Knowledge-Based Systems 210 (2020) 106530

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

How to implement MCDM tools and continuous logic into neural computation?

Towards better interpretability of neural networks

Orsolya Csiszár [a,b,*], Gábor Csiszár [c], József Dombi [d]

[a] Institute of Applied Mathematics, Óbuda University, Budapest, Hungary
[b] Faculty of Basic Sciences, University of Applied Sciences Esslingen, Esslingen, Germany
[c] Department of Materials Physics, Institute of Materials Science, University of Stuttgart, Stuttgart, Germany
[d] Institute of Informatics, University of Szeged, Szeged, Hungary

ARTICLE INFO

Article history:
Received 4 June 2020
Received in revised form 25 September 2020
Accepted 12 October 2020
Available online 15 October 2020

Keywords:
Multi-criteria decision-making
Preference modeling
Outranking methods
Fuzzy systems
Continuous logic
Nilpotent logical systems
Neural networks

ABSTRACT

The theories of multi-criteria decision-making (MCDM) and fuzzy logic both aim to model human thinking. In MCDM, aggregation processes and preference modeling play the central role. This paper suggests a consistent framework for modeling human thinking by using the tools of both fields: fuzzy logical operators as well as aggregation and preference operators. In this framework, aggregation, preference, and the logical operators are described by the same unary generator function. Similarly to the implication being defined as a composition of the disjunction and the negation operator, preference operators were introduced as a composition of the aggregative operator and the negation operator. After a profound examination of the main properties of the preference operator, our main goal is the implementation into neural networks. We show how preference can be modeled by a perceptron, and illustrate the results in practical neural applications.

1. Introduction

When it comes to modeling human thinking, two main approaches have received particular attention in the last decades: fuzzy logic and multi-criteria decision analysis (MCDA), or multi-criteria decision-making (MCDM). In real-world applications, a decision-maker, more often than not, faces decision situations where multiple criteria have to be considered simultaneously.

Since the modeling is always affected by the presence of different kinds of uncertainty due to the imperfect human knowledge, fuzzy set theory, as a language that is capable to deal with uncertainty has been successful also in MCDM models.

Fuzzy sets provide a theoretical framework to quantify a type of uncertainty, such as imprecision and ambiguity that is inherent in many decision-making processes. The seminal paper by Orlovsky [1] can be considered as the first attempt to use fuzzy set theory in preference modeling. In his paper, Orlovsky defines the strict preference relation and the indifference relation with the use of Łukasiewicz- and minimum t-norms. As a result, numerous approaches have been proposed to solve fuzzy MCDM problems [2]. A review and comparison of many of these methods can be found in [3].

In multiple criteria decision-making (MCDM), the decision-maker's preference plays a key role (see e.g. [4–6], for a comprehensive taxonomy of the MCDA process characteristics, see [7]), and therefore, preference modeling is fundamental. The classical MCDM procedures perform generally in two steps; aggregation and exploitation. First, the aggregation part defines an outranking relation that indicates the global preference between any ordered pair of alternatives. Second, the exploitation transforms the information into a global ranking, usually by using a ranking method to obtain a score function, like in classical procedures typical of the so-called European (or French) School, such as PROMETHEE [8] and ELECTRE III [9].

The preference operators of these outranking methods can be described well by the generator-based preference operator introduced in this article. The aggregation procedures in decision-making often use value functions or preference relations. In the classical theory, preference is a binary relation with the semantical meaning of

$$p(x, y) = truth\ of (x \leq y).$$

In order to deal with the preference operator and the logical operators in a consistent framework, the in- and output values need to

* Corresponding author at: Institute of Applied Mathematics, Óbuda University, Budapest, Hungary.
E-mail addresses: csiszar.orsolya@nik.uni-obuda.hu, ocsiszar@hs-esslingen.de (O. Csiszár), Gabor.Csiszar@mp.imw.uni-stuttgart.de (G. Csiszár), dombi@inf.u-szeged.hu (J. Dombi).

https://doi.org/10.1016/j.knosys.2020.106530
0950-7051/

be normalized. As we can easily see, the preference operator does not belong to the logical operators, since logical operators need to be consistent with the classical logic; i.e. on the boundaries we need to get crisp values from $\{0, 1\}$. However, if the two input values are the same, preference operators should give a neutral output value (greatest uncertainty level); i.e. different from 0 and 1. This means that for $(0, 0)$ and for $(1, 1)$, preference operators cannot have a crisp value from $\{0, 1\}$, and therefore they do not belong to the world of logical operators in the strict sense. In this work, we propose a suggestion on how we can still create a theoretical framework synthesizing the worlds of continuous logic and MCDM, and examine the main properties of the preference operator in nilpotent systems. This consistent framework supports a potential application of the results in the field of artificial intelligence, as an important step towards the interpretability of neural models.

Recently, intelligent learning methods, especially deep learning models have been revolutionizing the business and technology world. One of the greatest challenges is the increasing need for interpretability, transparency, and safety. Although deep neural networks have achieved impressive experimental results, they may surprisingly be unstable when it comes to adversarial perturbations. For example in image classification, minimal changes to the input image may cause the network to misclassify it [10–13]. In predictive modeling, interpretability (opening the "black box") becomes more and more important. In a high-risk environment, we also need to know the reasons why a decision was made.

Neural models have also been developed for multiple criteria decision-making [14,15]. In these models, the motivation is to model the decision-maker's underlying preference structures by means of supervised learning based on sampled preference data. The recent advances in theory and methodology of neural networks and fuzzy logic have laid a solid basis for developing models based on neural architecture for MCDM in a fuzzy environment. On the one hand, fuzzy systems can deal with uncertainty and linguistic terms, modeling the decision-maker's preferences using fuzzy rules. On the other hand, neural networks can exhibit learning capability. In this direction, Preference Learning (PL) is emerging as an extended paradigm in machine learning by inducing predictive preference models from experimental data [16–19]. PL involves in various research fields such as knowledge discovery or recommender systems.

Although combinations of neural networks and MCDM have been considered in different contexts, there has been little attempt to combine neural networks with continuous logical systems so far. The novelty of this paper is to suggest a consistent framework for modeling human thinking by using the tools of all three fields: fuzzy logical operators, MCDM tools (such as aggregation and preference operators), as well as deep learning methods. Beyond the theoretical demand, our objective here is to provide multicriteria decision tools to the nilpotent neural model introduced in [20].

The article is organized as follows. In Section 2, we recall some basic preliminaries regarding nilpotent systems. Section 3 introduces the problem of preference modeling in these systems and suggests a definition for the preference operator combining the aggregative operator with the negation operator of the system. The main properties of the preference operator are examined in Section 4. In Sections 5 and 6, we show how the nilpotent preference can be modeled by a perceptron and illustrate this result in applications in neural networks. To obtain differentiability, the squashing function as a smooth approximation of the cutting function is used in the formulae. The results are summarized in Section 7.

2. Operators of nilpotent systems - a general framework

First, we highlight some of our related preliminary results. Among other families of fuzzy logics, nilpotent fuzzy logic is beneficial from several perspectives. The fulfillment of the law of contradiction and the excluded middle, and the coincidence of the residual and the S-implication [21,22] make the application of nilpotent operators in logical systems promising. In [23–28], an abundant asset of operators was examined thoroughly: in [24], negations, conjunctions and disjunctions, in [25] implications, and in [26] equivalence operators. In [27], the aggregative operators were studied and a parametric form of a general operator o_ν was given by using a shifting transformation of the generator function. Varying the parameters, nilpotent conjunctive, disjunctive, aggregative (where a high input can compensate for a lower one) and negation operators can all be obtained. It was also demonstrated how the nilpotent generated operator can be applied for preference modeling. Moreover, as shown in [28], membership functions, which play a substantial role in the overall performance of fuzzy representation, can also be defined using a generator function. In [20], the authors showed that in the field of continuous logic, nilpotent logical systems are the most suitable for neural computation.

2.1. Normalization of the generator functions

Let us first consider the most important operators in classical logic are the conjunction, the disjunction, and the negation operator. These three basic operators together form a so-called connective system. When extending classical logic to continuous logic, compatibility and consistency are crucial. The negation should also be involutive; i.e. $n(n(x)) = x$, for $\forall x \in [0, 1]$. Involutive negations are called strong negations.

Definition 1 (*[24]*)**.** The triple (c, d, n), where c is a t-norm, d is a t-conorm and n is a strong negation, is called a connective system.

Definition 2 (*[24]*)**.** A connective system is nilpotent if the conjunction c is a nilpotent t-norm, and the disjunction d is a nilpotent t-conorm.

In the nilpotent case, the generator functions of the disjunction, and the conjunction (denoted by $t(x)$ and $s(x)$ respectively) are bounded functions, being determined up to a multiplicative constant. This means that they can be normalized the following way:

$$f_c(x) := \frac{t(x)}{t(0)}, \qquad f_d(x) := \frac{s(x)}{s(1)}. \tag{1}$$

Note that the normalized generator functions are now uniquely defined. Next, we recall the definition of the cutting function, to simplify the notations used. The differentiable approximation of the cutting function, the squashing function $S(x)$ introduced and examined in [29], is a ReLu-like bounded activation function in our model. In [27], the authors showed that all the nilpotent operators can be described by using one generator function $f(x)$ and the cutting function.

Definition 3 (*[24]*)**.** Let us define the cutting operation $[\ ]$ by

$$[x] = \begin{cases} 0 & \text{if} \quad x < 0 \\ x & \text{if} \quad 0 \le x \le 1 \\ 1 & \text{if} \quad 1 < x \end{cases}$$

Proposition 1 (*[24]*)**.** *With the help of the cutting operator, we can write the conjunction and disjunction in the following form, where* f_c

and f_d are decreasing and increasing normalized generator functions respectively.

$$c(x, y) = f_c^{-1}[f_c(x) + f_c(y)], \tag{2}$$

$$d(x, y) = f_d^{-1}[f_d(x) + f_d(y)]. \tag{3}$$

Remark 1. For the natural negations to coincide, as shown in [24], $f_c(x)+f_d(x) = 1$ must hold for $\forall x \in [0, 1]$, which means that only one generator function, e.g. $f_d(x)$ is needed to describe the operators. Henceforth, f_d is represented by $f(x)$.

Remark 2. Note that the min and max operators (often used as conjunction, and disjunction in applications) can also be expressed by [ ] in the following way:

$$\min(x, y) = [x + [y - x + 1] - 1], \tag{4}$$

$$\max(x, y) = [x + [y - x]], \tag{5}$$

where $x, y \in [0, 1]$.

The associativity of t-norms and t-conorms permits us to consider their extensions to the multivariable case.

2.2. The general parametric operator

In [27], the authors also examined a general parametric operator $o_\nu(\underline{x})$ of nilpotent systems.

Definition 4 (*[27]*). Let $f : [0, 1] \to [0, 1]$ be an increasing bijection, $\nu \in [0, 1]$, and $\underline{x} = (x_1, \ldots, x_n)$, where $x_i \in [0, 1]$ and let us define the general operator by

$$o_\nu(\underline{x}) = f^{-1}\left[\sum_{i=1}^{n} (f(x_i) - f(\nu)) + f(\nu)\right] = f^{-1}\left[\sum_{i=1}^{n} f(x_i) - (n-1)f(\nu)\right]. \tag{6}$$

Remark 3. Note that the general operator for $\nu = 1$ is conjunctive, for $\nu = 0$ it is disjunctive and for $\nu = \nu_* = f^{-1}\left(\frac{1}{2}\right)$ it is self-dual.

On the basis of Remark 3, the conjunction, the disjunction and the aggregative operator can be defined in the following way:

Definition 5 (*[27]*). Let $f : [0, 1] \to [0, 1]$ be an increasing bijection, $\underline{x} = (x_1, \ldots, x_n)$, where $x_i \in [0, 1]$. Let us define the conjunction, the disjunction and the aggregative operator by

$$c(\underline{x}) := o_1(\underline{x}) = f^{-1}\left[\sum_{i=1}^{n} f(x_i) - (n-1)\right], \tag{7}$$

$$d(\underline{x}) := o_0(\underline{x}) = f^{-1}\left[\sum_{i=1}^{n} f(x_i)\right], \tag{8}$$

$$a(\underline{x}) := o_{\nu_*}(\underline{x}) = f^{-1}\left[\sum_{i=1}^{n} f(x_i) - \frac{1}{2}(n-1)\right], \tag{9}$$

respectively, where $\nu_* = f^{-1}\left(\frac{1}{2}\right)$.

A conjunction, a disjunction and an aggregative operator differ only in one parameter of the general operator in (6). The parameter ν has the semantic meaning of the level of expectation: maximal for the conjunction, neutral for the aggregation, and minimal for the disjunction.

Next, let us recall the weighted form of the general operator:

Definition 6 (*[27]*). Let $\underline{w} \in \mathbb{R}^n, w_i > 0, f : [0, 1] \to [0, 1]$ an increasing bijection with $\nu \in [0, 1]$, $\underline{x} = (x_1, \ldots, x_n)$, where $x_i \in [0, 1]$. The weighted general operator is defined by

$$o_{\nu,\underline{w}}(\underline{x}) := f^{-1}\left[\sum_{i=1}^{n} w_i(f(x_i) - f(\nu)) + f(\nu)\right]. \tag{10}$$

Note that if the weight vector is normalized; i.e. for $\sum_{i=1}^{n} w_i = 1$,

$$o_{\nu,\underline{w}}(\underline{x}) = f^{-1}\left(\sum_{i=1}^{n} w_i f(x_i)\right). \tag{11}$$

For future application, we introduce a threshold-based operator in the following way.

Definition 7 (*[27]*). Let $\underline{w} \in \mathbb{R}^n$, $w_i > 0$, $\underline{x} = (x_1, \ldots, x_n) \in [0, 1]^n$, $\underline{\nu} = (\nu_1, \ldots \nu_n) \in [0, 1]^n$ and let $f : [0, 1] \to [0, 1]$ be a strictly increasing bijection. Let us define the threshold-based nilpotent operator by

$$o_{\underline{\nu},\underline{w}}(\underline{x}) = f^{-1}\left[\sum_{i=1}^{n} w_i\,(f(x_i) - f\,(\nu_i)) + f(\nu)\right] =$$

$$= f^{-1}\left[\sum_{i=1}^{n} w_i f(x_i) + C\right], \tag{12}$$

where

$$C = f(\nu) - \sum_{i=1}^{n} w_i f(\nu_i). \tag{13}$$

Remark 1. Note that the Equation in (12) describes the perceptron model in neural computation. Here, the parameters all have semantic meanings as importance (weights), decision level and level of expectancy.

The most commonly used operators for $n = 2$ and for special values of w_i and C, also for $f(x) = x$, are listed in Table 1.

2.3. The unary operators: negation, modifiers and hedges

Now let us focus on the unary (1-variable) case, examined in [28], which also plays a crucial role in the nilpotent neural model. The unary operators are mainly used to construct modifiers and membership functions by using a generator function. The membership functions can be interpreted as modeling an inequality [30]. Note that non-symmetrical membership functions can also be constructed by connecting two unary operators with a conjunction [23,28]. For the most important unary operators see Table 2.

Definition 8 (*[28]*). Let $\lambda \in \mathbb{R}^+, \lambda > 1, \nu \in [0, 1], f : [0, 1] \to [0, 1]$ be an increasing bijection. Let us define the unary operator $\tau_\nu^{(\lambda)}(x)$ in the following way.

$$\tau_\nu^{(\lambda)}(x) := f^{-1}\left[\lambda\,(f(x) - f(\nu)) + f(\nu)\right]. \tag{14}$$

Remark 2. For $\nu = 1$, $\nu = 0$ and $\nu = \nu_*$ (i.e. $f(\nu) = \frac{1}{2}$), we get the necessity, the possibility and the sharpness operators, respectively.

3. Preference modeling

Preference modeling is an inevitable part of several applied fields of decision-making and at the same time, it has its own

Table 1
The most important two-variable operators $o_{\underline{w}}(\underline{x})$.

	w_1	w_2	C	$o_{\underline{w}}(x,y)$	for $f(x)=x$	Notation
Logical operators						
Disjunction	1	1	0	$f^{-1}[f(x)+f(y)]$	$[x+y]$	$d(x,y)$
Conjunction	1	1	-1	$f^{-1}[f(x)+f(y)-1]$	$[x+y-1]$	$c(x,y)$
Implication	-1	1	1	$f^{-1}[f(y)-f(x)+1]$	$[y-x+1]$	$i(x,y)$
Multi-criteria Decision Tools						
Arithmetic mean	0.5	0.5	0	$f^{-1}\left[\frac{1}{2}\left(f(x)+f(y)\right)\right]$	$\frac{1}{2}(x+y)$	$m(x,y)$
Preference	-0.5	0.5	0.5	$f^{-1}\left[\frac{1}{2}\left(f(y)-f(x)+1\right)\right]$	$\frac{1}{2}\left(y-x+1\right)$	$p(x,y)$
Aggregative operator	1	1	-0.5	$f^{-1}\left[f(x)+f(y)-\frac{1}{2}\right]$	$\left[x+y-\frac{1}{2}\right]$	$a(x,y)$

Table 2
The most important unary operators $\tau_{\nu}^{(\lambda)}(x)$.

	ν	$\tau_{\nu}^{(\lambda)}(x)$	for $f(x)=x$	Notation
Possibility	1	$f^{-1}[\lambda f(x)]$	$[\lambda x]$	$\tau_P(x)$
Necessity	0	$f^{-1}[\lambda f(x)-(\lambda-1)]$	$[\lambda x-(\lambda-1)]$	$\tau_N(x)$
Sharpness	$\nu_*=f^{-1}\left(\frac{1}{2}\right)$	$f^{-1}[\lambda f(x)-\frac{1}{2}(\lambda-1)]$	$[\lambda x-\frac{1}{2}(\lambda-1)]$	$\tau_S(x)$
Negation ($\lambda=-1$)	$\nu_*=f^{-1}\left(\frac{1}{2}\right)$	$f^{-1}[-f(x)+1]$	$[-x+1]$	$n(x)$

intriguing theoretical problems [2]. Since the modeling is always affected by the presence of different kinds of uncertainty, the use of soft techniques is sensible. Fuzzy set theory is a language that is capable to deal with uncertainty.

In the classical theory, preference is a binary relation closely related to the implications:

$xRy \iff$ "y is not worse than x".

Preferences between alternatives can also be described by a valued preference relation p, such that the value $p(x,y)$ is normalized, and introduced as the degree to which the statement "y is not worse than x" is true:

$p(x,y)=$ truth of $(y \geq x)$.

Here, p is a continuous function, which is strictly decreasing in the first-, and strictly increasing in the second variable, meanwhile $p(x,y)=n(p(y,x))$ must also hold.

In accordance with the case of the implication defined as a composition of the disjunction and the negation operator, $i(x,y)=d(n(x),y)$, it seems convincing to define the preference operator by composing the aggregation and the negation operator, $p(x,y) = a(n(x),y)$. In other words, by substituting $n(x)$ and y in the commutative self-De Morgan weighted aggregative operator, the operator $a(n(x),y)$ has certain properties that are similar to those expected of a preference operator. Consequently, it is sensible to define the preference operator in the following way:

Definition 9. Let $w>0$ be a real parameter and $f:[0,1]\to[0,1]$ be an increasing bijection. Let us define the preference operator as $p_w(x,y)=a_w(n(x),y)=f^{-1}\left[w(f(y)-f(x))+\frac{1}{2}\right]$.

Remark 4. Note that for $w = \frac{1}{2}$, $p_{\frac{1}{2}}(x,y) = f^{-1}\left(\frac{1}{2}\left(f(y)-f(x)+1\right)\right)$, henceforth referred to as $p(x,y)$.

Preference operators with different generator functions and weights are illustrated in Fig. 1. The fact that the implication and the preference operators can be derived in a similar way as $o_{\nu,w}(n(x),y)$ with $\nu=1$, $\nu=f^{-1}\left(\frac{1}{2}\right)$ respectively, provides a possible explanation of the common misconception about their use. Let us consider the following two examples:

If $x<y$ and $y<z$, then $x<z$

If $x\to y$ and $y\to z$, then $x\to z$.

The first one is based on the property of the preference relation describing the transitivity of preferences, whereas the second one is based on the implication (hypothetical reasoning or hypothetical syllogism). In our everyday language, we do usually not distinguish between these two types of reasoning, and we tend to confuse them [31].

4. Properties of the preference operator

In this Section, we give a systematic overview of the main properties of the preference operator defined in Definition 9. First, the basic properties are examined in Section 4.1. In Section 4.2, we focus on the ordering properties, which play an outstanding role in preference modeling. In Sections 4.3 to 4.5 a wide range of compositions with other operators (namely the negation operator, the conjunction, the disjunction, the aggregation and some other unary operators) are examined. Finally, additive transitivity and bisymmetry are studied in Sections 4.6 and 4.7.

4.1. Basic properties

First, we examine some basic properties of the preference operator $p_w(x,y)$. Note the similarities to the properties of implications.

Proposition 2. *The preference operator $p_w(x,y)$ has the following properties:*

1. *Continuity;*
2. *Self-duality (SD, see also Section 4.3); i.e.*

$$p_w(x,y)=n(p_w(n(x),n(y))); \quad \text{(SD)}$$

3. *Neutrality: $p_w(x,x)=\nu_*$;*
4. *Weak dominance of falsity of antecedent (WDF):*

$$p_w(0,y)\geq \nu_* \quad \text{for all} \quad y\in[0,1]; \quad \text{(WDF)}$$

5. *Weak dominance of truth of consequent (WDT):*

$$p_w(x,1)\geq \nu_* \quad \text{for all} \quad x\in[0,1]; \quad \text{(WDT)}$$

6. *Boundary conditions ((BC), Compatibility)*

$$p_w(0,0)=p_w(1,1)=\nu_*;$$

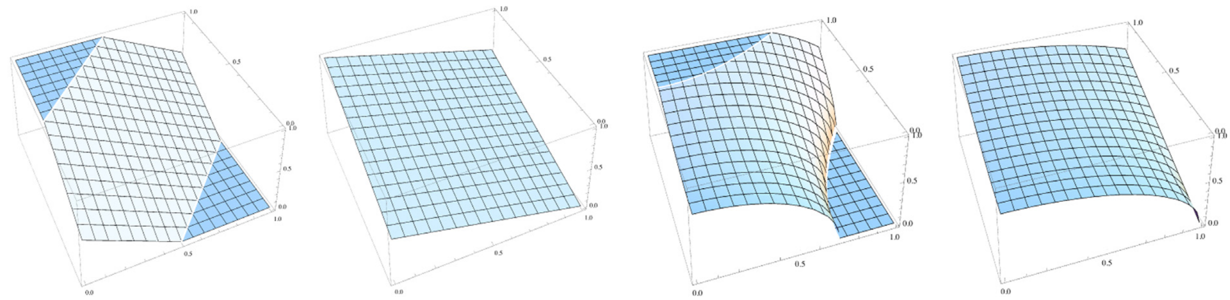

Fig. 1. Preference operator with generator functions $f(x) = x$ and $f(x) = \sqrt{x}$, for $w = 1$ and $w = 0.5$.

$$p_w(0, 1) \geq \nu_*, \qquad p_w(1, 0) \leq \nu_*; \tag{BC}$$

7. *Specially,*

$$p_w(0, 1) = 1, \quad \text{and} \quad p_w(1, 0) = 0$$

if and only if $w \geq \frac{1}{2}$;

8. *Preference property (PP); i.e.*

$$x < y \quad \text{if and only if} \quad p_w(x, y) > \nu_*; \tag{PP}$$

9. *Threshold transitivity (TT); i.e.*

$$p_w(x, y) > \nu_* \text{ and } p_w(y, z) > \nu_* \Rightarrow p_w(x, z) > \nu_*; \tag{TT}$$

where $\nu_* = f^{-1}\left(\frac{1}{2}\right)$.

Proof.

1. Follows directly from the continuity of f.
2. From the commutativity and self-duality of $a_w(x, y)$, we get $p_w(x, y) = a_w(n(x), y) = n(a_w(x, n(y))) = n(p_w(n(x), n(y)))$.
3. Follows from direct calculation.
4. $p_w(0, y) = f^{-1}\left[wf(y) + \frac{1}{2}\right] \geq f^{-1}\left(\frac{1}{2}\right) = \nu_*$.
5. $p_w(x, 1) = f^{-1}\left[w\left(1 - f(x)\right) + \frac{1}{2}\right] \geq f^{-1}\left(\frac{1}{2}\right) = \nu_*$.
6. $p_w(0, 0) = p_w(1, 1) = f^{-1}\left(\frac{1}{2}\right) = \nu_*$;
$p_w(0, 1) = f^{-1}\left[w\left(f(1) - f(0)\right) + \frac{1}{2}\right] = f^{-1}\left[\frac{1}{2} + w\right] \geq \nu_*$;
$p_w(1, 0) = f^{-1}\left[w\left(f(0) - f(1)\right) + \frac{1}{2}\right] = f^{-1}\left[\frac{1}{2} - w\right] \leq \nu_*$.
7. Follows directly from 6.
8. Since f is a strictly increasing function, $x < y$ if and only if $f(x) < f(y)$. $p_w(x, y) = f^{-1}\left[w\left(f(y) - f(x)\right) + \frac{1}{2}\right] > f^{-1}\left[\frac{1}{2}\right] = \nu_*$.
9. Follows directly from 8.

Remark 5. Note that in the first statement of (BC), ν_* represents the maximal level of uncertainty.

4.2. Ordering properties

Next, we focus on the ordering properties, which play an outstanding role in preference modeling. Note the similarities to the implications.

Proposition 3. *The preference operator* $p_w(x, y)$ *satisfies:*

1. *the first place antitonicity:*

$$\text{for all } x_1, x_2, y \in [0, 1]$$
$$(\text{if } x_1 \leq x_2 \text{ then } p_w(x_1, y) \geq p_w(x_2, y)). \tag{FA}$$

2. *the second place isotonicity:*

$$\text{for all } x, y_1, y_2 \in [0, 1]$$
$$(\text{if } y_1 \leq y_2 \text{ then } p_w(x, y_1) \leq p_w(x, y_2)); \tag{SI}$$

3. *the weak ordering property:*

$$p_w(x, y) = 1 \quad \text{if and only if}$$
$$x \leq \tau(y), \quad y \geq f^{-1}\left(\frac{1}{2w}\right), \tag{WOP}$$

where $x \in [0, 1]$, *and* $\tau(x) : [0, 1] \to [0, 1]$ *is an increasing function.*

4.

$$p_w(x, y) = 0 \quad \text{if and only if}$$
$$x \geq \rho(y), \quad y \leq f^{-1}\left[1 - \frac{1}{2w}\right],$$

where $x \in [0, 1]$, *and* $\rho(x) : [0, 1] \to [0, 1]$ *is an increasing function.*

Proof.

1. Follows directly from the monotonicity of $f(x)$:

$$p_w(x, y_1) = f^{-1}\left[w\left(f(y_1) - f(x)\right) + \frac{1}{2}\right] \leq f^{-1}\left[w\left(f(y_2) - f(x)\right) + \frac{1}{2}\right].$$

2. Follows directly from the monotonicity of $f(x)$:

$$p_w(x_1, y) = f^{-1}\left[w\left(f(y) - f(x_1)\right) + \frac{1}{2}\right] \geq f^{-1}\left[w\left(f(y) - f(x_2)\right) + \frac{1}{2}\right].$$

3. $p_w(x, y) = 1$ if and only if

$$f^{-1}\left[w\left(f(y) - f(x)\right) + \frac{1}{2}\right] = 1;$$

i.e. $f(y) - f(x) \geq \frac{1}{2w}$, which means $x \leq f^{-1}\left[f(y) - \frac{1}{2w}\right]$, where $y \geq f^{-1}\left(\frac{1}{2w}\right)$ must hold. Therefore,

$$\tau(y) = f^{-1}\left[f(y) - \frac{1}{2w}\right]$$

is an increasing function with the expected property.

4. Similarly with

$$\rho(y) = f^{-1}\left[f(y) + \frac{1}{2w}\right].$$

Remark 6. Note that in 3, for $w = \frac{1}{2}$, $\tau(y) = 0$ for $\forall y$, which means that $p_{\frac{1}{2}}(x, y) = 1$ if and only if $x = 0, y = 1$.

For $w = 1$, $p_1(x, y) = 1$ if and only if $y \geq \nu_*$ and $x \leq f^{-1}\left[f(y) - \frac{1}{2}\right] \leq \nu_*$.

Similarly, in 4, for $w = \frac{1}{2}$, $\rho(y) = 1$ for $\forall y$, which means that $p_{\frac{1}{2}}(x, y) = 0$ if and only if $x = 1, y = 0$.

For $w = 1$, $p_1(x, y) = 0$ if and only if $y \geq \nu_*$ and $x \geq f^{-1}\left[f(y) - \frac{1}{2}\right] \geq \nu_*$.

4.3. Preference and negation

Next, we consider several compositions of the preference operator with the negation operator. Here, we can see again that the preference is closely related to the implication. The next Proposition tells us that $p(x, y)$ has similar properties to the law of contraposition for implications.

Proposition 4. *The preference operator $p(x, y)$ satisfies the following properties with respect to a strong negation n:*

1.
$$p_w(x, y) = p_w(n(y), n(x)) \quad \text{for all } x, y \in [0, 1];$$
2.
$$p_w(x, y) = n\left(p_w(y, x)\right) \quad \text{for all } x, y \in [0, 1];$$
3.
$$n\left(p_w(x, y)\right) = p_w(n(x), n(y)) \quad \text{for all } x, y \in [0, 1];$$

Proof. From the commutativity and self-duality of $a_w(x, y)$, we get

$$p_w(x, y) = a_w(n(x), y) = a_w(y, n(x)) = p_w(n(y), n(x)).$$

Similarly for the other two statements.

4.4. Preference, conjunction and disjunction

Next, we examine compositions of the preference operator with the main logical operators (conjunction and disjunction).

Proposition 5. *The preference operator $p(x, y)$ satisfies the following properties:*

1. *Asymmetry:*
$$c\left(p_w(x, y), p_w(y, x)\right) = 0; \quad \text{(AS)}$$
2. *S-strong completeness:*
$$; d\left(p_w(x, y), p_w(y, x)\right) = 1 \quad \text{(SSC)}$$
3. *T-transitivity*
$$c\left(p(x, y), p(y, z)\right) \leq p(x, z). \quad \text{(TT)}$$

Proof.

1. Let $A := w(f(y) - f(x))$, then
$$c\left(p_w(x, y), p_w(y, x)\right) = f^{-1}\left[\left[A + \frac{1}{2}\right] + \left[-A + \frac{1}{2}\right] - 1\right] = f^{-1}(0) = 0.$$
2. Similarly for the disjunction.
3. Follows from direct calculation, based on the fact that the terms in [ ] have values between 0 and 1; i.e. the cutting functions can be omitted.

4.5. Preference and aggregation

Next, we examine compositions of the preference operator with the aggregative operator $a(x, y)$.

Proposition 6. *The preference operator $p(x, y)$ satisfies the following properties:*

1. *Transitivity*
$$a\left(p(x, y), p(y, z)\right) = p(x, z)$$
2. *Common Base*
$$p(x, y) = a\left(p(y, z), p(z, x)\right)$$
3. *Inverse Property*
$$y = a\left(x, p_1(y, z)\right)$$
4. *Neutrality*
$$\nu_* = a\left(p(x, y), p(y, x)\right)$$
for all $x_i, y_i \in [0, 1]$.

Proof. All statements follow from direct calculation, based on the fact that the terms in [ ] have values between 0 and 1; i.e. the cutting functions can be omitted.

4.6. Additive transitivity

In [32], Tanino examined different types of transitivities. Among others, he also considered the so-called additive transitivity, where we understand $p(x, y) - \frac{1}{2}$ to be an intensity of preference of y over x.

Definition 10. $p(x, y)$ is an additive preference, if

$$\left(p(x, y) - \frac{1}{2}\right) + \left(p(y, z) - \frac{1}{2}\right) = p(x, z) - \frac{1}{2} \quad (15)$$

holds.

Proposition 7. *$p(x, y)$ is an additive preference, if and only if for its generator function $f(x) = x$ holds.*

Proof.

1. The sufficiency of the condition follows from direct calculation.
2. To prove the necessity, let $p(x, y)$ be a preference operator generated by $f(x)$. Let us define $g(x) = f^{-1}\left(x + \frac{1}{2}\right) - \frac{1}{2}$, $a = \frac{1}{2}(f(y) - f(x))$ and $b = \frac{1}{2}(f(z) - f(y))$. $p(x, y)$ is an additive preference, if and only if
$$g(a) + g(b) = g(a + b). \quad (16)$$
The solution of this functional equation is $g(a) = ca$, $c \neq 0$, $c \in \mathbb{R}$. This means that
$$f^{-1}(x) = g\left(x - \frac{1}{2}\right) + \frac{1}{2} = c\left(x - \frac{1}{2}\right) + \frac{1}{2}. \quad (17)$$
From $f^{-1}(0) = 0$ and $f^{-1}(1) = 1$ follows $c = 1$ and $f(x) = x$.

Remark 7. From the above proposition follows that an additive preference has the form

$$p(x, y) = \frac{1}{2}(y - x + 1). \quad (18)$$

Remark 8. Note that the use of the generator function $f(x) = x$ leads to Łukasiewicz logic.

4.7. Bisymmetry and common base property

When it comes to the problem of consistent aggregation, associativity and bisymmetry play an important role [33]. From aggregation point of view, associativity is an excellent tool for extending a binary function to an n-ary one, however, in some cases, bisymmetry can come even more handy.

Proposition 8. *The preference operator $p(x, y)$ is bisymmetric; i.e.*

$$p\left(p(x_1, y_1), p(x_2, y_2)\right) = p\left(p(x_1, x_2), p(y_1, y_2)\right) \tag{BS}$$

holds for $\forall x_i, y_i \in [0, 1]$.

Proof. Taking into account that the terms in [ ] all have values between 0 and 1, the cutting functions can be omitted. This way, the statement follows from direct calculation.

Proposition 9. *The preference operator $p(x, y)$ satisfies the common base property; i.e.*

$$p(x, y) = p\left(p(z, x), p(z, y)\right) \tag{CB}$$

holds for $\forall x_i, y_i \in [0, 1]$.

Proof. Taking into account that the terms in [ ] all have values between 0 and 1, the cutting functions can be omitted. This way, the statement follows from direct calculation.

4.8. Preference and unary operators

Unary operators (see Table 2) and therefore also membership functions, which play a substantial role in the overall performance of fuzzy representation, can also be interpreted as preference operators, as the following Proposition states. In the literature, membership functions are usually chosen independently from the logical operators of the system. Parameters are normally fine-tuned on the basis of pure experimental results. As recalled in (14), modifiers and membership functions can be connected to the logical operators of the system. Using operator-dependent membership functions makes it possible to build up a system by using a single generator function and a few parameters. Moreover, this can provide a theoretical explanation for the choice of membership functions and modifiers. Now we show that the unary operators can be interpreted as preferences:

Proposition 10.

$$\tau_{\nu_*}^{(\lambda)}(x) = p_\lambda(\nu_*, x),$$

$$\tau_N^{\left(\frac{1}{2}\right)}(x) = p(1, x),$$

$$\tau_P^{\left(\frac{1}{2}\right)}(x) = p(0, x),$$

where $\nu_* = f^{-1}\left(\frac{1}{2}\right)$.

Proof. Follows from direct calculation.

5. Squashing functions and preference operators

Our attention can now be turned to the cutting function. The main drawback of the cutting function in the nilpotent operator family is the lack of differentiability, which would be necessary for numerous practical applications. Although most fuzzy applications (e.g. embedded fuzzy control) use piecewise linear membership functions owing to their easy handling, there are areas where the parameters are learned by a gradient-based optimization method. In this case, the lack of continuous derivatives makes the application impossible. For example, the membership functions have to be differentiable for each input in order to fine-tune a fuzzy control system by a simple gradient-based technique. This problem could be solved by using the so-called squashing function family, which provides a solution to the aforementioned problem by a continuously differentiable approximation of the cutting function. As it was shown in [20,34], these squashing functions have a promising application in the field of neural computation.

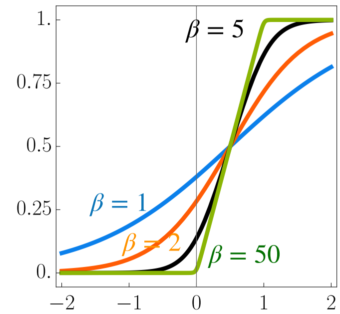

Fig. 2. Squashing functions for $a = 0.5$, $\lambda = 1$, for different β values ($\beta_1 = 1$, $\beta_2 = 2$, $\beta_3 = 5$, and $\beta_4 = 50$).

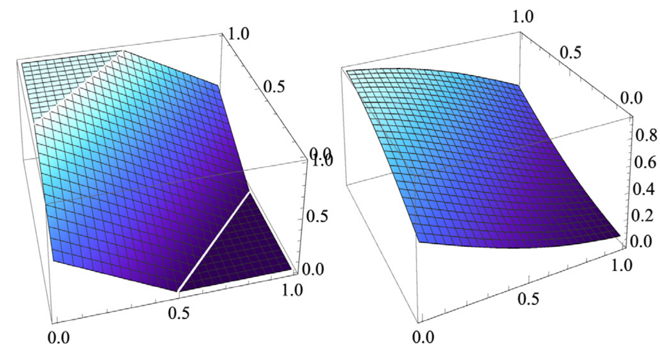

Fig. 3. Preference operators with $w = 1$ and $w = 2$, first using the cutting function, then the squashing function with $\beta = 3$.

Definition 11. The squashing function [28,29] is defined as

$$S_{a,\lambda}^{(\beta)}(x) = \frac{1}{\lambda\beta} \ln \frac{1 + e^{\beta(x-(a-\lambda/2))}}{1 + e^{\beta(x-(a+\lambda/2))}} = \frac{1}{\lambda\beta} \ln \frac{\sigma_{a+\lambda/2}^{(-\beta)}(x)}{\sigma_{a-\lambda/2}^{(-\beta)}(x)},$$

where $x, a, \lambda, \beta \in \mathbb{R}$ and $\sigma_d^{(\beta)}(x)$ denotes the logistic function:

$$\sigma_d^{(\beta)}(x) = \frac{1}{1 + e^{-\beta \cdot (x-d)}}. \tag{19}$$

The squashing function given in Definition 11 is a continuously differentiable approximation of the generalized cutting function by means of sigmoid functions (see Fig. 2). By increasing the value of β, the squashing function approaches the generalized cutting function. In other words, β shows the accuracy of the approximation, while the parameters a and λ determine the center and width. The error of the approximation can be upper bounded by c/β, which means that by increasing the parameter β, the error decreases by the same order of magnitude. The derivatives of the squashing function are easy to calculate and can be expressed by sigmoid functions and itself:

$$\frac{\partial S_{a,\lambda}^{(\beta)}(x)}{\partial x} = \frac{1}{\lambda}\left(\sigma_{a-\lambda/2}^{(\beta)}(x) - \sigma_{a+\lambda/2}^{(\beta)}(x)\right) \tag{20}$$

Preference operators using squashing functions are illustrated in Fig. 3.

6. Application in neural computation

While AI techniques, especially deep learning techniques, are revolutionizing the business and technology world, there is an increasing need to address the problem of interpretability and to improve model transparency, performance, and safety: a problem that is of vital importance to all our research community. This

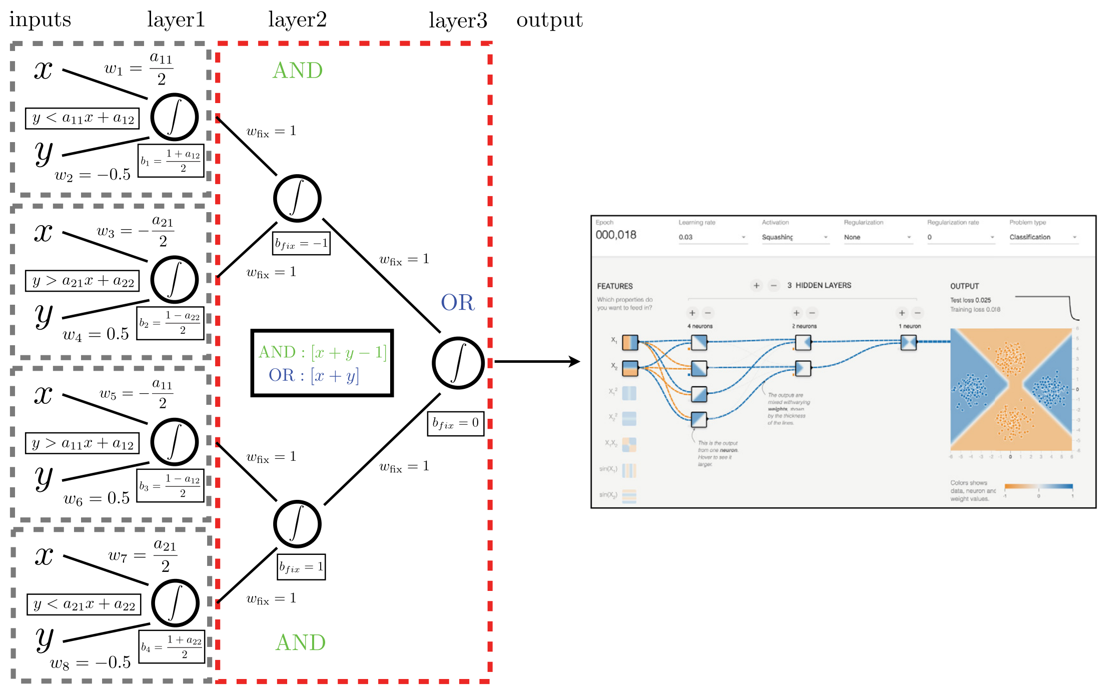

Fig. 4. Nilpotent neural network block designed for preference modeling, Example 1.

challenge is closely related to the fact that although deep neural networks have achieved impressive experimental results, especially in image classification, they have shown to be surprisingly unstable when it comes to adversarial perturbations: minimal changes to the input image may cause the network to misclassify it. In [20], the authors introduced an idea of achieving eXplainable Artificial Intelligence (XAI) by combining neural networks with continuous logic as a promising way to approach the problem: by this combination, the black-box nature of neural models can be reduced, and the neural network-based models can become more interpretable, transparent, and safe. Based on the results of the previous sections, beside logical operators, the preference operator can also be implemented in neural models. As Eq. (12) shows, perceptrons in neural networks can be modeled by the threshold-based nilpotent operator $o_{\underline{v},\underline{w}}(\underline{x})$. Approximating the cutting function with the squashing function given in Section 5 leads to an interpretable neural model introduced in [20]. Here, the weights of the first layer are to be learned, while the hidden layers of the pre-designed neural block work with frozen weights and biases. This explains how the preference operator, as a special case of $o_{\underline{v},\underline{w}}(\underline{x})$, can also be modeled by a perceptron. By learning the parameters of $o_{\underline{v},\underline{w}}(\underline{x})$, the network can be trained to find out which operator to use. Moreover, based on Remark 2, the min and max operators can also be modeled in this framework.

To demonstrate the use of the preference operator in solving classification problems by neural network, we extended the Tensorflow Playground, an interactive visualization of neural networks written in typescript, with the squashing function as activation function and created some new target data sets. In this section, we consider two examples to illustrate the architecture and performance of the suggested model, where the network has to find different regions of positive examples. The data points (represented by small circles) are initially colored orange or blue, which correspond to positive one and negative one. This color-coding is also used for neuron and weight values. In the output layer, the dots are colored orange or blue depending on their original values. The background color shows what the network is predicting for a particular area, while the intensity of the color shows how confident that prediction is.

In the first example (referred to as Example 1), the parameters of two straight lines separating the different regions have to be learned. Here, the target variable is positive when both $y < a_{11}x + a_{12}$ and $y > a_{21}x + a_{22}$ hold or where both $y > a_{11}x + a_{12}$ and $y < a_{21}x + a_{22}$ hold. In a logical network:

- If $y < a_{11}x + a_{12}$ AND $y > a_{21}x + a_{22}$ THEN predict +1
- If $y > a_{11}x + a_{12}$ AND $y < a_{21}x + a_{22}$ THEN predict +1
- Else predict −1

We designed the neural architecture corresponding to the preference modeling introduced in the previous sections: "AND" can be modeled by the conjunction operator, "OR" by the disjunction, while the expression $x > y$ by the preference operator $p(x, y)$. The architecture of the model is illustrated in Fig. 4 (for the weights see Table 1). Note that in the neural block modeling preference, all the weights in the hidden layers are frozen, only the weights of the input layer (the parameters of the straight lines separating the different regions) are to be learned.

In the second example (referred to as Example 2), a more complex region is to be found. Here, the positive examples lie inside a circle or outside a region bordered by two straight lines. The corresponding neural architecture can be seen in Fig. 5.

To show the practicality of the proposed method, a detailed study on the performance of the model and a comparison with further potential competitors are necessary. However, to provide more detailed comparison and a thorough study would be out of the scope of the present article and therefore is left for future work. As for now, to underline the nice performance of the squashing function, we present the results of an experiment using the second example illustrated in Fig. 5. Here, the neural architecture is designed as suggested in Example 2, but at this stage, all the weights and biases are learnable. Table 3 shows a comparison of the performance of the most popular activation functions. Note that in this case, the squashing function outperforms the other three types of activation functions (see Fig. 6).

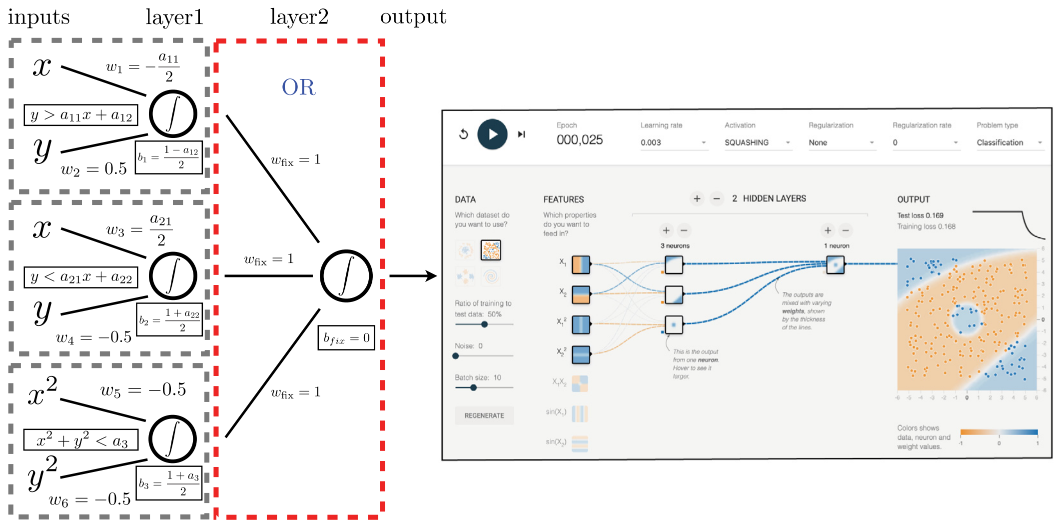

Fig. 5. Nilpotent neural network block designed for preference modeling, Example 2.

Fig. 6. Comparison of the performance of different activation functions in Example 2.

7. Conclusion and future work

This work, as a part of a series of studies about nilpotent systems, suggests a consistent framework for MCDM and continuous logical tools. In this framework, aggregation, preference, and the logical operators are described by the same unary generator function.

Similarly to the implication being defined as a composition of the disjunction and the negation operator, preference operators were introduced as a composition of the aggregative operators and the negation operator. The main properties were examined systematically and suggested a differentiable approximation for the cutting function, the squashing function. Finally, we have shown how the preference operator introduced in this work can be modeled by a perceptron pointing towards applications in the field of neural computation for MCDM. The implementation of this hybrid model in deeper networks (by combining the building blocks introduced here) and its application, illustrated with simulations and further validation of the results is left for future work. As a next step, we are working on a comparison with extreme learning machines introduced in [35], where, similarly to the model suggested in this work, the parameters of hidden nodes need not be tuned. Extreme learning machines are able to produce good generalization performance and learn thousands of times faster than networks trained using backpropagation. Combining extreme learning machines with the continuous logical background can be a very promising direction towards a more interpretable, transparent, and safe machine learning.

Supplemental research is also in progress aiming to explain the empirical success of squashing functions in neural networks by showing that the formulas describing this family follow from natural symmetry requirements [34]. Moreover, we are currently

Table 3
Performance of the squashing function with $\beta = 10, \lambda = 1, a = 0.5$, compared to ReLu, Sigmoid and Tanh in Example 2, with learning rate 0.003, ratio of training to test data: 70%.

Activation Function	Number of Epochs	Test Loss
Sigmoid	500	0.342
	700	0.342
	1000	0.341
	1500	**0.317**
ReLu	500	0.317
	700	0.295
	1000	0.286
	1500	**0.251**
Tanh	500	0.315
	700	0.283
	1000	0.252
	1500	**0.203**
Squashing	500	0.387
	700	0.367
	1000	0.108
	1500	**0.061**

in the process of investigating which "And"- and "Or"-operations can be represented by the fastest (i.e., 1-Layer) neural networks, and which activations functions allow such representations [36].

CRediT authorship contribution statement

Orsolya Csiszár: Writing, Methodology, Investigation, Formal analysis, Editing, Visualisation. **Gábor Csiszár:** Software Coding, Visualisation, Writing, Editing. **József Dombi:** Conceptualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the grant TUDFO/47138-1/2019-ITM from the Ministry of Technology and Innovation, Hungary. The authors are grateful to all anonymous referees whose comments and suggestions have significantly improved our original version of this paper.

References

[1] S.A. Orlovsky, Decision making with a fuzzy preference relation, Fuzzy Sets and Systems 1 (1978) 155–167.
[2] J. Fodor, M. Roubens, Fuzzy Preference Modelling and Multicriteria Decision Support, Kluwer Academic Publishers, Dordrecht, 1994.
[3] C. Kahraman, S.C. Onar, B. Oztaysi, Fuzzy multicriteria decision-making: A literature review, Int. J. Comput. Intell. Syst. 8 (4) (2015) 637–666.
[4] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukias, P. Vincke, Aggregation-overture, in: D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukias, P. Vincke (Eds.), Evaluation and Decision Models with Multiple Criteria. Stepping Stones for the Analyst, Springer, 2006, pp. 117–168.
[5] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukias, P. Vincke, Evaluation and Decision Models with Multiple Criteria. Stepping Stones for the Analyst, Springer, 2006.
[6] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukias, P. Vincke, Numbers and preferences, in: D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukias, P. Vincke (Eds.), Evaluation and Decision Models with Multiple Criteria. Stepping Stones for the Analyst, Springer, 2006, pp. 67–116.
[7] M. Cinelli, M. K, M. Gonzalez, R. Słowiński, How to support the application of multiple criteria decision analysis? in: Let Us Start with a Comprehensive Taxonomy, Omega, 2020, available onle.
[8] J.P. Brans, L'ingénierie de la décision: élaboration d'instruments d'aide á la décision. la méthode PROMETHEE, Presses de l'Université Laval, 1982.
[9] J.R. Figueira, V. Mousseau, B. Roy, ELECTRE methods, in: S. Greco, M. Ehrgott, J. Figueira (Eds.), Multiple Criteria Decision Analysis, in: International Series in Operations Research & Management Science, vol. 233, Springer, New York, NY, 2016.
[10] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: Lecture Notes in Computer Science, 2013, pp. 387–402.
[11] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2014, arXiv:1412.6572.
[12] C. Szegedy, Z. Wojciech, I. Sutskever J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2013, arXiv:1312.6199.
[13] S. Thys, W. Van Ranst, T. Goedemé, Fooling automated surveillance cameras: adversarial patches to attack person detection, 2019, arXiv:1904.08653.
[14] J. Wang, A neural network approach to modelling fuzzy preference relations for multiple criteria decision making, computers and, Oper. Res. 21 (9) (1994) 991–1000.
[15] J. Wang, B. Malakooti, A feedforward neural network for multiple criteria decision making, computers and, Oper. Res. 19 (1992) 151–167.
[16] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[17] R. Brafmanand, C. Domshlak, Preferencehandling - anintroductory tutorial, AI Mag. 30 (1) (2009) 58–86.
[18] A. Elgharabawy, M. Parsad, C. Lin, Preference neural network, IEEE Trans. Neural Netw. Learn. Syst. (2019) preprints.
[19] J. Fürnkranz, E. Hüllermeier, in: J. Fürnkranz, E. Hüllermeier (Eds.), Preference Learning: An Introduction, in Preference Learning, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 1–17.
[20] O. Csiszár, G. Csiszár, J. Dombi, Interpretable neural networks based on continuous-valued logic and multicriterion decision operators, Knowl.-Based Syst. 199 (2020) http://dx.doi.org/10.1016/j.knosys.2020.105972, available online.
[21] D. Dubois, H. Prade, Fuzzy sets in approximate reasoning. Part 1: Inference with possibility distributions, Fuzzy Sets and Systems 40 (1991) 143–202.
[22] E. Trillas, L. Valverde, On some functionally expressable implications for fuzzy set theory, in: Proceedings of the 3rd International Seminar on Fuzzy Set Theory, Linz, Austria, 1981, pp. 173-190.
[23] O. Csiszár, J. Dombi, Generator-based modifiers and membership functions in nilpotent operator systems, in: IEEE International Work Conference on Bioinspired Intelligence, iwobi 2019, July 3-5, 2019, Budapest, Hungary, 2019.
[24] J. Dombi, O. Csiszár, The general nilpotent operator system, Fuzzy Sets and Systems 261 (2015) 1–19.
[25] J. Dombi, O. Csiszár, Implications in bounded systems, Inform. Sci. 283 (2014) 229–240.
[26] J. Dombi, O. Csiszár, Equivalence operators in nilpotent systems, Fuzzy Sets and Systems 299 (2016) 113–129.
[27] J. Dombi, O. Csiszár, Self-dual operators and a general framework for weighted nilpotent operators, Int. J. Approx. Reason. 81 (2017) 115–127.
[28] J. Dombi, O. Csiszár, Operator-dependent modifiers in nilpotent logical systems, operator-dependent modifiers in nilpotent logical systems, in: Proceedings of the 10th International Joint Conference on Computational Intelligence, IJCCI 2018, 2018, pp. 126-134.
[29] J. Dombi, Zs. Gera, The approximation of piecewise linear membership functions and Łukasiewicz operators, Fuzzy Sets and Systems 154 (2005) 275–286.
[30] J. Dombi, Membership function as an evaluation, Fuzzy Sets and Systems 35 (1990) 1–21.
[31] J. Dombi, M. Baczynski, General characterization of implication's distributivity properties: The preference implication, IEEE Trans. Fuzzy Syst. (2019) http://dx.doi.org/10.1109/TFUZZ.2019.2946517.
[32] T. Tanino, Fuzzy preference orderings in group decision makeing, Fuzzy Sets and Systems 12 (1984) 117–131.
[33] J. Fodor, J. Kacprzyk (Eds.), Aspects of Soft Computing Intelligent Robotics and Control, Springer, 2019.
[34] J.C. Urenda, O. Csiszár, G. Csiszár, J. Dombi, O. Kosheleva, V. Kreinovich, G. Eigner, Why squashing functions in multi-layer neural networks, in: IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2020), 2020, in Press.
[35] G. Huang, Q. Zhu, C. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (1–3) (2006) 489–501.
[36] K. Alvarez, J.C. Urenda, O. Csiszár, G. Csiszár, J. Dombi, G. Eigner, V. Kreinovich, Towards fast and understandable computations: Which "and"- and "or"-operations can be represented by the fastest (i.e. 1-layer) neural networks? which activations functions allow such representations?, (UTEP-CS-20-42) University of Texas at El Paso, Department of Computer Science, 2020.