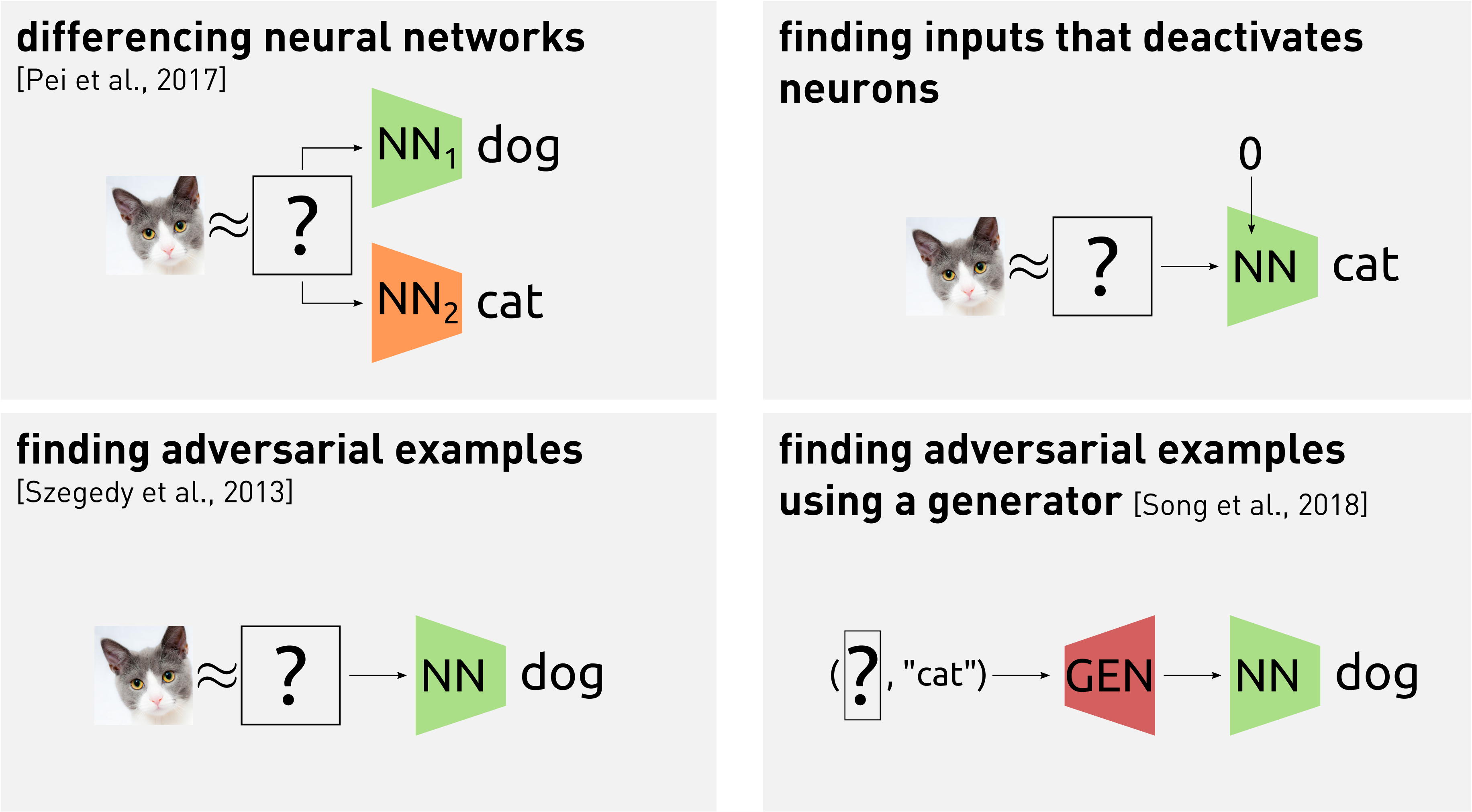
DL2: Training and Querying Neural Networks with Logic

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

SRILAB

github.com/eth-sri/dl2

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



differencing neural networks
[Pei et al., 2017]
?
NN1
dog
NN2
cat
finding inputs that deactivates neurons
0
?
NN
cat
finding adversarial examples
[Szegedy et al., 2013]
?
NN
dog
finding adversarial examples using a generator [Song et al., 2018]
(?, "cat")
GEN
NN
dog

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 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

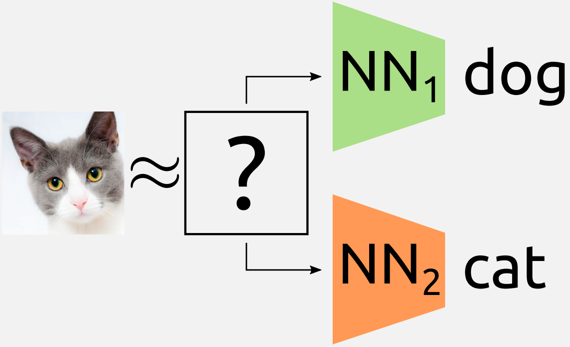
[Szegedy et al., 2013]

```
find i[224, 224, 3]
where i in [0, 1],
      class(NN1(i)) = dog,
     ‖i - image‖∞ < 25
```

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



DL2 Deep Learning with Differentiable Logic

```
find i[32, 32, 3]
where i in [0, 1],
      class(NN1(i)) = dog,
      class(NN2(i)) = cat,
      ‖i - image‖_2 < 2
```

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 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

[Szegedy et al., 2013]

```
find i[224, 224, 3]
where i in [0, 1],
      class(NN1(i)) = dog,
     ‖i - image‖∞ < 25
```

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

[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,
      NN1(i).p[dog] > 0.8,
      NN1(i).p[cat] < 0.1
```

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,
      i[:8, :8, :] = image[:8, :8, :]
```

finding adversarial examples

[Szegedy et al., 2013]

```
find i[224, 224, 3]
where i in [0, 1],
      class(NN1(i)) = dog,
      ‖i - image‖∞ < 25,
      ‖i - image‖∞ > 5
```

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

```
find i[100]
where i in [-1, 1],
   class(NN1(GEN(i, cat))) = dog,
   class(NN2(GEN(i, cat))) = car
return GEN(i, cat)
```

DL2 Querying

query

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

→

logical formula φ

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

→

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

$$\underset{\mathtt{i} \in [-1,1]^{100}}{\arg\min} \sum_{\substack{k \in \text{classes} \\ k \neq \text{dog}}} \max\left(\text{logit}_{\text{NN}}(\text{GEN}(\mathtt{i}, \text{cat}))_i - \text{logit}_{\text{NN}}(\text{GEN}(\mathtt{i}, \text{cat}))_{\text{dog}}, 0\right)$$

→

minimize loss

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

DL2 Training

Goal: φ holds for all inputs

θ

train with violation

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

query for violation

$$= \arg\min_{\boldsymbol{z}\in\mathbb{A}} \mathcal{L}(\neg\varphi)(\boldsymbol{x}, \boldsymbol{z}, \theta)$$

$\boldsymbol{z}^*$

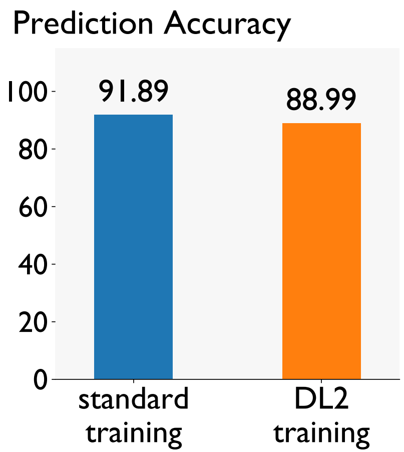
$\boldsymbol{z}^*$

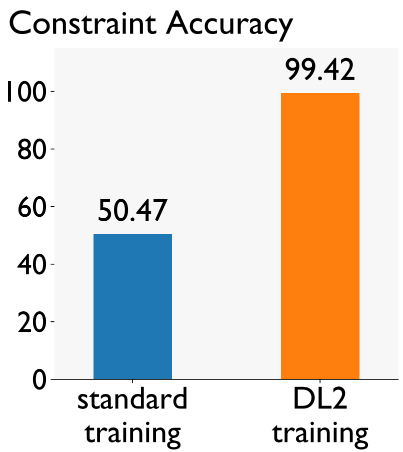
generalizes adversarial robustness training
generalizes previous work for training with constraints
applicable to supervised, semi-supervised and unsupervised training

Supervised Training Example

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

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



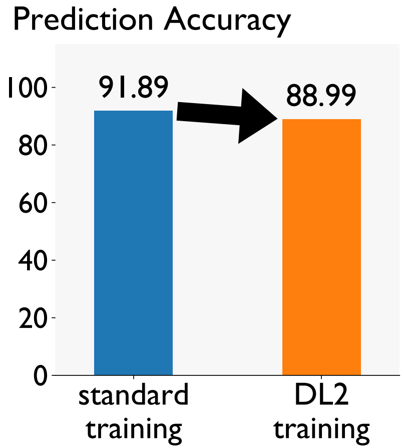


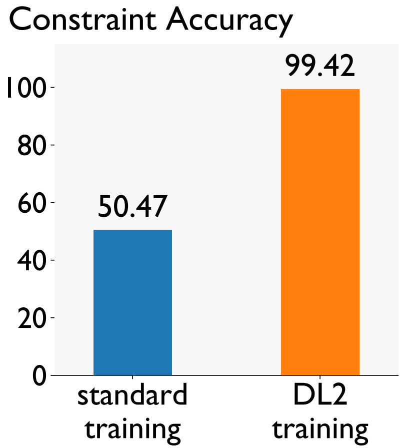
Resnet-18 on CIFAR-10

Supervised Training Example

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

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



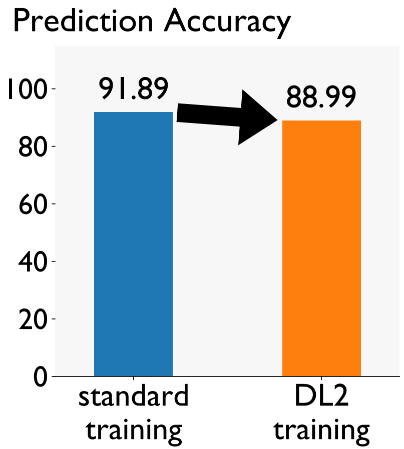


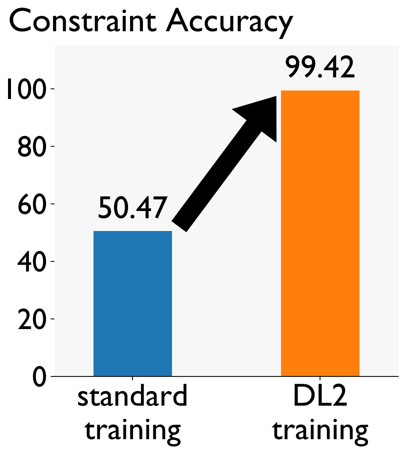
Resnet-18 on CIFAR-10

Supervised Training Example

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

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





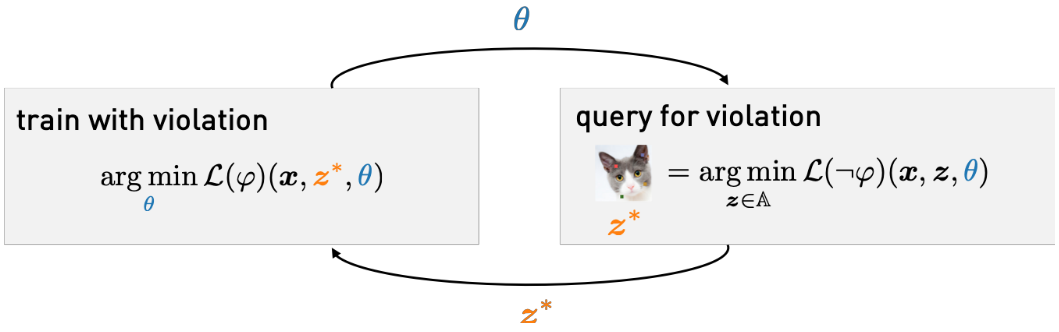
Resnet-18 on CIFAR-10

DL2: Training and Querying Neural Networks with Logic

Querying

```
find i[32, 32, 3]
where i in [0, 1],
      class(NN1(i)) = dog,
      class(NN2(i)) = cat,
      ||i - image||₂ < 2,
      NN1(i).p[7] > 0.8
      NN1(i).p[1] < 0.1
```

Training



generalizes adversarial robustness training
generalizes previous work for training with constraints
applicable to supervised, semi-supervised and unsupervised training

github.com/eth-sri/dl2