ECE595 / STAT598: Machine Learning I
Lecture 37 Robustness and Accuracy Trade Off

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

Today's Agenda

Two Fundamental Questions about Adversarial Attack

- Can We completely avoid adversarial attack?
  - Is there any classifier that cannot be attacked?
  - We will show that all classifiers are adversarial vulnerable
- If adversarial attack is unavoidable, what can we do?
  - There is a natural trade-off between accuracy and robustness
  - You can be absolutely robust but useless, or absolutely accurate but very vulnerable
  - We will characterize this trade-off

Our Plan: This lecture is based on two very recent papers

- Fawzi et al. Adversarial vulnerability for any classifier, arXiv: 1802.08686
- Zhang et al. Theoretically principled trade-off between robustness and accuracy, arXiv: 1901.08573

This lecture is theoretical. We will not go into the details. We will highlight the main conclusions and interpret their results.

Outline

- Lecture 33-35 Adversarial Attack Strategies
- Lecture 36 Adversarial Defense Strategies
- Lecture 37 Trade-off between Accuracy and Robustness

Today's Lecture

- Adversarial robustness of any classifier
  - Can we completely avoid adversarial attack?
  - Is there any classifier that cannot be attacked?
  - We will show that all classifiers are adversarial vulnerable
- Robustness-accuracy trade off
  - If adversarial attack is unavoidable, what can we do?
  - There is a natural trade-off between accuracy and robustness
  - You can be absolutely robust but useless, or absolutely accurate but very vulnerable
  - We will characterize this trade-off

Adversarial Robustness of Any Classifier

The first question we ask: Is adversarial attack **unavoidable**?

- There are several papers discussing this issue.
- We will be focusing on: Fawzi et al. Adversarial vulnerability for any classifier, arXiv: 1802.08686
- There is another paper: Shafahi et al. Are adversarial examples inevitable, arXiv 1809.02104
- The results we are going to study are both **general** and **restrictive**
- They are general because the results are universal bounds for all classifiers
- They are restrictive because they assume a generative model, require high dimensionality, and are ℓ_p ball additive attack
- Our plan: Understand the major claims, and not to worry about the specific proofing techniques (e.g., Gaussian isoperimetric inequality)

Notation

- There is an input $\boldsymbol{x}$
- Assume that $\boldsymbol{x}$ comes from a generator $\boldsymbol{x} = g(\boldsymbol{z})$ where $\boldsymbol{z}$ is i.i.d. Gaussian.
- Think about a generative adversarial network (GAN) [1]. You give me $\boldsymbol{z}$, and then I generate the image $\boldsymbol{x}$ according to $\boldsymbol{x} = g(\boldsymbol{z})$.
- $\boldsymbol{r}$ is perturbation
- f is classifier
- In-distribution robustness:

$$r_{\text{in}}(\boldsymbol{x}) = \min_{\boldsymbol{r} \in \mathcal{Z}} \|g(\boldsymbol{z}+\boldsymbol{r}) - \boldsymbol{x}\| \quad \text{subject to} \quad f(g(\boldsymbol{z}+\boldsymbol{r})) \neq f(\boldsymbol{x}). \quad (1)$$

[1] GAN is not the same as adversarial attack. GAN is a method that approximates the distribution.

$r_{\text{in}}(\boldsymbol{x})$

- Let us take a closer look at $r_{\text{in}}(\boldsymbol{x})$:

$$r_{\text{in}}(\boldsymbol{x}) = \min_{\boldsymbol{r} \in \mathcal{Z}} \|g(\boldsymbol{z}+\boldsymbol{r}) - g(\boldsymbol{z})\| \quad \text{subject to} \quad f(g(\boldsymbol{z}+\boldsymbol{r})) \neq f(g(\boldsymbol{z})).$$

- To make things clearer, let us replace all the $\boldsymbol{x}$ by $g(\boldsymbol{z})$
- You can do that because you **assume $\boldsymbol{x}$** is generated from g
- $f(g(\boldsymbol{z}+\boldsymbol{r})) \neq f(g(\boldsymbol{z}))$ says that the perturbed data is classified differently from the original
- $\min_{\boldsymbol{r} \in \mathcal{Z}} \|g(\boldsymbol{z}+\boldsymbol{r}) - \boldsymbol{x}\|$ says that for those that causes mis-classification, I will minimize the perturbation strength
- The smallest perturbation that still causes misclassification is then defined as the robustness of f
- You want $r_{\text{in}}(\boldsymbol{x})$ as **large** as possible. The larger it is, the stronger perturbation the hacker needs to launch in order to fool your classifier

Unconstrained Robustness

- Can we generalize the result to arbitrary perturbations?
- That is, we are not limited to generative models
- To do so we need to define the **unconstrained robustness**

$$r_{\text{unc}}(\boldsymbol{x}) = \min_{\boldsymbol{r} \in \mathcal{X}} \|\boldsymbol{r}\| \quad \text{subject to} \quad f(\boldsymbol{x} + \boldsymbol{r}) \neq f(\boldsymbol{x}) \tag{2}$$

- You can show that

$$r_{\text{unc}}(\boldsymbol{x}) \leq r_{\text{in}}(\boldsymbol{x}).$$

- For certain classifiers, you can further have $\frac{1}{2} r_{\text{in}}(\boldsymbol{x}) \leq r_{\text{unc}}(\boldsymbol{x})$. See Fawzi Theorem 2.
- So if you bound $r_{\text{in}}(\boldsymbol{x}) \leq \eta$, you can also bound $r_{\text{unc}}(\boldsymbol{x})$

Main Result

- Here we are going to summarize the main result.
- We will present the result in its simplest form, i.e., a very narrow case, so that we can bypass the technical details.
- Read the paper to learn more.

Theorem (Fawzi et al. 2018 Theorem 1)

Let $f : \mathbb{R}^d \to \{1, \ldots, K\}$ be an arbitrary classification function. Then, for any η,

$$\mathbb{P}[r_{in}(\boldsymbol{x}) \leq \eta] \geq 1 - \sqrt{\frac{\pi}{2}} e^{-\frac{\eta^2}{2L^2}} \tag{3}$$

where L is the Lipschitz constant of g.

Remark: Lipschitz constant defines the maximum slope of a function. See `https://en.wikipedia.org/wiki/Lipschitz_continuity`

Interpreting the Result

Let us look at this equation

$$\mathbb{P}[r_{\text{in}}(\boldsymbol{x}) \leq \eta] \geq 1 - \sqrt{\frac{\pi}{2}} e^{-\frac{\eta^2}{2L^2}} \tag{4}$$

- The event you are measuring is $r_{\text{in}}(\boldsymbol{x}) \leq \eta$.
- This says: You want the robustness to be no better than η. This a bad event.
- The equation says: The probability could be big.
- There exists a perturbation of magnitude $\eta \propto L$ such that the classifier can be fooled.
- Normally, $L \ll \sqrt{d}$, where d is the dimension of $\boldsymbol{x}$ (think of an image).
- If you plug in $\eta = 2L$, then you can show that $\mathbb{P}[r_{\text{in}}(\boldsymbol{x}) \leq 2L] \geq 0.8$.
- For just $2L$ perturbation magnitude, you have 0.8 probability of fooling the classifier.

What Does Attack Scale with d?

Let us also quickly look at Shafahi et al. Are adversarial examples inevitable, arXiv 1809.02104

- The findings are quite similar to Fawsi's.
- They showed that with probability at least

$$1 - V_c \left(\frac{\pi}{2}\right)^{\frac{1}{2}} \exp\left\{-\frac{d-1}{2}\epsilon^2\right\}, \tag{5}$$

  then one of the followings will hold
  - The data $\boldsymbol{x}$ is originally misclassified, or
  - $\boldsymbol{x}$ can be attacked within an ϵ-ball.
- You can ignore the constant V_c.
- As the data dimension d grows, the probability will go to 1.
- So for large images, the probability of attacking is high.

So what do we learned?

Existence of Attack:

- The results above are **existence** results.
- With high probability, there exists a direction which can almost certainly fool the classifier.
- This holds for all classifiers, as long as the dimension is high enough.
- Think in this way: Each perturbation pixel is small, but the sum can be big.
- How to find this attack direction? Not the focus here.

Can Random Noise Attack?

- Random noise cannot attack, especially for white-box.
- The probability of getting the correct attack direction is close to zero.

Outline

- Lecture 33-35 Adversarial Attack Strategies
- Lecture 36 Adversarial Defense Strategies
- Lecture 37 Trade-off between Accuracy and Robustness

Today's Lecture

- Adversarial robustness of any classifier
  - Can We completely avoid adversarial attack?
  - Is there any classifier that cannot be attacked?
  - We will show that all classifiers are adversarial vulnerable
- Robustness-accuracy trade off
  - If adversarial attack is unavoidable, what can we do?
  - There is a natural trade-off between accuracy and robustness
  - You can be absolutely robust but useless, or absolutely accurate but very vulnerable
  - We will characterize this trade-off

Trade Off Analysis

- If adversarial attack is unavoidable, what can we do?
  - We want to show that there is a natural trade-off between accuracy and robustness
  - You can be absolutely robust but useless, or absolutely accurate but very vulnerable
- Intuitively, the existence of trade-off makes sense:
  - You can be very robust, e.g., always claims class 1 regardless what you see. Then you are ultimately robust but not accurate.
  - You can be very accurate, e.g., a perceptron algorithm for linearly separable problems. But you have terrible robustness.
- Our discussion is based on this paper
  - Zhang et al. Theoretically principled trade-off between robustness and accuracy, arXiv: 1901.08573
  - Published in ICML 2019
- There is another very interesting paper
  - Tsipras et al., Robustness May Be at Odds with Accuracy, arXiv: 1805.12152
  - Some observations are quite intriguing.

Main Messages of Zhang et al. 2019

We will focus on Zhang et al. Theoretically principled trade-off between robustness and accuracy, arXiv: 1901.08573.

There are three messages:

- (1) There is an intrinsic trade off between robustness and accuracy
- (2) It is possible to upper bound both terms using a technique called classification-calibrated loss
- (3) You can develop a heuristic algorithm to minimize the empirical risk

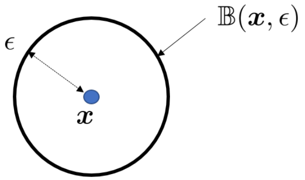
In addition, the paper showed a few very interesting results

- The trade-off optimization generalizes adversarial training
- They outperform defense methods in NIPS 2018 challenges

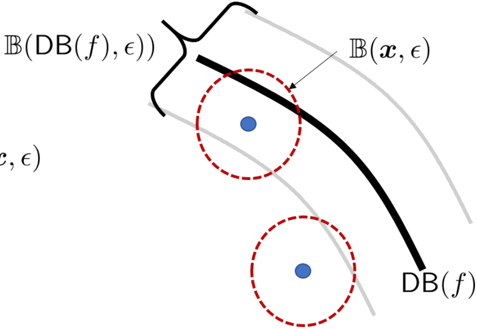
Notation

- $\boldsymbol{x} \in \mathcal{X}$: Input data. Random variable $\boldsymbol{X}$. Realization $\boldsymbol{x}$.
- $y \in \mathcal{Y} = \{+1, -1\}$: Label. Random variable Y. Realization y.
- Classifier: $f : \mathcal{X} \to \mathcal{Y}$
- $\mathbb{B}(\boldsymbol{x}, \epsilon)$ = an ϵ-**ball** surrounding the point $\boldsymbol{x}$
  - $\mathbb{B}(\boldsymbol{x}, \epsilon) = \{\mathbf{x}' \in \mathcal{X} : \|\mathbf{x}' - \mathbf{x}\| \leq \epsilon\}$
- **Decision boundary** of the classifier $\text{DB}(f) = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = 0\}$.
- **Neighborhood of the decision boundary** $\mathbb{B}(\text{DB}(f), \epsilon)$.
  - $\mathbb{B}(\text{DB}(f), \epsilon) = \{\boldsymbol{x} \in \mathcal{X} : \exists \mathbf{x}' \in \mathbb{B}(\boldsymbol{x}, \epsilon) \text{s.t.} f(\boldsymbol{x}) f(\boldsymbol{x}') \leq 0\}$
  - Basically: The band surrounding the decision boundary
  - Pick a point $\boldsymbol{x}$. If $\boldsymbol{x}$ is inside the band, then you can find $\boldsymbol{x}'$ with the epsilon ball of $\boldsymbol{x}$, where $f(\boldsymbol{x}) = +1$ and $f(\boldsymbol{x}') = -1$.
  - If $\boldsymbol{x}$ is outside the band, then within the same epsilon ball you will not be able to find a point that is predicted with an opposite label.

Notation



ϵ
$\boldsymbol{x}$
$\mathbb{B}(\boldsymbol{x}, \epsilon)$



$\mathbb{B}(\mathrm{DB}(f), \epsilon))$
$\mathbb{B}(\boldsymbol{x}, \epsilon)$
$\mathrm{DB}(f)$

Natural Classification Error

$$\mathcal{R}_{\text{nat}}(f) = \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}} \mathbb{I}\{f(\boldsymbol{X})Y \le 0\}. \tag{6}$$

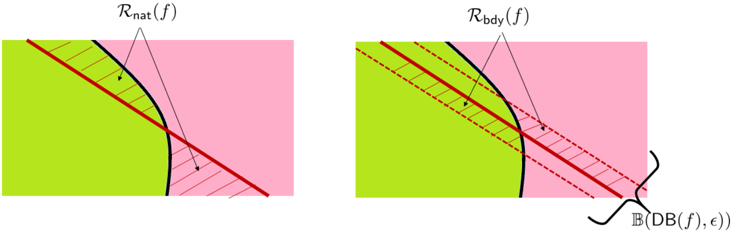
- You pick an input $\boldsymbol{X}$.
- The prediction is $f(\boldsymbol{X})$.
- You compare with the true label Y.
- If mismatch, then $f(\boldsymbol{X})Y \le 0$.
- The indicator function $\mathbb{I}$ will tell you whether this is indeed a mistake.
- Then you average over all the possible inputs $\boldsymbol{X} \sim D$.
- This will tell you the amount of error made by your classifier.
- Of course, you want this natural error as small as possible.
- $1 - \mathcal{R}_{\text{nat}}(f)$ is the **natural accuracy**. You want it as high as possible.

Accuracy and Robustness

Boundary Classification Error

$$\mathcal{R}_{\text{bdy}}(f) = \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbb{I}\{\boldsymbol{X} \in \mathbb{B}(\text{DB}(f), \epsilon), f(\boldsymbol{X})Y > 0\} \tag{7}$$

- $\boldsymbol{X} \in \mathbb{B}(\text{DB}(f)$ means the point $\boldsymbol{X}$ is inside the band.
- $f(\boldsymbol{X})Y > 0$ means that $\boldsymbol{X}$ is correctly classified.
- So, $\mathcal{R}_{\text{bdy}}(f)$ is anything inside the band **and** is correctly classified.

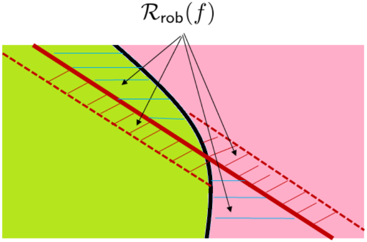


Accuracy and Robustness

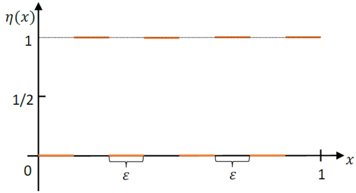
Robust Classification Error

$$\mathcal{R}_{\text{rob}}(f) = \mathcal{R}_{\text{nat}}(f) + \mathcal{R}_{\text{bdy}}(f) \tag{8}$$

- This is the sum of the two error: Anything that you have already made mistake (natural error), plus anything that you will likely to make mistake (boundary error)



Example



- The input is $x \in [0, 1]$.
- The true label y is either $+1$ or -1.
- Partition the input space into segments. Each segment has length ϵ.
- Odd segments are -1. Even segments are $+1$.
- Also define the posterior probability $\eta(x) \overset{\text{def}}{=} \mathbb{P}[Y = +1 | X = x]$

Example

	Bayes Optimal Classifier	All-One Classifier
$\mathcal{R}_{\text{nat}}$	0 (optimal)	1/2
$\mathcal{R}_{\text{bdy}}$	1	0
$\mathcal{R}_{\text{rob}}$	1	1/2 (optimal)

- Because you know the posterior distribution, Bayesian optimal classifier (based on MAP) will be exactly the same as $\eta(x)$. So $\mathcal{R}_{\text{nat}} = 0$ and it is optimal.
- The boundary error is 1, because the band is just the entire internal
- You can choose an all-one classifier. You always claim 1.
- This is a bad classifier in terms of natural accuracy. Half is correct, half is wrong.
- But the robustness error is actually better than Bayesian optimal.

Upper Bounding the Error

- After defining how to measure robustness, we can now ask about the **fundamental limit** of $\mathcal{R}_{\text{rob}}(f)$.
- The approach proposed by the paper is to
  - Define $\mathcal{R}^*_{\text{nat}} = \min_f \mathcal{R}_{\text{nat}}(f)$ be the best classifier (based on minimize the natural error).
  - We want to upper bound $\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}}$, so that we know $\mathcal{R}_{\text{rob}}(f)$ is more than $\mathcal{R}^*_{\text{nat}}$ by some maximum amount.
  - If we can find such upper bound, then we can perhaps minimizing the upper bound.
- Let us first state the theorem, and discuss the equations.
- We will skip the details. You should read the paper.

Theorem (Zhang et al. 2019 Theorem 3.1)

$$\mathcal{R}_{rob}(f) - \mathcal{R}^*_{nat} \leq \psi^{-1}\left(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi\right) + \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda). \quad (9)$$

Basic Argument

The theorem states that

$$\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}} \leq \psi^{-1}\left(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi\right) + \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda). \quad (10)$$

- The basic argue goes as follows.

$$\begin{aligned}
&\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}} \\
&\overset{(a)}{=} \mathcal{R}_{\text{nat}}(f) - \mathcal{R}^*_{\text{nat}} + \mathcal{R}_{\text{bdy}}(f) \qquad \text{because } \mathcal{R}_{\text{rob}} = \mathcal{R}_{\text{nat}} + \mathcal{R}_{\text{bdy}} \\
&\overset{(b)}{\leq} \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathcal{R}_{\text{bdy}}(f) \qquad \text{using surrogate loss } \psi \\
&\overset{(c)}{=} \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathbb{P}[\boldsymbol{X} \in \mathbb{B}(\text{DB}(f), \epsilon), f(\boldsymbol{X})Y > 0] \\
&\overset{(d)}{\leq} \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda), \qquad \text{for some } \lambda > 0.
\end{aligned}$$

Let us talk about these steps one by one.

Step (b)

$$\begin{aligned}\mathcal{R}_{\mathrm{rob}}(f) - \mathcal{R}^*_{\mathrm{nat}} &= \mathcal{R}_{\mathrm{nat}}(f) - \mathcal{R}^*_{\mathrm{nat}} + \mathcal{R}_{\mathrm{bdy}}(f) \\ &\overset{(b)}{\leq} \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \mathcal{R}_{\mathrm{bdy}}(f)\end{aligned}$$

- In principle, $\mathcal{R}_{\mathrm{nat}}(f)$ should be measured using $\mathcal{R}_{\mathrm{nat}}(f) = \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\mathbb{I}\{f(\boldsymbol{X})Y \leq 0\}$.
- The 0-1 loss is not differentiable, and poses difficulty in analysis.
- One way to handle that is to replace the 0-1 loss by the so-called classification-calibrated surrogate loss [2].
- Surrogate loss comes with a pair of functions ϕ and ψ.
- Here are some examples

Loss	$\phi(\alpha)$	$\psi(\theta)$
Hinge	$\max\{1-\alpha, 0\}$	θ
Sigmoid	$1 - \tanh(\alpha)$	θ
Exponential	$\exp(-\alpha)$	$1 - \sqrt{1-\theta^2}$
Logistic	$\log_2(1+\exp(-\alpha))$	$\psi_{\log}(\theta)$

[2] See Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. Journal of the American Statistical Association,

Step (b)

- So if you choose the hinge loss, for example, then $\phi(\alpha) = \max(1-\alpha, 0)$ and $\psi(\theta) = \theta$.
- Substituting these into the equation, you will have $\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}} \leq \mathcal{R}_\phi(f) - \mathcal{R}^*_\phi + \mathcal{R}_{\text{bdy}}(f)$
- If you can further upper bound $(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi)$ then you are good
- It turns out that $(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi)$ can be bounded using Theorem 2

Theorem (Zhang et al. 2019 Theorem 3.2)

$$
\begin{aligned}
\psi\left(\theta - \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)\right) &\leq \mathcal{R}_\phi(f) - \mathcal{R}^*_\phi \\
&\leq \psi\left(\theta - \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)\right) + \xi.
\end{aligned}
$$

Step (c) and (d)

Steps (c) and (d):

- (c) is just the definition of the $\mathcal{R}_{\text{bdy}}(f)$
- (d) follows from this

$$\begin{aligned}
&\mathbb{P}[\boldsymbol{X} \in \mathbb{B}(\text{DB}(f), \epsilon), f(\boldsymbol{X})Y > 0] \\
&\le \mathbb{P}[\boldsymbol{X} \in \mathbb{B}(\text{DB}(f), \epsilon)] \qquad \text{former is a subset of latter} \\
&= \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \mathbb{I}\{f(\boldsymbol{X}') \neq f(\boldsymbol{X})\} \\
&= \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \mathbb{I}\{f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda < 0\} \qquad \text{for all } \lambda \\
&\le \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)
\end{aligned}$$

- You can think of λ as a regularization parameter
- Theorem 3.1 holds for all λ
- Theorem 3.2 says that in order for theorem to hold, you need to carefully pick a λ

Optimization

- The theorem above suggest an optimization to minimize $\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}}$:

$$\min_f \; \underbrace{\psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi)}_{\text{accuracy}} + \underbrace{\mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)}_{\text{robustness}}$$

- You can replace the first term by the empirical risk $\phi(f(\boldsymbol{X})Y)$
- This will give you

$$\min_f \; \mathbb{E}\left\{ \underbrace{\phi(f(\boldsymbol{X})Y)}_{\text{accuracy}} + \underbrace{\max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X}, \epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)}_{\text{robustness}} \right\}$$

- There is a regularization parameter λ

What do you gain?

Let us look at this optimization again:

$$\min_f \; \mathbb{E}\left\{ \underbrace{\phi(f(\boldsymbol{X})Y)}_{\text{accuracy}} + \underbrace{\max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon)} \phi(f(\boldsymbol{X}')f(\boldsymbol{X})/\lambda)}_{\text{robustness}} \right\}$$

- This optimization is a trade-off between **accuracy** and **robustness**
- Recall **adversarial training** (Madry et al.)

$$\min_f \; \mathbb{E}\left\{ \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon)} \phi(f(\boldsymbol{X}')Y) \right\}$$

- It is an upper bound of $\mathcal{R}_{\text{rob}}(f)$
- The upper bound offered by the trade-off formulation is tighter

Summary

What do we learn from this lecture?

- **All classifiers are vulnerable**
  - Nature of the problem. As long as your perturbation is strong enough, you can fool the classifier
  - Especially true when the dimension of the data is high
- **There is a trade off between accuracy and robustness**
  - You need to trade the two through optimization
  - More general than adv. training, but still along the same line
  - Computational cost is as high as adversarial training

Some general advice for students

- The worst research project today is to develop new attack / defense.
- The trade-off is interesting but kind of expectable.
- The more difficult question is to go beyond the ℓ_p-ball.
- Much more valuable: Improve natural accuracy in different environment, not customized attack.
- If you want to defend attacks, defend new attacks that you have not seen, at scale.