Comparing Differentiable Logics for Learning Systems

A Research Preview

Thomas Flinkow   Barak A. Pearlmutter   Rosemary Monahan

Department of Computer Science
Maynooth University

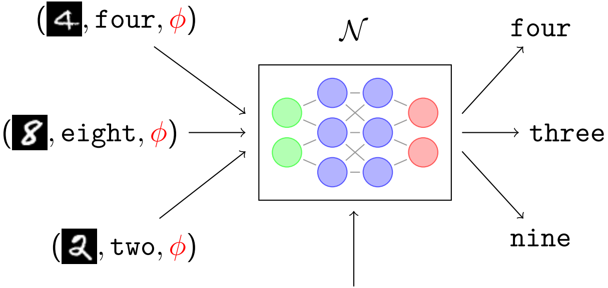
FMAS 2023

Contact: thomas.flinkow@mu.ie

Verification of Neural Networks

Task: verify that neural network $\mathcal{N}$ satisfies constraint ϕ.

1. **Train:** given data $\boldsymbol{x} = [$, , $]$, labels $\boldsymbol{y} = [\texttt{four}, \texttt{eight}, \texttt{two}]$ and loss function $\mathcal{L}$, iteratively update network weights.



$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{y}, \mathcal{N}(\boldsymbol{x})) + \lambda \mathcal{L}_{\phi}(\boldsymbol{y}, \mathcal{N}(\boldsymbol{x}))$$

2. **Verify:** with α, β-CROWN, Marabou, NNV, ERAN, ..., etc.[1]

[1] M.N. Müller *et al.*: The Third International Verification of Neural Networks Competition (VNN-COMP 2022): Summary and Results (2023) https://doi.org/10.48550/arXiv.2212.10376.

DL2[2]

- $\mathcal{L}$ maps logical constraints into $[0, \infty)$,
- $\mathcal{L}(\phi) = 0$ iff ϕ is satisfied,
- $\mathcal{L}(\phi)$ is differentiable almost everywhere.

Recursive definition of loss translation:

$$\begin{aligned}
\mathcal{L}_{\mathsf{DL2}}(x \leq y) &:= \max(x - y, 0) \\
\mathcal{L}_{\mathsf{DL2}}(x \wedge y) &:= \mathcal{L}_{\mathsf{DL2}}(x) + \mathcal{L}_{\mathsf{DL2}}(y) \\
\mathcal{L}_{\mathsf{DL2}}(x \vee y) &:= \mathcal{L}_{\mathsf{DL2}}(x) \cdot \mathcal{L}_{\mathsf{DL2}}(y).
\end{aligned}$$

[2] M. Fischer *et al.*: DL2: Training and Querying Neural Networks with Logic (2019).

Fuzzy Logics[3][4]

- logical system for reasoning with vagueness
- satisfaction of logical formulas expressed on $[0, 1]$
- operators are often differentiable almost everywhere

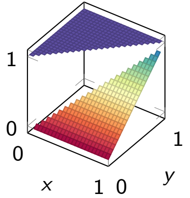
Name	**T-norm (Conjunction)**	**T-conorm (Disjunction)**
Gödel	$T_{\mathrm{G}}(x, y) = \min(x, y)$	$S_{\mathrm{G}}(x, y) = \max(x, y)$
Łukasiewicz	$T_{\mathrm{ŁK}}(x, y) = \max(0, x + y - 1)$	$S_{\mathrm{ŁK}}(x, y) = \min(1, x + y)$
Product	$T_{\mathrm{P}}(x, y) = xy$	$S_{\mathrm{PS}}(x, y) = x + y - xy$

[3] N. Ślusarz *et al.*: Logic of Differentiable Logics: Towards a Uniform Semantics of DL (2023) https://doi.org/10.48550/arXiv.2303.10650.

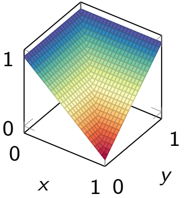
[4] E. van Krieken *et al.*: Analyzing Differentiable Fuzzy Logic Operators (2022) https://doi.org/10.1016/j.artint.2021.103602.

Fuzzy Implications[5]

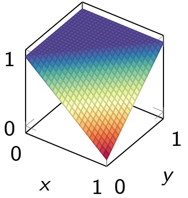
$$I_{\mathrm{G}}(x,y) = \begin{cases} 1, & \text{if } x < y \\ y \end{cases}$$



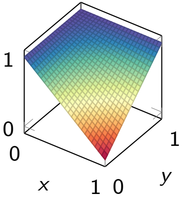
$$I_{\mathrm{KD}}(x,y) = \max(1-x, y)$$



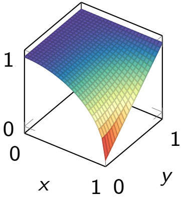
$$I_{\mathrm{ŁK}}(x,y) = \min(1-x+y, 1)$$



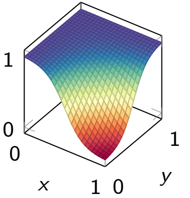
$$I_{\mathrm{RC}}(x,y) = 1 - x + xy$$



$$(I_{\mathrm{RC}})_{\phi=x^2}$$



$$(I_{\mathrm{RC}})_{s=9}$$



[5] M. Baczyński and B. Jayaram: Fuzzy Implications (2008).

Experimental Evaluation[6]

- Fashion-MNIST & CIFAR-10:
  $(\mathcal{N}(\boldsymbol{x})_{\texttt{dog}} \geq {}^{1}/_{10}) \longrightarrow (\mathcal{N}(\boldsymbol{x})_{\texttt{cat}} \geq \mathcal{N}(\boldsymbol{x})_{\texttt{truck}})$

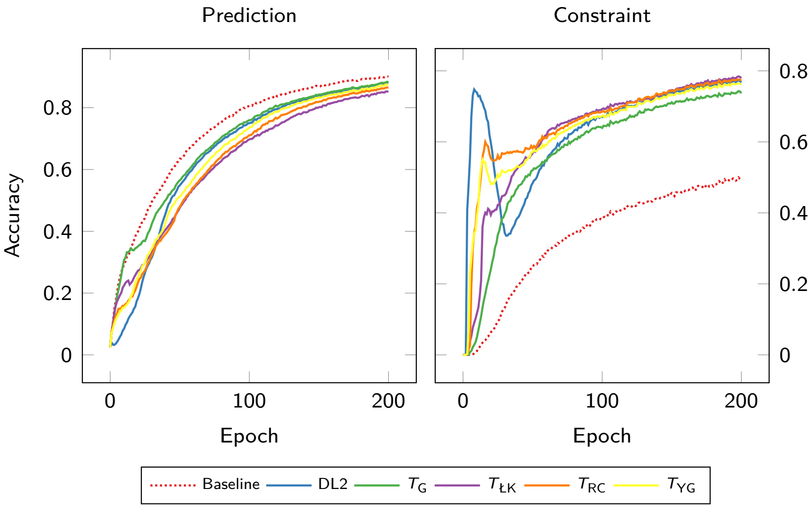
$$\mathcal{N}([\;\;]) = \begin{bmatrix} \texttt{dog:} & 0.15 \\ \texttt{cat:} & 0.12 \\ & \vdots \\ \texttt{plane:} & 0.02 \\ \texttt{truck:} & 0.08 \end{bmatrix}$$

- GTSRB: $\left(\sum_{k \in \text{limits}} \mathcal{N}(\boldsymbol{x})_k \leq \epsilon\right) \vee \left(\sum_{k \in \text{limits}} \mathcal{N}(\boldsymbol{x})_k \geq 1 - \epsilon\right)$

$$\mathcal{N}([\;\;]) = \begin{bmatrix} \texttt{limit 30 km/h:} & 0.12 \\ \texttt{limit 50 km/h:} & 0.31 \\ \texttt{limit 70 km/h:} & 0.11 \\ & \vdots \\ \texttt{no way:} & 0.02 \end{bmatrix} \quad \sum_{\text{limits}} = 0.54$$

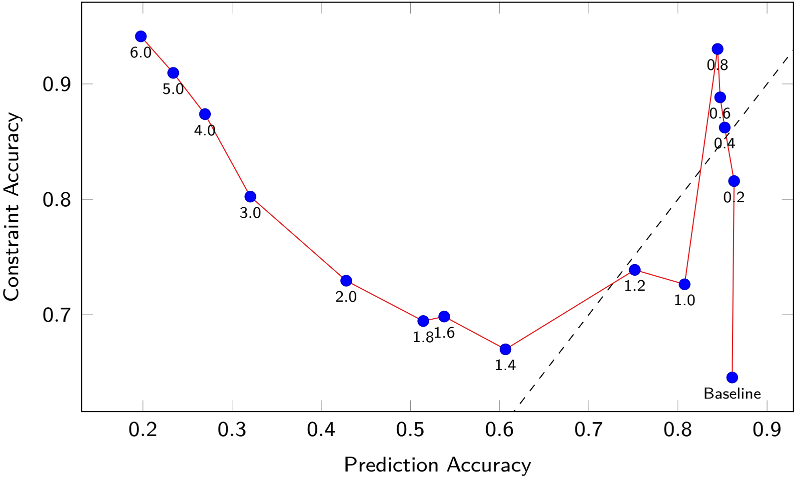
[6] https://github.com/tflinkow/dl-comparison

Results



Main finding: training with *any* logical loss improves constraint accuracy. Which loss translation is best? 😰

Hyperparameter in $\mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\phi}$



Effectiveness of training with logical constraints depends heavily on λ in a non-obvious, non-monotonic way.

Future Work

RQ1 What properties of learning systems to specify and verify?

RQ2 Are existing differentiable logics expressive enough?

RQ3 How to reuse constraints from training during inference?