Adversarial Robustness Theory and Algorithms

Genghis Luo, Jackie Chen, Daniel Jin

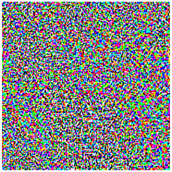
New York University

17 December, 2024

Adversarial Examples: An Illustration

How vulnerable are deep neural networks to small, imperceptible changes?

$+ .007 \times$



$=$

$\boldsymbol{x}$
"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Source: [GSS15]

Observation: Small perturbations can deceive the model completely!

Challenges in Adversarial Robustness

- Very small changes to the input image can fool state-of-the-art neural networks with high probability **[GSS15]**
- Existing defenses are often bypassed by stronger, adaptive adversaries **[CW17]**
- Theoretical guarantees for robustness remain limited **[PMW^{+}16]**

How can we learn models robust to adversarial inputs?

Overview of Key Papers

- **Paper 1 [Ma18]:** Towards Deep Learning Models Resistant to Adversarial Attacks
  - An Optimization Point of View.
- **Paper 2 [ZYJ+19]:** Theoretically Principled Trade-off between Robustness and Accuracy
  - Formalized the trade-off between adversarial robustness and standard accuracy.
  - Presented a mathematical framework to analyze this trade-off.
- **Remark [AMMZ23]:** H-consistency
  - Ensures that surrogate losses remain consistent with the classification loss.
  - A critical property for robust surrogate loss functions.

Goal: A comprehensive revisit of key theories, proofs, and connections between papers.

Notations

Notation	Description
$\mathcal{D}$	Data distribution over $(\boldsymbol{x}, y)$ pairs
$\mathcal{L}(\theta, x, y)$	Loss function with model parameters θ
$\mathcal{S} \subseteq \mathbb{R}^d$	Set of allowable adversarial perturbations
x^{adv}	Adversarial example generated from input x
$\mathbb{B}(x, \epsilon)$	ℓ_∞-ball around x: $\{x' \in \mathcal{X} : \|x' - x\|_\infty \leq \epsilon\}$
$\rho(\theta)$	Adversarial loss: $\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\mathcal{S}} \mathcal{L}(\theta, x+\delta, y)\right]$
θ	Model parameters to be optimized

Motivation: Why ERM Fails Against Adversarial Examples

Empirical risk minimization (ERM) has been the cornerstone of machine learning, defined as follows:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \mathcal{L}(\theta, x, y),$$

where $\mathcal{L}(\theta, x, y)$ is a loss function for a neural network parameterized by θ.

Observation: Despite its success, ERM fails to provide robustness against adversarial examples:

$$x^{\mathrm{adv}} = x + \delta \quad \text{such that} \quad \|\delta\| \le \epsilon,\ f(x^{\mathrm{adv}}) \neq y,$$

where δ represents an imperceptible perturbation constrained within an ℓ_∞-ball. These examples are misclassified even though they remain visually similar to x.

Robust Optimization: A Solution to Adversarial Vulnerabilities

To address the limitations of ERM, adversarial robustness can be formalized through a robust optimization framework. Instead of minimizing the loss on the original inputs x, we consider the worst-case adversarial perturbations within a given threat model $\mathcal{S}$:

$$\min_{\theta} \rho(\theta), \quad \text{where } \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in \mathcal{S}} \mathcal{L}(\theta, x + \delta, y) \right].$$

Key Components:

- *Threat Model:* Defines the set of allowable perturbations $\mathcal{S} \subseteq \mathbb{R}^d$.
- *Adversarial Loss:* Measures the model's performance under the worst-case perturbation δ.
- *Saddle-Point Problem:* Balances the adversary's goal to maximize the loss and the learner's goal to minimize it.

Gradients from Attacks and Saddle Point Optimization

To solve the robust optimization problem using SGD, we face two challenges:

- $\rho(\theta)$ involves an inner *maximization* problem.
- Standard backpropagation cannot be applied directly.

In practice, Both the gradients and the value of $\rho(\theta)$ will be computed using sampled input points. Therefore, we can consider, without loss of generality, the case of a single random example x with label y, in which case the problem becomes:

$$\min_{\theta} \max_{\delta \in \mathcal{S}} g(\theta, \delta), \quad \text{where } g(\theta, \delta) = \mathcal{L}(\theta, x + \delta, y).$$

If we assume that the loss $\mathcal{L}$ is continuously differentiable in θ, we can compute a descent direction for θ by utilizing the classical theorem of Danskin.

Theorem and Corollary for Saddle Point Optimization

Theorem C.1 (Danskin)

Let S be nonempty compact topological space and $g : \mathbb{R}^n \times S \to \mathbb{R}$ be such that $g(\cdot, \delta)$ is differentiable for every $\delta \in S$ and $\nabla g(\theta, \delta)$ is continuous on $\mathbb{R}^n \times S$. Also, let $\delta^*(\theta) = \{\delta \in \arg\max_{\delta \in S} g(\theta, \delta)\}$. Then the corresponding max-function:

$$\phi(\theta) = \max_{\delta \in S} g(\theta, \delta)$$

is locally Lipschitz continuous, directionally differentiable, and its directional derivatives satisfy:

$$\phi'(\theta, h) = \sup_{\delta \in \delta^*(\theta)} h^T \nabla_\theta g(\theta, \delta).$$

Theorem and Corollary for Saddle Point Optimization

In particular, if for some $\theta \in \mathbb{R}^n$ the set $\delta^*(\theta)$ is a singleton, the max-function is differentiable at θ and:

$$\nabla \phi(\theta) = \nabla_\theta g(\theta, \delta^*_\theta).$$

Intuition: since gradients are local objects, the function $\phi(\theta)$ is locally the same as $g(\theta, \delta^*_\theta)$, where δ^*_θ is the optimizer of the inner problem. Therefore, their gradients will be the same.

Theorem and Corollary for Saddle Point Optimization

Corollary C.2

Let $\bar{\delta}$ be such that $\bar{\delta} \in S$ and is a maximizer for

$$\max_{\delta \in S} L(\theta, x + \delta, y).$$

Then, as long as it is nonzero,

$$-\nabla_\theta L(\theta, x + \bar{\delta}, y)$$

is a descent direction for

$$\phi(\theta) = \max_{\delta \in S} L(\theta, x + \delta, y).$$

Theorem and Corollary for Saddle Point Optimization

Proof

We apply Theorem C.1 to $g(\theta, \delta) := L(\theta, x + \delta, y)$ and $S = B_{\|\cdot\|}(\epsilon)$, where $B_{\|\cdot\|}(\epsilon)$ denotes the ball of radius ϵ under a given norm. By Theorem C.1, the directional derivative of $\phi(\theta)$ in the direction of $h = \nabla_\theta L(\theta, x + \bar{\delta}, y)$ satisfies:

$$\phi'(\theta, h) = \sup_{\delta \in \delta^*(\theta)} h^T \nabla_\theta L(\theta, x + \delta, y)$$

$$\geq h^T h = \|\nabla_\theta L(\theta, x + \bar{\delta}, y)\|_2^2 \geq 0.$$

If the gradient is nonzero, then the inequality is strict, and the negative gradient $-\nabla_\theta L(\theta, x + \bar{\delta}, y)$ provides a descent direction for $\phi(\theta)$.

Theorem and Corollary for Saddle Point Optimization

Claim

For continuously differentiable functions, gradients at maximizers of the inner problem correspond to descent directions for the saddle point problem, the gradient is :

$$\nabla_\theta \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\nabla_\theta \mathcal{L}(\theta, x + \delta^*(\theta), y)\right],$$

where $\delta^*(\theta)$ solves the inner maximization:

$$\delta^*(\theta) = \arg\max_{\delta\in\mathcal{S}} \mathcal{L}(\theta, x + \delta, y).$$

Technical Challenges: Non-Differentiability of the Loss

Issue 1: ReLU and Max-Pooling Units

- Neural network architectures often include **ReLU** and **max-pooling** units.
- These components cause the loss function to be *not continuously differentiable*.

Key Insight:

- The set of discontinuities has **measure zero**.
- In practice, this issue is negligible since problematic points are rarely encountered.

Conclusion: Non-differentiability does not pose significant challenges during optimization.

Technical Challenges: Non-Concavity of the Inner Problem

Issue 2: Non-Concavity of the Inner Maximization Problem

- The inner maximization problem is **not concave**, making global maximizers hard to compute.

Proposed Solution:

- Consider a subset $S' \subseteq S$ where the local maximum is a global maximum in S'.
- Applying the theorem on S' ensures the gradient still provides a descent direction for the saddle point problem.

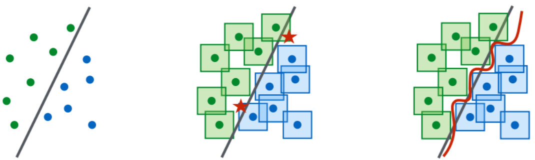
Practical Implication:

- If the inner maximum corresponds to a true adversarial example, **SGD** using this gradient will *decrease the loss*, improving model robustness.

Network Capacity and the Complexity of Decision Boundaries

Key Observation:

- Adversarial examples significantly alter the decision boundary.
- Simple linear boundaries fail to separate perturbed regions (middle figure).
- Increasing model capacity enables learning of complex decision boundaries to address adversarial perturbations (right figure).



The decision boundary becomes increasingly complex

Insight

Optimization Formulation:

$$\min_{\theta} \rho(\theta), \quad \text{where } \rho(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\mathcal{S}} \mathcal{L}(\theta, x+\delta, y)\right].$$

New Insight: This optimization formulation focuses exclusively on adversarial examples, neglecting the original data distribution.

- As a consequence, the model is inherently forced to **"overfit"**.
- This leads to unnecessarily complex decision boundaries and requires increased model capacity.

Is this approach fundamentally reasonable, or does it highlight an inherent trade-off between robustness and simplicity?

Notations I

Notation	Description
$\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{x}, \boldsymbol{y}$	Random vectors and their realizations
X, Y, x, y	Random variables and their realizations
$\mathcal{X}$	Sample space where $\mathcal{X} \subseteq \mathbb{R}^d$
$\text{sign}(x)$	Sign of scalar x, with $\text{sign}(0) = +1$
$\mathbf{1}_{\{\text{event}\}}$	Indicator function: 1 if an event happens, 0 otherwise
$\lVert \boldsymbol{x} \rVert$	Generic norm (if not specified)
$f : \mathcal{X} \to \mathbb{R}$	Score function mapping an instance to a prediction
$\mathbb{B}(\boldsymbol{x}, \epsilon)$	Neighborhood of $\boldsymbol{x}$: $\{\boldsymbol{x}' \in \mathcal{X} : \lVert \boldsymbol{x}' - \boldsymbol{x} \rVert \le \epsilon\}$

Notations II

Notation	Description
$\mathrm{DB}(f)$	Decision boundary of f: $\{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) = 0\}$
$\mathbb{B}(\mathrm{DB}(f), \epsilon)$	$\{\boldsymbol{x} \in \mathcal{X} : \exists \boldsymbol{x}' \in \mathbb{B}(\boldsymbol{x}, \epsilon)$ s.t. $f(\boldsymbol{x})f(\boldsymbol{x}') \leq 0\}$
$\psi^*(\boldsymbol{v})$	Conjugate function: $\sup_{\boldsymbol{u}}\{\boldsymbol{u}^T\boldsymbol{v} - \psi(\boldsymbol{u})\}$
ψ^{**}	Bi-conjugate of ψ
$\phi(\cdot)$	Surrogate of 0-1 loss

Problem Setup: Robust (Classification) Error

In the context of adversarial learning in binary classification, a set of instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ and labels $y_1, \ldots, y_n \in \{-1, +1\}$ is given. Assume that $(\boldsymbol{X}, Y) \sim \mathcal{D}$ with $\mathcal{D}$ unknown.
Define $\mathcal{R}_{\text{rob}}$ to characterize the robustness of a score function $f : \mathcal{X} \to \mathbb{R}$ by:

$$\mathcal{R}_{\text{rob}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}} \mathbf{1}_{\{\exists \boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon) \text{ s.t. } f(\boldsymbol{X}')Y \leq 0\}}$$

Write the natural generalization error as:

$$\mathcal{R}_{\text{nat}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}} \mathbf{1}_{\{f(\boldsymbol{X})Y \leq 0\}}$$

Note: The two errors satisfy $\mathcal{R}_{\text{rob}}(f) \geq \mathcal{R}_{\text{nat}}(f)$ for all f the robust error is equal to the natural error when $\epsilon = 0$.
Introduce the boundary error defined as:

$$\mathcal{R}_{\text{bdy}}(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}} \mathbf{1}_{\{\boldsymbol{X} \in \mathbb{B}(\mathrm{DB}(f),\epsilon), f(\boldsymbol{X})Y > 0\}}$$

Problem Setup: Key Relation

Claim

The following decomposition of $\mathcal{R}_{\text{rob}}(f)$ holds by definition:

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

Proof (Sketch)

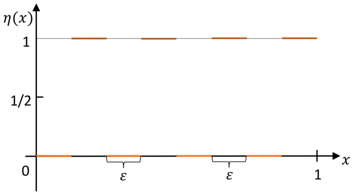
It is quite obvious since the first term $\mathcal{R}_{\text{nat}}(f)$ includes all misclassified points, and the second term $\mathcal{R}_{\text{bdy}}(f)$ includes all the points that are classified correctly but within $\mathbb{B}(\mathrm{DB}(f), \epsilon)$.

Toy Trade-off Example: Trade-off Between $\mathcal{R}_{\text{nat}}(f)$ and $\mathcal{R}_{\text{bdy}}(f)$

Toy Example [BJM06]: Trade-off Between $\mathcal{R}_{\text{nat}}(f)$ and $\mathcal{R}_{\text{bdy}}(f)$

Consider the case $(X, Y) \sim \mathcal{D}$, where the marginal distribution over the sample space $\mathcal{X}$ is a uniform distribution over $[0, 1]$, and for $k = 0, 1, \ldots, \lceil \frac{1}{2\epsilon} - 1 \rceil$,

$$\begin{aligned} \eta(x) &:= \Pr(Y = 1 \mid X = x) \\ &= \begin{cases} 0, & x \in [2k\epsilon, (2k+1)\epsilon) \\ 1, & x \in [(2k+1)\epsilon, (2k+2)\epsilon) \end{cases} \end{aligned}$$



	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)

Construction of Classification-calibrated Surrogate Loss

Introduce the **surrogate loss** $\mathcal{R}_\phi(f) := \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}}\phi(f(\boldsymbol{X})Y)$
Formally, for $\eta \in [0,1]$, define the **conditional ϕ-risk** by

$$H(\eta) := \inf_{\alpha\in\mathbb{R}} C_\eta(\alpha) := \inf_{\alpha\in\mathbb{R}}(\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)),$$

and define $H^-(\eta) := \inf_{\alpha(2\eta-1)\le 0} C_\eta(\alpha)$.
The classification-calibrated condition requires that imposing the constraint that α has an inconsistent sign with the Bayes decision rule $\mathrm{sign}(2\eta-1)$ leads to a strictly larger ϕ-risk:

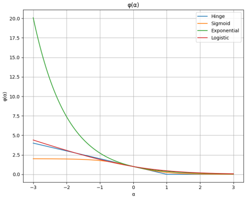
Assumption 1: Classification-Calibrated Condition

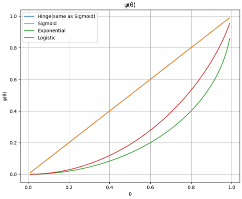
Assume that the surrogate loss ϕ is classification-calibrated, meaning that for any $\eta \neq 1/2$, $H^-(\eta) > H(\eta)$, i.e., Bayesian estimator is always the minimizer.

Construction of Classification-calibrated Surrogate Loss: Examples

Table 1: Examples of classification-calibrated loss ϕ and associated ψ-transform.

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	$\log_2(1+\exp(-\alpha))$	$\psi_{\log}(\theta)$





Construction of Classification-calibrated Surrogate Loss: Properties

Define the ψ transform of classification-calibrated surrogate loss ϕ: $[0,1] \to [0,\infty)$ by

$$\psi = \widetilde{\psi}^{**}$$

where $\widetilde{\psi}(\theta) := H^{-}\left(\frac{1+\theta}{2}\right) - H\left(\frac{1+\theta}{2}\right)$.

In fact, the function $\psi(\theta)$ is the largest convex lower bound on $\tilde{\psi}$.

The value $H^{-}\left(\frac{1+\theta}{2}\right) - H\left(\frac{1+\theta}{2}\right)$ characterizes how close the surrogate loss ϕ is to the class of non-classification-calibrated losses.

Lemma 2.1 [BJM06]

Under Assumption 1, the function ψ has the following properties: ψ is non-decreasing, continuous, convex on $[0,1]$ and $\psi(0) = 0$.

Guarantee on C.c. Surrogate Loss Minimization: Upper Bound

Theorem 3.1 [ZYJ+19]

Let $\mathcal{R}_\phi(f) := \mathbb{E}_\phi[f(\mathbf{X})Y]$ and $R^*_\phi := \min_f \mathcal{R}_\phi(f)$. Under Assumption 1, for any non-negative loss function ϕ such that $\phi(0) \geq 1$, any measurable $f : \mathcal{X} \to \mathbb{R}$, any probability distribution on $\mathcal{X} \times \{\pm 1\}$, and any $\lambda > 0$, we have:

$$\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}} = \mathcal{R}_{\text{nat}}(f) - \mathcal{R}^*_{\text{nat}} + \mathcal{R}_{\text{bdy}}(f)$$

$$\leq \psi^{-1}(\mathcal{R}_\phi(f) - \mathcal{R}^*_\phi) + \Pr[\mathbf{X} \in \mathbb{B}(\mathrm{DB}(f), \epsilon),\ f(\mathbf{X})Y > 0]$$

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

Proof of Theorem 3.1

Claim: Classification-Calibration [BJM06]

$$\psi(R_{\text{nat}}(f) - R^*_{\text{nat}}) \le R_\phi(f) - R^*_\phi.$$

Proof.

$$R_{\text{nat}}(f) - R^*_{\text{nat}} = R_{\text{nat}}(f) - R(\eta - \frac{1}{2})$$

$$\overset{①}{=} \mathbb{E}\left[\mathbf{1}_{\{\text{sign}(f(x))\neq\text{sign}\left(\eta(X)-\frac{1}{2}\right)\}} \cdot |2\eta(X) - 1|\right]$$

where ① is because $|(1-\eta)-\eta| = |2\eta - 1|$. Therefore,

$$\psi\left(R_{\text{nat}}\left(f\right) - R^*_{\text{nat}}\right) \overset{②}{\le} \mathbb{E}\left[\psi\left(\mathbf{1}_{\{\text{sign}(f(x))\neq\text{sign}\left(g(x)-\frac{1}{2}\right)\}} \cdot |2g(x) - 1|\right)\right]$$

$$\overset{③}{=} \mathbb{E}\left[\mathbf{1}_{\{\text{sign}(f(X))\neq\text{sign}\left(n(x)-\frac{1}{2}\right)\}} \cdot \varphi(|2n(X) - 1|)\right]$$

Proof of Theorem 3.1 (continued)

$$
\begin{aligned}
\psi(R_{\text{nat}}(f) - R^*_{\text{nat}}) &\overset{③}{=} \mathbb{E}[\mathbf{1}_{\{sign(f(X)) \neq sign(\eta(X) - \frac{1}{2})\}} \cdot \varphi(|2\eta(X) - 1|)] \\
&\overset{④}{\leq} \mathbb{E}[\mathbf{1}_{\{sign(f(X)) \neq sign(\eta(X) - \frac{1}{2})\}} \cdot \tilde{\psi}(|2g(X) - 1|)] \\
= \mathbb{E}[\mathbf{1}_{\{sign(f(X)) \neq sign(\eta(X) - \frac{1}{2})\}} &\cdot \left(\inf_{\alpha : \alpha(2\eta(X) - 1) \leq 0} C_{\eta(X)}(\alpha) - H(\eta(X)) \right) \\
&\overset{⑤}{\leq} \mathbb{E}[C_{g(X)}(f(X)) - H(g(X))] \\
&\overset{⑥}{=} R_\phi(f) - R^*_\phi.
\end{aligned}
$$

where ② is because of Jensen's Inequality, ③ is by $\psi(0) = 0$, ④ is by ψ being the convex lower bound of $\widetilde{\psi}$, ⑤ is because when $\text{sign}(f(X)) \neq \text{sign}\left(\eta(X) - \frac{1}{2}\right)$, $f(X)$ is a valid α for H^{-1}; otherwise clear, and ⑥ is because $\mathbb{E}[C_{\eta(X)}] = \mathcal{R}_\phi(f)$. □

Proof of Theorem 3.1 (continued)

By the claim, we can directly prove Theorem 3.1:
Proof.

$$
\begin{aligned}
\mathcal{R}_{\text{rob}}(f) - \mathcal{R}_{\text{nat}}^* &= \mathcal{R}_{\text{nat}}(f) - \mathcal{R}_{\text{nat}}^* + \mathcal{R}_{\text{bdy}}(f) \\
&\leq \psi^{-1}\left(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*\right) + \mathbb{E}_{(\boldsymbol{X},Y)\sim\mathcal{D}} \mathbf{1}_{\{\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon), f(\boldsymbol{X})Y>0\}} \\
&= \psi^{-1}\left(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*\right) + \Pr[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon), f(\boldsymbol{X})Y>0] \\
&\leq \psi^{-1}\left(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*\right) + \Pr[\boldsymbol{X}\in\mathbb{B}(\mathrm{DB}(f),\epsilon)] \\
&= \psi^{-1}\left(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*\right) + \mathbb{E} \max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)} 1\left\{f\left(\boldsymbol{X}'\right) f(\boldsymbol{X}) \leq 0\right\} \\
&= \psi^{-1}\left(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*\right) + \mathbb{E} \max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)} 1\left\{f\left(\boldsymbol{X}'\right) f(\boldsymbol{X})/\lambda \leq 0\right\} \\
&\leq \psi^{-1}\left(\mathcal{R}_\phi(f) - \mathcal{R}_\phi^*\right) + \mathbb{E} \max_{\boldsymbol{X}'\in\mathbb{B}(\boldsymbol{X},\epsilon)} \phi\left(f\left(\boldsymbol{X}'\right) f(\boldsymbol{X})/\lambda\right)
\end{aligned}
$$

□

Guarantee on C.c. Surrogate Loss Minimization: Lower Bound

Theorem 3.2 [ZYJ+19]

Suppose that $|\mathcal{X}| \geq 2$. Under Assumption 1, for any non-negative loss function ϕ such that $\phi(x) \to 0$ as $x \to +\infty$, any $\xi > 0$, and any $\theta \in [0,1]$, there exists a probability distribution on $\mathcal{X} \times \{\pm 1\}$, a function $f : \mathbb{R}^d \to \mathbb{R}$, and a regularization parameter $\lambda > 0$ such that $\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}} = \theta$ and

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

$$\leq \psi\left(\theta - \mathbb{E} \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon)} \phi\left(f\left(\boldsymbol{X}'\right) f(\boldsymbol{X})/\lambda\right)\right) + \xi$$

Interpretation of Theorem 3.2

Theorem 3.2 demonstrates that in the presence of extra conditions on the loss function, i.e., $\lim_{x \to +\infty} \phi(x) = 0$, the upper bound in Theorem 3.1 is tight. The condition holds for all the losses in Table 2.

TRADES Algorithm [ZYJ+19]: Optimization on Upper Bound

TRADES Algorithm [ZYJ+19]: Optimization on Upper Bound

Theorems 3.1 and 3.2 shed light on algorithmic designs of adversarial defenses. In order to minimize $\mathcal{R}_{\text{rob}}(f) - \mathcal{R}^*_{\text{nat}}$, the theorems suggest minimizing [a]

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

[a] For simplicity of implementation, we do not use the function ψ^{-1} and rely on λ to approximately reflect the effect of ψ^{-1}, the trade-off between the natural error and the boundary error, and the tight approximation of the boundary error using the corresponding surrogate loss function.

TRADES Algorithm: Heuristic Extension to Multi-class Classification

Heuristically, **[ZYJ+19]** use two heuristics to achieve more general defenses:
a) extending to multi-class problems by involving multi-class calibrated loss;
b) approximately solving the mini-max problem via alternating gradient descent.
For multi-class problems, a surrogate loss is calibrated if minimizers of the surrogate risk are also minimizers of the $0-1$ risk **[PS16]**. Examples of multi-class calibrated loss include cross-entropy loss. Algorithmically, **[ZYJ+19]** extend the problem to the case of multi-class classifications by replacing ϕ with a multi-class calibrated loss $\mathcal{L}(\cdot,\cdot)$:

$$\min_{f} \mathbb{E}\left\{\mathcal{L}(f(\boldsymbol{X}), \boldsymbol{Y}) + \max_{\boldsymbol{X}' \in \mathbb{B}(\boldsymbol{X},\epsilon)} \mathcal{L}\left(f(\boldsymbol{X}), f\left(\boldsymbol{X}'\right)\right)/\lambda\right\}$$

where $f(\boldsymbol{X})$ is the output vector of learning model (with soft-max operator in the top layer for the cross-entropy loss $\mathcal{L}(\cdot,\cdot)$), $\boldsymbol{Y}$ is the label-indicator vector, and $\lambda > 0$ is the regularization parameter.

TRADES Algorithm: Pseudocode

Algorithm 1 Adversarial training by TRADES

Input: Step sizes η_1 and η_2, batch size m, number of iterations K in inner optimization, network architecture parametrized by θ

Output: Robust network f_θ

Randomly initialize network f_θ, or initialize network with pre-trained configuration

repeat

  Read mini-batch $B = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ from training set

  for $i = 1, ..., m$ (in parallel) **do**

    $\boldsymbol{x}_i' \leftarrow \boldsymbol{x}_i + 0.001 \cdot \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is the Gaussian distribution with zero mean and identity variance

    for $k = 1, ..., K$ **do**

      $\boldsymbol{x}_i' \leftarrow \Pi_{\mathbb{B}(\boldsymbol{x}_i, \epsilon)}(\eta_1 \mathsf{sign}(\nabla_{\boldsymbol{x}_i'} \mathcal{L}(f_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i'))) + \boldsymbol{x}_i')$, where Π is the projection operator

    end for

  end for

  $\theta \leftarrow \theta - \eta_2 \sum_{i=1}^{m} \nabla_\theta [\mathcal{L}(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i) + \mathcal{L}(f_\theta(\boldsymbol{x}_i), f_\theta(\boldsymbol{x}_i'))/\lambda]/m$

until training converged

H-consistency: Motivation

Recall in TRADES, surrogate loss functions were introduced to create nice properties such as differentiability and convexity. And the algorithm is designed to minimize functions containing these surrogate losses.

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

Then is becomes essential to ensure that minimizing the surrogate loss aligns with minimizing the target loss.

H-consistency Bound: Definition

To ensure the surrogate loss aligns with the target loss, an H-**consistency bound** is introduced to connect surrogate loss minimization to target loss minimization.

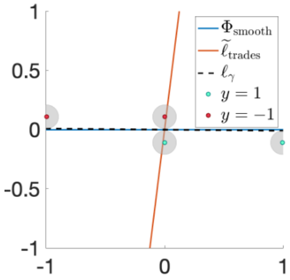
$$\forall h \in H, \quad \mathcal{R}_{\text{target}}(h) - \mathcal{R}_{\text{target},H} \leq f(\mathcal{R}_{\phi}(h) - \mathcal{R}_{\phi,H}),$$

where:

- $f : \mathbb{R}^+ \to \mathbb{R}^+$**: A non-increasing function.**
- $\mathcal{R}_{\text{target}}(h) - \mathcal{R}_{\text{target},H}$: True target loss within H.
- $\mathcal{R}_{\phi}(h) - \mathcal{R}_{\phi,H}$: Surrogate loss within H.

H-consistency Issue in TRADES

It has been proven that TRADES's original surrogate loss does not satisfy the H-consistency bound in certain cases.
Leading to inaccurate hypothesis found in classification tasks, especially in multi-class classification settings.



Smooth Adversarial Losses

A new family of surrogate functions **smooth adversarial losses** was later introduced that satisfy the H-consistency bound. This has lead to the creation of the **PSAL**(Principled Smooth Adversarial Loss) algorithm that optimizes on smooth adversarial losses, allowing it consistently outperform previous methods in both accuracy and robustness.

Method	Dataset	Norm	Maximum magnitude	Clean	$\mathrm{PGD}^{40}_{\mathrm{margin}}$	AutoAttack
Gowal et al. (2020) (WRN-70-16)	CIFAR-10	ℓ_∞	$\gamma = 8/255$	85.34 ± 0.04%	57.90 ± 0.13%	57.05 ± 0.17%
PSAL (WRN-70-16)				**86.63 ± 0.24%**	**59.01 ± 0.13%**	**57.46 ± 0.12%**
Gowal et al. (2020) (WRN-34-20)				85.21 ± 0.16%	57.54 ± 0.18%	56.70 ± 0.14%
PSAL (WRN-34-20)				**86.71 ± 0.08%**	**58.68 ± 0.16%**	**57.13 ± 0.18%**
Gowal et al. (2020) (WRN-28-10)				84.33 ± 0.18%	55.92 ± 0.20%	55.19 ± 0.23%
PSAL (WRN-28-10)				**86.07 ± 0.14%**	**57.12 ± 0.19%**	**55.66 ± 0.16%**
Pang et al. (2020) (WRN-34-20)				86.43%	—	54.39%
Rice et al. (2020) (WRN-34-20)				85.34%	—	53.42%
Wu et al. (2020) (WRN-34-10)				85.36%	—	56.17%
Qin et al. (2019) (WRN-40-8)				86.28%	—	52.84%
Xu et al. (2022) (ResNet-32)				80.43%	—	44.15%
Gowal et al. (2020) (WRN-70-16)	CIFAR-100	ℓ_∞	$\gamma = 8/255$	60.56 ± 0.31%	31.39 ± 0.19%	29.93 ± 0.14%
PSAL (WRN-70-16)				**62.25 ± 0.26%**	**34.11 ± 0.17%**	**30.63 ± 0.10%**
Gowal et al. (2020) (WRN-34-20)	SVHN	ℓ_∞	$\gamma = 8/255$	93.03 ± 0.13%	61.01 ± 0.16%	57.84 ± 0.19%
PSAL (WRN-34-20)				**94.31 ± 0.17%**	**63.12 ± 0.14%**	**58.08 ± 0.15%**

Pranjal Awasthi, Anqi Mao, Mehryar Mohri, and Yutao Zhong.
Theoretically Grounded Loss Functions and Algorithms for Adversarial Robustness.
Proceedings of the 26th International Conference on Artificial Intelligence and Statistics, 2023.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe.
Convexity, Classification, and Risk Bounds.
Journal of the American Statistical Association, 101(473):138–156, March 2006.

Nicholas Carlini and David Wagner.
Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods, November 2017.
arXiv:1705.07263 [cs].

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy.
Explaining and Harnessing Adversarial Examples, March 2015.

arXiv:1412.6572 [stat].

Aleksander Ma.
Towards Deep Learning Models Resistant to Adversarial Attacks.
International Conference on Learning Representations, 2018.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami.
Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks, March 2016.
arXiv:1511.04508 [cs].

Bernardo Ávila Pires and Csaba Szepesvári.
Multiclass Classification Calibration Functions, September 2016.
arXiv:1609.06385.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan.

Theoretically Principled Trade-off between Robustness and Accuracy.
Proceedings of the 36th International Conference on Machine Learning, 2019.