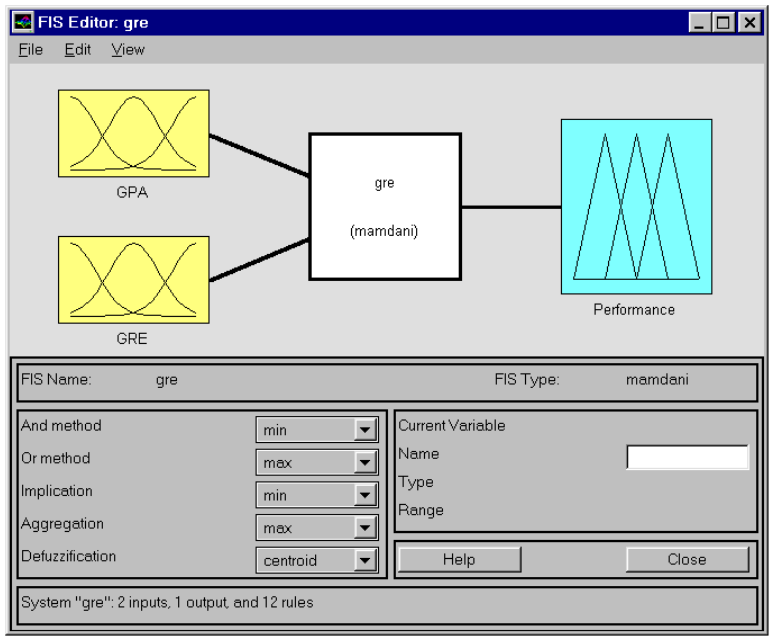
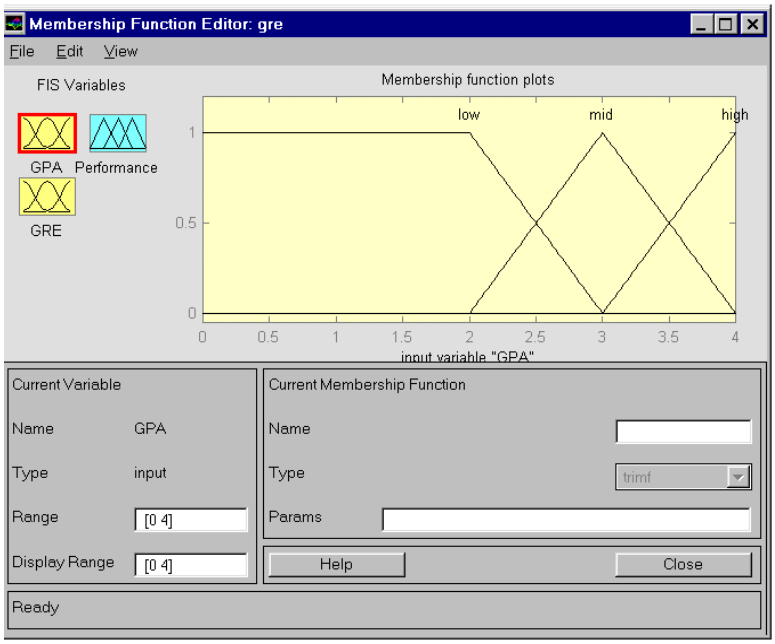
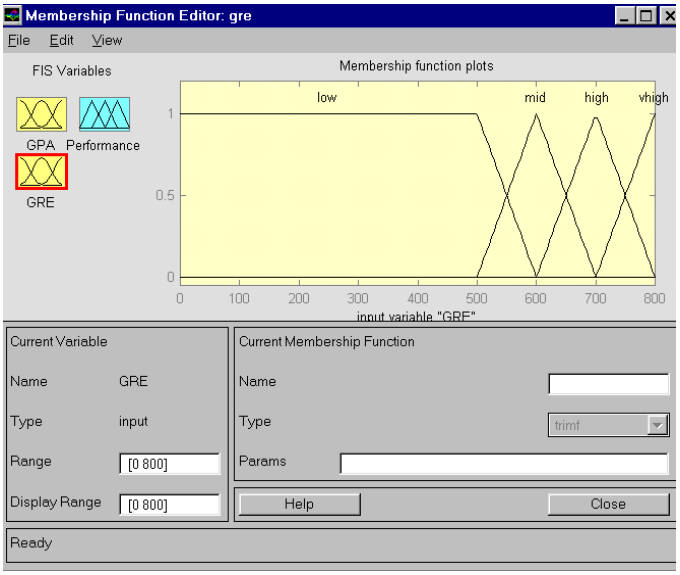
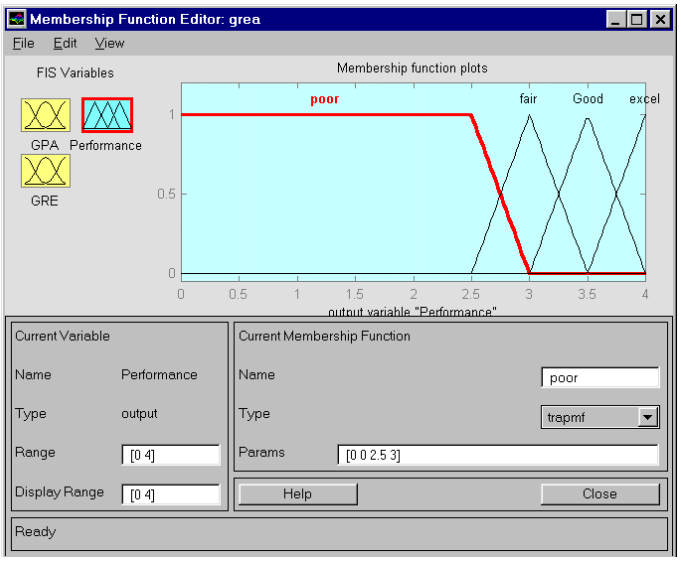
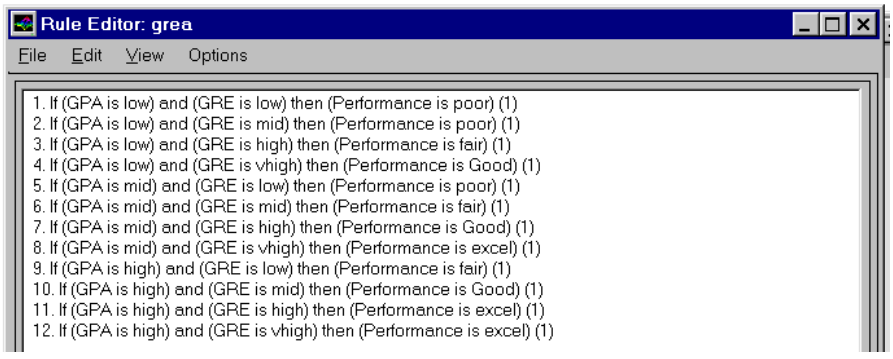
GRE/GPA example implemented using Matlab's Fuzzy Inference System Editor

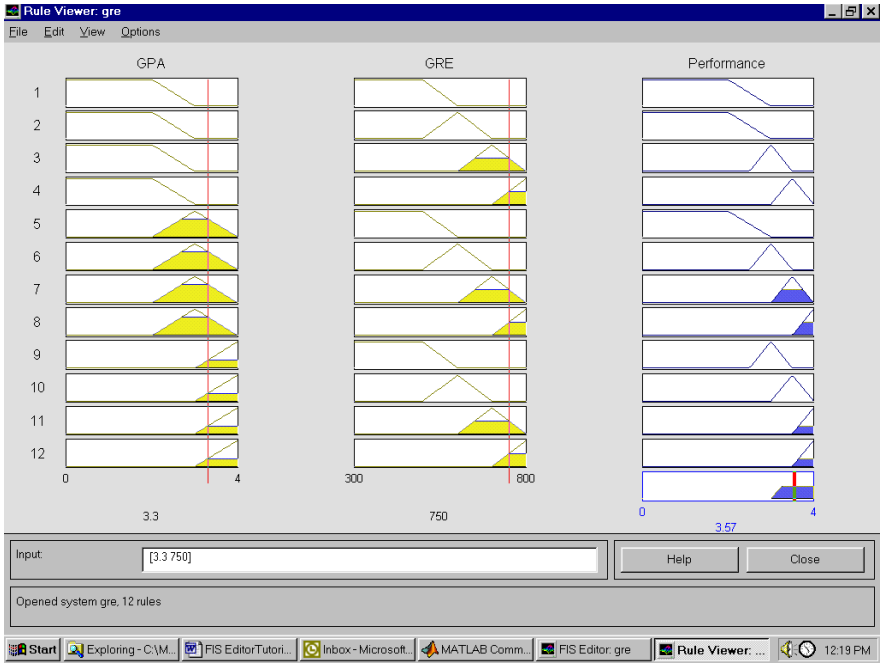





Membership Function Editor: gre
File Edit View
FIS Variables
GPA Performance
GRE
Membership function plots
low mid high vhigh
input variable "GRE"
Current Variable
Name GRE
Type input
Range [0 800]
Display Range [0 800]
Current Membership Function
Name
Type trimf
Params
Help Close
Ready


Membership Function Editor: grea
File Edit View
FIS Variables
GPA Performance
GRE
Membership function plots
poor fair Good excel
output variable "Performance"
Current Variable
Name Performance
Type output
Range [0 4]
Display Range [0 4]
Current Membership Function
Name poor
Type trapmf
Params [0 0 2.5 3]
Help Close
Ready


Rule Editor: grea
File Edit View Options
1. If (GPA is low) and (GRE is low) then (Performance is poor) (1)
2. If (GPA is low) and (GRE is mid) then (Performance is poor) (1)
3. If (GPA is low) and (GRE is high) then (Performance is fair) (1)
4. If (GPA is low) and (GRE is vhigh) then (Performance is Good) (1)
5. If (GPA is mid) and (GRE is low) then (Performance is poor) (1)
6. If (GPA is mid) and (GRE is mid) then (Performance is fair) (1)
7. If (GPA is mid) and (GRE is high) then (Performance is Good) (1)
8. If (GPA is mid) and (GRE is vhigh) then (Performance is excel) (1)
9. If (GPA is high) and (GRE is low) then (Performance is fair) (1)
10. If (GPA is high) and (GRE is mid) then (Performance is Good) (1)
11. If (GPA is high) and (GRE is high) then (Performance is excel) (1)
12. If (GPA is high) and (GRE is vhigh) then (Performance is excel) (1)



Adaptive Network based Fuzzy Inference System (ANFIS) Training routine

ANFIS Training routine for Sugeno-type FIS (MEX only). ANFIS applies the least-squares method and the back-propagation gradient descent for identifying linear and nonlinear parameters, respectively, in a Sugeno-type fuzzy inference systems.

FISMAT = ANFIS(TRN_DATA) attempts to fit a training data TRN_DATA using a Sugeno FIS with 2^N rules, where N is the number of inputs. (This is recommanded only when N is less than 7.) The final fuzzy inference system is returned as a FIS matrix FISMAT. The format of TRN_DATA is described below.

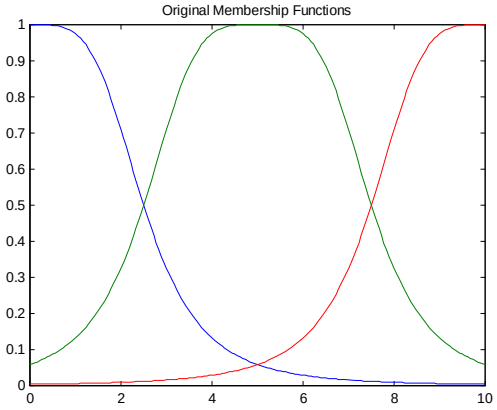
For More Detail use “Help ANFIS” in Matlab.

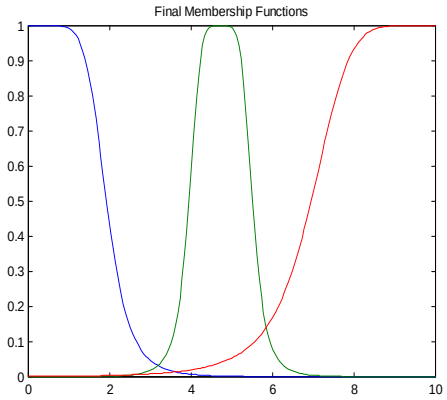
Example:

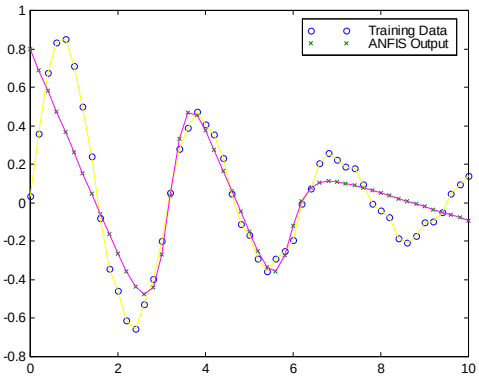
```
x = (0:0.2:10)';
 y = sin(2*x)./exp(x/5) + randn(size(x))/30;
 TrainData = [x y];
 NumMfs = 3;
 MfType = 'gbellmf';
 NumEpochs = 100;
 StepSize = 0.1;
 InputFismat = genfis1(TrainData, NumMfs, MfType);
 [xa,mfa]= plotmf(InputFismat,'input',1);
plot(xa,mfa);
title ('Original Membership Functions');
OutputFismat = anfis(TrainData, InputFismat, [NumEpochs nan
StepSize]);
yy = evalfis(x, OutputFismat);

[x1,mf1]= plotmf(OutputFismat,'input',1);
plot(x1,mf1);
title ('Final Membership Functions');
```

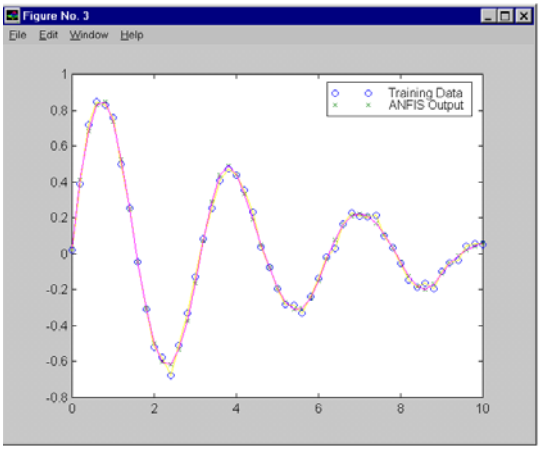
```
plot(x, y, 'o', x, yy, 'x', x, y, 'y', x, yy, 'm');
legend('Training Data', 'ANFIS Output');
```

See also GENFIS1, ANFISMEX.









Using three Membership Functions

Using five Membership Functions

GRE/GPA Example revisited

Number of Rules is reduced to 4 instead of 12