The Odds are Odd: A Statistical Test for Detecting Adversarial Examples

Kevin Roth, Yannic Kilcher, Thomas Hoffmann (ETH Zürich)

Meriel Stein

December 6, 2019

Background

Deep neural networks are used on a variety of classification problems (image classification, facial recognition) very effectively but are not robust to adversarial examples

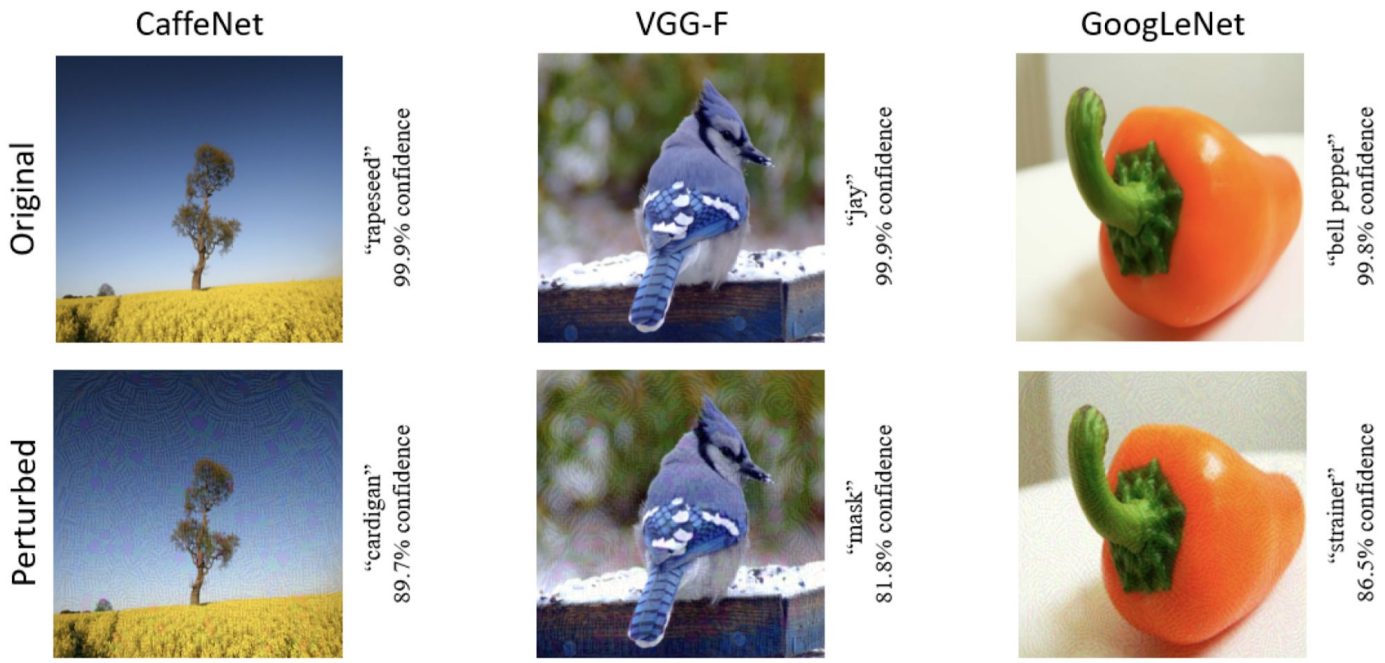
Small changes in input signal can lead to large changes in neural net output

Shape and robustness of perturbed log-odds statistics are different if $x = x^*$ versus $x = x^* + \Delta x$

Motivation

Robustification is usually limited to including adversarial examples in training data

Typical methods of defending neural networks against adversarial attacks not effective on well-designed attacks



Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. CoRR, abs/1801.00553, 2018.

Related Work

- Iterative Adversarial Attacks
  - **Madry et al., 2017**; **Kurakin et al., 2016** -- Projected Gradient Descent AKA Basic Iterative Method. Iconic iterative adversarial attack:

$$x^0 \sim \mathcal{U}(\mathcal{B}_\epsilon^p(x)) \tag{1}$$

$$x^{t+1} = \Pi_{\mathcal{B}_\epsilon^\infty(x)}\left(x^t - \alpha\,\mathrm{sign}(\nabla_x \mathcal{L}(f;x,y)|_{x^t})\right) \quad [L^\infty]$$

$$x^{t+1} = \Pi_{\mathcal{B}_\epsilon^2(x)}\left(x^t - \alpha \frac{\nabla_x \mathcal{L}(f;x,y)|_{x^t}}{||\,\nabla_x \mathcal{L}(f;x,y)|_{x^t}\,||_2}\right) \quad [L^2]$$

  - **Carlini & Wagner**, 2017b
- Detection
  - **Grosse et al., 2017** -- statistical tests can detect adversarial examples because adversarial examples come from a dissimilar distribution than the natural data does
  - **Metzen et al., 2017** -- add “detector” classification subnetwork that uses intermediate feature representations to distinguish between natural/adversarial activations.
  - **Feinman et al., 2017** -- test whether inputs lie in low-confidence areas of model.
  - **Xu et al., 2017** -- compare model predictions on “natural” input versus feature-squeezed version of that input, diff results and compare to a chosen threshold.
  - Also mentioned: Song et al., 2017; Li & Li, 2017; Lu et al., 2017; Carlini & Wagner, 2017a
- Origin of adversarial examples
  - **Gilmer et al. 2018** -- due to flaws in model and learning objective
  - **Schmidt et al, 2018** -- due to generalization error higher than zero
  - **Fawzi et al., 2018** -- due to high-dimensional statistics

Target Task

Networks can recover from adversarial input perturbations that force misclassifications $x=x^*+\Delta x$ by adding noise s.t. $Pr\{F(x+\eta)=y^*\}$ is "sufficiently" large and grouping together types of random adversarial transformations

Goals:

1. Instead of trying to recover from adversarial perturbation, try to detect it statistically through probabilistic classification
2. Accomplish this with a probabilistic classifier using a parameterized logit layer of scores that leverages the fact that perturbations are not robust.

Adversarial Perturbation with Noise

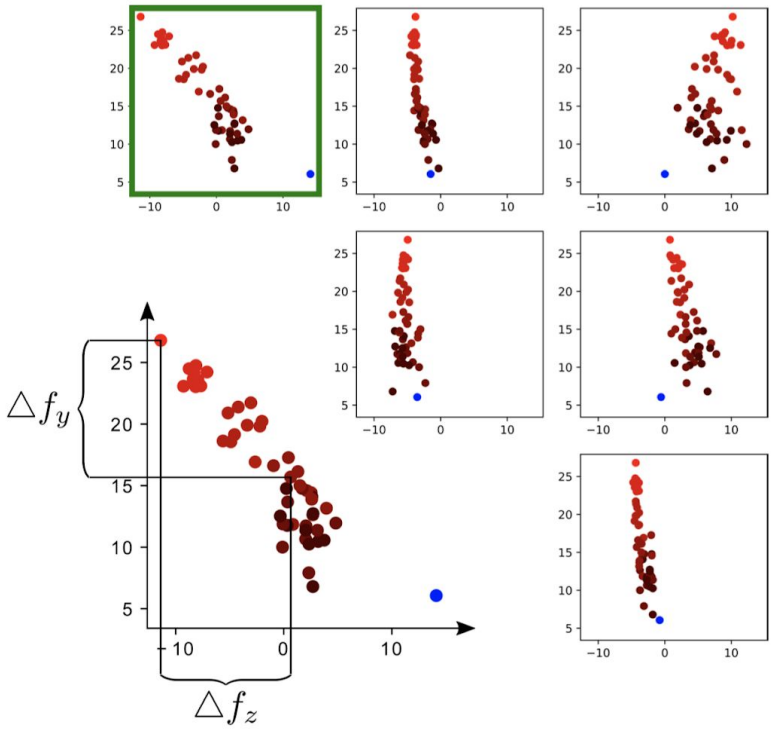


Figure 1. Change of logit scores f_y (on the vertical axis) and f_z (on the horizontal axis) when adding noise to an adversarially perturbed example $x = x^* + \triangle x$. Light red dot: x. Other red dots: $x + \eta$, with color coding of noise amplitude (light = small, dark = large). Light blue dot: x^*. Different plots correspond to different candidate classes z. The candidate class in the green box is selected by Equation 6 and the plot magnified in the lower left.

Proposed Solution

Parameterize logits: $f_y(x) = \langle w_y, \Phi(x) \rangle$

- w_y: weight vectors
- $\Phi(x)$: feature map from a trained network

Add noise to get perturbed log-odds: $f_{y,z}(x + \eta)$

- $\eta \sim N$
- $y = y^*$ during training or $y = F(x)$ during testing

Find suspected perturbation $F(x^*+\Delta x) = y \neq y^*$:

Find expected perturbed log-odds: $\bar{g}_{y,z}(x) = \mathbf{E}_\eta \left[\bar{g}_{y,z}(x, \eta) \right]$

Flag as possibly adversarial if: $\max_{z \neq y} \{ \bar{g}_{y,z}(x) - \tau_{y,z} \} \geqslant 0$

Define new classifier for test-time attacks:

$$\begin{cases} G(x) = \arg\max_z \{ \bar{g}_{y,z}(x) - \tau_{y,z} \} \\ y := F(x) \end{cases}$$

Implementation

1. Attack all examples provided to pre-trained deep neural networks

    a. Attack strategy: L∞-norm constrained Projected Gradient Descent white-box attack

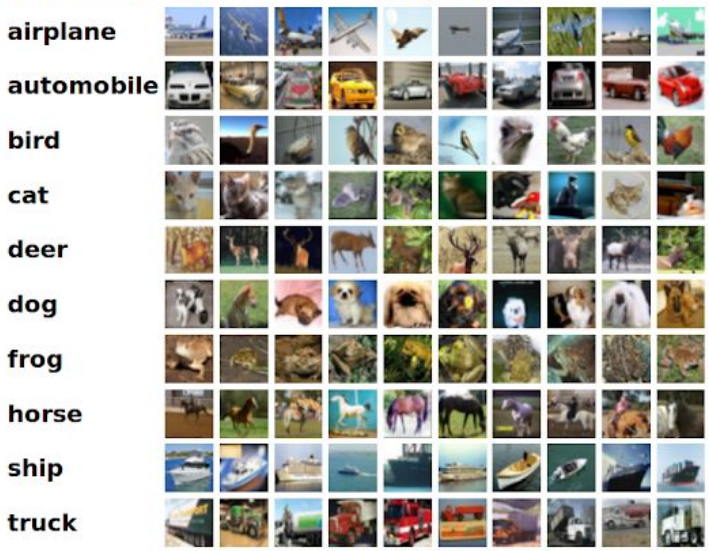
2. Compare the norm of the induced feature space perturbation $||\Delta\Phi||_2$ along adversarial directions and random directions and alignment of feature space and select weight vectors to characterize shift in feature representation

3. Compare distance to decision boundary for perturbed vs. natural examples to characterize classification output differences

Data Summary

CIFAR10

- Images: 60,000
- Classes: 10 (6000 images per class)
- 50,000 training and 10,000 test images.



ImageNet

- Non-empty synsets: 21,841
- Images: 14,197,122
- Images with bounding box annotations: 1,034,908

Experimental Results

Adversarial examples not necessarily detectable due to distance to decision boundary

Adversarial examples present in "cones" in feature space, surrounded by natural class

Softmax predictions of $x^*+\Delta x+\eta$ show that adding noise to adversarial example does not necessarily recover natural class

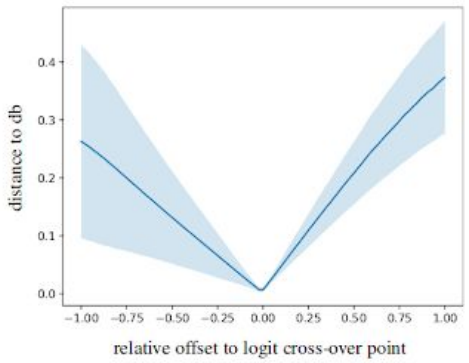


Figure 1. Average distance to the decision boundary

Figure 2. Adversarial cone

Experimental Analysis

Correction method able to cope with stronger attacks

Twice as effective as state-of-the-art adversarial training strategy

Comparable accuracy to feature squeezing detection

Significantly higher accuracy than dropout randomization detection

Table 2. Detection rates of our statistical test.

DATASET	MODEL	DETECTION RATE (CLEAN / PGD)
CIFAR10	WRESNET	0.2% / 99.1%
	CNN7	0.8% / 95.0%
	CNN4	1.4% / 93.8%
IMAGENET	INCEPTION V3	1.9% / 99.6%
	RESNET 101	0.8% / 99.8%
	RESNET 18	0.6% / 99.8%
	VGG11(+BN)	0.5% / 99.9%
	VGG16(+BN)	0.3% / 99.9%

Table 3. Accuracies of our correction method.

DATASET	MODEL	ACCURACY (CLEAN / PGD)
CIFAR10	WRESNET	96.0% / 92.7%
	CNN7	93.6% / 89.5%
	CNN4	71.0% / 67.6%

Table 4. Test set accuracies for adversarially trained models.

DATASET	ADVERSARIALLY TRAINED MODEL	ACCURACY (CLEAN / PGD)
CIFAR10	WRESNET	87.3% / 55.2%
	CNN7	82.2% / 44.4%
	CNN4	68.2% / 40.4%

Results Reproduction

ResNet models
- Clean
- Robustified

CIFAR10 Dataset
- 60,000 train; 10,000 test

Detection ratio of samples with valid clean classifications and invalid attacked classifications: 0.76

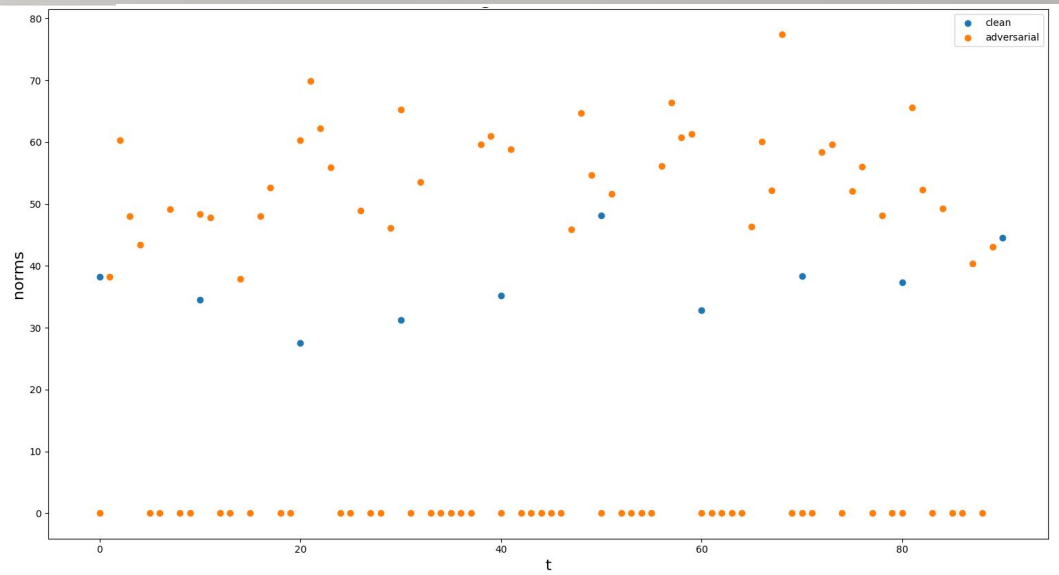
		n18		n24		n30	
		clean	robust	clean	robust	clean	robust
Legitimate examples	**without correction**	0.9600	0.8800	0.8800	0.8800	0.9600	0.8800
	with correction	0.9400	n/a	0.8300	n/a	0.9500	n/a
Adversarial examples	**without correction**	0.0300	0.5100	0.0300	0.5100	0.0300	0.5300
	with correction	0.1800	n/a	0.1700	n/a	0.2400	n/a

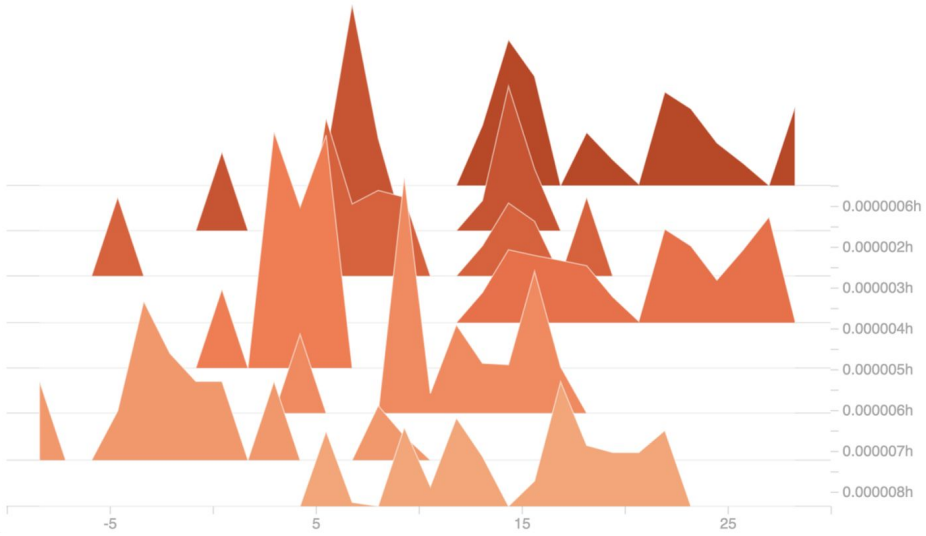
Results Reproduction

Alignments

- Computed by $\langle \Delta\Phi, \Delta\mathfrak{w} \rangle$
- Larger in magnitude and greater in number for adversarial examples

Reinforces implication that adversarial examples *cause atypically large feature space perturbations*





Conclusion and Future Work

Conclusions

- Adversarial perturbation changes the shape of the feature space that input vectors are projected onto.
- Adversarial perturbations of varying strength can be detected and corrected for through log-odds analysis with high accuracy.

Future Work

- Implement with L^2 and re-compare results against feature squeezing
- Research network architecture to better understand underlying properties that enable the success of this method
- Determine if this method generalizes to all models

References

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. InProceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 3–14. ACM, 2017a.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE, 2017b.

Fawzi, A., Fawzi, H., and Fawzi, O. Adversarial vulnerability for any classifier. arXiv preprint arXiv:1802.08686,2018.

Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B.Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410, 2017.

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu,M., Wattenberg, M., and Goodfellow, I. Adversarial spheres.arXiv preprint arXiv:1801.02774, 2018.

Grosse, K., Manoharan, P., Papernot, N., Backes, M., and McDaniel, P. On the (statistical) detection of adversarial examples.arXiv preprint arXiv:1702.06280, 2017.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world.arXiv preprint arXiv:1607.02533, 2016.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B.On detecting adversarial perturbations.arXiv preprint arXiv:1702.04267, 2017.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., andMadry, A. Adversarially robust generalization requiresmore data.arXiv preprint arXiv:1804.11285, 2018.

Xu, W., Evans, D., and Qi, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. arXiv preprint arXiv:1704.01155, 2017.