Measuring Neural Net Robustness with Constraints

Osbert Bastani[1], Yani Ioannou[2], Leonidas Lampropoulos[3],
Dimitrios Vytiniotis[4], Aditya V. Nori[4], Antonio Criminisi[4]

[1]Stanford University, [2]University of Cambridge, [3]University of Pennsylvania, [4]Microsoft Research Cambridge

Summary

Motivation: Despite having high accuracy, neural nets have been shown to be susceptible to adversarial examples, where a small perturbation to an input can cause it to become mislabeled

Algorithm: We propose metrics for measuring neural net robustness and devise a novel algorithm to approximate these metrics

Evaluation: We evaluate the robustness of deep neural nets with experiments on the MNIST and CIFAR-10 datasets:

- We generate more accurate estimates of robustness metrics than existing algorithms
- We use discovered adversarial examples to fine-tune neural nets, and show that existing algorithms for improving robustness "overfit" to specific kinds of adversarial examples

Related literature: Existing algorithms have been proposed for finding adversarial examples:

- Approximated as cost function minimization, and solved using L-BFGS-B (Szegedy et al. 2014)
- Fast signed-gradient heuristic (Goodfellow et al. 2015)

Robustness Metrics

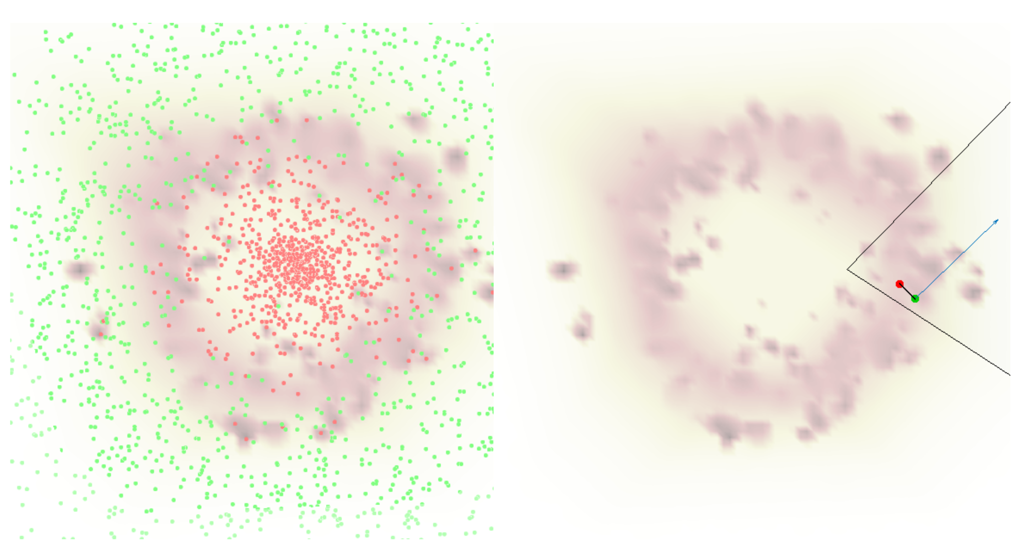
- **Problem setting:**
  - Input space $\mathcal{X} \subseteq \mathbb{R}^n$ and output labels $\mathcal{L} = \{1, \ldots, L\}$
  - Classifier $f: \mathcal{X} \to \mathcal{L}$
  - Distribution $\mathcal{D}$ over inputs $\mathcal{X}$
- Classifier f is (x_*, ϵ) **robust** if all points x s.t. $\|x_* - x\|_\infty \leq \epsilon$ have the same label as x_*
- The **pointwise robustness** of f at x_* is
$$\rho(f, x_*) = \inf\{\epsilon \geq 0 \mid f \text{ is not } (x_*, \epsilon) \text{ robust}\}$$
- The **adversarial frequency** of f is
$$\phi(f, \epsilon) = \Pr_{x_* \sim \mathcal{D}}[\rho(f, x_*) \leq \epsilon]$$
- The **adversarial severity** of f is
$$\mu(f, \epsilon) = \mathbb{E}_{x_* \sim \mathcal{D}}[\rho(f, x_*) \mid \rho(f, x_*) \leq \epsilon]$$

Constraint Formulation

- **Constraint systems:**
  - Linear inequalities: $\mathcal{C} \equiv (w^T x + b \geq 0)$
  - Conjunctions: $\mathcal{C} \equiv \mathcal{C}_1 \vee \mathcal{C}_2$
  - Disjunctions: $\mathcal{C} \equiv \mathcal{C}_1 \wedge \mathcal{C}_2$
- **Neural net f as a constraint system $\mathcal{C}_f(x, \ell)$:**
  - Encodes whether f outputs label ℓ on input x
  - Can be constructed when f is piecewise linear (e.g., ReLUs)
- **Pointwise robustness as constrained optimization:**
$$\rho(f, x_*, \ell) = \inf\left\{\epsilon \geq 0 \mid \mathcal{C}_f(x, \ell) \wedge \|x - x_*\|_\infty \text{ satisfiable}\right\}$$

Approximation

- **Approximation:**
  - Constraint formulation is NP-hard due to disjunctions
  - We restrict the search to a linear region around the input x_*
  - The resulting optimization problem is a linear program (LP)
  - The LP is very large, so we devise an abstraction-refinement constraint solving loop that significantly improves scalability
- Piecewise linear structure of neural nets:



- Generated adversarial examples:

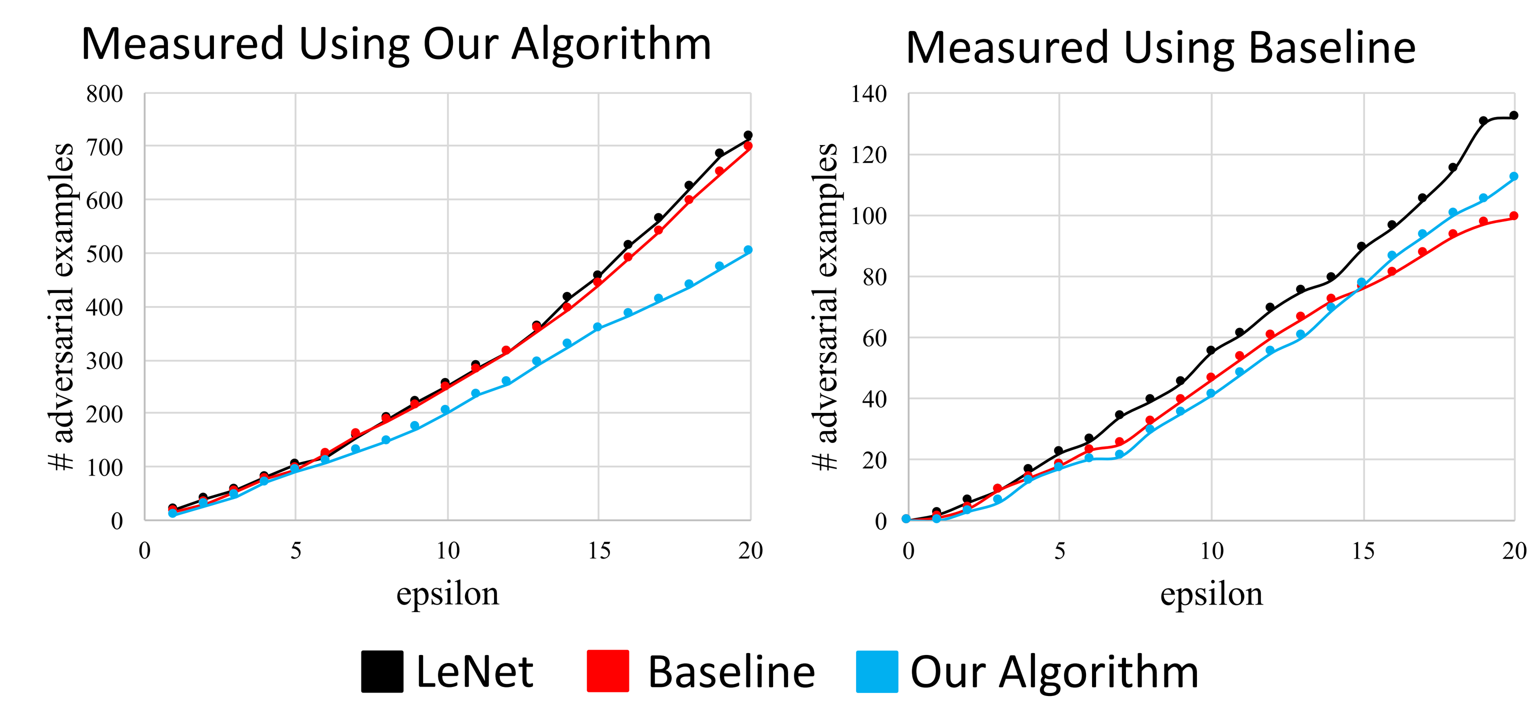
original adversarial perturbation

Evaluation on MNIST

Neural nets: (i) modified LeNet, (ii) fine-tuned using baseline (Szegedy et al. 2014), (iii) fine-tuned using our algorithm

Neural Net	Acc. (%)	Adv. Frequency (%)		Adv. Severity (pixels)	
		Baseline	Ours	Baseline	Ours
Original	99.08	1.32	7.15	11.9	12.4
Baseline	99.15	0.99	6.97	10.9	12.4
Ours	99.23	1.12	5.03	12.2	11.7

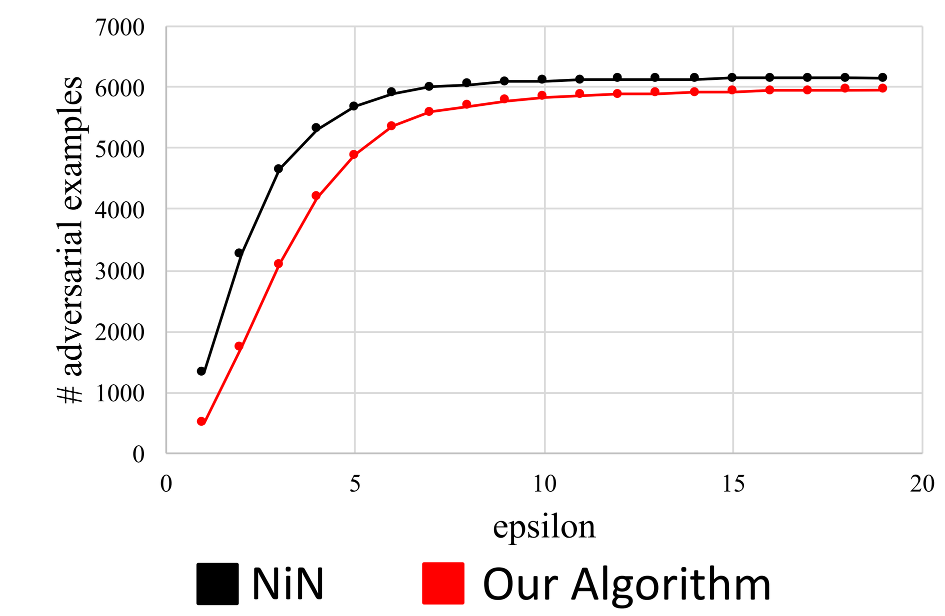
Count of test inputs x_* with adversarial example distance $\leq \epsilon$ away:



Evaluation on CIFAR-10

Neural nets: (i) NiN, (ii) fine-tuned using our algorithm

Count of test inputs x_* with adversarial example distance $\leq \epsilon$ away, measured using our algorithm:



References

Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, Fergus. Intriguing properties of neural networks. ICLR 2014.

Goodfellow, Shlens, Szegedy. Explaining and harnessing adversarial examples. ICLR 2015.