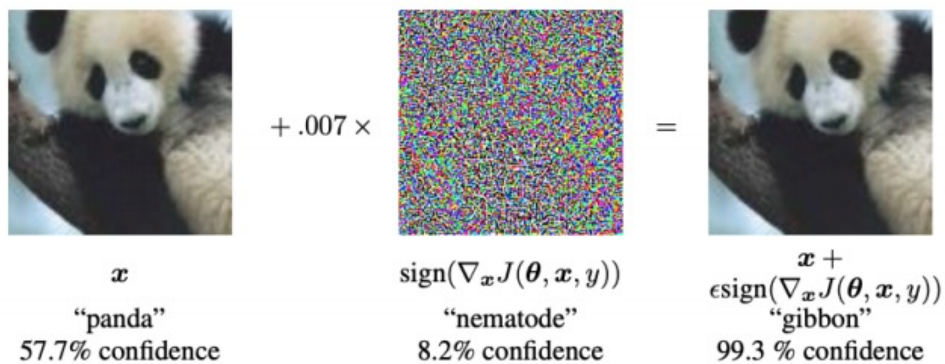
To be Robust or to be Fair: Towards Fairness in Adversarial Training

Han Xu*, Xiaorui Liu*, Yaxin Li, Anil Jain, Jiliang Tang
Michigan State University

July, 2021

Adversarial Attacks and Adversarial Training

- Adversarial Attacks

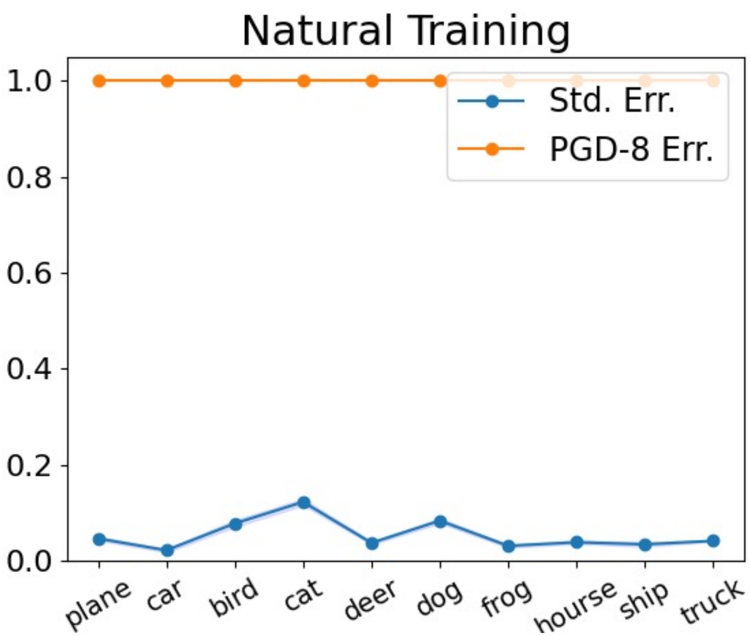


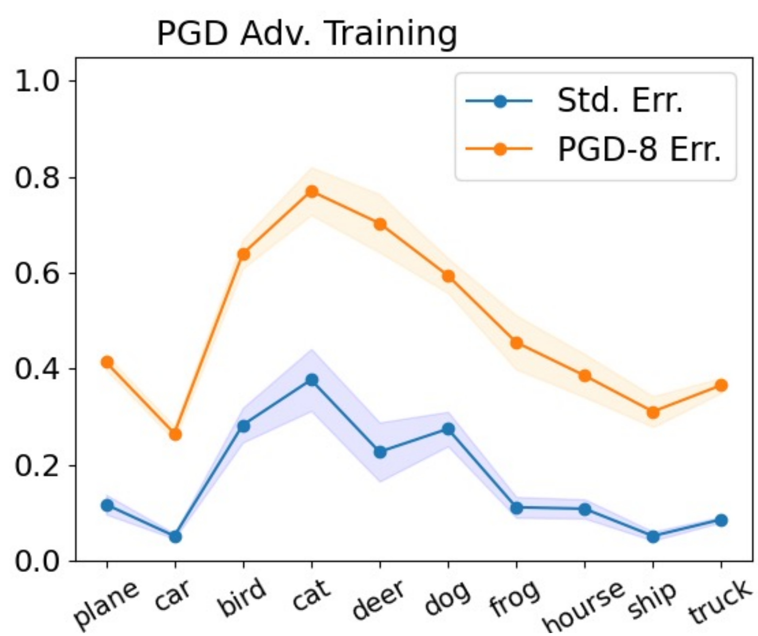
- Adversarial Training (Madry, et al., 2018, Zhang, et al., 2019)

$$\min_f \mathbb{E}_x \left[\max_{||\delta|| \leq \epsilon} \mathcal{L}(f(x+\delta), y) \right]$$

New finding: Unfairness of Adv Training

Adv. trained models are more likely to have big inter-class performance disparity





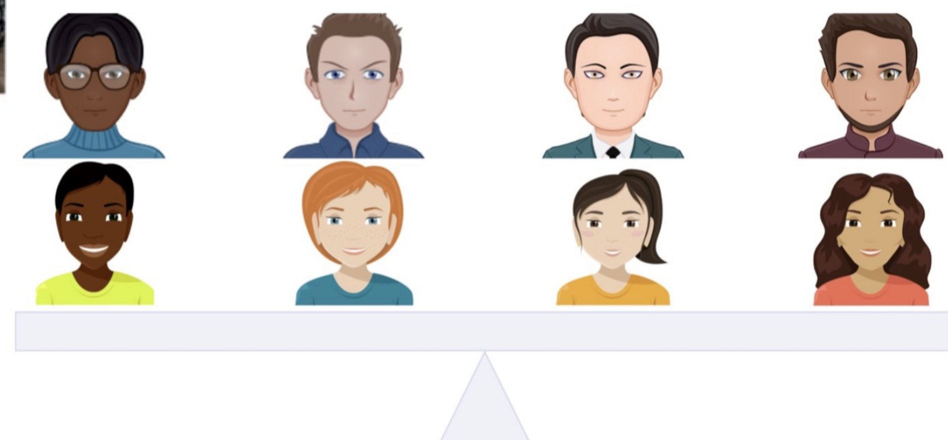
Each class's error rate is presented, CIFAR10, ResNet18 Model

Potential Consequences

Robust models are not equally safe.

(Image Credit: Eykholt et al., 2017)

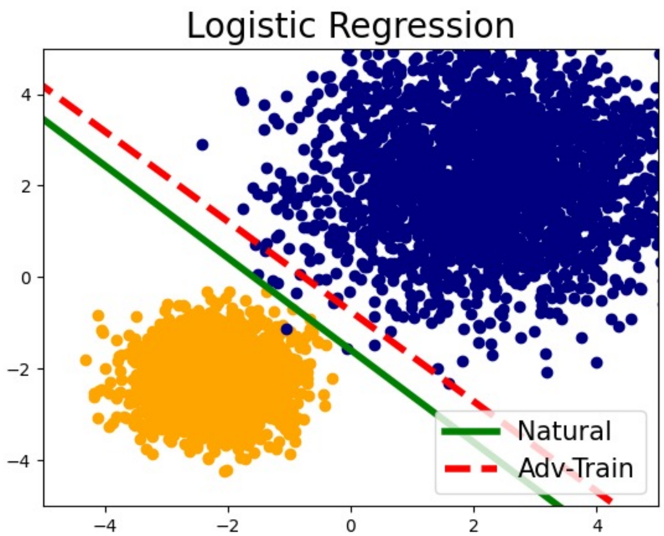
Unfair ML might have social ethnic issues



(Image Credit: Robinson et al., 2020)

Potential Reasons

- 1. There are indeed some classes whose data are harder to classify.
- 2. The decision boundary of an (natural) optimal classifier is closer to the easy class.
- 3. Adversarial training have more "neutral" decision boundaries.
- 4. In adversarial training, easy class become easier, hard class become harder.



Two Classes:

A "hard" class; $\mathcal{N}(\theta, \sigma_1^2)$.

An "easy" class; $\mathcal{N}(-\theta, \sigma_2^2)$.

$$\sigma_1^2 > \sigma_2^2$$

How to Deal With the Unfairness? The FRL method.

Algorithm 1 The Fair Robust Learning (FRL) Algorithm

Input: Fairness constraints specified by $\tau_1 > 0$ and $\tau_2 > 0$, test time attacking radius ϵ and hyper-param update rate α_1, α_2

Output: A fairly robust neural network f

Initialize network with a pre-trained robust model

Set $\phi^i_{\text{nat}} = 0$, $\phi^i_{\text{bndy}} = 0$ and $\phi = (\phi_{\text{nat}}, \phi_{\text{bndy}})$,

repeat

$\quad \mathcal{R}_{\text{nat}}(f), \mathcal{R}_{\text{nat}}(f, i) = \text{EVAL}(f)$

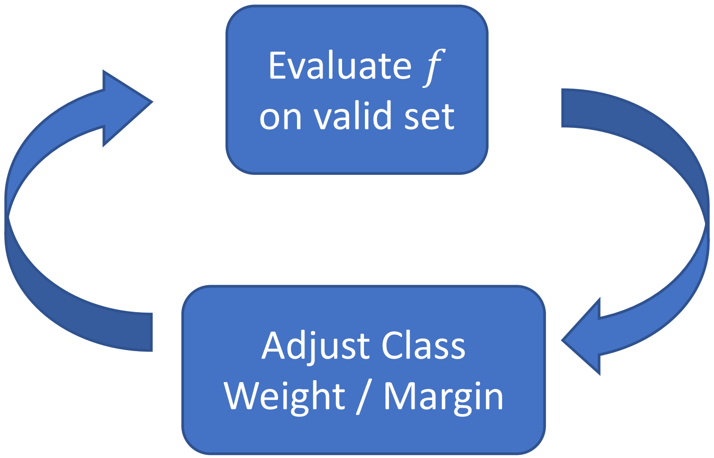
$\quad \mathcal{R}_{\text{bndy}}(f), \mathcal{R}_{\text{bndy}}(f, i) = \text{EVAL}(f, \epsilon)$

$\quad \phi^i_{\text{nat}} = \phi^i_{\text{nat}} + \alpha_1 \cdot (\mathcal{R}_{\text{nat}}(f, i) - \mathcal{R}_{\text{nat}}(f) - \tau_1)$

$\quad \phi^i_{\text{bndy}} = \phi^i_{\text{bndy}} + \alpha_2 \cdot (\mathcal{R}_{\text{bndy}}(f, i) - \mathcal{R}_{\text{bndy}}(f) - \tau_2)$

$\quad f \leftarrow \text{TRAIN}(f, \phi, \epsilon)$

until Model f satisfies all constraints



Thanks. Q&A