

Countering Adversarial Images using Input Transformations

Chuan Guo, Mayank Rana, Moustapha Cisse, Laurens Van Der Maaten

Presented by

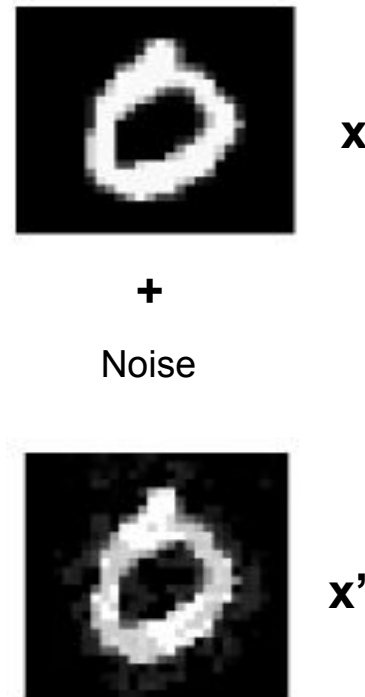
— **Hari Venugopalan, Zainul Abi Din**

Motivation: Why is this a hard problem to solve?

- Adversarial Examples are very easy to generate and very difficult to defend against - problem with neural nets, piecewise linear functions
- Adversarial Examples transfer from one model to another very easily
- This problem is not well understood, researchers are divided on the nature of these samples.

Problem Definition: Adversarial Example

- x is the original input, x' is the perturbed input which is found by adding some noise to x
- Given a classifier $h(x)$, the score $h(x)$ and $h(x')$ should not be the same.
- While $d(x, x')$ is smaller than some threshold.
- Keep the distortion low and move the decision boundary

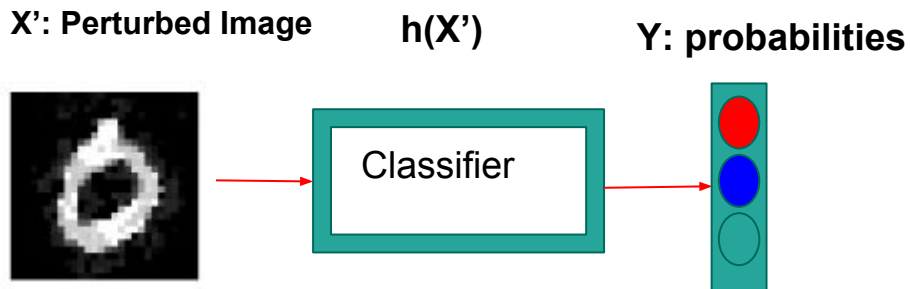
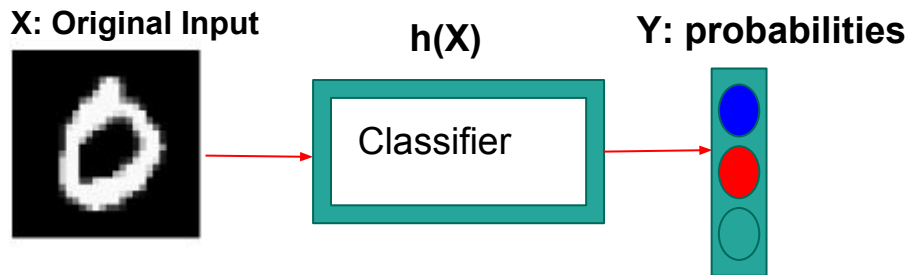


Measure of distortion

- Normalized L2-Dissimilarity
- X is the actual image
- X' is the adversarial image
- N is the total number of images
- Low L2-Dissimilarity

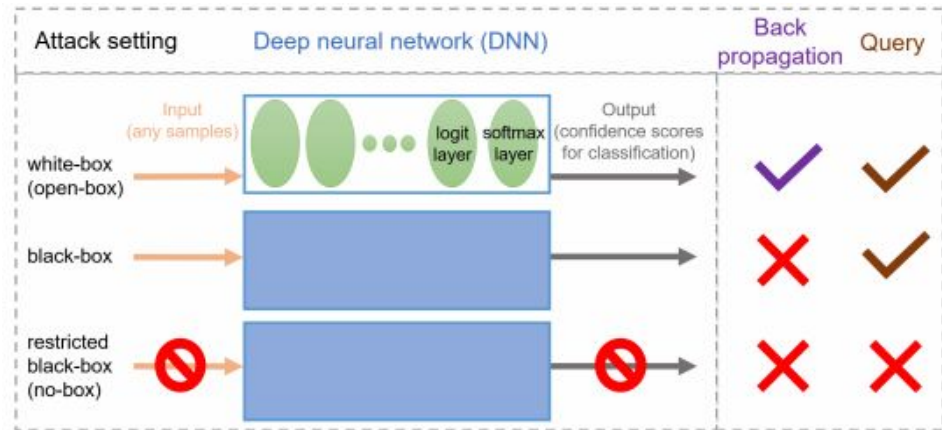
$$\frac{1}{N} \sum_{n=1}^N \frac{\|\mathbf{x}_n - \mathbf{x}'_n\|_2}{\|\mathbf{x}_n\|_2}.$$

Intuitively? What is an attack



Types of attacks in DNNs

- **White-Box:** The attacker is assumed to have complete knowledge of the networks weights and architecture
- **Black-Box:** Does not require internal model access
- **Gray-Box:** Attacker doesn't know the defense strategy used.
- **Targeted Attack:** Attacker selects the class they want the example to be misclassified as.
- **Untargeted Attack:** Any misclassification is the goal



How do you find this noise? Fast Gradient Sign Method

- Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy - 2015

1. $\mathbf{x}' = \mathbf{x} + \eta$
2. $\eta = \text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y))$.

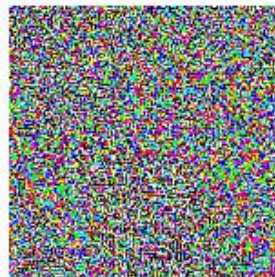


\mathbf{x}

“panda”

57.7% confidence

+ .007 ×



$\text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y))$

“nematode”

8.2% confidence

=



$\mathbf{x} +$

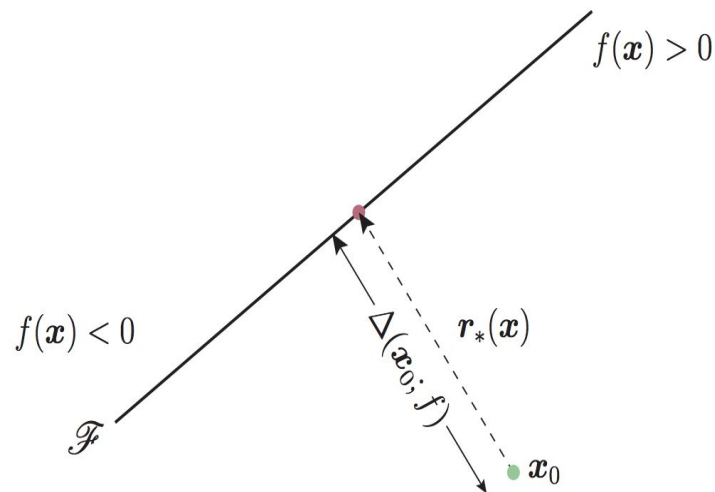
$\epsilon \text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y))$

“gibbon”

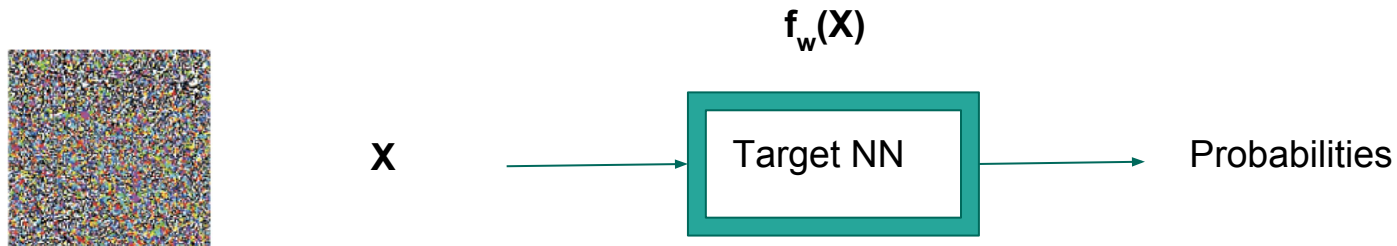
99.3 % confidence

Deep fool:

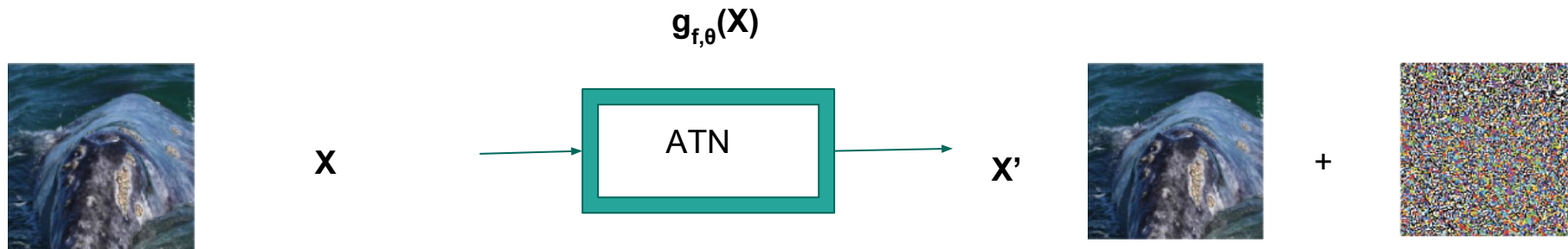
- $f(x)$ is the classifier
- Given a sample x_0 , change $f(x_0)$ sign
- Project x_0 orthogonally on $f(x)$
- $r = \{f(x_0) / \|\nabla_x f(x_0)\|^2\} \nabla_x f(x_0)$
- Keep adding r to x_0 till sign of $f(x_0)$ changes



Adversarial Transformative Networks



Learn the noise



Learn to generate and adversarial sample for Target NN

Image Source: DeepFool: A simple and accurate method to fool deep neural networks, Moosavi Dezfooli

Concept Source: Learning to generate adversarial examples, Shumeet Baluja

ATN Cost- Gradient W.R.T θ

$$\text{Cost} = \sum \beta L_x(g_{f,\theta}(x), x) + L_y(f_w(g_{f,\theta}(x)), r(f_w(x), t))$$

$$r(f(x), t) = \text{norm}(\alpha * \max(f(x); k = t, f(x)); \text{otherwise})$$

Carlini and Wagner Attack:

$$\min_{x'} [\|x - x'\|_2 + \lambda_f \max(-k, Z(x')_{h(x)} - \max\{Z(x')_k : k \neq h(x)\})]$$

- $Z(x')$: Input to the softmax layer
- $Z(x')_k$: k th component of $Z(x')$
- $\max\{Z(x')_k : k \neq h(x)\}$: second largest logit
- $Z(x')_{h(x)} - \max\{Z(x')_k : k \neq h(x)\}$: Difference between second largest logit and largest logit
- k : Confidence

Countering Adversarial Images

- Improve robustness of model
- Exploit randomness and non-differentiability
- **Model specific** and **model agnostic** defenses

Model Specific Defenses

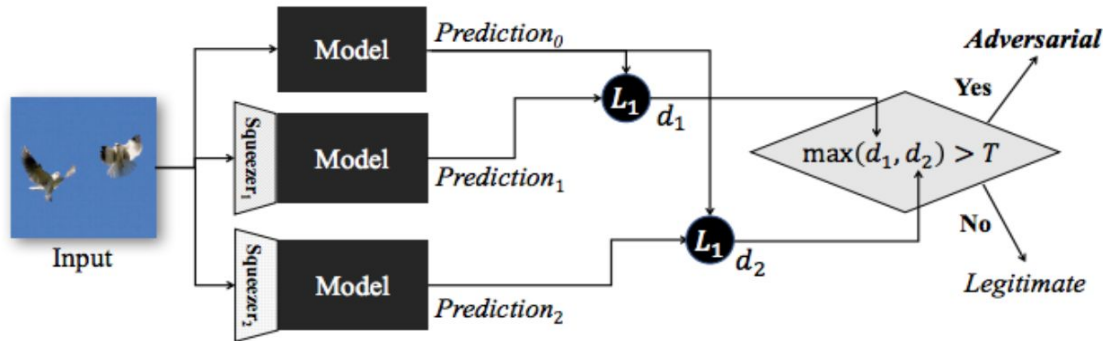
- Based on Robust Optimization
- Minimization-maximization approach is followed
- Learning algorithm and regularization scheme
- Makes assumptions on nature of adversary
- Do not satisfy Kerckhoff's principle

Model Agnostic Defenses

- Transforms images to remove perturbations
- JPEG compression and image re-scaling are examples
- Paper aims to increase effectiveness of model agnostic defenses

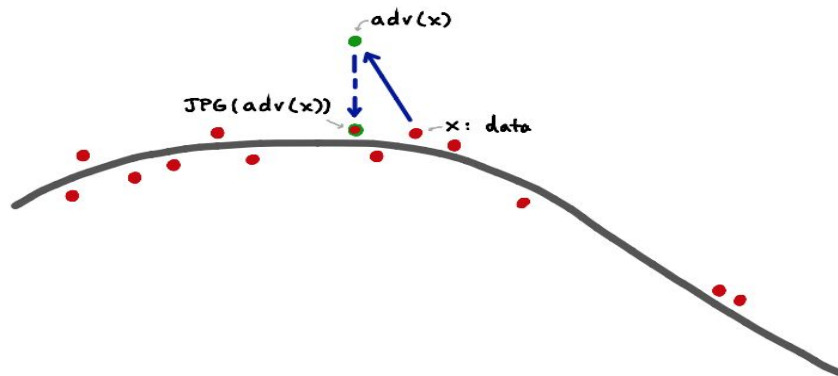
Feature Squeezing

- Proposed by Xu et al
- Detects adversarial inputs
- Input space is reduced by “squeezing out” features
- Outputs are compared in original and reduced spaces
- Different outputs means adversarial inputs



JPEG Compression

- Proposed by Dziugaite et al
- Removes perturbations by compressing images
- Compression could remove aspects of perturbation
- Effective against small-magnitude perturbation
- With larger perturbation, compression is unable to recover non-adversarial image



Total Variation Minimization

- Transformation is taken as an optimization problem
- Transformation is close to input, but also has low variation
- Inspired by noise removal

$$\min_{\mathbf{z}} \|(1 - X) \odot (\mathbf{z} - \mathbf{x})\|_2 + \lambda_{\text{TV}} \cdot \text{TV}_p(\mathbf{z}).$$

$$\text{TV}_p(\mathbf{z}) = \sum_{k=1}^K \left[\sum_{i=2}^N \|\mathbf{z}(i, :, k) - \mathbf{z}(i-1, :, k)\|_p + \sum_{j=2}^N \|\mathbf{z}(:, j, k) - \mathbf{z}(:, j-1, k)\|_p \right]$$

Original



Noisy image



Denoised image



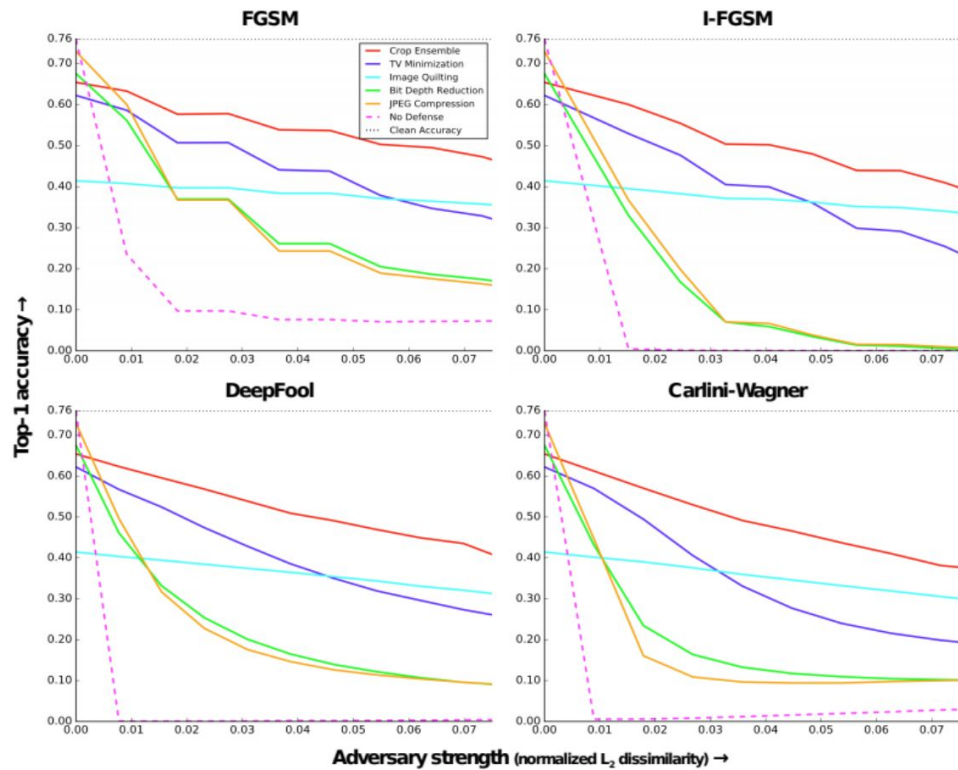
Image Quilting

- Images pieced together from small patches taken from database
- Non-parametric technique
- Database only has clean patches
- Patches are placed over predefined points and edges are smoothed
- Patches selected using KNN on database
- Resulting image does not have any perturbations

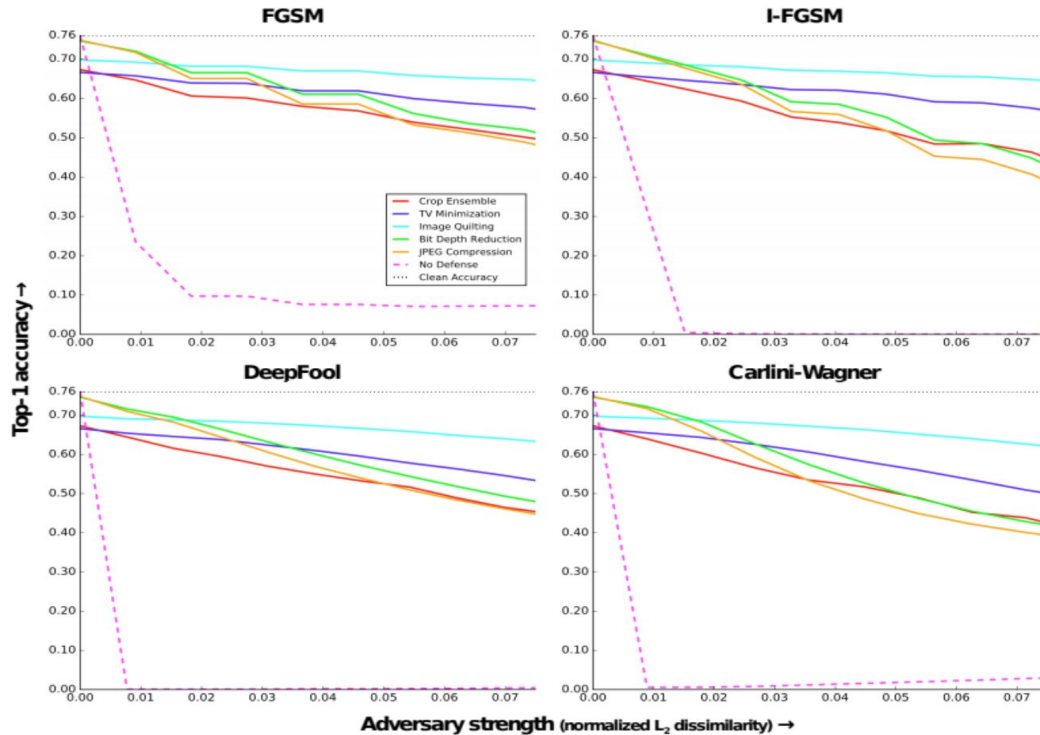
Experiments

- Performed in black box and gray box setting
- Performed on Imagenet dataset
- ResNet-50 model was attacked
- Attacks included FGSM, I-FGSM, Deepfool and Carlini and Wagner
- Top1 classification accuracy was reported for varying normalized L2 dissimilarities

Gray Box: Image Transformations at Test Time



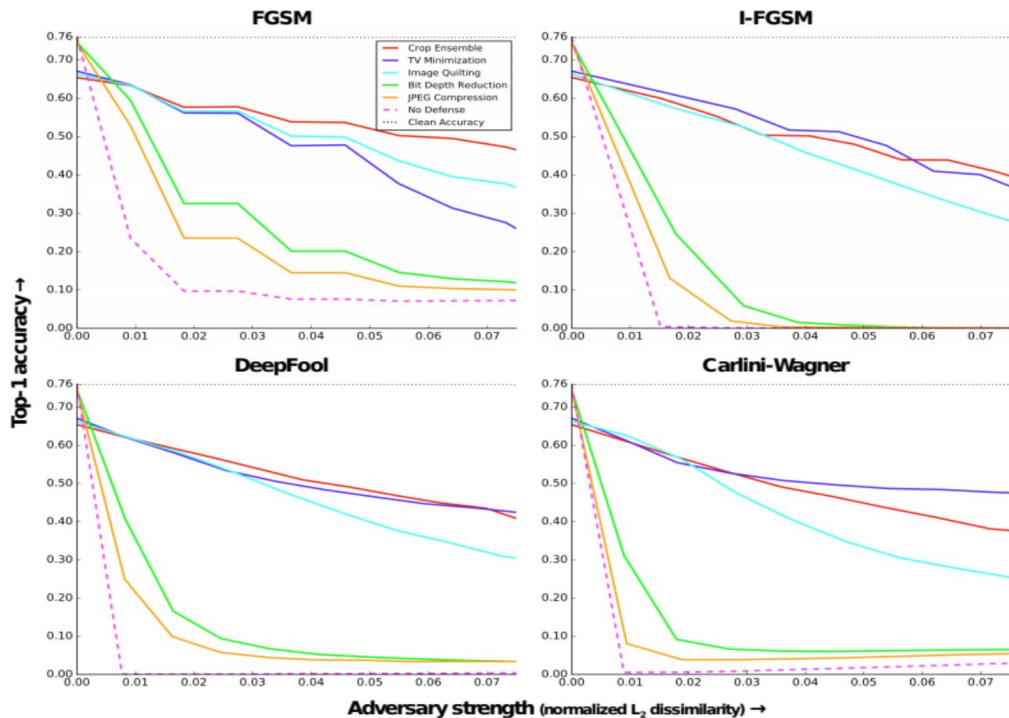
Black Box: Image transformations at test time



Black Box: Ensembling and model transfer

	Quilting				TVM + Quilting				Cropping + TVM + Quilting			
	RN50	RN101	DN169	Iv4	RN50	RN101	DN169	Iv4	RN50	RN101	DN169	Iv4
No Attack	70.07	72.56	70.18	73.01	72.38	74.74	73.10	75.55	72.14	74.53	72.92	75.10
FGSM	65.45	68.50	65.96	67.53	65.70	68.77	67.09	69.19	66.65	69.75	67.86	70.37
I-FGSM	65.59	68.72	66.16	69.29	65.84	69.10	67.32	71.05	67.03	70.14	68.20	71.52
DeepFool	65.20	68.73	65.86	68.70	65.80	69.34	67.40	71.03	67.11	70.49	68.62	71.47
CW-L2	64.11	67.72	65.00	68.14	63.99	68.20	66.08	70.13	65.31	69.14	66.96	70.50

Gray Box: Image transformations at test time



Comparison to Ensemble Adversarial Training

	Cropping	TVM	Quilting	Ensemble Training (Tramèr et al., 2017)
No Attack	65.41	66.29	69.66	80.3
FGSM	49.52	31.37	39.55	69.15
I-FGSM	43.89	40.99	33.22	5.07
DeepFool	44.92	44.69	34.54	1.84
CW-L2	41.06	48.41	30.51	22.23

Our experiment: Running TVM with an ATN

- We built an ATN that broke an ANN from error rate 0.34 to 0.91
- Error rate went from 0.91 to 0.90 when ATN perturbation was transformed using TVM
- Dataset consisted of 20000 non-MNIST images

Conclusions and Questions

- Image transformations are a more generic defense
- Benefits from randomization
- Benefits from non-differentiability
- Why were they not successful against ATN attack?
- Like ATNs, can the best transformation be “learned”?
- How good are these transformations in complete white box settings?

Merci Beaucoup