Reliable and Interpretable Artificial Intelligence

Lecture 4a: Adversarial Defenses

Martin Vechev
ETH Zurich

Fall 2020

SRILAB

http://www.sri.inf.ethz.ch

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Can we Avoid Adversarial Examples?

Many works have tried to, but follow-up works showed that all fail

The main **successful defenses** in practice now incorporate adversarial examples during training

Some pretty good experimental defenses exist

Adversarial Accuracy vs. Test Accuracy

Adversarial accuracy refers to a metric on the test set where for each data point we check if the network classifies the point correctly **and** the network is robust in a region around that point.

Example [$\boldsymbol{l}_\infty$ ball]: Let ϵ=0.3 , and let the test set T contain 100 examples. For each example $d_i \in T$, lets check if in the l_∞ region of size ≤ 0.3 around d_i, we find an (adversarial) example with a different classification than d_i. For that purpose we typically use a **PGD attack**. Now suppose, 95 of the 100 examples classify correctly and for 15 of these 95, we find an adversarial example. Then, our adversarial accuracy will be $\frac{80}{100} = 80\%$ and our test accuracy will be $\frac{95}{100} = 95\%$.

Adversarial accuracy and Test accuracy can be at odds: it is possible to raise the adversarial accuracy which tends to lower test accuracy. This trade off is being **actively investigated**.

Defense as Optimization Problem

find	θ
minimize	$\rho(\theta)$
where	$\rho(\theta) = \mathbf{E}_{(\mathrm{x},\mathrm{y})\sim D}\,[\max\limits_{x' \in S(x)} L(\theta, x', y)]$

Outer minimization problem

find $\boldsymbol{\theta}$ that minimizes the high loss, aiming to train a robust classifier

Inner maximization problem

find $\boldsymbol{x}'$ that achieves high loss

D is the underlying distribution

$\mathbf{E}_{(\mathrm{x},\mathrm{y})\sim D}$ is typically estimated with the empirical risk

$S(x)$ denotes the perturbation region around point x, that is, we want all points in $S(x)$ to classify the same as x . We can pick $S(x)$ to be:

$$S(x) = \{\, \boldsymbol{x}' \mid \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty < \boldsymbol{\epsilon}\}$$

Madry et.al, 2017

PGD Defense in Practice

Step 1: select a mini-batch B of examples from dataset D.

Step 2: compute B_{max} by applying PGD attack (actually computes an approximation) as follows to every point $(x, y) \in B$:

$$x_{max} = \underset{x' \in S(x)}{\mathbf{argmax}} \; L(\theta, x', y)$$

Note: x_{max} need not be adversarial example; it just aims to maximize L

Step 3: solve outer problem:

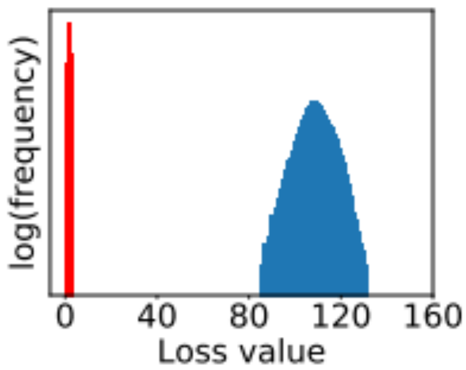
$$\theta' = \theta - \frac{1}{|B_{max}|} \sum_{(x_{max}, y) \in B_{max}} \nabla_\theta \, L(\theta, x_{max}, y)$$

Step 4: goto Step 1. Various stopping criteria, including reaching a certain number of epochs.

*The conversion of the original min-max problem to the 4 steps above is based on Danskin's theorem

Why do we think we can find a good approximate solution to the inner maximization problem?

Experiments show that many local maxima inside $S(x)$ have well-concentrated loss values. This is inline with why we believe neural network training is possible (many local minima with similar values).



This graph is for a **single example**: goal is to maximize the cross-entropy loss measured for 100,000 random starting points in $S(x)$.

The red graph indicates the value of the loss L for an adversarially trained network.

The blue graph is for the loss L of a non-adversarially trained network.

Points to Consider when Defending

Model capacity matters: larger networks are more defendable and less easy to be attacked with transferrable examples. Training smaller nets with PGD has negative effects on accuracy.

Training with **adversarial examples from PGD attacks (many steps and project)** tends to perform better than training with adversarial examples from FGSM attacks (one step, no projection).

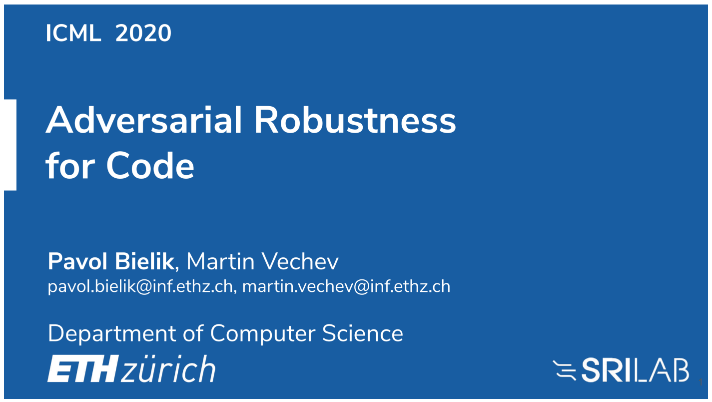
Even on larger networks, defenses can negatively affect accuracy (e.g. CIFAR). More research is needed here. By this we mean that after the network is trained, we test its accuracy on the test set. And there, it is more robust yet more points classify incorrectly.

"No free lunch in adversarial robustness", Tsipras et. al. 2018
Proves that if we want robust model, decrease in standard accuracy is inevitable!

"Adversarially Robust Generalization Requires More Data ", Schmidt et. al. 2018
Provides lower bound on number of samples needed to achieve adversarial robustness

"Theoretically Principled Trade-off between Robustness and Accuracy", Zhang et.al, 2019
Improves slightly on the PGD defense; also combines with standard (e.g., cross-entropy) loss.

Interesting Use Case: Robust models of code



- Involves adversarial training
- Learning representations
- Learning to abstain
- Rather unexplored area

https://www.sri.inf.ethz.ch/publications/bielik2020robustcode
ICML'2020

Lecture Summary

- We looked at a way to (experimentally) defend the network by training with adversarial examples, specifically the PGD defense. This results in a min-max nested optimization problem.

- Adversarial training can lower standard accuracy. Remains a question of research interest, how to avoid this from happening.