Stochastic optimization

- Optimization under random data $(X, Y) \sim P$
  - X: feature vector (e.g. image), Y: label

- Loss/Objective $\ell(\theta; X, Y)$ where $\theta \in \Theta$ is the parameter to be learned

- Optimize average performance under P

$$\text{minimize}_{\theta \in \Theta} \; \mathbb{E}_P[\ell(\theta; X, Y)]$$

Classification

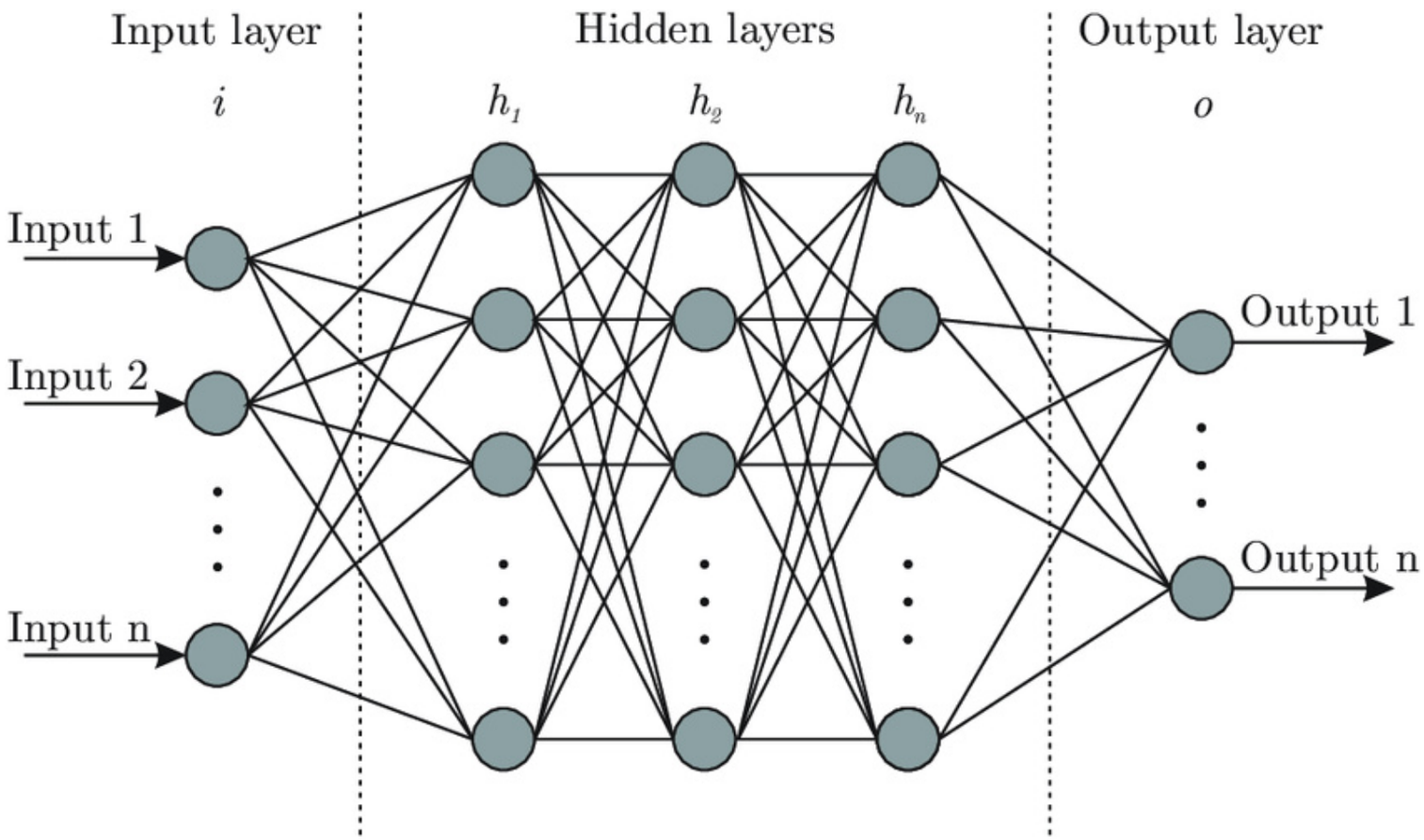
$Y \in \{1, \cdots, K\}$ $\quad \theta = [\theta_1 \cdots \theta_K] \in \mathbb{R}^{d \times K}$

Logit model $\quad p_\theta(y|x) = \dfrac{\exp(\theta_y^T x)}{\sum_{k=1}^K \exp(\theta_k^T x)}$

Max log likelihood (MLE)

$$\min_{\theta \in \Theta} \{ -\mathbb{E} \log p_\theta(Y|X) = -\mathbb{E}\, \theta_Y^T X + \mathbb{E} \log \textstyle\sum_{k=1}^K \exp(\theta_k^T x) \}$$

Neural networks



Instead of linear models $\theta_h^T x$, use $h_{\theta,h}(x)$ represented by a neural network

Neural networks

Weight matrices: $W_1, \cdots, W_L$, intercepts: $b_1, \cdots, b_L$

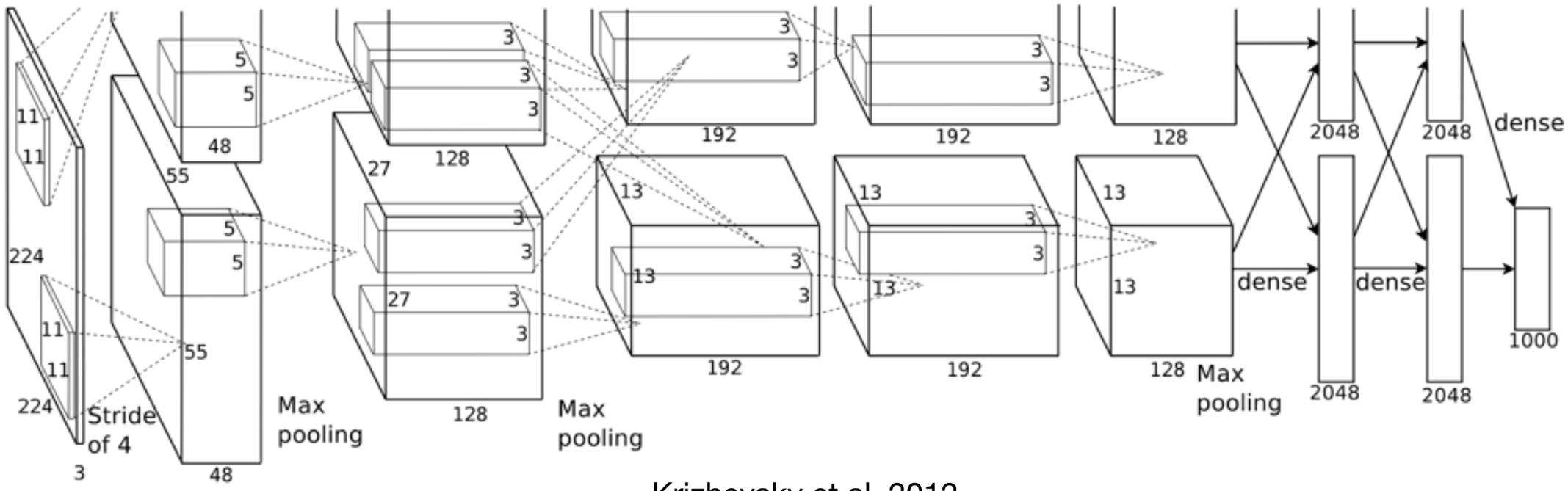
Activation function $\sigma_1, \cdots, \sigma_L$, $\sigma_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{d_i}$, $\begin{pmatrix} d_L = K \\ d_0 = \dim(x) \end{pmatrix}$

e.g. ReLU $\sigma(x) = \max(0, x)$ element-wise

$$h_\theta(x) = \sigma_L\left(W_L \sigma_{L-1}\left(W_{L-1} \cdots \sigma_2\left(W_2 \sigma_1(W_1 x + b_1) + b_2\right) \cdots\right) + b_L\right) \in \mathbb{R}^K$$

Final loss $\ell(\theta; x, y) = -\log \dfrac{\exp(h_{\theta,y}(x))}{\sum_{k=1}^{K} \exp(h_{\theta,k}(x))}$

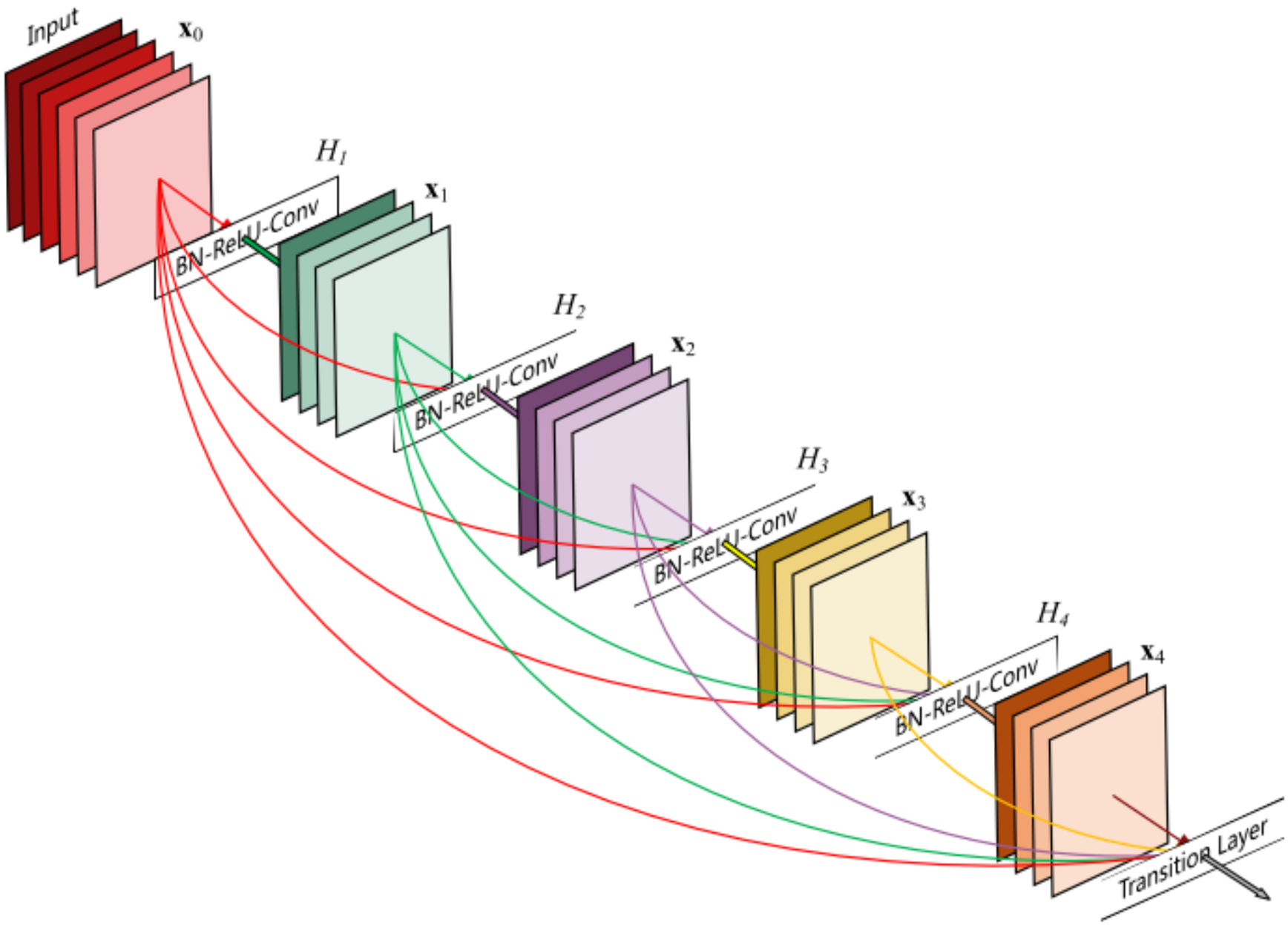
Convolutional nets



Krizhevsky et al. 2012

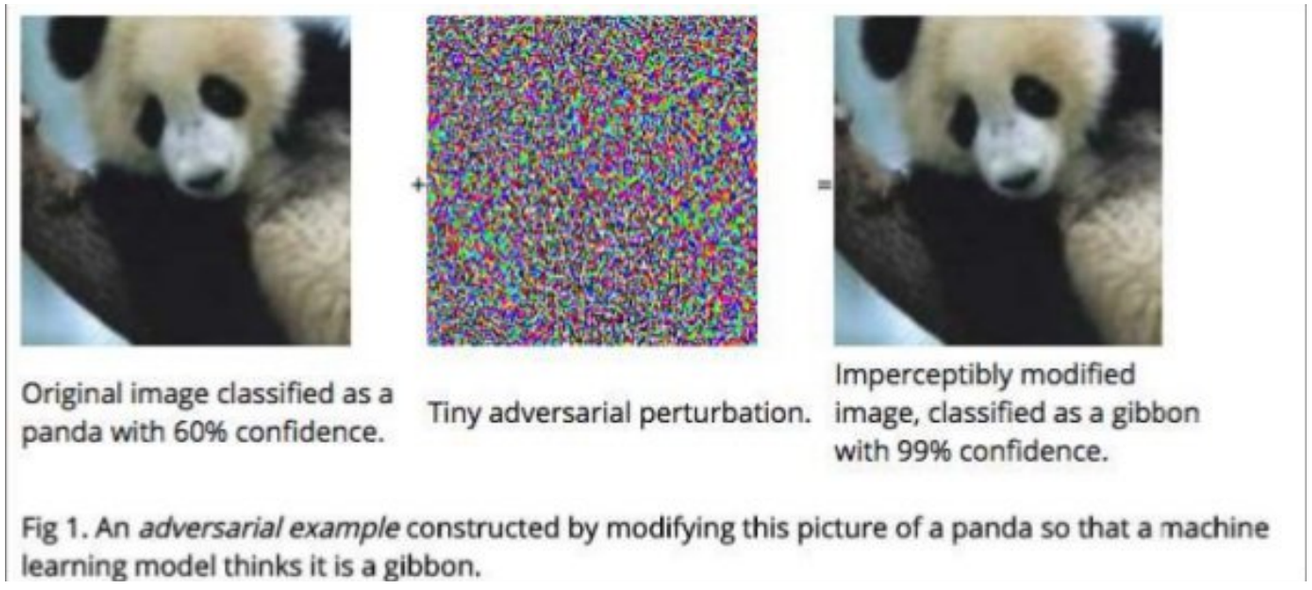
Residual nets

He et al. 2015



SOTA models are non-robust

- Deep networks are very brittle
  - imperceptible adversarial perturbations can fool them

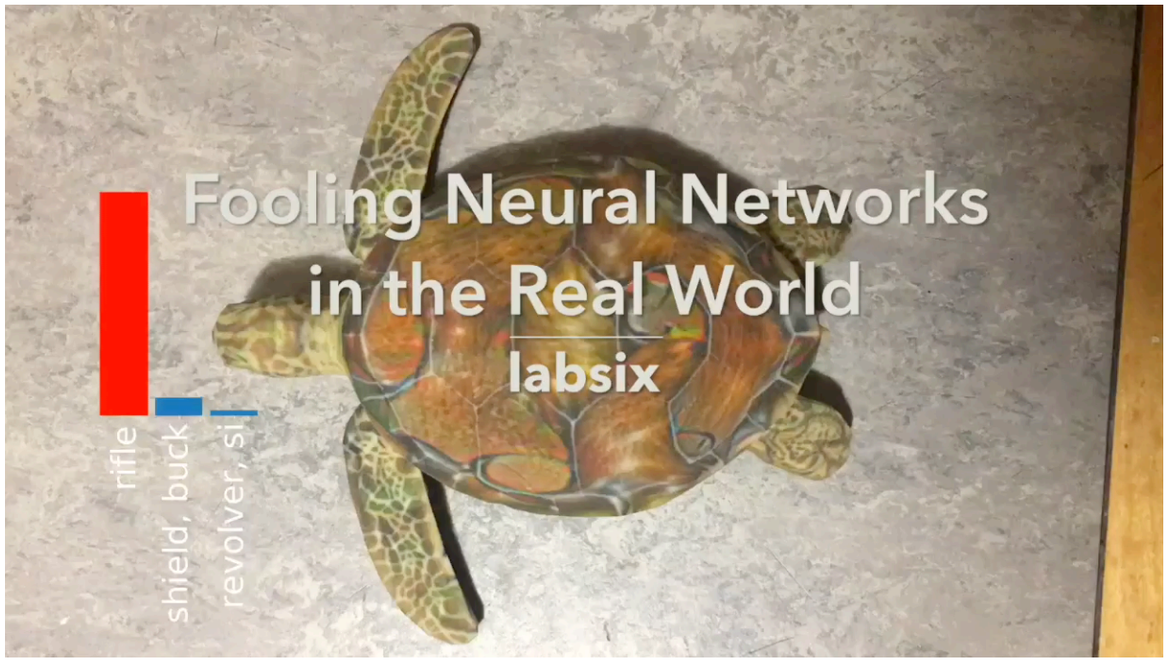


Goodfellow et al. (2015)

Nicholas Carlini

SOTA models are non-robust

- Deep networks are very brittle
  - imperceptible adversarial perturbations can fool them



[Athalye et al. ’17]

[Chen et al. ’18]

Robust optimization

- In the most basic form, want robustness against imperceptible pixel perturbations

$$\text{minimize}_{\theta \in \Theta} \; \mathbb{E}_P \left[\sup_{x' : \|x' - X\|_\infty \leq \epsilon} \ell(\theta; x', Y) \right]$$

- Consider $\epsilon > 0$ small enough
  - such that ϵ-perturbations to pixels don't change label

.

Envelope theorem

Let $(\theta, x') \mapsto \ell(\theta; x', y)$ be continuously differentiable, and let $\mathcal{X}$ be a compact set. Then,

$$\nabla_\theta \sup_{x' \in \mathcal{X}} \ell(\theta; x', y) = \nabla_\theta \ell(\theta; x^\star, y)$$

where $x^\star = \operatorname{argmax}_{x' \in \mathcal{X}} \ell(\theta; x', y)$ is the unique argmax

Remark: This theorem can be generalized in many ways.

Envelope theorem

Recall that the inf-compactness condition is said to hold if there exist $\alpha \in \mathbb{R}$ and a compact set $C \subset X$ such that for every u near u_0, the level set

$$\operatorname{lev}_\alpha f(\cdot, u) := \{x \in \Phi : f(x, u) \leq \alpha\}$$

is nonempty and contained in C.

Proposition 4.12 *Suppose that*

(i) *the function $f(x, u)$ is continuous on $X \times U$,*

(ii) *the inf-compactness condition holds,*

(iii) *for any $x \in \Phi$ the function $f_x(\cdot) := f(x, \cdot)$ is directionally differentiable at u_0,*

(iv) *if $d \in U$, $t_n \downarrow 0$ and $\{x_n\}$ is a sequence in C, then $\{x_n\}$ has a limit point $\bar{x}$ such that*

$$\limsup_{n \to \infty} \frac{f(x_n, u_0 + t_n d) - f(x_n, u_0)}{t_n} \geq f'_{\bar{x}}(u_0, d). \tag{4.27}$$

Then the optimal value function $v(u)$ is directionally differentiable at u_0 and

$$v'(u_0, d) = \inf_{x \in \mathcal{S}(u_0)} f'_x(u_0, d). \tag{4.28}$$

Moreover, if $x_n \in \mathcal{S}(u_0 + t_n d)$ for some $t_n \downarrow 0$, then any limit point $\bar{x}$ of $\{x_n\}$ belongs to $\mathcal{S}_1(u_0, d)$, where

$$\mathcal{S}_1(u_0, d) := \arg\min_{x \in \mathcal{S}(u_0)} f'_x(u_0, d). \tag{4.29}$$

Bonnans and Shapiro (2000)

Challenges

- Inner supremum is (very) non-concave in x
- Don’t even know how to efficiently evaluate robust loss
- Virtually all papers in this domain take heuristic approach to train/evaluate models
  - Non-convex optimization on NNs is fine, but we often want to actually certify robustness

Heuristic: adversarial training

$\mathcal{X}(x) := \{x' : \|x - x'\|_\infty \le \epsilon\}$

At each iteration,

Sample a training data point (X, Y)

Run T_{adv} number of gradient ascent steps on the adversarial loss

$$X' \leftarrow \text{Proj}_{\mathcal{X}(x)}\left(X' + \alpha_{t,x} \cdot \nabla_x \ell(\theta; X', Y)\right)$$

Take SGD step on θ at X'

$$\theta \leftarrow \theta - \alpha_{t,\theta} \cdot \nabla_\theta \ell(\theta; X', Y)$$

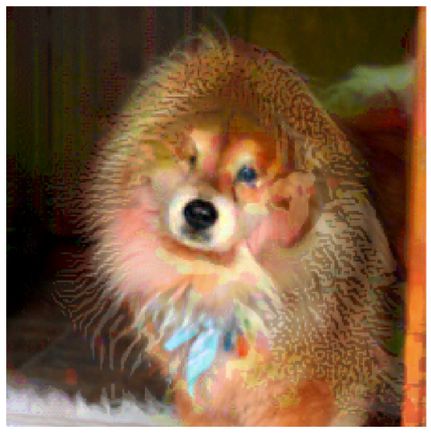
Challenges (cont.)

- Local search only provides a lower bound to robust loss
- During model updates, this incentivizes finding $\theta \in \Theta$ where this proxy is especially loose
- Generally a good idea to make attacks way more powerful during evaluation
  - Give it more computational budget (e.g. higher T_{adv})

Questions

- How should I choose $\epsilon > 0$?
- Why only infinity norm? How about other distances / perturbations?
- Do robustness carry over among different choices of norm and radius? A: Not really.

Beyond infinity norm



L_∞

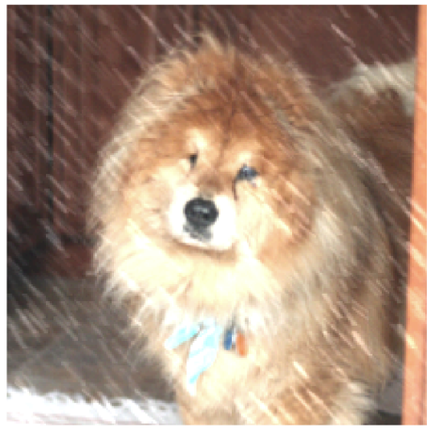
L_2

L_1

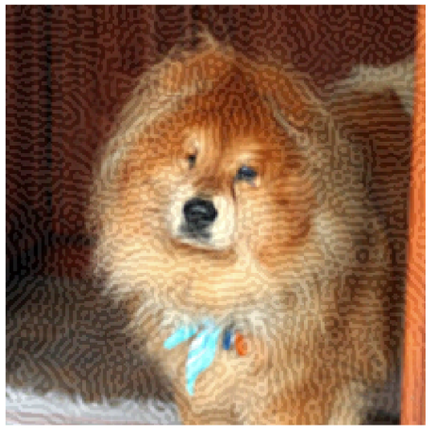
Elastic

JPEG

Fog



Snow



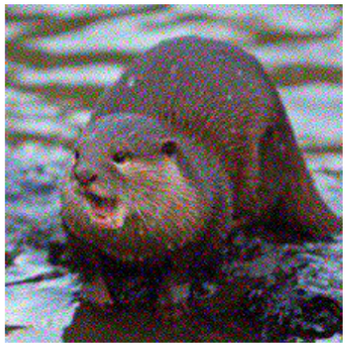
Gabor

Kang et al. (2020)

https://arxiv.org/abs/1909.04068

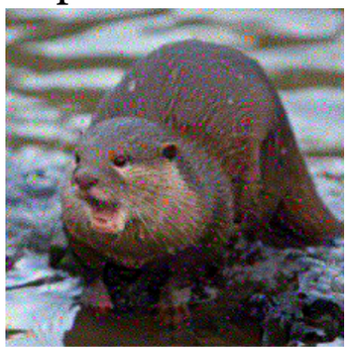
Beyond infinity norm

Randomly Initialized JPEG



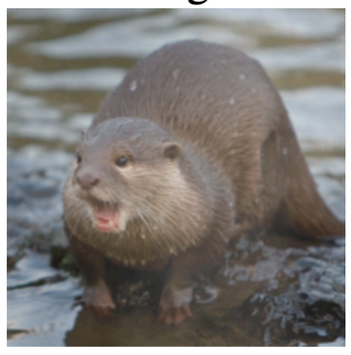
Otter (100.0%)

Adversarially Optimized JPEG



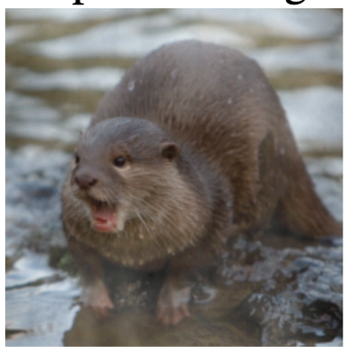
Basketball (100.0%)

Randomly Initialized Fog



Otter (100.0%)

Adversarially Optimized Fog

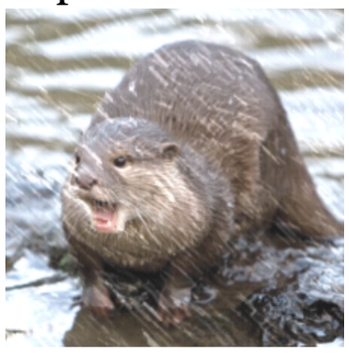


Titi Monkey (100.0%)

Randomly Initialized Snow

Otter (100.0%)

Adversarially Optimized Snow

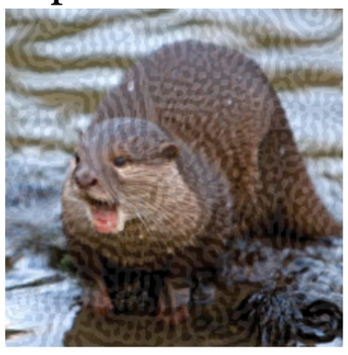


Loafer (98.0%)

Randomly Initialized Gabor

Otter (100.0%)

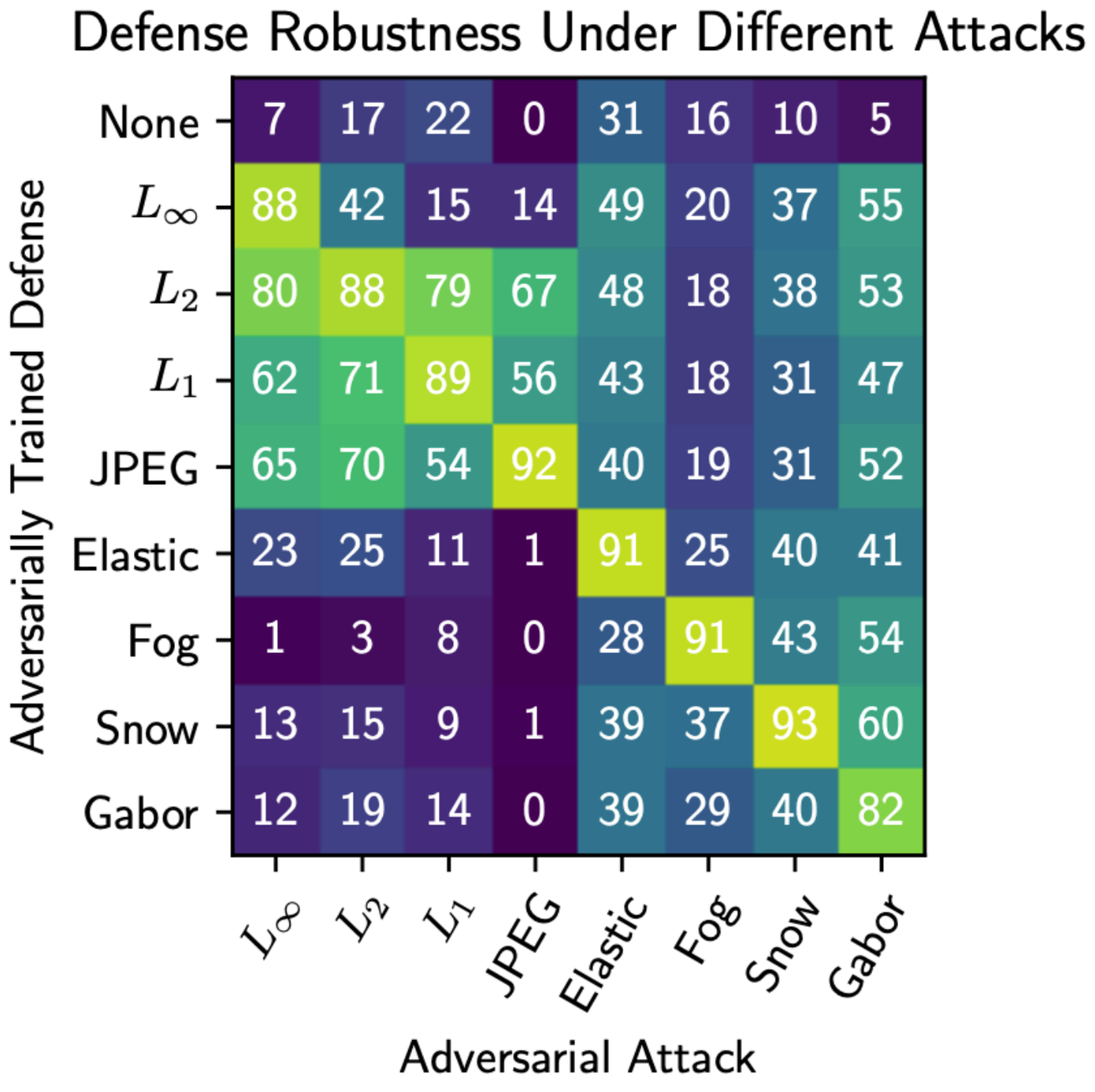
Adversarially Optimized Gabor



Zebra (100.0%)

Kang et al. (2020)

Choice of distance matters



Kang et al. (2020)

Further questions

- Generalization under robustness?
- Adaptive perturbations (e.g. on a learned feature space)
- How do we trade-off average-case and worst-case?

- Previous caveats for DRO largely applies here.
- Often good engineering >>>> theoretical attempts

Very, very active and quickly saturating area.

Rest of today

- Let’s compute upper bounds to robust loss, and train / evaluate models with respect to this bound

- Because this is an upper bound, we can certify robustness of models

- But these can be too conservative

- These bounds don’t (yet) scale to large datasets (e.g. ImageNet)

Certified Adversarial Training

We focus on NNs with ReLU activations. Fix a data point (x, y). Our goal is to obtain tractable upper bounds on the robust loss

$$\sup_{x' : \|x - x'\|_\infty \le \epsilon} \ell(\theta; x', y)$$

that can be computed via convex opt.

Neural network:

$$\begin{aligned} z_0 &:= x \\ z_1 &= W_0 z_0 + b_0 \\ x_1 &= \max(z_1, 0) \quad \text{(elementwise)} \\ z_2 &= W_1 x_1 + b_1 \\ &\vdots \\ x_{L-1} &= \max(z_{L-1}, 0) \\ z_L &= W_{L-1} x_{L-1} + b_{L-1} \end{aligned}$$

$$\theta = [W_i, b_i]_{i=0}^{L-1}$$

$$\ell(\theta; x, y) = -z_{L,y} + \log\left(\sum_{k=1}^{K} \exp(z_{L,k})\right)$$

We focus on $K = 2$, binary classification for ease of exposition.

Assume $Y \in \{1, 2\}$, and focus on $y = 1$ w.l.o.g.

Want to upper bound

$$\sup \left\{ -z_{L,1} + \log\left(e^{z_{L,1}} + e^{z_{L,2}}\right) = \log\left(1 + e^{z_{L,2} - z_{L,1}}\right) \right\}$$

$$\text{s.t.} \quad \begin{aligned} z_{i+1} &= W_i x_i + b_i \quad \forall i = 0, \dots, L-1 \\ x_i &= \max(z_i, 0) \quad \forall i = 1, \dots, L-1 \end{aligned}, \quad \|z_0 - x\|_\infty \le \epsilon$$

This is equivalent to upper bounding $\sup z_{L,2} - z_{L,1}$ subject to the same constraints as above.

$$\max \quad z_{L,2} - z_{L,1} \qquad \cdots (*)$$

$$\text{s.t.} \quad z_{i+1} = W_i x_i + b_i \qquad \forall i = 0, \cdots, L-1$$
$$x_i = \max(z_i, 0) \qquad \forall i = 1, \cdots, L-1 \quad \leftarrow \text{Non-convex}$$
$$|x_{0,j} - x_j| \le \epsilon \qquad \forall j = 1, \cdots, d$$

LP Relaxation (Kolter & Wong '18)

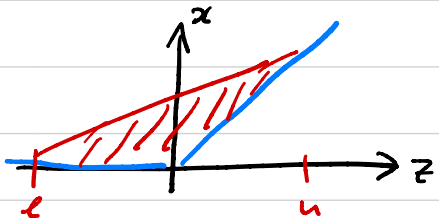
Relax $x_i = \max(z_i, 0)$ onto $x_i \ge z_i$, $x_i \ge 0$ (elementwise)

Problem: x_i may be unbounded so LP may also be unbounded.

What if we know that $z_i \in [l_i, u_i]$ (elementwise) ?

- If $l_{ij} \le 0 \le u_{ij}$, then relax to

$$x_{ij} \ge z_{ij}, \quad x_{ij} \ge 0, \quad x_{ij} \le \frac{z_{ij} - l_{ij}}{u_{ij} - l_{ij}} u_{ij}$$



- If $0 \le l_{ij} \le u_{ij}$, then $x_{ij} = z_{ij}$
- If $l_{ij} \le u_{ij} \le 0$, then $x_{ij} = 0$

We denote

$$I_i := \{j : l_{ij} \le 0 \le u_{ij}\}$$
$$I_i^+ := \{j : 0 < l_{ij} \le u_{ij}\}$$
$$I_i^- := \{j : l_{ij} \le u_{ij} < 0\}$$

We end up with the following relaxation

$$\max \quad z_{L,2} - z_{L,1}$$

$$\text{s.t.} \quad z_{i+1} = W_i x_i + b_i \qquad \forall i = 0, \cdots, L-1$$
$$x_i \ge z_i, \quad x_i \ge 0 \qquad \forall i = 1, \cdots, L-1$$
$$x_{ij} \le \frac{z_{ij} - l_{ij}}{u_{ij} - l_{ij}} u_{ij} \quad \forall j \in I_i, \quad x_{ij} = 0 \quad \forall j \in I_i^-, \quad x_{ij} = u_{ij} \quad \forall j \in I_i^+$$
$$-\epsilon \le x_{0,j} - x_j \le \epsilon \qquad \forall j = 1, \cdots, d$$

Computing this for every example (& every class for multi-class) is prohibitive. To make this more efficient, one approach is to "solve" the dual very inaccurately. Note that this will still be an upper bound to the problem (*) by weak duality.

Red: dual variables

max $z_{L,2} - z_{L,1}$

s.t. $z_{i+1} = W_i x_i + b_i$ (v_{i+1}) $\quad \forall i = 0, \cdots, L-1$

$z_i \to x_i \geq z_i, \quad x_i \geq 0$ ($+ M_i$) $\quad \forall i = 1, \cdots, L-1$

$\lambda_{ij} \to x_{ij} \leq \frac{z_{ij} - l_{ij}}{u_{ij} - l_{ij}} u_{ij} \quad \forall j \in \mathcal{I}_i, \quad x_{ij} = 0 \quad \forall j \in \mathcal{I}_i^-, \quad x_{ij} = u_{ij} \quad \forall j \in \mathcal{I}_i^+$

$-\varepsilon \leq x_{0,j} - x_j \leq \varepsilon \quad \forall j = 1, \cdots, d$

(ξ^- for the left inequality, ξ^+ for the right inequality)

min $(x+\varepsilon)^T \xi_+ - (x-\varepsilon)^T \xi_- + \sum_{i=1}^{L-1} v_{i+1}^T b_i - \sum_{i=2}^{L-1} \lambda_i^T (u_i - l_i)$

s.t. $v_L = [1, -1], \quad v_{ij} = \begin{cases} 0 & j \in \mathcal{I}_i^- \\ (W_i^T v_{i+1})_j & j \in \mathcal{I}_i^+ \end{cases} \quad \forall i = 0, \cdots, L-1$

$(u_{ij} - l_{ij})\lambda_{ij} - \mu_{ij} - \tau_{ij} = (W_i^T v_{i+1})_j$ (①), $\quad v_{ij} = u_{ij}\lambda_{ij} - \mu_{ij} \quad \forall j \in \mathcal{I}_i$ (②)

$W_0^T v_1 = \xi^+ - \xi^-, \quad \lambda, \mu, \tau, \xi_+, \xi_- \geq 0$

To simplify the dual, we use strict complementarity. At the dual optimum,

- For $j \in \mathcal{I}_i$, either λ_{ij} or $\mu_{ij} + \tau_{ij}$ is zero.

↳ since the bounds max $(z_{ij}, 0) \leq x_{ij} \leq \frac{z_{ij} - l_{ij}}{u_{ij} - l_{ij}} u_{ij}$ cannot be simultaneously tight

This implies, at the opt, $(u_{ij} - l_{ij})\lambda_{ij} = [(W_i^T v_{i+1})_j]_+$, $\tau_{ij} + \mu_{ij} = [(W_i^T v_{i+1})_j]_-$ from (①)

Plugging this into (②), $v_{ij} = \frac{u_{ij}}{u_{ij} - l_{ij}} [(W_i^T v_{i+1})_j]_+ - \alpha_{ij} [(W_i^T v_{i+1})_j]_-$

for some $\alpha_{ij} \in [0, 1]$.

Using α to remove μ, we get the simplified dual

$$\min_{\alpha_{ij} \in [0,1]} \sum_{i=0}^{L-1} \nu_{i+1}^T b_i - x^T \hat{\nu}_1 - \varepsilon \|\hat{\nu}_1\|_1 + \sum_{i=1}^{L-1} \sum_{j \in I_i} l_{ij} [\nu_{ij}]_+$$

$$\text{s.t.} \quad \nu_L = [I, -I], \quad \hat{\nu}_i = W_i^T \nu_{i+1} \quad i = L-1, \dots, 0$$

$$\nu_{ij} = \begin{cases} 0 & j \in \hat{I}_i^- \\ \hat{\nu}_{ij} & j \in \hat{I}_i^+ \\ \frac{u_{ij}}{u_{ij} - l_{ij}} [\hat{\nu}_{ij}]_+ - \alpha_{ij} [\hat{\nu}_{ij}]_- & j \in I_i \end{cases} \qquad \forall i = L-1, \dots, 1$$

For any fixed α, the objective can be computed via backward induction.

Wong & Kolter propose fixing $\alpha_{ij} = \frac{u_{ij}}{u_{ij} - l_{ij}}$, and doing backprop to get the upper bound on (*).

How do you know u, l? You don't, so we compute this inductively.

Assume you know $u_{1,j}, l_{1,j}, \dots, u_{I-1,j}, l_{I-1,j} \quad \forall j$.

$$\max / \min \quad z_{I,j}$$

$$\text{s.t.} \quad z_{i+1} = W_i x_i + b_i \qquad \forall i = 0, \dots, I-1$$

$$x_i \geq z_i, \quad x_i \geq 0 \qquad \forall i = 1, \dots, I-1$$

$$x_{ij} \leq \frac{z_{ij} - l_{ij}}{u_{ij} - l_{ij}} u_{ij} \quad \forall j \in I_i, \qquad x_{ij} = 0 \quad \forall j \in I_i^-, \qquad x_{ij} = u_{ij} \quad \forall j \in I_i^+$$

$$-\varepsilon \leq x_{0,j} - x_j \leq \varepsilon \qquad \forall j = 1, \dots, d$$

$\Rightarrow$ Formulate dual, plug-in a solution. This gives $u_{I,j}, l_{I,j}$.

While this is still computationally expensive, we now arrive at $\ell_{rob}(\theta; x, y) \geq \sup_{x' : \|x - x'\|_\infty \leq \epsilon} \ell(\theta; x', y)$, which can be used to

1) train models 2) certify robustness.

For a given test data point $(x, y=1)$, and a model θ, we can say that this model is robust to ϵ-adversarial noise if

$$\max \; z_{L,2} - z_{L,1} \leq 0$$

$$\text{s.t.} \quad z_{i+1} = W_i x_i + b_i \qquad \forall i = 0, \cdots, L-1$$

$$x_i \geq z_i, \quad x_i \geq 0 \qquad \forall i = 1, \cdots, L-1$$

$$x_{ij} \leq \frac{z_{ij} - l_{ij}}{u_{ij} - l_{ij}} u_{ij} \qquad \forall j \in \mathcal{I}_i, \qquad x_{ij} = 0 \quad \forall j \in \mathcal{I}_i^-, \qquad x_{ij} = u_{ij} \quad \forall j \in \mathcal{I}_i^+$$

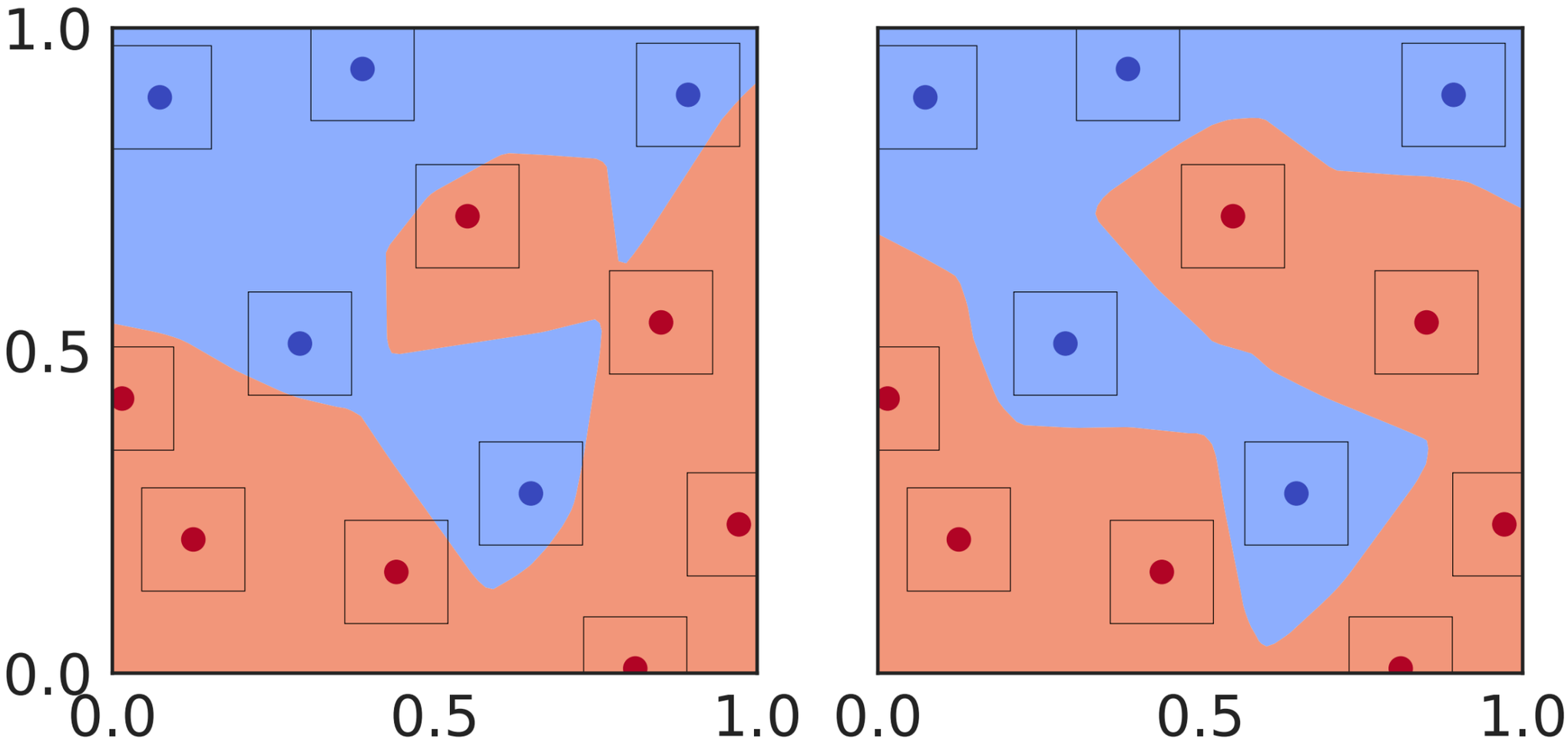
$$-\epsilon \leq x_{0,j} - x_j \leq \epsilon \qquad \forall j = 1, \cdots, d$$

Counting the number of test examples where this happens, we get the upper bound on the robust accuracy.

<u>Remark</u>

1) This approach can be quite loose. Too loose outside MNIST.
2) Computationally expensive, does not scale outside MNIST
3) Similar approach can be derived for any $\|\cdot\|_p$.

Certified adversarial training



Wong and Kolter (2018)

Certified adversarial training

Wong and Kolter (2018)

MNIST

Very weak ConvNet:
"Two conv layers with 16 and 32 channels (each with a stride of two, to decrease the resolution by half without requiring max pooling layers), and two fully connected layers stepping down to 100 and then 10 (the output dimension) hidden units, with ReLUs following each layer except the last."

Table 1. Error rates for various problems and attacks, and our robust bound for baseline and robust models.

PROBLEM	ROBUST	ϵ	TEST ERROR	FGSM ERROR	PGD ERROR	ROBUST ERROR BOUND
MNIST	$\times$	0.1	1.07%	50.01%	81.68%	100%
MNIST	$\surd$	0.1	1.80%	3.93%	4.11%	5.82%
SVHN	$\times$	0.01	16.01%	62.21%	83.43%	100%
SVHN	$\surd$	0.01	20.38%	33.28%	33.74%	40.67%

SDP Relaxations — Raghunathan et al. (2018)

$x = \max(z, 0) \iff x \geq z,\ x \geq 0,\ x(x-z) = 0$

Claim Let $S = \begin{bmatrix} 1 & x & z \\ x & a & b \\ z & b & c \end{bmatrix}$, where a, b, c represents x^2, xz, z^2.

$x \geq z,\ x \geq 0,\ x \cdot (x-z) = 0 \iff S \succeq 0,\ x \geq z,\ x \geq 0,\ a = b,$
$\text{rank}(S) = 1$

Pf '$\Rightarrow$' Take any such (x, z). Let $s = \begin{bmatrix} 1 \\ x \\ z \end{bmatrix}$. Then, $S := ss^T$ satisfies ↑

'$\Leftarrow$' $S = ss^T$ for some s, where $s = \begin{bmatrix} 1 \\ s_2 \\ s_3 \end{bmatrix}$, $s_2^2 = a$, $s_2 s_3 = b$, $s_3^2 = c$.
So $s_2 = x$, $s_3 = z$.

Instead of $x = \max(z, 0)$, consider $\begin{bmatrix} 1 & x & z \\ x & a & a \\ z & a & c \end{bmatrix} \succeq 0$, $x \geq z$, $x \geq 0$

SDP relaxation (no rank constr)

LP is generally faster, but SDP relaxation is often tighter.
LP has had more empirical success, but both only scale to CIFAR.

Open questions:
1) Scaling to ImageNet
2) Combine with architecture design
("folk thm" smooth networks are more robust)
3) Certification across other notions of robustness.