ME7021.1 - ARTIFICIAL INTELLIGENT TECHNIQUES IN ENGINEERING APPLICATIONS

Prof.Dr. ERTUĞRUL TAÇGIN

1. EXPERT SYSTEM WITH DOCASSEMBLE

A. PROBLEM

Body mass index (BMI) is a value derived from the mass (weight) and height of a person. The BMI is defined as the body mass divided by the square of the body height, and is universally expressed in units of kg/m2, resulting from mass in kilograms and height in metres.

In this study, it was aimed to calculate the Body Mass Index by obtaining the weight, height and age information from the user and to inform the person about the proximity of obesity.

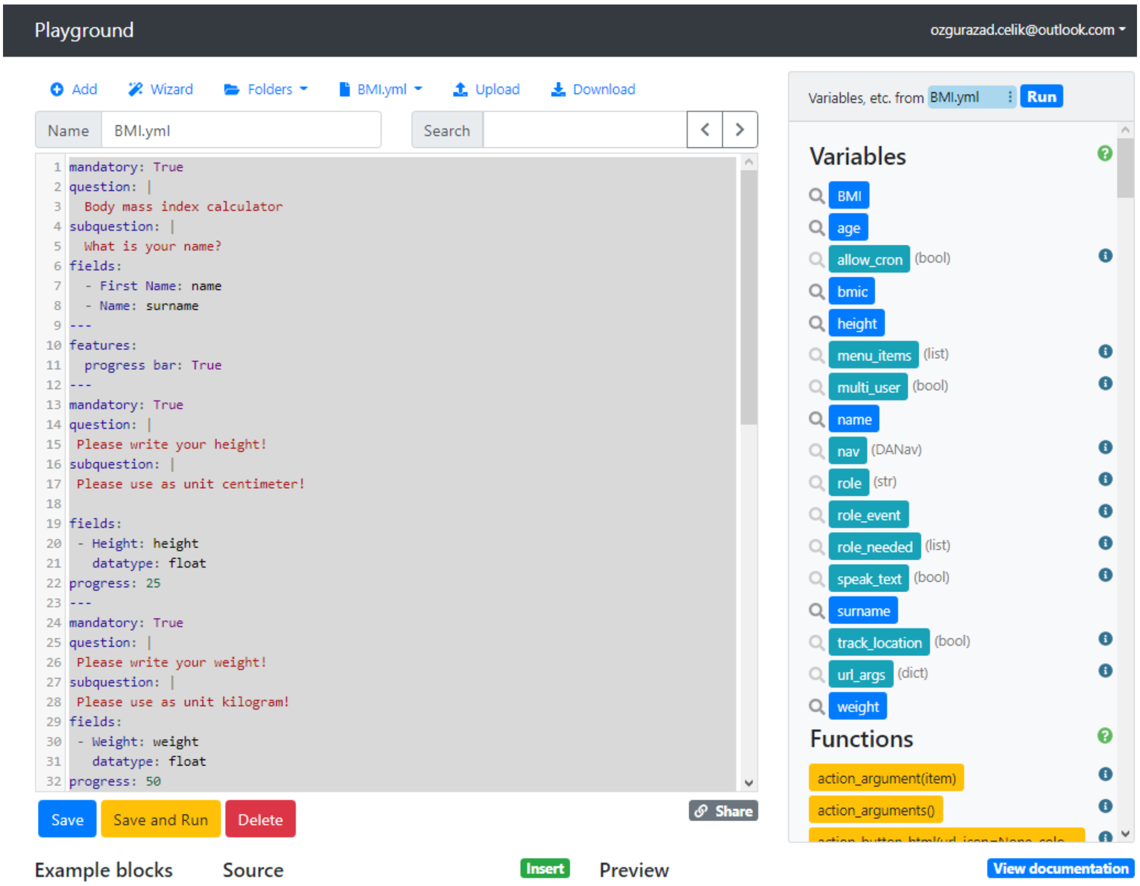
B. CODES

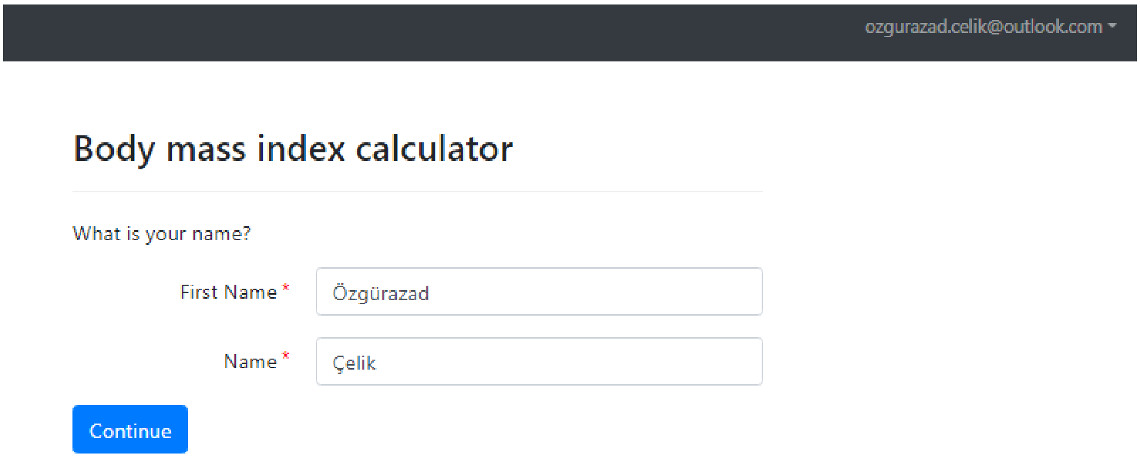
```
mandatory: True
question: |
  Body mass index calculator
subquestion: |
  What is your name?
fields:
  - First Name: name
  - Name: surname
---
features:
  progress bar: True
---
mandatory: True
question: |
 Please write your height!
subquestion: |
 Please use as unit centimeter!


fields:
 - Height: height
```

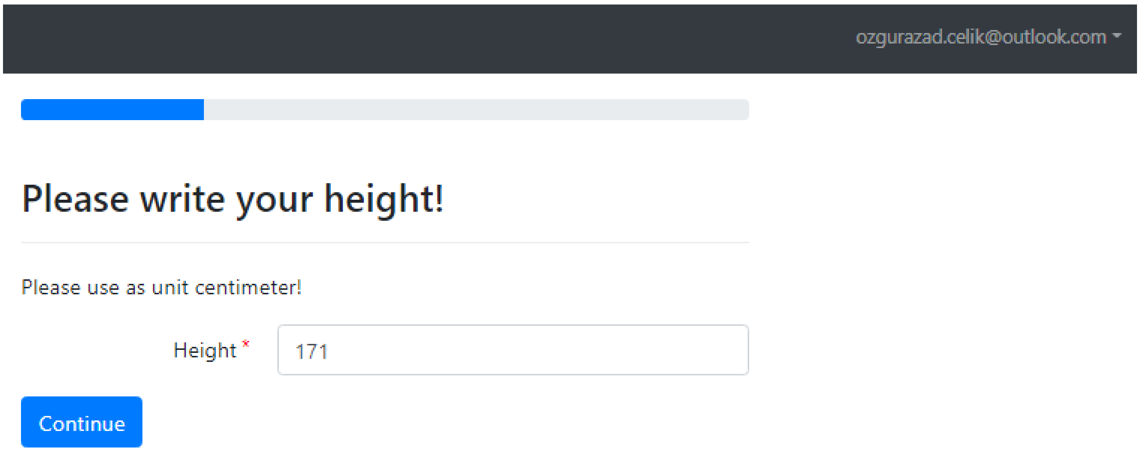
```
   datatype: float
progress: 25
---
mandatory: True
question: |
 Please write your weight!
subquestion: |
 Please use as unit kilogram!
fields:
 - Weight: weight
   datatype: float
progress: 50
---
mandatory: True
question: |
 Please enter your age?
fields:
 - Age: age
   datatype: float
progress: 75
---
code: |
 BMI = (weight*10000)/(height*height)

---
code: |
 if age <= 24:
   bmic = "24"

 elif age <= 34:
   bmic = "25"
```

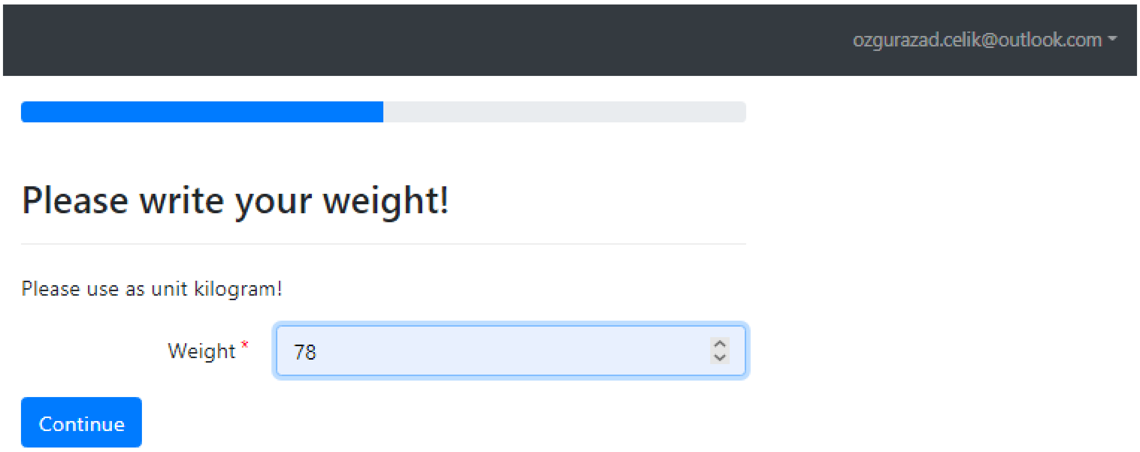
```
 elif age <= 44:
   bmic = "26"

 elif age <= 54:
   bmic = "27"

 elif age <= 65:
   bmic = "28"

 else:
   bmic = "29"
---
mandatory: True
question: |
 Hello, ${ name } ${ surname }!
subquestion: |
 Your Body Mass Index is ${ BMI }

 The minimum required limit for not being obese is ${ bmic }
buttons:
  - Exit: exit
  - Restart: restart
progress: 100
```

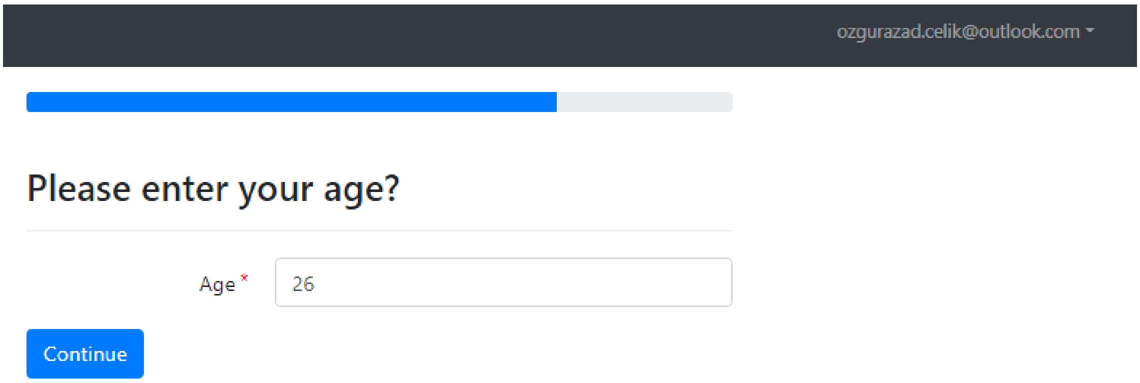
C. RESULTS

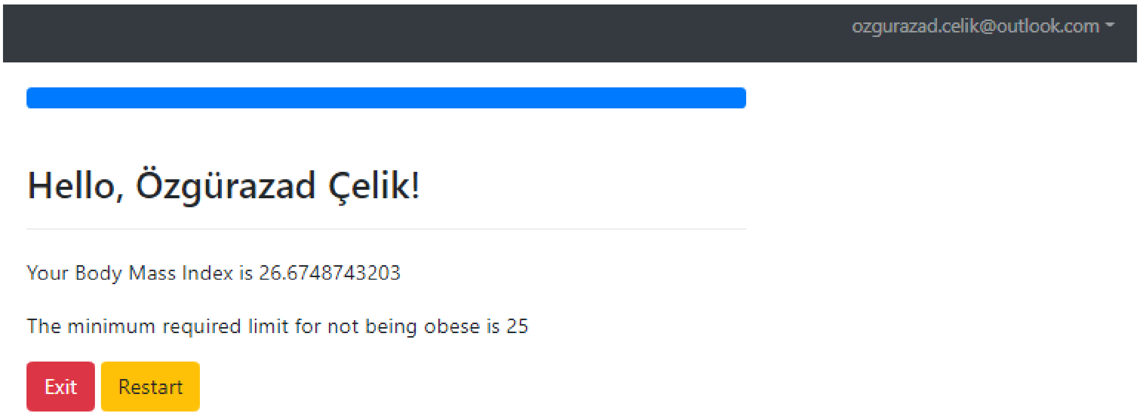

Playground
ozgurazad.celik@outlook.com
Add
Wizard
Folders
BMI.yml
Upload
Download
Name
BMI.yml
Search
mandatory: True
question: |
Body mass index calculator
subquestion: |
What is your name?
fields:
- First Name: name
- Name: surname
---
features:
progress bar: True
---
mandatory: True
question: |
Please write your height!
subquestion: |
Please use as unit centimeter!
fields:
- Height: height
datatype: float
progress: 25
---
mandatory: True
question: |
Please write your weight!
subquestion: |
Please use as unit kilogram!
fields:
- Weight: weight
datatype: float
progress: 50
Save
Save and Run
Delete
Share
Example blocks
Source
Insert
Preview
Variables, etc. from BMI.yml
Run
Variables
BMI
age
allow_cron (bool)
bmic
height
menu_items (list)
multi_user (bool)
name
nav (DANav)
role (str)
role_event
role_needed (list)
speak_text (bool)
surname
track_location (bool)
url_args (dict)
weight
Functions
action_argument(item)
action_arguments()
View documentation


ozgurazad.celik@outlook.com
Body mass index calculator
What is your name?
First Name *
Özgürazad
Name *
Çelik
Continue


ozgurazad.celik@outlook.com
Please write your height!
Please use as unit centimeter!
Height *
171
Continue


ozgurazad.celik@outlook.com
Please write your weight!
Please use as unit kilogram!
Weight *
78
Continue


ozgurazad.celik@outlook.com
Please enter your age?
Age *
26
Continue

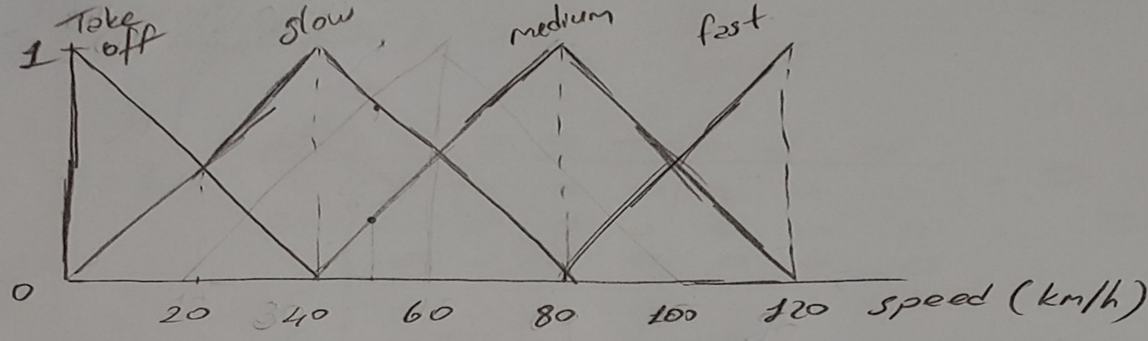

ozgurazad.celik@outlook.com
Hello, Özgürazad Çelik!
Your Body Mass Index is 26.6748743203
The minimum required limit for not being obese is 25
Exit
Restart

2. FUZZY LOGIC PROBLEM WITH HANDWRITE AND MATLAB SOLUTIONS

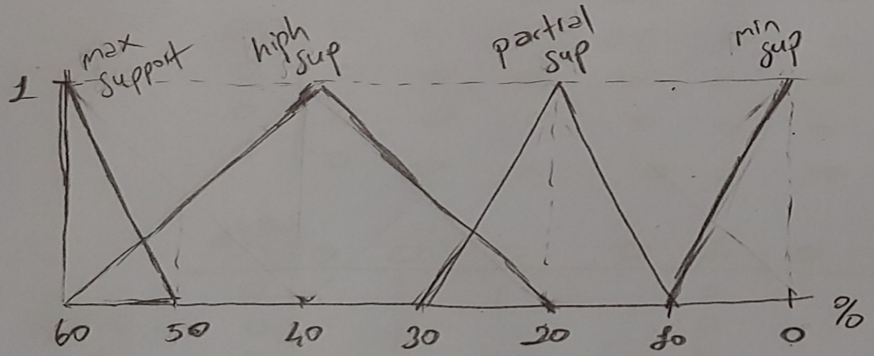
A. PROBLEM

A torque vectoring system has been designed to increase the acceleration of a vehicle. The vehicle's torque vectoring system performs torque vectoring to the front wheels of the vehicle until the vehicle reaches a speed of 120 km/h. The highest support is provided at 60% when the vehicle takes off. Support is cut off when the speed is 120 km/h. In this case, obtain the relationship of acceleration and torque support fuzzy sets. Then find the required percentage of torque support for 55 km/h.

B. HANDWRITE SOLUTION



	0	20	40	60	80	100	120
A_1	1	0.5	0	0	0	0	0
A_2	0	0.5	1	0.5	0	0	0
A_3	0	0	0	0.5	1	0.5	0
A_4	0	0	0	0	0	0.5	1



	60	50	40	30	20	10	0
B_1	1	0	0	0	0	0	0
B_2	0	0.5	1	0.5	0	0	0
B_3	0	0	0	0	1	0	0
B_4	0	0	0	0	0	0	1

$$\underset{\sim}{R} = \sum_{i=1}^{4} (A_i \times B_i)$$

RULES

- If car take off (A_1) THEN Torque vectoring level is max support (B_1)
- If speed is slow (A_2) THEN Torque vectoring level is High support (B_2)
- If speed is medium (A_3) THEN Torque vetoring level is Partial support (B_3)
- If speed is fast (A_4) THEN Torque vectoring level is min support (B_4)

$A_1 \times B_1 =$

$A_1 \backslash B_1$	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0.5	0.5	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0							
0							
0							
0							

$A_2 \times B_2 =$

$A_2 \backslash B_2$	0	0.5	1	0.5	0	0	0
0	0	0	0	0	0	0	0
0.5	0	0.5	0.5	0.5	0	0	0
1	0	0.5	1	0.5	0	0	0
0.5	0	0.5	0.5	0.5	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$A_3 \times B_3$

$A_3 \backslash B_3$	0	0	0	0	1	0	0
0					0		
0					0		
0					0		
0.5					0.5		
1					1		
0.5					0.5		
0					0		

$A_4 \times B_4$

$A_4 \backslash B_4$	0	0	0	0	0	0	1
0							0
0							0
0							0
0							0
0							0
0.5							0.5
1							1

$$\underset{\sim}{R} = (A_1 \times B_1) \cup (A_2 \times B_2) \cup (A_3 \times B_3) \cup (A_4 \cup B_4)$$

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 1 & 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

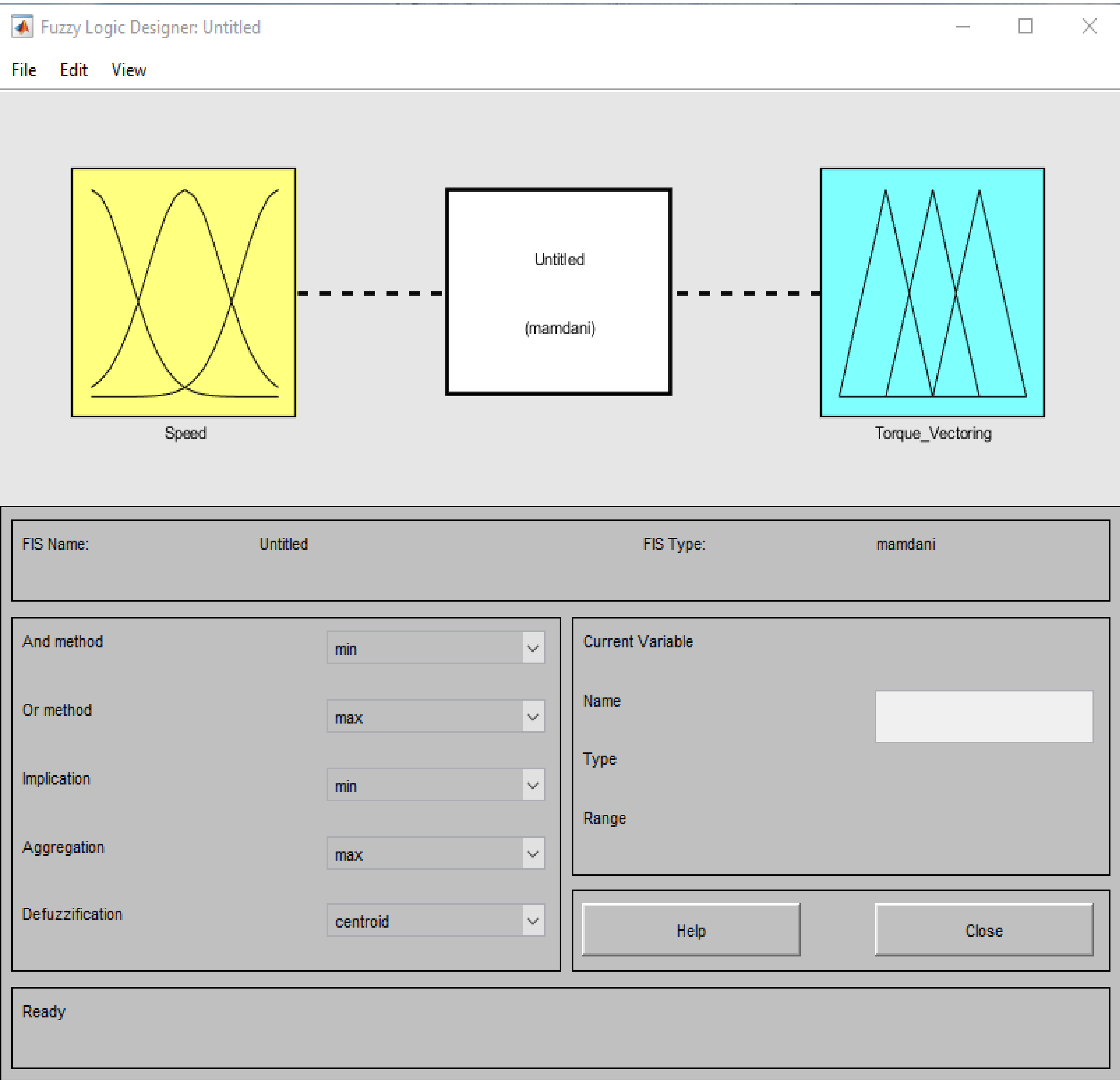
$$A' = [0 \quad 0.125 \quad 0.625 \quad 0.875 \quad 0.375 \quad 0 \quad 0] \cong [0 \quad 0.1 \quad 0.6 \quad 0.9 \quad 0.4 \quad 0 \quad 0]$$

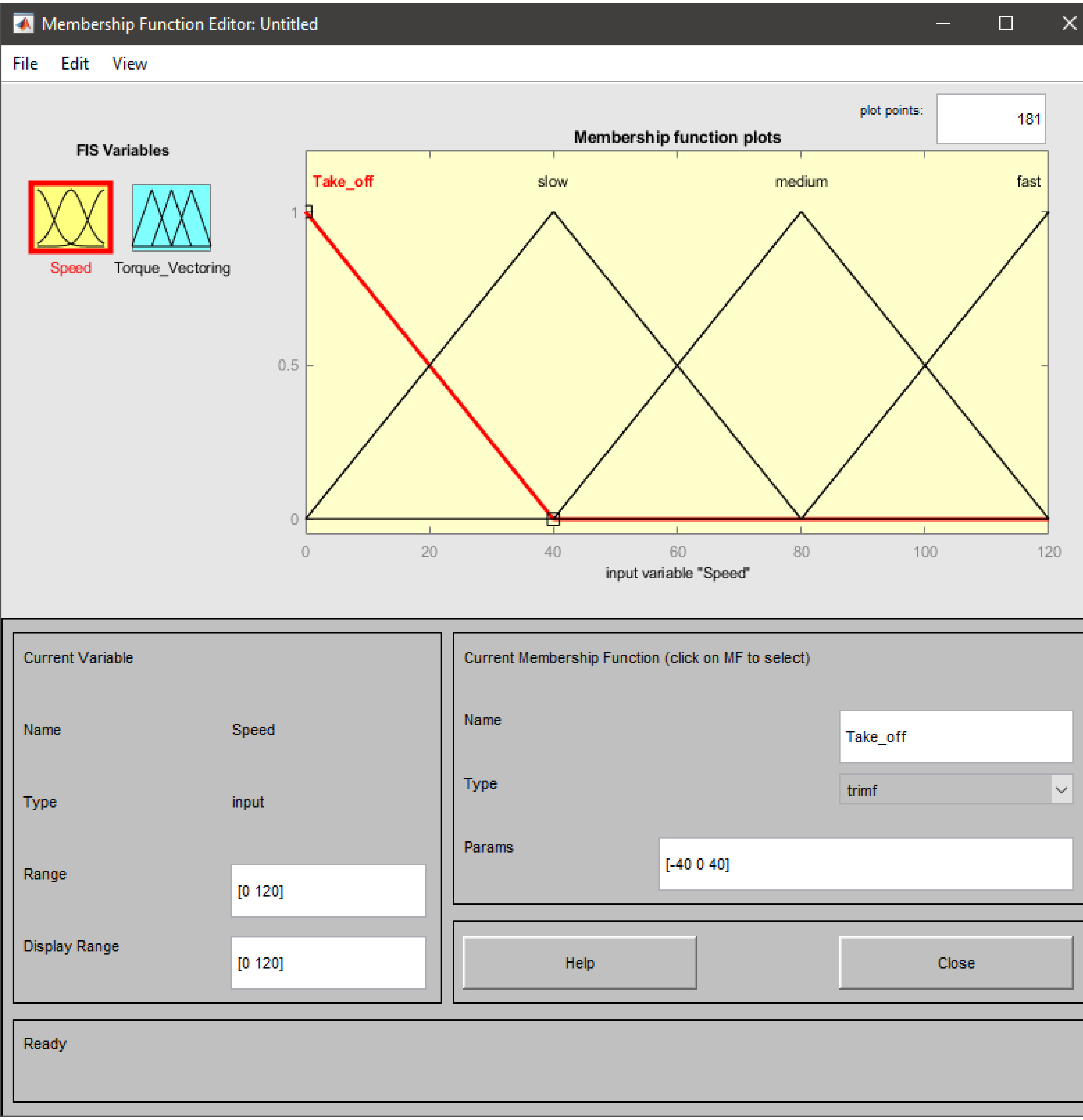
$$B' = A \circ R = [0.125 \quad 0.5 \quad 0.625 \quad 0.5 \quad 0.5 \quad 0 \quad 0]$$

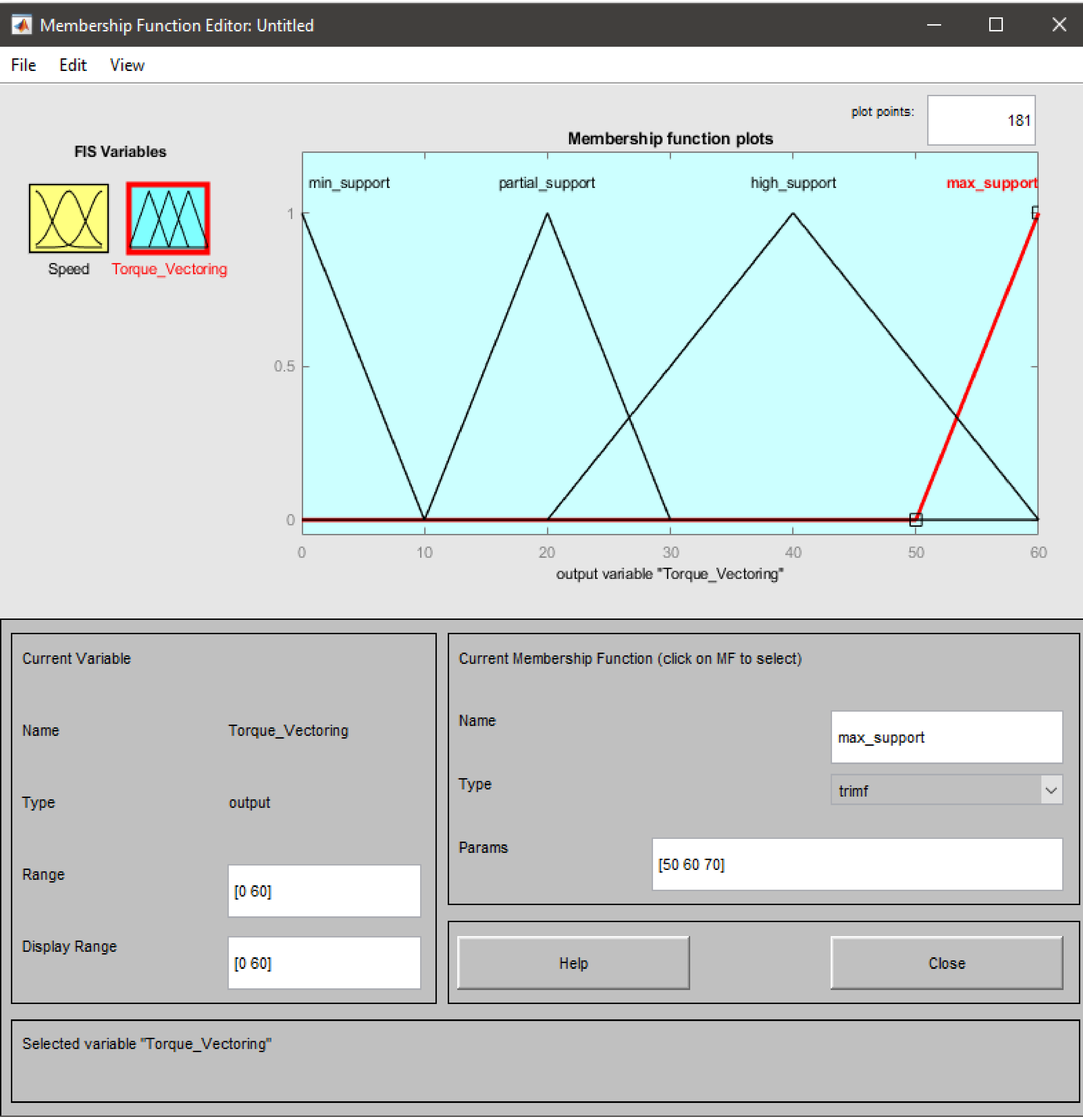
$$\theta = \frac{(0.125 \times 60) + (0.5 \times 50) + (0.625 \times 40) + (0.5 \times 30) + (0.5 \times 20)}{(0.125 + 0.5 + 0.625 + 0.5 + 0.5)} = \frac{82.5}{2.25}$$

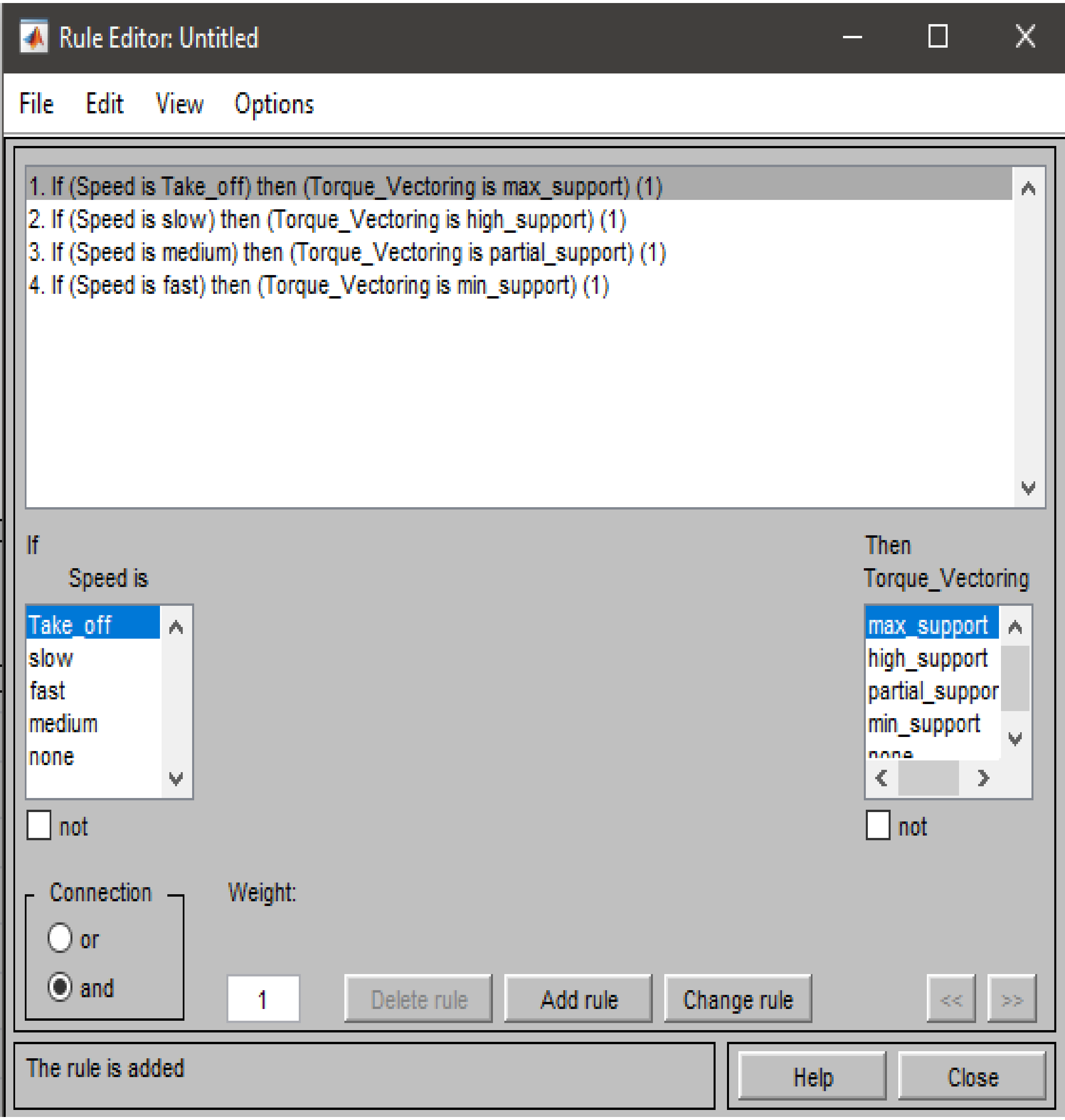
$\theta = 36.67\%$ Torque vectoring

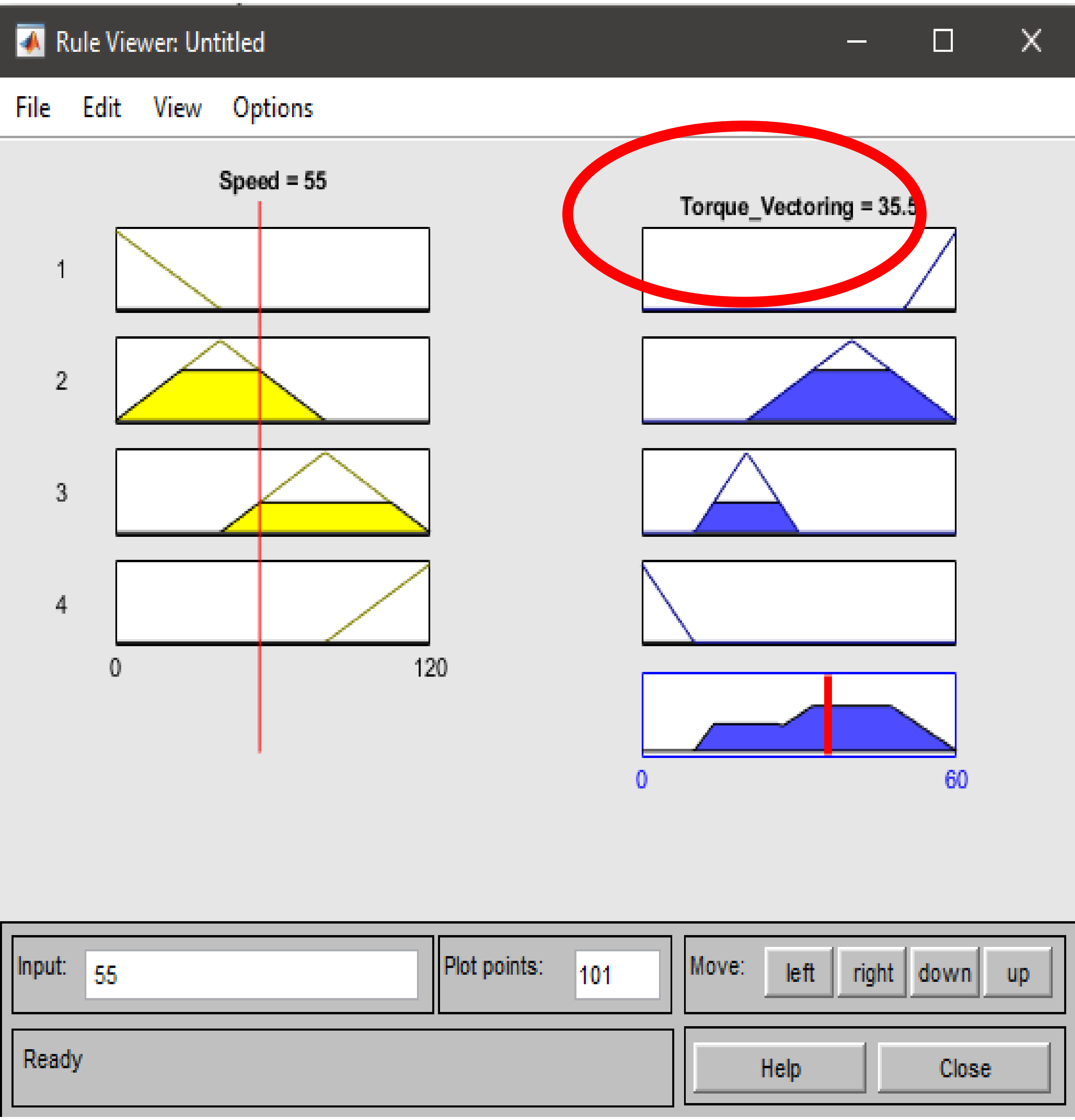
C. MATLAB SOLUTION

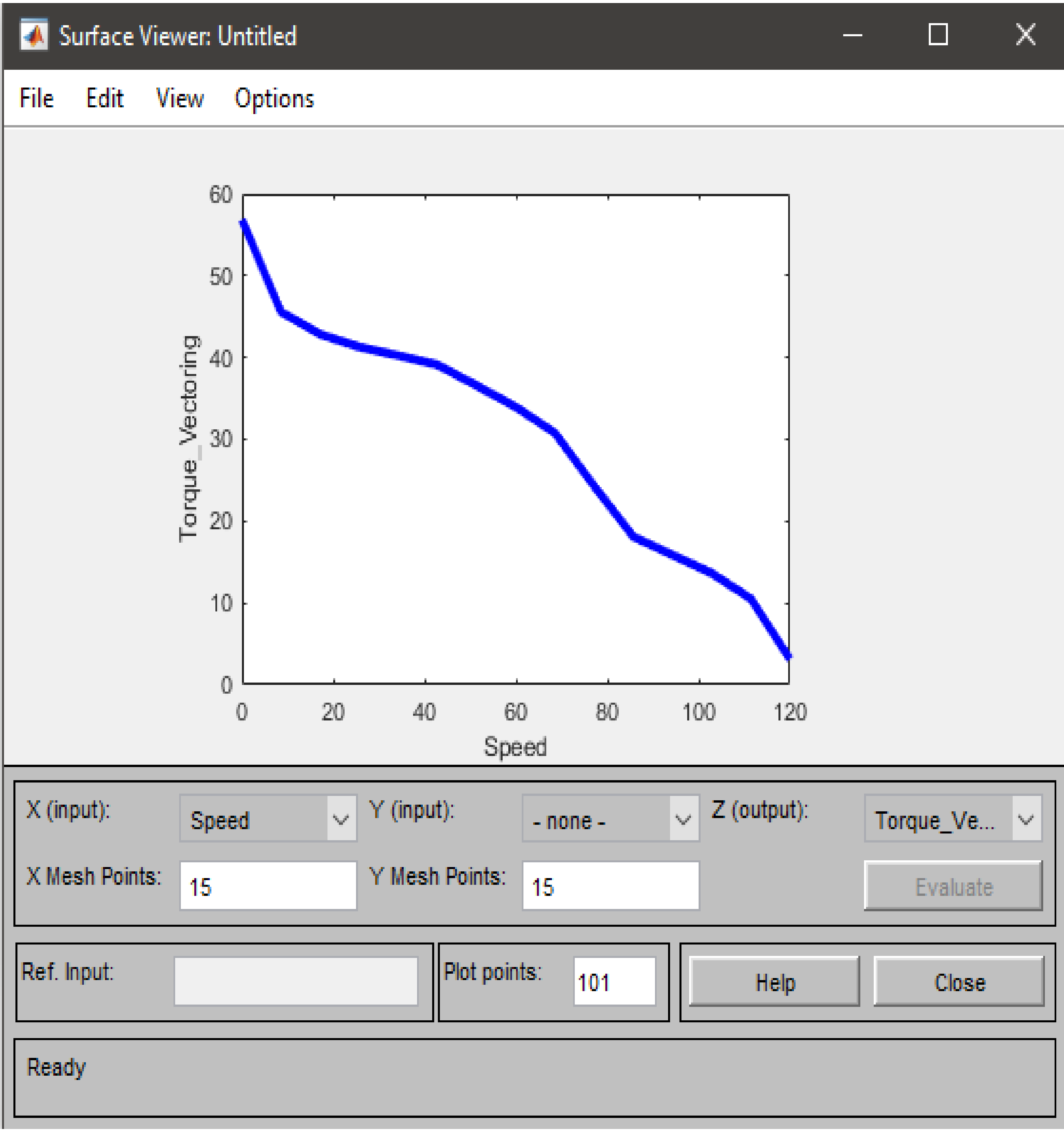



Membership Function Editor: Untitled
File
Edit
View
plot points:
181
Membership function plots
FIS Variables
Speed
Torque_Vectoring
Take_off
slow
medium
fast
1
0.5
0
0
20
40
60
80
100
120
input variable "Speed"
Current Variable
Name
Speed
Type
input
Range
[0 120]
Display Range
[0 120]
Current Membership Function (click on MF to select)
Name
Take_off
Type
trimf
Params
[-40 0 40]
Help
Close
Ready


Membership Function Editor: Untitled
File
Edit
View
plot points:
181
Membership function plots
FIS Variables
Speed
Torque_Vectoring
min_support
partial_support
high_support
max_support
1
0.5
0
0
10
20
30
40
50
60
output variable "Torque_Vectoring"
Current Variable
Name
Torque_Vectoring
Type
output
Range
[0 60]
Display Range
[0 60]
Current Membership Function (click on MF to select)
Name
max_support
Type
trimf
Params
[50 60 70]
Help
Close
Selected variable "Torque_Vectoring"


Rule Editor: Untitled
File Edit View Options
1. If (Speed is Take_off) then (Torque_Vectoring is max_support) (1)
2. If (Speed is slow) then (Torque_Vectoring is high_support) (1)
3. If (Speed is medium) then (Torque_Vectoring is partial_support) (1)
4. If (Speed is fast) then (Torque_Vectoring is min_support) (1)
If
Speed is
Take_off
slow
fast
medium
none
not
Then
Torque_Vectoring
max_support
high_support
partial_suppor
min_support
not
Connection
or
and
Weight:
1
Delete rule
Add rule
Change rule
<<
>>
The rule is added
Help
Close


Rule Viewer: Untitled
File Edit View Options
Speed = 55
Torque_Vectoring = 35.5
1
2
3
4
0
120
0
60
Input:
55
Plot points:
101
Move:
left
right
down
up
Ready
Help
Close


Surface Viewer: Untitled
File
Edit
View
Options
60
50
40
30
20
10
0
Torque_Vectoring
0
20
40
60
80
100
120
Speed
X (input):
Speed
Y (input):
- none -
Z (output):
Torque_Ve...
X Mesh Points:
15
Y Mesh Points:
15
Evaluate
Ref. Input:
Plot points:
101
Help
Close
Ready

3. GENETIC ALGORITHM WITH MATLAB

A. PROBLEM

In order for companies to keep up with the competition, the costs of engineering studies should be minimized with optimization methods. Therefore, part designs should be optimized, focusing on producing lighter, cheaper and efficient parts.

This study focuses on the problem of optimizing the weight of a coil spring. As a method, GENETIC ALGORITHM optimization method in MATLAB program was used.

mass
$m = \frac{1}{4}\pi^2 n d^2 \rho D$

Figure 3-1

The formula in Figure 2-1 is used to calculate the weight of the coil spring. This formula is the objective function for optimization.

$$M = \frac{(n+Q)\pi^2 D d^2 \rho}{4}$$

Figure 3-2 Mass function with Q correction

Parameters	Lower Limit [mm]	Upper Limit [mm]
Spring DİIameter (D)	80	89
Wire Diameter (d)	14	21
Quantity of active Spirals (n)	5	9

Table 3-1 Design variables values of coil springs

Constants	Value
Spring DİIameter (D)	$7800\ kgm^{-3}$
Wire Diameter (d)	$85.2 \times 10^3\ MPa$

Table 3-1 Design parameters of coil springs

Constraints:

$$\tau = \frac{8FD}{\pi d^3}\left(\frac{(4D-d)}{(4D-4d)} + \frac{0.615d}{D}\right)$$

Figure 3-3 Shear stress (τ) in a coil spring

$$\delta = \frac{8FD^3N}{d^4G}$$

Figure 3-4 deflection (δ) of a coil spring

B. CODES

a. Objective Function

```
function y = objective(input)


% X1: Spring Diameter [mm]
% X2: Wire Diameter [mm]
% X3: Quantity of active spirals [integer]
% X4: ρ – mass density [kg/m3]
Q = 4600; % Quantity of inactive turns [N]
X1=input(1);
X2=input(2);
X3=input(3);
X4=input(4);




y = ((X3+Q)*pi^2*X1*X2^2*X4)/4;
end
```

b. Constraint Function

```
function [const, const1] = springconstraint(input)
X1=input(1);
X2=input(2);
X3=input(3);
X4=input(4);

F = 85.2*10^-3; % Load [MPa]
G = 7800; % Modulus of rigidity [kgm^-3]
% X1: Spring Diameter [mm]
% X2: Wire Diameter [mm]
% X3: Quantity of active spirals [integer]
% X4: ρ – mass density [kg/m3]

const= (8*F*X1/(pi*X2^3)*(((4*X1-X2)/(4*X1-4*X2))+(0.615*X2/X1)); %The maximum shear stress tau
const1= ((8*F*X1^2*(X3+2)*((pi*X1)^2+X4^2)^(1/2))/(X2^4*G)); %The maximum deflection (δ) of a coil spring
end
```

C. RESULTS

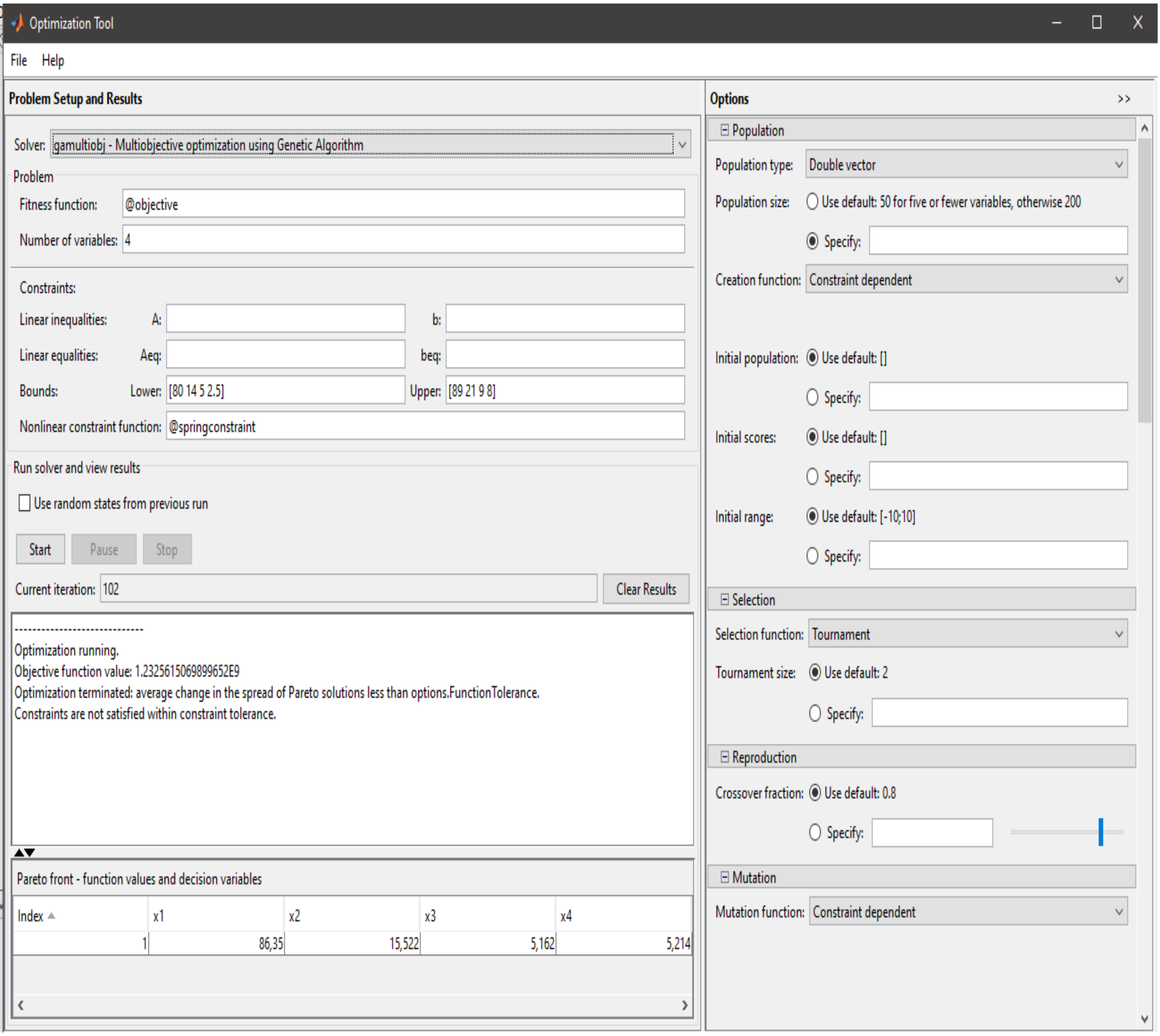


Figure 3-1 Results 1

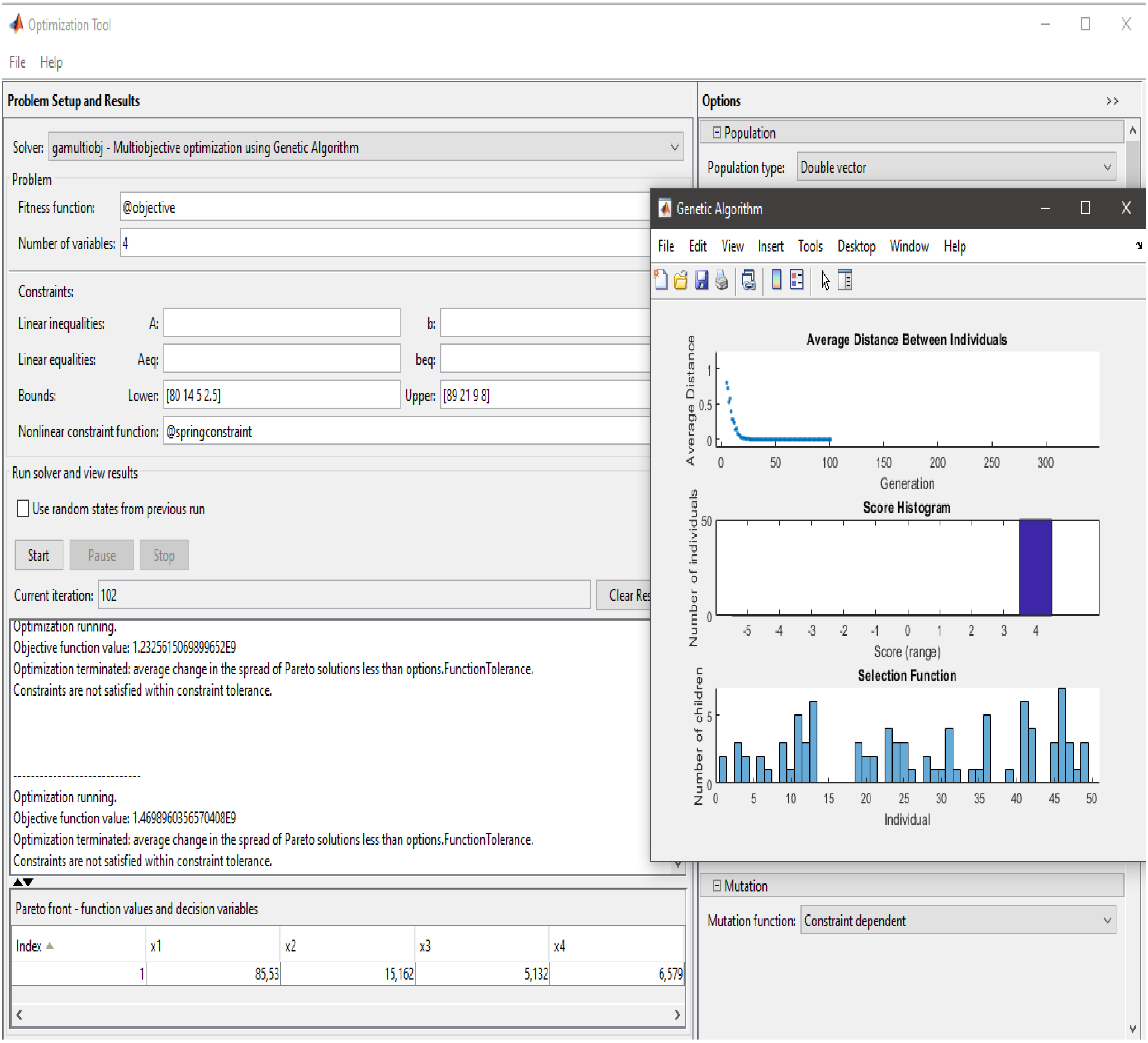


Figure 3-2 Results 2

It can be seen X input values and result value of objection function from Results figures.

4. ANN AND DEEP LEARNING PROBLEMS WITH MATLAB

A. AN ARTIFICIAL NEURAL NETWORK PROBLEM

In 1936, a scientist found 50 flowers of three types of Iris flower (setosa, versicolor, virginica), a total of 150 flowers, and measured the upper and lower flower petals of all. From this measurement, four qualities [sepal-length (lower leaf length cm), sepal-with (lower leaf width cm), pedal-length (upper leaf width cm), pedal-width (upper leaf length cm)] and 150-element data set has been obtained.

In this study, the Iris dataset was used and it was tried to find (classify) which of the three types of the flower it belongs to by using the upper and lower leaf widths and lengths of the iris flower.

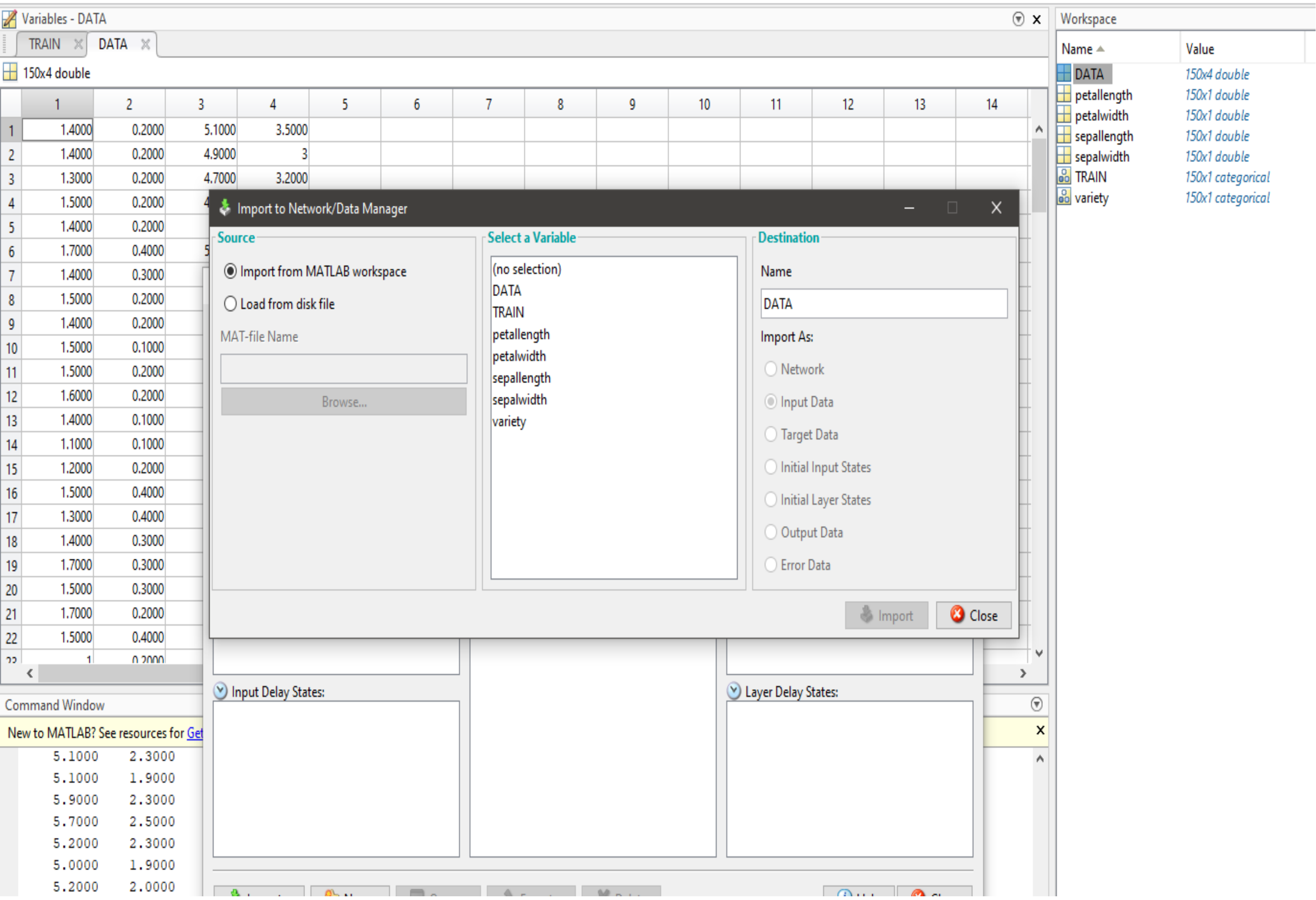
1	sepal_length	sepal_width	petal_length	petal_width	species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3.0	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5.0	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5.0	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa
11	4.9	3.1	1.5	0.1	setosa
12	5.4	3.7	1.5	0.2	setosa
13	4.8	3.4	1.6	0.2	setosa
14	4.8	3.0	1.4	0.1	setosa
15	4.3	3.0	1.1	0.1	setosa
16	5.8	4.0	1.2	0.2	setosa
17	5.7	4.4	1.5	0.4	setosa
18	5.4	3.9	1.3	0.4	setosa
19	5.1	3.5	1.4	0.3	setosa
20	5.7	3.8	1.7	0.3	setosa

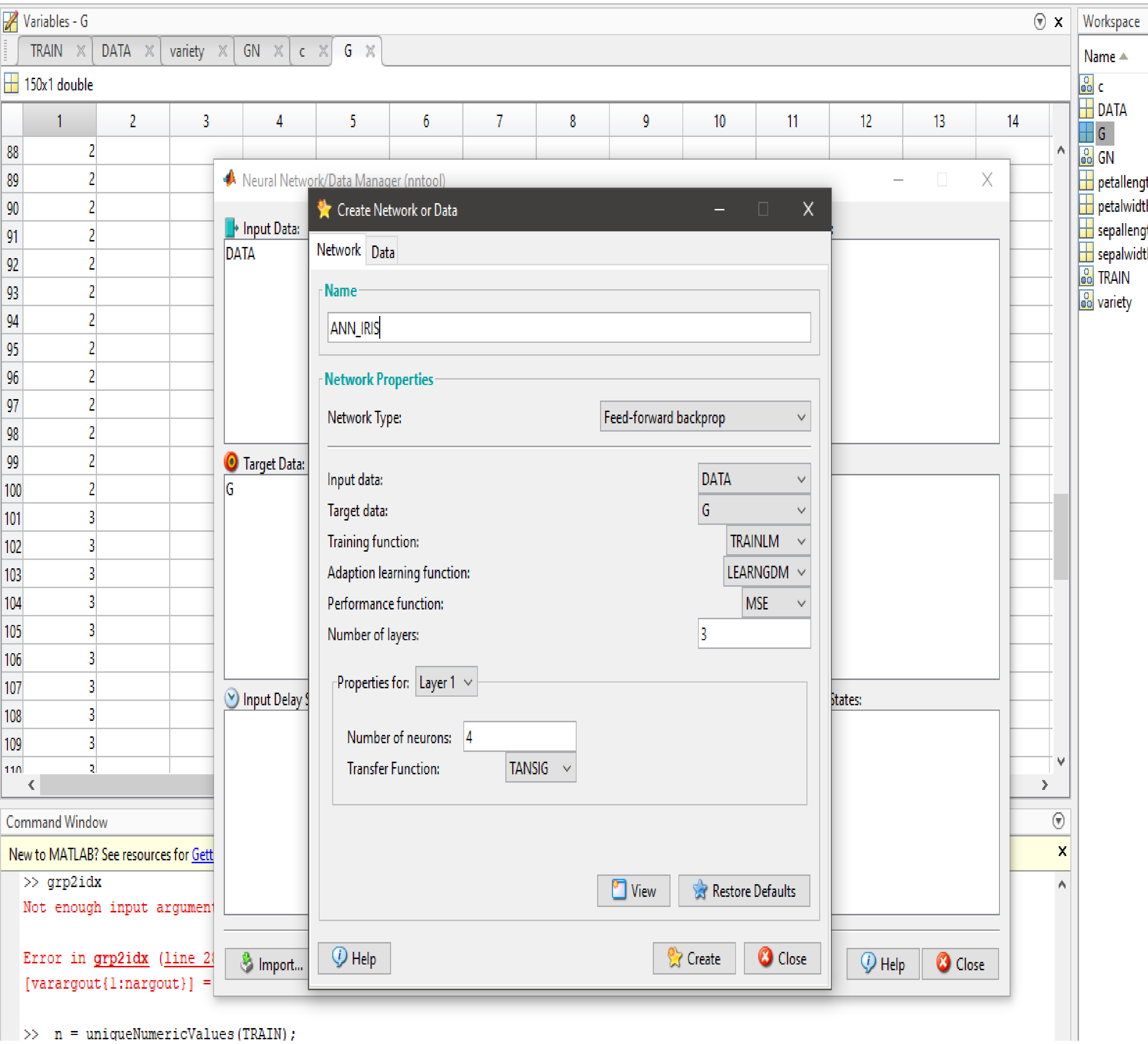
Table 4-1 IRIS DATASET

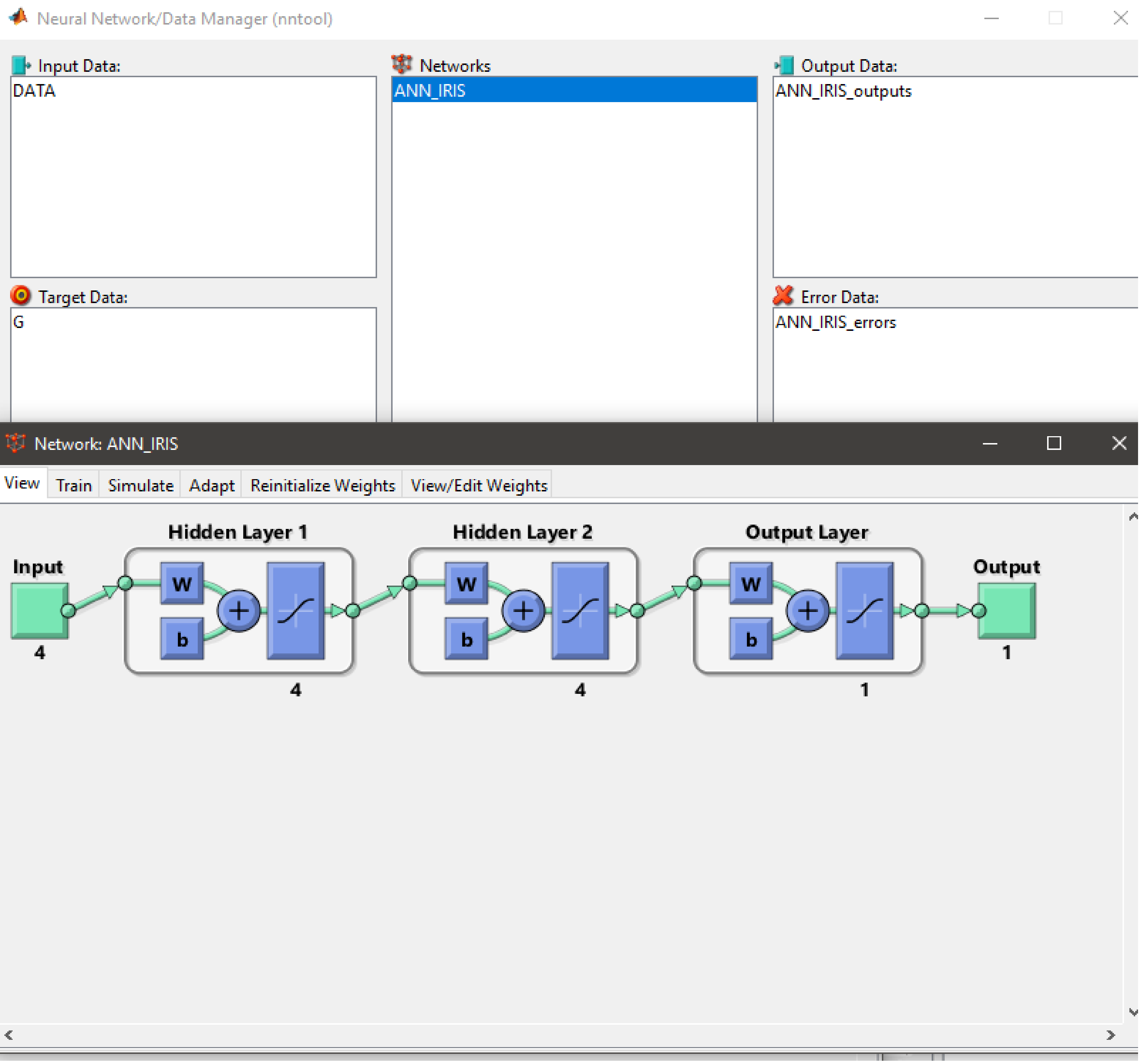
a. CODES

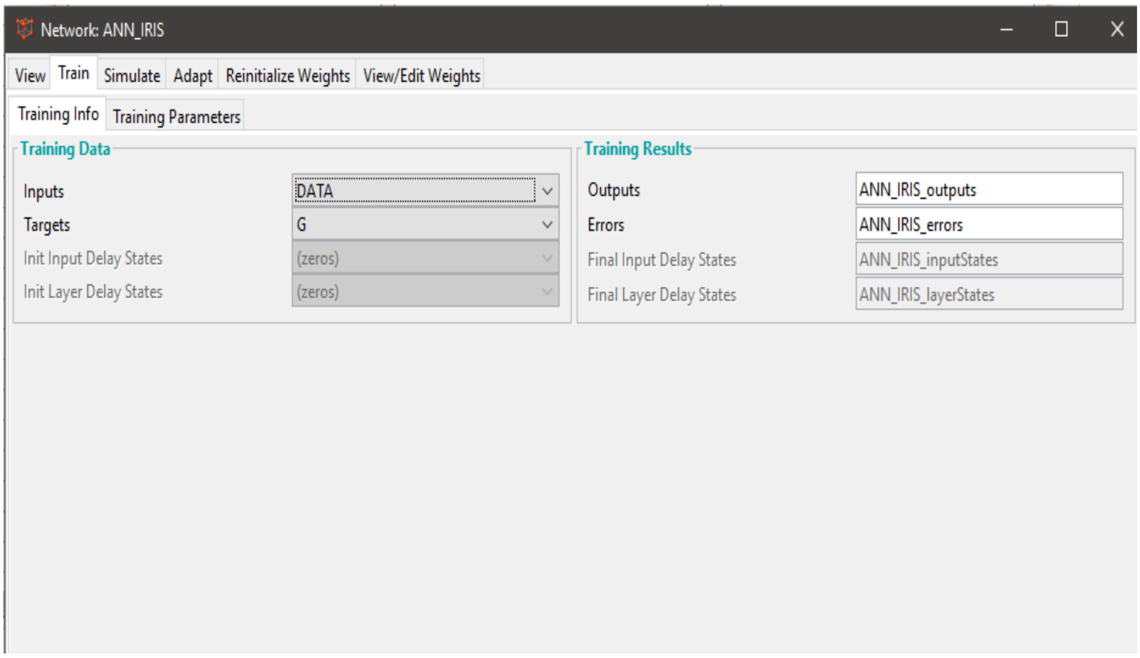
```
>> DATA = [petallength, petalwidth, sepallength, sepalwidth];
>> c = categorical(TRAIN);
[GN, ~, G] = unique(c);
>> nntool
```

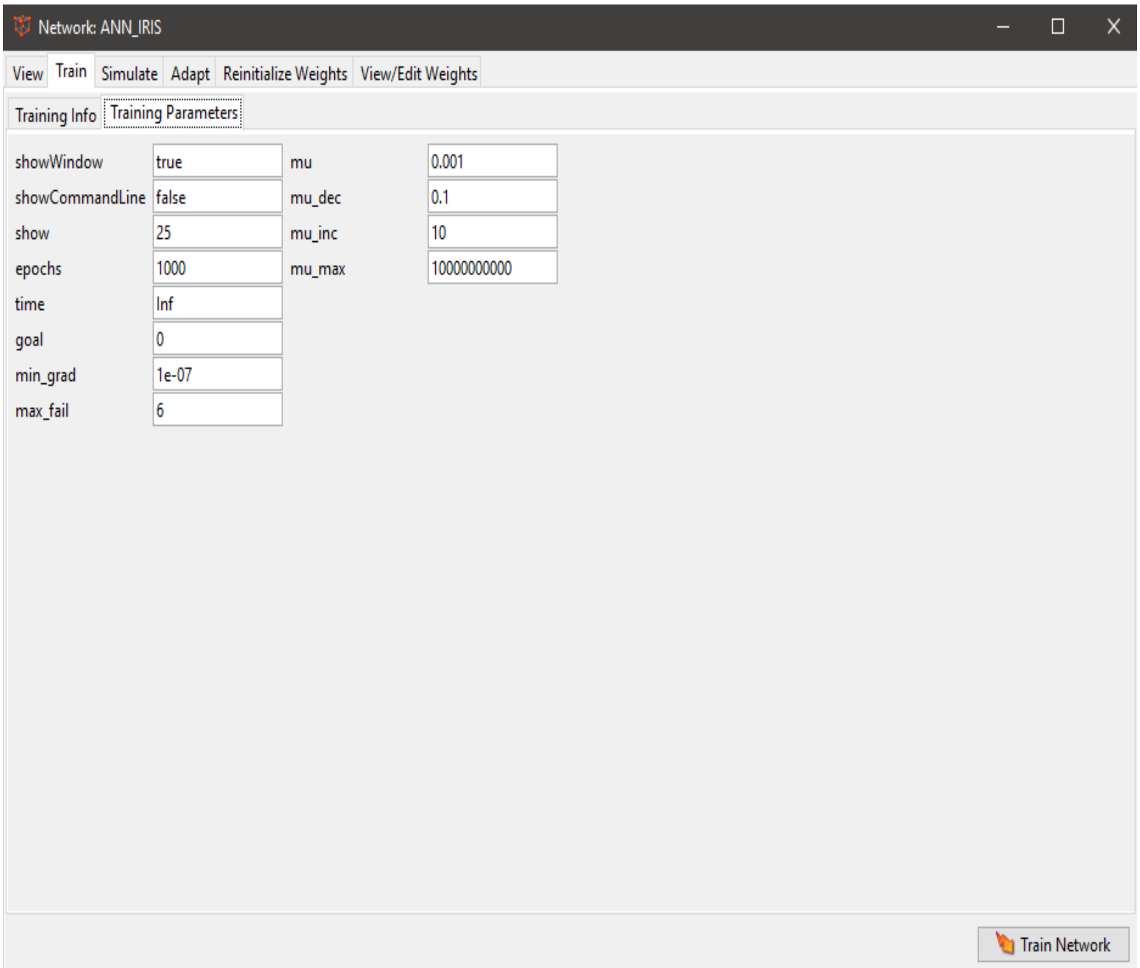
b. RESULTS

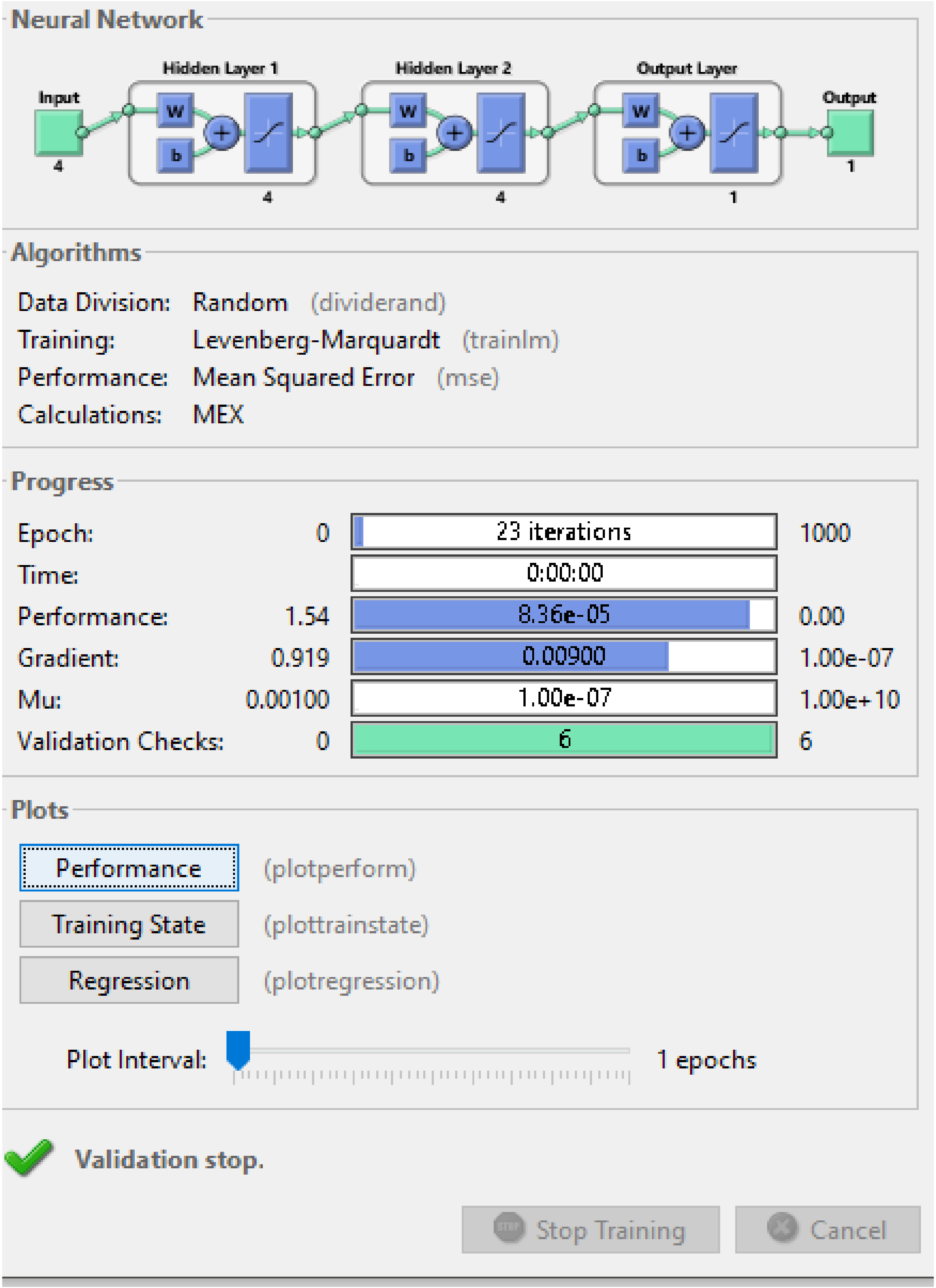


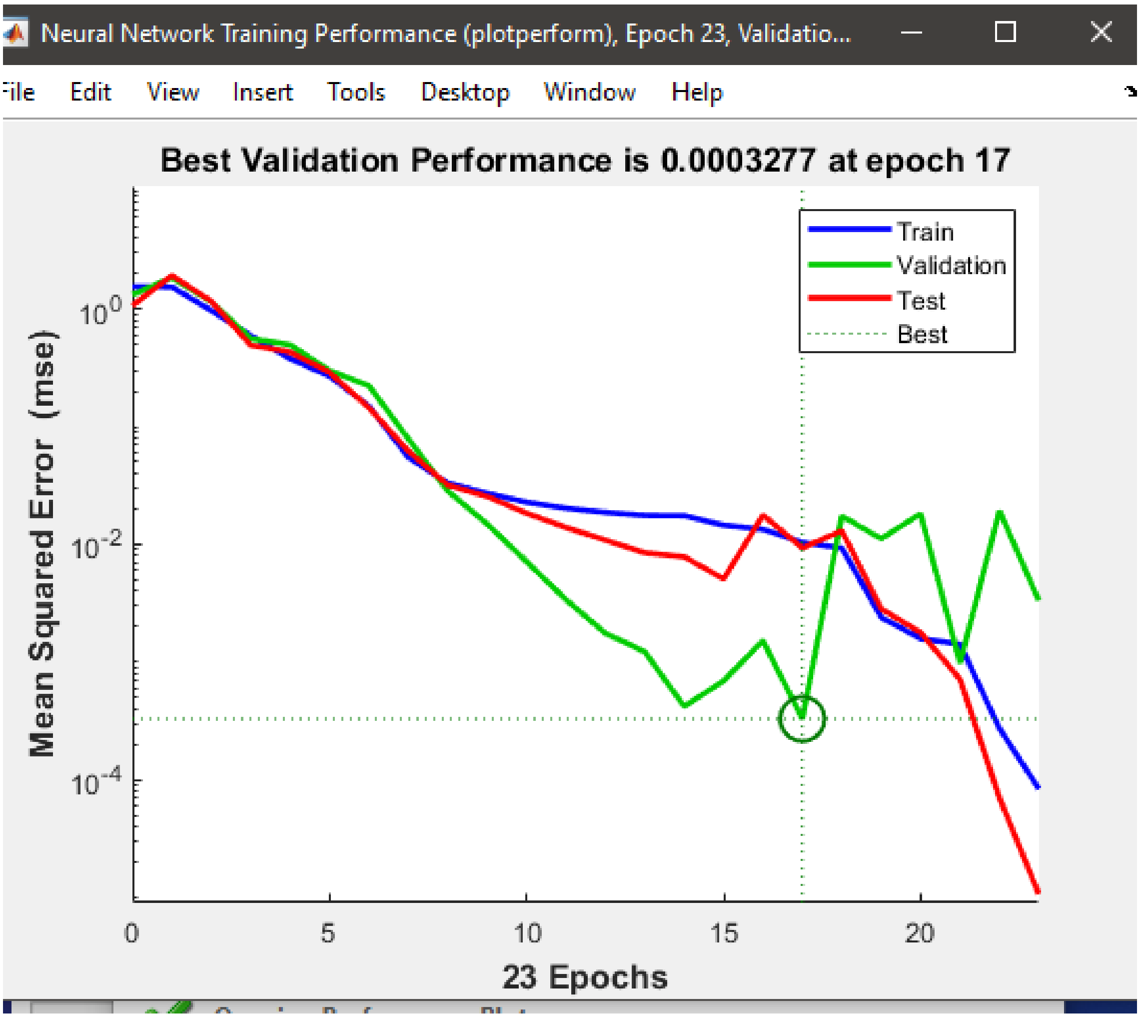

Variables - G
TRAIN DATA variety GN c G
150x1 double
Workspace
Name
c
DATA
G
GN
TRAIN
variety
Neural Network/Data Manager (nntool)
Input Data:
DATA
Target Data:
G
Create Network or Data
Network
Data
Name
ANN_IRIS
Network Properties
Network Type:
Feed-forward backprop
Input data:
DATA
Target data:
G
Training function:
TRAINLM
Adaption learning function:
LEARNGDM
Performance function:
MSE
Number of layers:
3
Properties for:
Layer 1
Number of neurons:
4
Transfer Function:
TANSIG
View
Restore Defaults
Help
Create
Close
Import...
Help
Close
Command Window
>> grp2idx
>> n = uniqueNumericValues(TRAIN);

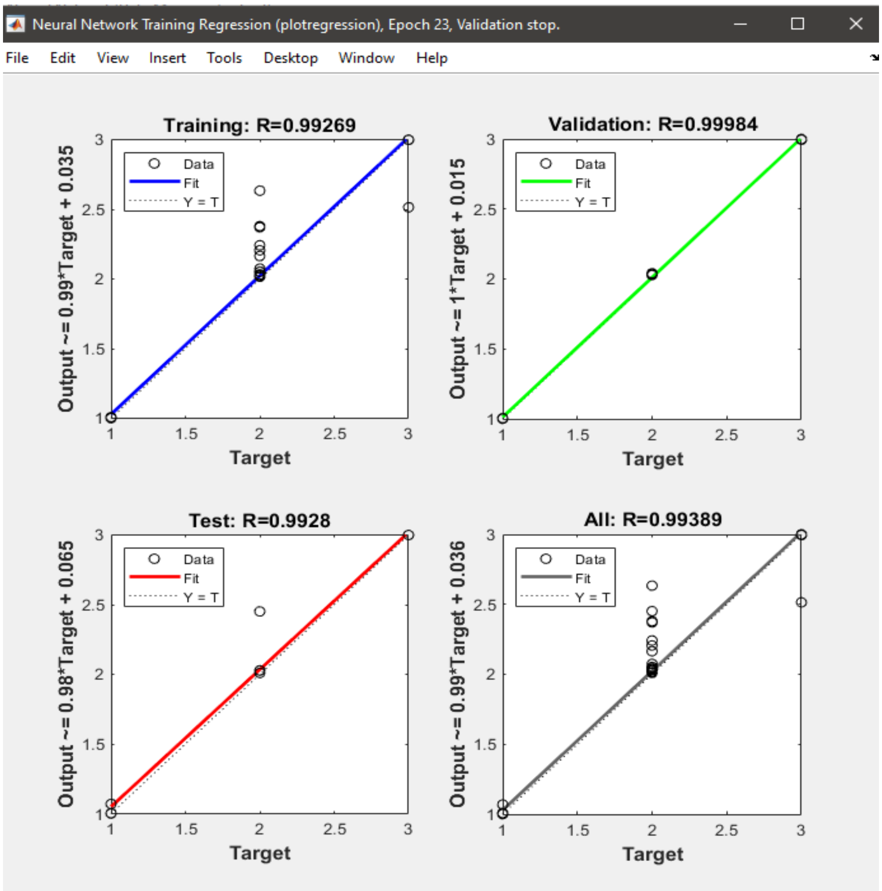

Neural Network/Data Manager (nntool)
Input Data:
DATA
Networks
ANN_IRIS
Output Data:
ANN_IRIS_outputs
Target Data:
G
Error Data:
ANN_IRIS_errors
Network: ANN_IRIS
View
Train
Simulate
Adapt
Reinitialize Weights
View/Edit Weights
Hidden Layer 1
Hidden Layer 2
Output Layer
Input
Output
W
b
W
b
W
b
4
4
4
1
1

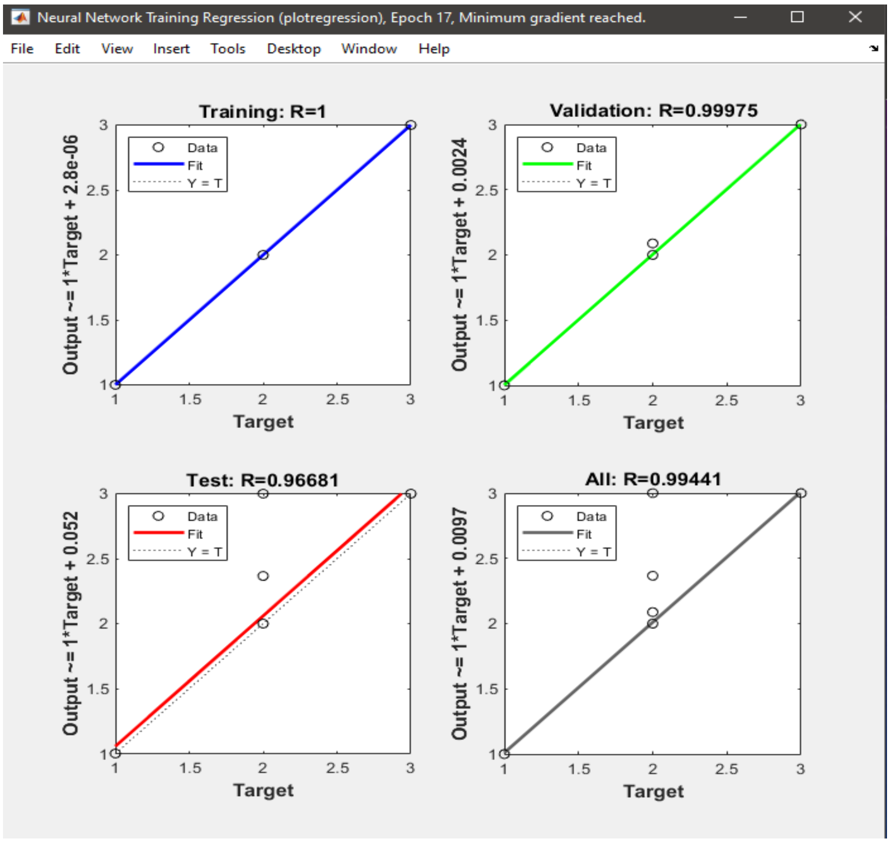

Network: ANN_IRIS
View
Train
Simulate
Adapt
Reinitialize Weights
View/Edit Weights
Training Info
Training Parameters
Training Data
Inputs
DATA
Targets
G
Init Input Delay States
(zeros)
Init Layer Delay States
(zeros)
Training Results
Outputs
ANN_IRIS_outputs
Errors
ANN_IRIS_errors
Final Input Delay States
ANN_IRIS_inputStates
Final Layer Delay States
ANN_IRIS_layerStates

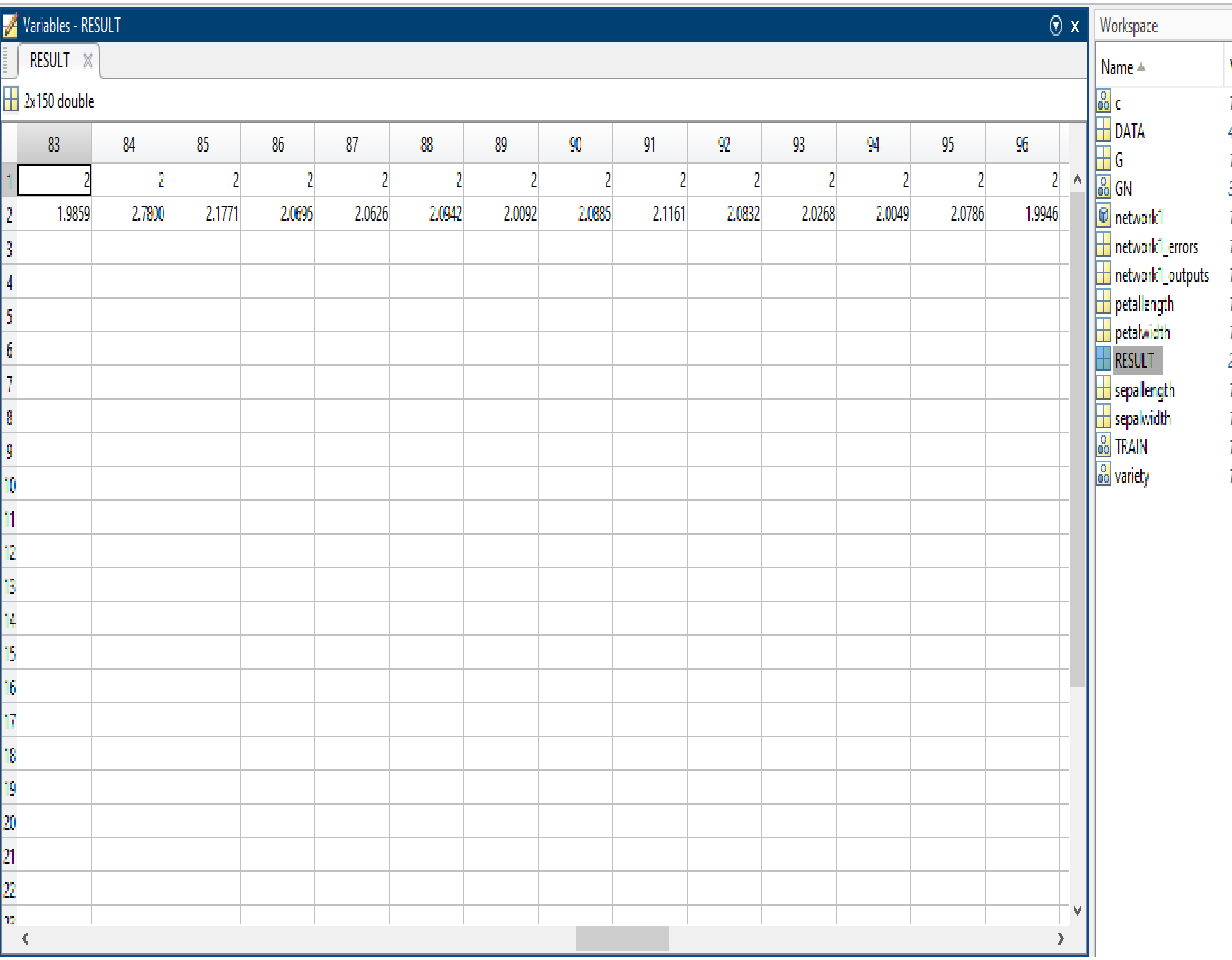

Network: ANN_IRIS
View
Train
Simulate
Adapt
Reinitialize Weights
View/Edit Weights
Training Info
Training Parameters
showWindow
true
showCommandLine
false
show
25
epochs
1000
time
Inf
goal
0
min_grad
1e-07
max_fail
6
mu
0.001
mu_dec
0.1
mu_inc
10
mu_max
10000000000
Train Network


Neural Network
Hidden Layer 1
Hidden Layer 2
Output Layer
Input
Output
W
b
4
4
4
1
1
Algorithms
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX
Progress
Epoch: 0 23 iterations 1000
Time: 0:00:00
Performance: 1.54 8.36e-05 0.00
Gradient: 0.919 0.00900 1.00e-07
Mu: 0.00100 1.00e-07 1.00e+10
Validation Checks: 0 6 6
Plots
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)
Plot Interval: 1 epochs
Validation stop.
Stop Training
Cancel


Neural Network Training Performance (plotperform), Epoch 23, Validatio...
File Edit View Insert Tools Desktop Window Help
Best Validation Performance is 0.0003277 at epoch 17
Train
Validation
Test
Best
Mean Squared Error (mse)
10^0
10^-2
10^-4
0
5
10
15
20
23 Epochs


Neural Network Training Regression (plotregression), Epoch 23, Validation stop.
File Edit View Insert Tools Desktop Window Help
Training: R=0.99269
Output ~= 0.99*Target + 0.035
Validation: R=0.99984
Output ~= 1*Target + 0.015
Test: R=0.9928
Output ~= 0.98*Target + 0.065
All: R=0.99389
Output ~= 0.99*Target + 0.036
Data
Fit
Y = T
Target


Neural Network Training Regression (plotregression), Epoch 17, Minimum gradient reached.
File Edit View Insert Tools Desktop Window Help
Training: R=1
Output ~= 1*Target + 2.8e-06
Validation: R=0.99975
Output ~= 1*Target + 0.0024
Test: R=0.96681
Output ~= 1*Target + 0.052
All: R=0.99441
Output ~= 1*Target + 0.0097
Data
Fit
Y = T
Target



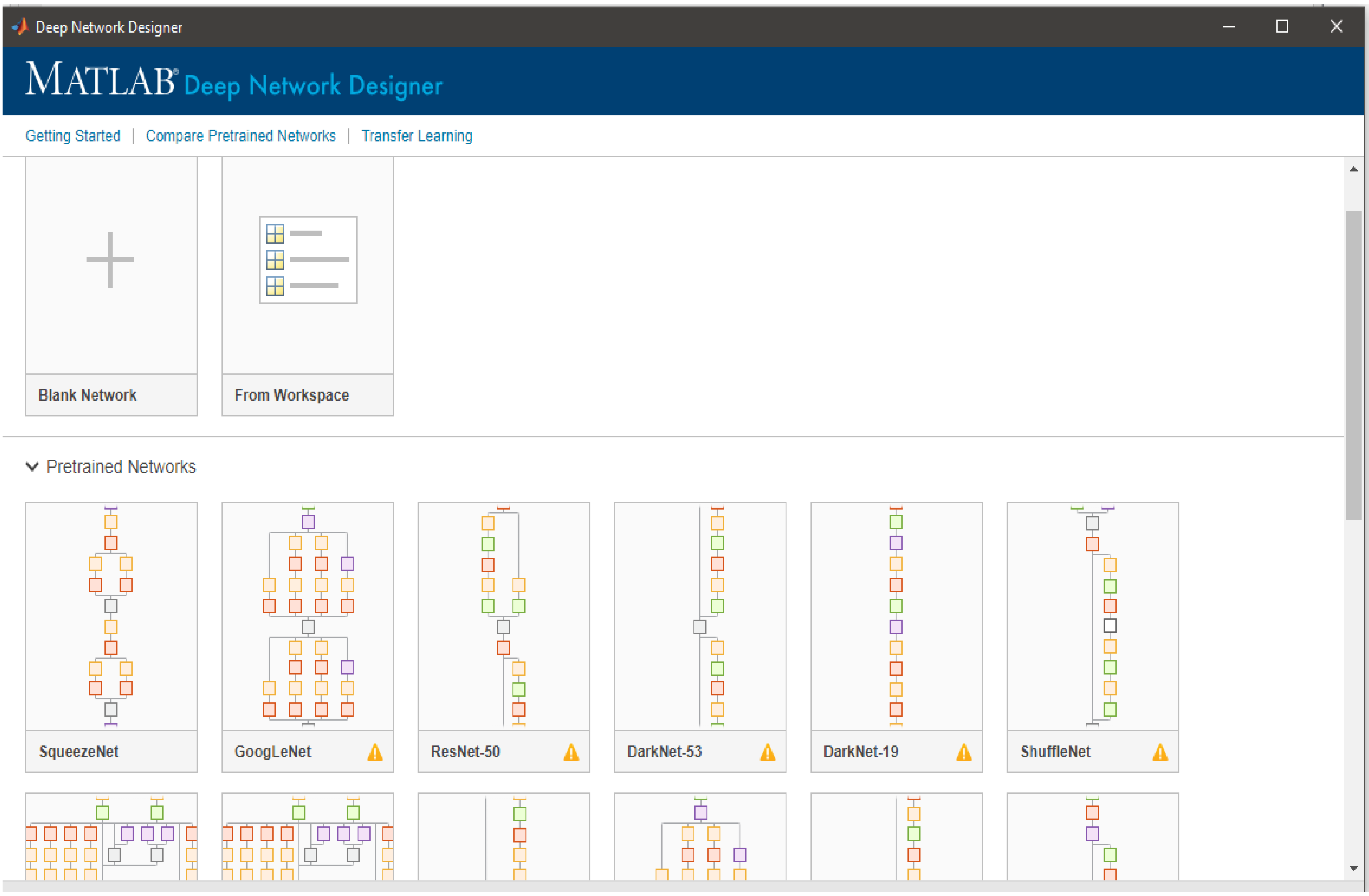
The training was completed as a result of the best results from the 17th of 23 iterations. As a result, when Iris data was tested, it was seen that all flowers were placed in the correct class with great accuracy.

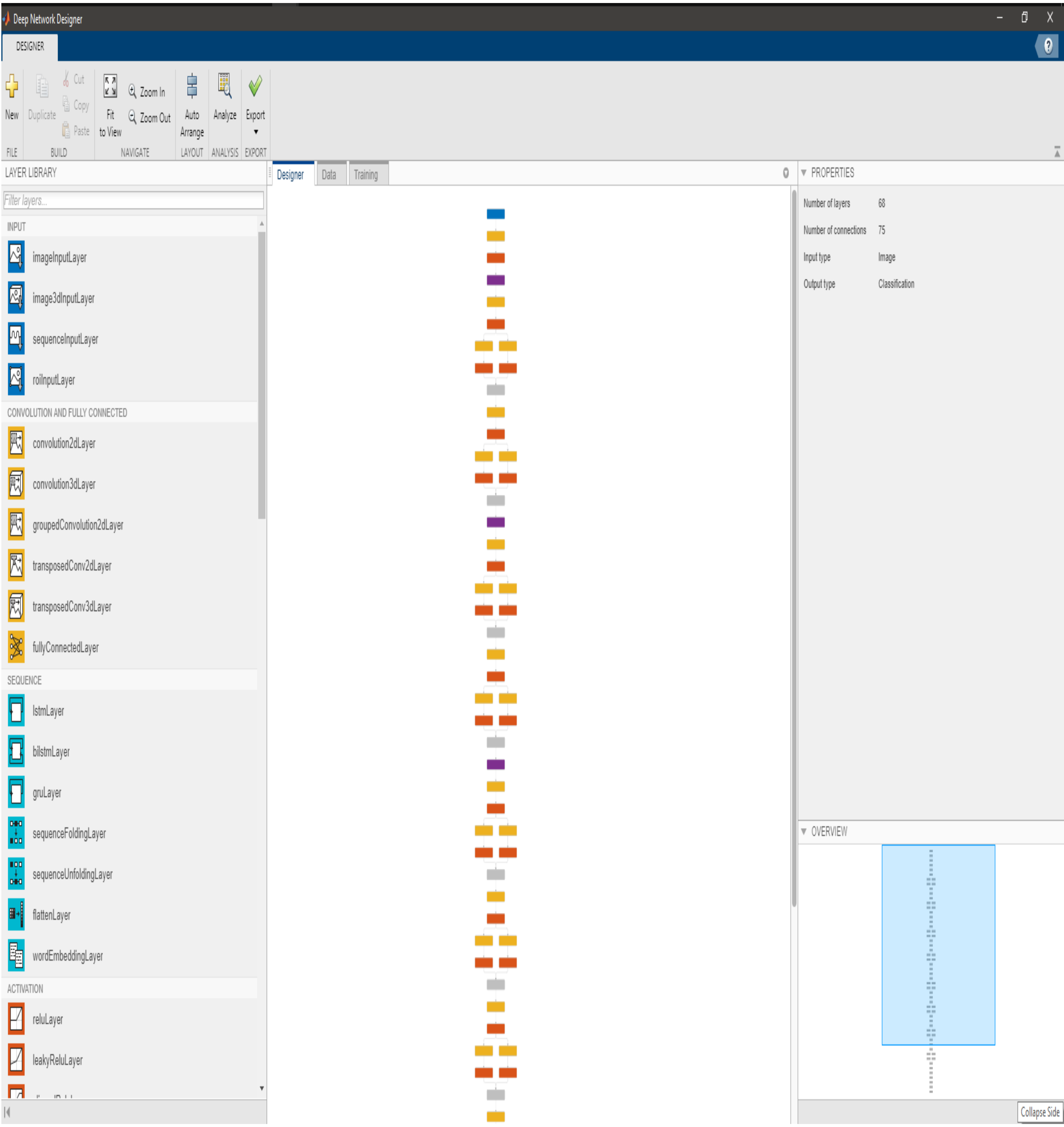
B. AN DEEP LEARNING STUDY WITH MATLAB SQUEEZENET

a. PROBLEM

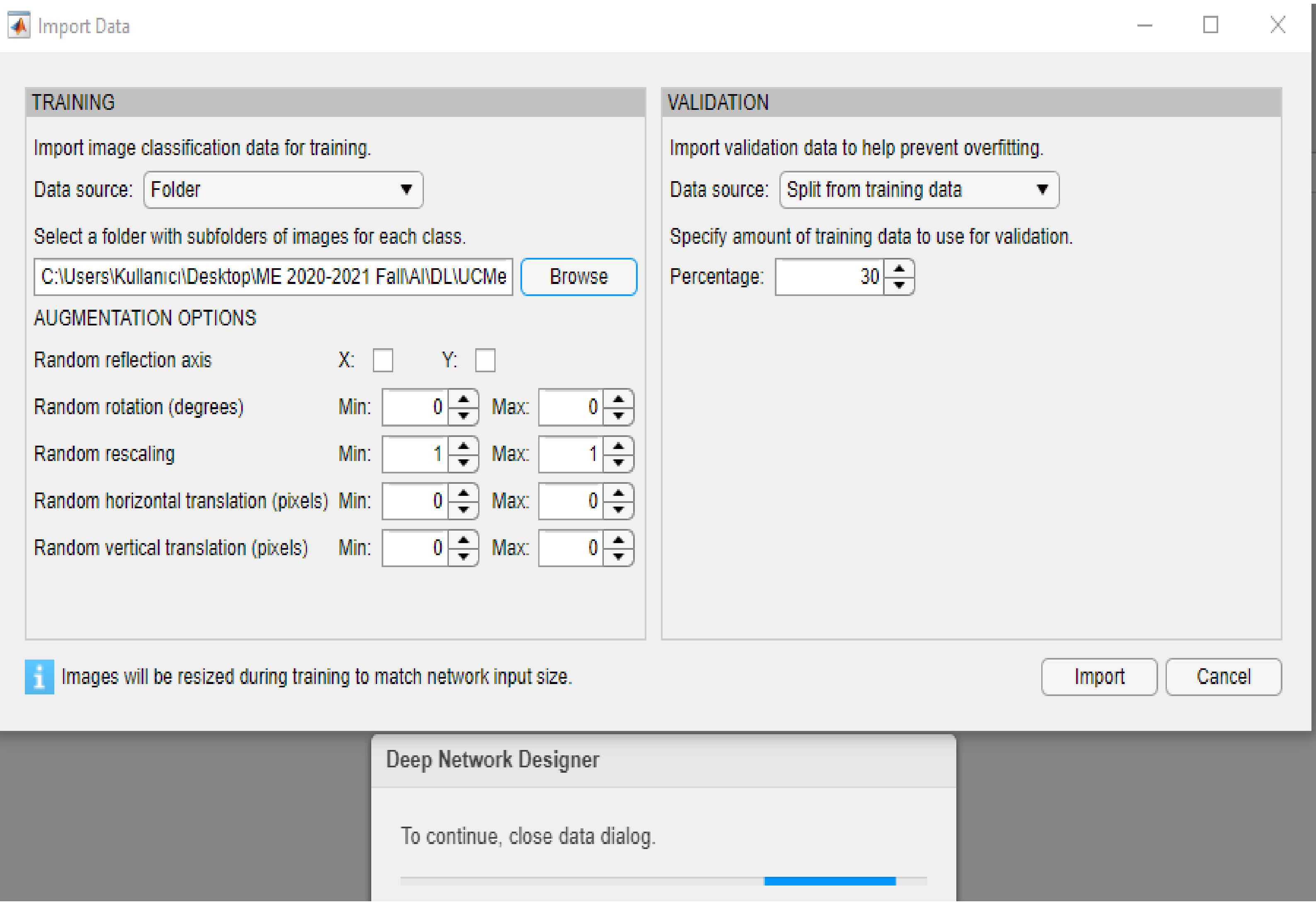
In this study, photographs taken by satellites belonging to 21 different classes were trained with deep learning using Matlab Squeezenet and it was tried to find which class the photographs belong to.

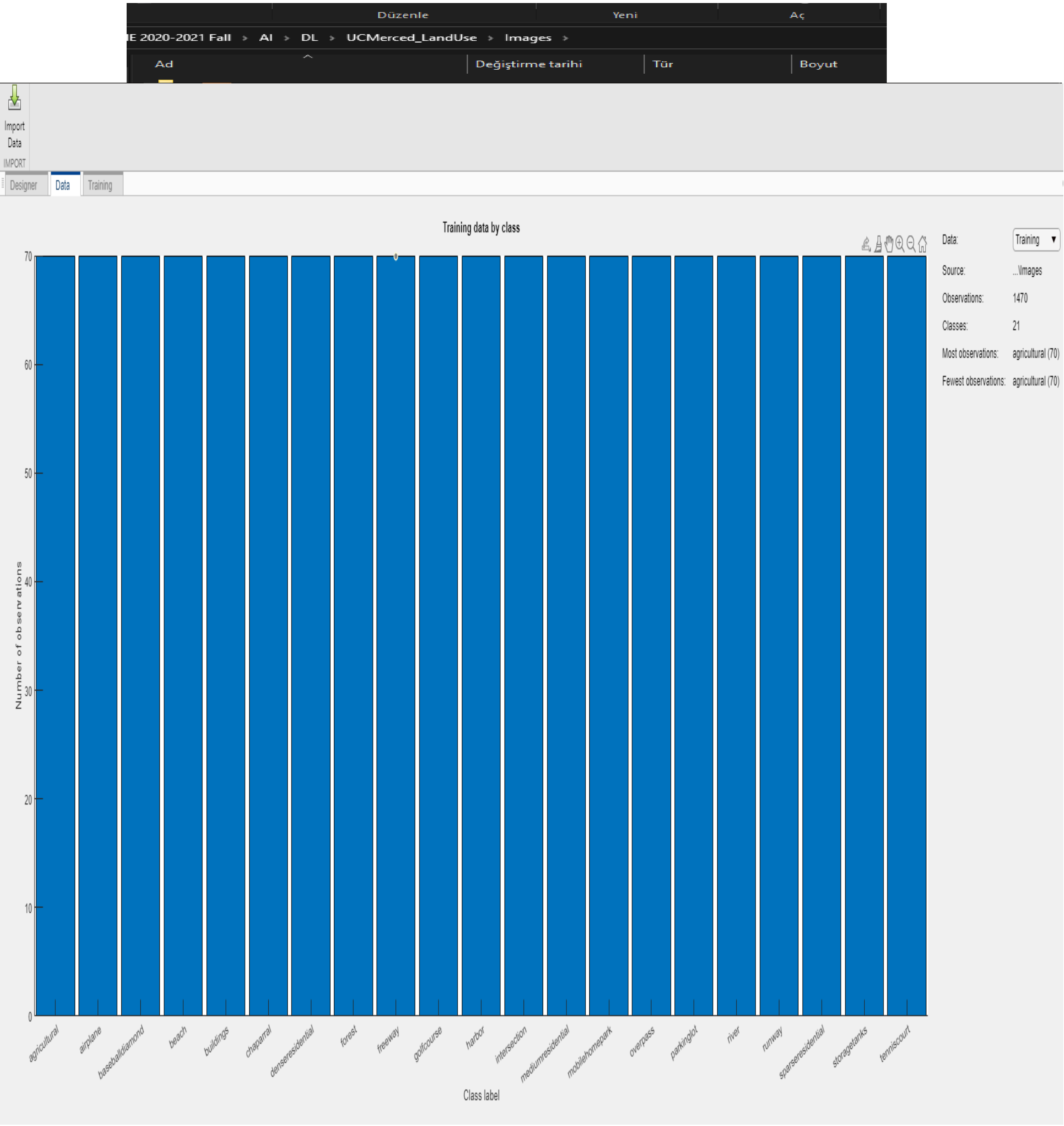
b. PREPARATIONS AND INSTALLATION



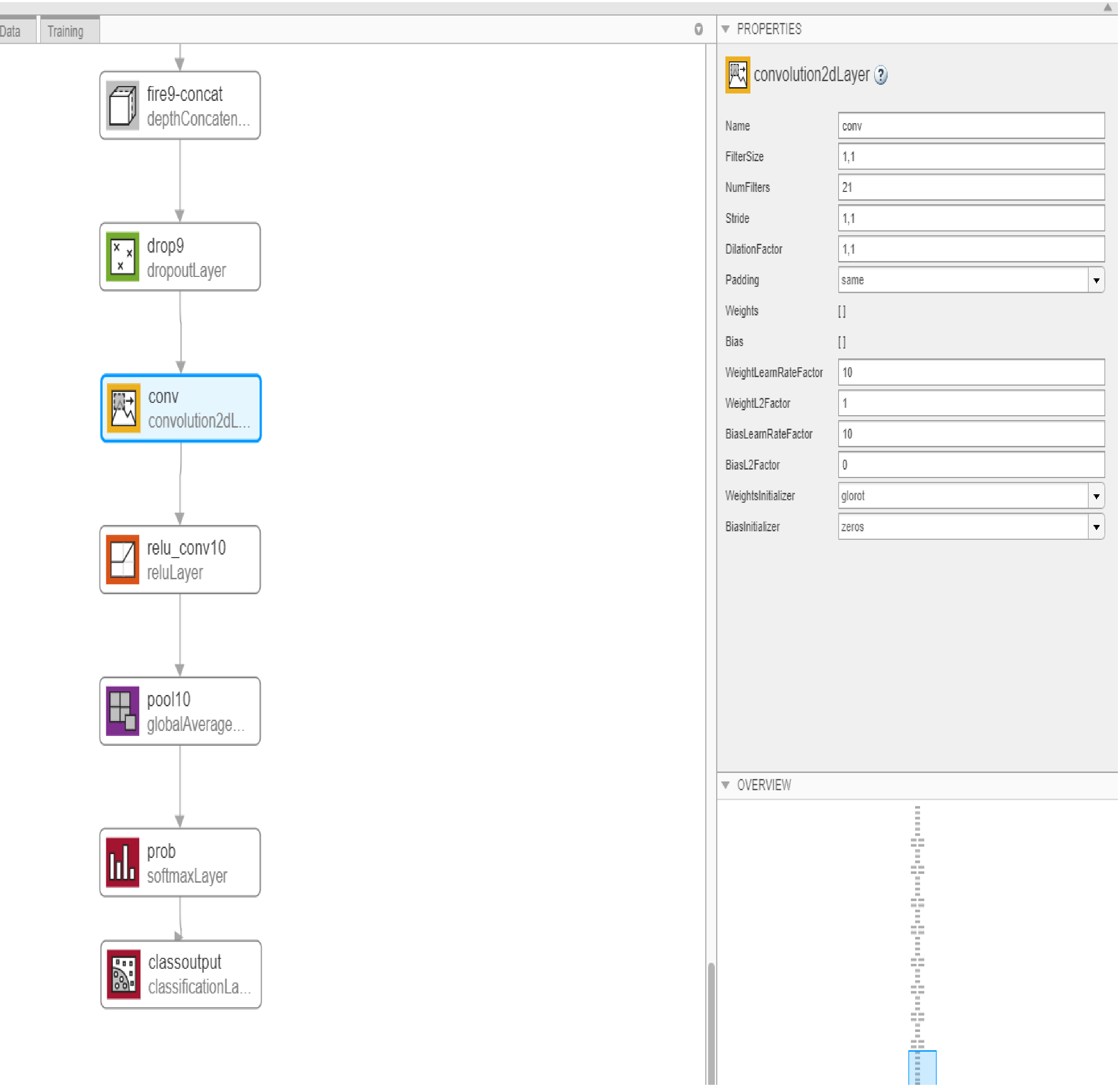

Deep Network Designer
DESIGNER
New
Duplicate
Cut
Copy
Paste
Fit to View
Zoom In
Zoom Out
Auto Arrange
Analyze
Export
FILE
BUILD
NAVIGATE
LAYOUT
ANALYSIS
EXPORT
LAYER LIBRARY
Filter layers...
INPUT
imageInputLayer
image3dInputLayer
sequenceInputLayer
roiInputLayer
CONVOLUTION AND FULLY CONNECTED
convolution2dLayer
convolution3dLayer
groupedConvolution2dLayer
transposedConv2dLayer
transposedConv3dLayer
fullyConnectedLayer
SEQUENCE
lstmLayer
bilstmLayer
gruLayer
sequenceFoldingLayer
sequenceUnfoldingLayer
flattenLayer
wordEmbeddingLayer
ACTIVATION
reluLayer
leakyReluLayer
Designer
Data
Training
PROPERTIES
Number of layers 68
Number of connections 75
Input type Image
Output type Classification
OVERVIEW
Collapse Side

An prepared Image Dateset which contain so many images download from internet. Than added to the MATLAB study field.



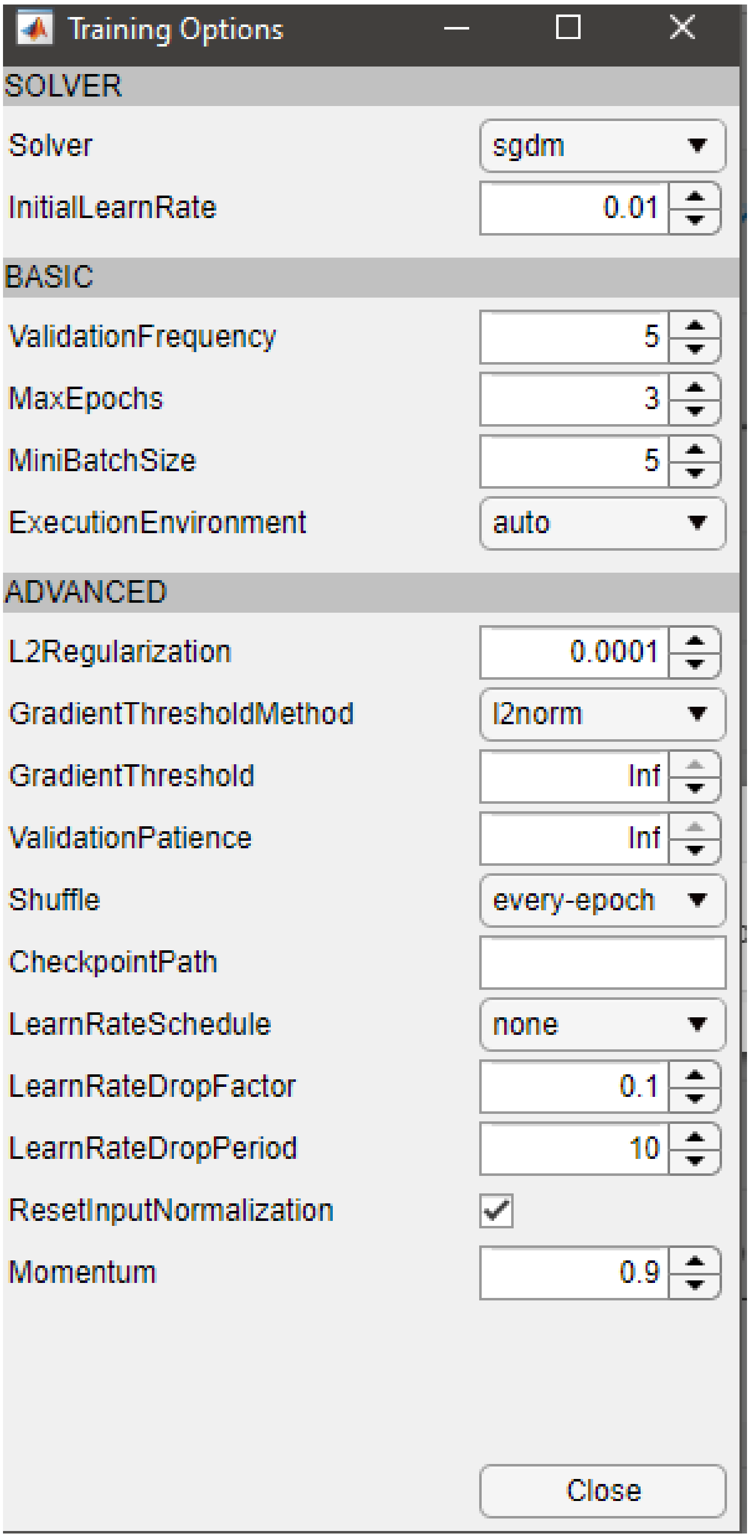


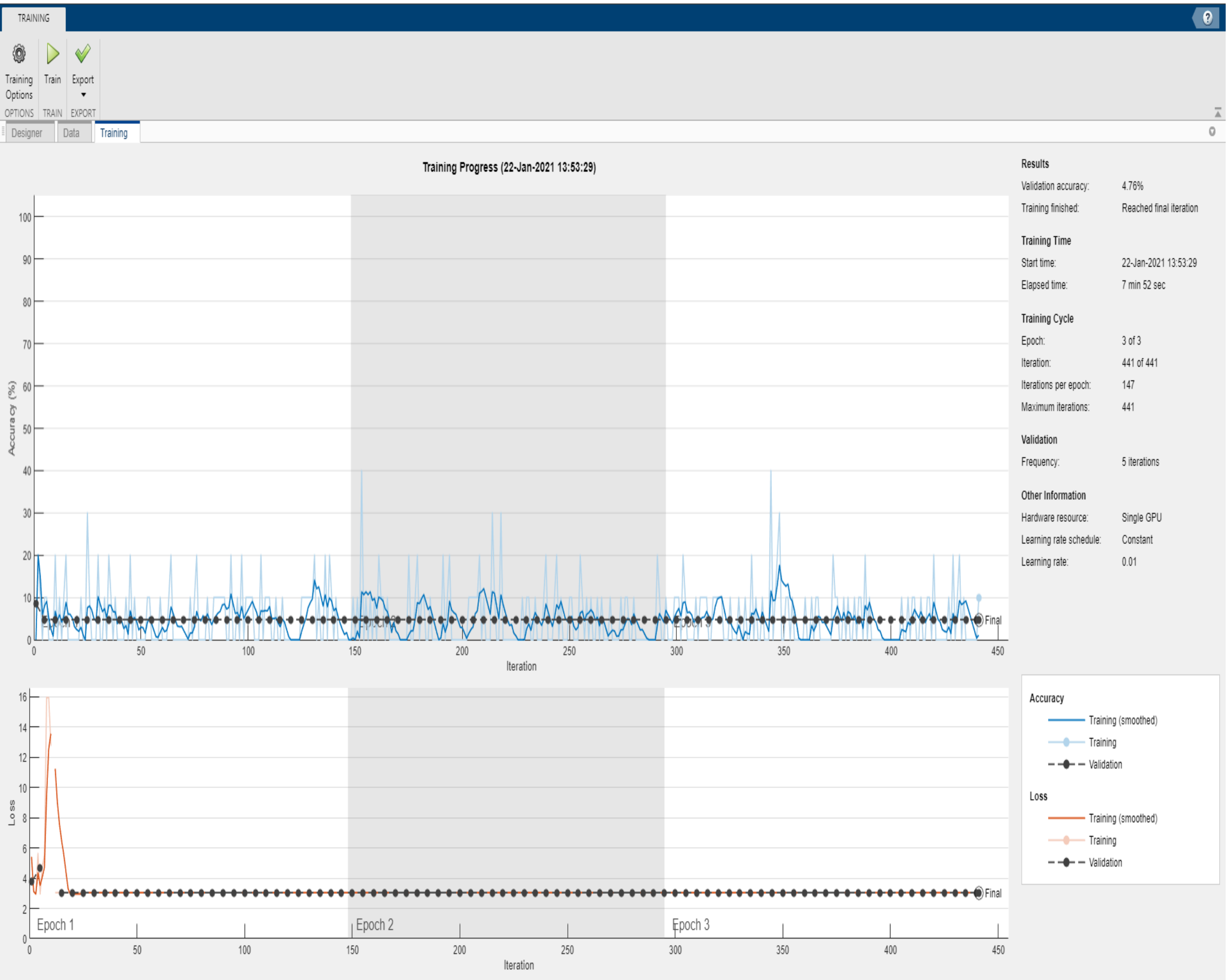
Düzenle
Yeni
Aç
E 2020-2021 Fall > AI > DL > UCMerced_LandUse > Images >
Ad
Değiştirme tarihi
Tür
Boyut
Import Data
IMPORT
Designer
Data
Training
Training data by class
Data:
Training
Source:
...\Images
Observations:
1470
Classes:
21
Most observations:
agricultural (70)
Fewest observations:
agricultural (70)
Number of observations
0
10
20
30
40
50
60
70
agricultural
airplane
baseballdiamond
beach
buildings
chaparral
denseresidential
forest
freeway
golfcourse
harbor
intersection
mediumresidential
mobilehomepark
overpass
parkinglot
river
runway
sparseresidential
storagetanks
tenniscourt
Class label


Data
Training
fire9-concat
depthConcaten...
drop9
dropoutLayer
conv
convolution2dL...
relu_conv10
reluLayer
pool10
globalAverage...
prob
softmaxLayer
classoutput
classificationLa...
PROPERTIES
convolution2dLayer
Name
conv
FilterSize
1,1
NumFilters
21
Stride
1,1
DilationFactor
1,1
Padding
same
Weights
[]
Bias
[]
WeightLearnRateFactor
10
WeightL2Factor
1
BiasLearnRateFactor
10
BiasL2Factor
0
WeightsInitializer
glorot
BiasInitializer
zeros
OVERVIEW

Covnet design was regulated for the study. Filter that recognise the classes and output operatör were renewed.

c. RESULTS


Training Options
SOLVER
Solver
sgdm
InitialLearnRate
0.01
BASIC
ValidationFrequency
5
MaxEpochs
3
MiniBatchSize
5
ExecutionEnvironment
auto
ADVANCED
L2Regularization
0.0001
GradientThresholdMethod
l2norm
GradientThreshold
Inf
ValidationPatience
Inf
Shuffle
every-epoch
CheckpointPath
LearnRateSchedule
none
LearnRateDropFactor
0.1
LearnRateDropPeriod
10
ResetInputNormalization
Momentum
0.9
Close

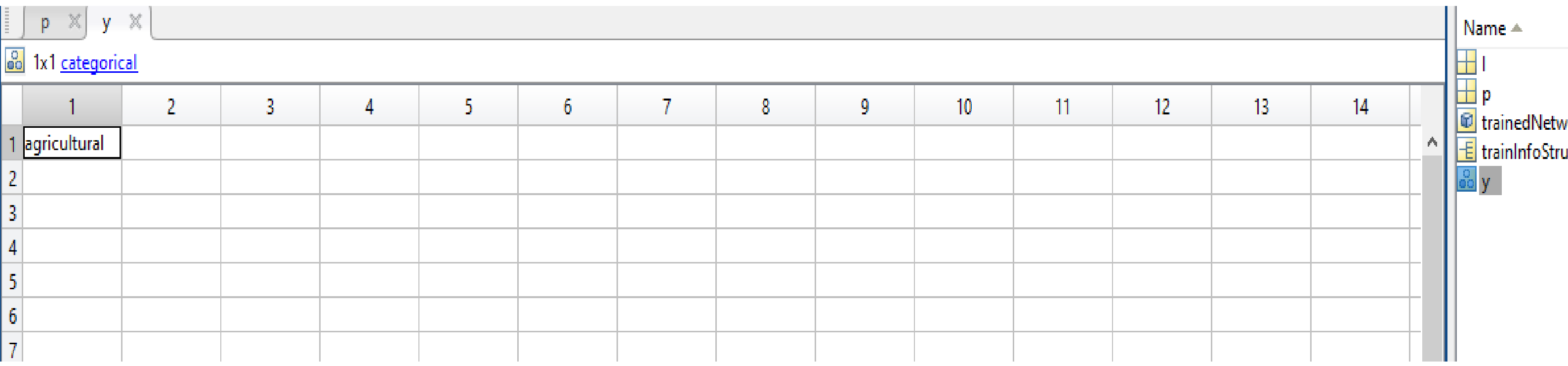


With first choosen settings model was not been able to trained as wanted.

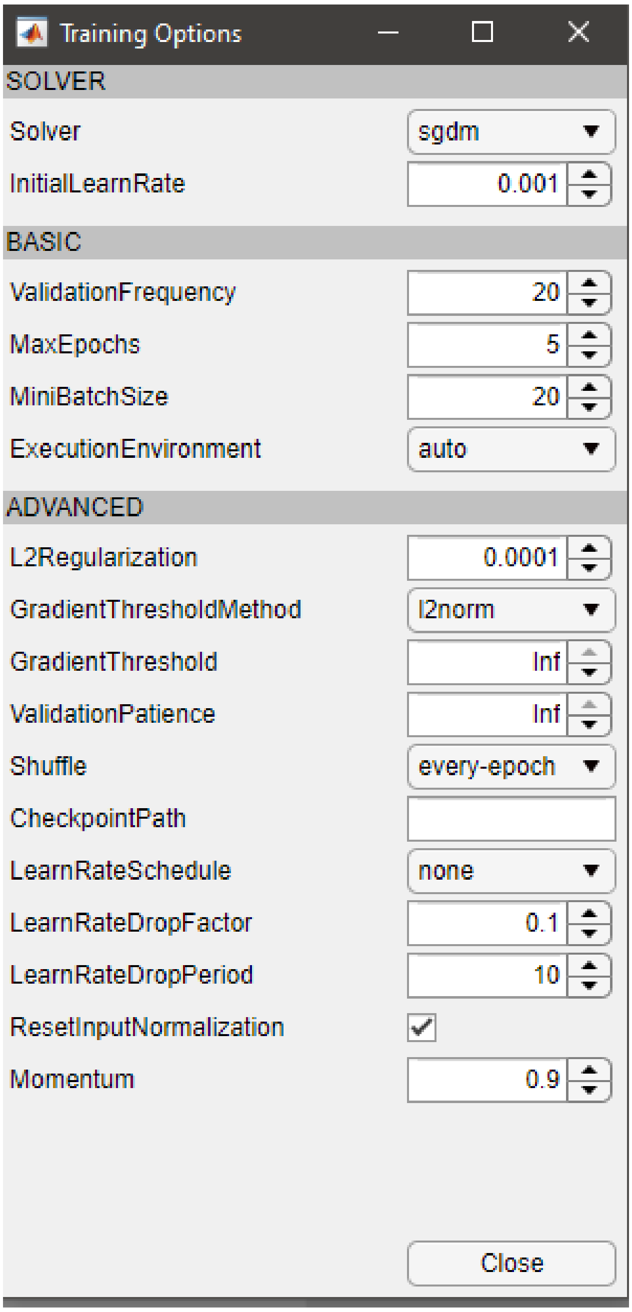
- I = imread ("C:\Users\Kullanıcı\Desktop\ME 2020-2021 Fall\AI\DL\UCMerced_LandUse\Images\airplane\airplane01.tif");
- I = imresize(I, [227 227]);
- [y,p] = classify(trainedNetwork_1,I);

With these codes image below was regulized and added to MATLAB.

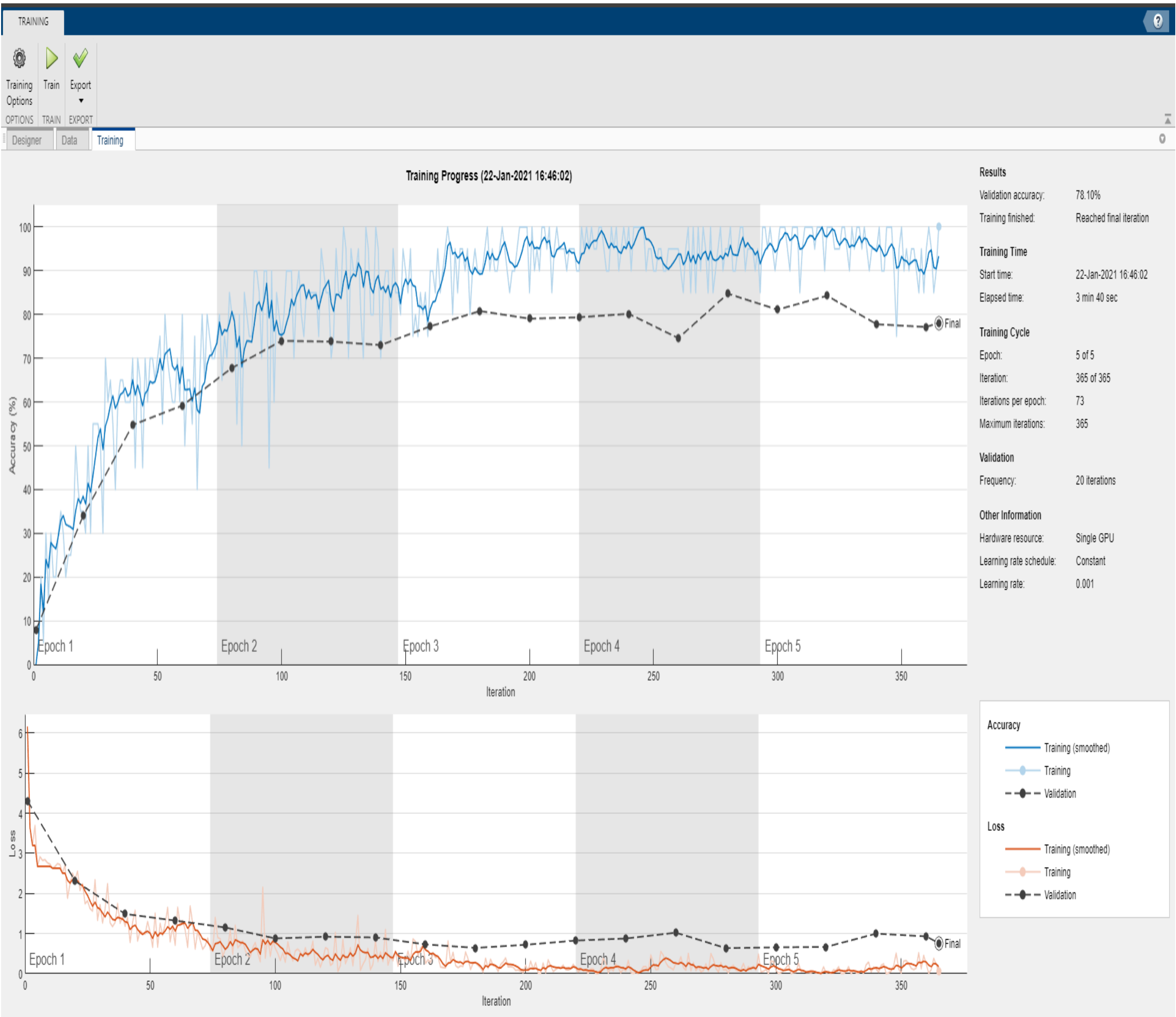
Image above was tested with trained model.



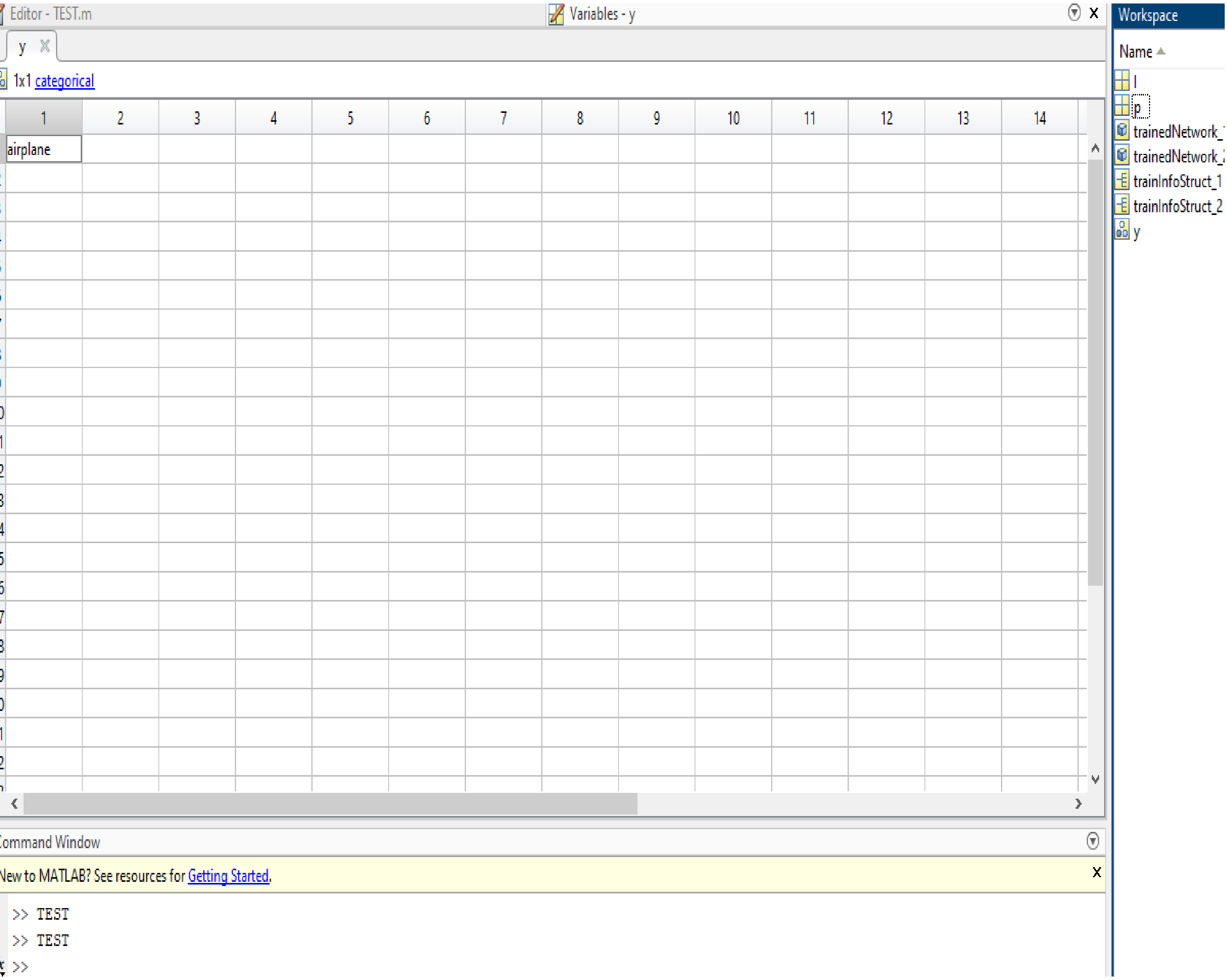
The image was forecasted as a member of agricultural class. So system was trained again due to the error.



Parameters were changed for success.



With these parameters model trained so much better accuracy with 78.10%.



With second test model can find the true class for the image. Study was completed with success.