9
Neuro-fuzzy systems of Mamdani, logical and Takagi-Sugeno type

9.1 Introduction

Within the last dozen of years, different structures of neuro-fuzzy networks have been presented, often referred to in the world literature as neuro-fuzzy systems. They combine the advantages of neural networks and classic fuzzy systems. In particular, the neuro-fuzzy networks are characterized – in contrast with neural networks – by a interpretable representation of knowledge represented by fuzzy rules. As generally known, the knowledge in neural networks is represented by the values of synaptic weights, and therefore is completely not interpretable, for instance, for a user of a medical expert system that uses neural networks. Moreover, neuro-fuzzy networks can be trained, using the idea of error backpropagation method, which is the basis of learning of multilayer neural networks. The learning usually applies to membership function parameters of the **IF** and **THEN** part of the fuzzy rules. As shown in Chapter 7, there is also the possibility to apply the evolutionary algorithms to learn not only the parameters of the membership functions but also the fuzzy rules themselves. The above discussed advantages of the neuro-fuzzy networks are the reason for their common application in classification, approximation and prediction problems. Most of neuro-fuzzy structures described in the world literature utilizes the Mamdani type inference or the Takagi-Sugeno schema. As mentioned in Chapter 4, the Mamdani type inference consists in connecting the antecedents and the consequents of rules using a t-norm (most often the t-norm of the min type or of the product type). Then the aggregation of particular rules is made using a t-conorm. In case of the Takagi-Sugeno schema, the consequents of the rules are not fuzzy in nature, but are functions of the input variables. Less often the logical inference is applied, which consists in connecting the antecedents and the consequents of rules using a fuzzy implication that satisfies the conditions of Definition 4.47. In case of an inference of logical type the aggregation of particular rules is made using a t-conorm. It is obvious that the designers and users of neuro-fuzzy systems would like to obtain a possibly high accuracy of these systems operation in the sense of the chosen quality criterion. In approximation and prediction problems, such quality criterion is the mean squared error, and in classification problems – the number of erroneously classified samples. In both problems, the experiments are made on learning sequences and testing sequences. It should be stressed that the satisfactory results obtained on a learning sequence do not guarantee a correct system operation on a testing sequence. In other words, the neuro-fuzzy system should have good properties of the so-called generalization. In particular, neuro-fuzzy systems designed using both the membership function and the weights describing the importance of rules and importance of linguistic variables in individual rules should be characterized by an appropriate number of all parameters which are to be subject of learning. A big number of parameters ensures a small learning error, but usually leads to wrong generalization. On the other hand, a small number of parameters in the system leads to a larger learning error. In this chapter, we will present the Mamdani, logical and Takagi-Sugeno systems, their learning algorithms and we will make a comparative analysis of their effectiveness. We will solve the issue of designing neuro-fuzzy systems, which are a compromise between accuracy and the number of parameters describing this system.

9.2 Description of simulation problems used

Neuro-fuzzy system discussed in this and the next chapter will be tested using standard testing problems (*benchmarks*).

Table 9.1 presents the name of the problem, number of input data, length of the learning sequence and length of the testing sequence.

Below, we present a detailed description of the problems listed in Table 9.1. Information on the number of rules and the number of epochs relates to the simulations performed in this chapter (problems 9.2.1 - 9.2.4).

9.2.1 Polymerization

We consider the problem of modeling the polymer manufacturing process. The device produces polymers (macromolecular compounds obtained from monomers, i.e. small-molecule compounds) as a result of chemical reaction

TABLE 9.1. Simulation problems

No.	Name of problem	Number of inputs	Length of the learning sequence	Length of the testing sequence
1	Polymerization	3	70	20
2	HANG (modeling a static nonlinear function)	2	50	20
3	NPD (modeling a dynamic nonlinear function)	2	1000	200
4	Modeling the taste of rice	5	75	30
5	Distinguishing of the brand of wine	13	125	53
6	Classification of iris flower	4	90	60

called polymerization, during which many small molecules of the same compound connect spontaneously (or under the influence of catalytic agents). In order to model the system, three continuous input variables are selected. They include: monomer concentration, change of monomer concentration and its current flow rate. Based on the values of input variables, the next value of the monomer flow rate should be determined. Simulation tests of systems made of 3 inputs, one output and 6 rules have been performed. The experiment was repeated many times for 6000 epochs (420 000 iterations) and its results were averaged.

9.2.2 Modeling a static non-linear function

It is an issue of approximation of a non-linear function – HANG, described by the formula

$$y(x_1, x_2) = \left(1 + x_1^{-2} + x_2^{-1.5}\right)^2, \tag{9.1}$$

where $x_1, x_2 \in [1, 5]$. The learning sequence consists of 50 input data vectors and the corresponding function values. Simulation tests of systems made of 2 inputs, one output and 8 rules were performed. The experiment was repeated many times for 8000 epochs (400 000 iterations) and its results were averaged.

9.2.3 Modeling a non-linear dynamic object (Nonlinear Dynamic Problem - NDP)

It is the problem of modeling a *nonlinear dynamic object* the behavior of which is described by the formula

$$y(t) = g(y(t-1), y(t-2)) + u(t), \tag{9.2}$$

where

$$g\left(u\left(t-1\right),y\left(t-2\right)\right)=\frac{y\left(t-1\right)y\left(t-2\right)\left(y\left(t-1\right)-0.5\right)}{1+y^{2}\left(t-1\right)+y^{2}\left(t-2\right)}, \tag{9.3}$$

and $u(t)$ is the output signal.

For the purpose of learning neuro-fuzzy systems, a sequence of model states of the objects for a random input signal with uniform distribution is used (first 500 samples) and for a sinusoidal input signal $u(t) = \sin(2\pi t/25)$ (next 500 samples). The sequence has been generated for a zero initial state. Simulation tests of systems made of 3 inputs, one output and 6 rules were performed. The experiment was repeated many times for 500 epochs (500 000 iterations) and its results were averaged.

9.2.4 Modeling the taste of rice

The problem to be solved in this example is to find a nonlinear dependency between input data, characterizing the rice samples, and the output signal containing the interpretation of the taste of rice. Data consist of 105 cases. Each sample has been described by 5 features: flavor, appearance, taste, viscosity and hardness, constituting the system input data. The system output is a general assessment of the taste of rice. Input and output data have been normalized to the interval $[0, 1]$. Simulation tests of systems made of 2 inputs, one output and 6 rules were performed. The experiment was repeated many times for 5000 epochs (375 000 iterations) and its results were averaged.

9.2.5 Distinguishing of the brand of wine

The problem to be solved is the correct classification of wine samples. Data in the problem of wine distinguishing consist of chemical analysis of 178 wines from same region of Italy, but from three different vineyards. The input data consist of 13 continuous attributes which include among other thing: alcohol contents, malic acid contents, sediment, sediment alkalinity, magnesium contents, total phenol contents, color intensity and shade. In the experiment discussed, all the data have been divided into a learning sequence (125 samples) and a testing sequence (53 samples).

9.2.6 Classification of iris flower

The problem consist in the classification of the Iris flower based on the length of the leaf in cm, width of the leaf in cm, length of the petal in cm, width of the petal in cm. We distinguish three classes: *Iris setosa*, *Iris Versicolor* and *Iris Virginica*. Data include 150 sets, which were divided at random into the learning sequence (90 sets) and the testing sequence (60 sets).

Remark 9.1
Gradient algorithms of the momentum type with learning coefficient $\eta = 0.25$ and with momentum coefficient 0.1 have been used for learning of all the neuro-fuzzy systems presented in this chapter. These algorithms have been derived in Subchapter 9.6 without taking account of the momentum term in particular iteration procedures. In all neuro-fuzzy systems considered in this chapter, the following principle has been adopted:

- particular rules are aggregated using a t-conorm of the max type in case of the Mamdani system and a t-norm of the min type in case of a logical system,
- the antecedents of rules are aggregated by means of t-norm of the product type.

The basis for assessment of neuro-fuzzy systems will be the value of error (mean squared error in case of approximation issues or number of erroneously classified samples in case of classification issues). At first, the mean error in particular epochs is determined, and then the minimum error is found among these errors.

9.3 Neuro-fuzzy systems of Mamdani type

Let us consider two types of neuro-fuzzy systems of Mamdani type, the so-called A type and B type systems. In both cases, the antecedents and the consequents of rules are connected by means of a t-norm. In A type systems at the inference block output we have N fuzzy sets, while in B-type systems at the block output we have one fuzzy set which is the result of aggregation of inference results in particular rules.

9.3.1 A-type systems

In A-type systems, the defuzzification is realized using the dependency:

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \mu_{\overline{B}^r}(\overline{y}^r)}{\sum_{r=1}^{N} \mu_{\overline{B}^r}(\overline{y}^r)}. \tag{9.4}$$

The membership functions of fuzzy sets $\overline{B}^r$, $r = 1, 2, \ldots, N$, are defined using the following formula:

$$\mu_{\overline{B}^r}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \left\{ \mu_{A^r}(\mathbf{x}) \overset{T}{*} \mu_{A^r \to B^r}(\mathbf{x}, y) \right\}. \tag{9.5}$$

With singleton type fuzzification, formula (9.5) takes the form

$$\mu_{\overline{B}^r}(y) = \mu_{A^r \to B^r}(\overline{\mathbf{x}}, y) = T\left(\mu_{A^r}(\overline{\mathbf{x}}), \mu_{B^r}(y)\right). \tag{9.6}$$

Since

$$\mu_{A^r}(\overline{\mathbf{x}}) = \overset{n}{\underset{i=1}{T}} \left(\mu_{A_i^r}(\overline{x}_i)\right), \tag{9.7}$$

we have

$$\mu_{\overline{B}^r}(y) = \mu_{A^r \to B^r}(\overline{\mathbf{x}}, y) = T\left[\overset{n}{\underset{i=1}{T}} \left(\mu_{A_i^r}(\overline{x}_i)\right), \mu_{B^r}(y)\right], \tag{9.8}$$

where T is any t-norm. Owing to the fact that

$$\mu_{B^r}(\overline{y}^r) = 1 \tag{9.9}$$

and

$$T(a, 1) = a, \tag{9.10}$$

we obtain the following dependency:

$$\mu_{\overline{B}^r}(\overline{y}^r) = \overset{n}{\underset{i=1}{T}} \left(\mu_{A_i^r}(\overline{x}_i)\right). \tag{9.11}$$

By substituting dependency (9.11) to formula (9.4), we get

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot T_{i=1}^{n} \left(\mu_{A_i^r}(\overline{x}_i)\right)}{\sum_{r=1}^{N} T_{i=1}^{n} \left(\mu_{A_i^r}(\overline{x}_i)\right)}. \tag{9.12}$$

In A-type systems, separate inference is made within each rule and $\mu_{\overline{B}^r}(\overline{y}^r)$, $r = 1, 2, \ldots, N$, is computed. Let us assume that input and output linguistic variables are described by means of Gaussian membership functions, that is

$$\mu_{A_i^r}(x_i) = \exp\left[-\left(\frac{x_i - \overline{x}_i^r}{\sigma_i^r}\right)^2\right], \tag{9.13}$$

$$\mu_{B^r}(y) = \exp\left[-\left(\frac{y - \overline{y}^r}{\sigma^r}\right)^2\right]. \tag{9.14}$$

By substituting the above dependencies to formula (9.4) and applying the Larsen rule, we will get the following formula:

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \left(\prod_{i=1}^{n} \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right)}{\sum_{r=1}^{N} \left(\prod_{i=1}^{n} \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right)}. \tag{9.15}$$

Let us notice that in dependency (9.15), there is no parameter σ^r of the output fuzzy set B^r, $r = 1, 2, \ldots, N$. Figure 9.1 presents a block schema of the structure reflecting dependency (9.15). As we can see, it is a multilayer network structure. Such a structure is called a neuro-fuzzy network. To train it, the idea of error backpropagation method may be applied.

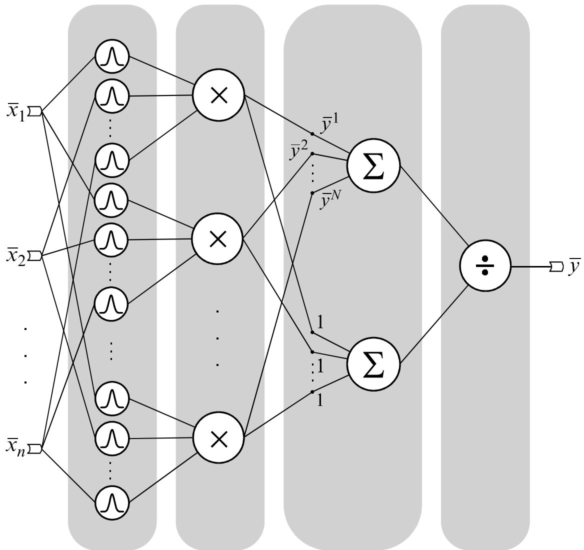

FIGURE 9.1. Network structure of a system described by formula (9.15)

9.3.2 B-type systems

In B-type systems, the defuzzification is made using the dependency

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \mu_{B'}\left(\overline{y}^r\right)}{\sum_{r=1}^{N} \cdot \mu_{B'}\left(\overline{y}^r\right)}. \tag{9.16}$$

In these systems, aggregation of particular fuzzy sets $\overline{B}^k$ given by formula (9.6) is made, which means that the fuzzy set B' is determined through operation of union of fuzzy sets $\overline{B}^k$

$$B' = \bigcup_{k=1}^{N} \overline{B}^k. \tag{9.17}$$

The membership function of fuzzy set B' is determined using a t-conorm, i.e.

$$\mu_{B'}\left(y\right) = \mathop{S}_{k=1}^{N} \left\{\mu_{\overline{B}^k}\left(y\right)\right\}. \tag{9.18}$$

Therefore

$$\begin{aligned} \mu_{B'}\left(\overline{y}^r\right) &= \mathop{S}_{k=1}^{N} \left\{\mu_{\overline{B}^k}\left(\overline{y}^r\right)\right\} = \mathop{S}_{k=1}^{N} \left\{T\left(\mu_{A^k}\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right)\right\} \\ &= \mathop{S}_{k=1}^{N} \left\{T\left(\mathop{T}_{i=1}^{n} \mu_{A_i^k}\left(\overline{x}_i\right), \mu_{B^k}\left(\overline{y}^r\right)\right)\right\}. \end{aligned} \tag{9.19}$$

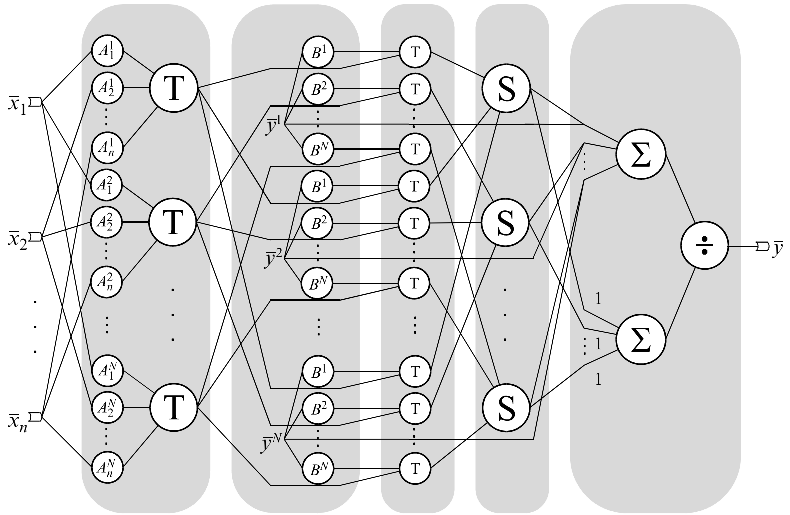

FIGURE 9.2. Network structure of a system described by formula (9.20)

By substituting formula (9.19) to dependency (9.16), we get

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{N} \left\{ T\left(T_{i=1}^{n} \left\{ \mu_{A_i^k}\left(\overline{x}_i\right) \right\}, \mu_{B^k}\left(\overline{y}^r\right) \right) \right\}}{\sum_{r=1}^{N} S_{k=1}^{N} \left\{ T\left(T_{i=1}^{n} \left\{ \mu_{A_i^k}\left(\overline{x}_i\right) \right\}, \mu_{B^k}\left(\overline{y}^r\right) \right) \right\}}. \tag{9.20}$$

In Fig. 9.2 the network structure of the system described by formula (9.20) is presented.

In B-type systems, separate inference is also made within each rule, but next, the aggregation of inference results is made in individual rules and only then $\mu_{B'}(\overline{y}^r)$, $r = 1, 2, \ldots, N$, is computed.

9.3.3 Mamdani type systems in modeling problems

Mamdani type systems will be applied to modeling problems. These problems were described in detail in Subchapter 9.2. We will assume that fuzzy sets A_i^r and B^r are characterized by Gaussian membership functions given by formula (9.13) and (9.14).

9.3.3.1. M1-type systems

Let us consider Mamdani type systems which are constructed using definitions of triangular norms without taking the weights into account. Using dependency (9.20) and min type Mamdani rule, we obtain the following description of the neuro-fuzzy system:

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{N} \left\{ \min \left(T_{i=1}^{n} \left\{ \mu_{A_i^k} (\overline{x}_i) \right\}, \mu_{B^k} (\overline{y}^r) \right) \right\}}{\sum_{r=1}^{N} S_{k=1}^{N} \left\{ \min \left(T_{i=1}^{n} \left\{ \mu_{A_i^k} (\overline{x}_i) \right\}, \mu_{B^k} (\overline{y}^r) \right) \right\}}. \tag{9.21}$$

Substituting dependencies (9.13) and (9.14) to formula (9.21) and using the contents of Remark 9.1, we obtain

$$\begin{aligned} \overline{y} &= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{N} \left\{ \min \left(T_{i=1}^{n} \left\{ \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\}, \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right) \right\}}{\sum_{r=1}^{N} S_{k=1}^{N} \left\{ \min \left(T_{i=1}^{n} \left\{ \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\}, \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right) \right\}} \\ &= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \max\limits_{1 \le k \le N} \left\{ \min \left(\prod_{i=1}^{n} \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right) \cdot \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right\}}{\sum_{r=1}^{N} \max\limits_{1 \le k \le N} \left\{ \min \left(\prod_{i=1}^{n} \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \cdot \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right) \right\}}. \end{aligned} \tag{9.22}$$

Using dependency (9.20) and product type Mamdani rule (known as Larsen rule), we obtain the following description of the neuro-fuzzy system:

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{N} \left\{ T_{i=1}^{n} \left\{ \mu_{A_i^k} (\overline{x}_i) \right\} \cdot \mu_{B^k} (\overline{y}^r) \right\}}{\sum_{r=1}^{N} S_{k=1}^{N} \left\{ T_{i=1}^{n} \left\{ \mu_{A_i^k} (\overline{x}_i) \right\} \cdot \mu_{B^k} (\overline{y}^r) \right\}}. \tag{9.23}$$

Substituting dependencies (9.13) and (9.14) to formula (9.23) and using the contents of Remark 9.1, we obtain

$$\begin{aligned} \overline{y} &= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{N} \left\{ T_{i=1}^{n} \left\{ \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\} \cdot \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right\}}{\sum_{r=1}^{N} S_{k=1}^{N} \left\{ T_{i=1}^{n} \left\{ \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\} \cdot \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right\}} \\ &= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \max\limits_{1 \le k \le N} \left\{ \prod_{i=1}^{n} \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \cdot \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right\}}{\sum_{r=1}^{N} \max\limits_{1 \le k \le N} \left\{ \prod_{i=1}^{n} \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \cdot \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right\}}. \end{aligned} \tag{9.24}$$

Further in this chapter, we will not remind the wording of Remark 9.1. We should however remember that particular rules are aggregated using a t-conorm of the max type in case of the Mamdani system and a t-norm of the min type in case of a logic system and that the antecedents of the rules are aggregated using a t-norm of the product type. Neuro-fuzzy systems (9.22) and (9.24) are special cases of B-type system described in point 9.3.2. In systems (9.22) and (9.24) the following parameters of the membership functions are subject to learning; $\overline{x}_i^k, \sigma_i^k, \overline{y}^k, \sigma^k,\ k = 1, 2, \ldots, N$. One of subjects of studies is also A-type Mamdani system described in point 9.3.1, the description of which, for the Reader's convenience, is recalled below:

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \left[\prod_{i=1}^{n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right)\right]}{\sum_{r=1}^{N} \left[\prod_{i=1}^{n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right)\right]}. \tag{9.25}$$

System (9.25) has been called a simplified Larsen structure. In this system, the parameters $\overline{x}_i^r$, σ_i^r, $\overline{y}^r$, $r = 1, 2, \ldots, N$, are subject to learning. It may be shown that system (9.25) is a special case of system (9.24) Neuro-fuzzy systems (9.22), (9.24) and (9.25) have been used to solve four problems specified in Table 9.1: polymerization, HANG, NDP and modeling the taste of rice. All the parameters of the neuro-fuzzy systems have been trained using error backpropagation method: centers and widths of Gaussian functions were trained. In case of structure (9.25), there are no widths of consequents of the Gaussian function.

9.3.3.1.1. Polymerization

Table 9.2 presents the smallest error for individual structures and the number of epochs corresponding to this error. Table 9.3 presents three desired error values and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.3, for the Larsen structure, it was impossible to train the system with error 0.0045.

9.3.3.1.2. HANG

Table 9.4 presents the smallest error for individual structures and the number of epochs corresponding to this error. Table 9.5 presents three

TABLE 9.2. The smallest error obtained as a result of learning

POLYMERIZATION		
Structure	The smallest error	Number of epochs
Mamdani	0.0041	3734
Larsen	0.0049	5984
Larsen (simplified)	0.0042	4689

TABLE 9.3. Number of epochs required to train the system which is characterized by a definite error

POLYMERIZATION			
Structure	Value of error		
	0.0055	0.0050	0.0045
Mamdani	1086	1479	1943
Larsen	3621	5984	–
Larsen (simplified)	807	2718	4454

TABLE 9.4. The smallest error obtained as a result of learning

HANG		
Structure	The smallest error	Number of epochs
Mamdani	0.0340	7848
Larsen	0.0387	8000
Larsen (simplified)	0.0240	7102

TABLE 9.5. Number of epochs required to train the system which is characterized by a given error

HANG			
POLYMERIZATION			
Structure	Value of error		
	0.028	0.026	0.024
Mamdani	–	–	–
Larsen	–	–	–
Larsen (simplified)	4071	6024	7102

desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.5, neither the Mamdani nor the Larsen structure did achieve any of the desired values of error.

9.3.3.1.3. NDP

Table 9.6 presents the smallest error for individual structures and the number of epochs corresponding to this error.

Table 9.7 presents three desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.7, the Mamdani structure did not attain any of the desired values of error.

TABLE 9.6. The smallest error obtained as a result of learning

NDP		
Structure	The smallest error	Number of epochs
Mamdani	0.0263	436
Larsen	0.0176	433
Larsen (simplified)	0.0140	393

TABLE 9.7. Number of epochs required to train the system which is characterized by a given error

NDP			
Structure	Value of error		
	0.026	0.023	0.020
Mamdani	–	–	–
Larsen	172	233	302
Larsen (simplified)	74	82	93

9.3.3.1.4. Modeling the taste of rice

Table 9.8 presents the smallest error for individual structures and the number of epochs corresponding to this error.

As it may be inferred from Table 9.9, only the simplified Larsen structure obtained all the desired values of error.

9.3.3.2. M2-type systems

Let us consider Mamdani type systems which are constructed using definitions of triangular norms taking into account the weights w_k, characterizing the importance of particular rules. Using the definition of weighted t-conorm and dependency (9.22), (9.24) and (9.25) we obtain the following description of the neuro-fuzzy systems:

TABLE 9.8. The smallest error obtained as a result of learning

MODELING THE TASTE OF RICE		
Structure	The smallest error	Number of epochs
Mamdani	0.0244	4459
Larsen	0.0252	2501
Larsen (simplified)	0.0205	3888

TABLE 9.9. Number of epochs required to train the system which is characterized by a definite error

MODELING THE TASTE OF RICE			
Structure	Value of error		
	0.028	0.025	0.022
Mamdani	233	1978	–
Larsen	506	–	–
Larsen (simplified)	67	451	2936

a) Mamdani system with weights of rules

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{*N} \left\{ \min \left(T_{i=1}^{n} \left\{ \mu_{A_i^k} \left(\overline{x}_i \right) \right\}, \mu_{B^k} \left(\overline{y}^r \right) \right), w_k \right\}}{\sum_{r=1}^{N} S_{k=1}^{*N} \left\{ \min \left(T_{i=1}^{n} \left\{ \mu_{A_i^k} \left(\overline{x}_i \right) \right\}, \mu_{B^k} \left(\overline{y}^r \right) \right), w_k \right\}} \tag{9.26}$$

$$= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{*N} \left\{ \min \left(T_{i=1}^{n} \left\{ \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\}, \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right), w_k \right\}}{\sum_{r=1}^{N} S_{k=1}^{*N} \left\{ \min \left(T_{i=1}^{n} \left\{ \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\}, \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right), w_k \right\}}.$$

b) Larsen system with weights of rules

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{*N} \left\{ T_{i=1}^{n} \left(\left\{ \mu_{A_i^k} \left(\overline{x}_i \right) \right\} \cdot \mu_{B^k} \left(\overline{y}^r \right) \right), w_k \right\}}{\sum_{r=1}^{N} S_{k=1}^{*N} \left\{ T_{i=1}^{n} \left(\left\{ \mu_{A_i^k} \left(\overline{x}_i \right) \right\} \cdot \mu_{B^k} \left(\overline{y}^r \right) \right), w_k \right\}} \tag{9.27}$$

$$= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{*N} \left\{ T_{i=1}^{n} \left\{ \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\} \cdot \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right], w_k \right\}}{\sum_{r=1}^{N} S_{k=1}^{*N} \left\{ T_{i=1}^{n} \left\{ \exp \left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\} \cdot \exp \left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right], w_k \right\}}.$$

In both a) and b) systems, the parameters of membership function, i.e. $\overline{x}_i^k, \sigma_i^k, \overline{y}^k, \sigma^k$ and weights w_k are subject to learning.

c) Simplified Larsen system with weights of rules

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot w_r \cdot \prod_{i=1}^{n} \left(\mu_{A_i^r} (\overline{x}_i) \right)}{\sum_{r=1}^{N} w_r \cdot \prod_{i=1}^{n} \left(\mu_{A_i^r} (\overline{x}_i) \right)} \tag{9.28}$$

$$= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot w_r \cdot \prod_{i=1}^{n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right)}{\sum_{r=1}^{N} w_r \cdot \prod_{i=1}^{n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right)}.$$

In c) system, the parameters of membership function $\overline{x}_i^r, \sigma_i^r, \overline{y}^r$ and weights w_r. are subject to learning. Neuro-fuzzy systems (9.26), (9.27) and (9.28), have been used to solve four problems specified in Table 9.1.

9.3.3.2.1. Polymerization

Table 9.10 presents the smallest error for individual structures and the number of epochs corresponding to this error.

Table 9.11 presents three desired values of error and the number of epochs, after which this error was obtained.

9.3.3.2.2. HANG

Table 9.12 presents the smallest error for individual structures and the number of epochs corresponding to this error. Table 9.13 presents three

TABLE 9.10. The smallest error obtained as a result of learning

POLYMERIZATION		
Structure	The smallest error	Number of epochs
Mamdani with weights	0.0039	4088
Larsen with weights	0.0043	4501
Larsen (simplified) weights	0.0039	3691

TABLE 9.11. Number of epochs required to train the system which is characterized by a definite error

POLYMERIZATION			
Structure	Value of error		
	0.0055	0.0050	0.0045
Mamdani with weights	26	44	2440
Larsen with weights	2646	3154	4099
Larsen (simplified) with weights	1633	1633	3443

TABLE 9.12. The smallest error obtained as a result of learning

HANG		
Structure	The smallest error	Number of epochs
Mamdani with weights	0.0318	7848
Larsen with weights	0.0353	6773
Larsen (simplified) with weights	0.0183	1955

TABLE 9.13. Number of epochs required to train the system which is characterized by a definite error

HANG			
Structure	Value of error		
	0.028	0.026	0.024
Mamdani with weights	–	–	–
Larsen with weights	–	–	–
Larsen (simplified) with weights	191	366	632

TABLE 9.14. The smallest error obtained as a result of learning

NDP		
Structure	The smallest error	Number of epochs
Mamdani with weights	0.0238	389
Larsen with weights	0.0164	495
Larsen (simplified) with weights	0.0136	487

desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.13, neither for the Mamdani nor for the Larsen structure the system was able to learn as to obtain the desired values of error.

9.3.3.2.3. NDP

Table 9.14 presents the smallest error for individual structures and the number of epochs corresponding to this error. Table 9.15 presents three desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.15, for the Mamdani structure with weights of rules, the system was unable to learn as to obtain the error 0.020 and 0.023.

9.3.3.2.4. Modeling the taste of rice

Table 9.16 presents the smallest error for individual structures and the number of epochs corresponding to this error. Table 9.17 presents three desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.17, for the Larsen structure with weights of rules, the system was unable to learn as to obtain the error 0.022.

9.3.3.3. M3-type systems

Let us consider Mamdani type systems which are constructed using definitions of triangular norms taking into account the weights w_k, characterizing

TABLE 9.15. Number of epochs required to train the system which is characterized by a definite error

NDP			
Structure	Value of error		
	0.028	0.020	0.023
Mamdani with weights	157	–	–
Larsen with weights	121	180	272
Larsen (simplified) with weights	55	59	90

TABLE 9.16. The smallest error obtained as a result of learning

MODELING THE TASTE OF RICE		
Structure	The smallest error	Number of epochs
Mamdani with weights	0.0178	4978
Larsen with weights	0.0229	4154
Larsen (simplified) with weights	0.0199	4935

TABLE 9.17. Number of epochs required to train the system which is characterized by a definite error

MODELING THE TASTE OF RICE			
Structure	Value of error		
	0.028	0.025	0.022
Mamdani with weights	335	716	1751
Larsen with weights	562	1479	–
Larsen (simplified) with weights	421	852	3264

the importance of particular rules, and the weights $w_{i,k}$, characterizing the importance of particular input linguistic variables. Using the definition of weighted t-conorm and dependencies (9.22), (9.24) and (9.25), we obtain the following description of the neuro-fuzzy systems:

a) Mamdani system with weights of inputs and rules

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{*N}\left\{\min\left(T_{i=1}^{*n}\left\{\mu_{A_i^k}\left(\overline{x}_i\right), w_{i,k}\right\}, \mu_{B^k}\left(\overline{y}^r\right)\right), w_k\right\}}{\sum_{r=1}^{N} S_{k=1}^{*N}\left\{\min\left(T_{i=1}^{*n}\left\{\mu_{A_i^k}\left(\overline{x}_i\right), w_{i,k}\right\}, \mu_{B^k}\left(\overline{y}^r\right)\right), w_k\right\}} \tag{9.29}$$

$$= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{*N}\left\{\min\begin{pmatrix} T_{i=1}^{*n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^k}{\sigma_i^k}\right)^2\right], w_{i,k}\right\}, \\ \exp\left[-\left(\frac{\overline{y}^r-\overline{y}^k}{\sigma^k}\right)^2\right] \end{pmatrix}, w_k\right\}}{\sum_{r=1}^{N} S_{k=1}^{*N}\left\{\min\begin{pmatrix} T_{i=1}^{*n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^k}{\sigma_i^k}\right)^2\right], w_{i,k}\right\}, \\ \exp\left[-\left(\frac{\overline{y}^r-\overline{y}^k}{\sigma^k}\right)^2\right] \end{pmatrix}, w_k\right\}}.$$

b) Larsen system with weights of inputs and rules

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{*N}\left\{T_{i=1}^{*n}\left\{\mu_{A_i^k}(\overline{x}_i), w_{i,k}\right\} \cdot \mu_{B^k}(\overline{y}^r), w_k\right\}}{\sum_{r=1}^{N} S_{k=1}^{*N}\left\{T_{i=1}^{*n}\left\{\mu_{A_i^k}(\overline{x}_i), w_{i,k}\right\} \cdot \mu_{B^k}(\overline{y}^r), w_k\right\}} \tag{9.30}$$

$$= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot S_{k=1}^{*N}\left\{T_{i=1}^{*n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^k}{\sigma_i^k}\right)^2\right], w_{i,k}\right\} \cdot \exp\left[-\left(\frac{\overline{y}^r-\overline{y}^k}{\sigma^k}\right)^2\right], w_k\right\}}{\sum_{r=1}^{N} S_{k=1}^{*N}\left\{T_{i=1}^{*n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^k}{\sigma_i^k}\right)^2\right], w_{i,k}\right\} \cdot \exp\left[-\left(\frac{\overline{y}^r-\overline{y}^k}{\sigma^k}\right)^2\right], w_k\right\}}.$$

In both a) and b) systems, the parameters of membership function, i.e. $\overline{x}_i^k, \sigma_i^k, \overline{y}^k, \sigma^k$ and weights $w_{i,k}$ and w_k are subject to learning.

c) Simplified Larsen system with weights of inputs and rules

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot w_r\left[T_{i=1}^{n}\left\{1 - w_{i,r}\left(1 - \mu_{A_i^r}\left(\overline{x}_i\right)\right)\right\}\right]}{\sum_{r=1}^{N} w_r\left[T_{i=1}^{n}\left\{1 - w_{i,r}\left(1 - \mu_{A_i^r}\left(\overline{x}_i\right)\right)\right\}\right]} \tag{9.31}$$

$$= \frac{\sum_{r=1}^{N} \overline{y}^r \cdot w_r\left[T_{i=1}^{n}\left\{1 - w_{i,r}\left(1 - \left(\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right)\right)\right\}\right]}{\sum_{r=1}^{N} w_r\left[T_{i=1}^{n}\left\{1 - w_{i,r}\left(1 - \left(\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right)\right)\right\}\right]}.$$

In system c) the parameters of membership function, i.e. $\overline{x}_i^r, \sigma_i^r, \overline{y}^r$ and weights $w_{i,r}$ and w_r are subject to learning. Neuro-fuzzy systems (9.29), (9.30) and (9.31) have been used to solve four problems specified in Table 9.1.

9.3.3.3.1. Polymerization

Table 9.18 presents the smallest error for individual structures and the number of epochs corresponding to this error. Table 9.19 presents three desired values of error and the number of epochs, after which this error was obtained.

9.3.3.3.2. HANG

Table 9.20 presents the smallest error for individual structures and the number of epochs corresponding to this error. Table 9.21 presents three desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.21, the desired error values could not be obtained for the Larsen structure with weights of inputs and rules.

TABLE 9.18. The smallest error obtained as a result of learning

POLYMERIZATION		
Structure	The smallest error	Number of epochs
Mamdani with weights of inputs and rules	0.0034	4704
Larsen with weights of inputs and rules	0.0035	3822
Larsen (simplified) with weights of inputs and rules	0.0031	2953

TABLE 9.19. Number of epochs required to train the system which is characterized by a definite error

POLYMERIZATION			
Structure	Value of error		
	0.0055	0.0050	0.0045
Mamdani with weights of inputs and rules	1915	2303	2549
Larsen with weights of inputs and rules	1	1	1
Larsen (simplified) with weights of inputs and rules	1	6	13

TABLE 9.20. The smallest error obtained as a result of learning

HANG		
Structure	The smallest error	Number of epochs
Mamdani with weights of inputs and rules	0.0209	5474
Larsen with weights of inputs and rules	0.0346	1541
Larsen (simplified) with weights of inputs and rules	0.0124	4252

TABLE 9.21. Number of epochs required to train the system which is characterized by a definite error

HANG			
Structure	Value of error		
	0.028	0.026	0.024
Mamdani with weights of inputs and rules	4213	5474	5474
Larsen with weights of inputs and rules	–	–	–
Larsen (simplified) with weights of inputs and rules	628	750	750

TABLE 9.22. The smallest error obtained as a result of learning

NDP		
Structure	The smallest error	Number of epochs
Mamdani with weights of inputs and rules	0.0181	498
Larsen with weights of inputs and rules	0.0146	500
Larsen (simplified) with weights of inputs and rules	0.0188	484

9.3.3.3.3. NDP

Table 9.22 presents the smallest error for individual structures and the number of epochs corresponding to this error. Table 9.23 presents three

TABLE 9.23. Number of epochs required to train the system which is characterized by a definite error

NDP			
Structure	Value of error		
	0.026	0.023	0.020
Mamdani with weights of inputs and rules	60	126	294
Larsen with weights of inputs and rules	24	31	74
Larsen (simplified) with weights of inputs and rules	–	–	–

TABLE 9.24. The smallest error obtained as a result of learning

MODELING THE TASTE OF RICE		
Structure	The smallest error	Number of epochs
Mamdani with weights of inputs and rules	0.0168	2218
Larsen with weights of inputs and rules	0.0218	2325
Larsen (simplified) with weights of inputs and rules	0.0190	4975

desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.23, the desired error values could not be obtained for the simplified Larsen structure with weights of inputs and rules.

9.3.3.3.4. Modeling the taste of rice

Table 9.24 presents the smallest error for individual structures and the number of epochs corresponding to this error.

Table 9.25 presents three desired values of error and the number of epochs, after which this error was obtained.

9.4 Neuro-fuzzy systems of logical type

In the previous subchapter, we have discussed neuro-fuzzy systems with Mamdani type inference. Currently, we will consider systems in which the

TABLE 9.25. Number of epochs required to train the system which is characterized by a definite error

MODELING THE TASTE OF RICE			
Structure	Value of error		
	0.028	0.025	0.022
Mamdani with weights of inputs and rules	1	3	3
Larsen with weights of inputs and rules	295	528	2325
Larsen (simplified) with weights of inputs and rules	1	1	5

antecedents and the consequents of rules are connected with each other using a fuzzy implication.

In logical type systems, the defuzzification is made by means of dependency

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \mu_{B'}\left(\overline{y}^r\right)}{\sum_{r=1}^{N} \cdot \mu_{B'}\left(\overline{y}^r\right)}. \tag{9.32}$$

In these systems, the fuzzy set B' is created as a result of intersection of fuzzy sets $\overline{B}^k$, i.e.

$$B' = \bigcap_{k=1}^{N} \overline{B}^k. \tag{9.33}$$

The membership function of fuzzy set B' is determined using a t-norm, which shall be notated as follows:

$$\mu_{B'}(y) = \overset{N}{\underset{k=1}{T}} \left\{ \mu_{\overline{B}^k}(y) \right\}. \tag{9.34}$$

Using formulas (9.34), (9.6) and (9.7), we have

$$\begin{aligned} \mu_{B'}\left(\overline{y}^r\right) &= \overset{N}{\underset{k=1}{T}} \left\{ \mu_{\overline{B}^k}\left(\overline{y}^r\right) \right\} = \overset{N}{\underset{k=1}{T}} \left\{ I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}\left(\overline{y}^r\right)\right) \right\} \\ &= \overset{N}{\underset{k=1}{T}} \left\{ I\left(\overset{N}{\underset{i=1}{T}} \mu_{A_i^k}(\overline{x}_i), \mu_{B^k}\left(\overline{y}^r\right) \right) \right\}, \end{aligned} \tag{9.35}$$

where I is a fuzzy implication defined in point 4.8.4. By substituting formula (9.35) to dependency (9.32), we obtain

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot T_{k=1}^{N} \left\{ I\left(T_{i=1}^{n} \left\{ \mu_{A_i^k}(\overline{x}_i) \right\}, \mu_{B^k}\left(\overline{y}^r\right) \right) \right\}}{\sum_{r=1}^{N} T_{k=1}^{N} \left\{ I\left(T_{i=1}^{n} \left\{ \mu_{A_i^k}(\overline{x}_i) \right\}, \mu_{B^k}\left(\overline{y}^r\right) \right) \right\}}. \tag{9.36}$$

The specific form of formula (9.36) depends on the chosen definition of I function. Logical type systems will be applied to solve modeling problems. We will consider M1 systems (without weights), M2 systems (with weights of rules) and M3 systems (with weights of rules and weights of input linguistic variables). We will apply the Łukasiewicz, binary, Reichenbach, Zadeh and Willmott fuzzy implications. Moreover we will present and test simplified neuro-fuzzy structures using Łukasiewicz and Zadeh implications.

9.4.1 M1-type systems

Let us consider logical type systems which are constructed using definitions of triangular norms without taking the weights into account. First, we will use Łukasiewicz implication. As a result of applying this implication, we will obtain the following dependency:

$$\begin{aligned}\mu_{A^k \to B^k}(\overline{\mathbf{x}}, y) &= I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right) = I\left(\mathop{T}_{k=1}^{n}\left(\mu_{A_i^k}(\overline{x}_i)\right), \mu_{B^k}(y)\right) \\ &= \min\left[1, 1 - \mathop{T}_{i=1}^{n}\left(\mu_{A_i^k}(\overline{x}_i)\right) + \mu_{B^k}(y)\right].\end{aligned} \tag{9.37}$$

By substituting dependency (9.37) to formula (9.36), we obtain

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{N} \left\{\min\left[1, 1 - T_{i=1}^{n}\left(\mu_{A_i^k}(\overline{x}_i)\right) + \mu_{B^k}(\overline{y}^r)\right]\right\}}{\sum_{r=1}^{N} T_{k=1}^{N} \left\{\min\left[1, 1 - T_{i=1}^{n}\left(\mu_{A_i^k}(\overline{x}_i)\right) + \mu_{B^k}(\overline{y}^r)\right]\right\}}. \tag{9.38}$$

By substituting dependencies (9.13) and (9.14) to formula (9.38), we obtain

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{N} \left\{\min\left[\begin{array}{l} 1, 1 - T_{i=1}^{n}\left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right) \\ + \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k}\right)^2\right] \end{array}\right]\right\}}{\sum_{r=1}^{N} T_{k=1}^{N} \left\{\min\left[\begin{array}{l} 1, 1 - T_{i=1}^{n}\left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right) \\ + \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k}\right)^2\right] \end{array}\right]\right\}}. \tag{9.39}$$

By applying the binary fuzzy implication, we obtain

$$\begin{aligned}\mu_{A^k \to B^k}(\overline{\mathbf{x}}, y) &= I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right) = I\left(\mathop{T}_{i=1}^{n}\left(\mu_{A_i^k}(\overline{x}_i)\right), \mu_{B^k}(y)\right) \\ &= \max\left[1 - \mathop{T}_{i=1}^{n}\left(\mu_{A_i^k}(\overline{x}_i)\right), \mu_{B^k}(y)\right].\end{aligned} \tag{9.40}$$

By substituting dependency (9.40) to formula (9.36), we obtain

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{N} \left\{ \max \left[1 - T_{i=1}^{n} \left(\mu_{A_i^k} \left(\overline{x}_i \right) \right), \mu_{B^k} \left(\overline{y}^r \right) \right] \right\}}{\sum_{r=1}^{N} T_{k=1}^{N} \left\{ \max \left[1 - T_{i=1}^{n} \left(\mu_{A_i^k} \left(\overline{x}_i \right) \right), \mu_{B^k} \left(\overline{y}^r \right) \right] \right\}}. \tag{9.41}$$

By substituting dependencies (9.13) and (9.14) to formula (9.41), we obtain

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{N} \left\{ \max \left[\begin{array}{c} 1 - T_{i=1}^{n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right), \\ \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \end{array} \right] \right\}}{\sum_{r=1}^{N} T_{k=1}^{N} \left\{ \max \left[\begin{array}{c} 1 - T_{i=1}^{n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right), \\ \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \end{array} \right] \right\}}. \tag{9.42}$$

Applying Reichenbach fuzzy implication we get:

$$\begin{aligned} \mu_{A^k \to B^k} \left(\overline{\mathbf{x}}, y \right) &= I \left(\mu_{A^k} \left(\overline{\mathbf{x}} \right), \mu_{B^k} \left(y \right) \right) = I \left(\mathop{T}_{i=1}^{n} \left(\mu_{A_i^k} \left(\overline{x}_i \right) \right), \mu_{B^k} \left(y \right) \right) \\ &= 1 - \mathop{T}_{i=1}^{n} \left(\mu_{A_i^k} \left(\overline{x}_i \right) \right) \left(1 - \mu_{B^k} \left(y \right) \right). \end{aligned} \tag{9.43}$$

By substituting dependency (9.43) to formula (9.36), we obtain

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{N} \left\{ 1 - T_{i=1}^{n} \left(\mu_{A_i^k} \left(\overline{x}_i \right) \right) \left(1 - \mu_{B^k} \left(\overline{y}^r \right) \right) \right\}}{\sum_{r=1}^{N} T_{k=1}^{N} \left\{ 1 - T_{i=1}^{n} \left(\mu_{A_i^k} \left(\overline{x}_i \right) \right) \left(1 - \mu_{B^k} \left(\overline{y}^r \right) \right) \right\}}. \tag{9.44}$$

By substituting dependencies (9.13) and (9.14) to formula (9.44), we have

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{N} \left\{ \begin{array}{l} 1 - T_{i=1}^{n} \left(\exp \left[-\left(\dfrac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right) \\ \left(1 - \exp \left[-\left(\dfrac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right) \end{array} \right\}}{\sum_{r=1}^{N} T_{k=1}^{N} \left\{ \begin{array}{l} 1 - T_{i=1}^{n} \left(\exp \left[-\left(\dfrac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right) \\ \left(1 - \exp \left[-\left(\dfrac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right) \end{array} \right\}}. \tag{9.45}$$

Applying Zadeh fuzzy implication we get:

$$\mu_{A^k \to B^k}(\overline{\mathbf{x}}, y) = I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right) = I\left(\mathop{T}_{i=1}^{n}\left(\mu_{A_i^k}(\overline{x}_i)\right), \mu_{B^k}(y)\right) \quad (9.46)$$
$$= \max\left\{\min\left[\mathop{T}_{i=1}^{n}\left(\mu_{A_i^k}(\overline{x}_i)\right), \mu_{B^k}(y)\right], 1 - \mathop{T}_{i=1}^{n}\left(\mu_{A_i^k}(\overline{x}_i)\right)\right\}.$$

By substituting dependency (9.46) to formula (9.36), we obtain

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T \left\{ \begin{array}{l} \max\left\{ \mathop{T}_{i=1}^{n}\left\{\mu_{A_i^r}(\overline{x}_i)\right\}, 1 - \mathop{T}_{i=1}^{n}\left\{\mu_{A_i^r}(\overline{x}_i)\right\}\right\}, \\ \mathop{T}_{\substack{k=1 \\ k \neq r}}^{N} \left\{ \max\left\{ \begin{array}{l} \min\left[\mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\overline{x}_i)\right\}, \mu_{B^k}(\overline{y}^r)\right], \\ 1 - \mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\overline{x}_i)\right\} \end{array} \right\} \right\} \end{array} \right\}}{\sum_{r=1}^{N} T \left\{ \begin{array}{l} \max\left\{ \mathop{T}_{i=1}^{n}\left\{\mu_{A_i^r}(\overline{x}_i)\right\}, 1 - \mathop{T}_{i=1}^{n}\left\{\mu_{A_i^r}(\overline{x}_i)\right\}\right\}, \\ \mathop{T}_{\substack{k=1 \\ k \neq r}}^{N} \left\{ \max\left\{ \begin{array}{l} \min\left[\mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\overline{x}_i)\right\}, \mu_{B^k}(\overline{y}^r)\right], \\ 1 - \mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\overline{x}_i)\right\} \end{array} \right\} \right\} \end{array} \right\}}. \quad (9.47)$$

By substituting dependencies (9.13) and (9.14) to formula (9.47), we get:

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T \left\{ \begin{array}{l} \max\left\{ \begin{array}{l} \mathop{T}_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\}, \\ 1 - \mathop{T}_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\} \end{array} \right\}, \\ \mathop{T}_{\substack{k=1 \\ k \neq r}}^{N} \left\{ \max\left\{ \begin{array}{l} \min\left\{ \begin{array}{l} \mathop{T}_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right\}, \\ \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k}\right)^2\right] \end{array} \right\}, \\ 1 - \mathop{T}_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right\} \end{array} \right\} \right\} \end{array} \right\}}{\sum_{r=1}^{N} T \left\{ \begin{array}{l} \max\left\{ \begin{array}{l} \mathop{T}_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\}, \\ 1 - \mathop{T}_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\} \end{array} \right\}, \\ \mathop{T}_{\substack{k=1 \\ k \neq r}}^{N} \left\{ \max\left\{ \begin{array}{l} \min\left\{ \begin{array}{l} \mathop{T}_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right\}, \\ \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k}\right)^2\right] \end{array} \right\}, \\ 1 - \mathop{T}_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right\} \end{array} \right\} \right\} \end{array} \right\}}. \quad (9.48)$$

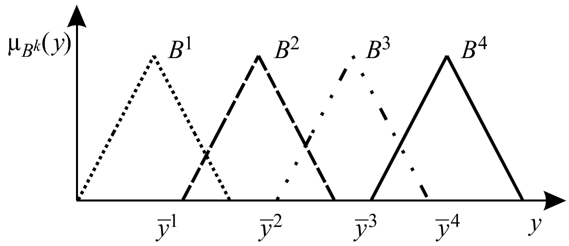

FIGURE 9.3. Example fuzzy sets satisfying the assumption $\mu_{B^k}\left(\overline{y}^r\right) \approx 0$

The simplified systems were also studied which are characterized by a small coincidence or total separation one from another of output fuzzy sets B^k. In this situation, the condition $\mu_{B^k}\left(\overline{y}^r\right) \approx 0$ is satisfied, which is illustrated by Fig. 9.3.

If $\mu_{B^k}\left(\overline{y}^r\right) \approx 0$, then we will obtain from dependency (9.39) a simplified Łukasiewicz structure of the following form

$$\overline{y} = \frac{\sum\limits_{r=1}^{N} \overline{y}^r \mathop{T}\limits_{\substack{k=1\\k\neq r}}^{N} \left\{1 - \mathop{T}\limits_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\} \right\}}{\sum\limits_{r=1}^{N} \mathop{T}\limits_{\substack{k=1\\k\neq r}}^{N} \left\{1 - \mathop{T}\limits_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\} \right\}}. \tag{9.49}$$

Similarly, if $\mu_{B^k}\left(\overline{y}^r\right) \approx 0$, then we will obtain from dependency (9.48) a simplified Zadeh structure given by the formula

$$\overline{y} = \frac{\sum\limits_{r=1}^{N} \overline{y}^r T \left\{ \begin{array}{l} \max \left\{ \begin{array}{l} \mathop{T}\limits_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right\}, \\ 1 - \mathop{T}\limits_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right\} \end{array} \right\}, \\ \mathop{T}\limits_{\substack{k=1\\k\neq r}}^{N} \left\{ 1 - \mathop{T}\limits_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\} \right\} \end{array} \right\}}{\sum\limits_{r=1}^{N} T \left\{ \begin{array}{l} \max \left\{ \begin{array}{l} \mathop{T}\limits_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right\}, \\ 1 - \mathop{T}\limits_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right\} \end{array} \right\}, \\ \mathop{T}\limits_{\substack{k=1\\k\neq r}}^{N} \left\{ 1 - \mathop{T}\limits_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right\} \right\} \end{array} \right\}}. \tag{9.50}$$

In systems (9.39), (9.42), (9.45) and (9.48) the following parameters of the membership functions are subject to learning: $\overline{x}_i^k, \sigma_i^k, \overline{y}^k, \sigma^k$. In simplified systems (9.49) and (9.50), the parameters $\overline{x}_i^k, \sigma_i^k, \overline{y}^k$ are subject to learning. We will solve the modeling problems using Łukasiewicz structure, binary structure, Reichenbach structure, Łukasiewicz simplified structure, Zadeh structure, Willmott structure and Zadeh simplified structure.

9.4.1.1. Polymerization

Table 9.26 presents the smallest error for individual structures and the number of epochs corresponding to this error.

As it may be inferred from Table 9.26, the smallest error was 0.0038 and was obtained for Zadeh structure.

Table 9.27 presents three desired values of error and the number of epochs, after which this error was obtained. As it may be inferred from this table, not all the structures were able to obtain the desired error value.

TABLE 9.26. The smallest error obtained as a result of learning

POLYMERIZATION		
Structure	The smallest error	Number of epochs
Łukasiewicz	0.0065	5863
Binary	0.0063	5980
Reichenbach	0.0040	5494
Łukasiewicz simplified	0.0059	3385
Zadeh	0.0038	3648
Willmott	0.0056	5918
Zadeh simplified	0.0049	3432

TABLE 9.27. Number of epochs required to train the system which is characterized by a definite error

POLYMERIZATION			
Structure	Value of error		
	0.0055	0.0050	0.0040
Łukasiewicz	–	–	–
Binary	–	–	–
Reichenbach	1627	1903	2783
Łukasiewicz simplified	–	–	–
Zadeh	949	1022	1809
Willmott	–	–	–
Zadeh simplified	2844	3432	–

9.4.1.2. HANG

Table 9.28 presents the smallest error for individual structures and the number of epochs corresponding to this error.

As it may be inferred from Table 9.28, the smallest error was 0.0177 and was obtained for binary structure. Table 9.29 presents three desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.29, not all the structures were able to obtain the desired value of error.

9.4.1.3. NDP

Table 9.30 presents the smallest error for individual structures and the number of epochs corresponding to this error.

TABLE 9.28. The smallest error obtained as a result of learning

HANG		
Structure	The smallest error	Number of epochs
Łukasiewicz	0.0289	3908
Binary	0.0177	7773
Reichenbach	0.0320	7989
Łukasiewicz simplified	0.0361	6536
Zadeh	0.0216	5288
Willmott	0.0366	7327
Zadeh simplified	0.0265	2317

TABLE 9.29. Number of epochs required to train the system which is characterized by a definite error

HANG			
Structure	Value of error		
	0.028	0.026	0.024
Łukasiewicz	–	–	–
Binary	1795	2996	3762
Reichenbach	–	–	–
Łukasiewicz simplified	–	–	–
Zadeh	3382	3875	4218
Willmott	–	–	–
Zadeh simplified	1787	–	–

TABLE 9.30. The smallest error obtained as a result of learning

NDP		
Structure	The smallest error	Number of epochs
Łukasiewicz	0.0166	457
Binary	0.0149	437
Reichenbach	0.0157	454
Łukasiewicz simplified	0.0229	497
Zadeh	0.0156	498
Willmott	0.0180	488
Zadeh simplified	0.0156	496

TABLE 9.31. Number of epochs required to train the system which is characterized by a definite error

NDP			
Structure	Value of error		
	0.026	0.023	0.020
Łukasiewicz	255	276	313
Binary	74	111	166
Reichenbach	166	173	251
Łukasiewicz simplified	190	497	–
Zadeh	38	59	109
Willmott	121	171	303
Zadeh simplified	39	83	147

As it may be inferred from Table 9.30, the smallest error was 0.0149 and was obtained for the binary structure. Table 9.31 presents three desired values of error and the number of epochs, after which this error was obtained. As it may be inferred from Table 9.31, for the simplified Łukasiewicz structure, it was impossible to obtain the error equal to 0.020.

9.4.1.4. Modeling the taste of rice

Table 9.32 presents the smallest error for individual structures and the number of epochs corresponding to this error.

As it may be inferred from Table 9.32, the smallest error was 0.0211 and was obtained for Willmott structure. Table 9.33 presents three desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.33, not all the structures were able to obtain the desired value of error equal to 0.022.

TABLE 9.32. The smallest error obtained as a result of learning

MODELING THE TASTE OF RICE		
Structure	The smallest error	Number of epochs
Łukasiewicz	0.0221	4048
Binary	0.0230	3201
Reichenbach	0.0212	4575
Łukasiewicz simplified	0.0243	2328
Zadeh	0.0219	4534
Willmott	0.0211	2605
Zadeh simplified	0.0246	4588

TABLE 9.33. Number of epochs required to train the system which is characterized by a definite error

MODELING THE TASTE OF RICE			
Structure	Value of error		
	0.028	0.025	0.022
Łukasiewicz	408	2327	–
Binary	286	817	–
Reichenbach	313	938	2354
Łukasiewicz simplified	1030	1850	–
Zadeh	344	1484	4534
Willmott	134	517	2605
Zadeh simplified	998	4588	–

9.4.2 M2-type systems

Let us consider logical type systems which are constructed using definitions of triangular norms taking into account the weights w_k, characterizing the importance of particular rules. Using the definition of weighted t-norm and dependencies (9.39), (9.42), (9.45), and (9.48) - (9.50), we obtain the following neuro-fuzzy systems:

a) Neuro-fuzzy system with weights of rules and the Łukasiewicz implication

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{*N} \left\{ \min \left[\begin{array}{l} 1, 1 - \underset{i=1}{\overset{n}{T}} \left(\exp \left[- \left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right) \\ + \exp \left[- \left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \end{array} \right], w_k \right\}}{\sum_{r=1}^{N} T_{k=1}^{*N} \left\{ \min \left[\begin{array}{l} 1, 1 - \underset{i=1}{\overset{n}{T}} \left(\exp \left[- \left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right] \right) \\ + \exp \left[- \left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \end{array} \right], w_k \right\}}. \quad (9.51)$$

b) Neuro-fuzzy system with weights of rules and the binary implication

$$\bar{y}=\frac{\sum_{r=1}^{N}\bar{y}^{r}\,\mathop{T^{*}}_{k=1}^{N}\left\{\max\left[\begin{array}{l}1-\mathop{T}\limits_{i=1}^{n}\left(\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{k}}{\sigma_i^{k}}\right)^{2}\right]\right),\\ \exp\left[-\left(\frac{\bar{y}^{r}-\bar{y}^{k}}{\sigma^{k}}\right)^{2}\right]\end{array}\right],w_k\right\}}{\sum_{r=1}^{N}\mathop{T^{*}}_{k=1}^{N}\left\{\max\left[\begin{array}{l}1-\mathop{T}\limits_{i=1}^{n}\left(\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{k}}{\sigma_i^{k}}\right)^{2}\right]\right),\\ \exp\left[-\left(\frac{\bar{y}^{r}-\bar{y}^{k}}{\sigma^{k}}\right)^{2}\right]\end{array}\right],w_k\right\}}. \tag{9.52}$$

c) Neuro-fuzzy system with weights of rules and the Reichenbach implication

$$\bar{y}=\frac{\sum_{r=1}^{N}\bar{y}^{r}\,\mathop{T^{*}}_{k=1}^{N}\left\{\begin{array}{l}1-\mathop{T}\limits_{i=1}^{n}\left(\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{k}}{\sigma_i^{k}}\right)^{2}\right]\right)\\ \left(1-\exp\left[-\left(\frac{\bar{y}^{r}-\bar{y}^{k}}{\sigma^{k}}\right)^{2}\right]\right),w_k\end{array}\right\}}{\sum_{r=1}^{N}\mathop{T^{*}}_{k=1}^{N}\left\{\begin{array}{l}1-\mathop{T}\limits_{i=1}^{n}\left(\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{k}}{\sigma_i^{k}}\right)^{2}\right]\right)\\ \left(1-\exp\left[-\left(\frac{\bar{y}^{r}-\bar{y}^{k}}{\sigma^{k}}\right)^{2}\right]\right),w_k\end{array}\right\}}. \tag{9.53}$$

d) Neuro-fuzzy system with weights of rules and a fuzzy Zadeh implication

$$\bar{y}=\frac{\sum\limits_{r=1}^{N}\bar{y}^{r}T^{*}\left\{\begin{array}{l}\max\left\{\begin{array}{l}\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{r}}{\sigma_i^{r}}\right)^{2}\right]\right\},\\ 1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{r}}{\sigma_i^{r}}\right)^{2}\right]\right\}\end{array}\right\},w_r,\\ \mathop{T}\limits_{\substack{k=1\\k\neq r}}^{N}\left\{\max\left\{\begin{array}{l}\min\left\{\begin{array}{l}\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{k}}{\sigma_i^{k}}\right)^{2}\right]\right\},\\ \exp\left[-\left(\frac{\bar{y}^{r}-\bar{y}^{k}}{\sigma^{k}}\right)^{2}\right]\end{array}\right\},\\ 1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{k}}{\sigma_i^{k}}\right)^{2}\right]\right\}\end{array}\right\},w_k\right\}\end{array}\right\}}{\sum\limits_{r=1}^{N}T^{*}\left\{\begin{array}{l}\max\left\{\begin{array}{l}\mathop{T^{*}}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{r}}{\sigma_i^{r}}\right)^{2}\right]\right\},\\ 1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{r}}{\sigma_i^{r}}\right)^{2}\right]\right\}\end{array}\right\},w_r,\\ \mathop{T}\limits_{\substack{k=1\\k\neq r}}^{N}\left\{\max\left\{\begin{array}{l}\min\left\{\begin{array}{l}\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{k}}{\sigma_i^{k}}\right)^{2}\right]\right\},\\ \exp\left[-\left(\frac{\bar{y}^{r}-\bar{y}^{k}}{\sigma^{k}}\right)^{2}\right]\end{array}\right\},\\ 1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\bar{x}_i-\bar{x}_i^{k}}{\sigma_i^{k}}\right)^{2}\right]\right\}\end{array}\right\},w_k\right\}\end{array}\right\}}. \tag{9.54}$$

e) Simplified neuro-fuzzy system with weights of rules and a fuzzy Łukasiewicz implication

$$\overline{y}=\frac{\sum\limits_{r=1}^{N}\overline{y}^{r}\mathop{T}\limits_{\substack{k=1\\k\neq r}}^{N}\left\{1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right\},w_k\right\}}{\sum\limits_{r=1}^{N}\mathop{T}\limits_{\substack{k=1\\k\neq r}}^{N}\left\{1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right\},w_k\right\}}. \tag{9.55}$$

f) Simplified neuro-fuzzy system with weights of rules and the Zadeh implication

$$\overline{y}=\frac{\sum\limits_{r=1}^{N}\overline{y}^{r}T^{*}\left\{\begin{array}{c}\left(\max\left\{\begin{array}{c}\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\},\\1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\}\end{array}\right\}\right),w_r,\\\mathop{T^{*}}\limits_{\substack{k=1\\k\neq r}}^{N}\left\{1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right\},w_k\right\}\end{array}\right\}}{\sum\limits_{r=1}^{N}T^{*}\left\{\begin{array}{c}\left(\max\left\{\begin{array}{c}\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\},\\1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\}\end{array}\right\}\right),w_r,\\\mathop{T^{*}}\limits_{\substack{k=1\\k\neq r}}^{N}\left\{1-\mathop{T}\limits_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^k}{\sigma_i^k}\right)^2\right]\right\},w_k\right\}\end{array}\right\}}. \tag{9.56}$$

In systems (9.51) - (9.54) the parameters of the membership functions, i.e. $\overline{x}_i^k, \sigma_i^k, \overline{y}^k, \sigma^k$ and weights w_k are subject to learning. In systems (9.55) and (9.56), the parameters of membership function $\overline{x}_i^k, \sigma_i^k, \overline{y}^k$ and weights w_k are subject to learning.

9.4.2.1. Polymerization

Table 9.34 presents the smallest error for individual structures and the number of epochs corresponding to this error. As it may be inferred from the table, the smallest error was 0.0030 and was obtained for Zadeh structure with weights of rules. Table 9.35 presents three desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from the table, not all the structures were able to obtain the desired value of error.

9.4.2.2. HANG

Table 9.36 presents the smallest error for individual structures and the number of epochs corresponding to this error.

TABLE 9.34. The smallest error obtained as a result of learning

POLYMERIZATION		
Structure	The smallest error	Number of epochs
Łukasiewicz with weights of rules	0.0041	4765
Binary with weights of rules	0.0054	5980
Reichenbach with weights of rules	0.0037	4653
Łukasiewicz simplified with weights of rules	0.0039	4694
Zadeh with weights of rules	0.0030	5650
Willmott with weights of rules	0.0047	5539
Zadeh simplified with weights of rules	0.0041	5151

TABLE 9.35. Number of epochs required to train the system which is characterized by a definite error

POLYMERIZATION			
Structure	Value of error		
	0.0055	0.0050	0.0040
Łukasiewicz with weights of rules	1258	1662	3266
Binary with weights of rules	5980	–	–
Reichenbach with weights of rules	1385	1385	2521
Łukasiewicz simplified with weights of rules	4	4	209
Zadeh with weights of rules	1497	1497	2726
Willmott with weights of rules	1405	3084	–
Zadeh simplified with weights of rules	367	701	3103

As it may be inferred from Table 9.36, the smallest error was 0.0115 and was obtained for Reichenbach structure with weights of rules. Table 9.37 presents three desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.37, not all the structures were able to obtain the desired value of error.

9.4.2.3. NDP

Table 9.38 presents the smallest error for individual structures and the number of epochs corresponding to this error.

TABLE 9.36. The smallest error obtained as a result of learning

HANG		
Structure	The smallest error	Number of epochs
Łukasiewicz with weights of rules	0.0247	6500
Binary with weights of rules	0.0161	6525
Reichenbach with weights of rules	0.0115	7580
Łukasiewicz simplified with weights of rules	0.0350	1840
Zadeh with weights of rules	0.0202	5290
Willmott with weights of rules	0.0335	7977
Zadeh simplified with weights of rules	0.0231	7935

TABLE 9.37. Number of epochs required to train the system which is characterized by a definite error

HANG			
Structure	Value of error		
	0.028	0.026	0.024
Łukasiewicz with weights of rules	3771	3908	–
Binary with weights of rules	1320	1320	1929
Reichenbach with weights of rules	506	660	660
Łukasiewicz simplified with weights of rules	–	–	–
Zadeh with weights of rules	3380	3393	4089
Willmott with weights of rules	–	–	–
Zadeh simplified with weights of rules	2483	2483	4139

As it may be inferred from Table 9.38, the smallest error was 0.0131 and was obtained for binary structure with weights of rules. Table 9.39 presents three desired values of error and the number of epochs, after which this error was obtained.

9.4.2.4. Modeling the taste of rice

Table 9.40 presents the smallest error for individual structures and the number of epochs corresponding to this error.

As it may be inferred from Table 9.40, the smallest error was 0.0199 and was obtained for Willmott structure with weights of rules. Table 9.41

TABLE 9.38. The smallest error obtained as a result of learning

NDP		
Structure	The smallest error	Number of epochs
Łukasiewicz with weights of rules	0.0161	492
Binary with weights of rules	0.0131	498
Reichenbach with weights of rules	0.0140	489
Łukasiewicz simplified with weights of rules	0.0177	459
Zadeh with weights of rules	0.0148	499
Willmott with weights of rules	0.0165	486
Zadeh simplified with weights of rules	0.0142	448

TABLE 9.39. Number of epochs required to train the system which is characterized by a definite error

NDP			
Structure	Value of error		
	0.026	0.023	0.020
Łukasiewicz with weights of rules	170	218	274
Binary with weights of rules	101	119	151
Reichenbach with weights of rules	139	153	186
Łukasiewicz simplified with weights of rules	267	277	364
Zadeh with weights of rules	222	281	352
Willmott with weights of rules	86	121	331
Zadeh simplified with weights of rules	58	78	212

TABLE 9.40. The smallest error obtained as a result of learning

MODELING THE TASTE OF RICE		
Structure	The smallest error	Number of epochs
Łukasiewicz with weights of rules	0.0207	3257
Binary with weights of rules	0.0219	3897
Reichenbach with weights of rules	0.0205	3800
Łukasiewicz simplified with weights of rules	0.0222	2841
Zadeh with weights of rules	0.0205	3531
Willmott with weights of rules	0.0199	3805
Zadeh simplified with weights of rules	0.0227	4432

TABLE 9.41. Number of epochs required to train the system which is characterized by a definite error

MODELING THE TASTE OF RICE			
Structure	Value of error		
	0.028	0.025	0.022
Łukasiewicz with weights of rules	1	16	462
Binary with weights of rules	143	1117	3387
Reichenbach with weights of rules	38	185	394
Łukasiewicz simplified with weights of rules	397	1152	–
Zadeh with weights of rules	108	314	879
Willmott with weights of rules	22	78	374
Zadeh simplified with weights of rules	461	696	–

presents three desired values of error and the number of epochs, after which this error was obtained.

As it may be inferred from Table 9.41, the error of 0.022 could not be obtained for the simplified Łukasiewicz structure with weights of rules and Zadeh structure with weights of rules.

9.4.3 M3-type systems

Let us consider logical type systems which are constructed using definitions of triangular norms taking into account the weights w_k, characterizing the importance of particular rules, and the weights $w_{i,k}$, characterizing the importance of particular input linguistic variables. Using the definition of weighted t-norm and dependencies (9.37), (9.40), (9.43), (9.46), (9.49) and (9.50), we obtain the following neuro-fuzzy systems:

a) Neuro-fuzzy system with weights of inputs and rules and the Łukasiewicz implication

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{*N} \left\{ \min \left[\begin{array}{l} 1, 1 - T_{i=1}^{*n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right) \\ + \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \end{array} \right], w_k \right\}}{\sum_{r=1}^{N} T_{k=1}^{*N} \left\{ \min \left[\begin{array}{l} 1, 1 - T_{i=1}^{*n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right) \\ + \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \end{array} \right], w_k \right\}}. \quad (9.57)$$

b) Neuro-fuzzy system with weights of inputs and rules and the binary implication

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{*N} \left\{ \max \begin{bmatrix} 1 - T_{i=1}^{*n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right), \\ \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \end{bmatrix}, w_k \right\}}{\sum_{r=1}^{N} T_{k=1}^{*N} \left\{ \max \begin{bmatrix} 1 - T_{i=1}^{*n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right), \\ \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \end{bmatrix}, w_k \right\}}. \tag{9.58}$$

c) Neuro-fuzzy system with weights of inputs and rules and the Reichenbach implication

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T_{k=1}^{*N} \left\{ \begin{array}{c} 1 - T_{i=1}^{*n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right) \\ \left(1 - \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right), w_k \end{array} \right\}}{\sum_{r=1}^{N} T_{k=1}^{*N} \left\{ \begin{array}{c} 1 - T_{i=1}^{*n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right) \\ \left(1 - \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \right), w_k \end{array} \right\}}. \tag{9.59}$$

d) Neuro-fuzzy system with weights of inputs and rules and the Zadeh implication

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T^* \left\{ \begin{array}{l} \max \left\{ \begin{array}{l} \overset{n}{\underset{i=1}{T^*}} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right], w_{i,r} \right\}, \\ 1 - \overset{n}{\underset{i=1}{T^*}} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right], w_{i,r} \right\} \end{array} \right\}, w_r, \\ \overset{N}{\underset{\substack{k=1 \\ k \neq r}}{T^*}} \left\{ \max \left\{ \begin{array}{l} \min \overset{n}{\underset{i=1}{T^*}} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right\}, \\ \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right], \\ 1 - \overset{n}{\underset{i=1}{T^*}} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right\} \end{array} \right\}, w_k \right\} \end{array} \right\}}{\sum_{r=1}^{N} T^* \left\{ \begin{array}{l} \max \left\{ \begin{array}{l} \overset{n}{\underset{i=1}{T^*}} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right], w_{i,r} \right\}, \\ 1 - \overset{n}{\underset{i=1}{T^*}} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right], w_{i,r} \right\} \end{array} \right\}, w_r, \\ \overset{N}{\underset{\substack{k=1 \\ k \neq r}}{T^*}} \left\{ \max \left\{ \begin{array}{l} \min \left\{ \begin{array}{l} \overset{n}{\underset{i=1}{T^*}} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right\}, \\ \exp\left[-\left(\frac{\overline{y}^r - \overline{y}^k}{\sigma^k} \right)^2 \right] \end{array} \right\}, \\ 1 - \overset{n}{\underset{i=1}{T^*}} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right\} \end{array} \right\}, w_k \right\} \end{array} \right\}}. \tag{9.60}$$

e) Simplified neuro-fuzzy system with weights of inputs and rules and the Łukasiewicz implication

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \underset{\substack{k=1 \\ k \neq r}}{\overset{N}{T^*}} \left\{ 1 - T_{i=1}^{*n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right\}, w_k \right\}}{\sum_{r=1}^{N} \underset{\substack{k=1 \\ k \neq r}}{\overset{N}{T^*}} \left\{ 1 - T_{i=1}^{*n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right\}, w_k \right\}}. \tag{9.61}$$

f) Simplified neuro-fuzzy system with weights of inputs and rules and the Zadeh implication

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r T^* \left\{ \begin{array}{l} \max \left\{ \begin{array}{l} T_{i=1}^{*n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right], w_{i,r} \right\}, \\ 1 - T_{i=1}^{*n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right], w_{i,r} \right\} \end{array} \right\}, w_r, \\ \underset{\substack{k=1 \\ k \neq r}}{\overset{N}{T^*}} \left\{ 1 - T_{i=1}^{*n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right\}, w_k \right\} \end{array} \right\}}{\sum_{r=1}^{N} T^* \left\{ \begin{array}{l} \max \left\{ \begin{array}{l} T_{i=1}^{*n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right], w_{i,r} \right\}, \\ 1 - T_{i=1}^{*n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right], w_{i,r} \right\} \end{array} \right\}, w_r, \\ \underset{\substack{k=1 \\ k \neq r}}{\overset{N}{T^*}} \left\{ 1 - T_{i=1}^{*n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^k}{\sigma_i^k} \right)^2 \right], w_{i,k} \right\}, w_k \right\} \end{array} \right\}}. \tag{9.62}$$

In systems (9.57) - (9.60), the parameters of membership function, i.e. $\overline{x}_i^k$, $\sigma_i^k, \overline{y}^k, \sigma^k$ and weights $w_{i,k}$ and w_k are subject to learning. In systems (9.61) and (9.62), the parameters of membership function $\overline{x}_i^k, \sigma_i^k, \overline{y}^k$ and weights $w_{i,k}$ and w_k are subject to learning. Neuro-fuzzy systems (9.57) - (9.62) have been used to solve four problems specified in Table 9.1.

9.4.3.1. Polymerization

Table 9.42 presents the smallest error for individual structures and the number of epochs corresponding to this error.

As it may be inferred from Table 9.42, the smallest error was 0.0028 and was obtained for Zadeh structure with weights of inputs and rules. Table 9.43 presents three desired values of error and the number of epochs, after which this error was obtained.

TABLE 9.42. The smallest error obtained as a result of learning

POLYMERIZATION		
Structure	The smallest error	Number of epochs
Łukasiewicz with weights of inputs and rules	0.0038	4773
Binary with weights of inputs and rules	0.0036	4896
Reichenbach with weights of inputs and rules	0.0034	4704
Łukasiewicz simplified with weights of inputs and rules	0.0037	4815
Zadeh with weights of inputs and rules	0.0028	5064
Willmott with weights of inputs and rules	0.0039	4810
Zadeh simplified with weights of inputs and rules	0.0038	5515

TABLE 9.43. Number of epochs required to train the system which is characterized by a definite error

POLYMERIZATION			
Structure	Value of error		
	0.0055	0.0050	0.0040
Łukasiewicz with weights of inputs and rules	1	9	867
Binary with weights of inputs and rules	2305	2386	2798
Reichenbach with weights of inputs and rules	1915	2303	2549
Łukasiewicz simplified with weights of inputs and rules	2502	2821	3225
Zadeh with weights of inputs and rules	1	1	6
Willmott with weights of inputs and rules	11	90	1341
Zadeh simplified with weights of inputs and rules	2	2	206

9.4.3.2. HANG

Table 9.44 presents the smallest error for individual structures and the number of epochs corresponding to this error. Table 9.45 presents the results analogous to those given in Table 9.43.

TABLE 9.44. The smallest error obtained as a result of learning

HANG		
Structure	The smallest error	Number of epochs
Łukasiewicz with weights of inputs and rules	0.0207	6502
Binary with weights of inputs and rules	0.0110	7882
Reichenbach with weights of inputs and rules	0.0092	7390
Łukasiewicz simplified with weights of inputs and rules	0.0203	7996
Zadeh with weights of inputs and rules	0.0105	5533
Willmott with weights of inputs and rules	0.0300	6545
Zadeh simplified with weights of inputs and rules	0.0178	8000

TABLE 9.45. Number of epochs required to train the system which is characterized by a definite error

HANG			
Structure	Value of error		
	0.028	0.026	0.024
Łukasiewicz with weights of inputs and rules	3724	3771	3771
Binary with weights of inputs and rules	556	608	608
Reichenbach with weights of inputs and rules	603	678	978
Łukasiewicz simplified with weights of inputs and rules	7992	7992	7992
Zadeh with weights of inputs and rules	666	666	1115
Willmott with weights of inputs and rules	–	–	–
Zadeh simplified with weights of inputs and rules	3943	4408	5407

TABLE 9.46. The smallest error obtained as a result of learning

NDP		
Structure	The smallest error	Number of epochs
Łukasiewicz with weights of inputs and rules	0.0140	498
Binary with weights of inputs and rules	0.0121	479
Reichenbach with weights of inputs and rules	0.0133	497
Łukasiewicz simplified with weights of inputs and rules	0.0162	457
Zadeh with weights of inputs and rules	0.0140	4
Willmott with weights of inputs and rules	0.0141	496
Zadeh simplified with weights of inputs and rules	0.0135	496

9.4.3.3. NDP

Table 9.46 presents the smallest error for individual structures and the number of epochs corresponding to this error.

As it may be inferred from Table 9.46, the smallest error was 0.0121 and was obtained for binary structure with weights of inputs and rules. Table 9.47 presents three desired values of error and the number of epochs, after which this error was obtained

9.4.3.4. Modeling the taste of rice

Table 9.48 presents the smallest error for individual structures and the number of epochs corresponding to this error.

As it may be inferred from Table 9.48 the smallest error was 0.0164 and was obtained for Zadeh structure with weights of inputs and rules. Table 9.49 presents three desired values of error and the number of epochs, after which this error was obtained.

9.5 Neuro-fuzzy systems of Takagi-Sugeno type

In the fuzzy Takagi-Sugeno type model [246], the base of rules is of a fuzzy character only in the **IF** part, whereas in the **THEN** part, there are functional dependencies

$$\begin{array}{c} R^{(r)}: \textbf{IF} \;\; (x_1 \text{ is } A_i^r \;\; \textbf{AND} \;\; x_2 \text{ is } A_2^r \text{...A}\textbf{ND} \;\; x_n \text{ is } A_n^r) \\ \textbf{THEN} \;\; y_r = f^{(r)}\left(x_1, x_2, ..., x_n\right) \end{array} \tag{9.63}$$

TABLE 9.47. Number of epochs required to train the system which is characterized by a definite error

NDP			
Structure	Value of error		
	0.026	0.023	0.020
Łukasiewicz with weights of inputs and rules	279	295	368
Binary with weights of inputs and rules	61	80	94
Reichenbach with weights of inputs and rules	107	176	237
Łukasiewicz simplified with weights of inputs and rules	285	299	315
Zadeh with weights of inputs and rules	95	109	142
Willmott with weights of inputs and rules	80	109	150
Zadeh simplified with weights of inputs and rules	60	82	170

TABLE 9.48. The smallest error obtained as a result of learning

MODELING THE TASTE OF RICE		
Structure	The smallest error	Number of epochs
Łukasiewicz with weights of inputs and rules	0.0192	3031
Binary with weights of inputs and rules	0.0194	4164
Reichenbach with weights of inputs and rules	0.0191	4460
Łukasiewicz simplified with weights of inputs and rules	0.0201	4804
Zadeh with weights of inputs and rules	0.0164	3994
Willmott with weights of inputs and rules	0.0187	3916
Zadeh simplified with weights of inputs and rules	0.0186	3646

If we assume that the input of the fuzzy system is signal $\overline{\mathbf{x}} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$, then in order to obtain the output signal $\overline{y}$ of the system, first we will determine

$$T\left(\mu_{A_1^r}(\overline{x}_1), \mu_{A_2^r}(\overline{x}_2), ..., \mu_{A_n^r}(\overline{x}_n)\right), \quad r = 1, \ldots, N. \tag{9.64}$$

TABLE 9.49. Number of epochs required to train the system which is characterized by a definite error

MODELING THE TASTE OF RICE			
Structure	Value of error		
	0.028	0.025	0.022
Łukasiewicz with weights of inputs and rules	74	331	2045
Binary with weights of inputs and rules	143	317	1679
Reichenbach with weights of inputs and rules	2	3	8
Łukasiewicz simplified with weights of inputs and rules	165	450	1702
Zadeh with weights of inputs and rules	40	143	197
Willmott with weights of inputs and rules	1	1	37
Zadeh simplified with weights of inputs and rules	76	202	404

The next step is to compute

$$\overline{y}_r = f^{(r)}\left(\overline{x}_1, \overline{x}_2, ..., \overline{x}_n\right), \quad r = 1, \ldots, N. \tag{9.65}$$

The output signal of the fuzzy Takagi-Sugeno system is a normalized weighted sum of particular inputs $\overline{y}_1, ..., \overline{y}_N$, i.e.

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}_r T_{i=1}^{n} \left\{\mu_{A_i^r}\left(\overline{x}_i\right)\right\}}{\sum_{r=1}^{N} T_{i=1}^{n} \left\{\mu_{A_i^r}\left(\overline{x}_i\right)\right\}}. \tag{9.66}$$

In the following part of this subchapter, we will consider the Takagi-Sugeno systems with linear dependencies in consequents of the base of rules, i.e.

$$\begin{array}{c} R^{(r)}: \textbf{IF } (x_1 \text{ is } A_i^r \textbf{ AND } x_2 \text{ is } A_2^r ... \textbf{AND } x_n \text{ is } A_n^r) \\ \textbf{THEN } y_r = c_0^{(r)} + c_1^{(r)} x_1 + ... + c_n^{(r)} x_n \end{array} \tag{9.67}$$

for $r = 1, ..., N$. It should be noted that if $c_i^{(r)} = 0$, $i = 1, \ldots, n$, then system (9.66) is reduced to a simplified Mamdani system given by formula (9.12), and then $c_0^{(r)} = \overline{y}^r$, $r = 1, ..., N$.

The systems of Takagi-Sugeno type have been used to solve approximation and identification problems (polymerization, HANG, NDP, modeling the taste of rice). Like in case of Mamdani type structures and logical type structures, we will consider three types of systems, i.e. without weights, with weights of rules and with weights of rules and weights of inputs reflecting the importance of individual linguistic variables.

9.5.1 M1-type systems

To construct a neuro-fuzzy system, Gaussian membership functions and the assumption that the antecedents in each rule are connected by a t-norm of the product type have been used. In this situation, dependency (9.66) takes the following form

$$\begin{aligned}\overline{y} &= \frac{\sum_{r=1}^{N} T_{i=1}^{n}\left\{\mu_{A_i^r}\left(\overline{x}_i\right)\right\}\left(c_0^{(r)}+c_1^{(r)}x_1+\ldots+c_n^{(r)}x_n\right)}{\sum_{r=1}^{N} T_{i=1}^{n}\left\{\mu_{A_i^r}\left(\overline{x}_i\right)\right\}} \qquad (9.68)\\ &= \frac{\sum_{r=1}^{N} T_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\}\left(c_0^{(r)}+c_1^{(r)}x_1+\ldots+c_n^{(r)}x_n\right)}{\sum_{r=1}^{N} T_{i=1}^{n}\left\{\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right\}}\\ &= \frac{\sum_{r=1}^{N}\left[\prod_{i=1}^{n}\left(\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right)\right]\left(c_0^{(r)}+c_1^{(r)}x_1+\ldots+c_n^{(r)}x_n\right)}{\sum_{r=1}^{N}\left[\prod_{i=1}^{n}\left(\exp\left[-\left(\frac{\overline{x}_i-\overline{x}_i^r}{\sigma_i^r}\right)^2\right]\right)\right]}.\end{aligned}$$

All the parameters of the neuro-fuzzy systems have been subject to learning using error backpropagation method: centers and widths of Gaussian functions and function parameters $c_0^{(r)}, ..., c_n^{(r)}, r = 1, ..., N$.

9.5.1.1. Polymerization

The smallest error for the Takagi-Sugeno structure was 0.0034 and was obtained in the 3430th epoch. Table 9.50 presents three desired error values and the number of epochs, after which this error was obtained.

9.5.1.2. HANG

The smallest error for the Takagi-Sugeno structure was 0.0197 and was obtained in the 7551st epoch. Table 9.51 presents three desired values of error and the number of epochs, after which this error was obtained.

TABLE 9.50. Number of epochs required to train the system which is characterized by a definite error

POLYMERIZATION			
Structure	Value of error		
	0.0055	0.0050	0.0045
Takagi-Sugeno	72	83	83

9.5.1.3. NDP

The smallest error for the Takagi-Sugeno structure was 0.0156 and was obtained in the 481st epoch. Table 9.52 presents three desired values of error and the number of epochs, after which this error was obtained.

9.5.1.4. Modeling the taste of rice

The smallest error for the Takagi-Sugeno structure was 0.0176 and was obtained in the 1264th epoch. Table 9.53 presents three desired values of error and the number of epochs, after which this error was obtained.

9.5.2 M2-type systems

By introducing to system (9.68) the weights specifying the importance of particular rules, we will obtain the following dependency:

TABLE 9.51. Number of epochs required to train the system which is characterized by a definite error

HANG			
Structure	Value of error		
	0.028	0.026	0.024
Takagi-Sugeno	4280	5159	5593

TABLE 9.52. Number of epochs required to train the system which is characterized by a definite error

NDP			
Structure	Value of error		
	0.026	0.023	0.020
Takagi-Sugeno	36	66	122

TABLE 9.53. Number of epochs required to train the system which is characterized by a definite error

MODELING THE TASTE OF RICE			
Structure	Value of error		
	0.028	0.025	0.022
Takagi-Sugeno	546	1060	1951

$$\overline{y} = \frac{\sum_{r=1}^{N} w_r T_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right\} \left(c_0^{(r)} + c_1^{(r)} x_1 + \ldots + c_n^{(r)} x_n \right)}{\sum_{r=1}^{N} w_r T_{i=1}^{n} \left\{ \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right\}} \tag{9.69}$$

$$= \frac{\sum_{r=1}^{N} w_r \left[\prod_{i=1}^{n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right) \right] \left(c_0^{(r)} + c_1^{(r)} x_1 + \ldots + c_n^{(r)} x_n \right)}{\sum_{r=1}^{N} w_r \left[\prod_{i=1}^{n} \left(\exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right) \right]}.$$

All the parameters of the neuro-fuzzy systems have been subject to learning using the error backpropagation method: centers and widths of Gaussian functions, weights of rules and function parameters $c_0^{(r)}, ..., c_n^{(r)}, r = 1, ..., N$.

9.5.2.1. Polymerization

The smallest error for the Takagi-Sugeno structure was 0.0031 and was obtained in the 3098th epoch. Table 9.54 presents three desired values of error and the number of epochs, after which this error was obtained.

9.5.2.2. HANG

The smallest error for the Takagi-Sugeno structure with weights of rules was 0.0145 and was obtained in the 3008th epoch. Table 9.55 presents three desired values of error and the number of epochs, after which this error was obtained.

9.5.2.3. NDP

The smallest error for the Takagi-Sugeno structure with weights of rules was 0.0140 and was obtained in the 497th epoch. Table 9.56 presents three desired values of error and the number of epochs, after which this error was obtained.

TABLE 9.54. Number of epochs required to train the system which is characterized by a definite error

POLYMERIZATION			
Structure	Value of error		
	0.0055	0.0050	0.0045
Takagi-Sugeno with weights of rules	56	57	95

TABLE 9.55. Number of epochs required to train the system which is characterized by a definite error

HANG			
Structure	Value of error		
	0.028	0.026	0.024
Takagi-Sugeno with weights of rules	478	779	1132

TABLE 9.56. Number of epochs required to train the system which is characterized by a definite error

HANG			
Structure	Value of error		
	0.026	0.023	0.020
Takagi-Sugeno with weights of rules	20	40	79

TABLE 9.57. Number of epochs required to train the system which is characterized by a definite error

MODELING THE TASTE OF RICE			
Structure	Value of error		
	0.028	0.025	0.022
Takagi-Sugeno with weights of rules	8	35	67

9.5.2.4. Modeling the taste of rice

The smallest error for the Takagi-Sugeno structure with weights of rules was 0.0149 and was obtained in the 1620th epoch. Table 9.57 presents three desired values of error and the number of epochs, after which this error was obtained.

9.5.3 M3-type systems

By introducing to system (9.69) the weights specifying the importance of particular linguistic variables in each rule, we will obtain the following dependency:

$$\overline{y} = \frac{\sum_{r=1}^{N} \begin{pmatrix} w_r \left[T_{i=1}^{n} \left\{ 1 - w_{i,r} \left(1 - \mu_{A_i^r} \left(\overline{x}_i \right) \right) \right\} \right] \cdot \\ \cdot \left(c_0^{(r)} + c_1^{(r)} x_1 + \ldots + c_n^{(r)} x_n \right) \end{pmatrix}}{\sum_{r=1}^{N} w_r \left[T_{i=1}^{n} \left\{ 1 - w_{i,r} \left(1 - \mu_{A_i^r} \left(\overline{x}_i \right) \right) \right\} \right]} \tag{9.70}$$

$$= \frac{\sum_{r=1}^{N} w_r \left[T_{i=1}^{n} \left\{ \begin{array}{c} 1 - w_{i,r} \cdot \\ \cdot \left(1 - \exp\left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \right) \end{array} \right\} \cdot \right] \cdot \left(c_0^{(r)} + c_1^{(r)} x_1 + \ldots + c_n^{(r)} x_n \right)}{\sum_{r=1}^{N} w_r \left[T_{i=1}^{n} \left\{ 1 - w_{i,r} \left(\begin{array}{c} 1 - \exp \cdot \\ \cdot \left[-\left(\frac{\overline{x}_i - \overline{x}_i^r}{\sigma_i^r} \right)^2 \right] \end{array} \right) \right\} \right]}.$$

All the parameters of the neuro-fuzzy systems have been subject to learning using the error backpropagation method: centers and widths of Gaussian functions, weights of inputs and rules and function parameters $c_0^{(r)}, ..., c_n^{(r)}$, $r = 1, ..., N$.

9.5.3.1. Polymerization

The smallest error for the Takagi-Sugeno structure with weights of rules was 0.0030 and was obtained in the 4859th epoch. Table 9.58 presents three desired values of error and the number of epochs, after which this error was obtained.

9.5.3.2. HANG

The smallest error for the Takagi-Sugeno structure with weights of rules was 0.0116 and was obtained in the 2381st epoch. Table 9.59 presents three

TABLE 9.58. Number of epochs required to train the system which is characterized by a definite error

POLYMERIZATION			
Structure	Value of error		
	0.0055	0.0050	0.0045
Takagi-Sugeno with weights of inputs and rules	36	110	324

TABLE 9.59. Number of epochs required to train the system which is characterized by a definite error

HANG			
Structure	Value of error		
	0.028	0.026	0.024
Takagi-Sugeno with weights of inputs and rules	265	465	478

TABLE 9.60. Number of epochs required to train the system which is characterized by a definite error

NDP			
Structure	Value of error		
	0.026	0.023	0.020
Takagi-Sugeno with weights of inputs and rules	60	77	111

TABLE 9.61. Number of epochs required to train the system which is characterized by a definite error

MODELING THE TASTE OF RICE			
Structure	Value of error		
	0.028	0.025	0.022
Takagi-Sugeno with weights of inputs and rules	1	1	1

desired values of error and the number of epochs, after which this error was obtained.

9.5.3.4. Modeling the taste of rice

The smallest error for the Takagi-Sugeno structure with weights of rules was 0.0129 and was obtained in the 4008th epoch. Table 9.61 presents three desired values of error and the number of epochs, after which this error was obtained.

9.5.3.3. NDP

The smallest error for the Takagi-Sugeno structure with weights of rules was 0.0085 and was obtained in the 495th epoch. Table 9.60 presents three desired values of error and the number of epochs, after which this error was obtained.

9.6 Learning algorithms of neuro-fuzzy systems

In Subchapters 9.3, 9.4 and 9.5 we have discussed the neuro-fuzzy systems of the Takagi-Sugeno, Mamdani and logical type. In this subchapter, we will derive the learning algorithms of the above specified systems. They have been used in simulation examples (Subchapters 9.3-9.5).

We will use the idea of error backpropagation method, which is the basic learning method of neural networks. Learning of the neuro-fuzzy systems will come down to application of gradient algorithms, minimizing the appropriately formulated quality criterion. By $\overline{\mathbf{x}}(t) \in \mathbf{R}^n$ and $d(t) \in \mathbf{R}$ we will notate, a sequence of input and desired (at the output of the neuro-fuzzy system) signals, respectively. The problem of learning of those systems comes down to determining, based on the learning sequence

$$(\overline{\mathbf{x}}(1), d(1)), (\overline{\mathbf{x}}(2), d(2)), \ldots \tag{9.71}$$

all the parameters of the membership function and weights (weights describing the importance of rules and importance of particular linguistic variables in each rule) so as to minimize the criterion

$$Q(t) = \frac{1}{2}\left[f(\overline{\mathbf{x}}(t)) - d(t)\right]^2, \tag{9.72}$$

where

$$\overline{y} = f(\overline{\mathbf{x}}(t)) \tag{9.73}$$

is the output of the neuro-fuzzy systems of the Mamdani, logical and Takagi-Sugeno type presented in previous subchapters. For example, in the Mamdani and logical type systems, the parameter $\overline{y}^r$, $r = 1, \ldots, N$, may be determined using the gradient algorithm

$$\overline{y}^r(t+1) = \overline{y}^r(t) - \eta \frac{\partial Q(t)}{\partial \overline{y}^r(t)}. \tag{9.74}$$

The direct determination of gradient $\frac{\partial Q(t)}{\partial \overline{y}^r(t)}$ in the above procedure is complicated from a computational point of view. That is why an analogy between the neural networks and the neuro-fuzzy networks has been used, considering the fact that the latter also have a multilayer structure. Therefore, the error backpropagation method may be applied to learning of neuro-fuzzy networks. The notation used in this subchapter shall be explained on the example of a single neuron described by formula

$$y = f(s), \quad s = \sum_{i=0}^{n} x_i w_i, \tag{9.75}$$

where f is a sigmoidal function, x_i and w_i, $i = 0, \ldots, n$ are inputs and weights of the neuron. Let d be the desired signal at the neuron output. Then

$$\varepsilon^f = \varepsilon = y - d \tag{9.76}$$

is the error at neuron output and the expression

$$\varepsilon^s = \varepsilon^f\{s\} = \varepsilon^f \frac{\partial f(s)}{\partial s} = (y - d) f'(s) \tag{9.77}$$

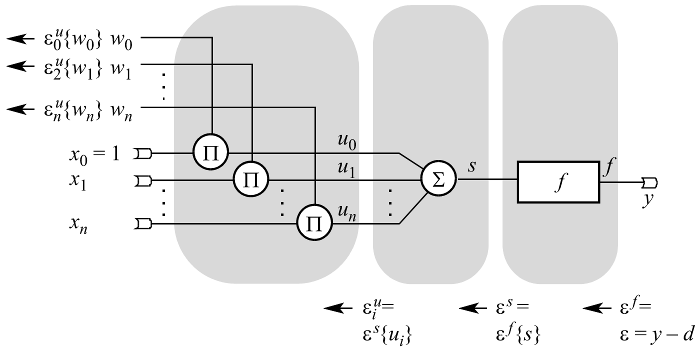

FIGURE 9.4. Flow of signals and errors in a single neuron

describes the error propagated from the functional block f to the summation block s. Figure 9.4 presents the flow of signals and errors in a single neuron.

At first, we will derive the learning algorithm for the Takagi-Sugeno system and next, for the Mamdani and logical type systems. We will modify the symbols used before to describe these systems, so that it would be possible to clearly present the flow of errors through particular blocks of the specified systems. The output signal of the Takagi-Sugeno system may be described as follows:

$$\overline{y} = \frac{\sum_{r=1}^{N}\left(\begin{array}{c} w_r^{\text{def}} \cdot T^*\left\{\begin{array}{c}\mu_{A_1^r}(\overline{x}_1), \mu_{A_2^r}(\overline{x}_2), \ldots, \mu_{A_n^r}(\overline{x}_n); \\ w_{1,r}^{\tau}, w_{2,r}^{\tau}, \ldots, w_{n,r}^{\tau}\end{array}\right\} \\ \cdot \left(c_{0,r}^f + \sum_{i=1}^{n} c_{i,r}^f \cdot \overline{x}_i\right)\end{array}\right)}{\sum_{r=1}^{N}\left(\begin{array}{c} w_r^{\text{def}} \cdot \\ \cdot\, T^*\left\{\begin{array}{c}\mu_{A_1^r}(\overline{x}_1), \mu_{A_2^r}(\overline{x}_2), \ldots, \mu_{A_n^r}(\overline{x}_n); \\ w_{1,r}^{\tau}, w_{2,r}^{\tau}, \ldots, w_{n,r}^{\tau}\end{array}\right\}\end{array}\right)}, \qquad (9.78)$$

where $w_{i,r}^{\tau} \in [0,1]$, $i = 1, 2, \ldots, n$, $r = 1, 2, \ldots, N$, mean the weights of antecedents of rules and $w_r^{\text{def}} \in [0,1]$, $r = 1, 2, \ldots, N$, mean the weights of rules. By substituting

$$T^*\left\{\begin{array}{c}\mu_{A_1^r}(\overline{x}_1), \mu_{A_2^r}(\overline{x}_2), \ldots, \mu_{A_n^r}(\overline{x}_n); \\ w_{1,r}^{\tau}, w_{2,r}^{\tau}, \ldots, w_{n,r}^{\tau}\end{array}\right\} = \tau_r(\overline{\mathbf{x}}) \qquad (9.79)$$

and

$$c_{0,r}^f + \sum_{i=1}^{n} c_{i,r}^f \cdot \overline{x}_i = f_r(\overline{\mathbf{x}}), \qquad (9.80)$$

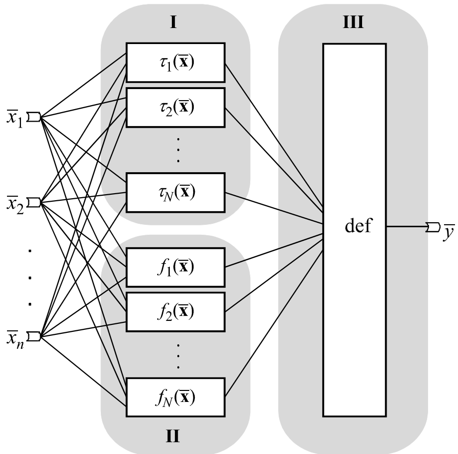

FIGURE 9.5. Network structure of the Takagi-Sugeno system

we get

$$\overline{y} = \frac{\sum_{r=1}^{N} w_r^{\mathrm{def}} \cdot \tau_r\left(\overline{\mathbf{x}}\right) \cdot f_r\left(\overline{\mathbf{x}}\right)}{\sum_{r=1}^{N} w_r^{\mathrm{def}} \cdot \tau_r\left(\overline{\mathbf{x}}\right)} = \mathrm{def}\left(\begin{array}{c} \tau_1\left(\overline{\mathbf{x}}\right), \ldots, \tau_N\left(\overline{\mathbf{x}}\right), \\ f_1\left(\overline{\mathbf{x}}\right), \ldots, f_N\left(\overline{\mathbf{x}}\right); \\ w_1^{\mathrm{def}}, \ldots, w_N^{\mathrm{def}} \end{array}\right). \tag{9.81}$$

The network structure of the Takagi-Sugeno system is presented in Fig. 9.5.

In the Takagi-Sugeno system, the following parameters are subject to learning:

- $p_{u,i,r}^{A}$, $u = 1, 2, \ldots, P^{A}$, parameters of input membership functions of the fuzzy sets,
- $c_{i,r}^{f}$, $i = 0, 1, \ldots, n$, $r = 1, 2, \ldots, N$, parameters of the functional blocks,
- $w_{i,r}^{\tau}$, $i = 1, 2, \ldots, n$, $r = 1, 2, \ldots, N$, weights of antecedents,
- w_r^{def}, $r = 1, 2, \ldots, N$, weights of rules.

The Takagi-Sugeno system parameters are modified by iteration according to the dependencies below:

$$p_{u,i,r}^{A}(t+1) = p_{u,i,r}^{A}(t) - \eta \Delta p_{u,i,r}^{A}(t), \tag{9.82}$$

$$w_{i,r}^{\tau}(t+1) = w_{i,r}^{\tau}(t) - \eta \Delta w_{i,r}^{\tau}(t), \tag{9.83}$$

$$c_{0,r}^{f}(t+1) = c_{0,r}^{f}(t) - \eta \Delta c_{0,r}^{f}(t), \tag{9.84}$$

$$c_{i,r}^{f}(t+1) = c_{i,r}^{f}(t) - \eta \Delta c_{i,r}^{f}(t), \quad i = 1, \ldots, n, \tag{9.85}$$

$$w_r^{\mathrm{def}}(t+1) = w_r^{\mathrm{def}}(t) - \eta \Delta w_r^{\mathrm{def}}(t). \tag{9.86}$$

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

$$\Delta p_{u,i,r}^{A}(t) = \varepsilon_r^{\tau}\left\{p_{u,i,r}^{A}\right\}, \tag{9.87}$$

$$\Delta w_{i,r}^{\tau}(t) = \varepsilon_r^{\tau}\left\{w_{i,r}^{\tau}\right\} \tag{9.88}$$

$$\Delta c_{0,r}^{f}(t) = \varepsilon_r^{f}\left\{c_{0,r}^{f}\right\}, \tag{9.89}$$

$$\Delta c_{i,r}^{f}(t) = \varepsilon_r^{f}\left\{c_{i,r}^{f}\right\}, \tag{9.90}$$

$$\Delta w_r^{\mathrm{def}}(t) = \varepsilon^{\mathrm{def}}\left\{w_r^{\mathrm{def}}\right\}. \tag{9.91}$$

The errors propagated by individual layers of the Takagi-Sugeno system are defined as follows (Fig. 9.6):

$$\varepsilon_r^{\tau} = \varepsilon^{\mathrm{def}}\left\{\tau_r(\overline{\mathbf{x}})\right\}, \tag{9.92}$$

$$\varepsilon_r^{f} = \varepsilon^{\mathrm{def}}\left\{f_r(\overline{\mathbf{x}})\right\}, \tag{9.93}$$

$$\varepsilon^{\mathrm{def}} = \varepsilon = \overline{y} - d. \tag{9.94}$$

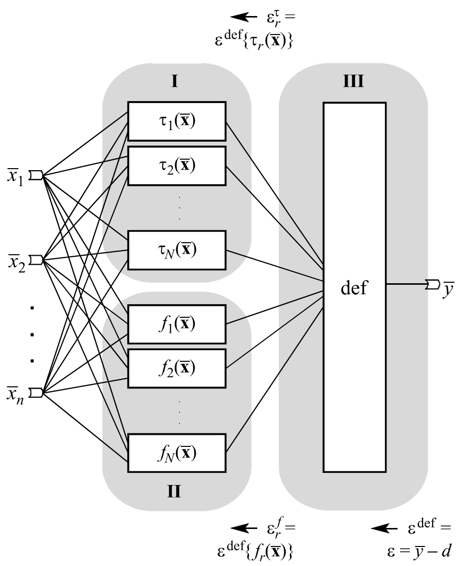

FIGURE 9.6. Flow of errors in the Takagi-Sugeno system

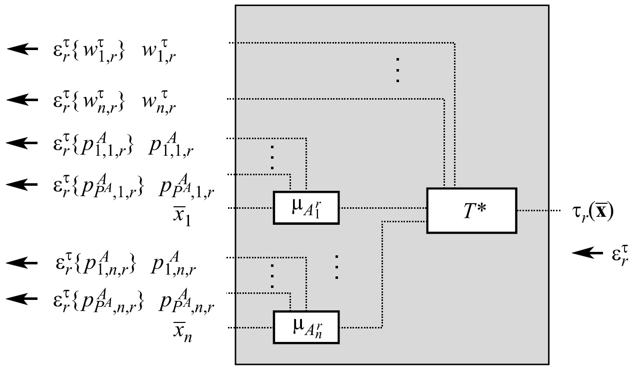

FIGURE 9.7. Block of rules activation of the Takagi-Sugeno system

The errors propagated by blocks of rules activation of the Takagi-Sugeno system are defined as follows (Fig. 9.7):

$$\varepsilon_r^\tau\left\{p_{u,i,r}^A\right\} = \varepsilon_r^\tau \frac{\partial T^*\begin{Bmatrix}\mu_{A_1^r}\left(\overline{x}_1\right), \mu_{A_2^r}\left(\overline{x}_2\right), \ldots, \mu_{A_n^r}\left(\overline{x}_n\right); \\ w_{1,r}^\tau, w_{2,r}^\tau, \ldots, w_{n,r}^\tau\end{Bmatrix}}{\partial \mu_{A_i^r}\left(\overline{x}_i\right)} \cdot \cdot \frac{\partial \mu_{A_i^r}\left(\overline{x}_i\right)}{\partial p_{u,i,r}^A}, \tag{9.95}$$

$$\varepsilon_r^\tau\left\{w_{i,r}^\tau\right\} = \varepsilon_r^\tau \frac{\partial T^*\begin{Bmatrix}\mu_{A_1^r}\left(\overline{x}_1\right), \mu_{A_2^r}\left(\overline{x}_2\right), \ldots, \mu_{A_n^r}\left(\overline{x}_n\right); \\ w_{1,r}^\tau, w_{2,r}^\tau, \ldots, w_{n,r}^\tau\end{Bmatrix}}{\partial w_{i,r}^\tau}. \tag{9.96}$$

We should notice that we are solving an optimization problem with constraints. That is why further in our considerations, we will apply the so-called *constraint function* $f_z(\cdot)$ given by dependency

$$f_z\left(x\right) = \frac{1}{1+\exp\left(-\left(p_1 x - p_2\right)\right)}, \tag{9.97}$$

while

$$\frac{\partial f_z\left(x\right)}{\partial x} = p_1\left(1 - f_z\left(x\right)\right) f_z\left(x\right). \tag{9.98}$$

In the simulations, it has been assumed that $p_1 = 10$ and $p_2 = 5$.

Example 9.1

We will show the method for the determination of partial derivatives in formula (9.95) and (9.96). Using the notation of the constraint function in the definition of weighted t-norm, we get

$$T^*\begin{Bmatrix}a_1, a_2, \ldots, a_n; \\ w_1, w_2, \ldots, w_n\end{Bmatrix} = T^*\left\{\mathbf{a}; \mathbf{w}\right\} = T_{i=1}^n\left\{1 - f_z\left(w_i\right)\left(1 - a_i\right)\right\}. \tag{9.99}$$

In case of an algebraic t-norm, we have

$$T^* \{\mathbf{a};\mathbf{w}\} = \prod_{i=1}^{n} (1 - f_z(w_i)(1-a_i)). \tag{9.100}$$

Then

$$\frac{\partial T^* \{\mathbf{a};\mathbf{w}\}}{\partial a_i} = f_z(w_i) \prod_{\substack{u=1 \\ u\neq i}}^{n} (1 - f_z(w_u)(1-a_u)) \tag{9.101}$$

and

$$\frac{\partial T^* \{\mathbf{a};\mathbf{w}\}}{\partial w_i} = -(1-a_i) \frac{\partial f_z(w_i)}{\partial w_i} \prod_{\substack{u=1 \\ u\neq i}}^{n} (1 - f_z(w_u)(1-a_u)). \tag{9.102}$$

Example 9.2
We will determine the partial derivatives of the Gaussian membership function of the input fuzzy set A (in order to have a clear notation, we will omit appropriate indexes)

$$\mu_A(x) = \exp\left(-\left(\frac{x-\overline{x}}{\sigma}\right)^2\right). \tag{9.103}$$

Let us notice that:

$$P^A = 2, \quad p_{1,i,r}^A = x, \quad p_{2,i,r}^A = \sigma. \tag{9.104}$$

Appropriate derivatives take the form

$$\frac{\partial \mu_A(x)}{\partial x} = -\mu_A(x) \frac{2(x-\overline{x})}{\sigma^2}, \tag{9.105}$$

$$\frac{\partial \mu_A(x)}{\partial \overline{x}} = \mu_A(x) \frac{2(x-\overline{x})}{\sigma^2}, \tag{9.106}$$

$$\frac{\partial \mu_A(x)}{\partial \sigma} = \mu_A(x) \frac{2(x-\overline{x})^2}{\sigma^3}. \tag{9.107}$$

The errors propagated by functional blocks of the Takagi-Sugeno system are determined as follows (Fig. 9.8):

$$\varepsilon_r^f \left\{c_{0,r}^f\right\} = \varepsilon_r^f, \tag{9.108}$$

$$\varepsilon_r^f \left\{c_{i,r}^f\right\} = \varepsilon_r^f \overline{x}_i. \tag{9.109}$$

The errors propagated by the defuzzification block of the Takagi-Sugeno system are determined as follows (Fig. 9.9):

$$\varepsilon^{\text{def}} \{\tau_r(\overline{\mathbf{x}})\} = \varepsilon^{\text{def}} \frac{\partial}{\partial \tau_r(\overline{\mathbf{x}})} \text{def} \begin{pmatrix} \tau_1(\overline{\mathbf{x}}), \ldots, \tau_N(\overline{\mathbf{x}}), \\ f_1(\overline{\mathbf{x}}), \ldots, f_N(\overline{\mathbf{x}}); \\ w_1^{\text{def}}, \ldots, w_N^{\text{def}} \end{pmatrix}, \tag{9.110}$$

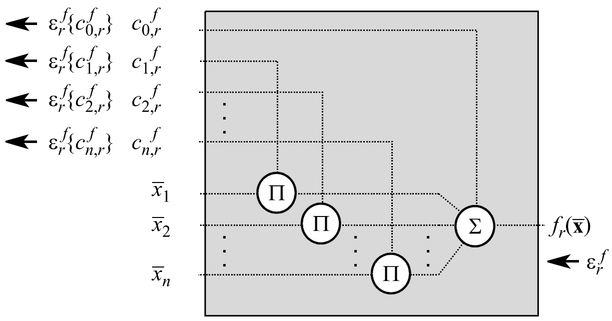

FIGURE 9.8. Functional block of the Takagi-Sugeno system

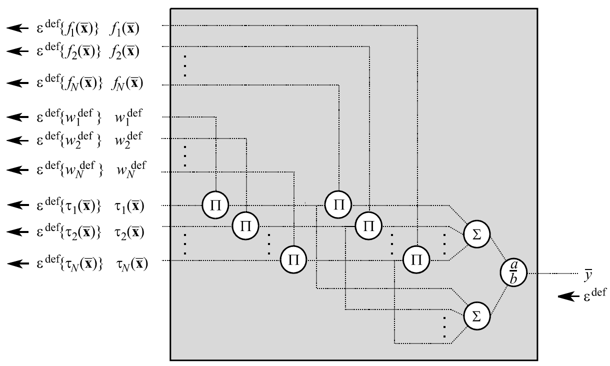

FIGURE 9.9. Defuzzification block of the Takagi-Sugeno system

$$\varepsilon^{\text{def}}\{f_r(\overline{\mathbf{x}})\} = \varepsilon^{\text{def}} \frac{\partial}{\partial f_r(\overline{\mathbf{x}})} \text{def} \begin{pmatrix} \tau_1(\overline{\mathbf{x}}), \ldots, \tau_N(\overline{\mathbf{x}}), \\ f_1(\overline{\mathbf{x}}), \ldots, f_N(\overline{\mathbf{x}}); \\ w_1^{\text{def}}, \ldots, w_N^{\text{def}} \end{pmatrix}, \tag{9.111}$$

$$\varepsilon^{\text{def}}\left\{w_r^{\text{def}}\right\} = \varepsilon^{\text{def}} \frac{\partial}{\partial w_r^{\text{def}}} \text{def} \begin{pmatrix} \tau_1(\overline{\mathbf{x}}), \ldots, \tau_N(\overline{\mathbf{x}}), \\ f_1(\overline{\mathbf{x}}), \ldots, f_N(\overline{\mathbf{x}}); \\ w_1^{\text{def}}, \ldots, w_N^{\text{def}} \end{pmatrix}, \tag{9.112}$$

where

$$\text{def} \begin{pmatrix} a_1, a_2, \ldots, a_n, \\ b_1, b_2, \ldots, b_n; \\ w_1, w_2, \ldots, w_n \end{pmatrix} = \text{def}(\mathbf{a}, \mathbf{b}; \mathbf{w}) = \frac{\sum_{i=1}^n w_i a_i b_i}{\sum_{i=1}^n w_i a_i}, \tag{9.113}$$

$$\frac{\partial \text{def}(\mathbf{a}, \mathbf{b}; \mathbf{w})}{\partial a_j} = (b_j - \text{def}(\mathbf{a}, \mathbf{b}; \mathbf{w})) \frac{w_j}{\sum_{i=1}^n w_i a_i}, \tag{9.114}$$

$$\frac{\partial \text{def}(\mathbf{a}, \mathbf{b}; \mathbf{w})}{\partial w_j} = (b_j - \text{def}(\mathbf{a}, \mathbf{b}; \mathbf{w})) \frac{a_j}{\sum_{i=1}^n w_i a_i}, \tag{9.115}$$

$$\frac{\partial \operatorname{def}(\mathbf{a}, \mathbf{b}; \mathbf{w})}{\partial b_j} = \frac{w_j a_j}{\sum_{i=1}^{n} w_i a_i}. \tag{9.116}$$

Now we will derive learning algorithms of the neuro-fuzzy systems of Mamdani and logical type. We will start our considerations with a generalized model which describes both types of systems. The output signal of such system may be described as follows:

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\sum_{r=1}^{N} \operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}. \tag{9.117}$$

The operation of operators $\operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)$, $r = 1, 2, \ldots, N$, depends on the type of inference applied in a given system, i.e.

$$\operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r) = \begin{cases} S^* \left\{ \begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\mathrm{agr}}, \ldots, w_N^{\mathrm{agr}} \end{array} \right\} \\ \text{for } \textit{Mamdani inference}, \\ \\ T^* \left\{ \begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\mathrm{agr}}, \ldots, w_N^{\mathrm{agr}} \end{array} \right\} \\ \text{for } \textit{logical inference}, \end{cases} \tag{9.118}$$

where

$$I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r) = \begin{cases} T\{\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\} \\ \text{for } \textit{Mamdani inference} \\ \\ I_{\text{fuzzy}}(\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)) \\ \text{for } \textit{logical inference} \end{cases} \tag{9.119}$$

and

$$I_{\text{fuzzy}}(\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)) = \begin{cases} S\{N(\tau_k(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r)\} \\ \text{for } S-\textit{implication}, \\ \\ t_{mul}^{-1}\left(\min\left\{1, \dfrac{t_{\text{mul}}(\mu_{B^k}(\overline{y}^r))}{t_{\text{mul}}(\tau_k(\overline{\mathbf{x}}))}\right\}\right) \\ \text{for } R-\textit{implication}, \\ \\ S\{N(\tau_k(\overline{\mathbf{x}})), T\{\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\}\} \\ \text{for } Q-\textit{implication}, \end{cases} \tag{9.120}$$

In formula (9.120), the definition of R-implication has been used, taking into consideration the multiplicative generators $t_{\text{mul}}(\cdot)$ of Archimedean t-norm. The rules activation operator $\tau_k(\overline{\mathbf{x}})$, $k = 1, 2, \ldots, N$, has been described similarly to the Takagi-Sugeno system considered earlier. Figure 9.10 presents the network structure of a generalized neuro-fuzzy system.

In the considered neuro-fuzzy system, the following parameters are subject to learning:

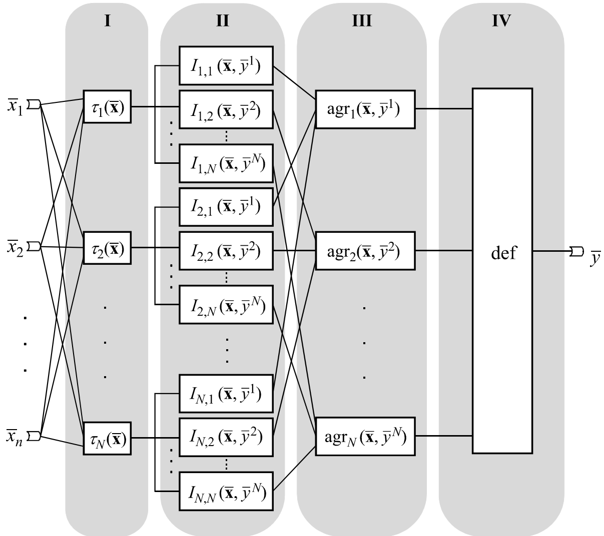

FIGURE 9.10. Network structure of the neuro-fuzzy system

- $p^A_{u,i,k}$, $u = 1, 2, \ldots, P^A$, $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, N$, parameters of input membership functions of the fuzzy sets,
- $p^B_{1,k} = \overline{y}^k$, $k = 1, 2, \ldots, N$, centers of membership functions of output fuzzy sets,
- $p^B_{u,k}$, $u = 2, 3, \ldots, P^B$, $k = 1, 2, \ldots, N$, other parameters of membership functions of output fuzzy sets,
- $w^\tau_{i,k}$, $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, N$, weights of antecedents,
- w^{agr}_k, $k = 1, 2, \ldots, N$, weights of rules.

The system parameters are modified by iteration according to the dependencies below:

$$p^A_{u,i,k}(t+1) = p^A_{u,i,k}(t) - \eta \Delta p^A_{u,i,k}(t), \tag{9.121}$$

$$w^\tau_{i,k}(t+1) = w^\tau_{i,k}(t) - \eta \Delta w^\tau_{i,k}(t), \tag{9.122}$$

$$p^B_{u,k}(t+1) = p^B_{u,k}(t) - \eta \Delta p^B_{u,k}(t), \quad u = 2, \ldots, P^B, \tag{9.123}$$

$$\overline{y}^r(t+1) = p^B_{1,r}(t+1) = \overline{y}^r(t) - \eta \Delta \overline{y}^r(t), \tag{9.124}$$

$$w^{\mathrm{agr}}_k(t+1) = w^{\mathrm{agr}}_k(t) - \eta \Delta w^{\mathrm{agr}}_k(t). \tag{9.125}$$

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

$$\Delta p^A_{u,i,k} = \varepsilon^\tau_k \left\{ p^A_{u,i,k} \right\}, \tag{9.126}$$

$$\Delta w^\tau_{i,k} = \varepsilon^\tau_k \left\{ w^\tau_{i,k} \right\}, \tag{9.127}$$

$$\Delta p^B_{u,k} = \sum_{r=1}^{N} \varepsilon^I_{k,r} \left\{ p^B_{u,k} \right\}, \quad u = 2, \ldots, P^B, \tag{9.128}$$

$$\Delta \overline{y}^r = \Delta p^B_{1,r} = \varepsilon^{\text{def}} \left\{ \overline{y}^r \right\} + \sum_{k=1}^{N} \varepsilon^I_{k,r} \left\{ \overline{y}^r \right\} + \sum_{k=1}^{N} \varepsilon^I_{r,k} \left\{ p^B_{1,r} \right\}, \tag{9.129}$$

$$\Delta w^{\text{agr}}_k = \sum_{r=1}^{N} \varepsilon^{\text{agr}}_r \left\{ w^{\text{agr}}_k \right\}. \tag{9.130}$$

The errors propagated by particular layers of the system are determined as follows (Fig. 9.11):

$$\varepsilon^\tau_k = \sum_{r=1}^{N} \varepsilon^I_{k,r} \left\{ \tau_k \left(\overline{\mathbf{x}} \right) \right\}, \tag{9.131}$$

$$\varepsilon^I_{k,r} = \varepsilon^{\text{agr}}_r \left\{ I_{k,r} \left(\overline{\mathbf{x}}, \overline{y}^r \right) \right\}, \tag{9.132}$$

$$\varepsilon^{\text{agr}}_r = \varepsilon^{\text{def}} \left\{ \text{agr}_r \left(\overline{\mathbf{x}}, \overline{y}^r \right) \right\}, \tag{9.133}$$

$$\varepsilon^{\text{def}} = \varepsilon = \overline{y} - d. \tag{9.134}$$

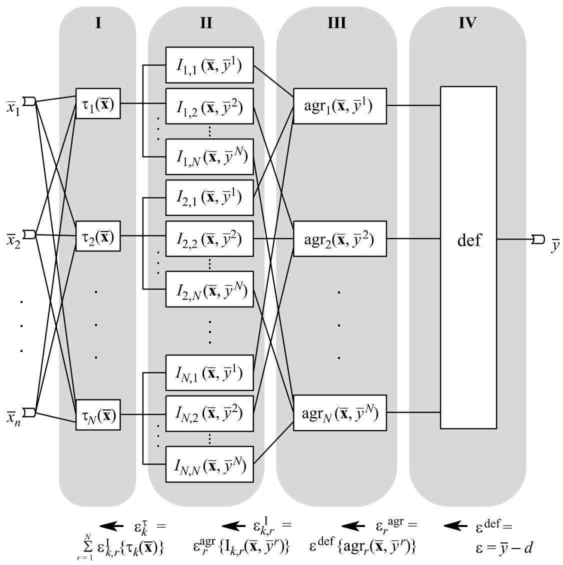

FIGURE 9.11. Flow of errors in the neuro-fuzzy system

The errors propagated by blocks of rules activation of the system are determined similarly as in Takagi-Sugeno system. The method of determination of errors propagated by implication blocks of the system depends of the chosen inference model (Mamdani or logical) as well as type of applied fuzzy implication (S, R, Q-implication) in case of logical inference.

The errors propagated by implication blocks of the system with Mamdani type inference are determined as follows (Fig. 9.12):

$$\varepsilon_{k,r}^{I}\left\{p_{u,k}^{B}\right\}=\varepsilon_{k,r}^{I}\frac{\partial T\left\{\tau_{k}\left(\overline{\mathbf{x}}\right),\mu_{B^{k}}\left(\overline{y}^{r}\right)\right\}}{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}\frac{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}{\partial p_{u,k}^{B}},\tag{9.135}$$

$$\varepsilon_{k,r}^{I}\left\{\overline{y}^{r}\right\}=\varepsilon_{k,r}^{I}\frac{\partial T\left\{\tau_{k}\left(\overline{\mathbf{x}}\right),\mu_{B^{k}}\left(\overline{y}^{r}\right)\right\}}{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}\frac{\partial\mu_{B^{k}}\left(\overline{y}^{r}\right)}{\partial\overline{y}^{r}},\tag{9.136}$$

$$\varepsilon_{k,r}^{I}\left\{\tau_{k}\left(\overline{\mathbf{x}}\right)\right\}=\varepsilon_{k,r}^{I}\frac{\partial T\left\{\tau_{k}\left(\overline{\mathbf{x}}\right),\mu_{B^{k}}\left(\overline{y}^{r}\right)\right\}}{\partial\tau_{k}\left(\overline{\mathbf{x}}\right)},\tag{9.137}$$

whereas the derivatives $\frac{\partial\mu_{B^{k}}(\overline{y}^{r})}{\partial p_{u,k}^{B}}$, $\frac{\partial\mu_{B^{k}}(\overline{y}^{r})}{\partial\overline{y}^{r}}$, $\frac{\partial T\left\{\tau_{k}(\overline{\mathbf{x}}),\mu_{B^{k}}(\overline{y}^{r})\right\}}{\partial\tau_{k}(\overline{\mathbf{x}})}$ and $\frac{\partial T\left\{\tau_{k}(\overline{\mathbf{x}}),\mu_{B^{k}}(\overline{y}^{r})\right\}}{\partial\mu_{B^{k}}(\overline{y}^{r})}$ are determined using the dependencies provided with the description of the learning method of the Takagi-Sugeno system.

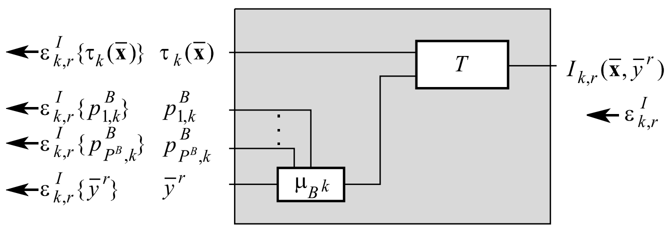

FIGURE 9.12. Implication block of the system with Mamdani type inference

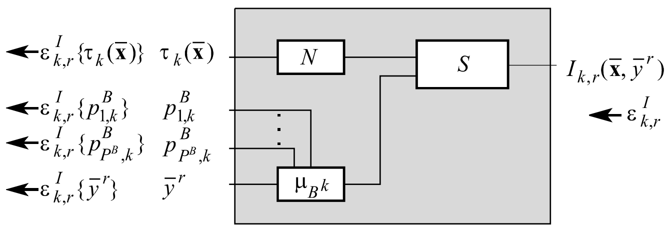

FIGURE 9.13. Implication block of the neuro-fuzzy system with inference of logical type (S-implication)

The errors propagated by implication blocks of the system with inference of logical type using the S-implication are determined as follows (Fig. 9.13):

$$\varepsilon_{k,r}^{I}\left\{p_{u,k}^{B}\right\} = \varepsilon_{k,r}^{I}\frac{\partial S\left\{N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right),\mu_{B^k}\left(\overline{y}^r\right)\right\}}{\partial\mu_{B^k}\left(\overline{y}^r\right)}\frac{\partial\mu_{B^k}\left(\overline{y}^r\right)}{\partial p_{u,k}^{B}}, \tag{9.138}$$

$$\varepsilon_{k,r}^{I}\left\{\overline{y}^r\right\} = \varepsilon_{k,r}^{I}\frac{\partial S\left\{N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right),\mu_{B^k}\left(\overline{y}^r\right)\right\}}{\partial\mu_{B^k}\left(\overline{y}^r\right)}\frac{\partial\mu_{B^k}\left(\overline{y}^r\right)}{\partial\overline{y}^r}, \tag{9.139}$$

$$\varepsilon_{k,r}^{I}\left\{\tau_k\left(\overline{\mathbf{x}}\right)\right\} = \varepsilon_{k,r}^{I}\frac{\partial S\left\{N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right),\mu_{B^k}\left(\overline{y}^r\right)\right\}}{\partial N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\frac{\partial N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}{\partial\tau_k\left(\overline{\mathbf{x}}\right)}, \tag{9.140}$$

while

$$N\left(a\right) = 1 - a, \tag{9.141}$$

$$\frac{\partial N\left(a\right)}{\partial a} = -1 \tag{9.142}$$

and the derivatives $\frac{\partial\mu_{B^k}\left(\overline{y}\right)}{\partial p_{u,k}^{B}}$ and $\frac{\partial\mu_{B^k}\left(\overline{y}^r\right)}{\partial\overline{y}^r}$ are determined using the dependencies provided with the description of the learning method of the Takagi-Sugeno system.

The method of determination of partial derivatives in formulas (9.138) - (9.140) will be shown in Example 9.3. This example relates to a more general case, taking into account any number of arguments and their weights in the definition of the t-conorm.

Example 9.3

Using the notation of the constraint function in the definition of weighted t-conorm, we get

$$S^*\left\{\begin{array}{c} a_1, a_2, \ldots, a_n; \\ w_1, w_2, \ldots, w_n \end{array}\right\} = S^*\left\{\mathbf{a};\mathbf{w}\right\} = \underset{i=1}{\overset{n}{S}}\left\{f_z\left(w_i\right)a_i\right\}. \tag{9.143}$$

In case of an algebraic t-conorm, we have

$$S^*\left\{\mathbf{a};\mathbf{w}\right\} = 1 - \prod_{i=1}^{n}\left(1 - f_z\left(w_i\right)a_i\right). \tag{9.144}$$

Then

$$\frac{\partial S^*\left\{\mathbf{a};\mathbf{w}\right\}}{\partial a_i} = f_z\left(w_i\right)\prod_{\substack{u=1\\u\neq i}}^{n}\left(1 - f_z\left(w_u\right)a_u\right) \tag{9.145}$$

and

$$\frac{\partial S^*\left\{\mathbf{a};\mathbf{w}\right\}}{\partial w_i} = a_i\frac{\partial f_z\left(w_i\right)}{\partial w_i}\prod_{\substack{u=1\\u\neq i}}^{n}\left(1 - f_z\left(w_u\right)a_u\right). \tag{9.146}$$

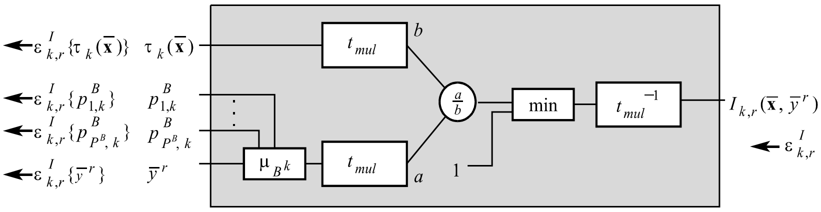

FIGURE 9.14. Implication block of the neuro-fuzzy system with inference of logical type (R-implication)

The errors propagated by implication blocks of the system with inference of logical type using the R-implication are determined as follows (Fig. 9.14):

$$\varepsilon_{k,r}^{I}\left\{p_{u,k}^{B}\right\}= = \varepsilon_{k,r}^{I}\left(\begin{array}{c}\dfrac{\partial t_{\text{mul}}^{-1}\left(\min\left\{1,\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\right\}\right)}{\partial\min\left\{1,\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\right\}} \\ \dfrac{\partial\min\left\{1,\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\right\}}{\partial\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}}\cdot\dfrac{\partial\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}}{\partial t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)} \\ \dfrac{\partial t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{\partial\mu_{B^k}\left(\overline{y}^r\right)}\dfrac{\partial\mu_{B^k}\left(\overline{y}^r\right)}{\partial p_{u,k}^{B}}\end{array}\right), \tag{9.147}$$

$$\varepsilon_{k,r}^{I}\left\{\overline{y}^r\right\}= = \varepsilon_{k,r}^{I}\left(\begin{array}{c}\dfrac{\partial t_{\text{mul}}^{-1}\left(\min\left\{1,\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\right\}\right)}{\partial\min\left\{1,\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\right\}} \\ \dfrac{\partial\min\left\{1,\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\right\}}{\partial\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}}\cdot\dfrac{\partial\dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}}{\partial t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)} \\ \dfrac{\partial t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{\partial\mu_{B^k}\left(\overline{y}^r\right)}\dfrac{\partial\mu_{B^k}\left(\overline{y}^r\right)}{\partial\overline{y}^r}\end{array}\right), \tag{9.148}$$

$$\varepsilon_{k,r}^{I}\left\{\tau_k\left(\overline{\mathbf{x}}\right)\right\} = \\ = \varepsilon_{k,r}^{I}\left(\begin{array}{c} \dfrac{\partial t_{mul}^{-1}\left(\min\left\{1, \dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\right\}\right)}{\partial \min\left\{1, \dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\right\}} \\ \dfrac{\partial \min\left\{1, \dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\right\}}{\partial \dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}} \cdot \dfrac{\partial \dfrac{t_{\text{mul}}\left(\mu_{B^k}\left(\overline{y}^r\right)\right)}{t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}}{\partial t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)} \\ \dfrac{\partial t_{\text{mul}}\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}{\partial \tau_k\left(\overline{\mathbf{x}}\right)} \end{array}\right). \tag{9.149}$$

In the above formulas, there are derivatives of the division operator and the minimum operator. The method of their determination has been provided in Subchapter 10.6 of the following chapter.

Example 9.4
To generate the Goguen R-implication, the following multiplicative generator of the t-norm may be used:

$$t_{\text{mul}}\left(a\right) = a^p, \quad p > 0. \tag{9.150}$$

Then in formulas (9.147) - (9.149) the following dependencies are used:

$$\frac{\partial t_{\text{mul}}\left(a\right)}{\partial a} = pa^{p-1}, \tag{9.151}$$

$$t_{\text{mul}}^{-1}\left(a\right) = a^{\frac{1}{p}}, \tag{9.152}$$

$$\frac{\partial t_{\text{mul}}^{-1}\left(a\right)}{\partial a} = \frac{1}{p}a^{\frac{1}{p}-1}. \tag{9.153}$$

The errors propagated by implication blocks of the system with inference of logical type using the Q-implication are determined as follows (Fig. 9.15):

$$\varepsilon_{k,r}^{I}\left\{p_{u,k}^{B}\right\} = \varepsilon_{k,r}^{I}\frac{\partial S\left\{N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right), T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}\right\}}{\partial T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}} \cdot \frac{\partial T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}}{\partial \mu_{B^k}\left(\overline{y}^r\right)}\frac{\partial \mu_{B^k}\left(\overline{y}^r\right)}{\partial p_{u,k}^{B}}, \tag{9.154}$$

$$\varepsilon_{k,r}^{I}\left\{\overline{y}^r\right\} = \varepsilon_{k,r}^{I}\frac{\partial S\left\{N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right), T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}\right\}}{\partial T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}} \cdot \frac{\partial T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}}{\partial \mu_{B^k}\left(\overline{y}^r\right)}\frac{\partial \mu_{B^k}\left(\overline{y}^r\right)}{\partial \overline{y}^r}, \tag{9.155}$$

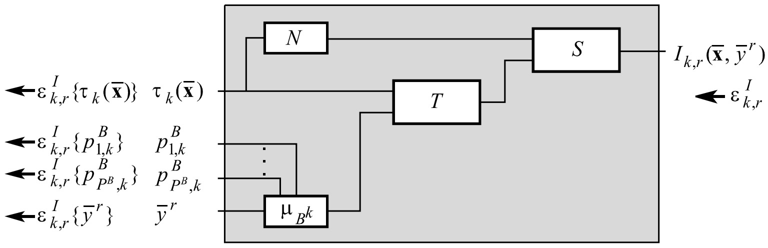

FIGURE 9.15. Implication block of the neuro-fuzzy system with inference of logical type (Q-implication)

$$\begin{aligned}&\varepsilon^I_{k,r}\left\{\tau_k\left(\overline{\mathbf{x}}\right)\right\}\\ &= \varepsilon^I_{k,r}\left(\begin{array}{c}\dfrac{\partial S\left\{N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right), T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}\right\}}{\partial N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}\\ \cdot\dfrac{\partial N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}{\partial \tau_k\left(\overline{\mathbf{x}}\right)} + \dfrac{\partial S\left\{N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right), T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}\right\}}{\partial T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}}\\ \cdot\dfrac{\partial T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}}{\partial \tau_k\left(\overline{\mathbf{x}}\right)}\end{array}\right), \quad (9.156)\end{aligned}$$

and the derivatives

$$\frac{\partial \mu_{B^k}\left(\overline{y}^r\right)}{\partial p^B_{u,k}}, \frac{\partial \mu_{B^k}\left(\overline{y}^r\right)}{\partial \overline{y}^r}, \frac{\partial T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}}{\partial \tau_k\left(\overline{\mathbf{x}}\right)}, \frac{\partial T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}}{\partial \mu_{B^k}\left(\overline{y}^r\right)},$$

$$\frac{\partial S\left\{N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right), T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}\right\}}{\partial N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}, \frac{\partial S\left\{N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right), T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}\right\}}{\partial T\left\{\tau_k\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(\overline{y}^r\right)\right\}}$$
$$\text{and} \frac{\partial N\left(\tau_k\left(\overline{\mathbf{x}}\right)\right)}{\partial \tau_k\left(\overline{\mathbf{x}}\right)}$$

are determined using the dependencies provided earlier and with the description of the learning method of the Takagi-Sugeno system.

Errors propagated by aggregation blocks of the system are determined depending on the chosen inference method. The errors propagated by aggregation blocks of the system with Mamdani type inference are determined as follows (Fig. 9.16):

$$\varepsilon^{\text{agr}}_r\left\{w^{\text{agr}}_r\right\} = \varepsilon^{\text{agr}}_r \frac{\partial S^*\left\{\begin{array}{c}I_{1,r}\left(\overline{\mathbf{x}}, \overline{y}^r\right), \ldots, I_{N,r}\left(\overline{\mathbf{x}}, \overline{y}^r\right);\\ w^{\text{agr}}_1, \ldots, w^{\text{agr}}_N\end{array}\right\}}{\partial w^{\text{agr}}_r}, \quad (9.157)$$

$$\varepsilon^{\text{agr}}_r\left\{I_{k,r}\left(\overline{\mathbf{x}}, \overline{y}^r\right)\right\} = \varepsilon^{\text{agr}}_r \frac{\partial S^*\left\{\begin{array}{c}I_{1,r}\left(\overline{\mathbf{x}}, \overline{y}^r\right), \ldots, I_{N,r}\left(\overline{\mathbf{x}}, \overline{y}^r\right);\\ w^{\text{agr}}_1, \ldots, w^{\text{agr}}_N\end{array}\right\}}{\partial I_{k,r}\left(\overline{\mathbf{x}}, \overline{y}^r\right)}, \quad (9.158)$$

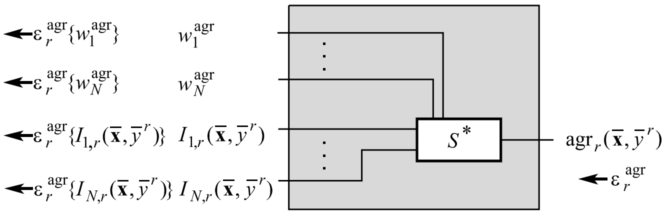

FIGURE 9.16. Aggregation block of the neuro-fuzzy system with Mamdani type inference

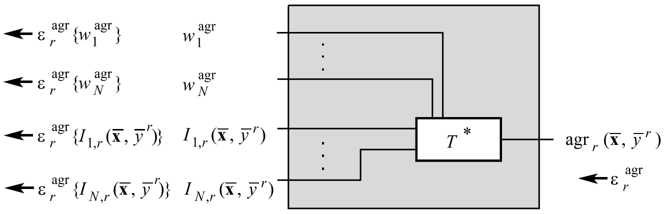

FIGURE 9.17. Aggregation block of the neuro-fuzzy system with inference of logical type

while the derivatives

$$\frac{\partial S^*\left\{\begin{array}{c} I_{1,r}(\bar{\mathbf{x}},\bar{y}^r),\ldots,I_{N,r}(\bar{\mathbf{x}},\bar{y}^r); \\ w_1^{\mathrm{agr}},\ldots,w_N^{\mathrm{agr}} \end{array}\right\}}{\partial I_{k,r}\left(\bar{\mathbf{x}},\bar{y}^r\right)} \quad \text{and} \quad \frac{\partial S^*\left\{\begin{array}{c} I_{1,r}(\bar{\mathbf{x}},\bar{y}^r),\ldots,I_{N,r}(\bar{\mathbf{x}},\bar{y}^r); \\ w_1^{\mathrm{agr}},\ldots,w_N^{\mathrm{agr}} \end{array}\right\}}{\partial w_k^{\mathrm{agr}}}$$

are determined based on the dependencies presented above.

The errors propagated by aggregation blocks of the system with inference of logical type are determined as follows (Fig. 9.17):

$$\varepsilon_r^{\mathrm{agr}}\left\{w_k^{\mathrm{agr}}\right\} = \varepsilon_r^{\mathrm{agr}} \frac{\partial T^*\left\{\begin{array}{c} I_{1,r}\left(\bar{\mathbf{x}},\bar{y}^r\right),\ldots,I_{N,r}\left(\bar{\mathbf{x}},\bar{y}^r\right); \\ w_1^{\mathrm{agr}},\ldots,w_N^{\mathrm{agr}} \end{array}\right\}}{\partial w_k^{\mathrm{agr}}}, \tag{9.159}$$

$$\varepsilon_r^{\mathrm{agr}}\left\{I_{k,r}\left(\bar{\mathbf{x}},\bar{y}^r\right)\right\} = \varepsilon_r^{\mathrm{agr}} \frac{\partial T^*\left\{\begin{array}{c} I_{1,r}\left(\bar{\mathbf{x}},\bar{y}^r\right),\ldots,I_{N,r}\left(\bar{\mathbf{x}},\bar{y}^r\right); \\ w_1^{\mathrm{agr}},\ldots,w_N^{\mathrm{agr}} \end{array}\right\}}{\partial I_{k,r}\left(\bar{\mathbf{x}},\bar{y}^r\right)}, \tag{9.160}$$

while the derivatives

$$\frac{\partial T^*\left\{\frac{I_{1,r}(\bar{\mathbf{x}},\bar{y}^r),\ldots,I_{N,r}(\bar{\mathbf{x}},\bar{y}^r);}{w_1^{\mathrm{agr}},\ldots,w_N^{\mathrm{agr}}}\right\}}{\partial I_{k,r}\left(\bar{\mathbf{x}},\bar{y}^r\right)} \quad \text{and} \quad \frac{\partial T^*\left\{\frac{I_{1,r}(\bar{\mathbf{x}},\bar{y}^r),\ldots,I_{N,r}(\bar{\mathbf{x}},\bar{y}^r);}{w_1^{\mathrm{agr}},\ldots,w_N^{\mathrm{agr}}}\right\}}{\partial w_k^{\mathrm{agr}}}$$

are determined using the dependencies provided with the description of the learning method of the Takagi-Sugeno system.

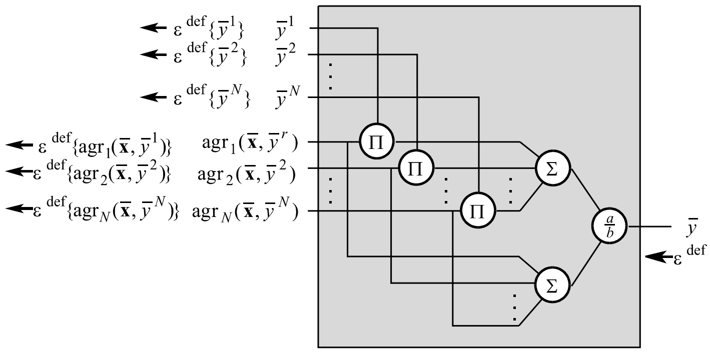

FIGURE 9.18. Defuzzification block of the neuro-fuzzy system

The errors propagated by the defuzzification block of the system are determined as follows (Fig. 9.18):

$$\varepsilon^{\text{def}}\left\{\overline{y}^r\right\} = \varepsilon^{\text{def}} \frac{\partial \operatorname{def}\left(\begin{array}{c}\operatorname{agr}_1\left(\overline{\mathbf{x}}, \overline{y}^1\right), \ldots, \operatorname{agr}_N\left(\overline{\mathbf{x}}, \overline{y}^N\right); \\ \overline{y}^1, \ldots, \overline{y}^N\end{array}\right)}{\partial \overline{y}^r}, \tag{9.161}$$

$$\varepsilon^{\text{def}}\left\{\operatorname{agr}_r\left(\overline{\mathbf{x}}, \overline{y}^r\right)\right\} = \varepsilon^{\text{def}} \frac{\partial \operatorname{def}\left(\begin{array}{c}\operatorname{agr}_1\left(\overline{\mathbf{x}}, \overline{y}^1\right), \ldots, \operatorname{agr}_N\left(\overline{\mathbf{x}}, \overline{y}^N\right); \\ \overline{y}^1, \ldots, \overline{y}^N\end{array}\right)}{\partial \operatorname{agr}_r\left(\overline{\mathbf{x}}, \overline{y}^r\right)}, \tag{9.162}$$

while

$$\operatorname{def}\left(a_1, a_2, \ldots, a_n; w_1, w_2, \ldots, w_n\right) = \operatorname{def}(\mathbf{a}; \mathbf{w}) = \frac{\sum_{i=1}^n w_i a_i}{\sum_{i=1}^n a_i}, \tag{9.163}$$

$$\frac{\partial \operatorname{def}(\mathbf{a}; \mathbf{w})}{\partial a_j} = \left(w_j - \operatorname{def}(\mathbf{a}; \mathbf{w})\right) \frac{1}{\sum_{i=1}^n a_i}, \tag{9.164}$$

$$\frac{\partial \operatorname{def}(\mathbf{a}; \mathbf{w})}{\partial w_j} = \left(a_j - \operatorname{def}(\mathbf{a}; \mathbf{w}) \frac{\partial a_j}{\partial w_j}\right) \frac{1}{\sum_{i=1}^n a_i}. \tag{9.165}$$

Let us notice that dependencies (9.163) - (9.165) are special cases of dependencies – (9.113) - (9.116).

9.7 Comparison of neuro-fuzzy systems

Simulation analyses in Subchapters 9.3 – 9.5 and an attempt to evaluate the studied neuro-fuzzy systems were based on the mean squared error as the criterion used to compare these systems. These considerations allow to conclude that usually the systems containing a higher number of trained parameters allowed to obtain better results. However, the desired neuro-fuzzy system should be characterized by the smallest possible error but at the same time should be as simple as possible. It should remember that systems with smaller number of trained parameters are characterized among others by better capabilities of generalization of the results obtained. Here, we should mention the so-called *parsimony principle* [235]. This principle is very useful when determining the appropriate order of the model. It may be formulated as follows: *from between two alternative and satisfactory models, we shall choose the one which contains less independent parameters*. This principle remains compliant with common sense: “do not enter any additional parameters into the process description unless they are necessary”.

Estimation methods of the system order have been best developed for autoregression processes [107, 132, 202]. Time series $u(n), u(n-1), ..,$ $u(n-p)$ is an autoregression process of order p, if the difference equation is satisfied

$$u(n) + \alpha_1 u(n-1) + ... + \alpha_p u(n-p) = e(n) \tag{9.166}$$

or equivalently

$$u(n) = -\sum_{k=1}^{p} \alpha_k u(n-k) + e(n), \tag{9.167}$$

where $\alpha_1, ..., \alpha_p$ are process coefficient, while $e(n)$ is the white noise

$$\mathbf{E}[e(n)] = 0, \quad \mathbf{E}[e(n)\,e(m)] = \begin{cases} \sigma^2 & \text{for} \quad n = m, \\ 0 & \text{for} \quad n \neq m. \end{cases} \tag{9.168}$$

In the autoregression theory, criteria allowing to estimate the order of predictor p, determining first the prediction error $\widehat{Q}_p$ based on the learning sequence of the length M, are well known. The most important is the Akaike information criterion (AIC), Schwarz method and the final prediction error (FPE) method.

In the following point, we will first present the basic models evaluation criteria (taking into account their complexity), initially applied to the estimation of orders of autoregression processes, and next they will be adapted to evaluate the effectiveness of neuro-fuzzy systems. By *the effectiveness of operation* of a neuro-fuzzy system, we shall understand the precision (accuracy) of operation achieved by such a system, (expressed by mean squared error or by the number of erroneously classified samples) in the context of its size. By *the system size* we shall understand the number of all parameters that are subject to learning. We shall also present the concept of the so-called *criteria isolines,* which allow to solve the problem of the compromise between the system accuracy and the number of parameters describing this system.

9.7.1 *Models evaluation criteria taking into account their complexity*

Two general criteria taking into account the complexity of the model, the dependencies between those criteria as well as their special forms are presented below.

9.7.1.1. Criterion A

The general form of criterion A, taking into account the complexity of the model, is given by formula

$$W(p) = \widehat{Q}_p \left[1 + \beta(M, p)\right], \tag{9.169}$$

where $\widehat{Q}_p$ is the mean square error, and $\beta(M, p)$ is the function of the length of the learning sequence M and the number of parameters p of the model. To eliminate too complex structures (according to the economy principle), we assume that

$$\lim_{p \to \infty} \beta(M, p) = \infty. \tag{9.170}$$

At the same time, in order to avoid the situation where the presence of the penalizing term in expression (9.169) hampers the observation of the decreasing of the mean square error $\widehat{Q}_p$ value with the increase of model complexity, we shall assume that

$$\lim_{M \to \infty} \beta(M, p) = 0. \tag{9.171}$$

The typical choice is $\beta(M, p) = 2p/M$ and then

$$W(p) = \widehat{Q}_p \left[1 + \frac{2p}{M}\right]. \tag{9.172}$$

9.7.1.2. Criterion B

An alternative criterion to formula (9.169) may be the following dependence:

$$W(p) = M \log \widehat{Q}_p + \gamma(M, p), \tag{9.173}$$

where the additional term $\gamma(M, p)$ should take into account the penalty for accepting models of an order which is too high. It is easy to check that if

$$\gamma(M, p) = M\beta(M, p), \tag{9.174}$$

then criteria (9.169) and (9.173) are asymptotically equivalent.

Below, we shall present the basic methods of the compromise selection of the model order. Most of these methods are the special cases of criterion A or B presented above.

9.7.1.3. Akaike information criterion (AIC) method

The assumption that $\gamma(M,p) = 2p$ in criterion (9.173) results in the so-called *Akaike Information Criterion*. The complexity of system p may be found by searching for the smallest value of the following expression

$$\mathrm{AIC}(p) = M \ln \widehat{Q}_p + 2p \tag{9.175}$$

9.7.1.4. Final prediction error (FPE) method

The FPE criterion was also proposed by Akaike. In *the Final Prediction Error* method, which does not result from any general formulas (9.169) and (9.173), the complexity of system p may be found by searching for the smallest value of the expression

$$\mathrm{FPE}(p) = \frac{M+p}{M-p}\widehat{Q}_p. \tag{9.176}$$

In expression (9.176) together with the increase of parameter p, the factor $\frac{M+p}{M-p}$ increases and the value of the mean square error $\widehat{Q}_p$ decreases. We shall notice that for high values of M, the following approximation may be used:

$$\mathrm{FPE}(p) = \widehat{Q}_p\left[1 + \frac{2p/M}{1-p/M}\right] \approx \widehat{Q}_p\left[1+\frac{2p}{M}\right], \tag{9.177}$$

which is of type (9.169), i.e.

$$\beta(M,p) = \frac{2p}{M}. \tag{9.178}$$

Expressions (9.169) and (9.173) are asymptotically equivalent, if condition (9.174) is satisfied, and hence

$$\gamma(M,p) = 2p. \tag{9.179}$$

In consequence:

$$\mathrm{FPE}(p) \approx \mathrm{AIC}(p) = M \ln \widehat{Q}_p + 2p. \tag{9.180}$$

FPE and AIC criteria show a tendency to select a model of a too small order. That is why literature [235] proposes three other methods described below:

9.7.1.5. Schwarz method

Assuming $\gamma(M,p) = p \log M$ in criterion (9.173) gives the so-called *Schwarz criterion*. In this method, the complexity of system p may be found by searching for the smallest value of the expression

$$S(p) = M \ln \widehat{Q}_p + p \ln M. \tag{9.181}$$

9.7.1.6. Södeström and Stoica method

Assuming $\gamma(M,p) = 2pc \log(\log M)$, where $c \geq 1$, in criterion (9.173) gives the so-called *Södeström and Stoica criterion*. In this method, the complexity of system p is found by searching for the smallest value of the expression

$$H(p) = M \ln \widehat{Q}_p + 2pc \log(\log M). \tag{9.182}$$

9.7.1.7. CAT method

In the CAT (*Criterion Autoregressive Transfer Function*) method, the complexity of system p may be found by searching for the smallest value of the expression

$$\mathrm{CAT}(p) = \frac{1}{M} \sum_{i=1}^{p} \frac{1}{\overline{Q}_i} - \frac{1}{\overline{Q}_p}, \tag{9.183}$$

where $\overline{Q}_i = \frac{m}{M-i} \widehat{Q}_i$.

The methods described above for determination of the order of the model have been first proposed for the analysis of data autoregression processes using formula (9.166). However, it should be stated that these methods allow to determine the appropriate order of the model regardless whether the system belongs to the class of the model structures or not [235].

9.7.2 Criteria isolines method

The estimation methods of the prediction order described in the previous point will be adapted now to the evaluation of fuzzy systems. Thanks to this, search for the desired fuzzy system based on two criteria (number of parameters and mean square error) will come down to one selected criterion, i.e. AIC, Schwarz or FPE. They have been adapted for the needs of evaluation of neuro-fuzzy systems in the following form:

$$\mathrm{AIC}(p, \widehat{Q}_p) = M \ln \widehat{Q}_p + 2p, \tag{9.184}$$

$$S(p, \widehat{Q}_p) = M \ln \widehat{Q}_p + p \ln M, \tag{9.185}$$

$$\mathrm{FPE}(p, \widehat{Q}_p) = \frac{Mn + p}{Mn - p} \widehat{Q}_p, \tag{9.186}$$

where p is the number of system parameters subject to learning (number of parameters of all membership functions and number of all weights if they occur in a given system), $\widehat{Q}_p$ is the measure of error used in simulations described in Subchapters 9.3 - 9.5, M is the number of samples in a learning sequence, and n is the number of system inputs. The product $M \cdot n$ may therefore be treated as a measure of size of the problem being solved. Tables 9.62a, 9.62b, 9.63a and 9.63b contain the computed values of criteria for particular tested structures in case of the learning and testing sequence used in the polymerization problem. Figures 9.19 - 9.24 illustrate the coordinates of the points corresponding to particular neuro-fuzzy systems tested. The coordinate p defines the number of parameters of a given system, coordinate Q defines the error with which the system realized the problem to be solved. The criteria isolines present constant values of the AIC, Schwarz and FPE criteria, with different values of the error and the number of parameters. Such an approach allows to solve the problem of the compromise between the system operation error and the number of parameters describing this system. Points located on the criteria isolines with the same values of AIC, Schwarz or FPE criterion characterize the neuro-fuzzy systems making up the Pareto set. In the Pareto set, none of the two values of contradictory criteria may be improved (mean square error *versus* system size), without worsening the other one. Points located on the criteria isolines with the smallest values of AIC, Schwarz or FPE criterion characterize the neuro-fuzzy systems which have been called suboptimal ones. The suboptimal neuro-fuzzy systems presented in graphs ensure the smallest value of criteria within tested structures (the terminology "optimum systems" is not used as all possible structures have not been tested).

Tables 9.62a, 9.62b, 9.63a and 9.63b and figures indicate that both for the learning sequence and for the testing sequence, the AIC criterion evaluates as the best system 1 (simplified Larsen structure), and next, system 29 (Zadeh structure with weights of rules), the FPE criterion – system 29, the Schwarz criterion – definitely system 1.

Analogically, the criteria isolines may be easily drawn for HANG, NDP and modeling the taste of rice problems. Having drawn these lines, it may be checked that in case of the HANG problem, both for the learning and the testing sequence, the AIC and FPE criteria indicate system 23 (Reichenbach structure with weights of rules), and the Schwarz criterion – system 1 (simplified Larsen structure). In case of the NDP problem all three criteria, both for the learning and the testing sequence indicate the selection of system 3 (simplified Larsen structure with weights of inputs and rules). In case of modeling the taste of rice, both for the learning and the testing sequence, the AIC and Schwarz criteria indicate system 1 (simplified Larsen structure) as the best one, and the FPE criterion indicates system 20 (Mamdani structure with weight of rules).

TABLE 9.62a. Values of criteria for the learning sequence

No.	Structure	Polymerization				
		error	p	AIC	FPE	Schwarz
1	Larsen simplified	0.0042	42	−299.09	0.0063	−204.65
2	Larsen simplified with weights of rules	0.0039	48	−292.27	0.0062	−184.35
3	Larsen simplified with weights of inputs and rules	0.0031	66	−272.34	0.0059	−123.94
4	Łukasiewicz simplified	0.0059	42	−275.30	0.0089	−180.86
5	Łukasiewicz simplified with weights of rules	0.0039	48	−292.27	0.0062	−184.35
6	Łukasiewicz simplified with weights of inputs and rules	0.0037	66	−259.96	0.0071	−111.56
7	Zadeh simplified	0.0049	42	−288.30	0.0074	−193.86
8	Zadeh simplified with weights of rules	0.0041	48	−288.77	0.0065	−180.85
9	Zadeh simplified with weights of inputs and rules	0.0038	66	−258.09	0.0073	−109.69
10	Binary	0.0063	48	−258.70	0.01	−150.78
11	Binary with weights of rules	0.0054	54	−257.49	0.0091	−136.08
12	Binary with weights of inputs and rules	0.0036	72	−249.88	0.0074	−87.99
13	Larsen	0.0049	48	−276.30	0.0078	−168.37
14	Larsen with weights of rules	0.0043	54	−273.44	0.0073	−152.02
15	Larsen with weights of inputs and rules	0.0035	72	−251.85	0.0072	−89.96
16	Łukasiewicz	0.0065	48	−256.52	0.0104	−148.59
17	Łukasiewicz with weights of rules	0.0041	54	−276.77	0.0069	−155.36
18	Łukasiewicz with weights of inputs and rules	0.0038	72	−246.09	0.0078	−84.20
19	Mamdani	0.0041	48	−288.77	0.0065	−180.85
20	Mamdani with weights of rules	0.0039	54	−280.27	0.0066	−158.86
21	Mamdani with weights of inputs and rules	0.0034	72	−253.88	0.0069	−91.99
22	Reichenbach	0.0040	48	−290.50	0.0064	−182.57
23	Reichenbach with weights of rules	0.0037	54	−283.96	0.0063	−162.54

TABLE 9.62b. Values of criteria for the learning sequence

No.	Structure	Polymerization				
		error	p	AIC	FPE	Schwarz
24	Reichenbach with weights of inputs and rules	0.0034	72	−253.88	0.0069	−91.990
25	Willmott	0.0056	48	−266.95	0.0089	−159.02
26	Willmott with weights of rules	0.0047	54	−267.21	0.008	−145.79
27	Willmott with weights of inputs and rules	0.0039	72	−244.27	0.008	−82.38
28	Zadeh	0.0038	48	−294.09	0.0061	−186.17
29	Zadeh with weights of rules	0.0030	54	−298.64	0.0051	−177.22
30	Zadeh with weights of inputs and rules	0.0028	72	−267.47	0.0057	−105.58
31	Takagi-Sugeno	0.0034	60	−277.88	0.0061	−142.97
32	Takagi-Sugeno with weights of rules	0.0031	66	−272.34	0.0059	−123.94
33	Takagi-Sugeno with weights of inputs and rules	0.0030	84	−238.64	0.007	−49.77

In all simulations performed so far, the effectiveness of operation of the neuro-fuzzy systems, assuming the defined number of rules to solve a specific problem has been analised. Thanks to it, it was possible to compare 33 different neuro-fuzzy systems. It should be stressed that the method of criteria isolines may be also applied to the appropriate designing of each of these systems. If we concentrate on a single specific neuro-fuzzy system, we may select the number of rules which will ensure the smallest value of one of the criteria listed in point 9.7.1. For example, for the problem of modeling the taste of rice, we have applied the simplified Larsen structure, changing gradually the number of rules from 10 to 2. Individual systems are characterized by the following number of parameters: 110, 99, 88, 77, 66, 55, 44, 33 and 22. Figures 9.25 and 9.26 show the function of dependency of the AIC and Schwarz criteria versus the number of parameters. As it may be inferred from the graphs, the AIC and Schwarz criteria suggest that four rules should be assumed. In the simulations performed, the problem of modeling the taste of rice was analyzed, assuming 5 rules, according to the principle of caution.

TABLE 9.63a. Value of criteria for the testing sequence

No.	Structure	Polymerization				
		error	p	AIC	FPE	Schwarz
1	Larsen simplified	0.0045	42	−294.26	0.0068	−199.82
2	Larsen simplified with weights of rules	0.0041	48	−288.77	0.0065	−180.85
3	Larsen simplified with weights of inputs and rules	0.0033	66	−267.97	0.0063	−119.57
4	Łukasiewicz simplified	0.0063	42	−270.70	0.0095	−176.27
5	Łukasiewicz simplified with weights of rules	0.0041	48	−288.77	0.0065	−180.85
6	Łukasiewicz simplified with weights of inputs and rules	0.0039	66	−256.27	0.0075	−107.87
7	Zadeh simplified	0.0053	42	−282.80	0.0080	−188.37
8	Zadeh simplified with weights of rules	0.0044	48	−283.83	0.0070	−175.90
9	Zadeh simplified with weights of inputs and rules	0.0040	66	−254.50	0.0077	−106.10
10	Binary	0.0067	48	−254.40	0.0107	−146.47
11	Binary with weights of rules	0.0057	54	−253.71	0.0096	−132.29
12	Binary with weights of inputs and rules	0.0038	72	−246.09	0.0078	−84.20
13	Larsen	0.0052	48	−272.14	0.0083	−164.21
14	Larsen with weights of rules	0.0045	54	−270.26	0.0076	−148.84
15	Larsen with weights of inputs and rules	0.0038	72	−246.09	0.0078	−84.20
16	Łukasiewicz	0.0069	48	−252.34	0.0110	−144.41
17	Łukasiewicz with weights of rules	0.0045	54	−270.26	0.0076	−148.84
18	Łukasiewicz with weights of inputs and rules	0.0041	72	−240.77	0.0084	−78.88
19	Mamdani	0.0043	48	−285.44	0.0068	−177.51
20	Mamdani with weights of rules	0.0042	54	−275.09	0.0071	−153.67
21	Mamdani with weights of inputs and rules	0.0037	72	−247.96	0.0076	−86.07
22	Reichenbach	0.0044	48	−283.83	0.0070	−175.90

TABLE 9.63b. Value of criteria for the testing sequence

No.	Structure	Polymerization				
		error	p	AIC	FPE	Schwarz
23	Reichenbach with weights of rules	0.0039	54	−280.27	0.0066	−158.86
24	Reichenbach with weights of inputs and rules	0.0037	72	−247.96	0.0076	−86.07
25	Willmott	0.0060	48	−262.12	0.0096	−154.19
26	Willmott with weights of rules	0.0049	54	−264.30	0.0083	−142.88
27	Willmott with weights of inputs and rules	0.0043	72	−237.44	0.0088	−75.55
28	Zadeh	0.0043	48	−285.44	0.0068	−177.51
29	Zadeh with weights of rules	0.0033	54	−291.97	0.0056	−170.55
30	Zadeh with weights of inputs and rules	0.0031	72	−260.34	0.0063	−98.45
31	Takagi-Sugeno	0.0036	60	−273.88	0.0065	−138.97
32	Takagi-Sugeno with weights of rules	0.0034	66	−265.88	0.0065	−117.48
33	Takagi-Sugeno with weights of inputs and rules	0.0033	84	−231.97	0.0077	−43.09

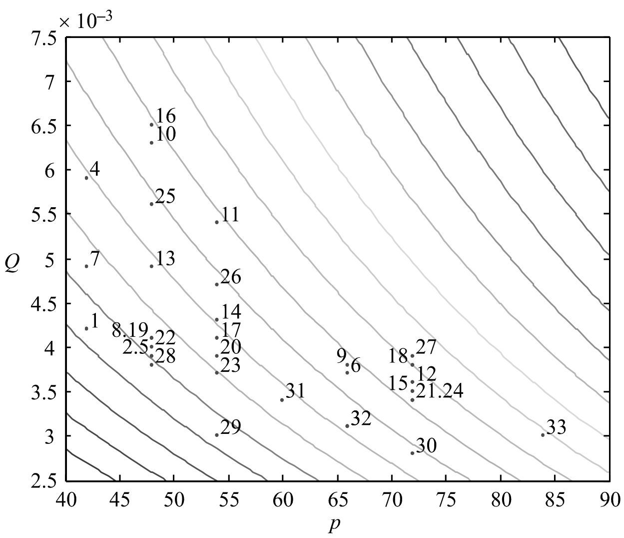

FIGURE 9.19. Criteria isolines: results obtained by particular systems for the Akaike criterion for the learning sequence – polymerization problem

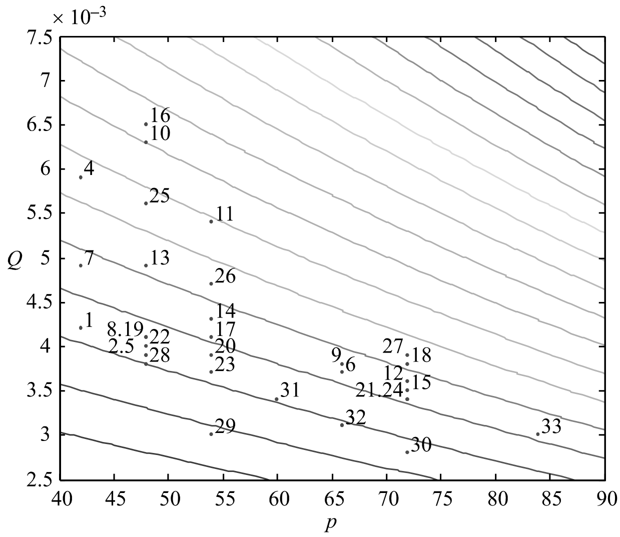

FIGURE 9.20. Criteria isolines: results obtained by particular systems for the FPE criterion for the learning sequence – polymerization problem

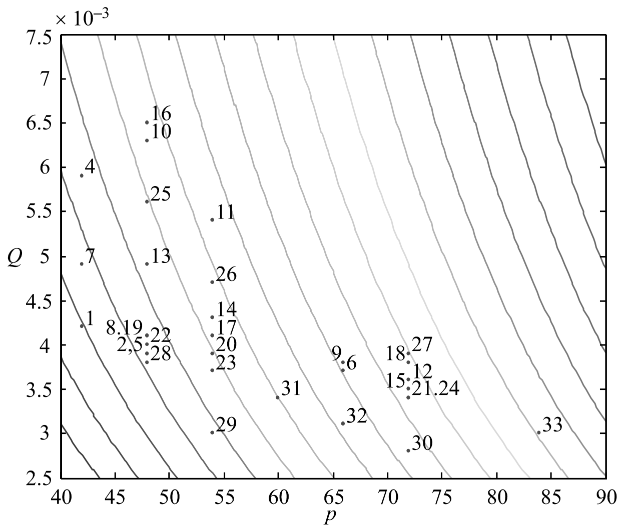

FIGURE 9.21. Criteria isolines: results obtained by particular systems for the Schwarz criterion for the learning sequence – polymerization problem

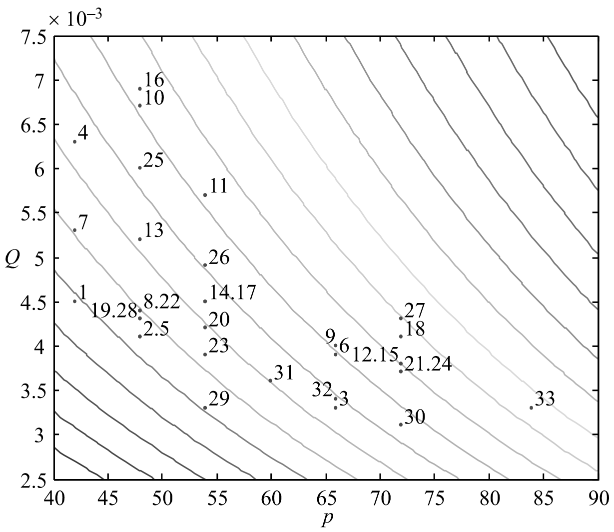

FIGURE 9.22. Criteria isolines: results obtained by particular systems for the Akaike criterion for the testing sequence – polymerization problem

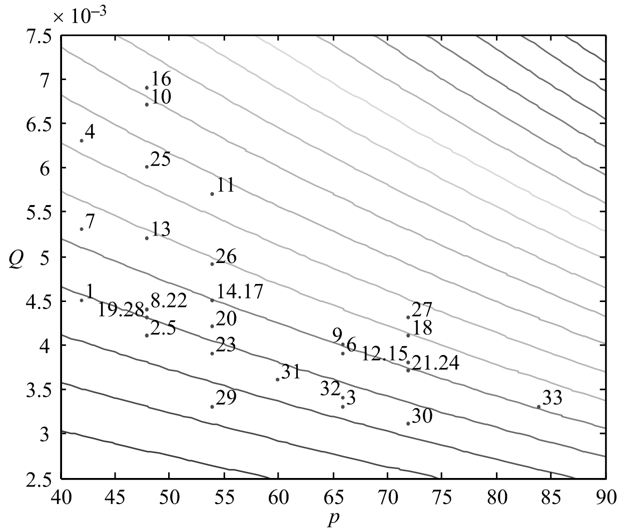

FIGURE 9.23. Criteria isolines: results obtained by particular systems for the FPE criterion for the testing sequence – polymerization problem

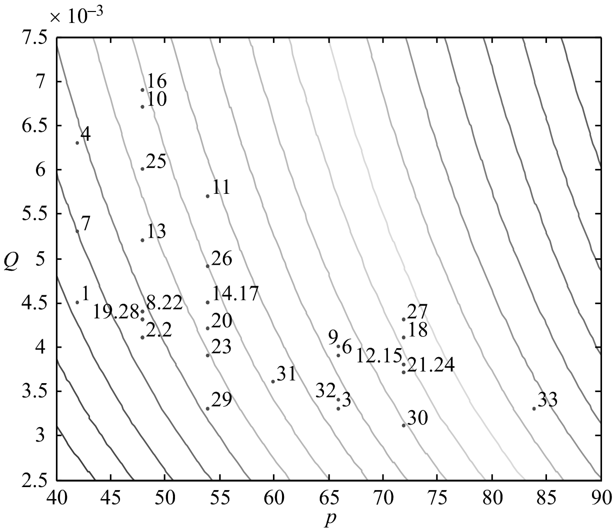

FIGURE 9.24. Criteria isolines: results obtained by particular systems for the Schwarz criterion for the testing sequence – polymerization problem

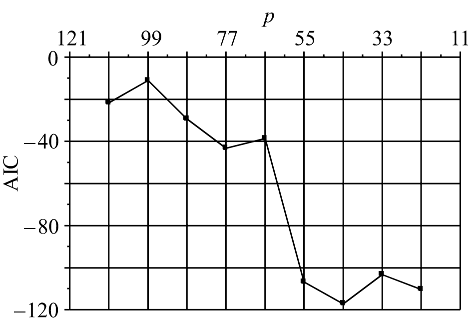

FIGURE 9.25. Values of the Akaike criterion

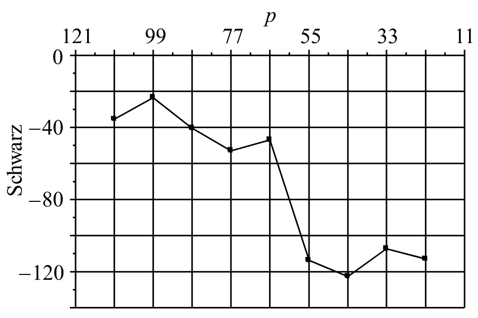

FIGURE 9.26. Values of the Schwarz criterion

9.8 Notes

In this chapter, the object of studies included the neuro-fuzzy systems of the Mamdani, logical and Takagi-Sugeno type. From the simulations performed we may conclude that if weights reflecting the importance of rules and importance of linguistic variables in the antecedents of rules are included, it significantly improves the operation of neuro-fuzzy systems. The Takagi-Sugeno systems are characterized by the smallest mean square error, but this result is obtained with a large number of parameters. Extended structures (characterized by a more extensive information on membership functions of the fuzzy sets in the consequents of rules) give better results than the simplified structures. Moreover, the issue of compromise between the system operation error and the number of parameters describing it has been presented in this chapter. From the analysis of criteria isolines corresponding to particular simulations we may conclude that in most cases the best system, in the meaning of proposed criteria, is the simplified Larsen structure given by formula (9.25). The logical type systems have been studied in monographs by Czogała and Łęski [34], Rutkowska [187] as well as Rutkowski [225]. Different approaches to the issue of designing neuro-fuzzy networks have been presented in works [65, 126, 142, 145, 148, 149, 176, 185, 186, 213, 214, 216, 239, 253, 254]. Neuro-fuzzy structures associated with the rough sets theory have been proposed by Nowicki [151, 152], while in association with the type-2 fuzzy sets theory have been proposed by Starczewski [238]. Relational neuro-fuzzy systems have been analyzed by Scherer [231]. The learning method of neuro-fuzzy structures has been developed by Piliński [172 - 174]. Models evaluation criteria taking into account their complexity have been discussed in detail in monograph [235].