

extremeansfis

Training routine for Sugeno-type Fuzzy Inference System

Syntax

```
[finalRMSE, Parameters]=extremeansfis(trainData)
[finalRMSE, Parameters]=extremeansfis(trainData, nMembershipFn)
[finalRMSE, Parameters]=extremeansfis(trainData, nMembershipFn, nOutput)
[finalRMSE, Parameters]=extremeansfis(trainData, nMembershipFn, nOutput, nEpochs)
```

Description

This syntax is the major training routine for Sugeno-type fuzzy inference systems. `extremeansfis` uses its own extreme-learning algorithm to identify parameters of Sugeno-type fuzzy inference systems to emulate a given training data set.

`extremeansfis` only supports Sugeno-type systems, and these must have the following properties:

- Be first order Sugeno-type systems.
- Have a single or multiple output, obtained using weighted average defuzzification. All output membership functions must be of bell shaped.
- Have no rule sharing. Different rules cannot share the same output membership function, namely the number of output membership functions must be equal to the number of rules.
- Have unity weight for each rule.

Moreover, `extremeansfis` cannot accept all the customization options that basic fuzzy inference allows. That is, you cannot make your own membership functions and defuzzification functions; you must use the ones provided.

The arguments in the description for `extremeansfis` are as follows. Note that you can specify the arguments `nMembershipFn`, `nOutput`, `nEpochs` as empty, `[]`, when necessary:

- `trainData`: the name of a training data set. Data should be arranged in column format, initial columns must represent inputs and then output vectors followed by inputs.
- `nMembershipFn`: the number of membership functions. Use `nMembershipFn`, an integer scalar value, as the second argument to `extremeansfis`. Each input to this FIS is characterized by two or more membership functions. Specify the number of membership functions in `nMembershipFn`. (default: 2)
- `nOutput`: the number of output vectors in `trainData`. If there are more than one outputs, then you have to specify an integer scalar as number of output vectors. (default: 1)
- `nEpochs`: the number of epochs. Use optional argument `nEpochs`, an integer scalar value (>20 recommended) to set number of epochs. (default: 50)

The variables return as output by `extremeanfis` are as follows:

- `finalRMSE`: returning variable `finalRMSE` gives the root mean square error of trained network for given training data.
- `Parameters`: is the structure containing designed premise and consequent parameters which are set according to a minimum training error criterion.

Examples

```
% fisheriris classification problem
load fisheriris
l=[[2*ones(45,1);zeros(45,1);zeros(45,1)] [zeros(45,1);2*ones(45,1);zeros(45,1)]
[zeros(45,1);zeros(45,1);2*ones(45,1)]];
trainData= [[meas(1:45,:);meas(51:95,:);meas(101:145,:)] 1];
[finalRMSE, Parameters] = extremeanfis(trainData, 5,3);
output = simextremeanfis([meas(1:45,:);meas(51:95,:);meas(101:145,:)],Parameters);
Test_data= [meas(46:50,:);meas(96:100,:);meas(146:150,:)];
test_output = simextremeanfis(Test_data,Parameters);
for i=1:3
    for j=1:length(test_output)
        if(test_output(j,i)<1)
            test_output(j,i)=0;
        end
        if(test_output(j,i)>=1)
            test_output(j,i)=2;
        end
    end
end
end
```