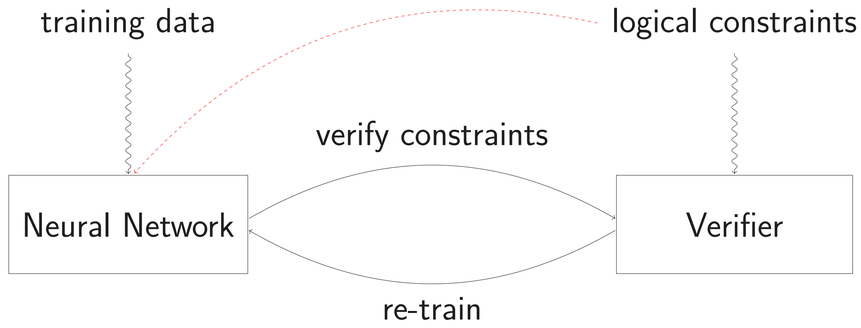
Differentiable Logics for Machine Learning: What difference do they make?

Thomas Flinkow[1], Barak A. Pearlmutter[1,2], Rosemary Monahan[1,2]

[1]Department of Computer Science, Maynooth University
[2]Hamilton Institute, Maynooth University
thomas.flinkow@mu.ie

1. Motivation: Correct-by-construction ML models

Task: train a neural network $\mathcal{N}$ to satisfy constraint ϕ.
Train: given data, labels, and loss function, iteratively update weights.
Verify: α, β-CROWN, Marabou, NNV, ERAN, ...



2. Background: Property-driven ML

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

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

DL2: approximate $\forall x.\, x \vDash \phi$ by finding x^* such that $x^* \nvDash \phi$.

- Approximate counterexample *outside* of training set using PGD:

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

- Use this counterexample in training:

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

3. Background: Differentiable Logics

- **DL2:** $[\![\cdot]\!]_{\text{DL2}} : \Phi \to [0, \infty)$, where $[\![\top]\!]_{\text{DL2}} = 0$, and

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

- **Fuzzy Logics:** $[\![\cdot]\!]_L : \Phi \to [0, 1]$, where $[\![\top]\!]_L = 1$ and $[\![\bot]\!]_L = 0$

Logic	Conjunction	Disjunction	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			
Goguen	xy	$x + y - xy$	$\begin{cases} 1, & \text{if } x < y, \\ y^x, & \text{else.} \end{cases}$

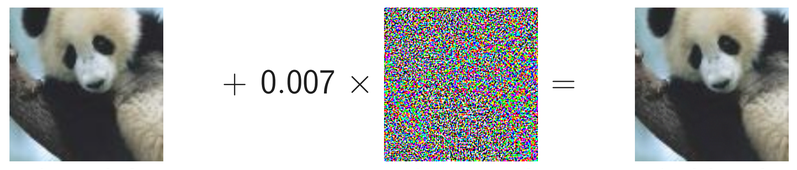
4. Methods

General-purpose framework implementation in PyTorch:

```
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()
```

(https://github.com/tflinkow/comparing-differentiable-logics)

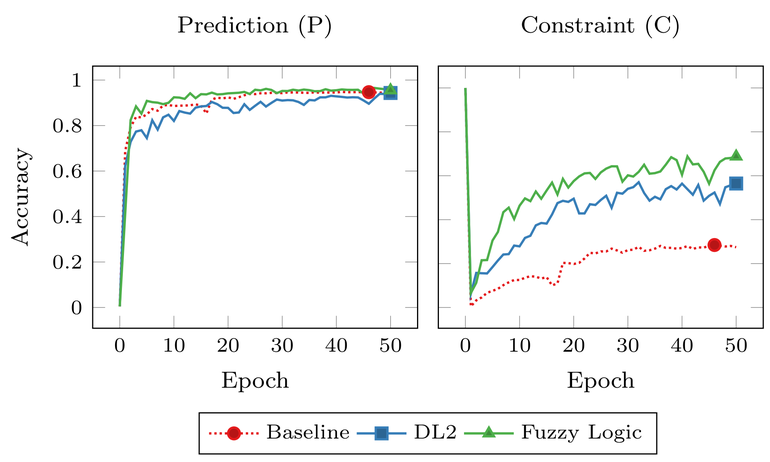
5. Experiment – Local Robustness Constraint



'panda' 57.7 % confidence — adversarial noise — 'gibbon' 99.3 % confidence

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

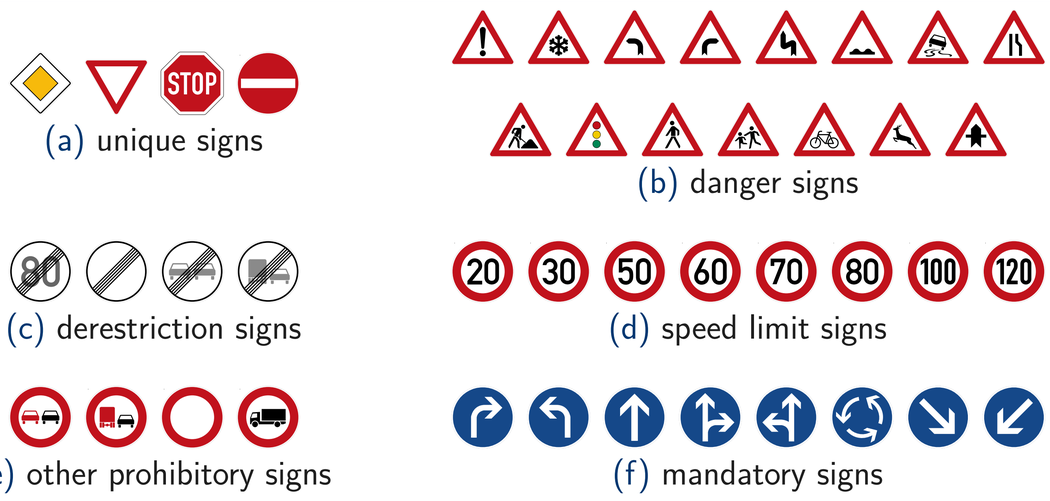
$$\underbrace{\forall \boldsymbol{x}.\, \|\boldsymbol{x} - \boldsymbol{x}_0\|_\infty \leq \varepsilon}_{\text{all elements in the input space close to } \boldsymbol{x}_0} \quad \text{implies} \quad \underbrace{\|\mathcal{N}(\boldsymbol{x}) - \mathcal{N}(\boldsymbol{x}_0)\|_\infty \leq \delta}_{\text{the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



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

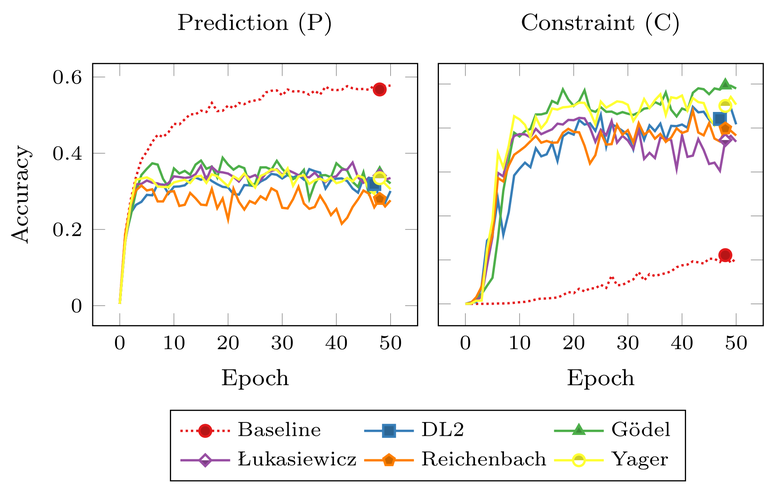
Figure: The $\texttt{Robustness}(\epsilon = 0.4, \delta = 0.01)$ constraint on GTSRB.

6. Experiment – Group Constraint



(a) unique signs
(b) danger signs
(c) derestriction signs
(d) speed limit signs
(e) other prohibitory signs
(f) mandatory signs

Constraint: The sum of probabilities of groups of related signs must be either very high or very low.



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: The $\texttt{Groups}(\epsilon = 0.6, \delta = 0.02)$ constraint on GTSRB.

7. Results

Main finding: Training with *any* differentiable logic works well! Operators with favourable theoretical properties (e.g. shadow-lifting conjunctions, implications with Modus Ponens and Modus Tollens reasoning) are not necessarily the best choice in practice.

8. Future Work

- Expressive specifications for ML & temporal differentiable logics.
- Certified training for guaranteed constraint satisfaction.

Thomas Flinkow, Barak A. Pearlmutter, Rosemary Monahan (2024). 'Comparing Differentiable Logics for Learning with Logical Constraints', *Submitted to Science of Computer Programming*. arXiv: https://arxiv.org/abs/2407.03847

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under grant number 20/FFP-P/8853.