Neural Networks into Łukasiewicz Logic, with Applications to Formal Verification

Sandro Preto[1,†], Marcelo Finger[2,†]

[1]Federal University of ABC (UFABC), Santo André, Brazil
[2]University of São Paulo (USP), São Paulo, Brazil

Abstract

This paper presents an overview of a line of research on representing neural networks in Łukasiewicz logic and applying this representation to formal property verification. The approach relies on the correspondence between neural network computations and rational McNaughton functions, enabling the translation of certain neural architectures into logical formulas. We summarize previously published results on the logical representation of ReLU–TId neural networks and on the encoding of reachability and robustness properties in Łukasiewicz logic.

Keywords

Łukasiewicz logic, neural network, formal verification

1. Introduction

This paper is an abstract of previously published work and summarizes a line of research on representing neural networks in Łukasiewicz logic and its applications to formal verification.

Although neural networks are widely recognized as state of the art in artificial intelligence, they suffer from the black-box problem [1, 2]. That is, they do not provide justifications for the actions they take, nor for the weights learned during the training process. Representing neural networks in a logical language paves the way for the formal verification of their properties and offers a possible approach to addressing the black-box issue.

As neural networks typically operate on rational numbers and take such numbers as input and output values, formulas of a propositional fuzzy logic can represent them by leveraging the truth values in the unit interval $[0, 1]$. Within this approach, the truth value of a formula φ corresponds to the output of a neural network when the propositional variables $X_1, \ldots, X_n$ are assigned truth values matching the network's inputs.

Łukasiewicz logic (Ł) brings together several properties that qualify it as a suitable target system for this endeavor [3, 4]. This logic has a well-established algebraic semantics, and its associated computational problems have been extensively studied. In particular, the satisfiability problem is **NP**-complete and the validity problem is **coNP**-complete [5], which are reasonable complexities for such problems. Moreover, algorithms have been proposed and empirically evaluated for these problems, including studies on the detection of the phase-transition phenomenon [6].

Once a neural network is represented by Łukasiewicz formulas, this representation can be used within Ł-formulas and Ł-entailment instances to express properties of the network. Properties such as *reachability*—that is, whether there exists an input for which the neural network produces a given output—and *robustness*—whether it preserves its output in the presence of small perturbations of the input—are examples of properties that can be encoded in the language of Łukasiewicz logic.

In Section 2, we introduce Łukasiewicz logic and, in Section 3, we present results concerning the translation of neural networks into Łukasiewicz logic formulas. Section 4 shows how the properties

LOGICNN 2026: Workshop on Logical Methods for Neural Network Analysis, July 25, 2026, Lisbon, Portugal
† These authors contributed equally.
sandro.preto@ufabc.edu.br (S. Preto); mfinger@ime.usp.br (M. Finger)
0000-0003-4448-5364 (S. Preto); 0000-0002-1391-1175 (M. Finger)

Table 1
Derived operators in Ł

Operator	Ł-valuation
Conjunction: $\varphi \odot \psi =_{\text{def}} \neg(\neg\varphi \oplus \neg\psi)$	$v(\varphi \odot \psi) = \max(0, v(\varphi) + v(\psi) - 1)$
Maximum (weak disjunction): $\varphi \vee \psi =_{\text{def}} \neg(\neg\varphi \oplus \psi) \oplus \psi$	$v(\varphi \vee \psi) = \max(v(\varphi), v(\psi))$
Minimum (weak conjunction): $\varphi \wedge \psi =_{\text{def}} \neg(\neg\varphi \vee \neg\psi)$	$v(\varphi \wedge \psi) = \min(v(\varphi), v(\psi))$
Implication: $\varphi \to \psi =_{\text{def}} \neg\varphi \oplus \psi$	$v(\varphi \to \psi) = \min(1, 1 - v(\varphi) + v(\psi))$
Bi-implication: $\varphi \leftrightarrow \psi =_{\text{def}} (\varphi \to \psi) \wedge (\psi \to \varphi)$	$v(\varphi \leftrightarrow \psi) = 1 - \lvert v(\varphi) - v(\psi)\rvert$
Truncated subtraction: $\varphi \ominus \psi =_{\text{def}} \varphi \odot \neg\psi$	$v(\varphi \ominus \psi) = \max(0, v(\varphi) - v(\psi))$

of reachability and robustness can be encoded in Łukasiewicz logic. Finally, Section 5 indexes the published work in this line of research and outlines future directions.

2. Łukasiewicz Logic

Łukasiewicz logic (Ł) is based on a propositional language $\mathcal{L}_Ł$ freely generated from a countable set of propositional variables through the binary Ł-disjunction operator $\oplus$ and the unary Ł-negation operator $\neg$. We denote Ł-formulas (in $\mathcal{L}_Ł$) by lower Greek letters. For the semantics of Ł, a Ł-valuation is a function $v : \mathcal{L}_Ł \to [0, 1]$ that obeys, for $\varphi, \psi \in \mathcal{L}_Ł$:

$$v(\neg\varphi) = 1 - v(\varphi);\ \text{and} \tag{1}$$

$$v(\varphi \oplus \psi) = \min(1, v(\varphi) + v(\psi)). \tag{2}$$

From Ł-conjunction and Ł-negation one may derive the Ł-operators in Table 1.

We say that a Ł-formula $\varphi \in \mathcal{L}_Ł$ is *satisfiable* if there is a Ł-valuation v such that $v(\phi) = 1$; in this case we say that v *satisfies* ϕ and that φ is *satisfied* by v. A set of formulas $\Phi \subseteq \mathcal{L}_Ł$ is *satisfiable* if there is a valuation v that simultaneously satisfies all formulas in Φ; again, we say in this case that v *satisfies* Φ and that Φ is *satisfied* by v. An Ł-*entailment instance* $\Phi \models \varphi$, where $\varphi \in \mathcal{L}_Ł$ and $\Phi \subseteq \mathcal{L}_Ł$ is *valid* iff φ is satisfied by every Ł-valuation that satisfies Φ; formulas in Φ are called *premises* and formula φ is the *conclusion*.

Some important properties for our encodings are: $v(\varphi) \leq v(\psi)$ iff $\varphi \to \psi$ is satisfied by v; and $v(\varphi) = v(\psi)$ iff $\varphi \leftrightarrow \psi$ is satisfied by v, for formulas $\varphi, \psi \in \mathcal{L}_Ł$ and Ł-valuation v.

3. Neural Networks into Łukasiewicz Logic

The McNaughton theorem identifies formulas of Łukasiewicz logic with *McNaughton functions*: continuous piecewise linear functions from $[0, 1]^n$ to $[0, 1]$ whose linear pieces have integer coefficients [7, 8]. *Rational McNaughton functions* generalize McNaughton functions by allowing rational numbers as coefficients. One way to represent such functions within a logical system is to rely on extensions of Ł, such as rational Łukasiewicz logic, which introduce division operators δ_n inducing division by $n \in \mathbb{N}^*$ in the semantics [9].

Another way to handle rational McNaughton functions, without extending Ł, is to employ *semantics modulo satisfiability* [10, 11, 12]. In this approach, rational McNaughton functions are represented by a pair $\langle\varphi, \Phi\rangle$, where φ is a Ł-formula encoding the function and Φ is a set of Ł-formulas constraining Ł-valuations to those satisfying Φ. For instance, any constant $\frac{a}{b} \in [0,1] \cap \mathbb{Q}$, where $a, b \in \mathbb{N}$, $a \leq b$ and $b \neq 0$, can be represented by the pair $\langle aZ_{\frac{1}{b}}, \varphi_{\frac{1}{b}}\rangle$, where $Z_{\frac{1}{b}}$ is a propositional variable and

$$\varphi_{\frac{1}{b}} \doteq Z_{\frac{1}{b}} \leftrightarrow \neg(b-1)Z_{\frac{1}{b}}.$$

Here, $nZ_{\frac{1}{b}}$ denotes the n-fold strong Ł-disjunction $Z_{\frac{1}{b}} \oplus \cdots \oplus Z_{\frac{1}{b}}$, for $n \in \mathbb{N}$. In this way, any Ł-valuation v satisfying the formula $\varphi_{\frac{1}{b}}$ assigns the truth value $\frac{a}{b}$ to the formula $aZ_{\frac{1}{b}}$. In the representation $\langle\varphi, \Phi\rangle$ of a nonconstant rational McNaughton function $f : [0,1]^n \to [0,1]$, there are distinguished propositional variables $X_1, \ldots, X_n$ such that, for every Ł-valuation v satisfying Φ, we have

$$f\big(v(X_1), \ldots, v(X_n)\big) = v(\varphi).$$

Moreover, for every point $\langle x_1, \ldots, x_n\rangle \in [0,1]^n$, there exists at least one Ł-valuation v satisfying Φ such that $v(X_1) = x_1, \ldots, v(X_n) = x_n$.

Representing rational McNaughton functions within the plain Ł system via semantics modulo satisfiability allows one to leverage existing algorithms—together with their empirically evaluated implementations—for satisfiability solving and theorem proving in Ł.

The functional expressive power of Łukasiewicz logic can then be employed to represent neural networks. Fully connected feedforward neural networks with ReLU activation function—given by $\max(0, x)$—in the hidden layers and truncated identity activation function—given by $\max(0, \min(1, x))$—in the output layer compute exactly rational McNaughton functions. That is, each output value is a rational McNaughton function of the network's input values [13]. We call these *ReLU–TId neural networks.*

Algorithms have been proposed to effectively translate a ReLU–TId neural network into a representation modulo satisfiability pair $\langle\varphi, \Phi\rangle$. The translation proceeds in two steps. First, the rational McNaughton functions computed by the network are put into *regional format*, which explicitly encodes each linear piece of the function together with the region of the domain $[0,1]^n$ on which that piece coincides with the function [14]. Second, each function in regional format is represented modulo satisfiability in polynomial time [11, 12]. While the second step runs in polynomial time, the first may incur an exponential blow-up in the number of neurons of the neural network. Still, the regional formats obtained in the first step may not be sufficiently refined for the second step to yield exact representations, and may therefore require further refinements.

Since a ReLU–TId neural network with m output neurons computes m rational McNaughton functions, one for each output neuron, a representation modulo satisfiability of the network can be viewed as a sequence of pairs

$$\langle\varphi_1, \Phi_1\rangle, \ldots, \langle\varphi_m, \Phi_m\rangle.$$

Each pair $\langle\varphi_i, \Phi_i\rangle$ represents the rational McNaughton function computed by the i-th output neuron.

4. Applications to Formal Verification

In this section, we present the encoding of two properties of binary classification neural networks in Łukasiewicz logic. Let $\langle\varphi, \Phi\rangle$ denote the representation modulo satisfiability of a rational McNaughton function $f : [0,1]^n \to [0,1]$, computed by such a network, assumed to have a single output neuron.

The *reachability* property states that a neural network attains at least a given constant value $\frac{a}{b} \in [0,1]$, where $a, b \in \mathbb{N}$, $a \leq b$ and $b \neq 0$. That is, it asks whether there exist input values $x_1, \ldots, x_n \in [0,1]$ such that $f(x_1, \ldots, x_n) \geq \frac{a}{b}$. The network reaches at least the value $\frac{a}{b}$ if, and only if, Ł-formula

$$\bigwedge \Phi \wedge \varphi_{\frac{1}{b}} \wedge aZ_{\frac{1}{b}} \to \varphi$$

is satisfiable. Further details can be found in [15].

For the *robustness* property, we first clarify how a binary decision is obtained from the network output. If the output value is greater than or equal to 0.5, the network decides positively; otherwise, it decides negatively. Such output values are commonly interpreted as probabilities of a positive decision. We say that the network is robust with respect to a "large" probability threshold $\frac{a}{b} \geq \frac{1}{2}$ and a "small" perturbation bound $\frac{\alpha}{\beta} \in [0, 1]$ if, whenever

$$f(x_1, \ldots, x_n) \geq \frac{a}{b},$$

it holds that

$$f(x_1 + p_1, \ldots, x_n + p_n) \geq \frac{1}{2},$$

for all perturbations $p_i \in \mathbb{R}$ such that $|p_i| \leq \frac{\alpha}{\beta}$. The constants satisfy $a, b, \alpha, \beta \in \mathbb{N}$, $a \leq b$, $\alpha \leq \beta$, $b \neq 0$ and $\beta \neq 0$. The network is robust if, and only if, the corresponding Ł-entailment instance

$$\begin{array}{rl} \Phi,\ \Phi',\ \varphi_{\frac{1}{\beta}},\ \varphi_{\frac{1}{b}},\ \varphi_{\frac{1}{2}}, & \\ P_1 \to \alpha Z_{\frac{1}{\beta}},\ \ldots,\ P_n \to \alpha Z_{\frac{1}{\beta}}, & \\ (X_1' \leftrightarrow X_1 \oplus P_1) \vee (X_1' \leftrightarrow X_1 \ominus P_1), & \\ \ldots, & \\ (X_n' \leftrightarrow X_n \oplus P_n) \vee (X_n' \leftrightarrow X_n \ominus P_n), & \\ aZ_{\frac{1}{b}} \to \varphi & \models Z_{\frac{1}{2}} \to \varphi'. \end{array}$$

is valid. The pair $\langle \varphi', \Phi' \rangle$ is a copy of $\langle \varphi, \Phi \rangle$ obtained by replacing the propositional variables X_i in the latter pair with fresh propositional variables X_i' in the former. The propositional variables P_i stand for perturbations of the input values.

Let v be a Ł-valuation satisfying the premises of the Ł-entailment instance above. Then:

- The satisfaction of Φ, Φ', $\varphi_{\frac{1}{\beta}}$, $\varphi_{\frac{1}{b}}$, and $\varphi_{\frac{1}{2}}$ ensures that the representation modulo satisfiability is respected and that the required constants are defined;
- The satisfaction of the formulas $P_i \to \alpha Z_{\frac{1}{\beta}}$ ensures that each perturbation is at most $\frac{\alpha}{\beta}$;
- The satisfaction of the formulas $(X_i' \leftrightarrow X_i \oplus P_i) \vee (X_i' \leftrightarrow X_i \ominus P_i)$ ensures that the propositional variables X_i' are evaluated as perturbed versions of the values assigned to the propositional variables X_i, according to the values assigned to P_i;
- The value $v(\varphi)$ is the output of the network on inputs $v(X_1), \ldots, v(X_n)$;
- The satisfaction of formula $aZ_{\frac{1}{b}} \to \varphi$ ensures that the output of the network is at least $\frac{a}{b}$ for input values $v(X_1), \ldots, v(X_n)$;
- The value $v(\varphi')$ is the output of the network on the perturbed inputs $v(X_1'), \ldots, v(X_n')$.

Thus, if the Ł-entailment instance is valid, every valuation satisfying the premises also satisfies the conclusion $Z_{\frac{1}{2}} \to \varphi'$. Hence, the output of the network on the perturbed input values is at least $\frac{1}{2}$. The validity of this Ł-entailment instance therefore yields robustness of the network. Notice that such robustness property is global, since it is verified for all input values of the network. Further details can be found in [15].

5. Published Work and Future Perspectives

This line of research has been developed over the past years through a series of publications addressing different aspects of the logical representation of neural networks in Łukasiewicz logic and its applications to formal verification.

The representation of rational McNaughton functions directly in Łukasiewicz logic via semantics modulo satisfiability, including an algorithm that takes such functions in regional format as input, was investigated in [10, 11, 12]. The extraction of the functions computed by ReLU–TId neural networks into regional format is developed in [14]. The encoding of reachability and robustness properties for binary classification neural networks is detailed in [12]. In addition, the empirical behavior of Ł-solvers for satisfiability and theorem proving when applied to these formal verification instances was studied in [16]. A more comprehensive overview of this line of research can be found in [13].

As a promising future perspective, we aim to avoid the exponential blow-up arising from the extraction of functions in regional format from ReLU–TId neural networks by revising the translation approach. Instead of a two-step translation, we plan to propose a polynomial-time direct representation modulo satisfiability that is closely tied to the neuronal architecture of the network. Addressing this exponential blow-up would be an important step toward making these methods applicable to larger real-world neural networks. This approach also might be generalized to other neural network architectures. Indeed, preliminary results establishing representations of recurrent neural networks in Łukasiewicz logic can already be found in [17].

Another source of complexity comes from the underlying computational problems of Łukasiewicz logic. Although satisfiability and validity are, respectively, **NP**-complete and **coNP**-complete [5], these complexity classes are expected for many exact formal verification problems. Moreover, besides the existing algorithms for satisfiability solving and theorem proving in Ł, one may also pursue approximate solutions for formal verification queries by using some framework of approximate logics for Ł, as the one in [18].

Another still little-explored approach consists in approximating neural networks by rational McNaughton functions [19]. Indeed, continuous functions can be approximated by rational McNaughton functions via a Weierstrass-like approximation result [20, 21]. Besides potentially yielding regional format encodings of lower complexity for ReLU–TId networks, this approach may also provide logical representations for neural architectures beyond the ReLU–TId setting.

Acknowledgments

This work was carried out at the Center for Artificial Intelligence (C4AI-USP), with support by the University of São Paulo and the São Paulo Research Foundation (FAPESP), Brasil. Process Number #2019/07665-4. Preto acknowledges that this study was financed, in part, by the São Paulo Research Foundation (FAPESP), Brasil. Process Number #2026/02601-1. Finger was partially supported by CNPQ Grant PQ 302963/2022-7.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to: Grammar and spelling check, Paraphrase and reword. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] D. Castelvecchi, Can we open the black box of AI?, Nature 538 (2016) 20–23.

[2] M. Finger, Logic in times of big data, in: J. A. de Barros, D. Krause (Eds.), A True Polymath: A Tribute to Francisco Antonio Doria, College Publications, 2020, pp. 184–198. URL: www.collegepublications.co.uk/ABF/?00002.

[3] R. Cignoli, I. D'Ottaviano, D. Mundici, Algebraic Foundations of Many-Valued Reasoning, Trends in Logic, Springer Netherlands, 2000.

[4] P. Hájek, Metamathematics of Fuzzy Logic, Trends in Logic, Springer, Dordrecht, 1998.

[5] D. Mundici, Satisfiability in many-valued sentential logic is NP-complete, Theoretical Computer Science 52 (1987) 145–153.

[6] M. Bofill, F. Manyà, A. Vidal, M. Villaret, New complexity results for Łukasiewicz logic, Soft Computing 23 (2019) 2187–2197.

[7] R. McNaughton, A theorem about infinite-valued sentential logic, Journal of Symbolic Logic 16 (1951) 1–13.

[8] D. Mundici, A constructive proof of McNaughton's theorem in infinite-valued logic, The Journal of Symbolic Logic 59 (1994) 596–602.

[9] B. Gerla, Rational Łukasiewicz logic and DMV-algebras, Neural Network World 11 (2001) 579–594.

[10] M. Finger, S. Preto, Probably partially true: Satisfiability for Łukasiewicz infinitely-valued probabilistic logic and related topics, Journal of Automated Reasoning 64 (2020) 1269–1286. URL: https://doi.org/10.1007/s10817-020-09558-9. doi:10.1007/s10817-020-09558-9.

[11] S. Preto, M. Finger, An efficient algorithm for representing piecewise linear functions into logic, Proceedings of the 15th International Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2020), Electronic Notes in Theoretical Computer Science 351 (2020) 167–186. doi:10.1016/j.entcs.2020.08.009.

[12] S. Preto, M. Finger, Efficient representation of piecewise linear functions into Łukasiewicz logic modulo satisfiability, Mathematical Structures in Computer Science 32 (2022) 1119–1144. doi:10.1017/S096012952200010X.

[13] S. Preto, M. Finger, Effective reasoning over neural networks using Łukasiewicz logic, in: P. Hitzler, M. Kamruzzaman Sarker, A. Eberhart (Eds.), Compendium of Neurosymbolic Artificial Intelligence, volume 369 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2023, pp. 609–630. doi:10.3233/FAIA230160.

[14] S. Preto, M. Finger, Regional, lattice and logical representations of neural networks, Proceedings of the 19th International Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2024), Electronic Proceedings in Theoretical Computer Science 421 (2025) 64–79. doi:10.4204/eptcs.421.4.

[15] S. Preto, M. Finger, Proving properties of binary classification neural networks via Łukasiewicz logic, Logic Journal of the IGPL 31 (2023) 805–821. doi:10.1093/jigpal/jzac050.

[16] S. Preto, F. Manyà, M. Finger, Benchmarking Łukasiewicz logic solvers with properties of neural networks, in: 2023 IEEE 53rd International Symposium on Multiple-Valued Logic (ISMVL), IEEE, 2023, pp. 158–163.

[17] S. Preto, Towards logical representations of recurrent neural networks, in: Anais do VI Workshop Brasileiro de Lógica, SBC, 2025, pp. 1–8. doi:10.5753/wbl.2025.6543.

[18] M. Finger, S. Preto, Polyhedral semantics and the tractable approximation of łukasiewicz infinitely-valued logic, Journal of Logic and Computation 35 (2025) exad059. doi:10.1093/logcom/exad059.

[19] J. Lobo, M. Finger, S. Preto, Logical approximation and formal verification of neural networks, in: Anais do V Workshop Brasileiro de Lógica, SBC, 2024, pp. 1–8. doi:10.5753/wbl.2024.2347.

[20] P. Amato, M. Porto, An algorithm for the automatic generation of a logical formula representing a control law, Neural Network World 10 (2000) 777–786.

[21] S. Aguzzoli, D. Mundici, Weierstrass Approximation Theorem and Łukasiewicz Formulas with one Quantified Variable, Physica-Verlag HD, Heidelberg, 2003, pp. 315–335.