Robustness May Be at Odds with Accuracy

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, Aleksander Madry (ICLR 2019)

COSC 594 Special Topics in Cybersecurity: Adversarial Machine Learning

Aukkawut Ammartayakun

University of Tennessee, Knoxville

Table of contents

Long Story Short

Long Story Short

- The more robust your model is, the less the accuracy is.

Long Story Short

- The more robust your model is, the less the accuracy is.
  - Or is it?
- Naively training model using adversarial examples for robustness might not be a wise idea.

Introduction

Review ML

The goal of ML is to minimize the criterion

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[L(x, y; \theta)]$$

Review ML

The goal of ML is to minimize the criterion

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[L(x, y; \theta)]$$

or in a way, minimize *adversarial loss*

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in \Delta} L(x + \delta, y; \theta)\right]$$

Review ML

The goal of ML is to minimize the criterion

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[L(x, y; \theta)]$$

or in a way, minimize *adversarial loss*

$$\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta \in \Delta} L(x + \delta, y; \theta)\right]$$

Will this cost us anything?

Visualization

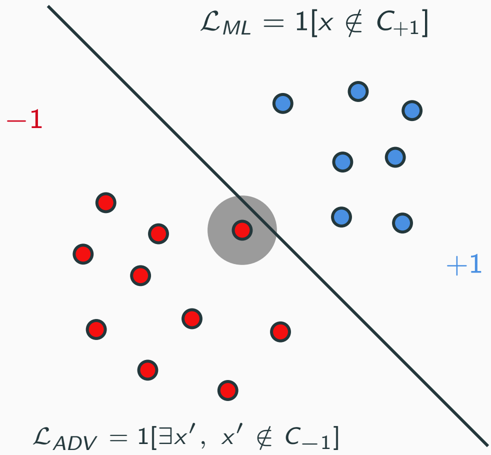

Figure 1: Traditional ML loss VS adversarial loss on binary classification with 0-1 loss

Training with Adversarial Loss

- Clearly we need 2 optimizations

Training with Adversarial Loss

- Clearly we need 2 optimizations (minimax):

Training with Adversarial Loss

- Clearly we need 2 optimizations (minimax):
  - Adversarial Attack (inner max)

Training with Adversarial Loss

- Clearly we need 2 optimizations (minimax):
    - Adversarial Attack (inner max)
    - Model training (outer min)

Training with Adversarial Loss

- Clearly we need 2 optimizations (minimax):
  - Adversarial Attack (inner max)
  - Model training (outer min)
  - We want to, at worst case scenario (maximize), get the robust prediction (minimize)

How can we solve the minimax optimization?

- Danskin's theorem[1] implies that

How can we solve the minimax optimization?

- Danskin's theorem[1] implies that

$$\nabla_\theta \max_{\delta \in \Delta} f(x+\delta, y; \theta) = \nabla_\theta f(x+\delta^*, y; \theta)$$

given that your constraint set is compact (ex. ℓ^p-ball on finite dimensional space is compact) and your loss is a continuous function.

How can we solve the minimax optimization?

- Danskin's theorem[1] implies that

$$\nabla_\theta \max_{\delta \in \Delta} f(x+\delta, y; \theta) = \nabla_\theta f(x+\delta^*, y; \theta)$$

  given that your constraint set is compact (ex. ℓ^p-ball on finite dimensional space is compact) and your loss is a continuous function.
- Even if we can't get the optimal δ^*, we can pretend what we have is good enough and perform gradient descent.

How does that differ compared to normal training?

- One can say this is data augmentation.

How does that differ compared to normal training?

- One can say this is data augmentation.
  - We essentially add (adversarial) points to move the decision boundary.

How does that differ compared to normal training?

- One can say this is data augmentation.
  - We essentially add (adversarial) points to move the decision boundary.

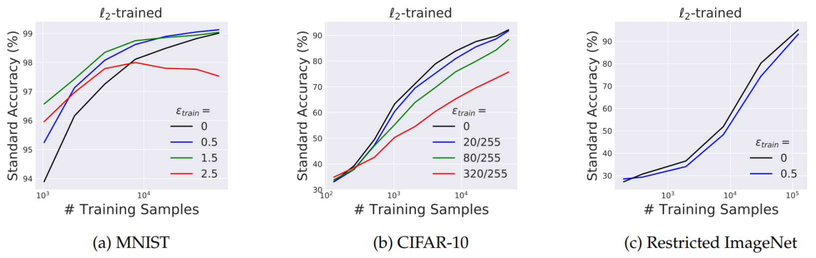

Figure 2: Accuracy across the dataset with different numbers of training samples and different levels of perturbation ε

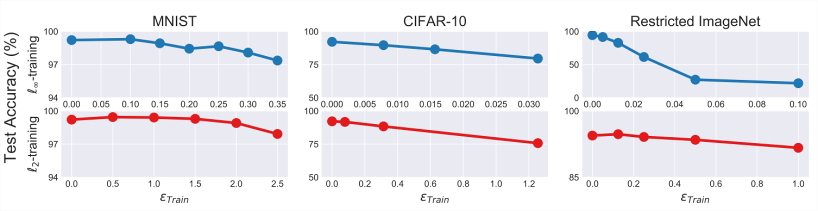

Figure 3: Decrease in accuracy when the strength of perturbation increases

Theoretical Modeling

Classification Problem

- Consider the data generating model for a point (x, y)

Classification Problem

- Consider the data generating model for a point (x, y)

$$
\begin{aligned}
y &\sim 2\,\mathrm{Ber}(0.5) - 1 \\
\xi &\sim 2\,\mathrm{Ber}(p) - 1 \\
x_1 &\sim y\xi \\
x_2, \ldots, x_{d+1} &\overset{\text{iid}}{\sim} \mathcal{N}(\eta y, 1) \\
\eta &\text{ large}
\end{aligned}
$$

Classification Problem

- Consider the data generating model for a point (x, y)

$$\begin{aligned} y &\sim 2\,\mathrm{Ber}(0.5) - 1 \\ \xi &\sim 2\,\mathrm{Ber}(p) - 1 \\ x_1 &\sim y\xi \\ x_2, \ldots, x_{d+1} &\overset{\text{iid}}{\sim} \mathcal{N}(\eta y, 1) \\ \eta &\text{ large} \end{aligned}$$

- Now, consider the simple classification model with a decision rule

$$f(x) = \mathrm{sign}(w^\top x)$$

(linear model essentially)

Classification Problem

- Consider the data generating model for a point (x, y)

$$\begin{aligned} y &\sim 2\,\mathrm{Ber}(0.5) - 1 \\ \xi &\sim 2\,\mathrm{Ber}(p) - 1 \\ x_1 &\sim y\xi \\ x_2, \ldots, x_{d+1} &\overset{\text{iid}}{\sim} \mathcal{N}(\eta y, 1) \\ \eta &\text{ large} \end{aligned}$$

- Now, consider the simple classification model with a decision rule

$$f(x) = \mathrm{sign}(w^\top x)$$

  (linear model essentially)
- Formulate the adversarial problem with this.

Classification Model

- Natural model would be

$$w_n = \begin{bmatrix} 0 & \frac{1}{d} & \dots & \frac{1}{d} \end{bmatrix}$$

(given x_1 weight of 0 and the rest equal weight)

Classification Model

- Natural model would be

$$w_n = \begin{bmatrix} 0 & \frac{1}{d} & \dots & \frac{1}{d} \end{bmatrix}$$

(given x_1 weight of 0 and the rest equal weight)

$$\begin{aligned}
\mathbb{P}[f(x) = y] &= \mathbb{P}\left[y\frac{1}{d}\sum_{k=2}^{d+1} x_k > 0\right] \\
&= \mathbb{P}\left[y\frac{1}{d}\sum_{k=2}^{d+1} \mathcal{N}(\eta y, 1) > 0\right] \\
&= \mathbb{P}\left[y\frac{1}{d}\mathcal{N}\left(\eta d y, d\right) > 0\right] \\
&= \mathbb{P}\left[y\mathcal{N}\left(\eta y, \frac{1}{d}\right) > 0\right] \\
&= \mathbb{P}\left[\mathcal{N}\left(\eta, \frac{1}{d}\right) > 0\right]
\end{aligned}$$

- $\mathbb{P}[\mathcal{N}\left(\eta, \frac{1}{d}\right) > 0] = 1 - \Phi\left(-\eta\sqrt{d}\right)$, since $\Phi(3) \approx 0.9986$, therefore if $\eta \geq \frac{3}{\sqrt{d}}$ then we have over 99% accuracy.

Adversarial Accuracy

- Recall accuracy of natural model

$$\mathbb{P}[f(x) = y] = \mathbb{P}\left[\mathcal{N}\left(\eta, \frac{1}{d}\right) > 0\right]$$

Adversarial Accuracy

- Recall accuracy of natural model

$$\mathbb{P}[f(x) = y] = \mathbb{P}\left[\mathcal{N}\left(\eta, \frac{1}{d}\right) > 0\right]$$

- Suppose we perturb those point under ℓ^∞ with $\varepsilon = [0, 2\eta, \ldots, 2\eta]$, then the accuracy is

Adversarial Accuracy

- Recall accuracy of natural model

$$\mathbb{P}[f(x) = y] = \mathbb{P}\left[\mathcal{N}\left(\eta, \frac{1}{d}\right) > 0\right]$$

- Suppose we perturb those point under ℓ^∞ with $\varepsilon = [0, 2\eta, \ldots, 2\eta]$, then the accuracy is

$$\begin{aligned}\mathbb{P}[f(x - \varepsilon) = y] &= \mathbb{P}\left[\mathcal{N}\left(\eta, \frac{1}{d}\right) - \epsilon > 0\right] \\ &= \mathbb{P}\left[\mathcal{N}\left(-\eta, \frac{1}{d}\right) > 0\right]\end{aligned}$$

Adversarial Accuracy

- Recall accuracy of natural model

$$\mathbb{P}[f(x) = y] = \mathbb{P}\left[\mathcal{N}\left(\eta, \frac{1}{d}\right) > 0\right]$$

- Suppose we perturb those point under ℓ^∞ with $\varepsilon = [0, 2\eta, \dots, 2\eta]$, then the accuracy is

$$\begin{aligned}\mathbb{P}[f(x - \varepsilon) = y] &= \mathbb{P}\left[\mathcal{N}\left(\eta, \frac{1}{d}\right) - \epsilon > 0\right] \\ &= \mathbb{P}\left[\mathcal{N}\left(-\eta, \frac{1}{d}\right) > 0\right]\end{aligned}$$

- Then, if $\eta \geq \frac{3}{\sqrt{d}}$, we can't get accuracy under this adversarial more than $100\% - 99\% = 1\%$

Visualization

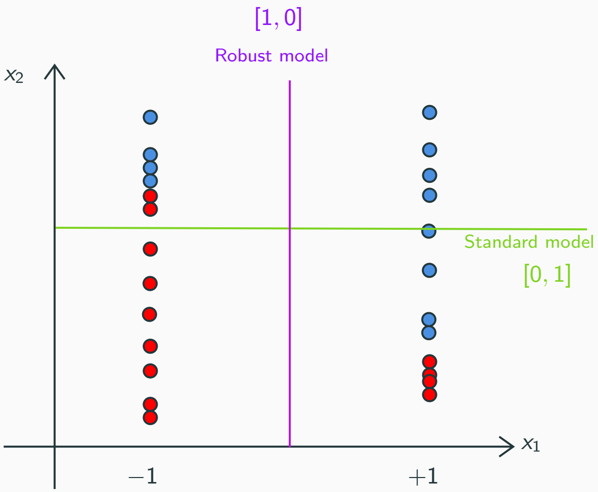

Figure 4: Decision Boundary of the standard and robust model on $d = 1$

Empirical Analysis

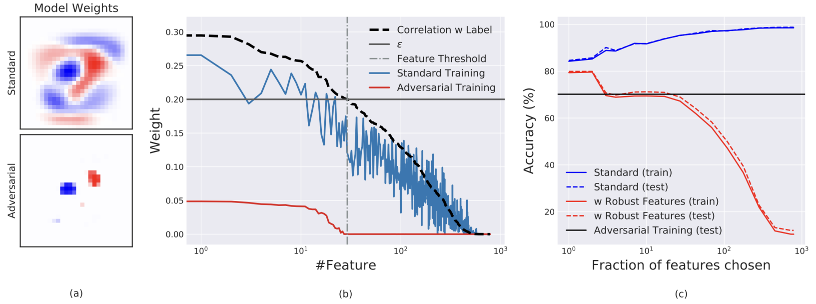

Figure 5: Empirical analysis on binary classification task of MNIST (5 and 7). Adversarial training indeed uses strongly correlated (i.e., robust feature) weight more and ignores the weakly correlated feature.

What does that tell us?

- Under a robust model, the only feature we can use is x_1.

What does that tell us?

- Under a robust model, the only feature we can use is x_1.
  - Because, say in ℓ^∞, one can perturb the "weakly correlated" features towards an anti-correlation direction.

What does that tell us?

- Under a robust model, the only feature we can use is x_1.
  - Because, say in ℓ^∞, one can perturb the "weakly correlated" features towards an anti-correlation direction.
  - That is $x_2, \ldots, x_{d+1}$ are subject to the adversarial attack while x_1 is invariant.

What does that tell us?

- Under a robust model, the only feature we can use is x_1.
  - Because, say in ℓ^∞, one can perturb the "weakly correlated" features towards an anti-correlation direction.
  - That is $x_2, \ldots, x_{d+1}$ are subject to the adversarial attack while x_1 is invariant.
- Therefore, the robust accuracy is at most bounded by a function of $p = \mathbb{P}[\xi = 1]$

Semantics under Embeddings

“Unexpected Results”

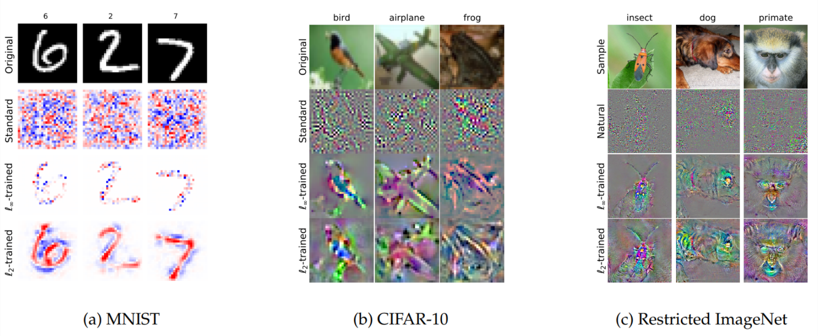

Figure 6: Loss gradient with respect to pixel value shows some “semantic” of the image

Perturbation under large ε

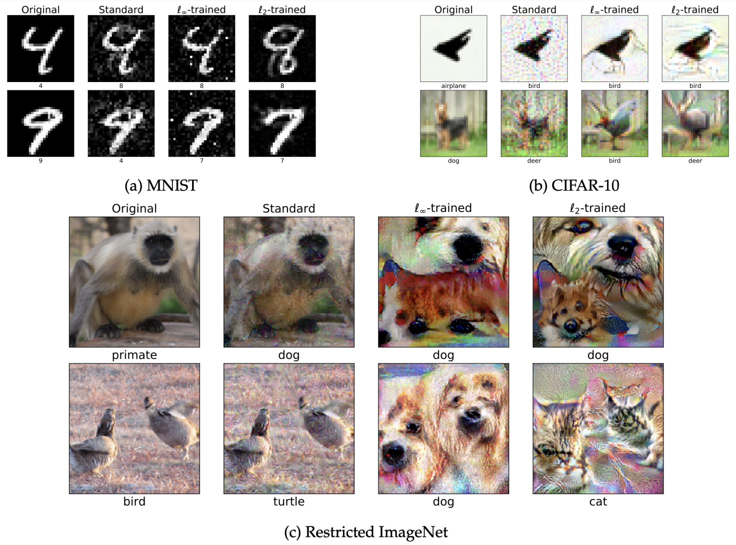

Figure 7: Large ε example on standard and robust model. One can see that the example from robust model is semantically within the target class.

Interpolation

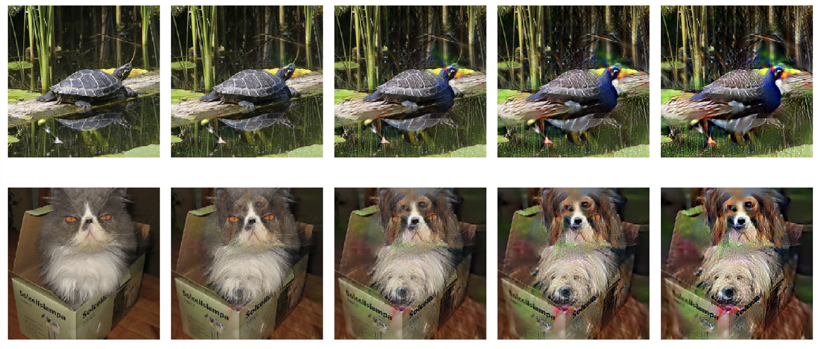

Figure 8: GAN-like interpolation effect from large ε example

Not so unexpected explanation

- As discussed under the robust model, the only feature we can use is x_1.

Not so unexpected explanation

- As discussed under the robust model, the only feature we can use is x_1.
    - Then, in a way, robust models are forced to learn the invariant aspect of the data.
    - In a way, you are required to perturb the invariant direction in order to get an adversarial example under the robust model.
    - This direction of perturbation seems to be smooth, akin to GAN.

Tradeoff

Long Story Short

- More sample complexity

Long Story Short

- More sample complexity
    - More data required to make the robust model.

Long Story Short

- More sample complexity
  - More data required to make the robust model.
- More time complexity

Long Story Short

- More sample complexity
  - More data required to make the robust model.
- More time complexity
  - You need to calculate the minimax problem (two optimizations) instead of just the minimization problem

Robustness Accuracy Tradeoff

Theorem (Robustness-accuracy trade-off)

Any classifier that attains at least $1-\delta$ standard accuracy on $\mathcal{D}$ has robust accuracy at most $\frac{p}{1-p}\delta$ against an ℓ^∞-bounded adversary with $\varepsilon \geq 2\eta$

Proof.

- Let G_+ be the distribution of $x_2, \ldots, x_{d+1}$ when $y = +1$ and G_- for otherwise. Fix $\varepsilon = 2\eta$.
- Let

$$\begin{aligned} p_{++} &= \mathbb{P}_{x_2,\ldots,x_{d+1}\sim G_+}[f(x) = 1 | x_1 = 1] \\ p_{-+} &= \mathbb{P}_{x_2,\ldots,x_{d+1}\sim G_+}[f(x) = 1 | x_1 = -1] \\ p_{--} &= \mathbb{P}_{x_2,\ldots,x_{d+1}\sim G_-}[f(x) = 1 | x_1 = -1] \\ p_{+-} &= \mathbb{P}_{x_2,\ldots,x_{d+1}\sim G_-}[f(x) = 1 | x_1 = 1] \end{aligned}$$

- Then, from law of total probability

$$\begin{aligned} \mathbb{P}[f(x) = y] = \mathbb{P}[y = +1]\mathbb{P}[f(x) = 1 | y = +1] \\ + \mathbb{P}[y = -1]\mathbb{P}[f(x) = -1 | y = -1] \end{aligned}$$

$\square$

Proof

Proof (con't).

- Standard accuracy

$$\begin{aligned}\mathbb{P}[f(x) = y] &= \mathbb{P}[y = +1]\,(pp_{++} + (1-p)p_{-+}) \\ &\quad + \mathbb{P}[y = -1]\,(p(1-p_{--}) + (1-p)(1-p_{+-})) \\ &= \frac{1}{2}\,[p(1 + p_{++} - p_{--}) + (1-p)(1 + p_{-+} - p_{+-})]\end{aligned}$$

- Now, for adversarial accuracy

$$\begin{aligned}\mathbb{P}[f(x-\varepsilon) = y] &= \mathbb{P}[y = +1]\,(pp_{+-} + (1-p)p_{--}) \\ &\quad + \mathbb{P}[y = -1]\,(p(1-p_{-+}) + (1-p)(1-p_{++})) \\ &= \frac{1}{2}\,[p(1 + p_{+-} - p_{-+}) + (1-p)(1 + p_{--} - p_{++})]\end{aligned}$$

- Let $a = 1 + p_{--} - p_{++}$, $b = 1 + p_{+-} - p_{-+}$ then,

$$\mathbb{P}[f(x-\varepsilon) = y] = \frac{1}{2}(pb + (1-p)a) \leq 2\delta$$

(since $\mathbb{P}[f(x) = y] = 1 - \mathbb{P}[f(x-\varepsilon) = y] \geq 1 - \delta$)

$\square$

Proof

Proof (con't).

- Since $a \geq 0$,

$$\begin{aligned}\frac{1}{2}(pa + (1-p)b) &\leq \frac{1}{2}\left(pb + (1-p)\left[\frac{p}{1+p}\right]^2 a\right) \\ &= \frac{p}{2(1-p)}\left((1-p)b + pa\right) \\ &\leq \frac{p}{1-p}\delta\end{aligned}$$

$\square$

- Optimistically, if you have a good standard model, then the robust accuracy will be small.

$$\liminf_{\delta \to 0} \frac{p}{1-p}\delta = 0$$

(lim sup is 1; hence limit does not exist).

- Moreover, it is tight in the sense that if $\delta = 1 - p$ (only first feature), then the robust accuracy is indeed p.

SVM

Theorem

For $\eta \geq 4/\sqrt{d}$ and $p \leq 0.975$ (the first feature is not perfect), a soft-margin SVM classifier of unit weight norm minimizing the distributional loss achieves a standard accuracy of $> 99\%$ and adversarial accuracy of $< 1\%$ against an ℓ^∞-bounded adversary of $\varepsilon \geq 2\eta$. Minimizing the distributional adversarial loss instead leads to a robust classifier that has standard and adversarial accuracy of p against any $\varepsilon < 1$

Theorem
For $\eta \geq 4/\sqrt{d}$ and $p \leq 0.975$ (the first feature is not perfect), a soft-margin SVM classifier of unit weight norm minimizing the distributional loss achieves a standard accuracy of $> 99\%$ and adversarial accuracy of $< 1\%$ against an ℓ^∞-bounded adversary of $\varepsilon \geq 2\eta$. Minimizing the distributional adversarial loss instead leads to a robust classifier that has standard and adversarial accuracy of p against any $\varepsilon < 1$

Implication: Adversarial training is necessary to achieve non-trivial adversarial accuracy

Proof

Theorem
Adversarial training is necessary to achieve non-trivial adversarial accuracy

Proof.

- Recall the soft-margin SVM training

$$\min_{w} \underbrace{\mathbb{E}\left[\max\left\{0, 1 - yw^\top x\right\}\right]}_{\text{margin}} + \overbrace{\frac{1}{2}\lambda\|w\|_2^2}^{\text{regularization}}$$

- Assume we use λ such that $\|w\|_2^2 = 1$

Lemma
Optimal solution w^* *has* $w_i = w_j, \forall i, j \in \{2, \ldots, d+1\}$

- This lemma is true due to exchangability and convexity of the loss with respect to w (more detail on paper).

$\square$

Proof (con't).

- Since the weight is equal, we can define "meta-feature"

$$z = \frac{1}{\sqrt{d}} \sum_{k=2}^{d+1} x_k$$

- Then, we can decompose the sum as $w^\top x = w_1 x_1 + \nu z$
- $z \sim \mathcal{N}(\eta y \sqrt{d}, 1) \overset{\eta = 4/\sqrt{d}}{\Longrightarrow} \nu z \sim \mathcal{N}(4y\nu, \nu^2)$
- **Claim**: $\nu^* \geq \frac{1}{\sqrt{2}}$ (weight on meta-feature should be at least equal to w_1)
- If this claim is not true, then

$$\mathbb{E}\left[\max\left\{0, 1 - yw^\top x\right\}\right] > \underbrace{(1-p)}_{x_1 \text{ is wrong}} \mathbb{E}\left[\max\left\{0, 1 + w_1 - \mathcal{N}(4\nu, \nu^2)\right\}\right]$$
$$\overset{p=0.975}{=} 0.0004$$

but

$$\mathbb{E}\left[\max\left\{0, 1 - yw^\top x\right\}\right] < \underbrace{\mathbb{E}\left[\max\left\{0, 1 - \mathcal{N}(4, 1)\right\}\right]}_{w_1 = 0 \implies \nu = 1} = 0.0004$$

Proof

Proof (con't).

- **Claim** With probability at least 99%, $\nu z y > \frac{1}{\sqrt{2}} \geq w_1$

$$\mathbb{P}[zy > 1] = 1 - \Phi(-3) \geq 0.9986$$

  multiply both side in probability with $\nu > 0$ should not change the equality.

- This implies from νz alone, we can get accuracy more than 99% independent of x_1.
- Now the adversarial objective

$$\min_w \underbrace{\mathbb{E}\left[\max\left\{0, 1 - yw^\top x + \varepsilon\|w\|_1\right\}\right]}_{\text{margin}} + \overbrace{\frac{1}{2}\lambda\|w\|_2^2}^{\text{regularization}}$$

- If $w_i > 0$ for $i > 2$, then it implies that we are contributing the weight with negative mean $(\mathcal{N}(\eta, 1) - 2\eta \sim \mathcal{N}(-\eta, 1))$!
  - In another word, to maintain non-trivial robust accuracy, w_1 must dominates ν and that cost us with standard accuracy.

$\square$

Visualization

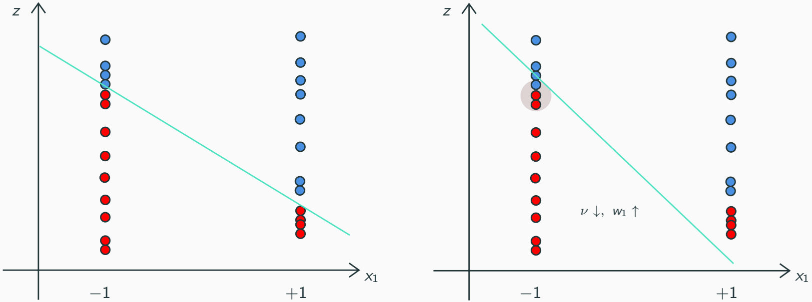

Figure 9: Under adversarial training, the contribution over invariant direction increase whereas the contribution to weakly correlated direction decrease

Reference

References i

Penot, J.-P.
Calculus Without Derivatives.
Springer New York, 2013.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A.
Robustness may be at odds with accuracy.
In *International Conference on Learning Representations* (2019).