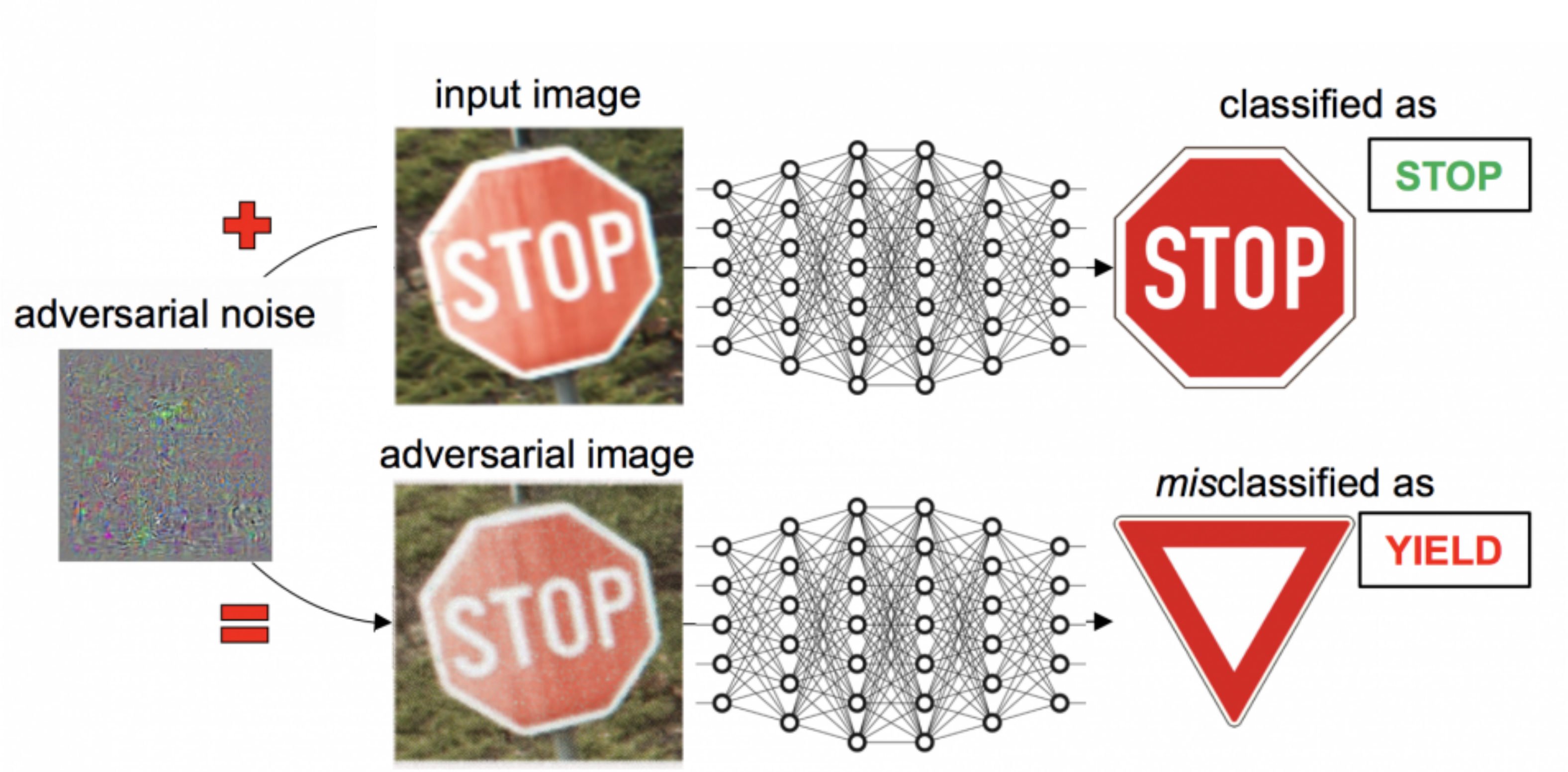
Second-order provable defenses against adversarial examples

Sahil Singla, Soheil Feizi
Department of Computer Science
University of Maryland

https://github.com/singlasahil14/so-robust

What are adversarial examples?



Empirical Defenses against adversarial attacks

- Work empirically but **no theoretical guarantee**

- **Examples**: Adversarial training [Madry et al. 2017, Kurakin et al.'17, Carlini & Wagner '16], Defensive distillation [Papernot et al. 2015], Defense-GAN [Samangouei et al. 2018], CURE [Moosavi et al. 2018], etc.

- Broken by newer adaptive attacks [e.g. Carlini et al. 2017] !

☹

Certified Defenses against adversarial attacks

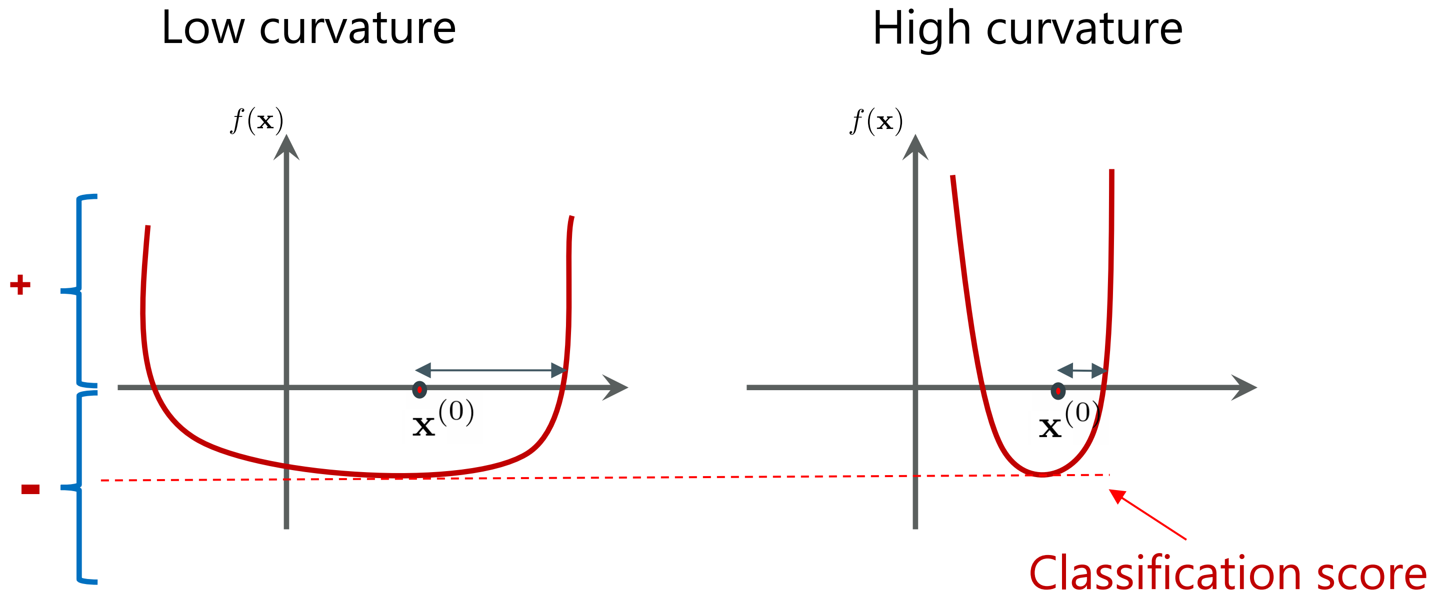
- **Theoretical guarantees** against all attacks within a certain threat model

- **Examples**: Convex-relaxations [Wong et al. 2017], Interval bound propagation [Gowal et al. 2018], Randomized smoothing [Cohen et al. 2019], CROWN-IBP [Zhang et al. 2019], CNN-Cert [Boopathy et al. 2018], etc.

- All use **first-order information of the model** (i.e. gradients)

Question: can higher-order information be used in improving provable robustness?

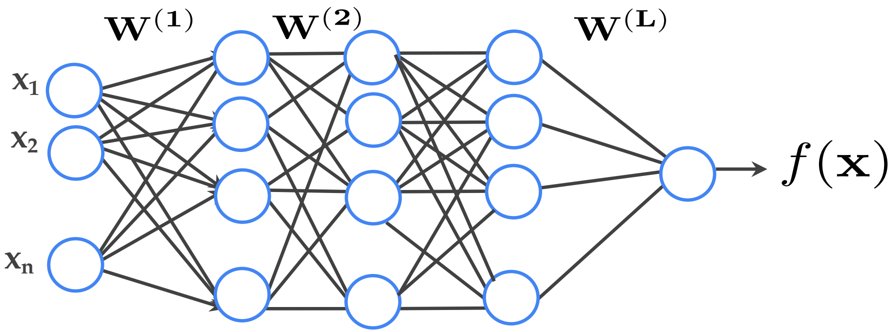
Intuition: Curvature Effect in Robustness



Low curvature translates to large robustness radius

Problem Setup

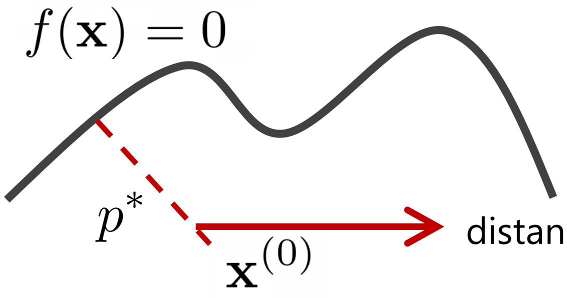
- Classification using deep fully-connected network



$$f(\mathbf{x}) = \mathbf{W}^{(\mathbf{L})}\sigma\left(\mathbf{W}^{(\mathbf{L-1})}\ldots\sigma\left(\mathbf{W}^{(\mathbf{1})}\mathbf{x}\right)\ldots\right)$$

- Differentiable activations (e.g. sigmoid, tanh, softplus, etc.)
- Gradient: $\mathbf{g}(\mathbf{x}) := \nabla_{\mathbf{x}} f(\mathbf{x})$
- Hessian: $\mathbf{H}(\mathbf{x}) := \nabla^2_{\mathbf{x}} f(\mathbf{x})$
- Input to layer I: $\mathbf{z}^{(I)}$
- Output of layer I: $\mathbf{a}^{(I)} = \sigma\left(\mathbf{z}^{(I)}\right)$

Certification problem framework



distance to decision boundary

$$p^* = \min_{\substack{\mathbf{x} \\ f(\mathbf{x})=0}} \frac{1}{2}\|\mathbf{x}-\mathbf{x_0}\|^2 \overset{\text{lagrangian}}{=} \min_{\mathbf{x}} \max_{\eta} \frac{1}{2}\|\mathbf{x}-\mathbf{x_0}\|^2 + \eta f(\mathbf{x})$$

non-convex optimization

$$\overset{\text{min-max}}{\geq} \max_{\eta} \overbrace{\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{x_0}\|^2 + \eta f(\mathbf{x})}^{d(\eta)\ \text{still non-convex}}$$

$$= \max_{\eta} d(\eta)$$

Curvature-based Certificate

- **Theorem**

If $m\mathbf{I} \preccurlyeq \nabla^2_{\mathbf{x}} f \preccurlyeq M\mathbf{I}$ $\forall \mathbf{x} \in \mathbb{R}^n$

$d(\eta)$ can be computed via convex opt for $\frac{-1}{M} \leq \eta \leq \frac{-1}{m}$

$$p^* \geq d^* := \max_{-1/M \leq \eta \leq -1/m} d(\eta)$$

Curvature-based Robustness Certificate (**CRC**)

Tightness property of the proposed approach

$$p^* \geq d^* := \underbrace{\max_{-1/M \leq \eta \leq -1/m} \quad d(\eta)}$$

solution: $(\eta^*, \mathbf{x}^*)$

If $f(\mathbf{x}^*) = 0 \implies primal = dual$

- **No such guarantee** exists for first-order robustness methods!

Similar results for the attack problem framework

	Certificate problem $(-) = cert$	Attack problem $(-) = attack$
primal problem, $p^*_{(-)}$	$\min_{f(\mathbf{x})=0} 1/2\|\mathbf{x}-\mathbf{x}^{(0)}\|^2$	$\min_{\|\mathbf{x}-\mathbf{x}^{(0)}\|\leq\rho} f(\mathbf{x})$
dual function, $d_{(-)}(\eta)$	$\min_{\mathbf{x}} 1/2\|\mathbf{x}-\mathbf{x}^{(0)}\|^2+\eta f(\mathbf{x})$	$\min_{\mathbf{x}} f(\mathbf{x})+\eta/2(\|\mathbf{x}-\mathbf{x}^{(0)}\|^2-\rho^2)$
When is dual solvable?	$-1/M\leq\eta\leq-1/m$	$-m\leq\eta$
dual problem, $d^*_{(-)}$	$\max_{-1/M\leq\eta\leq-1/m} d_{cert}(\eta)$	$\max_{-m\leq\eta} d_{attack}(\eta)$
When primal = dual?	$f(\mathbf{x}^{(cert)})=0$	$\|\mathbf{x}^{(attack)}-\mathbf{x}^{(0)}\|=\rho$

f denotes the classifier. ρ is the radius of the ball.

How to compute the curvature bounds?

- **Theorem**

$$\mathbf{H}(\mathbf{x}) = \sum_{I=1}^{L-1} \left(\mathbf{J}^{(I)}\right)^T \operatorname{diag}\left(\mathbf{J}^{(L,I)} \odot \sigma''\left(\mathbf{z}^{(I)}\right)\right) \mathbf{J}^{(I)}$$

Jacobian of $\mathbf{z}^{(I)}$ w.r.t $\mathbf{x}$

Jacobian of $\mathbf{z}^{(L)}$ w.r.t $\mathbf{a}^{(I)}$

- We use this formula to compute the curvature bounds

How to compute the curvature bounds?

- Example: two layer network

$$H(\mathbf{x}) = (\mathbf{W}^{(1)})^T \mathrm{diag}\left(\mathbf{W}^{(2)} \odot \sigma''(\mathbf{z}^{(1)})\right) \mathbf{W}^{(1)}$$

Depends on weights (not the input)

Depends on the input

- For activations tanh, sigmoid, softplus we have

$$h_L \leq \sigma''(x) \leq h_U \qquad \forall x \in \mathbb{R}$$

$$\min(\mathbf{W}_i^{(2)} h_L, \mathbf{W}_i^{(2)} h_U) \leq \mathbf{W}_i^{(2)} \sigma''(\mathbf{z}_i^{(1)}) \leq \max(\mathbf{W}_i^{(2)} h_L, \mathbf{W}_i^{(2)} h_U) \qquad \forall \mathbf{x}$$

How to compute the curvature bounds?

$$\mathbf{N} = \left(\mathbf{W}^{(1)}\right)^T \operatorname{diag}\left(\min(\mathbf{W}^{(2)} h_L, \mathbf{W}^{(2)} h_U)\right) \mathbf{W}^{(1)}$$
$$\mathbf{P} = \left(\mathbf{W}^{(1)}\right)^T \operatorname{diag}\left(\max(\mathbf{W}^{(2)} h_L, \mathbf{W}^{(2)} h_U)\right) \mathbf{W}^{(1)}$$

- This gives the following matrix inequalities:

$$\mathbf{N} \preccurlyeq H(\mathbf{x}) \preccurlyeq \mathbf{P} \qquad \forall \mathbf{x} \in \mathbb{R}^n$$

$$m = -\|\mathbf{N}\|_2, \qquad M = \|\mathbf{P}\|_2$$

$$\Rightarrow \quad \boxed{m\mathbf{I} \preccurlyeq H(\mathbf{x}) \preccurlyeq M\mathbf{I} \qquad \forall \mathbf{x} \in \mathbb{R}^n}$$

- Similar result for deeper nets (with more complex proof)

Confronting the Hessian

- **Newton Step Update (Certificate):**

$$\mathbf{x}^{(k+1)} = -(\mathbf{I} + \eta\mathbf{H}^{(k)})^{-1}\left(\eta\mathbf{g}^{(k)} - \mathbf{x}^{(0)} - \eta\mathbf{H}^{(k)}\mathbf{x}^{(k)}\right)$$

- Since $\frac{-1}{M} \leq \eta \leq \frac{-1}{m} \implies \|\eta\mathbf{H}^{(k)}\|_2 < 1,$

$$(\mathbf{I} + \eta\mathbf{H}^{(k)})^{-1} \approx \mathbf{I} - \eta\mathbf{H}^{(k)} + (\eta\mathbf{H}^{(k)})^2 - (\eta\mathbf{H}^{(k)})^3 \dots$$

- Can efficiently be computed via Hessian vector product!

Training with Curvature Regularization

- Deep networks computed by standard/adversarial training can have very high curvature bounds
- Curvature-based Robust Training (CRT)

Computed using our attack optimization

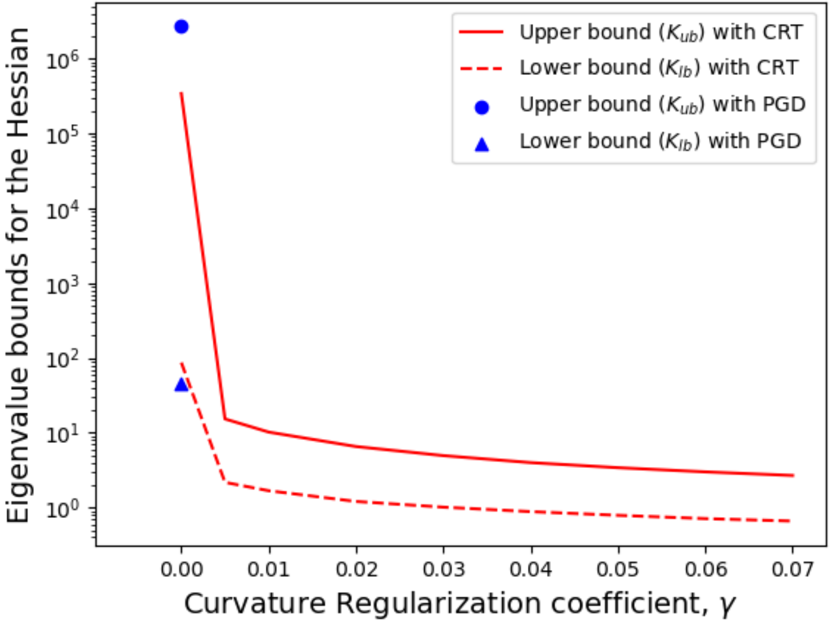
$$\min_\theta \left[\underbrace{\frac{1}{m} \sum_{i=1}^{m} \ell(f_\theta(\mathbf{x}_i^*), y_i)} + \underbrace{\gamma}\, \underbrace{K(\theta)} \right]$$

Cross entropy

Curvature regularization coefficient

Differentiable curvature bound

Empirical results with Curvature Regularization



- 3 layer fully connected network, sigmoid activations, MNIST

Certified Robust accuracy comparison

Network	Training	Standard Accuracy	Certified Robust Accuracy
2×[1024], softplus	**CRT, 0.01**	**98.68%**	**69.79%**
	CROWN-IBP	88.48%	42.36%
2×[1024], relu	COAP	89.33%	44.29%
	CROWN-IBP	89.49%	44.96%
3×[1024], softplus	**CRT, 0.05**	**97.43%**	**57.78%**
	CROWN-IBP	86.58%	42.14%
3×[1024], relu	COAP	89.12%	44.21%
	CROWN-IBP	87.77%	44.74%
4×[1024], softplus	**CRT, 0.07**	**95.60%**	**53.19%**
	CROWN-IBP	82.74%	41.34%
4×[1024], relu	COAP	90.17%	44.66%
	CROWN-IBP	84.4%	43.83%

Comparison between Convex Outer Adversarial Polytope (COAP), CROWN-IBP and Curvature-based Robust Training i.e CRT (ours) with Attack radius $\rho = 1.58$ on the MNIST dataset.

Certificate comparison

Network	Training	Certificate (mean)	
		CROWN	CRC
2×[1024], sigmoid	standard	0.28395	**0.48500**
	$\gamma = 0.01$	0.32548	**0.84719**
	CRT, 0.01	0.43061	**1.54673**
3×[1024], sigmoid	standard	**0.24644**	0.06874
	$\gamma = 0.01$	0.39799	**1.07842**
	CRT, 0.01	0.39603	**1.24100**
4×[1024], sigmoid	standard	**0.19501**	0.00454
	$\gamma = 0.01$	0.40620	**1.05323**
	CRT, 0.01	0.40327	**1.06208**

Comparison between CROWN and Curvature-based Robustness Certificate i.e CRC (ours) on the MNIST dataset.

How frequently primal equals dual?

Network	γ	Accuracy	Certificate success	Attack success
2×[1024], sigmoid	0.	98.77%	2.24%	5.05%
	0.03	98.30%	44.17%	100%
3×[1024], sigmoid	0.	98.52%	0.12%	0.%
	0.05	97.60%	22.59%	100%
4×[1024], sigmoid	0.	98.22%	0.01%	0.%
	0.07	95.24%	19.53%	100%

Certificate success rate is the fraction of points satisfying $f(\mathbf{x}^*) = 0$.

Attack success rate is the fraction satisfying $\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_2 = \rho = 0.5$

Both imply *primal=dual*. Results are on the MNIST dataset.

Results using local, not global curvature bounds

Network	Training	CRC (Global)	CRC (Local)
2×[1024], sigmoid	standard	0.5013	**0.5847**
	CRT, 0.0	1.0011	**1.1741**
	CRT, 0.01	1.5705	**1.6047**
	CRT, 0.02	1.6720	**1.6831**

Comparison between CRC computed using global and local curvature bound on the MNIST dataset with attack radius $\rho = 0.5$ for a 2 layer network.

Extension to convolutional neural networks

γ	MNIST				
	Standard Accuracy	Certified Robust Accuracy	CNN-Cert [4]	**CRC (Ours)**	Certificate Improvement (Percentage %)
0	98.35%	0.0%	0.1503	**0.1770**	17.76%
0.01	94.85%	75.26%	0.2135	**0.8427**	294.70%
0.02	93.18%	74.42%	0.2378	**0.9048**	280.49%
0.03	91.97%	72.89%	0.2547	**0.9162**	259.71%

Comparison between CRC and CNN-Cert for different values of the regularization parameter γ for a single hidden layer convolutional network with the tanh activation function [Singla & Feizi, 2019]. For Certified Robust Accuracy, we use $\rho = 0.5$.

Summary

- We derive a new formulation for the robustness certification that uses the second-order information of the network (i.e. curvature values)
- Our curvature-based certificate is based on two key results:
  - ✓ We derive a closed-form formula for the Hessian of a network with smooth activation functions
  - ✓ We derive differentiable global upper bounds on the curvatures values of the network
- Curvature-based certificates are exact for significant fraction of test inputs.

https://github.com/singlasahil14/so-robust

Questions?