Differentiable Logics for Machine Learning with Logical Constraints in Practice

Thomas Flinkow

Department of Computer Science
Maynooth University

ITU Copenhagen
13th August 2024

Contents

Introduction & Motivation

Comparing Differentiable Logics for Learning with Logical Constraints*

Thomas Flinkow[a,*], Barak A. Pearlmutter[a,b], Rosemary Monahan[a,b]

[a]*Department of Computer Science, Maynooth University, Maynooth, Co. Kildare, Ireland*
[b]*Hamilton Institute, Maynooth University, Maynooth, Co. Kildare, Ireland*

Abstract

Extensive research on formal verification of machine learning systems indicates that learning from data alone often fails to capture underlying background knowledge such as specifications implicitly available in the data. Various neural network verifiers have been developed to ensure that a machine-learnt model satisfies correctness and safety properties, however, they typically assume a trained network with fixed weights. A promising approach for creating machine learning models that inherently satisfy constraints after training is to encode background knowledge as explicit logical constraints that guide the learning process via so-called differentiable logics. In this paper, we experimentally compare and evaluate various logics from the literature, presenting our findings and highlighting open problems for future work.

Keywords: machine learning, neuro-symbolic, differentiable logic, verification

1. Introduction

Advancements in machine learning (ML) in the past few years indicate great potential for applying ML to various domains. Autonomous systems are one such application domain, but using ML components in such a safety-critical domain presents unique new challenges for formal verification. These include

- ML failing to learn background knowledge from data alone [2],
- neural networks being susceptible to adversarial inputs [3, 4],
- and a lack of specifications, generally and especially when continuous learning is permitted [5–7].

Addressing these challenges is even more important and more difficult when the ML-enabled autonomous system is permitted to continue to learn after deployment, either to adapt to changing environments or to correct and improve itself when errors are detected [8].

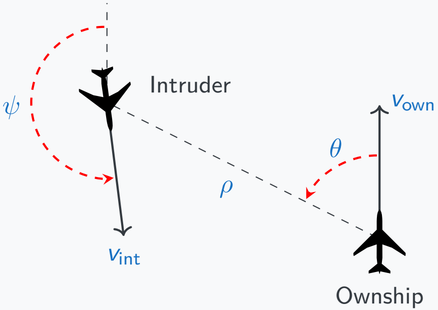
1.1. Formal verification of neural networks

A multitude of neural network verifiers have been presented in the past few years. We refer the reader to the Neural Network Verification Competition (VNN-COMP) reports [9–12] for an overview of state-of-the-art

Motivation

Issue:
Neural networks fail to learn (safety) properties from data alone!

Example: Reluplex (Katz et al., 2017)
'If an intruder is *near* and *approaching from the left*, network should advise *strong right*'.

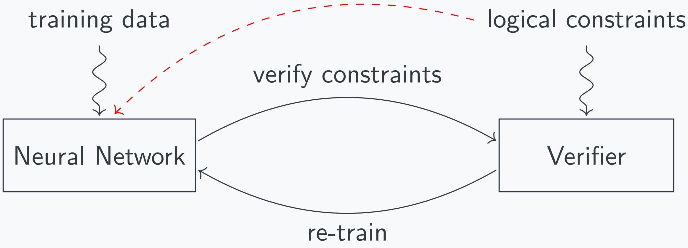


Training with Logical Constraints

Task: train a neural network $\mathcal{N}$ to satisfy constraint ϕ.

Train: given data, labels, and loss function, iteratively update network weights.

Verify: afterwards, α, β-CROWN, Marabou, NNV, ERAN, ...



Note

Training with constraints does not guarantee their satisfaction!

- **Is training with logical constraints useful in practice?**

- **Which logic translation is best?**

Background

Training with Differentiable Logics

Given data $\boldsymbol{x}_0$ and label $\boldsymbol{y}$, and constraint ϕ,
obtain optimal network weights $\boldsymbol{\theta}^+$ by

$$\boldsymbol{\theta}^+ = \arg\min_{\boldsymbol{\theta}} \ \alpha \mathcal{L}_{\mathsf{CE}}(\boldsymbol{x}_0, \boldsymbol{y}) + \beta \mathcal{L}_{\mathsf{C}}(\boldsymbol{x}_0, \boldsymbol{y}, \phi).$$

Insight from DL2 (Fischer et al., 2019)

Learning to satisfy $\forall x.\, x \vDash \phi$ by finding x^* such that $x^* \nvDash \phi$.

1. Approximate counterexample *outside* of training set using PGD:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \|\boldsymbol{x}-\boldsymbol{x}_0\|_\infty \leq \epsilon} \ \mathcal{L}_{\mathsf{C}}(\boldsymbol{x}_0, \boldsymbol{x}, \boldsymbol{y}, \phi)$$

2. Use this counterexample in training:

$$\boldsymbol{\theta}^+ = \arg\min_{\boldsymbol{\theta}} \ \alpha \mathcal{L}_{\mathsf{CE}}(\boldsymbol{x}_0, \boldsymbol{y}) + \beta \mathcal{L}_{\mathsf{C}}(\boldsymbol{x}_0, \boldsymbol{x}^*, \boldsymbol{y}, \phi).$$

DL2 (Fischer et al., 2019)

- mapping $[\![\cdot]\!]_{\mathsf{DL2}} : \Phi \to [0, \infty)$,
- $[\![\phi]\!]_{\mathsf{DL2}} = 0$ iff ϕ is satisfied,
- $[\![\phi]\!]_{\mathsf{DL2}}$ is differentiable almost everywhere.

Recursive definition of loss translation:

$$[\![x \leq y]\!]_{\mathsf{DL2}} := \max\{x - y, 0\}$$
$$[\![\phi \wedge \psi]\!]_{\mathsf{DL2}} := [\![\phi]\!]_{\mathsf{DL2}} + [\![\psi]\!]_{\mathsf{DL2}}$$
$$[\![\phi \vee \psi]\!]_{\mathsf{DL2}} := [\![\phi]\!]_{\mathsf{DL2}} \cdot [\![\psi]\!]_{\mathsf{DL2}}.$$

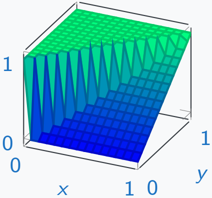
Fuzzy Logics (Ślusarz et al., 2023; van Krieken et al., 2022)

- logical system for reasoning with vagueness
- mapping $[\![\cdot]\!]_L : \Phi \to [0,1]$, where $[\![\top]\!]_L = 1$ and $[\![\bot]\!]_L = 0$,
- operators happen to be differentiable almost everywhere

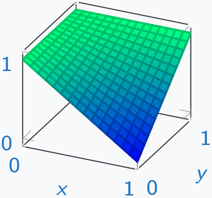
Logic	T-norm	S-norm	Implication
Gödel	$\min\{x, y\}$	$\max\{x, y\}$	$\begin{cases} 1, & \text{if } x < y, \\ y, & \text{else.} \end{cases}$
Kleene-Dienes			$S(N(x), y)$
Łukasiewicz	$\max\{0, x + y - 1\}$	$\min\{1, x + y\}$	
Reichenbach	xy	$x + y - xy$	
Goguen			$\begin{cases} 1, & \text{if } x < y, \\ y^x, & \text{else.} \end{cases}$

How do these logics differ?

$$[\![x \rightarrow y]\!]_G = \begin{cases} 1, & \text{if } x \leq y \\ y, & \text{else} \end{cases}$$



$$[\![x \rightarrow y]\!]_{RC} = 1 - x + xy$$



Example: 'If it rains, the ground will be wet.'

Let $[\![\text{rain}]\!] = 0.1$ and $[\![\text{ground wet}]\!] = 0$.

- With $\nabla[\![x \rightarrow y]\!]_G = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, we have no choice but to *gaslight*.

Findings

Only the Reichenbach implication closely follows MP and MT.

Shadow-Lifting (Varnai & Dimarogonas, 2020)

Definition

$$\left.\frac{\partial [\![x_1 \wedge x_2]\!]_L}{\partial x_i}\right|_{x_1=x_2=\rho} > 0 \quad \text{for all } i \in \{1,2\}.$$

Highly desirable for learning: allows for gradual improvement.

Example

The formula $0.1 \wedge 0.9$ should be more true than $0.1 \wedge 0.2$, but the Gödel t-norm $\min\{x, y\}$ yields the same truth value in both cases.

Findings

DL2 and the Reichenbach logic are the only shadow-lifting logics.

Consistency (Grespan et al., 2021)

Definition

Given a fuzzy logic tautology τ, its consistency is defined as

$$\int \cdots \int_{[0,1]} [\![\tau(x_1, \ldots, x_n)]\!]_L \, \mathrm{d}x_n \cdots \mathrm{d}x_1.$$

Findings

For the set of axioms chosen, Gödel logic was the least, and the sigm. Reichenbach and Łukasiewicz logics were the most consistent.

Experimental Setup & Results

Integration into PyTorch[1]

```
def train(..):
   for _, (inputs, labels) in enumerate(train_loader):
      outputs = NN(inputs)
      ce_loss = F.cross_entropy(outputs, labels)

      adv = pgd.attack(NN, inputs, labels, constraint)
      dl_loss = constraint.eval(NN, inputs, adv, labels)

      loss = alpha * ce_loss + beta * dl_loss

      optimizer.zero_grad()
      loss.backward()
      optimizer.step()
```

[1] https://github.com/tflinkow/comparing-differentiable-logics

Balancing Loss

$$\boldsymbol{\theta}^{+} = \arg\min_{\boldsymbol{\theta}} \; \alpha \mathcal{L}_{\mathsf{CE}}(\boldsymbol{x}_0, \boldsymbol{y}) + \beta \mathcal{L}_{\mathsf{C}}(\boldsymbol{x}_0, \boldsymbol{x}^{*}, \boldsymbol{y}, \phi).$$

Problem

It is *crucial* to find close to optimal values for α and β to allow each logic to perform at its best and to yield a fair comparison.

Adaptive Loss Balancing with GradNorm (Chen et al., 2018)

- Key point: $\alpha(t)$ and $\beta(t)$
- Better results than expensive grid search!

Local Robustness Constraint

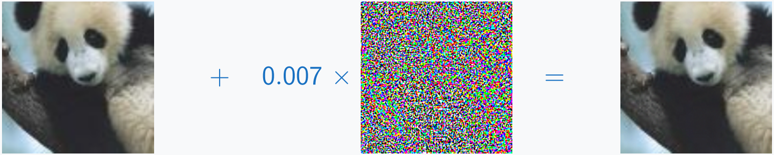


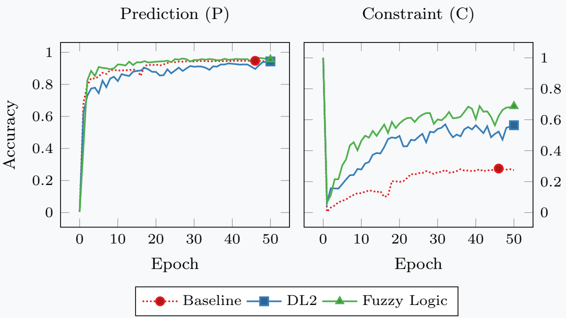
Figure 2: Adversarial attack (Goodfellow et al., 2015).

Definition

A neural network is locally robust in input $\boldsymbol{x}_0$, if

$$\underbrace{\forall \boldsymbol{x}.\, \|\boldsymbol{x} - \boldsymbol{x}_0\|_\infty \leq \varepsilon}_{\text{all elements in the input space close to } \boldsymbol{x}_0} \quad \text{implies} \quad \underbrace{\|\mathcal{N}(\boldsymbol{x}) - \mathcal{N}(\boldsymbol{x}_0)\|_\infty \leq \delta}_{\text{the classification is roughly the same}}$$

Local Robustness Constraint – Results



Logic	P	C
Baseline	94.68	28.45
DL2	94.32	56.46
Fuzzy Logic	**95.50**	**68.76**

Figure 3: The Robustness($\epsilon = 0.4, \delta = 0.01$) constraint on GTSRB.

Observation

The fuzzy logic translation $[\![x \leq y]\!]_L = \dfrac{1 - \max\{x - y, 0\}}{|x| + |y|}$ seems to perform better than the DL2 one $[\![x \leq y]\!]_{\mathrm{DL2}} = \max\{x - y, 0\}$.

Group Constraint

(a) unique signs

(b) danger signs

(c) derestriction signs

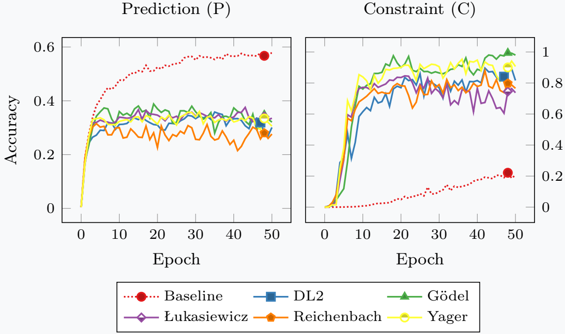
(d) speed limit signs

(e) other prohibitory signs

(f) mandatory signs

Definition

$$\underbrace{\forall \boldsymbol{x} \in ||\boldsymbol{x} - \boldsymbol{x}_0|| \le \epsilon}_{\text{handled by PGD}} \quad \rightarrow \quad \bigwedge_{G \in \mathcal{G}} p_G \le \delta \;\vee\; p_G \ge 1 - \delta.$$



Logic	P	C
Baseline	56.73	22.15
DL2	31.92	84.12
Gödel	**34.76**	**99.39**
Łukasiewicz	34.13	74.54
Reichenbach	28.00	79.76
Yager	33.44	90.06

Figure 5: The Groups($\epsilon = 0.6, \delta = 0.02$) constraint on GTSRB.

Observation

The shadow-lifting conjunctions (Reichenbach and DL2) do not perform as well as the Gödel one (which always has strong derivatives).

Class Similarity Constraint

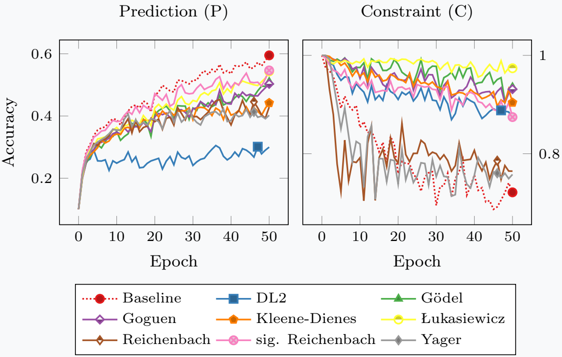
Introduce background knowledge into the network on CIFAR-10, i.e.

- A cat is more similar to a dog than to a frog.

Definition

$$\underbrace{\forall \boldsymbol{x} \in ||\boldsymbol{x} - \boldsymbol{x}_0|| \leq \epsilon}_{\text{handled by PGD}} \quad \rightarrow \quad \bigwedge_{\langle a,b,c \rangle \in \mathcal{T}} \left(\mathcal{N}(\boldsymbol{x})_a \geq {}^{1}\!/_{10} \rightarrow \mathcal{N}(\boldsymbol{x})_b \geq \mathcal{N}(\boldsymbol{x})_c \right).$$

Class Similarity Constraint – Results



Logic	P	C
Baseline	59.50	72.15
DL2	30.13	88.82
Gödel	50.93	93.51
Goguen	50.34	93.13
Kleene-Dienes	44.20	90.40
Łukasiewicz	**54.34**	**97.38**
Reichenbach	44.58	78.58
sig. Reichenbach	54.66	87.45
Yager	41.33	76.04

Figure 6: The `ClassSimilarity`$(\epsilon = 0.6)$ constraint on CIFAR-10.

Observation

DL2 introduces a significant hit to prediction accuracy. The only implication following MP and MT closely (Reichenbach) does not perform extraordinarily well.

Future Work

Expressive Power

Example: ROAD-R Data Set (Giunchiglia et al., 2023)

Videos annotated with background knowledge (propositional logic).

$$\{\neg\text{Ped}, \neg\text{Cyc}\} \cup \{\neg\text{Red}, \neg\text{Green}\} \cup \{\neg\text{Green}, \neg\text{Mov}\} \cup \ldots$$

Is there a need for more expressive logics? e.g.

- temporal,
- probabilistic

Summary

Problem
Investigate the effectiveness of various differentiable logics in practice.

Result
Training with *any* loss translation works well.

Future Work
Investigate logics for properties beyond propositional logic.

Thomas Flinkow

Department of Computer Science
Maynooth University

Email: thomas.flinkow@mu.ie

Thank you! Any questions?

References

Chen, Z., Badrinarayanan, V., Lee, C.-Y., & Rabinovich, A. (2018).GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. *Proceedings of the 35th International Conference on Machine Learning*, 794–803.

Fischer, M., Balunovic, M., Drachsler-Cohen, D., Gehr, T., Zhang, C., & Vechev, M. (2019).DL2: Training and Querying Neural Networks with Logic. *Proceedings of the 36th International Conference on Machine Learning*, 1931–1941. Retrieved April 13, 2023, from https://proceedings.mlr.press/v97/fischer19a.html

Giunchiglia, E., Stoian, M. C., Khan, S., Cuzzolin, F., & Lukasiewicz, T. (2023).ROAD-R: The autonomous driving dataset with logical requirements. *Machine Learning*, *112*(9), 3261–3291. https://doi.org/10.1007/s10994-023-06322-z

References (cont.)

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015, March). Explaining and Harnessing Adversarial Examples. https://doi.org/10.48550/arXiv.1412.6572

Grespan, M. M., Gupta, A., & Srikumar, V. (2021, July). Evaluating Relaxations of Logic for Neural Networks: A Comprehensive Study. https://doi.org/10.48550/arXiv.2107.13646

Katz, G., Barrett, C., Dill, D. L., Julian, K., & Kochenderfer, M. J. (2017). Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In R. Majumdar & V. Kunčak (Eds.), *Computer Aided Verification* (pp. 97–117). Springer International Publishing. https://doi.org/10.1007/978-3-319-63387-9_5

Ślusarz, N., Komendantskaya, E., Daggitt, M., Stewart, R., & Stark, K. (2023).Logic of Differentiable Logics: Towards a Uniform Semantics of DL. *EPiC Series in Computing*, *94*, 473–493. https://doi.org/10.29007/c1nt

References (cont.)

van Krieken, E., Acar, E., & van Harmelen, F. (2022).Analyzing Differentiable Fuzzy Logic Operators. *Artificial Intelligence*, *302*, 103602. https://doi.org/10.1016/j.artint.2021.103602

Varnai, P., & Dimarogonas, D. V. (2020).On Robustness Metrics for Learning STL Tasks. *2020 American Control Conference (ACC)*, 5394–5399. https://doi.org/10.23919/ACC45564.2020.9147692