Section 8: Robustness in Deep Learning: Adversarial Attacks and Defenses

By Taha Majlesi

July 30, 2025

Abstract

Deep neural networks, despite their remarkable success in various domains, exhibit a significant vulnerability to adversarial examples. These are inputs that have been meticulously crafted by adding small, often imperceptible, perturbations to legitimate inputs, with the intent of causing the model to misclassify them. This fragility has ignited a fervent area of research dedicated to understanding and mitigating these threats. This document provides a comprehensive and structured overview of the key adversarial attack methodologies and the corresponding defense strategies that have been developed. We will delve into the important formulas and core concepts for each, offering a clear and understandable explanation of this ongoing arms race in the field of deep learning.

1 Adversarial Attacks on Deep Neural Networks

Adversarial attacks are techniques designed to generate adversarial examples. These attacks can be broadly classified based on two main criteria: the attacker's knowledge of the target model and the method used to create the perturbations.

1.1 Attacker's Knowledge

- **White-box Attacks:** In this scenario, the attacker has complete knowledge of the model, including its architecture, parameters (weights and biases), and the gradients of the loss function with respect to the input. [1] This level of access allows for highly efficient and effective gradient-based attacks.
- **Black-box Attacks:** Here, the attacker has no direct access to the model's internal workings. [1] They can only interact with the model by providing inputs and observing the outputs (e.g., class labels or confidence scores).

1.2 Attack Techniques

1.2.1 Fast Gradient Sign Method (FGSM)

Proposed by Goodfellow et al. (2015), FGSM is one of the earliest and most straightforward white-box attacks. [2] It operates by taking a single step in the direction of the sign of the gradient of the loss function with respect to the input. This efficiently increases the model's loss, often leading to misclassification.

The perturbation δ is calculated as:

$$\delta = \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

where:

- ϵ is a small scalar that controls the magnitude of the perturbation.
- $\text{sign}(\cdot)$ is the sign function.
- $\nabla_x J(\theta, x, y)$ is the gradient of the loss function J with respect to the input x.
- θ represents the model's parameters, and y is the true label of the input x.

The adversarial example x_{adv} is then generated by:

$$x_{\text{adv}} = x + \delta$$

Effect: FGSM is computationally inexpensive and often effective at creating misclassifications. However, as a single-step method, the generated perturbations may not be optimal (i.e., the smallest possible) and can be defended against by more robust, iterative methods. [2]

1.2.2 Randomized FGSM (R+FGSM)

A variation of FGSM, Randomized FGSM (R+FGSM) introduces a small random perturbation to the input before applying the standard FGSM step. [3] This technique, introduced by Tramèr et al. (2018), helps to circumvent "gradient masking," a phenomenon where defenses obscure or reduce the usefulness of gradients. [3]

The process is as follows:

1. Start with a random initial perturbation: $x' = x + \alpha \cdot \text{sign}(\mathcal{N}(0, I))$, where α is a small magnitude and $\mathcal{N}(0, I)$ is a standard normal distribution.
2. Apply a scaled FGSM step with the remaining perturbation budget: $x_{\text{adv}} = x' + (\epsilon - \alpha) \cdot \text{sign}(\nabla_x J(\theta, x', y))$.

In essence, R+FGSM takes a random step followed by a gradient-based step, which can help the attack escape local optima and overcome certain defenses. [3]

1.2.3 Basic Iterative Method (BIM) and Projected Gradient Descent (PGD)

Iterative attacks generally achieve higher success rates than single-step methods by applying multiple, smaller gradient steps.

Basic Iterative Method (BIM), also known as I-FGSM, was introduced by Kurakin et al. It iteratively applies the FGSM update with a small step size α and clips the result after each step to ensure the perturbation remains within the allowed ϵ-ball. [2] The update rule for each iteration i is:

$$x_{\text{adv}}^{(i+1)} = \text{Clip}_{x,\epsilon}\{x_{\text{adv}}^{(i)} + \alpha \cdot \text{sign}(\nabla_x J(\theta, x_{\text{adv}}^{(i)}, y))\}$$

where $\text{Clip}_{x,\epsilon}\{\cdot\}$ ensures that the perturbed example remains within an ϵ-neighborhood of the original input x. [2]

Projected Gradient Descent (PGD) is a more powerful iterative attack. It is often considered the strongest first-order adversary. [4] PGD typically starts with a random perturbation within the ϵ-ball around the input x:

$$x_{\text{adv}}^{(0)} = x + \text{Uniform}(-\epsilon, \epsilon)$$

Then, it iteratively applies the same update rule as BIM, projecting the result back onto the ϵ-ball after each step. [4] The random start helps PGD to avoid local minima and find more robust adversarial examples. [5]

Effect: Iterative methods like BIM and PGD generally produce smaller and more effective perturbations than single-step attacks. PGD, in particular, is a widely used benchmark for evaluating the robustness of defenses. [5]

1.2.4 Momentum Iterative FGSM (MI-FGSM)

Proposed by Dong et al. (2018), MI-FGSM enhances iterative attacks by incorporating a momentum term into the gradient updates. [4] This helps to stabilize the updates and improve the transferability of adversarial examples to other models, a crucial aspect of black-box attacks. [5]

At each iteration t, the momentum term g_t is updated:

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

where μ is a decay factor. The adversarial example is then updated using the sign of the momentum term:

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

By accumulating gradients over iterations, MI-FGSM can overcome local maxima and create more transferable adversarial examples. [5]

1.2.5 DeepFool

DeepFool, developed by Moosavi-Dezfooli et al. (2016), takes a geometric approach to finding the minimal perturbation required to change a model's classification. [2] It iteratively linearizes the classifier's decision boundaries and moves the input towards the closest boundary until a misclassification occurs. [2]

DeepFool is an untargeted attack that typically produces very small L_2 perturbations. While computationally more expensive than FGSM, it often finds smaller perturbations. [2]

1.2.6 Carlini & Wagner (C&W) Attack

The C&W attack, proposed by Carlini & Wagner (2017), is a powerful, optimization-based attack. It formulates the search for an adversarial example as an optimization problem that balances the size of the perturbation with the misclassification objective. For an L_2-based attack, the objective is:

$$\min_{\delta} \|\delta\|_2^2 + c \cdot f(x + \delta)$$

where $f(x+\delta)$ is a loss function that is low when $x+\delta$ is misclassified, and c is a constant that controls the trade-off between the two terms. [4] The C&W attack is highly effective and can bypass many defenses.

1.2.7 Elastic-Net Attack (EAD)

The Elastic-Net Attack to DNNs (EAD), introduced by Chen et al. (2018), extends the C&W attack by incorporating an L_1 norm penalty. [4] The optimization problem becomes:

$$\min_{\delta} \|\delta\|_2^2 + \beta\|\delta\|_1 + c \cdot f(x + \delta)$$

The inclusion of the L_1 term encourages sparse perturbations, where only a few pixels are changed significantly. EAD has been shown to produce highly transferable adversarial examples. [4]

1.2.8 Black-Box Attack Strategies

In black-box settings, where gradient information is unavailable, attackers employ different strategies:

- **Transfer Attacks:** Adversarial examples are crafted on a surrogate model and then transferred to the target model. [5]
- **Score-Based and Decision-Based Attacks:** These attacks rely on querying the model and using the output scores or decisions to guide the search for an adversarial example.
- **Evolutionary/Optimization Approaches:** Techniques like genetic algorithms can be used to evolve adversarial examples over many queries.
- **Approximate Gradient with Finite Differences:** The gradient can be estimated by making small perturbations to the input and observing the change in the output.

2 Defenses Against Adversarial Attacks

In response to the threat of adversarial attacks, a wide range of defense mechanisms have been proposed. These can be broadly categorized as proactive defenses that aim to build robust models and reactive defenses that detect or mitigate attacks at test time.

2.1 Adversarial Training (Robust Optimization)

Adversarial training is a proactive defense that involves training the model on adversarial examples. The goal is to solve a min-max optimization problem:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\delta\|\leq\epsilon} J(\theta, x + \delta, y) \right]$$

This means we aim to train the model to minimize the worst-case loss under any perturbation of size up to ϵ. [4]

2.1.1 Types of Adversarial Training

- **FGSM Adversarial Training:** An early form that uses FGSM to generate adversarial examples for training. While initially promising, it was found to be insufficient against stronger iterative attacks. [4]
- **PGD Adversarial Training:** Proposed by Madry et al. (2018), this is considered the gold standard for empirical robustness. It uses the more powerful PGD attack to generate adversarial training examples. [4]
- **Ensemble Adversarial Training:** This technique, introduced by Tramèr et al. (2018), augments the training data with adversarial examples transferred from other pre-trained models. This helps to improve robustness against black-box attacks.
- **Generative Adversarial Training:** This approach uses a generator network to produce adversarial perturbations, which are then used to train the classifier.

Trade-offs: Adversarial training, especially with PGD, is one of the most effective empirical defenses. However, it often leads to a decrease in accuracy on clean data and significantly increases the computational cost of training. [4]

2.2 Defensive Distillation

Introduced by Papernot et al. (2016), defensive distillation is a technique that trains a model using "soft" labels produced by a previously trained "teacher" network. The teacher network is trained with a high temperature in its softmax function, which smooths the output probability distribution. The student network is then trained on these soft labels.

The effect is a smoother decision surface with smaller gradients, making it harder for gradient-based attacks to succeed. [4] However, later research showed that this defense can be broken by more sophisticated attacks like the C&W attack.

2.3 Gradient Masking / Obfuscation (and Why It Fails)

Gradient masking refers to defenses that intentionally or unintentionally hide or reduce the usefulness of the model's gradients. This can be achieved through:

- **Shattered gradients:** Using non-differentiable operations in the model.
- **Stochastic gradients:** Introducing randomness into the model's prediction process.
- **Exploding/vanishing gradients:** Constructing the model in a way that leads to numerically unstable gradients.

A seminal paper by Athalye et al. (2018), "Obfuscated Gradients Give a False Sense of Security," demonstrated that these types of defenses can often be circumvented by adaptive attacks that are specifically designed to bypass the masking mechanism. [4]

2.4 Adversarial Example Detection

This line of defense focuses on identifying and rejecting adversarial inputs before they are classified. Strategies include:

- **Auxiliary Detector Models:** Training a separate model to distinguish between normal and adversarial inputs based on the main model's internal activations. [2]
- **Statistical Anomalies:** Detecting adversarial examples by looking for statistical irregularities in the model's outputs or feature representations.
- **Consistency Checks:** Applying transformations to the input and checking for significant changes in the model's output.

While appealing, detection methods can often be evaded by adaptive attacks that are aware of the detection mechanism.

2.5 Input Transformations and Denoising

These defenses aim to purify or reconstruct the input to remove adversarial perturbations before classification.

- **Defense-GAN:** Uses a Generative Adversarial Network (GAN) to project the input onto the manifold of clean images.
- **Defense-VAE:** Employs a Variational Autoencoder (VAE) for the same purpose.
- **Simple Denoising/Filtering:** Applying standard image processing techniques like Gaussian blur or JPEG compression.

These methods can be effective to some extent, but adaptive attacks can often be designed to bypass the transformation or denoising step.

2.6 Ensemble and Randomized Strategies

- **Ensemble of Models:** Using multiple diverse classifiers and aggregating their predictions.
- **Randomized Model Rotation:** Training a set of adversarially disjoint models and randomly selecting one for each input. [4]

These strategies increase the difficulty for the attacker, especially in black-box scenarios.

2.7 Adversarial Logit Pairing (ALP)

Proposed by Kannan et al. (2018), ALP is a regularization technique that encourages the model to produce similar logits for clean and adversarial examples. The pairing loss is:

$$L_{\text{pair}} = \|z(x) - z(\tilde{x})\|_2^2$$

where $z(x)$ and $z(\tilde{x})$ are the logits for the clean and adversarial inputs, respectively. While initially showing promise, later analysis revealed that its effectiveness was partly due to gradient masking.

2.8 Certified Defenses and Robustness Guarantees

In contrast to empirical defenses, certified defenses provide a mathematical guarantee that no attack within a certain perturbation budget can cause a misclassification.

- **Convex Relaxations / Verification Methods:** These methods use formal verification techniques to prove the robustness of a network.
- **Randomized Smoothing:** This powerful technique creates a "smoothed" classifier by averaging the predictions over random noise added to the input. This provides a probabilistic guarantee of robustness.

Certified defenses offer strong guarantees but often come at the cost of reduced accuracy on clean data and are typically limited to smaller models and datasets.

3 The Arms Race Continues

The field of adversarial robustness is in a constant state of flux, with new attacks and defenses being developed in a continuous cycle. Currently, adversarial training remains the most reliable empirical defense, though it is computationally expensive and its effectiveness is limited to the specific threat model it was trained on. Many early defenses that relied on "security through obscurity" have been broken by adaptive attacks.

3.1 Key Takeaway

Building truly robust deep learning models is a significant challenge. A critical approach is necessary when evaluating new defenses, ensuring they are tested against strong, adaptive attacks. The focus should be on developing defenses that provide genuine robustness rather than simply obscuring gradients. The combination of robust optimization techniques like adversarial training with other clever regularization or preprocessing methods appears to be the most promising direction for future research.

References

[1] Purdue University College of Engineering. (n.d.). *Robustness in Deep Learning: Adversarial Attacks and Defenses*. Retrieved from engineering.purdue.edu

[2] Goodfellow, I. J., Shlens, J., Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572*.

[3] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P. (2018). Ensemble Adversarial Training: Attacks and Defenses. In *International Conference on Learning Representations*.

[4] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A. (2018). Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

[5] Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., & Li, J. (2018). Boosting Adversarial Attacks with Momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.