DEPARTMENT OF
COMPUTER
SCIENCE

The
Alan Turing
Institute

How benign is benign overfitting ?

-Amartya Sanyal, Puneet Dokania, Varun Kanade, Philip Torr

Aug 7, 2020

Collaborators

Puneet Dokania

Varun Kanade

Philip H.S. Torr

Benign Overfitting

Overfitting in Deep Networks

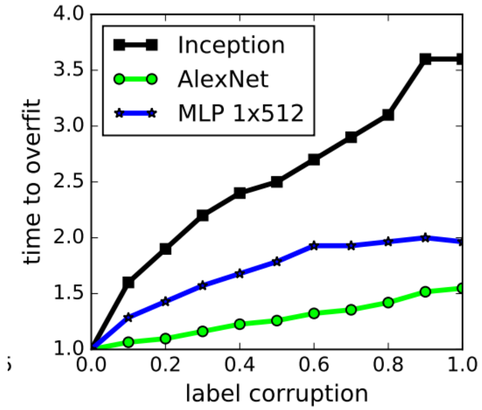
- Deep Networks are often trained to zero training error

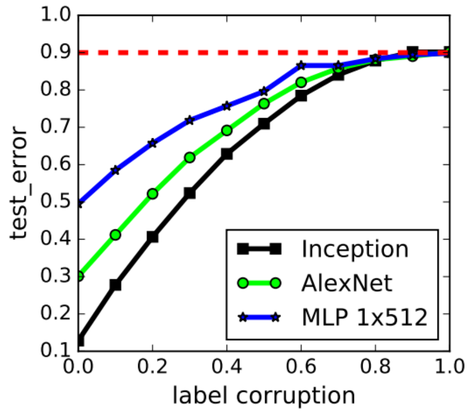
Overfitting in Deep Networks

- Deep Networks are often trained to zero training error

- Even in the presence of label noise

Overfitting in Deep Networks



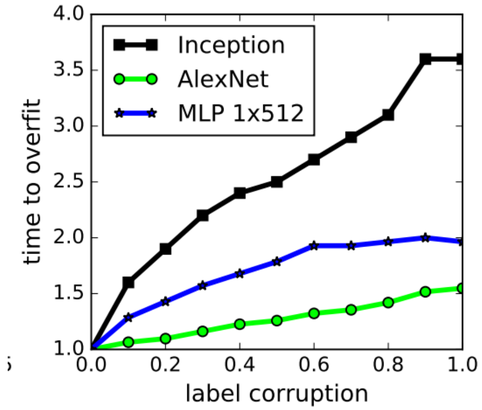


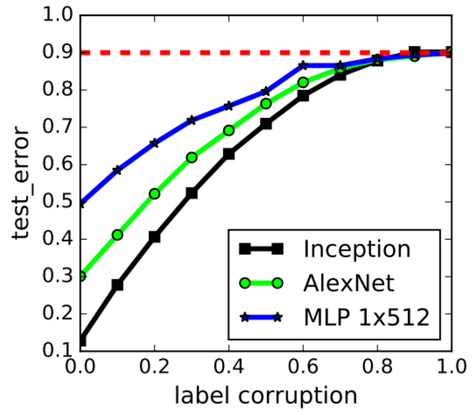
- Deep Networks are often trained to zero training error

- Even in the presence of label noise

Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization."

Overfitting in Deep Networks





- Deep Networks are often trained to zero training error
- Even in the presence of label noise
- Benign overfitting.

Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization."

How benign is benign overfitting ?

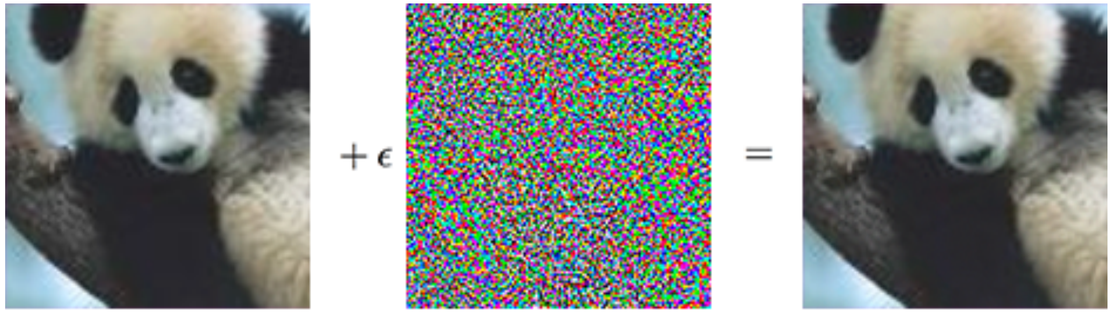
-Amartya Sanyal, Puneet Dokania, Varun Kanade, Philip Torr

How benign is benign overfitting ?

- Not very for Adversarial Robustness!

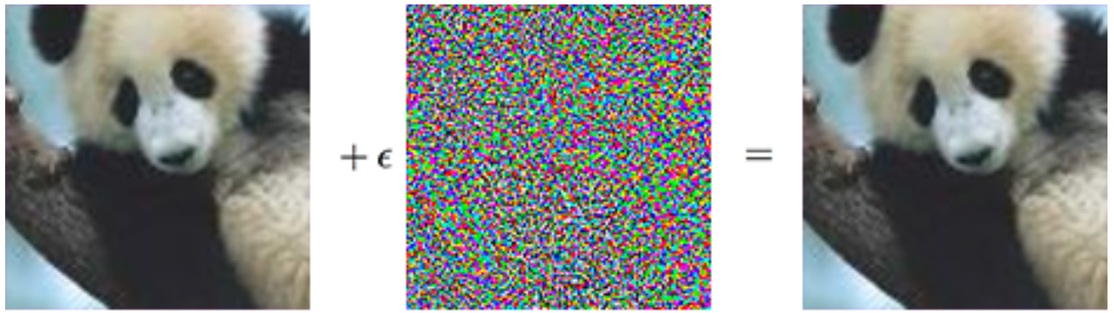
-Amartya Sanyal, Puneet Dokania, Varun Kanade, Philip Torr

Adversarial Robustness



Goodfellow et. al. 2014

Adversarial Robustness



Definition 1 (Natural and Adversarial Error). *For any distribution* $\mathcal{D}$ *defined over* $(\mathbf{x}, y) \in \mathbb{R}^d \times \{0,1\}$ *and any binary classifier* $f : \mathbb{R}^d \to \{0,1\}$,

- *the* natural *error is*

$$\mathcal{R}(f;\mathcal{D}) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\left[f\left(\mathbf{x}\right) \neq y\right], \tag{1}$$

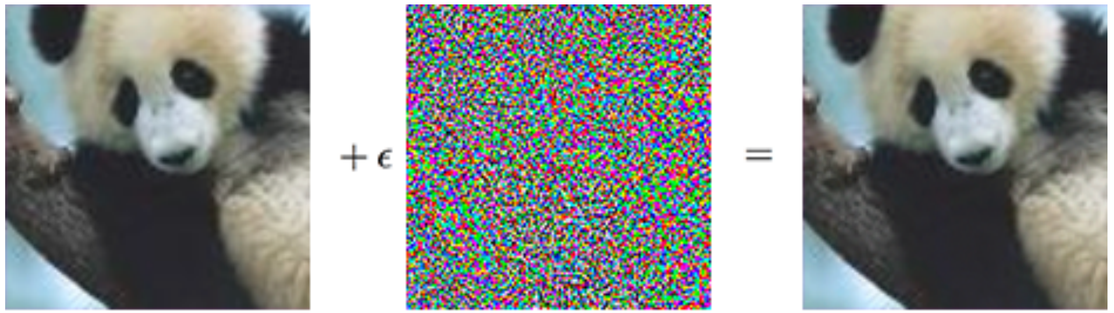
- *if* $\mathcal{B}_\gamma\left(\mathbf{x}\right)$ *is a ball of radius* $\gamma \geq 0$ *around* $\mathbf{x}$ *under some norm*[2], *the* γ-adversarial *error is*

$$\mathcal{R}_{\mathrm{Adv},\gamma}(f;\mathcal{D}) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists \mathbf{z} \in \mathcal{B}_\gamma\left(\mathbf{x}\right); f\left(\mathbf{z}\right) \neq y\right], \tag{2}$$

Goodfellow et. al. 2014

https://www.theverge.com/2018/1/3/16844842/ai-computer-vision-trick-adversarial-patches-google

Adversarial Robustness



Definition 1 (Natural and Adversarial Error). *For any distribution $\mathcal{D}$ defined over $(\mathbf{x}, y) \in \mathbb{R}^d \times \{0,1\}$ and any binary classifier $f : \mathbb{R}^d \to \{0,1\}$,*

- *the* natural *error is*

$$\mathcal{R}(f;\mathcal{D}) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\left[f\left(\mathbf{x}\right) \neq y\right], \tag{1}$$

- *if $\mathcal{B}_\gamma\left(\mathbf{x}\right)$ is a ball of radius $\gamma \geq 0$ around $\mathbf{x}$ under some norm[2], the* γ-adversarial *error is*

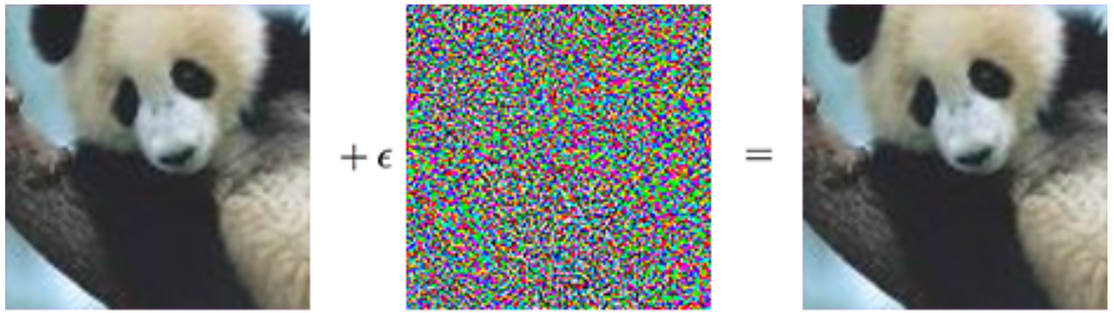
$$\mathcal{R}_{\mathrm{Adv},\gamma}(f;\mathcal{D}) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists \mathbf{z} \in \mathcal{B}_\gamma\left(\mathbf{x}\right); f\left(\mathbf{z}\right) \neq y\right], \tag{2}$$

On Average am I “close” to being wrong ?

Goodfellow et. al. 2014

https://www.theverge.com/2018/1/3/16844842/ai-computer-vision-trick-adversarial-patches-google

Adversarial Robustness



Definition 1 (Natural and Adversarial Error). *For any distribution $\mathcal{D}$ defined over $(\mathbf{x}, y) \in \mathbb{R}^d \times \{0,1\}$ and any binary classifier $f : \mathbb{R}^d \to \{0,1\}$,*

- *the* natural *error is*

$$\mathcal{R}(f;\mathcal{D}) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\left[f\left(\mathbf{x}\right) \neq y\right], \tag{1}$$

- *if $\mathcal{B}_\gamma\left(\mathbf{x}\right)$ is a ball of radius $\gamma \geq 0$ around $\mathbf{x}$ under some norm[2], the* γ-adversarial *error is*

$$\mathcal{R}_{\mathrm{Adv},\gamma}(f;\mathcal{D}) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exists \mathbf{z} \in \mathcal{B}_\gamma\left(\mathbf{x}\right); f\left(\mathbf{z}\right) \neq y\right], \tag{2}$$

On Average am I “close” to being wrong ?

Goodfellow et. al. 2014

https://www.theverge.com/2018/1/3/16844842/ai-computer-vision-trick-adversarial-patches-google

Adversarial Defense vs Attacks

Defenses

Deflecting Adversarial Attacks with Pixel Deflection (Prakash et al.) (code)	CVPR 2018	ImageNet	$\ell_2(\epsilon = 0.05)$	98.9% accuracy (on images originally classified correctly by underlying model)	81% accuracy (on images originally classified correctly)	• 0% accuracy [AC18] (code)
Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser (Liao et al.) (code)	CVPR 2018	ImageNet	$\ell_\infty(\epsilon = 4/255)$	75% accuracy	75% accuracy	• 0% accuracy [AC18] (code)
Towards Deep Learning Models Resistant to Adversarial Attacks (Madry et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 8/255)$	87% accuracy	46% accuracy	
Mitigating Adversarial Effects Through Randomization (Xie et al.) (code)	ICLR 2018	ImageNet	$\ell_\infty(\epsilon = 10/255)$	99.2% accuracy (on images originally classified correctly by underlying model)	86% accuracy (on images originally classified correctly)	• 0% accuracy [ACW18] (code)
Thermometer Encoding: One Hot Way To Resist Adversarial Examples (Buckman et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 8/255)$	90% accuracy	79% accuracy	• 30% accuracy [ACW18] (code)
Countering Adversarial Images using Input Transformations (Guo et al.) (code)	ICLR 2018	ImageNet	$\ell_2(\epsilon = 0.06)$	75% accuracy	70% accuracy on ImageNet with average normalized ℓ_2 perturbation of 0.06	• 0% accuracy [ACW18] (code)
Stochastic Activation Pruning for Robust Adversarial Defense (Dhillon et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 4/255)$	83% accuracy	51% accuracy	• 0% accuracy [ACW18] (code)
PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples (Song et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 8/255)$	90% accuracy	70% accuracy	• 9% accuracy [ACW18] (code)
Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks (Papernot et al.) (code)	S&P 2016	MNIST	$\ell_0(\epsilon = 112)$	99.51% accuracy	0.45% adversary success rate in changing classifier's prediction	• 3.6% accuracy [CW16] (code)

Adversarial Defense vs Attacks

Defenses

Deflecting Adversarial Attacks with Pixel Deflection (Prakash et al.) (code)	CVPR 2018	ImageNet	$\ell_2(\epsilon = 0.05)$	98.9% accuracy (on images originally classified correctly by underlying model)	81% accuracy (on images originally classified correctly)	• 0% accuracy [AC18] (code)
Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser (Liao et al.) (code)	CVPR 2018	ImageNet	$\ell_\infty(\epsilon = 4/255)$	75% accuracy	75% accuracy	• 0% accuracy [AC18] (code)
Towards Deep Learning Models Resistant to Adversarial Attacks (Madry et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 8/255)$	87% accuracy	46% accuracy	
Mitigating Adversarial Effects Through Randomization (Xie et al.) (code)	ICLR 2018	ImageNet	$\ell_\infty(\epsilon = 10/255)$	99.2% accuracy (on images originally classified correctly by underlying model)	86% accuracy (on images originally classified correctly)	• 0% accuracy [ACW18] (code)
Thermometer Encoding: One Hot Way To Resist Adversarial Examples (Buckman et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 8/255)$	90% accuracy	79% accuracy	• 30% accuracy [ACW18] (code)
Countering Adversarial Images using Input Transformations (Guo et al.) (code)	ICLR 2018	ImageNet	$\ell_2(\epsilon = 0.06)$	75% accuracy	70% accuracy on ImageNet with average normalized ℓ_2 perturbation of 0.06	• 0% accuracy [ACW18] (code)
Stochastic Activation Pruning for Robust Adversarial Defense (Dhillon et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 4/255)$	83% accuracy	51% accuracy	• 0% accuracy [ACW18] (code)
PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples (Song et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 8/255)$	90% accuracy	70% accuracy	• 9% accuracy [ACW18] (code)
Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks (Papernot et al.) (code)	S&P 2016	MNIST	$\ell_0(\epsilon = 112)$	99.51% accuracy	0.45% adversary success rate in changing classifier's prediction	• 3.6% accuracy [CW16] (code)

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples

Anish Athalye [*1] **Nicholas Carlini** [*2] **David Wagner** [2]

On Adaptive Attacks to Adversarial Example Defenses

Florian Tramèr*
Stanford University

Nicholas Carlini*
Google Brain

Wieland Brendel*
University of Tübingen

Aleksander Mądry
MIT

Adversarial Defense vs Attacks

Defenses

Deflecting Adversarial Attacks with Pixel Deflection (Prakash et al.) (code)	CVPR 2018	ImageNet	$\ell_2(\epsilon = 0.05)$	98.9% accuracy (on images originally classified correctly by underlying model)	81% accuracy (on images originally classified correctly)	• 0% accuracy [AC18] (code)
Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser (Liao et al.) (code)	CVPR 2018	ImageNet	$\ell_\infty(\epsilon = 4/255)$	75% accuracy	75% accuracy	• 0% accuracy [AC18] (code)
Towards Deep Learning Models Resistant to Adversarial Attacks (Madry et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 8/255)$	87% accuracy	46% accuracy	
Mitigating Adversarial Effects Through Randomization (Xie et al.) (code)	ICLR 2018	ImageNet	$\ell_\infty(\epsilon = 10/255)$	99.2% accuracy (on images originally classified correctly by	86% accuracy (on images originally classified correctly)	• 0% accuracy [ACW18]
Countering Adversarial Images using Input Transformations (Guo et al.) (code)	ICLR 2018	ImageNet	$\ell_2(\epsilon = 0.06)$	75% accuracy	70% accuracy on ImageNet with average normalized ℓ_2 perturbation of 0.06	• 0% accuracy [ACW18] (code)
Stochastic Activation Pruning for Robust Adversarial Defense (Dhillon et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 4/255)$	83% accuracy	51% accuracy	• 0% accuracy [ACW18] (code)
PixelDefend: Leveraging Generative Models to Understand and Defend against Adversarial Examples (Song et al.) (code)	ICLR 2018	CIFAR-10	$\ell_\infty(\epsilon = 8/255)$	90% accuracy	70% accuracy	• 9% accuracy [ACW18] (code)
Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks (Papernot et al.) (code)	S&P 2016	MNIST	$\ell_0(\epsilon = 112)$	99.51% accuracy	0.45% adversary success rate in changing classifier's prediction	• 3.6% accuracy [CW16] (code)

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples

Anish Athalye [*1] **Nicholas Carlini** [*2] **David Wagner** [2]

Stanford University Google Brain University of Tübingen

Aleksander Mądry
MIT

Goodhart's Law:
"When a measure becomes a target, it ceases to be a good measure."

Causes of Adversarial Vulnerability

Causes of Adversarial Vulnerability

Bad Data

Noisy Labels are quite ubiquitous in real data and benchmark datasets.

Causes of Adversarial Vulnerability

Bad Data

Noisy Labels are quite ubiquitous in real data and benchmark datasets.

Bad Representations

We would ideally want representations that can reflect the perceptual similarity between real world objects as seen by humans.

Impact of Bad data

Overfitting Label Noise

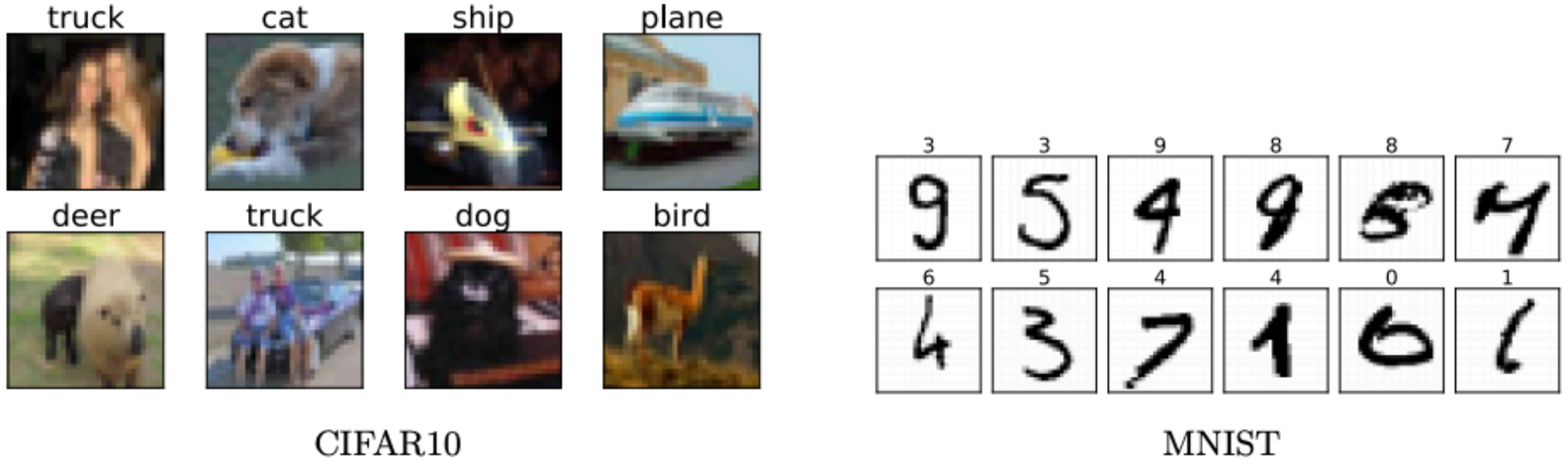


Figure 1: Label Noise in CIFAR10 and MNIST. Text above the image indicates the training set label.

Impact of Bad data

Overfitting Label Noise

Theorem 1. *Let c be the target classifier, and let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y)$, such that $y = c(\mathbf{x})$ in its support. Using the notation $\mathbb{P}_D[A]$ to denote $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathbf{x} \in A]$ for any measurable subset $A \subseteq \mathbb{R}^d$, suppose that there exist $c_1 \geq c_2 > 0$, $\rho > 0$, and a finite set $\zeta \subset \mathbb{R}^d$ satisfying*

$$\mathbb{P}_{\mathcal{D}}\left[\bigcup_{\mathbf{s}\in\zeta} \mathcal{B}_\rho^p(\mathbf{s})\right] \geq c_1 \quad \textit{and} \quad \forall \mathbf{s} \in \zeta,\ \mathbb{P}_{\mathcal{D}}\left[\mathcal{B}_\rho^p(\mathbf{s})\right] \geq \frac{c_2}{|\zeta|} \tag{3}$$

where $\mathcal{B}_\rho^p(\mathbf{s})$ represents a ℓ_p-ball of radius ρ around $\mathbf{s}$. Further, suppose that each of these balls contain points from a single class i.e. for all $\mathbf{s} \in \zeta$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{B}_\rho^p(\mathbf{s}) : c(\mathbf{x}) = c(\mathbf{z})$.

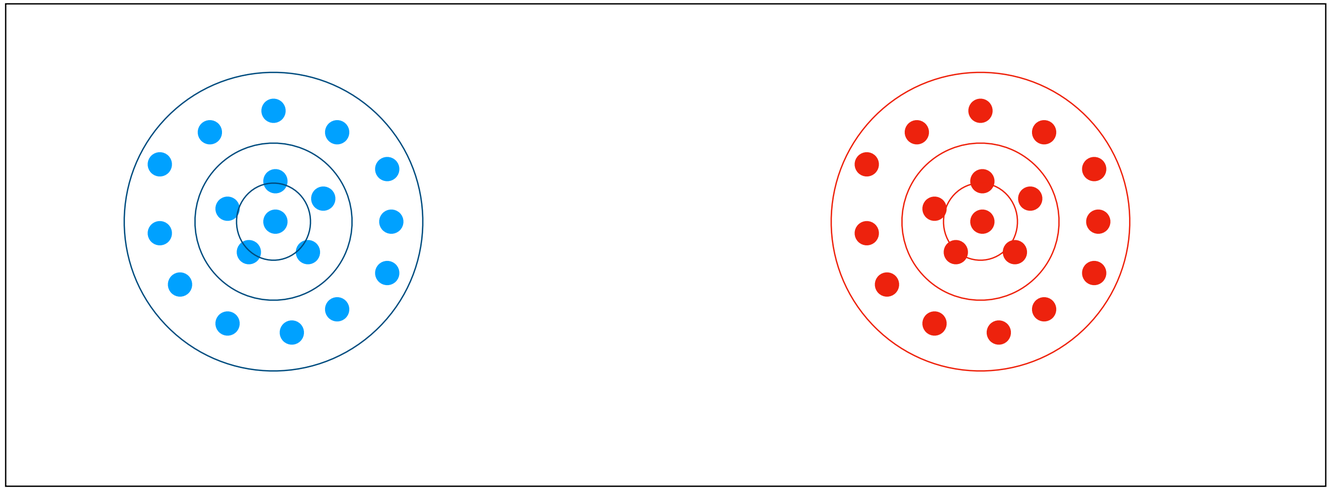
Let $\mathcal{S}_m$ be a dataset of m i.i.d. samples drawn from $\mathcal{D}$, which subsequently has each label flipped independently with probability η. For any classifier f that perfectly *fits the training data $\mathcal{S}_m$ i.e. $\forall\ \mathbf{x}, y \in \mathcal{S}_m, f(\mathbf{x}) = y$, $\forall \delta > 0$ and $m \geq \frac{|\zeta|}{\eta c_2} \log\left(\frac{|\zeta|}{\delta}\right)$, with probability at least $1 - \delta$, $\mathcal{R}_{\mathrm{Adv},2\rho}(f; \mathcal{D}) \geq c_1$.*

Theorem 1. *Let c be the target classifier, and let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y)$, such that $y = c(\mathbf{x})$ in its support. Using the notation $\mathbb{P}_D[A]$ to denote $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathbf{x} \in A]$ for any measurable subset $A \subseteq \mathbb{R}^d$, suppose that there exist $c_1 \geq c_2 > 0$, $\rho > 0$, and a finite set $\zeta \subset \mathbb{R}^d$ satisfying*

$$\mathbb{P}_{\mathcal{D}}\left[\bigcup_{\mathbf{s}\in\zeta} \mathcal{B}_\rho^p(\mathbf{s})\right] \geq c_1 \quad \textit{and} \quad \forall \mathbf{s} \in \zeta,\ \mathbb{P}_{\mathcal{D}}\left[\mathcal{B}_\rho^p(\mathbf{s})\right] \geq \frac{c_2}{|\zeta|} \tag{3}$$

where $\mathcal{B}_\rho^p(\mathbf{s})$ represents a ℓ_p-ball of radius ρ around $\mathbf{s}$. Further, suppose that each of these balls contain points from a single class i.e. for all $\mathbf{s} \in \zeta$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{B}_\rho^p(\mathbf{s}) : c(\mathbf{x}) = c(\mathbf{z})$.

Let $\mathcal{S}_m$ be a dataset of m i.i.d. samples drawn from $\mathcal{D}$, which subsequently has each label flipped independently with probability η. For any classifier f that perfectly *fits the training data $\mathcal{S}_m$ i.e. $\forall\ \mathbf{x}, y \in \mathcal{S}_m, f(\mathbf{x}) = y$, $\forall \delta > 0$ and $m \geq \frac{|\zeta|}{\eta c_2} \log\left(\frac{|\zeta|}{\delta}\right)$, with probability at least $1 - \delta$, $\mathcal{R}_{\mathrm{Adv},2\rho}(f; \mathcal{D}) \geq c_1$.*

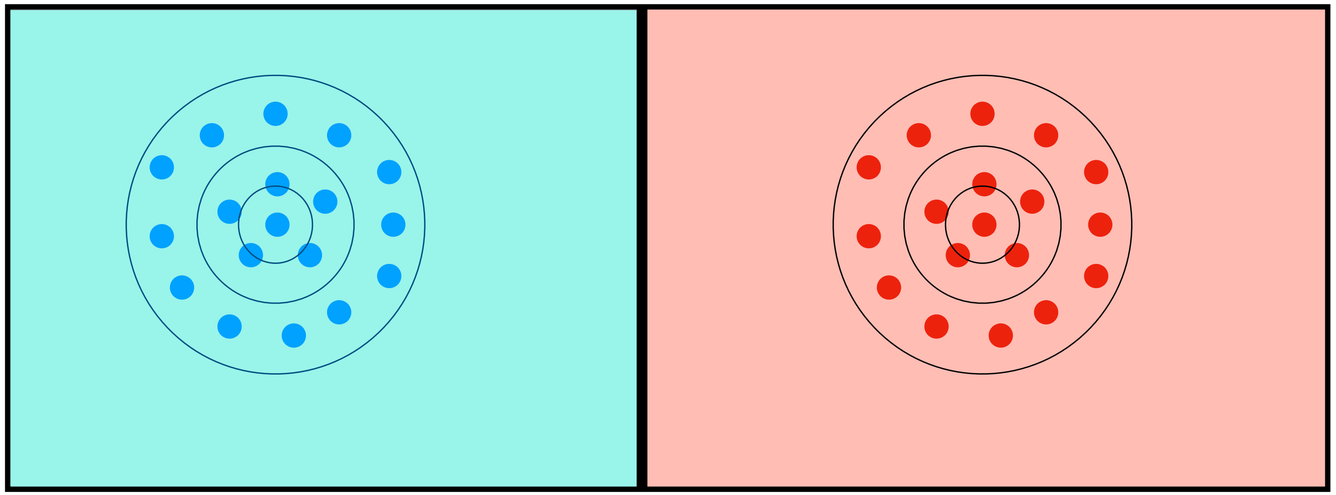


Theorem 1. *Let c be the target classifier, and let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y)$, such that $y = c(\mathbf{x})$ in its support. Using the notation $\mathbb{P}_D[A]$ to denote $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathbf{x} \in A]$ for any measurable subset $A \subseteq \mathbb{R}^d$, suppose that there exist $c_1 \geq c_2 > 0$, $\rho > 0$, and a finite set $\zeta \subset \mathbb{R}^d$ satisfying*

$$\mathbb{P}_{\mathcal{D}}\left[\bigcup_{\mathbf{s}\in\zeta} \mathcal{B}_\rho^p(\mathbf{s})\right] \geq c_1 \quad \textit{and} \quad \forall \mathbf{s} \in \zeta,\ \mathbb{P}_{\mathcal{D}}\left[\mathcal{B}_\rho^p(\mathbf{s})\right] \geq \frac{c_2}{|\zeta|} \tag{3}$$

where $\mathcal{B}_\rho^p(\mathbf{s})$ represents a ℓ_p-ball of radius ρ around $\mathbf{s}$. Further, suppose that each of these balls contain points from a single class i.e. for all $\mathbf{s} \in \zeta$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{B}_\rho^p(\mathbf{s}) : c(\mathbf{x}) = c(\mathbf{z})$.

Let $\mathcal{S}_m$ be a dataset of m i.i.d. samples drawn from $\mathcal{D}$, which subsequently has each label flipped independently with probability η. For any classifier f that perfectly *fits the training data $\mathcal{S}_m$ i.e. $\forall\ \mathbf{x}, y \in \mathcal{S}_m, f(\mathbf{x}) = y$, $\forall \delta > 0$ and $m \geq \frac{|\zeta|}{\eta c_2} \log\left(\frac{|\zeta|}{\delta}\right)$, with probability at least $1 - \delta$, $\mathcal{R}_{\mathrm{Adv},2\rho}(f; \mathcal{D}) \geq c_1$.*



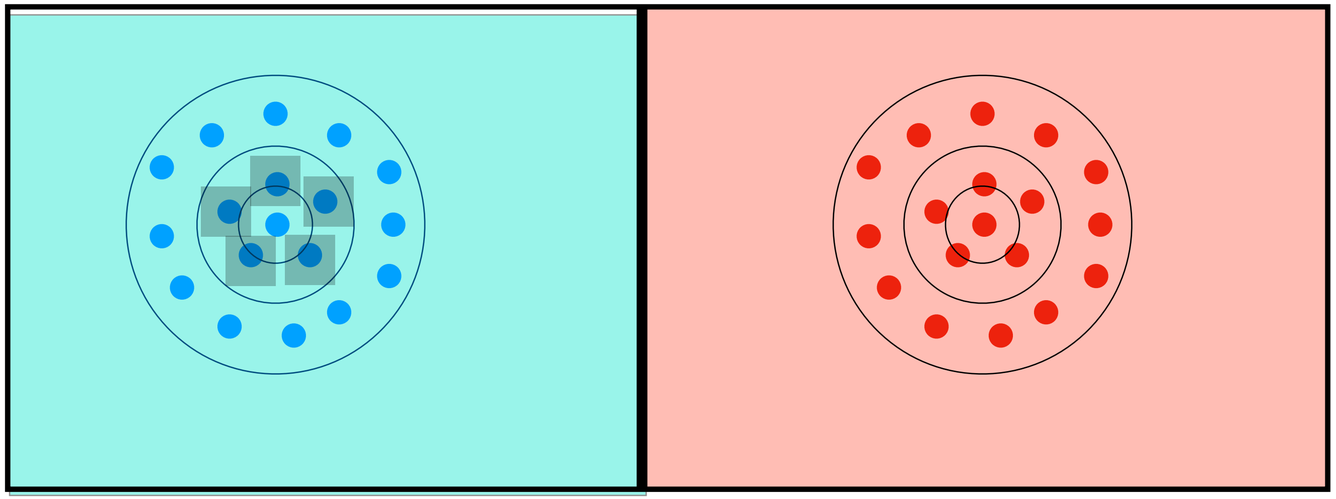
Theorem 1. *Let c be the target classifier, and let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y)$, such that $y = c(\mathbf{x})$ in its support. Using the notation $\mathbb{P}_D[A]$ to denote $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathbf{x} \in A]$ for any measurable subset $A \subseteq \mathbb{R}^d$, suppose that there exist $c_1 \geq c_2 > 0$, $\rho > 0$, and a finite set $\zeta \subset \mathbb{R}^d$ satisfying*

$$\mathbb{P}_{\mathcal{D}}\left[\bigcup_{\mathbf{s}\in\zeta} \mathcal{B}_\rho^p(\mathbf{s})\right] \geq c_1 \quad \text{and} \quad \forall \mathbf{s} \in \zeta,\ \mathbb{P}_{\mathcal{D}}\left[\mathcal{B}_\rho^p(\mathbf{s})\right] \geq \frac{c_2}{|\zeta|} \tag{3}$$

where $\mathcal{B}_\rho^p(\mathbf{s})$ represents a ℓ_p-ball of radius ρ around $\mathbf{s}$. Further, suppose that each of these balls contain points from a single class i.e. for all $\mathbf{s} \in \zeta$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{B}_\rho^p(\mathbf{s}) : c(\mathbf{x}) = c(\mathbf{z})$.

Let $\mathcal{S}_m$ be a dataset of m i.i.d. samples drawn from $\mathcal{D}$, which subsequently has each label flipped independently with probability η. For any classifier f that perfectly *fits the training data $\mathcal{S}_m$ i.e. $\forall\ \mathbf{x}, y \in \mathcal{S}_m, f(\mathbf{x}) = y$, $\forall \delta > 0$ and $m \geq \frac{|\zeta|}{\eta c_2} \log\left(\frac{|\zeta|}{\delta}\right)$, with probability at least $1 - \delta$, $\mathcal{R}_{\mathrm{Adv},2\rho}(f; \mathcal{D}) \geq c_1$.*

ζ = (, , , ,)



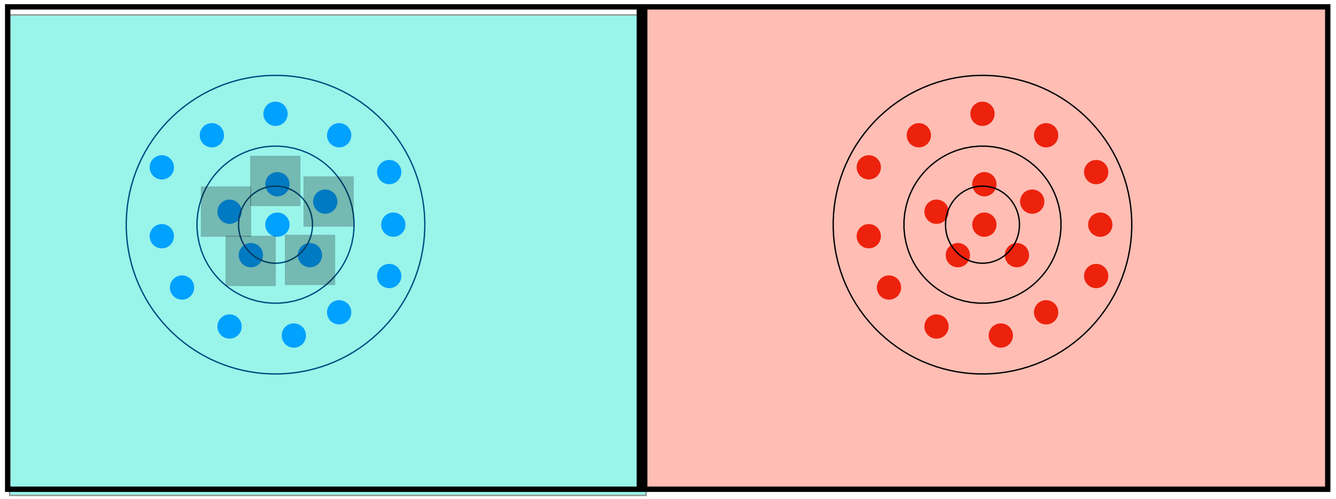
Theorem 1. *Let c be the target classifier, and let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y)$, such that $y = c(\mathbf{x})$ in its support. Using the notation $\mathbb{P}_D[A]$ to denote $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathbf{x} \in A]$ for any measurable subset $A \subseteq \mathbb{R}^d$, suppose that there exist $c_1 \geq c_2 > 0$, $\rho > 0$, and a finite set $\zeta \subset \mathbb{R}^d$ satisfying*

$$\mathbb{P}_{\mathcal{D}}\left[\bigcup_{\mathbf{s}\in\zeta} \mathcal{B}_\rho^p(\mathbf{s})\right] \geq c_1 \quad \text{and} \quad \forall \mathbf{s} \in \zeta,\ \mathbb{P}_{\mathcal{D}}\left[\mathcal{B}_\rho^p(\mathbf{s})\right] \geq \frac{c_2}{|\zeta|} \tag{3}$$

where $\mathcal{B}_\rho^p(\mathbf{s})$ represents a ℓ_p-ball of radius ρ around $\mathbf{s}$. Further, suppose that each of these balls contain points from a single class i.e. for all $\mathbf{s} \in \zeta$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{B}_\rho^p(\mathbf{s}) : c(\mathbf{x}) = c(\mathbf{z})$.

Let $\mathcal{S}_m$ be a dataset of m i.i.d. samples drawn from $\mathcal{D}$, which subsequently has each label flipped independently with probability η. For any classifier f that perfectly *fits the training data $\mathcal{S}_m$ i.e. $\forall\ \mathbf{x}, y \in \mathcal{S}_m, f(\mathbf{x}) = y$, $\forall \delta > 0$ and $m \geq \frac{|\zeta|}{\eta c_2} \log\left(\frac{|\zeta|}{\delta}\right)$, with probability at least $1 - \delta$, $\mathcal{R}_{\mathrm{Adv},2\rho}(f; \mathcal{D}) \geq c_1$.*

ζ = (, , , ,)



Theorem 1. *Let c be the target classifier, and let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y)$, such that $y = c(\mathbf{x})$ in its support. Using the notation $\mathbb{P}_D[A]$ to denote $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathbf{x} \in A]$ for any measurable subset $A \subseteq \mathbb{R}^d$, suppose that there exist $c_1 \geq c_2 > 0$, $\rho > 0$, and a finite set $\zeta \subset \mathbb{R}^d$ satisfying*

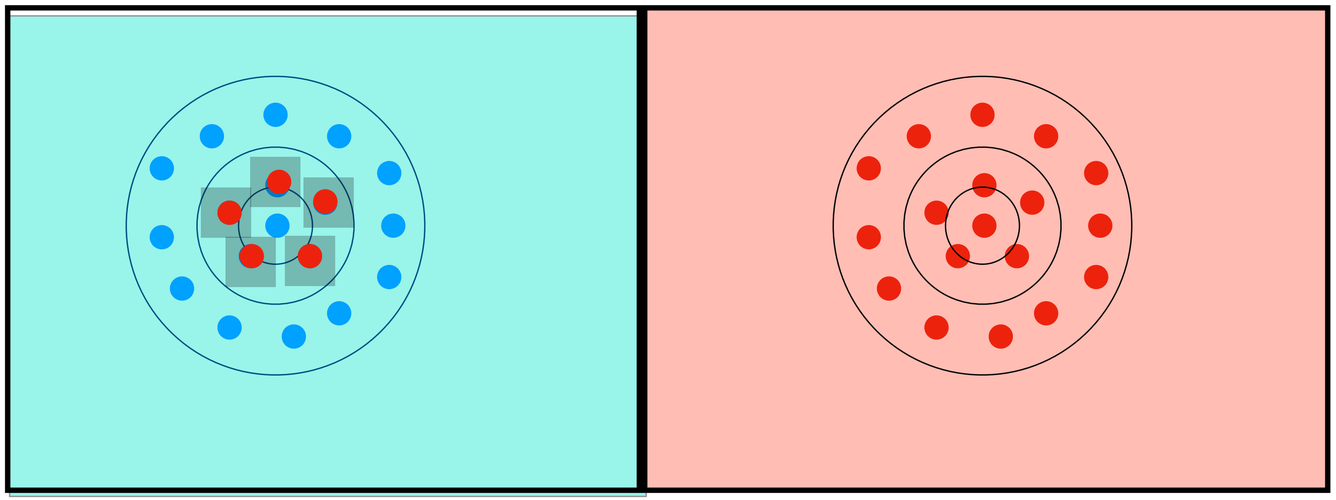
$$\mathbb{P}_{\mathcal{D}}\left[\bigcup_{\mathbf{s}\in\zeta} \mathcal{B}_\rho^p(\mathbf{s})\right] \geq c_1 \quad \text{and} \quad \forall \mathbf{s} \in \zeta,\ \mathbb{P}_{\mathcal{D}}\left[\mathcal{B}_\rho^p(\mathbf{s})\right] \geq \frac{c_2}{|\zeta|} \tag{3}$$

where $\mathcal{B}_\rho^p(\mathbf{s})$ represents a ℓ_p-ball of radius ρ around $\mathbf{s}$. Further, suppose that each of these balls contain points from a single class i.e. for all $\mathbf{s} \in \zeta$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{B}_\rho^p(\mathbf{s}) : c(\mathbf{x}) = c(\mathbf{z})$.

Let $\mathcal{S}_m$ be a dataset of m i.i.d. samples drawn from $\mathcal{D}$, which subsequently has each label flipped independently with probability η. For any classifier f that perfectly *fits the training data $\mathcal{S}_m$ i.e. $\forall\ \mathbf{x}, y \in \mathcal{S}_m, f(\mathbf{x}) = y$, $\forall \delta > 0$ and $m \geq \frac{|\zeta|}{\eta c_2} \log\left(\frac{|\zeta|}{\delta}\right)$, with probability at least $1 - \delta$, $\mathcal{R}_{\text{Adv},2\rho}(f; \mathcal{D}) \geq c_1$.*

$h(\mathbf{x}) = c(\mathbf{x})$, if $(\mathbf{x}, b) \notin \mathcal{S}_m$ for $b = 0, 1$, and $h(\mathbf{x}) = y$ if $(\mathbf{x}, y) \in \mathcal{S}_m$.

ζ = (, , , ,)



Theorem 1. *Let c be the target classifier, and let $\mathcal{D}$ be a distribution over $(\mathbf{x}, y)$, such that $y = c(\mathbf{x})$ in its support. Using the notation $\mathbb{P}_D[A]$ to denote $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathbf{x} \in A]$ for any measurable subset $A \subseteq \mathbb{R}^d$, suppose that there exist $c_1 \geq c_2 > 0$, $\rho > 0$, and a finite set $\zeta \subset \mathbb{R}^d$ satisfying*

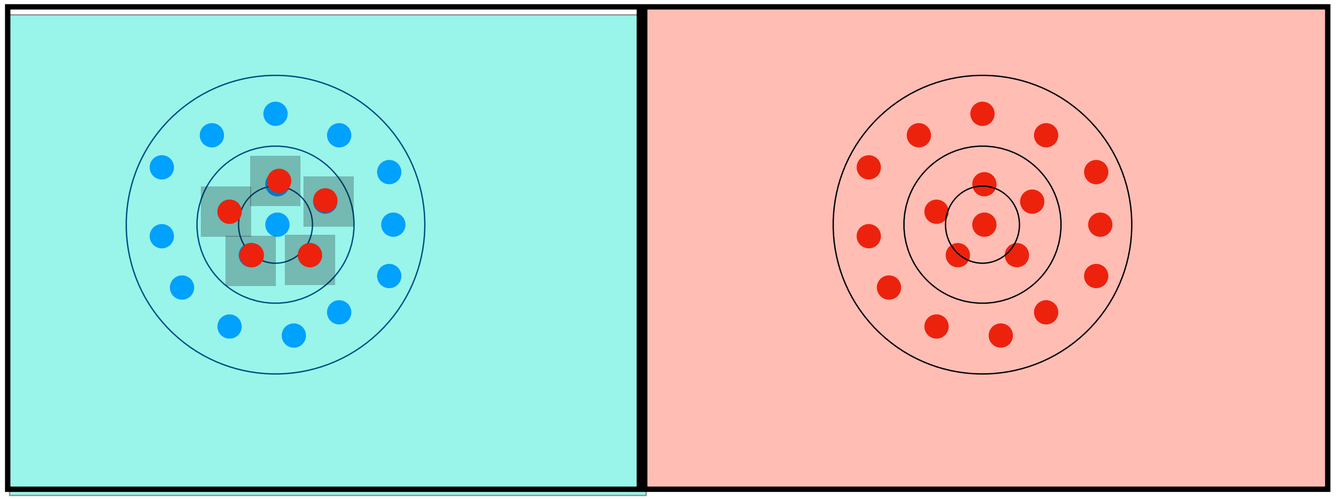
$$\mathbb{P}_{\mathcal{D}}\left[\bigcup_{\mathbf{s}\in\zeta} \mathcal{B}_\rho^p(\mathbf{s})\right] \geq c_1 \quad \text{and} \quad \forall \mathbf{s} \in \zeta,\ \mathbb{P}_{\mathcal{D}}\left[\mathcal{B}_\rho^p(\mathbf{s})\right] \geq \frac{c_2}{|\zeta|} \tag{3}$$

where $\mathcal{B}_\rho^p(\mathbf{s})$ represents a ℓ_p-ball of radius ρ around $\mathbf{s}$. Further, suppose that each of these balls contain points from a single class i.e. for all $\mathbf{s} \in \zeta$, for all $\mathbf{x}, \mathbf{z} \in \mathcal{B}_\rho^p(\mathbf{s}) : c(\mathbf{x}) = c(\mathbf{z})$.

Let $\mathcal{S}_m$ be a dataset of m i.i.d. samples drawn from $\mathcal{D}$, which subsequently has each label flipped independently with probability η. For any classifier f that perfectly *fits the training data $\mathcal{S}_m$ i.e. $\forall\ \mathbf{x}, y \in \mathcal{S}_m, f(\mathbf{x}) = y$, $\forall \delta > 0$ and $m \geq \frac{|\zeta|}{\eta c_2} \log\left(\frac{|\zeta|}{\delta}\right)$, with probability at least $1 - \delta$, $\mathcal{R}_{\mathrm{Adv},2\rho}(f; \mathcal{D}) \geq c_1$.*

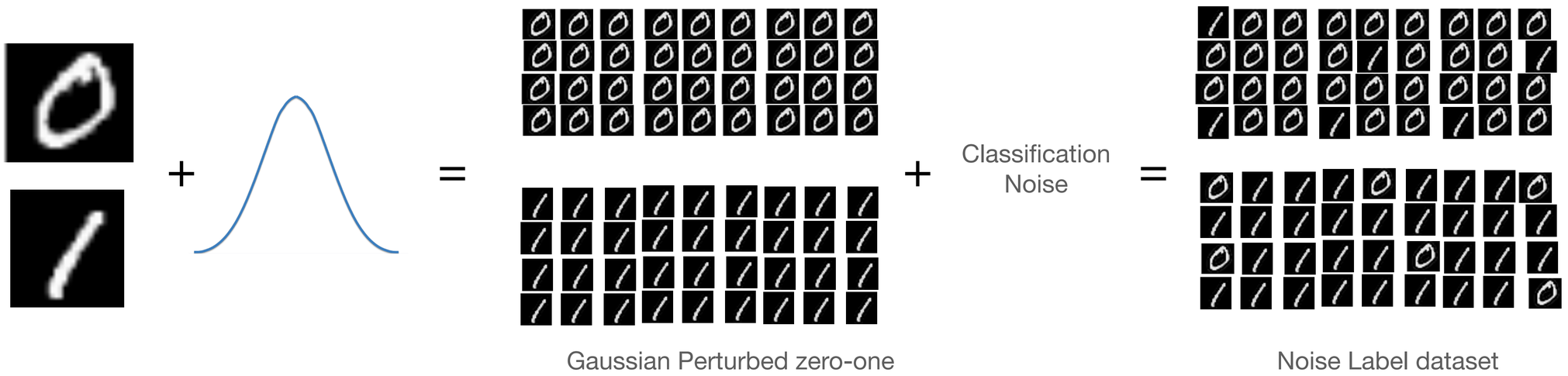
$h(\mathbf{x}) = c(\mathbf{x})$, if $(\mathbf{x}, b) \notin \mathcal{S}_m$ for $b = 0, 1$, and $h(\mathbf{x}) = y$ if $(\mathbf{x}, y) \in \mathcal{S}_m$.

ζ = (, , , ,)



Impact of Bad data

Overfitting Label Noise (Synthetic Experiment)



Impact of Bad data

Overfitting Label Noise (Synthetic Experiment)

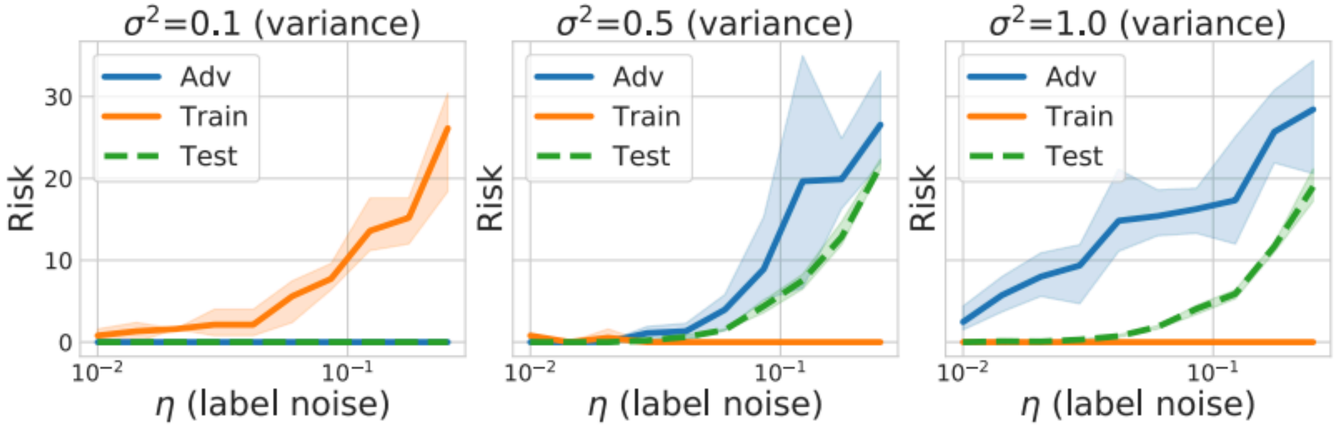


Figure 3: Adversarial error increases with label noise (η) if training error is 0. Shaded region shows 95% confidence interval.

When training error is zero/interpolation happens, adversarial error increases faster than test error for increasing noise.

Impact of Bad data

Overfitting Label Noise (Synthetic Experiment)

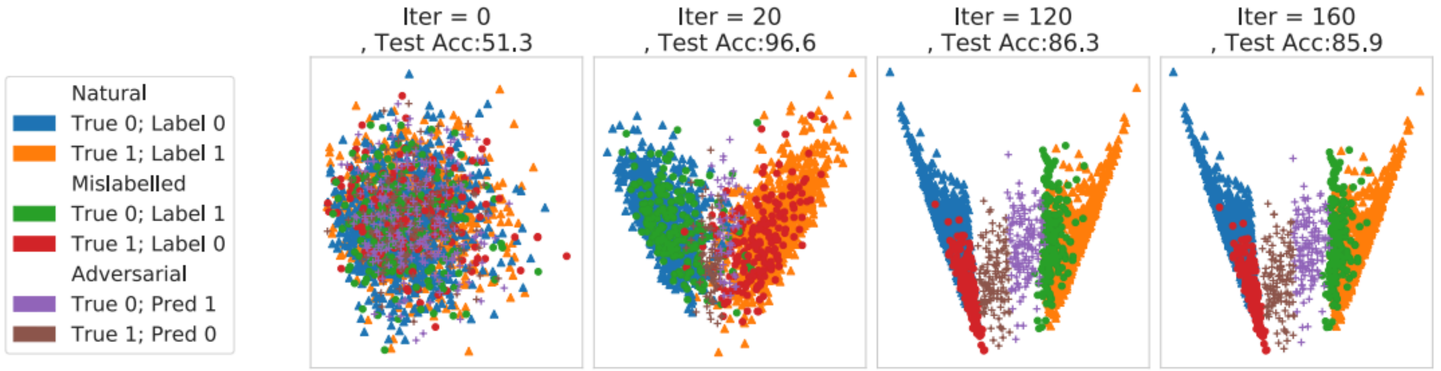
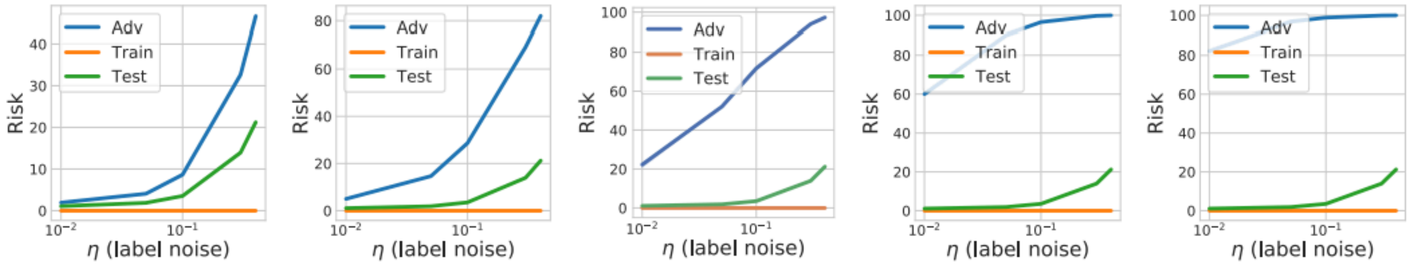


Figure 5: Two dimensional PCA projections of the original correctly labelled (blue and orange), original mis-labelled (green and red), and adversarial examples (purple and brown) at different stages of training. The correct label for *True 0* (blue), *Noisy 0* (green), *Adv 0* (purple +) are the same i.e. 0 and similar for the other class.

Adversarial examples are formed by moving towards mis-labelled points of the opposite class

Impact of Bad data

Overfitting Label Noise (Full MNIST Experiment)



(a) $\epsilon = 0.01$ (b) $\epsilon = 0.025$ (c) $\epsilon = 0.05$ (d) $\epsilon = 0.1$ (e) $\epsilon = 0.2$

Figure 4: Shows the adversarial for the full MNIST dataset for varying levels of adversarial perturbation. There is negligible variance between runs and thus the shaded region showing the confidence interval is invisible.

Increasing noise increases Adversarial vulnerability much faster than it increases test error.

What does Robust Training Algorithms do ?

Adversarial Training loses clean accuracy

ϵ	Train-Acc. (%)	Test-Acc (%)
0.0	99.98	95.25
0.25	97.23	92.77
1.0	86.03	81.62

Table 1: Train and test accuracies on clean dataset for ResNet-50 models trained using ℓ_2 adversaries of perturbation ϵ. The $\epsilon = 0$ setting represents the natural training.

Adversarial Training loses clean accuracy

ϵ	Train-Acc. (%)	Test-Acc (%)
0.0	99.98	95.25
0.25	97.23	92.77
1.0	86.03	81.62

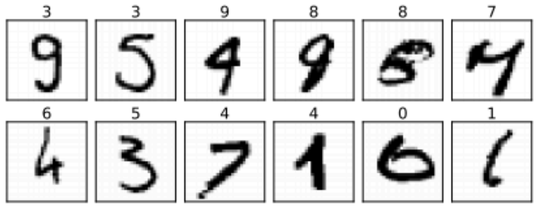
Table 1: Train and test accuracies on clean dataset for ResNet-50 models trained using ℓ_2 adversaries of perturbation ϵ. The $\epsilon = 0$ setting represents the natural training.

Two observations:

- AT decreases clean Training Accuracy
- AT decreases clean Test Accuracy

Adversarial Training loses train accuracy

ϵ	Train-Acc. (%)	Test-Acc (%)
0.0	99.98	95.25
0.25	97.23	92.77
1.0	86.03	81.62



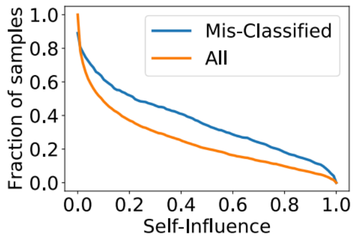
MNIST

AT misclassifies all the mis-labelled images in both CIFAR-10 and MNIST training sets that we could identify.

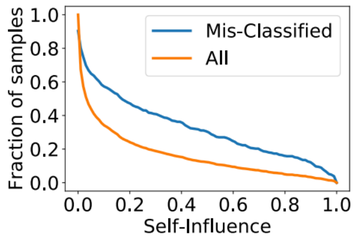
CIFAR10

Adversarial Training loses train accuracy

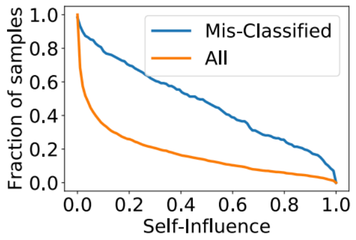
The self-influence of images mis-classified by AT is much higher than the average self-influence of the datasets.



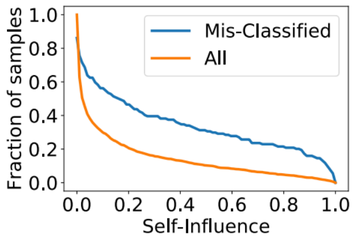
DOG



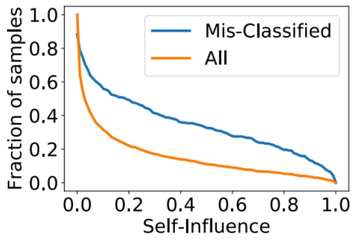
FROG



HORSE



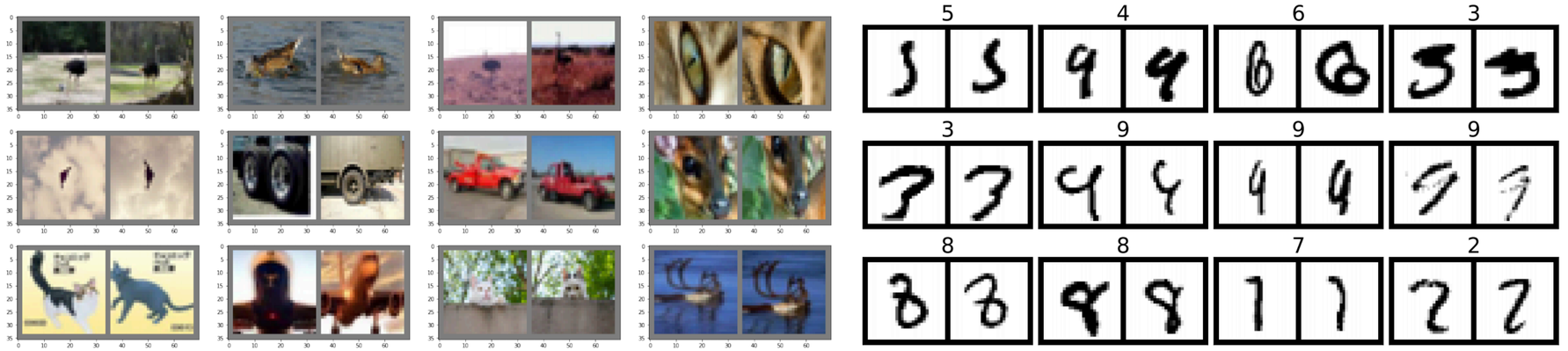
SHIP



TRUCK

TL;DR: Adversarial Training avoids rare examples

Adversarial Training loses test accuracy



CIFAR10 MNIST

Figure 6: Each pair is a training (left) and test (right) image mis-classified by the adversarially trained model. They were both correctly classified by the naturally-trained model.

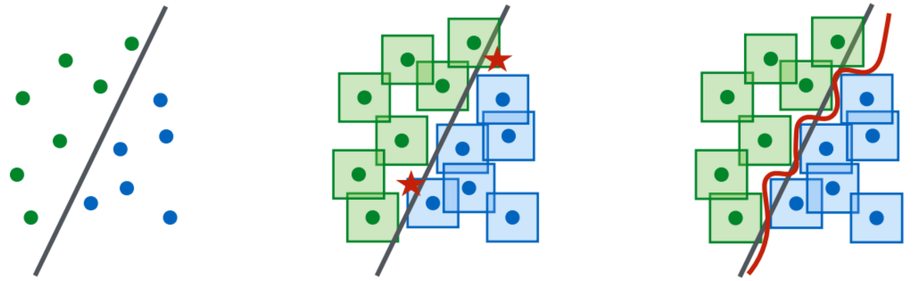
Proper representation learning and adversarial robustness

Proper representation learning and adversarial robustness

- Choice of model affects representation and invariances in the representations.
- Invariances in the representations affects the shape of decision boundary.

Proper representation learning and adversarial robustness

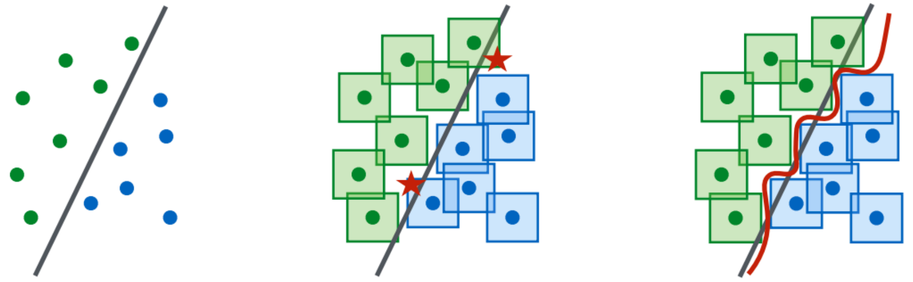
- Choice of model affects representation and invariances in the representations.
- Invariances in the representations affects the shape of decision boundary.
- Adversarial robustness can require more “complex” decision boundaries.



Madry et. al. 2017

Proper representation learning and adversarial robustness

- Choice of model affects representation and invariances in the representations.
- Invariances in the representations affects the shape of decision boundary.
- Adversarial robustness can require more “complex” decision boundaries.
- However, this visual complexity need not refer to statistical complexity.

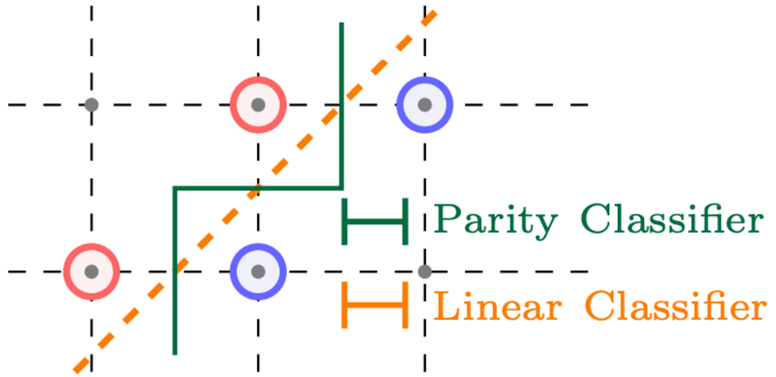


Madry et. al. 2017

Proper representation learning and adversarial robustness

There exists a distribution such that

- With representation A:
  - Both training error and test error can be zero.
  - But Adversarial error will be large.



Theorem 2. *For some universal constant c, and any $0 < \gamma_0 < 1/\sqrt{2}$, there exists a family of distributions $\mathcal{D}$ defined on $\mathcal{X} \times \{0,1\}$ where $\mathcal{X} \subseteq \mathbb{R}^2$ such that for all distributions $\mathcal{P} \in \mathcal{D}$, and denoting by $\mathcal{S}_m = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)\}$ a sample of size m drawn i.i.d. from $\mathcal{P}$,*

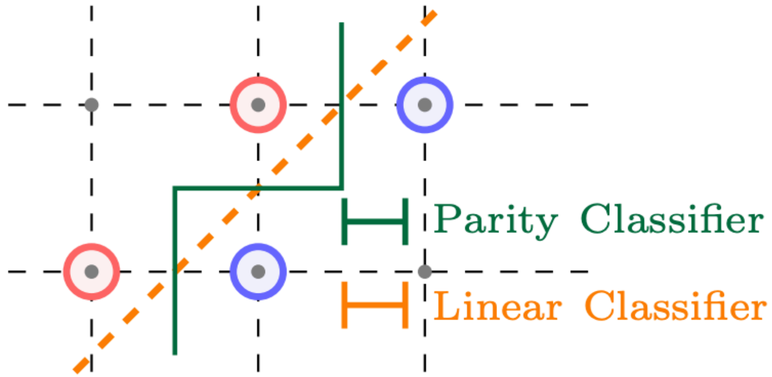
(i) For any $m \geq 0$, $\mathcal{S}_m$ is linearly separable i.e., $\forall (\mathbf{x}_i, y_i) \in \mathcal{S}_m$, there exist $\mathbf{w} \in \mathbb{R}^2, w_0 \in \mathbb{R}$ s.t. $y_i \left(\mathbf{w}^\top \mathbf{x}_i + w_0\right) \geq 0$. Furthermore, for every $\gamma > \gamma_0$, any linear separator f that perfectly fits the training data $\mathcal{S}_m$ has $\mathcal{R}_{\mathrm{Adv},\gamma}(f;\mathcal{P}) \geq 0.0005$, even though $\mathcal{R}(f;\mathcal{P}) \to 0$ as $m \to \infty$.

(ii) There exists a function class $\mathcal{H}$ such that for some $m \in O(\log(\delta^{-1}))$, any $h \in \mathcal{H}$ that perfectly fits the $\mathcal{S}_m$, satisfies with probability at least $1-\delta$, $\mathcal{R}(h;\mathcal{P}) = 0$ and $\mathcal{R}_{\mathrm{Adv},\gamma}(h;\mathcal{P}) = 0$, for any $\gamma \in [0, \gamma_0 + 1/8]$.

Proper representation learning and adversarial robustness

There exists a distribution such that

- With representation A:
  - Both training error and test error can be zero.
  - But Adversarial error will be large.
- With representation B:
  - Both training error and test error can be zero.
  - And Adversarial error will also be zero.



Theorem 2. *For some universal constant c, and any $0 < \gamma_0 < 1/\sqrt{2}$, there exists a family of distributions $\mathcal{D}$ defined on $\mathcal{X} \times \{0,1\}$ where $\mathcal{X} \subseteq \mathbb{R}^2$ such that for all distributions $\mathcal{P} \in \mathcal{D}$, and denoting by $\mathcal{S}_m = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)\}$ a sample of size m drawn i.i.d. from $\mathcal{P}$,*

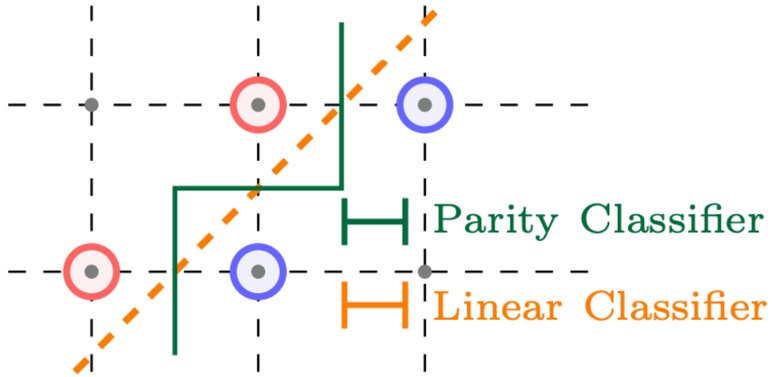
(i) For any $m \geq 0$, $\mathcal{S}_m$ is linearly separable i.e., $\forall (\mathbf{x}_i, y_i) \in \mathcal{S}_m$, there exist $\mathbf{w} \in \mathbb{R}^2, w_0 \in \mathbb{R}$ s.t. $y_i \left(\mathbf{w}^\top \mathbf{x}_i + w_0\right) \geq 0$. Furthermore, for every $\gamma > \gamma_0$, any linear separator f that perfectly fits the training data $\mathcal{S}_m$ has $\mathcal{R}_{\mathrm{Adv},\gamma}(f;\mathcal{P}) \geq 0.0005$, even though $\mathcal{R}(f;\mathcal{P}) \to 0$ as $m \to \infty$.

(ii) There exists a function class $\mathcal{H}$ such that for some $m \in O(\log(\delta^{-1}))$, any $h \in \mathcal{H}$ that perfectly fits the $\mathcal{S}_m$, satisfies with probability at least $1-\delta$, $\mathcal{R}(h;\mathcal{P}) = 0$ and $\mathcal{R}_{\mathrm{Adv},\gamma}(h;\mathcal{P}) = 0$, for any $\gamma \in [0, \gamma_0 + 1/8]$.

Proper representation learning and adversarial robustness

There exists a distribution such that

- With representation A:
  - Both training error and test error can be zero.
  - But Adversarial error will be large.
- With representation B:
  - Both training error and test error can be zero.
  - And Adversarial error will also be zero.



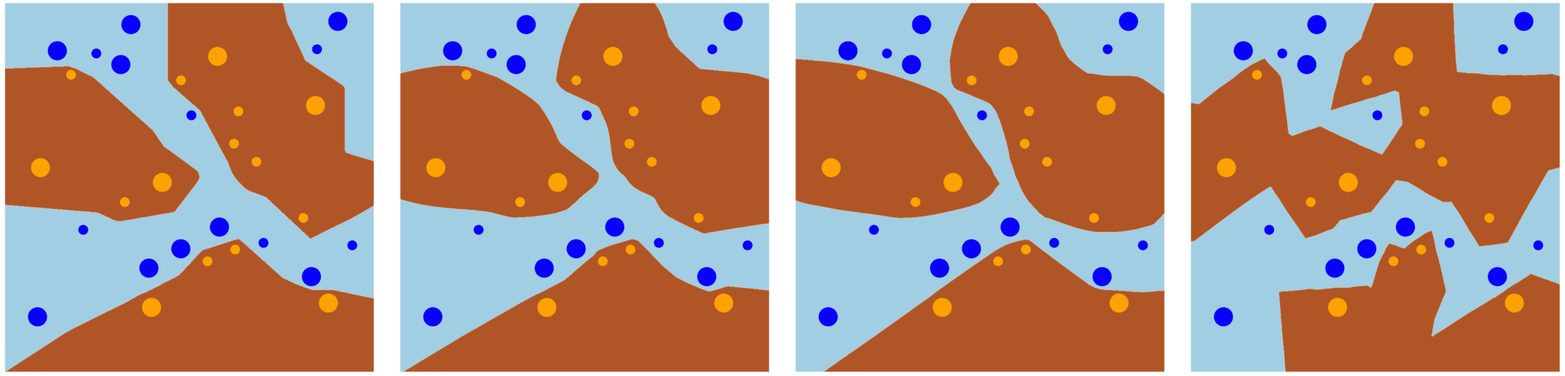
- Classifiers from B (though visually more complex) have lower VC

Theorem 2. *For some universal constant c, and any $0 < \gamma_0 < 1/\sqrt{2}$, there exists a family of distributions $\mathcal{D}$ defined on $\mathcal{X} \times \{0,1\}$ where $\mathcal{X} \subseteq \mathbb{R}^2$ such that for all distributions $\mathcal{P} \in \mathcal{D}$, and denoting by $\mathcal{S}_m = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)\}$ a sample of size m drawn i.i.d. from $\mathcal{P}$,*

(i) For any $m \geq 0$, $\mathcal{S}_m$ is linearly separable i.e., $\forall (\mathbf{x}_i, y_i) \in \mathcal{S}_m$, there exist $\mathbf{w} \in \mathbb{R}^2, w_0 \in \mathbb{R}$ s.t. $y_i \left(\mathbf{w}^\top \mathbf{x}_i + w_0\right) \geq 0$. Furthermore, for every $\gamma > \gamma_0$, any linear separator f that perfectly fits the training data $\mathcal{S}_m$ has $\mathcal{R}_{\mathrm{Adv},\gamma}(f;\mathcal{P}) \geq 0.0005$, even though $\mathcal{R}(f;\mathcal{P}) \to 0$ as $m \to \infty$.

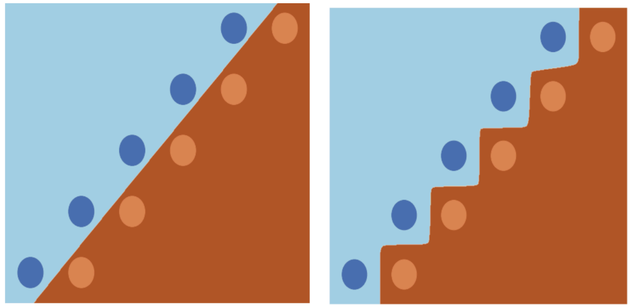
(ii) There exists a function class $\mathcal{H}$ such that for some $m \in O(\log(\delta^{-1}))$, any $h \in \mathcal{H}$ that perfectly fits the $\mathcal{S}_m$, satisfies with probability at least $1-\delta$, $\mathcal{R}(h;\mathcal{P}) = 0$ and $\mathcal{R}_{\mathrm{Adv},\gamma}(h;\mathcal{P}) = 0$, for any $\gamma \in [0, \gamma_0 + 1/8]$.

Neural networks learn “simple” boundaries.



(a) Shallow NN (b) Shallow-Wide NN (c) Deep NN (d) Large Margin

Figure 9: Decision boundaries of neural networks are much simpler than they should be.

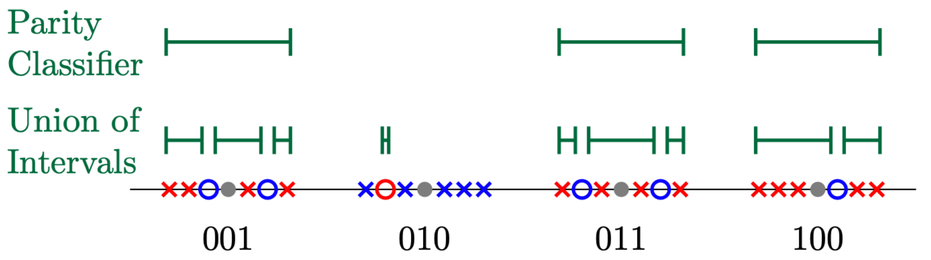


NATURAL AT

Representation Learning with label noise

There exists a (noisy) distribution such that

- With representation A:
  - Both training error and test error can be zero.
  - But Adversarial error will be large.
- With representation B:
  - Test error will be zero.
  - Training error cannot be zero.
  - And Adversarial error will be zero.
- Both Classifiers have polynomial sample complexity.



Theorem 3. *[Formal version of Theorem 3] For any $n \in \mathbb{Z}_+$, there exists a family of distributions $\mathcal{D}^n$ over $\mathbb{R} \times \{0,1\}$ and function classes $\mathcal{C}, \mathcal{H}$, such that for any $\mathcal{P}$ from $\mathcal{D}^n$, and for any $0 < \gamma < 1/4$, and $\eta \in (0, 1/2)$ if $\mathcal{S}_m = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ denotes a sample of size m where*

$$m = O\left(\max\left\{n \log \frac{n}{\delta}\left(\frac{(1-\eta)}{(1-2\eta)^2} + 1\right), \frac{n}{\eta\gamma^2}\log\left(\frac{n}{\gamma\delta}\right)\right\}\right)$$

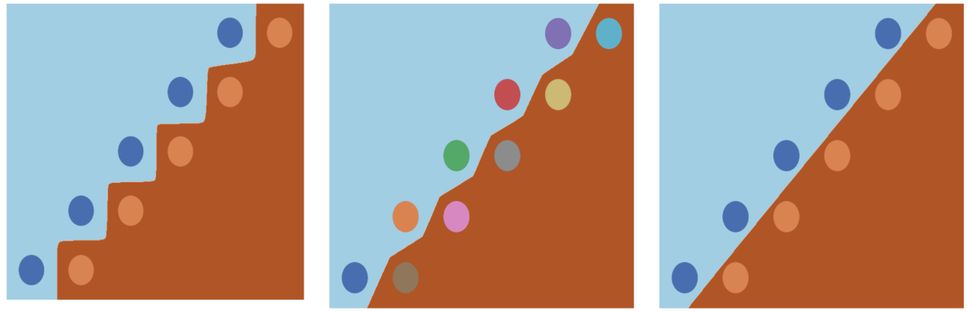
drawn from $\mathcal{P}$, and if $\mathcal{S}_{m,\eta}$ denotes the sample where each label is flipped independently with probability η.

(i) the classifier $c \in \mathcal{C}$ that minimizes the training error on $\mathcal{S}_{m,\eta}$, has $\mathcal{R}(c;\mathcal{P}) = 0$ and $\mathcal{R}_{\mathrm{Adv},\gamma}(c;\mathcal{P}) = 0$ for $0 \le \gamma < 1/4$.

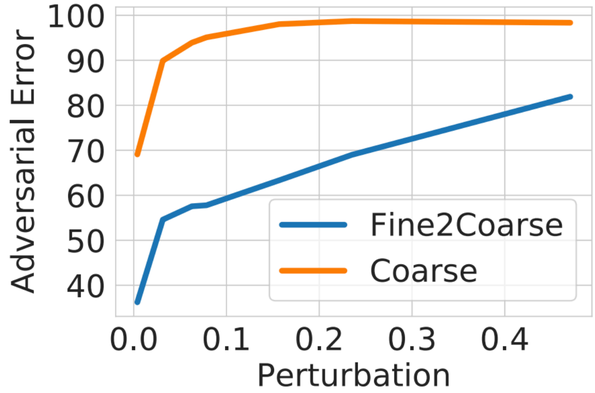
(ii) there exist $h \in \mathcal{H}$, h has zero training error on $\mathcal{S}_{m,\eta}$, and $\mathcal{R}(h;\mathcal{P}) = 0$. However, for any $\gamma > 0$, and for any $h \in \mathcal{H}$ with zero training error on $\mathcal{S}_{m,\eta}$, $\mathcal{R}_{\mathrm{Adv},\gamma}(h;\mathcal{P}) \ge 0.1$.

Furthermore, the required $c \in \mathcal{C}$ and $h \in \mathcal{H}$ above can be computed in $O\left(\mathrm{poly}\left(n\right), \mathrm{poly}\left(\frac{1}{\frac{1}{2}-\eta}\right), \mathrm{poly}\left(\frac{1}{\delta}\right)\right)$ time.

Intriguing Experiment



(a) Decision Region of neural networks are more complex for adversarially trained models. Treating it as a multi-class classification problem, with natural training (MULTICLASS), also increases robustness by increasing the margin.



(b) Adversarial error on coarse labels of CIFAR-100.

Figure 10: Assigning a separate class to each sub-population within the original class during training increases robustness by learning more meaningful representations.