Adversarial Training: theories and applications

Yan Li

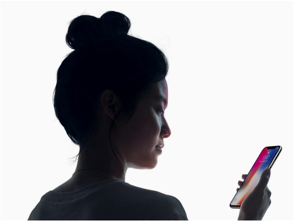
Collaboration with: Haoming Jiang (Gatech), Qianli Shen (Peking U), Ethan X. Fang (Penn State), Zhaoran Wang (Northwestern), Huan Xu (Alibaba), Tuo Zhao (Gatech)

December 2020

Background

Successes of Deep Learning (DL)

- Face recognition
  - bio-authentification
  - security

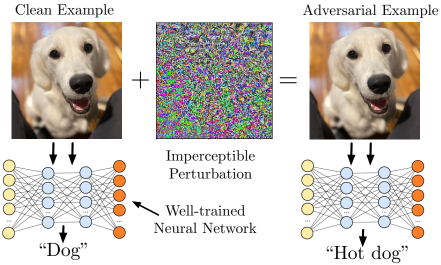


- Natural language processing
  - machine translation
  - machine understanding
  - text generation

- Planning
  - autonomous driving

Blindspots of DL: adversarial examples

Deep neural networks are vulnerable to adversarial examples (Goodfellow et al. 2014).

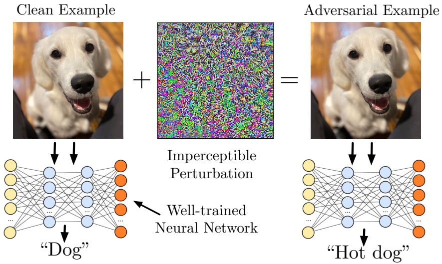


Mathematical description: Given model $f(\cdot, \theta)$, loss function $\ell(\cdot, \cdot)$, data point (x, y), a (specified) perturbation set $\mathcal{B}$.

$$\widehat{x} = \arg\max_{x' \in \{x\} + \mathcal{B}} \ell\left(f(x', \theta), y\right).$$

Blindspots of DL: adversarial examples

Deep neural networks are vulnerable to adversarial examples (Goodfellow et al. 2014).



Mathematical description: Given model $f(\cdot, \theta)$, loss function $\ell(\cdot, \cdot)$, data point (x, y), a (specified) perturbation set $\mathcal{B}$.

$$\widehat{x} = \arg\max_{x' \in \{x\} + \mathcal{B}} \ell\left(f(x', \theta), y\right).$$

Defend against adversarial examples

Provable defense:

- discrete optimization: (Tjeng et al. 2017).
  – heavy computation, no scalable
- randomized smoothing: (Cohen et al. 2019).
  – scalable to ImageNet level dataset.
  – hard to defend against ℓ_∞ attack: $\mathcal{B} = \{\delta : \|\delta\|_\infty \leq \epsilon\}$.

Summary: limited practical performance, assumes adversary has infinite computational power (reasonable?).

Adversarial Training (AT), (Madry et al 2017)

$$\min_{\theta} \widehat{L}_{\mathrm{adv}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \max_{\delta_i \in \mathcal{B}} \ell(f(\underbrace{x_i + \delta_i}_{\text{Adversarial Example:}\widehat{x}_i};\theta), y_i).$$

- Great empirical performance. Building block for most defense methods. Matches state-of-art algorithm with early-stopping (Rice et al. 2020)
- Adaptive robustness: defending against stronger attack → more robust model (Gao et al. 2019).
  – gradient descent based adversary (GDAT).
- Lack of theoretical guarantees.

Adversarial Training (AT), (Madry et al 2017)

$$\min_{\theta} \widehat{L}_{\text{adv}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \max_{\delta_i \in \mathcal{B}} \ell(f(\underbrace{x_i + \delta_i}_{\text{Adversarial Example:}\widehat{x}_i};\theta), y_i).$$

- Great empirical performance. Building block for most defense methods. Matches state-of-art algorithm with early-stopping (Rice et al. 2020)
- Adaptive robustness: defending against stronger attack $\rightarrow$ more robust model (Gao et al. 2019).
  – gradient descent based adversary (GDAT).
- Lack of theoretical guarantees.

Outline

Address the following questions, (tentatively)

- **Q**: How does AT achieve robustness?

  A: Understand AT through its algorithmic implicit bias.

- **Q**: AT beyond robustness?

  A: Improve reinforcement learning (RL) algorithm with AT.

Outline

Address the following questions, (tentatively)

- **Q**: How does AT achieve robustness?

  A: Understand AT through its algorithmic implicit bias.

- **Q**: AT beyond robustness?

  A: Improve reinforcement learning (RL) algorithm with AT.

Outline

Address the following questions, (tentatively)

- **Q**: How does AT achieve robustness?

  A: Understand AT through its algorithmic implicit bias.

- **Q**: AT beyond robustness?

  A: Improve reinforcement learning (RL) algorithm with AT.

Part I: Understand Adversarial Training

Robust generalization

Population robustness:

$$\min_\theta \mathcal{L}_{\text{adv}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{\delta\in\mathcal{B}} \ell(f(\underbrace{x+\delta}_{\text{Adversarial Example: } \widehat{x}};\theta), y).$$

Empirical robustness: AT replaces unknown data distribution $\mathcal{D}$ with empirical distribution $\widehat{\mathcal{D}} = \{(x_i, y_i)\}_{i=1}^n$.

$$\min_\theta \widehat{\mathcal{L}}_{\text{adv}}(\theta) = \mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}} \max_{\delta\in\mathcal{B}} \ell(f(\underbrace{x+\delta}_{\text{Adversarial Example: } \widehat{x}};\theta), y).$$

Robust generalization: $\left\| \widehat{\mathcal{L}}_{\text{adv}}(\cdot) - \mathcal{L}_{\text{adv}}(\cdot) \right\|_\infty \leq \epsilon.$

$$\text{Small } \widehat{\mathcal{L}}_{\text{adv}} \Rightarrow \text{Small } \mathcal{L}_{\text{adv}}.$$

Robust generalization

Population robustness:

$$\min_\theta \mathcal{L}_{\mathrm{adv}}(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \max_{\delta\in\mathcal{B}} \ell(f(\underbrace{x+\delta}_{\text{Adversarial Example: } \widehat{x}};\theta), y).$$

Empirical robustness: AT replaces unknown data distribution $\mathcal{D}$ with empirical distribution $\widehat{\mathcal{D}} = \{(x_i, y_i)\}_{i=1}^n$.

$$\min_\theta \widehat{\mathcal{L}}_{\mathrm{adv}}(\theta) = \mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}} \max_{\delta\in\mathcal{B}} \ell(f(\underbrace{x+\delta}_{\text{Adversarial Example: } \widehat{x}};\theta), y).$$

Robust generalization: $\left\|\widehat{L}_{\mathrm{adv}}(\cdot) - \mathcal{L}_{\mathrm{adv}}(\cdot)\right\|_\infty \leq \epsilon$.

Small $\widehat{\mathcal{L}}_{\mathrm{adv}} \Rightarrow$ Small $\mathcal{L}_{\mathrm{adv}}$.

Road to robust generalization

Known Results:

- Complexity-based approach: robust shattering dimension (Montasser et al. 2019), adversarial Rademacher complexity (Ying et al. 2018), function transformation (Khim et al. 2018).

  Solely depends on complexity of the model class. Not applicable to DNNs (# Params $\gg 10^8$) .

- How about a margin based approach?

  Parameter-free (e.g., kernel SVM).

- Our approach: algorithmic implicit bias.
- Result in short: AT finds large margin solution.

Road to robust generalization

Known Results:

- Complexity-based approach: robust shattering dimension (Montasser et al. 2019), adversarial Rademacher complexity (Ying et al. 2018), function transformation (Khim et al. 2018).

  Solely depends on complexity of the model class. Not applicable to DNNs (# Params $\gg 10^8$) .

- How about a margin based approach?

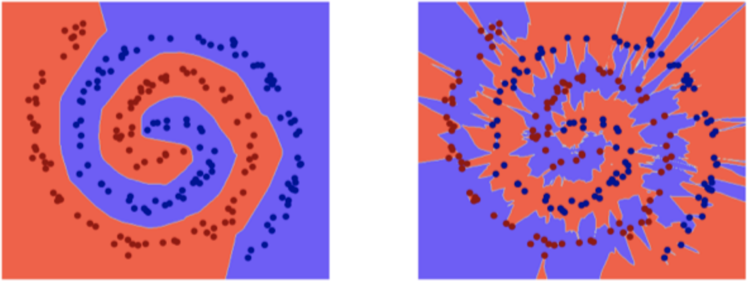
  Parameter-free (e.g., kernel SVM).

- Our approach: algorithmic implicit bias.
- Result in short: AT finds large margin solution.

Implicit Bias

Definition: In solving an under-specified problem, the optimization algorithm biases toward solutions with certain properties.

Implicit bias in training DNNs:



(a). (b).

Implicit Bias of Algorithms: network (a) is learnt by SGD (Smooth Boundary). Both networks overfits training data. Only network (a) generalizes well.

Implicit Bias

Provable examples:

- **shallow models**:

  • over-determined linear system + GD → minimum ℓ_2 norm solution (well-known).

  • logistic regression (linearly separable data) + GD → ℓ_2 SVM in direction (Soudry et al. 2018).

  • extension to SGD and Mirror Descent (Gunasekar et al. 2018).

- **deep models**:

  • Linear model + gradient flow → weight-matrix alignment + low-rankness (Ji et al. 2018).

  • Linear convolutional network (Gunasekar et al. 2018).

Implicit Bias

Provable examples:

- **shallow models**:

  • over-determined linear system + GD $\rightarrow$ minimum ℓ_2 norm solution (well-known).

  • logistic regression (linearly separable data) + GD $\rightarrow$ ℓ_2 SVM in direction (Soudry et al. 2018).

  • extension to SGD and Mirror Descent (Gunasekar et al. 2018).

- **deep models**:

  • Linear model + gradient flow $\rightarrow$ weight-matrix alignment + low-rankness (Ji et al. 2018).

  • Linear convolutional network (Gunasekar et al. 2018).

Training linear model with GD

Problem setup: linearly separable data $(x_i, y_i)_{i=1}^n$, tight exponential tail loss (e.g., logistic/exp loss),

Learning linear classifier:

$$\min_\theta \widehat{\mathcal{L}}_{\text{clean}}(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(y_i x_i^\top \theta).$$

Observations:

- no finite minimizer! $\|\theta^t\| \to \infty$ if $\widehat{\mathcal{L}}_{\text{clean}}(\theta^t) \to 0$.
- only the direction matter.

Training linear model with GD

Problem setup: linearly separable data $(x_i, y_i)_{i=1}^n$, tight exponential tail loss (e.g., logistic/exp loss),

Learning linear classifier:

$$\min_\theta \widehat{\mathcal{L}}_{\text{clean}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i x_i^\top \theta).$$

Observations:

- no finite minimizer! $\left\|\theta^t\right\| \to \infty$ if $\widehat{\mathcal{L}}_{\text{clean}}(\theta^t) \to 0$.
- only the direction matter.

Training linear model with GD

Theorem (Soudry et al. 2018)

GD converges in direction to the ℓ_2 SVM, that is

$$\left\|\theta^t\right\|_2 = \Omega(\log t), \quad 1 - \left\langle \theta^t / \left\|\theta^t\right\|_2, \theta_2 \right\rangle = \mathcal{O}(1/\log t).$$

- θ_2 (and generally θ_q) = the optimal $\ell_2(\ell_q)$ margin SVM,

$$\theta_q = \arg\max_{\|\theta\|_p = 1} \min_{i=1,\dots,n} y_i x_i^\top \theta, \quad 1/p + 1/q = 1, p, q \in [1, \infty].$$

- γ_q = optimal ℓ_q margin (max of RHS).
- Slow rate ($\exp(1/\epsilon)$ time for ϵ accuracy in direction), tight (unfortunately).

Training linear model with AT

Problem setup: linearly separable data $(x_i, y_i)_{i=1}^n$, tight exponential tail loss (e.g., logistic/exp loss), ℓ_q perturbation: $\mathcal{B} = \{\delta : \|\delta\|_q \leq c\}$.

Learning robust linear classifier:

$$\min_\theta \widehat{\mathcal{L}}_{\mathrm{adv}}(\theta) = \frac{1}{n}\sum_{i=1}^n \max_{\|\delta\|_q \leq c} \ell(y_i(x_i + \delta)^\top \theta).$$

Observations:

- When $\epsilon = 0$, standard training, converges in direction to ℓ_2 SVM (Soudry et al. 2018).
- Inner max can be solved exactly.
- When $c < \gamma_q$, no finite solution, $\|\theta_t\| \to \infty$!

Training linear model with AT

Problem setup: linearly separable data $(x_i, y_i)_{i=1}^n$, tight exponential tail loss (e.g., logistic/exp loss), ℓ_q perturbation: $\mathcal{B} = \{\delta : \|\delta\|_q \leq c\}$.

Learning robust linear classifier:

$$\min_\theta \widehat{\mathcal{L}}_{\text{adv}}(\theta) = \frac{1}{n} \sum_{i=1}^n \max_{\|\delta\|_q \leq c} \ell(y_i (x_i + \delta)^\top \theta).$$

Observations:

- When $\epsilon = 0$, standard training, converges in direction to ℓ_2 SVM (Soudry et al. 2018).
- Inner max can be solved exactly.
- When $c < \gamma_q$, no finite solution, $\|\theta_t\| \to \infty$!

GDAT – Gradient Descent based Adversarial Training

GDAT on Separable Data with ℓ_q Perturbation

Input: Data points $\{(x_i, y_i)\}_{i=1}^n$, perturbation level $c < \gamma_q$ and step sizes $\{\eta^t\}_{t=0}^{T-1}$.

Init: Set $\theta^0 = 0$.

For $t = 0 \ldots T-1$:

For $i = 1 \ldots n$, $\widehat{\delta}_i = \arg\max_{\|\delta_i\|_q \le c} \ell(y_i(x_i + \delta_i)^\top \theta^t)$.

Set $\widetilde{x}_i = x_i + \widehat{\delta}_i$, for $i = 1 \ldots n$.

Update $\theta^{t+1} = \theta^t - (\eta^t/n) \cdot \sum_{i=1}^n \nabla \ell(y_i \widetilde{x}_i \theta^t)$.

Questions: Does GDAT possess implicit bias, and whether it relates to robustness?

A Robust SVM

Consider the following large margin classifier that adapts to adversary geometry $\mathcal{B} = \{\delta : \|\delta\|_q \leq c\}$

$$\theta_{q,c} = \arg\max_{\|\theta\|_2 = 1} \min_{i=1,\ldots,n} \min_{\|\delta_i\|_q \leq c} y_i(x_i + \delta_i)^\top \theta.$$

Robustness: $\theta_{q,c}$ is in the same direction to the solution of

$$\min_{\theta \in \mathbb{R}^d} \|\theta\|_2 \quad \text{s.t. } y_i \widetilde{x}_i^\top \theta \geq 1 \text{ for all } \|\widetilde{x}_i - x_i\|_q \leq c, \forall i = 1 \ldots n.$$

A Robust SVM

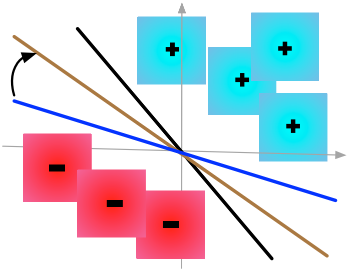
Consider the following large margin classifier that adapts to adversary geometry $\mathcal{B} = \{\delta : \|\delta\|_q \leq c\}$

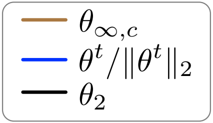
$$\theta_{q,c} = \arg\max_{\|\theta\|_2 = 1} \min_{i=1,\ldots,n} \min_{\|\delta_i\|_q \leq c} y_i (x_i + \delta_i)^\top \theta.$$

Robustness: $\theta_{q,c}$ is in the same direction to the solution of

$$\min_{\theta \in \mathbb{R}^d} \|\theta\|_2 \quad \text{s.t. } y_i \widetilde{x}_i^\top \theta \geq 1 \text{ for all } \|\widetilde{x}_i - x_i\|_q \leq c, \forall i = 1 \ldots n.$$

A Robust SVM





Minimum mix-norm solution: $\theta_{q,c}$ is in the same direction to the solution of (here $1/p + 1/q = 1$)

$$\min_{\theta \in \mathbb{R}^d} \|\theta\|_2 + \eta(c) \|\theta\|_p \quad \text{s.t. } y_i x_i^\top \theta \geq 1, \forall i = 1 \ldots n.$$

GDAT Adapts to Adversary Examples

Theorem (Li et al. 2019)

Let perturbation level $c < \gamma_q$, *Then*

$$1 - \left\langle \theta^t / \left\|\theta^t\right\|_2, \theta_{q,c} \right\rangle = \mathcal{O}(\log n / \log t).$$

Remarks:

- Guaranteed robustness against ℓ_q perturbation bounded by c.
- Adaptive implicit bias. Converges to the most ℓ_2 robust linear classifier with ℓ_q margin at least c.
  Special case: $q = 2$, converge to ℓ_2 SVM.
- Complementary of well known results on non-separable data: SVM + AT ⇒ Robust SVM (Xu el al. 2009).

GDAT Adapts to Adversary Examples

Theorem (Li et al. 2019)

Let perturbation level $c < \gamma_q$, Then

$$1 - \left\langle \theta^t / \left\|\theta^t\right\|_2, \theta_{q,c} \right\rangle = \mathcal{O}(\log n / \log t).$$

Remarks:

- Guaranteed robustness against ℓ_q perturbation bounded by c.
- Adaptive implicit bias. Converges to the most ℓ_2 robust linear classifier with ℓ_q margin at least c.
  Special case: $q = 2$, converge to ℓ_2 SVM.
- Complementary of well known results on non-separable data:
  SVM + AT $\Rightarrow$ Robust SVM (Xu el al. 2009).

GDAT Accelerates Convergence ($q = 2$)

Theorem (Li et al. 2019)

Let c and number of iterations T satisfy $\gamma_2 - c = \mathcal{O}\left(\frac{\log^2 T}{T}\right)^{1/2}$, We have $\theta_{2,c} = \theta_2$, and

$$1 - \left\langle \theta^T / \left\|\theta^T\right\|_2, \theta_2 \right\rangle = \mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right).$$

Exponential Acceleration by GDAT!

Corollary: Convergence on clean loss by GDAT is almost exponentially faster than GD.

- GDAT: $\widehat{\mathcal{L}}_{\text{clean}}(\theta_T) = \mathcal{O}\left(\exp(-\sqrt{T}/\log T)\right)$
- GD: $\widehat{\mathcal{L}}_{\text{clean}}(\theta_T) = \mathcal{O}\left(1/T\right)$

GDAT Accelerates Convergence ($q = 2$)

Theorem (Li et al. 2019)

Let c and number of iterations T satisfy $\gamma_2 - c = \mathcal{O}\left(\frac{\log^2 T}{T}\right)^{1/2}$,
We have $\theta_{2,c} = \theta_2$, and

$$1 - \left\langle \theta^T / \left\|\theta^T\right\|_2, \theta_2 \right\rangle = \mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right).$$

Exponential Acceleration by GDAT!

Corollary: Convergence on clean loss by GDAT is almost exponentially faster than GD.

- GDAT: $\widehat{\mathcal{L}}_{\text{clean}}(\theta_T) = \mathcal{O}\left(\exp(-\sqrt{T}/\log T)\right)$
- GD: $\widehat{\mathcal{L}}_{\text{clean}}(\theta_T) = \mathcal{O}\left(1/T\right)$

GDAT Accelerates Convergence ($q = 2$)

Theorem (Li et al. 2019)

Let c and number of iterations T satisfy $\gamma_2 - c = \mathcal{O}\left(\frac{\log^2 T}{T}\right)^{1/2}$,
We have $\theta_{2,c} = \theta_2$, and

$$1 - \left\langle \theta^T / \left\| \theta^T \right\|_2, \theta_2 \right\rangle = \mathcal{O}\left(\frac{\log T}{\sqrt{T}}\right).$$

Exponential Acceleration by GDAT!

Corollary: Convergence on clean loss by GDAT is almost exponentially faster than GD.

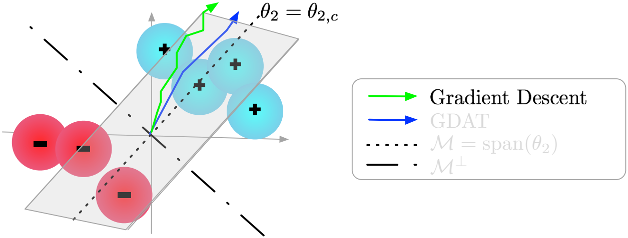
- GDAT: $\widehat{\mathcal{L}}_{\text{clean}}(\theta_T) = \mathcal{O}\left(\exp(-\sqrt{T}/\log T)\right)$
- GD: $\widehat{\mathcal{L}}_{\text{clean}}(\theta_T) = \mathcal{O}\left(1/T\right)$

Intuition: We have $\theta_{2,c} = \theta_2$: implicit bias of GD coincides with explicit bias of AT!

Key Technical Ingredients:

- Projection of θ^t onto the orthogonal space $\mathcal{M}^\perp = \{\theta : \langle \theta, \theta_2 \rangle = 0\}$ is bounded for all $t \geq 0$.
- For projection of θ^t onto the space $\mathcal{M} = \mathrm{span}(\theta_2)$, its increment satisfies **Generalized Perceptron Lemma**:

$$\langle \theta^{t+1} - \theta^t, \theta_2 \rangle \geq \eta \widehat{\mathcal{L}}_{\mathrm{adv}}(\theta^t)(\gamma_2 - c).$$

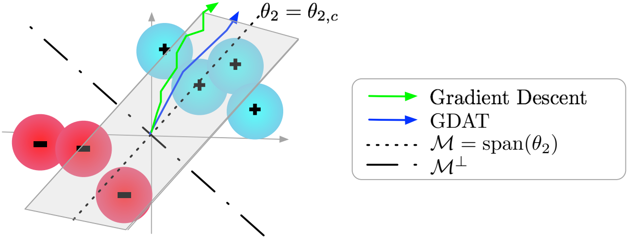


Intuition: We have $\theta_{2,c} = \theta_2$: implicit bias of GD coincides with explicit bias of AT!

Key Technical Ingredients:

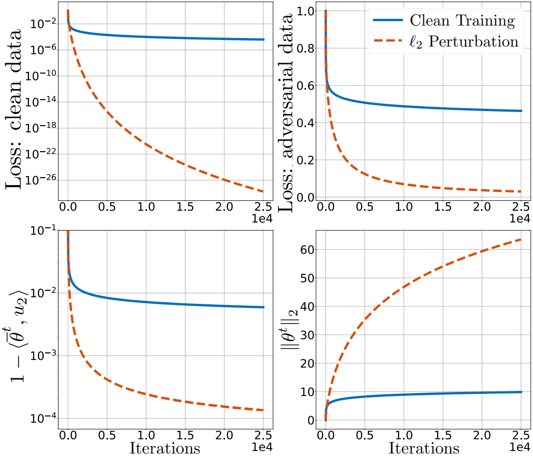
- Projection of θ^t onto the orthogonal space $\mathcal{M}^\perp = \{\theta : \langle \theta, \theta_2 \rangle = 0\}$ is bounded for all $t \geq 0$.
- For projection of θ^t onto the space $\mathcal{M} = \mathrm{span}(\theta_2)$, its increment satisfies **Generalized Perceptron Lemma**:

$$\left\langle \theta^{t+1} - \theta^t, \theta_2 \right\rangle \geq \eta \widehat{\mathcal{L}}_{\mathrm{adv}}(\theta^t)(\gamma_2 - c).$$



Empirical Study

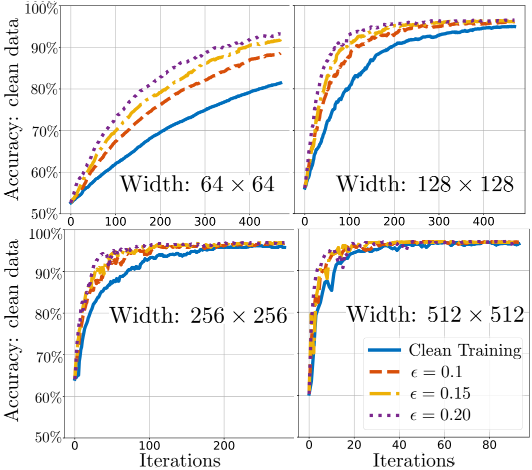
Linear Classifiers: We generate data with $\gamma_2 = 1$. We set $c = 0.95$. $\eta = 0.1$ for GDAT and $\eta = 1$ for standard training.



Clean Training v.s. GDAT (ℓ_2 perturbation)

Empirical Study

Neural Networks: We use MNIST dataset. The width of hidden layer varies in $\{64 \times 64, 128 \times 128, 256 \times 256, 512 \times 512\}$. We use ℓ_∞ perturbation with perturbation level $\epsilon \in \{0.1, 0.15, 0.20\}$.



Additional experimental results: (Xie et al. 2019) show acceleration effect of AT with practical deep networks.

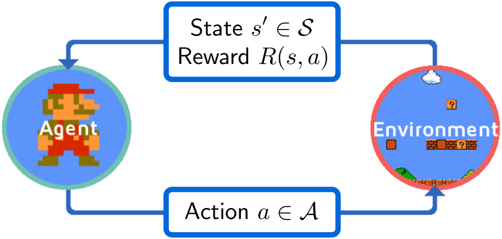
Summary

- GDAT adapts the classifier to adversary geometry.
- GDAT with ℓ_2 perturbation provides exponential speed-up on clean loss.

Part II: AT for better Reinforcement Learning

Reinforcement Learning

Markov Decision Process: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$



- $P(s'|s,a)$: Transition Kernel;
- $\gamma \in (0,1)$: Discount Factor;
- $\mathbb{E}\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$: Expected Total Discounted Reward.

Solving an RL.

Goal: maximize expected (discounted) reward

$$\max_{\pi} V(\pi) = \mathbb{E}_{s_0,a_0,\dots}\left[\sum_{t\geq 0}\gamma^t r(s_t,a_t)\right],$$

with $s_0 \sim p_0, a_t \sim \pi(s_t), s_{t+1} \sim \mathbb{P}(s_{t+1}|s_t,a_t)$.

Policy gradient algorithms:

- estimate the gradient of the expected reward through trajectory samples: $\widehat{g}_t$, update: $\pi_{t+1} = \pi_t - \eta\widehat{g}_t$.
- suffers from large variance, aggressive update, unstable training.

Improved variants [Actor-critic]: TRPO (Schulman et al. 2015), DDPG (Lillicrap et al. 2015).

Solving an RL.

Goal: maximize expected (discounted) reward

$$\max_{\pi} V(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \right],$$

with $s_0 \sim p_0, a_t \sim \pi(s_t), s_{t+1} \sim \mathbb{P}(s_{t+1} | s_t, a_t)$.

Policy gradient algorithms:

- estimate the gradient of the expected reward through trajectory samples: $\widehat{g}_t$, update: $\pi_{t+1} = \pi_t - \eta \widehat{g}_t$.
- suffers from large variance, aggressive update, unstable training.

Improved variants [Actor-critic]: TRPO (Schulman et al. 2015), DDPG (Lillicrap et al. 2015).

RL with smooth environments

Motivation: smooth reward function, smooth transition ⇒ exists smooth policy.

Promoting smoothness in policy: adversarially defined regularization (Shen*, Li*, Jiang, Wang, Zhao, 2020)

$$\mathcal{R}_{\mathrm{s}}^{\pi}(\theta) = \mathop{\mathbb{E}}_{s \sim \rho^{\pi_\theta}} \max_{\widetilde{s} \in \mathbb{B}_d(s,\epsilon)} \mathcal{D}(\pi_\theta(s), \pi_\theta(\widetilde{s})).$$

$\mathcal{D}(\cdot,\cdot)$ appropriate metric, $\mathbb{B}_d(s,\epsilon) = \{s', \|s - s'\|_\infty \leq \epsilon\}$, ρ^{π_θ} the stationary state distribution induced by π_θ.

- the inner max measures local Lipschitz smoothness of policy under metric $\mathcal{D}$.
- can solve the inner max with projected gradient ascent.
- take expectation w.r.t. state-visitation distribution: smoothness along trajectory.

RL with smooth environments

Motivation: smooth reward function, smooth transition ⇒ exists smooth policy.

Promoting smoothness in policy: adversarially defined regularization (Shen*, Li*, Jiang, Wang, Zhao, 2020)

$$\mathcal{R}_{\mathrm{s}}^{\pi}(\theta) = \mathop{\mathbb{E}}_{s\sim\rho^{\pi_\theta}} \max_{\widetilde{s}\in\mathbb{B}_d(s,\epsilon)} \mathcal{D}(\pi_\theta(s), \pi_\theta(\widetilde{s})).$$

$\mathcal{D}(\cdot,\cdot)$ appropriate metric, $\mathbb{B}_d(s,\epsilon) = \{s', \|s - s'\|_\infty \le \epsilon\}$, ρ^{π_θ} the stationary state distribution induced by π_θ.

- the inner max measures local Lipschitz smoothness of policy under metric $\mathcal{D}$.
- can solve the inner max with projected gradient ascent.
- take expectation w.r.t. state-visitation distribution: smoothness along trajectory.

RL with smooth environments

Motivation: smooth reward function, smooth transition $\Rightarrow$ exists smooth policy.

Promoting smoothness in policy: adversarially defined regularization (Shen*, Li*, Jiang, Wang, Zhao, 2020)

$$\mathcal{R}_{\mathrm{s}}^{\pi}(\theta) = \mathop{\mathbb{E}}_{s \sim \rho^{\pi_\theta}} \max_{\widetilde{s} \in \mathbb{B}_d(s,\epsilon)} \mathcal{D}(\pi_\theta(s), \pi_\theta(\widetilde{s})).$$

$\mathcal{D}(\cdot,\cdot)$ appropriate metric, $\mathbb{B}_d(s,\epsilon) = \{s', \|s - s'\|_\infty \leq \epsilon\}$, ρ^{π_θ} the stationary state distribution induced by π_θ.

- the inner max measures local Lipschitz smoothness of policy under metric $\mathcal{D}$.
- can solve the inner max with projected gradient ascent.
- take expectation w.r.t. state-visitation distribution: smoothness along trajectory.

Application: TRPO (stochastic policy)

Policy update:

$$\theta_{k+1} = \arg\min_{\theta} -\mathbb{E}_{\substack{s\sim\rho^{\pi_{\theta_k}}, \\ a\sim\pi_{\theta_k}}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a)\right]$$
$$+ \lambda_{\mathrm{s}} \mathbb{E}_{s\sim\rho^{\pi_{\theta_k}}} \max_{\widetilde{s}\in\mathbb{B}_d(s,\epsilon)} \mathcal{D}_{\mathrm{J}}(\pi_\theta(\cdot|s) \mid\mid \pi_\theta(\cdot|\widetilde{s})),$$
$$\text{s.t.} \quad \mathbb{E}_{s\sim\rho^{\pi_{\theta_k}}} \left[\mathcal{D}_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \| \pi_\theta(\cdot|s_)\right] \le \delta.$$

- Jeffrey's divergence: $\mathcal{D}_{\mathrm{J}}(P||Q) = \frac{1}{2}\mathcal{D}_{\mathrm{KL}}(P||Q) + \frac{1}{2}\mathcal{D}_{\mathrm{KL}}(Q||P)$.
- first part – approximate linearization of objective function (proximal update).
- second part – smoothness regularization.

Application: TRPO (stochastic policy)

Policy update:

$$\theta_{k+1} = \arg\min_{\theta} -\mathbb{E}_{\substack{s\sim\rho^{\pi_{\theta_k}},\\ a\sim\pi_{\theta_k}}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a)\right]$$

$$+ \lambda_{\mathrm{s}} \mathbb{E}_{s\sim\rho^{\pi_{\theta_k}}} \max_{\widetilde{s}\in\mathbb{B}_d(s,\epsilon)} \mathcal{D}_{\mathrm{J}}(\pi_\theta(\cdot|s) \mid\mid \pi_\theta(\cdot|\widetilde{s})),$$

$$\text{s.t.} \quad \mathbb{E}_{s\sim\rho^{\pi_{\theta_k}}} \left[\mathcal{D}_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \| \pi_\theta(\cdot|s_{)}\right] \leq \delta.$$

- Jeffrey's divergence:
  $\mathcal{D}_{\mathrm{J}}(P||Q) = \frac{1}{2}\mathcal{D}_{\mathrm{KL}}(P||Q) + \frac{1}{2}\mathcal{D}_{\mathrm{KL}}(Q||P)$.
- first part – approximate linearization of objective function (proximal update).
- second part – smoothness regularization.

Application: DDPG (deterministic policy)

Actor-critic framework:

- actor: policy network $\mu_\theta(s)$.
- critic: network to approximate Q-function $Q_\phi(s, a)$ (expected future reward given initial state-action pair (s, a)).

Idea: use critic to help update the policy (actor).

DDPG with smoothness inducing regularization. We can induce smoothness in either the actor or the critic.

Application: DDPG (deterministic policy)

Smooth critic:

$$\phi_{t+1} = \arg\min_{\phi} \sum_{i \in B} \left(y_t^i - Q_\phi(s_t^i, a_t^i) \right)^2 + \lambda_{\mathrm{s}} \sum_{i \in B} \max_{\widetilde{s}_t^i \sim \mathbb{B}_d(s_t^i, \epsilon)} (Q_\phi(s_t^i, a_t^i) - Q_\phi(\widetilde{s}_t^i, a_t^i))^2,$$

$$\text{with} \quad y_t^i = r_t^i + \gamma Q_{\phi_t'}(s_{t+1}^i, \mu_{\theta_t'}(s_{t+1}^i)), \forall i \in B,$$

B denotes the mini-batch sampled from the replay buffer.

- first part – approximate Bellman error for Q-function evaluation.
- second part – smoothness inducing regularization.
- use target network ϕ_t' to generate y_t^i, improve stability.
- update the target network with exponential averaging: $\phi_{t+1}' = \tau \phi_{t+1} + (1 - \tau)\phi_t'$.

Application: DDPG (deterministic policy)

Smooth critic:

$$\phi_{t+1} = \arg\min_{\phi} \sum_{i \in B} \left(y_t^i - Q_\phi(s_t^i, a_t^i) \right)^2 + \lambda_{\mathrm{s}} \sum_{i \in B} \max_{\widetilde{s}_t^i \sim \mathbb{B}_d(s_t^i, \epsilon)} (Q_\phi(s_t^i, a_t^i) - Q_\phi(\widetilde{s}_t^i, a_t^i))^2,$$

$$\text{with} \quad y_t^i = r_t^i + \gamma Q_{\phi_t'}(s_{t+1}^i, \mu_{\theta_t'}(s_{t+1}^i)), \forall i \in B,$$

B denotes the mini-batch sampled from the replay buffer.

- first part – approximate Bellman error for Q-function evaluation.
- second part – smoothness inducing regularization.
- use target network ϕ_t' to generate y_t^i, improve stability.
- update the target network with exponential averaging: $\phi_{t+1}' = \tau \phi_{t+1} + (1 - \tau) \phi_t'$.

Application: DDPG (deterministic policy)

Smooth actor:

$$\mu_{\theta_{t+1}} = \mu_{\theta_t} - \eta \mathop{\mathbb{E}}_{s\sim\rho^\beta} \Big[- \nabla_a Q_\phi(s,a)\big|_{a=\mu_{\theta_t}(s)} \nabla_\theta \mu_{\theta_t}(s) + \lambda_{\mathrm{s}} \nabla_\theta \left\| \mu_{\theta_t}(s) - \mu_{\theta_t}(\widetilde{s}) \right\|_2^2 \Big],$$

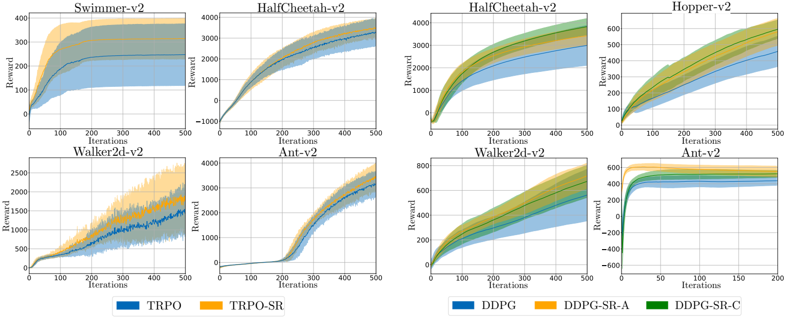
with $\widetilde{s} = \mathop{\arg\max}_{\widetilde{s}\sim\mathbb{B}_d(s,\epsilon)} \left\| \mu_{\theta_t}(s) - \mu_{\theta_t}(\widetilde{s}) \right\|_2^2$ for $s \sim \rho^\beta$.

- first part – policy gradient.
- second part – gradient of the smoothness inducing regularizer.

Experiments

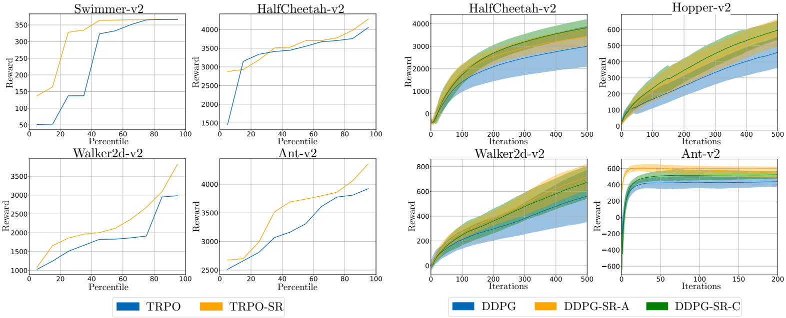
Environments: OpenAI gym (Swimmer, HalfCheetah, Walker, Ant, Hopper)

Faster learning:



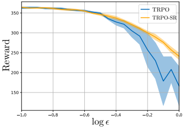
Faster learning compared to strong implementation of baseline.

Significant improvement: Repeated 10 runs with random initializations, plot the quantiles of the final cumulative rewards.

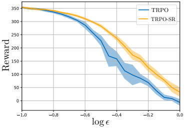


Improvement for both worse-case and base-case scenario.

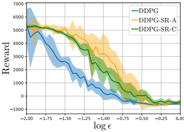
Evaluation of robustness



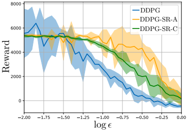
(a) Swimmer - Random Disturbed Rollout



(b) Swimmer - Adversarially Disturbed Rollout



(a) HalfCheetah - Adversarially Disturbed Rollout



(b) HalfCheetah - Random Disturbed Rollout

Robust against adversarial error & measurement error.

Summary

- Study implicit bias of AT when training linear model.
  – AT adapts the model to adversary geometry.
  – AT with ℓ_2 perturbation speeds up training.
- Improve reinforcement learning algorithm with AT.
  – adversarially defined smoothness regularizer.
  – improve sample complexity and robustness against state perturbations.

References

[1] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples."
[2] Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks."
[3] Rice, Leslie, Eric Wong, and J. Zico Kolter. "Overfitting in adversarially robust deep learning."
[4] Gao, Ruiqi, et al. "Convergence of Adversarial Training in Overparametrized Neural Networks."
[5] Tjeng, Vincent, Kai Xiao, and Russ Tedrake. "Evaluating robustness of neural networks with mixed integer programming."
[6] Cohen, Jeremy M., Elan Rosenfeld, and J. Zico Kolter. "Certified adversarial robustness via randomized smoothing."
[7] Montasser, Omar, Steve Hanneke, and Nathan Srebro. "VC classes are adversarially robustly learnable, but only improperly."
[8] Yin, Dong, Kannan Ramchandran, and Peter Bartlett. "Rademacher complexity for adversarially robust generalization."
[9] Khim, Justin, and Po-Ling Loh. "Adversarial risk bounds via function transformation."

References

[10] Soudry, Daniel, et al. "The implicit bias of gradient descent on separable data."
[11] Gunasekar, Suriya, et al. "Characterizing implicit bias in terms of optimization geometry."
[12] Ji, Ziwei, and Matus Telgarsky. "Gradient descent aligns the layers of deep linear networks."
[13] Gunasekar, Suriya, et al. "Implicit bias of gradient descent on linear convolutional networks."
[14] Xu, Huan, Constantine Caramanis, and Shie Mannor. "Robustness and regularization of support vector machines."
[15] Xie, Cihang, et al. "Adversarial Examples Improve Image Recognition."
[16] Schulman, John, et al. "Trust region policy optimization."
[17] Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning."

Thank You!