Shanghai Jiao Tong University

Topic chosen: An instance of an AI poisoning attack

Ng Tze Kean
Student number: 721370290002

Contents

1 Abstract

Deep neural networks have been revolutionary in their fields but their vulnerability towards cyber attacks have been raising concerns. Over the years, there have been many different ways that have been developed to attack DNNs of which we will be looking into attacks on training data which will cause misclassifications by DNNs. We will take a look at a case study of an event where a DNN was attacked by image poisoning and we will analyze the impacts. We will implement the attack proposed by SpotOn research paper as an attempt to simulate an image poisoning and we will test the implementation on a DNN model to show how an image can be poisoned without noticeable difference to the human eye.

2 Introduction

In this assignment we will look into data poisoning of images and how it can raise security concerns when it comes to training DNNs. The paper SpotOn describes how a different approach is taken compared to the traditional implementation of the Fast Gradient Sign Method (FSGM). In brief, FSGM works by leveraging on how DNNs are designed to learn, which is through gradient descent. In FSGM, the gradient of the loss function is calculated to estimate how much the DNN prediction changes for a change in the input data. A small amount of noise is then added to the input data in the direction of the gradient whereby the DNN will make the largest mistake. Typically an ϵ is used to control the amount of noise added to the image, the greater the ϵ the larger the degradation of the image.

These images that are modified typically contain correct labeling and is difficult to discern by the human eye if there was any modification. The poisoned images are usually fed along with legitimate training data to the target DNN. As the DNN is trained, it will be poisoned by the training data. This can cause effects such as wrong associations being learnt by the model. For example, real data points that are similar to the poisoned input can be wrongly classified. Another impact could be a decision boundary shift. The poisoned image can act like a force that shifts the decision boundary of the DNN, causing what should be classified correctly to be misclassified due to the shift.

These are some of the consequences of a DNN being trained on poisoned images. We will now explore how the images could be poisoned and a implementation of such an technique.

3 Case study

3.1 Report: Analysis of Data Poisoning on Generative AI

This report analyzes the impact of data poisoning on Generative AI (GANs) through the lens of the article “This new data poisoning tool lets artists fight back against generative AI” by MIT Technology Review. We will explore how the poisoning attack is carried out, its potential consequences, and its broader implications for neural networks in other fields.

3.2 In-Depth Analysis of Nightshade: Pixel-Level Poisoning of Generative AI

The Nightshade tool disrupts Generative Adversarial Networks (GANs) by manipulating the training data at a pixel level.

1. Carrier Selection: The artist chooses an original image they wish to protect. Nightshade can handle various image formats like JPEG or PNG.

2. Imperceptible Perturbations: Nightshade employs a technique called adversarial patching. It introduces slight modifications to a small subset of pixels within the image. These changes are statistically insignificant and imperceptible to the human eye.

3. Fast Fourier Transforms (FFTs): Nightshade likely utilizes FFTs to manipulate the image in the frequency domain. Here, the image data is transformed from the spatial domain (pixel positions) to the frequency domain (representing spatial frequencies).

4. Targeted Frequency Band: Within the frequency domain, Nightshade focuses on specific frequency bands that influence the model’s perception of the image’s features. These bands might correspond to edges, textures, or specific object characteristics.

5. Shifting the Spectrum: Nightshade subtly alters the chosen frequency bands. This could involve scaling specific frequencies up or down, introducing phase shifts, or injecting carefully crafted noise.

6. Inverse FFT: After manipulating the chosen frequency bands, Nightshade applies an inverse FFT to convert the data back to the spatial domain. The resulting image appears visually identical to the original but carries the embedded ”poison.”

3.3 Impact on GAN Training:

Corrupted Gradients and Loss Landscape:

1. Gradient Descent: During training, GANs rely on a technique called gradient descent to optimize their internal parameters (weights and biases). These parameters determine how the network interprets and generates images. Gradient descent works by iteratively adjusting the parameters in the direction that minimizes the loss function - a measure of how well the model performs.

2. Poisoned Gradients: Nightshade’s manipulation in the frequency domain introduces misleading information into the training data. This disrupts the natural distribution of pixel values the network expects. As the model processes the poisoned image, it calculates gradients based on the corrupted data points. These ”poisoned” gradients steer the parameter updates in the wrong direction, pushing the model away from the optimal solution.

3. Loss Landscape Distortion: Imagine the loss function as a landscape with valleys representing optimal solutions. The training process aims to navigate this landscape and reach the lowest valley (minimum loss). Nightshade introduces additional ”hills” and ”valleys” into the landscape due to the misleading gradients. This distorts the model’s search path, making it difficult to find the true minimum and hindering its ability to learn accurate representations of the data.

3.4 The Result: Disrupted GAN Outputs

The overall effect is a significant degradation in the GAN's performance. This can manifest in several ways:

1. Mode Collapse: This can lead to a phenomenon called mode collapse, where the network gets stuck generating a limited set of outputs that reflect the ”poisoned” information. Imagine a GAN consistently producing blurry cat faces with misplaced features due to its reliance on the misleading cues injected by Nightshade.

2. Nonsensical Outputs: In severe cases, the model might generate entirely nonsensical outputs. The conflicting information from the poisoned data disrupts the network's internal representation to such an extent that it can no longer produce coherent images.

3. Reduced Image Quality: The generated images might become blurry, distorted, or lack coherence due to the model's inability to accurately represent the underlying data.

4. Reduced Generalizability: Nightshade effectively reduces the network's ability to generalize from the training data. The model becomes fixated on the ”poisoned” information, struggling to capture the true diversity of real-world images.

3.5 Potential Impacts of Data Poisoning

1. Disruption of AI Development: Widespread use of Nightshade could hinder the development of GANs. If AI companies can't trust the training data, they might be hesitant to invest in these models.

2. Malicious Applications: The concept of data poisoning could be exploited for malicious purposes. Hackers could target critical AI systems in fields like finance or healthcare, causing significant disruptions.

3. Data Arms Race: A ”data arms race” might emerge, with artists using Nightshade and AI developers scrambling to create countermeasures. This could lead to a constant back-and-forth struggle.

4. Data poisoning is not limited to GANs. It's a potential threat to any neural network that relies on vast amounts of training data. Fields like self-driving cars or facial recognition systems could be vulnerable if bad actors tamper with the data used to train their underlying neural networks.

3.6 Conclusion

Nightshade highlights the vulnerability of AI models to data poisoning. While it empowers artists to protect their work, it also raises concerns about the robustness of AI systems in various fields. Moving forward, researchers need to develop methods for detecting and mitigating data poisoning attacks to ensure the safe and reliable development of AI.

4 Technique studied for demo

4.1 Background

As mentioned in the introduction, the use of FSGM as a technique has its limitations whereby all the images of the pixels are poisoned, that is the image quality is ostensibly degraded. We want to implement a demonstration of a program that tries to poison an image and to show that the final image may not be obvious but has impacts on the DNN being trained. We will looking into the paper, SpotOn, which proposes the use of Gradient weighted Class Activation Mapping(Grad-CAM) to identify the areas to be targeted, known as Region of Importance (ROI) such that there is minimal modification to the original image.

The use of Grad-CAM allows the attacker to find the neurons in the last convolutional layer that play a greater role in making the classification. The specifics of Grad-CAM will not be discussed here but can be found in the paper “Grad-cam: Visual explanations from deep networks via gradient-based localization". Through Grad-CAM, the pixels which are of high ROI can be identified, allowing the attacker to modify the image with minimal alterations to the overall image.

4.2 Purpose of demo

The SpotOn technique offers a valuable lens to examine Nightshade's potential impact on AI security. we try to draw parallels to the Nightshade through this demo, along with ways to strengthen the demo to better simulate a Nightshade attack:

Targeted Manipulation: Both Nightshade and SpotOn manipulate data to influence the model's behavior. Nightshade alters training data at a pixel level, while SpotOn targets specific regions (ROI) within an image.

Impact on Model Learning: Nightshade injects imperceptible changes during training, causing the model to learn incorrect associations between pixels and concepts. SpotOn manipulates a single image to deceive a trained model, altering its classification for that specific image.

Batch Poisoning: Instead of a single image, manipulate a batch of training images containing the target class. Apply targeted noise to the ROI in each image using similar techniques as SpotOn. This injects "poisoned" data into the training process.

Iterative Refinement: Train the model on the poisoned dataset. Analyze the misclassified outputs and refine the noise manipulation strategy for subsequent poisoning iterations. This mimics how Nightshade might evolve its tactics over multiple training cycles.

We also recognise the limitations of this demo below.

1. Our demo uses a pre-trained classification model (Alexnet). In a real Nightshade attack, generative models are the target. We can improve the simulation by using a generative model for testing, like a Variational Autoencoder (VAE) or Generative Adversarial Network (GAN).

2. The demo uses basic noise addition for ROI manipulation. Nightshade likely employs more sophisticated techniques to make manipulations imperceptible. We can explore advanced gradient masking or optimization algorithms to create more realistic "poisoned" images.

3. Our demo focuses on manipulating a specific ROI. Nightshade might target more complex relationships between objects or manipulate the entire image in a subtle manner. We can explore incorporating techniques like backdoor embedding or spatial transformations to achieve a more nuanced poisoning effect.

4.3 Implementation

We referred to an existing implementation of the SpotOn attack on the GitHub, which can be found here (https://github.com/CandleLabAI/SpotOn-AttackOnDNNs). We modified the code base as it was not working as intended due to package differences.

Overall, we can study the code to try and understand how the attack is carried out. We can see that for each image, GradCam is used to identify the ROI and the image is modified with the noise added to the ROI. The function `extract_salmap` is used to extract the saliency map of the image and the ROI is identified. Since there are 2 types of attacks, we will be focusing on the only_in_roi attack where the noise is added only to the ROI. The noise is added to the ROI. A structural similarity comparison is done to compare the original image and the modified image. The image is then tested on the model to see if the image is

The image is then tested on the model to see if the image is misclassified. We see that there is no significant difference to the human eye but the model is poisoned as it is trained on the dataset.

To summarise the implementation, we can refer to the process below

1. Load Pre-trained Model: We'll use a pre-trained model Alexnet with default weights loaded as the model for testing

2. Load and Preprocess Image: Load an image containing the target class.
3. Grad-CAM Calculation: Use Grad-CAM to generate a heatmap highlighting the ROI influencing the target class prediction.
4. ROI Manipulation: Apply noise (calculated using gradients) to the ROI to nudge the model towards the misclassification.
5. Test the Model: Feed the original and poisoned image to the model and observe the change in prediction.

4.4 Usage

The program modified is tested and is able to run on

- Python 3.11.9
- scikit-image 0.23.2
- scikit-learn 1.3.2
- torch 2.3.0
- opencv-python 4.9.0.80

Some slight modifications made to the program include the modification of the structural similarity comparison to test for only colored images. The original code also implemented 3 different models to test on, which were pretrained, instead for this simulation it was modified to take in a default model from torch. The dataset used for this test was the 101_ObjectCategories dataset which can be downloaded here.

```
    python run_attack.py
        --epsilons 0.005 0.006 0.007
        --mean 0.485 0.456 0.406
        --std 0.229 0.224 0.225
        --logs only_roi.txt
        --data 101_ObjectCategories/
        --type_attack only_in_roi
        --thresh_lambda 0.3
```

5 Conclusion

In conclusion, we can see that through running the code, we are able to poison the image without any noticible change to the human eye. Such image poisoning can be carried out over a large scale on the internet such that images that might be scrapped by webcrawlers could be poisoned. This might lead to GenAI that are trained over such images to be slowed poisoned as the incremental amount of data being fed to these deep learning algorithms increase.

This also pertains to the case study that we have examined above in the case of Nightshade. Though we have to also recognised that the techniques used in the study is far more advanced than what was done in this lab experiment. But the implications are the same, that the poisoning of images can have a significant impact on the DNNs being trained.

References

[1] Yash Khare et al. "SpotOn: A Gradient-based Targeted Data Poisoning Attack on Deep Neural Networks". In: *24th International Symposium on Quality Electronic Design (ISQED)*. California, USA, 2023.

[2] Fei-Fei Li et al. *Caltech 101*. Apr. 2022. DOI: 10.22002/D1.20086.