Adaptive Batch Normalization Networks for Adversarial Robustness

AVSS 2024

Shao-Yuan Lo       Vishal M. Patel
Johns Hopkins University
What are Adversarial Examples

- Deep networks are **vulnerable** to adversarial examples.

\[ f_\theta(\text{Dog}) = \text{"Dog"} \]

\[ f_\theta(\text{Dog} + 0.001 \times \text{noise}) = \text{"Cat"} \]
What are Adversarial Examples

• Dataset: CIFAR-10
• Network: ResNet-50
How to Generate Adversarial Examples

• Train a model
  • \( \min \text{Loss}(f(x), y; \theta) \)
  • **Minimize** the loss function w.r.t. *model parameters \( \theta \)*

• Generate adversarial examples
  • Most common method: Gradient-based method, e.g., FGSM.
  • \( \max \text{Loss}(f(x+\delta), y; \theta) \)
  • **Maximize** the loss function w.r.t. *adversarial perturbation \( \delta \)*
Defense by Adversarial Training

• Adversarial Training (AT) is a strong defense against adversarial examples.
• **Core idea:** Train with adversarial examples.

\[
\theta^* = \arg \min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} L(x, y; \theta)
\]

\[
\theta^* = \arg \min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[ \max_{\delta \in S} L(x + \delta, y; \theta) \right]
\]

Generate adversarial examples
Train model parameters

[Madry et al. ICLR’18]
Defense by Adversarial Training

• However, AT involves a **min-max optimization**, which is extremely expensive.

Iterative adversarial example generation

$$x^{t+1} = \prod_{x+S} (x^t + \alpha \cdot \text{sign}(\nabla_x L(x, y; \theta))) \rightarrow x + \delta$$

Adversarial Training

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{(x, y) \sim \mathcal{D}} \max_{\delta \in \mathcal{S}} L(x + \delta, y; \theta)$$

Generate adversarial examples

Train model parameters

[Madry et al. ICLR’18]
How to design a defense method that gets rid of AT but is still robust against strong adversarial examples?
Inspiration from Image Harmonization

• Image harmonization: Match a foreground object to a new background scene.

• A style code is extracted by a style encoder and is passed to the adaptive instance norm layers of the harmonizer network.

[Valanarasu et al. ICLR’23]
Adaptive Batch Normalization Network

- **Pre-trained substitute model:** A public model trained on large-scale datasets (e.g., ImageNet)

- **Target model:** The model that we are training for a downstream task
Adaptive Batch Normalization Network

- **Standard BN**

  \[ z' = \gamma \left( \frac{z - \mu(z)}{\sigma(z)} \right) + \beta \]

- **Proposed adaptive BN**

  \[ z'_t = \gamma_s \left( \sigma(z_s) \left[ \frac{z_t - \mu(z_t)}{\sigma(z_t)} \right] + \mu(z_s) \right) + \beta_s \]
Adversary would perturb the target model’s BN.

The substitute model’s BN are relatively unaffected.

The substitute model is trained on large-scale datasets different from the target task dataset, making it harder for adversary to transfer the attack.

The model is trained on only clean data without using AT.
Training Time Complexity

• Let us set that each network pass (i.e., a forward pass or a backward pass) has $N$ computational complexity.

• ABNN:
  $2N + N = 3N$

• PGT-AT:
  $2N \times t_{\text{max}} + 2N = 2N (t_{\text{max}} + 1)$

• PGT-AT has
  $2N (t_{\text{max}} + 1) / 3N = 0.67 (t_{\text{max}} + 1)$
  times more training complexity than ABNN
Results

- **Dataset**: UCF-101
- **Target model**: 3D ResNeXt-101
- **Substitute model**: 3D ResNet-18 pre-trained on Kinetics-400
- **Attack**: ROA with 10% area, $t_{\max}=5$

<table>
<thead>
<tr>
<th>Method</th>
<th>Clean</th>
<th>ROA</th>
<th>Training cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Defense [14]</td>
<td>93.0</td>
<td>7.0</td>
<td>2N</td>
</tr>
<tr>
<td>OUDefend [21]</td>
<td>62.0</td>
<td>13.6</td>
<td>24N</td>
</tr>
<tr>
<td>ABNN (Ours)</td>
<td>68.3</td>
<td>24.4</td>
<td>3N</td>
</tr>
</tbody>
</table>

(a) Better clean data performance
(b) Better robustness generalization
(c) Better training efficiency
Results

Table 1. Evaluation results (%) under the PGD attack on the CIFAR-10 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Clean</th>
<th>PGD</th>
<th>Training cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Defense</td>
<td>93.4</td>
<td>0.0</td>
<td>2N</td>
</tr>
<tr>
<td>PGD-AT [23]</td>
<td>83.3</td>
<td>51.6</td>
<td>12N</td>
</tr>
<tr>
<td>ABNN (Ours)</td>
<td>87.5</td>
<td>31.5</td>
<td>3N</td>
</tr>
</tbody>
</table>

Table 2. Evaluation results (%) under the PGD attack on the UCF-101 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Clean</th>
<th>PGD</th>
<th>Training cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Defense [14]</td>
<td>93.0</td>
<td>0.0</td>
<td>2N</td>
</tr>
<tr>
<td>OUDDefend [21]</td>
<td>62.0</td>
<td>58.6</td>
<td>24N</td>
</tr>
<tr>
<td>ABNN (Ours)</td>
<td>68.3</td>
<td>43.4</td>
<td>3N</td>
</tr>
</tbody>
</table>
Conclusion

• The proposed adversarial defense ABNN is a non-AT method that gets rid of the extremely time-consuming AT.

• Compared to traditional AT-based approaches, the proposed ABNN achieves higher clean data performance, better robustness generalization, and significantly lower training time complexity.