Friendly Adversarial Training:
Attacks Which Do Not Kill Training Make Adversarial Learning Stronger

2021.5.18

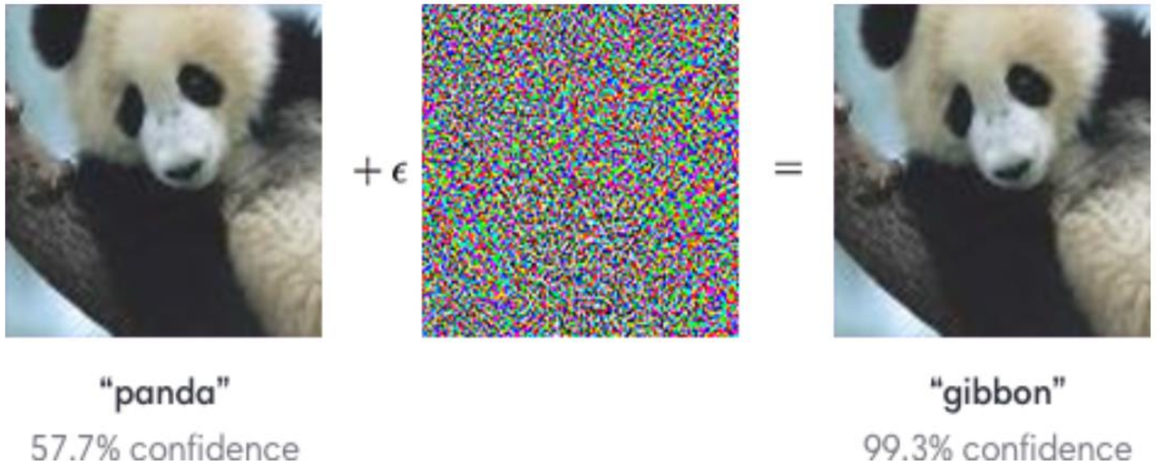
Jingfeng Zhang   Xilie Xu   Bo Han   Gang Niu   Lizhen Cui
Masashi Sugiyama   Mohan Kankanhalli

ICML 2020

Background

What is adversarial data?

Adversarial data = neural data + synthetic noise.

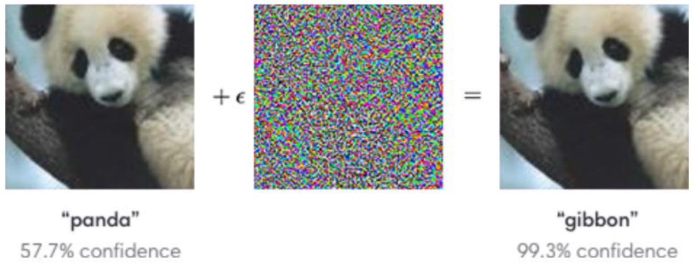


The danger of adversarial data is enormous

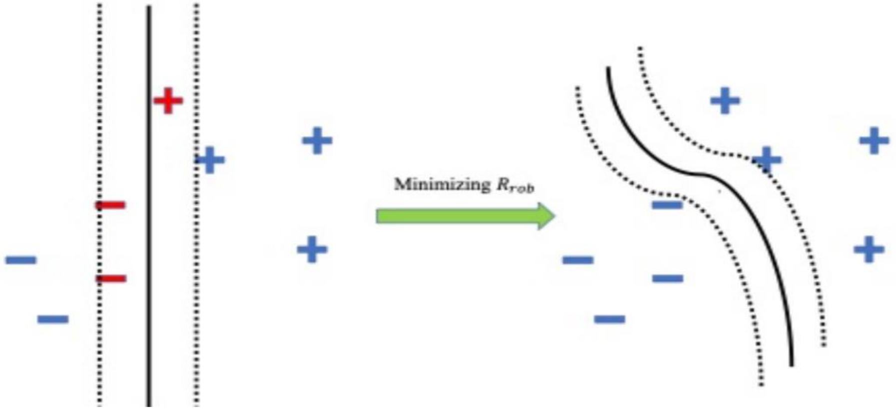
Background

What is adversarial learning ?

- **Adversarial training** so far is the most effective method for obtaining the adversarial robustness of the trained classifier..



- **Purpose 1:** correctly classify the data.
- **Purpose 2:** make the decision boundary thick so that no data is encouraged to fall inside the decision boundary.



Background

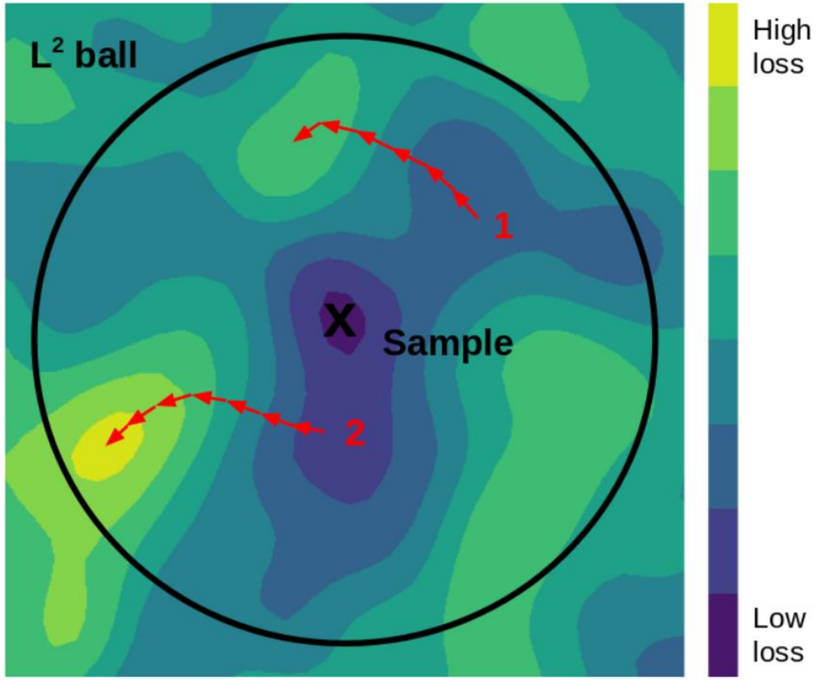
Conventional formulation of adversarial training

Minimax formulation:

$$\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f\,(\tilde{x}_i)\,, y_i), \text{ where } \tilde{x}_i = argmax_{x \in B(x_i)}\, \ell(f\,(\tilde{x}), y_i)$$

Outer minimization   Inner maximization

Projected gradient descent(PGD) adversarial training
Approximately realizes this minimax formulation.
PGD formulates the problem of **finding the most adversarial data** as a constrained optimization problem.



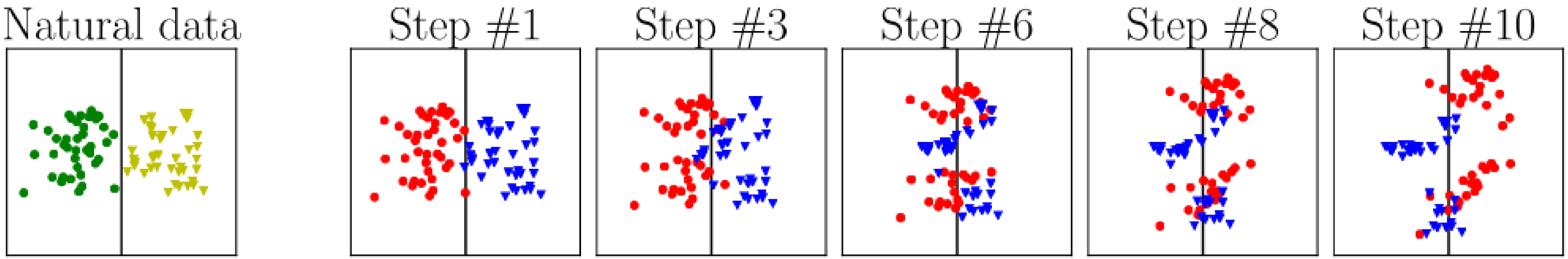
Projected gradient descent with restart. 2nd run finds a high loss adversarial example within the L^2 ball. Sample is in a region of low loss.

Motivation

The minimax formulation is conservative and pessimistic.

Many existing studies found the minimax-based adversarial training causes the **server degradation** of the natural generalization. **Why?**

The adversarial data generated by PGD



The cross-over mixture problem

Is the minimax formulation suitable to the adversarial training?

Idea

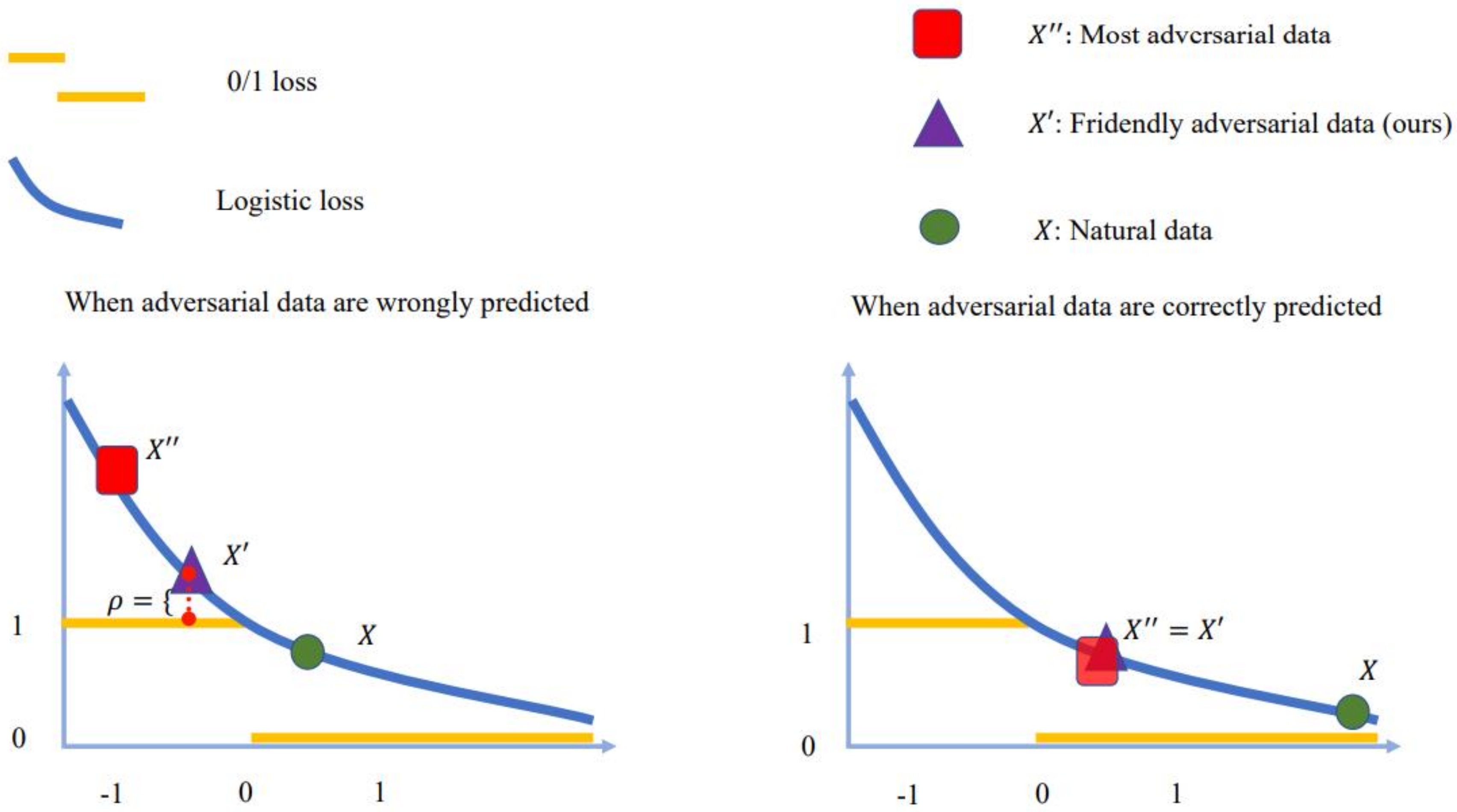
Min-min formulation for the adversarial training.

The outer minimization keeps the same. Instead of generating adversarial data via inner maximization, we generate x˜i as follows:

$$\tilde{x}_i = \arg min_{\tilde{x} \in B(x_i)} \ell(f(\tilde{x}), y_i) \ \text{ s.t. } \ell(f(\tilde{x}), y_i) - \min_{y \in \mathcal{Y}} \ell(f(\tilde{x}), y_i) \geq \rho$$

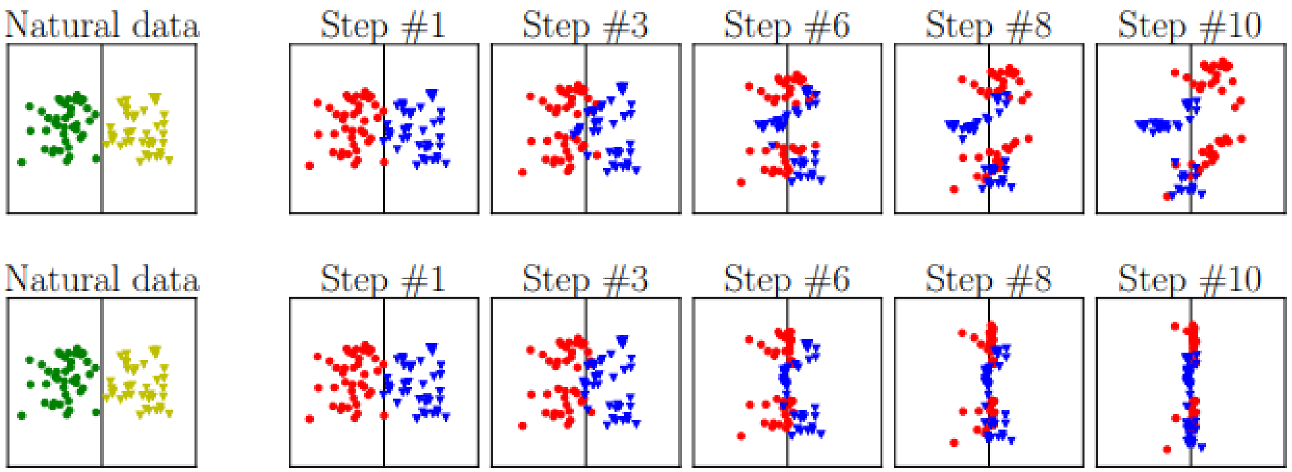
Idea

Adversarial data generated by min-min and minimax formulation



Idea

Realization of our min-min formulation-friendly adversarial training.



Conventional PGD generating most adversarial data

Early stopped PGD generating friendly adversarial data

Algorithm

Algorithm 1 PGD-K-τ

Input: data $x \in \mathcal{X}$, label $y \in \mathcal{Y}$, model f, loss functior ℓ, maximum PGD step K, step τ, perturbation bound ϵ step size α

Output: $\tilde{x}$

$\tilde{x} \leftarrow x$

while $K > 0$ **do**

  if $\arg\max_i f(\tilde{x}) \neq y$ and $\tau = 0$ **then**

    break

  else if $\arg\max_i f(\tilde{x}) \neq y$ **then**

    $\tau \leftarrow \tau - 1$

  end if

  $\tilde{x} \leftarrow \Pi_{\mathcal{B}[x,\epsilon]}(\alpha\,\mathrm{sign}(\nabla_{\tilde{x}}\ell(f(\tilde{x}), y)) + \tilde{x})$

  $K \leftarrow K - 1$

end while

Algorithm 2 Friendly Adversarial Training (FAT)

Input: network f_θ, training dataset $S = \{(x_i, y_i)\}_{i=1}^{n}$, learning rate η, number of epochs T, batch size m, number of batches M

Output: adversarially robust network f_θ

for epoch $= 1, \ldots, T$ **do**

  for mini-batch $= 1, \ldots, M$ **do**

    Sample a mini-batch $\{(x_i, y_i)\}_{i=1}^{m}$ from S

    for $i = 1, \ldots, m$ (in parallel) **do**

      Obtain adversarial data $\tilde{x}_i$of x_i by Algorithm 1

    end for

    $\theta \leftarrow \theta - \eta \frac{1}{m} \sum_{i-1}^{m} \nabla_\theta \ell(f_\theta(\tilde{x}_i), y_i)$

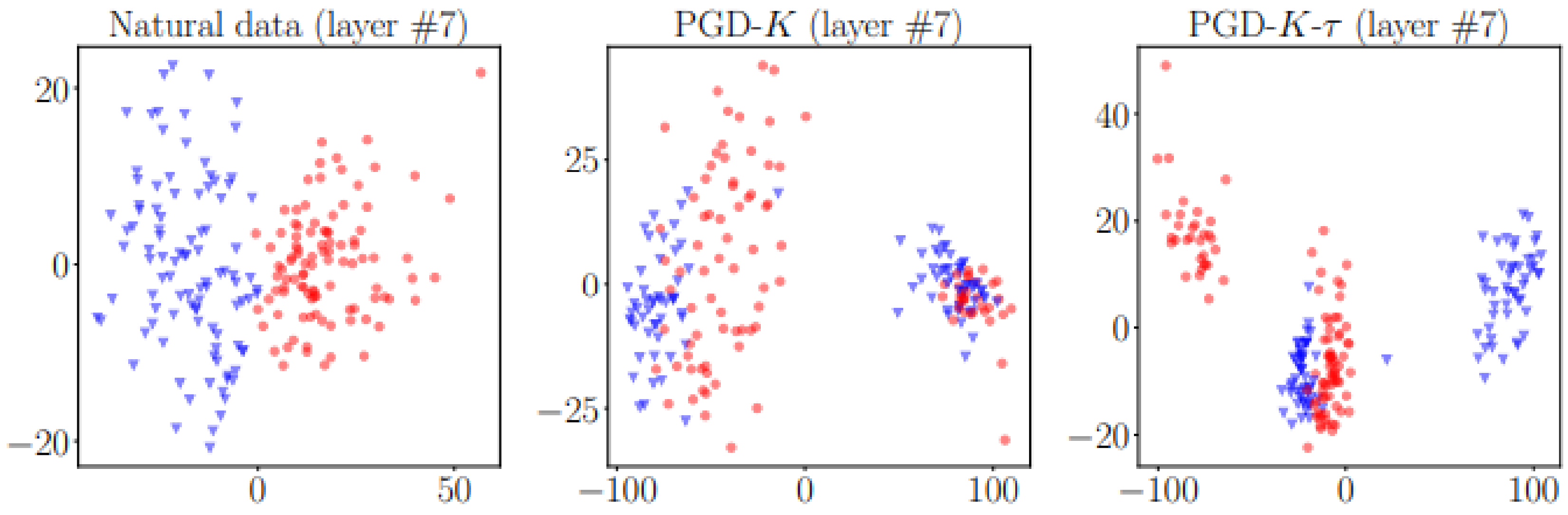
  end for

end for

Benefits

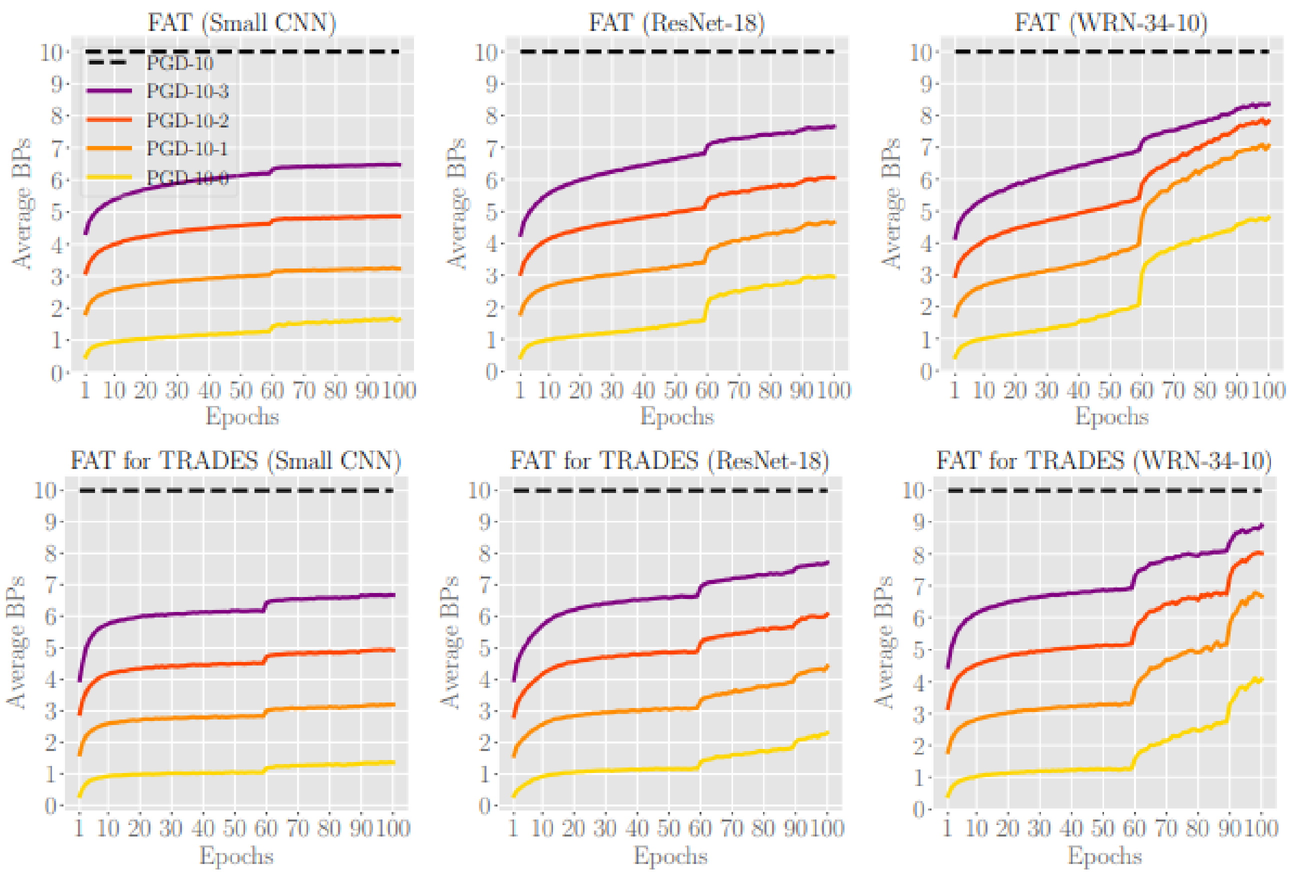
Benefit(a): Alleviate cross-over mixture problem

- In the classification of the CIFAR-10 dataset, the cross-over mixture problem may not appear in the input space, but in the middle layers.



Benefits

Benefit(b): FAT is computationally efficient



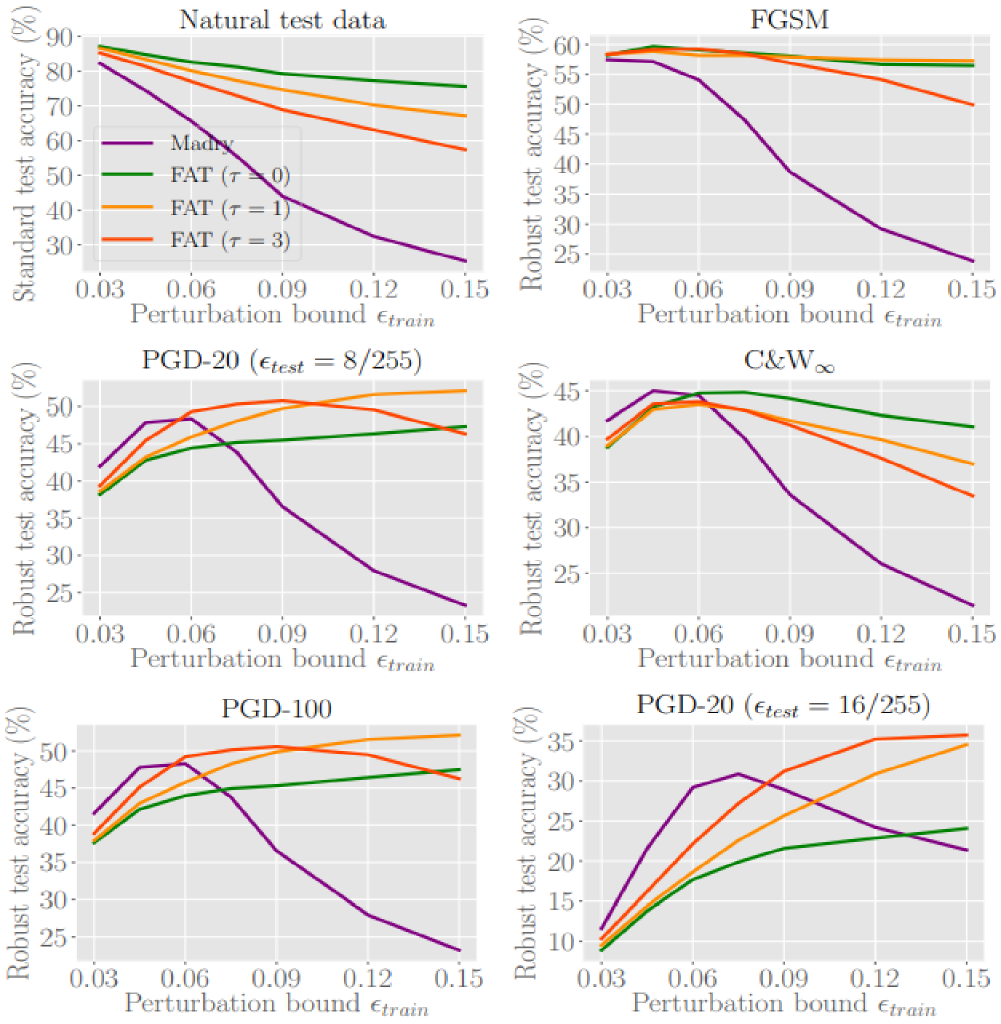
We report the average backward propagations (BPs) per epoch over training process.

Dashed line is existing adversarial training based on conventional PGD.

Solid lines are friendly adversarial trainings based on early stopped PGD.

Benefits

Benefit(c): FAT can enable larger defense parameter ϵ train



For CIFAR-10 dataset, we adversarially train deep neural network with ϵ train:[0.03,0.15], and evaluate each robust model with 6 evaluation metrics.

The purple line represents existing adversarial training.

Lines of other colors represent friendly adversarial training with different configurations.

Benefits

Benefit(d):Benchmarking on Wide ResNet

Friendly Adversarial Training

Table 1. Evaluations (test accuracy) of deep models (WRN-32-10) on CIFAR-10 dataset

Defense	Natural	FGSM	PGD-20	C&W$_\infty$	PGD-100
Madry	87.30	56.10	45.80	46.80	-
CAT	77.43	57.17	46.06	42.28	-
DAT	85.03	63.53	48.70	47.27	-
FAT ($\epsilon_{train} = 8/255$)	**89.34** $\pm$ 0.221	65.52 $\pm$ 0.355	46.13 $\pm$ 0.409	46.82 $\pm$ 0.517	45.31 $\pm$ 0.531
FAT ($\epsilon_{train} = 16/255$)	87.00 $\pm$ 0.203	**65.94** $\pm$ 0.244	**49.86** $\pm$ 0.328	**48.65** $\pm$ 0.176	**49.56** $\pm$ 0.255

Results of Madry, CAT and DAT are reported in (Wang et al., 2019). FAT has the same evaluations.

Table 2. Evaluations (test accuracy) of deep models (WRN-34-10) on CIFAR-10 dataset

Defense	Natural	FGSM	PGD-20	C&W$_\infty$	PGD-100
TRADES ($\beta = 1.0$)	88.64	56.38	49.14	-	-
FAT for TRADES ($\epsilon_{train} = 8/255$)	**89.94** $\pm$ 0.303	**61.00** $\pm$ 0.418	**49.70** $\pm$ 0.653	49.35 $\pm$ 0.363	48.35 $\pm$ 0.240
TRADES ($\beta = 6.0$)	84.92	61.06	56.61	**54.47**	55.47
FAT for TRADES ($\epsilon_{train} = 8/255$)	**86.60** $\pm$ 0.548	**61.97** $\pm$ 0.570	55.98 $\pm$ 0.209	54.29 $\pm$ 0.173	55.34 $\pm$ 0.291
FAT for TRADES ($\epsilon_{train} = 16/255$)	84.39 $\pm$ 0.030	61.73 $\pm$ 0.131	**57.12** $\pm$ 0.233	54.36 $\pm$ 0.177	**56.07** $\pm$ 0.155

Results of TRADES ($\beta = 1.0$ and 6.0) are reported in (Zhang et al., 2019b). FAT for TRADES has the same evaluations.

FAT can improve standard accuracy while maintain the superior adversarial robustness.

Thanks