DataScience Lab: Training robust neural networks

Nan An, Hangyue Zhao, Lucas Hennecon

PGD L∞ attack model

PGD attack+ Adversarial training

Randomized Smoothing $$g(x) = \arg\max_c \mathbb{P}(f(x+\delta) = c), \quad \delta \sim \mathcal{N}(0, \sigma^2 I)$$

- smooth out the decision boundary

Spectral Normalization

- ensures that small perturbations do not cause the model's predictions to overreact

Improved the model's robustness against attacks

Effective against L2 perturbations, limited in $L\,\infty$, like FGSM

Carlini & Wagner Attack

Loss function

$$\mathcal{L} = \|\delta\|_2^2 + c \cdot g(x + \delta)$$

g(x′): An auxiliary function that ensures x′, is misclassified.

δ: Optimized perturbation.

c: Hyperparameter, Balances perturbation size and misclassification.

$$g(x') = \max(Z(x')_t - \max_{i \neq t} Z(x')_i, -\kappa)$$

Z(x′): Model logits for x′x'x′, before softmax.

t: True label of the input.

κ\kappa: Confidence parameter; larger values mean stronger attacks

The C&W attack explicitly optimizes the loss function, allowing it to bypass defenses like defensive distillation.

Analysis Accuracy

PGD attack+ Adversarial training (PGD attack sample) with smooth and Spectral Normalize

C&W Attack + Same mechanism but 50% C&W attack samples and 50% PGD attack samples

Table 1. Comparison of Accuracy for Different Models

Model	PGD L_2 Accuracy (%)	PGD L_∞ Accuracy (%)
PGD Model	24.79	36.14
C&W Model	2.73	27.49

- Combining L∞ and L2 defenses can lead to conflict if attacks are not complementary.
- Complexity and diversity of attacks do not guarantee improved robustness.
- The key is finding the right adversarial training strategy.

Randomized Adversarial Training

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\tau\|_p \leq \epsilon} \mathcal{L}\left(\tilde{f}_\theta(x+\tau), y\right)\right]$$

	AT	RAT
Natural Accuracy	31.93	38.87
ℓ_2	31.34	38.18
ℓ_∞	23.82	24.7

Mixed Adversarial Training

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\mathop{\mathbb{E}}_{p\sim\mathcal{U}(\{2,\infty\})} \max_{\|\tau\|_p \leq \epsilon} \mathcal{L}\left(f_\theta(x+\tau), y\right) \right].$$

MAT-Rand :

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\mathop{\mathbb{E}}_{p\sim\mathcal{U}(\{2,\infty\})} \max_{\|\tau\|_p \leq \epsilon} \mathcal{L}\left(f_\theta(x+\tau), y\right) \right].$$

MAT-Max :

$$\min_{\theta} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} \left[\max_{p\in\{2,\infty\}} \max_{\|\tau\|_p \leq \epsilon} \mathcal{L}\left(f_\theta(x+\tau), y\right) \right].$$

	PDG	MAT
Natural Accuracy	38.87	40.72
ℓ_2	38.18	39.84
ℓ_∞	24.7	27.14

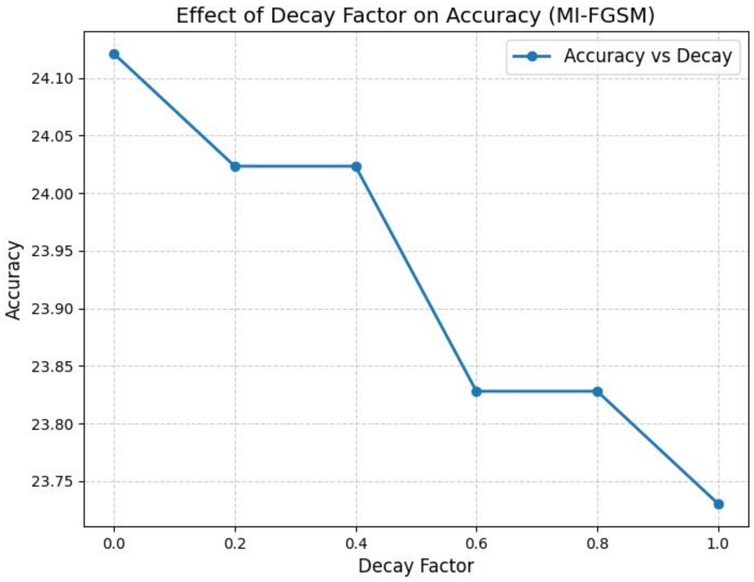
MI-FGSM

- <u>Accumulate gradient:</u>

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(x_t^*, y)}{\|\nabla_x J(x_t^*, y)\|_1}$$

- <u>Update</u>:

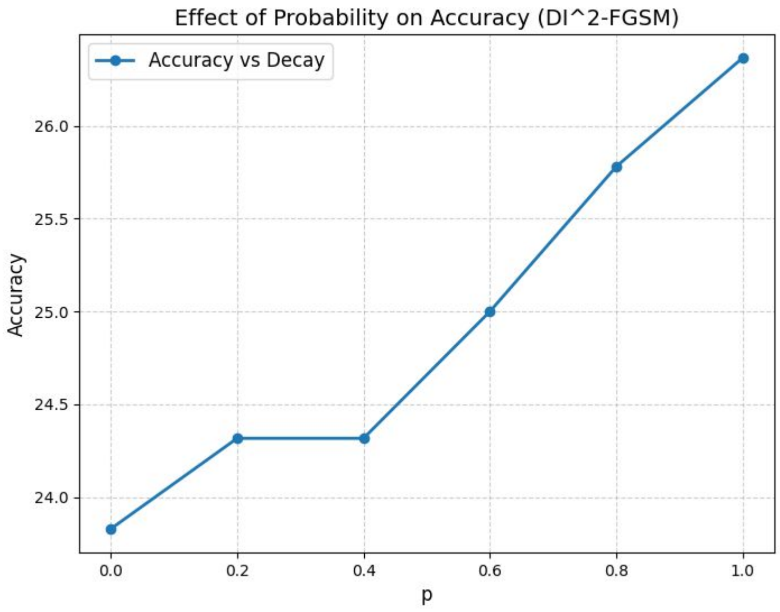
$$x_{t+1}^* = x_t^* + \alpha \cdot \text{sign}(g_{t+1})$$



M-DI^2-FGSM

- Variant of MI-FGSM
- Apply transformations (resizing, 0-padding) with probability p

-> Less efficient for whitebox attacks



Thank you for listening