Big Project Build a ResNet20 Models

Khaula Sayyidatunadia

June 16, 2024

1 Introduction about ResNet

Adversarial images refer to specially crafted inputs that are designed to exploit vulnerabilities in machine learning models, particularly deep neural networks (DNNs). These inputs are intentionally perturbed in a way that causes the model to misclassify them, despite their appearances being virtually indistinguishable from normal examples to human observers

1.1 Types of Attack

1.2 White Box Attacks

Attackers have complete knowledge of the model architecture, weights, and training data, enabling them to craft highly effective adversarial examples.

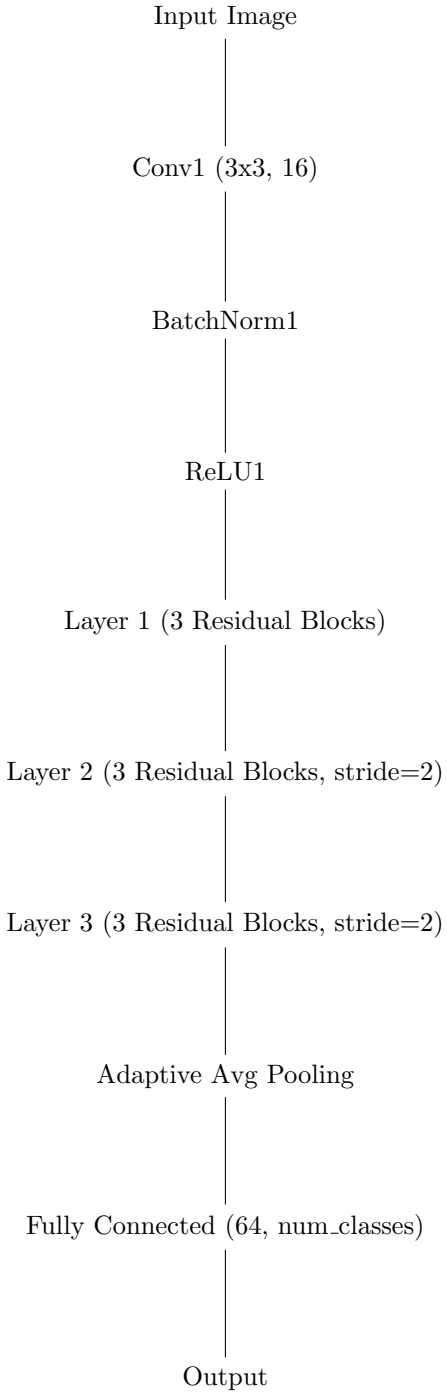
1.3 Black Box Attacks

Attackers have limited information about the model, often restricted to input-output pairs or confidence scores. Techniques like model inversion and query-based attacks are utilized in these scenarios. So, we start with a clean or true image that has a high accuracy of 99. Then, we apply an attack that reduces its accuracy. The goal is to assess whether our model architecture can correctly classify the image after it has been attacked. This involves comparing the model's performance before and after the attack.

1.4 Methodology

my structured approach to evaluate and defend against adversarial attacks using the Fast Gradient Sign Method (FGSM).

1. **Clean Image:** Start with an image correctly classified by the model with a high accuracy rate (e.g., 99%). This serves as the baseline "clean" or "true" image.
2. **Apply Attack:** Introduce an attack such as FGSM to perturb the clean image, aiming to manipulate it in a way that the model misclassifies it.
3. **Evaluate Model Performance:** Assess the model's ability to classify the attacked image correctly. Measure the accuracy on both the clean and attacked images.
4. **Compare Results:** Compare the accuracy of the model on the clean image (pre-attack) and the attacked image (post-attack). A significant drop in accuracy post-attack indicates vulnerability to adversarial examples.
5. **Develop Defense Strategies:** Based on the evaluation, develop and test defense strategies using different model architectures, training techniques, or adversarial training approaches. Aim to enhance the model's robustness against adversarial attacks.
6. **Iterative Improvement:** Iterate through steps 2 to 5 to refine defense strategies and improve the model's resilience to adversarial attacks.



1.5 Fast Gradient Methods

Objective: Generate an adversarial example x_{adv} from a given input x that causes misclassification by a machine learning model M.

Steps:

1. **Input:**
   - x: Original input image.
   - y_{true}: True label of x.
   - M: Machine learning model to attack.

- ϵ: Perturbation magnitude (small value to ensure imperceptibility).

2. **Compute Gradient:** Compute the gradient of the loss function $J(\theta, x, y_{true})$ with respect to x:
$$\nabla_x J(\theta, x, y_{true}) = \frac{\partial J(\theta, x, y_{true})}{\partial x}$$

3. **Generate Adversarial Perturbation:** Calculate the sign of the gradient to determine the direction of perturbation:
$$\text{sign}(\nabla_x J(\theta, x, y_{true})) = \text{sign}\left(\frac{\partial J(\theta, x, y_{true})}{\partial x}\right)$$
Create the perturbation vector:
$$\text{perturbation} = \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y_{true}))$$

4. **Construct Adversarial Example:** Generate the adversarial example by adding the perturbation to the original input:
$$x_{adv} = x + \text{perturbation}$$

5. **Output:** x_{adv} is the adversarial example that is perceptually similar to x but is likely to be misclassified by M.

Experiment

I conducted numerous experiments for this project. Initially, I tested the baseline code provided by my professor and achieved an accuracy of 0.00375. However, the misclassified images were still being incorrectly predicted. The results from the baseline code are as follows:

No	Origin	Prediction	Label
0	305	850	778
1	883	887	377
2	243	243	122
3	559	559	740
4	438	297	695

This indicates that the attack on the images has not been well-defined, and thus, we need to develop defense strategies to correctly identify the target labels.

Next, I tried changing the hyperparameter epsilon from 16 to 0.5, hoping that this would more effectively attack the images, resulting in an accuracy significantly lower than that for clean images. This would allow us to develop defense strategies and evaluate the model to correct the labels. However, the results remained unchanged, with the target labels still incorrect. The table after changing epsilon is as follows:

No	Origin	Prediction	Label
0	305	305	778
1	883	883	377
2	243	243	122
3	559	559	740
4	438	438	695

The resulting accuracy was 0.00. This indicates that the predictions were not adversarial images but rather identical to the original clean images. The FGSM (Fast Gradient Sign Method) attack defined in the training loop did not work effectively, resulting in an accuracy of 0.00. FGSM is intended to create adversarial examples by adding a small, calculated perturbation to the input images, which should mislead the model into making incorrect predictions. However, in this case, the FGSM attack did not generate effective adversarial examples, leading to no reduction in accuracy. This does not help in classifying the targets correctly, even though the predicted values matched the original images.

Next, I ran the code using a well-known dataset with many examples available online, the MNIST dataset, to test my understanding of the workflow of adversarial attacks. I replaced the ResNet50 model with a simple LeNet model to observe the accuracy on both clean and adversarial images and to see how accuracy increases when the machine correctly classifies the images.

Here is the workflow of my approach:

- **Import Libraries:** Necessary for handling data, building and training models, and implementing adversarial attacks.
- **Load Dataset "MNIST":** The MNIST dataset consists of handwritten digits and is widely used for benchmarking image classification models.
- **Divide into Test and Train Sets:** Essential for training the model on one subset of the data and evaluating its performance on another.
- **Define Normalization and Hyperparameters:** Normalization ensures the data is scaled properly, and hyperparameters control the learning process.
- **Define a Simple LeNet Model:** LeNet is a straightforward convolutional neural network architecture suitable for MNIST.
- **Train the Model on the Training Dataset:** Training involves learning patterns from the data to make accurate predictions.
- **Add FGSM Attack:** FGSM creates adversarial examples by adding perturbations to the input images, challenging the model's robustness.
- **Test the Dataset with the Defense Attack Function:** Evaluating the model's performance on adversarial examples helps in understanding its vulnerabilities.
- **Print the Accuracy, Plot, and Image:** Visualizing the results provides insight into the model's performance and the effectiveness of adversarial attacks.

Observations from FGSM Attack on LeNet Model

From the provided accuracy results using the LeNet model on the MNIST dataset with different epsilon values for the Fast Gradient Sign Method (FGSM) attack, we observe the following:

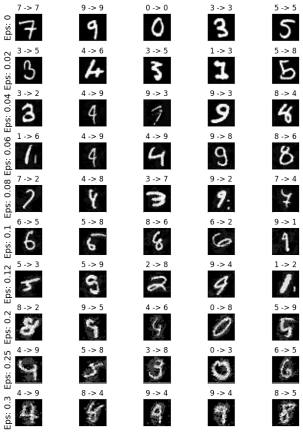


Figure 1: A result for Lesnet model with Mnist Dataset

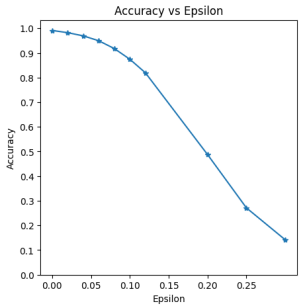


Figure 2: Comparing original images vs Adversarial Images

Epsilon	Test Accuracy
0	9912 / 10000 = 0.9912
0.02	9821 / 10000 = 0.9821
0.04	9693 / 10000 = 0.9693
0.06	9493 / 10000 = 0.9493
0.08	9177 / 10000 = 0.9177
0.1	8743 / 10000 = 0.8743
0.12	8194 / 10000 = 0.8194
0.2	4876 / 10000 = 0.4876
0.25	2714 / 10000 = 0.2714
0.3	1418 / 10000 = 0.1418

The accuracy results from applying the LeNet model to the MNIST dataset using the Fast Gradient Sign Method (FGSM) with different epsilon values reveal a clear trend. Initially, the model achieves a high accuracy of 99.12% on clean images, demonstrating its ability to accurately classify digit images. However, as we gradually increase the epsilon values from 0.02 to 0.3, the accuracy decreases significantly, dropping to 14.18%. This decline shows how FGSM effectively generates adversarial examples that cause the model to make incorrect predictions. While FGSM successfully reduces accuracy, it also highlights a key challenge: these attacks do not inherently help the model learn to classify correctly. Addressing this challenge requires developing robust defense strategies that can mitigate the impact of adversarial examples, thereby improving the model's overall reliability and accuracy in practical applications.

Experiment with 2D Convolutional Model

Finally, I experimented with a simple 2D convolutional model on the MNIST dataset, where I found that the FGSM attack did not significantly affect many images because I continued to obtain correct classifications even after increasing the epsilon values.

Figure 3: A result for Lesnet model with Mnist Dataset

Conclusion

The results from the baseline code and the updated code indicate that the initial model struggled with adversarial attacks, as evidenced by the incorrect labels even before applying the attacks. This failure highlights the model's inherent vulnerability to adversarial perturbations.

In conclusion, the experiments conducted using the LeNet model on the MNIST dataset with varying epsilon values for the Fast Gradient Sign Method (FGSM) attack highlight the vulnerability of deep learning models to adversarial examples. Initially, the model demonstrated high accuracy on clean images, but the introduction of adversarial perturbations significantly reduced the accuracy, indicating the effectiveness of the FGSM attack. This underscores the importance of developing robust defense strategies to mitigate the impact of such attacks.

The experimentation with a simple 2D convolutional model showed that certain model architectures might exhibit resilience to adversarial attacks, as evidenced by the continued correct classifications despite increased epsilon values. This suggests that model architecture plays a crucial role in adversarial robustness.

Overall, the results emphasize the need for ongoing research in adversarial defense mechanisms to ensure the reliability and accuracy of machine learning models in real-world applications. Future work should focus on refining these defense strategies to enhance model robustness against a variety of adversarial attacks.