Defenses Against Adversarial Examples

Neil Gong
Prevention – robust classifiers

• Empirically robust classifier
  • A particular attack cannot find adversarial example within a $L_p$ norm ball
  • $(p, \varepsilon)$-robust against an attack for $x$, if the attack does not find adversarial perturbation whose $L_p$ norm is no larger than $\varepsilon$.

• Certifiably robust classifier
  • No adversarial examples exist within a $L_p$ norm ball.
  • $(p, \varepsilon)$-certifiably robust for $x$, if no adversarial perturbation whose $L_p$ norm is no larger than $\varepsilon$ exists.
Training empirically robust classifier

An attack
\[
\max_{\delta \in B_p(x, \epsilon)} L(x + \delta, y|\theta)
\]

Adversarial training
\[
\min_{\theta} \sum_{(x, y)} \max_{\delta \in B_p(x, \epsilon)} L(x + \delta, y|\theta)
\]
Adversarial training

\[
\min_\theta \sum_{(x, y)} \max_{\delta \in B_p(x, \varepsilon)} L(x + \delta, y | \theta)
\]

• Alternate between max and min
• Inner max
  • Finding adversarial perturbation \( \delta \), e.g., Projected Gradient Descent (PGD)
• Outer min
  • Updating model parameters \( \theta \) using both normal and adversarial examples
Issues of adversarial training

• No certifiable guarantee

• May not be empirically robust against unseen attacks
  • Use multiple attacks during training

• May not be robust to perturbation larger than $\varepsilon$ used in training
We observe that for Table 2: CIFAR10: Performance of the adversarially trained network against different adversaries, the gradients (the decision-based attack does not utilize gradients) of the model. This is potentially due to the threshold filters learned by the model masking the loss of PGD on the MNIST better. For MNIST, there is a sharp drop shortly after. Moreover, we observe that the performance of Boundary Attack (DBA) in the case of the MNIST adversarially trained networks, we also evaluate the performance of the Decision adversaries respectively (the training strength). The MNIST and CIFAR10 networks were trained against with 2000 steps and PGD on standard and adversarially trained models.

For each model of attack we show the most effective attack in bold. The source networks considered for the attack are: the network itself (A) (white-box attack), an independently initialized and trained copy of the network (A'), a copy of the network trained on natural examples (A nat), and the network trained on natural examples and the source network trained against PGD (A nat). For MNIST, # Accuracy = 20.40.60.80.0. For CIFAR10, # Accuracy = 20.40.60.80.0. There is a sharp drop shortly after. Moreover, we observe that the performance is equal or less or equal to the value used during training, the performance is equal or less or equal to the value used during training.

DBA: decision boundary attack

(a) MNIST, $\ell_\infty$-norm
Evaluating an empirically robust classifier

\[ x' \] \rightarrow \text{Classifier} \rightarrow \text{Incorrect label}

\[ x'' \] \rightarrow \text{Robust Classifier} \rightarrow \text{Incorrect label}
Evaluating an empirically robust classifier

• Metric 1
  • Whether human perceives x’’ and x as the same
  • no-> defense is effective
  • Hard to implement

• Metric 2
  • d(x’,x) vs. d(x’’, x)
  • d(x’’, x) > d(x’,x) -> defense is effective
  • d(x’’, x) - d(x’,x) measures effectiveness
  • Consider strong adaptive attacks
Certifiably robust classifier

- A classifier is \((p, \varepsilon)\)-certifiably robust for \(x\), if no adversarial perturbation whose \(L_p\) norm is no larger than \(\varepsilon\) exists.

- Verification
  - Given a classifier and \(x\), verify whether the classifier is \((p, \varepsilon)\)-certifiably robust for \(x\)

- Certification
  - Given a classifier and \(x\), deriving \(p\) and \(\varepsilon\)
Verification via interval analysis

• Given $x, p=\infty, \varepsilon$, we propagate the intervals from the input to the output

• Limitations
  • False negatives
  • Limited to $p=\infty$
  • Not effective for certain classifiers
Certification via randomized smoothing

• Given a classifier and $x$, deriving $p$ and $\varepsilon$

• Many methods have been developed

• Randomized smoothing
  • Applicable to any classifier
  • Scalable to large neural networks
Adversarial example is close to classification boundary?
Measuring Adversarial Examples

A normal example: digit 0

An adversarial example with a target label 9
Randomized smoothing
Formal definition of randomized smoothing

• Input
  • a classifier $f$
  • an example $x$
  • a noise distribution

• Output
  • $g(x) = \arg\max_c \Pr(f(x + r) = c)$
Deriving \((\rho, \varepsilon)\)

- Noise is isotropic Gaussian distribution

- \(g(x + \delta) = C_A\) when \(|\delta|_2 \leq \varepsilon\)

\[
\varepsilon = \frac{\sigma}{2} \left( \Phi^{-1}(\rho_A) - \Phi^{-1}(\rho_B) \right)
\]

Certified radius
Tightness of the bound

• Given
  • No assumptions on the classifier $f$
  • Randomized smoothing with Gaussian noise

• The derived bound is tight
Estimating the label probabilities

• Sampling a large number of noise

• Predicting labels for the noisy examples

• Estimating label probabilities with probabilistic guarantees
Generalization to top-k

• Input
  • a classifier $f$
  • an example $x$
  • a noise distribution

• Output
  • $p_c = \Pr(f(x + r) = c)$
  • The smoothed classifier predicts $k$ labels with the largest label probabilities

• A label is among the top-$k$ labels if the adversarial perturbation is bounded
Training to improve certified accuracy

- Adding random noise during training

- Adding certified radius as a regularization term

\[
\begin{align*}
\frac{1}{N} \sum_{i=1}^{N} & \left[ \frac{1}{N} \sum_{j=1}^{N} \left[ \mathbb{1}_{\{g_\theta(x) \neq y\}} + \mathbb{1}_{\{g_\theta(x) = y, CR(g_\theta; x, y) < \epsilon\}} \right] \right] \\
& = \frac{1}{N} \sum_{i=1}^{N} \left[ \frac{1}{N} \sum_{j=1}^{N} \left[ \mathbb{1}_{\{g_\theta(x) \neq y\}} + \mathbb{1}_{\{g_\theta(x) = y, CR(g_\theta; x, y) < \epsilon\}} \right] \right]
\end{align*}
\]

0/1 Classification Error + 0/1 Robustness Error

MACER: Attack-free and Scalable Robust Training via Maximizing Certified Radius
Randomized smoothing

• Strengths
  • Applicable to any classifier
  • Scalable to large classifier

• Limitations
  • Efficiency – need many predictions
  • Probabilistic guarantees