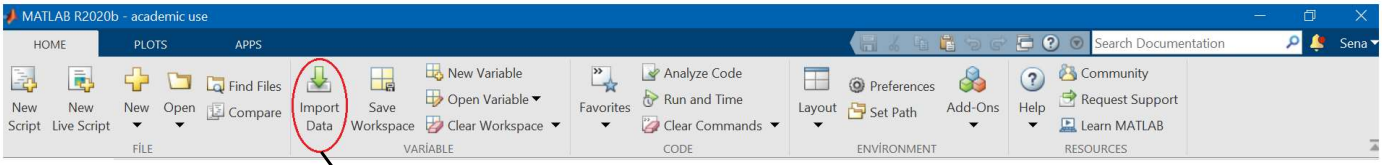
YAPAY ZEKA VİZE PROJESİ

İki farklı veri kümesinin excel üzerinde oluşturup yapay sinir ağlarıyla eğittikten sonra yine Excel üzerinde oluşturduğumuz test verileri ile doğruluk paylarını karşılaştırdığımız projenin detayları aşağıdaki gibidir;

A ve B sayısının birbirleriyle bölünmesi sonucu SONUC kısmı ortaya çıktı.

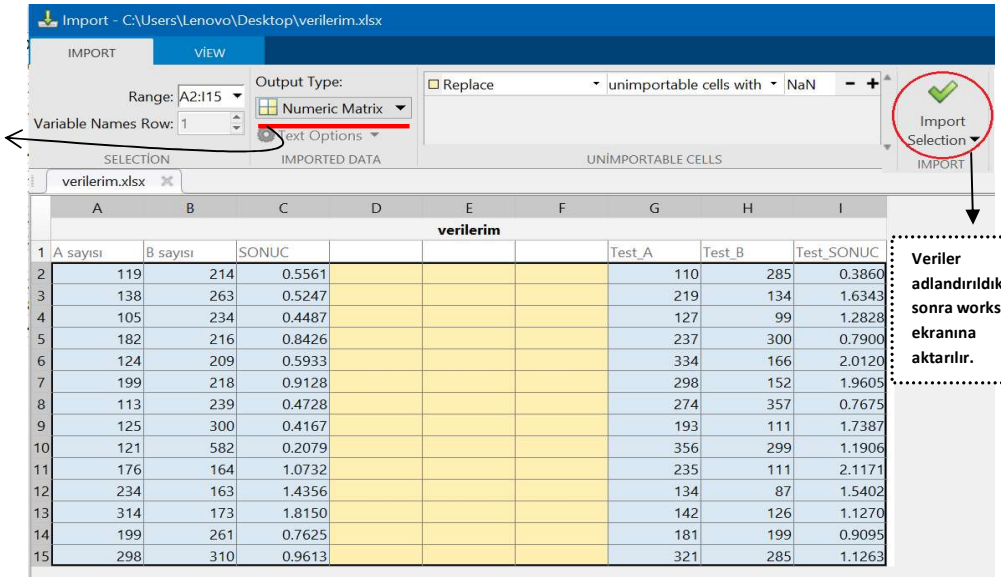
	A	B	C	D	E	F	G	H	I
1	A sayısı	B sayısı	SONUC				Test_A	Test_B	Test_SONUC
2	119	214	0,556075				110	285	0,38596491
3	138	263	0,524715				219	134	1,63432836
4	105	234	0,448718				127	99	1,28282828
5	182	216	0,842593				237	300	0,79
6	124	209	0,593301				334	166	2,01204819
7	199	218	0,912844				298	152	1,96052632
8	113	239	0,472803				274	357	0,767507
9	125	300	0,416667				193	111	1,73873874
10	121	582	0,207904				356	299	1,19063545
11	176	164	1,073171				235	111	2,11711712
12	234	163	1,435583				134	87	1,54022989
13	314	173	1,815029				142	126	1,12698413
14	199	261	0,762452				181	199	0,90954774
15	298	310	0,96129				321	285	1,12631579

Gerçek test sonuç verilerinin Yapay Sinir Ağları ile eğitilen tahmin sonuçlarıyla karşılaştırılması sonucu projenin amacına ulaşıldı.



Import data butonuna tıkladıktan sonra Excel dosyamızı seçip Excel üzerindeki verilerin Matlab üzerine aktarımı sağlanır.

Burada bulunan seçeneklerden 'Numeric Matrix' seçeneği seçilir ve veriler sınıflandırılmaya başlanır.



	A	B	C	D	E	F	G	H	I
1	A sayısı	B sayısı	SONUC				Test_A	Test_B	Test_SONUC
2	119	214	0.5561				110	285	0.3860
3	138	263	0.5247				219	134	1.6343
4	105	234	0.4487				127	99	1.2828
5	182	216	0.8426				237	300	0.7900
6	124	209	0.5933				334	166	2.0120
7	199	218	0.9128				298	152	1.9605
8	113	239	0.4728				274	357	0.7675
9	125	300	0.4167				193	111	1.7387
10	121	582	0.2079				356	299	1.1906
11	176	164	1.0732				235	111	2.1171
12	234	163	1.4356				134	87	1.5402
13	314	173	1.8150				142	126	1.1270
14	199	261	0.7625				181	199	0.9095
15	298	310	0.9613				321	285	1.1263

Veriler adlandırıldıktan sonra workspace ekranına aktarılır.

İlk iki satıra 'input' adı verildi.

Üçüncü satıra 'target' adı verildi.

İlk iki test satırına 'test_input' adı verildi.

Üçüncü test satırına 'test_target' adı verildi.

input

	A sayısı	B sayısı	SONUC
1	119	214	0.5561
2	138	263	0.5247
3	105	234	0.4487
4	182	216	0.8426
5	124	209	0.5933
6	199	218	0.9128
7	113	239	0.4728
8	125	300	0.4167
9	121	582	0.2079
10	176	164	1.0732
11	234	163	1.4356
12	314	173	1.8150
13	199	261	0.7625
14	298	310	0.9613

target

	A sayısı	B sayısı	SONUC
1	119	214	0.5561
2	138	263	0.5247
3	105	234	0.4487
4	182	216	0.8426
5	124	209	0.5933
6	199	218	0.9128
7	113	239	0.4728
8	125	300	0.4167
9	121	582	0.2079
10	176	164	1.0732
11	234	163	1.4356
12	314	173	1.8150
13	199	261	0.7625
14	298	310	0.9613

test_input

	Test_A	Test_B	Test_SONUC
1	110	285	0.3860
2	219	134	1.6343
3	127	99	1.2828
4	237	300	0.7900
5	334	166	2.0120
6	298	152	1.9605
7	274	357	0.7675
8	193	111	1.7387
9	356	299	1.1906
10	235	111	2.1171
11	134	87	1.5402
12	142	126	1.1270
13	181	199	0.9095
14	321	285	1.1263

test_target

	Test_A	Test_B	Test_SONUC
1	110	285	0.3860
2	219	134	1.6343
3	127	99	1.2828
4	237	300	0.7900
5	334	166	2.0120
6	298	152	1.9605
7	274	357	0.7675
8	193	111	1.7387
9	356	299	1.1906
10	235	111	2.1171
11	134	87	1.5402
12	142	126	1.1270
13	181	199	0.9095
14	321	285	1.1263

Workspace

Name	Value	Size
input	14x2 double	14x2
target	14x1 double	14x1
test_input	14x2 double	14x2
test_target	14x1 double	14x1

Veriler sınıflandırılıp workspace kısmına aktarıldıktan sonra dört nümerik matrislerin transpozu alınır.

Transpoze = Satırların sütunlarıyla yer değiştirmesidir.

Dört nümerik matrisin transpozu alınır.

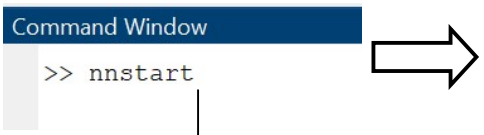
Command Window

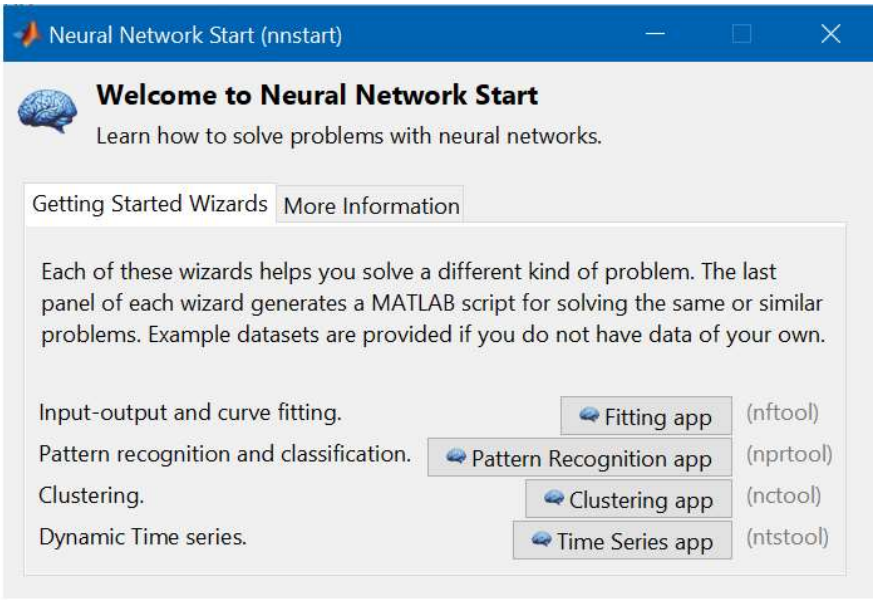
```
>> input=input'

input =

   119   138   105   182   124   199   113   125   121   176   234   314   199   298
   214   263   234   216   209   218   239   300   582   164   163   173   261   310

>> target=target'

target =

  Columns 1 through 11

    0.5561    0.5247    0.4487    0.8426    0.5933    0.9128    0.4728    0.4167    0.2079    1.0732    1.4356

  Columns 12 through 14

    1.8150    0.7625    0.9613

>> test_input=test_input'

test_input =

   110   219   127   237   334   298   274   193   356   235   134   142   181   321
   285   134    99   300   166   152   357   111   299   111    87   126   199   285

>> test_target=test_target'

test_target =

  Columns 1 through 11

    0.3860    1.6343    1.2828    0.7900    2.0120    1.9605    0.7675    1.7387    1.1906    2.1171    1.5402

  Columns 12 through 14

    1.1270    0.9095    1.1263
```

Girildikten sonra matrislerin boyutları yer değiştirir.

Workspace

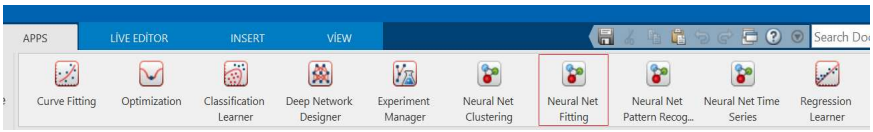
Name	Value	Size
input	2x14 double	2x14
target	1x14 double	1x14
test_input	2x14 double	2x14
test_target	1x14 double	1x14

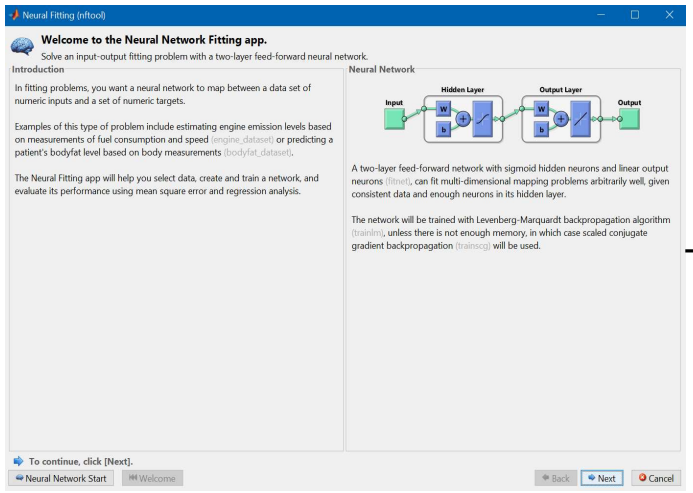


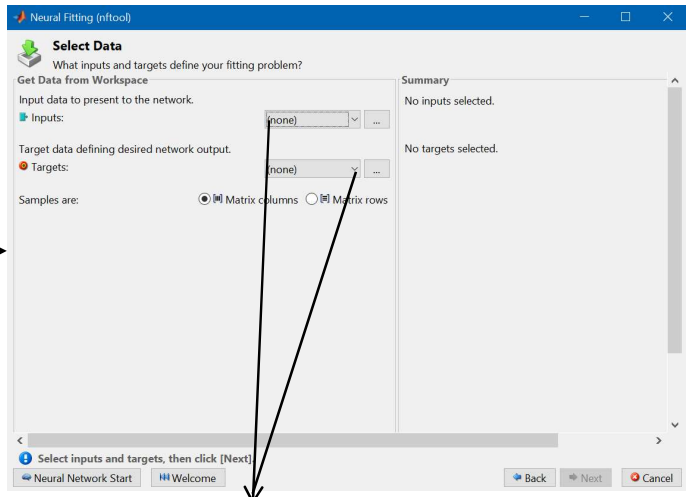


Command window kısmına yazılan nnstart (neural network) komutu sağ tarafta görünen araç kutusu kısmını açar ve buradan Fitting app uygulaması açılır.

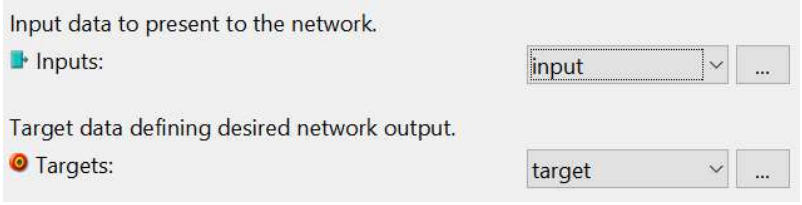
Fitting app uygulamasının açılmasının başka bir yolu ise matlab-->apps kısmından Neural Net Fitting seçeneğinden tıklanarak açılabilir.



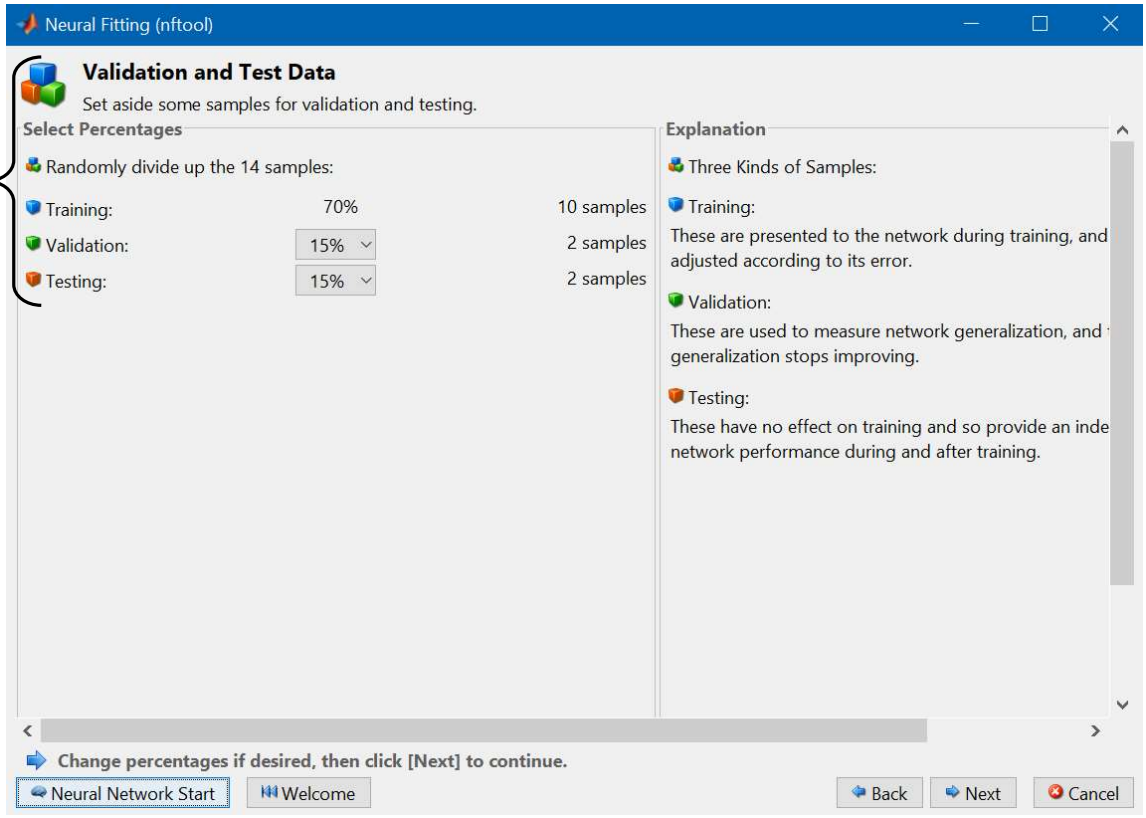




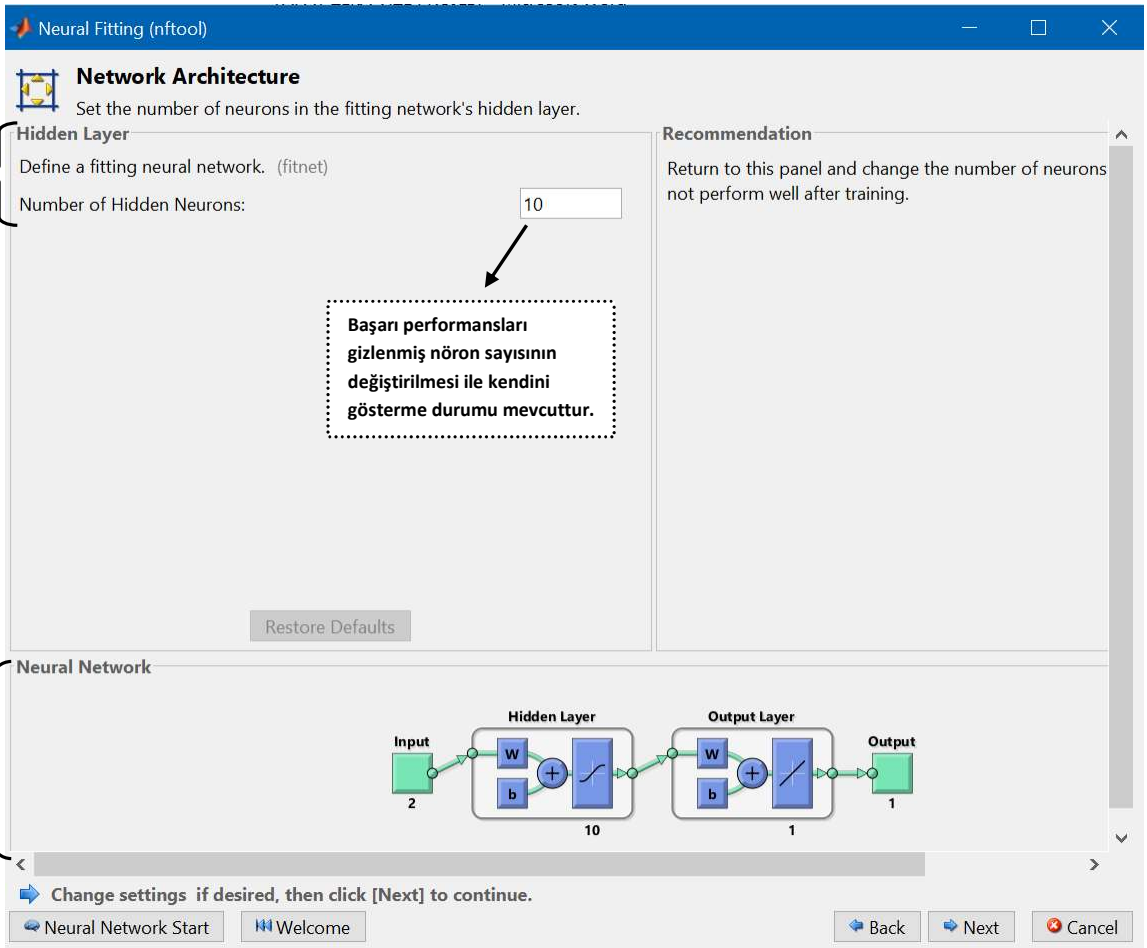
Buradan asıl verilerin giriş ve çıkış dataları seçilir.



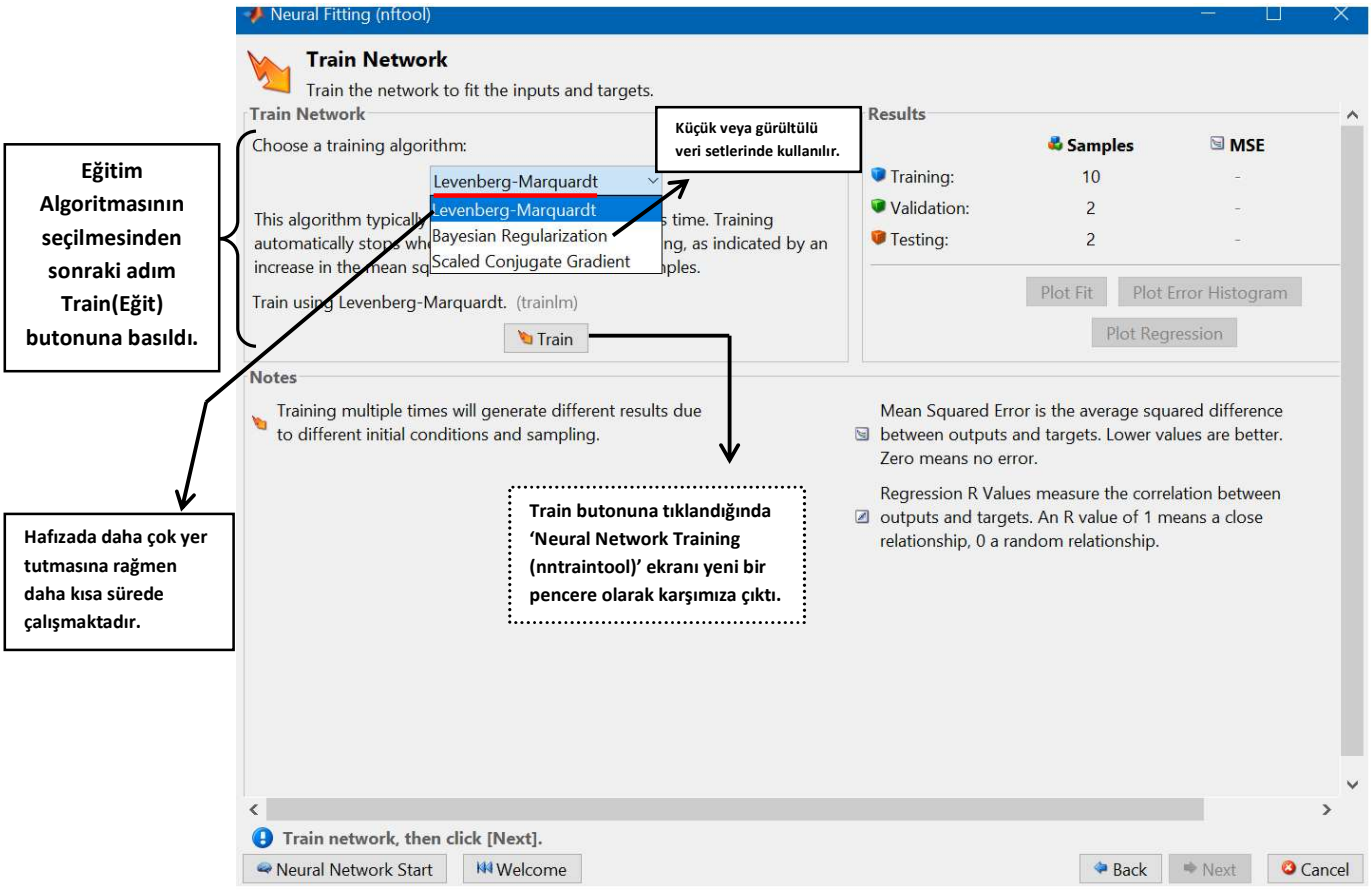
Training,val ve test oranları 0.70,0.15,0.15 alındı.Eğitim için oran %70 alındı.

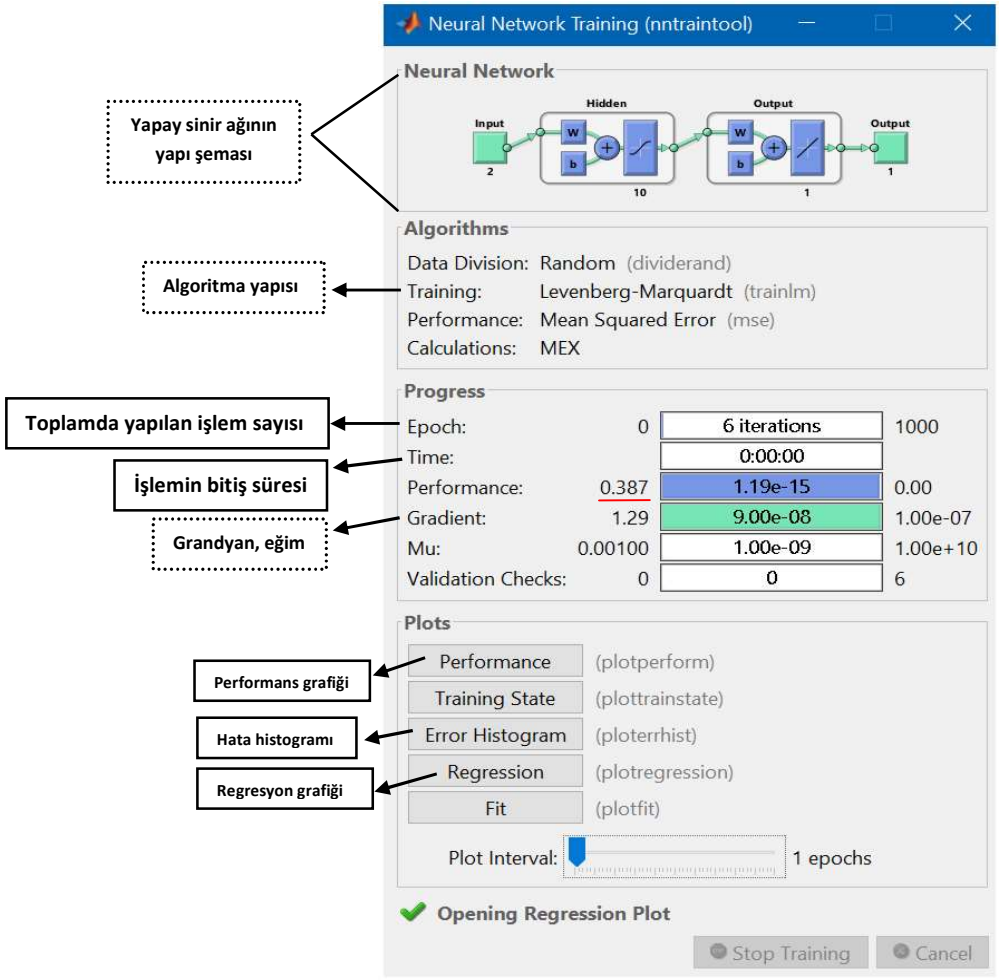


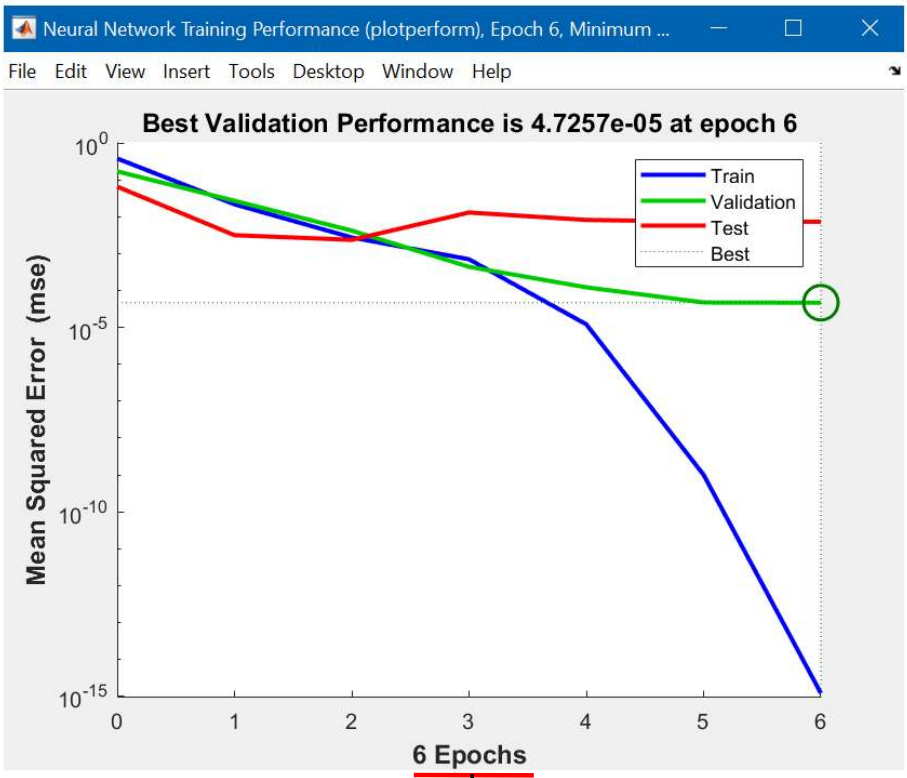
Gizlenmiş Nöron sayısı 10' a ayarlanarak sonraki kısma geçildi.



Burada ağın yapı şeması ortaya çıktı.

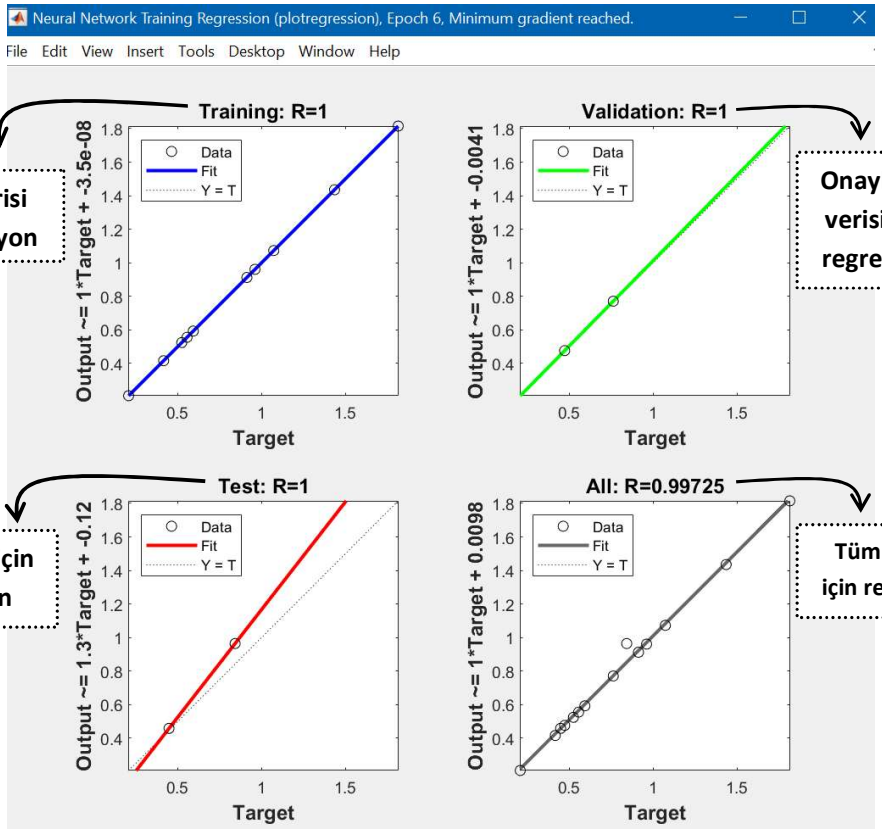

Neural Fitting (nftool)
Train Network
Train the network to fit the inputs and targets.
Train Network
Choose a training algorithm:
Levenberg-Marquardt
Levenberg-Marquardt
Bayesian Regularization
Scaled Conjugate Gradient
Küçük veya gürültülü veri setlerinde kullanılır.
Eğitim Algoritmasının seçilmesinden sonraki adım Train(Eğit) butonuna basıldı.
Train using Levenberg-Marquardt. (trainlm)
Train
Results
Samples
MSE
Training: 10
Validation: 2
Testing: 2
Plot Fit
Plot Error Histogram
Plot Regression
Notes
Training multiple times will generate different results due to different initial conditions and sampling.
Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.
Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.
Hafızada daha çok yer tutmasına rağmen daha kısa sürede çalışmaktadır.
Train butonuna tıklandığında 'Neural Network Training (nntraintool)' ekranı yeni bir pencere olarak karşımıza çıktı.
Train network, then click [Next].
Neural Network Start
Welcome
Back
Next
Cancel


Neural Network Training (nntraintool)
Neural Network
Yapay sinir ağının yapı şeması
Input
Hidden
Output
Output
Algorithms
Algoritma yapısı
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX
Progress
Toplamda yapılan işlem sayısı
İşlemin bitiş süresi
Grandyan, eğim
Epoch: 0 6 iterations 1000
Time: 0:00:00
Performance: 0.387 1.19e-15 0.00
Gradient: 1.29 9.00e-08 1.00e-07
Mu: 0.00100 1.00e-09 1.00e+10
Validation Checks: 0 0 6
Plots
Performans grafiği
Hata histogramı
Regresyon grafiği
Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Regression (plotregression)
Fit (plotfit)
Plot Interval: 1 epochs
Opening Regression Plot
Stop Training
Cancel



Performans Grafiği (Öğrenme Grafiği)

Algoritmanın veri setini kaç kez ziyaret ettiğini gösterir.



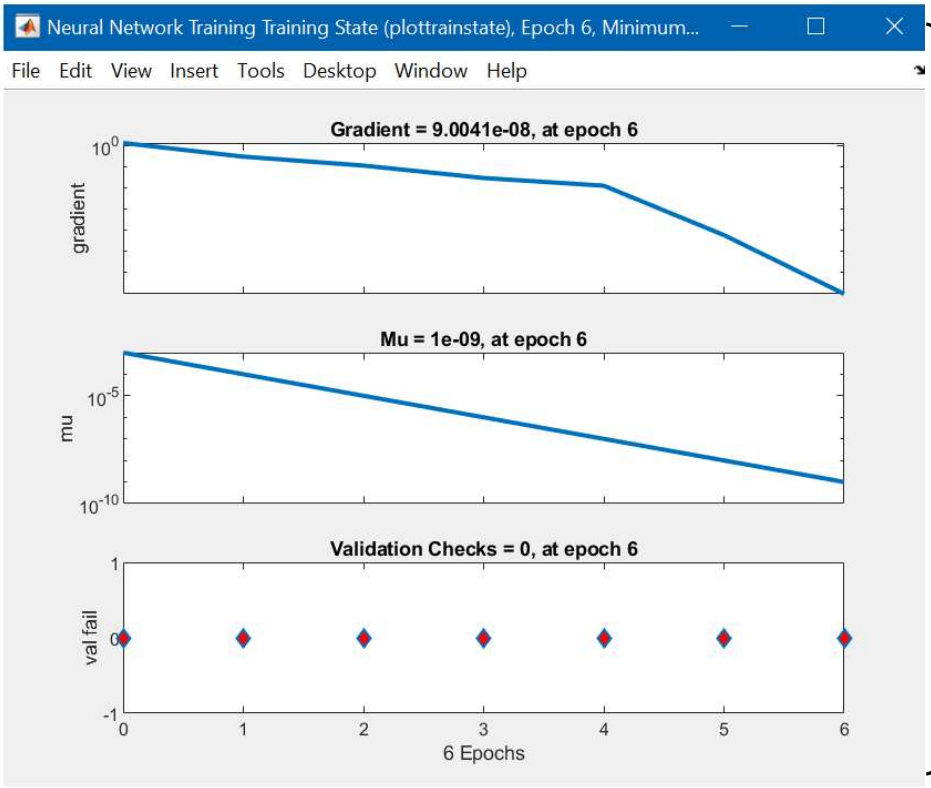
Eğitim verisi için regresyon

Onaylama verisi için regresyon

Test serisi için regresyon

Tüm veriler için regresyon

Regresyon Grafiği

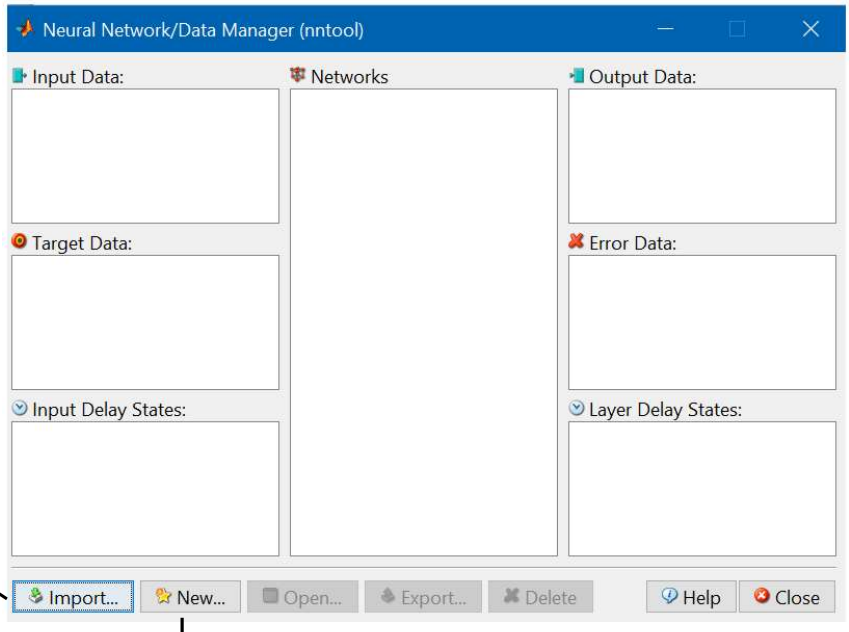


Eğitim durumu değerleri grafiği

Buradaki grafikler incelendikten sonra ekran kapatılır ve command window ekranına son kez kullanabileceğimiz nntool eklentisi yazılır.

```
Command Window
>> nntool
Warning: nntool will be removed in a future release. Use nnstart instead.
> In nntool (line 17)
```

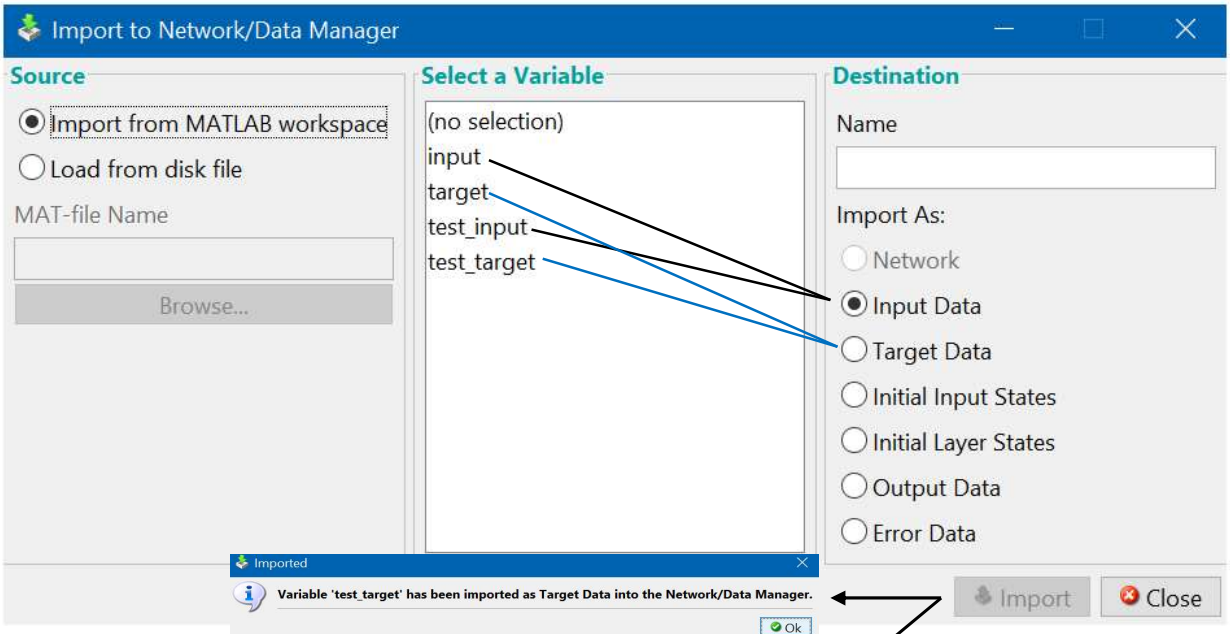
Enter tuşuna basıldığında ekrana veri yöneticisi ekranı açılır.



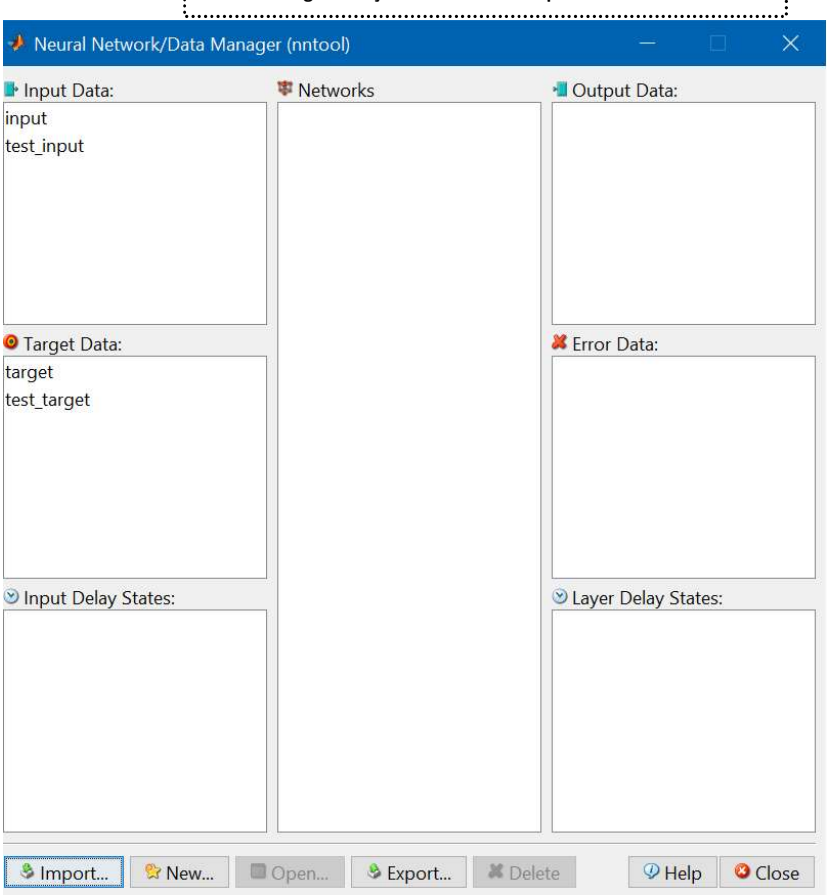
Bu kısımda veriler sınıflandırılır.

Yeni network oluşturur.

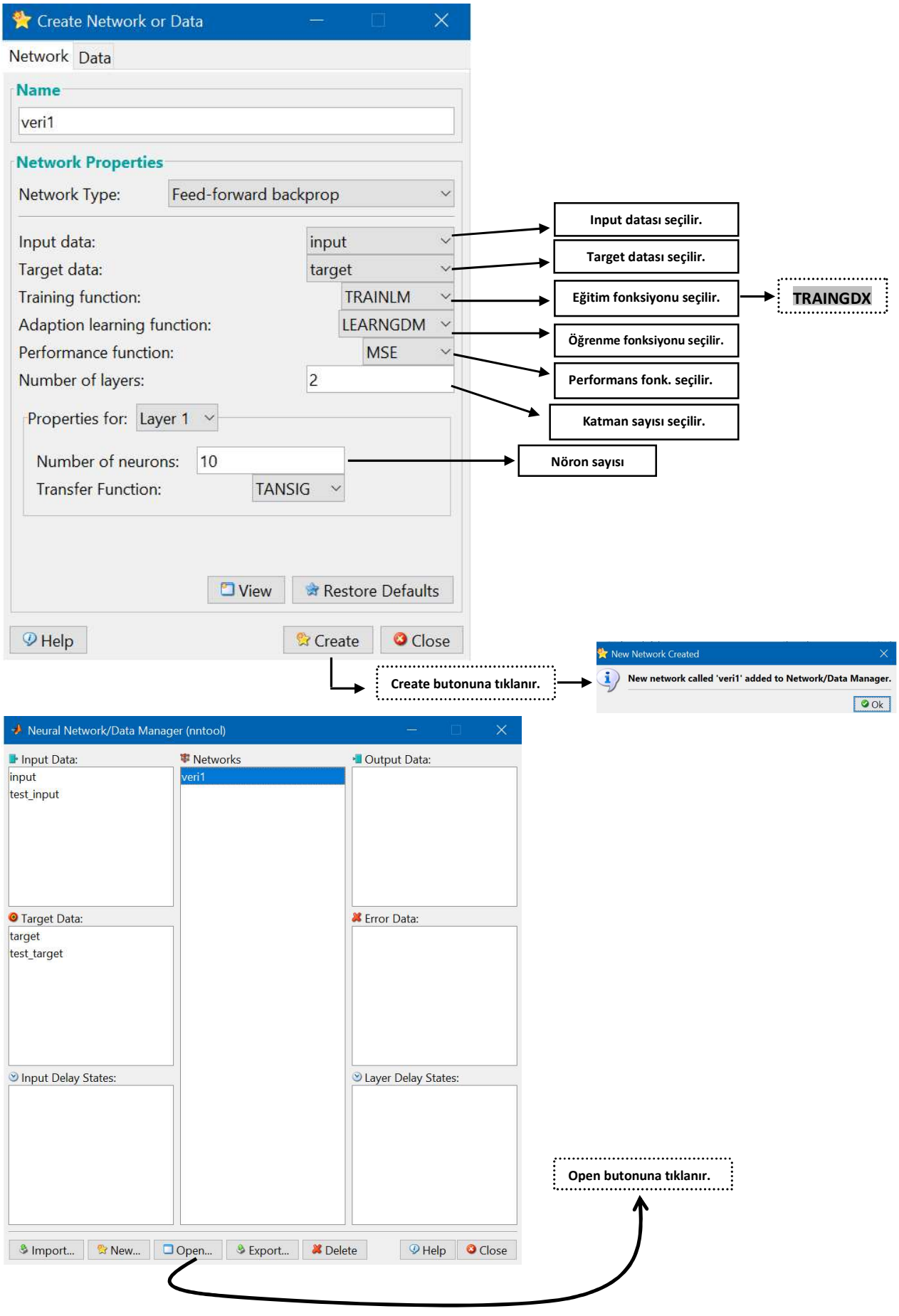
Import butonuna tıklandığında;

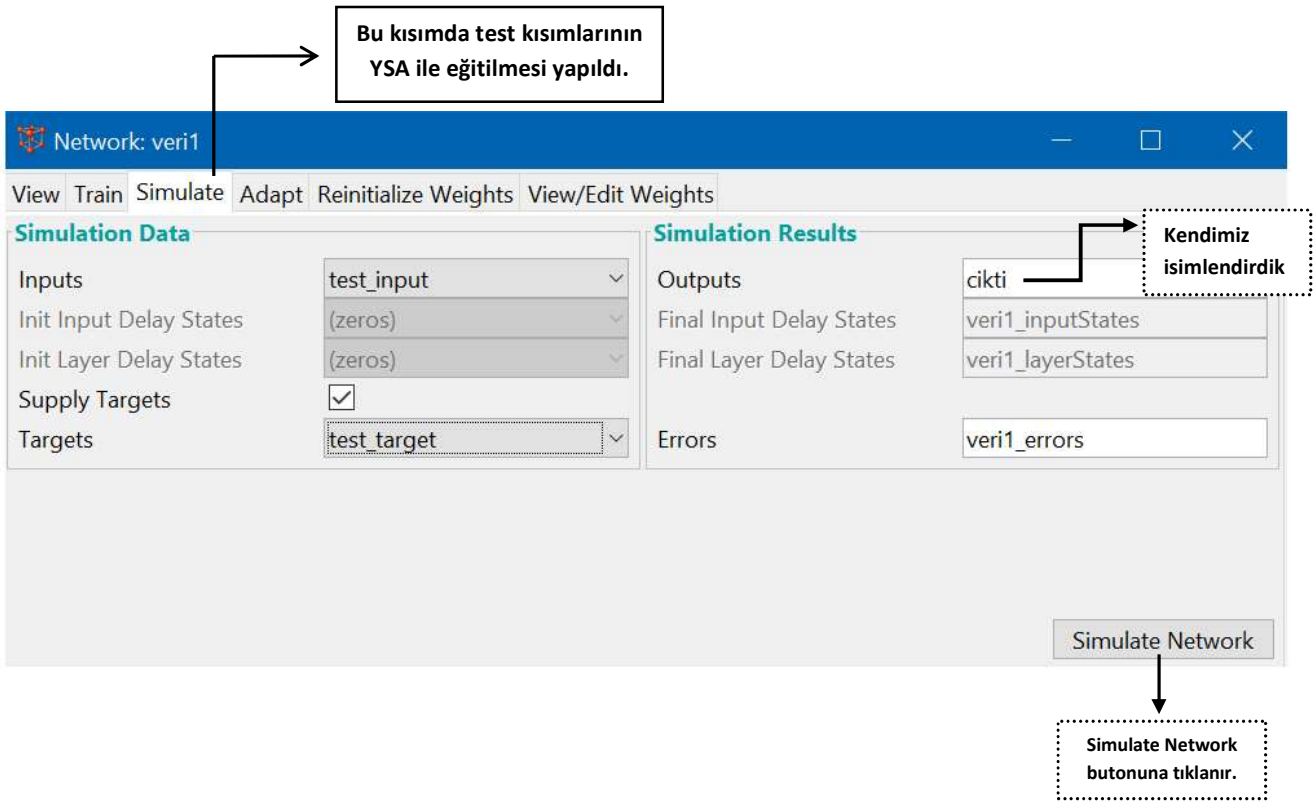


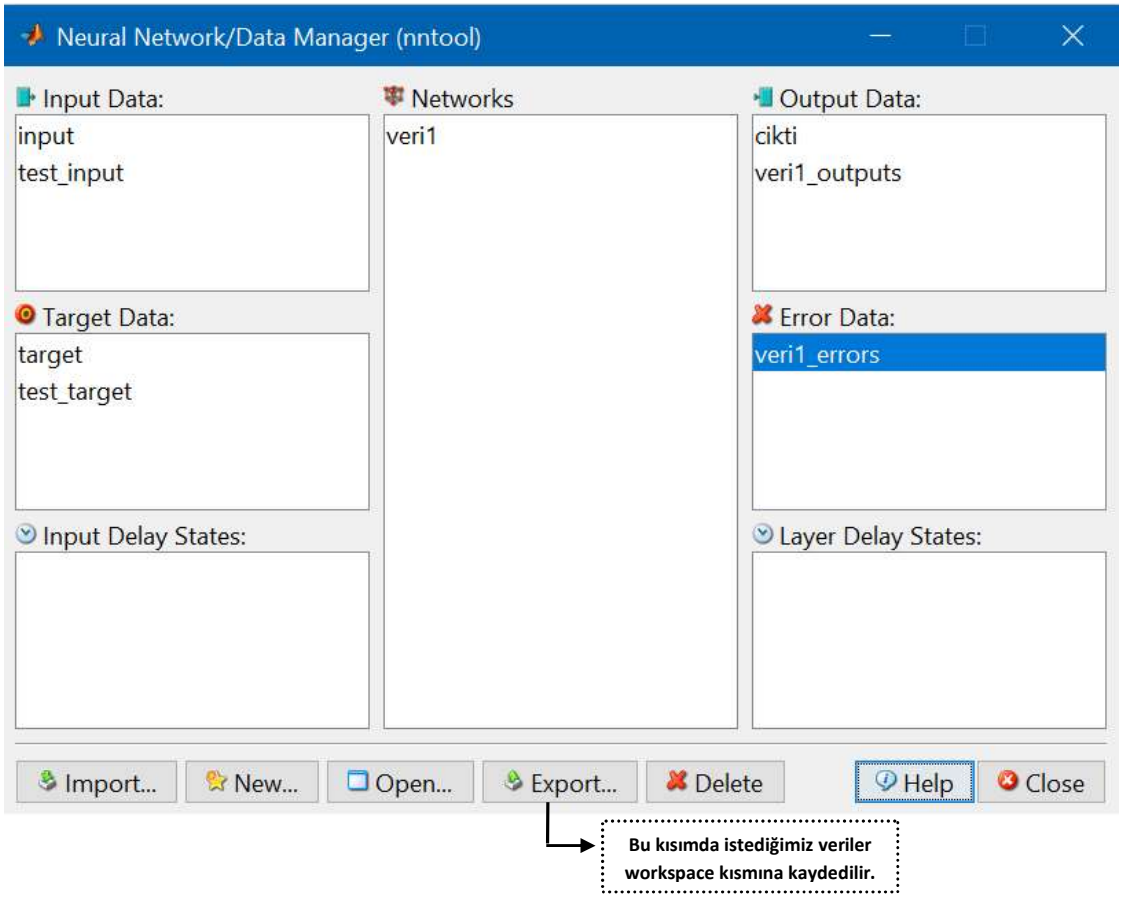
Verilerin hangi sınıfa ait olduğu seçildikten sonra ımport butonuna tıklandığında seçilen dataların workspace kısmına aktarılır.

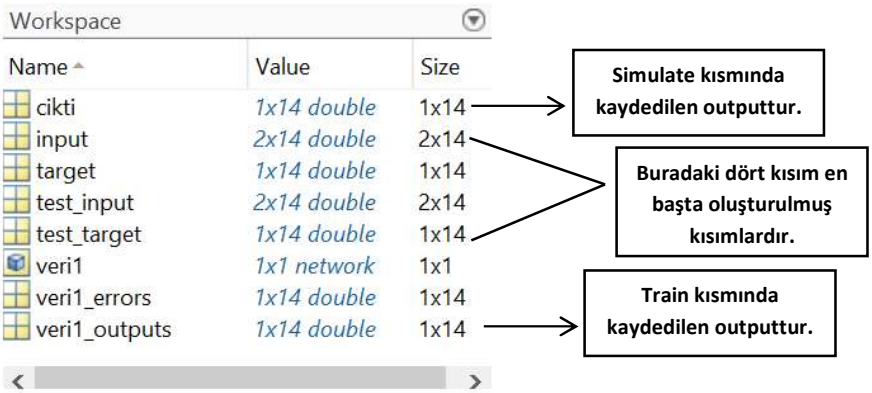


New butonuna tıklandığında

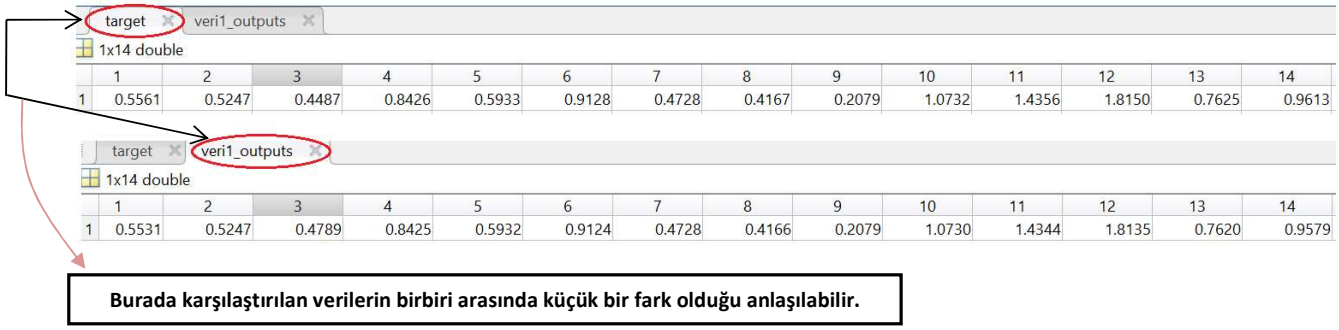

Create Network or Data
Network
Data
Name
veri1
Network Properties
Network Type:
Feed-forward backprop
Input data:
input
Target data:
target
Training function:
TRAINLM
Adaption learning function:
LEARNGDM
Performance function:
MSE
Number of layers:
2
Properties for:
Layer 1
Number of neurons:
10
Transfer Function:
TANSIG
View
Restore Defaults
Help
Create
Close
Input datası seçilir.
Target datası seçilir.
Eğitim fonksiyonu seçilir.
TRAINGDX
Öğrenme fonksiyonu seçilir.
Performans fonk. seçilir.
Katman sayısı seçilir.
Nöron sayısı
Create butonuna tıklanır.
New Network Created
New network called 'veri1' added to Network/Data Manager.
Ok
Neural Network/Data Manager (nntool)
Input Data:
input
test_input
Networks
veri1
Output Data:
Target Data:
target
test_target
Error Data:
Input Delay States:
Layer Delay States:
Import...
New...
Open...
Export...
Delete
Help
Close
Open butonuna tıklanır.


Bu kısımda test kısımlarının YSA ile eğitilmesi yapıldı.
Network: veri1
View
Train
Simulate
Adapt
Reinitialize Weights
View/Edit Weights
Simulation Data
Inputs
test_input
Init Input Delay States
(zeros)
Init Layer Delay States
(zeros)
Supply Targets
Targets
test_target
Simulation Results
Outputs
cikti
Kendimiz isimlendirdik
Final Input Delay States
veri1_inputStates
Final Layer Delay States
veri1_layerStates
Errors
veri1_errors
Simulate Network
Simulate Network butonuna tıklanır.

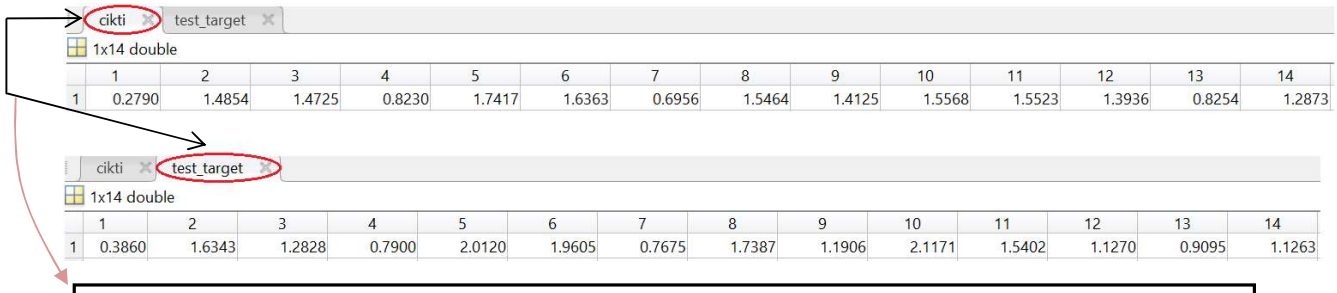

Neural Network/Data Manager (nntool)
Input Data:
input
test_input
Networks
veri1
Output Data:
cikti
veri1_outputs
Target Data:
target
test_target
Error Data:
veri1_errors
Input Delay States:
Layer Delay States:
Import...
New...
Open...
Export...
Delete
Help
Close
Bu kısımda istediğimiz veriler workspace kısmına kaydedilir.



Son olarak veriler workspace kısmına kaydedildikten sonra 'target' ve 'veri1_outputs' kısımları birbirleriyle karşılaştırılır.



Burada karşılaştırılan verilerin birbiri arasında küçük bir fark olduğu anlaşılabilir.

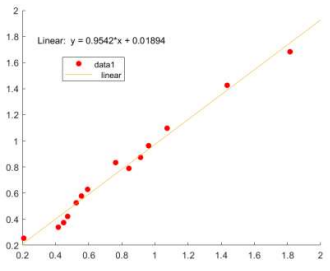


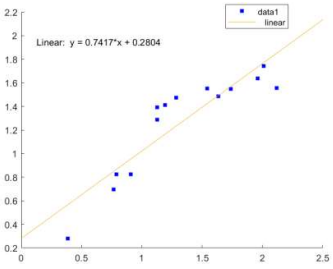
Test verilerinin sonuçlarının YSA ile eğittiğimiz cevaplarımız ile birbirleriyle benzer sonuçları ile benzerlik göstermektedir.

Son olarak 'scatter' plot ile belirlenen data dağılımlarına göre grafik oluşturulur.

```
scatter(target,veri1_outputs,'red',"filled")

scatter(test_target,cikti,'blue',"filled","square")
```

Filled= İşaretlerin içini doldurur.
Marker Type= Belirli şekiller arasından seçim yapılarak grafik üzerinden gösterilir.





Verilerin karşılaştırılması sonucunda görüldüğü üzere Yapay sinir ağları ile test sonuçları eğitilmiştir.

1700001922- Sena Bilgici