Combining thresholded real values for designing an artificial neuron in a neural network

Olivier Strauss, Agnès Rico, Jerome Pasquet, Lionel Pibre

Combining thresholded real values for designing an artificial neuron in a neural network

Olivier Strauss[a], Agnès Rico[b], Jérôme Pasquet[c], Lionel Pibre[c]

[a] *LIRMM, Université de Montpellier, CNRS, France*
[b] *ERIC, Université Claude Bernard Lyon 1, CNRS, France*
[c] *TETIS, Université Paul Valery, France*

Abstract

This study emanates from a simple observation: as specified by Vapnik [37] in his study, an artificial neural network cannot generate a universal approximator if the aggregation function chosen to design the artificial neuron does not include non-linearity. The usual option is to follow a linear aggregation by a non-linear function, or so-called activation function. We wonder if this approach could be replaced by one using a natively non-linear aggregation function.

Among all of the available non-linear aggregation functions, here we are interested in aggregations based on weighted minimum and weighted maximum operations [8]. As these operators were originally developed within a possibility theory and fuzzy rule framework, such operators cannot be easily integrated into a neural network because the values that are usually considered belong to $[0, 1]$. For gradient descent based learning, a neuron must be an aggregation function derivable with respect to its inputs and synaptic weights, whose variables (synaptic weights, inputs and outputs) must all be signed real values. We thus propose an extension of weighted maximum based aggregation to enable this learning process. We show that such an aggregation can be seen as a combination of four Sugeno integrals. Finally, we compare this type of approach with the classical one.

Key words: Neural network, Sugeno integral, non-additive aggregation, non-monotonic set functions

1. Introduction

One of the basic principles of conventional neural networks is to cascade non-linear aggregation functions that are supposed to mimic the functioning of natural neurons, hence their name "artificial neurons". Here we are adopting a broad view of the notion of aggregation function as proposed in [13]: *The*

Email addresses: Olivier.Strauss@lirmm.fr (Olivier Strauss), Agnes.Rico@univ-lyon1.fr (Agnès Rico), jerome.pasquet@univ-montp3.fr (Jérôme Pasquet), lionel.pibre@univ-montp3.fr (Lionel Pibre)

essence of aggregation is that the output value computed by the aggregation function should represent or synthesize "in some sense" all individual inputs, where quotes are put to emphasize the fact that the precise meaning of this expression is highly dependent on the context. Most aggregation functions used to define an artificial neuron is achieved by computing a non-linear function of a weighted sum of its inputs. The learning process is a regression to adjust the *synaptic* weights of each aggregation function, i.e. each artificial neuron. The non-linear function has a very important role. Through this non-linearity, learning is possible because it allows us to generate a universal approximator if the number of neurons and their arrangement are adequate. The non-linearity achieves soft thresholding. Different functions have been used, including sigmoid [20], rectified linear unit (ReLU) [25], parametric rectified linear unit (PReLU) [15] functions, etc.

What we propose aims to reverse this principle: here it is a matter of thresholding first and then summing. If such a neuron is inserted in a network, the learning goal is no longer to find the best multiplicative weights associated with each input, but rather the best threshold associated with each input.

The aggregation function we propose extends the weighted disjunction proposed by Dubois and Prade [8] in the possibility theory framework. We show that this approach can be related to a Sugeno integral with respect to a possibility measure, which is a well-known tool in decision theory research.

Very few published articles have proposed to use max-min approaches in neural networks, even though Sugeno himself mentioned this possibility in his thesis [33][1].

On the one hand, most articles referring to the use max-min approaches in connection with neural networks focus on Takagi-Sugeno fuzzy neural networks, which is a very particular use for an if-then-else representation of the relation between inputs and outputs [5]. For example, in [35], such a network is used for improving an adaptive sliding mode control strategy. Some other approaches refer to the Sugeno integral. Such an approach is reported in [24], where this integral is used for ranking the relevance of three type-2 Mamdani inference-based neural network modules in a decision process in order to improve an image recognition method. In [29], the monotonicity property of the Sugeno integral is used to replace the arithmetic aggregation operator in order to reduce the features extracted by convolutional layers in the pooling process. However, the fuzzy measure used in the Sugeno integral is not learned in any of these studies. It is stated as being known.

On the other hand, some authors propose methods to learn the 2^n values of a fuzzy measure used for aggregating n inputs (see e.g. [5] and references therein). In [5], the proposed method consists of representing the input-output relation of a dataset by a capacity. Each value of the capacity is represented by an interval. In [28], the learning process consists of updating the bounds

[1] In the conclusion, Sugeno wrote: *It is particularly hoped that this research will serve in future for the studies of artificial intelligence.*

of each interval through the use of a simulated annealing-based method by iteratively incorporating new data. One of the weaknesses of this approach is that incoherence between the model and the dataset often leads to empty intervals. In [1] the authors propose to learn fuzzy measure values used with a Sugeno integral for binary classification based on ordinal data. In that case, learning is achieved based on the empirical risk minimization principle. In this approach, the authors often restrict themselves to k-interactive measures in order to reduce the parameter space, and also to simplify the fulfilment of the monotony constraint associated with fuzzy measures. Due to the complexity of this problem, only an approximation of the solution is obtained via linear programming.

In recent work [4], a promising approach proposing the approximate solution of a max-min matrix equation could lead to the learning of a max-min neural network. However, this preliminary solution does not currently allow us to backpropagate an error in the network and thus learn the synaptic weights. This approach could be particularly well suited to creating classification neural networks that can potentially be interpreted by if-then rules. In [38], min-sum and max-sum approaches, also called tropical arithmetic, have been proposed to improve classification in convolutional layers, with an application to breast cancer diagnosis. Most of the work proposes very low-depth neural architecture [11]. A more theoretical study of this type of approach can be found in [10]. Finally, a recent article [9] offers a broad overview of the use of possibilistic approaches to learning.

A number of recent papers have shown that neurons operate in a way that is quite different from the thresholded weighted sum model usually used (see e.g. [6]). In particular, it seems that the transfer of an electrochemical message from one neuron to another involves a threshold specific to each connection, this threshold being soft in one way or another [14]. Similarly, the sum-of-inputs model as causing neuron excitation is sometimes questioned, with a winner-takes-all model possibly being more appropriate [32]. This behaviour is well modelled by a fuzzy relation of the if-then type corresponding to weighted disjunction. Such a model therefore seems to better correspond to the behaviour of a natural neuron as proposed in [6]. However, as noted in [14], message transmission between two neurons can be excitatory or inhibitory, which corresponds to positive or negative weights or excitations in the classical model. It would therefore be interesting, in order to perfect a model based on weighted disjunction, to extend this model to signed values. .

The approach we propose in this paper is thus particularly unique since it consists of extending the max-min approach proposed by Dubois and Prade [8] to generate an aggregation function so as to enable conventional learning by quadratic regression. This kind of extension has already been proposed by Grabisch [12], and then Sugeno [34], in a purely ordinal framework. Our approach is fundamentally different in spirit since we position ourselves in a quantitative context. We call this approach ST aggregation (i.e. the " sum of thresholded values"). Because of the summation, unlike Grabisch's approach, this aggregation operator cannot be considered as an extension of the Sugeno integral but rather

as an additive balance of Sugeno integrals. For comparison, we also consider another approach, that is closer to symmetric Sugeno integral and cumulative prospect theory Sugeno integral proposals of Grabisch and Sugeno. We call this other approach MT aggregation (i.e. the " maximum of thresholded values").

We show, in several experiments, that these new aggregation functions can be used in a conventional neural network to replace the commonly used aggregation function (weighted sum with threshold). Our experimental studies focus on regression experiments with fully connected networks. We compare four approaches involving the same number of layers and neurons: the classical approach, the ST-aggregation based approach, the MT-aggregation based approach, and finally a hybrid approach where ST neurons are mixed with classical neurons.

The first interesting result is that these qualitative aggregation based approaches work: a purely ST network can learn a relationship from examples, while achieving learning performance that is inferior to but comparable to that of conventional networks. An MT-aggregation based neural network performs less well when used as a conventional neuron.

The second result is that the ST approach is perfectly compatible with the classical approach: it is possible to interleave layers of ST neurons with layers of classical neurons in the same network.

Finally, since the derivatives w.r.t. inputs and weights are integer values, both ST and MT methods are potentially not subject to the vanishing gradient problem.

After this introduction, in Section 2 we define the notations we use and provide some important preliminary notions to help understand the proposed approach. In Section 3, we present two signed extensions of the weighted maximum, one based on a maximal trade-off (the MT-aggregation) and another based on an additive trade-off (the ST-aggregation). We outline some key properties of both extensions. We attempt to give an interpretation of the use of such aggregation functions in the neural network framework. Section 4 concerns some pathways for computing the derivatives of both aggregations with respect to the inputs and parameters, which are essential for the learning process. Section 5 is devoted to experiments comparing both approaches to the conventional approach. We show that the use of neurons based on ST aggregation allows us to obtain performances close to those that may be obtained with the conventional approach, thereby highlighting the relevance of this new approach. On the other hand, the MT approach seems less promising in the context of gradient descent learning, a result which is not surprising given the highly non-linear nature of this approach. Finally we conclude and discuss the relevance of using such approaches in the neural network framework.

2. Theoretical background

2.1. Notations

- $\Omega = \{1, \ldots, n\} \subset \mathbb{N}$.

- A **real vector** $\boldsymbol{x} : \Omega \to \mathbb{R}^n$ is a discrete function defined by a discrete subset of $\mathbb{R}^n$: $\boldsymbol{x} = (x_1, \cdots, x_n) \in \mathbb{R}^n$. $\mathbf{0} = (0, \cdots, 0)$ is the null vector and $\mathbf{1} = (1, \cdots, 1)$ is the unit vector.

- On $\mathbb{R}$, the maximum operator is denoted by $\vee$ and the minimum operator is denoted by $\wedge$. This notation is also used on $\mathbb{R}^n$ where the maximum and the minimum are calculated coordinate by coordinate.

- A **set function** is a function $\vartheta : 2^\Omega \to \mathbb{R}$ that associates a real value to any subset of Ω.

- A **kernel** of Ω is a real vector $\boldsymbol{\varphi} : \Omega \to \mathbb{R}$ used as a parameter of a set function defined by $\boldsymbol{\varphi} = (\varphi_1, \cdots, \varphi_n)$.

- A set function ϑ of Ω is said to be **additive** if $\forall A, B \subseteq \Omega$, $\vartheta(A \cup B) + \vartheta(A \cap B) = \vartheta(A) + \vartheta(B)$.

- A **capacity** is a normalized increasing set function $\upsilon : 2^\Omega \to \mathbb{R}^+$ with $\upsilon(\varnothing) = 0$. Normalized means $\upsilon(\Omega) = 1$ and increasing means that $\forall A \subseteq B \subseteq \Omega$, $\upsilon(A) \leq \upsilon(B)$.

- A **possibility measure** is a maxitive capacity Π: $\forall A, B \subseteq \Omega$, $\Pi(A \cup B) = \Pi(A) \vee \Pi(B)$. A discrete possibility measure is generally associated with a normalized kernel $\boldsymbol{\pi}$, such that $\pi_i = \Pi(\{i\})$, a so called possibility distribution. In that case, $\forall A \subseteq \Omega$, $\Pi(A) = \vee_{i \in A} \pi_i$. Normalized means that, since Π is a capacity: $\Pi(\Omega) = \vee_{i=1}^n \pi_i = 1$, thus $\forall i \in \Omega$, $\pi_i \in [0, 1]$.

- A **necessity measure** is a minitive capacity N: $\forall A, B \subseteq \Omega$, $N(A \cap B) = N(A) \wedge N(B)$. It can also be associated with a possibility distribution $\boldsymbol{\pi} \in [0, 1]^n$. In that case, $\forall A \subseteq \Omega$, $N(A) = 1 - \vee_{i \in A^c} \pi_i = \wedge_{i \in A^c} (1 - \pi_i)$, where A^c is the complementary set of A in Ω.

- To any subset $A \subseteq \Omega$ is associated a function $\mathbf{1}_A : \Omega \to \{0, 1\}$, which is called the **characteristic function** of A, such that $\forall i \in \Omega$, $\mathbf{1}_A(i) = 1$ if $i \in A$ and 0 otherwise.

2.2. Weighted maximum and weighted minimum

The minimum $\wedge$ and the maximum $\vee$ operators have been extended to weighted min and weighted max aggregations. Let $\boldsymbol{\varphi} \in [0, 1]^n$ (with $\vee_{i \in \Omega} \varphi_i = 1$) be a kernel of Ω whose components are used as weights of weighted maximum and weighted minimum. Formally, let $\boldsymbol{x} \in [0, 1]^n$, then we have:

- the weighted maximum operator associated with $\boldsymbol{\varphi}$ is:

$$\mathrm{wmax}_{\boldsymbol{\varphi}}(\boldsymbol{x}) = \bigvee_{i \in \Omega} \varphi_i \wedge x_i.$$

- the weighted minimum operator associated with $\boldsymbol{\varphi}$, supposed to be such that $\wedge_{i\in\Omega}\varphi_i = 0$, is:

$$\mathrm{wmin}_{\boldsymbol{\varphi}}(\boldsymbol{x}) = \bigwedge_{i\in\Omega} \varphi_i \vee x_i.$$

Weighted maximum and weighted minimum are particular cases of the Sugeno integral [22]. The Sugeno integral of $\boldsymbol{x}$ with respect to a capacity v is the qualitative aggregation operator defined as follows [33]:

$$S_v(\boldsymbol{x}) = \bigvee_{A\subseteq\Omega} v(A) \wedge \left(\bigwedge_{i\in A} x_i\right).$$

The Sugeno integral with respect to a possibility measure associated with the kernel $\boldsymbol{\varphi}$ equals $\mathrm{wmax}_{\boldsymbol{\varphi}}$ and the Sugeno integral with respect to a necessity measure associated with $\boldsymbol{\varphi}$ equals $\mathrm{wmin}_{\boldsymbol{\varphi}}$.

2.3. Extension of maximum and minimum operations to signed values

In [12], Garbisch proposes a symmetric extension of the Sugeno integral to signed values. This extension is purely ordinal and in the spirit of the symmetric (Šipos) extension of the Choquet integral. It is based on extending the maximum ($\vee$) and minimum ($\wedge$) operations so that the algebraic structure is close to a ring. Let us briefly recall some interesting results in our context.

The negative numbers are modeled as follows. A linear order set $\{L^+, \leq\}$, with bottom and top elements denoted 0 and 1, is considered and a reverse set is defined by $L^- = \{-\alpha | \alpha \in L^+\}$ with the reversed order $-\alpha \leq -\beta$ if and only if $\beta \leq \alpha$ in L^+. The bottom and top of L^- are -1 and -0, respectively, where -0 is equal to 0. L denotes the union of L^+ and L^-. Thus the top of L is 1 and its bottom is -1.

The absolute value of $\alpha \in L$ is defined by $|\alpha|$ where

$$|\alpha| = \begin{cases} \alpha & \text{if } \alpha \in L^+ \\ -\alpha & \text{if } \alpha \in L^- \end{cases}$$

Where $\alpha, \beta \in L$, the symmetric extensions of the $\vee$ and $\wedge$ operations are defined as follow:

$$\alpha \veebar \beta = \begin{cases} -\vee(|\alpha|, |\beta|) & \text{if } \beta \neq -\alpha, \\ 0 & \text{if } \beta = -\alpha, \\ \vee(|\alpha|, |\beta|) & \text{otherwise.} \end{cases}$$

$$\alpha \barwedge \beta = \begin{cases} -(|\alpha| \wedge |\beta|) & \text{if } sign(\alpha) \neq sign(\beta), \\ |\alpha| \wedge |\beta| & \text{otherwise.} \end{cases}$$

where $sign(\alpha) = 1$ is $\alpha \in L^+$ and $sign(\alpha) = -1$ is $\alpha \in L^-$.
These expressions can be simplified:

$$\alpha \circledvee \beta = sign(\alpha + \beta)\left(|\alpha| \vee |\beta|\right),$$
$$\alpha \circledwedge \beta = sign(\alpha.\beta)\left(|\alpha| \wedge |\beta|\right).$$

As proved in [12],

- both $\circledvee$ are $\circledwedge$ are commutative,
- 0 is the sole neutral element of $\circledvee$ and absorbant of $\circledwedge$,
- 1 is the sole neutral element of $\circledwedge$ and absorbant of $\circledvee$,
- $\circledwedge$ is associative,
- $\circledvee$ is associative for any expression involving $\alpha_1 \cdots \alpha_n$ in L such that $\vee_{i=1}^{n}\alpha_i \neq -\wedge_{i=1}^{n}\alpha_i$.
- $\circledwedge$ is distributive w.r.t. $\circledvee$ in L^+ and L^-.

In this paper, we consider a rule combining separately positive and negative values. In this case if $\vee_{i=1}^{n}\alpha_i = -\wedge_{i=1}^{n}\alpha_i$ then we have $\circledvee_{i=1}^{n}\alpha_i = 0$.

Based on this extension of $\vee$ and $\wedge$ operations, several extensions have been proposed in [12] to define a symmetric extension of the Sugeno integral. Here we are interested in the *natural extension* w.r.t. a possibility measure.

Let $\boldsymbol{\varphi} \in [0,1]^n$ and $\Pi_{\boldsymbol{\varphi}}$ be the possibility measure associated with $\boldsymbol{\varphi}$. The Sugeno integral of $\boldsymbol{x} \in [0,1]^n$ w.r.t. $\Pi_{\boldsymbol{\varphi}}$ is:

$$S_{\Pi_{\boldsymbol{\varphi}}}(\boldsymbol{x}) = \bigvee_{i\in\Omega}(\varphi_i \wedge x_i).$$

Now considering $\boldsymbol{x} \in [-1,1]^n$, then the symmetric the Sugeno integral of $\boldsymbol{x}$ w.r.t. $\Pi_{\boldsymbol{\varphi}}$ is :

$$\check{S}_{\Pi_{\boldsymbol{\varphi}}}(\boldsymbol{x}) = \left(S_{\Pi_{\boldsymbol{\varphi}}}(\boldsymbol{x}^+)\right) \circledvee \left(-S_{\Pi_{\boldsymbol{\varphi}}}(-\boldsymbol{x}^-)\right),$$

with $\boldsymbol{x}^+ = \boldsymbol{x} \vee \mathbf{0}$ and $\boldsymbol{x}^- = \boldsymbol{x} \wedge \mathbf{0}$.

Note that $S_{\Pi_{\boldsymbol{\varphi}}}(\boldsymbol{x}^+) = \bigvee_{i\in\Omega}(\varphi_i \wedge x_i^+) = \circledvee_{i\in\Omega}\left(\varphi_i \circledwedge x_i^+\right)$
and $-S_{\Pi_{\boldsymbol{\varphi}}}(-\boldsymbol{x}^-) = -\bigvee_{i\in\Omega}(\varphi_i \wedge (-x_i^-)) = \circledvee_{i\in\Omega}\left(\varphi_i \circledwedge x_i^-\right)$ thus

$$\check{S}_{\Pi_{\boldsymbol{\varphi}}}(\boldsymbol{x}) = \left(\bigcircledvee_{i\in\Omega}\left(\varphi_i \circledwedge x_i^+\right)\right) \circledvee \left(\bigcircledvee_{i\in\Omega}\left(\varphi_i \circledwedge x_i^-\right)\right). \tag{1}$$

This expression cannot be further reduced due to the non-associativity of the $\circledvee$ operator.

3. Extending weighted maximum to signed values and weights: MT- and ST-aggregations

Recall that our goal here is to create a new formal neuron by replacing the usual nonlinear aggregation function, composed of a linear aggregation and an activation function, with a natively nonlinear aggregation function. We also would like this neuron to be easily integrated into a conventional neural network, thereby enabling gradient descent adjustment. These specifications entail two major constraints for this new aggregation function:

i) it must admit signed real inputs and weights while outputing a signed value,
ii) in contrast to what is practiced with maximum operators, the scale must be able to change when the weights are updated,
iii) it must be derivable with respect to inputs and weights, and those derivatives should be continuous,

The extension proposed in Section 2.3 takes an important step in this direction, albeit with two substential restrictions that contradict previous specifications: first in expression (1) the weights are not signed, and second the values are fixed on an irremovable scale.

The aim of this section is twofold: 1– to show that bounds of wmax aggregation are arbitrary and can be modified, 2– to propose an extension of wmax to signed real values which is derivable and whose derivative is continuous.

3.1. Non-normalized weighted maximum

In the approach of [8], the weighted maximum is only defined when $\boldsymbol{x}$ and $\boldsymbol{\varphi}$ belong to $[0,1]^n$. In this section, we intend to consider that values of both $\boldsymbol{x}$ and $\boldsymbol{\varphi}$ may be unbounded, i.e. $\boldsymbol{x} \in \mathbb{R}^{n+}$ and $\boldsymbol{\varphi} \in \mathbb{R}^{n+}$, without normalization condition. We define $\mathcal{M}$ as being an unbounded extension of the weighted maximum:

$$\begin{array}{rccl} \mathcal{M} : \mathbb{R}^{n+} \times \mathbb{R}^{n+} & \rightarrow & \mathbb{R}^{+} \\ (\boldsymbol{x}, \boldsymbol{\varphi}) & \mapsto & \bigvee_{i\in\Omega} (\varphi_i \wedge x_i) \end{array} \tag{2}$$

For each $\boldsymbol{\varphi} \in \mathbb{R}^{n+}$, $\mathcal{M}(.,\boldsymbol{\varphi})$ can be considered as an aggregation function on $\mathbb{R}^{n+}$.

If $\boldsymbol{\varphi}$ is bounded then there is a link with a Sugeno integral. Indeed, since $\forall i \in \Omega\ \varphi_i \in [\underline{\varphi}, \overline{\varphi}]$, where $\underline{\varphi} = \wedge_{i\in\Omega}\varphi_i$ and $\overline{\varphi} = \vee_{i\in\Omega}\varphi_i$, we have $\forall \boldsymbol{x} \in \mathbb{R}^{n+}$, $\mathcal{M}(\boldsymbol{x},\boldsymbol{\varphi}) = \mathcal{M}(\boldsymbol{x} \wedge \boldsymbol{\varphi}, \boldsymbol{\varphi})$. So we can thus consider the restriction of $\mathcal{M}(.,\boldsymbol{\varphi})$ to $[0,\overline{\varphi}]$ denoted by $\mathcal{M}(.,\boldsymbol{\varphi})|_{[0,\overline{\varphi}]}$. This restriction is a weighted lattice polynomial as defined in [23]. So it is easy to check that $\mathcal{M}(.,\boldsymbol{\varphi})|_{[0,\overline{\varphi}]}$ is a Sugeno integral w.r.t. the set function μ defined by $\forall A \subseteq \Omega$, $\mu(A) = \bigvee_{i\in A} \varphi_i$.

Considering a given $\boldsymbol{\varphi}$, we are going to prove that $\mathcal{M}(\cdot,\boldsymbol{\varphi})$ is bounded. More precisely, if one value x_i of $\boldsymbol{x}$ is greater than φ_i, then $\mathcal{M}(\boldsymbol{x},\boldsymbol{\varphi}) \in [\underline{\varphi}, \overline{\varphi}]$.

Proposition 3.1. *Let* $\boldsymbol{x} \in \mathbb{R}^{n+}$

- *if* $\exists i_0 \in \Omega$ *such that* $x_{i_0} > \underline{\varphi}$, *then* $\mathcal{M}(\boldsymbol{x},\boldsymbol{\varphi}) \in [\underline{\varphi}, \overline{\varphi}]$,

- *if* $\forall i \in \Omega$, $x_i \leq \underline{\varphi}$, *then* $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) = \vee_{i\in\Omega} x_i \leq \underline{\varphi}$.

Proof.

- If $\exists i_0 \in \Omega$ such that $x_{i_0} > \underline{\varphi}$, then $x_{i_0} \wedge \varphi_{i_0}$ is either $\varphi_{i_0} \geq \underline{\varphi}$ or $x_{i_0} > \underline{\varphi}$, so we have $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \geq \underline{\varphi}$.

  $\forall i \in \Omega, x_i \wedge \varphi_i \leq \varphi_i \leq \overline{\varphi}$ so we have $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \leq \overline{\varphi}$.

  Hence in this case we have $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \in [\underline{\varphi}, \overline{\varphi}]$.

- If $\forall i \in \Omega$, $x_i \leq \underline{\varphi}$, then $\forall i \in \Omega$, $x_i \wedge \varphi_i = x_i$ and $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) = \vee_{i\in\Omega} x_i$. In this case $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \leq \underline{\varphi}$.

□

According to the definition of $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi})$, the roles of $\boldsymbol{x}$ and $\boldsymbol{\varphi}$ can be exchanged. So we have the following result considering a given $\boldsymbol{x}$.

Proposition 3.2. *Let* $\boldsymbol{\varphi} \in \mathbb{R}^{n+}$

- *if* $\exists i_0 \in \Omega$ *such that* $\varphi_{i_0} > \underline{x}$, *then* $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \in [\underline{x}, \overline{x}]$,
- *if* $\forall i \in \Omega$, $\varphi_i \leq \underline{x}$, *then* $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) = \vee_{i\in\Omega} \varphi_i$.

According to the previous properties, we have the following result.

Corollary. *Let* $\boldsymbol{x}, \boldsymbol{\varphi} \in \mathbb{R}^{n+}$, $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \in [\underline{x} \wedge \underline{\varphi}, \overline{x} \wedge \overline{\varphi}]$.

Proof.

- Let us prove that $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \leq \overline{x} \wedge \overline{\varphi}$.

  According to proposition 3.1 we have $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \leq \overline{\varphi}$.

  $\forall i \in \Omega$, $x_i \wedge \varphi_i \leq x_i \leq \overline{x}$ so we have $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \leq \overline{x}$, which concludes the proof of the inequality.

- Let us prove that $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \geq \underline{x} \wedge \underline{\varphi}$.

  According to proposition 3.1,

  if $\exists i_0 \in \Omega$ such that $x_{i_0} > \underline{\varphi}$, then $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) \geq \underline{\varphi} \geq \underline{x} \wedge \underline{\varphi}$.

  If $\forall i \in \Omega$, $x_i \leq \underline{\varphi}$, then $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) = \vee_{i\in\Omega} x_i \geq \underline{x} \geq \underline{x} \wedge \underline{\varphi}$, which concludes the proof of the inequality.

□

Now let $\boldsymbol{\delta} \in \mathbb{R}^n$ be a bounded vector, the vector $\boldsymbol{\phi} = \mathbf{0} \vee (\boldsymbol{\varphi} + \boldsymbol{\delta})$ is also bounded, i.e. increasing (or decreasing) the values of $\boldsymbol{\varphi}$ leads to another bounded aggregation $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\phi})$.

This is an important property because, when used in a neural network, the kernel associated with each neuron is intended to be updated additively. The fact that modifying the kernel does not change the intrinsic property of the $\mathcal{M}$ operator is therefore of prime interest.

3.2. Signed extension

We now propose to combine the proposals put forward in Section 2.3 with those outlined in Section 3.1.

Let $\boldsymbol{\varphi} \in \mathbb{R}^n$ and $\boldsymbol{x} \in \mathbb{R}^n$. Let $\boldsymbol{\varphi}^+ = \mathbf{0} \vee \boldsymbol{\varphi}$, $\boldsymbol{\varphi}^- = \mathbf{0} \wedge \boldsymbol{\varphi}$, $\boldsymbol{x}^+ = \mathbf{0} \vee \boldsymbol{x}$ and $\boldsymbol{x}^- = \mathbf{0} \wedge \boldsymbol{x}$. According to the signed extensions proposed in [12], an extended weighted maximum aggregation of $\boldsymbol{x}$ w.r.t. $\boldsymbol{\varphi}$ can be divided in four parts:

$$\bullet\ \mathcal{M}(\boldsymbol{x}^+, \boldsymbol{\varphi}^+) = \bigvee_{i\in\Omega} \left(x_i^+ \wedge \varphi_i^+\right) = \bigvee\!\!\!\!\bigcirc_{i\in\Omega} \left(x_i^+ \circledwedge \varphi_i^+\right), \tag{3}$$

$$\bullet\ \mathcal{M}(-\boldsymbol{x}^-, \boldsymbol{\varphi}^+) = \bigvee_{i\in\Omega} \left(-x_i^- \wedge \varphi_i^+\right) = -\bigvee\!\!\!\!\bigcirc_{i\in\Omega} \left(x_i^- \circledwedge \varphi_i^+\right), \tag{4}$$

$$\bullet\ \mathcal{M}(\boldsymbol{x}^+, -\boldsymbol{\varphi}^-) = \bigvee_{i\in\Omega} \left(x_i^+ \wedge -\varphi_i^-\right) = -\bigvee\!\!\!\!\bigcirc_{i\in\Omega} \left(x_i^+ \circledwedge \varphi_i^-\right), \tag{5}$$

$$\bullet\ \mathcal{M}(-\boldsymbol{x}^-, -\boldsymbol{\varphi}^-) = \bigvee_{i\in\Omega} \left(-x_i^- \wedge -\varphi_i^-\right) = \bigvee\!\!\!\!\bigcirc_{i\in\Omega} \left(x_i^- \circledwedge \varphi_i^-\right). \tag{6}$$

Remark 1. *Note that $\mathcal{M}(.,\boldsymbol{\varphi}^+)|_{[0,\overline{\varphi}]}$ is a Sugeno integral w.r.t. the set function μ^+ defined by $\forall A \subseteq \Omega$, $\mu^+(A) = \bigvee\limits_{i\in A} \varphi_i^+$ and $\mathcal{M}(.,-\boldsymbol{\varphi}^-)|_{[0,|\underline{\varphi}|]}$ is a Sugeno integral w.r.t. the set function μ^- defined by $\forall A \subseteq \Omega$, $\mu^-(A) = \bigwedge\limits_{i\in A} \varphi_i^-$.*

Now, several proposals can be made to combine these four parts and achieve aggregation. We limit ourselves to two proposals.

3.3. Natural signed extension: MT-aggregation

The first idea that comes to mind after reading Section 3.2 is to follow proposition 27 put forward by Grabisch [12], which would lead to proposing the maximum of thresholded values aggregation $\mathcal{A}_{MT}$ by:

$$\begin{aligned}
\mathcal{A}_{MT}(\boldsymbol{x}, \boldsymbol{\varphi}) &= \left(\mathcal{M}(\boldsymbol{x}^+, \boldsymbol{\varphi}^+)\right) \bigvee\!\!\!\!\bigcirc \left(-\mathcal{M}(-\boldsymbol{x}^-, \boldsymbol{\varphi}^+)\right) \\
&\quad \bigvee\!\!\!\!\bigcirc \left(-\mathcal{M}(\boldsymbol{x}^+, -\boldsymbol{\varphi}^-)\right) \bigvee\!\!\!\!\bigcirc \left(\mathcal{M}(-\boldsymbol{x}^-, -\boldsymbol{\varphi}^-)\right), \\
&= \left(\bigvee\!\!\!\!\bigcirc_{i\in\Omega} \left(x_i^+ \circledwedge \varphi_i^+\right)\right) \bigvee\!\!\!\!\bigcirc \left(\bigvee\!\!\!\!\bigcirc_{i\in\Omega} \left(x_i^- \circledwedge \varphi_i^+\right)\right) \\
&\quad \bigvee\!\!\!\!\bigcirc \left(\bigvee\!\!\!\!\bigcirc_{i\in\Omega} \left(x_i^+ \circledwedge \varphi_i^-\right)\right) \bigvee\!\!\!\!\bigcirc \left(\bigvee\!\!\!\!\bigcirc_{i\in\Omega} \left(x_i^- \circledwedge \varphi_i^-\right)\right), \\
&= \bigvee_{\star,\bullet\in\{+,-\}} \bigvee\!\!\!\!\bigcirc_{i\in\Omega} \left(x_i^\star \circledwedge \varphi_i^\bullet\right).
\end{aligned} \tag{7}$$

Let $\overline{\varphi} = \max_{i\in\Omega} \varphi_i$ and $\underline{\varphi} = \min_{i\in\Omega} \varphi_i$, then the aggregated value $\mathcal{A}_{MT}(\boldsymbol{x}, \boldsymbol{\varphi})$ belongs to $[-\Delta, \Delta]$ where $\Delta = \max\left(|\underline{\varphi}|, |\overline{\varphi}|\right)$.

This proposition is in the line with the idea of extending the Choquet integral to negative values as well as using signed (and thus non-monotonic) set functions (see e.g. [27]). This type of extension has already been proposed by Sugeno in [34] in the normalized framework by introducing the notion of pairs of fuzzy measures. In the present case, considering Remark 1, the pair of set functions used for MT aggregation are: $\forall A \subseteq \Omega$ $\mu^+(A) = \bigvee\limits_{i\in A} \varphi_i^+$ and $\mu^-(A) = \bigwedge\limits_{i\in A} \varphi_i^-$.

Thus we have:

$$\begin{aligned}\mathcal{A}_{MT}(\boldsymbol{x},\boldsymbol{\varphi}) = &\left(\bigvee_{A\subseteq\Omega}\mu^+(A)\wedge\left(\bigwedge_{i\in A}x_i^+\right)\right)\vee\left(\bigvee_{A\subseteq\Omega}\mu^+(A)\wedge\left(\bigwedge_{i\in A}x_i^-\right)\right)\\ &\vee\left(\bigvee_{A\subseteq\Omega}\mu^-(A)\wedge\left(\bigwedge_{i\in A}x_i^+\right)\right)\vee\left(\bigvee_{A\subseteq\Omega}\mu^-(A)\wedge\left(\bigwedge_{i\in A}x_i^-\right)\right).\end{aligned} \tag{8}$$

This aggregation is perfectly in line with specifications (i) and (ii) listed at the beginning of this section, but it does not fulfill specification (iii):

i) it admits signed real inputs and weights and outputs a signed value,
ii) as proved in Section 3.1, when the values of parameter $\boldsymbol{\varphi}$ change by adding bounded signed values, each of the four parts can be considered as Sugeno integrals and thus the combination can be considered as an aggregation,
iii) it is derivable with respect to inputs and weights, and those derivatives are not continuous.

Proposition 3.3. *The derivatives of $\mathcal{A}_{MT}$ w.r.t. parameters and inputs are not defined everywhere.*

Proof. The role of parameters and inputs can be exchanged in $\mathcal{A}_{MT}$ so we present the proof of derivation w.r.t. the input values. The proof for the parameters is similar.
By construction, $\exists p, q \in \Omega$ such that
$\bigvee_{i\in\Omega}\left(x_i^+\wedge\varphi_i^+\right)\vee\bigvee_{i\in\Omega}\left(x_i^-\wedge\varphi_i^-\right) = x_p\wedge\varphi_p = \alpha \geq 0$ and
$\bigvee_{i\in\Omega}\left(x_i^-\wedge\varphi_i^+\right)\vee\bigvee_{i\in\Omega}\left(x_i^+\wedge\varphi_i^-\right) = x_q\wedge\varphi_q = \beta \leq 0$.
Since such a case may arise, let's assume that $\alpha = -\beta$. In that case, $\mathcal{A}_{MT}(\boldsymbol{x},\boldsymbol{\varphi}) = \alpha\vee(-\alpha) = 0$.
We show that in such a case the derivatives of $\mathcal{A}_{MT}$ w.r.t. x_p and φ_p may not be defined. There are two cases:

1) $|x_p| > |\varphi_p|$.

Let us consider $\boldsymbol{x}'$ such that $\boldsymbol{x}' = \boldsymbol{x}$ except for x'_p and x_p for witch we assume that $x'_p = x_p + \epsilon$. We can consider ϵ small as being small enough to have $|x_p + \epsilon| > |\varphi_p|$. In such a case, $\frac{\mathcal{A}_{MT}(\boldsymbol{x}',\boldsymbol{\varphi})-\mathcal{A}_{MT}(\boldsymbol{x},\boldsymbol{\varphi})}{\epsilon} = 0/\epsilon = 0$. Hence $\lim_{\epsilon\to 0}\frac{\mathcal{A}_{MT}(\boldsymbol{x}',\boldsymbol{\varphi})-\mathcal{A}_{MT}(\boldsymbol{x},\boldsymbol{\varphi})}{\epsilon} = 0$.

2) $|x_p| \leq |\varphi_p|$.

Let us consider $\boldsymbol{x}'$ such that $\boldsymbol{x}' = \boldsymbol{x}$ except for x'_p and x_p for which we assume that $x'_p = x_p + \epsilon$. We can consider ϵ as being small enough to have $|x_p + \epsilon| \leq |\varphi_p|$. In such a case, $\frac{\mathcal{A}_{MT}(\boldsymbol{x}',\boldsymbol{\varphi})-\mathcal{A}_{MT}(\boldsymbol{x},\boldsymbol{\varphi})}{\epsilon} = (x_p+\epsilon)/\epsilon = x_p/\epsilon + 1$. Hence $\lim_{\epsilon\to 0}\frac{\mathcal{A}_{MT}(\boldsymbol{x}',\boldsymbol{\varphi})-\mathcal{A}_{MT}(\boldsymbol{x},\boldsymbol{\varphi})}{\epsilon} = sign(x_p).\infty$. □

The fact that the derivative is not defined for certain input and parameter values makes the gradient descent method theoretically unfeasible. However, as this situation is potentially quite rare, we propose to replace the derivative by an arbitrary value when it is not defined (see section 4.1).

Worse still, the probability of an MT neuron being blocked in a position such that it no longer transmits information via the chain rule is not low.

Proposition 3.4. *Let* $\Delta = |\underline{\varphi}| \vee |\overline{\varphi}|$ *where* $\overline{\varphi} = \bigvee_{i\in\Omega} \varphi_i$ *and* $\underline{\varphi} = \bigwedge_{i\in\Omega} \varphi_i$. *If* $\forall i \in \Omega$, $|x_i| > \Delta$ *and* $\exists p, q \in \Omega$ *such that* $\varphi_p = -\varphi_q = \Delta$, *thus* $\mathcal{A}_{MT}(\boldsymbol{x}, \boldsymbol{\varphi}) = 0$ *and all the derivatives of* $\mathcal{A}_{MT}(\boldsymbol{x}, \boldsymbol{\varphi})$ *w.r.t. the input values equal* 0.

Proof. The proof is trivial. Let us suppose, without any loss of generality, that $\varphi_p \geq 0$ and $\varphi_q \leq 0$.

There are two possible cases:

- either $x_p > 0$ and thus $\bigvee_{i\in\Omega} \left(x_i^+ \wedge \varphi_i^+\right) = \varphi_p$ and $\bigvee_{i\in\Omega} \left(x_i^- \wedge \varphi_i^+\right) = 0$,
- or $x_p < 0$ and thus $\bigvee_{i\in\Omega} \left(x_i^+ \wedge \varphi_i^+\right) = 0$ and $\bigvee_{i\in\Omega} \left(x_i^- \wedge \varphi_i^+\right) = \varphi_p$.

Hence $\bigvee_{i\in\Omega} \left(x_i^+ \wedge \varphi_i^+\right) \vee \bigvee_{i\in\Omega} \left(x_i^- \wedge \varphi_i^+\right) = \varphi_p = \Delta$.

The same type of reasoning leads to $\bigvee_{i\in\Omega} \left(x_i^+ \wedge \varphi_i^-\right) \vee \bigvee_{i\in\Omega} \left(x_i^- \wedge \varphi_i^-\right) = \varphi_q = -\Delta$.

Thus in that case $\mathcal{A}_{MT}(\boldsymbol{x}, \boldsymbol{\varphi}) = \Delta \vee (-\Delta) = 0$ and this result does not depend on the input values. Thus

$$\forall i \in \Omega, \frac{\delta \mathcal{A}_{MT}}{\delta x_i}(\boldsymbol{x}, \boldsymbol{\varphi}) = 0.$$

□

We propose to consider an extension that does not have this defect, mainly caused by an abrupt transition to 0 when the negative and positive parts are equal.

3.4. Additive signed extension: ST-aggregation

To solve the problem caused by the non-continuity induced by the extended max operator, we propose to combine the four terms defined in Section 3.2 in a linear fashion. This combination is no longer a Sugeno integral, but each part of the combination can be considered as a Sugeno integral. This additive extension is compatible with the general neural network approach.

The additive signed extension is given by:

$$\begin{aligned}\mathcal{A}_{ST}(\boldsymbol{x}, \boldsymbol{\varphi}) &= \mathcal{M}(\boldsymbol{x}^+, \boldsymbol{\varphi}^+) - \mathcal{M}(-\boldsymbol{x}^-, \boldsymbol{\varphi}^+) - \mathcal{M}(\boldsymbol{x}^+, -\boldsymbol{\varphi}^-) + \mathcal{M}(-\boldsymbol{x}^-, -\boldsymbol{\varphi}^-)), \\ &= \bigvee_{i\in\Omega} \left(x_i^+ \mathbin{\textcircled{\wedge}} \varphi_i^+\right) + \bigvee_{i\in\Omega} \left(x_i^- \mathbin{\textcircled{\wedge}} \varphi_i^+\right) + \bigvee_{i\in\Omega} \left(x_i^+ \mathbin{\textcircled{\wedge}} \varphi_i^-\right) + \bigvee_{i\in\Omega} \left(x_i^- \mathbin{\textcircled{\wedge}} \varphi_i^-\right). \end{aligned} \tag{9}$$

Let $\overline{\varphi} = \bigvee_{i\in\Omega} \varphi_i$ and $\underline{\varphi} = \bigwedge_{i\in\Omega} \varphi_i$, then the aggregated value $\mathcal{A}_{ST}(\boldsymbol{x}, \boldsymbol{\varphi})$ belongs to $[-2.\Delta, 2.\Delta]$, where $\Delta = |\underline{\varphi}| \vee |\overline{\varphi}|$ since two members of this sum belong to $[0, \Delta]$ while other members belong to $[-\Delta, 0]$.

3.5. Adding a bias

We have not mentioned an essential point, which is the bias associated with each neuron. It plays an important role in the transmission of an inhibitory or excitatory message to the next neuron. Adding a bias to this type of neuron can be done in a similar way to conventional neurons, by adding a value to the input of each neuron that does not depend on the inputs. Let's call this additional input φ_0. Let $\varphi_0^+ = \max(0, \varphi_0)$ and $\varphi_0^- = \min(0, \varphi_0)$. In both extensions, the four terms have to be replaced by a term taking the bias into account, i.e.

- $\bigvee_{i\in\Omega} \left(x_i^+ \wedge \varphi_i^+\right) \longrightarrow \varphi_0^+ \vee \left(\bigvee_{i\in\Omega} \left(x_i^+ \wedge \varphi_i^+\right)\right)$,
- $\bigvee_{i\in\Omega} \left(x_i^- \wedge \varphi_i^+\right) \longrightarrow \varphi_0^- \vee \left(\bigvee_{i\in\Omega} \left(x_i^- \wedge \varphi_i^+\right)\right)$,
- $\bigvee_{i\in\Omega} \left(x_i^+ \wedge \varphi_i^-\right) \longrightarrow \varphi_0^- \vee \left(\bigvee_{i\in\Omega} \left(x_i^+ \wedge \varphi_i^-\right)\right)$,
- $\bigvee_{i\in\Omega} \left(x_i^- \wedge \varphi_i^-\right) \longrightarrow \varphi_0^+ \vee \left(\bigvee_{i\in\Omega} \left(x_i^- \wedge \varphi_i^-\right)\right)$.

3.6. Neural interpretation

A possible neural interpretation might be as follows. The electrical flow that transmits information from one neuron to another can be positive (positive activation solicitation), negative (negative activation solicitation) or zero (no solicitation).

Each synapse has an associated threshold: a positive threshold means that the receiving neuron is expecting a positive solicitation, while a negative threshold indicates that a negative solicitation is expected. The neuron aggregates these solicitations activate the next neuron. In the MT-aggregation, the neuron outputs its best or worst state – or 0 when there is an exact counterbalance between the best and worst case. In the ST version, positive and negative states are additively counterbalanced. The difference in values between thresholds associated with synapses enables us to differentiate the importance of each synapse in the aggregation.

This interpretation is close to that of conventional neurons. In additive aggregation, each synapse is associated with a weight. If the weight is of the same sign as the input, then the response is used positively. If the weight is of the opposite sign, the response is negative. The set of responses for each synapse is aggregated additively. Then a threshold-based function transforms this aggregation into a positive or negative activation for the next neuron (depending on the activation function used).

The MT approach is rather a *winner takes in all*, where the ST approach looks like a compromise between opposing coalitions. The interpretation of the classical approach is less easy because of the multiplicative weights.

Example 3.1. Suppose you're part of a financial group that wants to make a share deal with another group, based on information provided by the group's experts. Each expert gives you his or her opinion: sell (increase the group's capital base) or buy (decrease the group's capital base). You have an opinion on each member of the panel. Some are positive (you think the expert is loyal

to the group) and others are negative (you think the expert is trying to favor an opposing group). The weight associated with the i^{th} synapse (i^{th} expert) is the maximum transaction you would accept to make on the basis of this expert.

Naturally, if the i^{th} expert is associated with a negative threshold, then it is more prudent to make the reverse transaction, while taking into account the maximum threshold of this transaction. On the basis of the opinions of all the experts you have consulted, you pass your opinion on this transaction to your management. In the MT approach, this opinion is the maximum transaction you arrive at after consulting the experts. In the event of the buy and sell opinions arriving at the same sum, you prefer to abstain. The ST approach gives an equivalent but more nuanced result. However, even in this case, a small variation in the entries leads to a small variation in the aggregation, which is not the case with the MT approach (as shown in Section 3.3).

We can assume that you're not the only one to be consulted by your superior, and that your colleagues have also received information from the group's expert panel. In this way, quantitative investment information moves from one decision-making layer to another one right up to the final decision-maker who commits to the transaction.

In a context like this, an entirely additive aggregation would make no sense, any more than multiplicative weights would make sense, unless you assume an expected value like aggregation. Indeed, multiplying the amount proposed for the transaction by the i^{th} expert by a factor and adding this to the results of the other experts has no consonant interpretation, unless we assume that the sum of the weights is unitary. Finally, since we're talking about modelling human-like reasoning here, *the ordinal theory based only on comparisons is more in line with human subjective preferences than the cardinal theory since it is hardly assumed that ordinary people make numerical calculations in their brains* [34].

The MT and ST approaches give qualitatively close results, with the main difference being that the transition in the case of close positive and negative appraisals is smoother (derivable) in the ST case than in the MT case.

4. Computing MT and ST aggregation derivatives

In order to perform learning, by replacing in a neural network, the non-linear equation of the neural aggregation by an MT- or an ST-aggregation, it must be possible to derive both expressions (7) and (9) with respect to $\boldsymbol{x}$ and $\boldsymbol{\varphi}$. Although these expression are non-linear, an approximate derivative may be obtained [36].

To perform these derivations, a preliminary remark must be made about Expression (2): as shown in [21], a small decrease in the i^{th} value of $\boldsymbol{\varphi}$ induces a variation in the value of $\bigvee_{i\in\Omega}(x_i \wedge \varphi_i)$ if and only if $\varphi_i = \bigvee_{i\in\Omega}(x_i \wedge \varphi_i)$, i.e.

$$\frac{\delta\mathcal{M}}{\delta\varphi_i}(\boldsymbol{x},\boldsymbol{\varphi}) = \begin{cases} 1 & \text{if } \mathcal{M}(\boldsymbol{x},\boldsymbol{\varphi}) = \varphi_i, \\ 0 & \text{else}, \end{cases} \tag{10}$$

The approximation of the partial derivatives of $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi})$ is based on replacing the notion of derivative by that of left-hand derivative, i.e.

$$\frac{\delta \mathcal{M}}{\delta \varphi_i}(\boldsymbol{x}, \boldsymbol{\varphi}) \approx \lim_{\epsilon \to 0^+} \frac{\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}) - \mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}^{-\epsilon,i})}{\epsilon},$$

where $\boldsymbol{\varphi}^{-\epsilon,i}$ is the vector such that $\forall j \neq i$, $\varphi_j^{-\epsilon,i} = \varphi_j$ and $\varphi_i^{\epsilon,i} = \varphi_i - \epsilon$ $(\epsilon > 0)$.

Let $\alpha = \mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi})$.

- If $\varphi_i < \alpha$, then by construction $\varphi_i - \epsilon < \alpha$. Thus $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}^{-\epsilon,i}) = \alpha$, therefore $\frac{\delta \mathcal{M}}{\delta \varphi_i}(\boldsymbol{x}, \boldsymbol{\varphi}) = 0$.
- If $\varphi_i = \alpha$, then $\varphi_i < x_i$ and thus $\mathcal{M}(\boldsymbol{x}, \boldsymbol{\varphi}^{-\epsilon,i}) = \alpha - \epsilon$, therefore $\frac{\delta \mathcal{M}}{\delta \varphi_i}(\boldsymbol{x}, \boldsymbol{\varphi}) = 1$.

The construction is identical for partial derivatives with respect to inputs.

Now computing the derivation of Expression(9) w.r.t. parameter values or input values is trivial because of the additive combination. As pointed out previously, the case of deriving Expression (7) is less straightforward, and may result in zero values for any element of the input and/or parameter vectors, or to infinite values for some of them.

4.1. Derivation of MT-aggregation

Let $\alpha = \left(\bigotimes_{i\in\Omega}\left(x_i^+ \oslash \varphi_i^+\right)\right) \otimes \left(\bigotimes_{i\in\Omega}\left(x_i^- \oslash \varphi_i^-\right)\right)$
and $\beta = \left(\bigotimes_{i\in\Omega}\left(x_i^+ \oslash \varphi_i^-\right)\right) \otimes \left(\bigotimes_{i\in\Omega}\left(x_i^- \oslash \varphi_i^+\right)\right)$.

By construction, $\alpha \in [0, \Delta]$ and $\beta \in [-\Delta, 0]$.

Referring to the preliminary remark and Proposition 3.3, there can be two cases.

1) $\alpha \neq -\beta$, thus $\forall i \in \Omega$,

$$\frac{\delta \mathcal{A}_{MT}}{\delta x_i}(\boldsymbol{x}, \boldsymbol{\varphi}) = \begin{cases} 1 & \text{if } x_i = \alpha, \\ -1 & \text{if } x_i = \beta, \\ 0 & \text{else .} \end{cases} \qquad \frac{\delta \mathcal{A}_{MT}}{\delta \varphi_i}(\boldsymbol{x}, \boldsymbol{\varphi}) = \begin{cases} 1 & \text{if } \varphi_i = \alpha, \\ -1 & \text{if } \varphi_i = \beta, \\ 0 & \text{else .} \end{cases}$$

2) $\alpha = -\beta$ thus $\forall i \in \Omega$,

- either $|x_i \oslash \varphi_i| = \alpha$ then $\frac{\delta \mathcal{A}_{MT}}{\delta \varphi_i}(\boldsymbol{x}, \boldsymbol{\varphi})$ is undetermined, while $\frac{\delta \mathcal{A}_{MT}}{\delta x_i}(\boldsymbol{x}, \boldsymbol{\varphi}) = 0$ or reverse,
- or $\frac{\delta \mathcal{A}_{MT}}{\delta \varphi_i}(\boldsymbol{x}, \boldsymbol{\varphi}) = \frac{\delta \mathcal{A}_{MT}}{\delta x_i}(\boldsymbol{x}, \boldsymbol{\varphi}) = 0$.

In experiments, we avoid this problem by arbitrarily replacing indeterminate values randomly with either 1, 0 or -1.

4.2. Derivation of ST-aggregation

Computation of the derivatives of $\mathcal{M}_{ST}$ w.r.t. input and parameter values is rather easy when considering equation (10).
– Let $\alpha_\star^\bullet = \bigtriangledown_{i\in\Omega}\,(x_i^\bullet \bigtriangleup \varphi_i^\star)$ with $\bullet, \star \in \{+,-\}$.
– Let $\forall p \in \Omega$, $\delta_\star^\bullet(p)$ be the derivative of $\bigtriangledown_{i\in\Omega}\,(x_i^\bullet \bigtriangleup \varphi_i^\star)$ w.r.t. x_p.
– Let $\zeta_\star^\bullet(p)$ be the derivative of $\bigtriangledown_{i\in\Omega}\,(x_i^\bullet \bigtriangleup \varphi_i^\star)$ w.r.t. φ_p.
– Let $sign(\star,\bullet) = 1$, if $\star = \bullet$ and -1 otherwise.

By construction $\delta_\star^\bullet(p) = sign(\star,\bullet)$ if $x_p = \alpha_\star^\bullet$ and 0 otherwise. Symmetrically, $\zeta_\star^\bullet(p) = sign(\star,\bullet)$ if $\varphi_p = \alpha_\star^\bullet$ and 0 otherwise.

Thus $\forall p \in \Omega$, $\frac{\delta \mathcal{A}_{ST}}{\delta x_p}(\boldsymbol{x},\boldsymbol{\varphi}) = \sum_{\bullet,\star\in\{+,-\}} \delta_\star^\bullet(p)$ and $\frac{\delta \mathcal{A}_{ST}}{\delta \varphi_p}(\boldsymbol{x},\boldsymbol{\varphi}) = \sum_{\bullet,\star\in\{+,-\}} \zeta_\star^\bullet(p)$.

$\frac{\delta \mathcal{A}_{ST}}{\delta x_p}(\boldsymbol{x},\boldsymbol{\varphi})$ (and also $\frac{\delta \mathcal{A}_{ST}}{\delta \varphi_p}(\boldsymbol{x},\boldsymbol{\varphi})$) is a sum of four terms, two of which can be equal to 1 or 0 and the other two of which can be equal to -1 or 0. Thus, by construction, both values belong to $\{-2,-1,0,1,2\}$.

5. Experiment

5.1. Learning an ST aggregation

The following experiment shows that it is possible to learn an ST relation based on a dataset by using simple gradient descent.

To achieve this experiment, we considered an ST aggregation relation of ten inputs ($n = 10$). First, the value of each element of $\boldsymbol{\varphi}$ was randomly drawn from a centered normal distribution of standard deviation equal to 20. Then, we drew 500 vectors ${}^k\boldsymbol{x} \in \mathbb{R}^{10}$ ($k = 1 \dots 500$) from a centered normal distribution of standard deviation equal to 30. We subsequently computed the 500 output values ${}^k y = \mathcal{A}_{ST}({}^k\boldsymbol{x},\boldsymbol{\varphi})$ ($k = 1 \dots 500$).

Then we started a $\boldsymbol{\varphi}$ vector learning procedure with the learning base $\{{}^k\boldsymbol{x},{}^k y\}_{k=1\dots500}$ to see if it would be possible to recover the simulated parameter values. The initial value of $\boldsymbol{\varphi}$ is taken at random from the same distribution as that used to simulate this parameter. As a learning criterion to be minimized, we considered the quadratic distance between the current output of the aggregation function and the target. The learning rate was arbitrarily set at 10^{-3} and we performed a learning on 300 epochs (an epoch refers to one learning cycle through the 500 dataset elements). The variations in the logarithm of the quadratic criterion are presented in Figure (1) while those of the logarithm of the distance between the simulated parameter and the actual parameter is presented in Figure (2). After 300 epochs, the quadratic criterion value was $\approx 2.10^{-6}$ and the distance between $\boldsymbol{\varphi}$ and its estimate after 300 iterations was $\approx 5.10^{-5}$. The coefficient of determination value was very close to 1 (in fact $1-10^{-11}$). Similar results were also obtained with a test dataset of 500 elements.

We performed the same experiment by noising the output with normal centered additive noise with a standard deviation of 3 (while knowing that the standard deviation of the output was 11.5). We observed the same uniform convergence. After 300 epochs, the quadratic criterion was ≈ 8.3, while the

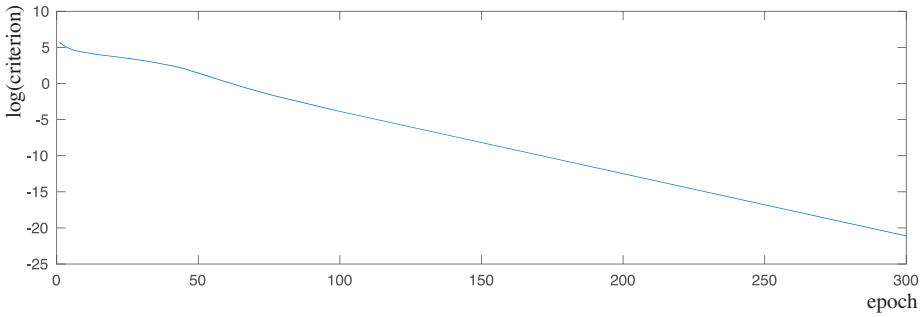


Figure 1: Variations in the quadratic criterion according to the iterations

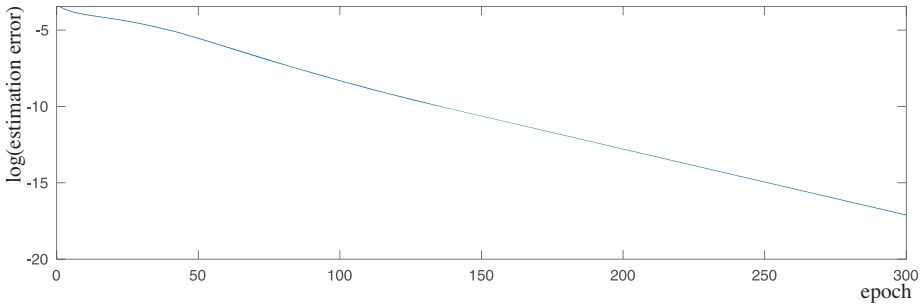


Figure 2: Variations in the distance between φ and its estimation according to the iterations

distance between the estimated and simulated parameters was 0.16. This estimation was very stable, i.e. performing 1000 epochs did not change this result. Of course, convergence depends on the number and variety of examples. This experiment shows that, if the model used is close to the true input/output relationship, then the parameters of the aggregation function can be recovered by regression.

5.2. Comparing MT and ST aggregation based neural networks versus a conventional neural network

In this section, we propose to compare three (fully connected) neural networks with the same architecture (same number of layers and neurons per layer). The first using conventional neurons, i.e. linear aggregation followed by a nonlinear function (here RLU), the second using neurons based on ST aggregation, and the third using neurons based on MT aggregation. For the conventional neural network, the last neuron is obviously a linear neuron (no activation function), while for the ST and MT networks we tested the use of their respective aggregations, and of a linear aggregation.

5.2.1. The datasets

We evaluated the different methods on 8 regression problems from [7]. The databases include the Poker hand dataset, SGEMM GPU kernel performance Dataset[26], 3D road network dataset[18], SARCOS dataset[2], Query Analytics Workloads Dataset[2, 30], Wave Energy Converters Data Set (WECs), The Kyoto Encyclopedia of Genes and Genomes [17] (KEGG) and the Stock dataset [3]. Since the results for all these databases had similar patterns, we only present the results for the 4 following databases.

SGEMM GPU kernel performance Dataset [26]: This dataset measures the running time of a matrix-matrix product A*B = C, where all matrices are 2048×2048 in size, using a parameterizable SGEMM GPU kernel with 241600 possible parameter combinations. For each tested combination, 4 runs were performed and their results are reported in the 4 last columns. All times were measured in milliseconds. There were 14 parameters, the first 10 were ordinal and could only take up to 4 powers of two different values, and the last 4 variables were binary. Out of a total of 1327104 parameter combinations, only 241600 were feasible (due to various kernel constraints). This dataset contains the results of all of these feasible combinations.

Wave Energy Converters Dataset (WECs): This dataset consists of positions and absorbed power outputs of wave energy converters (WECs) in four real wave scenarios from the southern coast of Australia (Sydney, Adelaide, Perth and Tasmania). This dataset is composed of 72000 data with 48 parameters per data entry.

The Kyoto Encyclopedia of Genes and Genomes [17] (KEGG) is the primary resource database of the Japanese GenomeNet service [4] for understanding higher order functional meanings and utilities of a cell or organism from its genome information. KEGG consists of the PATHWAY database for computerized knowledge on molecular interaction networks such as pathways and complexes. This dataset is composed of 64608 data with 27 parameters per data entry.

Stock dataset: The data provided are daily stock prices from January 1988 through October 1991 for 10 aerospace companies. This dataset is composed of 59049 data with 9 parameters per data entry.

5.2.2. Experimental settings and network architecture

For each database, we shuffled all the elements and split the database into five folds. We then trained the neural networks on four folds and tested the remaining fold. Each network was trained with a decreasing learning rate initialized at 1e-3 and a step decay down to 1e-5. The networks were trained for 400 epochs with a batch size of 64, and a stochastic gradient descent (SGD) optimizer. We normalized the input and output data between -1 and 1 using

[2] http://gaussianprocess.org/gpml/data/

[3] https://www.openml.org/search?type=data&status=active&id=223

[4] http://www.genome.ad.jp/

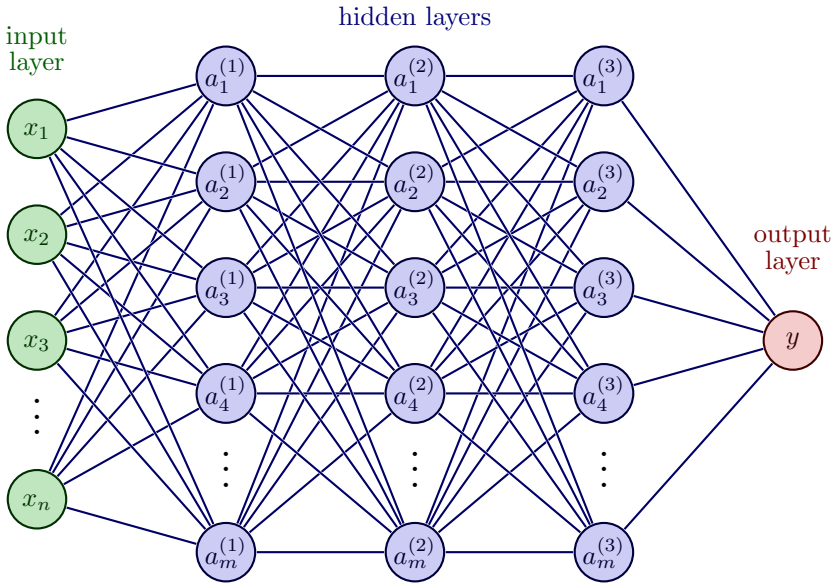


Figure 3: Architecture of the networks used to perform our experiments.

min max values.

We tested several architectures on all bases, but since the results were all very close, we decided to show a representative experiment on four bases. The architecture we used is shown in Figure 3. Each neural network is composed of five layers: an input layer (green), an output layer (red) and three hidden layers of $m = 128$ neurons (blue). The number of inputs (n) depends on the database used. This choice of architecture is entirely empirical and does not favor any of the approaches. Better yet, adding more layers does not substantially change the learning performance of either method. As this architecture is very simple, it does not require any special regularization (batch normalization[16], dropout[31], or weight penalty[19]).

To compare the performances of the different architectures, we looked at the prediction error pattern of the network via the root of mean square error (RMSE), which is the square root of the mean square deviation between the predicted output for a given input and its target, i.e. the output associated with this input in the database.

This variation in the RMSE on the training base highlights the ability of the equation carried out by the neural network to represent the relationship between inputs and outputs. The variations in the RMSE on the test base highlight the ability of the network to generalize what it has learned with the training base on other data from the same database.

5.2.3. Results

Figure (4) plots the variations in the RMSE with epochs obtained by learning with the four databases presented in Section 5.2.1. On each figure, plots with solid lines correspond to the RMSE obtained with the test base (with label Te), while plots with dotted lines correspond to the RMSE obtained on the training base (with label Tr). Blue plots correspond to an ST-based network with the last neuron also being ST-based (label ST). Green plots correspond to an ST-based network with the last neuron being linear (label $ST + 1FC$). Red plots correspond to the conventional neural network, with the last neuron being linear (label FC). Orange plots correspond to an MT-based network with the last neuron also being MT-based (label MT). Brown plots correspond to an MT-based network with the last neuron being linear (label $MT + 1FC$). The pink plot corresponds to a hybrid network alternating between an ST layer and a conventional layer twice in a row.

What can be seen at first on the different plots of Figure (4), and which is the most surprising result of this study is that, although an aggregation based on the Sugeno integral was used, the ST neurons could be used to perform regression learning through a completely conventional regression approach (gradient descent). The same is true for MT neurons, although this aggregation seems less suitable and saturates faster due to the winners take all strategy of this approach (explained in Section 3.6). The second general result is that the ST and MT networks performed better when the last neuron was a linear aggregation, which was similar to the results obtained with a conventional neural network. The third general result is that the overall behavior of the ST network was comparable on both the training dataset and the test dataset. This shows that learning using an ST neural network is relevant and efficient prediction may be achieved. The fourth general result is that ST layers can be perfectly integrated within a network with fully connected layers. Indeed, if we look at the pink curves, we can see that the performance is close to that of a conventional neural network.

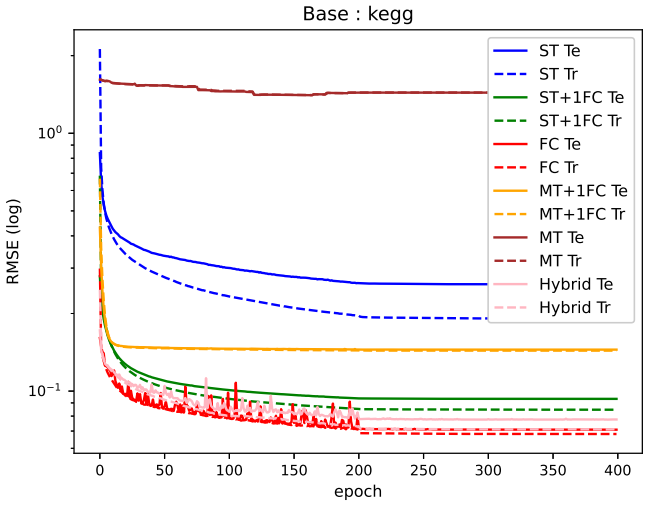
In contrast, regarding the data on which we compared the different structures, the overall results showed faster and more efficient convergence for conventional neurons than for ST neurons. This difference in behavior was considerably reduced when considering an ST neural network whose last neuron was linear, and this reduction was further enhanced in the hybrid network.The addition of a bias, as described in Section 3.5 yields somewhat conflicting results. In 75% of cases, we observed an improvement compared to a version without bias. Also, on hybrid architectures, we found relative improvements of 3%, 7%, and 25% on the stock, wecs, and sgemm databases respectively. Conversely, on the kegg database, the use of bias reduced performance by 31%.

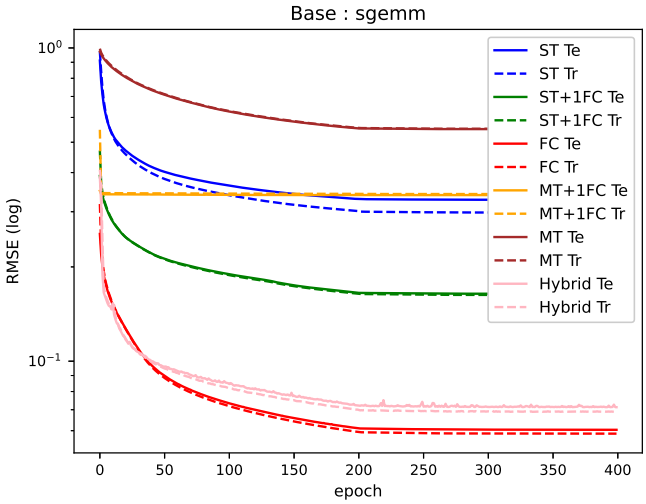
Concerning the training-validation gap (difference between RMSE with training and test data), it was much smaller for conventional neural networks and ST neural networks with a linear output neuron than for the full ST-neuron based network. In the case of classic neural networks, this phenomenon shows that, despite a particularly high number of parameters, overfitting is naturally

regulated by the updating algorithm, i.e. the stochastic gradient, and limited by the complexity of the task. In the case of ST-neural networks, this requires further investigation.

Even if the update algorithm is the same for both networks, it does not apply in the same way. Indeed, regarding conventional networks, the update algorithm modifies the weights that will be applied to the input data of each neuron. Regarding the ST network, the update algorithm modifies how the input data will be positively or negatively updated.

Finally, with regard to MT aggregation, it seems clear that conventional regression based on the chain rule is not optimal. Very quickly, the neural network based on MT aggregation is stuck in a local minimum from which it can no longer escape, despite the last linear layer. This experience in no way invalidates this approach, but shows that it would be necessary to find an updating technique more in line with this modeling.


Base : kegg
RMSE (log)
epoch
ST Te
ST Tr
ST+1FC Te
ST+1FC Tr
FC Te
FC Tr
MT+1FC Te
MT+1FC Tr
MT Te
MT Tr
Hybrid Te
Hybrid Tr


Base : sgemm
RMSE (log)
epoch
ST Te
ST Tr
ST+1FC Te
ST+1FC Tr
FC Te
FC Tr
MT+1FC Te
MT+1FC Tr
MT Te
MT Tr
Hybrid Te
Hybrid Tr

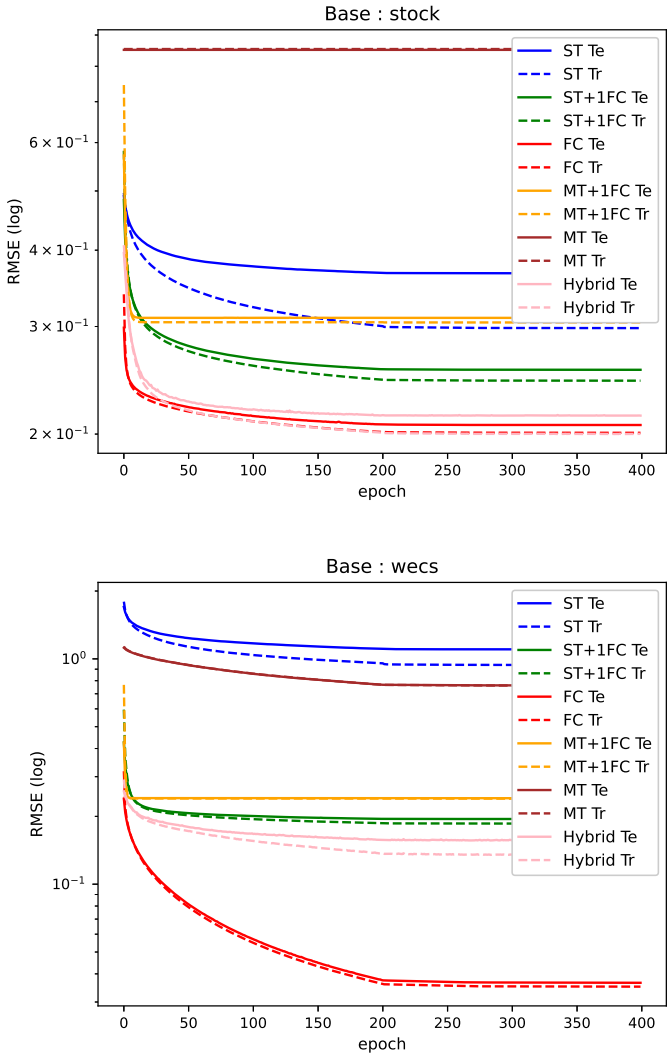


Figure 4: Variations in RMSE when learning on four datasets according to epochs. Solid lines correspond to the RMSE obtained with the test base (with label Te) while dotted lines correspond to the RMSE obtained on the training base (with label Tr). Blue lines correspond to a network based only on ST neurons (label ST). Green plots correspond to a network with ST neurons, but the last neuron is linear (label $ST + 1FC$). Brown plots correspond to a network with only MT neurons. Orange plots correspond to a network with MT neurons, and the last neuron is linear (label $MT + 1FC$). Red plots correspond to a conventional neural network. Pink plots correspond to an hybrid network with 2 layers with ST neurons, and 2 layers with linear neurons.

6. Conclusion and discussion

In this paper we have proposed two signed extensions of the weighted maximum approach. The purpose of these extensions is to replace the coupling of an additive aggregation with a non-linear function, commonly used in neural networks, by a natively non-linear function. Although the conventional neuron is the succession of a weighted addition performing a balance and a non-linearity performing a choice, our main proposal consists of reversing this order by performing the choice first, followed by the additive balance. As a second proposal, we considered an approach based on the work of Grabisch and Sugeno in an more ordinal context. We have shown that these approaches combine four Sugeno integrals w.r.t. a maxitive measure, additively for the ST approach and maximally for the MT approach.

One of the features of these approaches, which differentiates it from the original weighted maximum approach – or even from Sugeno integrals – is that it does not require the fulfillment of any normalization of the weights used in the aggregation or of monotonicity. Due to this dual property, it can be used freely in conventional neural networks without requiring any training modification. Moreover, this kind of aggregation is in perfect agreement with the aggregative model generally used in neural networks in the sense that these aggregations involve the same number of parameters and inputs, contrary to what has been previously reported when it comes to learning a fuzzy integral. In this approach, the use of a bias is non-trivial, unlike in the additive case. We have proposed a method for adding a bias that is consistent with the proposed model. However, experiments have not convincingly shown that adding a bias improves learning. In fact, for some databases, it seems to impair learning. This dependence requires further investigation.

What is surprising, is that this type of approach allows for regressions (i.e. quantitative learning) while the Sugeno integral is known to be used for modeling qualitative relationships. What we can deduce from our experiments is that the use of this new aggregator we proposed makes it possible to achieve learning with performances quite comparable to those achieved with the conventional approach, as far as the ST aggregation is concerned. In the conventional approach, all of the neurons are non-linear except the last one. We noticed that by respecting this procedure with the ST or MT neurons, we obtained, as for the conventional neurons, an increase in performance compared to using the ST or MT aggregation in the last neuron. This increase in performance is fairly easy to explain since, in essence, MT and ST neurons cannot create values other than those present in the threshold and input values. Using additive aggregation at the end of the process corrects this potential defect.

Speaking of future work, as you will have noticed, this is very preliminary work which should pave the way to many complementary studies. For example, it could be particularly interesting to study the approximation power of such an approach, and the advantage of using ST layers in deep neural networks. We proposed this study within the regression framework. It will of course be particularly interesting to study this approach in the classification context, an

area which is more in line with the very origin of this type of approach.

Based on the interpretation of the negative opinions we have given, it should be possible to get back to an interpretation of the decision produced by the network, if the network is not too deep, because otherwise there is a risk of losing this interpretation because of the plethora of cascading neurons.

Finally, we plan to take a closer look at the MT approach and, more specifically, its learning process. It seems clear that gradient descent learning does not really dovetail with this approach. We are considering an alternative approach, inspired, for example, from the work of Baaj[3].

References

[1] S. Abbaszadeh and E. Hüllermeier. Machine learning with the Sugeno integral: The case of binary classification. *IEEE Transactions on Fuzzy Systems*, 29(12):3723–3733, 2021.

[2] C. Anagnostopoulos, F. Savva, and P. Triantafillou. Scalable aggregation predictive analytics - a query-driven machine learning approach. *Applied Intelligence*, 48:2546–2567, 2018.

[3] I. Baaj. Learning rule parameters of possibilistic rule-based system. In *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, 2022.

[4] I. Baaj. On the handling of inconsistent systems of max-min fuzzy relational equations. *Fuzzy Sets and Systems*, 482:108912, 2024.

[5] Q. Brabant, M. Couceiro, D. Dubois, H. Prade, and A. Rico. Learning rule sets and Sugeno integrals for monotonic classification problems. *Fuzzy Sets and Systems*, 401:4–37, 2020.

[6] A. Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological Cybernetics*, 95(1):1–19, 2006.

[7] D. Dua and C. Graff. UCI machine learning repository, 2017.

[8] D. Dubois and H Prade. Weighted minimum and maximum operations in fuzzy set theory. *Information Sciences*, 39(2):205–210, 1986.

[9] D. Dubois and H. Prade. Reasoning and learning in the setting of possibility theory - overview and perspectives. *International Journal of Approximate Reasoning*, page 109028, 2023.

[10] S. Fan, L. Liu, and Y. Luo. An alternative practice of tropical convolution to traditional convolutional neural networks. In *Proceedings of the 5th International Conference on Compute and Data Analysis*, pages 162–168, 02 2021.

[11] Y. Forghani and Sadoghi Y. Fuzzy min–max neural network for learning a classifier with symmetric margin. *Neural Processing Letters*, 42(2):315–353, 2015.

[12] M. Grabisch. The symmetric Sugeno integral. *Fuzzy Sets and Systems*, 139(3):473–490, 2003.

[13] M. Grabisch, J-L. Marichal, R. Mesiar, and E. Pap. Aggregation functions: Means. *Information Sciences*, 181(1):1–22, 2011.

[14] K. Gurney, T. Prescott, and P. Redgrave. A computational model of action selection in the basal ganglia. I. A new functional anatomy. *Biological Cybernetics*, 84(6):401–410, 2011.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, Los Alamitos, CA, USA, 2015.

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456, 2015.

[17] M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya. The kegg databases at genomenet. *Nucleic acids research*, 30(1):42–46, 2002.

[18] M. Kaul, B. Yang, and C. Jensen. Building accurate 3d spatial networks to enable next generation intelligent transportation systems. In *2013 IEEE 14th International Conference on Mobile Data Management*, volume 1, pages 137–146, 2013.

[19] A. Krogh and J. Hertz. A simple weight decay can improve generalization. In J. Moody, S. Hanson, and R. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] L. Li, Z. Qiao, Y. Liu, and Y. Chen. A convergent smoothing algorithm for training max–min fuzzy neural networks. *Neurocomputing*, 260:404–410, 2017.

[22] J.-L. Marichal. *Aggregation Operators for Multicriteria Decision Aid*. PhD thesis, Institute of Mathematics, University of Liège, Liège, Belgium, 1998.

[23] J-L. Marichal. Weighted lattice polynomials. *Discrete Mathematics*, 309(4):814–820, 2009.

[24] O. Mendoza, P. Melin, and G. Licea. A hybrid approach for image recognition combining type-2 fuzzy logic, modular neural networks and the Sugeno integral. *Information Sciences*, 179(13):2078–2101, 2009.

[25] V. Nair and G. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. International Conference on Machine Learning*, pages 807–814, 2010.

[26] C. Nugteren and V. Codreanu. Cltune: A generic auto-tuner for opencl kernels. In *2015 IEEE 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip*, pages 195–202. IEEE, 2015.

[27] Strauss O., Rico A., and Hmidy Y. Macsum: A new interval-valued linear operator. *International Journal of Approximate Reasoning*, 145:121–138, 2022.

[28] H. Prade, A. Rico, and M. Serrurier. Elicitation of Sugeno integrals: A version space learning perspective. In J. Rauch, Z. Raś, P. Berka, and T. Elomaa, editors, *Foundations of Intelligent Systems*, pages 392–401, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[29] I. Rodriguez-Martinez, J. Lafuente, R. Santiago, G. Pereira Dimuro, F. Herrera, and H. Bustince. Replacing pooling functions in convolutional neural networks by linear combinations of increasing functions. *Neural Networks*, 152:380–393, 2022.

[30] F. Savva, C. Anagnostopoulos, and P. Triantafillou. Explaining aggregates for exploratory analytics. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 478–487, 2018.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[32] T. Stewart, X. Choo, and C. Eliasmith. Symbolic reasoning in spiking neurons: A model of the cortex/basal ganglia/thalamus loop. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32, 2010.

[33] M. Sugeno. *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974.

[34] M. Sugeno. Ordinal preference models based on s-integrals and their verification. In S. Li, X. Wang, Y. Okazaki, J. Kawabe, T. Murofushi, and L. Guan, editors, *Nonlinear Mathematics for Uncertainty and its Applications*, pages 1–18, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[35] X. Sun, L. Zhang, and J. Gu. Neural-network based adaptive sliding mode control for Takagi-Sugeno fuzzy systems. *Information Sciences*, 628:240–253, 2023.

[36] L-N. Teow, H. Mui, K. Terrace, and K-F Loe. An effective learning method for max-min neural networks. In *IJCAI'97: Proceedings of the Fifteenth international joint conference on Artifical intelligence*, volume 2, pages 1134–1139, 06 1997.

[37] V. Vapnik. *Statistical learning theory.* Wiley, 1998.

[38] Perwej. Y., A. Dwivedi, N. Akhtar, M. Shan, and D. Dumka. Tropical convolutional neural networks (tcnns) based methods for breast cancer diagnosis. *International Journal of Scientific Research in Science and Technology*, 10:1100 – 1116, 2023.