Chapter 10

Neuro-Fuzzy Systems

10.1. Similitude between the RBFNN and the TSK fuzzy systems

10.2. The ANFIS architecture

10.3. Neuro-Fuzzy systems type Mamdani

10.4. ANFIS in Matlab

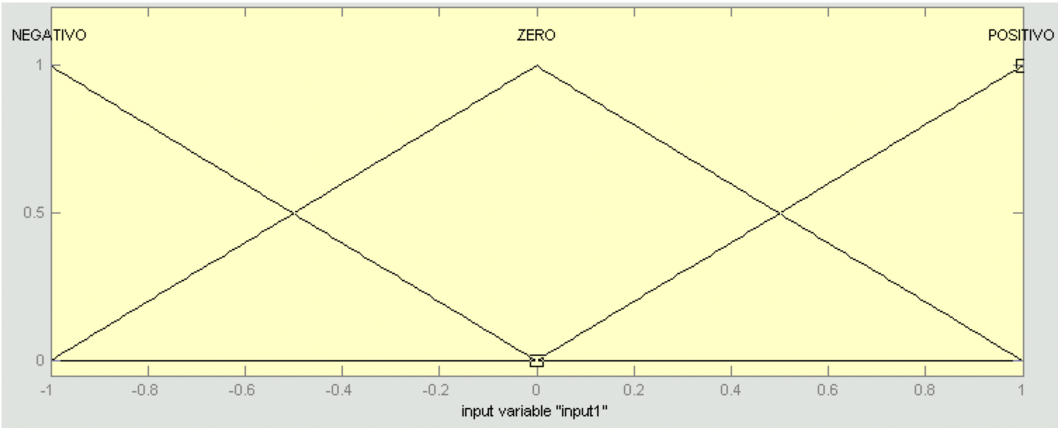
10.5. Fuzzy c-means clustering

10.6. Derivation of the rules from the fuzzy partitions

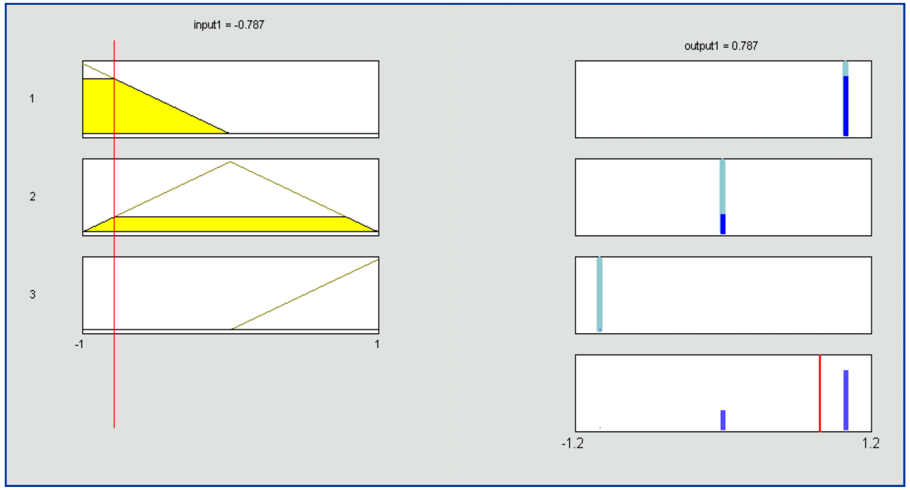
10.7. Derivation f the rules from subtractive clustering

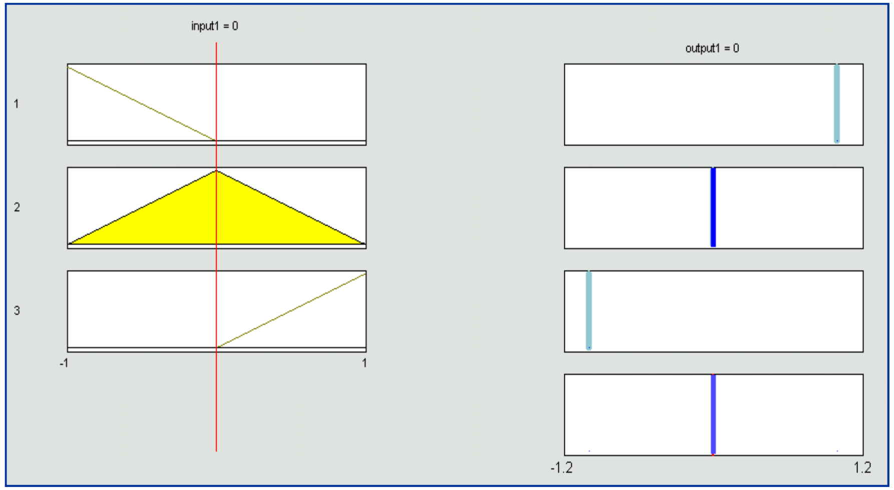
10.8. Conclusions

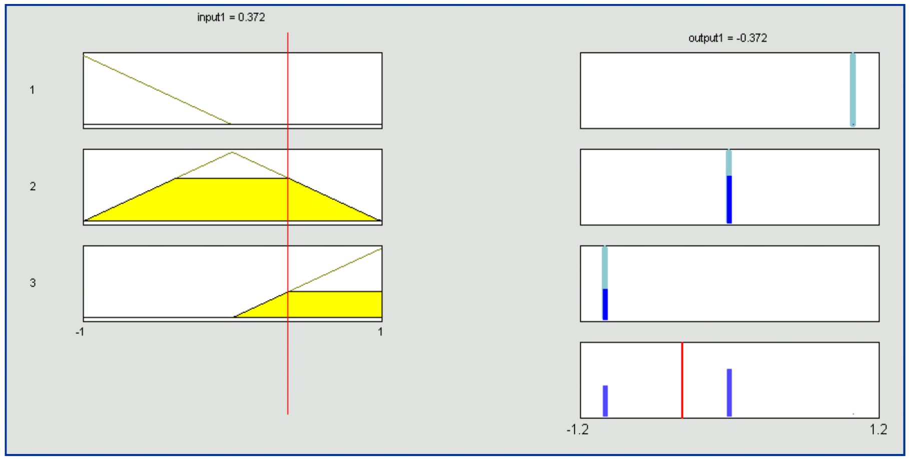
10.1. Similitude between the RBFNN and the zero order TSK systems

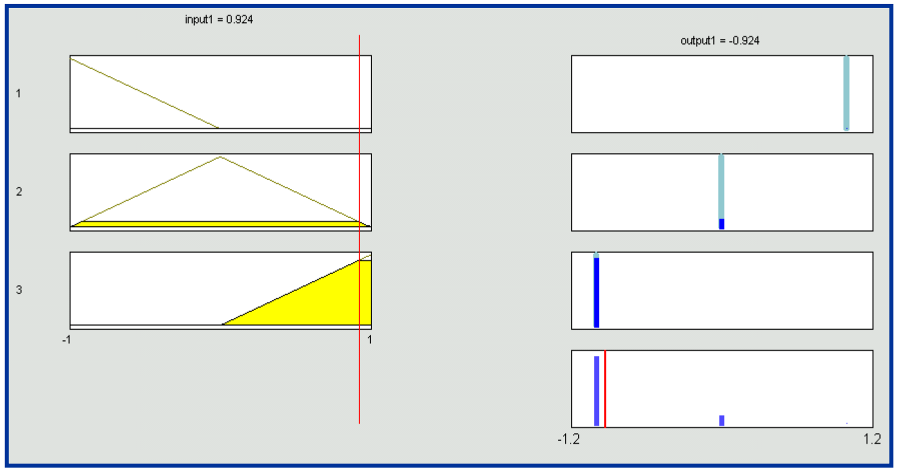


1. IF p is NEGATIVE THEN y is 1
2. IF p is ZERO THEN y is 0
3. IF p is POSITIVE THEN y is -1

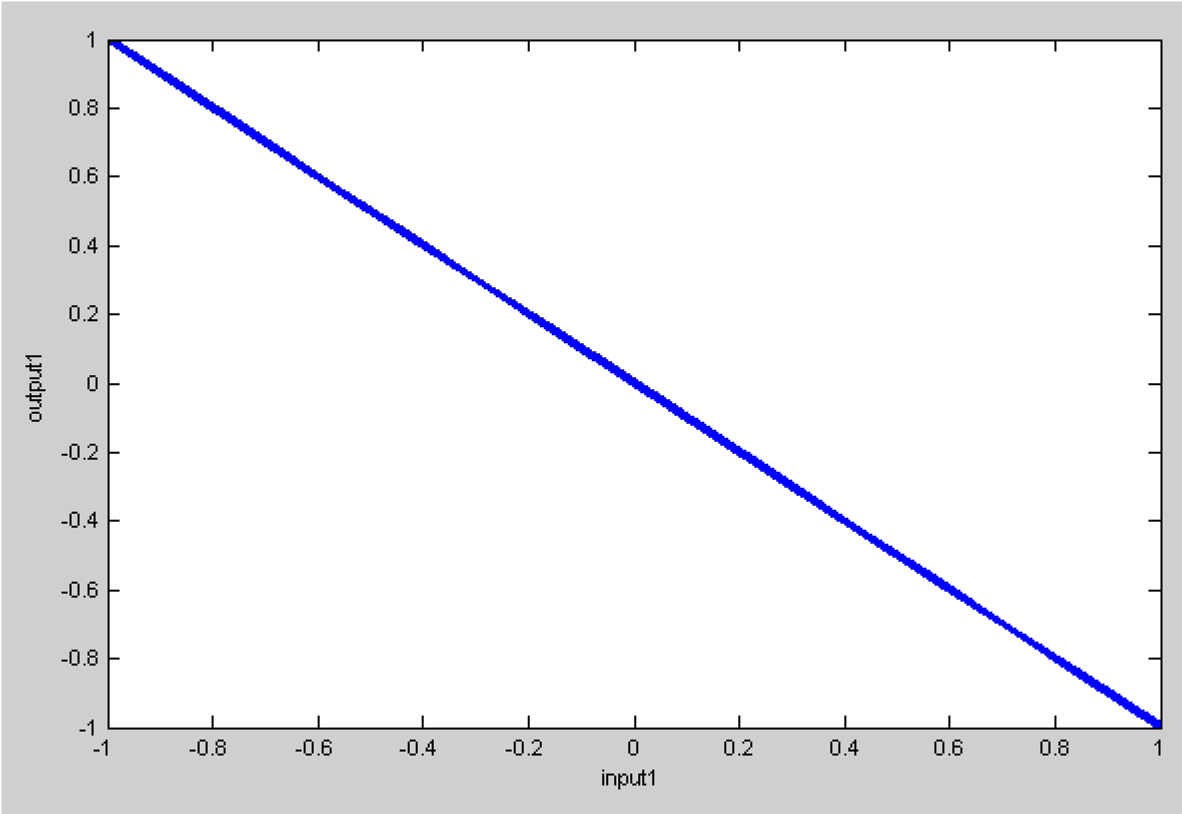

input1 = -0.787
output1 = 0.787
1
2
3
-1
1
-1.2
1.2


input1 = 0
output1 = 0
1
2
3
-1
1
-1.2
1.2


input1 = 0.372
output1 = -0.372
1
2
3
-1
1
-1.2
1.2

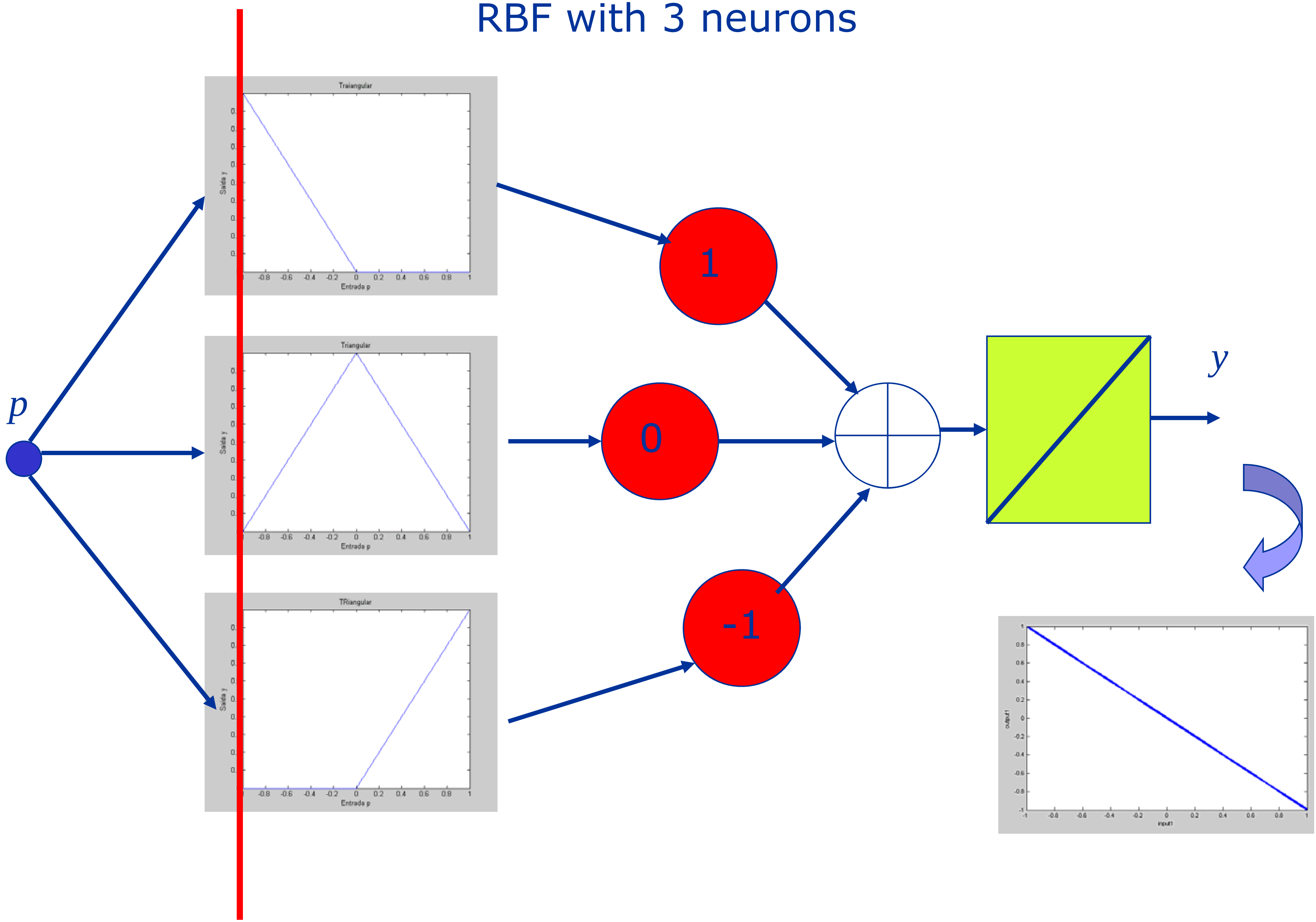

input1 = 0.924
output1 = -0.924
1
2
3
-1
1
-1.2
1.2

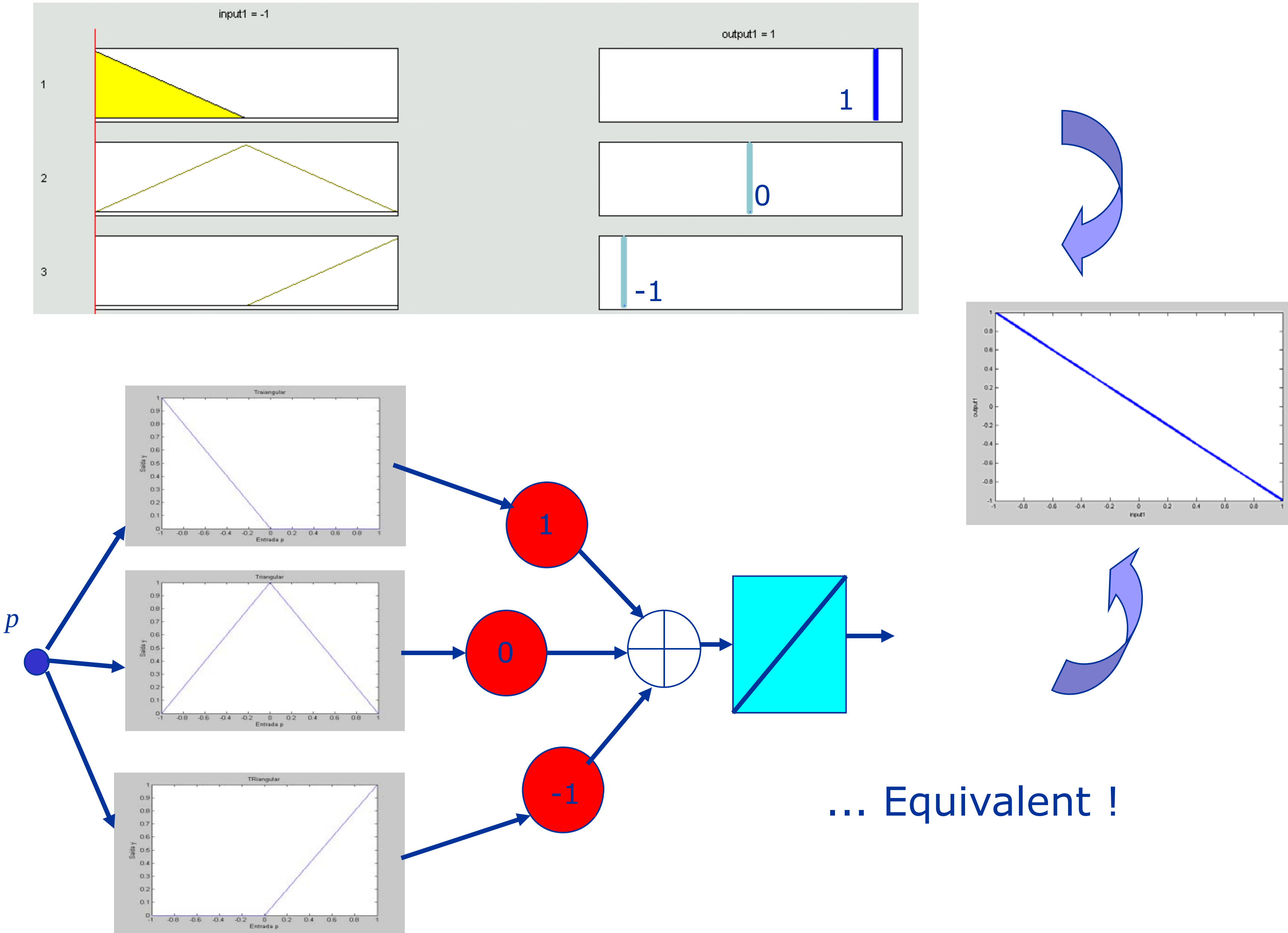
Input-output relation (*view surface*)



Note the (almost) perfect approximation of the function $y = -p$

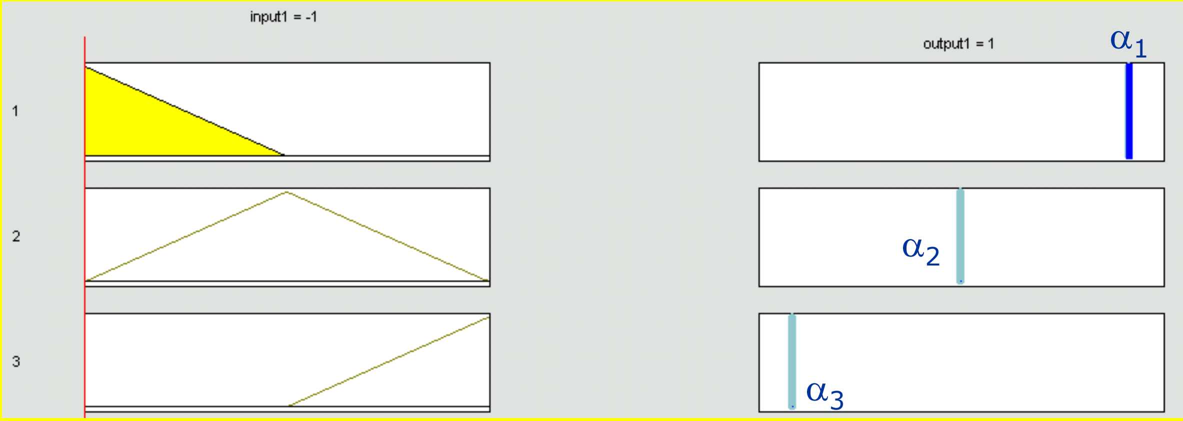
RBF with 3 neurons

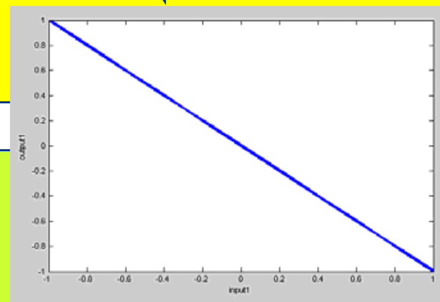


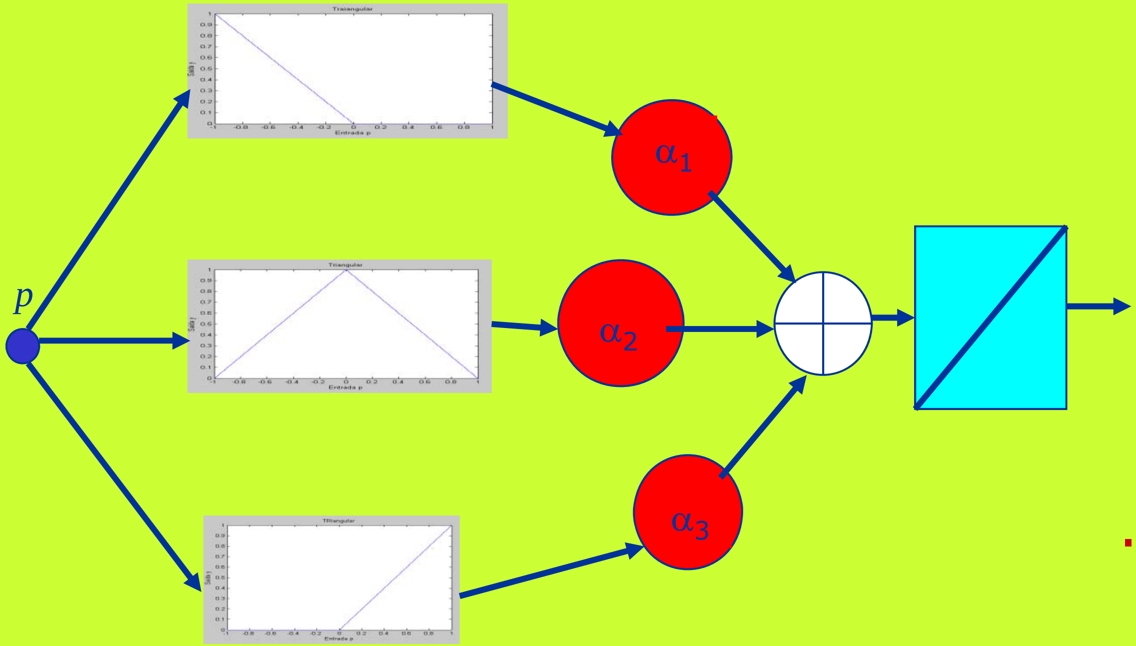


... Equivalent !

1. IF p is NEGATIVE THEN y is α_1
2. IF p is ZERO THEN y is α_2
3. IF p is POSITIVE THEN y is α_3

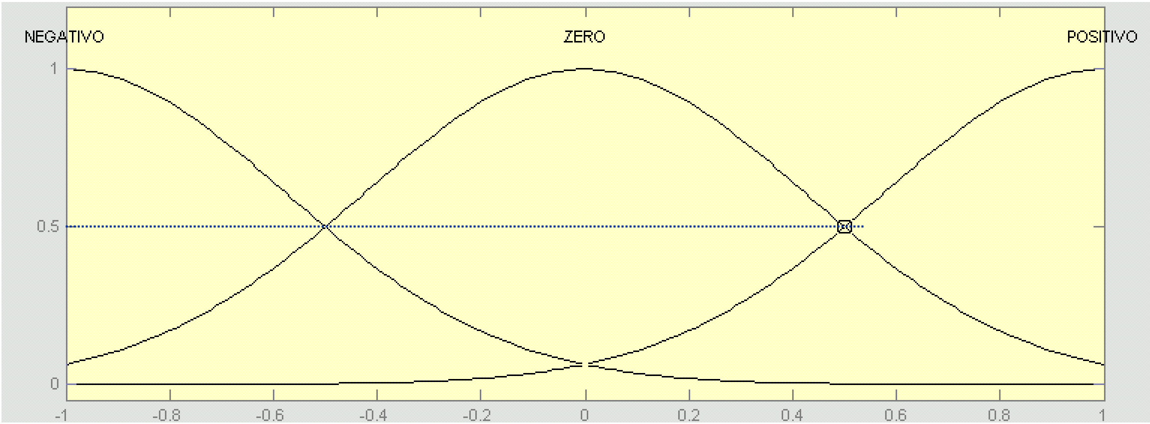






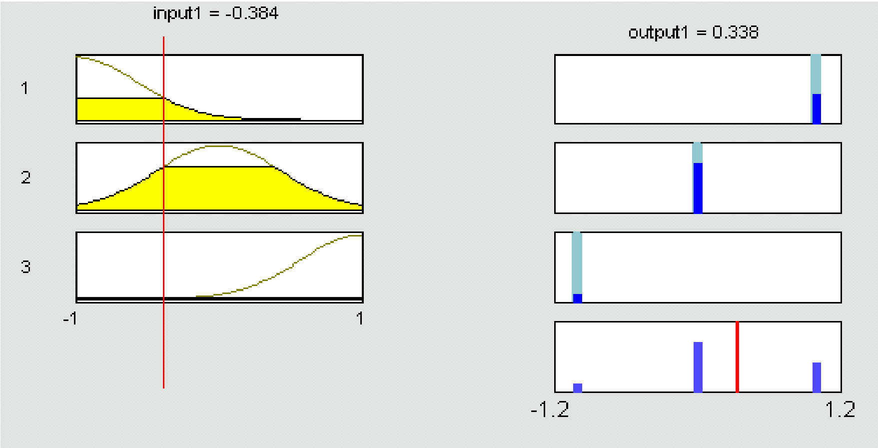
... Equivalent !

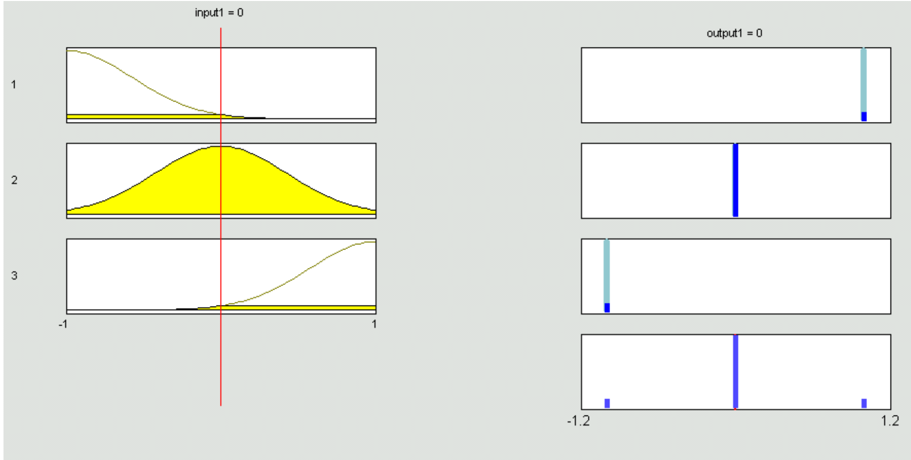
Case of Gaussian functions

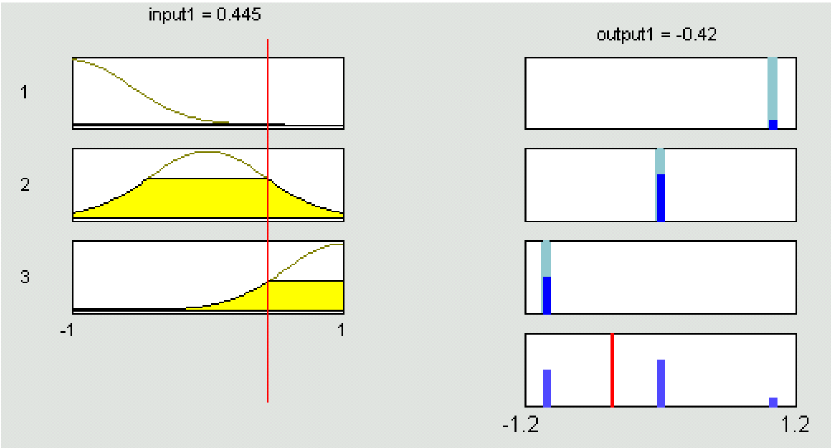


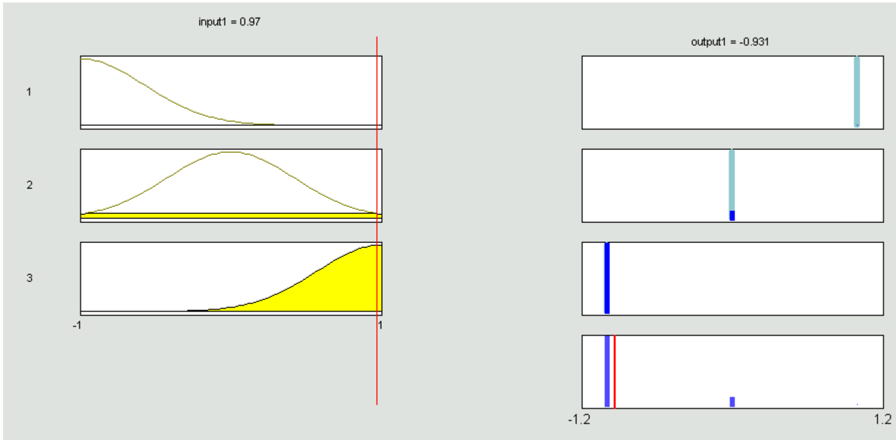
[SIGMA, c]=[0.4246 1]

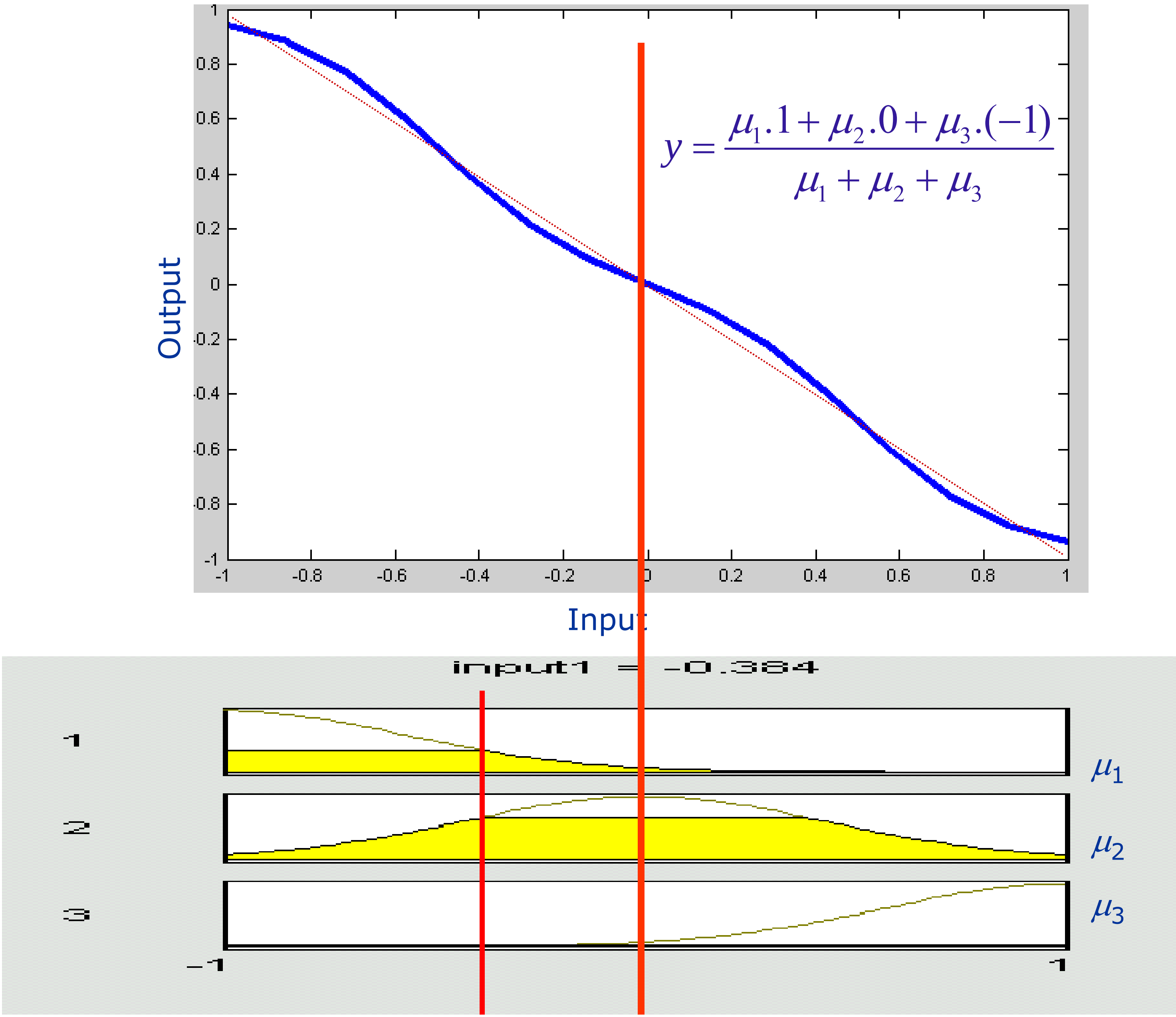
1. IF p is NEGATIVE THEN y is *1*
2. IF p is ZERO THEN y is 0
3. IF p is POSITIVE THEN y is -1


input1 = -0.384
output1 = 0.338
1
2
3
-1
1
-1.2
1.2

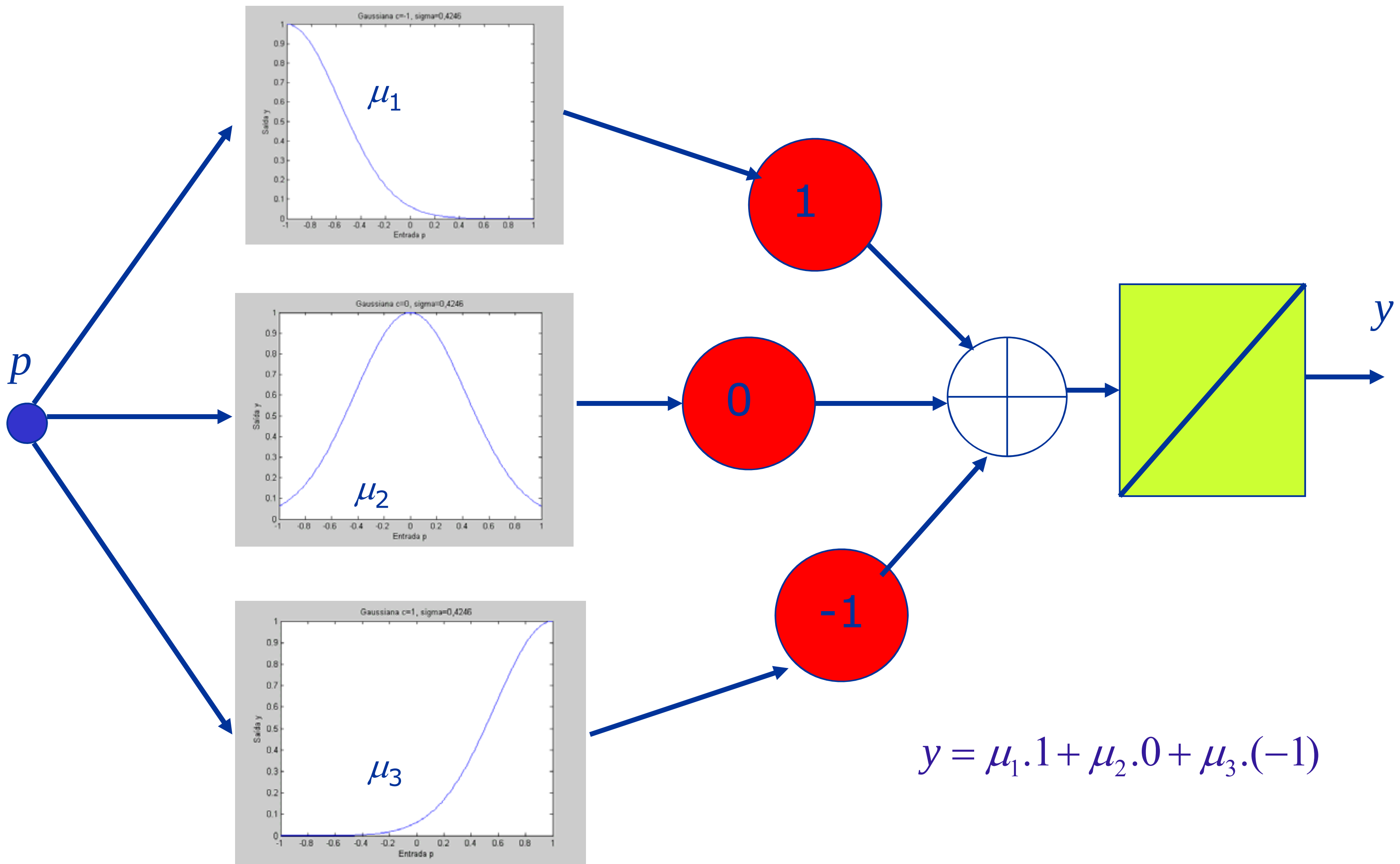

input1 = 0
output1 = 0
1
2
3
-1
1
-1.2
1.2


input1 = 0.445
output1 = -0.42
1
2
3
-1
1
-1.2
1.2

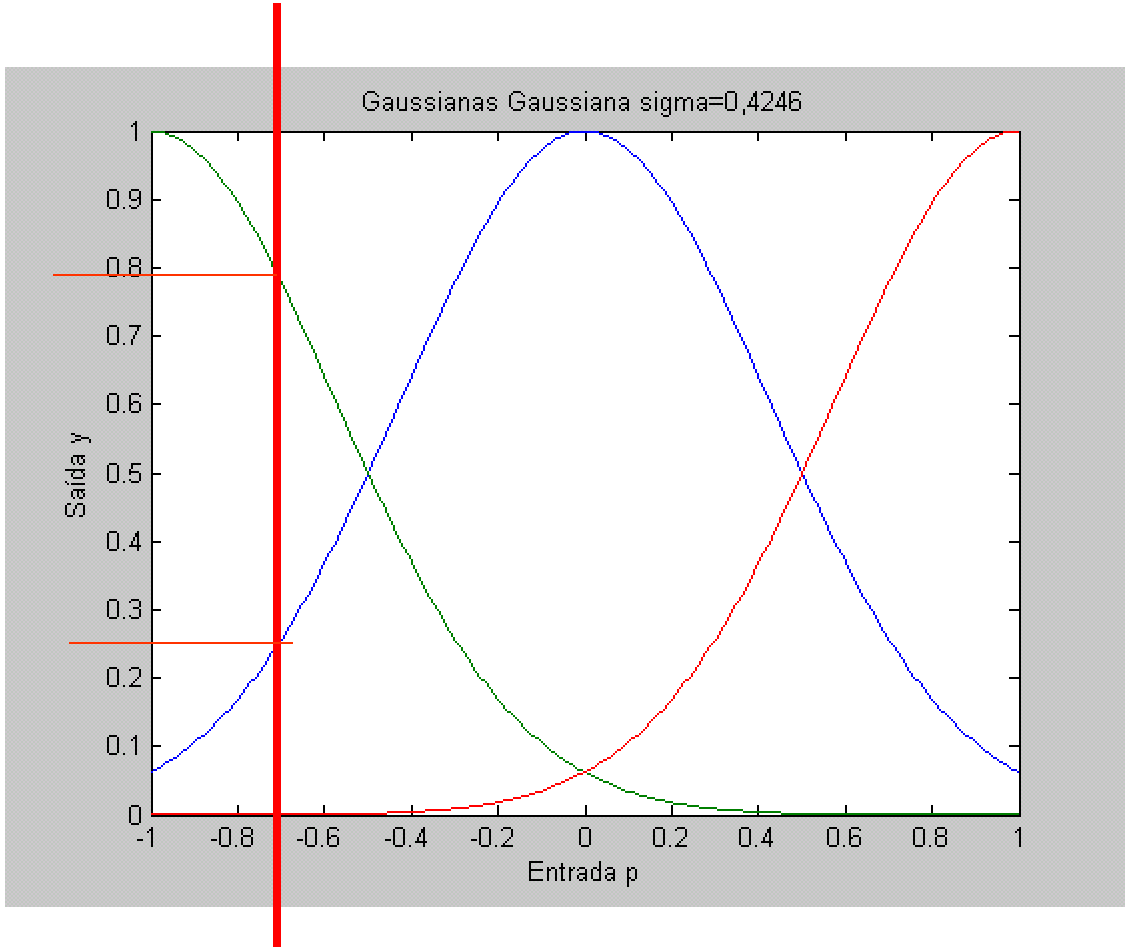

input1 = 0.97
output1 = -0.931
1
2
3
-1
1
-1.2
1.2

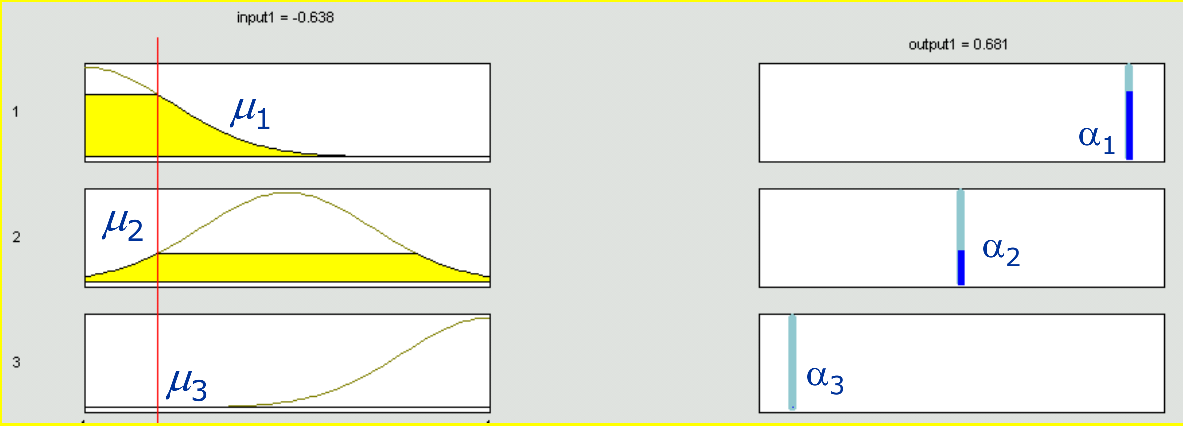


Output
Input
$y = \frac{\mu_1.1 + \mu_2.0 + \mu_3.(-1)}{\mu_1 + \mu_2 + \mu_3}$
input1 = -0.384
1
2
3
-1
1
μ_1
μ_2
μ_3



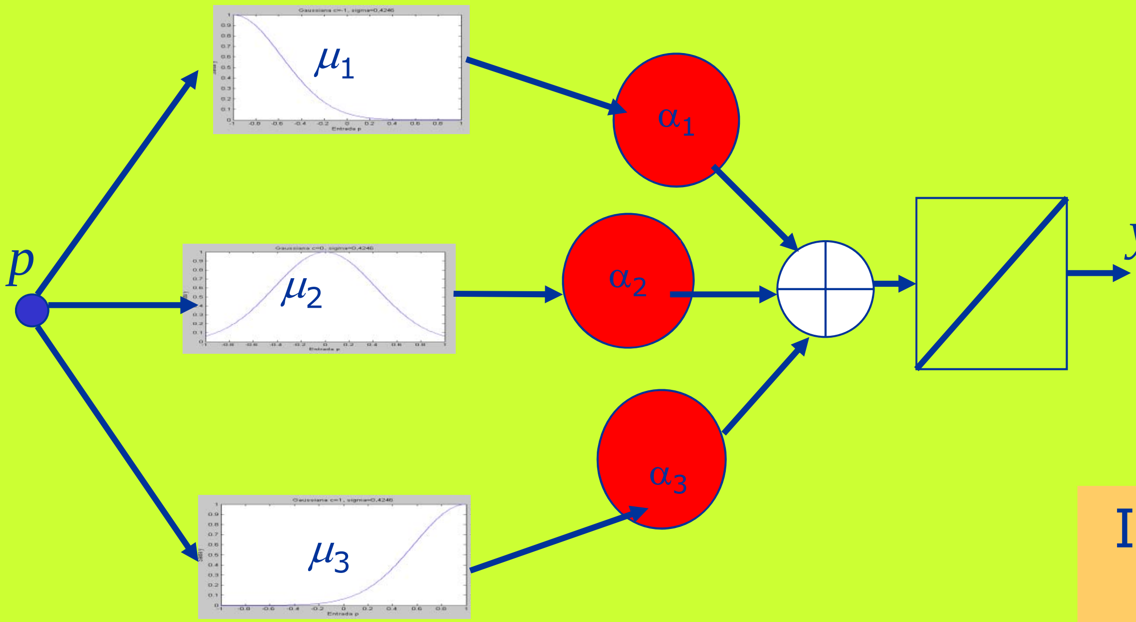
$$y = \mu_1.1 + \mu_2.0 + \mu_3.(-1)$$


Gaussianas Gaussiana sigma=0,4246
Saída y
Entrada p
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1



1. IF p is NEGATIVE THEN y is α_1
2. IF p is ZERO THEN y is α_2
3. IF p is POSITIVE THEN y is α_3

$$y = \frac{\mu_1.\alpha_1 + \mu_2.\alpha_2 + \mu_3.\alpha_3}{\mu_1 + \mu_2 + \mu_3}$$

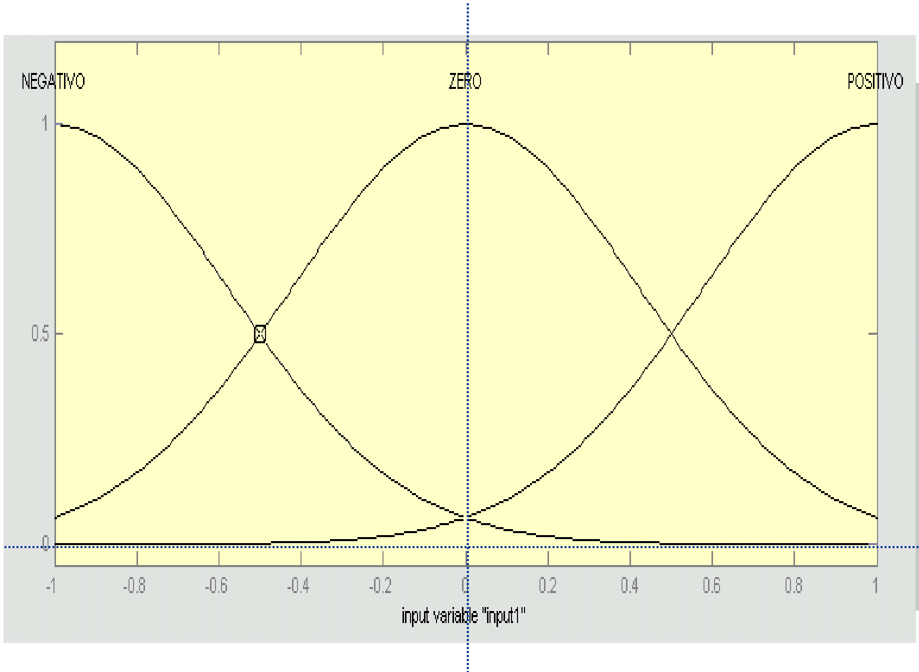


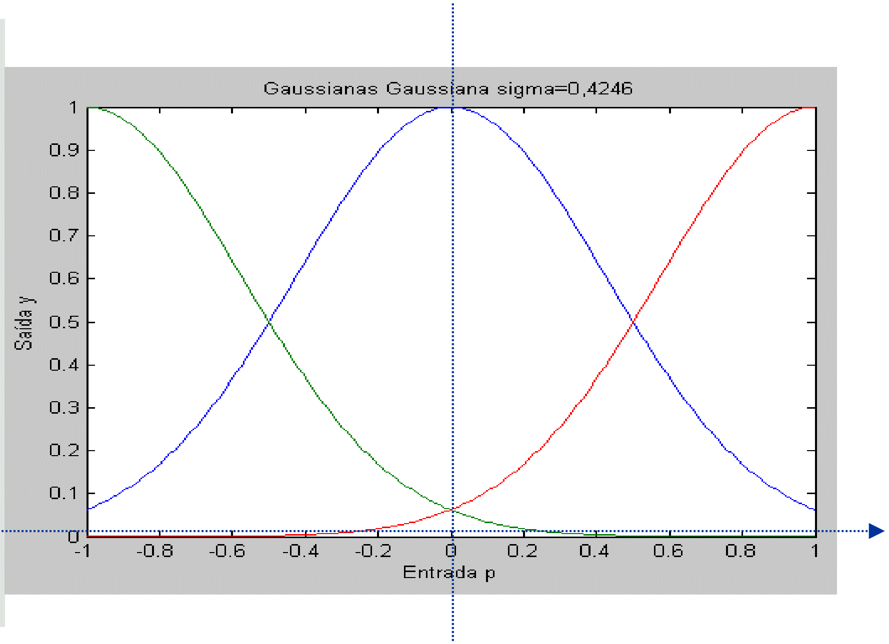
$$y = \mu_1.\alpha_1 + \mu_2.\alpha_2 + \mu_3.\alpha_3$$

If $\mu_1 + \mu_2 + \mu_3 = 1$, they are equivalent

Membership functions

RBFs





In some intervals, the sum of membership is greater than 1.

Conditions of equivalence:

1- the membership functions are equal to the radial basis functions.

2- The sum of memberships for each value of the input is 1.

And if each rule has more than one antecedent ?

And if the consequents are not of zero order ?

Does it exist any NN equivalent ?

10.2. The architecture ANFIS (*Adaptive Network Fuzzy Inference System*)

System TSK of 1st order:

Rule 1: IF p_1 is $\underset{\sim 1}{A}$ AND p_2 is $\underset{\sim 1}{B}$ THEN $y^1 = \alpha_1 + \beta_1 p_1 + \gamma_1 p_2$

Rule 2: IF p_1 is $\underset{\sim 2}{A}$ AND p_2 is $\underset{\sim 2}{B}$ THEN $y^2 = \alpha_2 + \beta_2 p_1 + \gamma_2 p_2$

Two inputs $p_1{}^*$ and $p_2{}^*$ are presented and the rules are fired with intensities r_1 and r_2, (conjunction by the product), producing the outputs y^1 and y^2 :

$$r_1 = \underset{\sim 1}{A}(p_1^*) \times \underset{\sim 1}{B}(p_2^*)$$

$$r_2 = \underset{\sim 2}{A}(p_1^*) \times \underset{\sim 2}{B}(p_2^*)$$

$$y^1 = \alpha_1 + \beta_1 p_1^* + \gamma_1 p_2^*$$

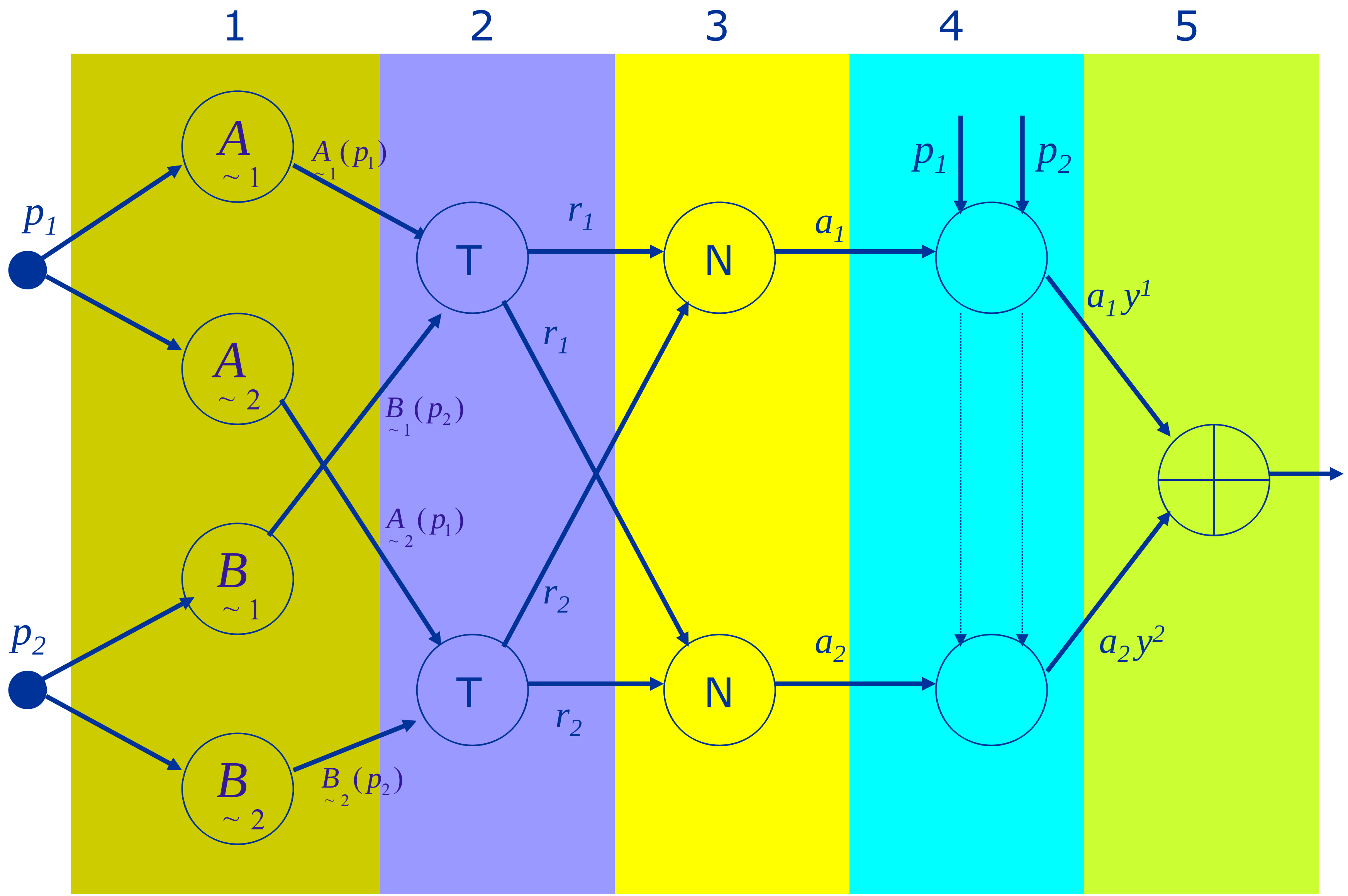
$$y^2 = \alpha_2 + \beta_2 p_1^* + \gamma_2 p_2^*$$

The total output will be

$$y^* = \frac{r_1 y^1 + r_2 y^2}{r_1 + r_2} = a_1 y^1 + a_2 y^2, \qquad a_1 = \frac{r_1}{r_1 + r_2} \qquad a_2 = \frac{r_2}{r_1 + r_2}$$

To realize these operations a neural network with five layers is implemented.

ANFIS



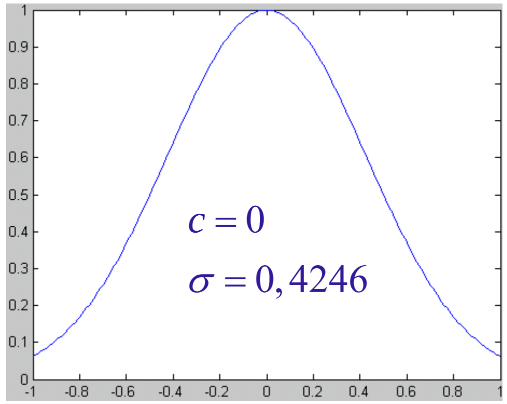
1
2
3
4
5
p_1
p_2
$\underset{\sim}{A}_1$
$\underset{\sim}{A}_2$
$\underset{\sim}{B}_1$
$\underset{\sim}{B}_2$
$\underset{\sim}{A}_1(p_1)$
$\underset{\sim}{B}_1(p_2)$
$\underset{\sim}{A}_2(p_1)$
$\underset{\sim}{B}_2(p_2)$
T
T
r_1
r_1
r_2
r_2
N
N
a_1
a_2
p_1
p_2
$a_1 y^1$
$a_2 y^2$
y

Layer 1

The output of the neuron is the membership value of the crisp inputs to the respective fuzzy sets

$(p_1$ to $\underset{\sim}{A}_1$ and $\underset{\sim}{A}_2$, p_2 to $\underset{\sim}{B}_1$ and $\underset{\sim}{B}_2)$

Frequently, because of differentiability, the membership functions of $\underset{\sim}{A}_1, \underset{\sim}{A}_2, \underset{\sim}{B}_1, \underset{\sim}{B}_2$ are Gaussians.



$$\underset{\sim}{A}_i(p) = e^{-\frac{(p-c_{i1})^2}{2\sigma_{i1}^2}} \qquad \underset{\sim}{B}_i(p) = e^{-\frac{(p-c_{i2})^2}{2\sigma_{i2}^2}}$$

$\{c_{i1} \quad c_{i2} \quad \sigma_{i1} \quad \sigma_{i1}\} \triangleq$ parameters of layer 1

parameters of the antecedents of the rules.

Layer 2

Each neuron computes the fire strength of the rule it is associated with. The outputs of the first neuron (upper) and of the second neuron (lower) are respectively r_1 and r_2:

$$r_1 = \underset{\sim}{A}_1(p_1^*) \times \underset{\sim}{B}_1(p_2^*)$$

$$r_2 = \underset{\sim}{A}_2(p_1^*) \times \underset{\sim}{B}_2(p_2^*)$$

Usually the algebraic product is used as the conjunction operator, because it is differentiable (advantageous for training using retropropagation).

Layer 3

Each neuron normalizes (N) the firing strengths of the rules

$$a_1 = \frac{r_1}{r_1 + r_2} \qquad a_2 = \frac{r_2}{r_1 + r_2}$$

Layer 4

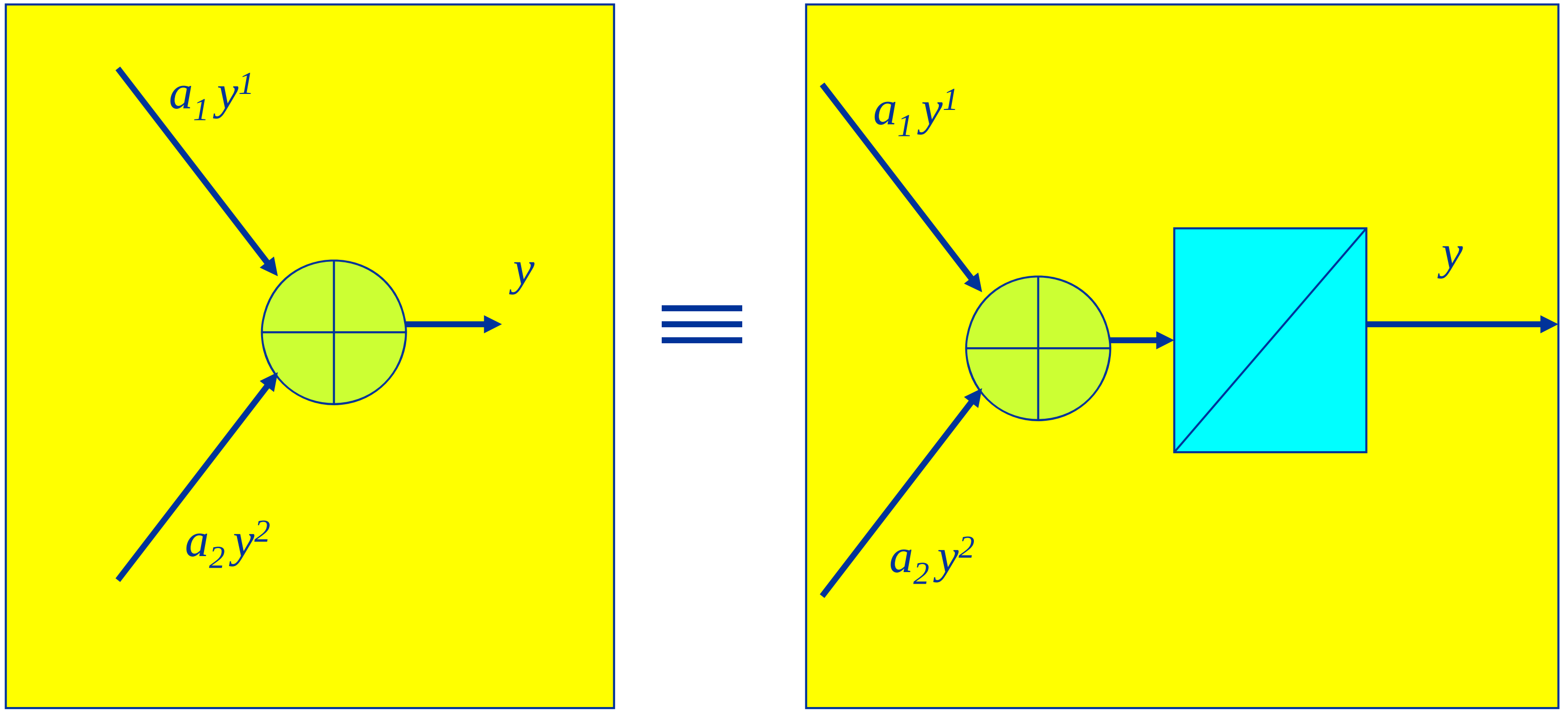
The output of each neuron is the product of a_i by the individual output of each rule

$$a_1 y^1 = a_1(\alpha_1 + \beta_1 p_1^* + \gamma_1 p_2^*)$$
$$a_2 y^2 = a_2(\alpha_2 + \beta_2 p_1^* + \gamma_2 p_2^*)$$

Layer 5

The single neuron (in the case there is only one output) of this layer computes the overall output of the system:

$$y^* = a_1 y^1 + a_2 y^2$$



How does it work:

Give a training set $\{(p_1^k, p_2^k), k=1,...Q\}$.

Compute the output of the network y^k.

Compute the error of the output $e_k = y_k - y_d^k,\ y_d^k$, it is the desired (target) output.

A learning procedure is implemented, using retropropagation (from this comes the name of the architecture ANFIS- *Adaptive Network Fuzzy Inference System*).

For this aim the different activation functions in each layer must be differentiable.

Initialization

The parameters of the antecedents are chosen such that in each dimension the membership functions fulfil the requirements:

- Completeness: they cover all the possible values of the inputs
- Normality: they are normal fuzzy sets
- Convexity: the fuzzy sets are convex.

The final result depends on the initialization.

A global minimum cannot be guaranteed (usually one attains a local minimum).

Minimization algorithms implemented in ANFIS (*Fuzzy Logic Toolbox MatLab*)

(i) retropropagation

All parameters are optimized:

- of the antecedents (membership functions)
- of the consequents (the polynomial coefficients)

order zero: α_i

order 1: α_i, β_i, γ_i , i= 1,..., number of rules

Requires the derivatives of all activation functions.

(ii) Least squares

If the antecedents are fixed, (i.e., the layers 1, 2 and 3) then

$$y^* = a_1 y^1 + a_2 y^2 = a_1(\alpha_1 + \beta_1 p_1^* + \gamma_1 p_2^*) + a_2(\alpha_2 + \beta_2 p_1^* + \gamma_2 p_2^*)$$

$$= a_1\alpha_1 + (a_1 p_1^*)\beta_1 + (a_1 p_2^*)\gamma_1 + a_2\alpha_2 + (a_2 p_1^*)\beta_2 + (a_2 p_2^*)\gamma_2 =$$

$$= \underbrace{\begin{bmatrix} a_1 & (a_1 p_1^*) & (a_1 p_2^*) & a_2 & (a_2 p_1^*) & (a_2 p_2^*) \end{bmatrix}}_{\mathbf{p}^T} \underbrace{\begin{bmatrix} \alpha_1 \\ \beta_1 \\ \gamma_1 \\ \alpha_2 \\ \beta_2 \\ \gamma_2 \end{bmatrix}}_{\boldsymbol{\theta}}$$

$$= \mathbf{p}^T \boldsymbol{\theta} = \boldsymbol{\theta}^{\mathbf{T}} \mathbf{p}$$

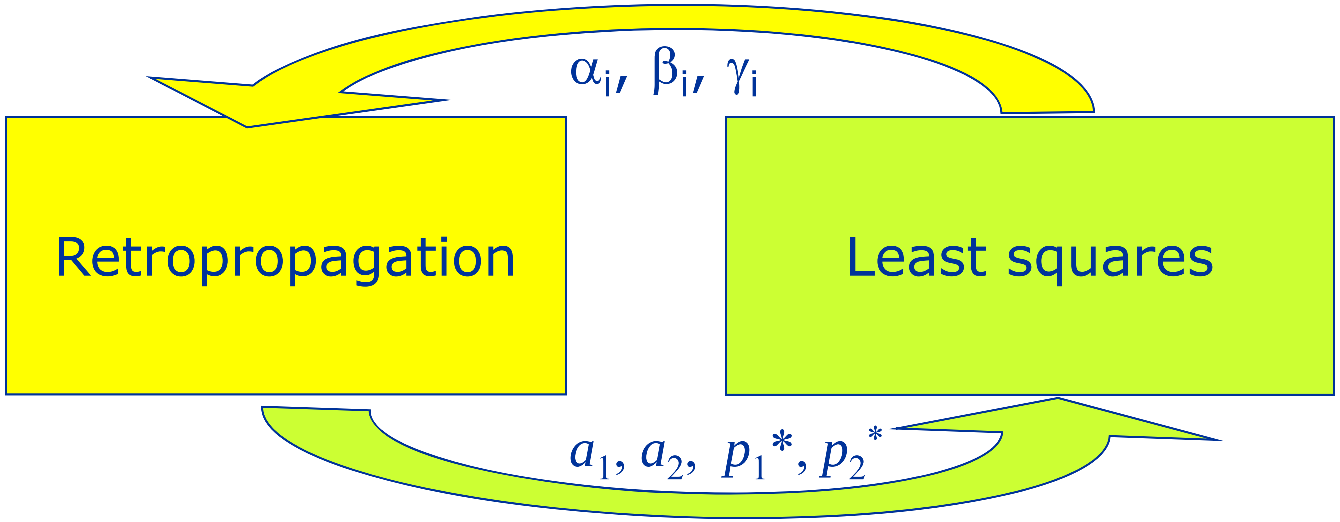
Given a_1, a_2, p_1^*, p_2^* the Widrow-Hoff or the RLS algorithm can be applied.

(ii) Hybrid method

-consequents (α_i, β_i, γ_i , i= 1,..., number of rules):

least squares (recursive or not) (fixing the antecedents)

- antecedents (coefficients of the membership functions): retropropagation (fixed the consequents)



10.3. Mamdani type neuro-fuzzy systems

Rule 1: IF p_1 is $\underset{\sim}{A}_1$ AND p_2 is $\underset{\sim}{B}_1$ THEN y^1 is $\underset{\sim}{C}_1$

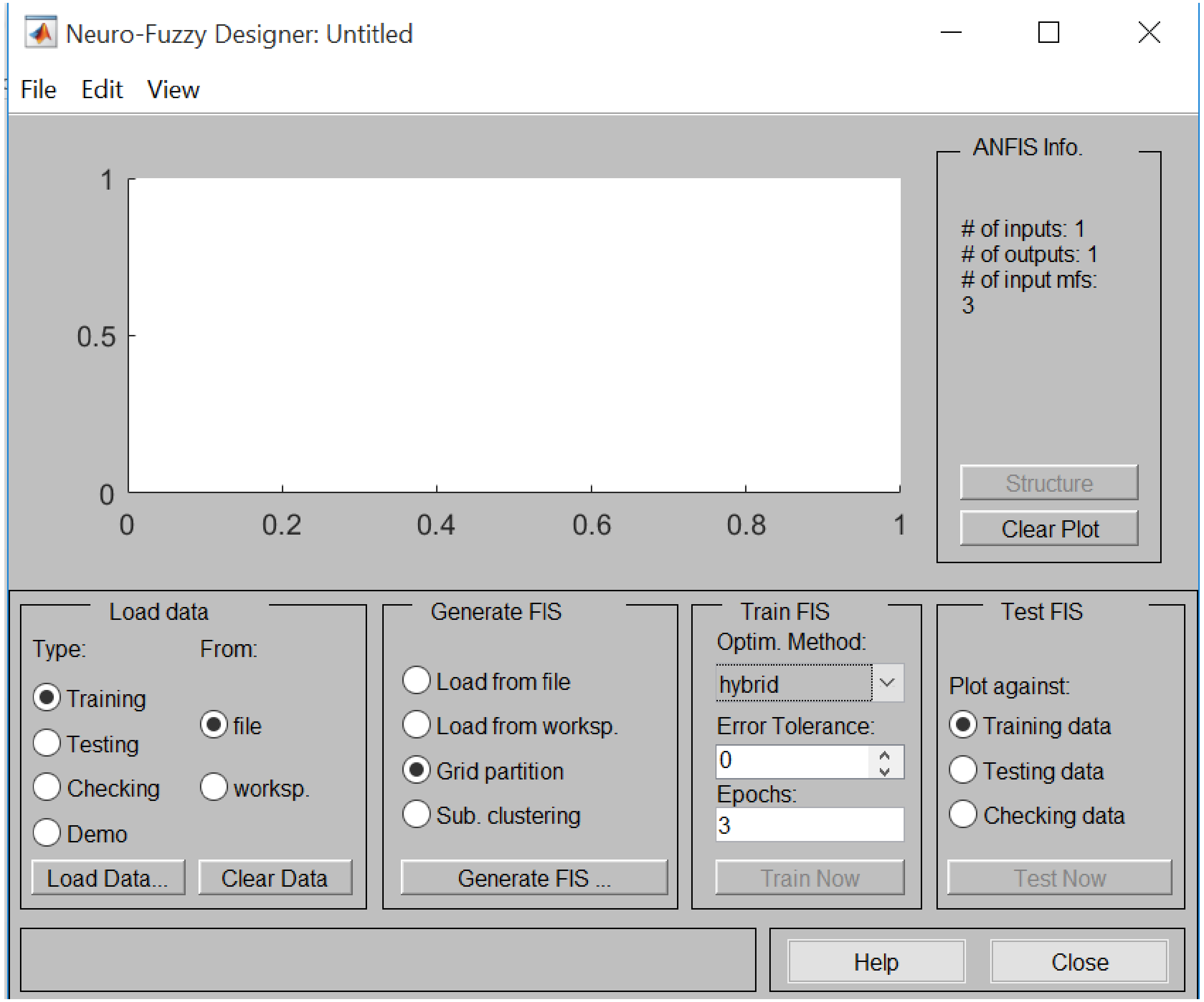
Rule 2: IF p_1 is $\underset{\sim}{A}_2$ AND p_2 is $\underset{\sim}{B}_2$ THEN y^1 is $\underset{\sim}{C}_2$

Several architectures of NN proposed in literature.

Learning difficulties not yet overcome.

Less used that TSK.

10.4. ANFIS in MatLab (*anfisedit*)

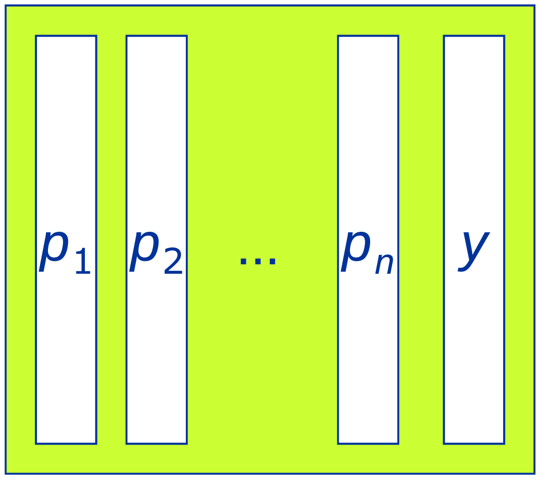


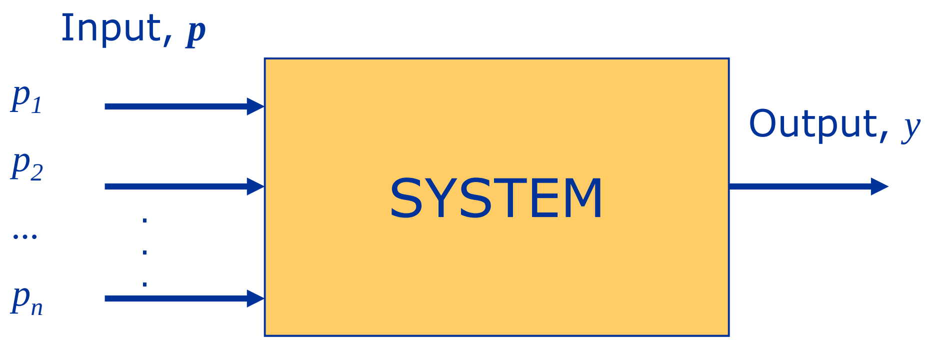
Data format

-Matrices:

-n columns: dimension of the input data p_1, p_2, ..p_n

- last column are the output y data





Initialization

Create a *fis* initial file using

- Grid partition

- Subtractive clustering (see Chap. 6)

Training

- Retropropagation

- Hybrid (retropropagation + least squares)

10.5. The fuzzy c-means clustering

$X \triangleq$ universe of points to group (classify)

$\left\{ \underset{\sim}{A}_i, i = 1, 2, \ldots, c \right\}$ a fuzzy c-partition in X

Each element in X belongs to each of the partitions $\underset{\sim}{A}_i$, with some membership value.

In the limit, one point may belong to all partitions (with sum of membership functions equal to 1).

Let $$\mu_{ik} = \mu_{\underset{\sim}{A}_i}(x_k), \;\; \mu_{ik} \in [0,1]$$

with the constrain

$$\sum_{i=1}^{c} \mu_{ik} = 1, \quad k = 1,2,\ldots,n$$

One class cannot be empty, and one class cannot have all points with membership 1. So

$$0 < \sum_{k=1}^{n} \mu_{ik} < n, \;\; i = 1,2,\ldots,c$$

It can happen that $\mu_{ik} \wedge \mu_{jk} \neq 0$, since the k point may belong to both classes i and j.

We have

$$\bigcup_{i=1}^{c} \mu_{\underset{\sim}{A}_i}(x_k) = 1, \text{ the sum of memberships of each } x_k \text{, for all } k$$

$$0 < \sum_{k=1}^{n} \mu_{\underset{\sim}{A}_i}(x_k) < n, \text{ for all } i \text{ (no class has all points with membership 1)}$$

c-means clustering algorithm

n points

c clusters, with centers in $v_1, v_2, \ldots, v_c$

Objective function

m', fuzziness degree

Crisp case (Cap. 6)

$$J_m(\underset{\sim}{U}, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (\mu_{ik})^{m'} . (d_{ik})^2$$

$$J(U, \mathbf{v}) = \sum_{k=1}^{n} \sum_{i=1}^{c} \psi_{ik} (d_{ik})^2$$

$$d_{ik} = d(x_k - v_i) = \sqrt{\sum_{j=1}^{m} (x_{kj} - v_{ij})^2}$$

μ_{ik} : value of membership of point k to the class i

$m' \in [1, \infty)$: ponderation coefficient, measures the fuzziness degree of the classification

The m coordinates of each center v_i are

Crisp case (Chap. 6)

$$v_{ij} = \frac{\sum_{k=1}^{n} \mu_{ik}^{m'} . x_{kj}}{\sum_{k=1}^{n} \mu_{ik}^{m'}}, \; j = 1, 2, \ldots, m$$

$$v_{ij} = \frac{\sum_{k=1}^{n} \psi_{ik} x_{ik}}{\sum_{k=1}^{n} \psi_{ik}}$$

The c-fuzzy optimal partition will minimize J_m

A global minimum cannot be guaranteed, but only the best solution under a pre-specified accuracy.

For two classes $\underset{\sim}{A}_i$ and $\underset{\sim}{A}_j$ $\underset{\sim}{A}_i \cap \underset{\sim}{A}_j \neq \varnothing$

$$\varnothing \subset \underset{\sim}{A}_i \subset X$$

A family of fuzzy partition matrices , M_{fc}, can be defined, to classify n points into c-classes,

$$M_{fc} = \{\underset{\sim}{U} \mid \mu_{ik} \in [0,1]; \sum_{i=1}^{c} \mu_{ik} = 1; 0 < \sum_{k=1}^{n} \mu_{ik} < n\}$$

$$i = 1, 2, \ldots, c \qquad k = 1, 2, \ldots, n$$

Any $\underset{\sim}{U} \in M_{fc}$ is a fuzzy partition.

M_{fc} has infinite cardinality.

Iterative procedure (Ross, 382)

1th Fix c $(2 \leq c < n)$

Chose a value for m'

Initialize the partition matrix $\underset{\sim}{U}^{(0)}$ (ex. randomly)

For r=1,2,..., do:

2nd Compute the centers $\{v_i^{(r)}, i=1,2,...,c\}$

3th Update the partition matrix at iteration $r, \underset{\sim}{U}^{(r)}$

$$\mu_{ik}^{(r+1)} = \left[\sum_{j=1}^{c} \left(\frac{d_{ik}^{(r)}}{d_{jk}^{(r)}} \right)^{\frac{2}{m'-1}} \right]^{-1}$$

for $I_k \neq \varnothing$ (for the classes whose indexes belong to I_k)

or

$\mu_{ik}^{(r+1)} = 0,$ for all the classes i in which $i \in \tilde{I}_k$

$I_k = \left\{ i \mid 2 \leq i < c \ : \ d_{ik}^{(r)} = 0 \right\}$ (centers whose distance to point k is null point k will be a center, so membership 1)

$\tilde{I}_k = \left\{ 2, \ldots, c \right\} - I_k$ (centers whose distance to the point k is non null)

$$\sum_{i \in I_k} \mu_{ik}^{(r+1)} = 1$$

4th If $\left\| U^{(r+1)} - U^{(r)} \right\| < \varepsilon_L$ stop.

If not, $r = r+1$

go to 2nd

J_m: criterion of minimum of squared distances.

The squared distances are weighted by the membership values $(\mu_{ik})^{m'}$.

J_m: minimizes the squared distances of the points to their centers

maximizes the distances between the centers of the clusters.

Which is the good value for m'?

$$m'=1,\quad \frac{2}{m'-1}=\frac{2}{0}=\infty$$

and then

$$\mu_{ik}^{(r+1)}=\begin{cases} 1, \text{ if for all } j, d_{ik}^{(r)}<d_{jk}^{(r)}, i\neq j, \left(\dfrac{d_{ik}}{d_{jk}}\right)^{\infty}=0, \left(\dfrac{d_{ik}}{d_{ik}}\right)^{\infty}=1 \\ 0, \text{ if for some } j, d_{ik}^{(r)}>d_{jk}^{(r)}, j\neq i, \left(\dfrac{d_{ik}}{d_{jk}}\right)^{\infty}=\infty, \infty^{-1}=0 \end{cases}$$

(crisp case)

$$m'=\infty, (\mu_{ik})^{m'}=0 \Rightarrow J_m(U,v)=0$$

(completely fuzzy)

The greater m', the stronger is the fuzziness of the membership to the clusters; m' controls this fuzzy character of membership to the clusters.

m' increases, J_m decreases (keeping constant all the other parameters), slower is the convergence.

There is no theoretical optimum for m'; generally it is chosen between 1.25 and 2 (Ross).In Matlab default is 2.

Measures for the fuzzy classification

-which is the uncertainty (fuzziness) degree of a classification ?

-"How fuzzy is a fuzzy c-partition?" (Ross, 387).

- which is the level of superposition of the defined classes ?

Let x_k be a classified element of the Universe

$\mu_i(x_k)$- value of membership of x_k to the class i

$\mu_j(x_k)$- value of membership of x_k to the class j

$\mu_i \, \mu_j$, algebraic product, depending directly from the relative superposition between non-empty clusters. This is a good measure for the uncertainty of the classification.

Coefficient of the fuzzy partition

$$F_C(\underset{\sim}{U}) = \frac{tr(\underset{\sim}{U}\,\underset{\sim}{U}^T)}{n}, \quad \underset{\sim}{U} = [\mu_{ik}]:$$ matriz of fuzzy partition

Interprets the results of the fuzzy partition.

Properties:

$F_C(\underset{\sim}{U}) = 1$ if the partitions are crisp

$F_C(\underset{\sim}{U}) = \frac{1}{c}$ if $\mu_i = \frac{1}{c}$, $i=1, 2, ..., c$ (total ambiguity)

$\frac{1}{c} \leq F_c(\underset{\sim}{U}) \leq 1$ in any case

The elements of the diagonal of $\underset{\sim}{U}\,\underset{\sim}{U}^T$ are proportional to the quantities of non-shared memberships in the fuzzy clusters.

The elements out of the diagonal of $\underset{\sim}{U}\,\underset{\sim}{U}^T$ represent the quantity of shared partition between pairs of fuzzy clusters. If they are null, the partition is crisp.

As the fuzzy partition coefficient approaches 1, the fuzzy uncertainty is minimized in the overlapping of the clusters. The greater $F_C(\underset{\sim}{U})$, the better succeeded is the partition of the dataset into clusters.

Remarks:

1. If x_k belongs with memberships $\mu_i(k)$ and $\mu_j(k)$ to the classes i and j, then $min(\mu_i(k), \mu_j(k))$ gives the quantity of membership claimed both by i and by j, and so not shared. It is a good measure of self - superposition.

2. The elements of the diagonal of $\underset{\sim}{U}\,\underset{\sim}{U}^T$ are the sum of the square of the elements of each row of $\underset{\sim}{U}$

Defuzzification of a fuzzy partition

Method of maximum membership

- the highest element of each columns of U passes to 1, and all the others to zero.

$$\mu_{ik} = \underset{i \in c}{máx}\{u_{ik}\} \Rightarrow \mu_{ik} = 1 \wedge \mu_{jk} = 0,\ j \neq i$$

$$i, j = 1, 2, \ldots, c \qquad k = 1, 2, \ldots n$$

Classification by the nearest center

- each point is affected to the cluster whose center is closest to it,

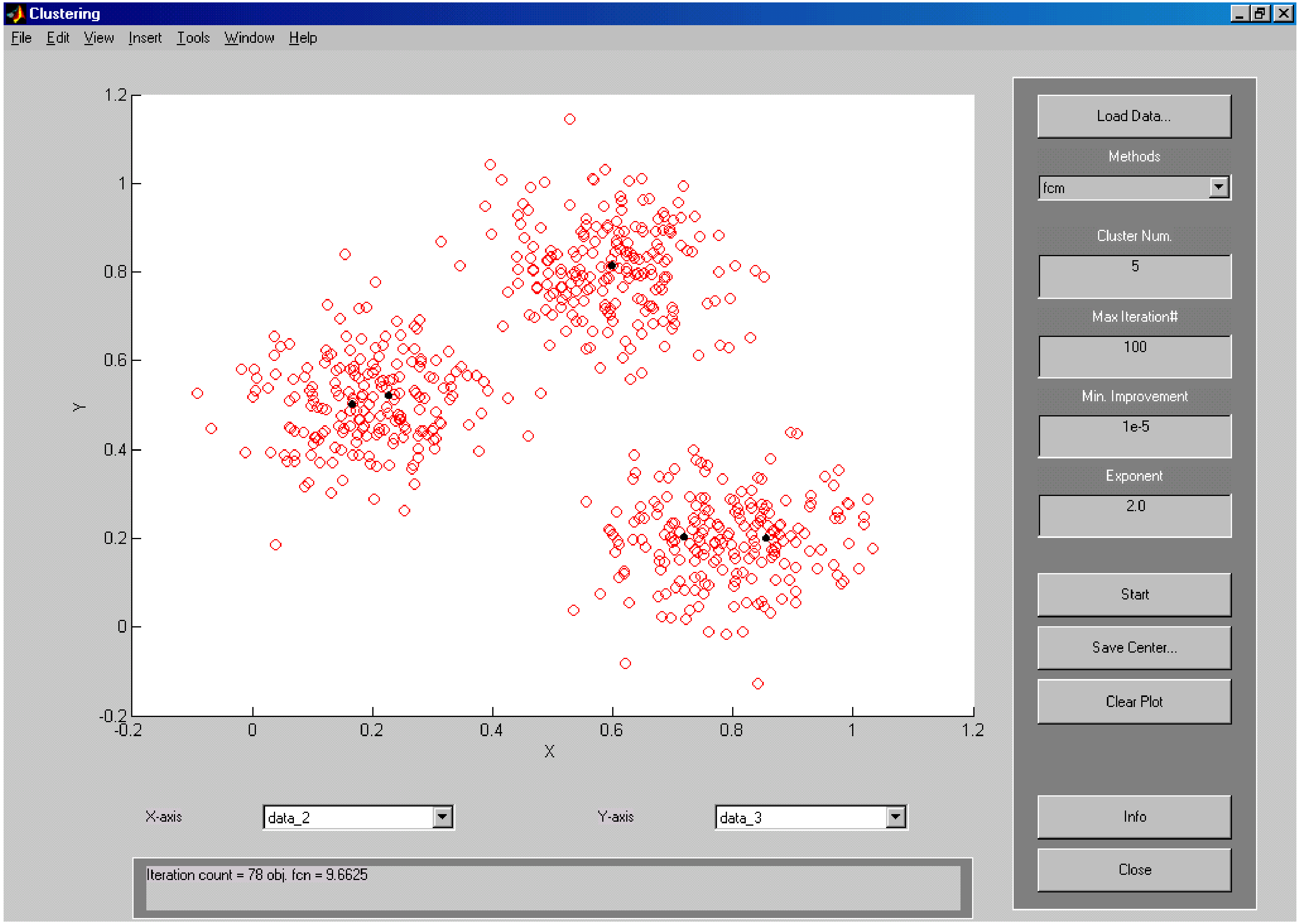
$$d_{ik} = \underset{i \in c}{mín}\{d_{ik}\} \Rightarrow \mu_{ik} = 1 \wedge \mu_{ij} = 0,\ j \neq i$$

$$d_{jk} = \left\| x_k - v_j \right\|$$

In Fuzzy Logic Toolbox: *fcm*

GUI for clustering: *findcluster* in command line.

Clustering GUI , Fuzzy Logic Toolbox



10.6. Derivation of the rules from fuzzy partitions

The fuzzy c-means clustering may be applied to obtain the initial membership functions in ANFIS.

one input,
one output

i) Collect the training data made up by pairs (p,y)

ii) Represent these data in the plane (p,y)

iii) Apply the clustering method

Each center $v_i = (p_i, y_i)$ obtained will define a rule

Definition of the rule from the center

Center $v_i=(p_i, y_i)$

Rule TSK zero order

IF input is $\underset{\sim}{p}_i$ THEN output is y_i

$\underset{\sim}{p}_i$ Fuzzy set obtained by defining, in the dimension of p , a membership function centered in p_i

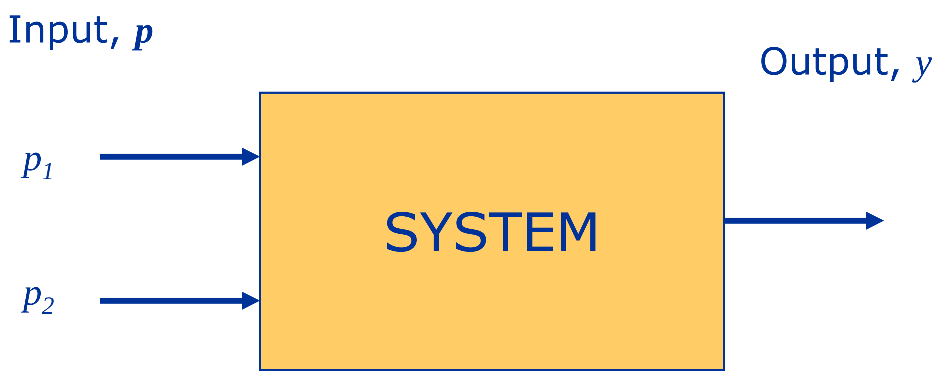
y_i value of the coordinate of the center v_i in the dimension of y , *i.e.*, y_i

D1

Diapositivo 52

D1 Dourado; 26-11-2016

Case of two inputs



i) Collect the training data made up by 3-tuples (p_1,p_2,y)

ii) Represent these data in the tridimensional space (p_1,p_2,y)

iii) Apply the clustering method

Each obtained center (p_{i1},p_{i2},y_i) will define one rule

Definition of a rule from the center

Center $v_i=(p_{i1}, p_{i2}, y_i)$

Rule Mamdani type:

IF p_1 is $\underset{\sim}{p}_{i1}$ AND p_2 is $\underset{\sim}{p}_{i2}$ THEN output is $\underset{\sim}{y}_i$

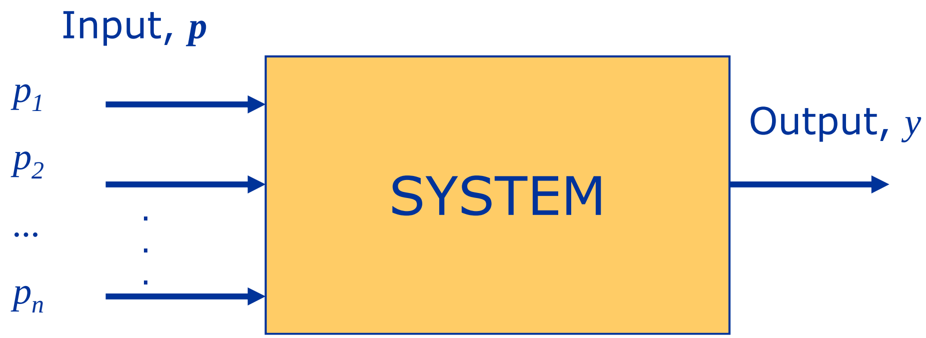
$\underset{\sim}{p}_{i1}, \underset{\sim}{p}_{i2}, \underset{\sim}{y}_i$ Fuzzy sets obtained projecting the center v_i in the dimensions p_1, p_2, y, respectively; these projections are the centers of the membership functions

Rule TSK of order zero:

IF p_1 is $\underset{\sim}{p}_{i1}$ AND p_2 is $\underset{\sim}{p}_{i2}$ THEN output is y_i

y_i value of the coordinate of the center in the dimension y.

Case de n inputs



Training data : tuples $(p_1, p_2, \ldots, p_n, y)$

Centers: $v_i = (p_{i1}, p_{i2}, \ldots, p_{in}, y_i)$

Rule TSK of order zero:

IF Input1 is $\underset{\sim}{p}_{i1}$ AND Input2 is $\underset{\sim}{p}_{i2}$ AND ... AND Input$_n$ is $\underset{\sim}{p}_{in}$ THEN output is y_i

Rules type Mamdani

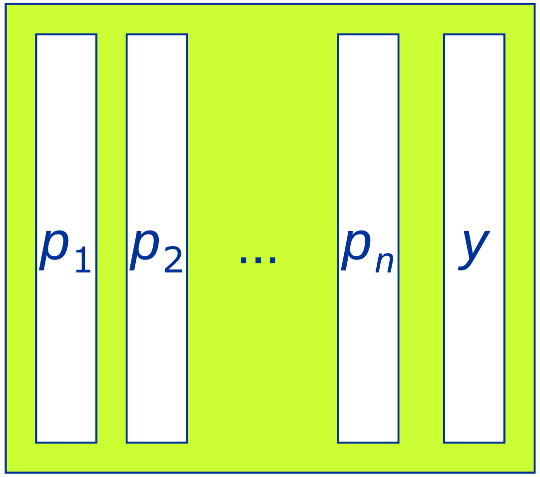
- For the antecedents the same procedure as TSK is applied.

- For the consequents the cluster is projected in the y dimension.

- Algorithms improving o FCM
  - Gustafson-Kessel
  - Gath-Geva

10.7. Derivation of the rules from subtractive clustering

- The subtractive clustering produces a set of centers.



- Each center has n+1 coordinates
  - -n for the inputs
  - -1 for the output

- center one membership function in each center

- the openness is computed as in the case of RBF.

- project the membership function in each dimension

- The resulting fuzzy sets define the rules (one for each center).

10.8. Conclusions

Neuro-fuzzy systems are useful whenever there exist available experimental data end there isn’t available sufficient knowledge (theoretical or empirical) about the system to write directly fuzzy rules.

They are used to optimize rule-based fuzzy systems.

The rules issued form optimization should preferably have a semantic meaning for the human user, i.e., they should be transparent and interpretable. This leads to special learning techniques, an actual research subject.

Bibliografia

Fuzzy Logic Toolbox Users' Guide, The Mathworks, 2018.

Fuzzy Logic With Engineering Applications, Timothy Ross, McGraw Hill, 3rd Edition, 2010.

Fuzzy Set Theory and Its Applications, 4th ed. H. Zimmermann, Springer Verlag, 2001.

Fuzzy Cluster Analysis, Methods for Classification, Data Analysis and Image Recognition, Höpner, F., F. Klawonn, R. Kruse, T. Runkler, John Wiley and Sons, 1999.

Fuzzy Systems Theory and its Applications, T. Terano, K. Asai and M. Sugeno, Academic Press, 1987

Introduction to Neuro-Fuzzy Systems, Robert Fullér, Springer Verlag 2000.

An Introduction to Fuzzy Control, D. Driankov, H. Hellendoorn and M. Reinfrank, Springer Verlag 1996.

Fuzzy Modelling and Control, Andrzej Piegat, Springer Verlag, 2001.

Redução da Complexidade e Análise de Interpretabilidade de Sistemas Neuro-Difusos, Carlos M. J. S. Pereira, Tese de Doutoramento, DEI/FCTUC 2002.

Interpretability and learning in neuro-fuzzzy systems, Rui Paiva and António Dourado, Fuzzy Sets and Systems, 147(2004) pp 17-38, Elsevier.

Building interpretable systems in real time, Jose V. Ramos, Carlos Pereira, Antonio Dourado, in *Evolving Intelligent Systems: Methodology and Applications*, Edited by Plamen Angelov, Dimitar P. Filev, and Nikola Kasabov 2010, Institute of Electrical and Electronics Engineers.

Evolving Fuzzy Systems - Methodologies, Advanced Concepts and Applications, Lughofer, Edwin Studies in Fuzziness and Soft Computing, Vol. 266, 2011, XXIV, Springer.

A new method for designing neuro-fuzzy systems for nonlinear modelling with interpretability aspects, K. Cpałka nn, K. Łapa n, A.Przybył, M.Zalasiński , Neurocomputing, DOI:10.1016/j.neucom.2013.12.031, Elsevier.