DL2: Training and Querying Neural Networks with Logic

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

github.com/eth-sri/dl2
differencing neural networks
[Pei et al., 2017]

finding adversarial examples
[Szegedy et al., 2013]

finding inputs that deactivates neurons

finding adversarial examples using a generator
[Song et al., 2018]
DL2: Deep Learning with Differentiable Logic
**differencing neural networks**  
[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
```

**finding adversarial examples**  
[Szegedy et al., 2013]

```
find i[224, 224, 3] where i in [0, 1],
    class(NN1(i)) = dog,
    \|i - image\|_\infty < 25
```

**finding inputs that deactivates neurons**

```
find i[32, 32, 3] where i in [0, 1],
    NN(i).l3[17] = 0,
    class(NN(i)) = cat,
    \|i - image\|_1 < 100
```

**finding adversarial examples using a generator**  
[Song et al., 2018]

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

\[ \text{find } i \in [0, 1], \text{ where } i \text{ in } [0, 1], \]
\[ \text{class}(\text{NN1}(i)) = \text{dog}, \]
\[ \text{class}(\text{NN2}(i)) = \text{cat}, \]
\[ ||i - \text{image}||_2 < 2 \]

\[ \sim \]

\begin{align*}
\text{NN}_1 & \quad \text{dog} \\
\text{NN}_2 & \quad \text{cat}
\end{align*}
differencing neural networks [Pei et al., 2017]

\[
\text{find } i[32, 32, 3] \\
\text{where } i \in [0, 1], \\
\text{class}(\text{NN1}(i)) = \text{dog}, \\
\text{class}(\text{NN2}(i)) = \text{cat}, \\
\|i - \text{image}\|_2 < 2
\]

finding adversarial examples [Szegedy et al., 2013]

\[
\text{find } i[224, 224, 3] \\
\text{where } i \in [0, 1], \\
\text{class}(\text{NN1}(i)) = \text{dog}, \\
\|i - \text{image}\|_\infty < 25
\]

finding inputs that deactivates neurons

\[
\text{find } i[32, 32, 3] \\
\text{where } i \in [0, 1], \\
\text{class}(\text{NN1}(i)) = \text{dog}, \\
\text{class}(\text{NN2}(i)) = \text{cat}, \\
\|i - \text{image}\|_2 < 2
\]

finding adversarial examples using a generator [Song et al., 2018]

\[
\text{find } i[100] \\
\text{where } i \in [-1, 1], \\
\text{class}(\text{NN}(\text{GEN}(i, \text{cat}))) = \text{dog} \\
\text{return } \text{GEN}(i, \text{cat})
\]
differencing neural networks [Pei et al., 2017]

\[
\text{find } i[32, 32, 3] \\
\text{where } i \text{ in } [0, 1], \\
\quad \text{class}(\text{NN1}(i)) = \text{dog}, \\
\quad \text{class}(\text{NN2}(i)) = \text{cat}, \\
\quad \lVert i - \text{image} \rVert_2 < 2, \\
\quad \text{NN1}(i).p[\text{dog}] > 0.8, \\
\quad \text{NN1}(i).p[\text{cat}] < 0.1
\]

finding adversarial examples [Szegedy et al., 2013]

\[
\text{find } i[224, 224, 3] \\
\text{where } i \text{ in } [0, 1], \\
\quad \text{class}(\text{NN1}(i)) = \text{dog}, \\
\quad \lVert i - \text{image} \rVert_\infty < 25, \\
\quad \lVert i - \text{image} \rVert_\infty > 5
\]

finding adversarial examples using a generator [Song et al., 2018]

\[
\text{find } i[100] \\
\text{where } i \text{ in } [-1, 1], \\
\quad \text{class}(\text{NN1}(\text{GEN}(i, \text{cat}))) = \text{dog}, \\
\quad \text{class}(\text{NN2}(\text{GEN}(i, \text{cat}))) = \text{car} \\
\text{return } \text{GEN}(i, \text{cat})
\]

finding inputs that deactivates neurons

\[
\text{find } i[32, 32, 3] \\
\text{where } i \text{ in } [0, 1], \\
\quad \text{NN}(i).l3[17] = 0, \\
\quad \text{class}(\text{NN}(i)) = \text{cat}, \\
\quad \lVert i - \text{image} \rVert_1 < 100, \\
\quad i[:8, :8, :] = \text{image}[:8, :8, :]
\]
**DL2 Querying**

**query**

find $i[100]$
where $i$ in $[-1, 1]$
class$(NN(\text{GEN}(i, \text{cat}))) = \text{dog}$
return $\text{GEN}(i, \text{cat})$

**logical formula $\varphi$**

$$\varphi := \left( \bigwedge_{j=1}^{100} (-1 \leq i_j \wedge i_j \leq 1) \right) \wedge \left( \bigwedge_{k \in \text{classes}} \min_{k \neq \text{dog}} \logit_{NN}(\text{GEN}(i, \text{cat}))_k < \logit_{NN}(\text{GEN}(i, \text{cat}))_{\text{dog}} \right)$$

**differentiable loss**

$$\mathcal{L}(\varphi) \geq 0$$

**minimize loss**

$$\arg \min_{i \in [-1, 1]^{100}} \sum_{k \neq \text{dog}} \max \left( \logit_{NN}(\text{GEN}(i, \text{cat}))_k, 0 \right)$$

**Theorem:** $\mathcal{L}(\varphi) = 0$ if and only if $\varphi$ is satisfied
DL2 Training

Goal: $\varphi$ holds for all inputs

\[ \arg\min_{\theta} \mathcal{L}(\varphi)(x, z^*, \theta) \]

\[ = \arg\min_{z \in A} \mathcal{L}(\neg \varphi)(x, z, \theta) \]

generalizes adversarial robustness training
generalizes previous work for training with constraints
applicable to supervised, semi-supervised and unsupervised training
A car should be considered more similar to a truck than a dog.

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

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

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

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

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

Prediction Accuracy

<table>
<thead>
<tr>
<th></th>
<th>standard training</th>
<th>DL2 training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>91.89</td>
<td>88.99</td>
</tr>
</tbody>
</table>

Constraint Accuracy

<table>
<thead>
<tr>
<th></th>
<th>standard training</th>
<th>DL2 training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>50.47</td>
<td>99.42</td>
</tr>
</tbody>
</table>

Resnet-18 on CIFAR-10
DL2: Training and Querying Neural Networks with Logic

Querying

find \( i \in [32, 32, 3] \)

where \( i \) in \([0, 1]\),

class(NN1(\( i \))) = dog,

class(NN2(\( i \))) = cat,

\( \|i - \text{image}\|_2 < 2 \),

NN1(\( i \)).p[7] > 0.8

NN1(\( i \)).p[1] < 0.1

Training

\[
\min_{\theta} \mathcal{L}(\varphi)(x, z^*, \theta)
\]

query for violation

\[
\min_{z \in A} \mathcal{L}(\neg \varphi)(x, z, \theta)
\]

\( z^* \)

generalizes adversarial robustness training

generalizes previous work for training with constraints

applicable to supervised, semi-supervised and unsupervised training

github.com/eth-sri/dl2