Mathematical Theory of Robustness of Neural Networks

EDIC Candidacy Exam

Thomas Weinberger

EPFL
LTHC

February 2023

Overview

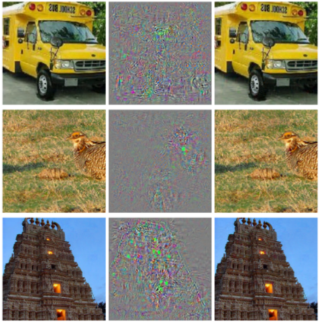
- Introduction
- **[Bubeck et al., NeurIPS '21]:** Typical random neural networks are non-robust
- **[Vardi et al., NeurIPS '22]:** Gradient flow is biased towards selecting non-robust neural networks
- **[Schmidt et al., NeurIPS '18]:** Learning robustly is much harder than only learning
- Research proposal

Introduction: Robustness of NNs

- **Empirical** studies ca. 2014[1]: classification with neural networks highly susceptible to small perturbations on the input

[1] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R., 2014. Intriguing properties of neural networks. ICLR 2014.

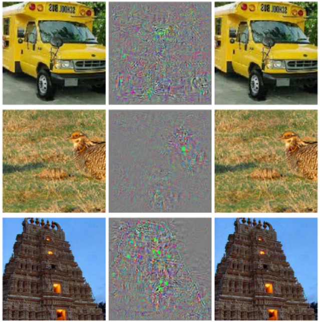
- **Empirical** studies ca. 2014[1]: classification with neural networks highly susceptible to small perturbations on the input
- Perturbations typically imperceptible to the human eye



[1] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R., 2014. Intriguing properties of neural networks. ICLR 2014.

Introduction: Robustness of NNs

- **Empirical** studies ca. 2014[1]: classification with neural networks highly susceptible to small perturbations on the input
- Perturbations typically imperceptible to the human eye



- **Theory** still rather poorly understood → recent results

[1] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R., 2014. Intriguing properties of neural networks. ICLR 2014.

Paper 1

Bubeck, S., Cherapanamjeri, Y., Gidel, G. and Tachet des Combes, R., 2021. **A single gradient step finds adversarial examples on random two-layers neural networks.** Advances in Neural Information Processing Systems, 34.

Setup

- **Random** two-layer neural networks $f : \mathbb{R}^d \to \mathbb{R}$

$$f(x) = \frac{1}{\sqrt{k}} \sum_{l=1}^{k} a_l \cdot \sigma(w_l^\top x) \tag{1}$$

- d: input dimension, k: $\#_{\text{neurons}}$ in hidden layer, $\sigma : \mathbb{R} \to \mathbb{R}$: non-linearity
- $w_l \sim \mathcal{N}(0, \frac{1}{d} I_d)$, $a_l \sim \mathcal{U}(\{-1, +1\})$ all i.i.d.

Setup

- **Random** two-layer neural networks $f : \mathbb{R}^d \to \mathbb{R}$

$$f(x) = \frac{1}{\sqrt{k}} \sum_{l=1}^{k} a_l \cdot \sigma(w_l^\top x) \tag{1}$$

- d: input dimension, k: $\#_{\text{neurons}}$ in hidden layer, $\sigma : \mathbb{R} \to \mathbb{R}$: non-linearity
- $w_l \sim \mathcal{N}(0, \frac{1}{d} I_d)$, $a_l \sim \mathcal{U}(\{-1, +1\})$ all i.i.d.
- **Task:** binary classification based on $\text{sign}(f(x))$

Setup

- **Random** two-layer neural networks $f : \mathbb{R}^d \to \mathbb{R}$

$$f(x) = \frac{1}{\sqrt{k}} \sum_{l=1}^{k} a_l \cdot \sigma(w_l^\top x) \tag{1}$$

- d: input dimension, k: $\#_{\text{neurons}}$ in hidden layer, $\sigma : \mathbb{R} \to \mathbb{R}$: non-linearity
- $w_l \sim \mathcal{N}(0, \frac{1}{d} I_d)$, $a_l \sim \mathcal{U}(\{-1, +1\})$ all i.i.d.
- **Task:** binary classification based on $\text{sign}(f(x))$
- Data: $x \in \sqrt{d} \cdot \mathbb{S}^{d-1}$

Setup

- **Random** two-layer neural networks $f : \mathbb{R}^d \to \mathbb{R}$

$$f(x) = \frac{1}{\sqrt{k}} \sum_{l=1}^{k} a_l \cdot \sigma(w_l^\top x) \tag{1}$$

- d: input dimension, k: $\#_{\text{neurons}}$ in hidden layer, $\sigma : \mathbb{R} \to \mathbb{R}$: non-linearity
- $w_l \sim \mathcal{N}(0, \frac{1}{d} I_d)$, $a_l \sim \mathcal{U}(\{-1, +1\})$ all i.i.d.
- **Task:** binary classification based on $\operatorname{sign}(f(x))$
- Data: $x \in \sqrt{d} \cdot \mathbb{S}^{d-1}$
- **Question:** how large δ needed s.t. $\forall x : \operatorname{sign}(f(x+\delta)) \neq \operatorname{sign}(f(x))$?

Setup

- **Random** two-layer neural networks $f : \mathbb{R}^d \to \mathbb{R}$

$$f(x) = \frac{1}{\sqrt{k}} \sum_{l=1}^{k} a_l \cdot \sigma(w_l^\top x) \tag{1}$$

- d: input dimension, k: $\#_{\text{neurons}}$ in hidden layer, $\sigma : \mathbb{R} \to \mathbb{R}$: non-linearity
- $w_l \sim \mathcal{N}(0, \frac{1}{d} I_d)$, $a_l \sim \mathcal{U}(\{-1, +1\})$ all i.i.d.
- **Task:** binary classification based on $\text{sign}(f(x))$
- Data: $x \in \sqrt{d} \cdot \mathbb{S}^{d-1}$
- **Question:** how large δ needed s.t. $\forall x : \text{sign}(f(x+\delta)) \neq \text{sign}(f(x))$?
  - W.h.p. over weights
  - Size of δ: ℓ_2-norm

The High Level Idea

- For k large + our scale of input/weights + σ "nice", should expect w.h.p. (CLT)

$$|f(x)| \in \Theta(1). \tag{2}$$

The High Level Idea

- For k large + our scale of input/weights + σ "nice", should expect w.h.p. (CLT)

$$|f(x)| \in \Theta(1). \tag{2}$$

- Bounded gradient:

$$\|\nabla f(x)\| \in \Omega(1) \tag{3}$$

The High Level Idea

- For k large + our scale of input/weights + σ "nice", should expect w.h.p. (CLT)
$$|f(x)| \in \Theta(1). \tag{2}$$

- Bounded gradient:
$$\|\nabla f(x)\| \in \Omega(1) \tag{3}$$

- Gradient locally stable: for δ small,
$$\|\nabla f(x) - \nabla f(x+\delta)\| \in o(\|\nabla f(x)\|) \tag{4}$$

- Together with an *upper* bound on $\|\nabla f(x)\|$ implies that the gradient essentially is constant on the "macroscopic" scale.

The High Level Idea

- For k large + our scale of input/weights + σ "nice", should expect w.h.p. (CLT)
$$|f(x)| \in \Theta(1). \tag{2}$$
- Bounded gradient:
$$\|\nabla f(x)\| \in \Omega(1) \tag{3}$$
- Gradient locally stable: for δ small,
$$\|\nabla f(x) - \nabla f(x+\delta)\| \in o(\|\nabla f(x)\|) \tag{4}$$
- Together with an *upper* bound on $\|\nabla f(x)\|$ implies that the gradient essentially is constant on the "macroscopic" scale.
- **Combined:** apply constant sized perturbation in direction $\pm\nabla f$ to locally linear function with constant size output
$\rightarrow$ can change output to constant sized output of opposite sign!

Main Theorem

Theorem

Let $\gamma \in (0,1)$ and let σ be non-constant, Lipschitz and with Lipschitz derivative. Assume $k \geq C_1 \log^3(1/\gamma)$, $d \geq C_2 \log(k/\gamma)\log(1/\gamma)$, and let $\eta \in \mathbb{R}$ such that $|\eta| = C_3\sqrt{\log(1/\gamma)}\|\nabla f(x)\|^{-2}$ and $sign(\eta) = -sign(f(x))$. Then with probability at least $1-\gamma$:

$$sign(f(x)) \neq sign(f(x + \eta\nabla f(x))). \tag{5}$$

Moreover, we have $\|\eta\nabla f(x)\| \leq C_4\sqrt{\log(1/\gamma)}$.

Main Theorem

Theorem

Let $\gamma \in (0,1)$ and let σ be non-constant, Lipschitz and with Lipschitz derivative. Assume $k \geq C_1 \log^3(1/\gamma)$, $d \geq C_2 \log(k/\gamma)\log(1/\gamma)$, and let $\eta \in \mathbb{R}$ such that $|\eta| = C_3\sqrt{\log(1/\gamma)}\|\nabla f(x)\|^{-2}$ and $sign(\eta) = -sign(f(x))$. Then with probability at least $1-\gamma$:

$$sign(f(x)) \neq sign(f(x + \eta \nabla f(x))). \tag{5}$$

Moreover, we have $\|\eta \nabla f(x)\| \leq C_4\sqrt{\log(1/\gamma)}$.

- Covers sub-exponential width regime
- Constants depend only on σ

Lower Bound on Gradient

- Assume: σ is 1-Lipschitz, L-smooth, and $\sigma(0) = 0$.
- Then, $|\sigma(X)| \leq |X|$
- $\forall l : {w_l}^\top x \sim \mathcal{N}(0, 1)$
  $\rightarrow |f(x)| \in \Theta(1)$ follows from Bernstein's inequality
- We have
$$\nabla f(x) = \frac{1}{\sqrt{k}} \sum_{l=1}^{k} a_l w_l \sigma'({w_l}^\top x) \tag{6}$$
- Bound $\|\nabla f(x)\| \geq \|P \nabla f(x)\|$ with $P := I_d - xx^\top / d$ projection onto orthogonal complement of x
- This decouples the product of the two random variables appearing inside the sum
- Bernstein's inequality and a standard ξ^2 concentration bound

Stability of Gradient

- Evoke variational description of euclidean norm:

$$\sup_{\delta\in\mathbb{R}^d:\|\delta\|\leq R} \|\nabla f(x) - \nabla f(x+\delta)\| \tag{7}$$

$$= \sup_{v\in\mathbb{S}^{d-1}\delta\in\mathbb{R}^d:\|\delta\|\leq R} \frac{1}{\sqrt{k}} \sum_{l=1}^{k} a_l(w_l^\top v)\cdot\Big(\sigma'(w_l^\top x) - \sigma'(w_l^\top\cdot(x-\delta))\Big) \tag{8}$$

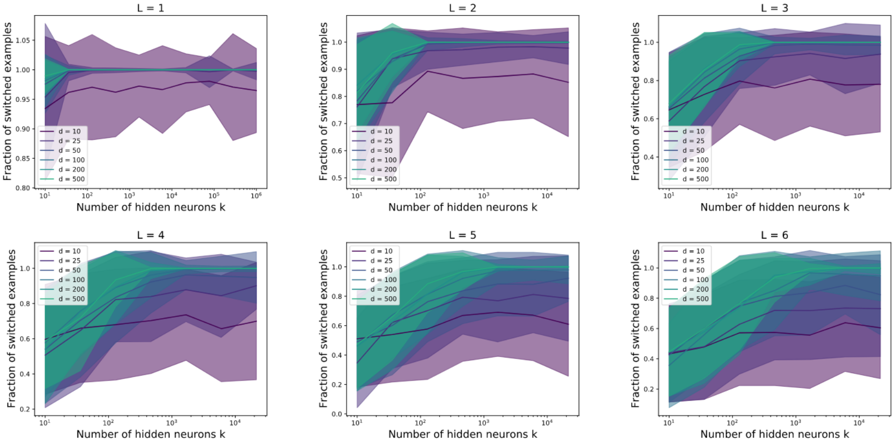
- ϵ-net jointly over v, δ with metric $\|v - v'\| + \|\delta - \delta'\|$
- Union bound and upper bound approximation error $\to$ upper bound on gradient deviation
- Finally, use following standard descend Lemma:

$$f(x - \eta\nabla f(x)) \leq f(x) - \eta\|\nabla f(x)\| \tag{9}$$

$$\times\left(\|\nabla f(x)\| - \sup_{\frac{\|\delta\|}{\|\nabla f(x)\|}\leq\eta} \|\nabla f(x) - \nabla f(x+\delta)\|\right) \tag{10}$$

Experiments

- Averaged over 100 random inputs and 100 networks
- Search over $|\eta| \leq 20$ (empirically: ≈ 1 almost always)



Conclusion

Conclusion

- Results also hold for ReLU, random inputs, Gaussian output neuron weights

Conclusion

- Results also hold for ReLU, random inputs, Gaussian output neuron weights
- Only shallow NNs

Conclusion

- Results also hold for ReLU, random inputs, Gaussian output neuron weights
- Only shallow NNs
- Doesn't explain non-robustness of trained NNs (e.g. with first order methods)

Conclusion

- Results also hold for ReLU, random inputs, Gaussian output neuron weights
- Only shallow NNs
- Doesn't explain non-robustness of trained NNs (e.g. with first order methods)
- Could be that stationary points selected by gradient descent are very "atypical"

Paper 2

Vardi, G., Yehudai, G. and Shamir, O., 2022. **Gradient Methods Provably Converge to Non-Robust Networks.** In Advances in Neural Information Processing Systems, 36.

Setup

$$f_\theta(x) = \sum_{l=1}^{k} a_l \cdot \sigma(w_l^\top x + b_l) \tag{11}$$

- $w_l \in \mathbb{R}^d$, $a, b \in \mathbb{R}^k$, stack in param. vector $\theta = [w_1, \ldots, w_k, b, a]$
- d: input dimension, k: $\#_{\text{neurons}}$ hidden layer, n: $\#_{\text{samples}}$
  $\sigma(x) = \max\{0, x\}$

Setup

$$f_\theta(x) = \sum_{l=1}^{k} a_l \cdot \sigma(w_l^\top x + b_l) \tag{11}$$

- $w_l \in \mathbb{R}^d$, $a, b \in \mathbb{R}^k$, stack in param. vector $\theta = [w_1, \ldots, w_k, b, a]$
- d: input dimension, k: $\#_{\text{neurons}}$ hidden layer, n: $\#_{\text{samples}}$
  $\sigma(x) = \max\{0, x\}$
- **Assumption:** data sufficiently separated: $|\langle x_i, x_j \rangle| \in o(d)$ for all $i \neq j$

Setup

$$f_\theta(x) = \sum_{l=1}^{k} a_l \cdot \sigma(w_l^\top x + b_l) \tag{11}$$

- $w_l \in \mathbb{R}^d$, $a, b \in \mathbb{R}^k$, stack in param. vector $\theta = [w_1, \ldots, w_k, b, a]$
- d: input dimension, k: $\#_{\text{neurons}}$ hidden layer, n: $\#_{\text{samples}}$
  $\sigma(x) = \max\{0, x\}$
- **Assumption:** data sufficiently separated: $|\langle x_i, x_j \rangle| \in o(d)$ for all $i \neq j$
- Holds w.h.p. when $x_i \sim \mathcal{U}(\sqrt{d} \cdot \mathbb{S}^{d-1})$ i.i.d. and $n \in \mathcal{O}(\text{poly}(d))$

Setup

$$f_\theta(x) = \sum_{l=1}^{k} a_l \cdot \sigma(w_l^\top x + b_l) \tag{11}$$

- $w_l \in \mathbb{R}^d$, $a, b \in \mathbb{R}^k$, stack in param. vector $\theta = [w_1, \ldots, w_k, b, a]$
- d: input dimension, k: $\#_{\text{neurons}}$ hidden layer, n: $\#_{\text{samples}}$
  $\sigma(x) = \max\{0, x\}$
- **Assumption:** data sufficiently separated: $|\langle x_i, x_j \rangle| \in o(d)$ for all $i \neq j$
- Holds w.h.p. when $x_i \sim \mathcal{U}(\sqrt{d} \cdot \mathbb{S}^{d-1})$ i.i.d. and $n \in \mathcal{O}(\text{poly}(d))$
- Empirical loss

$$\mathcal{L}(\theta) := \sum_{i=1}^{n} \ell(y_i f_\theta(x_i)). \tag{12}$$

with either $\ell(z) = e^{-z}$ or $\ell(z) = \log(1 + e^{-z})$.

Robust Networks Exist

Achieving robustness is not hard for separated data:

Robust Networks Exist

Achieving robustness is not hard for separated data:

Theorem (Robust Networks Exist)

Assume that for all $i \neq j$ it holds that $|\langle x_i, x_j \rangle| \leq c \cdot d$, where $0 < c < 1$. Then, there always exists some f_θ such that for every x_i, an adversarial perturbation must be of size at least $\Omega(\sqrt{d})$.

Robust Networks Exist

Achieving robustness is not hard for separated data:

Theorem (Robust Networks Exist)

Assume that for all $i \neq j$ it holds that $|\langle x_i, x_j \rangle| \leq c \cdot d$, where $0 < c < 1$. Then, there always exists some f_θ such that for every x_i, an adversarial perturbation must be of size at least $\Omega(\sqrt{d})$.

Proof sketch: construct NN such that exactly one neuron is active per training sample. Then, requires $\|\delta\| \in \Omega(\sqrt{d})$ to turn off neuron and turn on other neuron of opposite sign.

Dynamics and KKT points

- Start at $\theta(0)$ and perform **gradient flow** on the empirical loss:

$$\frac{d\theta(t)}{dt} \in -\partial^{\circ}\mathcal{L}(\theta(t)) \tag{13}$$

- Convergence in direction of $\theta(t)$ to $\tilde{\theta}$: $\lim_{t\to\infty} \frac{\theta(t)}{\|\theta(t)\|} = \frac{\tilde{\theta}}{\|\tilde{\theta}\|}$

Dynamics and KKT points

- Start at $\theta(0)$ and perform **gradient flow** on the empirical loss:

$$\frac{d\theta(t)}{dt} \in -\partial^\circ \mathcal{L}(\theta(t)) \tag{13}$$

- Convergence in direction of $\theta(t)$ to $\tilde{\theta}$: $\lim_{t\to\infty} \frac{\theta(t)}{\|\theta(t)\|} = \frac{\tilde{\theta}}{\|\tilde{\theta}\|}$

Theorem (GF and KKT Points, Lyu and Li '19, Ji and Telgarsky '20)

Let f_θ be a homogenous ReLU network. Consider minimizing either the exponential or logistic loss using gradient flow.
Assume that $\exists t_0$ s.t. $\mathcal{L}(\theta(t_0)) < 1$, that is, $y_i f_{\theta(t_0)}(x_i) > 0$ for every x_i.
Then, gradient flow converges in direction to a first order stationary point (KKT point) of the following maximum margin problem in param. space:

$$\min_\theta \frac{1}{2}\|\theta\|^2 \;\; s.t. \;\; \forall i \in [n] \;\; y_i f_\theta(x_i) \geq 1 \tag{14}$$

Moreover, $\mathcal{L}(\theta(t)) \to 0$ and $\|\theta(t)\| \to \infty$ as $t \to \infty$.

Main Theorem

Theorem (Main Theorem)

Let $\{(x_i, y_i)\}_{i=1}^n \subset (\sqrt{d} \cdot \mathbb{S}^{d-1}) \times \{\pm 1\}$, such that the two classes are balanced (at least a constant fraction each) and let $n \leq \frac{d+1}{3(\max_{i \neq j} |\langle x_i, x_j \rangle| + 1)}$. Let f_θ be a network with θ a KKT point as above. Then there exists a vector $\delta = \eta \sum_{i=1}^n y_i x_i$ for some $\eta > 0$ with $|\eta| \in \mathcal{O}\left(\sqrt{\frac{d}{c^2 n}}\right)$ which is a universal perturbation over the whole training set, i.e., $\forall i \in [n] : sign(f_\theta(x_i - y_i \delta)) = -y_i$.

Main Theorem

Theorem (Main Theorem)

Let $\{(x_i, y_i)\}_{i=1}^n \subset (\sqrt{d} \cdot \mathbb{S}^{d-1}) \times \{\pm 1\}$, such that the two classes are balanced (at least a constant fraction each) and let $n \leq \frac{d+1}{3(\max_{i \neq j} |\langle x_i, x_j \rangle| + 1)}$. Let f_θ be a network with θ a KKT point as above. Then there exists a vector $\delta = \eta \sum_{i=1}^n y_i x_i$ for some $\eta > 0$ with $|\eta| \in \mathcal{O}\left(\sqrt{\frac{d}{c^2 n}}\right)$ which is a universal perturbation over the whole training set, i.e., $\forall i \in [n] : sign(f_\theta(x_i - y_i \delta)) = -y_i$.

- If $x_i \sim \mathcal{U}(\sqrt{d} \cdot \mathbb{S}^{d-1})$, then w.h.p. $\max_{i \neq j} |\langle x_i, x_j \rangle| \in \mathcal{O}(\sqrt{d} \log(d))$ and hence for $n \in \Theta\left(\frac{\sqrt{d}}{\log(d))}\right)$ we have $\|\delta\| \in o(\sqrt{d})$

Main Theorem

Theorem (Main Theorem)

Let $\{(x_i, y_i)\}_{i=1}^n \subset (\sqrt{d} \cdot \mathbb{S}^{d-1}) \times \{\pm 1\}$, such that the two classes are balanced (at least a constant fraction each) and let $n \leq \frac{d+1}{3(\max_{i\neq j} |\langle x_i, x_j \rangle| + 1)}$. Let f_θ be a network with θ a KKT point as above. Then there exists a vector $\delta = \eta \sum_{i=1}^n y_i x_i$ for some $\eta > 0$ with $|\eta| \in \mathcal{O}\left(\sqrt{\frac{d}{c^2 n}}\right)$ which is a universal perturbation over the whole training set, i.e., $\forall i \in [n] : sign(f_\theta(x_i - y_i\delta)) = -y_i$.

- If $x_i \sim \mathcal{U}(\sqrt{d} \cdot \mathbb{S}^{d-1})$, then w.h.p. $\max_{i\neq j} |\langle x_i, x_j \rangle| \in \mathcal{O}(\sqrt{d} \log(d))$ and hence for $n \in \Theta\left(\frac{\sqrt{d}}{\log(d))}\right)$ we have $\|\delta\| \in o(\sqrt{d})$
- Independent of the width and number of parameters

Experiments

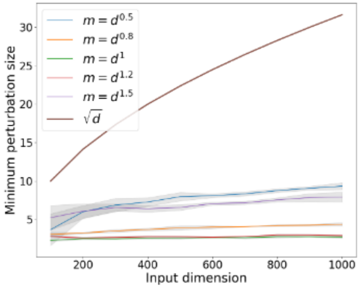
- $x_i \sim \mathcal{U}(\sqrt{d}\mathbb{S}^{d-1})$, $y_i \sim \mathcal{U}(\{\pm 1\})$

Experiments

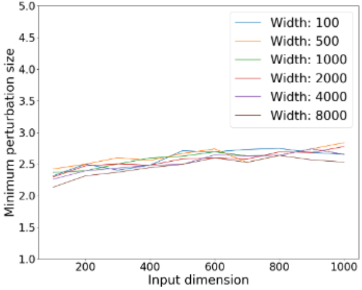
- $x_i \sim \mathcal{U}(\sqrt{d}\mathbb{S}^{d-1})$, $y_i \sim \mathcal{U}(\{\pm 1\})$
- Trained with SGD and exponential loss until $\mathcal{L} \leq 10^{-30}$

Experiments

- $x_i \sim \mathcal{U}(\sqrt{d}\mathbb{S}^{d-1})$, $y_i \sim \mathcal{U}(\{\pm 1\})$
- Trained with SGD and exponential loss until $\mathcal{L} \leq 10^{-30}$
- Consider maximum η_{min} over all n samples
- Averaged over 5 networks per data point



(a)



(b)

Conclusion

Conclusion

- Proof technique only applicable to shallow NNs

Conclusion

- Proof technique only applicable to shallow NNs
- Strong separation condition; implies upper bound on dataset size

Conclusion

- Proof technique only applicable to shallow NNs
- Strong separation condition; implies upper bound on dataset size
- Doesn't allow for clusters of points within which examples have large inner products

Conclusion

- Proof technique only applicable to shallow NNs
- Strong separation condition; implies upper bound on dataset size
- Doesn't allow for clusters of points within which examples have large inner products
- Experiments: separation condition probably too pessimistic

Conclusion

- Proof technique only applicable to shallow NNs
- Strong separation condition; implies upper bound on dataset size
- Doesn't allow for clusters of points within which examples have large inner products
- Experiments: separation condition probably too pessimistic
- Universal perturbations

Conclusion

- Proof technique only applicable to shallow NNs
- Strong separation condition; implies upper bound on dataset size
- Doesn't allow for clusters of points within which examples have large inner products
- Experiments: separation condition probably too pessimistic
- Universal perturbations
- Robust networks exist but gradient flow converges to non-robust ones $\rightarrow$ yet another implicit bias of neural networks trained with gradient based methods

Paper 3

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K. and Madry, A., 2018. **Adversarially robust generalization requires more data.** Advances in neural information processing systems, 31.

- For any learning task, robustness alone is trivially achievable (constant hypothesis)

- For any learning task, robustness alone is trivially achievable (constant hypothesis)
- **Natural question:** inherent trade-off between robustness and generalization?

- For any learning task, robustness alone is trivially achievable (constant hypothesis)
- **Natural question:** inherent trade-off between robustness and generalization?
- This paper answers question for two simple learning tasks

- For any learning task, robustness alone is trivially achievable (constant hypothesis)
- **Natural question:** inherent trade-off between robustness and generalization?
- This paper answers question for two simple learning tasks
- Holds for any learning algorithm, including NNs trained with ERM

- For any learning task, robustness alone is trivially achievable (constant hypothesis)
- **Natural question:** inherent trade-off between robustness and generalization?
- This paper answers question for two simple learning tasks
- Holds for any learning algorithm, including NNs trained with ERM
- Separation between (linear) vs. (non-linearity $\circ$ linear) classifiers (not shown here)

Setup

Binary classification with $0-1$ loss

Setup

Binary classification with $0 - 1$ loss

Definition (Standard Classification Error)

Let $\mathcal{D} : \mathbb{R}^d \times \{\pm 1\} \to \mathbb{R}$ be a distribution. Then, the classification error β of a classifier f is defined as

$$\beta := \mathbb{P}_{(x,y)\sim\mathcal{D}}(f(x') \neq y) \tag{15}$$

Setup

Binary classification with $0-1$ loss

Definition (Standard Classification Error)

Let $\mathcal{D} : \mathbb{R}^d \times \{\pm 1\} \to \mathbb{R}$ be a distribution. Then, the classification error β of a classifier f is defined as

$$\beta := \mathbb{P}_{(x,y)\sim\mathcal{D}}(f(x') \neq y) \tag{15}$$

Definition (Robust Classification Error)

The ($\mathcal{B}$-)robust classification error β_r of a classifier f is defined as $\beta_r := \mathbb{P}_{(x,y)\sim\mathcal{D}}[\exists x' \in \mathcal{B}(x) : f(x') \neq y]$.

Here: $\mathcal{B}(x) = \mathcal{B}^{\epsilon}_{\infty}(x) = \{x' \in \mathbb{R}^d | \|x' - x\|_{\infty} \leq \epsilon\}$

- **Considered regime:** single sample sufficient to obtain low standard classification error (w.h.p.) from a single sample.

- **Considered regime:** single sample sufficient to obtain low standard classification error (w.h.p.) from a single sample.
- Applies to following two data models + linear classifiers:

- **Considered regime:** single sample sufficient to obtain low standard classification error (w.h.p.) from a single sample.
- Applies to following two data models + linear classifiers:

Definition (Gaussian Model)

Let θ^* be a per-class mean vector and let $\sigma > 0$ be the variance. A (θ^*, σ)-Gaussian model is defined by the distribution over $\mathbb{R}^d \times \{\pm 1\}$ by first drawing a label $y \in \{\pm 1\}$ uniformly at random and then sampling the input point $x \in \mathbb{R}^d$ from $\mathcal{N}(y \cdot \theta^*, \sigma^2 I)$.

- **Considered regime:** single sample sufficient to obtain low standard classification error (w.h.p.) from a single sample.
- Applies to following two data models + linear classifiers:

Definition (Gaussian Model)

Let θ^* be a per-class mean vector and let $\sigma > 0$ be the variance. A (θ^*, σ)-Gaussian model is defined by the distribution over $\mathbb{R}^d \times \{\pm 1\}$ by first drawing a label $y \in \{\pm 1\}$ uniformly at random and then sampling the input point $x \in \mathbb{R}^d$ from $\mathcal{N}(y \cdot \theta^*, \sigma^2 I)$.

Definition (Bernoulli Model)

Let $\theta^* \in \{\pm 1\}^d$ and let $\tau > 0$. Then the $(\theta^*, \tau)-$Bernoulli model is defined by the following distribution over $(x, y) \in \{\pm\}^d \times \{\pm 1\}$: First, draw a label y uniformly at random from $\{\pm 1\}$. Then sample the data point $x \in \{\pm 1\}^d$ by sampling each coordinate according to

$$x_i = \begin{cases} y \cdot \theta_i^* & \textit{with probability } 1/2 + \tau \\ -y \cdot \theta_i^* & \textit{with probability } 1/2 - \tau \end{cases}$$

Separation under Gaussian Models

Given w, define linear classifier $f_w : \mathbb{R}^d \to \{\pm 1\}$ as $f_w(x) = \text{sgn}(\langle w, x \rangle)$

Separation under Gaussian Models

Given w, define linear classifier $f_w : \mathbb{R}^d \to \{\pm 1\}$ as $f_w(x) = \text{sgn}(\langle w, x \rangle)$

Theorem

Let (x, y) be drawn from a (θ^*, σ)-Gaussian model with $\|\theta^*\|_2 = \sqrt{d}$ and $\sigma \leq c \cdot d^{1/4}$. Let $\hat{w} \in \mathbb{R}^d$ be the vector $\hat{w} = y \cdot x$. Then w.h.p., the linear classifier $f_{\hat{w}}$ has classification error at most 1%.

Separation under Gaussian Models

Given w, define linear classifier $f_w : \mathbb{R}^d \to \{\pm 1\}$ as $f_w(x) = \text{sgn}(\langle w, x \rangle)$

Theorem

Let (x, y) be drawn from a (θ^*, σ)-Gaussian model with $\|\theta^*\|_2 = \sqrt{d}$ and $\sigma \leq c \cdot d^{1/4}$. Let $\hat{w} \in \mathbb{R}^d$ be the vector $\hat{w} = y \cdot x$. Then w.h.p., the linear classifier $f_{\hat{w}}$ has classification error at most 1%.

Theorem

Let $\{x_i, y_i\}_{i=1}^n$ be drawn i.i.d. from a (θ^*, σ)-Gaussian model with $\|\theta^*\|_2 = \sqrt{d}$ and $\sigma \leq c_1 d^{1/4}$. Let $\hat{w} = \sum_{i=1}^n y_i x_i$. Then w.h.p., the linear classifier $f_{\hat{w}}$ has ℓ_∞^ϵ-robust classification error at most 1% if

$$n \geq \begin{cases} 1 & \text{for} \quad \epsilon \leq \frac{1}{4} d^{-1/4} \\ c_2 \epsilon^2 \sqrt{d} & \text{for} \quad \frac{1}{4} d^{-1/4} \leq \epsilon \leq \frac{1}{4} \end{cases}.$$

Theorem

Let g_n be any learning algorithm, i.e., a function mapping n samples to a binary classifier $f(x)$. Let $\sigma = c_1 d^{1/4}$, let $\epsilon \geq 0$, and let $\theta \in \mathbb{R}^d$ be drawn from $\mathcal{N}(0, I)$. Moreover, let the samples be drawn from the (θ, σ)-Gaussian model. Then, the expected ℓ_∞^ϵ- robust classification error of f_n is at least $(1 - \frac{1}{d})\frac{1}{2}$ if

$$n \leq c_2 \frac{\epsilon^2 \sqrt{d}}{\log d} \tag{16}$$

Theorem

Let g_n be any learning algorithm, i.e., a function mapping n samples to a binary classifier $f(x)$. Let $\sigma = c_1 d^{1/4}$, let $\epsilon \geq 0$, and let $\theta \in \mathbb{R}^d$ be drawn from $\mathcal{N}(0, I)$. Moreover, let the samples be drawn from the (θ, σ)-Gaussian model. Then, the expected ℓ_∞^ϵ- robust classification error of f_n is at least $(1 - \frac{1}{d})\frac{1}{2}$ if

$$n \leq c_2 \frac{\epsilon^2 \sqrt{d}}{\log d} \tag{16}$$

Together with previous Thm.: Robust sample complexity in the range

$$c \frac{\epsilon^2 \sqrt{d}}{\log d} \leq n \leq c' \epsilon^2 \sqrt{d} \tag{17}$$

Bernoulli Model

Theorem

Let (x, y) be drawn from a (θ^*, τ)-Bernoulli model with $\tau \geq c \cdot d^{-1/4}$ where c is a universal constant. Let $\hat{w} \in \mathbb{R}^d$ be the vector $\hat{w} = y \cdot x$. Then with probability, the linear classifier $f_{\hat{w}}$ has classification error at most 1%.

Theorem (Informal)

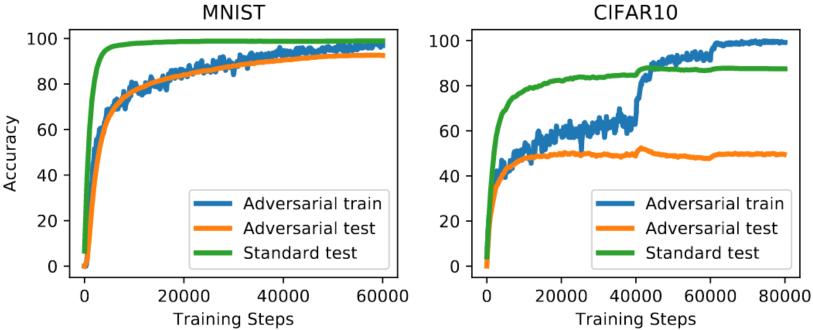
Assume that the data is generated by the Bernoulli model. Then, a linear model f has expected ℓ_∞^ϵ-robust classification error of at least $\frac{1}{2} - \gamma$

$$n \in \tilde{\mathcal{O}}(\gamma^2 \cdot d) \tag{18}$$

while $f \circ sign$ has ℓ_∞^ϵ-robust classification error at most 1% when using a single sample.

Experiments

- CNNs trained with robustness maximization algorithm
- Compare standard classification error to robust classification error on MNIST and CIFAR-10



Conclusion

Conclusion

- Results pessimistic: existence of a familiy of distributions (Gaussian model) with separation between the two metrics

Conclusion

- Results pessimistic: existence of a familiy of distributions (Gaussian model) with separation between the two metrics
- Perhaps separation vanishes for:

Conclusion

- Results pessimistic: existence of a familiy of distributions (Gaussian model) with separation between the two metrics
- Perhaps separation vanishes for:
  - Structured data models (e.g. distributions on low-dimensional manifolds)

Conclusion

- Results pessimistic: existence of a familiy of distributions (Gaussian model) with separation between the two metrics
- Perhaps separation vanishes for:
  - Structured data models (e.g. distributions on low-dimensional manifolds)
  - Learning tasks where single-sample standard generalization is not possible

Conclusion

- Results pessimistic: existence of a familiy of distributions (Gaussian model) with separation between the two metrics
- Perhaps separation vanishes for:
  - Structured data models (e.g. distributions on low-dimensional manifolds)
  - Learning tasks where single-sample standard generalization is not possible
- Specialized results for NNs or models trained with first order methods?

Conclusion

- Results pessimistic: existence of a familiy of distributions (Gaussian model) with separation between the two metrics
- Perhaps separation vanishes for:
  - Structured data models (e.g. distributions on low-dimensional manifolds)
  - Learning tasks where single-sample standard generalization is not possible
- Specialized results for NNs or models trained with first order methods?
- Similar results for ℓ_2-perturbations?

Research Proposal

Research Proposal

- Sample complexity of neural networks, **not** ignoring the training algorithm, and **not** just under worst-case distributions

Research Proposal

- Sample complexity of neural networks, **not** ignoring the training algorithm, and **not** just under worst-case distributions
- Generalization/robustness under realistic data models, e.g., low-dim. manifolds, clusters, sparse models

Research Proposal

- Sample complexity of neural networks, **not** ignoring the training algorithm, and **not** just under worst-case distributions
- Generalization/robustness under realistic data models, e.g., low-dim. manifolds, clusters, sparse models
- Separation of sample complexity: NNs vs Kernels, linear classifiers, ...

Research Proposal

- Sample complexity of neural networks, **not** ignoring the training algorithm, and **not** just under worst-case distributions
- Generalization/robustness under realistic data models, e.g., low-dim. manifolds, clusters, sparse models
- Separation of sample complexity: NNs vs Kernels, linear classifiers, ...
- Generalization measures: beyond VC, beyond flatness

Thank you!
Questions?

Bubeck, S., Cherapanamjeri, Y., Gidel, G. and Tachet des Combes, R., 2021. **A single gradient step finds adversarial examples on random two-layers neural networks.** Advances in Neural Information Processing Systems, 34.

Vardi, G., Yehudai, G. and Shamir, O., 2022. **Gradient Methods Provably Converge to Non-Robust Networks.** In Advances in Neural Information Processing Systems, 36.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K. and Madry, A., 2018. **Adversarially robust generalization requires more data.** Advances in neural information processing systems, 31.