Neurosymbolic Reasoning and Learning with Restricted Boltzmann Machines

Son N. Tran[1] , **Artur d'Avila Garcez**[2]
[1]Deakin University
[2]City St George's, University of London
son.tran@deakin.edu.au, a.garcez@citystgeorges.ac.uk

Abstract

This paper presents Logical Boltzmann Machines (LBMs), a novel neurosymbolic system that integrates propositional logic reasoning with learning in Restricted Boltzmann Machines (RBMs). We formally prove that logical satisfiability corresponds to energy minimisation in RBMs, enabling any propositional logic formula to be encoded into the network. Unlike prior approaches limited to if-then rules, LBMs support full propositional logic, including negation, conjunction, disjunction, implication, and biconditional operators. The system performs reasoning by searching for low-energy configurations and demonstrates the ability to integrate learning and reasoning. The original paper is at https://ojs.aaai.org/index.php/AAAI/article/view/25806

1 Main Contributions of the paper

Symbolic knowledge is often incorporated into neural networks as general or transferable rules, commonly using if-then representations and Modus Ponens inference. However, relying solely on Modus Ponens is limiting, as it cannot capture more general reasoning patterns like Modus Tollens or logical expressions involving negation, conjunction, disjunction, and biconditionals. While earlier work established a theoretical link between propositional logic and neural networks, the resulting knowledge representations were often complex and difficult to align with modern learning methods.

This paper introduces a method to translate logical formulae into simple 2-layer neural networks. The networks, called Logical Boltzmann Machines (LBM), work as a neurosymbolic system capable of (i) representing any formula in propositional logic, (ii) reasoning given such knowledge, (iii) learning from knowledge and data. To this end, we show that any strict disjunctive normal form (SDNF) can be represented within a restricted Boltzmann machine (RBM), such that an assignment of variables minimises the energy function if and only if it satisfies the SDNF. We further present effective strategies for translating various logical forms into SDNF, highlighting the practical applicability of our approach.

Definition 1. *A strict DNF (SDNF) is a DNF with at most one conjunctive clause that maps to true for any assignment of truth-values x. A full DNF is a DNF where each propositional variable must appear at least once in every conjunctive clause*

Theorem 1. *Any SDNF* $\varphi \equiv \bigvee_j (\bigwedge_{t\in\mathcal{S}_{T_j}} \mathrm{x}_t \wedge \bigwedge_{k\in\mathcal{S}_{K_j}} \neg \mathrm{x}_k)$ *can be mapped onto an equivalent RBM with energy:*

$$E(\mathbf{x},\mathbf{h}) = -\sum_j h_j(\sum_{t\in\mathcal{S}_{T_j}} x_t - \sum_{k\in\mathcal{S}_{K_j}} x_k - |\mathcal{S}_{T_j}| + \epsilon) \quad (1)$$

where $0 < \epsilon < 1$, $\mathcal{S}_{T_j}$ *and* $\mathcal{S}_{K_j}$ *are, respectively, the sets of indices of the positive and negative literals in each conjunctive clause* j *of the SDNF, and* $|\mathcal{S}_{T_j}|$ *is the number of positive literals in conjunctive clause* j.

2 Relevance to KR

In this section, we demonstrate how knowledge is represented in LBMs by transforming various logical forms into SDNF, and how these representations enable logical reasoning within the networks.

2.1 Knowledge Representation in Neural Networks

Clausal Form.

$$\begin{aligned}\varphi_{\text{CF}} &\equiv \bigvee_{t\in\mathcal{S}_T} \neg\mathrm{x}_t \vee \bigvee_{k\in\mathcal{S}_K} \mathrm{x}_k \\ &\equiv \bigvee_{j\in\mathcal{S}_T\cup\mathcal{S}_K} (\bigwedge_{t\in\mathcal{S}_T\backslash j} \mathrm{x}_t \wedge \bigwedge_{k\in\mathcal{S}_K\backslash j} \neg\mathrm{x}_k \wedge \mathrm{x}'_j)\end{aligned} \quad (2)$$

where $\mathcal{S}\backslash j$ denotes a set $\mathcal{S}$ from which j has been removed. $\mathrm{x}'_j \equiv \neg\mathrm{x}_j$ if $j \in \mathcal{S}_T$. Otherwise, $\mathrm{x}'_j \equiv \mathrm{x}_j$. This SDNF only has $|\mathcal{S}_T| + |\mathcal{S}_K|$ clauses, making the translation efficient.

Disjunctive Normal Form (DNF).

$$\begin{aligned}\varphi_{\text{DNF}} &\equiv \bigvee_{m=1}^{M} \underbrace{(\bigwedge_{t\in\mathcal{S}_T^m} x_t \wedge \bigwedge_{k\in\mathcal{S}_K^m} \neg x_k)}_{\varphi^m} \\ &\equiv \bigvee_{m'\in\mathcal{M}} (\bigwedge_{m\in\mathcal{M}\backslash m'} \varphi^m \wedge \neg\varphi^{m'})\end{aligned} \quad (3)$$

$$\mathcal{M} = \{1, ..., M\} \quad (4)$$

Conjunctive Normal Form (CNF).

$$\varphi_{\text{CNF}} \equiv \bigwedge_{m=1}^{M} (\bigvee_{j \in \mathcal{S}_T^m \cup \mathcal{S}_K^m} (\bigwedge_{t \in \mathcal{S}_T^m \setminus j} \mathrm{x}_t \wedge \bigwedge_{k \in \mathcal{S}_K^m \setminus j} \neg \mathrm{x}_k \wedge \mathrm{x}'_j)) \tag{5}$$

2.2 Reasoning

Reasoning as Sampling Since satisfying assignments correspond to energy minima, they would have the highest probability, as:

$$P(\mathbf{x}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{x},\mathbf{h})}}{Z} \tag{6}$$

Therefore, sampling methods can be applied to search for satisfying assignments. Techniques such as Gibbs sampling are particularly suitable, as they support parallel computation in this context.

Reasoning as Optimisation Although the energy function of an LBM is intractable, its free-energy function can be computed analytically as: $\mathcal{F} = \sum_j \left(-\log\left(1 + \exp\left(c \sum_i w_{ij} x_i + b_j\right)\right)\right)$. Each term in this expression is a scaled negative softplus function, where the scaling factor c, referred to as the *confidence value*, is a non-negative constant. This function yields negative values when the input is positive, and values near zero when the input is negative. The smoothness of the function can be controlled by adjusting the value of c.

Each free-energy term corresponds to a conjunctive clause in the SDNF, linked via the weighted sum $\sum_i w_{ij} x_i + b_j$. When a truth-value assignment $\mathbf{x}$ does not satisfy the formula, all such terms approach zero. In contrast, if the assignment does satisfy the formula, at least one free-energy term evaluates to $-\log(1 + \exp(c\epsilon))$, where $0 < \epsilon < 1$ as established in Theorem 1. Consequently, the more likely a truth assignment is to satisfy the formula, the lower the resulting free energy.

$$s_\varphi(\mathbf{x}) = -\frac{1}{c\epsilon} \min_{\mathbf{h}} E(\mathbf{x}, \mathbf{h}) = \lim_{c \to \infty} -\frac{1}{c\epsilon} \mathcal{F}(\mathbf{x}) \tag{7}$$

3 Results

3.1 Solve SAT Problems

This set of experiments evaluates the capability of Logical Boltzmann Machines (LBMs) to perform logical reasoning and solve SAT problems efficiently. For reasoning tasks, LBMs are used to find satisfying assignments of propositional logic formulae with exponentially large search spaces. By leveraging Gibbs sampling and free-energy minimisation, LBMs achieve 100% accuracy while exploring less than 0.75% of possible assignments, demonstrating high sample efficiency and scalability. In the SAT solving task, LBMs encode CNF formulae as energy functions and convert the discrete problem into a continuous optimisation task. Using methods like dual annealing and differential evolution, LBMs can solve SAT instances with up to 200 variables without any training data. Although not yet competitive with symbolic SAT solvers in efficiency, LBMs offer a differentiable, data-free approach to symbolic reasoning, with strong potential for integration into hybrid neuro-symbolic systems.

3.2 Learning from Data and Knowledge

This experiment evaluates the effectiveness of Logical Boltzmann Machines (LBMs) in combining symbolic background knowledge with labeled data for supervised learning. Using seven standard benchmarks from the ILP and neurosymbolic AI domains—including Mutagenesis, KRK, UW-CSE, and Alzheimer's drug-related datasets—the study compares LBMs against three baselines: the symbolic system Aleph, the neurosymbolic CILP++, and a standard RBM. LBMs are initialised from logical clauses and trained discriminatively, allowing them to revise and adapt the encoded knowledge during learning. Results show that LBMs outperform the baselines in five of the seven datasets, achieving high accuracy in both symbolic-rich and data-driven tasks. These outcomes demonstrate the LBM's capacity to effectively integrate structured knowledge with data to enhance predictive performance and generalisation.

3.3 Integration of Learning and Reasoning

The experiment demonstrates how Logical Boltzmann Machines (LBMs) can serve as a logical reasoning layer on top of deep neural networks for tasks such as semantic image interpretation. In this setup, LBMs are used to encode logical rules—such as part-whole relationships between objects—and are integrated with neural models that extract visual features and make predictions (e.g., using Faster R-CNN and neural regressors). The system is trained end-to-end, allowing the LBM to guide learning through logical inference, such as Modus Ponens and Modus Tollens, by enforcing consistency between predictions and background knowledge. Experimental results show that the LBM-based model outperforms other neurosymbolic systems, including Deep Logic Nets (DLN)(Tran and Garcez 2018), Logic Tensor Networks (LTN) (Donadello, Serafini, and d'Avila Garcez 2017; Badreddine et al. 2022), and Compositional Neural Logic Programming (CNLP)(Tran 2021), especially in predicting part-of relationships. This highlights the effectiveness of combining symbolic reasoning with sub-symbolic perception in a unified learning framework

References

Badreddine, S.; d'Avila Garcez, A.; Serafini, L.; and Spranger, M. 2022. Logic tensor networks. *Artificial Intelligence* 303:103649.

Donadello, I.; Serafini, L.; and d'Avila Garcez, A. S. 2017. Logic tensor networks for semantic image interpretation. In *IJCAI-17*, 1596–1602.

Tran, S., and Garcez, A. 2018. Deep logic networks: Inserting and extracting knowledge from deep belief networks. *IEEE T. Neur. Net. Learning Syst.* (29):246–258.

Tran, S. N. 2021. Compositional neural logic programming. In Zhou, Z.-H., ed., *Proceedings of IJCAI-21*, 3059–3066.