JAISCR, 2022, Vol. 12, No. 3, pp. 197 – 206
10.2478/jaiscr-2022-0013

A NOVEL APPROACH TO TYPE-REDUCTION AND DESIGN OF INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS

Janusz T. Starczewski[1,*], Krzysztof Przybyszewski, [2], Aleksander Byrski[3], Eulalia Szmidt[4], Christian Napoli[5]

[1]*Department of Computational Intelligence, Częstochowa University of Technology, al. Armii Krajowej 36, 42-200 Częstochowa, Poland*

[2]*Institute of Information Technologies, University of Social Sciences, ul. Sienkiewicza 9, 90-113 Łódź*

[3]*Institute of Computer Science, AGH University of Science and Technology, 30-059 Kraków, Poland*

[4]*Systems Research Institute of the Polish Academy of Sciences, 01-447 Warsaw, Poland*

[5]*Department of Computer, Control and Management Engineering, Sapienza University of Rome, Via Ariosto 25 Roma 00185, Italy*

[*]*E-mail: janusz.starczewski@pcz.pl*

Submitted: 5th January 2022; Accepted: 24th June 2022

Abstract

Fuzzy logic systems, unlike black-box models, are known as transparent artificial intelligence systems that have explainable rules of reasoning. Type 2 fuzzy systems extend the field of application to tasks that require the introduction of uncertainty in the rules, e.g. for handling corrupted data. Most practical implementations use interval type-2 sets and process interval membership grades. The key role in the design of type-2 interval fuzzy logic systems is played by the type-2 inference defuzzification method. In type-2 systems this generally takes place in two steps: type-reduction first, then standard defuzzification. The only precise type-reduction method is the iterative method known as Karnik-Mendel (KM) algorithm with its enhancement modifications. The known non-iterative methods deliver only an approximation of the boundaries of a type-reduced set and, in special cases, they diminish the profits that result from the use of type-2 fuzzy logic systems. In this paper, we propose a novel type-reduction method based on a smooth approximation of maximum/minimum, and we call this method a smooth type-reduction. Replacing the iterative KM algorithm by the smooth type-reduction, we obtain a structure of an adaptive interval type-2 fuzzy logic which is non-iterative and as close to an approximation of the KM algorithm as we like.

Keywords: smooth type-reduction, interval type-2 fuzzy logic systems

1 Introduction

The need to exercise control over Artificial Intelligence tools has focused researchers' interest on explainable models [5, 19]. Among several explainable intelligent methods, fuzzy logic systems are seen as particularly interesting due to their transparency of fuzzy rules. However, fuzzy rules operating in an uncertain or non-stationary environment require a higher order of fuzziness. Therefore, conditional propositions are frequently equipped with fuzzy sets of type-2 [23] Such type-2 fuzzy sets are described with a fuzzified membership function of the assumed shape. A rectangular shape of membership fuzzification, in interval type-2 sets, sufficiently describes uncertainty in the modeling of most practical processes. While other general type-2 sets, like Gaussian or triangular, are computationally expensive in defuzzification of output information, defuzzification of interval type-2 conclusions requires a sufficiently small number of computations. Technically, defuzzification of type-2 fuzzy sets is split into two phases: type-reduction mapping from a type-2 to a type-1 fuzzy set, and final defuzzification into a crisp fuzzy set.

For interval type-2 fuzzy sets, a common defuzzification algorithm is known as the Karnik–Mendel (KM) iterative procedure [7, 8, 11], which began two decades of interest in applications of interval type-2 fuzzy logic. In recent years, mostly modifications of the basic KM algorithm have been being developed [22, 15, 10]. Optimization of initial switch points underlie enhanced KM algorithms [21, 4]. Independent noniterative approaches have been addressed for solving the centroids of interval type-2 fuzzy sets as the Nie–Tan algorithm [9, 14] or the Nagar–Bardini method [6, 4]. However, the KM algorithm together with its acceleration modifications is the only accurate type-reduction method, while the known non-iterative methods just approximate a type-reduced set.

The iteration procedure in the KM algorithm is time-consuming and make problems with the determination of exact derivatives with respect to parameters of the type-2 fuzzy system. Moreover, a standard backpropagation algorithm for supervised learning of adaptive fuzzy systems propagates errors through a small part of the system's internal connections, selected by the KM algorithm. This results in a chaotic and inaccurate learning curve. The reason for this is that the non-differentiable maximum and minimum functions are used in the selection process with the KM algorithm. Therefore, we need continuous derivative functions that approximate the maximum and minimum with sufficient precision. In mathematical analysis, there is a class of functions differentiable everywhere known as smooth function. Such approximation of the maximum function, called a smooth maximum function, is used in this article to construct a novel type-reduction method for type-2 interval fuzzy logic. Properties of the smooth maximum function are useful for optimization of interval type-2 fuzzy systems with such techniques as gradient descent [2], conjugate gradient [3], or second-order gradient [1], smoothly optimizes the interval type-2 fuzzy logic system as new data is available. Consequently, in this paper, we derive a new adaptive structure of the interval type-2 fuzzy logic system equipped with smooth type-reduction. We summarize this paper as follows:

1. We propose the use of smooth maximum in a new formulation of type-reduction for interval type-2 fuzzy sets.
2. We design a new structure of an adaptive interval type-2 fuzzy logic system with multiple layers corresponding to a new type-reduction method.
3. We apply the backpropagation optimization method that makes use of the smooth maximum and smooth minimum functions.
4. We demonstrate that the proposed design outperforms the standard type-2 fuzzy logic systems using exact calculations of the KM algorithm.

2 An overview of interval type-2 fuzzy logic systems

A typical type-2 fuzzy system contains rules of the following form:

$$\widetilde{R}_k\colon \text{IF } \mathbf{x} \text{ is } \widetilde{A}_k \text{ THEN } y \text{ is } \widetilde{B}_k. \tag{1}$$

In this case, fuzzy rules are composed of type-2 fuzzy sets that operate under the most common reasoning schema. Input to the system is the fact that x is $\widetilde{A}'$. This premise is confronted with all rules having antecedents $\widetilde{A}_k$ and consequents $\widetilde{B}_k$. Finally, a conclusion $\widetilde{B}'$ is determined as an aggregation of type-2 fuzzy consequents of active rules. In the interval case, memberships are subintervals of $\widetilde{[0,1]}$ expressed by of upper and lower bounds, e.g. $\widetilde{A}_k =$

$\left[\underline{\mu}_{A_k}(x), \overline{\mu}_{A_k}(x)\right] \subseteq [0,1]$ for each $x \in X$. The conditional fuzzy proposition (1) is considered an interval type-2 fuzzy relation $\widetilde{R} = \left[\overline{\mu}_{R_k}(\mathbf{x}, y), \overline{\mu}_{R_k}(\mathbf{x}, y)\right]$ decomposed into

$$\underline{\mu}_{R_k}(\mathbf{x}, y) = \widetilde{R}\left(\underline{\mu}_{A_k}(\mathbf{x}), \underline{\mu}_{B_k}(y)\right) \quad (2)$$

$$\overline{\mu}_{R_k}(\mathbf{x}, y) = \widetilde{R}\left(\overline{\mu}_{A_k}(\mathbf{x}), \overline{\mu}_{B_k}(y)\right) \quad (3)$$

Consequently, an interval type-2 fuzzy conclusion can be determined by a compositional rule of inference

$$\widetilde{B}'_k = \widetilde{A}' \circ \widetilde{R}_k \quad (4)$$

which is implemented as an interval version of sup-T composition

$$\left[\underline{\mu}_{\widetilde{B}'_k}(y), \overline{\mu}_{\widetilde{B}'_k}(y)\right] = \left[\begin{array}{l} \sup_{\mathbf{x} \in \mathbf{X}} \left\{T\left(\underline{\mu}_{\widetilde{A}'}(\mathbf{x}), R\left(\underline{\mu}_{\widetilde{A}_k}(\mathbf{x}), \underline{\mu}_{\widetilde{B}_k}(y)\right)\right)\right\}, \\ \sup_{\mathbf{x} \in \mathbf{X}} \left\{T\left(\overline{\mu}_{\widetilde{A}'}(\mathbf{x}), R\left(\overline{\mu}_{\widetilde{A}_k}(\mathbf{x}), \overline{\mu}_{\widetilde{B}_k}(y)\right)\right)\right\} \end{array}\right] \quad (5)$$

The composition formula can be significantly simplified if there is no need for fuzzifing input values $\mathbf{x}'$. Namely $\overline{\mu}_{\widetilde{A}'}(\mathbf{x})$ is non-zero only at $\mathbf{x}'$, and consequently, there is no need to find supremum over the whole $\mathbf{X}$,

$$\begin{aligned} &\left[\underline{\mu}_{\widetilde{B}'_k}(y), \overline{\mu}_{\widetilde{B}'_k}(y)\right] \\ &= \left[T\left(1, R\left(\underline{\mu}_{\widetilde{A}_k}(\mathbf{x}'), \underline{\mu}_{\widetilde{B}_k}(y)\right)\right),\right. \\ &\left.T\left(1, R\left(\overline{\mu}_{\widetilde{A}_k}(\mathbf{x}'), \overline{\mu}_{\widetilde{B}_k}(y)\right)\right)\right] \\ &= \left[R\left(\underline{\mu}_{\widetilde{A}_k}(\mathbf{x}'), \underline{\mu}_{\widetilde{B}_k}(y)\right), R\left(\overline{\mu}_{\widetilde{A}_k}(\mathbf{x}'), \overline{\mu}_{\widetilde{B}_k}(y)\right)\right] \end{aligned} \quad (6)$$

In most engineering cases, conjunctions in the form of t-norms are employed as relations. In consequence, the conclusion should be aggregated by $\widetilde{B}' = \bigcup_{k=1}^{R} \widetilde{B}'_k$. Consequently,

$$\left[\underline{\mu}_{\widetilde{B}'}(y), \overline{\mu}_{\widetilde{B}'}(y)\right] = \left[\mathop{S}_{k=1}^{K} \underline{\mu}_{\widetilde{B}'_k}(y), \mathop{S}_{k=1}^{K} \overline{\mu}_{\widetilde{B}'_k}(y)\right] \quad (7)$$

However, R can be realized by material fuzzy implications, as well, and thus aggregated as $\widetilde{B}' = \bigcap_{k=1}^{R} \widetilde{B}'_k$. Nevertheless, this logical approach is not a case for this paper.

Type-reduction results in a type-1 fuzzy set, called a type-reduced set. Obviously, an interval type-2 fuzzy set after such reduction is characterized by a normal and rectangular membership function. Naturally, the next step needs a final defuzzification of the traditional type, which for the rectangle, requires only a trivial calculation of the average of the two bounds of the type-reduced set. This all together leads to an interval fuzzy structure presented in Figure 1.

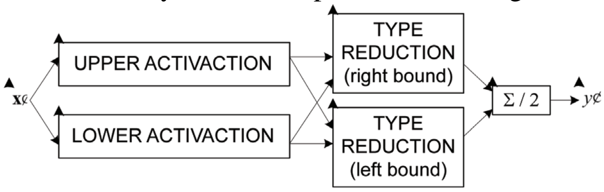

Figure 1. Interval type-2 fuzzy logic system

2.1 Type-reduction for interval type-2 sets

Formally, type-reduction is an application of an extension principle to the centroid defuzzification, or to its discrete form called a height defuzzification (in the case of singleton consequents). Let K denotes the number of consequents for the extended height defuzzification. Consequently, the centroid of an interval type-2 fuzzy set, given by its lower and upper memberships, $\underline{\mu}_k$ and $\overline{\mu}_k$, can be fuzzified via the extension principle, i.e.,

$$\mu(y) = \begin{cases} 1 & \text{if } y = \left.\frac{\sum_{k=1}^{K} y_k u_k}{\sum_{k=1}^{K} u_k}\right|_{u_k \in [\underline{\mu}_k, \overline{\mu}_k]}, \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where $k = 1, \ldots, K$.

To perform the optimization, assume that discrete points of the interval type-2 set to be defuzzified are arranged in the following order $y_1 < y_2 < \ldots, y_K$. In this case, the extended centroid is a normal and rectangular fuzzy set bounded in a classical interval, i.e. $[y_{\min}, y_{\max}]$. Generally, the bounds for this set can be expressed by

$$y_{\min} = \frac{\sum_{k=1}^{L-1} \overline{\mu}_k y_k + \sum_{k=L}^{K} \underline{\mu}_k y_k}{\sum_{k=1}^{L-1} \overline{\mu}_k + \sum_{k=L}^{K} \underline{\mu}_k}, \quad (9)$$

$$y_{\max} = \frac{\sum_{k=1}^{R} \underline{\mu}_k y_k + \sum_{k=R+1}^{K} \overline{\mu}_k y_k}{\sum_{k=1}^{R} \underline{\mu}_k + \sum_{k=R+1}^{K} \overline{\mu}_k}, \quad (10)$$

where L and R may be determined by the KM iterative procedure or other methods as e.g. smooth type-reduction. The calculation of L and R is a major computational problem for interval type-2 fuzzy logic systems.

The final defuzzification can be calculated formally by

$$y' = \frac{y_{\min} + y_{\max}}{2} \; . \tag{11}$$

2.2 KM type-reduction

While calculating the centroid given by (8), $y_{\max}$ in (10) is obtained by the maximization of centroid $\sum_{k=1}^{K} y_k u_k / \sum_{k=1}^{K} u_k$ and $y_{\min}$ in (9) can be obtained by the minimization of the same centroid, both in boundaries $u_k \in \left[\underline{\mu}_k, \overline{\mu}_k\right]$. Using differentiation,

$$\frac{\partial y(u_1,\ldots,u_k)}{\partial u_q} = \frac{y_q - y(u_1,\ldots,u_k)}{\sum_{k=1}^{K} u_k} \; . \tag{12}$$

From the necessary condition $\partial y / \partial u_q = 0$, we obtain

$$y_q \sum_{k=1}^{K} u_k = \sum_{k=1}^{K} y_k u_k \tag{13}$$

$$y_q \sum_{k=1,k\neq q}^{K} u_k = \sum_{k=1,k\neq q}^{K} y_k u_k \tag{14}$$

Since the necessary condition leads us to the equation that does not depend on u_q, inequality $y_q > y(u_1,\ldots,u_k)$ implies that y is increasing with u_q and $y_q < y(u_1,\ldots,u_k)$ implies that y is decreasing with u_q.

Within bounds $\left[\underline{\mu}_k, \overline{\mu}_k\right]$, y attains its maximum if

$$u_q = \begin{cases} \overline{\mu}_q & \text{for } y_q > y(u_1,\ldots,u_k) \\ \underline{\mu}_k & \text{for } y_q \leq y(u_1,\ldots,u_k) \end{cases}$$

and y attains its minimum if

$$u_q = \begin{cases} \overline{\mu}_q & \text{for } y_q < y(u_1,\ldots,u_k) \\ \underline{\mu}_k & \text{for } y_q \geq y(u_1,\ldots,u_k) \end{cases}$$

for all $q = 1,\ldots,K$.

Therefore, the KM type-reduction algorithm in its simplest form can be summarized as follows.

Algorithm 1. *Let the consequent values be aranged in the ascending order $y_1 < y_2 < \ldots < y_K$. and let us perform the following enumerations:*

1. *calculate type-1 system output y_0 as an average of y_k weighted by mean membership grades, i.e., $\left(\underline{\mu}_k + \overline{\mu}_k\right)/2$,*
2. *set the initial values $y_{\min} = y_{\max} = y_0$,*
3. *for each $k = 1,2,\ldots,K$, if $y_k > y_{\max}$, then $\overrightarrow{\mu}_k = \overline{\mu}_k$, otherwise $\overrightarrow{\mu}_k = \underline{\mu}_k$,*
4. *find the closest $y_{\text{next}} = \min_{k=1,\ldots,K} y_k : y_k > y_{\max}$,*
5. *calculate $y_{\max}$ as an average of y_k weighted by new grades $\overrightarrow{\mu}_k$,*
6. *if $y_{\max} \leq y_{\text{next}}$, continue, else go to step 3,*
7. *for each $k = 1,2,\ldots,K$, if $y_k < y_{\min}$, then $\underleftarrow{\mu}_k = \overline{\mu}_k$, otherwise $\underleftarrow{\mu}_k = \underline{\mu}_k$,*
8. *find the closest $y_{\text{next}} = \max_{k=1,\ldots,K} y_k : y_k < y_{\min}$,*
9. *calculate $y_{\min}$ as an average of y_k weighted by new grades $\underleftarrow{\mu}_k$,*
10. *if $y_{\min} \geq y_{\text{next}}$, finish, else go to step 7.*

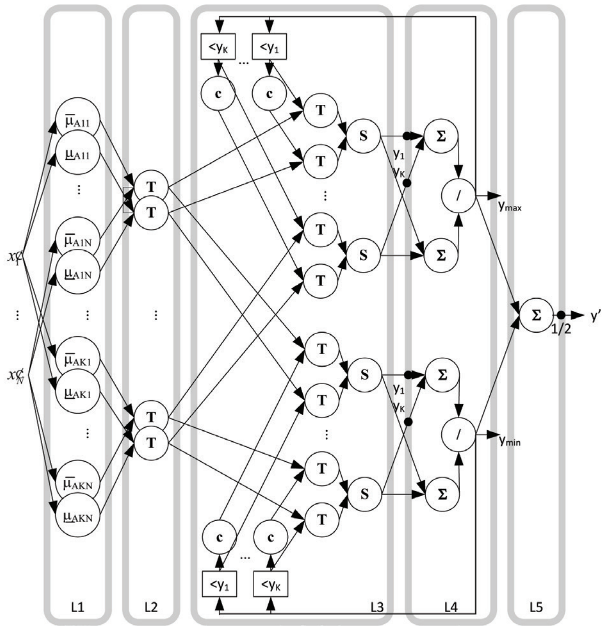

Figure 2. Adaptive interval type-2 fuzzy logic system using KM type-reduction

The direct application of Algorithm 1 leads to the structure of the adaptive interval type-2 fuzzy logic system, shown in Figure 2, which contains recursions between layers 4 and 3. This feature complicates the application of gradient descent techniques since the output errors are propagated backward only by the nodes selected with the $y_{\max}$ and $y_{\min}$ functions. Consequently, for each of the rules, either the parameters of the upper membership functions or of the lower membership functions are learned. This makes the learning process chaotic in time and poorly convergent. Thus, a natural remedy for this will be to use aggregate functions that depend on all the arguments, rather than functions depending on just the extreme argument, such as min and max. Therefore, we use the smooth maximum and smooth minimum functions to create a new type-reduction algorithm.

3 Smooth type-reduction

Suppose each embedded system has an output value υ_p, for $p = 1,\dots,P$. A smooth maximum of $\upsilon_1,\dots,\upsilon_P$ is a differentiable approximation to the maximum function with continuous derivatives. A universal smooth maximum/minimum function is formulated as

$$y_\alpha(\upsilon_1,\dots,\upsilon_P) = \frac{\sum_{p=1}^{P} \upsilon_p e^{\alpha\upsilon_p}}{\sum_{p=1}^{P} e^{\alpha\upsilon_p}} \quad (15)$$

which converges to maximum as parameter α tends to positive infinity, or to minimum as $\alpha \to -\infty$. Note that y_0 is the arithmetic mean of $\upsilon_1,\dots,\upsilon_P$. Consequently, $y_{-\infty}$ and y_∞ stand for the left $y_{\min}$ and the right $y_{\max}$ end points of the type-reduced set, respectively, while y_0 is the output of the type-0 fuzzy system embedded in the interval type-2 fuzzy system. However, the number of output values can be limited independently for $y_{\min}$ and $y_{\max}$, since not all tuples provide the optimal (maximal and minimal) output values. For instance, in the search for $y_{\max}$, only tuples with lower memberships for consequents not larger than y_0 should be taken into account. Moreover, having y_k arranged in ascending order, we initialize the algorithm with a binary word L....LU...U, where L and U stand for the lower and the upper memberships, respectively. The mask is implemented by a binary mask vector M_r, which is exploited in step 3 of Algorithm 1. Now, only right shifting operations can be performed to calculate the next output values υ_p, which can maximize the output. The operation has been inspired by a type of digital circuit called the shift register. The exemplary shifts in order to determine $y_{\max}$ are presented in Figure 3.

r\k	1	2	R	R+1	K-1	K
R	0	0	1	1	1	1
R+1	0	0	0	1	1	1
K-1	0	0	0	0	1	1
K	0	0	0	0	0	1

Figure 3. Right shifted masks for a membership selection between upper and lower memberships to calculate $y_{\max}$.

In the search for $y_{\min}$, similar mask $1\dots10\dots0$ initialize left shifting operations. Therefore, using the smooth extremum function (15), the proposed algorithm can be summarized as follows:

Algorithm 2. *Let the consequent values be arranged in the ascending order $y_1 < y_2 < \dots < y_K$ and the values in vector forms, i.e.,*

$$\mathbf{y} = [y_1,\dots,y_K]$$
$$\overline{\mu} = [\overline{\mu}_1,\dots,\overline{\mu}_K]$$
$$\underline{\mu} = \left[\underline{\mu}_1,\dots,\underline{\mu}_K\right]$$

To compute the right and the left endpoints of the type-reduced set, perform the following steps:

1. *calculate type-1 system output y_0 as an average of elements of* **y***weighted by mean membership grades, i.e., $\left(\underline{\mu}+\overline{\mu}\right)/2$,*
2. *find index R of the closest $y_R = \min_{k=1,\dots,K} y_k : y_k > y_0$,*
3. *for $r = R,\dots,K-1$:*
   a) *set a mask $M_r = \underset{1\ \dots}{0\dots0}\underset{R\ \dots\ K}{1\dots1}$,*
   b) *apply the mask to upper and lower memberships $\overrightarrow{\mu} = (1-M_r)\odot\underline{\mu} + M_r\odot\overline{\mu}$ (where $\odot$ is the Hadamard product),*
   c) *calculate $\overrightarrow{y}_r$ as an average of elements* **y** *weighted by $\overrightarrow{\mu}$,*
4. *return $y_{\max}$ as an aggregation of all $\overrightarrow{y}_r$ with the use of smooth maximum (15), $r = R,\dots,K-1$,*
5. *find index L of the closest $y_L = \min_{k=1,\dots,K} y_k : y_k < y_0$,*
6. *for $l = 2,\dots,L$:*

a) set a mask $M_l = \underset{1}{1}\dots\underset{L}{1}0\dots\underset{K}{0}$,

b) apply the mask to upper and lower memberships $\overleftarrow{\mu} = (1-M_l)\odot\underline{\mu}+M_l\odot\overline{\mu}$,

c) calculate $\overleftarrow{y}_l$ *as an average of elements* $\mathbf{y}$ *weighted by* $\overleftarrow{\mu}$,

7. *return* $y_{\min}$ *as an aggregation of all* $\overleftarrow{y}_l$ *with the use of smooth minimum given by (15),* $r = 2,\dots,L$.

From the machine learning point of view, a fundamental property of the smooth maximum function is the differentiability. The derivative of y_α can be easily calculated as:

$$\frac{\partial y_\alpha(\upsilon_1,\dots,\upsilon_P)}{\partial\upsilon_q} = \frac{e^{\alpha\upsilon_q}\left(1+\alpha\upsilon_q-\alpha y_\alpha(\upsilon_1,\dots,\upsilon_P)\right)}{\sum_{p=1}^{P}e^{\alpha\upsilon_p}} \tag{16}$$

Alternatively, smooth maximum can be defined by LogSumExp function, $LSE(\upsilon_1,\dots,\upsilon_P) = \frac{1}{\alpha}\log\sum_p\exp(\alpha\upsilon_p)$, or the p-norm, $\|(\upsilon_1,\dots,\upsilon_R)\|_p = \left(\sum_r|\upsilon_r|^p\right)^{\frac{1}{p}}$. However, when applied to the gradient machine learning, they give almost identical results.

4 A smooth interval type-2 fuzzy logic system

Using shifted mask vectors $\overrightarrow{\mathbf{M}}$ and $\overleftarrow{\mathbf{M}}$, the right end-point and the left boundary of the type-reduced set are calculated as

$$\begin{aligned}\overrightarrow{y}_i &= \frac{\sum_{k=1}^{K}\overrightarrow{\mu}_{i,k}y_k}{\sum_{k=1}^{K}\overrightarrow{\mu}_{i,k}}\\ &= \frac{\sum_{k=1}^{K}\left(\overline{\tau}_k\overrightarrow{M}_{i,k}+\underline{\tau}_k\left(1-\overrightarrow{M}_{i,k}\right)\right)y_k}{\sum_{k=1}^{K}\left(\overline{\tau}_k\overrightarrow{M}_{i,k}+\underline{\tau}_k\left(1-\overrightarrow{M}_{i,k}\right)\right)}\end{aligned} \tag{17}$$

$$\overleftarrow{y}_j = \frac{\sum_{k=1}^{K}\left(\overline{\tau}_k\overleftarrow{M}_{j,k}+\underline{\tau}_k\left(1-\overleftarrow{M}_{j,k}\right)\right)y_k}{\sum_{k=1}^{K}\left(\overline{\tau}_k\overleftarrow{M}_{j,k}+\underline{\tau}_k\left(1-\overleftarrow{M}_{j,k}\right)\right)} \tag{18}$$

where i and j indicate different shifted masks with respoect to steps 3 and 6 of Algorithm 2. The proposed structure of adaptive interval type-2 fuzzy system using smooth type-reduction is presented in Figure 4. Distinct masks for selection of upper/lower grades of rule firing result with several sublayers $3i$ and $4j$. The sublayers are then aggregated by the smooth max and the smooth min operations, i.e.,

$$y_{\max}(\overrightarrow{y}_1,\dots,\overrightarrow{y}_R) = \frac{\sum_{i=1}^{R}\overrightarrow{y}_i e^{\alpha\overrightarrow{y}_i}}{\sum_{i=1}^{R}e^{\alpha\overrightarrow{y}_i}} \tag{19}$$

$$y_{\min}(\overleftarrow{y}_1,\dots,\overleftarrow{y}_L) = \frac{\sum_{j=1}^{L}\overleftarrow{y}_j e^{-\alpha\overleftarrow{y}_j}}{\sum_{j=1}^{L}e^{-\alpha\overleftarrow{y}_j}} \tag{20}$$

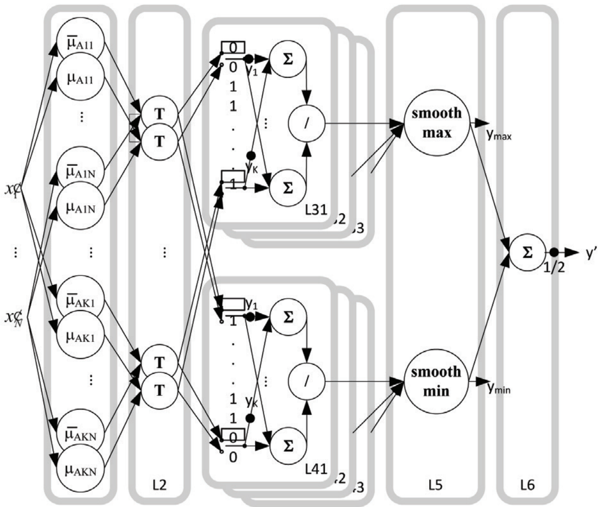

Figure 4. Adaptive interval type-2 fuzzy logic system using smooth type-reduction

In order to derive learning formulae, we can calculate partial derivatives of (17) and (18) with respect to upper and lower firing grades, i.e.,

$$\frac{\partial y_{\mathrm{r}}(i)}{\partial\overline{\tau}_q} = \frac{\overrightarrow{M}_{i,q}\left(y_q-y_{\mathrm{r}}(i)\right)}{\sum_{k=1}^{K}\left(\overline{\tau}_k\overrightarrow{M}_{i,k}+\underline{\tau}_k\left(1-\overrightarrow{M}_{i,k}\right)\right)} \tag{21}$$

$$\frac{\partial y_{\mathrm{r}}(i)}{\partial\underline{\tau}_q} = \frac{\left(1-\overrightarrow{M}_{i,q}\right)\left(y_q-y_{\mathrm{r}}(i)\right)}{\sum_{k=1}^{K}\left(\overline{\tau}_k\overrightarrow{M}_{i,k}+\underline{\tau}_k\left(1-\overrightarrow{M}_{i,k}\right)\right)} \tag{22}$$

$$\frac{\partial y_{\mathrm{l}}(j)}{\partial\overline{\tau}_q} = \frac{\overleftarrow{M}_q\left(y_q-y_{\mathrm{l}}(j)\right)}{\sum_{k=1}^{K}\left(\overline{\tau}_k\overleftarrow{M}_k+\underline{\tau}_k\left(1-\overleftarrow{M}_k\right)\right)} \tag{23}$$

$$\frac{\partial y_{\mathrm{l}}(j)}{\partial\underline{\tau}_q} = \frac{\left(1-\overleftarrow{M}_q\right)\left(y_q-y_{\mathrm{left}}(j)\right)}{\sum_{k=1}^{K}\left(\overline{\tau}_k\overleftarrow{M}_k+\underline{\tau}_k\left(1-\overleftarrow{M}_k\right)\right)} \tag{24}$$

The gradient of smooth maximum (analogously smooth minimum) is calculated using (16), i.e.,

$$\nabla_{\overrightarrow{y}_q}y_{\max}(\overrightarrow{y}_1,\dots,\overrightarrow{y}_R) = \frac{e^{\alpha\overrightarrow{y}_q}\left(1+\alpha\overrightarrow{y}_q-\alpha y_{\max}\right)}{\sum_{i=1}^{R}e^{\alpha\overrightarrow{y}_i}} \tag{25}$$

$$\nabla_{\overleftarrow{y}_q}y_{\min}(\overleftarrow{y}_1,\dots,\overleftarrow{y}_L) = \frac{e^{-\alpha\overleftarrow{y}_q}\left(1-\alpha\overleftarrow{y}_q+\alpha y_{\min}\right)}{\sum_{i=1}^{L}e^{-\alpha\overleftarrow{y}_i}} \tag{26}$$

The rest of the derivations of the gradient descent algorithm relies on the composition of partial derivatives to minimize the following square error as the cost function

$$Q = \frac{1}{2}\left(d - y'\right)^2 \tag{27}$$

5 Experimental results

In this paper, the interval type-2 fuzzy logic system with smooth type-reduction is compared with the type-2 fuzzy system based on KM type-reduction and the type-1 FLS. We demonstrate performance on two regression datasets: Nonlinear Dynamic Plant and Kinematics. As it is the first demonstration of the smooth interval type-2 FLS, we use the standard back propagation method with first-order derivatives of the square error propagated backward. The root mean square error measures and compares the performance of simulated systems

$$RMSE = \sqrt{\frac{\sum_{p=1}^{P} Q_p}{P}} \tag{28}$$

which is calculated as a mean value over P patterns in each fold of cross-validation, and then averaged over all folds.

In all experiments, the errors were measured as testing results using 10-folds cross-validation. Whole datasets were used in 50 learning epochs, after which the errors were stabilized and the testing errors were still close to the training errors. The learning hyperparameters of the systems were tuned to achieve the best performance. Each hyperparmeter was estimated independently using cross-validation. The final set of hyperparameters was chosen as: $\eta_y = 0.1$, $\eta_\sigma = \eta_m = \eta_\theta = 0.05$, and the optimal smoothness parameter was either $\alpha = 100$ in the case of smooth maximum or $\alpha = -100$ in the case of smooth minimum. The choice of the alpha parameter was a compromise between the accuracy of the approximation of the maximum function and the value of the derivative of this function with respect to submaximal input variables. It is worth noting that the derivative of the smooth maximum function with respect to υ_q given by (16), however, related to the known soft-max function, is dependent on the α times difference between the function's output and υ_q.

The Nonlinear Dynamic Plant is an original dataset studied by Wang [20]. It consists of 400 samples generated by the second-order difference equation:

$$y(t) = g\left(y(t-1), y(t-2)\right) + u(t), \tag{29}$$

where

$$g\left(y(t-1), y(t-2)\right) = \frac{y(t-1)\, y(t-2)\left(y(t-1) - \frac{1}{2}\right)}{1 + y^2(t-1) + y^2(t-2)} \tag{30}$$

Starting from the equilibrium state $(0,0)$, 200 samples were obtained for a random input u uniformly distributed in $[-1.5, 1.5]$, and the latter samples were collected with a sinusoidal input signal given, i.e., $u(t) = \sin(2\pi t/25)$.

The result of comparing systems with 6 rules for the original data and with the noise of individual inputs is presented in Table 1. In order to test the systems' performance under conditions of greater uncertainty of data, we added random values from the uniform distribution to the input variables. A parameter Δ_i corresponds to the width of the probability distribution in relation to the width of the domain of the input variable. The noise applied to all inputs was intended to test the benefits of using Type-2 systems with all rules being uniformly uncertain.

Table 1. Nonlinear Dynamic Plant approximation with optional uniform disturbance applied to inputs (T1 stands for the type-1 FLS, KM-IT2 for interval type-2 FLS, and S-IT2 for smooth interval type-2 FLS)

	T1 RMSE	KM-IT2 RMSE	S-IT2 RMSE
	original dataset		
	0.1066	0.1060	**0.0943**
Δ_1	noised 1st input		
0.1	0.1460	0.1249	**0.1095**
0.5	0.2534	0.2427	**0.2407**
Δ_2	noised 2nd input		
0.1	0.1514	0.1355	**0.1308**
0.5	0.2641	0.2305	**0.2131**
$\Delta_{1,2}$	noised all inputs		
0.1	0.2380	0.2065	**0.1858**
0.5	0.2907	0.2741	**0.2721**

It can be observed that smooth type-reduction decreases the RMSE in all cases. The difference between the performance of the type-1 FLS, the standard interval type-2 fuzzy system, and the proposed smooth interval type-2 FLS is especially noticeable for the original data, however, with noisy data, the smooth system still performs better.

The Kinematics data set is taken from the Data for Evaluating Learning in Valid Experiments (DELVE) repository of the University of Toronto, and is concerned with the forward kinematics of an 8 link all-revolute robot arm. The desired output presents predictions of the distance between the end-effector and a target. We have selected the 8nm data variant with 8192 instances, which is highly non-linear and medium noisy.

The simulation results of training systems with 13 fuzzy rules (type-1 and type-2) are summarized in Table 2. The obtained RMSE confirm that the smooth type-reduction increases the accuracy of regression.

Table 2. Kinematics predictions with optional uniform disturbance applied to all inputs (T1 stands for the type-1 FLS, KM-IT2 for interval type-2 FLS, and S-IT2 for smooth interval type-2 FLS)

	IT1 RMSE	KM-IT2 RMSE	S-IT2 RMSE
	original dataset		
	0.1947	0.1903	**0.1911**
Δ_n	noised all inputs		
0.1	0.2447	0.2583	**0.2379**
0.5	0.3532	0.3497	**0.3368**

6 Conclusions

In this paper, we demonstrated that there is a differentiable type-reduction method, which serves up better training in adaptive interval type-2 fuzzy logic systems. Compared to the type-2 interval system based on KM type-reduction, the system based on smooth type-reduction shows non-chaotic learning processes and achieves much lower training and testing error values. Both type-2 fuzzy systems significantly exceed the learning ability of the type-1 fuzzy system.

The proposed system is a good approach to solving problems with increased model uncertainty or with uncertain measurements of the input data. In cases where we have even a small portion of undistorted training data, we perform the initial training of type-2 systems by treating them as type-1 fuzzy systems, and then we use the method of generating type-2 fuzzy rules for uncertain data using the fuzzy-rough approximation [12, 18, 17] or possibilistic fuzzification [16, 13]. So we reaffirm that type-2 fuzzy systems are key to extracting explanatory fuzzy rules, especially in the case of uncertainty or even ambivalency of these rules.

In the case of many rules, the structure of the fuzzy system with smooth type-reduction has many layers to be aggregated by the smooth maximum or smooth minimum functions, hence some similarities to deep neural networks are not unintended. The model complexity reduction methods, developed e.g. for convolutional neural networks, will be adapted to fuzzy type-2 smooth systems in our future work.

Acknowledgments

The project financed under the program of the Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019 - 2022 project number 020/RID/2018/19, the amount of financing 12,000,000 PLN.

References

[1] Bilski, J., Kowalczyk, B., Marchlewska, A., and Zurada, J. (2020). Local Levenberg-Marquardt algorithm for learning feedforward neural networks. Journal of Artificial Intelligence and Soft Computing Research, 10(4):299–316.

[2] Bilski, J., Kowalczyk, B., Marjański, A., Gandor, M., and Zurada, J. (2021). A novel fast feedforward neural networks training algorithm. Journal of Artificial Intelligence and Soft Computing Research, 11(4):287–306.

[3] Bilski, J. and Smoląg, J. (2020). Fast conjugate gradient algorithm for feedforward neural networks. In Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., and Zurada, J. M., editors, Artificial Intelligence and Soft Computing, pages 27–38, Cham. Springer International Publishing.

[4] Chen, Y. and Wang, D. (2018). Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted enhanced Karnik–Mendel algorithms. Soft Computing, 22(4):1361–1380.

[5] De Magistris, G., Russo, S., Roma, P., Starczewski, J. T., and Napoli, C. (2022). An explainable fake news detector based on named entity recognition and stance classification applied to covid-19. Information, 13(3):137.

[6] El-Nagar, A. M. and El-Bardini, M. (2014). Simplified interval type-2 fuzzy logic system based on new type-reduction. Journal of Intelligent & Fuzzy Systems, 27(4):1999–2010.

[7] Karnik, N. N., Mendel, J. M., and Liang, Q. (1999). Type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems, 7(6):643–658.

[8] Liang, Q. and Mendel, J. M. (2000). Interval type-2 fuzzy logic systems: Theory and design. IEEE Transactions on Fuzzy Systems, 8:535–550.

[9] Maowen Nie and Woei Wan Tan (2008). Towards an efficient type-reduction method for interval type-2 fuzzy logic systems. In 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), pages 1425–1432.

[10] Melgarejo, M. (2007). A fast recursive method to compute the generalized centroid of an interval type-2 fuzzy set. In Proc. NAFIPS 2007, pages 190–194.

[11] Mendel, J. M. (2017). Uncertain rule-based fuzzy systems. Introduction and new directions, page 684.

[12] Nowicki, R. K. and Starczewski, J. T. (2017). A new method for classification of imprecise data using fuzzy rough fuzzification. Inf. Sci., 414:33–52.

[13] Nowicki, R. K., Starczewski, J. T., and Grycuk, R. (2019). Extended possibilistic fuzzification for classification. In Guervós, J. J. M., Garibaldi, J., Linares-Barranco, A., Madani, K., and Warwick, K., editors, Proceedings of the 11th International Joint Conference on Computational Intelligence, IJCCI 2019, Vienna, Austria, September 17-19, 2019, pages 343–350. ScitePress.

[14] Rojas, J. D., Salazar, O., and Serrano, H. (2016). Nie-Tan Method and its Improved Version: A Counterexample. IngenierÃa, 21:138 – 153.

[15] Sepulveda, R., Castillo, O., Melin, P., and Montiel, O. (2007). An efficient computational method to implement type-2 fuzzy logic in control applications. In Melin, P. and et al., editors, Analysis and Design of Intelligent Systems using Soft Computing Techniques, volume 41, chapter 5, pages 45–52. Springer-Verlag, Germany, 1 edition.

[16] Starczewski, J. T. (2013). Advanced Concepts in Fuzzy Logic and Systems with Membership Uncertainty, volume 284 of Studies in Fuzziness and Soft Computing. Springer.

[17] Starczewski, J. T., Goetzen, P., and Napoli, C. (2020). Triangular fuzzy-rough set based fuzzification of fuzzy rule-based systems. Journal of Artificial Intelligence and Soft Computing Research, 10.

[18] Starczewski, J. T., Nowicki, R. K., and Nieszporek, K. (2019). Fuzzy-rough fuzzification in general FL classifiers. In Guervós, J. J. M., Garibaldi, J., Linares-Barranco, A., Madani, K., and Warwick, K., editors, Proceedings of the 11th International Joint Conference on Computational Intelligence, IJCCI 2019, Vienna, Austria, September 17-19, 2019, pages 335–342. ScitePress.

[19] Staszewski, P., Jaworski, M., Rutkowski, L., and Tao, D. (2020). Explainable cluster-based rules generation for image retrieval and classification. In Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., and Zurada, J. M., editors, Artificial Intelligence and Soft Computing, pages 85–94, Cham. Springer International Publishing.

[20] Wang, L. and Yen, J. (1999). Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and kalman filter. Fuzzy Sets and Systems, 101:353–362.

[21] Wu, D. and Mendel, J. M. (2009). Enhanced karnik-mendel algorithms. IEEE Transactions on Fuzzy Systems, 17(4):923–934.

[22] Wu, D. and Tan, W. (2005). Computationally efficient type-reduction strategies for a type-2 fuzzy logic controller. In Proc. IEEE Fuzzy Conference, pages 353–358, Reno, NV.

[23] Zadeh, L. A. (1975). The concept of a linguistic variable and its application to approximate reasoning — I. Information Sciences, 8:199–249.

Janusz T. Starczewski is an Associate Professor with the Department of Intelligent Computer Systems at Czestochowa University of Technology, where he is Head of education for discipline of information and communication technology. He holds the Ph.D and D.Sc. degrees in Computer Science and an M.Sc. in Electrical Engineering. He is an experienced project contributor in artificial intelligence and IT systems. The mainstream of his scientific achievements comprise studies on advanced concepts of fuzzy logic, including type-2 fuzzy logic systems and their combinations with the rough set theory. He has authored more than 50 publications. His book ”Advanced Concepts in Fuzzy Logic and Systems with Membership Uncertainty” has been granted by the Polish Minister of Science and Higher Education.

Krzysztof Przybyszewski is a professor at the University of Social Sciences in Łódź. His adventure with applied computer science began in the 1980s with a simulation of non-quantum collective processes (the subject of a Ph.D. dissertation). At present, he is involved in research and applications of various artificial intelligence technologies and soft computing methods in selected IT problems (in particular, in expert systems supporting the management of education quality in universities - the use of fuzzy numbers and sets). As a deputy dean at the University of Social Sciences, he is the designer and organizer of the on-Computer Science Faculty education program. He is the author of over 80 publications in the field of computer science and IT applications.

Aleksander Byrski obtained Ph.D. in 2007 and D.Sc. in 2013 at the Department of Computer Science of the AGH University of Science and Technology in Krakow, Poland. His main research interests are metaheuristics, agentbased systems, high performance computing and simulation. He works as a Full Professor in the Institute of Computer Science at the AGH University of Science and Technology.

Eulalia Szmidt is a Full Professor of Computer Science at the Systems Research Institute Polish Academy of Sciences, and WIT - Warsaw School of Information Technology, in Warsaw, Poland. She has a M.Sc and Ph.D. in automatic control and computer science from the Warsaw University of Technology, MBA in management and marketing from the University of Illinois at Urbana-Champaign, and D.Sc. in artificial intelligence from the Bulgarian Academy of Sciences.

Her main interests concern representation and processing of imprecise information, fuzzy sets, intuitionistic fuzzy sets, artificial intelligence, soft computing. She is a co-editor of many volumes, and an (co)author of 200 papers published in the international journals, edited volumes and proceedings of prestigious national and international conferences. She has been a member of many program committees at national and international congresses, conferences and workshops, local co-chair of 19 scientific meetings, and a reviewer for many international journals. She is an IFSA Fellow and a member of the EUSFLAT Board.

Christian Napoli is Associate Professor with the Department of Computer, Control, and Management Engineering “Antonio Ruberti”, Sapienza University of Rome, since 2019, where he also collaborates with the department of Physics and the Faculty of Medicine and Psychology, as well as holding the office of Scientific Director of the International School of Advanced and Applied Computing (ISAAC).

He received the B.Sc. degree in Physics from the Department of Physics and Astronomy, University of Catania, in 2010, where he also got the M.Sc. degree in Astrophysics in 2012 and the Ph.D. in Computer Science in 2016 at the Department of Mathematics and Computer Science, he obtained the National Scientific Abilitation as associate professor in Computer Engineering (2017) and computer science (2019).

Christian Napoli has been Research Associate with the Department of Mathematics and Computer Science, University of Catania, from 2018 to 2019, while, previously, Research Fellow and Adjunct Professor with the same department from 2015 to 2018. He has been a Student Research Fellow with the Department of Electrical, Electronics, and Informatics Engineering, University of Catania, from 2009 to 2016, a collaborator of the Astrophysical Observatory of Catania and the National Institute for Nuclear Physics, since 2010.

He has been several time Invited Professor at the Silesian University of Technology, Visiting Academic at the New York University, and responsible of many different institutional topics from 2011 until now for Undegraduate, Graduate and PhD students in Computer Science, Computer Engineering and Electronics Engineering. His teaching activity focused on Artificial Intelligence, Neural Networks, Machine Learning, Computing Systems, Computer Architectures, Distributed Systems, and High Performance Computing. He is involved in several international research projects, serves as reviewer and member of the board program committee for major international journals and international conferences. His current research interests include neural networks, artificial intelligence, human-computer interation and computational neuropsychology.