

Łukasiewicz Equivalent Neural Networks

Antonio Di Nola, Giacomo Lenzi and Gaetano Vitale

Abstract In this paper we propose a particular class of multilayer perceptrons, which describes possibly non-linear phenomena, linked with Łukasiewicz logic; we show how we can name a neural network with a formula and, viceversa, how we can associate a class of neural networks to each formula. Moreover, we introduce the definition of *Łukasiewicz Equivalent Neural Networks* to stress the strong connection between different neural networks via Łukasiewicz logical objects.

Keywords MV-algebras · Riesz MV-algebras · Łukasiewicz Logic · Feedforward Neural Networks · Multilayer Perceptron

1 Introduction

Many-valued logic has been proposed in [1] to model neural networks: it is shown there that, by taking as activation functions ρ the identity truncated to zero and one (i.e., $\rho(x) = (1 \wedge (x \vee 0))$), it is possible to represent the corresponding neural network as combination of propositions of Łukasiewicz calculus.

In [3] the authors showed that multilayer perceptrons, whose activation functions are the identity truncated to zero and one, can be fully interpreted as logical objects, since they are equivalent to (equivalence classes of) formulas of an extension of Łukasiewicz propositional logic obtained by considering scalar multiplication with real numbers (corresponding to Riesz MV-algebras, defined in [4, 5]).

Now we propose more general multilayer perceptrons which describe not necessarily linear events. We show how we can name a neural network with a formula and,

A. Di Nola · G. Lenzi · G. Vitale (✉)
Department of Mathematics, University of Salerno,
Via Giovanni Paolo II 132, 84084 Fisciano (SA), Italy
e-mail: gvitale@unisa.it

A. Di Nola
e-mail: adinola@unisa.it

G. Lenzi
e-mail: gilenzi@unisa.it

viceversa, how we can associate a class of neural networks to each formula; moreover we introduce the idea of *Łukasiewicz Equivalent Neural Networks* to stress the strong connection between (very different) neural networks via Łukasiewicz logical objects.

2 Multilayer Perceptrons

Artificial neural networks are inspired by the nervous system to process information. There exist many typologies of neural networks used in specific fields. We will focus on feedforward neural networks, in particular multilayer perceptrons, as in [3], which have applications in different fields, such as speech or image recognition. This class of networks consists of multiple layers of neurons, where each neuron in one layer has directed connections to the neurons of the subsequent layer. If we consider a multilayer perceptron with n inputs, l hidden layers, ω_{ij}^h as weight (from the j th neuron of the hidden layer h to the i th neuron of the hidden layer $h + 1$), b_i real number and ρ an activation function (a monotone-nondecreasing continuous function), then each of these networks can be seen as a function $F : [0, 1]^n \rightarrow [0, 1]$ such that

$$F(x_1, \dots, x_n) = \rho\left(\sum_{k=1}^{n^{(l)}} \omega_{0,k}^l \rho\left(\dots \left(\sum_{i=1}^n \omega_{l,i}^1 x_i + b_i\right) \dots\right)\right).$$

3 Łukasiewicz Logic and Riesz MV-Algebras

MV-algebras are the algebraic structures corresponding to Łukasiewicz many valued logic, as Boolean algebras correspond to classical logic. An MV-algebra is a structure $A = (A, \oplus, *, 0)$ that satisfies the following properties:

- $x \oplus (y \oplus z) = (x \oplus y) \oplus z$
- $x \oplus y = y \oplus x$
- $x \oplus 0 = x$
- $x^{**} = x$
- $x \oplus 0^* = 0^*$
- $(x^* \oplus y)^* \oplus y = (y^* \oplus x)^* \oplus x$

The standard MV-algebra is the real unit interval $[0, 1]$, where the constant 0 is the real number 0 and the operations are

$$x \oplus y = \min(1, x + y)$$

$$x^* = 1 - x$$

for any $x, y \in [0, 1]$. Another example of MV-algebra is the standard Boolean algebra $\{0, 1\}$ where all the elements are idempotent, i.e. $x \oplus x = x$ and the \oplus is the connective \vee in classical logic. A further class of examples of MV-algebras are M_n (for each $n \in \mathbb{N}$), where the elements are the continuous functions from the cube $[0, 1]^n$ to the real interval $[0, 1]$ which are piecewise linear with integer coefficients. These functions are called *McNaughton functions* and a major result in MV-algebra theory states that M_n is the free MV-algebra with n generators. For more details on the theory of the MV-algebras see [2].

A Riesz MV-algebra (RMV-algebra) is a structure $(A, \oplus, *, 0, \cdot)$, where $(A, \oplus, *, 0)$ is an MV-algebra and the operation $\cdot : [0, 1] \times A \rightarrow A$ has the following properties for any $r, q \in [0, 1]$ and $x, y \in A$:

- $r \cdot (x \odot y^*) = (r \cdot x) \odot (r \cdot y)^*$
- $r \cdot (x \odot y^*) = (r \cdot x) \odot (r \cdot y)^*$
- $r \cdot (q \cdot x) = (rq) \cdot x$
- $1 \cdot x = x$

where $x \odot y = (x^* \oplus y^*)^*$. We will denote by RM_n the Riesz MV-algebra of the continuous functions from the cube $[0, 1]^n$ to the real interval $[0, 1]$ which are piecewise linear with real coefficients (*Riesz McNaughton functions*). In analogy with the MV-algebra case RM_n is the free Riesz MV-algebra on n generators.

When we talk about a (Riesz) MV-formula, i.e. a syntactic polynomial, we can consider it also as a (Riesz) McNaughton function. Actually, in most of the literature there is no distinction between a (Riesz) McNaughton function and a (Riesz) MV-formula, but it results that, with a different interpretation of the free variables, we can give meaning to MV-formulas by means of other, possibly nonlinear, functions (e.g. we consider generators different from the canonical projections π_1, \dots, π_n , such as polynomial functions, Lyapunov functions, logistic functions, sigmoidal functions and so on).

4 The Connection Between Neural Networks and Riesz MV-Algebras

Already in the first half of *XXth* century Claude Shannon understood the strong relation between switching circuits and Boolean algebras, and so Boolean algebras were (and they are still) used to describe and analyze circuits by algebraic methods. In an analogous way, in [3] the authors describe the connection between (a particular class of) neural networks and RMV-algebras; for instance the authors define the one-layer neural networks which encode $\min(x, y)$ and $\max(x, y)$ as follows:

$$\min(x, y) = \rho(y) - \rho(y - x)$$

$$\max(x, y) = \rho(y) + \rho(x - y)$$

where ρ is the identity truncated to zero and one. In [3] the following theorem was proved.

Theorem 1 *Let the function ρ be the identity truncated to zero and one (i.e., $\rho(x) = (1 \wedge (x \vee 0))$).*

- *For every $l, n, n^{(2)}, \dots, n^{(l)} \in \mathbb{N}$, and $\omega_{i,j}^h, b_i \in \mathbb{R}$, the function $F : [0, 1]^n \rightarrow [0, 1]$ defined as*

$$F(x_1, \dots, x_n) = \rho\left(\sum_{k=1}^{n^{(l)}} \omega_{0,k}^l \rho\left(\dots \left(\sum_{i=1}^n \omega_{l,i}^1 x_i + b_i\right) \dots\right)\right)$$

is a Riesz McNaughton function;

- *for any Riesz McNaughton function f , there exist $l, n, n^{(2)}, \dots, n^{(l)} \in \mathbb{N}$, and $\omega_{i,j}^h, b_i \in \mathbb{R}$ such that*

$$f(x_1, \dots, x_n) = \rho\left(\sum_{k=1}^{n^{(l)}} \omega_{0,k}^l \rho\left(\dots \left(\sum_{i=1}^n \omega_{l,i}^1 x_i + b_i\right) \dots\right)\right).$$

4.1 Łukasiewicz Equivalent Neural Networks

If we consider particular kinds of functions from $[0, 1]^n$ to $[0, 1]^n$ (surjective functions) we can still describe non linear phenomena with a Riesz MV-formula, which, this time, does not correspond to a piecewise linear function but rather to a function which can still be decomposed into “regular pieces” (e.g. a piecewise sigmoidal function).

The crucial point is that we can try to apply, with a suitable choice of generators, all the well established methods of the study of piecewise linear functions to piecewise non-linear functions.

As in [3] we have the following definition.

Definition 1 We denote by \mathcal{N} the class of the multilayer perceptrons such that the all activation functions of all neurons coincide with the identity truncated to zero and one.

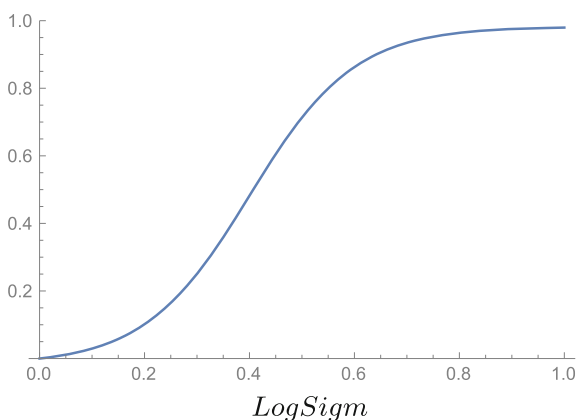
We can generalize this definition as follows.

Definition 2 We call \mathcal{LN} the class of the multilayer perceptrons such that:

- the activation functions of all neurons from the second hidden layer on is $\rho(x) = (1 \wedge (x \vee 0))$, i.e. the identity truncated to zero and one;
- the activation functions of neurons of the first hidden layer have the form $\varphi \circ \rho(x)$ where φ is a continuous functions from $[0, 1]$ to $[0, 1]$.

Example

An example of $\varphi(x)$ could be LogSigm, the logistic sigmoid function “adapted” to the interval $[0, 1]$, as showed in the next figure.



The first hidden layer (which we will call *interpretation layer*) is an interpretation of the free variables (i.e. the input data) or, in some sense, a change of variables.

Roughly speaking we interpret the input variables of the network x_1, \dots, x_n as continuous functions from $[0, 1]^n$ to $[0, 1]$; so, from the logical point of view, we have not changed the (Riesz) MV-formula which describes the neural network but only the interpretation of the variables.

For these reasons we introduce the definition of *Lukasiewicz Equivalent Neural Networks* as follows.

Definition 3 Given a network in \mathcal{LN} , the Riesz MV-formula associated to it is the one obtained first by replacing $\varphi \circ \rho$ with ρ , and then building the Riesz MV-formula associated to the resulting network in \mathcal{N} .

Definition 4 We say that two networks of \mathcal{LN} are *Lukasiewicz Equivalent* iff the two networks have logically equivalent associated Riesz MV-formulas.

5 Examples of Łukasiewicz Equivalent Neural Networks

Let us see now some examples of Łukasiewicz equivalent neural networks (seen as the functions $\psi(\varphi(\bar{x}))$). In every example we will consider a Riesz MV-formula $\psi(\bar{x})$ with many different φ interpretations of the free variables \bar{x} , i.e. the activation functions of the interpretation layers.

Example 1

A simple one-variable example of Riesz MV-formula could be $\psi = \bar{x} \odot \bar{x}$. Let us plot the functions associated with this formula when the activation functions of the

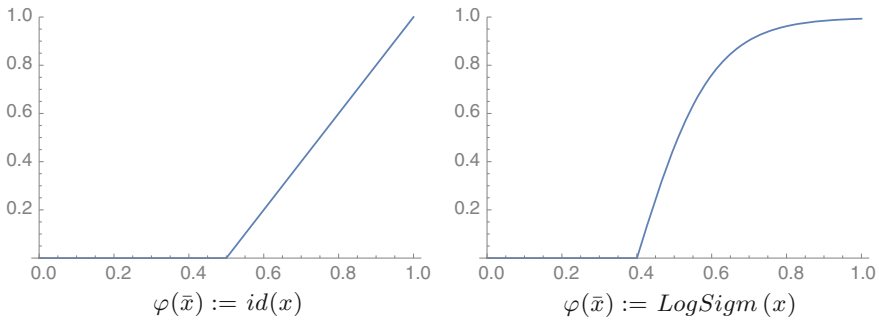


Fig. 1 $\psi(\bar{x}) = \bar{x} \odot \bar{x}$

interpretation layer is respectively the identity truncate function to 0 and 1 and the *LogSigm* (Fig. 1).

In all the following examples we will have (a), (b) and (c) figures, which indicate respectively these variables interpretations:

- (a) x and y as the canonical projections π_1 and π_2 ;
- (b) both x and y as *LogSigm* functions, applied only on the first and the second coordinate respectively, i.e. $LogSigm \circ \rho(\pi_1)$ and $LogSigm \circ \rho(\pi_2)$ (as in the example 1);
- (c) x as *LogSigm* function, applied only on the first coordinate, and y as the cubic function π_2^3 .

We show how, by changing projections with arbitrary functions φ , we obtain functions (b) and (c) “similar” to the standard case (a), which, however, are no more “linear”. The “shape” of the function is preserved, but distortions are introduced.

Example 2: The \odot Operation

We can also consider, in a similar way, the two-variables formula $\psi(\bar{x}, \bar{y}) = \bar{x} \odot \bar{y}$, which is represented in the following graphs (Fig. 2).

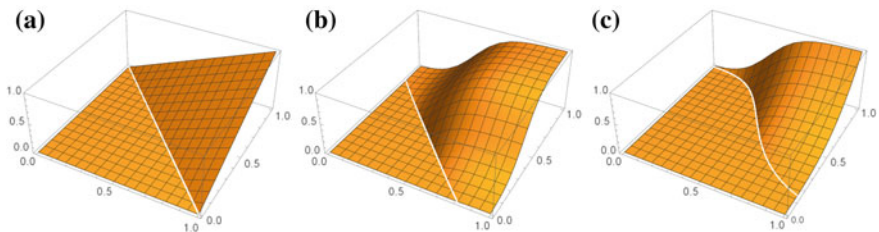


Fig. 2 $\psi(\bar{x}, \bar{y}) = \bar{x} \odot \bar{y}$

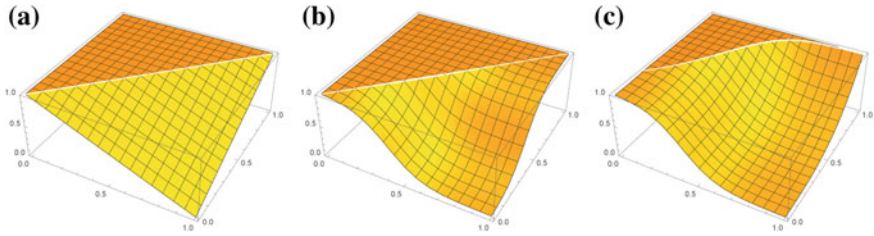


Fig. 3 $\psi(\bar{x}, \bar{y}) = \bar{x} \rightarrow \bar{y}$

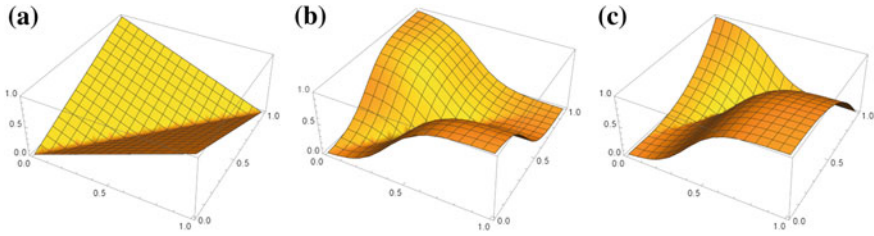


Fig. 4 $\psi(\bar{x}, \bar{y}) = (\bar{x} \odot \bar{y}^*) \oplus (\bar{x}^* \odot \bar{y})$

Example 3: The Łukasiewicz Implication

As in classical logic, also in Łukasiewicz logic we have *implication* (\rightarrow), a propositional connective which is defined as follows: $\bar{x} \rightarrow \bar{y} = \bar{x}^* \oplus \bar{y}$ (Fig. 3).

Example 4: The Chang Distance

An important MV-formula is $(\bar{x} \odot \bar{y}^*) \oplus (\bar{x}^* \odot \bar{y})$, called *Chang Distance*, which is the absolute value of the difference between x and y (in the usual sense) (Fig. 4).

6 Conclusions and Future Investigations

To sum up we:

1. propose \mathcal{LN} as a privileged class of multilayer perceptrons;
2. link \mathcal{LN} with Łukasiewicz logic (one of the most important many-valued logics);
3. show that we can use many properties of (Riesz) McNaughton functions for a larger class of functions;
4. propose an equivalence between particular types of multilayer perceptrons, defined by Łukasiewicz logic objects;
5. compute many examples of Łukasiewicz equivalent multilayer perceptrons to show the action of the free variables interpretation.

We think that using (in various ways) the *interpretation layer* it is possible to encode and describe many phenomena (e.g. degenerative diseases, distorted signals, etc.), always using the descriptive power of the Łukasiewicz logic formal language.

In the future investigations we will focus on:

- the composition of many multilayer perceptrons, to describe more complicated phenomena and their relations;
- the implementation of a back propagation model, regarded as a dynamical system.

References

1. Castro, J.L., Trillas, E.: The logic of neural networks. *Mathw. Soft Comput.* **5**, 23–27 (1998)
2. Cignoli, R., D'Ottaviano, I., Mundici, D.: Algebraic foundations of many valued reasoning, Kluwer (2000)
3. Di Nola, A., Gerla, B., Leustean, I.: Adding real coefficients to Łukasiewicz logic: an application to neural networks. *Fuzzy Logic and Applications*, pp. 77–85. Springer International Publishing, Switzerland (2013)
4. Di Nola, A., Leustean, I.: Riesz MV-algebras and their logic. In: *Proceedings of the 7th Conference of the European Society for Fuzzy Logic and Technology*. Atlantis Press, pp. 140–145 (2011)
5. Di Nola, A., Leustean, I.: Łukasiewicz logic and Riesz spaces. *Soft Comput.* **18**(12), 2349–2363 (2014)