DL2: Training and Querying Neural Networks with Logic

Marc Fischer, Mislav Balunović, Dana Drachsler-Cohen, Timon Gehr, Ce Zhang, Martin Vechev

Department of Computer Science
ETH zürich

Training & Querying with Constraints

DL2 is a system which allows to (i) **query** networks for inputs i, satisfying a constraint φ and (ii) **train** networks with weights θ to satisfy a constraint φ.

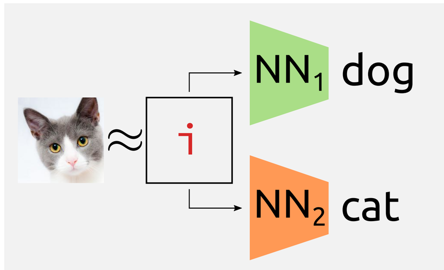
Key insight: Translate the constraint φ to a loss $\mathcal{L}(\varphi)$.

Querying Neural Networks with Constraints

Users can write queries in a simple SQL-like language.

Differencing neural networks as in [Pei et al., 2017]

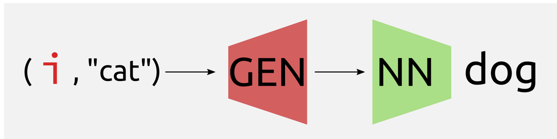
```
find i[32, 32, 3]
where i in [0, 1],
      class(NN1(i)) = dog,
      class(NN2(i)) = cat,
      ||i - image||2 < 2
```

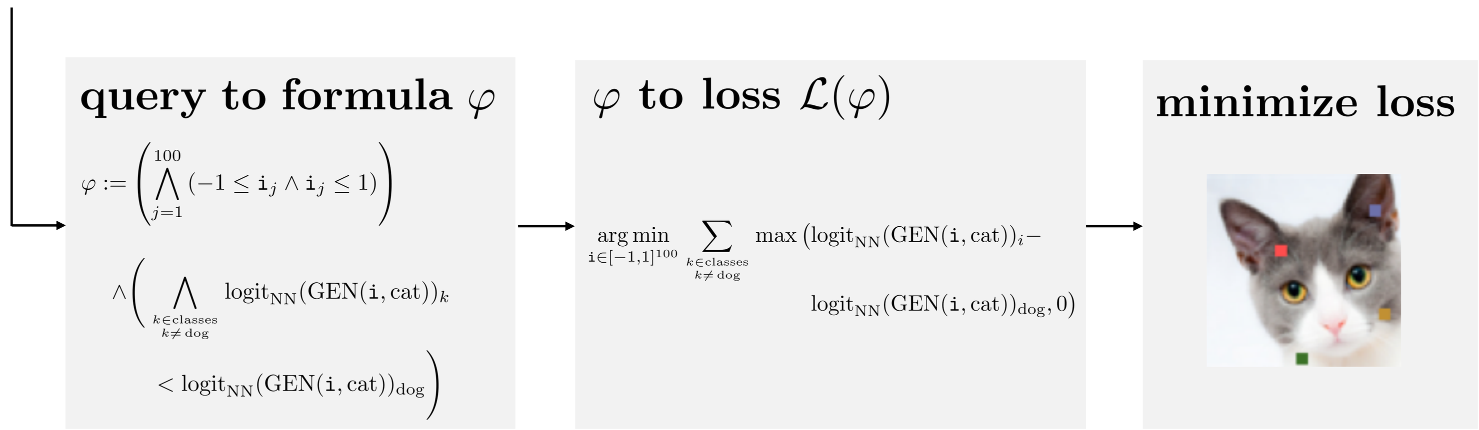


Read query as: *Find* i *of size* $32 \times 32 \times 3$ *which (i) has values in* $[0,1]$*, (ii) gets classified as* dog *by* NN1*, (iii) as* cat *by* NN2*, and (iv) is close to a given* image *(w.r.t.* L_2*-norm).*

Finding adversarial examples using a generator as in [Song et al., 2018]

```
find i[100]
where i in [-1, 1],
  class(NN(GEN(i, cat))) = dog
```

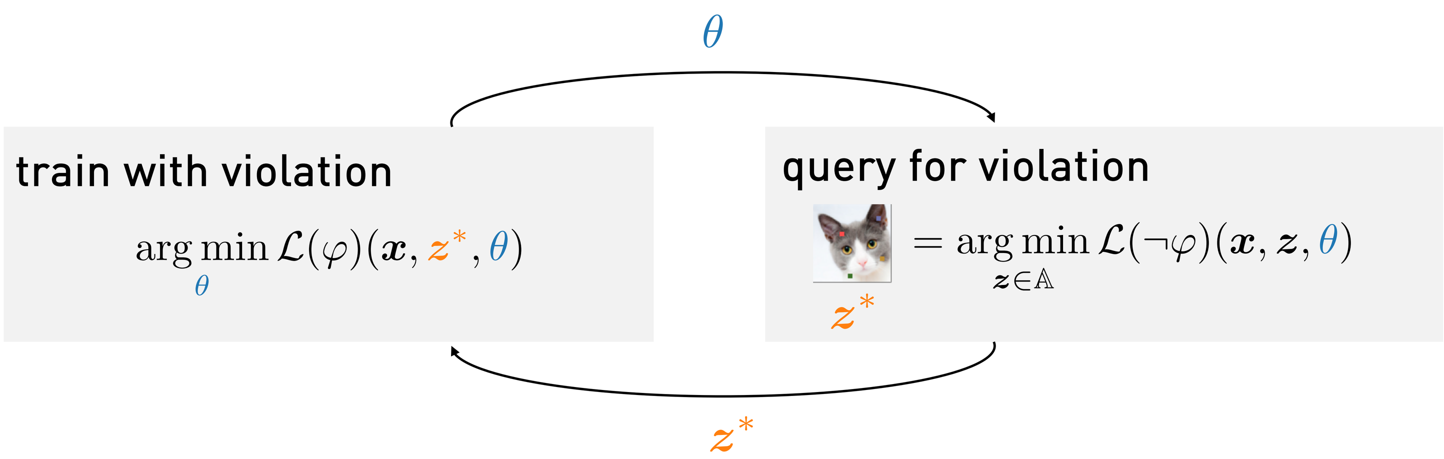




Training with Constraints (Supervised, Semi-Supervised, Unsupervised)

Goal: Train a neural network, with weights θ, on data $\mathcal{T}$, such that a constraint φ holds for all inputs from a set $\mathbb{A}$.

$$\arg\min_{\theta} \mathop{\mathbb{E}}_{x\sim\mathcal{T}} \left[\mathcal{L}(\varphi)(\boldsymbol{x}, \arg\min_{\boldsymbol{z}\in\mathbb{A}} \mathcal{L}(\neg\varphi)(\boldsymbol{x},\boldsymbol{z},\theta),\theta)\right]$$



- **generalizes** adversarial robustness training
- **generalizes** prior work on training with constraints
- **applicable** to supervised, semi-supervised and unsupervised training

Key Ingredient: Constraints to Loss

DL2 translates a constraint φ, given as a quantifier-free first order logical formula, to a differentiable loss $\mathcal{L}(\varphi)$ with desirable properties.

Key Properties of the Translation

- $\mathcal{L}(\varphi) \geq 0$
- $\mathcal{L}(\varphi) = 0$ **if and only if** φ is **satisfied**
- $\mathcal{L}(\varphi)$ is **differentiable** almost everywhere w.r.t. all variables and network parameters θ
- allowing the use of **standard optimizers** such as (projected) gradient descent or L-BFGS-B

Recursive Translation

$$\begin{aligned}
\mathcal{L}(t \leq t') &:= \max(t - t', 0)\\
\mathcal{L}(t \neq t') &:= \xi \cdot [t = t']\\
\mathcal{L}(t = t') &:= \mathcal{L}(t \leq t' \wedge t' \leq t)\\
\mathcal{L}(t < t') &:= \mathcal{L}(t \leq t' \wedge t \neq t')\\
\mathcal{L}(\varphi' \wedge \varphi'') &:= \mathcal{L}(\varphi') + \mathcal{L}(\varphi'')\\
\mathcal{L}(\varphi' \vee \varphi'') &:= \mathcal{L}(\varphi') \cdot \mathcal{L}(\varphi'')
\end{aligned}$$

Negation

In a negated formula $\neg\psi$, the negation is recursively pushed into subformulas until no negation remains, e.g.

$$\mathcal{L}(\neg((a > 3) \wedge (b < 1))) \rightarrow \mathcal{L}((a \leq 3) \vee (b \geq 1)).$$

Query Specific

$$(\texttt{class}(\texttt{NN}(\texttt{i})) = \texttt{c}) := \bigwedge_{\substack{k\in\text{classes}\\ k\neq c}} \texttt{logits}_{\texttt{NN}}(\texttt{i})_k < \texttt{logits}_{\texttt{NN}}(\texttt{i})_c$$

DL2 Deep Learning with Differentiable Logic — SRILAB

github.com/eth-sri/dl2

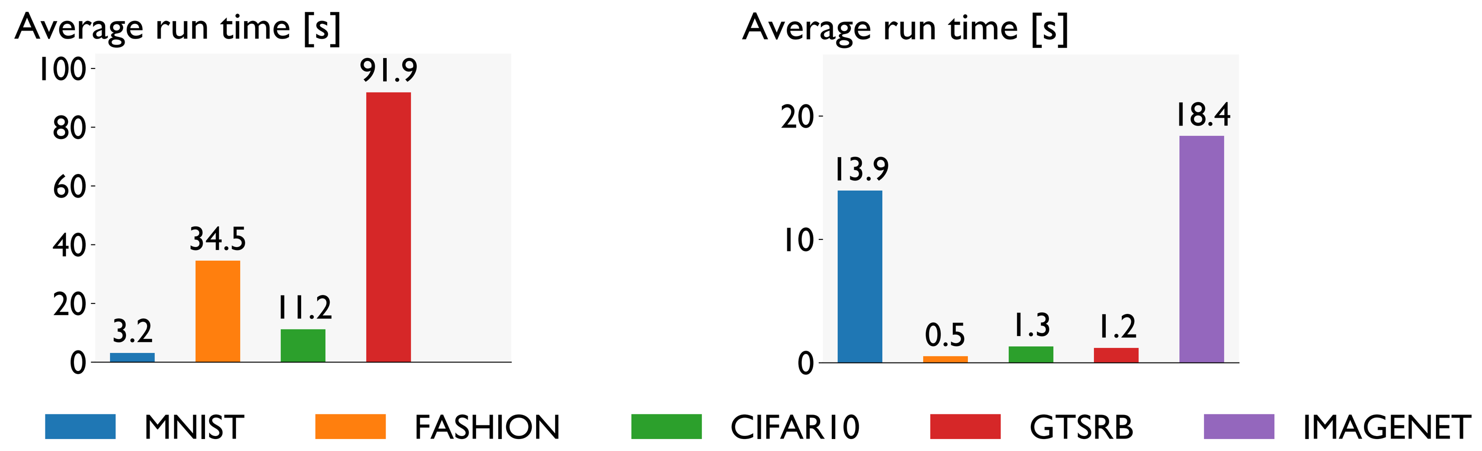
References

Pei, K., Cao, Y., Yang, J., and Jana, S. (2017) DeepXplore: Automated whitebox testing of deep learning systems.
Song, Y., Shu, R., Kushman, N., and Ermon, S. (2018) Generative adversarial examples.

Evaluating DL2

Querying

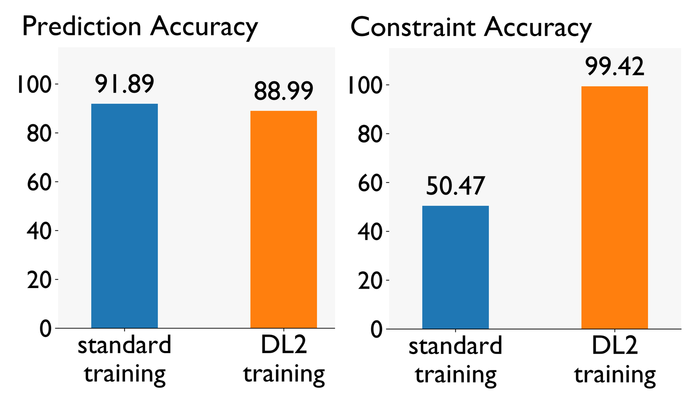
- 18 templated queries (similar to the queries in the left column)
- 2 minute timeout
- we report the average run time for 10 instantiations of each template (results for two queries are shown below)



Supervised Training

"A car should be considered more similar to a truck than a dog."

$$\forall \boldsymbol{z} \in B_\epsilon(\boldsymbol{x}) \cap [0,1]^d . y = \text{car} \implies \text{logit}_\theta(\boldsymbol{z})_{\text{truck}} > \text{logit}_\theta(\boldsymbol{z})_{\text{dog}} + \delta$$

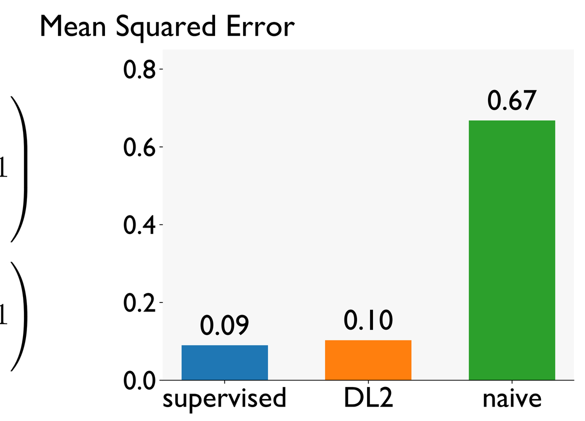


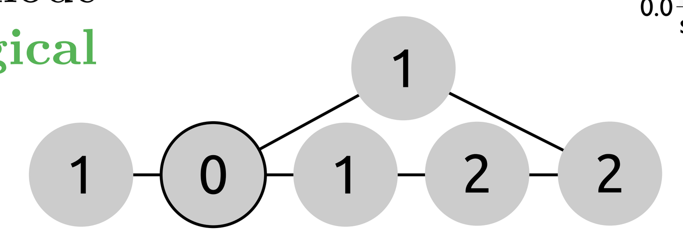
Trend over 7 queries on 3 datasets: **slight drop** of prediction accuracy, but **large gains in constraint accuracy**.

Unsupervised Training

Given an unweighted graph with 15 nodes and random edges, predict the length of the shortest path from a root node **only from a logical description**.

$$d(0) = 0 \wedge \bigwedge_{v\in V} d(v) \geq 0 \wedge \left(\bigvee_{\substack{(v,v')\in E\\ v'\neq 0}} d(v') = d(v) + 1\right) \wedge \left(\bigwedge_{(v,v')\in E} d(v') \leq d(v) + 1\right)$$

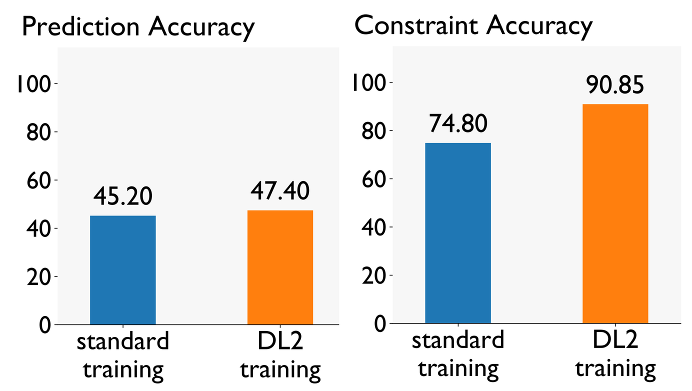




Semi-supervised Training

For CIFAR-100, train with:

- cross-entropy loss on 25% labeled data
- the DL2 loss matching the following constraint on the remaining unlabelled data



"Class groups have either high probability or low probability."

$$(p_{people} < \epsilon \vee p_{people} > 1 - \epsilon) \wedge \ldots \wedge (p_{trees} < \epsilon \vee p_{trees} > 1 - \epsilon)$$

beyond the reach of prior work