M2177.0043 Introduction to Deep Learning

Lecture 16: Defense against adversarial attacks

Hyun Oh Song[1]

[1]Dept. of Computer Science and Engineering, Seoul National University

May 14, 2020

Last time

- Adversarial attacks
- White-box adversarial attacks
- Black-box adversarial attacks
- One-pixel attack

Outline

Defense against adversarial attacks

Adversarial Training - Goodfellow

- Train with an adversarial objective function
- "Robust" training loss

$$\tilde{J}(x, y, \theta) = \alpha J(x, y, \theta) + (1 - \alpha) J(x + \epsilon \text{sign}(\nabla_x J(x, y, \theta)), y, \theta)$$

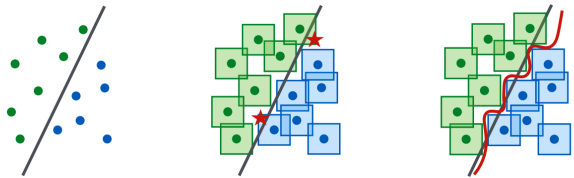


Figure: A conceptual illustration of "natural" vs "adversarial" decision boundaries

Ian Goodfellow and Jonathon Shlens and Christian Szegedy, *Explaining and Harnessing Adversarial Examples, 2015*

Adversarial training - Madry

- Perform adversarial training (PGD attacks) during training. Concretely minimize the following loss function

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in B(\epsilon)} \ell(\theta, x+\delta, y) \right]$$

- The inner maximization

$$\max_{\delta \in B(\epsilon)} \ell(\theta, x+\delta, y)$$

is typically performed via PGD which performs the following gradient step

$$\delta \leftarrow \prod_{B(\epsilon)} (\delta + \eta \nabla_{\delta} \ell(x+\delta, y, \theta))$$

Madry et al. *Towards deep learning models resistant to adversarial attacks, 2017*

- **Pros**: This approach is simple, intuitive, and shows good adversarial robustness. This is one of the mostly used adversarial defense method.

- **Cons**:
  1. Adversarial training requires many rounds of PGD attacks per each mini-batch. This can slow down the network training quite significantly.
  2. Tends to overfit to the training set compared to vanilla training and the clean accuracy on previously unseen test set drops significantly

- How to improve adversarial training in both the training speed and in the generalization performance is an active research area.

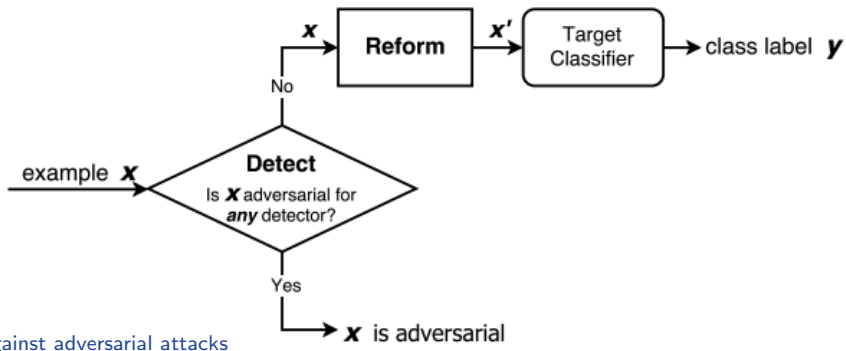
Madry et al. *Towards deep learning models resistant to adversarial attacks, 2017*

Magnet - Preprocessor based defense

- MagNet neither modifies the classifier nor requires knowledge of the process for generating adversarial examples.
- Magnet consists of a detector network and a reformer network.

Dongyu Meng and Hao Chen, *MagNet: a Two-Pronged Defense against Adversarial Examples, 2017*

- The detector network learns to differentiate between normal unperturbed examples versus adversarially perturbed examples.
- The reformer moves adversarial examples towards the manifold of normal examples which is effective for correctly classifying adversarial examples with small perturbation.



Magnet: Detector

- Train an autoencoder to minimize a mse loss function over the training set:

$$L(X_{train}) = \frac{1}{|X_{train}|} \sum_{x \in X_{train}} \|x - ae(x)\|_2$$

For test time, check the reconstruction error on a test example x and threshold

If $E(x) = ||x - ae(x)||_p \geqslant \tau$ "perturbed" Else "normal"

- Alternatively, check the probability divergence on a test example x: The divergence between $f(x)$ and $f(ae(x))$

$$JSD(P||Q) = \frac{1}{2}(D_{KL}(P||M) + D_{KL}(Q||M)),$$

where

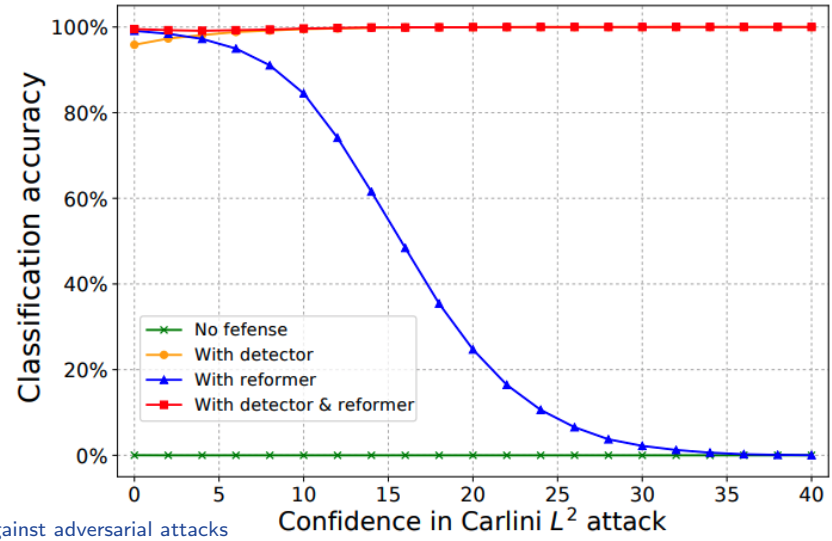
$$D_{KL}(P||Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}, \; M = \frac{1}{2}(P + Q)$$

- Detector rejects "unusual" test samples claiming it's been perturbed

Magnet: Reformer

- The reformer tries to reconstruct the test input. The output of the reformer is fed to the target classifier.
- The ideal reformer:
  - should not change the classification results of normal examples.
  - should change adversarial examples adequately so that the reconstructed examples are close to normal examples.
- Use the autoencoder that used to train for detector as reformer. The autoencoder is expected to output an example that approximates the adversarial example and that is closer to the manifold of the normal examples.

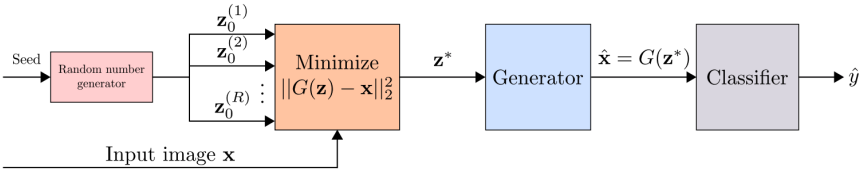
Magnet Result



Defense-GAN

Defense-GAN is a framework to defend adversarial attacks using a generative model. Given :

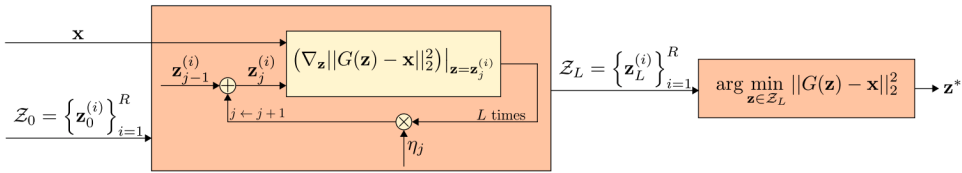
- Generator G of trained on training samples (un-perturbed)
- Image x to be classified
- Classifier



$$z^* = \underset{z}{\text{argmin}} ||G(z) - x||_2^2$$

Pouya Samangouei and Maya Kabkab and Rama Chellappa, *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models, 2018*

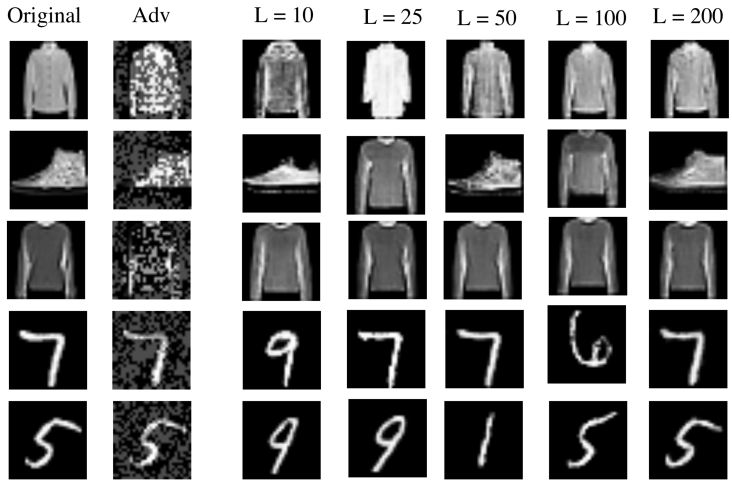
Defense-GAN



- Minimize the reconstruction error $||G(z) - x||_2^2$, using L steps of gradient descent and R random re-initializations.

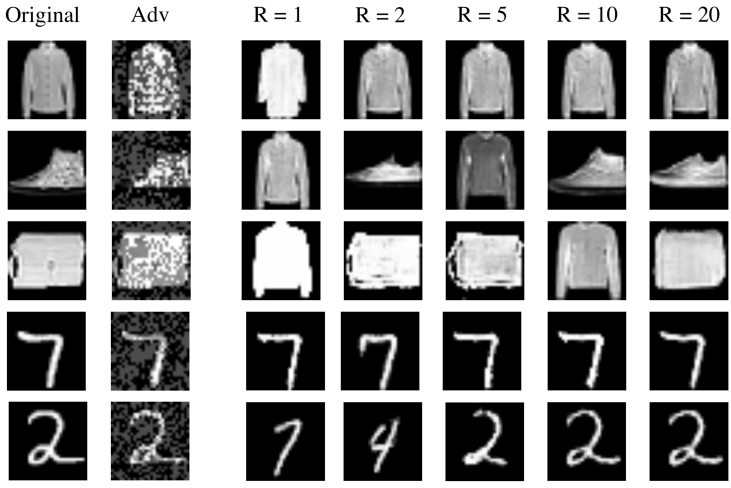
Defense-GAN: Effect of L

FGSM adversarial ϵ=0.3, R = 1



Defense-GAN: Effect of R

FGSM adversarial ϵ=0.3, L = 25



Local linearization regularization

- Consider *local linearity measure* to find the point δ in $B(\epsilon)$ where the linear approximation $\ell(x) + \delta^\intercal \nabla_x \ell(x)$ is maximally violated,

$$\gamma(\epsilon, x) = \max_{\delta \in B(\epsilon)} |\ell(x+\delta) - \ell(x) - \delta^\intercal \nabla_x \ell(x)|$$

Proposition 1

Consider a loss function $\ell(x)$ that is once-differentiable, and a local neighborhood defined by $B(\epsilon)$. Then for all $\delta \in B(\epsilon)$

$$|\ell(x+\delta) - \ell(x)| \leqslant |\delta^\intercal \nabla_x \ell(x)| + \gamma(\epsilon, x) \tag{1}$$

- From Equation (1), it is clear that $\ell(x+\delta) \to \ell(x)$, as both $|\delta^\intercal \nabla_x \ell(x)| \to 0$ and $\gamma(\epsilon; x) \to 0$ for all $\delta \in B(\epsilon)$. And assuming $\ell(x+\delta) \geqslant \ell(x)$ one also has the upper bound $\ell(x+\delta) \leqslant \ell(x) + |\delta^\intercal \nabla_x \ell(x)| + \gamma(\epsilon, x)$.

- LLR propose the following objective for adversarial robust training,

$$\min_{\theta} \mathbb{E}_{\mathcal{D}} \left[\ell(x) + \underbrace{\lambda\gamma(\epsilon, x) + \mu|\delta_{LLR}^{\intercal} \nabla_x \ell(x)|}_{\text{LLR}} \right],$$

where δ_{LLR} is the point in $B(\epsilon)$ where the linear approximation $\ell(x) + \delta^{\intercal}\nabla_x \ell(x)$ is maximally violated. Concretely,

$$\begin{aligned} \delta_{LLR} &= \underset{\delta \in B(\epsilon)}{\operatorname{argmax}} \ \gamma(\epsilon, x) \\ &= \underset{\delta \in B(\epsilon)}{\operatorname{argmax}} \ |\ell(x+\delta) - \ell(x) - \delta^{\intercal}\nabla_x \ell(x)| \end{aligned}$$

- Quantitative results on ImageNet dataset

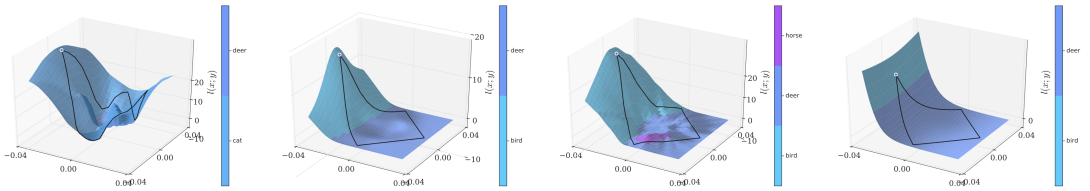
Methods	PGD steps	Nominal Accuracy	Untargeted Accuracy	Random-Targeted Success Rate
ImageNet: ResNet-152 (4/255)				
ADV	30	69.20%	39.70%	0.50%
DENOISE	30	69.70%	38.90%	**0.40%**
LLR	**2**	**72.70%**	**47.00%**	**0.40%**
ImageNet: ResNet-152 (16/255)				
ADV [28]	30	64.10%	6.30%	40.00%
DENOISE [28]	30	**66.80%**	**7.50%**	**38.00%**
LLR	**10**	51.20%	6.10%	43.80%

Table 3: LLR gets 47% adversarial accuracy for 4/255 – 7.30% higher than DENOISE and ADV. For 16/255, LLR gets similar robustness results, but it comes at a significant cost to the nominal accuracy. Note Multi-Targeted attacks for ImageNet requires looping over 1000 labels, this evaluation can take up to several days even on 50 GPUs thus is omitted from this table. The column of the strongest attack is highlighted.

Note, ADV baseline is the adversarial training from Madry et al.

Qin et al., *Adversarial robustness through local linearization, 2019*

- $\ell(x)$ projection onto 2D plane, where one direction is the adversarial perturbation while the other is random.



(a) ADV-1 (b) LLR-1 (c) ADV-2 (d) LLR-2

Figure 4: Comparing the loss surface, $\ell(x)$, after we train using just 1 or 2 steps of PGD for the inner maximization of either the adversarial objective (ADV) $\max_{\delta \in B(\epsilon)} \ell(x+\delta)$ or the linearity objective (LLR) $\gamma(\epsilon, x) = \max_{\delta \in B(\epsilon)} \left|\ell(x+\delta) - \ell(x) - \delta^T \nabla \ell(x)\right|$. Results are shown for image 126 in test set of CIFAR-10, the nominal label is deer. ADV-i refers to adversarial training with i PGD steps, similarly with LLR-i.

Note, ADV baseline is the adversarial training from Madry et al.

Qin et al., *Adversarial robustness through local linearization, 2019*

Statistical testing: The odds are odd

- Exploit certain anomalies that adversarial attacks introduce
- Define $f(x)$ as the normalized logit for a given input x and let $F(x) = \text{argmax}_y f_y(x)$. Let (x^*, y^*) be the normal data and $x = x^* + \Delta x$ as its adversarially perturbed counterpart such that $F(x) \neq y^* = F(x^*)$.
- Also define pairwise log-odds between classes y and z given input x,

$$f_{y,z}(x) = f_z(x) - f_y(x)$$

- Log-odds behave differently against noise perturbation ($f_{y,z}(x + \eta)$ with $\eta \sim \mathbb{N}$) whether the input is normal or adv.

Roth et al., *The odds are odd: A statistical test for detection adversarial examples, 2019*

- (Normal input) It is common to use small noise during normal training (*i.e.* data augmentation) as a way to robustify the models or to use regularization techniques (*i.e.* dropout) which improve model generalization. Intuition: log-odds with regard to the true class remain stable under noise. $f_{y^*,z}(x^* + \eta) \approx f_{y^*,z}(x^*)$

- (Adversarially perturbed) Posit the conjecture that common attacks find perturbations Δx that are not *robust* but that overfit to specifics of x. If the adversarial perturbation is not robust w.r.t. the noise process, then this will yield $f_{y,y^*}(x + \eta) > f_{y,y^*}(x)$, meaning that noise will partially undo the effect of the adversarial manipulation.

Roth et al., *The odds are odd: A statistical test for detection adversarial examples, 2019*

- The log-odds may behave differently for different class pairs, as they reflect class confusion probabilities that are task-specific. This can be addressed by performing a Z-score standardization across data points x and perturbations η. For each fixed class pair (y, z) define:

$$
\begin{aligned}
g_{y,z}(x,\eta) &= f_{y,z}(x+\eta) - f_{y,z}(x) \\
\mu_{y^*,z} &= \mathbb{E}_{x^*|y^*}\mathbb{E}_{\eta}\left[g_{y^*,z}(x^*,\eta)\right] \\
\sigma^2_{y^*,z} &= \mathbb{E}_{x^*|y^*}\mathbb{E}_{\eta}\left[(g_{y^*,z}(x^*,\eta) - \mu_{y^*,z})^2\right] \\
\bar{g}_{y,z}(x,\eta) &= (g_{y,z}(x,\eta) - \mu_{y,z})/\sigma_{y,z} \\
\bar{g}_{y,z}(x) &= \mathbb{E}_{\eta}\left[\bar{g}_{y,z}(x,\eta)\right]
\end{aligned}
$$

Roth et al., *The odds are odd: A statistical test for detection adversarial examples, 2019*

- Flag an example $(x, y := F(x))$ as manipulated, if

$$\max_{z \neq y} \bar{g}_{y,z}(x) - \tau_{y,z} \geqslant 0,$$

  otherwise it is considered clean.

- Also, one could *correct* the label prediction by defining a new classifier G via

$$G(x) = \operatorname*{argmax}_{z} \bar{g}_{y,z}(x) - \tau_{y,z}, \;\; y := F(x)$$

Roth et al., *The odds are odd: A statistical test for detection adversarial examples, 2019*

- Experiment on CIFAR10 showing the histograms of standardized log-odds $\bar{g}_{y,z}(x)$. Shows a good separation between clean data x^* and manipulated data $x = x^* + \Delta x$.

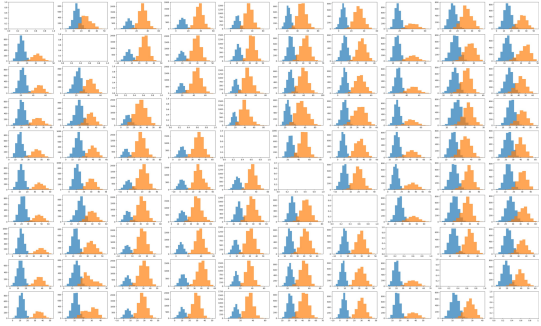


Figure 2. Histograms of the test statistic $\bar{g}_{y,z}(x)$ aggregated over all data points in the training set. Blue represents natural data, orange represents adversarially perturbed data. Columns correspond to predicted labels y, rows to candidate classes z.

Roth et al., *The odds are odd: A statistical test for detection adversarial examples, 2019*

- Quantitative results. Detection rates and accuracies of the corrected predictions.

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%

Roth et al., *The odds are odd: A statistical test for detection adversarial examples*, 2019

- A recent paper by Hosseini et al. 2019 titled "Are Odds Really Odd? Bypassing Statistical Detection of Adversarial Examples" reports a revised attack which can completely fool the statistical detector.
- In general, in white-box setting, devising a viable adversarial defense method is much more difficult than devising new attacks