# Short Tutorial on Genfis/Anfis

And don’t forget ...

**VALIDATION**

# GenFIS

- The **genfis** functions in the MATLAB Fuzzy Logic Toolbox enable initial Fuzzy Inference Systems to be constructed from your training data prior to optimisation by ANFIS
- genfis1  uses a *grid partition* on the data
  - This results in equally spaced Membership Functions which are unlikely to be optimal
- genfis2 and genfis3 use clustering to create initial MFs which will make better use of the available computional resources
- Try each of them out on the invkine_codepad example

# Validation

- It is pointless to try different genfis/anfis options without a **quantitative** method of evaluation
- The most effective way of comparing accuracy is to use a set of validation points in addition to the training points as we did in the curve-fitting exercises
  - The error on the validation set gives a good indication of how well the system generalizes
- Look at the example code on the next slide to see how to use validation data in anfis
- Note that, if you supply validation data, anfis will return two output FISs
  - See slide 7 for an explanation
- Note also that you can experiment with each joint angle independently

```
%fizzmat1 = genfis1(train_data1);
fizzmat1 = genfis2(train_data1(:,1:2),train_data1(:,3),0.2);  cluster radius 0.2
%pifizzmat1 = genfis3(train_data1(:,1:2),train_data1(:,3), 20);

figure();
plotmf(fizzmat1,'input',1);


% trnOpt: a vector of training options.

% trnOpt(1): maximum training epoch number (default: 10)
% trnOpt(2): training error goal (default: 0)
% trnOpt(3): initial step size (default: 0.01)
% trnOpt(4): step size decrease rate (default: 0.9)
% trnOpt(5): step size increase rate (default: 1.1)

trnOpt = [200 0 0.01 0.9 1.1];



[anfis1_out,error1,stepsize1,val_fis1,val_err1] =
anfis(train_data1,fizzmat1,trnOpt,dispOpt,validation_data1,optMethod);
```

genfis2 generates a Sugeno-type FIS structure using **subtractive clustering** and requires separate sets of input and output data as input arguments

**fismat = genfis2(Xin,Xout,radii)**

The rule extraction method first uses the subclust function to determine the number of rules and antecedent membership functions

and then uses linear least squares estimation to determine each rule's consequent equations.

This function returns a FIS structure that contains a set of fuzzy rules to cover the feature space.

The arguments for genfis2 are as follows:

**Xin** is a matrix in which each row contains the input values of a data point.

**Xout** is a matrix in which each row contains the output values of a data point.

**radii** is a vector that specifies a cluster center's range of influence in each of the data dimensions,

*assuming the data falls within a unit hyperbox*.

For example, if the data dimension is 3 (e.g., Xin has two columns and Xout has one column), radii = [0.5 0.4 0.3] specifies that the ranges of influence in the first, second, and third data dimensions (i.e., the first column of Xin, the second column of Xin, and the column of Xout) are 0.5, 0.4, and 0.3 times the width of the data space, respectively.

If radii is a scalar value, then this scalar value is applied to all data dimensions, i.e., each cluster center has a spherical neighborhood of influence with the given radius.

Typically, cluster radii are between 0.2 and 0.5

The input membership function type is 'gaussmf', and the output membership function type is 'linear'

**genfis3** generates an FIS using ***fuzzy c-means*** (FCM) clustering by extracting a set of rules that models the data behavior.

The function requires separate sets of input and output data as input arguments.

When there is only one output, you can use genfis3 to generate an initial FIS for anfis training.

The rule extraction method first uses the fcm function to determine the number of rules and membership functions for the antecedents and consequents.

**fismat = genfis3(Xin,Xout)** generates a Sugeno-type FIS structure (fismat) given input data Xin and output data Xout. The matrices Xin and Xout have one column per FIS input and output, respectively.

**fismat = genfis3(Xin,Xout,type,cluster_n)** generates an FIS structure of the specified type and allows you to specify the number of clusters (cluster_n) to be generated by FCM.

The input membership function type is 'gaussmf'. By default, the output membership function type is 'linear'.

# ANFIS Outputs

Output arguments for anfis are:

Fis — FIS structure whose parameters are tuned using the training data, returned as a structure.

Error — Root mean squared training data errors at each training epoch, returned as an array of scalars.

stepsize — Step sizes at each training epoch, returned as an array of scalars.

If the error measure undergoes two consecutive combinations of an increase followed by a decrease, then anfis scales the step size by the decrease rate, trnOpt(4).

If the error measure undergoes four consecutive decreases, then anfis scales the step size by the increase rate, trnOpt(5).

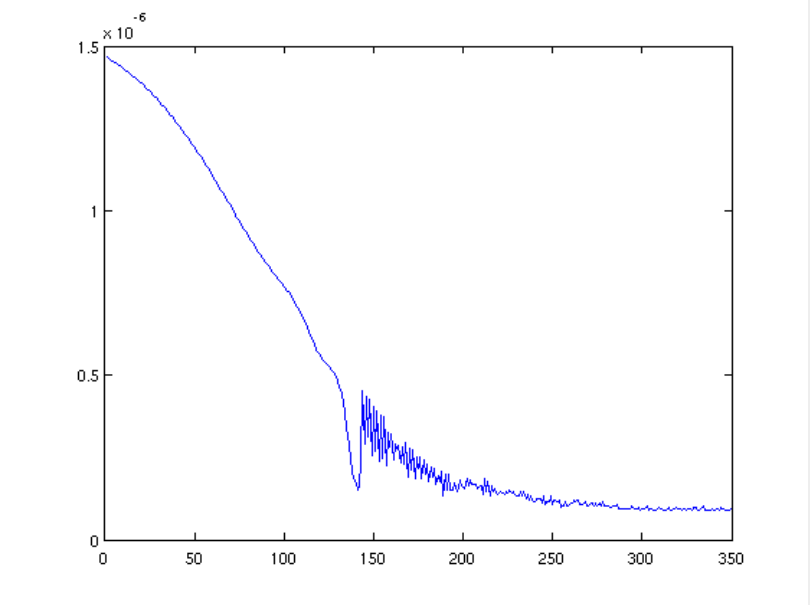
**ChkFis — FIS structure that corresponds to the epoch at which chkErr is minimum.**

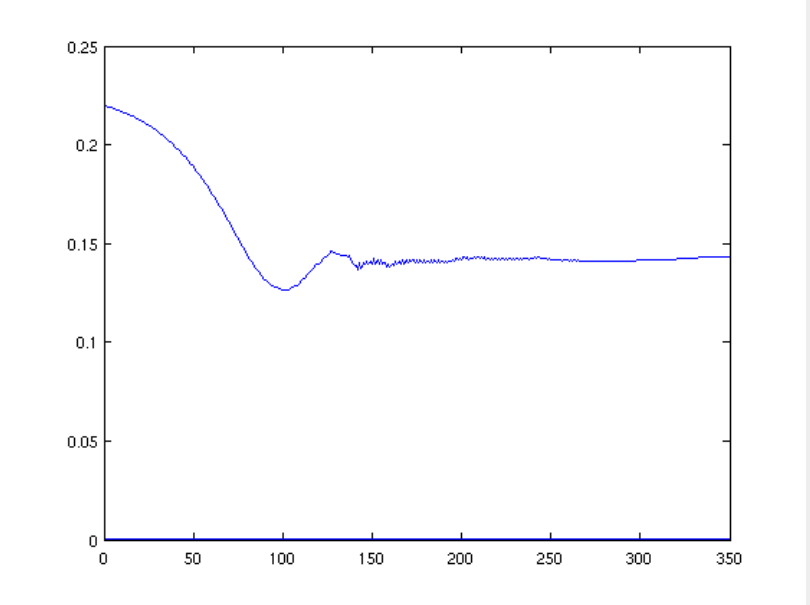
The function returns chkFis only when you supply chkData as an input argument.

chkErr — Root mean squared checking data errors at each training epoch, returned as an array of scalars.

Th**e function returns chkErr only when you supply chkData as an input argument.**

# An Example




- Training Error
- Validation Error
  - Shows over-fitting