10

Flexible neuro-fuzzy systems

10.1 Introduction

In the previous chapter we considered Mamdani and logical neuro-fuzzy systems. In the present chapter we will build a neuro-fuzzy system, the inference method (Mamdani or logical) of which will be found as a result of the learning process. The structure of such a system will be changing during the learning process. Its operation will be possible thanks to specially constructed adjustable triangular norms. Adjustable triangular norms, applied to aggregate particular rules, take the form of a classic t-norm or t-conorm after the learning process is finished. Adjustable implications, which finally take the form of a "correlation function" between premises and consequents (Mamdani approach) or fuzzy S-implication (logical approach), will be constructed in analogical way. Moreover, the following concepts will be used for construction of the neuro-fuzzy systems: the concept of soft triangular norms, parameterized triangular norms as well as weights which describe the importance of particular rules and antecedents in those rules.

10.2 Soft triangular norms

Soft equivalents of triangular norms shall be defined in the following way:

$$\widetilde{T}\{\mathbf{a};\alpha\} = (1-\alpha)\frac{1}{n}\sum_{i=1}^{n} a_i + \alpha T\{a_1,\ldots,a_n\} \tag{10.1}$$

and

$$\widetilde{S}\{\mathbf{a};\alpha\} = (1-\alpha)\frac{1}{n}\sum_{i=1}^{n} a_i + \alpha S\{a_1,\ldots,a_n\}, \tag{10.2}$$

where $\mathbf{a} = [a_1, ..., a_n]$ and $\alpha \in [0,1]$. The above operators allow smooth balancing between the arithmetic average of arguments $a_1, \ldots, a_n$ and a classic t-norm or t-conorm operator.

Example 10.1
The soft Zadeh t-norm (of the min type) shall be defined as follows:

$$\widetilde{T}\{a_1,a_2;\alpha\} = (1-\alpha)\frac{1}{2}(a_1+a_2) + \alpha\min\{a_1,a_2\}. \tag{10.3}$$

Its operation is illustrated by Fig. 10.1.

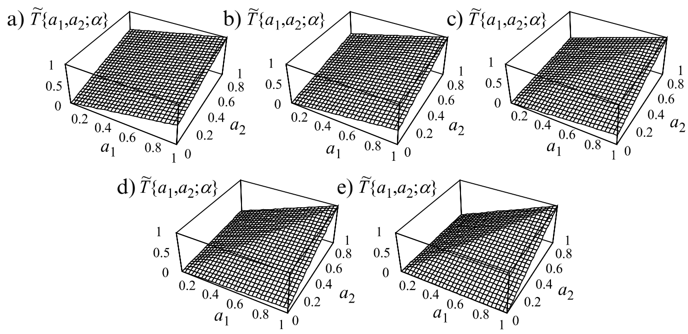

FIGURE 10.1. Hyperplanes of function (10.3) for a) $\alpha = 0.00$, b) $\alpha = 0.25$, c) $\alpha = 0.50$, d) $\alpha = 0.75$, e) $\alpha = 1.00$

The soft Zadeh t-conorm takes the following form

$$\widetilde{S}\{a_1,a_2;\alpha\} = (1-\alpha)\frac{1}{2}(a_1+a_2) + \alpha\max\{a_1,a_2\}. \tag{10.4}$$

Its operation is illustrated by Fig. 10.2.

As we remember, the "correlation function" in the Mamdani approach shall be defined through the t-norm. A soft equivalent of this function shall be notated as follows:

$$\widetilde{I}(a,b;\beta) = (1-\beta)\frac{1}{2}(a+b) + \beta T\{a,b\}. \tag{10.5}$$

The soft S-implication takes the form

$$\widetilde{I}(a,b;\beta) = (1-\beta)\frac{1}{2}(1-a+b) + \beta S\{1-a,b\}, \tag{10.6}$$

where $\beta \in [0,1]$ in both cases.

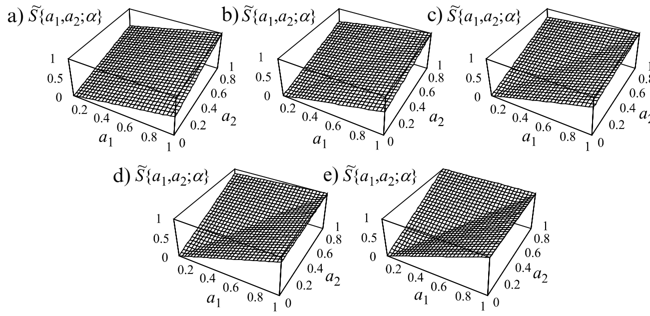

FIGURE 10.2. Hyperplanes of function (10.4) for a) $\alpha = 0.00$, b) $\alpha = 0.25$, c) $\alpha = 0.50$, d) $\alpha = 0.75$, e) $\alpha = 1.00$

Example 10.2
The soft binary S-implication is given by the following formula

$$\widetilde{I}\,(a, b; \beta) = (1-\beta)\,\frac{1}{2}\,(1-a+b) + \beta \max\{1-a, b\}\,. \tag{10.7}$$

Its operation is illustrated by Fig. 10.3.

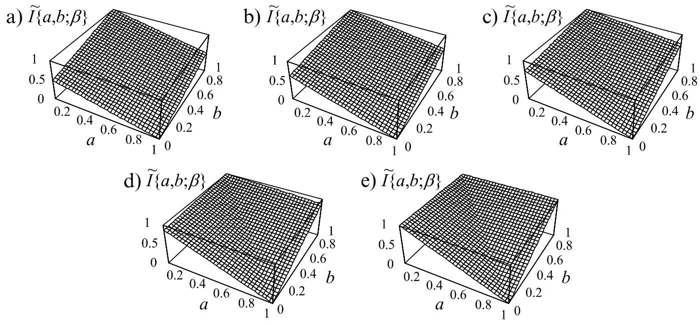

FIGURE 10.3. Hyperplanes of function (10.7) for a) $\beta = 0.00$, b) $\beta = 0.25$, c) $\beta = 0.50$, d) $\beta = 0.75$, e) $\beta = 1.00$

To construct Mamdani systems, we can use the following soft triangular norms:

- $\widetilde{T}_1\,\{\mathbf{a}; \alpha^{\tau}\} = (1-\alpha^{\tau})\,\frac{1}{n}\sum_{i=1}^{n} a_i + \alpha^{\tau} T_{i=1}^{n}\,\{a_i\}$ to aggregate the premises in particular rules;
- $\widetilde{T}_2\,\{b_1, b_2; \alpha^{I}\} = \left(1-\alpha^{I}\right)\frac{1}{2}\,(b_1+b_2) + \alpha^{I} T\,\{b_1, b_2\}$ to combine the premises and consequents of the rules;
- $\widetilde{S}\,\{\mathbf{c}; \alpha^{\mathrm{agr}}\} = (1-\alpha^{\mathrm{agr}})\,\frac{1}{N}\sum_{k=1}^{N} c_k + \alpha^{\mathrm{agr}} S_{k=1}^{N}\,\{c_k\}$ to aggregate the rules,

where n is the number of inputs while N is the number of rules.

To construct logical systems using the S-implication, we can use the following soft triangular norms:

- $\widetilde{T}_1\{\mathbf{a};\alpha^\tau\} = (1-\alpha^\tau)\frac{1}{n}\sum_{i=1}^n a_i + \alpha^\tau T_{i=1}^n\{a_i\}$ to aggregate the premises in particular rules;
- $\widetilde{S}\{b_1,b_2;\alpha^I\} = \left(1-\alpha^I\right)\frac{1}{2}(1-b_1+b_2) + \alpha^I S\{1-b_1,b_2\}$ to combine the premises and consequents of the rules;
- $\widetilde{T}_2\{\mathbf{c};\alpha^{\text{agr}}\} = (1-\alpha^{\text{agr}})\frac{1}{N}\sum_{k=1}^N c_k + \alpha^{\text{agr}} T_{k=1}^N\{c_k\}$ to aggregate the rules,

where n is the number of inputs while N is the number of rules. It should be emphasized that parameters α^τ, α^I and α^{agr} can be found as a result of learning.

10.3 Parameterized triangular norms

In order to construct flexible systems, we can also use parameterized variations of triangular norms. These include among other things Dombi, Hamacher, Yager, Frank, Weber, Dubois and Prade, Schweizer and Mizumoto triangular norms. The notations $\overset{\leftrightarrow}{T}\{a_1,a_2,\ldots,a_n;p\}$ and $\overset{\leftrightarrow}{S}\{a_1,a_2,\ldots,a_n;p\}$ will be used to notate them. Parameterized triangular norms are characterized by the fact that their corresponding hyperplanes can be modified as a result of learning the parameter p.

Example 10.3
Parameterized Dombi t-norm is defined as follows:

$$\overset{\leftrightarrow}{T}\{\mathbf{a};p\} = \begin{cases} \textit{Łukasiewicz t-norm} & \text{for} \quad p=0, \\ \left(1+\left(\sum\limits_{i=1}^n\left(\dfrac{1-a_i}{a_i}\right)^p\right)^{\frac{1}{p}}\right)^{-1} & \text{for} \quad p\in(0,\infty), \\ \textit{Zadeh t-norm} & \text{for} \quad p=\infty. \end{cases} \tag{10.8}$$

Its operation for $n=2$ is illustrated by Fig. 10.4.

Parameterized Dombi t-conorm is defined as follows:

$$\overset{\leftrightarrow}{S}\{a;p\} = \begin{cases} \textit{Łukasiewicz}\, t-conorm & \text{for} \quad p=0\ , \\ 1-\left(1+\left(\sum_{i=1}^n\left(\frac{a_i}{1-a_i}\right)^p\right)^{\frac{1}{p}}\right)^{-1} & \text{for} \quad p\in(0,\infty)\ , \\ Zadeh\ t-conorm & \text{for} \quad p=\infty\,. \end{cases} \tag{10.9}$$

Figure 10.5 illustrates its operation.

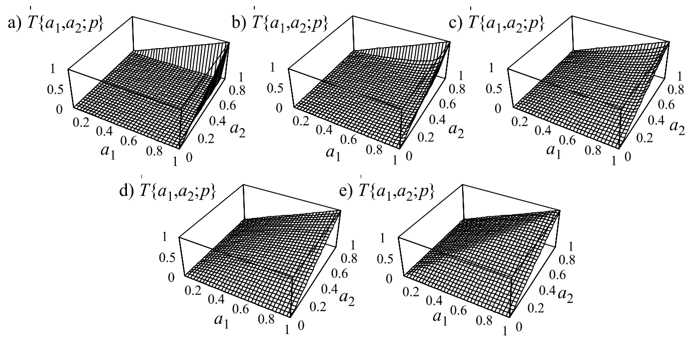

FIGURE 10.4. Hyperplanes of function (10.8) for a) $p = 0.10$, b) $p = 0.25$, c) $p = 0.50$, d) $p = 1.00$, e) $p = 10.00$

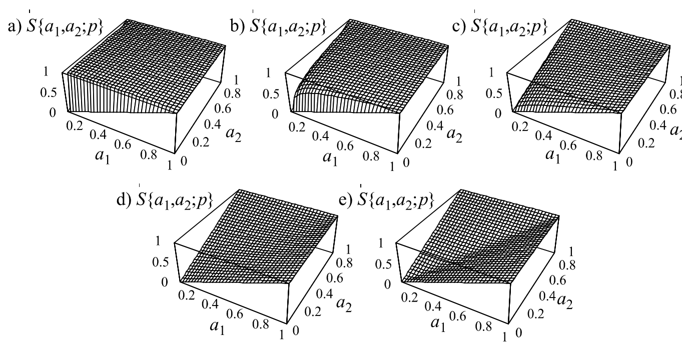

FIGURE 10.5. Hyperplanes of function (10.9) for a) $p = 0.10$, b) $p = 0.25$, c) $p = 0.50$, d) $p = 1.00$, e) $p = 10.00$

The additive generator of parameterized Dombi t-norm takes the form

$$t_{\text{add}}(x) = \left(\frac{1-x}{x}\right)^p, \tag{10.10}$$

while the additive generator of parameterized Dombi t-conorm is defined as follows:

$$s_{\text{add}}(x) = \left(\frac{x}{1-x}\right)^p. \tag{10.11}$$

Parameterized Dombi t-norm for $n = 2$ may play the role of a "correlation function". By combining the concept of parameterized Dombi t-conorm with the concept of S-implication we obtain the parameterized Dombi S-implication which is notated as follows:

$$\overset{\leftrightarrow}{I}(a, b; p) = 1 - \left(1 + \left(\left(\frac{1-a}{a}\right)^p + \left(\frac{b}{1-b}\right)^p\right)^{\frac{1}{p}}\right)^{-1} \tag{10.12}$$

for $p \in (0, \infty)$. The operation of parameterized Dombi S-implication is illustrated by Fig. 10.6.

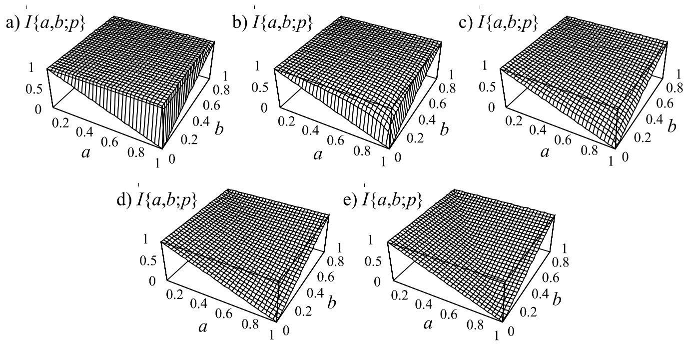

FIGURE 10.6. Hyperplanes of function (10.12) for a) $p = 0.10$, b) $p = 0.25$, c) $p = 0.50$, d) $p = 1.00$, e) $p = 10.00$

Example 10.4

Parameterized Yager t-norm is defined as follows:

$$\overset{\leftrightarrow}{T}\{\mathbf{a};p\} = \begin{cases} \textit{Łukasiewicz t-norm} & \text{for } p = 0 \\ \max\left\{0, 1 - \left(\sum\limits_{i=1}^{n}(1-a_i)^p\right)^{\frac{1}{p}}\right\} & \text{for } p \in (0,\infty) \\ \textit{Zadeh t-norm} & \text{for } p = \infty \end{cases} \quad (10.13)$$

for $p > 0$. Its operation for $n = 2$ is illustrated by Fig. 10.7.

Parameterized Yager t-conorm is defined as follows:

$$\overset{\leftrightarrow}{S}\{\mathbf{a};p\} = \begin{cases} \textit{boundary t-conorm} & \text{for } p = 0\,, \\ \min\left\{1, \left(\sum_{i=1}^{n}(a_i)^p\right)^{\frac{1}{p}}\right\} & \text{for } p \in (0,\infty), \\ \textit{Zadeh t-conorm} & \text{for } p = \infty\,. \end{cases} \quad (10.14)$$

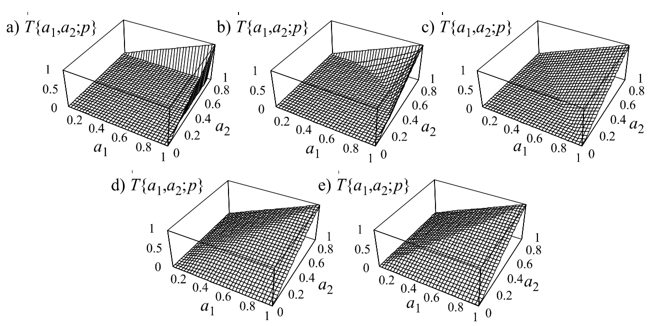

FIGURE 10.7. Hyperplanes of function (10.13) for a) $p = 0.1$, b) $p = 0.5$, c) $p = 1.0$, d) $p = 10.0$, e) $p = 100.0$

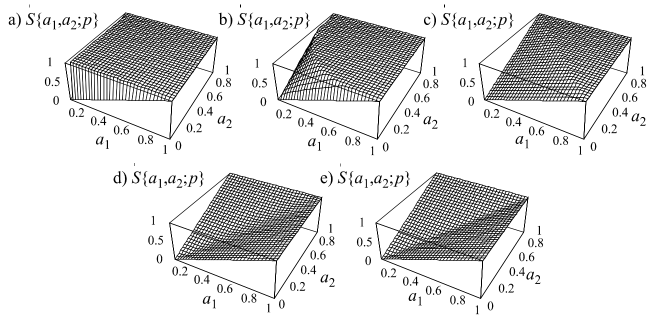

FIGURE 10.8. Hyperplanes of function (10.14) for a) $p = 0.1$, b) $p = 0.5$, c) $p = 1.0$, d) $p = 10.0$, e) $p = 100.0$

Figure 10.8 illustrates its operation.

The additive generator of parameterized Yager t-norm takes the form

$$t_{\text{add}}(x) = (1 - x)^p, \tag{10.15}$$

while the additive generator of parameterized Yager t-conorm is defined as follows:

$$s_{\text{add}}(x) = x^p. \tag{10.16}$$

Parameterized Yager t-norm for $n = 2$ can be used as "correlation function". By combining the concept of parameterized Yager t-conorm with the concept of S-implication we obtain parameterized Yager S-implication which is notated as follows:

$$\overset{\leftrightarrow}{I}(a, b; p) = \min\left\{1, \left((1-a)^p + b^p\right)^{\frac{1}{p}}\right\}. \tag{10.17}$$

The operation of parameterized Yager S-implication is illustrated by Fig. 10.9.

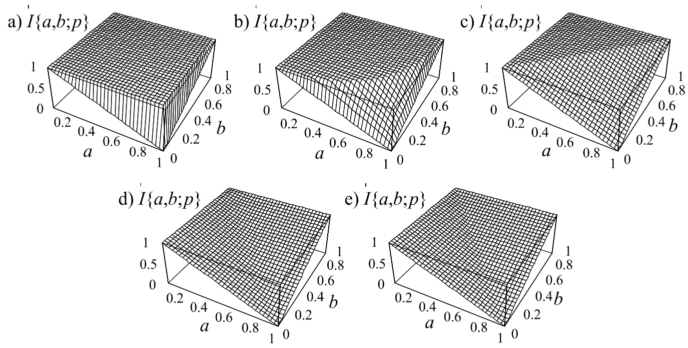

FIGURE 10.9. Hyperplanes of function (10.17) for a) $p = 0.1$, b) $p = 0.5$, c) $p = 1.0$, d) $p = 10.0$, e) $p = 100.0$

To construct Mamdani systems, we can use the following parameterized triangular norms:

- $\overset{\leftrightarrow}{T}_1 \{a_1, a_2, \ldots, a_n; p^\tau\}$ to aggregate the premises in particular rules;
- $\overset{\leftrightarrow}{T}_2 \left\{b_1, b_2; p^I\right\}$ to combine the premises and consequents of the rules;
- $\overset{\leftrightarrow}{S} \{c_1, c_2, \ldots, c_N; p^{\mathrm{agr}}\}$ to aggregate the rules,

where n is the number of inputs and N is the number of rules.

In order to construct logical systems using the S-implication, we can use the following parameterized triangular norms:

- $\overset{\leftrightarrow}{T}_1 \{a_1, a_2, \ldots, a_n; p^\tau\}$ to aggregate the premises in particular rules;
- $\overset{\leftrightarrow}{S} \left\{1 - b_1, b_2; p^I\right\}$ to combine the premises and consequents of the rules;
- $\overset{\leftrightarrow}{T}_2 \{c_1, c_2, \ldots, c_N; p^{\mathrm{agr}}\}$ to aggregate the rules,

where n is the number of inputs and N is the number of rules.

It should be emphasized that parameters p^τ, p^I and p^{agr} can be found in the process of learning.

10.4 Adjustable triangular norms

We will build the function $H(\mathbf{a}; \nu)$ which, depending on the value of the parameter ν, takes the form of t-norm or t-conorm. To construct this function we will use the compromise operator defined below.

Definition 10.1
Function

$$\widetilde{N}_\nu : [0,1] \to [0,1] \tag{10.18}$$

defined as

$$\widetilde{N}_\nu(a) = (1 - \nu) N(a) + \nu a \tag{10.19}$$

is called a *compromise operator*, where $\nu \in [0,1]$ and $N(a) = \widetilde{N}_0(a) = 1 - a$.

It could be observed that $\widetilde{N}_{1-\nu}(a) = \widetilde{N}_\nu(1-a) = 1 - \widetilde{N}_\nu(a)$ and

$$\widetilde{N}_\nu(a) = \begin{cases} N(a) & \text{for} \quad \nu = 0, \\ \frac{1}{2} & \text{for} \quad \nu = \frac{1}{2}, \\ a & \text{for} \quad \nu = 1. \end{cases} \tag{10.20}$$

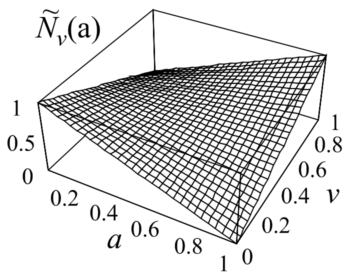

FIGURE 10.10. Illustration of the operation of the compromise operator (10.19)

Function $\widetilde{N}_\nu$ for $\nu = 0$ is *a strong type negation*. Its operation is illustrated by Fig. 10.10.

Definition 10.2
Function

$$H : [0,1]^n \rightarrow [0,1] \tag{10.21}$$

defined as

$$H(\mathbf{a};\nu) = \widetilde{N}_\nu \left(\overset{n}{\underset{i=1}{S}} \left\{ \widetilde{N}_\nu (a_i) \right\} \right) = \widetilde{N}_{1-\nu} \left(\overset{n}{\underset{i=1}{T}} \left\{ \widetilde{N}_{1-\nu} (a_i) \right\} \right) \tag{10.22}$$

is called H*-function*, where $\nu \in [0,1]$.

Theorem 10.1
Let T and S be dual triangular norms. Then function H, defined by formula (10.22), changes its shape from the t-norm to the t-conorm, when ν changes from 0 to 1.

Proof. The assumption says that

$$T\{\mathbf{a}\} = N\left(S\left\{N(a_1), N(a_2), \ldots, N(a_n)\right\}\right). \tag{10.23}$$

For $\nu = 0$ formula (10.23) can be notated as follows:

$$T\{\mathbf{a}\} = \widetilde{N}_0 \left(S \left\{ \widetilde{N}_0(a_1), \widetilde{N}_0(a_2), \ldots, \widetilde{N}_0(a_n) \right\} \right). \tag{10.24}$$

At the same time

$$S\{\mathbf{a}\} = \widetilde{N}_1 \left(S \left\{ \widetilde{N}_1(a_1), \widetilde{N}_1(a_2), \ldots, \widetilde{N}_1(a_n) \right\} \right) \tag{10.25}$$

for $\nu = 1$. The right sides of formulas (10.24) and (10.25) can be notated as follows:

$$H(\mathbf{a};\nu) = \widetilde{N}_\nu \left(\overset{n}{\underset{i=1}{S}} \left\{ \widetilde{N}_\nu (a_i) \right\} \right) \tag{10.26}$$

for, respectively, $\nu = 0$ and $\nu = 1$. If parameter ν changes its value from 0 to 1, then function H is smoothly switched between the t-norm and the t-conorm. It could easily be observed that:

$$H(\mathbf{a};\nu) = \begin{cases} T\{\mathbf{a}\} & \text{for} \quad \nu = 0\,, \\ \dfrac{1}{2} & \text{for} \quad \nu = \dfrac{1}{2}\,, \\ S\{\mathbf{a}\} & \text{for} \quad \nu = 1. \end{cases} \tag{10.27}$$

Example 10.5
The adjustable H-function constructed with the use of Zadeh t-norm or t-conorm takes the form

$$\begin{aligned} H(a_1,a_2;\nu) &= \widetilde{N}_{1-\nu}\left(\min\left\{\widetilde{N}_{1-\nu}(a_1)\,,\widetilde{N}_{1-\nu}(a_2)\right\}\right) \\ &= \widetilde{N}_{\nu}\left(\max\left\{\widetilde{N}_{\nu}(a_1)\,,\widetilde{N}_{\nu}(a_2)\right\}\right), \end{aligned} \tag{10.28}$$

while ν changes from value 0 to 1. It could easily be observed that:

$$H(a_1,a_2;0) = T\{a_1,a_2\} = \min\{a_1,a_2\}\,, \tag{10.29}$$

$$H(a_1,a_2;1) = S\{a_1,a_2\} = \max\{a_1,a_2\}\,. \tag{10.30}$$

The operation of Zadeh H-function is illustrated by Fig. 10.11.

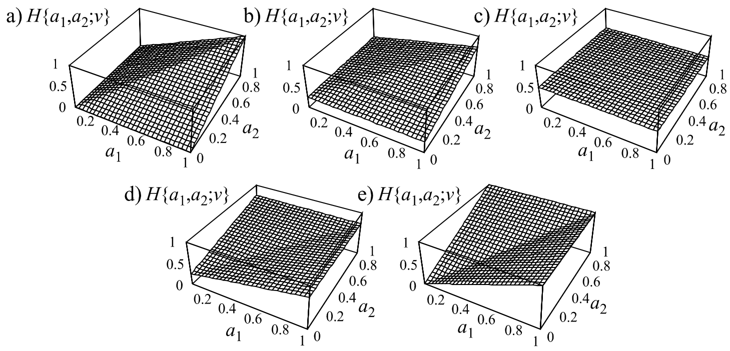

FIGURE 10.11. Hyperplanes of function (10.28) for a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$

Example 10.6
The adjustable H-function constructed with the use of algebraic t-norm or t-conorm takes the form:

$$\begin{aligned} H(a_1,a_2;\nu) &= \widetilde{N}_{1-\nu}\left(\widetilde{N}_{1-\nu}(a_1)\,\widetilde{N}_{1-\nu}(a_1)\right) \\ &= \widetilde{N}_{\nu}\left(1-\left(1-\widetilde{N}_{\nu}(a_1)\right)\left(1-\widetilde{N}_{\nu}(a_1)\right)\right), \end{aligned} \tag{10.31}$$

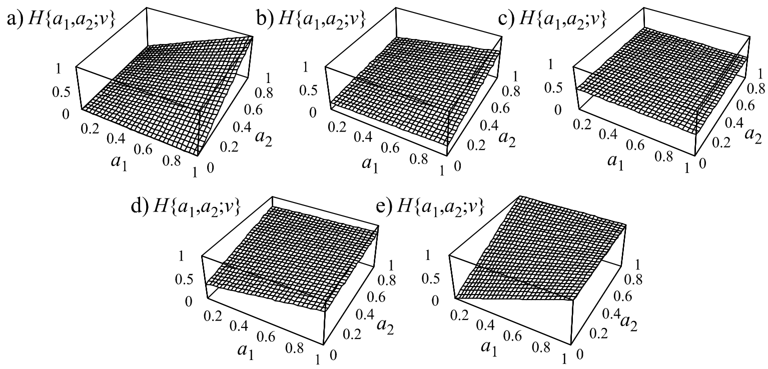

FIGURE 10.12. Hyperplanes of function (10.31) for a) $v = 0.00$, b) $v = 0.15$, c) $v = 0.50$, d) $v = 0.85$, e) $v = 1.00$

while ν changes from value 0 to 1. It could easily be observed that:

$$T\{a_1, a_2\} = H(a_1, a_2; 0) = a_1 a_2, \tag{10.32}$$

$$S\{a_1, a_2\} = H(a_1, a_2; 1) = a_1 + a_2 - a_1 a_2. \tag{10.33}$$

The operation of algebraic H-function is illustrated by Fig. 10.12.

Now we will construct the so-called H-implication which may be switched between the "correlation function" (t-norm) and fuzzy implication (S-implication).

Theorem 10.2
Let T and S be dual triangular norms. Then the H-implication defined as follows:

$$I(a, b; \nu) = H\left(\widetilde{N}_{1-\nu}(a), b; \nu\right) \tag{10.34}$$

changes from the "engineering implication"

$$I_{\text{cor}}(a, b) = I(a, b; 0) = T\{a, b\} \tag{10.35}$$

to the fuzzy implication

$$I_{\text{fuzzy}}(a, b) = I(a, b; 1) = S\{1 - a, b\} \tag{10.36}$$

when parameter ν changes its value from 0 to 1.

Proof. Theorem 10.2 is a direct consequence of Theorem 10.1.

Example 10.7

The adjustable H-implication which may be switched between the "correlation function" expressed by the Zadeh t-norm

$$\begin{aligned} I_{\text{eng}}(a,b) &= H(a,b;0) \\ &= T\{a,b\} \\ &= \min\{a,b\} \end{aligned} \tag{10.37}$$

and binary S-implication

$$\begin{aligned} I_{\text{fuzzy}}(a,b) &= H\left(\widetilde{N}_0(a), b; 1\right) \\ &= S\{N(a), b\} \\ &= \max\{N(a), b\} \end{aligned} \tag{10.38}$$

may be expressed as follows:

$$I(a,b;\nu) = H\left(\widetilde{N}_{1-\nu}(a), b; \nu\right), \tag{10.39}$$

while ν changes from 0 to 1.The operation of H-implication given by formula (10.39) is illustrated by Fig. 10.13.

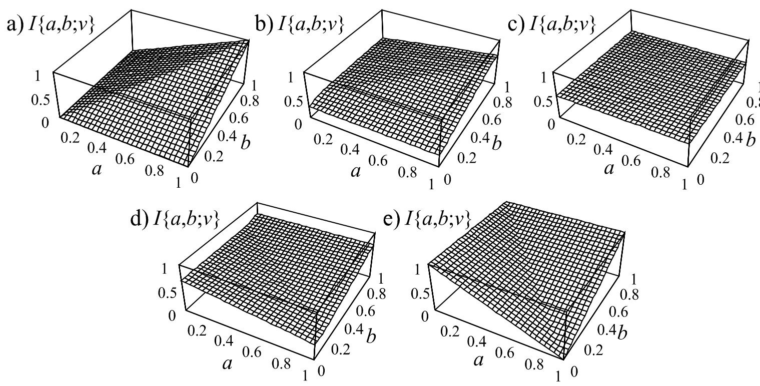

FIGURE 10.13. Hyperplanes of function (10.39) for a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$

Example 10.8

The adjustable H-implication which may be switched between the "correlation function" expressed by algebraic t-norm

$$\begin{aligned} I_{\text{eng}}(a,b) &= H(a,b;0) \\ &= T\{a,b\} \\ &= ab, \end{aligned} \tag{10.40}$$

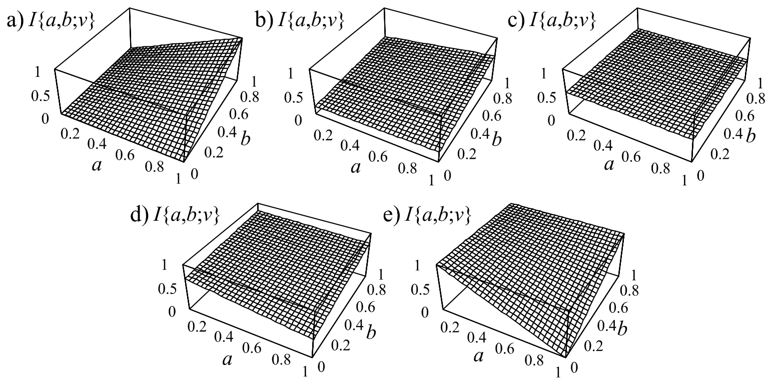

FIGURE 10.14. Hyperplanes of function (10.42) for a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$

and binary S-implication

$$I_{\text{fuzzy}}(a,b) = H\left(\widetilde{N}_0(a), b; 1\right) \tag{10.41}$$
$$= S\{N(a), b\}$$
$$= 1 - a + ab,$$

may be expressed as follows:

$$I(a,b;\nu) = H\left(\widetilde{N}_{1-\nu}(a), b; \nu\right), \tag{10.42}$$

while ν changes from 0 to 1. The operation of H-implication given by formula (10.42) is illustrated by Fig. 10.14.

10.5 Flexible systems

Using the concept of adjustable triangular norms and adjustable implications, we will build a neuro-fuzzy system the structure of which can change between the system of Mamdani type and the logical type system.

Theorem 10.3
Let T and S be dual triangular norms. Then the neuro-fuzzy system

$$\tau_k(\overline{\mathbf{x}}) = H\left(\begin{array}{c}\mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n); \\ 0\end{array}\right), \tag{10.43}$$

$$I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r) = H\left(\begin{array}{c}\widetilde{N}_{1-\nu}(\tau_k(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r); \\ \nu\end{array}\right), \tag{10.44}$$

$$\operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r) = H\begin{pmatrix} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ 1-\nu \end{pmatrix}, \tag{10.45}$$

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\sum_{r=1}^{N} \operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)} \tag{10.46}$$

changes between the Mamdani type system ($\nu = 0$) and the logical type system ($\nu = 1$) together with the change of parameter ν from 0 to 1.

Proof. For $\nu = 0$ formula (10.46) takes the form

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{N}\left\{T\left\{\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\right\}\right\}}{\sum_{r=1}^{N} S_{k=1}^{N}\left\{T\left\{\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\right\}\right\}}. \tag{10.47}$$

It could easily be observed that the above formula describes the Mamdani type system. For $\nu = 1$ we have

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot T_{k=1}^{N}\left\{S\left\{N\left(\tau_k(\overline{\mathbf{x}})\right), \mu_{B^k}(\overline{y}^r)\right\}\right\}}{\sum_{r=1}^{N} T_{k=1}^{N}\left\{S\left\{N\left(\tau_k(\overline{\mathbf{x}})\right), \mu_{B^k}(\overline{y}^r)\right\}\right\}}. \tag{10.48}$$

Dependency (10.48) describes a logical system using the S-implication. For the value of parameter $\nu \in (0,1)$ the inference is performed according to the definition of the H-implication, which ends the proof.

Table 10.1 presents implication and aggregation operators for changing parameter ν. The system described by means of dependencies (10.43) - (10.46) is a flexible system as it enables the choice of inference model as a result of the learning process. However, that system does not include the other flexibility aspects described in Subchapters 10.2 and 10.3.

At present the concept of soft triangular norms, parameterized triangular norms, weights of rules and weights of rules premises will be introduced to system (10.46) given in Theorem 10.3. Then the flexible neuro-fuzzy system takes the following form:

$$\tau_k(\overline{\mathbf{x}}) = \begin{pmatrix} (1-\alpha^\tau)\operatorname{avg}\left(\mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n)\right) + \\ +\alpha^\tau \overset{\leftrightarrow}{H}{}^{*}\begin{pmatrix} \mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n); \\ w_{1,k}^\tau, \ldots, w_{n,k}^\tau, p^\tau, 0 \end{pmatrix} \end{pmatrix}, \tag{10.49}$$

TABLE 10.1. Implication and aggregation operators for changing parameter ν

Parameter ν	Implication	Aggregation
$\nu = 0$	$T\{a,b\}$	t-conorma
$\nu = 1$	$S\{1-a,b\}$	t-norma
$0 < \nu < 1$	$H\left(\widetilde{N}_{1-\nu}(a), b; \nu\right)$	$H(a,b;1-\nu)$
$\nu = 0.5$	$H(a,b;0.5) = 0.5$	$H(a,b;0.5) = 0.5$

$$I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)=\left(\begin{array}{c}\left(1-\alpha^{I}\right)\operatorname{avg}\left(\widetilde{N}_{1-\nu}\left(\tau_{k}\left(\overline{\mathbf{x}}\right)\right),\mu_{B^{k}}\left(\overline{y}^{r}\right)\right)+\\+\alpha^{I}\overset{\leftrightarrow}{H}\left(\begin{array}{c}\widetilde{N}_{1-\nu}\left(\tau_{k}\left(\overline{\mathbf{x}}\right)\right),\mu_{B^{k}}\left(\overline{y}^{r}\right);\\p^{I},\nu\end{array}\right)\end{array}\right),\tag{10.50}$$

$$\operatorname{agr}_{r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)=\left(\begin{array}{c}\left(1-\alpha^{\text{agr}}\right)\operatorname{avg}\left(I_{1,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right),\ldots,I_{N,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)\right)+\\+\alpha^{\text{agr}}\overset{\leftrightarrow}{H}^{*}\left(\begin{array}{c}I_{1,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right),\ldots,I_{N,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right);\\w_{1}^{\text{agr}},\ldots,w_{N}^{\text{agr}},p^{\text{agr}},1-\nu\end{array}\right)\end{array}\right).\tag{10.51}$$

In the system described by means of dependencies (10.46) and (10.49) - (10.51) we can distinguish the following parameters:

- $\nu\in[0,1]$, parameter of the type of inference model,
- $\alpha^{\tau}\in[0,1]$, $\alpha^{I}\in[0,1]$, $\alpha^{\text{agr}}\in[0,1]$, flexibility parameters (in the sense of Yager and Filev) in operators of premises aggregation, operators of inference and operators of rules aggregation,
- $p^{\tau}\in[0,\infty)$, $p^{I}\in[0,\infty)$, $p^{\text{agr}}\in[0,\infty)$, parameters of the hyperplanes shape of premises aggregation operators, operators of inference and operators of rules aggregation,
- $w_{i,k}^{\tau}\in[0,1]$, $i=1,\ldots,n$, $k=1,\ldots,N$, weights of rules premises,
- $w_{k}^{\text{agr}}\in[0,1]$, $k=1,\ldots,N$, weights of rules,
- $p_{u,i,k}^{A}$, $u=1,2,\ldots,P^{A}$, $i=1,2,\ldots,n$, parameters of the shape of membership function of input fuzzy sets,
- $p_{1,k}^{B}=\overline{y}^{k}$, $k=1,2,\ldots,N$, centers of membership functions of output fuzzy sets,
- $p_{u,k}^{B}$, $u=2,3,\ldots,P^{B}$, $k=1,2,\ldots,N$, parameters of the shape of membership functions of output fuzzy sets.

The above mentioned parameters will be subject to learning in the following subchapter.

10.6 Learning algorithms

Now we will derive gradient learning algorithms of the system described by means of dependencies (10.46) and (10.49) - (10.51). Those parameters are modified by iteration according to the dependencies below:

$$\nu\left(t+1\right)=\nu\left(t\right)-\eta\Delta\nu\left(t\right),\tag{10.52}$$

$$\alpha^{\tau}\left(t+1\right)=\alpha^{\tau}\left(t\right)-\eta\Delta\alpha^{\tau}\left(t\right)\tag{10.53}$$

$$\alpha^{I}(t+1) = \alpha^{I}(t) - \eta \Delta \alpha^{I}(t), \tag{10.54}$$

$$\alpha^{\mathrm{agr}}(t+1) = \alpha^{\mathrm{agr}}(t) - \eta \Delta \alpha^{\mathrm{agr}}(t) \tag{10.55}$$

$$p^{\tau}(t+1) = p^{\tau}(t) - \eta \Delta p^{\tau}(t), \tag{10.56}$$

$$p^{I}(t+1) = p^{I}(t) - \eta \Delta p^{I}(t), \tag{10.57}$$

$$p^{\mathrm{agr}}(t+1) = p^{\mathrm{agr}}(t) - \eta \Delta p^{\mathrm{agr}}(t), \tag{10.58}$$

$$w^{\tau}_{i,k}(t+1) = w^{\tau}_{i,k}(t) - \eta \Delta w^{\tau}_{i,k}(t), \tag{10.59}$$

$$w^{\mathrm{agr}}_{k}(t+1) = w^{\mathrm{agr}}_{k}(t) - \eta \Delta w^{\mathrm{agr}}_{k}(t), \tag{10.60}$$

$$p^{A}_{u,i,k}(t+1) = p^{A}_{u,i,k}(t) - \eta \Delta p^{A}_{u,i,k}(t), \tag{10.61}$$

$$p^{B}_{u,k}(t+1) = p^{B}_{u,k}(t) - \eta \Delta p^{B}_{u,k}(t); \quad u = 2, \ldots, P^{B}, \tag{10.62}$$

$$\overline{y}^{r}(t+1) = p^{B}_{1,r}(t+1) = \overline{y}^{r}(t) - \eta \Delta \overline{y}^{r}(t). \tag{10.63}$$

The terms Δ in the above dependencies are defined as follows:

$$\Delta \nu = \sum_{k=1}^{N} \sum_{r=1}^{N} \varepsilon^{I}_{k,r}\{\nu\} + \sum_{r=1}^{N} \varepsilon^{\mathrm{agr}}_{r}\{\nu\}, \tag{10.64}$$

$$\Delta \alpha^{\tau} = \sum_{k=1}^{N} \varepsilon^{\tau}_{k}\{\alpha^{\tau}\}, \tag{10.65}$$

$$\Delta \alpha^{I} = \sum_{k=1}^{N} \sum_{r=1}^{N} \varepsilon^{I}_{k,r}\left\{\alpha^{I}\right\}, \tag{10.66}$$

$$\Delta \alpha^{\mathrm{agr}} = \sum_{r=1}^{N} \varepsilon^{\mathrm{agr}}_{r}\{\alpha^{\mathrm{agr}}\}, \tag{10.67}$$

$$\Delta p^{\tau} = \sum_{k=1}^{N} \varepsilon^{\tau}_{k}\{p^{\tau}\}, \tag{10.68}$$

$$\Delta p^{I} = \sum_{k=1}^{N} \sum_{r=1}^{N} \varepsilon^{I}_{k,r}\left\{p^{I}\right\}, \tag{10.69}$$

$$\Delta p^{\mathrm{agr}} = \sum_{r=1}^{N} \varepsilon^{\mathrm{agr}}_{r}\{p^{\mathrm{agr}}\}, \tag{10.70}$$

$$\Delta w^{\tau}_{i,k} = \varepsilon^{\tau}_{k}\left\{w^{\tau}_{i,k}\right\}, \tag{10.71}$$

$$\Delta w^{\mathrm{agr}}_{k} = \sum_{r=1}^{N} \varepsilon^{\mathrm{agr}}_{r}\{w^{\mathrm{agr}}_{k}\}, \tag{10.72}$$

$$\Delta p^A_{u,i,k} = \varepsilon^\tau_k \left\{ p^A_{u,i,k} \right\}, \tag{10.73}$$

$$\Delta p^B_{u,k} = \sum_{r=1}^{N} \varepsilon^I_{k,r} \left\{ p^B_{u,k} \right\}; \quad u = 2, \ldots, P^B, \tag{10.74}$$

$$\Delta \overline{y}^r = \Delta p^B_{1,r} = \varepsilon^{\mathrm{def}} \left\{ \overline{y}^r \right\} + \sum_{k=1}^{N} \varepsilon^I_{k,r} \left\{ \overline{y}^r \right\} + \sum_{k=1}^{N} \varepsilon^I_{r,k} \left\{ p^B_{1,r} \right\}. \tag{10.75}$$

The errors propagated through particular system layers are defined similarly to the learning algorithms related to non-flexible systems which have been described in point 9.6. The method of error propagation is illustrated in Fig. 9.11.

The errors propagated by blocks of rules activation are defined as follows (Fig. 10.15):

$$\varepsilon^\tau_k \left\{ \alpha^\tau \right\} = \varepsilon^\tau_k \frac{\partial \tau_k \left(\overline{\mathbf{x}} \right)}{\partial \alpha^\tau}, \tag{10.76}$$

$$\varepsilon^\tau_k \left\{ p^\tau \right\} = \varepsilon^\tau_k \frac{\partial \tau_k \left(\overline{\mathbf{x}} \right)}{\partial b^\tau_k \left(\overline{\mathbf{x}} \right)} \frac{\partial b^\tau_k \left(\overline{\mathbf{x}} \right)}{\partial p^\tau}, \tag{10.77}$$

$$\varepsilon^\tau_k \left\{ w^\tau_{i,k} \right\} = \varepsilon^\tau_k \frac{\partial \tau_k \left(\overline{\mathbf{x}} \right)}{\partial b^\tau_k \left(\overline{\mathbf{x}} \right)} \frac{\partial b^\tau_k \left(\overline{\mathbf{x}} \right)}{\partial w^\tau_{i,k}}, \tag{10.78}$$

$$\varepsilon^\tau_k \left\{ p^A_{u,i,k} \right\} = \varepsilon^\tau_k \left(\begin{array}{c} \dfrac{\partial \tau_k \left(\overline{\mathbf{x}} \right)}{\partial b^\tau_k \left(\overline{\mathbf{x}} \right)} \dfrac{\partial b^\tau_k \left(\overline{\mathbf{x}} \right)}{\partial \mu_{A^k_i} \left(\overline{x}_i \right)} + \\ \\ + \dfrac{\partial \tau_k \left(\overline{\mathbf{x}} \right)}{\partial a^\tau_k \left(\overline{\mathbf{x}} \right)} \dfrac{\partial a^\tau_k \left(\overline{\mathbf{x}} \right)}{\partial \mu_{A^k_i} \left(\overline{x}_i \right)} \end{array} \right) \frac{\partial \mu_{A^k_i} \left(\overline{x}_i \right)}{\partial p^A_{u,i,k}}, \tag{10.79}$$

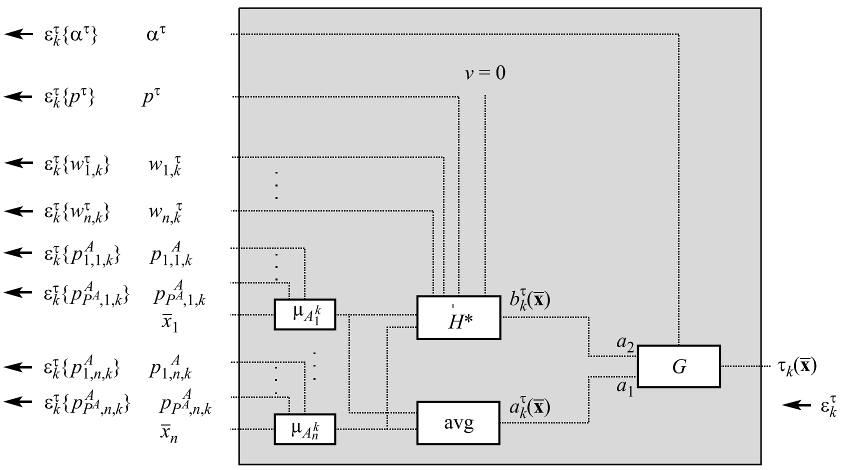

FIGURE 10.15. Block of rules activation of a flexible system

while

$$\frac{\partial \tau_k(\overline{\mathbf{x}})}{\partial a_k^\tau(\overline{\mathbf{x}})} = \frac{\partial}{\partial a_k^\tau(\overline{\mathbf{x}})} G\left(\begin{array}{c} a_k^\tau(\overline{\mathbf{x}}), b_k^\tau(\overline{\mathbf{x}}); \\ \alpha^\tau \end{array}\right), \tag{10.80}$$

$$\frac{\partial \tau_k(\overline{\mathbf{x}})}{\partial b_k^\tau(\overline{\mathbf{x}})} = \frac{\partial}{\partial b_k^\tau(\overline{\mathbf{x}})} G\left(\begin{array}{c} a_k^\tau(\overline{\mathbf{x}}), b_k^\tau(\overline{\mathbf{x}}); \\ \alpha^\tau \end{array}\right), \tag{10.81}$$

$$\frac{\partial \tau_k(\overline{\mathbf{x}})}{\partial \alpha^\tau} = \frac{\partial}{\partial \alpha^\tau} G\left(\begin{array}{c} a_k^\tau(\overline{\mathbf{x}}), b_k^\tau(\overline{\mathbf{x}}); \\ \alpha^\tau \end{array}\right), \tag{10.82}$$

$$\frac{\partial a_k^\tau(\overline{\mathbf{x}})}{\partial \mu_{A_i^k}(\overline{x}_i)} = \frac{\partial}{\partial \mu_{A_i^k}(\overline{x}_i)} \operatorname{avg}\left(\mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n)\right), \tag{10.83}$$

$$\frac{\partial b_k^\tau(\overline{\mathbf{x}})}{\partial p^\tau} = \frac{\partial}{\partial p^\tau} \overset{\leftrightarrow}{H}{}^{*}\left(\begin{array}{c} \mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n); \\ w_{1,k}^\tau, \ldots, w_{n,k}^\tau, p^\tau, 0 \end{array}\right), \tag{10.84}$$

$$\frac{\partial b_k^\tau(\overline{\mathbf{x}})}{\partial w_{i,k}^\tau} = \frac{\partial}{\partial w_{i,k}^\tau} \overset{\leftrightarrow}{H}{}^{*}\left(\begin{array}{c} \mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n); \\ w_{1,k}^\tau, \ldots, w_{n,k}^\tau, p^\tau, 0 \end{array}\right), \tag{10.85}$$

$$\frac{\partial b_k^\tau(\overline{\mathbf{x}})}{\partial \mu_{A_i^k}(\overline{x}_i)} = \frac{\partial}{\partial \mu_{A_i^k}(\overline{x}_i)} \overset{\leftrightarrow}{H}{}^{*}\left(\begin{array}{c} \mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n); \\ w_{1,k}^\tau, \ldots, w_{n,k}^\tau, p^\tau, 0 \end{array}\right). \tag{10.86}$$

The derivatives in the above dependencies are determined with use of the formulas specified in the further part of this subchapter.

The errors propagated by blocks of implications are defined as follows (Fig. 10.16):

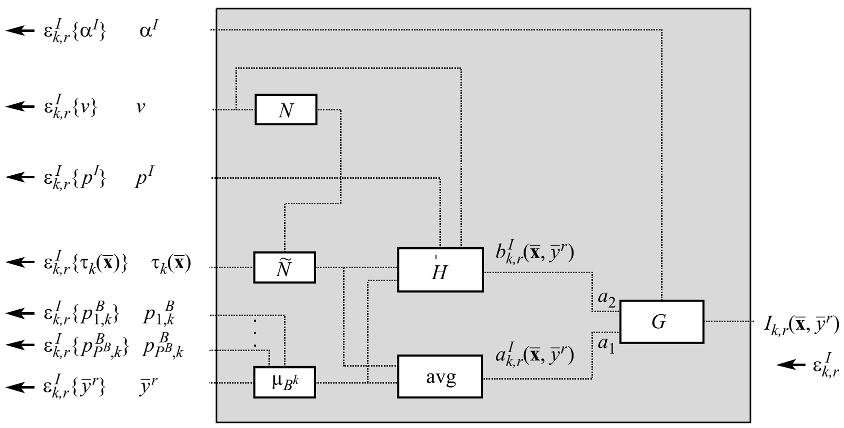

FIGURE 10.16. Block of implications of a flexible system

$$\varepsilon_{k,r}^{I}\left\{\nu\right\}=\varepsilon_{k,r}^{I}\left(\begin{array}{c}\dfrac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}\dfrac{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial\nu}+\\ +\left(\begin{array}{c}\dfrac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}\dfrac{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial\widetilde{N}_{1-\nu}\left(\tau_{k}\left(\overline{\mathbf{x}}\right)\right)}+\\ +\dfrac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}\dfrac{\partial a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial\widetilde{N}_{1-\nu}\left(\tau_{k}\left(\overline{\mathbf{x}}\right)\right)}\end{array}\right)\cdot\\ \cdot\dfrac{\partial\widetilde{N}_{1-\nu}\left(\tau_{k}\left(\overline{\mathbf{x}}\right)\right)}{\partial\left(1-\nu\right)}\dfrac{\partial N\left(\nu\right)}{\partial\nu}\end{array}\right),\tag{10.87}$$

$$\varepsilon_{k,r}^{I}\left\{\alpha^{I}\right\}=\varepsilon_{k,r}^{I}\frac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial\alpha^{I}},\tag{10.88}$$

$$\varepsilon_{k,r}^{I}\left\{p^{I}\right\}=\varepsilon_{k,r}^{I}\frac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}\frac{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial p^{I}},\tag{10.89}$$

$$\varepsilon_{k,r}^{I}\left\{p_{u,k}^{B}\right\}=\varepsilon_{k,r}^{I}\left(\begin{array}{c}\dfrac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}\dfrac{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}+\\ +\dfrac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}\dfrac{\partial a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}\end{array}\right)\frac{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}{\partial p_{u,k}^{B}},\tag{10.90}$$

$$\varepsilon_{k,r}^{I}\left\{\overline{y}^{r}\right\}=\varepsilon_{k,r}^{I}\left(\begin{array}{c}\dfrac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}\dfrac{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}+\\ +\dfrac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}\dfrac{\partial a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}\end{array}\right)\frac{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}{\partial\overline{y}^{r}},\tag{10.91}$$

$$\varepsilon_{k,r}^{I}\{\tau_{k}(\overline{\mathbf{x}})\}=\varepsilon_{k,r}^{I}\left(\begin{array}{c}\dfrac{\partial I_{k,r}(\overline{\mathbf{x}},\overline{y}^{r})}{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}\dfrac{\partial b_{k,r}^{I}(\overline{\mathbf{x}},\overline{y}^{r})}{\partial\widetilde{N}_{1-\nu}\left(\tau_{k}\left(\overline{\mathbf{x}}\right)\right)}+\\ +\dfrac{\partial I_{k,r}(\overline{\mathbf{x}},\overline{y}^{r})}{\partial a_{k,r}^{I}(\overline{\mathbf{x}},\overline{y}^{r})}\dfrac{\partial a_{k,r}^{I}(\overline{\mathbf{x}},\overline{y}^{r})}{\partial\widetilde{N}_{1-\nu}(\tau_{k}\left(\overline{\mathbf{x}}\right))}\end{array}\right)\frac{\partial\widetilde{N}_{1-\nu}(\tau_{k}\left(\overline{\mathbf{x}}\right))}{\partial\tau_{k}(\overline{\mathbf{x}})},\tag{10.92}$$

while

$$\frac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}=\frac{\partial}{\partial a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}G\left(\begin{array}{c}a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right),b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right);\\ \alpha^{I}\end{array}\right),\tag{10.93}$$

$$\frac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}=\frac{\partial}{\partial b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right)}G\left(\begin{array}{c}a_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right),b_{k,r}^{I}\left(\overline{\mathbf{x}},\overline{y}^{r}\right);\\ \alpha^{I}\end{array}\right),\tag{10.94}$$

$$\frac{\partial I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^r\right)}{\partial \alpha^I}=\frac{\partial}{\partial \alpha^I}G\left(\begin{array}{c} a^I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^r\right),b^I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^r\right); \\ \alpha^I \end{array}\right), \tag{10.95}$$

$$\frac{\partial a^I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^r\right)}{\partial \widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}=\frac{\partial}{\partial \widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\operatorname{avg}\left(\widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right),\mu_{B^k}\left(\overline{y}^r\right)\right), \tag{10.96}$$

$$\frac{\partial a^I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^r\right)}{\partial \mu_{B^k}\left(\overline{y}^r\right)}=\frac{\partial}{\partial \mu_{B^k}\left(\overline{y}^r\right)}\operatorname{avg}\left(\widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right),\mu_{B^k}\left(\overline{y}^r\right)\right), \tag{10.97}$$

$$\frac{\partial b^I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^r\right)}{\partial \nu}=\frac{\partial}{\partial \nu}\overset{\leftrightarrow}{H}\left(\begin{array}{c} \widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right),\mu_{B^k}\left(\overline{y}^r\right); \\ p^I,\nu \end{array}\right), \tag{10.98}$$

$$\frac{\partial b^I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^r\right)}{\partial p^I}=\frac{\partial}{\partial p^I}\overset{\leftrightarrow}{H}\left(\begin{array}{c} \widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right),\mu_{B^k}\left(\overline{y}^r\right); \\ p^I,\nu \end{array}\right), \tag{10.99}$$

$$\frac{\partial b^I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^r\right)}{\partial \widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}=\frac{\partial}{\partial \widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\overset{\leftrightarrow}{H}\left(\begin{array}{c} \widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right),\mu_{B^k}\left(\overline{y}^r\right); \\ p^I,\nu \end{array}\right), \tag{10.100}$$

$$\frac{\partial b^I_{k,r}\left(\overline{\mathbf{x}},\overline{y}^r\right)}{\partial \mu_{B^k}\left(\overline{y}^r\right)}=\frac{\partial}{\partial \mu_{B^k}\left(\overline{y}^r\right)}\overset{\leftrightarrow}{H}\left(\begin{array}{c} \widetilde{N}_{1-\nu}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right),\mu_{B^k}\left(\overline{y}^r\right); \\ p^I,\nu \end{array}\right). \tag{10.101}$$

The derivatives in the above dependencies are determined with use of the formulas specified in the further part of this subchapter.

The errors propagated by blocks of aggregation are defined as follows (Fig. 10.17):

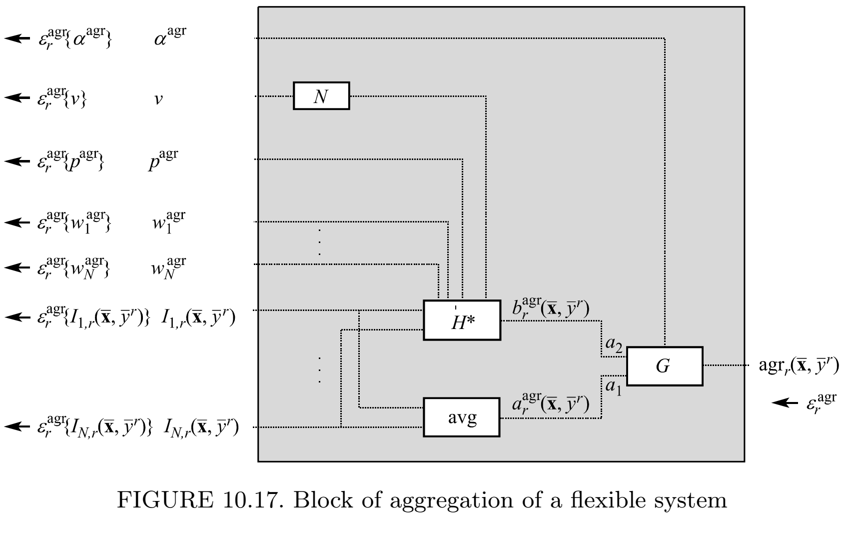

FIGURE 10.17. Block of aggregation of a flexible system

$$\varepsilon_r^{\mathrm{agr}}\{\nu\} = \varepsilon_r^{\mathrm{agr}} \frac{\partial \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)} \frac{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial(1-\nu)} \frac{\partial N(\nu)}{\partial \nu}, \tag{10.102}$$

$$\varepsilon_r^{\mathrm{agr}}\{\alpha^{\mathrm{agr}}\} = \varepsilon_r^{\mathrm{agr}} \frac{\partial \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\partial \alpha^{\mathrm{agr}}}, \tag{10.103}$$

$$\varepsilon_r^{\mathrm{agr}}\{p^{\mathrm{agr}}\} = \varepsilon_r^{\mathrm{agr}} \frac{\partial \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)} \frac{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial p^{\mathrm{agr}}}, \tag{10.104}$$

$$\varepsilon_r^{\mathrm{agr}}\{w_k^{\mathrm{agr}}\} = \varepsilon_r^{\mathrm{agr}} \frac{\partial \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)} \frac{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial w_k^{\mathrm{agr}}}, \tag{10.105}$$

$$\varepsilon_r^{\mathrm{agr}}\{I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)\} = \varepsilon_r^{\mathrm{agr}} \left(\begin{array}{c} \dfrac{\partial \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)} \dfrac{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)} + \\ + \dfrac{\partial \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\partial a_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)} \dfrac{\partial a_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)} \end{array} \right), \tag{10.106}$$

while

$$\frac{\partial \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\partial a_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)} = \frac{\partial}{\partial a_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)} G \left(\begin{array}{c} a_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r), b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r); \\ \alpha^{\mathrm{agr}} \end{array} \right), \tag{10.107}$$

$$\frac{\partial \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)} = \frac{\partial}{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)} G \left(\begin{array}{c} a_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r), b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r); \\ \alpha^{\mathrm{agr}} \end{array} \right), \tag{10.108}$$

$$\frac{\partial \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\partial \alpha^{\mathrm{agr}}} = \frac{\partial}{\partial \alpha^{\mathrm{agr}}} G \left(\begin{array}{c} a_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r), b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r); \\ \alpha^{\mathrm{agr}} \end{array} \right), \tag{10.109}$$

$$\frac{\partial a_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)} = \frac{\partial}{\partial I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)} \mathrm{avg}\left(I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r)\right), \tag{10.110}$$

$$\frac{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial(1-\nu)} = \frac{\partial}{\partial(1-\nu)} \overset{\leftrightarrow}{H}^* \left(\begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\mathrm{agr}}, \ldots, w_N^{\mathrm{agr}}, p^{\mathrm{agr}}, 1-\nu \end{array} \right), \tag{10.111}$$

$$\frac{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial p^{\mathrm{agr}}} = \frac{\partial}{\partial p^{\mathrm{agr}}} \overset{\leftrightarrow}{H}^* \left(\begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\mathrm{agr}}, \ldots, w_N^{\mathrm{agr}}, p^{\mathrm{agr}}, 1-\nu \end{array} \right), \tag{10.112}$$

$$\frac{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial w_k^{\mathrm{agr}}} = \frac{\partial}{\partial w_k^{\mathrm{agr}}} \overset{\leftrightarrow}{H}^* \left(\begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\mathrm{agr}}, \ldots, w_N^{\mathrm{agr}}, p^{\mathrm{agr}}, 1-\nu \end{array} \right), \tag{10.113}$$

$$\frac{\partial b_r^{\mathrm{agr}}(\overline{\mathbf{x}}, \overline{y}^r)}{\partial I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)} = \frac{\partial}{\partial I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)} \overset{\leftrightarrow}{H}^* \left(\begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\mathrm{agr}}, \ldots, w_N^{\mathrm{agr}}, p^{\mathrm{agr}}, 1-\nu \end{array} \right). \tag{10.114}$$

The errors propagated by defuzzification block are defined similarly to the learning algorithms related to non-flexible systems which have been described in Subchapters 9.3 - 9.5.

The learning algorithms derived above of a flexible neuro-fuzzy system require determining derivatives for different types of operators. Below a computation method of those derivatives is presented.

10.6.1 Basic operators

Summation operator

$$y = \sum_{i=1}^{n} x_i \tag{10.115}$$

$$\frac{\partial y}{\partial x_i} = 1 \tag{10.116}$$

Multiplication operator

$$y = \prod_{i=1}^{n} x_i \tag{10.117}$$

$$\frac{\partial y}{\partial x_i} = \prod_{\substack{j=1 \\ j \neq i}}^{n} x_j \tag{10.118}$$

Division operator

$$y = \frac{a}{b} \tag{10.119}$$

$$\frac{\partial y}{\partial a} = \frac{1}{b} \tag{10.120}$$

$$\frac{\partial y}{\partial b} = -\frac{a}{b^2} \tag{10.121}$$

Minimum operator

$$y = \min_{i=1\ldots n} \{x_i\} \tag{10.122}$$

$$\frac{\partial y}{\partial x_i} = \begin{cases} 1 & \text{for} \quad x_i = y \\ 0 & \text{for} \quad x_i \neq y \end{cases} \tag{10.123}$$

Maximum operator

$$y = \max_{i=1\ldots n} \{x_i\} \tag{10.124}$$

$$\frac{\partial y}{\partial x_i} = \begin{cases} 1 & \text{for} \quad x_i = y \\ 0 & \text{for} \quad x_i \neq y \end{cases} \tag{10.125}$$

Compromise operator

$$\widetilde{N}_\nu (a) = (1 - f_z(\nu))(1 - a) + f_z(\nu)\, a \tag{10.126}$$

$$\frac{\partial \widetilde{N}_\nu (a)}{\partial a} = 2 f_z(\nu) - 1 \tag{10.127}$$

$$\frac{\partial \widetilde{N}_{\nu}(a)}{\partial \nu} = (2a-1)\frac{\partial f_z(\nu)}{\partial \nu} \tag{10.128}$$

Arithmetic average operator

$$\operatorname{avg}(a_1, a_2, ..., a_n) = \frac{1}{n}\sum_{i=1}^{n} a_i \tag{10.129}$$

$$\frac{\partial \operatorname{avg}(a_1, a_2, \ldots, a_n)}{\partial a_i} = \frac{1}{n} \tag{10.130}$$

Aggregation operator

$$G(a_1, a_2; \phi) = (1 - f_z(\phi))\, a_1 + f_z(\phi)\, a_2 \tag{10.131}$$

$$\frac{\partial G(a_1, a_2; \phi)}{\partial a_1} = 1 - f_z(\phi) \tag{10.132}$$

$$\frac{\partial G(a_1, a_2; \phi)}{\partial a_2} = f_z(\phi) \tag{10.133}$$

$$\frac{\partial G(a_1, a_2; \phi)}{\partial \phi} = -(a_1 - a_2)\frac{\partial f_z(\phi)}{\partial \phi} \tag{10.134}$$

Defuzzification operator

$$\operatorname{def}(a_1, a_2, \ldots, a_n; w_1, w_2, \ldots, w_n) = \operatorname{def}(\mathbf{a}; \mathbf{w}) = \frac{\sum_{i=1}^{n} w_i a_i}{\sum_{i=1}^{n} a_i} \tag{10.135}$$

$$\frac{\partial \operatorname{def}(\mathbf{a}; \mathbf{w})}{\partial a_j} = (w_j - \operatorname{def}(\mathbf{a}; \mathbf{w}))\frac{1}{\sum_{i=1}^{n} a_i} \tag{10.136}$$

$$\frac{\partial \operatorname{def}(\mathbf{a}; \mathbf{w})}{\partial w_j} = \left(a_j - \operatorname{def}(\mathbf{a}; \mathbf{w})\frac{\partial a_j}{\partial w_j}\right)\frac{1}{\sum_{i=1}^{n} a_i} \tag{10.137}$$

10.6.2 Membership functions

Gaussian membership function

$$\mu_A(x) = \exp\left(-\left(\frac{x - \overline{x}}{\sigma}\right)^2\right) \tag{10.138}$$

$$\frac{\partial \mu_A(x)}{\partial x} = -\mu_A(x)\frac{2(x - \overline{x})}{\sigma^2} \tag{10.139}$$

$$\frac{\partial \mu_A(x)}{\partial \overline{x}} = \mu_A(x) \frac{2(x-\overline{x})}{\sigma^2} \tag{10.140}$$

$$\frac{\partial \mu_A(x)}{\partial \sigma} = \mu_A(x) \frac{2(x-\overline{x})^2}{\sigma^3} \tag{10.141}$$

Triangular membership function

$$\mu_A(x) = \begin{cases} 0 & \text{for} \quad x \le a \text{ or } x \ge c \\ \dfrac{x-a}{b-a} & \text{for} \quad a \le x \le b \\ \dfrac{c-x}{c-b} & \text{for} \quad b \le x \le c \end{cases} \tag{10.142}$$

$$\frac{\partial \mu_A(x)}{\partial x} = \begin{cases} 0 & \text{for} \quad x < a \text{ or } x > c \\ \dfrac{1}{2(b-a)} & \text{for} \quad x = a \\ \dfrac{1}{b-a} & \text{for} \quad a < x < b \\ \dfrac{c-2b+a}{2(c-b)(b-a)} & \text{for} \quad x = b \\ -\dfrac{1}{c-b} & \text{for} \quad b < x < c \\ -\dfrac{1}{2(c-b)} & \text{for} \quad x = c \end{cases} \tag{10.143}$$

$$\frac{\partial \mu_A(x)}{\partial a} = \begin{cases} 0 & \text{for} \quad x \le a \text{ or } x > b \\ \dfrac{1}{2(b-a)} & \text{for} \quad x = b \\ \dfrac{x-a}{(b-a)^2} & \text{for} \quad a \le x < b \end{cases} \tag{10.144}$$

$$\frac{\partial \mu_A(x)}{\partial b} = \begin{cases} 0 & \text{for} \quad x \le a \text{ or } x \ge c \\ \dfrac{a-x}{(b-a)^2} & \text{for} \quad a \le x < b \\ \dfrac{a-2b+c}{2(c-b)(b-a)} & \text{for} \quad x = b \\ \dfrac{c-x}{(c-b)^2} & \text{for} \quad b < x \le c \end{cases} \tag{10.145}$$

$$\frac{\partial \mu_A(x)}{\partial c} = \begin{cases} 0 & \text{for} \quad x \le b \text{ or } x > c \\ \dfrac{1}{2(c-b)} & \text{for} \quad x = c \\ \dfrac{x-b}{(c-b)^2} & \text{for} \quad b < x < c \end{cases} \tag{10.146}$$

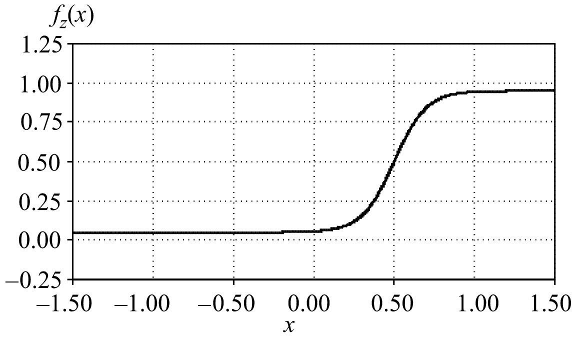

FIGURE 10.18. Plot of function (10.147) for $p_{z1} = 10$, $p_{z2} = 5$, $p_{z3} = 0.9$, $p_{z4} = 0.05$

10.6.3 Constraints

Constraints for parameters $\nu \in [0,1]$, $\lambda \in [0,1]$, $\alpha^{\tau} \in [0,1]$, $\alpha^{I} \in [0,1]$, $\alpha^{\mathrm{agr}} \in [0,1]$, $w_{i,k}^{\tau} \in [0,1]$, $i = 1,\ldots,n$, $k = 1,\ldots,N$, $w_{k}^{\mathrm{agr}} \in [0,1]$, $k = 1,\ldots,N$

$$f_z(x) = \frac{p_{z3}}{1+\exp\left(-\left(p_{z1}x - p_{z2}\right)\right)} + p_{z4} \tag{10.147}$$

$$\frac{\partial f_z(x)}{\partial x} = -\frac{p_{z1}}{p_{z3}}\left(p_{z3} + p_{z4} - f_z(x)\right)\left(p_{z4} - f_z(x)\right) \tag{10.148}$$

Constraints for parameters $p^{\tau} \in [0,\infty)$, $p^{I} \in [0,\infty)$, $p^{\mathrm{agr}} \in [0,\infty)$

$$f_z(x) = \frac{x}{1+\exp\left(-\left(p_{z1}x - p_{z2}\right)\right)} + p_{z3} \tag{10.149}$$

$$\frac{\partial f_z(x)}{\partial x} = \frac{-p_{z3} + f_z(x)}{x}\left(1 + p_{z1}\left(p_{z3} + x - f_z(x)\right)\right) \tag{10.150}$$

In Figures 10.18 and 10.19 we show plots of functions 10.147 and 10.149, respectively.

10.6.4 H-functions

Argument of *H*-functions

$$\arg_i(a_i, w_i, \nu) = G\begin{pmatrix} N\left(f_z(w_i)\,N(a_i)\right), f_z(w_i)\,a_i; \\ \nu \end{pmatrix} \tag{10.151}$$

$$\frac{\partial \arg_i(a_i, w_i, \nu)}{\partial a_i} = f_z(w_i) \tag{10.152}$$

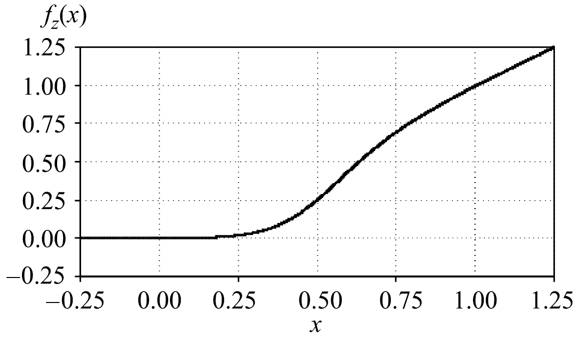

FIGURE 10.19. Plot of function (10.149) for $p_{z1} = 10$, $p_{z2} = 5$, $p_{z3} = 0$

$$\frac{\partial \arg_i (a_i, w_i, \nu)}{\partial w_i} = (a + \nu - 1) \frac{\partial f_z (w_i)}{\partial w_i} \tag{10.153}$$

$$\frac{\partial \arg_i (a_i, w_i, \nu)}{\partial \nu} = f_z (w_i) - 1 \tag{10.154}$$

Zadeh *H*-function

$$H^* (\mathbf{a}; \mathbf{w}, \nu) = \widetilde{N}_\nu \left(\max_{i=1,\ldots,n} \left\{ \widetilde{N}_\nu (\arg_i (a_i, w_i, \nu)) \right\} \right) \tag{10.155}$$

$$H^* (\mathbf{a}; \mathbf{w}, \nu) = \widetilde{N}_\nu (h^* (\mathbf{a}; \mathbf{w}, \nu)) \tag{10.156}$$

where

$$h^* (\mathbf{a}; \mathbf{w}, \nu) = \max_{i=1,\ldots,n} \left\{ \widetilde{N}_\nu (\arg_i (a_i, w_i, \nu)) \right\} \tag{10.157}$$

$$\frac{\partial H^* (\mathbf{a}; \mathbf{w}, \nu)}{\partial a_i} = \begin{cases} (2f_z (\nu) - 1)^2 \cdot \dfrac{\partial \arg_i (a_i, w_i, \nu)}{\partial a_i} \\ \text{for} \quad h^* (\mathbf{a}; \mathbf{w}, \nu) = \widetilde{N}_\nu (\arg_i (a_i, w_i, \nu)) \\ \\ 0 \quad \text{for} \quad h^* (\mathbf{a}; \mathbf{w}, \nu) \neq \widetilde{N}_\nu (\arg_i (a_i, w_i, \nu)) \end{cases} \tag{10.158}$$

$$\frac{\partial H^* (\mathbf{a}; \mathbf{w}, \nu)}{\partial w_i} = \begin{cases} (2f_z (\nu) - 1)^2 \cdot \dfrac{\partial \arg_i (a_i, w_i, \nu)}{\partial w_i} \\ \text{for} \quad h^* (\mathbf{a}; \mathbf{w}, \nu) = \widetilde{N}_\nu (\arg_i (a_i, w_i, \nu)) \\ \\ 0 \quad \text{for} \quad h^* (\mathbf{a}; \mathbf{w}, \nu) \neq \widetilde{N}_\nu (\arg_i (a_i, w_i, \nu)) \end{cases} \tag{10.159}$$

$$\frac{\partial H^*(\mathbf{a};\mathbf{w},\nu)}{\partial \nu} = \frac{\partial f_z(\nu)}{\partial \nu}\left(2h^*(\mathbf{a};\mathbf{w},\nu)-1\right) + \left(2f_z(\nu)-1\right)\max_{i=1,\dots,n}\left\{\begin{array}{c}\left(2f_z(\nu)-1\right)\dfrac{\partial \arg_i(a_i,w_i,\nu)}{\partial \nu}+ \\ +\left(2\arg_i(a_i,w_i,\nu)-1\right)\dfrac{\partial f_z(\nu)}{\partial \nu}\end{array}\right\} \tag{10.160}$$

Algebraic *H*-function

$$H^*(\mathbf{a};\mathbf{w},\nu) = \widetilde{N}_\nu\left(1-\prod_{i=1}^{n}\left(1-\widetilde{N}_\nu\left(\arg_i(a_i,w_i,\nu)\right)\right)\right) \tag{10.161}$$

$$H^*(\mathbf{a};\mathbf{w},\nu) = \widetilde{N}_\nu\left(h^*(\mathbf{a};\mathbf{w},\nu)\right) \tag{10.162}$$

where

$$h^*(\mathbf{a};\mathbf{w},\nu) = 1-\prod_{i=1}^{n}\left(1-\widetilde{N}_\nu\left(\arg_i(a_i,w_i,\nu)\right)\right) \tag{10.163}$$

$$\frac{\partial H^*(\mathbf{a};\mathbf{w},\nu)}{\partial a_i} = \left(2f_z(\nu)-1\right)^2\frac{\partial \arg_i(a_i,w_i,\nu)}{\partial a_i} \cdot\prod_{\substack{u=1\\u\neq i}}^{n}\left(1-\widetilde{N}_\nu\left(\arg_u(a_u,w_u,\nu)\right)\right) \tag{10.164}$$

$$\frac{\partial H^*(\mathbf{a};\mathbf{w},\nu)}{\partial w_i} = \left(2f_z(\nu)-1\right)^2\frac{\partial \arg_i(a_i,w_i,\nu)}{\partial w_i}\cdot \cdot\prod_{\substack{u=1\\u\neq i}}^{n}\left(1-\widetilde{N}_\nu\left(\arg_u(a_u,w_u,\nu)\right)\right) \tag{10.165}$$

$$\frac{\partial H^*(\mathbf{a};\mathbf{w},\nu)}{\partial \nu} = \left(2h^*(\mathbf{a};\mathbf{w},\nu)-1\right)\frac{\partial f_z(\nu)}{\partial \nu}+ + \left(2f_z(\nu)-1\right)\sum_{i=1}^{n}\left(\begin{array}{c}\left(\begin{array}{c}\left(2\arg_i(a_i,w_i,\nu)-1\right)\dfrac{\partial f_z(\nu)}{\partial \nu}+ \\ +\left(2f_z(\nu)-1\right)\dfrac{\partial \arg_i(a_i,w_i,\nu)}{\partial \nu}\end{array}\right)\cdot \\ \cdot\prod_{\substack{u=1\\u\neq i}}^{n}\left(1-\widetilde{N}_\nu\left(\arg_u(a_u,w_u,\nu)\right)\right)\end{array}\right) \tag{10.166}$$

Dombi $\boldsymbol{H}$-function

$$\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)$$
$$=\widetilde{N}_{\nu}\left(1-\left(1+\left(\sum_{i=1}^{n}\left(\widetilde{N}_{\nu}\left(\arg_i\left(a_i,w_i,\nu\right)\right)^{-1}-1\right)^{-f_{z1}(p)}\right)^{\frac{1}{f_{z1}(p)}}\right)^{-1}\right) \tag{10.167}$$

$$\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)=\widetilde{N}_{\nu}\left(1-\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)\right) \tag{10.168}$$

$$p\in(0,\infty) \tag{10.169}$$

where

$$\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)$$
$$=\left(1+\left(\sum_{i=1}^{n}\left(\widetilde{N}_{\nu}\left(\arg_i\left(a_i,w_i,\nu\right)\right)^{-1}-1\right)^{-f_{z1}(p)}\right)^{\frac{1}{f_{z1}(p)}}\right)^{-1} \tag{10.170}$$

$$\begin{aligned}\frac{\partial\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)}{\partial a_i}&=\left(2f_z(\nu)-1\right)^2\cdot\\ &\cdot\frac{\left(\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-1}-1\right)^{1-f_{z1}(p)}}{\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-2}}\cdot\\ &\cdot\frac{\left(\widetilde{N}_{\nu}\left(\arg_i\left(a_i,w_i,\nu\right)\right)^{-1}-1\right)^{-f_{z1}(p)-1}}{\widetilde{N}_{\nu}\left(\arg_i\left(a_i,w_i,\nu\right)\right)^2}\cdot\\ &\cdot\frac{\partial\arg_i\left(a_i,w_i,\nu\right)}{\partial a_i}\end{aligned} \tag{10.171}$$

$$\begin{aligned}\frac{\partial\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)}{\partial w_i}&=\left(2f_z(\nu)-1\right)^2\cdot\\ &\cdot\frac{\left(\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-1}-1\right)^{1-f_{z1}(p)}}{\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-2}}\cdot\\ &\cdot\frac{\left(\widetilde{N}_{\nu}\left(\arg_i\left(a_i,w_i,\nu\right)\right)^{-1}-1\right)^{-f_{z1}(p)-1}}{\widetilde{N}_{\nu}\left(\arg_i\left(a_i,w_i,\nu\right)\right)^2}\cdot\\ &\cdot\frac{\partial\arg_i\left(a_i,w_i,\nu\right)}{\partial w_i}\end{aligned} \tag{10.172}$$

$$\frac{\partial \overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)}{\partial p} = \frac{2f_z(\nu)-1}{f_{z1}(p)} \frac{\left(\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-1}-1\right)^{1-f_{z1}(p)}}{\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-2}} \cdot$$
$$\cdot \left(\sum_{i=1}^{n} \frac{-\ln\left(\tilde{N}_\nu(\arg_i(a_i,w_i,\nu))^{-1}-1\right)}{\left(\tilde{N}_\nu(\arg_i(a_i,w_i,\nu))^{-1}-1\right)^{f_{z1}(p)}} + \atop + \frac{\ln\left(\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-1}-1\right)}{\left(\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-1}-1\right)^{-f_{z1}(p)}} \right) \cdot$$
$$\cdot \frac{\partial f_{z1}(p)}{\partial p} \tag{10.173}$$

$$\frac{\partial \overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)}{\partial \nu} = \left(1-2\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)\right)\frac{\partial f_z(\nu)}{\partial \nu} +$$
$$+ (2f_z(\nu)-1) \frac{\left(\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-1}-1\right)^{1-f_{z1}(p)}}{\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-2}} \cdot$$
$$\cdot \sum_{i=1}^{n} \left(\frac{\left(\tilde{N}_\nu(\arg_i(a_i,w_i,\nu))^{-1}-1\right)^{-f_{z1}(p)-1}}{\tilde{N}_\nu(\arg_i(a_i,w_i,\nu))^{2}} \cdot \atop \cdot \left((2f_z(\nu)-1)\frac{\partial \arg_i(a_i,w_i,\nu)}{\partial \nu} + \atop + (2\arg_i(a_i,w_i,\nu)-1)\frac{\partial f_z(\nu)}{\partial \nu} \right) \right) \tag{10.174}$$

Yager *H*-function

$$\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu) = \widetilde{N}_\nu\left(\min\left\{1, \left(\sum_{i=1}^{n} \widetilde{N}_\nu(\arg_i(a_i,w_i,\nu))^{f_{z1}(p)}\right)^{\frac{1}{f_{z1}(p)}}\right\}\right) \tag{10.175}$$

$$\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu) = \widetilde{N}_\nu\left(\min\left\{1, \overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)\right\}\right) \tag{10.176}$$

$$\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu) = \begin{cases} \widetilde{N}_\nu\left(\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)\right) & \text{for } \overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu) \le 1 \\ \widetilde{N}_\nu(1) & \text{for } \overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu) > 1 \end{cases} \tag{10.177}$$

$$p \in (0,\infty) \tag{10.178}$$

where

$$\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)=\left(\sum_{i=1}^{n}\widetilde{N}_{\nu}\left(\arg_{i}\left(a_{i},w_{i},\nu\right)\right)^{f_{z1}(p)}\right)^{\frac{1}{f_{z1}(p)}} \tag{10.179}$$

$$\frac{\partial\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)}{\partial a_{i}}=\begin{cases}\begin{array}{c}\left(2f_{z}\left(\nu\right)-1\right)^{2}\cdot\\ \cdot\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{1-f_{z1}(p)}\cdot\\ \cdot\widetilde{N}_{\nu}\left(\arg_{i}\left(a_{i},w_{i},\nu\right)\right)^{f_{z1}(p)-1}\cdot\\ \cdot\dfrac{\partial\arg_{i}\left(a_{i},w_{i},\nu\right)}{\partial a_{i}}\\ \text{for}\quad\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)\leq 1\end{array}\\ 0\quad\text{for}\quad\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)>1\end{cases} \tag{10.180}$$

$$\frac{\partial\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)}{\partial w_{i}}=\begin{cases}\begin{array}{c}\left(2f_{z}\left(\nu\right)-1\right)^{2}\cdot\\ \cdot\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{1-f_{z1}(p)}\cdot\\ \cdot\widetilde{N}_{\nu}\left(\arg_{i}\left(a_{i},w_{i},\nu\right)\right)^{f_{z1}(p)-1}\cdot\\ \cdot\dfrac{\partial\arg_{i}\left(a_{i},w_{i},\nu\right)}{\partial w_{i}}\\ \text{for}\quad\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)\leq 1\end{array}\\ 0\quad\text{for}\quad\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)>1\end{cases} \tag{10.181}$$

$$\frac{\partial\overset{\leftrightarrow}{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)}{\partial p}=\begin{cases}\begin{array}{c}\dfrac{2f_{z}\left(\nu\right)-1}{f_{z1}\left(p\right)}\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{1-f_{z1}(p)}\cdot\\ \cdot\left(\begin{array}{c}\displaystyle\sum_{i=1}^{n}\frac{\ln\left(\widetilde{N}_{\nu}\left(\arg_{i}\left(a_{i},w_{i},\nu\right)\right)\right)}{\widetilde{N}_{\nu}\left(\arg_{i}\left(a_{i},w_{i},\nu\right)\right)^{-f_{z1}(p)}}+\\ -\dfrac{\ln\left(\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)\right)}{\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{-f_{z1}(p)}}\end{array}\right)\cdot\\ \cdot\dfrac{\partial f_{z1}\left(p\right)}{\partial p}\\ \text{for}\quad\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)\leq 1\end{array}\\ 0\quad\text{for}\quad\overset{\leftrightarrow}{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)>1\end{cases} \tag{10.182}$$

$$\frac{\partial \overleftrightarrow{H}^{*}(\mathbf{a};\mathbf{w},p,\nu)}{\partial \nu}=\begin{cases}\begin{array}{c}\left(2\overleftrightarrow{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)-1\right)\dfrac{\partial f_z(\nu)}{\partial \nu}+\\ +\left(2f_z(\nu)-1\right)\dfrac{1}{\overleftrightarrow{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)^{f_{z1}(p)-1}}\cdot\\ \cdot\left(\begin{array}{c}\tilde{N}_{\nu}\left(\arg_i(a_i,w_i,\nu)\right)^{f_{z1}(p)-1}\cdot\\ \sum\limits_{i=1}^{n}\left(\begin{array}{c}(2f_z(\nu)-1)\dfrac{\partial \arg_i(a_i,w_i,\nu)}{\partial \nu}+\\ +(2\arg_i(a_i,w_i,\nu)-1)\dfrac{\partial f_z(\nu)}{\partial \nu}\end{array}\right)\end{array}\right)\end{array} & \text{for } \overleftrightarrow{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)\le 1\\[2ex] \dfrac{\partial f_z(\nu)}{\partial \nu} & \text{for } \overleftrightarrow{h}^{*}(\mathbf{a};\mathbf{w},p,\nu)>1\end{cases} \qquad (10.183)$$

10.7 Simulation examples

We will present the results of simulation for previously described flexible neuro-fuzzy systems. Simulations concern problems of polymerization, modeling the taste of rice, classification of iris flower and classification of wine presented in Subchapter 9.2. To remind of those problems, they have been listed in Table 10.2. Two simulation series have been conducted for each simulation example. Each series has been conducted and described in analogic way:

- In the first experiment only the parameters of membership function of input and output fuzzy sets and the parameter of inference model $\nu \in [0,1]$ were learnt. The value of this parameter after completion of the learning process belongs to the set $\nu \in \{0,1\}$.
- In the second experiment the parameters of membership function of input and output fuzzy sets were also learnt, whereas the value of

TABLE 10.2. Simulation examples used

Simulation problem	Type of the problem	Number of inputs	Length of the learning sequence	Length of the testing sequence
Polymerization	approximation	3	70	–
Modeling the taste of rice	approximation	5	75	30
Classification of iris flowers	classification	4	105	45
Classification of wine	classification	13	125	53

the parameter of inference model ν was chosen as a opposite value (0 or 1) to the one obtained in the first experiment. As we will see, the accuracy obtained in this experiment is worse than the one obtained in the first experiment.

- In the third experiment the same parameters as in the first experiment and also flexibility parameters $\alpha^{\tau} \in [0,1]$, $\alpha^{I} \in [0,1]$, $\alpha^{\text{agr}} \in [0,1]$ and parameters of the shape of the applied operators $p^{\tau} \in [0,\infty)$, $p^{I} \in [0,\infty)$, $p^{\text{agr}} \in [0,\infty)$ were learnt. The latter parameters occur in case of applying adjustable Dombi and Yager H-functions (in the second series of experiments).

- In the fourth experiment the same parameters as in the third experiment, as well as weights of rules premises $w_{i,k}^{\tau} \in [0,1]$, $i = 1,\ldots,n$, $k = 1,\ldots,N$ and weights of particular rules $w_{k}^{\text{agr}} \in [0,1]$, $k = 1,\ldots,N$ were learnt. The values of weights, after completion of the learning process, are illustrated in the diagrams in which the weights of premises and the weights of rules are separated with a vertical dotted line. In the diagrams we assume that the grayer the field which symbolizes a given weight, the value of the weight is closer to zero.

In the first simulation series, in each of the four experiments described above, non-adjustable Zadeh and algebraic H-functions and H-implications were applied. In the second series of experiments adjustable H-functions and Dombi and Yager H-implications were applied, instead of non- adjustable operators.

10.7.1 Polymerization

The results of simulation for the polymerization problem are presented in Tables 10.3a and 10.3b for non-adjustable H-functions (Zadeh and algebraic) and in Tables 10.4a and 10.4b for adjustable H-functions (Dombi and Yager). Moreover, for the experiment (iv) the values of weights of rules premises $w_{i,k}^{\tau} \in [0,1]$ and values of weights of rules $w_{k}^{\text{agr}} \in [0,1]$ of the considered systems with non-adjustable H-functions are symbolically presented in Fig. 10.20, while the values of weight of systems with adjustable H-functions are presented in Fig. 10.21.

10.7.2 Modeling the taste of rice

The results of simulation for the problem of modeling the taste of rice are presented in Tables 10.5a and 10.5b for non-adjustable H-functions (Zadeh and algebraic) and in Tables 10.6a and 10.6b for adjustable H-functions (Dombi and Yager). Moreover, for the experiment (iv) the values of weights of rules premises $w_{i,k}^{\tau} \in [0,1]$ and values of weights of rules

TABLE 10.3a. The results of simulation of a flexible system with non-parameterized H-functions – the problem of polymerization

Flexible system with non-parameterized H-functions (Polymerization)		
Simulation number	Name of flexibility parameter	Initial value
i	ν	0.5
ii	ν	1
iii	ν	0.5
	α^{τ}	1
	α^{I}	1
	α^{agr}	1
iv	ν	0.5
	α^{τ}	1
	α^{I}	1
	α^{agr}	1
	$\mathbf{w}^{\tau}$	1
	$\mathbf{w}^{\mathrm{arg}}$	1

TABLE 10.3b. The results of simulation of a flexible system with non-parameterized H-functions – the problem of polymerization

Flexible system with non-parametrized H-functions (Polymerization)				
Simulation number	Final value after learning		RMSE (learning sequence)	
	Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function
i	0.0000	0.0000	0.0096	0.0060
ii	–	–	0.0115	0.0063
iii	0.0000	0.0000	0.0059	0.0056
	0.7158	0.9678		
	0.7613	0.9992		
	0.7277	0.9930		
iv	0.0000	0.0000	0.0056	0.0044
	0.6941	0.9987		
	0.7783	0.9992		
	0.6713	0.9334		
	Fig. 10.20a	Fig. 10.20b		
	Fig. 10.20a	Fig. 10.20b		

TABLE 10.4a. The results of simulation of a flexible system with parameterized H-functions – the problem of polymerization

Flexible system with parameterized H-functions (Polymerization)		
Simulation number	Name of flexibility parameter	Initial value
i	ν	0.5
ii	ν	1
iii	ν	0.5
	p^{τ}	10
	p^{I}	10
	p^{agr}	10
	α^{τ}	1
	α^{I}	1
	α^{agr}	1
iv	ν	0.5
	p^{τ}	10
	p^{I}	10
	p^{agr}	10
	α^{τ}	1
	α^{I}	1
	α^{agr}	1
	$\mathbf{w}^{\tau}$	1
	$\mathbf{w}^{\text{arg}}$	1

$w_k^{\text{agr}} \in [0,1]$ of considered systems with non-adjustable H-functions are symbolically presented in Fig. 10.22, while the values of weights of systems with adjustable H-functions are presented in Fig. 10.23.

10.7.3 *Classification of iris flower*

The results of simulation for the problem of classification of iris flower are presented in Tables 10.7a and 10.7b for non-adjustable H-functions (Zadeh and algebraic) and in Tables 10.8a and 10.8b for adjustable H-functions (Dombi and Yager). Moreover, for the experiment (iv) the values of weights of rules premises $w_{i,k}^{\tau} \in [0,1]$ and values of weights of rules $w_k^{\text{agr}} \in [0,1]$ of the considered systems with non-adjustable H-functions are symbolically presented in Fig. 10.24, while the values of weight of systems with adjustable H-functions are presented in Fig. 10.25.

TABLE 10.4b. The results of simulation of a flexible system with parameterized H-functions – the problem of polymerization

Flexible system with parameterized H-functions (Polymerization)				
Simulation number	Final value after learning		RMSE (learning sequence)	
	Dombi H-function	Yager H-function	Dombi H-function	Yager H-function
i	0.0000	0.0000	0.0117	0.0110
ii	–	–	0.0133	0.0113
iii	0.0000 9.9714 10.0042 9.9835 0.6996 0.7743 0.9941	0.0000 10.2089 10.2594 9.3991 0.1624 0.5344 0.9942	0.0077	0.0061
iv	0.0000 13.1310 15.3619 3.4720 0.7127 0.7148 0.9335 Fig. 10.21a Fig. 10.21a	0.0000 7.5714 11.7834 13.9273 0.1375 0.4742 0.9910 Fig. 10.21b Fig. 10.21b	0.0069	0.0053

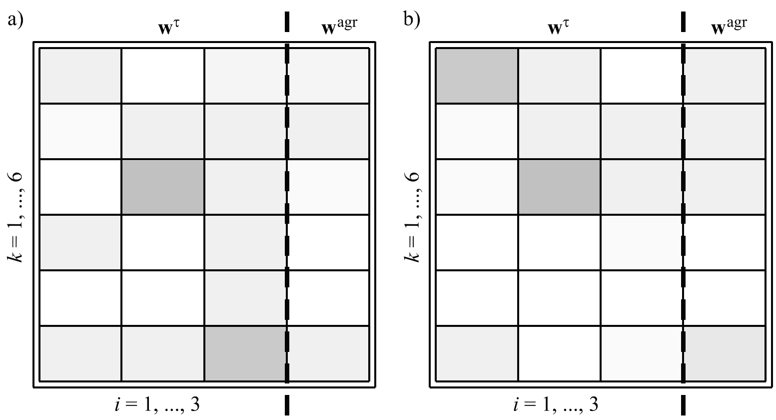

FIGURE 10.20. Weights of rules premises and weights of rules for a flexible system which solves the problem of polymerization in case of a) Zadeh H-function, b) algebraic H-function

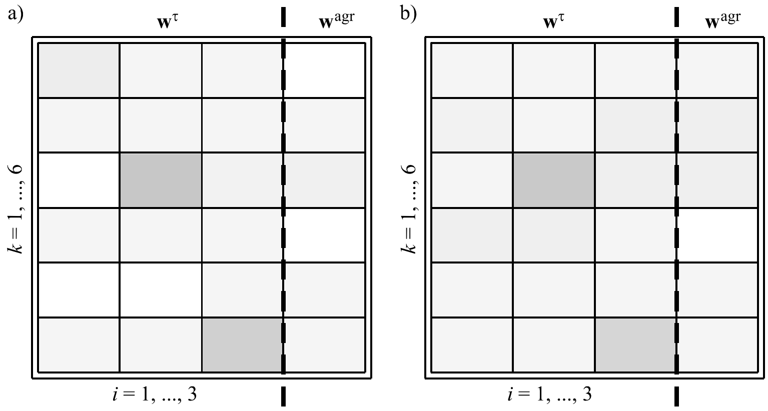

FIGURE 10.21. Weights of rules premises and weights of rules for a flexible system which solves the problem of polymerization in case of a) Dombi H-function, b) Yager H-function

TABLE 10.5a. The results of simulation of a flexible system with non-parameterized H-functions – the problem of modeling the taste of rice

Flexible system with non-parameterized H-functions (Modeling the taste of rice)		
Simulation number	Name of flexibility parameter	Initial value
i	ν	0.5
ii	ν	1
iii	ν	0.5
	α^{τ}	1
	α^{I}	1
	α^{agr}	1
iv	ν	0.5
	α^{τ}	1
	α^{I}	1
	α^{agr}	1
	$\mathbf{w}^{\tau}$	1
	$\mathbf{w}^{\mathrm{arg}}$	1

10.7.4 Classification of wine

The results of simulation for the problem of wine classification are presented in Tables 10.9a and 10.9b for non-adjustable H-functions (Zadeh and algebraic) and in Tables 10.10a and 10.10b for adjustable H-functions (Dombi

TABLE 10.5b. The results of simulation of a flexible system with non-parameterized H-functions – the problem of modeling the taste of rice

Flexible system with non-parameterized H-functions (Modeling the taste of rice)				
Simulation number	Final value after learning		RMSE (learning sequence)	
	Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function
i	0.0000	0.0000	0.0184	0.0185
ii	–	–	0.0186	0.0192
iii	0.0000 0.2954 0.9843 0.4658	0.0000 0.9972 0.9979 0.9958	0.0163	0.0173
iv	0.0000 0.3101 0.9575 0.5496 Fig. 10.22a Fig. 10.22a	0.0000 0.9519 0.9512 0.9085 Fig. 10.22b Fig. 10.22b	0.0140	0.0159

TABLE 10.6a. The results of simulation of a flexible system with parameterized H-functions – the problem of modeling the taste of rice

Flexible system with parametrized H-functions (Modeling the taste of rice)		
Simulation number	Name of flexibility parameter	Initial value
i	ν	0.5
ii	ν	1
iii	ν p^{τ} p^{I} p^{agr} α^{τ} α^{I} α^{agr}	0.5 10 10 10 1 1 1
iv	ν p^{τ} p^{I} p^{agr} α^{τ} α^{I} α^{agr} $\mathbf{w}^{\tau}$ $\mathbf{w}^{\mathrm{arg}}$	0.5 10 10 10 1 1 1 1 1

TABLE 10.6b. The results of simulation of a flexible system with parameterized H-functions – the problem of modeling the taste of rice

Flexible system with parametrized H-functions (Modeling the taste of rice)				
Simulation number	Final value after learning		RMSE (learning sequence)	
	Dombi H-function	Yager H-function	Dombi H-function	Yager H-function
i	0.0000	0.0000	0.0186	0.0187
ii	–	–	0.0192	0.0197
iii	0.0000 9.9268 10.0026 9.7692 0.4606 0.9943 0.9865	0.0000 10.7365 10.1154 10.8200 0.6895 0.9993 0.9728	0.0181	0.0184
iv	0.0000 10.1449 9.9448 9.2063 0.4380 0.9201 0.8967 Fig. 10.23a Fig. 10.23a	0.0000 10.9117 10.0472 10.0148 0.6763 0.9263 0.9927 Fig. 10.23b Fig. 10.23b	0.0160	0.0169

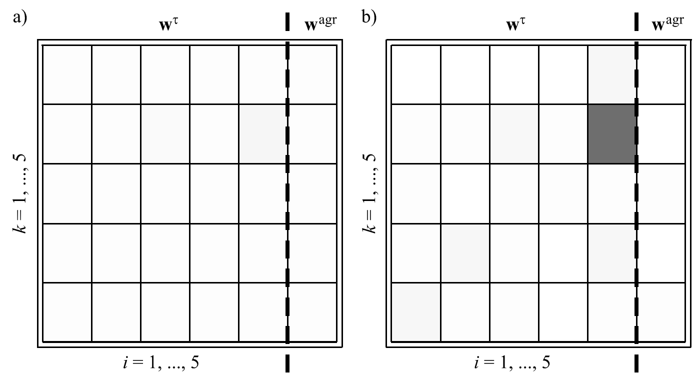

FIGURE 10.22. Weights of rules premises and weights of rules for a flexible system which solves the problem of modeling the taste of rice in case of a) Zadeh H-function, b) algebraic H-function

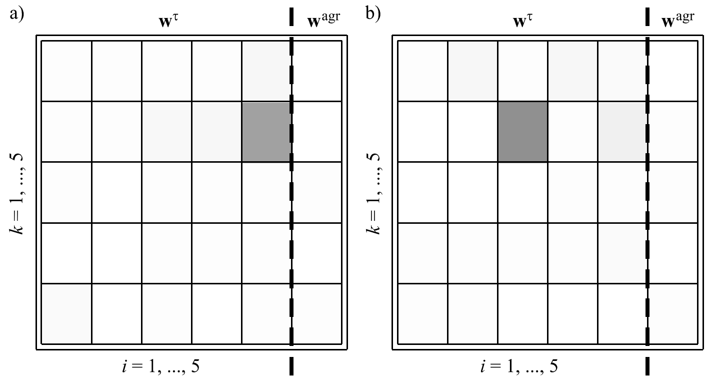

FIGURE 10.23. Weights of rules premises and weights of rules for a flexible system which solves the problem of modeling the taste of rice in case of a) Dombi H-function, b) Yager H-function

TABLE 10.7a. The results of simulation of a flexible system with non-parameterized H-functions – the problem of classification of iris flowers

Flexible system with non-parameterized H-functions (Classification of iris flowers)				
Simulation number	Name of flexibility parameter	Initial value	Final value after learning	
			Zadeh H-function	Algebraic H-function
i	ν	0.5	1.0000	1.0000
ii	ν	0	–	–
iii	ν	0.5	1.0000	1.0000
	α^{τ}	1	0.2032	0.9922
	α^{I}	1	0.9891	0.6082
	α^{agr}	1	0.9994	
iv	ν	0.5	1.0000	1.0000
	α^{τ}	1	0.2442	0.9592
	α^{I}	1	0.9845	0.5753
	α^{agr}	1	0.9650	0.9937
	$\mathbf{w}^{\tau}$	1	Fig. 10.24a	Fig. 10.24b
	$\mathbf{w}^{arg}$	1	Fig. 10.24a	Fig. 10.24b

and Yager). Moreover, for the experiment (iv) the values of weights of rules premises $w_{i,k}^{\tau} \in [0,1]$ and values of weights of rules $w_{k}^{agr} \in [0,1]$ of considered systems with non-adjustable H-functions are symbolically presented in Fig. 10.26, while the values of weights of systems with adjustable H-functions are presented in Fig. 10.27.

TABLE 10.7b. The results of simulation of a flexible system with non-parameterized H-functions – the problem of classification of iris flowers

Flexible system with non-parameterized H-functions (Classification of iris flowers)				
Simulation number	Number of errors [%] (learning sequence)		Number of errors [%] (testing sequence)	
	Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function
i	0.95	0.95	4.44	4.44
ii	0.95	0.95	6.67	6.67
iii	0.00	0.95	4.44	4.44
iv	0.00	0.00	4.44	4.44

TABLE 10.8a. The results of simulation of a flexible system with parameterized H-functions – the problem of classification of iris flowers

Flexible system with parameterized H-functions (Classification of iris flowers)				
Simulation number	Name of flexibility parameter	Initial value	Final value after learning	
			Dombi H-function	Yager H-function
i	ν	0.5	1.0000	1.0000
ii	ν	0	–	–
iii	ν	0.5	1.0000	1.0000
	p^{τ}	10	13.2031	4.3306
	p^{I}	10	10.0001	7.5741
	p^{agr}	10	9.9974	10.1209
	α^{τ}	1	0.8259	0.7846
	α^{I}	1	0.9924	0.9931
	α^{agr}	1	0.9985	0.9985
iv	ν	0.5	1.0000	1.0000
	p^{τ}	10	13.5253	4.3621
	p^{I}	10	10.8610	8.0120
	p^{agr}	10	9.4218	9.3590
	α^{τ}	1	0.8739	0.8068
	α^{I}	1	0.9871	0.9731
	α^{agr}	1	0.9698	0.9661
	$\mathbf{w}^{\tau}$	1	Fig. 10.25a	Fig. 10.25b
	$\mathbf{w}^{\text{arg}}$	1	Fig. 10.25a	Fig. 10.25b

TABLE 10.8b. The results of simulation of a flexible system with parameterized H-functions – the problem of classification of iris flowers

Flexible system with parameterized H-functions (Classification of iris flowers)				
Simulation number	Number of errors [%] (learning sequence)		Number of errors [%] (testing sequence)	
	Dombi H-function	Yager H-function	Dombi H-function	Yager H-function
i	0.00	0.95	4.44	4.44
ii	0.95	0.95	6.67	6.67
iii	0.00	0.00	4.44	4.44
iv	0.00	0.00	2.22	2.22

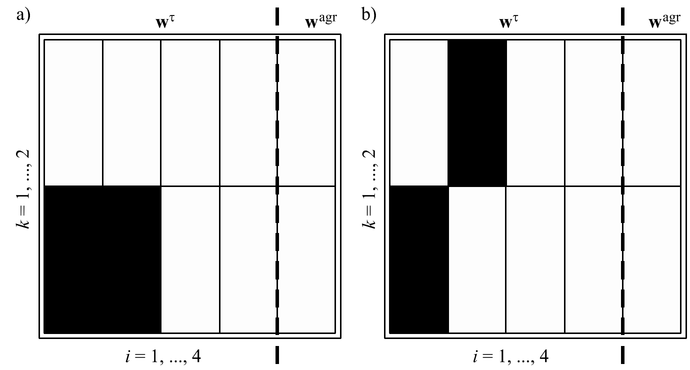

FIGURE 10.24. Weights of rules premises and weights of rules for a flexible system which solves the problem of classification of iris flowers in case of a) Zadeh H-function, b) algebraic H-function

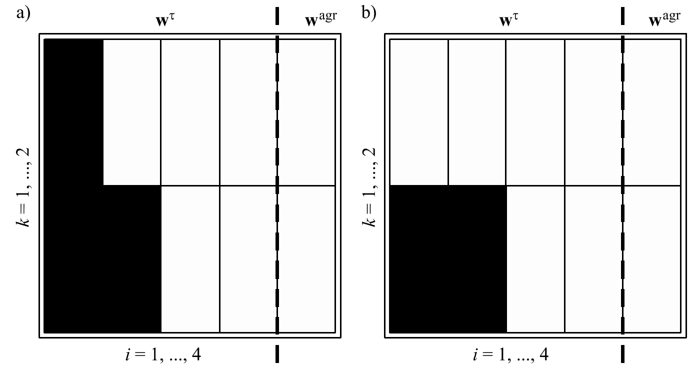

FIGURE 10.25. Weights of rules premises and weights of rules for a flexible system which solves the problem of classification of iris flowers in case of a) Dombi H-function, b) Yager H-function

TABLE 10.9a. The results of simulation of a flexible system with non-parameterized H-functions – the problem of wine classification

Flexible system with non-parameterized H-functions (Classification of wine)				
Simulation number	Name of flexibility parameter	Initial value	Final value after learning	
			Zadeh H-function	Algebraic H-function
i	ν	0.5	1.0000	1.0000
ii	ν	0	–	–
iii	ν	0.5	1.0000	1.0000
	α^{τ}	1	0.0004	0.0036
	α^{I}	1	0.9907	0.9986
	α^{agr}	1	0.9938	0.9908
iv	ν	0.5	1.0000	1.0000
	α^{τ}	1	0.0329	0.0180
	α^{I}	1	0.9987	0.9756
	α^{agr}	1	0.9896	0.9861
	$\mathbf{w}^{\tau}$	1	Fig. 10.26a	Fig. 10.26b
	$\mathbf{w}^{\text{arg}}$	1	Fig. 10.26a	Fig. 10.26b

TABLE 10.9b. The results of simulation of a flexible system with non-parameterized H-functions – the problem of wine classification

Flexible system with non-parameterized H-functions (Classification of wine)				
Simulation number	Number of errors [%] (learning sequence)		Number of errors [%] (testing sequence)	
	Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function
i	0.00	0.00	3.77	1.89
ii	0.80	0.80	3.77	3.77
iii	0.00	0.00	1.89	1.89
iv	0.00	0.00	0.00	0.00

TABLE 10.10a. The results of simulation of a flexible system with parameterized H-functions – the problem of wine classification

Flexible system with parameterized H-functions (Classification of wine)				
Simulation number	Name of flexibility parameter	Initial value	Final value after learning	
			Dombi H-function	Yager H-function
i	ν	0.5	1.0000	1.0000
ii	ν	0	–	–
iii	ν	0.5	1.0000	1.0000
	p^{τ}	10	9.9999	10.0498
	p^{I}	10	10.0005	9.9936
	p^{agr}	10	9.9991	10.0014
	α^{τ}	1	0.0032	0.0029
	α^{I}	1	0.9911	0.9917
	α^{agr}	1	0.9919	0.9920
iv	ν	0.5	1.0000	1.0000
	p^{τ}	10	7.8330	6.9528
	p^{I}	10	11.7084	13.3122
	p^{agr}	10	14.3699	12.1427
	α^{τ}	1	0.0028	0.0389
	α^{I}	1	0.9826	0.9740
	α^{agr}	1	0.9914	0.9599
	$\mathbf{w}^{\tau}$	1	Fig. 10.27a	Fig. 10.27b
	$\mathbf{w}^{\text{arg}}$	1	Fig. 10.27a	Fig. 10.27b

TABLE 10.10b. The results of simulation of a flexible system with parameterized H-functions – the problem of wine classification

Flexible system with parameterized H-functions (Classification of wine)				
Simulation number	Number of errors [%] (learning sequence)		Number of errors [%] (testing sequence)	
	Dombi H-function	Yager H-function	Dombi H-function	Yager H-function
i	0.00	0.00	1.89	1.89
ii	0.00	0.00	3.77	3.77
iii	0.00	0.00	1.89	1.89
iv	0.00	0.00	0.00	0.00

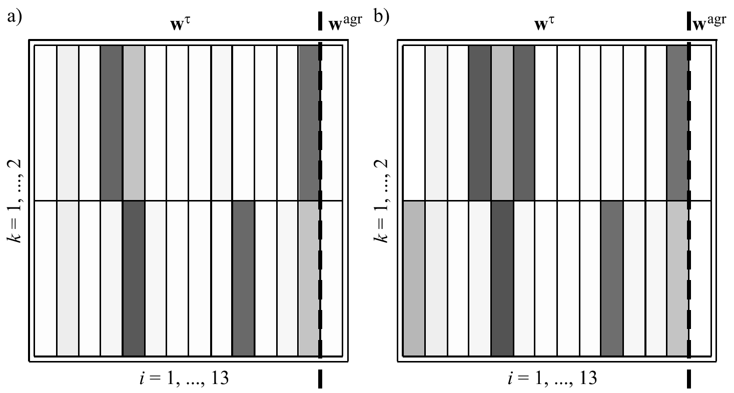

FIGURE 10.26. Weights of rules premises and weights of rules for a flexible system which solves the problem of distinguishing the brand of wine in case of a) Zadeh H-function, b) algebraic H-function

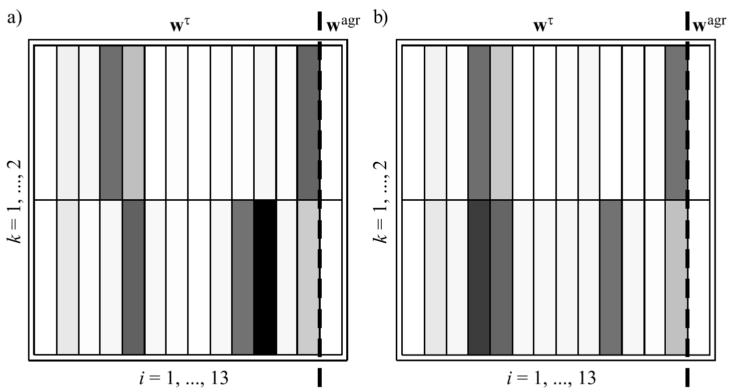

FIGURE 10.27. Weights of rules premises and weights of rules for a flexible system which solves the problem of distinguishing the brand of wine in case of a) Dombi H-function, b) Yager H-function

10.8 Notes

The concept of flexible neuro-fuzzy systems presented in this chapter allows us to determine the type of system (Mamdani or logical) as a result of the learning process. It could be inferred from the simulation examples presented in Subchapter 10.7 that a flexible system becomes a Mamdani system after completion of the learning process (parameter $\nu = 0$) for the problems of approximation or identification. In contrast, for the problems of classification a flexible system becomes a logical system (parameter $\nu = 1$) as a result of learning. The above results could be treated as a recommendation of the Mamdani system to solve the problems of approximation or identification and the logical system to solve the problems of classification. It should be mentioned that the concept of soft triangular norms was presented by Yager and Filev [262], while Klement [111] and Lowen [128] presented in detail various types of parameterized triangular norms. Various types of flexible neuro-fuzzy structures were proposed by Cpałka [30]. The subject of those systems is discussed in more detail in monograph [225]. We refer the interested Reader to the following works [210–212, 215, 217, 218, 220, 223, 227].