МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО АТОМНОЙ ЭНЕРГИИ

РОССИЙСКАЯ АКАДЕМИЯ НАУК

РОССИЙСКАЯ АССОЦИАЦИЯ НЕЙРОИНФОРМАТИКИ

МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)

НАУЧНАЯ СЕССИЯ МИФИ–2005

НЕЙРОИНФОРМАТИКА–2005

**VII ВСЕРОССИЙСКАЯ
НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ**

ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ

По материалам Школы-семинара
«Современные проблемы нейроинформатики»

Москва 2005

УДК 001 (06)+004.032.26 (06)
ББК 72я5+32.818я5
М82

НАУЧНАЯ СЕССИЯ МИФИ–2005. VII ВСЕРОССИЙСКАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «НЕЙРОИНФОРМАТИКА–2005»: ЛЕКЦИИ ПО НЕЙРОИНФОРМАТИКЕ. – М.: МИФИ, 2005. – 214 с.

В книге публикуются тексты лекций, прочитанных на Школе-семинаре «Современные проблемы нейроинформатики», проходившей 26–28 января 2005 года в МИФИ в рамках VII Всероссийской конференции «Нейроинформатика–2005».

Материалы лекций связаны с рядом проблем, актуальных для современного этапа развития нейроинформатики, включая ее взаимодействие с другими научно-техническими областями.

Ответственный редактор
Ю. В. Тюменцев, кандидат технических наук

ISBN 5–7262–0526–X

Содержание

***L. Rutkowski*. Neuro-fuzzy inference systems** **136**
Introduction 137
Non-flexible neuro-fuzzy systems 137
Description of fuzzy systems 137
Mamdani-type neuro-fuzzy systems 141
Logical-type neuro-fuzzy systems 141
Generalized neuro-fuzzy systems 142
Flexibility in fuzzy systems 142
Compromise operators 142
Weighted triangular norms 143
Parameterized triangular norms 144
Soft fuzzy norms 145
Flexible neuro-fuzzy systems 146
Compromise neuro-fuzzy systems 152
Learning procedures 153
Simulation results 154
Final remarks: Design of flexible neuro-fuzzy systems 163
References 163

Л. РУТКОВСКИЙ
Кафедра вычислительной техники
Ченстоховского технического университета,
Ченстохова, Польша
E-mail: lrutko@kik.pcz.czest.pl

НЕЙРО-НЕЧЕТКИЕ СИСТЕМЫ ВЫВОДА

Аннотация

В Лекции рассматривается новый класс нейро-нечетких систем вывода, основанных на логическом подходе и на подходе, предложенном Мамдани. Рассматриваются средства обеспечения гибкости при проектировании нейро-нечетких систем. Для предлагаемого подхода характерно автоматическое формирование правил нечеткого вывода в процессе обучения системы. Высокая точность нейро-нечетких систем демонстрируется на примере задачи моделирования динамического объекта.

LESZEK RUTKOWSKI
Department of Computer Engineering,
Technical University of Czestochowa
Czestochowa, Poland
E-mail: lrutko@kik.pcz.czest.pl

NEURO-FUZZY INFERENCE SYSTEMS

Abstract

In this paper we present a new class of neuro-fuzzy inference systems based on Mamdani and logical approaches. We introduce several flexibility concepts in the design of neuro-fuzzy systems. Our approach is characterized by automatic determination of fuzzy inference in the process of learning. A high accuracy of neuro-fuzzy systems is demonstrated in simulations of a dynamic plant.

УДК 004.032.26 (06) Нейронные сети

Introduction

In the literature various neuro-fuzzy inference systems (NFIS) have been proposed (see e.g. [1–5], [16–20). They have been used in a wide range of problems, e.g. system identification, pattern classification, system control, prediction and image processing. It is well known that traditional fuzzy systems suffer from the lack of learning properties. On the other hand neural networks are not able to incorporate a linguistic information coming from human experts. Neuro-fuzzy systems presented by several authors (see e.g. [7–12]) exhibit advantages of neural networks and fuzzy systems, i.e. they combine the natural language description of fuzzy systems and the learning properties of neural networks. In this paper we develop a new class of neuro-fuzzy systems. Our approach is characterized by automatic determination of fuzzy inference in the process of learning. It is well known that introducing additional parameters to be tuned in neuro-fuzzy systems improves their performance and they are able to better represent the patterns encoding in data. Therefore, in this papare we introduce several flexibility concepts in the design of neuro-fuzzy systems. Due to additional parameters incorporated into a neuro-fuzzy system, we achieve an excellent performance of neuro-fuzzy systems. A high accuracy of neuro-fuzzy systems is demonstrated in simulations of a dynamic plant. In the paper by T, S, and N we denote t-norm, t-conorm and negation, respectively.

Non-flexible neuro-fuzzy systems

Description of fuzzy systems

In this paper, we consider multi-input-single-output fuzzy system mapping $X \to Y$, where $X \subset R^n$ and $Y \subset R$. The system (see Fig. 1) is composed of a fuzzifier, a fuzzy rule base, a fuzzy inference engine and a defuzzifier. The fuzzifier performs a mapping from the observed crisp input to a fuzzy set. The most commonly used fuzzifier is the singleton fuzzifier which maps $\overline{\mathbf{x}} = [\overline{x}_1, \ldots, \overline{x}_n] \in X$ into a fuzzy set $A' \subseteq X$ characterized by the membership function

$$\mu_{A'}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \overline{\mathrm{x}} \\ 0 & \text{if } \mathbf{x} \neq \overline{\mathrm{x}} \end{cases} . \tag{1}$$

The fuzzy rule base consists of a collection of N fuzzy IF-THEN rules, aggregated by the disjunction or the conjunction, in the form

$$R^{(k)} : \begin{cases} \text{IF} & x_1 & \text{is} & A_1^k & \text{AND} \\ & x_2 & \text{is} & A_2^k & \text{AND} & \ldots \\ & x_n & \text{is} & A_n^k \\ \text{THEN} & y & \text{is} & B^k \end{cases} \tag{2}$$

or

$$R^{(k)} : \text{IF} \quad \mathbf{x} \quad \text{is} \quad A^k \quad \text{THEN} \quad y \quad \text{is} \quad \text{B}^k, \tag{3}$$

where $\mathbf{x} = [x_1, \ldots, x_n] \in X$, $y \in Y$, $A^k = A_1^k \times A_2^k \times \ldots \times A_n^k$, $A_1^k, A_2^k, \ldots, A_n^k$ are fuzzy sets characterized by membership functions

$$\mu_{A_1^k}(x_i), \; i = 1, \ldots, n; \; k = 1, \ldots, N,$$

whereas B^k are fuzzy sets characterized by membership functions $\mu_{B^k}(y), k = 1, \ldots, N$. The firing strength of the $k-th$ rule, $k = 1, \ldots, N$, is defined by

$$\tau_k(\overline{\mathbf{x}}) = \mathop{T}_{i=1}^{n} \left\{ \mu_{A_1^k}(\overline{x}_i) \right\} = \mu_{A^k}(\overline{\mathbf{x}}). \tag{4}$$

In the paper notations τ_k and $\mu_{A^k}(\overline{\mathbf{x}})$ will be used interchangeably.

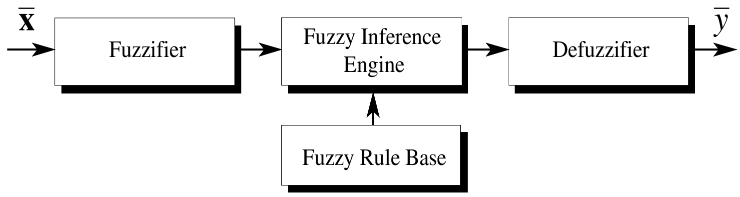

Figure 1. Fuzzy inference system

The fuzzy inference engine determines the mapping from the fuzzy sets in the input space X to the fuzzy sets in the output space Y. Each of N rules (3) determines a fuzzy set $\overline{B}^k \subseteq Y$ given by the compositional rule of inference

$$\overline{B}^k = A' \circ \left(A^k \to B^k\right), \tag{5}$$

where $A^k = A_1^k \times A_2^k \times \ldots \times A_n^k$. Fuzzy sets $\overline{B}^k$ are characterized by membership functions expressed by the sup-star composition

$$\mu_{\overline{B}^k}(y) = \sup_{\mathrm{x} \in \mathbf{X}} \left\{ \mu_{A'}(\mathbf{x}) * \mu_{A_1^k \times \ldots \times A_n^k \to B^k}(\mathbf{x}, y) \right\}, \tag{6}$$

where $*$ can be any operator in the class of t-norms. It is easily seen that for a crisp input $\overline{\mathbf{x}} = [\overline{x}_1, \ldots, \overline{x}_n] \in \mathbf{X}$, i.e., the singleton fuzzifier (1), formula (6) becomes

$$\begin{aligned} \mu_{\overline{B}^k}(y) &= \mu_{A_1^k \times \ldots \times A_n^k \to B^k}(\overline{\mathbf{x}}, y) \\ &= \mu_{A^k \to B^k}(\overline{\mathbf{x}}, y) \\ &= I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right), \end{aligned} \tag{7}$$

where $I(\cdot)$ is given by

$$I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right) = T\left\{\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\mathbf{y})\right\}. \tag{8}$$

in the case of the Mamdani approach or by

$$I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right) = S\left\{1 - \mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right\}. \tag{9}$$

in the case of the logical approach. In equation (9) we use an S-implication. We refer to (8) and (9) as to “engineering implication” or “fuzzy implication”, respectively (see [16]). Typical examples of the S-implication include the following formulas:

a) The Kleene-Dienes implication

$$I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right) = \max\left\{1 - \mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right\} \tag{10}$$

b) The Lukasiewicz implication

$$I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right) = \min\left\{1, 1 - \mu_{A^k}(\overline{\mathbf{x}}) + \mu_{B^k}(y)\right\} \tag{11}$$

c) The Reichenbach implication

$$I\left(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\right) = 1 - \mu_{A^k}(\overline{\mathbf{x}}) + \mu_{A^k}(\overline{\mathbf{x}}) \cdot \mu_{B^k}(y) \tag{12}$$

d) The Fodor implication

$$I\left(\mu_{A^k}\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(y\right)\right) \tag{13}$$
$$= \begin{cases} 1 & \text{if } \mu_{A^k}\left(\overline{\mathbf{x}}\right) \leqslant \mu_{B^k}\left(y\right) \\ \max\left\{1-\mu_{A^k}\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(y\right)\right\} & \text{if } \mu_{A^k}\left(\overline{\mathbf{x}}\right) > \mu_{B^k}\left(y\right) \end{cases}$$

Obviously, an S-implication can be generated by other t-conorms. The aggregation operator, applied in order to obtain the fuzzy set B' based on fuzzy sets $\overline{B}^k$, is the t-norm or t-conorm operator, depending on the type of fuzzy implication. In case of the Mamdani approach, the aggregation is carried out by

$$B' = \bigcup_{k=1}^{N} \overline{B}^k. \tag{14}$$

The membership function of B' is computed by the use of a t-conorm, that is

$$\mu_{B'}\left(y\right) = \mathop{S}_{k=1}^{N} \mu_{\overline{B}^k}\left(y\right). \tag{15}$$

When we use the logical model, the aggregation is carried out by

$$\overline{B} = \bigcap_{k=1}^{N} \overline{B}^k. \tag{16}$$

The membership function of B' is determined by the use of a t-norm, i.e.

$$\mu_{B'}\left(y\right) = \mathop{T}_{k=1}^{N} \left\{\mu_{\overline{B}^k}\left(y\right)\right\}. \tag{17}$$

The defuzzifier performs a mapping from the fuzzy set B' to a crisp point $\overline{y}$ in $Y \subset R$. The COA (center of area) method is defined by the following formula

$$\overline{y} = \frac{\int_{\mathrm{Y}} y \cdot \mu_{B'}\left(y\right) dy}{\int_{\mathrm{Y}} \mu_{B'}\left(y\right) dy} \tag{18}$$

or by

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \mu_{B'}\left(\overline{y}^r\right)}{\sum_{r=1}^{N} \mu_{B'}\left(\overline{y}^r\right)} \tag{19}$$

in the discrete form, where $\overline{y}^r$ are centers of the membership functions $\mu_{B^r}(y)$, i.e., for $r = 1, \ldots, N$

$$\mu_{B^r}(\overline{y}^r) = \max_{y \in \mathrm{Y}} \{\mu_{B^r}(y)\}. \tag{20}$$

Mamdani-type neuro-fuzzy systems

In this approach function $I(\cdot)$ is given by formula (8). Consequently,

$$I(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)) = T\{\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\}. \tag{21}$$

The aggregated output fuzzy set $B' \subseteq Y$ is given by

$$\mu_{B'}(\overline{y}^r) = \mathop{S}_{k=1}^{N} \left\{\mu_{\overline{B}^k}(\overline{y}^r)\right\} = \mathop{S}_{k=1}^{N} \{T\{\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\}\}. \tag{22}$$

Consequently, formula (19) takes the form

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \mathop{S}_{k=1}^{N} \left\{T\left\{\mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\overline{x}_i)\right\}, \mu_{B^k}(\overline{y}^r)\right\}\right\}}{\sum_{r=1}^{N} \mathop{S}_{k=1}^{N} \left\{T\left\{\mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\overline{x}_i)\right\}, \mu_{B^k}(\overline{y}^r)\right\}\right\}}. \tag{23}$$

Logical-type neuro-fuzzy systems

In this approach function $I(\cdot)$ is given by formula (9). Therefore

$$I(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)) = S\{1 - \mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\}. \tag{24}$$

The aggregated output fuzzy set $B' \subseteq Y$ is given by

$$\begin{aligned} \mu_{B'}(\overline{y}^r) &= \mathop{T}_{k=1}^{N} \left\{\mu_{\overline{B}^k}(\overline{y}^r)\right\} \\ &= \mathop{T}_{k=1}^{N} \{S\{N(\mu_{A^k}(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r)\}\}. \end{aligned} \tag{25}$$

and formula (19) becomes

$$\overline{y} = \frac{\sum_{r=1}^{N} \overline{y}^r \cdot \mathop{T}_{k=1}^{N} \left\{S\left\{N\left(\mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\overline{x}_i)\right\}\right), \mu_{B^k}(\overline{y}^r)\right\}\right\}}{\sum_{r=1}^{N} \mathop{T}_{k=1}^{N} \left\{S\left\{N\left(\mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\overline{x}_i)\right\}\right), \mu_{B^k}(\overline{y}^r)\right\}\right\}}. \tag{26}$$

Generalized neuro-fuzzy systems

We will generalize both approaches described in sections "Mamdani-type neuro-fuzzy systems" and "Logical-type neuro-fuzzy systems" and propose a general architecture of fuzzy systems. Observe that systems (23) and (26) can be presented in the form

$$\overline{y} = f(\overline{\mathbf{x}}) = \frac{\sum\limits_{r=1}^{N} \overline{y}^r \cdot \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\sum\limits_{r=1}^{N} \mathrm{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}, \tag{27}$$

where

$$agr_r(\overline{\mathbf{x}}, \overline{y}^r) = \begin{cases} \mathop{S}\limits_{k=1}^{N} \{I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)\} & \text{for } \textit{the Mamdani approach} \\ \mathop{T}\limits_{k=1}^{N} \{I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r)\} & \text{for } \textit{the logical approach} \end{cases} \tag{28}$$

$$I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r) = \begin{cases} T\{\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)\} \\ \quad \text{for } \textit{the Mamdani approach} \\ S(1-\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)) \\ \quad \text{for } \textit{the logical approach} \end{cases}. \tag{29}$$

The firing strength of rules τ_k is defined by formula (4). The general architecture of system (27) is depicted in Fig. 2.

Flexibility in fuzzy systems

We will present various concepts leading to the designing flexible neuro-fuzzy systems.

Compromise operators

In this section we propose the following combination of the Mamdani-type and logical-type systems:

$$\begin{aligned} I(\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)) &= (1-\lambda)\, T\{\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\} \\ &\quad + \lambda S\{1-\mu_{A^k}(\overline{\mathbf{x}}), \mu_{B^k}(y)\} \end{aligned} \tag{30}$$

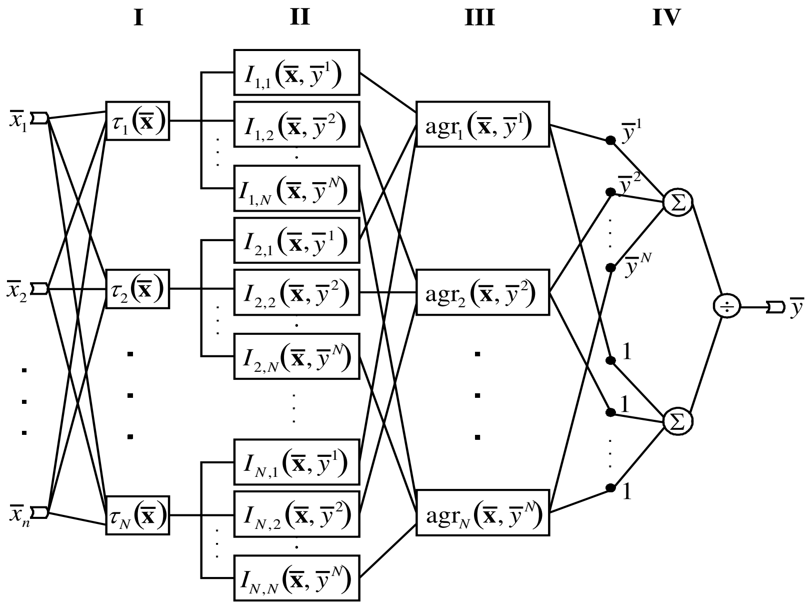

Figure 2. General architecture of fuzzy systems studied in the paper (flexible and nonflexible)

where parameter λ should satisfy $0 \leqslant \lambda \leqslant 1$.

Example 1. As an example we define the following combination

$$I\left(\mu_{A^k}\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(y\right)\right) = \begin{array}{l}(1-\lambda)\min\left\{\mu_{A^k}\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(y\right)\right\} \\ +\lambda\max\left\{1-\mu_{A^k}\left(\overline{\mathbf{x}}\right), \mu_{B^k}\left(y\right)\right\}\end{array} \tag{31}$$

Weighted triangular norms

We propose the weighted t-norm

$$T^*\left\{a_1,\ldots,a_n\,; w_1^\tau,\ldots,w_n^\tau\right\} = \mathop{T}_{i=1}^{n}\left\{1-w_i^\tau\left(1-\alpha_i\right)\right\} \tag{32}$$

to connect the antecedents in each rule, $k = 1, \ldots, N$, and the weighted t-norm and t-conorm:

$$T^* \left\{a_1, \ldots, a_N \;; w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}\right\} = \mathop{T}_{k=1}^{N} \left\{1 - w_k^{\text{agr}} \left(1 - a_k\right)\right\} \tag{33}$$

$$S^* \left\{a_1, \ldots, a_N \;; w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}\right\} = \mathop{S}_{k=1}^{N} \left\{w_k^{\text{agr}} a_k\right\} \tag{34}$$

to aggregate the individual rules in the logical and Mamdani models, respectively. The weights w_i^{τ}, w_k and w_k^{agr} are in the interval [0,1]. It is easily seen that formula (32) can be applied to the evaluation of an importance of input linguistic values, and the weighted t-norm (33) or t-conorm (34) to a selection of important rules.

Parameterized triangular norms

Parameterized triangular norms include the Dombi, Hamacher, Yager, Frank, Weber I, Weber II, Dubois-Prade and other families [6]. We use notation $\overleftrightarrow{T}\{a_1, a_2, \ldots, a_n; p\}$ and $\overleftrightarrow{S}\{a_1, a_2, \ldots, a_n; p\}$ for parameterized triangular norms. The hyperplanes corresponding to them can be adjusted in the process of learning of parameter p. As an example we present the Dombi family of parameterized triangular norms. The t-norm and t-conorm are given as follows:

a) The Dombi t-norm

$$\overleftrightarrow{T}\{\mathbf{a}; p\} = \begin{cases} \text{drastic } t\text{-norm} & \text{for } p = 0 \\ \dfrac{1}{1 + \left(\sum\limits_{i=1}^{n} \left(\dfrac{1 - a_i}{a_i}\right)^p\right)^{\frac{1}{p}}} & \text{for } p \in (0, \infty) \\ \text{Zadeh } t\text{-norm} & \text{for } p = \infty \end{cases} \tag{35}$$

where $\overleftrightarrow{T}$ stands for a t-norm of the Dombi family parameterized by p.

b) The Dombi t-conorm

$$\overset{\leftrightarrow}{S}\{\mathbf{a};p\} = \begin{cases} \text{drastic } t\text{-conorm} & \text{for } p = 0 \\ 1 - \dfrac{1}{1 + \left(\sum\limits_{i=1}^{n}\left(\dfrac{a_i}{1-a_i}\right)^p\right)^{\frac{1}{p}}} & \text{for } p \in (0,\infty) \\ \text{Zadeh } t\text{-conorm} & \text{for } p = \infty \end{cases} \tag{36}$$

where $\overset{\leftrightarrow}{S}$ stands for a t-conorm of the Dombi family parameterized by p.

Combining the S-implication and (36) we get the fuzzy S-implication generated by the Dombi family

$$\overset{\leftrightarrow}{I}(a,b;p) = 1 - \frac{1}{1 + \left(\left(\dfrac{1-a}{a}\right)^p + \left(\dfrac{b}{1-b}\right)^p\right)^{\frac{1}{p}}}. \tag{37}$$

Soft fuzzy norms

In this section we recall a concept of soft fuzzy norms proposed by Yager and Filev [21]. Let $a_1,\ldots,a_n$ be numbers in the unit interval that are to be aggregated. The soft version of triangular norms suggested by Yager and Filev is defined by

$$\widetilde{T}\{\mathbf{a};\alpha\} = (1-\alpha)\frac{1}{n}\sum_{i=1}^{n} a_i + \alpha \overset{n}{\underset{i=1}{T}}\{a_i\} \tag{38}$$

$$\widetilde{S}\{\mathbf{a};\alpha\} = (1-\alpha)\frac{1}{n}\sum_{i=1}^{n} a_i + \alpha \overset{n}{\underset{i=1}{S}}\{a_i\} \tag{39}$$

where $\alpha \in [0,1]$. They allow to balance between the arithmetic average aggregator and the triangular norm aggregator depending on parameter α.

Example 2. (An example of soft algebraic triangular norms) The soft algebraic triangular norms are based on classical algebraic triangular norms (see e.g. [6]). The soft algebraic t-norm is described as follows

$$\widetilde{T}\{a_1,a_2;\alpha\} = (1-\alpha)\frac{1}{2}(a_1+a_2) + \alpha a_1 a_2. \tag{40}$$

Flexible neuro-fuzzy systems

In this section we will derive flexible neuro-fuzzy systems by making use of the concepts presented in section "Flexibility in fuzzy systems". We start with a definition which is a generalization of a strong negation (see [6]).

Definition 1. (Compromise operator) Function

$$\widetilde{N}_\nu : [0,1] \to [0,1] \tag{41}$$

given by

$$\begin{aligned} \widetilde{N}_\nu (a) &= (1-\nu) N(a) + \nu N(N(a)) \\ &= (1-\nu) N(a) + \nu a \end{aligned} \tag{42}$$

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

Observe that

$$\widetilde{N}_\nu (a) = \begin{cases} N(a) & \text{for} \quad \nu = 0 \\ \frac{1}{2} & \text{for} \quad \nu = \frac{1}{2} \\ a & \text{for} \quad \nu = 1 \end{cases} \tag{43}$$

Obviously function $\widetilde{N}_\nu$ is a strong negation for $\nu = 0$. The 3D plot of function (42) is depicted in Fig. 3.

Definition 3. (H-function) Function

$$H : [0,1]^n \to [0,1]$$

given by

$$\begin{aligned} H(\mathbf{a};\nu) &= \widetilde{N}_\nu \left(\mathop{S}_{i=1}^{n} \left\{ \widetilde{N}_\nu (a_i) \right\} \right) \\ &= \widetilde{N}_{1-\nu} \left(\mathop{T}_{i=1}^{n} \left\{ \widetilde{N}_{1-\nu} (a_i) \right\} \right) \end{aligned} \tag{44}$$

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

Theorem 1. Let T and S be dual triangular norms. Function H defined by (44) varies between a t-norm and a t-conorm as ν goes from 0 to 1.

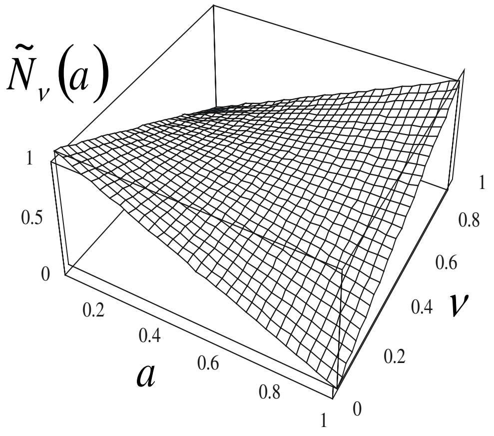

Figure 3. 3D plot of function (42)

Proof. From the assumption it follows that

$$T\{\mathbf{a}\} = N\left(S\{N(a_1), N(a_2), \ldots, N(a_n)\}\right). \tag{45}$$

For $\nu = 0$ formula (45) can be rewritten with the notation of the compromise operator (42)

$$T\{\mathbf{a}\} = \widetilde{N}_0\left(S\left\{\widetilde{N}_0(a_1), \widetilde{N}_0(a_2), \ldots, \widetilde{N}_0(a_n)\right\}\right). \tag{46}$$

Apparently

$$S\{\mathbf{a}\} = \widetilde{N}_1\left(S\left\{\widetilde{N}_1(a_1), \widetilde{N}_1(a_2), \ldots, \widetilde{N}_1(a_n)\right\}\right) \tag{47}$$

for $\nu = 1$.

The right-hand sides of (46) and (47) can be written as follows

$$H(\mathbf{a};\nu) = \widetilde{N}_\nu\left(\overset{n}{\underset{i=1}{S}}\left\{\widetilde{N}_\nu(a_i)\right\}\right) \tag{48}$$

for $\nu = 0$ and $\nu = 1$, respectively. If parameter ν changes from 0 to 1, then the result is established.

Remark 1. Observe that

$$H(\mathbf{a};\nu)=\begin{cases} T\{\mathbf{a}\} & \text{for} \quad \nu=0 \\ \frac{1}{2} & \text{for} \quad \nu=\frac{1}{2} \\ S\{\mathbf{a}\} & \text{for} \quad \nu=1. \end{cases} \tag{49}$$

It is easily seen, that for $0<\nu<0.5$ the H-function resembles a t-norm and for $0.5<\nu<1$ the H-function resembles a t-conorm.

Example 3. (An example of the H-function generated by the algebraic triangular norms) We will apply Theorem 1 to illustrate (for $n=2$) how to switch between the algebraic t-norm

$$T\{a_1,a_2\}=H(a_1,a_2;0)=a_1a_2 \tag{50}$$

and the algebraic t-conorm

$$S\{a_1,a_2\}=H(a_1,a_2;1)=a_1+a_2-a_1a_2. \tag{51}$$

The H-function generated by formulas (50) or (51) takes the form

$$\begin{aligned} H(a_1,a_2;\nu) &= \widetilde{N}_{1-\nu}\left(\widetilde{N}_{1-\nu}(a_1)\,\widetilde{N}_{1-\nu}(a_1)\right) \\ &= \widetilde{N}_{\nu}\left(1-\left(1-\widetilde{N}_{\nu}(a_1)\right)\left(1-\widetilde{N}_{\nu}(a_1)\right)\right) \end{aligned} \tag{52}$$

and varies from (50) to (51) as ν goes from zero to one. In Fig. 4, we illustrate function (52) for $\nu=0.00, \nu=0.15, \nu=0.50, \nu=0.85$, $\nu=1.00$.

Theorem 2. Let T and S be dual triangular norms.

Then

$$I(a,b;\nu)=H\left(\widetilde{N}_{1-\nu}(a),b;\nu\right) \tag{53}$$

switches between an "engineering implication"

$$I_{\text{eng}}(a,b)=I(a,b;0)=T\{a,b\} \tag{54}$$

and an S-implication

$$I_{\text{fuzzy}}(a,b)=I(a,b;1)=S\{1-a,b\} \tag{55}$$

Proof. Theorem 2 is a straightforward consequence of Theorem 1.

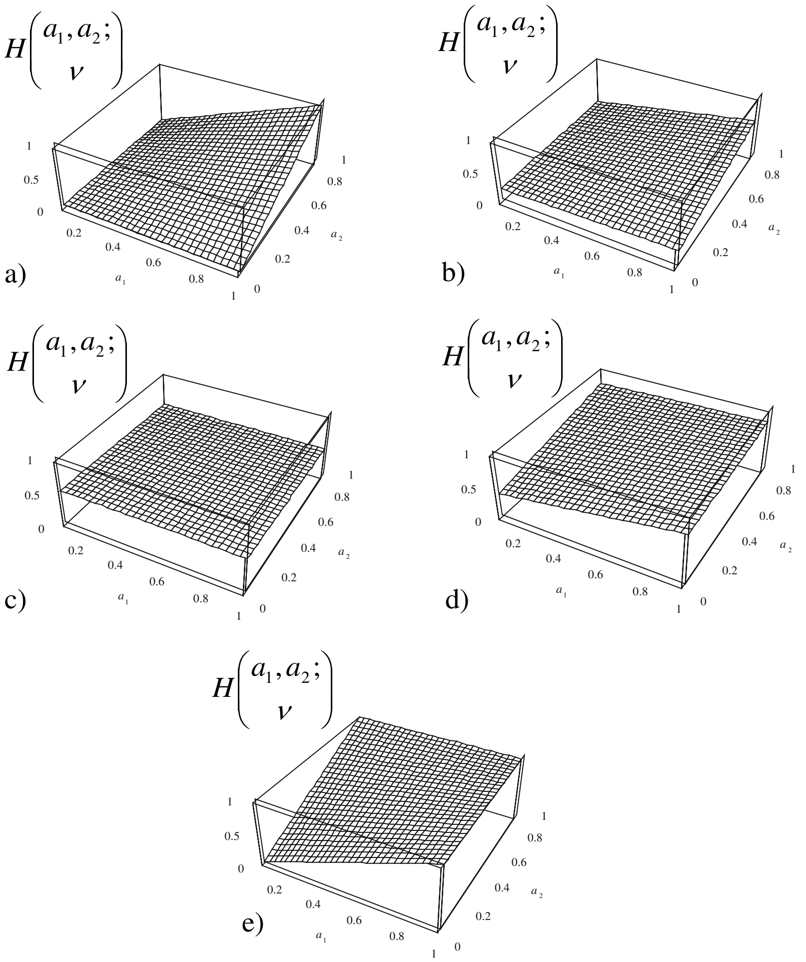

Figure 4. 3D plots of function (52) for a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$.

Example 4. (An example of the H-implication generated by the algebraic triangular norms) We will define the H-implication generated by the algebraic triangular norms. Let

$$\begin{aligned} I_{\text{eng}}(a,b) &= H(a,b;0) \\ &= T\{a,b\} \\ &= ab \end{aligned} \tag{56}$$

$$\begin{aligned} I_{\text{fuzzy}}(a,b) &= H\left(\widetilde{N}_0(a), b; 1\right) \\ &= S\{N(a), b\} \\ &= 1-a+ab. \end{aligned} \tag{57}$$

Then

$$I(a,b;\nu) = H\left(\widetilde{N}_{1-\nu}(a), b; \nu\right) \tag{58}$$

goes from (56) to (57) as ν varies from 0 to 1. The 3D plots of function (58) are depicted in Fig. 5.

The concept of generalized triangular norms leads to the following flexible neuro-fuzzy systems:

$$\overline{y} = f(\overline{\mathbf{x}}) = \frac{\sum\limits_{r=1}^{N} \overline{y}^r \cdot \text{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\sum\limits_{r=1}^{N} \text{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)} \tag{59}$$

where

$$\tau_k(\overline{\mathbf{x}}) = \begin{pmatrix} (1-\alpha^\tau)\,\text{avg}\left(\mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n)\right) + \\ +\alpha^\tau \overset{\leftrightarrow}{H}^{*} \begin{pmatrix} \mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n); \\ w_{1,k}^\tau, \ldots, w_{n,k}^\tau, p^\tau, 0 \end{pmatrix} \end{pmatrix} \tag{60}$$

$$I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r) = \begin{pmatrix} \left(1-\alpha^I\right)\text{avg}\left(\widetilde{N}_{1-\nu}(\tau_k(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r)\right) + \\ +\alpha^I \overset{\leftrightarrow}{H} \begin{pmatrix} \widetilde{N}_{1-\nu}(\tau_k(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r); \\ p^I, \nu \end{pmatrix} \end{pmatrix} \tag{61}$$

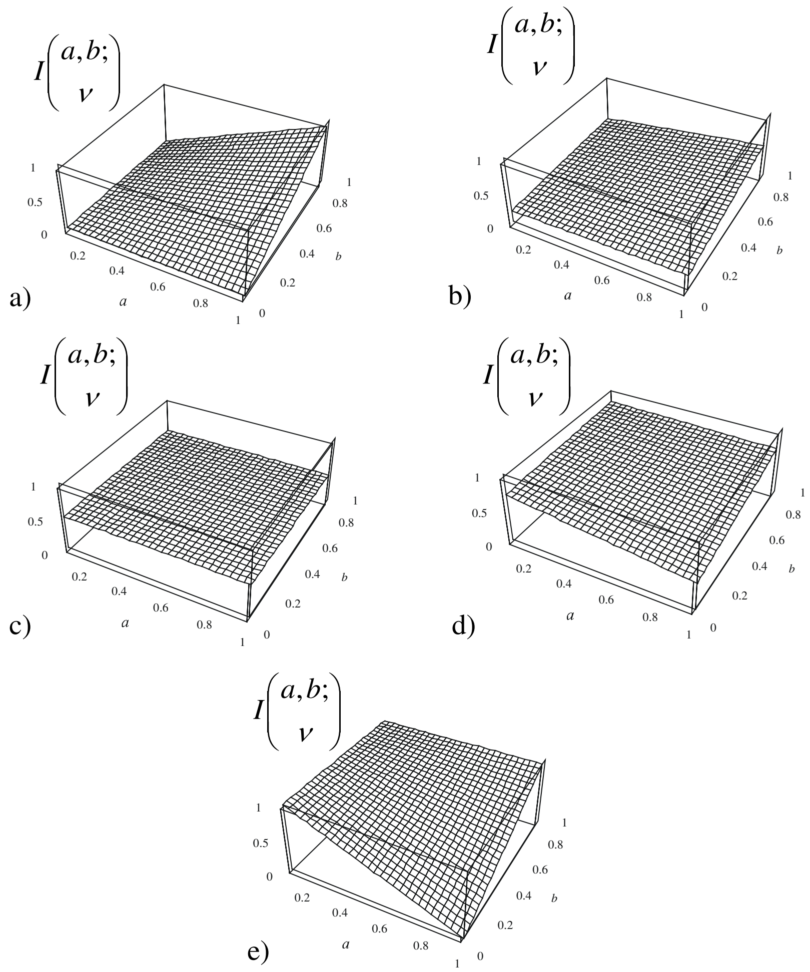

Figure 5. 3D plots of function (58) for a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$.

$$\operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r) = \begin{pmatrix} (1-\alpha^{\text{agr}}) \operatorname{avg}\left(I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r)\right) + \\ +\alpha^{\text{agr}} \overset{\leftrightarrow}{H}^{*} \begin{pmatrix} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}, p^{\text{agr}}, 1-\nu \end{pmatrix} \end{pmatrix} \tag{62}$$

In the above system we use parameterised families $\overset{\leftrightarrow}{H}(\cdot)$ and parameterised families with weights $\overset{\leftrightarrow}{H}^{*}(\cdot)$ analogously to formula (35) and (33). More specifically, in (60) and (62) we use the following definition

$$\overset{\leftrightarrow}{H}^{*} \begin{pmatrix} a_1, \ldots, a_n; \\ w_1, \ldots, w_n, p, \nu \end{pmatrix} = \overset{\leftrightarrow}{H} \begin{pmatrix} \arg_1(a_1, w_1, \nu), \ldots, \arg_n(a_n, w_n, \nu); \\ p, \nu \end{pmatrix} \tag{63}$$

where

$$\arg_i(a_i, w_i, \nu) = (1-\nu)(1 - w_i(1-a_i)) + \nu w_i a_i. \tag{64}$$

Compromise neuro-fuzzy systems

In this section we will derive neuro-fuzzy systems based on compromise operators presented in section "Compromise operators". The compromise neuro-fuzzy systems are given by the following formulas:

$$\overline{y} = f(\overline{\mathbf{x}}) = \frac{\sum\limits_{r=1}^{N} \overline{y}^r \cdot \operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)}{\sum\limits_{r=1}^{N} \operatorname{agr}_r(\overline{\mathbf{x}}, \overline{y}^r)} \tag{65}$$

where

$$\tau_k(\overline{\mathbf{x}}) = \begin{pmatrix} (1-\alpha^{\tau}) \operatorname{avg}\left(\mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n)\right) + \\ +\alpha^{\tau} \overset{\leftrightarrow}{T}^{*} \begin{pmatrix} \mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n); \\ w_{1,k}^{\tau}, \ldots, w_{n,k}^{\tau}, p^{\tau} \end{pmatrix} \end{pmatrix} \tag{66}$$

$$I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r) = \begin{pmatrix} \left(1-\alpha^{I}\right) \operatorname{avg}\left(\widetilde{N}_{1-\lambda}(\tau_k(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r)\right) + \\ +\alpha^{I} \begin{pmatrix} (1-\lambda) \overset{\leftrightarrow}{T} \begin{pmatrix} \tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r); \\ p^{I} \end{pmatrix} + \\ +\lambda \overset{\leftrightarrow}{S} \begin{pmatrix} 1-\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r); \\ p^{I} \end{pmatrix} \\ p^{I}, \nu \end{pmatrix} \end{pmatrix} \tag{67}$$

$$agr_r(\overline{\mathbf{x}}, \overline{y}^r) = \tag{68}$$
$$= \left(\begin{array}{c} (1-\alpha^{\text{agr}}) \operatorname{avg}\left(I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r)\right) + \\ +\alpha^{\text{agr}} \left(\begin{array}{c} (1-\lambda) \overset{\leftrightarrow}{S}^* \left\{ \begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}, p^{\text{agr}} \end{array} \right\} + \\ \lambda \overset{\leftrightarrow}{T}^* \left\{ \begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}, p^{\text{agr}} \end{array} \right\} \end{array} \right) \end{array} \right)$$

The above three formulas can be rewritten by making use of the H-function concept:

$$\tau_k(\overline{\mathbf{x}}) = \left(\begin{array}{c} (1-\alpha^\tau) \operatorname{avg}\left(\mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n)\right) + \\ +\alpha^\tau \overset{\leftrightarrow}{H}^* \left(\begin{array}{c} \mu_{A_1^k}(\overline{x}_1), \ldots, \mu_{A_n^k}(\overline{x}_n); \\ w_{1,k}^\tau, \ldots, w_{n,k}^\tau, p^\tau, 0 \end{array} \right) \end{array} \right) \tag{69}$$

$$I_{k,r}(\overline{\mathbf{x}}, \overline{y}^r) = \left(\begin{array}{c} \left(1-\alpha^I\right) \operatorname{avg}\left(\widetilde{N}_{1-\lambda}(\tau_k(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r)\right) + \\ +\alpha^I \left(\begin{array}{c} (1-\lambda) \overset{\leftrightarrow}{H} \left(\begin{array}{c} \widetilde{N}_1(\tau_k(\overline{\mathbf{x}}), \mu_{B^k}(\overline{y}^r)); \\ p^I, 0 \end{array} \right) + \\ +\lambda \overset{\leftrightarrow}{H} \left(\begin{array}{c} \widetilde{N}_0(\tau_k(\overline{\mathbf{x}})), \mu_{B^k}(\overline{y}^r); \\ p^I, 1 \end{array} \right) \\ p^I, \nu \end{array} \right) \end{array} \right) \tag{70}$$

$$agr_r(\overline{\mathbf{x}}, \overline{y}^r) = \tag{71}$$
$$= \left(\begin{array}{c} (1-\alpha^{\text{agr}}) \operatorname{avg}\left(I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r)\right) + \\ +\alpha^{\text{agr}} \left(\begin{array}{c} (1-\lambda) \overset{\leftrightarrow}{H}^* \left(\begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}, p^{\text{agr}}, 1 \end{array} \right) + \\ \lambda \overset{\leftrightarrow}{H}^* \left(\begin{array}{c} I_{1,r}(\overline{\mathbf{x}}, \overline{y}^r), \ldots, I_{N,r}(\overline{\mathbf{x}}, \overline{y}^r); \\ w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}, p^{\text{agr}}, 0 \end{array} \right) \end{array} \right) \end{array} \right)$$

Learning procedures

We will explain how to learn parameters of membership functions and flexability parameters in systems presented in sections “Flexible neuro-fuzzy systems” and “Compromise neuro-fuzzy systems”. Let $\overline{\mathbf{x}}(t) \in R^n$ and $d(t) \in R$ be a sequence of inputs and desirable output signals, respectively.

Based on the learning sequence $(\overline{\mathbf{x}}(1), d(1)), (\overline{\mathbf{x}}(2), d(2)), \ldots$ we wish to determine all parameters (including the compromise parameter λ and system's type ν) and weights of neuro-fuzzy systems such that

$$e(t) = \frac{1}{2}\left[f(\overline{\mathbf{x}}(t)) - d(t)\right]^2 \qquad (72)$$

is minimized, where $f(\cdot)$ is given by (59) or (65). The steepest descent optimization algorithm can be applied to solve this problem. For instance, parameters $\overline{y}^r, r = 1, \ldots, N$, are trained by the iterative procedure

$$\overline{y}^r(t+1) = \overline{y}^r(t) - \eta \frac{\partial e(t)}{\partial \overline{y}^r(t)}. \qquad (73)$$

Directly calculating partial derivatives in recursion (73) is rather complicated. Therefore, we recall that our system has a layered architecture (see Fig. 2) and apply the idea of the backpropagation method to train the system. The exact recursions are not shown here, however they can be derived analogously to the method given in [16]. We can apply the gradient optimization with constraints in order to optimize:

$$\begin{aligned}
\nu &\in [0,1], \ \lambda \in [0,1] \\
\alpha^{\tau} &\in [0,1], \ \alpha^{I} \in [0,1], \ \alpha^{\text{agr}} \in [0,1], \\
p^{\tau} &\in [0,\infty), \ p^{I} \in [0,\infty), \ p^{\text{agr}} \in [0,\infty), \\
w_{i,k}^{\tau} &\in [0,1], \ i = 1,\ldots,n, \ k = 1,\ldots,N, \\
w_{i,k}^{\text{agr}} &\in [0,1], \ k = 1,\ldots,N.
\end{aligned}$$

The same technique can be used in order to find in the process of learning parameters of the membership functions $\mu_{A_i^k}(x_i)$ and $\mu_{B^k}(y), i = 1,\ldots,n, k = 1,\ldots,N$.

Simulation results

In this section we apply neuro-fuzzy systems presented in sections “Flexible neuro-fuzzy systems” and “Compromise neuro-fuzzy systems” to identify a plant described by the difference equation (see [19])

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + g(u(k)) \qquad (74)$$

where the unknown function has the form

$$g(u) = 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u) \tag{75}$$

and the input is described by

$$u(k) = \sin\left(\frac{2\pi k}{250}\right) \tag{76}$$

In order to identify the plant, we use a model governed by the difference equation

$$\widehat{y}(k+1) = 0.3y(k) + 0.6y(k-1) + \widehat{f}(u(k)) \tag{77}$$

where $\widehat{f}(\cdot)$ is a flexible neuro-fuzzy system presented in section 4 (called OR-type system) or a compromise neuro-fuzzy system presented in section 5 (called AND-type system).

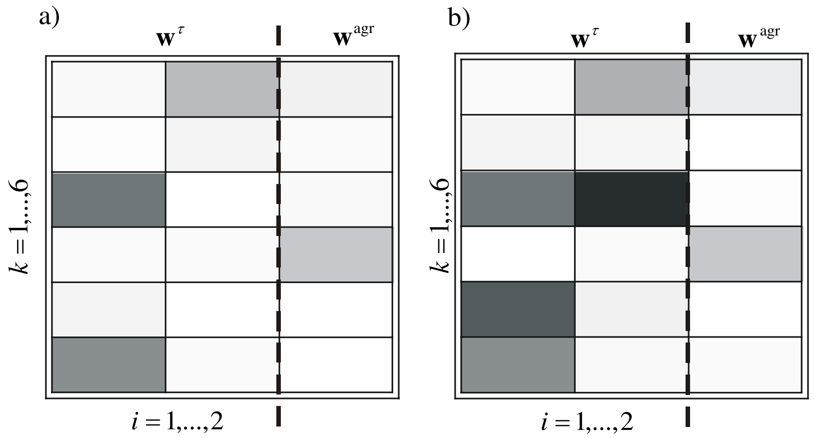

Figure 6. Weights representation in the Nonlinear Dynamic Plant problem for OR-type NFIS and a) Zadeh H-function, b) algebraic H-function

All the simulations are designed in the same fashion. We will gradually incorporate flexibility parameters in experiments (i)-(iv):

- In the first experiment (i), based on the input-output data, we learn the parameters of the membership functions and a system type $\nu \in [0,1]$ or

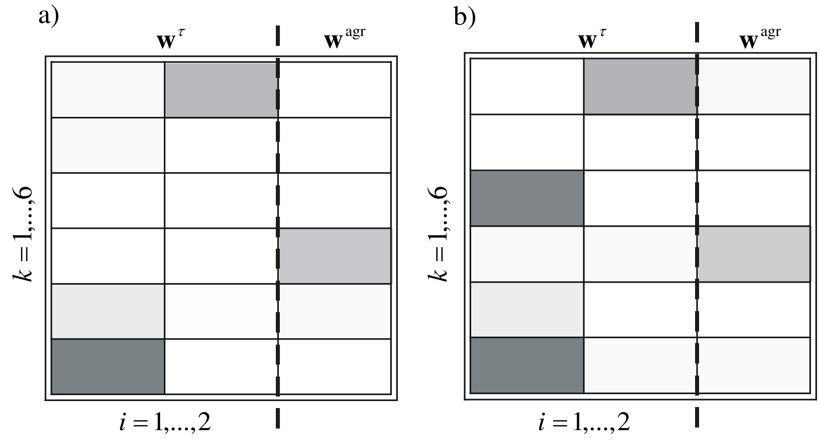

Figure 7. Weights representation in the Nonlinear Dynamic Plant problem for OR-type NFIS and a) Dombi H-function, b) Yager H-function

$\lambda \in [0,1]$ assuming that there are no other flexibility parameters in the system description. It will be seen that the optimal values of ν or λ, determined by a gradient procedure, are either zero or one.

- In the second experiment (ii), we learn the parameters of the membership functions choosing value ν or λ as opposite to that obtained in experiment (i). Obviously, we expect a worse performance of the neuro-fuzzy system comparing with experiment (i).

- In the third experiment (iii), we learn the parameters of the membership functions, system type $\nu \in [0,1]$ or $\lambda \in [0,1]$ and soft parameters $\alpha^T \in [0,1], \alpha^I \in [0,1], \alpha^{\mathrm{agr}} \in [0,1]$ of the flexible system assuming that classical (not-parameterized) triangular norms are applied.

- In the fourth experiment (iv), we learn the same parameters as in the third experiment and, moreover, the weights $w^{\tau}_{i,k} \in [0,1]$, $i = 1,\ldots,n, k = 1,\ldots,N$, in the antecedents of rules and weights $w^{\mathrm{agr}}_{i,k} \in [0,1], k = 1,\ldots,N$, of the aggregation operator of rules. In all diagrams (weights representation) we separate $w^{\tau}_{i,k} \in [0,1], i = 1,\ldots,n, k = 1,\ldots,N$, from $w^{\mathrm{agr}}_{i,k} \in [0,1], k = 1,\ldots,N$, by a vertical dashed line. In the

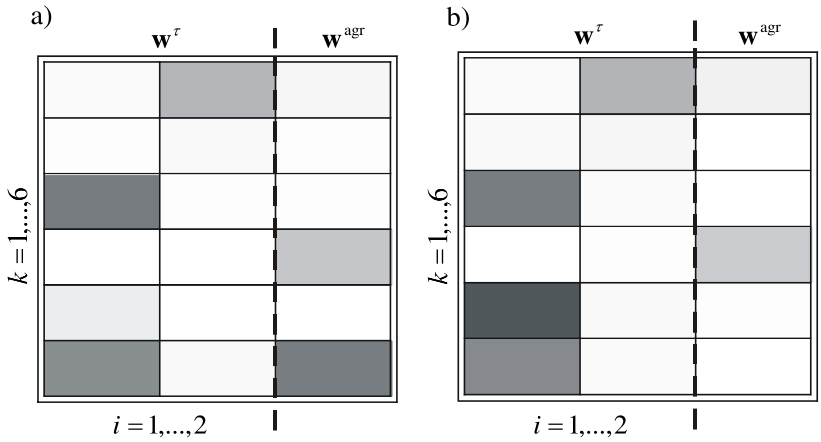

Figure 8. Weights representation in the Nonlinear Dynamic Plant problem for AND-type NFIS and a) Zadeh H-function, b) algebraic H-function

diagrams dark arears correspond to low values of weights and vice versa.

In each of the above simulations we apply the Zadeh H-implication (generated by the min/max triangular norms) and the algebraic H-implication (generated by the algebraic triangular norms). In separate experiments we repeat simulations (i)–(iv) replacing the Zadeh H-implication and the algebraic H-implication by quasi-implications generated by parameterized triangular norms: the Dombi H-implication and the Yager H-implication. In these simulations we additionally incorporate parameters $p^{\tau} \in [0, \infty)$, $p^{I} \in [0, \infty)$, $p^{\text{agr}} \in [0, \infty)$.

a) Flexible neuro-fuzzy systems

We apply neuro-fuzzy systems described in section 4 and given by formulas (59)–(62). The experimental results for the Nonlinear Dynamic Plant problem (72) are depicted in Tables 1 and 2 for the not-parameterized (Zadeh and algebraic) and parameterized (Dombi and Yager) H-functions, respectively. For experiment (iv) the final values (after learning) of weights $w^{\tau}_{i,k} \in [0, 1]$ and $w^{\text{agr}}_{k} \in [0, 1]$, $i = 1, \ldots, 2$, $k = 1, \ldots, 6$, are shown in Fig. 6 (Zadeh and algebraic H-functions) and Fig. 7 (Dombi and Yager H-functions).

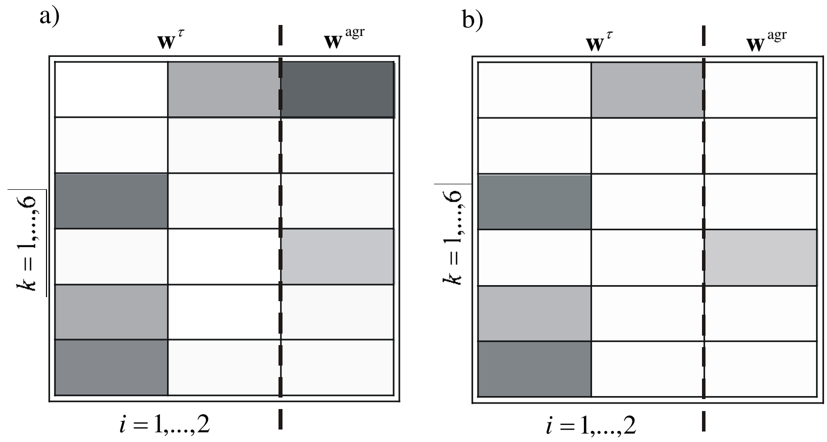

Figure 9. Weights representation in the Nonlinear Dynamic Plant problem for AND-type NFIS and a) Dombi H-function, b) Yager H-function

b) Compromise neuro-fuzzy systems

We apply neuro-fuzzy systems described in section 5 and given by formulas (65)–(68). The experimental results for the Nonlinear Dynamic Plant problem (72) are depicted in Tables 3 and 4 for the not-parameterized (Zadeh and algebraic) and parameterized (Dombi and Yager) H-functions, respectively. For experiment (iv) the final values (after learning) of weights $w^{\tau}_{i,k} \in [0,1]$ and $w^{\text{agr}}_{k} \in [0,1]$, $i = 1,\ldots,2$, $k = 1,\ldots,6$, are shown in Fig. 7 (Zadeh and algebraic H-functions) and Fig. 8 (Dombi and Yager H-functions).

It should be noted that flexible and compromise neuro-fuzzy systems perfectly approximate the plant. For example in Fig. 10 we show signals $y\,(k)$ and $\widehat{y}\,(k)$, applying system given by (59)–(62), after 4000 epochs. From tables 1 and 2 it follows that the flexible neuro-fuzzy system (59)–(62) becomes of a Mamdani-type $(\nu = 0)$ when the learning process is completed. The same conclusion $(\lambda = 0)$ is valid for compromise neuro-fuzzy system (65)–(68). In other investigations (see [12,16,17]) it was found out that neuro-fuzzy systems become of a logical-type for classification problems.

Table 1. Experimental results — system given by (59)–(62) and non-parametrized H-functions

OR-TYPE NFIS WITH NON-PARAMETRISED H-FUNCTIONS (NONLINEAR DYNAMIC PLANT)						
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		RMSE (learning sequence)	
			Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function
i	ν	0.5	0.0000	0.0000	0.0468	0.0311
ii	ν	1	-	-	0.0562	0.0370
iii	ν	0.5	0.0000	0.0000	0.0384	0.0291
	α^{τ}	1	0.8012	0.9896		
	α^{I}	1	0.7313	0.9964		
	α^{agr}	1	0.9936	0.9812		
iv	ν	0.5	0.0000	0.0000	0.0271	0.0168
	α^{τ}	1	0.8331	0.9925		
	α^{I}	1	0.6991	0.9968		
	α^{agr}	1	0.9959	0.9847		
	$\mathbf{w}^{\tau}$	**1**	Fig. 1-a	Fig. 1-b		
	$\mathbf{w}^{agr}$	**1**	Fig. 1-a	Fig. 1-b		

Table 2. Experimental results — system given by (59)–(62) and parametrized H-functions

OR-TYPE NFIS WITH PARAMETRISED H-FUNCTIONS (NONLINEAR DYNAMIC PLANT)						
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		RMSE (learning sequence)	
			Dombi H-function	Yager H-function	Dombi H-function	Yager H-function
i	ν	0.5	0.0000	0.0000	0.0392	0.0343
ii	ν	1	-	-	0.0437	0.0312
iii	ν	0.5	0.0000	0.0000	0.0304	0.0248
	p^{τ}	10	9.9986	10.9834		
	p^{I}	10	8.6509	11.9798		
	p^{agr}	10	12.8795	7.6195		
	α^{τ}	1	0.8218	0.9954		
	α^{I}	1	0.8913	0.9514		
	α^{agr}	1	0.9847	0.9945		
iv	ν	0.5	0.0000	0.0000	0.0202	0.0176
	p^{τ}	10	9.1298	9.9465		
	p^{I}	10	7.0674	15.2263		
	p^{agr}	10	11.0981	4.1546		
	α^{τ}	1	0.8694	0.9947		
	α^{I}	1	0.7943	0.9273		
	α^{agr}	1	0.9896	0.9187		
	$\mathbf{w}^{\tau}$	**1**	Fig. 2-a	Fig. 2-b		
	$\mathbf{w}^{agr}$	**1**	Fig. 2-a	Fig. 2-b		

Table 3. Experimental results — system given by (65)–(68) and non-parametrized H-functions

AND-TYPE NFIS WITH NON-PARAMETRISED H-FUNCTIONS (NONLINEAR DYNAMIC PLANT)						
Experiment number	**Name of flexibility parameter**	**Initial values**	**Final values after learning**		**RMSE (learning sequence)**	
			Zadeh H-function	**Algebraic H-function**	**Zadeh H-function**	**Algebraic H-function**
i	λ	0.5	0.0000	0.0000	0.0361	0.0342
ii	λ	1	-	-	0.0616	0.0406
iii	λ α^{τ} α^{I} α^{agr}	0.5 1 1 1	0.0000 0.8570 0.6453 0.9724	0.0000 0.9891 0.9839 0.9952	0.0279	0.0216
iv	λ α^{τ} α^{I} α^{agr} $\mathbf{w}^{\tau}$ $\mathbf{w}^{agr}$	0.5 1 1 1 **1** **1**	0.0000 0.8645 0.7957 0.9834 Fig. 3-a Fig. 3-a	0.0000 0.9847 0.9587 0.9893 Fig. 3-b Fig. 3-b	0.0213	0.0137

Table 4. Experimental results — system given by (65)–(68) and parametrized H-functions

AND-TYPE NFIS WITH PARAMETRISED H-FUNCTIONS (NONLINEAR DYNAMIC PLANT)						
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		RMSE (learning sequence)	
			Dombi H-function	Yager H-function	Dombi H-function	Yager H-function
i	λ	0.5	0.0000	0.0000	0.0391	0.0365
ii	λ	1	-	-	0.0645	0.0578
iii	λ	0.5	0.0000	0.0000	0.0266	0.0274
	p^{τ}	10	9.6428	10.9448		
	p^{I}	10	7.9106	13.8753		
	p^{agr}	10	10.2587	3.1987		
	α^{τ}	1	0.8927	0.9817		
	α^{I}	1	0.7314	0.9678		
	α^{agr}	1	0.9927	0.9824		
iv	λ	0.5	0.0000	0.0000	0.0207	0.0186
	p^{τ}	10	10.0024	9.5846		
	p^{I}	10	7.0568	10.8733		
	p^{agr}	10	11.5201	6.5864		
	α^{τ}	1	0.8942	0.9817		
	α^{I}	1	0.8917	0.9439		
	α^{agr}	1	0.9949	0.9486		
	$\mathbf{w}^{\tau}$	**1**	Fig. 4-a	Fig. 4-b		
	$\mathbf{w}^{agr}$	**1**	Fig. 4-a	Fig. 4-b		

Final remarks: Design of flexible neuro-fuzzy systems

In the paper we introduced several flexibility concepts in the design of neuro-fuzzy systems:

- softness to fuzzy implication operators, to the aggregation of rules and to the connectives of antecedents,
- certainty weights to the aggregation of rules and to the connectives of antecedents,
- parameterized families of t-norms and t-conorms to fuzzy implication operators, to the aggregation of rules and to the connectives of antecedents.

Moreover, our design process was characterized by the automatic determination of fuzzy inference described by parameter ν or λ. The main advantage of our approach is a high accuracy of neuro-fuzzy systems. In the future research we plan to apply neuro-fuzzy systems in a time-varying environment [13]–[15].

References

1. *Chen M.Y., Linkens D.A.* A systematic neuro-fuzzy modeling framework with application to material property prediction // *IEEE Trans. on Fuzzy Systems*, **31** (2001) 781–790.
2. *González, Pérez R.* SLAVE: A genetic learning system based on an iterative approach // *IEEE Trans. on Fuzzy Systems*, **7** (1999) 176–191.
3. *Jang J.S., Sun C.T., Mizutani E.* Neuro-Fuzzy and Soft Computing, Prentice Hall, Englewood Cliffs 1997.
4. *Kasabov N.* DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction // *IEEE Trans. on Fuzzy Systems*, **10** (2002) 144–154.
5. *Kim E., Park M., Ji S., Park M.* A new approach to fuzzy modeling // *IEEE Trans. on Fuzzy Systems*, **5** (1997) 328–337.
6. *Klement E.P., Mesiar R., Pap E.* Triangular Norms, Kluwer Academic Publishers, Netherlands 2000.
7. *Roubos H., Setnes M.* Compact and transparent fuzzy models and classifiers through iterative complexity reduction // *IEEE Trans. on Fuzzy Systems*, **9** (2001) 516–524.
8. *Rutkowski L., Cpalka K.* Flexible Structures of Neuro-Fuzzy Systems, Quo Vadis Computational Intelligence // *Studies in Fuzziness and Soft Computing*, Springer-Verlag, **54** (2000) 479–484.

9. *Rutkowski L., Cpalka K.* A general approach to neuro-fuzzy systems // *Proc. of the 10th IEEE Intern. Conference on Fuzzy Systems*, Melbourne 2001.

10. *Rutkowski L., Cpalka K.* A neuro-fuzzy controller with a compromise fuzzy reasoning // *Control and Cybernetics*, **31**, 2 (2002) 297–308.

11. *Rutkowski L., Cpalka K.* Flexible weighted neuro-fuzzy systems // *Proc. 9th Intern. Conference on Neural Information Processing (ICONIP'02)*, Orchid Country Club, Singapore 2002.

12. *Rutkowski L., Cpalka K.* Flexible neuro-fuzzy systems // *IEEE Trans. Neural Networks*, **14** (May 2003) 554–574.

13. *Rutkowski L.* Adaptive probabilistic neural networks for pattern classification in time-varying environment // *IEEE Trans. on Neural Networks*, **15** (2004) 811–827.

14. *Rutkowski L.* Generalized regression neural networks in time-varying environment // *IEEE Trans. on Neural Networks*, **15** (2004) 576–596.

15. *Rutkowski L.* New Soft Computing Techniques For System Modeling, Pattern Classification and Image Processing, Springer-Verlag, 2004.

16. *Rutkowski L.* Flexible Neuro-Fuzzy Systems, Kluwer Academic Publishers, 2004.

17. *Rutkowski L.* A new method for system modelling and pattern classification // *Bulletin of the Polish Academy of Sciences*, **52** (2004) 11–24.

18. *Setnes M., Roubos H.* GA-fuzzy modeling and classification complexity and performance // *IEEE Trans. on Fuzzy Systems*, **8** (2000) 509–521.

19. *Wang L.* Adaptive Fuzzy Systems and Control, Design and Stability Analysis, Prentice Hall, 1994.

20. *Wang L., Yen J.* Application of statistical information criteria for optimal fuzzy model construction // *IEEE Trans. on Fuzzy Systems*, **6** (1998) 362–371.

21. *Yager R.R., Filev D.P.* Essentials of Fuzzy Modeling and Control, New York, John Wiley & Sons, 1994.

Leszek RUTKOWSKI was born in Wroclaw, Poland, in 1952. He received M. Sc., Ph. D. and D. Sc. degrees in 1977, 1980, 1986, respectively, all from the Technical University of Wroclaw, Poland. Since 1980, he has been with the Technical University of Czestochowa where he is currently a Professor and Chairman of the Computer Engineering Department. From 1987 to 1990 he held a visiting position in the School of Electrical and Computer Engineering at Oklahoma State University. His research interests include neural networks, fuzzy systems, computational intelligence, pattern recognition and systems identification. He published over 100 technical papers including 17 in various series

IEEE Transactions. He is the author of the books "New Soft Computing Techniques For System Modelling, Pattern Classification and Image Processing" published by Springer, "Flexible Neuro-Fuzzy Systems" published by Kluwer Academic Publishers, "Adaptive Filters and Adaptive Signal Processing" (in Polish), and co-author of two others (in Polish and Russian) "Neural Networks, Genetic Algorithms and Fuzzy Systems" and "Neural Networks for Image Compression". He is also President and Founder of the Polish Neural Networks Society. He organized and served as General Chair of the Polish Neural Networks Conferences held in: 1996, 1997, 1999, 2000, 2002. Prof. Leszek Rutkowski is an Associate Editor of the IEEE Transactions on Neural Networks and recently elected member of the Polish Academy of Sciences.

Лешек РУТКОВСКИЙ родился во Вроцлаве, Польша, в 1952 г. Степени M. Sc., Ph. D. и D. Sc. получил 1977, 1980, 1986, соответственно, все во Вроцлавском техническом университете. Начиная с 1980 года он работает в Ченстоховском техническом университете, Ченстохова, Польша, где является в настоящее время профессором и заведующим кафедрой вычислительной техники. С 1987 по 1990 год он работал в качестве приглашенного профессора на факультете электротехники и вычислительной техники Оклахомского государственного университета, Оклахома, США. Исследовательские интересы проф. Рутковского включают нейронные сети, нечеткие системы, вычислительный интеллект, распознавание образов и идентификацию систем. Им опубликовано свыше 100 научных работ, включая 17 статей в различных сериях журнала IEEE Transactions. Проф. Рутковский является автором нескольких книг, в том числе: «Новые методы мягких вычислений для моделирования систем, классификации образов и обработки изображений» (издательство Springer, на английском языке), «Гибкие нейро-нечеткие системы» (издательство Kluwer Academic Publishers, на английском языке), «Адаптивные фильтры и адаптивная обработка сигналов» (издано на польском языке), а также соавтором книг «Нейронные сети, генетические алгоритмы и нечеткие системы» (издано на польском и русском языках) и «Нейронные сети для сжатия изображений». Проф. Рутковский является членом Польской Академии наук, основателем и президентом Польского общества нейронных сетей. Он был председателем оргкомитетов Польских конференций по нейронным сетям, проводившихся в 1996, 1997, 1999, 2000 и 2002 годах. Проф. Рутковский входит в состав редакционной коллегии журнала IEEE Transactions on Neural Networks.