Fuzzy Systems - Assignment #3 - Arian Tashakkor - 40023494

Problem 1

First we will train an ANFIS programmatically. For this problem we have chosen $\#mf = 7$ to keep the number of rules below 50 $(7^2 = 49 < 50)$.

```
warning('off', 'all');
epoch_n = 100;
mf_n = 7;

% ====== collect training data
point_n = 11;
x = linspace(-10, 10, point_n);
y = linspace(-10, 10, point_n);
[xx, yy] = meshgrid(x, y);

zz = (xx.^2 - yy.^2)*sin(xx./2);
trn_data = [xx(:) yy(:) zz(:)];

% ====== training options
ss = 0.1;
ss_dec_rate = 0.9;
ss_inc_rate = 1.1;
mf_type = 'gbellmf';

% ====== generate the initial FIS
in_fismat = genfis1(trn_data, mf_n, mf_type);

% ====== start training
[trn_out_fismat, trn_error, step_size] = ...
    anfis(trn_data, in_fismat, [epoch_n nan ss ss_dec_rate ss_inc_rate], ...
        [1,1,1,1]);
```

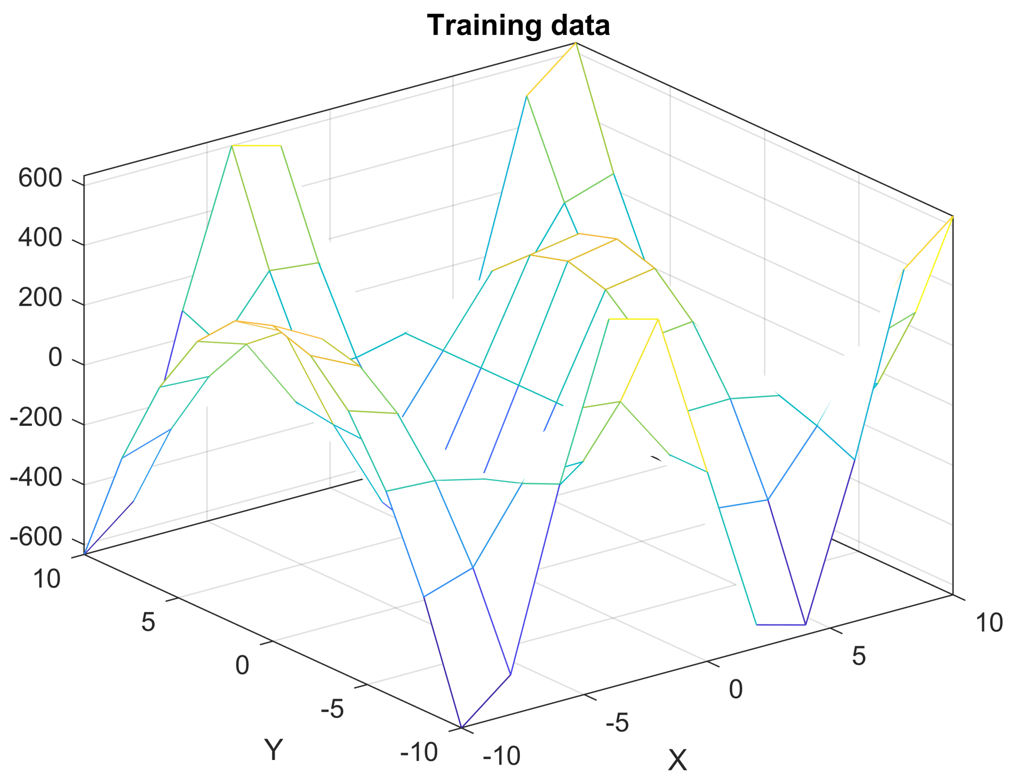
```
ANFIS info:
    Number of nodes: 131
    Number of linear parameters: 147
    Number of nonlinear parameters: 42
    Total number of parameters: 189
    Number of training data pairs: 121
    Number of checking data pairs: 0
    Number of fuzzy rules: 49


Start training ANFIS ...

1      4.59496
2      3.64365
3      2.8878
4      2.42837
Step size increases to 0.110000 after epoch 5.
5      2.11692
6      1.83026
7      1.55015
8      1.32854
Step size increases to 0.121000 after epoch 9.
```

```
9      1.19757
10      1.09258
11      1.03145
12      0.929942
Step size increases to 0.133100 after epoch 13.
13      0.881159
14      0.806335
15      0.825245
16      0.712934
17      0.734281
Step size decreases to 0.119790 after epoch 18.
18      0.638907
19      0.663525
20      0.514858
21      0.603611
Step size decreases to 0.107811 after epoch 22.
22      0.469325
23      0.556878
24      0.375481
25      0.515551
Step size decreases to 0.097030 after epoch 26.
26      0.345976
27      0.48352
28      0.272226
29      0.453586
Step size decreases to 0.087327 after epoch 30.
30      0.25232
31      0.430856
32      0.192905
33      0.407855
Step size decreases to 0.078594 after epoch 34.
34      0.179073
35      0.391145
36      0.130492
37      0.371886
Step size decreases to 0.070735 after epoch 38.
38      0.120834
39      0.359129
40      0.0808887
41      0.340252
Step size decreases to 0.063661 after epoch 42.
42      0.0749885
43      0.330219
44      0.0422345
45      0.303462
Step size decreases to 0.057295 after epoch 46.
46      0.0453858
47      0.296347
48      0.0165978
49      0.240771
Step size decreases to 0.051566 after epoch 50.
50      0.0557743
51      0.237393
52      0.0263361
53      0.229784
Step size decreases to 0.046409 after epoch 54.
54      0.030037
55      0.223563
56      0.00741432
57      0.169483
Step size decreases to 0.041768 after epoch 58.
58      0.0546612
59      0.167236
60      0.0325537
```

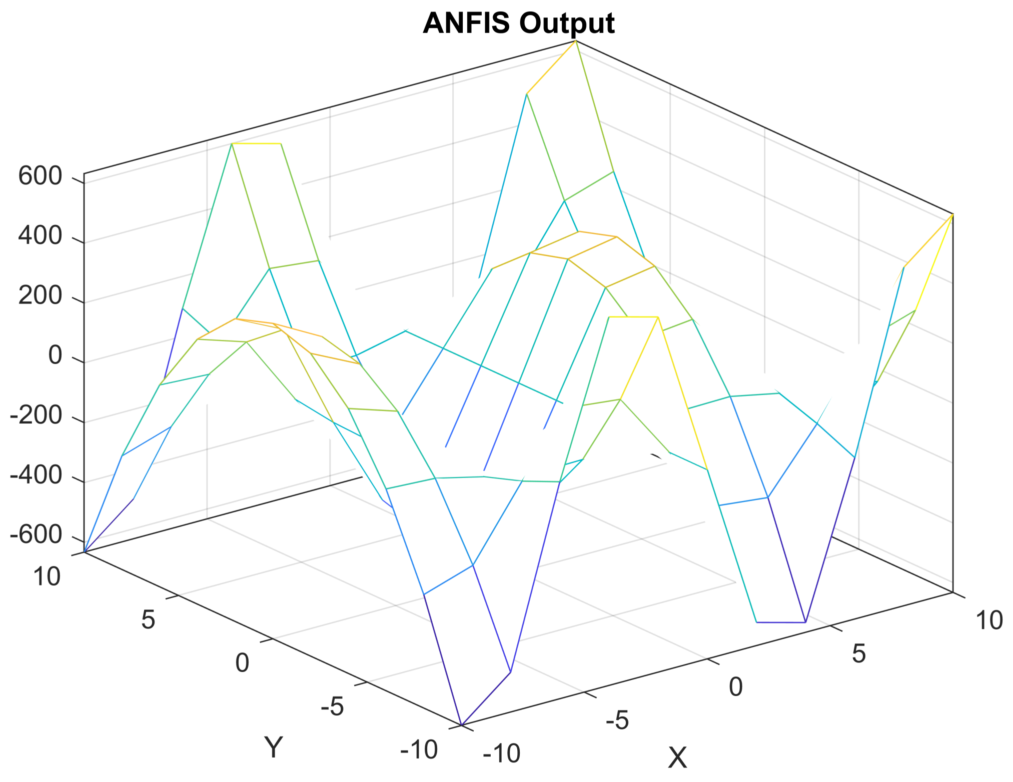
```
61      0.163317
Step size decreases to 0.037591 after epoch 62.
62      0.0343148
63      0.159717
64      0.0163803
65      0.149086
Step size decreases to 0.033832 after epoch 66.
66      0.024448
67      0.143779
68      0.0108044
69      0.124805
Step size decreases to 0.030449 after epoch 70.
70      0.0270283
71      0.120999
72      0.0144295
73      0.111008
Step size decreases to 0.027404 after epoch 74.
74      0.0226273
75      0.106562
76      0.0126567
77      0.0956097
Step size decreases to 0.024664 after epoch 78.
78      0.0220119
79      0.0916057
80      0.0133992
81      0.0829742
Step size decreases to 0.022197 after epoch 82.
82      0.0207849
83      0.0791569
84      0.0135194
85      0.0717639
Step size decreases to 0.019978 after epoch 86.
86      0.0198962
87      0.0682121
88      0.013697
89      0.061932
Step size decreases to 0.017980 after epoch 90.
90      0.0191417
91      0.0586622
92      0.0138166
93      0.0533027
Step size decreases to 0.016182 after epoch 94.
94      0.0184777
95      0.0503249
96      0.013866
97      0.0457314
Step size decreases to 0.014564 after epoch 98.
98      0.0178626
99      0.0430478
100      0.0138348

Designated epoch number reached. ANFIS training completed at epoch 100.

Minimal training RMSE = 0.00741432
```

```
% ====== compute ANFIS output
z_hat = evalfis([xx(:) yy(:)], trn_out_fismat);

% ====== plot of training data
mesh(xx, yy, zz);
limit = [min(xx(:)) max(xx(:)) min(yy(:)) max(yy(:)) ...
    min(zz(:)) max(zz(:))];
axis(limit); set(gca, 'box', 'on');
```

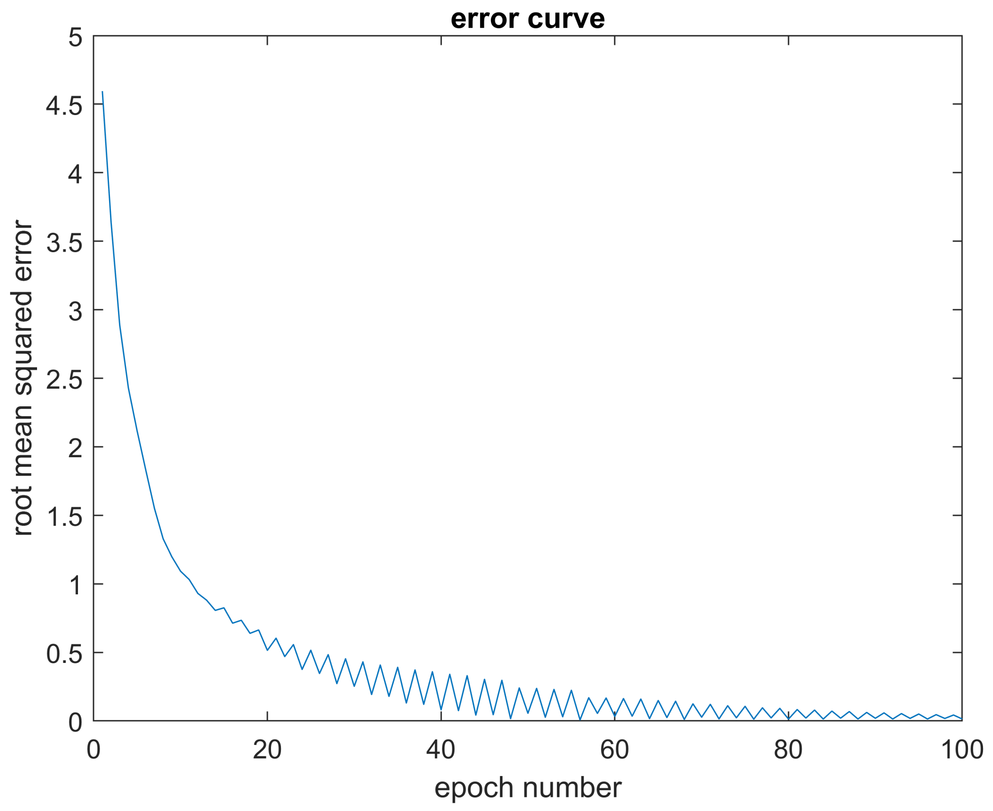
```
xlabel('X'); ylabel('Y'); title('Training data');
hold off
```



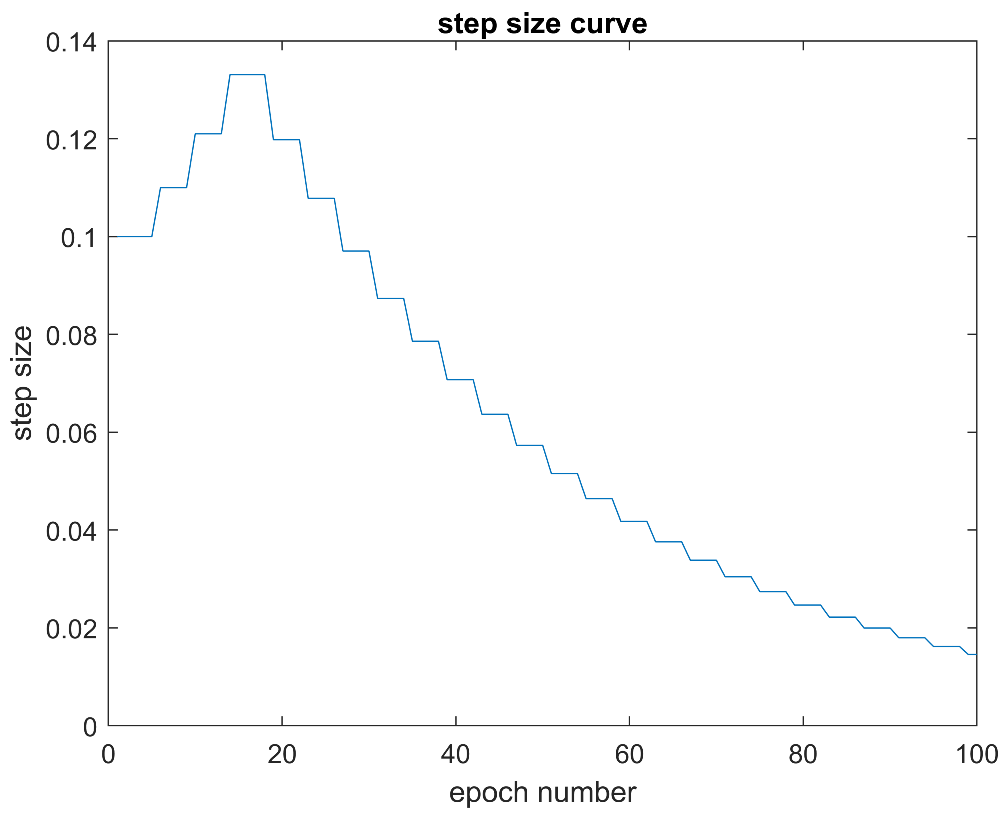
```
zz_hat = evalfis([xx(:) yy(:)], trn_out_fismat);
mesh(xx, yy, reshape(zz_hat, point_n, point_n));
axis(limit); set(gca, 'box', 'on');
xlabel('X'); ylabel('Y'); title('ANFIS Output');
hold off
```



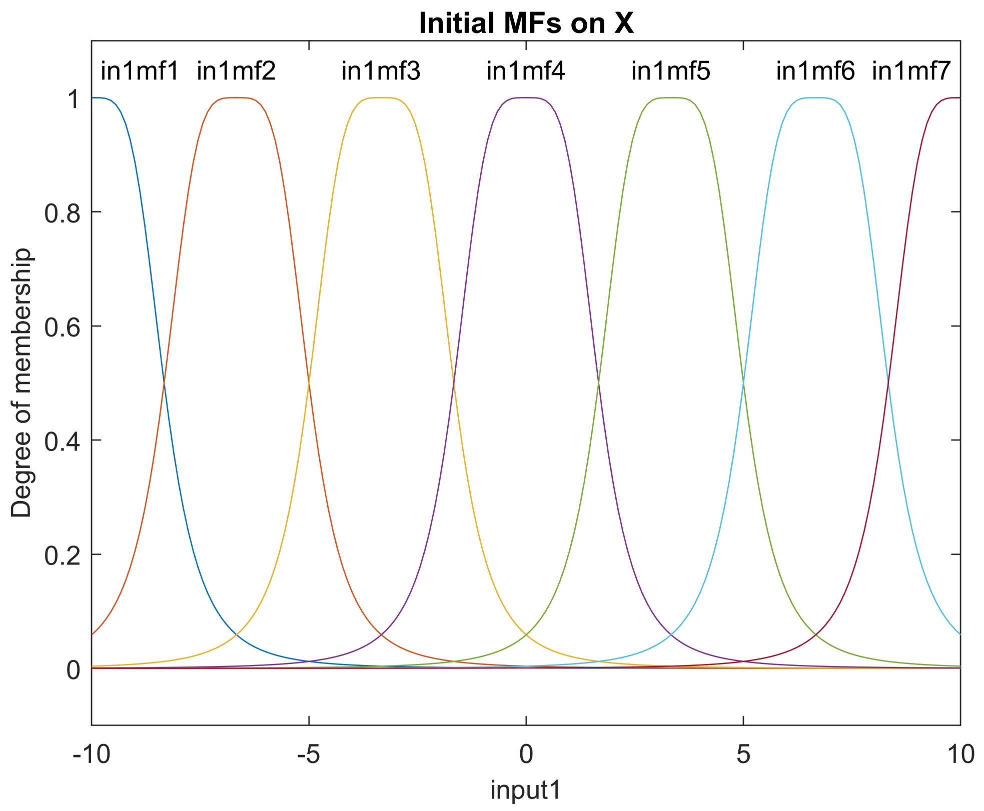
```
plot(1:epoch_n, trn_error);
xlabel('epoch number'); ylabel('root mean squared error');
title('error curve');
hold off
```



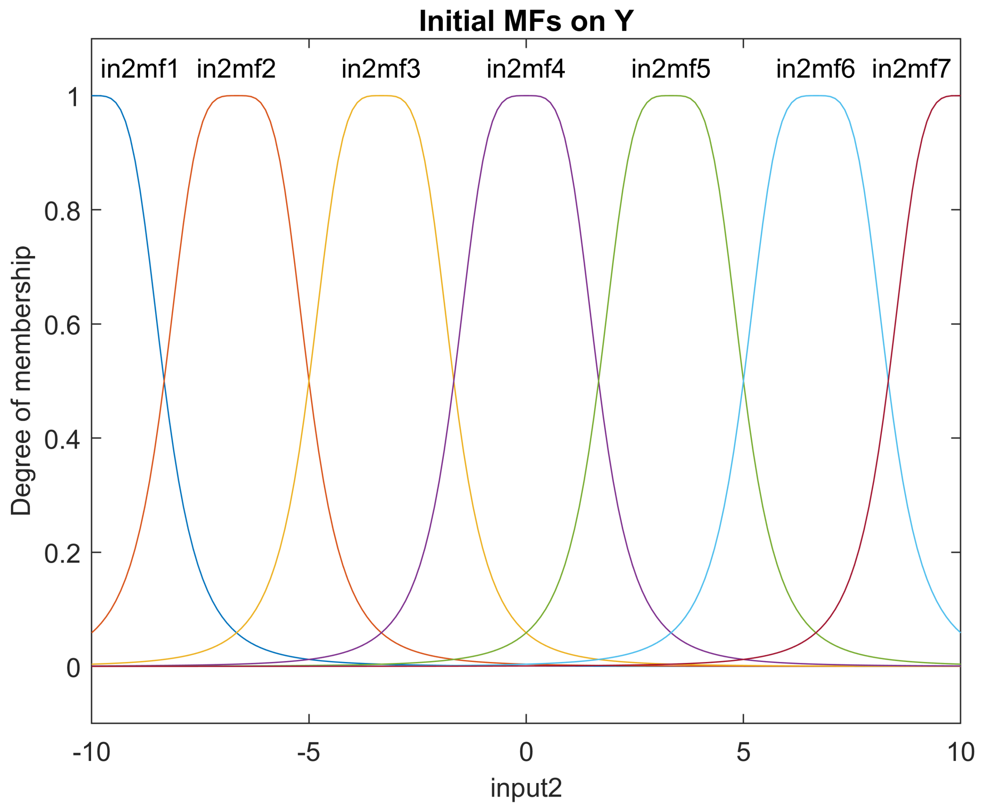
```
plot(1:epoch_n, step_size);
xlabel('epoch number'); ylabel('step size');
title('step size curve');
hold off
```



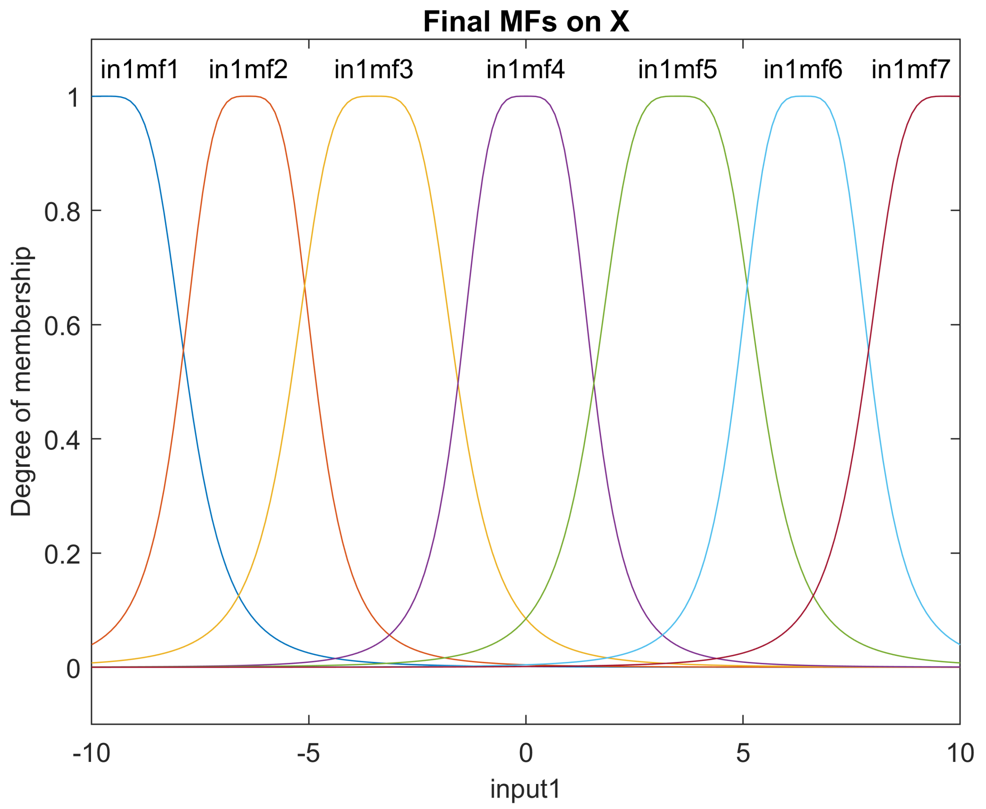
```
% ====== plot MFs
% plot initial MFs on x and y
plotmf(in_fismat, 'input', 1); title('Initial MFs on X');
hold off
```



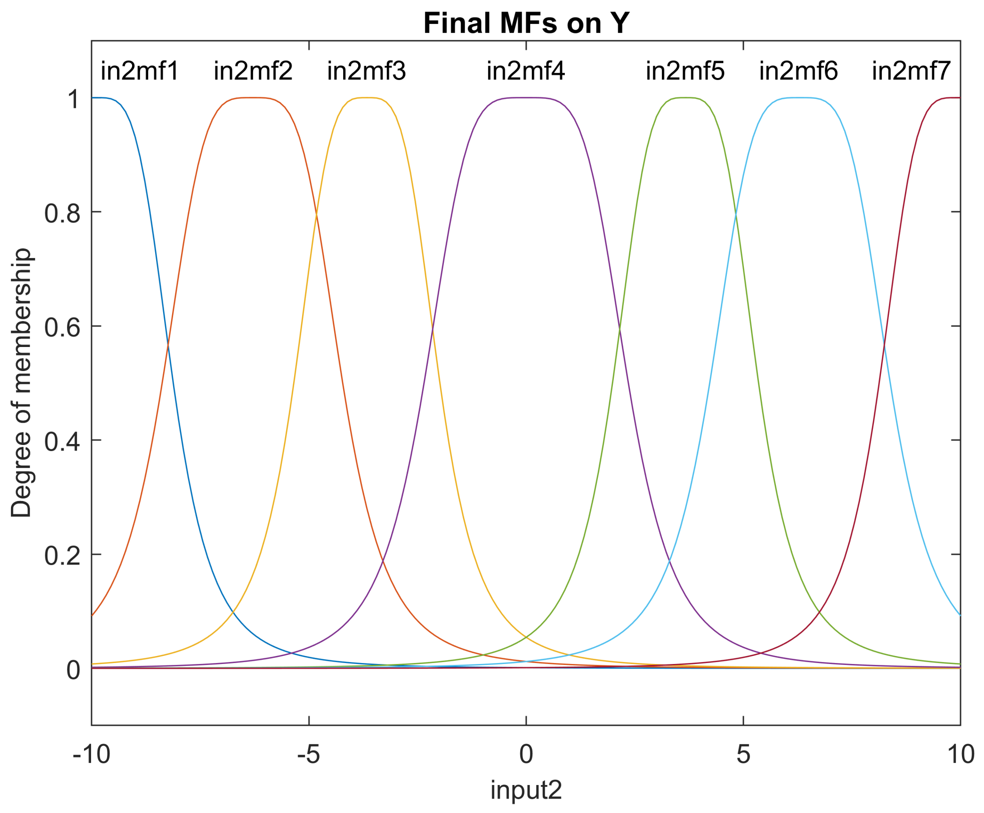
```
plotmf(in_fismat, 'input', 2); title('Initial MFs on Y');
hold off
```



```
% plot final MFs on X and Y
plotmf(trn_out_fismat, 'input', 1); title('Final MFs on X');
hold off
```

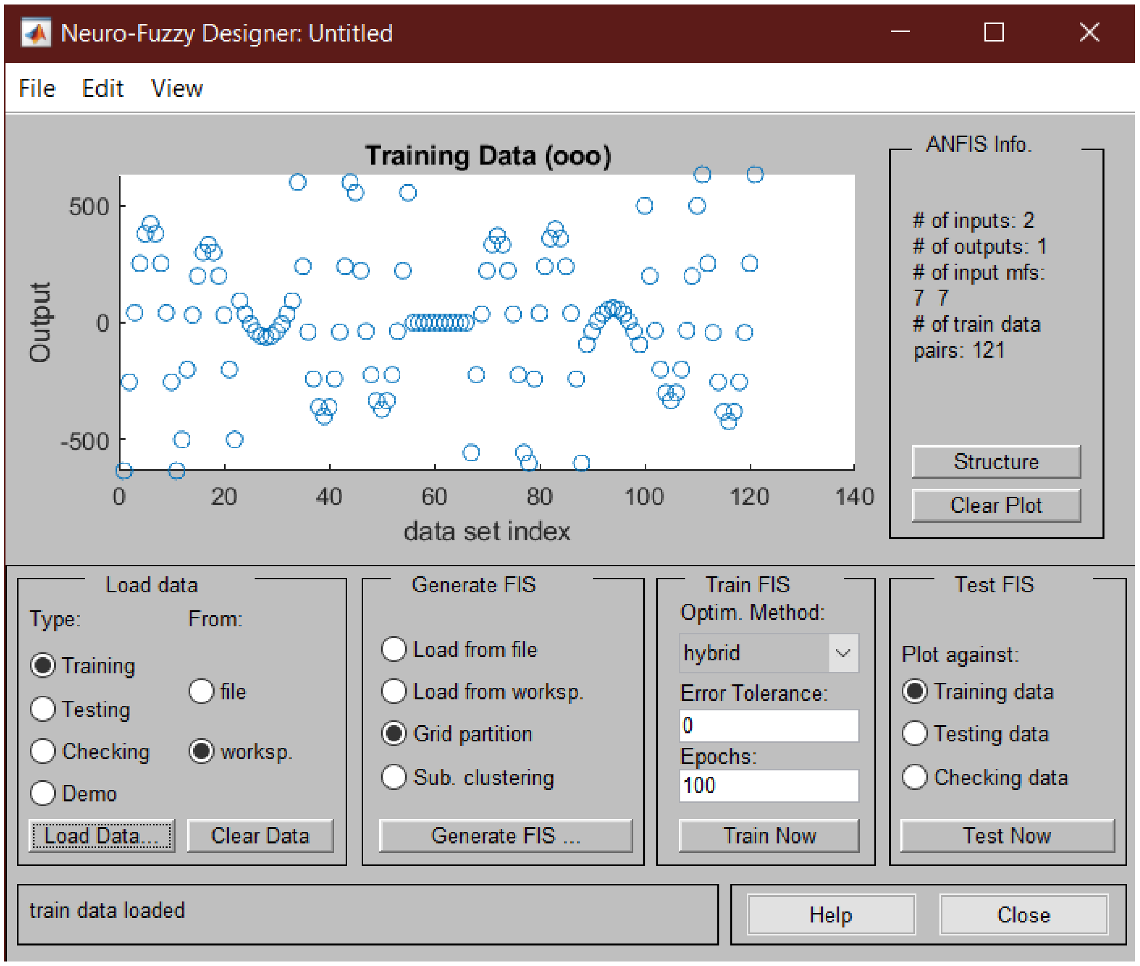


```
plotmf(trn_out_fismat, 'input', 2); title('Final MFs on Y');
hold off
```



Next we will do the same thing using the MATLAB Neuro-Fuzzy Designer.

1. First we will load in the training data:



# of inputs: 2
# of outputs: 1
# of input mfs:
7 7
# of train data
pairs: 121

Load data — Type: [✓ Training] Testing Checking Demo; From: file [✓ worksp.]

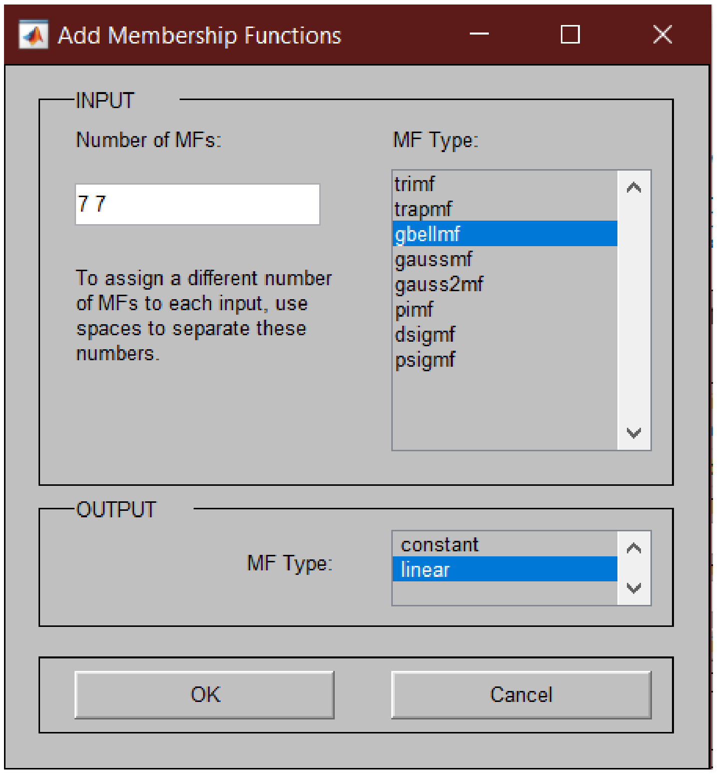
Generate FIS: Load from file, Load from worksp., [✓ Grid partition], Sub. clustering

Train FIS — Optim. Method: hybrid; Error Tolerance: 0; Epochs: 100

Test FIS — Plot against: [✓ Training data] Testing data Checking data

train data loaded

2. Now we will set 7 bell functions per input after pressing "Genearte FIS ...":



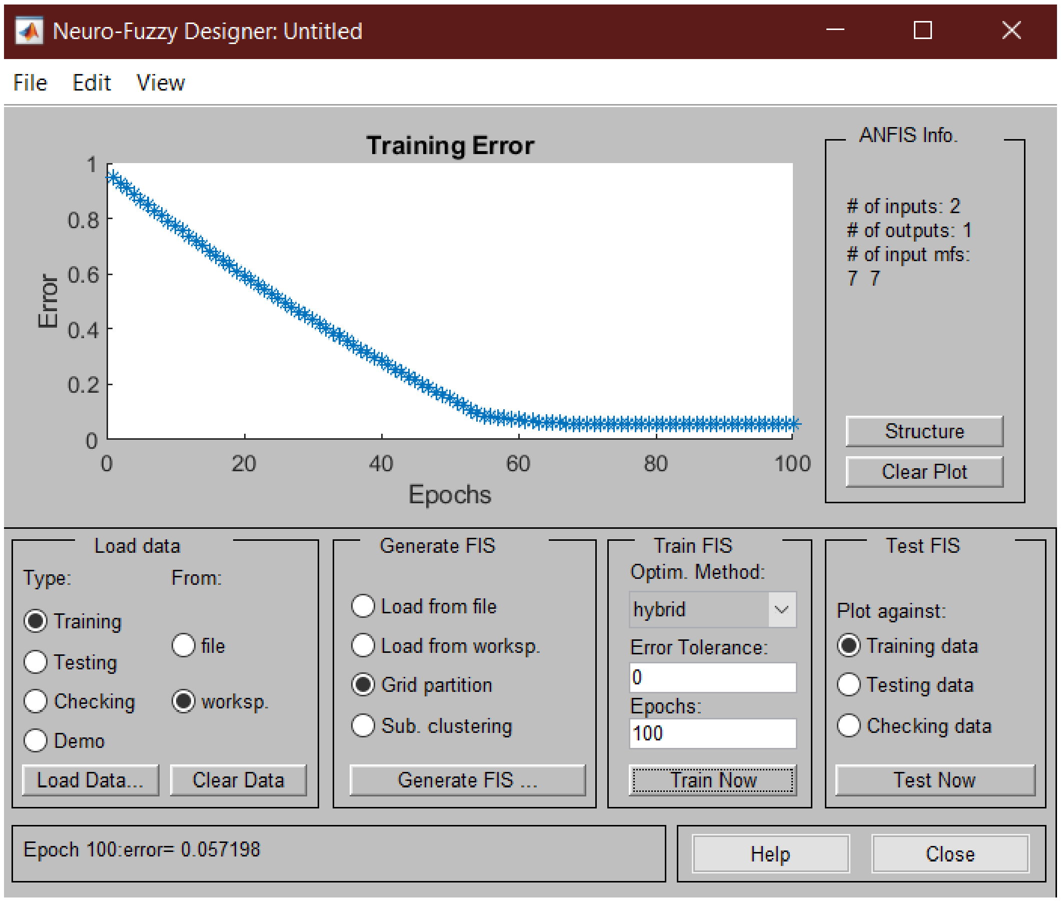
INPUT — Number of MFs: 7 7

To assign a different number of MFs to each input, use spaces to separate these numbers.

MF Type: trimf, trapmf, [✓ gbellmf], gaussmf, gauss2mf, pimf, dsigmf, psigmf

OUTPUT — MF Type: constant, [✓ linear]

3. Finally we will train for 100 epochs using the hybrid method by setting the number of epochs under the "Epochs" field and pressing "Train Now":



ANFIS Info.: # of inputs: 2; # of outputs: 1; # of input mfs: 7 7

Load data — Type: [✓ Training] Testing Checking Demo; From: file [✓ worksp.]

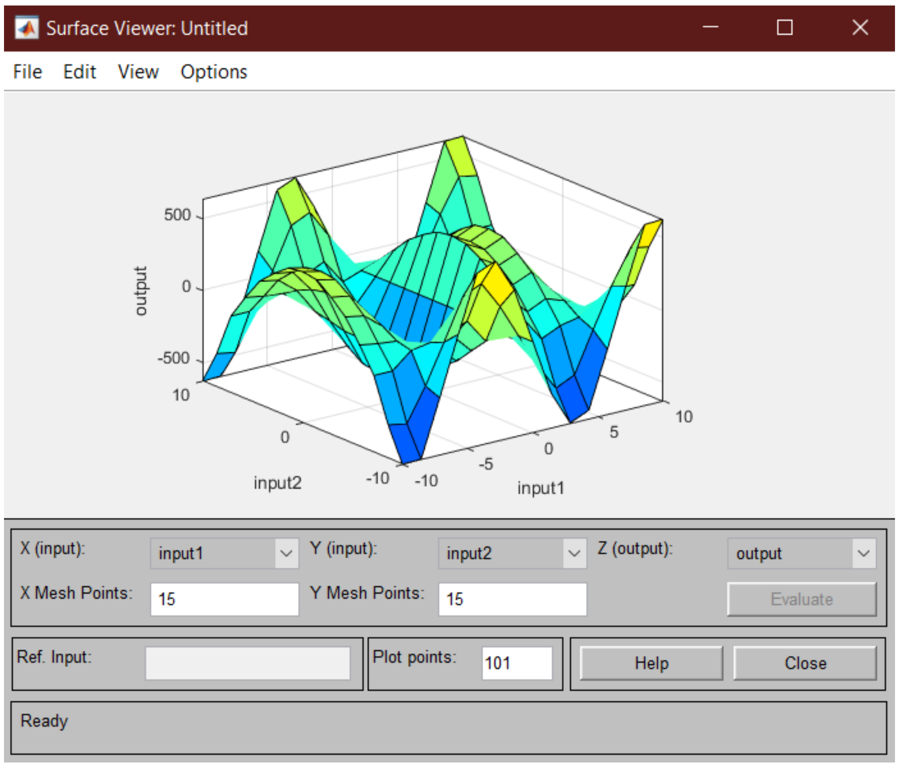
Generate FIS: Load from file, Load from worksp., [✓ Grid partition], Sub. clustering

Train FIS — Optim. Method: hybrid; Error Tolerance: 0; Epochs: 100

Test FIS — Plot against: [✓ Training data] Testing data Checking data

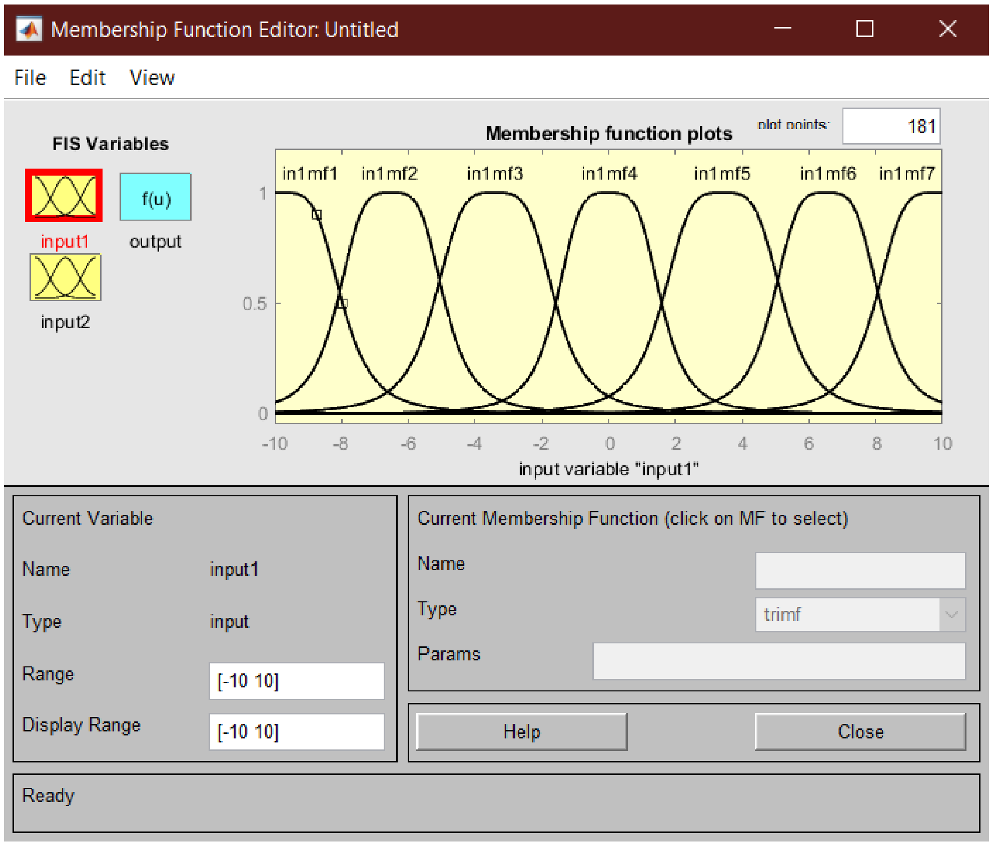
Epoch 100:error= 0.057198

4. We can see the produced surface by the ANFIS using View->Surface:



X (input): input1
Y (input): input2
Z (output): output
X Mesh Points: 15
Y Mesh Points: 15
Ref. Input:
Plot points: 101
Ready

5. Input 1's membership functions after training:



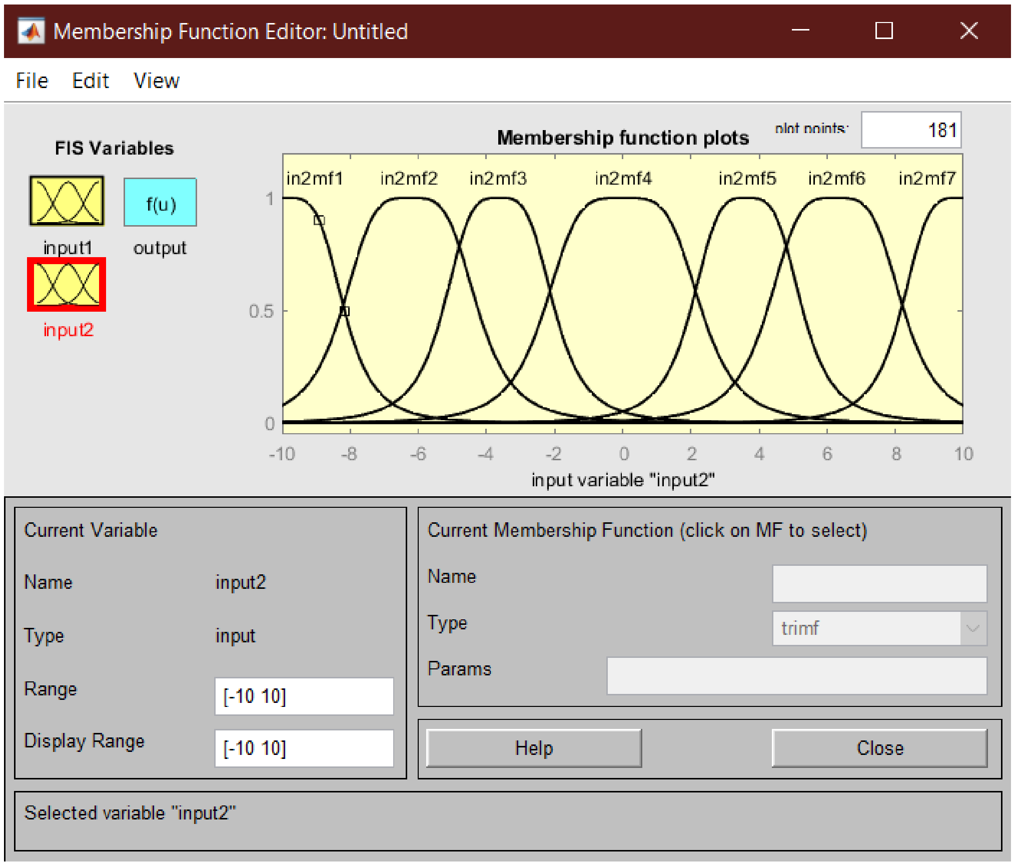
plot points: 181

Current Variable
Name input1
Type input
Range [-10 10]
Display Range [-10 10]

Current Membership Function (click on MF to select)
Name
Type trimf
Params

Ready

6. Input 2's membership functions after training:



Current Variable

Name input2

Type input

Range [-10 10]

Display Range [-10 10]

Current Membership Function (click on MF to select)

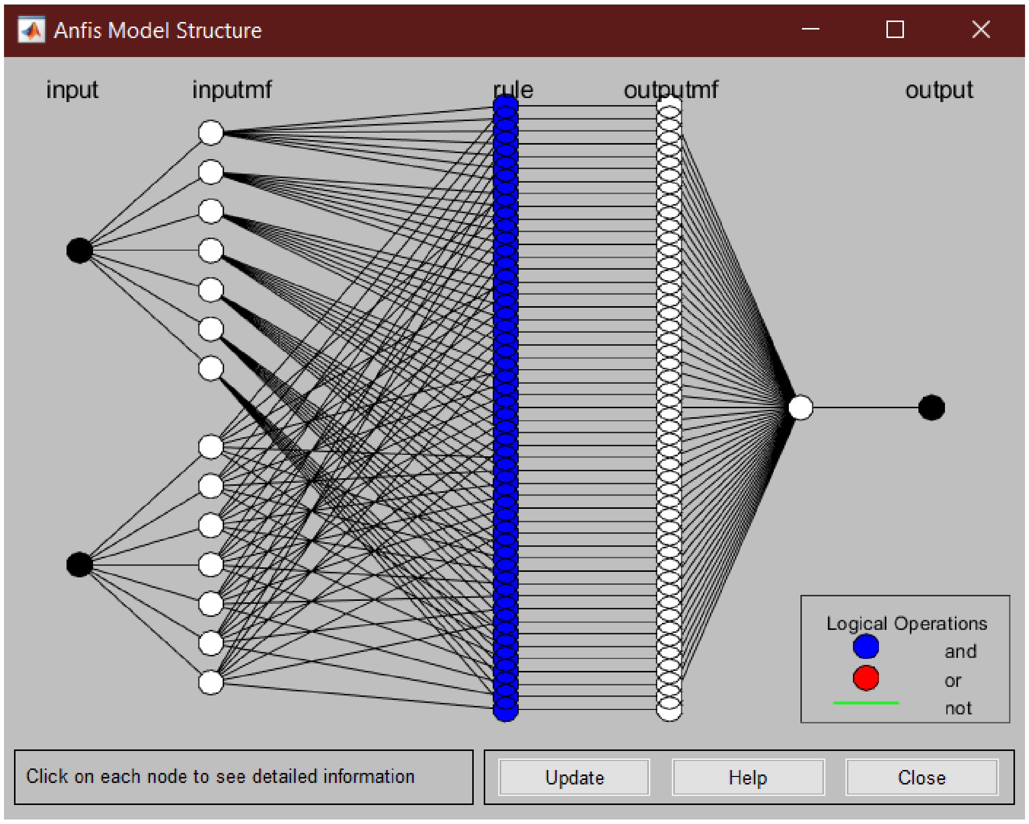
Name

Type trimf

Params

Selected variable "input2"

7. And finally network structure:



Click on each node to see detailed information