M2177.0043 Introduction to Deep Learning

Lecture 17: Defense against adversarial attacks

Hyun Oh Song[1]

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

May 11, 2023

Last time

- Adversarial attacks
- White-box adversarial attacks
- Black-box adversarial attacks

Outline

MagNet

- 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.

Meng and Chen, *MagNet: a Two-Pronged Defense against Adversarial Examples*, CCS 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.

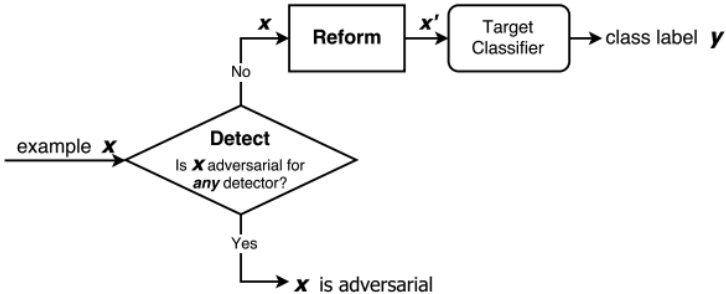


Figure adapted from Meng and Chen, *MagNet: a Two-Pronged Defense against Adversarial Examples*, CCS 2017.

MagNet: detector

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

$$L(\mathcal{X}_{\text{train}}) = \frac{1}{|\mathcal{X}_{\text{train}}|} \sum_{x \in \mathcal{X}_{\text{train}}} \|x - ae(x)\|_2.$$

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

$$\text{If } D(x) = \|x - ae(x)\|_2 \gtrless \tau \text{ "perturbed" else "normal".}$$

- Alternatively, check the Jansen-Shannon divergence between the softmax probabilities (with temperature T) of x and $ae(x)$:

$$D(x) = \text{JSD}(p \parallel q), \; p_i = \frac{\exp(Z(x)_i/T)}{\sum_j \exp(Z(x)_j/T)}, \; q_i = \frac{\exp(Z(ae(x))_i/T)}{\sum_j \exp(Z(ae(x))_j/T)},$$

  where $Z(x)$ is the logit of x.

- 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 is used to train for detector as the 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 results on MNIST

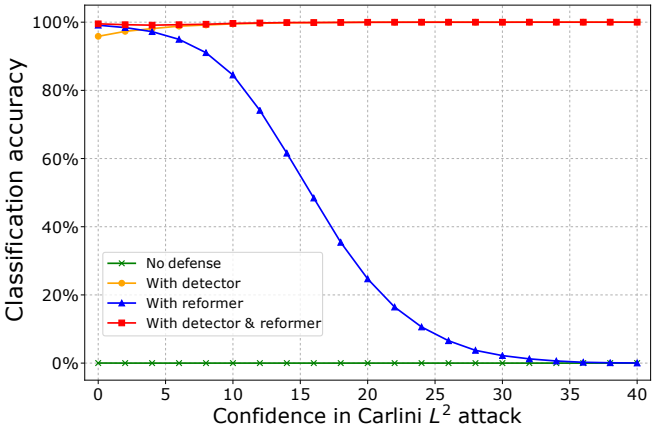
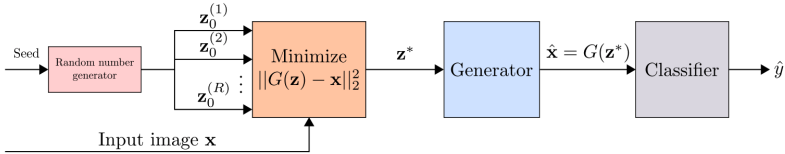


Figure adapted from Meng and Chen, *MagNet: a Two-Pronged Defense against Adversarial Examples*, CCS 2017.

Defense-GAN

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

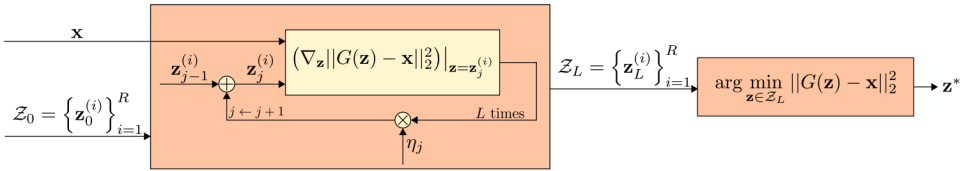
- Generator G of trained on unperturbed training samples
- Image x to be classified
- Classifier



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

Samangouei et al., *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models*, ICLR 2018.

Defense-GAN



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

Figure adapted from Samangouei et al., *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models*, ICLR 2018.

Defense-GAN: effect of L

FGSM adversarial examples ($\epsilon = 0.3, R = 1$)

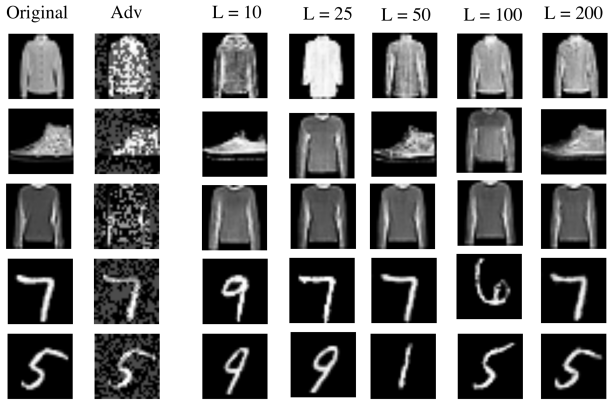


Figure adapted from Samangouei et al., *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models*, ICLR 2018.

Defense-GAN: effect of R

FGSM adversarial examples ($\epsilon = 0.3, L = 25$)

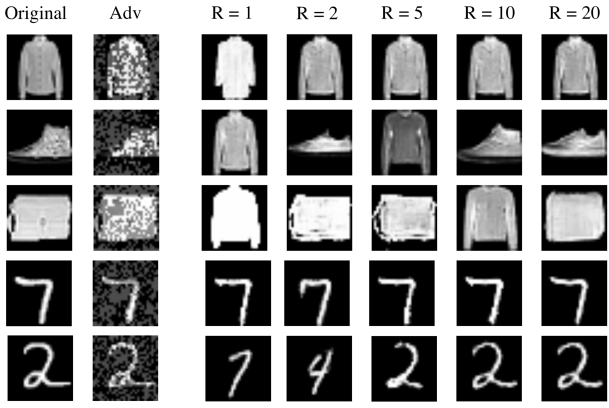


Figure adapted from Samangouei et al., *Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models*, ICLR 2018.

Breaking preprocessing-based defenses

- In fact, many preprocessing-based defenses have been shown vulnerable to adversarial attacks that are adaptively designed.

- MagNet: broken by an adaptive CW attack, whose objective is given by

$$\begin{aligned}
&\underset{x'}{\text{minimize}} \ \|x' - x\|_2^2 + c \cdot L_c(x') + d \cdot L_d(x') \\
&L_c(x') = \max(\max_{i \neq t} Z(R(x'))_i - Z(R(x'))_t, -\kappa) \\
&L_d(x') = \max(D(x') - \tau, 0),
\end{aligned}$$

  where D and R is the detector and reformer used in MagNet, respectively.

- This adaptive attack generates adversarial examples that bypass the detector and fool the classifier after reconstruction.

Carlini and Wagner, *MagNet and "Efficient Defenses Against Adversarial Attacks" are Not Robust to Adversarial Examples*, 2017.

- Defense-GAN: want to find x' such that x' is sufficiently close to x and $G(z^*(x'))$ is misclassified.

- One obvious way is to solve the following optimization problem:

$$\underset{x'}{\text{minimize}} \ \|x' - x\|_2^2 + c \cdot L_c(G(z^*(x'))), \tag{1}$$

  where L_c is the loss used in CW attack.

- However, since the operation

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

  is non-differentiable, we cannot compute the exact gradient of Equation (1) using back-propagation.

Athalye et al., *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*, ICML 2018.

- Observation: $G(z^*(x')) \approx x'$ by the definition of z^*.
- Workaround: perform the forward pass through $G(z^*(\cdot))$, but on the backward pass, replace $G(z^*(\cdot))$ to the identity function:

$$\begin{aligned}\left[\frac{d}{dx} L_c(G(z^*(x)))\right]_{x=x'} &= \left[\frac{d}{dx} L_c(x)\right]_{x=G(z^*(x'))} \left[\frac{d}{dx} G(z^*(x))\right]_{x=x'} \\ &\approx \left[\frac{d}{dx} L_c(x)\right]_{x=G(z^*(x'))} \left[\frac{d}{dx} x\right]_{x=x'} \\ &= \left[\frac{d}{dx} L_c(x)\right]_{x=G(z^*(x'))}.\end{aligned}$$

- This method is called **backward pass differential approximation** (BPDA) and received the best paper award at ICML 2018.
- Empirically, slightly inaccurate gradients still prove useful in constructing an adversarial example.

Outline

Adversarial training

- **Adversarial training** trains a robust classifier by augmenting each minibatch of training data with adversarial examples.
- It is generally regarded as one of the strongest empirical defense against adversarial attacks.

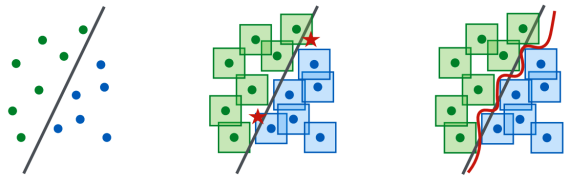


Figure: A conceptual illustration of "natural" vs "adversarial" decision boundaries.

Figure adapted from Madry et al., *Towards deep learning models resistant to adversarial attacks*, ICLR 2018.

FGSM adversarial training

- Trains a network on both natural and FGSM adversarial examples.
- "Robust" training objective:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\tilde{\ell}(x,y,\theta) \triangleq \alpha\ell(x,y,\theta) + (1-\alpha)\ell(\underbrace{x + \epsilon\ \mathrm{sign}(\nabla_x \ell(x,y,\theta))}_{\text{FGSM adversarial example}}, y, \theta) \right]$$

- While it is robust against one-step adversaries, it is completely vulnerable to more powerful iterative attacks such as PGD.

Goodfellow et al. *Explaining and Harnessing Adversarial Examples*, ICLR 2015.

PGD adversarial training

- Trains a network solely on adversarial examples generated via PGD. Concretely, the training objective is given by

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

- The inner maximization

$$\max_{\|\delta\|_\infty \leqslant \epsilon} \ell(x+\delta, y, \theta)$$

is typically solved via PGD which performs the following gradient step:

$$\delta \leftarrow \prod_{B_\epsilon(0)} \left(\delta + \eta \ \mathrm{sign}(\nabla_\delta \ell(x+\delta, y, \theta))\right).$$

Madry et al., *Towards deep learning models resistant to adversarial attacks*, ICLR 2018.

- While adversarial training is simple and intuitive, it still has many limitations that
  1. There exists a trade-off between the standard and robust accuracy. In other words, the standard accuracy of an adversarially trained network is lower than the standard network.
  2. It requires many rounds of PGD attacks per each mini-batch. This can slow down the network training quite significantly.
  3. It tends to overfit to the training set compared to vanilla training and the robust accuracy on a previously unseen test set drops significantly (large generalization gap).

Trade-off between robustness and accuracy

- Consider the following binary classification problem, where an input $x \in \mathbb{R}^{d+1}$ and its label $y \in \{-1, +1\}$ are sampled from

$$y \sim \mathcal{U}(\{-1, +1\}),$$
$$x_1 = \begin{cases} +y, & \text{with probability } \; p \\ -y, & \text{with probability } \; 1-p, \end{cases}$$
$$x_2, \ldots, x_{d+1} \overset{i.i.d}{\sim} \mathcal{N}(\eta y, 1).$$

- We assume $p = 0.95$ and $\eta = \Theta(1/\sqrt{d})$.

- The input consists of a single feature x_1 that is highly correlated with its label and d other features $x_2, \ldots, x_{d+1}$ that are weakly correlated with it.

Tsipras et al., *Robustness May be at Odds with Accuracy*, ICLR 2019.

- Each of $x_2, \ldots, x_{d+1}$ is hardly predictive of the correct label.
- However, the average of these features can be highly predictive.
- Consider the following linear classifier

$$f_{\mathrm{std}}(x) = \mathrm{sign}(w_{\mathrm{std}}^{\intercal} x), \;\; w_{\mathrm{std}} = \left[0, \frac{1}{d}, \ldots, \frac{1}{d}\right]. \tag{2}$$

Tsipras et al., *Robustness May be at Odds with Accuracy*, ICLR 2019.

- Consider the following linear classifier

$$f_{\text{std}}(x) = \text{sign}(w_{\text{std}}^\intercal x), \;\; w_{\text{std}} = \left[0, \frac{1}{d}, \ldots, \frac{1}{d}\right].$$

- It can achieve standard accuracy better than 99% when $\eta \geqslant 3/\sqrt{d}$:

$$\begin{aligned}
\mathbb{P}[f_{\text{std}}(x) = y] &= \mathbb{P}[\text{sign}(w_{\text{std}}^\intercal x) = y] \\
&= \mathbb{P}\Big[y \underbrace{\frac{1}{d}\sum_{i=1}^{d} \mathcal{N}(\eta y, 1)}_{\text{mean of Gaussian random variables}} > 0\Big] \\
&= \mathbb{P}\left[y\mathcal{N}(\eta y, \frac{1}{d}) > 0\right] \\
&= \mathbb{P}\left[\mathcal{N}(\eta, \frac{1}{d}) > 0\right] \qquad (y^2 = 1)
\end{aligned}$$

$$
\begin{aligned}
\mathbb{P}[f_{\text{std}}(x) = y] =& \mathbb{P}\left[\mathcal{N}(\eta, \frac{1}{d}) > 0\right] && (y^2 = 1) \\
\geqslant& \mathbb{P}\left[\mathcal{N}(\frac{3}{\sqrt{d}}, \frac{1}{d}) > 0\right] && (\eta \geqslant 3/\sqrt{d}) \\
=& \mathbb{P}\left[\mathcal{N}(0,1) > \frac{0 - 3/\sqrt{d}}{1/\sqrt{d}}\right] && \text{(standardization)} \\
=& \mathbb{P}\left[\mathcal{N}(0,1) > -3\right] > 0.99.
\end{aligned}
$$

- The standard classifier will take advantage of the weakly correlated features to achieve perfect standard accuracy better than $p = 0.95$.

Tsipras et al., *Robustness May be at Odds with Accuracy*, ICLR 2019.

Trade-off between robustness and accuracy

- However, this breaks in the ℓ_∞ adversarial setting.
- For example, with the perturbation budget $\epsilon = 2\eta$, an adversary can modify the weakly correlated features toward $-y$ by adding $-2\eta y$ for each feature:

$$x'_2, \ldots, x'_{d+1} \overset{i.i.d}{\sim} \mathcal{N}(\eta y, 1) - 2\eta y = \mathcal{N}(-\eta y, 1).$$

- Then, the classifier f_{std} in Equation (2) which relies solely on the weakly correlated features cannot have robust accuracy better than 1%.

Tsipras et al., *Robustness May be at Odds with Accuracy*, ICLR 2019.

- Concretely, given a perturbed input $x' = [x_1, x'_2, \ldots, x'_{d+1}]$, we have

$$\begin{aligned}
\mathbb{P}[f_{\text{std}}(x') = y] &= \mathbb{P}[\text{sign}(w_{\text{std}}^{\intercal} x') = y] \\
&= \mathbb{P}\Big[y\frac{1}{d}\sum_{i=1}^{d}\mathcal{N}(-\eta y, 1) > 0\Big] \\
&= \mathbb{P}\left[\mathcal{N}(-\eta, \frac{1}{d}) > 0\right] \\
&\leqslant \mathbb{P}\left[\mathcal{N}(-\frac{3}{\sqrt{d}}, \frac{1}{d}) > 0\right] && (\eta \geqslant 3/\sqrt{d}) \\
&= \mathbb{P}\left[\mathcal{N}(0,1) > \frac{0 + 3/\sqrt{d}}{1/\sqrt{d}}\right] && \text{(standardization)} \\
&= \mathbb{P}\left[\mathcal{N}(0,1) > 3\right] < 0.01.
\end{aligned}$$

Tsipras et al., *Robustness May be at Odds with Accuracy*, ICLR 2019.

Trade-off between robustness and accuracy

- Now, let's consider the following linear classifier that predicts labels only using the highly correlated feature x_1.

$$f_{\text{rob}}(x) = \text{sign}(w_{\text{rob}}^{\intercal} x), \;\; w_{\text{rob}} = [1, 0, \ldots, 0] \,.$$

- Clearly, it can achieve 95% standard accuracy ($p = 0.95$), which is less than f_{std}.

- However, if $\eta < 1/2$, then it is always robust against ℓ_∞ adversarial attacks with the perturbation budget $\epsilon = 2\eta$ since the sign of x_1 cannot be changed.

- Therefore, if d is sufficiently large and $3/\sqrt{d} \leqslant \eta < 1/2$, there exists a trade-off between robust and standard accuracy.

Tsipras et al., *Robustness May be at Odds with Accuracy*, ICLR 2019.

Free adversarial training

- Updates both network parameters and image perturbations using one **simultaneous backward pass**.
- Concretely, the network is trained on every intermediate adversarial example generated via PGD by repeating the following steps:
  1. Backward pass: compute $\nabla_\theta \ell(x+\delta, y, \theta)$ and $\nabla_\delta \ell(x+\delta, y, \theta)$ simultaneously.
  2. Parameter update: $\theta \leftarrow \theta - \beta \nabla_\theta \ell(x+\delta, y, \theta)$.
  3. Perturbation update: $\delta \leftarrow \prod_{B_\epsilon(0)} (\delta + \eta \ \mathrm{sign}(\nabla_\delta \ell(x+\delta, y, \theta)))$.
- Achieves comparable robustness to PGD adversarial training with almost no additional cost relative to vanilla training for the same number of weight updates.

Shafahi et al. *Adversarial Training for Free!*, NeurIPS 2019.

Fast adversarial training

- Empirically shows that FGSM adversarial training, combined with **random initialization**, is as effective as PGD adversarial training but has significantly lower computational cost.

$$\delta \sim \mathrm{Uniform}(-\epsilon, \epsilon) \qquad \text{(random initialization)}$$

$$\delta \leftarrow \prod_{B_\epsilon(0)} (\delta + \eta \ \mathrm{sign}(\nabla_\delta \ell(x + \delta, y, \theta))) \qquad \text{(FGSM step)}$$

$$\theta \leftarrow \theta - \beta \nabla_\theta \ell(x + \delta, y, \theta) \qquad \text{(parameter update)}$$

Wong et al. *Fast is better than free: Revisiting adversarial training*, ICLR 2020.

Overfitting in adversarial training

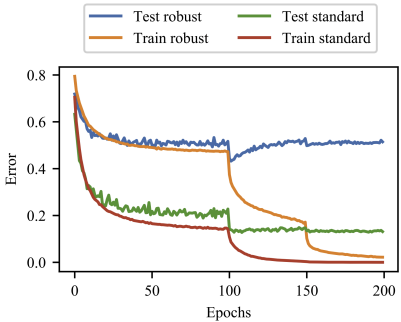


Figure: The learning curves for PGD adversarial training. The learning rate is decayed at 100 and 150 epochs.

- After a certain point, further training will continue to substantially decrease the robust training errors, while increasing the test errors.
- Simply using **early stopping** can significantly improve the robust performance.

Rice et al. *Overfitting in adversarially robust deep learning*, ICML 2020.

Adversarial training with additional unlabeled data

- Adversarial robustness can significantly benefit from augmenting training dataset with extra relevant **unlabeled** data.
- Obtain pseudo-labels for unlabeled data using a highly accurate standard classifier and feed them into adversarial training.
- Given labeled data $\mathcal{D} = (x_1, y_1, \ldots, x_n, y_n)$ and unlabeled data $\tilde{\mathcal{D}} = (\tilde{x}_1, \ldots, \tilde{x}_{\tilde{n}})$, the training algorithm is summarized by
  1. Learn a standard classifier f_{std} on $\mathcal{D}$.
  2. Generate pseudo-labels $\tilde{y}_i = f_{\mathrm{std}}(\tilde{x}_i)$ for $i = 1, \ldots, \tilde{n}$.
  3. Do adversarial training on $\mathcal{D} \cup \tilde{\mathcal{D}}$ to obtain a robust classifier f_{rob}.

Cameron et al. *Unlabeled Data Improves Adversarial Robustness*, NeurIPS 2020.

Outline

Certified defense

- Although adversarial training can significantly improve the empirical robustness of neural networks, there is no guarantee that a stronger, newly-discovered attack would not break them.
- A line of research has focused on providing **certified robustness** for neural networks that there are no possible adversarial attacks that could potentially break the models.

Randomized smoothing

- **Randomized smoothing** has been considered the most successful certified defense approach.
- It transforms any base classifier f into a new "smoothed classifier" g that returns the most probable prediction by f under Gaussian noise.
- Formally, given a classifier $f : \mathbb{R}^k \to \mathcal{Y}$ which maps images to class labels, the smoothed classifier $g : \mathbb{R}^k \to \mathcal{Y}$ is defined by

$$g(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \, \mathbb{P}[f(x + \eta) = y]$$
$$\text{where } \eta \sim \mathcal{N}(0, \sigma^2 I).$$

Cohen et al., *Certified Adversarial Robustness via Randomized Smoothing*, ICML 2019.

- Why is the smoothed classifier g robust?
- To see this, we first consider the soft classifier $F : \mathbb{R}^k \to \mathcal{P}(\mathcal{Y})$ that returns the probability vector and the soft smoothed classifier $G : \mathbb{R}^k \to \mathcal{P}(\mathcal{Y})$ which is defined by

$$G(x) = \mathbb{E}_{\eta \sim \mathcal{N}(0,\sigma^2 I)} \left[F(x + \eta) \right] = \left(F * \mathcal{N}(0, \sigma^2 I) \right)(x).$$

  Here $*$ denotes the convolution operator defined by

$$(f * g)(x) = \int_{\mathbb{R}^k} f(x - t) g(t) dt.$$

- G is also called the **Weierstrass transform** of F.

Lipschitz property of G

One remarkable property of the Weierstrass transform is that the transformed function, with an inverse CDF transform, is $1/\sigma$-Lipschitz.

Theorem

Let $\sigma > 0$, $F : \mathbb{R}^k \to [0, 1]$, and $G = F * \mathcal{N}(0, \sigma^2 I)$. Let Φ be the cdf of the standard normal distribution. Then, the function $\Phi^{-1} \circ G$ is $\frac{1}{\sigma}$-Lipschitz.

Proof (sketch)

Prove that the gradient of G (and $\Phi^{-1} \circ G$) is bounded $\to$ Lipschitz by the mean value theorem. Since the proof requires knowledge of real analysis and probability theory, we skip the proof.

Certified robustness of G

Due to the Lipschitz property of G, the output probability of G and its argmax do not change much with respect to the changes in the input.

Theorem

Let $\sigma > 0$, $F : \mathbb{R}^k \to \mathcal{P}(\mathcal{Y})$ be a soft classifier, and G be the smoothed soft classifier of F. Let

$$y_1 = \operatorname*{argmax}_{y \in \mathcal{Y}} G(x)_y, \;\; y_2 = \operatorname*{argmax}_{y \in \mathcal{Y} \setminus \{y_1\}} G(x)_y.$$

Then, for all $x' \in \mathbb{R}^k$ satisfying

$$\|x' - x\|_2 \leqslant \frac{\sigma}{2} \left(\Phi^{-1}(G(x)_{y_1}) - \Phi^{-1}(G(x)_{y_2}) \right),$$

we have $\operatorname{argmax}_y G(x')_y = \operatorname{argmax}_y G(x)_y = y_1$.

Proof

For any $y \in \mathcal{Y}$, $x \mapsto F(x)_y$ is a map from $\mathbb{R}^k$ to $[0,1]$. Therefore, $x \mapsto \Phi^{-1}(G(x)_y)$ is $\frac{1}{\sigma}$-Lipschitz. By the definition of Lipschitz continuity, we have

$$\begin{aligned}\Phi^{-1}(G(x)_{y_1}) - \Phi^{-1}(G(x')_{y_1}) &\leqslant |\Phi^{-1}(G(x')_{y_1}) - \Phi^{-1}(G(x)_{y_1})| \\ &\leqslant \frac{1}{\sigma}\|x' - x\|_2 \\ &\leqslant \frac{1}{2}\left(\Phi^{-1}(G(x)_{y_1}) - \Phi^{-1}(G(x)_{y_2})\right),\end{aligned}$$

which implies that

$$\frac{1}{2}\left(\Phi^{-1}(G(x)_{y_1}) + \Phi^{-1}(G(x)_{y_2})\right) \leqslant \Phi^{-1}(G(x')_{y_1}). \tag{3}$$

For any $y \neq y_1$, we do the same logic as follows:

$$\begin{aligned}\Phi^{-1}(G(x')_y) - \Phi^{-1}(G(x)_y) &\leqslant |\Phi^{-1}(G(x')_y) - \Phi^{-1}(G(x)_y)| \\ &\leqslant \frac{1}{\sigma}\|x' - x\|_2 \\ &\leqslant \frac{1}{2}\left(\Phi^{-1}(G(x)_{y_1}) - \Phi^{-1}(G(x)_{y_2})\right),\end{aligned}$$

which implies that

$$\Phi^{-1}(G(x')_y) \leqslant \Phi^{-1}(G(x)_y) + \frac{1}{2}\left(\Phi^{-1}(G(x)_{y_1}) - \Phi^{-1}(G(x)_{y_2})\right) \tag{4}$$

Since y_2 is the second-most likely class and Φ^{-1} is monotone increasing,

$$\Phi^{-1}(G(x)_y) \leqslant \Phi^{-1}(G(x)_{y_2}). \tag{5}$$

Plugging Equation (5) into Equation (4), we have

$$\begin{aligned}\Phi^{-1}(G(x')_y) &\leqslant \Phi^{-1}(G(x)_y) + \frac{1}{2}\left(\Phi^{-1}(G(x)_{y_1}) - \Phi^{-1}(G(x)_{y_2})\right) \\ &\leqslant \frac{1}{2}\left(\Phi^{-1}(G(x)_{y_1}) + \Phi^{-1}(G(x)_{y_2})\right). \end{aligned} \tag{6}$$

Combining Equation (3) and Equation (6), we have

$$\Phi^{-1}(G(x')_y) \leqslant \frac{1}{2}\left(\Phi^{-1}(G(x)_{y_1}) + \Phi^{-1}(G(x)_{y_2})\right) \leqslant \Phi^{-1}(G(x')_{y_1}),$$

which implies that $G(x')_y \leqslant G(x')_{y_1} \Rightarrow \operatorname{argmax}_y G(x')_y = y_1$. ■

Certified robustness of g

- In other words, we can compute the ℓ_2 radius in the image space such that the most likely class for any image x' within the radius will be the same as x.
- Now, let's consider the hard classifier $f : \mathbb{R}^k \to \mathcal{Y}$.
- Note that we can transform f to a soft classifier $F(x) = e_{f(x)}$.
- Then, the smoothed soft classifier G is defined by

$$\begin{aligned} G(x) &= \mathbb{E}_{\eta \sim \mathcal{N}(0,\sigma^2 I)} \left[F(x+\eta) \right] = \mathbb{E}_{\eta \sim \mathcal{N}(0,\sigma^2 I)} \left[e_{f(x+\eta)} \right] \\ &= \left[\mathbb{P}\left[f(x+\eta) = y_1 \right], \ldots, \mathbb{P}\left[f(x+\eta) = y_{|\mathcal{Y}|} \right] \right] \end{aligned}$$

- Applying the theorem to G, we can derive that the smoothed hard classifier $g(x) = \operatorname{argmax}_y \mathbb{P}\left[f(x+\eta) = y \right]$ is robust within some ℓ_2 radius.

Certified radius

- The radius

$$\frac{\sigma}{2}\left(\Phi^{-1}(\underbrace{\mathbb{P}\left[f(x+\eta)=y_1\right]}_{G(x)_{y_1}}) - \Phi^{-1}(\underbrace{\mathbb{P}\left[f(x+\eta)=y_2\right]}_{G(x)_{y_2}})\right)$$

  is called the **certified radius** of x.

- An image x has a high certified radius if
  1. the noise level σ is high $\rightarrow$ but it usually drops the accuracy.
  2. $\mathbb{P}\left[f(x+\eta)=y_1\right] \gg \mathbb{P}\left[f(x+\eta)=y_2\right]$, *i.e.*, the f predicts x with high probability under the additive Gaussian noise.

- Training a base classifier f with Gaussian noise augmentation will help to improve the certified robustness.

Outline

Motivation

- Adversarial examples can significantly degrade the performance of neural networks, raising security concerns about their deployment to real-world applications.
- When these neural networks are deployed to social media, such as Facebook or Instagram, this vulnerability can pose another serious threat to individual users.

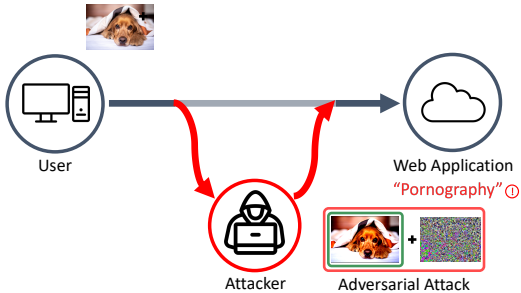
Moon et al., *Preemptive Image Robustification for Protecting Users against Man-in-the-Middle Adversarial Attacks*, AAAI 2022.

Motivation

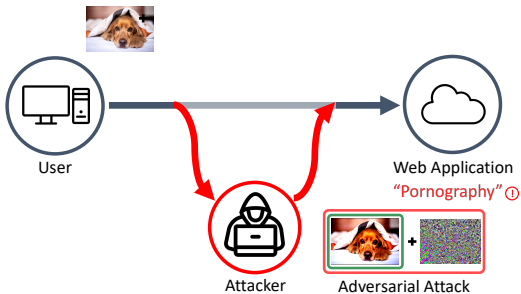
- Consider the situation where social media users upload their images from local machines to remote storage.

Motivation

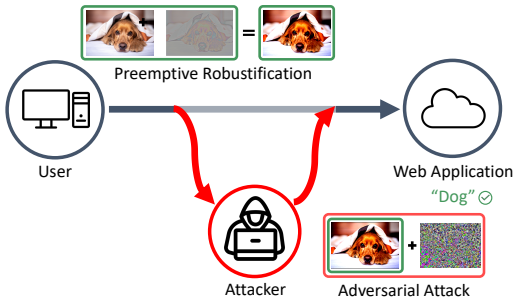
- Consider the situation where social media users upload their images from local machines to remote storage.
- Suppose there exists a **man-in-the-middle** (MitM) adversary that can intercept and add perturbations to the images web users upload during transmission.



- This adversary can easily vandalize neural network-based web services such as image auto-tagging by perturbing the images to be misclassified.
- This type of attack can severely deteriorate user experience, especially as the adversary can further use this attack to insult the uploaders beyond simple misclassification.



- How can the users protect their images from the MitM attacks?
- Based on the fact that the users hold control of their images before uploading, the users can take action to slightly modify the images **ahead of the adversary** to be robust against adversarial attacks.



Problem setup

- Suppose an image $x_o \in \mathbb{R}^k$ with the label $y_o \in \mathcal{Y}$ and a classifier f are given.
- A defender preemptively modifies x_o to produce a new image x_r that is visually indistinguishable from x_o with the modification budget δ.
- After the modification, the defender discards the original image x_o so that an adversary can only see the modified image x_r.
- Finally, the adversary attempts to generate adversarial examples from x_r with the perturbation budget ϵ.
- The defender's objective is to make x_r be robust against adversarial attacks.

Robust region

- To start, extend the notion of adversarial robustness to the whole image space and define the **robust region** of a classifier f as the set of images that f can output robust prediction under adversarial perturbations.

Definition (ϵ-robust region)

Let $f : \mathbb{R}^k \to \mathcal{Y}$ be a classifier and $\epsilon > 0$ be the perturbation budget of an adversary. The *ϵ-robust region* of the classifier f is defined by $R_\epsilon(f) \triangleq \{x \in \mathbb{R}^k \mid f(x') = f(x),\ \forall x'\ s.t.\ x' \in B_\epsilon(x)\}$.

Preemptive robustness

- Then, the defender's optimal strategy against the adversary is to
  1. make x_r be correctly classified as y_o
  2. locate x_r in the robust region $R_\epsilon(f)$ so that x_r is robust against adversarial perturbations.
- If both of these two conditions are satisfied, we say x_o is **preemptively robust** against adversarial attacks.
- x_r is called preemptive robustification of x_r.

Definition (preemptive robustness)

Given a classifier $f : \mathbb{R}^k \to \mathcal{Y}$, an image x_o with its label y_o is called *preemptively robust* against adversarial attacks if there exists $x_r \in B_\delta(x_o)$ such that (1) $f(x_r) = y_0$ and (2) $x_r \in R_\epsilon(f)$.

Preemptive robustification algorithm

- How can we preemptively robustify an image?
- Finding a preemptively robustified image x_r from the original image x_o can be formulated as the following optimization problem, which is directly from the definition:

$$\begin{aligned} &\underset{x_r}{\text{minimize}} \quad \mathbb{1}_{f(x_r)\neq y_o} + \mathbb{1}_{x_r \notin R_\epsilon(f)} \\ &\text{subject to} \quad x_r \in B_\delta(x_o). \end{aligned} \tag{7}$$

- Replacing the 0-1 loss with the cross-entropy loss and via a mild assumption, Equation (7) can be reformulated as

$$\begin{aligned} &\underset{x_r}{\text{minimize}} \quad \sup_{x_r^a} \ \ell(x_r^a, f(x_o)) \\ &\text{subject to} \quad x_r \in B_\delta(x_o) \text{ and } x_r^a \in B_\epsilon(x_r), \end{aligned}$$

where x_r^a denotes an adversarial example of x_r.

$$\begin{aligned} &\underset{x_r}{\text{minimize}} \quad \sup_{x_r^a} \ \ell(x_r^a, f(x_o)) \\ &\text{subject to} \ \ x_r \in B_\delta(x_o) \text{ and } x_r^a \in B_\epsilon(x_r) \end{aligned}$$

- To solve the optimization problem above, first compute the approximate solution $x_r^{a,T}$ of the inner maximization problem by T-step PGD starting from x_r.
- Note that each update step in the inner maximization can be represented by computational graph.
- Then, replace x_r^a with $x_r^{a,T}$ and compute the gradient of $\ell(x_r^{a,T}, f(x_o))$ with respect to x_r, which can be easily done via back-propagation.
- After computing the gradient, update x_r by gradient descent.

Preemptively robust training

- Train a classifier where data points are preemptively robust with high probability.
- To induce data points to be preemptively robust, the defender's optimal training objective should have the following form:

$$\begin{aligned}&\underset{\theta}{\text{minimize}} \quad \underset{(x_o,y_o)\sim\mathcal{D}}{\mathbb{E}} \left[\ell(\widehat{x}_r^a, y_o; \theta)\right] \\ &\text{subject to} \quad \widehat{x}_r^a = \underset{x_r^a \in B_\epsilon(\widehat{x}_r)}{\operatorname{argmax}} \ \ell(x_r^a, y_o) \quad \text{and} \quad \widehat{x}_r = \underset{x_r \in B_\delta(x_o)}{\operatorname{argmin}} \ \sup_{x_r^a \in B_\epsilon(x_r)} \ell(x_r, y_o).\end{aligned}$$

- In other words, first find a robust point $\widehat{x}_r$ near the training point x_o and run adversarial training on $\widehat{x}_r$.

Preemptively robust training

$$\underset{\theta}{\text{minimize}} \quad \underset{(x_o,y_o)\sim\mathcal{D}}{\mathbb{E}} \left[\ell(\widehat{x}_r^a, y_o; \theta)\right]$$

$$\text{subject to} \quad \widehat{x}_r^a = \underset{x_r^a \in B_\epsilon(\widehat{x}_r)}{\operatorname{argmax}} \ \ell(x_r^a, y_o) \quad \text{and} \quad \widehat{x}_r = \underset{x_r \in B_\delta(x_o)}{\operatorname{argmin}} \ \sup_{x_r^a \in B_\epsilon(x_r)} \ \ell(x_r, y_o).$$

- However, finding a robust point $\widehat{x}_r$ is computationally expensive.

Preemptively robust training

$$\underset{\theta}{\text{minimize}} \quad \underset{(x_o,y_o)\sim\mathcal{D}}{\mathbb{E}} \left[\ell(\widehat{x}_r^a, y_o; \theta)\right]$$

$$\text{subject to} \quad \widehat{x}_r^a = \underset{x_r^a \in B_\epsilon(\widehat{x}_r)}{\operatorname{argmax}} \ \ell(x_r^a, y_o) \quad \text{and} \quad \widehat{x}_r = \underset{x_r \in B_\delta(x_o)}{\operatorname{argmin}} \ \cancel{\underset{x_r^a \in B_\epsilon(x_r)}{\sup}} \ \ell(x_r, y_o).$$

- However, finding a robust point $\widehat{x}_r$ is computationally expensive.
- In practice, we omit the inner maximization step to ease the high computational cost.

Preemptively robust training

- Different from adversarial training, preemptively robust training does not enforce the original images x_o to be robust against adversarial perturbation.
- Considering the trade-off between robust and standard accuracy, it may be less prone to suffering from standard accuracy drop.

Evaluations

- First craft preemptively robustified images from the original images, given the modification budgets of the defender and the adversary.
- Assume the defender has the same modification budget as the adversary (*i.e.*, $\delta = \epsilon$).
- Then, measure the standard and robust accuracy of the preemptively robustified images.
- For measuring robust accuracy, use 20-step PGD and AutoAttack[1], an ensemble of state-of-the-art white and black-box attacks.

[1] Croce and Hein, *Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks*, ICML 2020.

CIFAR-10 results

Model	Preempt.	Clean	PGD	AA
ADV[2]	None	86.72	54.59	51.68
ADV	**Ours**	86.72	86.23	81.70
Ours	**Ours**	**88.54**	**87.10**	**82.88**

Table: Classification accuracy under adversaries with ℓ_∞ perturbation, $\epsilon = 8$.

Model	Preempt.	Clean	PGD	AA
ADV	None	90.85	71.90	71.21
ADV	**Ours**	90.85	84.81	83.56
Ours	**Ours**	**92.57**	**91.81**	**89.32**

Table: Classification accuracy under adversaries with ℓ_2 perturbation, $\epsilon = 0.5$.

[2] ADV denotes PGD adversarial training with early stopping.

ImageNet results

Model	Preempt.	Clean	PGD	AA
ADV[3]	None	56.24	32.03	27.52
ADV	**Ours**	56.24	55.79	47.14
Ours	**Ours**	**61.01**	**59.66**	**48.24**

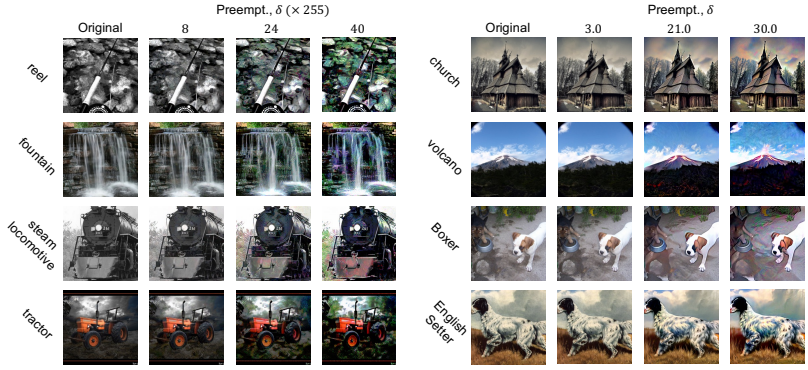
Table: Classification accuracy under adversaries with ℓ_∞ perturbation, $\epsilon = 4$.

Model	Preempt.	Clean	PGD	AA
ADV	None	54.99	32.07	27.58
ADV	**Ours**	55.05	51.70	43.32
Ours	**Ours**	**61.60**	**58.13**	**43.60**

Table: Classification accuracy under adversaries with ℓ_2 perturbation, $\epsilon = 3.0$.

[3] ADV denotes free adversarial training.

Examples of preemptively robustified images



Summary

	Pros	Cons
Preprocessing-based defense	▶ Can be applied to existing classifiers. ▶ No standard accuracy drop.	▶ Broken by stronger adaptive attacks.
Adversarial training	▶ Simple and intuitive. ▶ Empirically very strong.	▶ Drop in standard accuracy. ▶ High computational cost for training.
Randomized smoothing	▶ Theoretical guarantee on robustness.	▶ High computational cost for inference.
Preemptive robustness	▶ Achieves high standard and robust accuracy.	▶ Needs assumption that images can be preemptively modified.