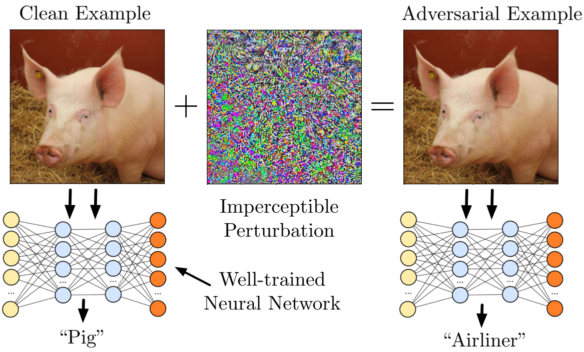
Implicit Bias of Gradient Descent based Adversarial Training on Separable Data

Yan Li*, Ethan X.Fang†, Huan Xu‡, Tuo Zhao*

*Georgia Tech, †Pennsylvania State University, ‡Georgia Tech, Alibaba Inc

Adversarial Examples



All current deep neural network (DNN) models are subject to adversarial examples.

Training Robust Models

Adversarial training directly minimize the worst-case loss for a given perturbation set Δ:

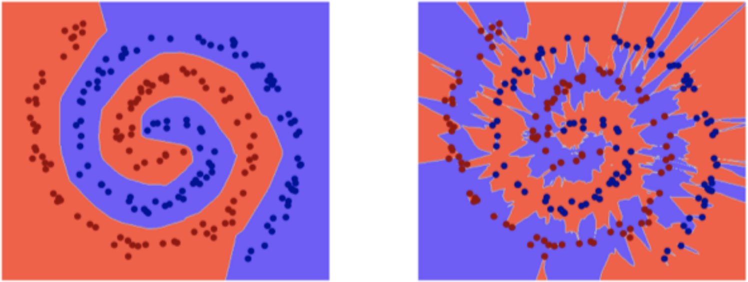
$$\theta_{\text{robust}} = \arg\min_{\theta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \max_{\delta_i \in \Delta} \ell(x_i + \delta_i, y_i, \theta).$$

Question: How does adversarial training promote robustness?

We propose to study from a computational perspective – implicit bias of the optimization algorithm.

Implicit Bias

Neural network can easily overfit training data. Training algorithm biases toward a certain kind of solutions.



(a). (b).

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

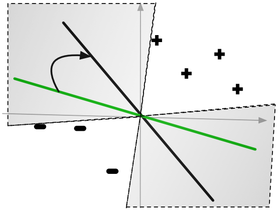
Training a Linear Classifier

Directly analyzing DNNs is beyond current technical limit.
▷ A simplified yet non-trivial example, training a linear classifier on linearly separable data $\{(x_i, y_i)\}_{i=1}^n$. We aim to solve

$$\min_{\theta \in \mathbb{R}^d} \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i x_i^\top \theta), \ell \text{ exponential/logistic loss.}$$

• Only the **direction** of the linear classifier is important.
• There is **no finite minimizer** of $\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i x_i^\top \theta)$. But there exists infinite amount of solutions at infinity.

Implicit Bias of Gradient Descent



$\theta^t / \|\theta^t\|_2$: Standard Training
θ_2: Maximum ℓ_2 Margin Classifier

Classifiers within the shaded cone overfit the clean examples.

Gradient descent converges in direction to the ℓ_2 norm max margin classifier (Soudry et al 2017; Ji and Telgarsky 2018):

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

where θ_q (here $q = 2$) and the optimal value γ_q is defined by:

$$\theta_q = \underset{\|\theta\|_p = 1}{\arg\max} \min_{i=1,\ldots,n} y_i x_i^\top \theta, \quad \text{with } 1/p + 1/q = 1.$$

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 $t = 0 \dots T-1$:

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

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

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

Questions: Can we characterize the implicit bias of GDAT on separable data? How is it related to adversarial robustness?

GDAT Adapts to Adversary Examples

Consider the following large margin classifier:

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

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

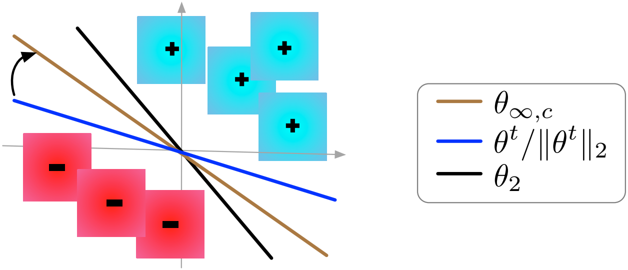
$$\min_{\theta\in\mathbb{R}^d} \|\theta\|_2 \quad \text{s.t. } y_i \widetilde{x}_i^\top \theta \geq 1 \text{ for all } \|\widetilde{x}_i - x_i\|_q \leq c, \forall i = 1 \ldots n.$$

GDAT Adapts to Adversary Examples

Theorem (Informal)

Let perturbation level $c < \gamma_q$, Then

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



GDAT Accelerates Convergence ($q = 2$)

Theorem (Informal)

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

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

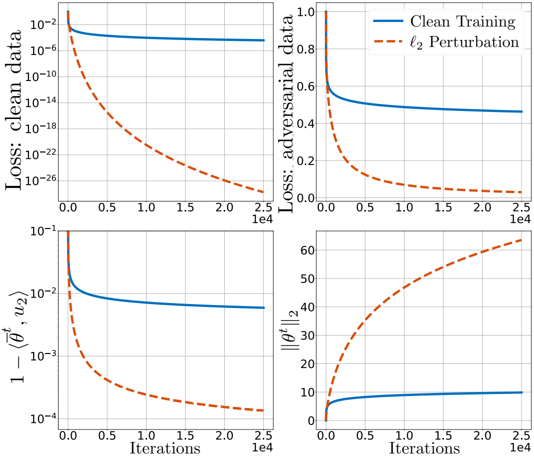
Exponential Acceleration by GDAT

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

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

Empirical Study

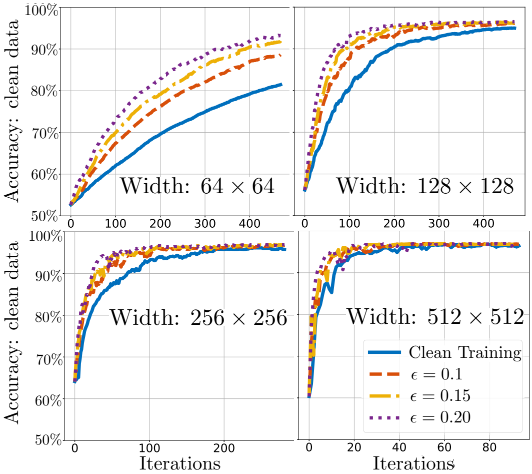
Linear Classifiers: We generate data with $\gamma_2 = 1$. We set $c = 0.95$. $\eta = 0.1$ for GDAT and $\eta = 1$ for standard training.



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 $\epsilon \in \{0.1, 0.15, 0.20\}$.



Thank you!