 1.1784750693533032617  
1.2040110552283136425 1.1184126091706616535 1.1915294787630676954 -  
5.2999993144233821241 1.9153665984339200978 0.036397972964384101013  
0.22643735367681300241 -0.10990308326885808721 0.20890084556845164854  
0.0072192364721500139335;0.062162075576465081195 -0.37578658083338201124  
1.0408586387488849567 0.35627394952564217512 0.77473033805248525674  
0.82923643881482933526 -1.9360857387644603289 -0.41930106708541947658  
0.22800794437280993976 -0.26269613332343416667 -4.4513435050773617618 -  
0.75669256804238560488 -0.64410569536272677116 4.3126862801523007107;-  
0.892339884709226260539 -1.1600763551710413513 0.016087400080169157734  
1.6377080023160086597 0.36904020572372087106 0.33136761549358839662 -  
0.099279938372391907109 -0.30602977555447447022 -0.24107728138141654162 -  
0.049778806178215298517 -4.1037193502520183941 0.86348745046445818385 -  
0.25210531466099289633 1.3900961808951328713;-0.92047830366777061428 -  
0.78964974763558082316 -0.71062462689085459377 -0.39328544352440364396  
0.22532456417559978967 0.30727046190077428323 1.8521020152962623939 -  
0.23230564205926071408 -0.18236618346606109364 -0.02783512198419121203 -  
0.65809122921881091361 0.028596370473067386775 -0.1298749020044716318 -  
0.092087539076401533911;-0.029822287974859949117 -0.16446075764911691208 -  
0.18441840145617005575 -0.40033357018802456784 -0.15687072883745581398 -  
0.073845343267246912888 -0.1805555361469949327 -0.0077754641549531433586 -  
0.28389323281212008876 -0.34293491227138606092 -3.2852644873105827372 -  
0.025293439820239765892 -0.1550745436425108037 -0.14057524000447710333;-  
0.5739725799308019738 -0.4665045740167272359 -0.39267368823090292862 -  
0.42898736630516637325 0.25016100862483586775 0.29317446790418560942  
0.50881176113693638019 -0.1185907322316969853 -0.026935109863320341045  
0.0038862288158129434934 -0.92976366350711259212 -0.049101919764832177573 -  
0.039296917507289932936 0.14033254243234852265;-0.9586587020990846586  
0.57828645024490288318 0.86166687409214648419 1.7532810900729858261  
1.0262741169029596655 0.91607149389532771533 -0.70122664947122903545 -  
0.64228775057413178828 -0.55472679515723322119 -0.064093481114655695419  
1.2412168008834454369 -1.2191773574687987391 -0.26118490938373839505 -  
1.2525934457860503635;-0.58825147417396150828 -1.372001110028784332 -  
0.95699309179650127177 1.038116076196837323 0.49984660516915085937  
0.6777028528648626482 0.0064666952854191619982 -2.2966855609565444141
```

```

5.3961680505105515948 0.65894638255314241793 1.3279173932979744244
2.0027243418031788202 0.88644617256734814514
2.8303257279659312928;0.27222866440446613634 0.28015161683316441854
0.27707315203815519045 0.27664452004136302898 0.2556131419272650307
0.25360250334938760908 0.24097908586218247384 0.0063488176412316629088 -
0.0090778593022683845559 -0.018583636298028814032 -0.35989872536193556751 -
0.086968983625766188572 -0.015459936876456401023 -0.012794269936644488139];

% Layer 2
b2 = [-1.5713027532411611986;1.0476244603017785817;0.16960542794613633899];
LW2_1 = [0.16346300509423192238 0.3954724806898258227 0.26485777965764251141 -
0.43386836172512538434 1.7971810174115294689 -0.40521452386109646504 -
2.1151985547486402695 -0.41731106200186129929 0.1686815130761820658
3.7598780299035405861;-0.011260199648902029582 0.0041527956847107602364
0.021415942475212793039 -0.0097084843115047420037 0.11208440574411912549
0.12741735018037952942 -0.21388063600307519185 -0.02587790730168156722
0.053214577193162033997 -2.2285416810687062039;0.049758669606714760192 -
0.033289492112634502807 -0.1887455873565156983 0.33198961972895912886 -
1.2412496436866811145 0.40838085065445506316 1.519804611242738579
0.24693655125949273699 -0.040415303282299412657 -0.70187163163809040967];

% Output 1
y1_step1.ymin = -1;
y1_step1.gain = [4.6518443399847;3.33333333333333;3.10405827683171];
y1_step1.xoffset = [0.259145;0.075;0.0818182];

% ===== SIMULATION =====

% Dimensions
Q = size(x1,2); % samples

% Input 1
xp1 = mapminmax_apply(x1,x1_step1);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
y1 = mapminmax_reverse(a2,y1_step1);
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

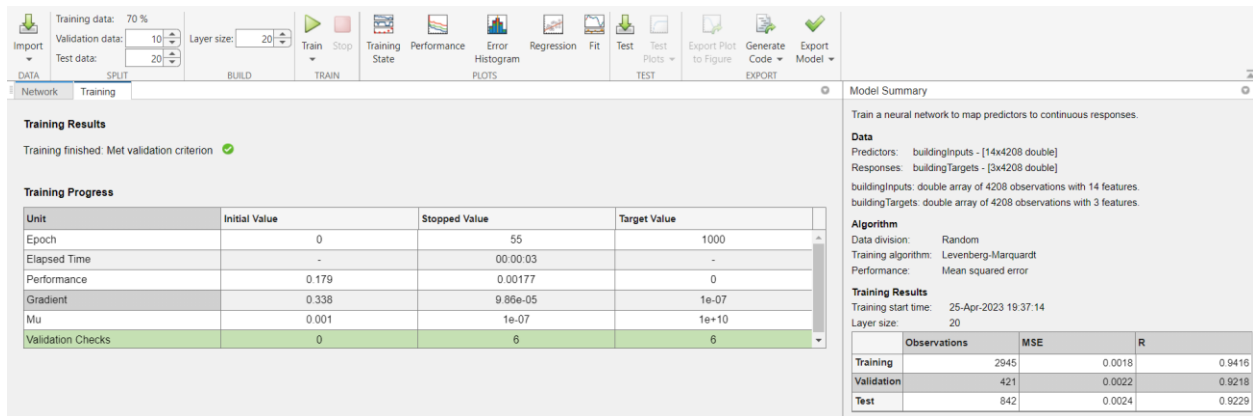
% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)

```

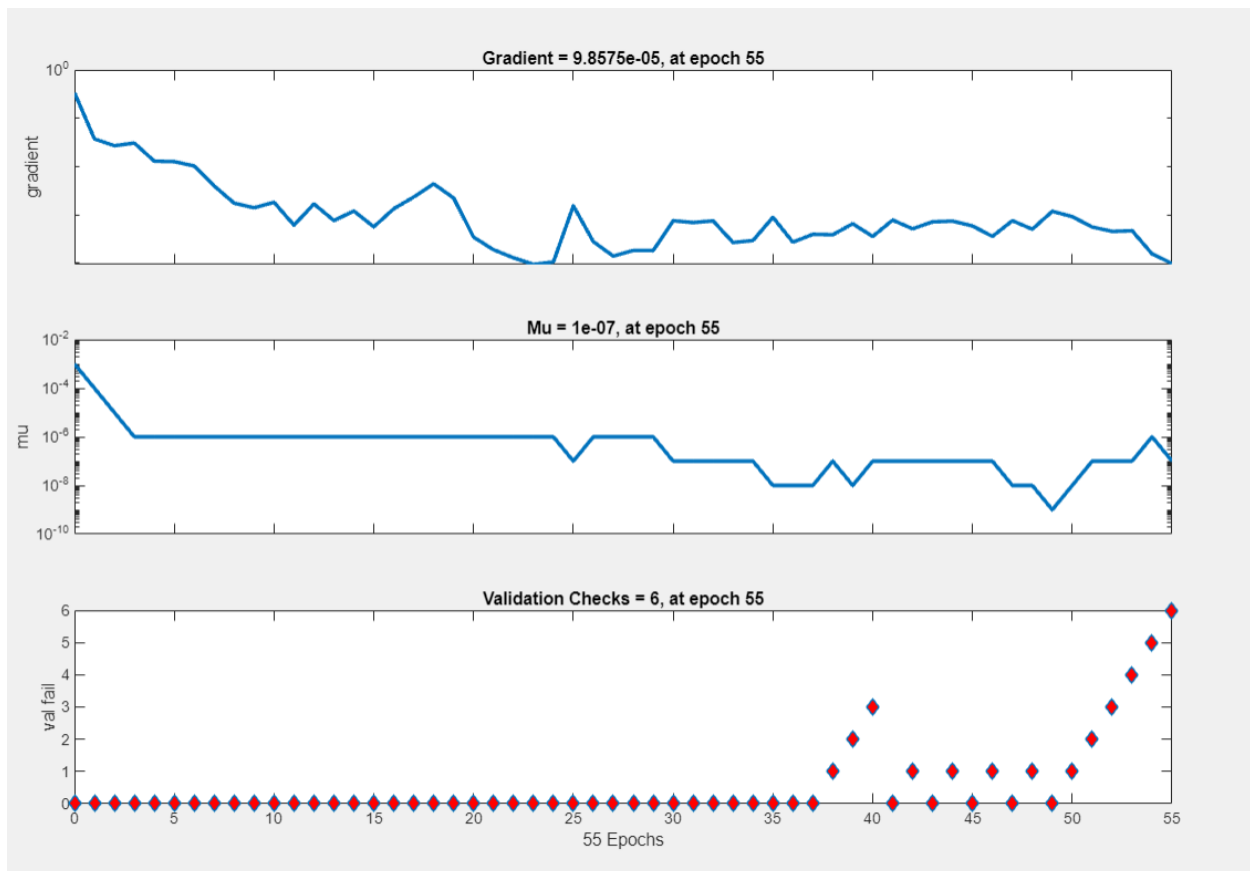
```
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end
```

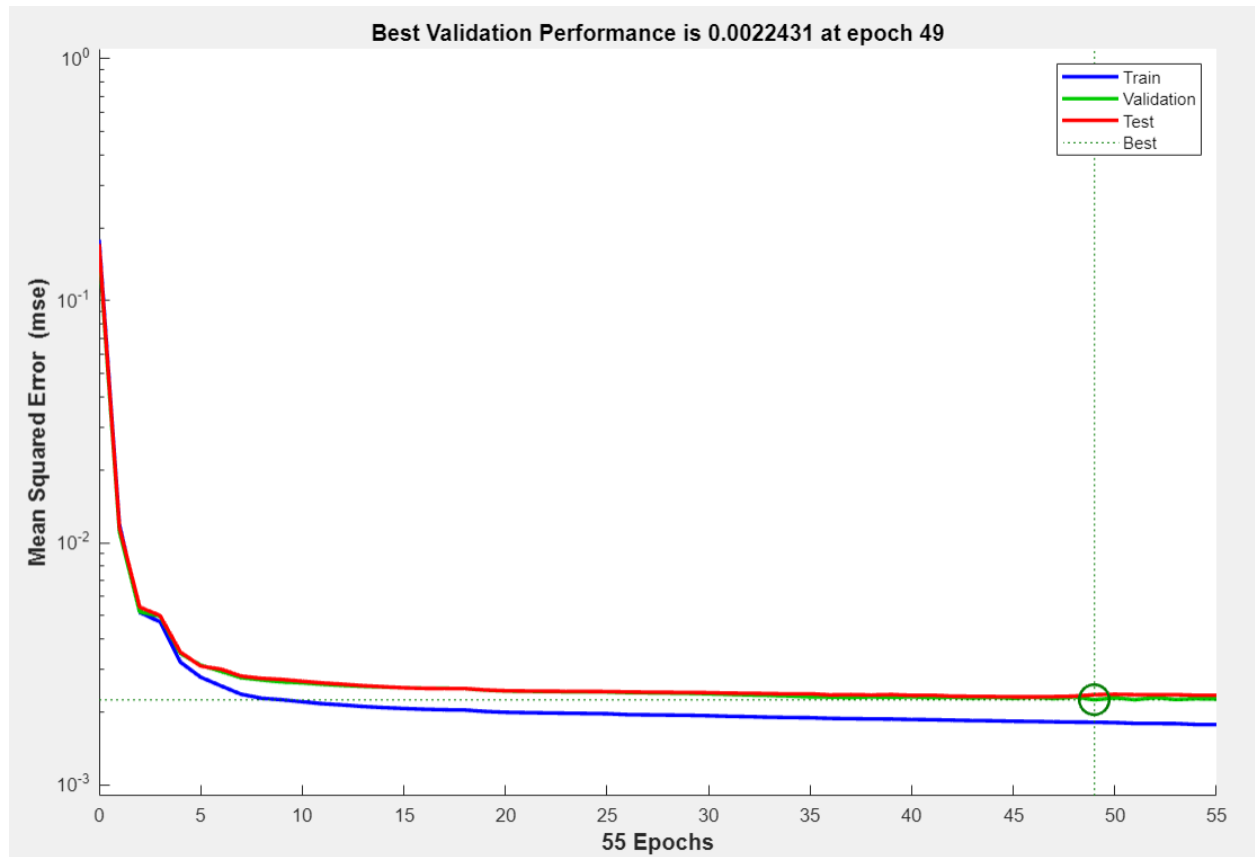

با کمی تغییرات به صورت زیر با استفاده از ۲۰ لایه میانی عملیات را انجام میدهیم. داده تست ۲۰ درصد و داده ولیدیشن هم ۱۰ درصد است.



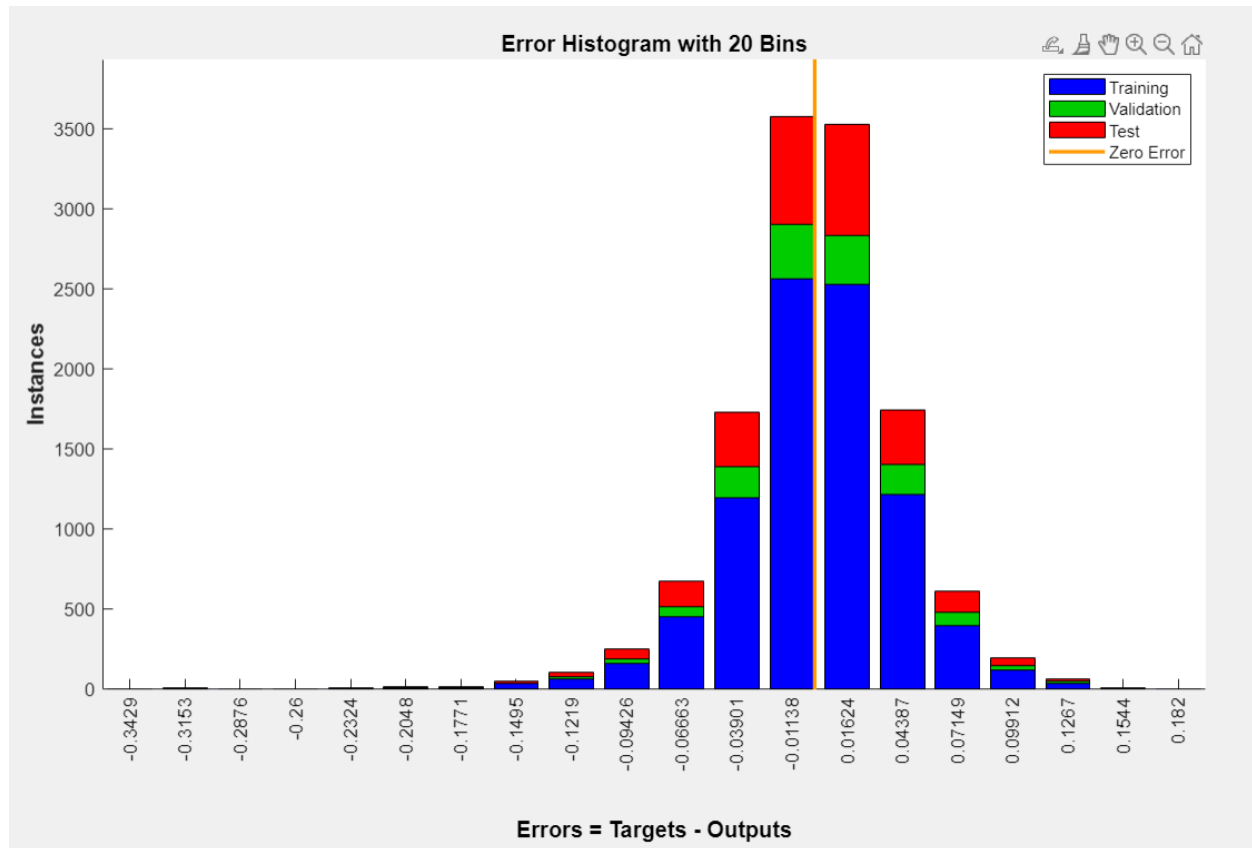
وضعیت آموزش به صورت زیر است:



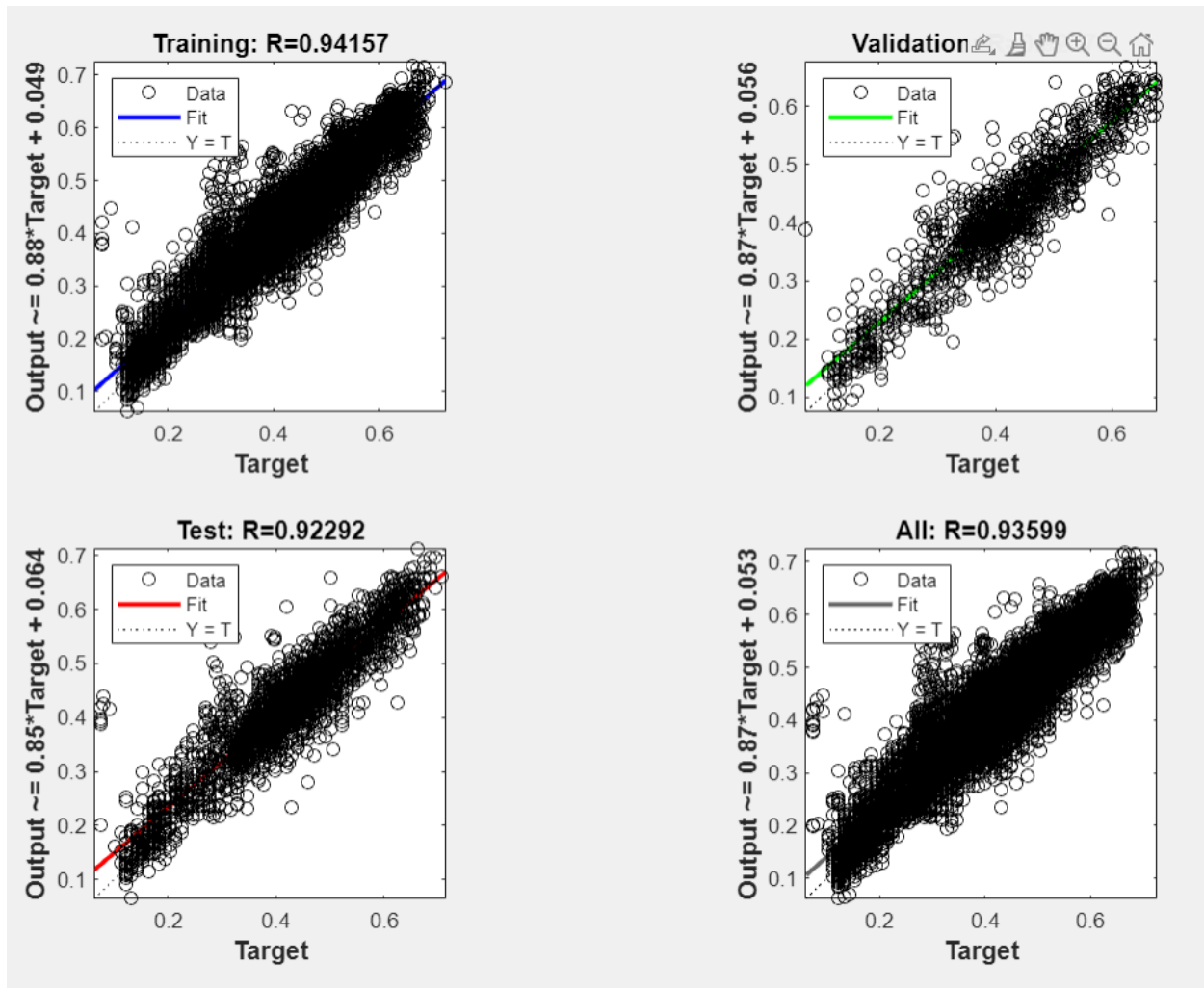
پرفورمنس به صورت زیر است:



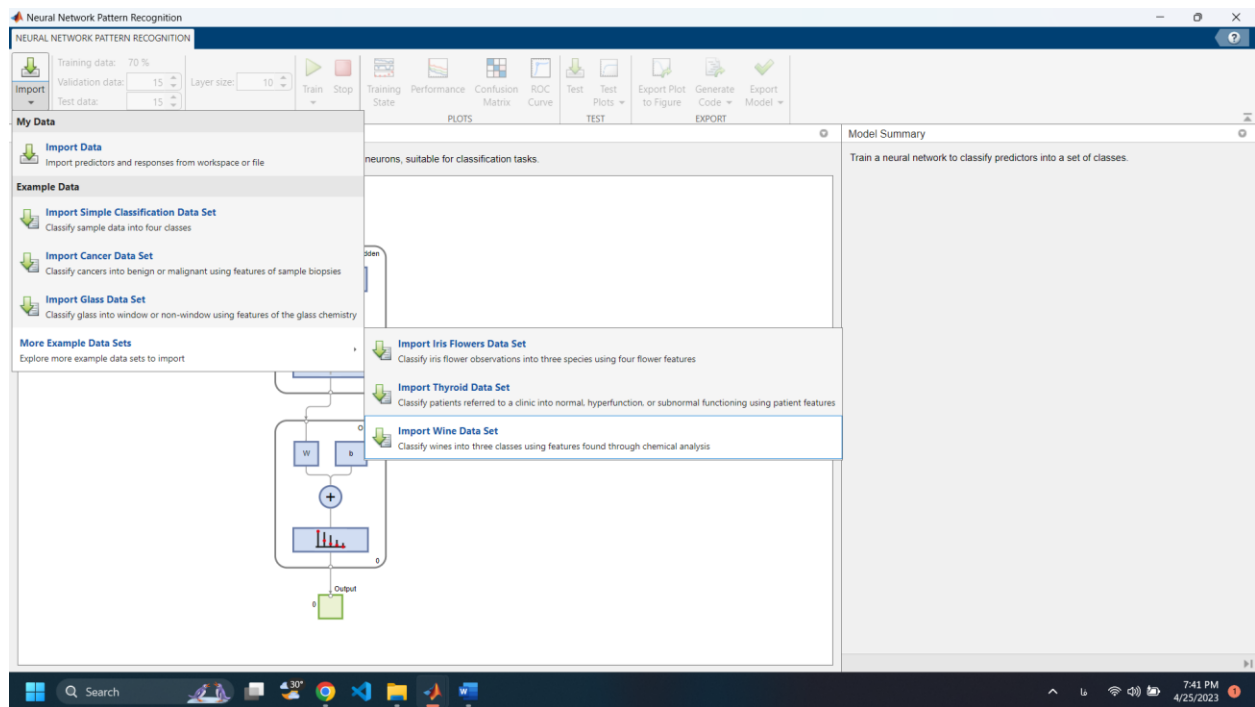
هیستوگرام ارور هم به صورت زیر است که تغییرات زیادی نسبت به نمونه قبلی در آن میبینیم:



اطلاعات رگرسیون به صورت زیر می باشد:



برای کلاسیفای هم به صورت زیر یک نمونه ساختیم:



نتیجه آموزش با مقادیر دیفالت و ۱۰ لایه نهان به صورت زیر است:

Neural Network Pattern Recognition

Training data: 70 %
Validation data: 15 %
Test data: 15 %
Layer size: 10

Import Training Performance Confusion Matrix ROC Curve Test Test Plots Export Plot to Figure Generate Code Export Model

DATA SPLIT BUILD TRAIN PLOTS TEST EXPORT

Model Summary

Train a neural network to classify predictors into a set of classes.

Training Results

Training finished: Reached minimum gradient

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	37	1000
Elapsed Time	-	00:00:00	-
Performance	0.588	2.33e-07	0
Gradient	0.396	5.63e-07	1e-06
Validation Checks	0	0	6

Algorithm

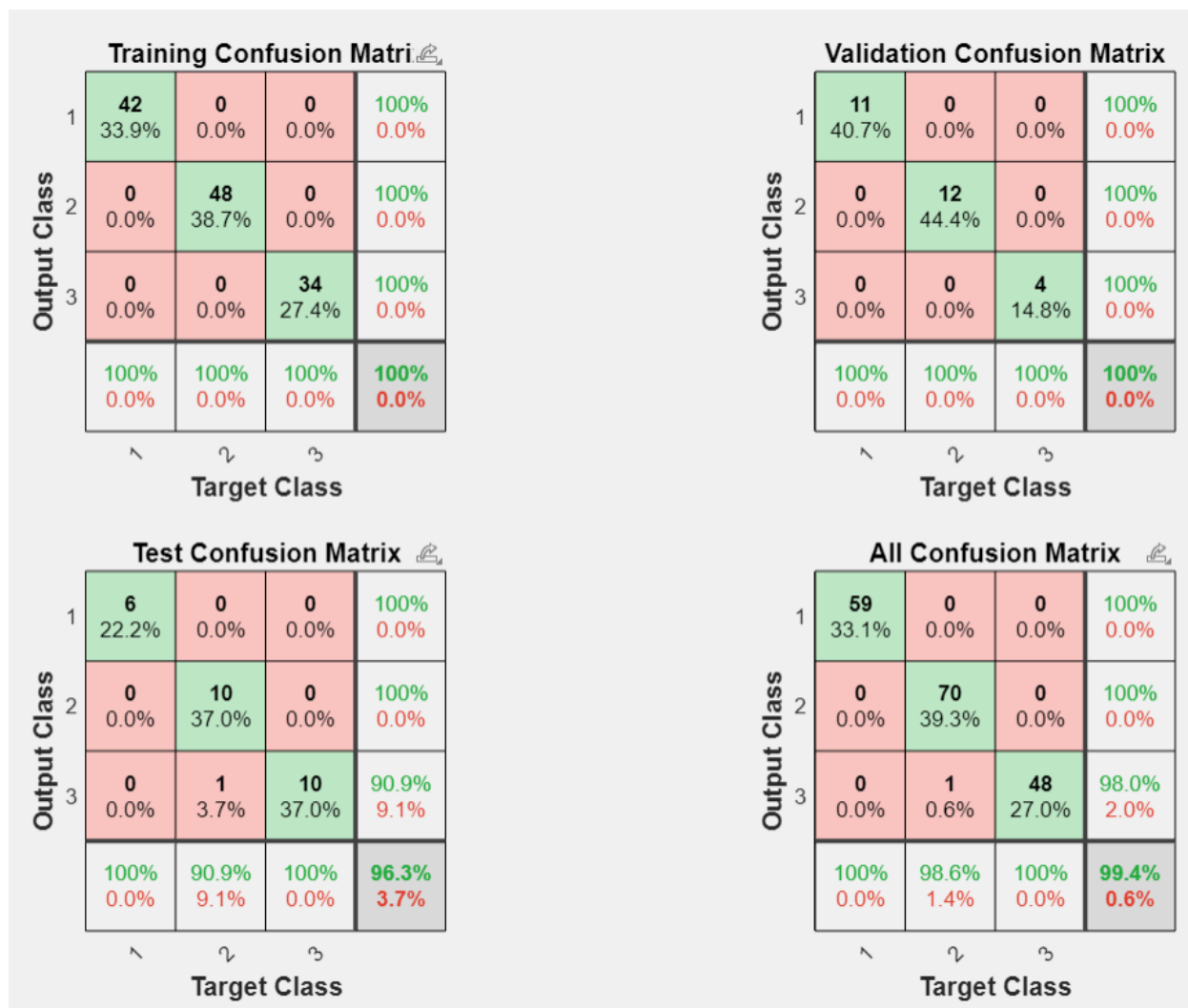
Data division: Random
Training algorithm: Scaled conjugate gradient
Performance: Cross-entropy error

Training Results

Training start time: 25-Apr-2023 19:42:27
Layer size: 10

	Observations	Cross-entropy	Error
Training	124	0.0000	0
Validation	27	0.0000	0
Test	27	0.0488	0.0370

ماتریس کانفیژن آن به صورت زیر است:



نمودار roc آن هم به صورت زیر است:

