Package ‘anfis’

February 19, 2015

Type Package

Title Adaptive Neuro Fuzzy Inference System in R

Version 0.99.1

Date 2015-01-16

Author Cristobal Fresno, Andrea S. Llera and Elmer A. Fernandez

Maintainer Cristobal Fresno <cfresno@bdmg.com.ar>

Description The package implements ANFIS Type 3 Takagi and Sugeno's fuzzy if-then rule network with the following features:
(1) Independent number of membership functions (MF) for each input, and also different MF extensible types. (2) Type 3 Takagi and Sugeno's fuzzy if-then rule (3) Full Rule combinations, e.g. 2 inputs 2 membership functions -> 4 fuzzy rules (4) Hybrid learning, i.e. Descent Gradient for precedents and Least Squares Estimation for consequents (5) Multiple outputs.

URL http://www.bdmg.com.ar

License GPL (>= 2)

Depends R (>= 3.0), methods, parallel

Collate 'MembershipFunction.R' 'MembershipFunction-show.R' 'BellMF.R'
'GaussianMF.R' 'NormalizedGaussianMF.R'
'MembershipFunction-evaluateMF.R' 'Anfis.R'
'Anfis-initialize.R' 'Anfis-getters.R' 'Anfis-printshow.R'
'Anfis-metrics.R' 'Anfis-package.R' 'Anfis-plotMF.R'
'Anfis-plot.R' 'Anfis-predict.R' 'Anfis-training.R'
'Anfis-trainSet.R' 'Anfis3-example.R'
'MembershipFunction-derivateMF.R' 'MembershipFunction-getset.R'
'MembershipFunction-print.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2015-01-16 16:22:46
Description

The package implements ANFIS Type 3 Takagi and Sugeno’s fuzzy if-then rule network. This package includes the new following features:

1. Membership Functions (MF) flexible framework:
   - Flexible user-defined membership functions (MF) extensible class.
   - Independent number of (MF) for each input.
   - Different MF types, if required, for each input.
2. Type 3 Takagi and Sugeno’s fuzzy if-then rule
3. Full Rule combinations, e.g. 2 inputs 2 membership functions this means that 4 fuzzy rules will be created.
4. Different learning strategies:
   - `trainHybridJangOffLine` Hybrid learning, i.e. Descent Gradient for precedents and Least Squares Estimation for consequents.
   - `trainHybridJangOnLine` on-line version with hybrid learning.
   - `trainHybridOffLine` Adaptive learning coefficient and momentum term.
5. Multiple outputs support, i.e., the same input partition can be used to predict more than one output variable.
ANFIS-class

Author(s)
Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

References

ANFIS-class  ANFIS S4 class implementation in R

Description
Represent a concrete S4 class that represents an Adaptive Neuro Fuzzy Inference System in R, using type 3 Takagi and Sugeno’s fuzzy if-then rule with multiple outputs.

Slots
premises list with the MembershipFunctions for each input.
consequents numeric matrix with nrow= #rules, ncol= #outputs.
rules matrix with the connectivity of the membership functions to the rules.
X input matrix with ncol=#inputs and nrow=#individuals.
Y output matrix with ncol=#output and nrow=#individuals.
errors numeric vector with training errors.
trainingType character describing the training algorithm used: trainHybridJangOffLine, trainHybridOffLine or trainHybridJangOnLine.
fitted.values numeric matrix with predicted values for training data X.
residuals numeric matrix with residuals values for training data X.
call call class object with training call.

Features
1. Membership Functions (MF) flexible framework:
   • Flexible user-defined membership functions(MF) extensible class.
   • Independent number of (MF) for each input.
   • Different MF types, if required, for each input.
2. Type 3 Takagi and Sugeno’s fuzzy if-then rule
3. Full Rule combinations, e.g. 2 inputs 2 membership functions this means that 4 fuzzy rules will be created.
4. Different learning strategies:
   trainHybridJangOffLine Hybrid learning, i.e. Descent Gradient for precedents and Least Squares Estimation for consequents.
**trainHybridJangOnLine** on-line version with hybrid learning.

**trainHybridOffLine** Adaptive learning coefficient and momentum term.

5. Multiple outputs support, i.e., the same input partition can be used to predict more than one output variable.

**Functions**

ANFIS S4 class includes the following functions:

- **initialize** constructor of ANFIS Architecture to generate the rule set and consequents
- **show/print** generic output of the object
- **getRules, getPremises, getConsequents, getErrors, getTrainingType** return the respective ANFIS slots
- **plotMF** plot MembershipFunctions domain
- **plotMFs** plot all the MembershipFunctions for the input domain
- **plot** plot training error according with training Type
- **LSE** auxiliary function for Least Square Estimation to avoid singular matrix system in off-line training

**trainHybridJangOffLine** Jang’s Hybrid off-line training

**trainHybridJangOnLine** Jang’s Hybrid on-line training

**trainHybridOffLine** Hybrid off-line training with momentum and adaptive learning rate

**summary, fitted, fitted.values, coef, coefficients, resid, residuals** wrappers for traditional model functions

**Author(s)**

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

BellMF-class, GaussianMF-class and NormalizedGaussianMF-class

Examples

```r
# Set 2 cores using global options for parallel library
require(parallel)
if(.Platform$OS.type == "windows"){
  options(mc.cores=1)
} else{
  options(mc.cores=2) # You could use all calling detectCores()
}

# Example domain for bidimensional sinc(x,y) function
x <- seq(-10, 10, length= 11)
trainingSet <- trainSet(x,x)
Z <- matrix(trainingSet[,"z"], ncol=length(x), nrow=length(x))
persp(x,x,Z,theta = 45, phi = 15, expand = 0.8, col = "lightblue",
ticktype="detailed",main="sinc(x)*sinc(y)")

# Training domain patterns
X <- trainingSet[,1:2]
Y <- trainingSet[,3,drop=FALSE]

# Defining the required MembershipFunctions for the ANFIS
membershipFunction<-list(
  x=c(new(Class="NormalizedGaussianMF",parameters=c(mu=-10,sigma=2)),
    new(Class="NormalizedGaussianMF",parameters=c(mu=-5,sigma=2)),
    new(Class="NormalizedGaussianMF",parameters=c(mu=0,sigma=2)),
    new(Class="NormalizedGaussianMF",parameters=c(mu=5,sigma=2)),
    new(Class="NormalizedGaussianMF",parameters=c(mu=10,sigma=2))),
  y=c(new(Class="NormalizedGaussianMF",parameters=c(mu=-10,sigma=2)),
    new(Class="NormalizedGaussianMF",parameters=c(mu=-5,sigma=2)),
    new(Class="NormalizedGaussianMF",parameters=c(mu=0,sigma=2)),
    new(Class="NormalizedGaussianMF",parameters=c(mu=5,sigma=2)),
    new(Class="NormalizedGaussianMF",parameters=c(mu=10,sigma=2))))

# Creating the ANFIS network with 2 inputs and 4 MembershipFunctions in
# each input
anfis3 <- new(Class="ANFIS",X,Y,membershipFunction)
anfis3

# Check for epsilon-completeness in each input
plotMFS(anfis3)

# Training the ANFIS network.
trainOutput <- trainHybridJangOffline(anfis3, epochs=10)
## We will use instead an already trained object to reduce example time.
data(anfis3)

## How the training went. You can keep on training as the training error
## is still descending.
plot(anfis3)

## Test the fit, i.e., how the MembershipFunctions partition the input space
plotMFS(anfis3)
```
Description

The example consist in learning of a bidimensional sinc(x,y) function using a regular grid of 121 points in the domain [-10,10]x[-10,10] and five independent Normalized Gaussian Membership Function (MF) for each input (x and y). The training process used the Hybrid off-line Jang`s strategy for 10 epochs.

Format

An ANFIS trained object for demonstration.

Details

Trainning Set  
• dim(x)= 121x2, the grid points.
  • dim(y)= 121x1, the sinc(x, y) output.

Architecture  2 (5x5) - 25 - 75 (75x1) - 1, i.e., 2 inputs with five MFs in each input, 25 rules and 75 consequents for the single output (75x1)

Last training error  0.01916307
BellMF-class

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

Source

see trainSet

See Also


BellMF-class

Bell Membership Function S4 class

Description

Represent a concrete Bell shaped Membership Function S4 class with parameters a, b, c. Slots inherited of MembershipFunction class and related functions: show, print, derivateMF, evaluateMF, [, and |<-.

Slots

parameters  named numeric vector with parameters of Membership Function.
nParameters  integer with the number of parameters for validity check.
name  character The description of the membership function.
expression  expression object just to display purposes.

Note

derivateMF, evaluateMF are extended. Prototype is defined and validity is inherited.
Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

GaussianMF-class and NormalizedGaussianMF-class

Other Membership Functions: GaussianMF, GaussianMF-class; MembershipFunction, MembershipFunction-method; NormalizedGaussianMF, NormalizedGaussianMF-class; extract-methods, extract-methods; derivateMF, derivateMF, derivateMF, derivateMF, derivateMF.


Examples

#BellMF example I
#A bell membership function with default prototype (a=1, b=1, c=0)
#The membership of x in the bell, should be 1
#The derivate of the first parameter at x, should be 0
#The derivate of the first parameter at x, should be also 0
bell <- new(Class="BellMF")
bell
evaluateMF(object=bell, x=0)
derivateMF(object=bell, x=0, i=1)
derivateMF(object=bell, x=0, i="a")
#
#BellMF example II
#A bell membership function with parameters (a=4, b=1, c=10)
#The membership of x in the bell, should be 0.137931
#The derivate of the first parameter at x, should be 0.05945303
#The derivate on "a" at x=0, should be 0.05945303
bell2 <- new(Class="BellMF", parameters=c(a=4,b=1,c=-10))
bell2
evaluateMF(object=bell2, x=0)
derivateMF(object=bell2, x=0, i=1)
derivateMF(object=bell2, x=0, i="a")

derivateMF
derivateMF derivative membership function

Description

Derivate de membership of x with respect to i of MembershipFunction object heirs.
derivateMF

Usage

```
derivateMF(object, x, i)
```

## S4 method for signature 'MembershipFunction'
derivateMF(object, x, i)

## S4 method for signature 'BellMF'
derivateMF(object, x, i)

## S4 method for signature 'GaussianMF'
derivateMF(object, x, i)

## S4 method for signature 'NormalizedGaussianMF'
derivateMF(object, x, i)

Arguments

- **object**: MembershipFunction class heirs
- **x**: numeric of the MembershipFunction to be evaluated
- **i**: index of the ith parameter to partially derivate

Value

numeric with the value obtained from the ith derivative at x

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <ALLera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

- MembershipFunction-class and evaluateMF
- Other Membership Functions: BellMF, BellMF-class, GaussianMF, GaussianMF-class, MembershipFunction, MembershipFunction-class, NormalizedGaussianMF, NormalizedGaussianMF-class, MembershipFunction-method, MembershipFunction-class-method, extract-methods, extract-methods: evaluateMF, evaluateMF, evaluateMF, evaluateMF, evaluateMF, BellMF-method, evaluateMF, GaussianMF-method, evaluateMF, MembershipFunction-method, evaluateMF, NormalizedGaussianMF-method, evaluateMF-methods; print, MembershipFunction-method; show, MembershipFunction-method

Examples

```
#BellMF example I
#A bell membership function with default prototype (a=1, b=1, c=0)
#The membership of x in the bell, should be 1
#The derivate of the first parameter at x, should be 0
#The derivate of the first parameter at x, should be also 0
bell <- new(Class="BellMF")
bell
```
evaluateMF(object=bell, x=0)
derivatemF(object=bell, x=0, i=1)
derivatemF(object=bell, x=0, i="a")
#
#BellMF example II
#A bell membership function with parameters (a=4,b=1,c=-10)
#The membership of x in the bell, should be 0.137931
#The derivate of the first parameter at x, should be 0.05945303
#The derivate on "a" at x=0, should be 0.05945303
bell2 <- new(Class="BellMF",parameters=c(a=4,b=1,c=-10))
bell2
evaluateMF(object=bell2, x=0)
derivatemF(object=bell2, x=0, i=1)
derivatemF(object=bell2, x=0, i="a")
#GaussianMF example I
#A Gaussian membership function with default prototype (mu=0, sigma=1)
#The membership of x in the gaussian, should be 1/sqrt(2*pi) = 0.3989423
#The derivate of the first parameter at x, should be 0
#The derivate on "mu" parameter at x, should be 0
 gaussian <- new(Class="GaussianMF")
gaussian
evaluateMF(object=gaussian, x=0)
derivatemF(object=gaussian, x=0, i=1)
derivatemF(object=gaussian, x=0, i="mu")
#
#GaussianMF example II
#A Gaussian membership function with parameters (mu=0, sigma=1)
#The membership of x in the Gaussian, should be 1/sqrt(2*pi) = 0.3989423
#The derivate of the first parameter at x, should be 0
#The derivate on "mu" parameter at x, should be 0
gaussian2 <- new(Class="GaussianMF",parameters=c(m=0,s=1))
gaussian2
evaluateMF(object=gaussian2, x=0)
derivatemF(object=gaussian2, x=0, i=1)
derivatemF(object=gaussian2, x=0, i="mu")
#NormalizedGaussianMF example I
#A normalized Gaussian membership function with default parameters (mu=0, sigma=1)
#The derivate of the first parameter at x, should be 1
#The derivate of the first parameter at x, should be 0
#The derivate on "mu" parameter at x, should be 0
normalizedGaussian <- new(Class="NormalizedGaussianMF")
normalizedGaussian
evaluateMF(object=normalizedGaussian, x=0)
derivatemF(object=normalizedGaussian, x=0, i=1)
derivatemF(object=normalizedGaussian, x=0, i="mu")
#
#NormalizedGaussianMF example II
#A normalized Gaussian membership function with parameters (mu=0, sigma=1)
#The derivate of the first parameter at x, should be 1
#The derivate of the first parameter at x, should be 0
#The derivate on "mu" parameter at x, should be 0
normalizedGaussian2 <- new(Class="NormalizedGaussianMF", parameters=c(m=0,s=1))
evaluateMF

evaluateMF(object=normalizedGaussian2, x=0)
derivatemf(object=normalizedGaussian2, x=0, i=1)
derivatemf(object=normalizedGaussian2, x=0, i="mu")

description

Evaluate de membership of x to the object MembershipFunction heirs.

Usage

evaluateMF(object, x)

## S4 method for signature 'MembershipFunction'
evaluateMF(object, x)

## S4 method for signature 'BellMF'
evaluateMF(object, x)

## S4 method for signature 'GaussianMF'
evaluateMF(object, x)

## S4 method for signature 'NormalizedGaussianMF'
evaluateMF(object, x)

Arguments

object MembershipFunction class heirs
x numeric of the MembershipFunction to be evaluated

Value

0 <= numeric <=1 with the obtained membership value

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>
evaluateMF

See Also

MembershipFunction-class and derivateMF

Other Membership Functions: BellMF, BellMF-class, GaussianMF, GaussianMF-class, MembershipFunction, MembershipFunction-class, NormalizedGaussianMF, NormalizedGaussianMF-class; [,MembershipFunction-method, derivateMF, derivateMF, derivateMF, derivateMF, derivateMF, derivateMF, BellMF-method, derivateMF, GaussianMF-method, derivateMF, MembershipFunction-method, derivateMF, NormalizedGaussianMF-method, derivateMF-methods; print, MembershipFunction-method; show, MembershipFunction-method

Examples

#BellMF example I
#A bell membership function with default prototype (a=1, b=1, c=0)
#The membership of x in the bell, should be 1
#The derivate of the first parameter at x, should be 0
#The derivate of the first parameter at x, should be also 0
bell <- new(Class="BellMF")
bell
evaluateMF(object=bell, x=0)
derivateMF(object=bell, x=0, i=1)
derivateMF(object=bell, x=0, i="a")
#
#BellMF example II
#A bell membership function with parameters (a=4,b=1,c=-10)
#The membership of x in the bell, should be 0.137931
#The derivate of the first parameter at x, should be 0.05945303
#The derivate on "a" at x=0, should be 0.05945303
bell2 <- new(Class="BellMF", parameters=c(a=4,b=1,c=-10))
bell2
evaluateMF(object=bell2, x=0)
derivateMF(object=bell2, x=0, i=1)
derivateMF(object=bell2, x=0, i="a")
#GaussianMF example I
#A Gaussian membership function with default prototype (mu=0, sigma=1)
#The membership of x in the Gaussian, should be 1/sqrt(2*pi) = 0.3989423
#The derivate of the first parameter at x, should be 0
#The derivate on "mu" parameter at x, should be 0
gaussian <- new(Class="GaussianMF")
gaussian
evaluateMF(object=gaussian, x=0)
derivativeMF(object=gaussian, x=0, i=1)
derivativeMF(object=gaussian, x=0, i="mu")
#
#GaussianMF example II
#A Gaussian membership function with parameters (mu=0, sigma=1)
#The membership of x in the gaussian, should be 1/sqrt(2*pi) = 0.3989423
#The derivate of the first parameter at x, should be 0
#The derivate on "mu" parameter at x, should be 0
gaussian2 <- new(Class="GaussianMF", parameters=c(mu=0,sigma=1))
gaussian2
evaluateMF(object=gaussian2, x=0)
extract-methods

Modify membership function parameters

Description

Get/set membership function parameters.

Usage

```r
## S4 method for signature 'MembershipFunction'
x[i]
```

```r
## S4 replacement method for signature 'MembershipFunction'
x[i] <- value
```

Arguments

- `x`: MembershipFunction class heirs
- `i`: numeric or character to access parameters vector [i]
- `value`: numeric parameter/s values

Value

- `numeric`: parameter/s in the case of object[i]
- `object`: MembershipFunction object in the case of object[i] <- value
See Also

_MembershipFunction-class_

Other Membership Functions: _BellMF, BellMF-class, GaussianMF, GaussianMF-class, MembershipFunction, MembershipFunction-class, NormalizedGaussianMF, NormalizedGaussianMF-class, derivateMF, derivateMF, derivateMF, derivateMF, derivateMF, derivateMF, BellMF-method, derivateMF, GaussianMF-method, derivateMF, MembershipFunction-method, derivateMF, NormalizedGaussianMF-method, derivateMF-methods: evaluateMF, evaluateMF, evaluateMF, evaluateMF, evaluateMF, evaluateMF, BellMF-method, evaluateMF, GaussianMF-method, evaluateMF, MembershipFunction-method, evaluateMF, NormalizedGaussianMF-methods: print, MembershipFunction-method; show, MembershipFunction-method_

---

## fitted

**ANFIS training results**

---

### Description

Obtain ANFIS slot information, according to training output

### Usage

```r
## S4 method for signature 'ANFIS'
fitted(object, ...)

## S4 method for signature 'ANFIS'
fitted.values(object, ...)

## S4 method for signature 'ANFIS'
coef(object, ...)

## S4 method for signature 'ANFIS'
coefficients(object, ...)

## S4 method for signature 'ANFIS'
resid(object, ...)

## S4 method for signature 'ANFIS'
residuals(object, ...)

## S4 method for signature 'ANFIS'
summary(object, ...)
```

### Arguments

- **object**: ANFIS class object
- **...**: required by resid, residuals, coef and coefficients
GaussianMF-class

Value

according to the call one of the following objects can be returned

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>list with premises and consequents.</td>
</tr>
<tr>
<td>numeric</td>
<td>numeric vector with training errors, fitted training values and residuals.</td>
</tr>
<tr>
<td>printed</td>
<td>statistics of the training process.</td>
</tr>
</tbody>
</table>

Note

see full example in ANFIS-class

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <Allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also


GaussianMF-class GaussianMF Membership Function S4 class

Description

Represent a concrete GaussianMF shaped Membership Function S4 class with parameters mu, sigma. Slots inherited of MembershipFunction class and related functions: show, print, derivateMF, evaluateMF, [ and [<. 

Slots

parameters named numeric vector with parameters of Membership Function. 
nParameters integer with the number of parameters for validity check. 
name character The description of the membership function. 
expression expression object just to display purposes.
Note

derivativeMF, evaluateMF are extended. Prototype is defined and validity is inherited.

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

BellMF-class and NormalizedGaussianMF-class

Other Membership Functions: BellMF, BellMF-class; MembershipFunction, MembershipFunction-class; NormalizedGaussianMF, NormalizedGaussianMF-class; [], MembershipFunction-method, [<=, MembershipFunction-method, extract-methods, extract-methods; derivativeMF, derivativeMF, derivativeMF, derivativeMF, derivativeMF, derivativeMF, derivateM, BellMF-method, derivateMF, GaussianMF-method, derivateMF, MembershipFunction-method, derivateMF, NormalizedGaussianMF-method, derivateMF-methods; evaluateMF, evaluateMF, evaluateMF, evaluateMF, evaluateMF, BellMF-method, evaluateMF, GaussianMF-method, evaluateMF, MembershipFunction-method, evaluateMF, NormalizedGaussianMF-method, evaluateMF-methods; print, MembershipFunction-method; show, MembershipFunction-method

Examples

# GaussianMF example I
# A Gaussian membership function with default prototype (mu=0, sigma=1)
# The membership of x in the Gaussian, should be 1/sqrt(2*pi) = 0.3989423
# The derivate of the first parameter at x, should be 0
# The derivate on "mu" parameter at x, should be 0
gaussian <- new(Class="GaussianMF")
geussian
evaluateMF(object=gaussian, x=0)
derivativeMF(object=gaussian, x=0, i=1)
derivativeMF(object=gaussian, x=0, i="mu")
#
# GaussianMF example II
# A Gaussian membership function with parameters (mu=0, sigma=1)
# The membership of x in the Gaussian, should be 1/sqrt(2*pi) = 0.3989423
# The derivate of the first parameter at x, should be 0
# The derivate on "mu" parameter at x, should be 0
gaussian2 <- new(Class="GaussianMF", parameters=c(mu=0, sigma=1))
geussian2
evaluateMF(object=gaussian2, x=0)
derivativeMF(object=gaussian2, x=0, i=1)
derivativeMF(object=gaussian2, x=0, i="mu")
getRules

Getters for ANFIS object

Description

Obtain ANFIS’s slot information, according to the given function call.

Usage

getRules(object)

## S4 method for signature 'ANFIS'
getRules(object)

getPremises(object)

## S4 method for signature 'ANFIS'
getPremises(object)

getConsequents(object)

## S4 method for signature 'ANFIS'
getConsequents(object)

getErrors(object)

## S4 method for signature 'ANFIS'
getErrors(object)

getTrainingType(object)

## S4 method for signature 'ANFIS'
getTrainingType(object)

Arguments

object   ANFIS class object

Value

according to the call one of the following objects can be returned

matrix   numeric matrix with rules or consequents
list     list with MembershipFunctions or premises and consequents
character name of the trainingType
numeric   numeric vector with training errors, fitted training values and residuals
Note

see full example in ANFIS-class

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <Altera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also


initialize initialize ANFIS object constructor

Description

Create the ANFIS object architecture for the trainingSet (X,Y) with full rules.

Usage

## S4 method for signature 'ANFIS'
initialize(.Object, X, Y, membershipFunction)

Arguments

- `.Object` ANFIS class
- `X` input matrix with `ncol=#inputs and nrow=#individuals`
- `Y` output matrix with `ncol=#output and nrow=#individuals`
- `membershipFunction` list with the MembershipFunction for each input

Value

ANFIS object
Note

see full example in ANFIS-class

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

ANFIS-class


LSE

Train ANFIS network

Description

ANFIS on-line or off-line hybrid Jang dynamic learning training process. In addition for off-line learning there is also adaptive learning coefficient and momentum term.

Usage

LSE(object, A, B, initialGamma = 1000)

## S4 method for signature 'ANFIS'
LSE(object, A, B, initialGamma = 1000)

trainHybridJangOffLine(object, epochs = 5, tolerance = 1e-05, initialGamma = 1000, k = 0.01)

## S4 method for signature 'ANFIS'
trainHybridJangOffLine(object, epochs = 5, tolerance = 1e-05, initialGamma = 1000, k = 0.01)
trainHybridOffline(object, epochs = 5, tolerance = 1e-05,
    initialGamma = 1000, eta = 0.05, phi = 0.2, a = 0.01, b = 0.1,
    delta_alpha_t_1 = list())

# S4 method for signature 'ANFIS'
trainHybridOffline(object, epochs = 5, tolerance = 1e-05,
    initialGamma = 1000, eta = 0.05, phi = 0.2, a = 0.01, b = 0.1,
    delta_alpha_t_1 = list())

trainHybridJangOnLine(object, epochs = 5, tolerance = 1e-15,
    initialGamma = 1000, k = 0.01, lamda = 0.9, S = matrix(nrow = 0, ncol
    = 0))

# S4 method for signature 'ANFIS'
trainHybridJangOnLine(object, epochs = 5,
    tolerance = 1e-15, initialGamma = 1000, k = 0.01, lamda = 0.9,
    S = matrix(nrow = 0, ncol = 0))

Arguments

object          ANFIS’ class object.
A               internal matrix for Iterative Least Squares Estimation of AX=B.
B               internal matrix for Iterative Least Squares Estimation of AX=B.
initialGamma   numeric large number >= 0. Default 1000.
ePOCHS          the max number of training epochs. Default 5.
tolerance      convergence error to stop training. Default 1e-5.
k               numeric with the initial step size for learning rule. Default 0.01.
etta             numeric learning rule coefficient. Default 0.05.
phi             numeric momentum rule coefficient. Default 0.2.
a               numeric step to increase eta if delta_e is < 0, i.e. descending. Default value 0.01.
b               numeric fraction to decrease eta if delta_e is > 0, i.e. ascending. Default value is 0.1.
delta_alpha_t_1 list with numeric matrix with last time step. Default list().
lamda           0 < numeric < 1 forgetting factor. Default 0.9.
S               covariance matrix for on-line LSE. Default matrix(nrow=0, ncol=0).

Value

matrix          with the system solution for LSE output.
error           numeric vector with training associated errors (pattern or epoch) according to trainingType.
convergence     TRUE/FALSE if it reached convergence or not.
updated          trainingType, premises, consequents, error, residuals, fitted.values and coefficient.
MembershipFunction-class

Note

see full example in ANFIS-class

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

ANFIS-class

Other ANFIS: ANFIS-class: anfis3; coef, coef, ANFIS-method, coefficients, coefficients, ANFIS-method, fitted, fitted, ANFIS-method, fitted.values, fitted.values, ANFIS-method, resid, resid, ANFIS-method, residuals, residuals, ANFIS-method, summary, summary, ANFIS-method; getConsequents, getConsequents, ANFIS-method, getErrors, getErrors, getErrors, ANFIS-method, getErrors, ANFIS-method, getPremises, getPremises, getPremises, getPremises, ANFIS-method, getPremises-methods, getRules, getRules, getRules, ANFIS-method, getRules-methods, getTrainingType, getTrainingType, getTrainingType, ANFIS-method, getTrainingType, ANFIS-method; initialize, initialize, ANFIS-method; plotMF, plotMF, plotMF, ANFIS-method, plotMF-methods, plotMFs, plotMFs, plotMFs, ANFIS-method, plotMFs-methods; plot, plot, ANFIS-method; predict, predict, ANFIS-method; print, print, ANFIS-method, show, show, ANFIS-method; trainSet

MembershipFunction-class

MembershipFunction S4 class

Description

Represent a generic virtual S4 MembershipFunction class, for fuzzy further redefinition. The actual subclasses available are GaussianMF, NormalizedGaussianMF and BellMF.

Slots

parameters named numeric vector with parameters of Membership Function.

nParameters integer with the number of parameters for validity check.

name character The description of the membership function.

eexpression expression object just to display purposes.

Functions

MembershipFunction S4 class includes the following functions:

show/print generic output of the object.

"[", "<-" getter and setter of the parameters values.

evaluateMF return membership value at x.

derivateMF return the derivate membership at x.
NormalizedGaussianMF-class

Description

Represent a concrete NormalizedGaussianMF shaped [0,1] Membership Function S4 class with parameters mu, sigma. Slots inherited of MembershipFunction class and related functions: show, print, derivateMF, evaluateMF, [ and [<-.

Slots

parameters named numeric vector with parameters of Membership Function.

nParameters integer with the number of parameters for validity check.

name character The description of the membership function.

expression expression object just to display purposes.

Note

derivateMF, evaluateMF are extended. Prototype is defined and validity is inherited.

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <ALLlera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>
plot

See Also

BellMF-class and GaussianMF-class

Other Membership Functions: BellMF, BellMF-class; GaussianMF, GaussianMF-class; MembershipFunction,
MembershipFunction-class; [, MembershipFunction-method, [, MembershipFunction-method,
extract-methods, extract-methods; derivateMF, derivateMF, derivateMF, derivateMF, derivateMF,
derivateMF, BellMF-method, derivateMF, GaussianMF-method, derivateMF, MembershipFunction-method,
derivateMF, NormalizedGaussianMF-method, derivateMF-methods; evaluateMF, evaluateMF,
evaluateMF, evaluateMF, evaluateMF, evaluateMF, BellMF-method, evaluateMF, GaussianMF-method,
evaluateMF, MembershipFunction-method, evaluateMF, NormalizedGaussianMF-method, evaluateMF-methods;
print, MembershipFunction-method; show, MembershipFunction-method

Examples

# NormalizedGaussianMF example I
# A normalized Gaussian membership function with default parameters (mu=0, sigma=1)
# The derivate of the first parameter at x, should be 1
# The derivate of the first parameter at x, should be 0
# The derivate on "mu" parameter at x, should be 0
normalizedGaussian <- new(Class="NormalizedGaussianMF"
normalizedGaussian
evaluateMF(object=normalizedGaussian, x=0)
derivateMF(object=normalizedGaussian, x=0, i=1)
derivateMF(object=normalizedGaussian, x=0, i="mu")
#
# NormalizedGaussianMF example II
# A normalized Gaussian membership function with parameters (mu=0, sigma=1)
# The derivate of the first parameter at x, should be 1
# The derivate of the first parameter at x, should be 0
# The derivate on "mu" parameter at x, should be 0
normalizedGaussian2 <- new(Class="NormalizedGaussianMF",
parameters=c(mu=0, sigma=1))
normalizedGaussian2
evaluateMF(object=normalizedGaussian2, x=0)
derivateMF(object=normalizedGaussian2, x=0, i=1)
derivateMF(object=normalizedGaussian2, x=0, i="mu")

plot

Plot ANFIS training errors

Description

Plot the training error of the network. If trainingType is "on-line" then full pattern errors along the patterns of the whole training process; for a specific epoch or the epoch summary error.

Usage

## S4 method for signature 'ANFIS'
plot(x, y, epoch = Inf, ...)

plot
**Arguments**

- **x**: ANFIS class object.
- **y**: not used but necessary for redefining the generic function.
- **epoch**: for on-line only: epoch == Inf the whole training error; epoch == integer > 0 the give epoch trainings errors, epoch == 0 the abs epoch training sum of errors.
  
  ... plot additional parameters.

**Value**

output graphics.

**Note**

see full example in ANFIS-class

**Author(s)**

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

**See Also**

ANFIS-class


---

**plotMF**

*PlotMF/s ANFIS' MembershipFunction domain/s*

**Description**

Plot the corresponding MembershipFunctions for each/all input/s domain.
Usage

plotMF(object, x, input, ...)

## S4 method for signature 'ANFIS'
plotMF(object, x, input, ...)

plotMFs(object, ...)

## S4 method for signature 'ANFIS'
plotMFs(object, ...)

Arguments

object ANFIS class object.

x numeric sequence to evaluate each MembershipFunction.

input integer with the input MembershipFunctions to plot.

... plot additional parameters.

Value

output graphics

Note

see full example in ANFIS-class

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <Allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

ANFIS-class

predict

Predict ANFIS' network output

Description

Forward Pass to predict the ANFIS' output

Usage

```r
## S4 method for signature 'ANFIS'
predict(object, x)
```

Arguments

- `object` : ANFIS class object.
- `x` : numeric matrix [patterns x inputs] of input patterns.

Value

matrix with the output values

Note

see full example in `ANFIS-class`

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <ALLera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

`ANFIS-class`

Description

Generic Print/Show Method for ANFIS class output visualization.

Usage

```r
## S4 method for signature 'ANFIS'
print(x, ...)

## S4 method for signature 'ANFIS'
show(object)
```

Arguments

- `x` ANFIS class object
- `...` not used but included for generic print compatibility
- `object` ANFIS class object

Value

Console output of the object

Note

See full example in `ANFIS-class`

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <Allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

`ANFIS-class`


print,MembershipFunction-method

Print a MembershipFunction object

Description

Generic Print Method for MembershipFunction class and descendants. Usage: print(x, ...)

Usage

## S4 method for signature 'MembershipFunction'
print(x, ...)

Arguments

x MembershipFunction class object

... not used but included for generic print compatibility

Value

console output of the object

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <ALLlera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

MembershipFunction-class

Other Membership Functions: BellMF, BellMF-class, GaussianMF, GaussianMF-class, MembershipFunction, MembershipFunction-class, NormalizedGaussianMF, NormalizedGaussianMF-class; [,MembershipFunction-method, MembershipFunction-method, extract-methods, extract-methods; derivateMF, derivateMF, derivateMF, derivateMF, derivateMF, BellMF-method, derivateMF, GaussianMF-method, derivateMF, MembershipFunction-method, derivateMF, NormalizedGaussianMF-method, derivateMF-methods; evaluateMF, evaluateMF, evaluateMF, evaluateMF, evaluateMF, evaluateMF, BellMF-method, evaluateMF, GaussianMF-method, evaluateMF, MembershipFunction-method, evaluateMF, NormalizedGaussianMF-method, evaluateMF-methods; show, MembershipFunction-method
show.MembershipFunction-method

Show a MembershipFunction object

Description

Generic display method for MembershipFunction class and its descendants. Usage: show(object)

Usage

```r
## S4 method for signature 'MembershipFunction'
show(object)
```

Arguments

- **object**: MembershipFunction class object

Value

console output of the object

Author(s)

Cristobal Fresno <cfresno@bdmg.ar.com>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also

- `MembershipFunction-class`

Description

Generates the training set of sinc(x)*sinc(y) for the (x,y) regular grid

Usage

trainSet(x, y)

Arguments

x numeric vector with the x-th grid coordinates
y numeric vector with the x-th grid coordinates

Value

matrix numeric matrix with the columns x, y and z=sinc(x,y)

Author(s)

Cristobal Fresno <cfresno@bdmg.com.ar>, Andrea S. Llera <allera@leloir.org.ar> and Elmer A. Fernandez <efernandez@bdmg.com.ar>

See Also


Examples

# Domain definition for a regular (x,y) grid with 11 points for each coordinate
x <- seq(-10, 10, length= 11)
trainingSet <- trainSet(x,x)
\[ Z \leftarrow \text{matrix}(\text{trainingSet[,"z"], ncol=\text{length}(x), nrow=\text{length}(x)}) \]

# Plot the domain
persp(x, x, Z, theta=45, phi=15, expand=0.8, col="lightblue",
ticktype="detailed", main="sinc(x)*sinc(y)"
Index

*Topic ANFIS
  Anfis-package, 2
*Topic fuzzy
  Anfis-package, 2
*Topic membership
  Anfis-package, 2
  [,MemberShipFunction-method
    (extract-methods), 13
  []- MemberShipFunction-method
    (extract-methods), 13

ANFIS-class, 3
Anfis-package, 2
anfis3, 4, 6, 13, 18, 19, 21, 24–27, 30
BellMF, 9, 12, 14, 16, 22, 23, 28, 29
BellMF (BellMF-class), 7
BellMF-class, 7
coef, 4, 7, 18, 19, 21, 24–27, 30
coef (fitted), 14
coef, ANFIS-method (fitted), 14
coefficients, 4, 7, 18, 19, 21, 24–27, 30
coefficients (fitted), 14
coefficients, ANFIS-method (fitted), 14
derivateMF, 8, 8, 12, 14, 16, 22, 23, 28, 29
derivateMF, BellMF-method (derivateMF), 8
derivateMF, GaussianMF-method (derivateMF), 8
derivateMF, MemberShipFunction-method (derivateMF), 8
derivateMF, NormalizedGaussianMF-method (derivateMF), 8
derivateMF-methods (derivateMF), 8
evaluateMF, 8, 9, 11, 14, 16, 22, 23, 28, 29
evaluateMF, BellMF-method (evaluateMF), 11
evaluateMF, GaussianMF-method (evaluateMF), 11
evaluateMF, MemberShipFunction-method (evaluateMF), 11
evaluateMF, NormalizedGaussianMF-method (evaluateMF), 11
evaluateMF-methods (evaluateMF), 11
fitted, 4, 7, 14, 18, 19, 21, 24–27, 30
fitted, ANFIS-method (fitted), 14
fitted.values, 4, 7, 18, 19, 21, 24–27, 30
fitted.values (fitted), 14
fitted.values, ANFIS-method (fitted), 14
GaussianMF, 8, 9, 12, 14, 22, 23, 28, 29
GaussianMF (GaussianMF-class), 15
GaussianMF-class, 15
getConsequents, 4, 7, 15, 19, 21, 24–27, 30
getConsequents (getRules), 17
getConsequents, ANFIS-method (getRules), 17
getErrors, 4, 7, 15, 19, 21, 24–27, 30
getErrors (getRules), 17
getErrors, ANFIS-method (getRules), 17
getPremises, 4, 7, 15, 19, 21, 24–27, 30
getPremises (getRules), 17
getPremises, ANFIS-method (getRules), 17
getPremises-methods (getRules), 17
getRules, 4, 7, 15, 19, 21, 24–27, 30
getRules, ANFIS-method (getRules), 17
getRules-methods (getRules), 17
getTrainingType, 4, 7, 15, 19, 21, 24–27, 30
getTrainingType (getRules), 17
getTrainingType, ANFIS-method (getRules), 17
initialize, 4, 7, 15, 18, 18, 21, 24–26, 28, 30
initialize, ANFIS-method (initialize), 18
LSE, 4, 7, 15, 18, 19, 24–27, 30
LSE, ANFIS-method (LSE), 19
LSE-methods (LSE), 19

MembershipFunction, 8, 9, 12, 14, 16, 23, 28, 29
MembershipFunction
(MembershipFunction-class), 21
MembershipFunction-class, 21

NormalizedGaussianMF, 8, 9, 12, 14, 16, 22, 28, 29
NormalizedGaussianMF
(NormalizedGaussianMF-class), 22
NormalizedGaussianMF-class, 22

plot, 4, 7, 15, 18, 19, 21, 23, 25, 26, 28, 30
plot, ANFIS-method (plot), 23
plotMF, 4, 7, 15, 18, 19, 21, 24, 26, 28, 30
plotMF, ANFIS-method (plotMF), 24
plotMF-methods (plotMF), 24
plotMFs, 4, 7, 15, 18, 19, 21, 24, 26, 28, 30
plotMFs (plotMF), 24
plotMFs, ANFIS-method (plotMF), 24
plotMFs-methods (plotMF), 24
predict, 4, 7, 15, 18, 19, 21, 24, 25, 26, 28, 30
predict, ANFIS-method (predict), 26
print, 4, 7, 15, 18, 19, 21, 24–26, 27, 30
print, ANFIS-method (print), 27
print, MembershipFunction-method, 28

resid, 4, 7, 18, 19, 21, 24–27, 30
resid (fitted), 14
resid, ANFIS-method (fitted), 14
residuals, 4, 7, 18, 19, 21, 24–27, 30
residuals (fitted), 14
residuals, ANFIS-method (fitted), 14

show, 4, 7, 15, 18, 19, 21, 24–26, 30
show (print), 27
show, ANFIS-method (print), 27
show, MembershipFunction-method, 29
summary, 4, 7, 18, 19, 21, 24–27, 30
summary (fitted), 14
summary, ANFIS-method (fitted), 14

trainHybridJangOffline, 4, 7, 15, 18, 19, 24–27, 30
trainHybridJangOffline (LSE), 19
trainHybridJangOffline, ANFIS-method (LSE), 19