CS F437

REPORT TITLE:

Literature Review on Adversarial Attacks in Latent Space

SUBMITTED IN PARTIAL FULFILLMENT OF THE COURSE CS F437: GEN-AI

INSTRUCTOR IN CHARGE: Prof. Poonam Goyal

SUBMITTED BY:

Aabir Shubhajit Sarkar	2022A3PS0473P
Rana Kashyap Chitrang	2022A7PS0448P

Literature Review on Adversarial Attacks in Deep Learning

1. Introduction

Deep neural networks (DNNs) have transformed areas like computer vision, natural language processing, and reinforcement learning, reaching impressive levels of performance in tasks such as image classification, speech recognition, and autonomous decision-making. Despite these achievements, their susceptibility to adversarial attacks—tiny, often invisible changes to inputs that lead to incorrect outputs—has raised serious concerns about their reliability and robustness.

This literature review explores the main strategies for crafting adversarial attacks, with a focus on both pixel-space and latent-space approaches. In particular, it highlights the method introduced by Shukla and Banerjee (2023) in their CVPR workshop paper *Generating Adversarial Attacks in the Latent Space*. The review also delves into gradient-based techniques, GAN-driven frameworks, and the geometric interpretations behind these attacks, drawing from both foundational and recent research to provide a well-rounded view of the current landscape.

2. Background on Adversarial Attacks

Adversarial attacks take advantage of the fact that deep neural networks can be surprisingly sensitive to small changes in their input. Even tiny, almost invisible tweaks can cause a model to make completely wrong predictions. This phenomenon was first highlighted by Szegedy et al. (2014), who showed that top-performing models could be fooled by minimal, human-imperceptible changes.

To keep these perturbations subtle, most attack methods enforce constraints using norms like the L_1 or $L\infty$ norm. These constraints ensure the altered input still looks very similar

to the original. Based on the amount of information available to the attacker, adversarial attacks fall into two broad types:

- **White-box attacks**, where the attacker has full knowledge of the model's structure and weights.

- **Black-box attacks**, where the attacker can only query the model and doesn't know its internal workings.

The core goal in both settings is to cause the model to misclassify the input, all while keeping the changes visually imperceptible.

3. Gradient-Based Adversarial Attacks

Gradient-based methods are among the earliest and most widely studied approaches for generating adversarial attacks. These methods leverage the gradients of the model's loss function concerning the input to craft perturbations.

3.1 Fast Gradient Sign Method (FGSM)

Goodfellow et al. (2015) introduced the Fast Gradient Sign Method (FGSM), a single-step attack that perturbs the input in the direction of the loss gradient's sign. For an input (x), label (y), and loss function (L), the adversarial example is computed as:

$$x' = x + \varepsilon \cdot \text{sign}(\nabla_x L(x, y))$$

where ε controls the perturbation magnitude. FGSM is computationally efficient but often less effective than iterative methods due to its single-step nature.

3.2 Projected Gradient Descent (PGD)

Madry et al. (2018) proposed Projected Gradient Descent (PGD), an iterative extension of FGSM. PGD applies multiple small perturbations, projecting each step onto an ε-ball to ensure the perturbation remains within the allowed margin:

$$x^{t+1} = \Pi_e(x^t + \alpha \cdot \text{sign}(\nabla_x L(x^t, y)))$$

PGD is more robust and achieves higher attack success rates (ASRs) but is computationally intensive. Shukla and Banerjee (2023) compare their latent-space method to PGD, noting comparable performance without requiring noise margin constraints.

3.3 DeepFool

Moosavi-Dezfooli et al. (2016) introduced DeepFool, which minimizes the perturbation norm by iteratively pushing the input toward the nearest decision boundary. DeepFool linearizes class boundaries around the current input, forming a convex polyhedron, and updates the input until it crosses the boundary. This method achieves high ASRs with smaller perturbations than FGSM but assumes a locally linear decision boundary, which may not always hold.

3.4 Jacobian-Based Saliency Map Attack (JSMA)

Papernot et al. (2016) proposed the Jacobian-Based Saliency Map Attack (JSMA), which selectively perturbs pixels with the highest impact on the loss gradient. By focusing on a small subset of pixels, JSMA produces sparse perturbations, making it suitable for scenarios where minimal changes are desired. However, its iterative nature increases computational cost.

4. Pixel-Space vs. Latent-Space Attacks

Traditional adversarial attacks, such as **FGSM**, **PGD**, and **DeepFool**, operate in the pixel (input) space, adding noise constrained by **L□-norm** bounds to ensure visual similarity. These methods rely on geometric priors (e.g., **ε-balls**) to regulate perturbation magnitude. However, **Shukla and Banerjee (2023)** argue that since classification decisions occur in the high-dimensional **latent (feature) space**, directly perturbing features could bypass the need for such priors while maintaining visual realism.

4.1 Pixel-Space Attacks

Pixel-space attacks are intuitive and widely used due to their direct manipulation of input data. However, they face several limitations:

- **Noise Margin Dependency:** The reliance on **L□-norm** constraints can limit the attack's flexibility and effectiveness.

- **Visual Artifacts:** Even small perturbations can introduce noticeable artifacts, especially in complex datasets like **CIFAR-100** or **Stanford Dogs**.

- **Interpretability:** Pixel-space perturbations offer limited insight into the model's **feature-level vulnerabilities**.

4.2 Latent-Space Attacks

Shukla and Banerjee (2023) propose a **GAN-based framework** to generate adversarial attacks in the **latent space**, eliminating the need for margin-based priors. Their method uses an **encoder-decoder architecture** as the generator and a **discriminator** to classify samples, achieving high **Attack Success Rates (ASRs)** (e.g., **92.86%** on **MNIST** for targeted attacks) while preserving **visual realism** (**SSIM = 0.85** for untargeted attacks on MNIST).

Key advantages include:

- **Minimal Regulation:** Latent-space perturbations do not require strict noise margins, allowing greater flexibility.

- **Geometric Interpretation:** Perturbations can be visualized as movements toward the **convex hull** of the target class, providing intuitive insights.

- **Generalizability:** The method supports both **targeted** and **untargeted attacks** across datasets like **MNIST**, **CIFAR-10**, **Fashion-MNIST**, **CIFAR-100**, and **Stanford Dogs**.

Creswell et al. (2017) explored latent-space attacks in **LatentPoison**, focusing on the vulnerability of **variational autoencoders**. However, their method requires **ε-bounds** and is limited to **security perspectives**. **Upadhyay and Mukherjee (2021)** extended this work to untargeted attacks by generating **out-of-distribution samples**, but it lacks the **end-to-end framework** of Shukla and Banerjee (2023).

5. GAN-Based Adversarial Attacks

Generative Adversarial Networks (GANs) have emerged as powerful tools for generating adversarial examples, leveraging their ability to model data distributions.

5.1 AdvGAN

Xiao et al. (2018) introduced AdvGAN, which uses a GAN to learn the distribution of original samples and generate adversarial examples efficiently. AdvGAN employs a soft hinge loss to constrain perturbation magnitude, achieving high ASRs. However, it operates in the pixel space and requires a user-specified bound, unlike the latent-space approach of Shukla and Banerjee (2023).

5.2 ATGAN

Yang et al. (2021) proposed ATGAN, a GAN-based attack that does not require a target model. ATGAN uses an autoencoder and weight parameters to balance attack strength and image quality. Shukla and Banerjee (2023) compare their method to ATGAN, reporting higher ASRs (90.83% vs. 81.78% on MNIST for untargeted attacks).

5.3 Shukla and Banerjee's GAN Framework

The framework proposed by **Shukla and Banerjee (2023)** uses a **GAN** with an **encoder-decoder generator** and a **ResNet-18-based discriminator**. The loss functions are designed to simultaneously ensure **realistic image generation** and **successful misclassification**:

- **Discriminator Loss:**
  $\mathcal{L}^D = \lambda_1 \cdot (1/m) \sum_{i=1}^{m} \mathcal{L}\_ce(\mathcal{D}(x^{(\mathbf{i})}), y^{(i)}) + \lambda_2 \cdot (1/m) \sum_{i=1}^{m} \mathcal{L}\_ce(\mathcal{D}(\mathcal{G}(x^{(\mathbf{i})})), \tau)$

- **Generator Loss:**
  $\mathcal{L}^G = \gamma_1 \cdot (1/m) \sum_{i=1}^{m} \mathcal{L}\_ce(\mathcal{D}(\mathcal{G}(x^{(\mathbf{i})})), \tau) + \gamma_2 \cdot \mathcal{L}_1(x, \bar{x})$

This approach achieves high **visual realism** (**PSNR = 21.7** for untargeted attacks on **MNIST**) and **outperforms pixel-space baselines**, which rely on **autoencoders** and **convex hulls** but achieve lower **ASRs** (e.g., **32.06%** on MNIST for targeted attacks).

6. Geometric Interpretations of Adversarial Attacks

Geometric perspectives provide valuable insights into adversarial attacks, particularly in the latent space. Yousefzadeh (2020) analyzed the convex hulls of features produced by DNNs to study model generalizability. Shukla and Banerjee (2023) extend this idea, interpreting latent-space perturbations as movements toward the convex hull of the target class (for targeted attacks) or the nearest class (for untargeted attacks). Their visualizations, using t-SNE embeddings and class activation maps, show well-separated clusters for original and adversarial samples, indicating effective attacks.

In contrast, pixel-space attacks like DeepFool (Moosavi-Dezfooli et al., 2016) linearize class boundaries to form convex polyhedra, but their geometric interpretation is less intuitive due to the high-dimensional input space. The latent-space approach offers a clearer visualization of how perturbations alter feature representations.

7. Evaluation Metrics and Datasets

Adversarial attack methods are evaluated using metrics like Attack Success Rate (ASR), Structural Similarity Index Measure (SSIM), and Peak Signal-to-Noise Ratio (PSNR). ASR measures the proportion of samples successfully misclassified, while SSIM and PSNR quantify visual similarity between original and perturbed images. Shukla and Banerjee (2023) report high ASRs (e.g., 98.94% on Stanford Dogs for targeted attacks) and reasonable SSIM/PSNR values, indicating effective attacks with minimal visual degradation.

Common datasets include:

- **MNIST**: Handwritten digits, used for simple classification tasks.
- **CIFAR10/CIFAR100**: Small-scale color images, testing robustness on complex data.
- **Fashion-MNIST**: Grayscale clothing images, bridging MNIST and CIFAR-10 in complexity.
- **Stanford Dogs**: Fine-grained classification, challenging due to high intra-class variability.

8. Challenges and Future Directions

Despite significant progress, adversarial attacks face several challenges:

- **Scalability**: Latent-space methods, while effective, require training complex GANs, which can be computationally expensive.
- **Generalizability**: Methods must generalize across diverse datasets and model architectures.
- **Defense Mechanisms**: Adversarial training (Madry et al., 2018) and robust optimization techniques are improving model resilience, necessitating stronger attacks.
- **Interpretability**: Understanding why certain perturbations succeed remains an open question, particularly in the latent space.

Future research could focus on:

- Developing hybrid pixel- and latent-space attacks to combine their strengths.
- Exploring unsupervised or semi-supervised GAN frameworks to reduce training costs.
- Investigating the transferability of latent-space attacks across different models and datasets.
- Enhancing geometric and visualization techniques to better understand feature-space vulnerabilities.

9. Implementation Plan

To implement adversarial attacks in the latent space, here's my overall approach:

1. **Model Setup**: I'll either build or use an existing generative model, such as a GAN, which can transform a random input (latent vector) into a realistic image. The model should be capable of producing images that resemble those in the dataset.

2. **Latent Vector Perturbation**: Next, I'll devise a method to subtly modify (perturb) the input latent vector before passing it into the generator. The goal is to make small changes that don’t drastically alter the image’s appearance to humans but are enough to mislead a machine learning classifier into making an incorrect

prediction.

3. **Attack Validation**: After generating the image, I'll check if it successfully fools the classifier. If it does, the attack is considered successful. If not, I'll tweak the latent vector and try again.

4. **Repeat and Track**: This process will be repeated across different images, and I'll keep track of how often the attack succeeds, as well as how realistic the images appear.

5. **Comparison**: Finally, I'll compare the effectiveness of this latent-space attack to traditional methods that directly modify pixel values. This comparison will help determine if the new approach is more effective and produces more natural-looking results.

By making these subtle modifications to the model's internal representation rather than adding visible noise to the image itself, this method avoids altering the image in ways that are obvious to humans, while still creating adversarial examples that confuse the classifier.

10. Conclusion

Adversarial attacks highlight critical vulnerabilities in deep neural networks, prompting extensive research into robust defense mechanisms. Gradient-based methods like FGSM, PGD, and DeepFool dominate pixel-space attacks, while GAN-based approaches, such as those by Xiao et al. (2018) and Shukla and Banerjee (2023), offer flexible alternatives. The latent-space framework proposed by Shukla and Banerjee (2023) represents a significant advancement, achieving high ASRs without noise margin constraints and providing geometric insights via convex hulls. By synthesizing findings from seminal works (Szegedy et al., 2014; Goodfellow et al., 2015) and recent innovations, this review underscores the evolving landscape of adversarial attacks and the need for continued exploration to enhance model robustness.

References

1. Goodfellow, I., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. *International Conference on Learning Representations*.
2. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*.
3. Moosavi-Dezfooli, S., Fawzi, A., & Frossard, P. (2016). DeepFool: A simple and accurate method to fool deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*.
4. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. *IEEE European Symposium on Security and Privacy*.
5. Shukla, N., & Banerjee, S. (2023). Generating adversarial attacks in the latent space. *CVPR Workshop*.
6. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014). Intriguing properties of neural networks. *International Conference on Learning Representations*.
7. Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., & Song, D. (2018). Generating adversarial examples with adversarial networks. *International Joint Conference on Artificial Intelligence*.
8. Yang, G., Li, M., Fang, J. X., Zhang, & Liang, X. (2021). Generating adversarial examples without specifying a target model. *PeerJ Computer Science*.
9. Yousefzadeh, R. (2020). Deep learning generalization and the convex hull of training sets. *NeurIPS Workshop: Deep Learning through Information Geometry*.
10. Creswell, A., Bharath, A. A., & Sengupta, B. (2017). LatentPoison - Adversarial attacks on the latent space. *arXiv:1711.02879*.
11. Upadhyay, U., & Mukherjee, P. (2021). Generating out of distribution adversarial attack using latent space poisoning. *IEEE Signal Processing Letters*.