Deceptive Pixels: Exploring the Landscape of Visualization Attacks in Neural Networks

Mohamed Abdel-Hamid

Department of Computer and Information Sciences, University of St. Thomas

Abstract

Adversarial model manipulation and adversarial training have emerged as significant areas of study in the context of enhancing the robustness of machine learning models against adversarial attacks. This study focuses on the exploration and analysis of these techniques, employing PyTorch for the implementation and evaluation of various adversarial attacks. The research builds upon and leverages two types of adversarial attacks: Passive and Active fooling, each with its unique approach to misleading model interpretations. Unlike traditional adversarial attacks that perturb input data, this attack fine-tunes a pre-trained model to generate misleading interpretations while maintaining the model's original performance. Passive fooling, including Location Fooling, Top-k Fooling, and Center-mass Fooling, generates uninformative interpretations while maintaining the model's accuracy. In contrast, Active fooling manipulates the model to produce completely incorrect interpretations that switch explanations between target classes.

Introduction

Basic Concepts

- Adversarial Model Manipulation: A malicious tactic in which model parameters are changed to fool interpretation methods. Unlike classic adversarial attacks, which alter input data, this method discreetly refines an existing model to produce false interpretations while retaining the model's original performance.
- Passive Fooling: This is a sort of adversarial approach in which non-informative interpretations are generated while the model's correctness is maintained. Location Fooling, Top-k Fooling, and Center-mass Fooling are all examples.
- Location Fooling: This type of attack manipulates the model to generate interpretations that highlight irrelevant locations in the image.
- Top-k Fooling: This attack manipulates the model to generate interpretations that highlight irrelevant top-k pixels in the image.
- Center-mass Fooling: This attack manipulates the model to generate interpretations that shift the center of mass of the interpretation.
- Active Fooling: This is another sort of adversarial approach in which the model is manipulated to give altogether inaccurate interpretations that point to irrelevant objects inside the image.
- Layer-wise Relevance Propagation (LRP), Grad-CAM, and Simple-Grad: These are interpretation methods used to understand the decision-making process of neural networks.
- Resnet50: This is a 50-layer deep convolutional neural network. It's a popular model for image categorization and other vision-related tasks.

Objectives

- Exploration and analysis of adversarial model manipulation and adversarial training approaches using PyTorch for development and assessment of various adversarial attacks.
- To comprehend the distinction between passive and active fooling tactics, as well as their influence on model interpretations.
- Determine the efficacy of various adversarial attacks on the model's performance and interpretation.
- To evaluate the strength of different attacks and their impact on the robustness of the model.

Background on Adversarial Training

- Need for Adversarial Training: Machine learning models, while highly accurate under normal conditions, can be easily fooled by adversarial examples. These examples, which are often indistinguishable from normal inputs to the human eye, can cause the model to make incorrect predictions. Adversarial training is used to make models more resistant to these types of attacks.
- Adversarial Training Process: During adversarial training, the model is trained not only on the original data but also on adversarial examples. This helps the model learn to correctly classify these tricky inputs, thereby improving its robustness.
- Challenges in Adversarial Training: One of the main challenges in adversarial training is the generation of effective adversarial examples. These examples need to be carefully crafted to fool the model, but not so different from normal inputs that the model can easily tell them apart.

General Theory

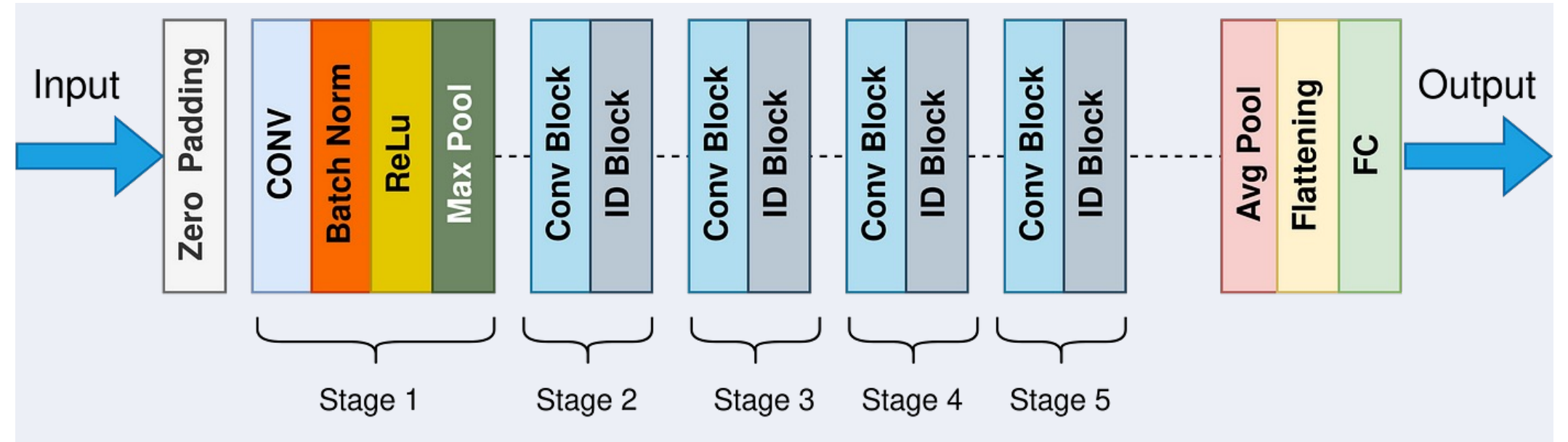


Figure 1: Resnet-50 model architecture

ResNet50 is a deep convolutional neural network with 50 layers, built upon the principle of residual learning with "skip" or "shortcut" connections to alleviate the problem of vanishing gradients. It consists of a single convolutional layer and max-pooling layer, followed by four stages each with a varying number of convolutional layers, and concludes with an average pooling layer and fully connected layer. Each stage, except the first, starts with a convolutional layer that reduces the spatial dimension, and utilizes identity and convolutional blocks to extract complex hierarchical features.

Methods

SOFTWARE AND TOOLS

Hardware

All the tests in the research were performed on a 2020 M1 Mac Machine with an M1 chip, featuring an 8-core CPU and 8-core GPU. The machine also had 16GB of unified memory and ran on macOS Ventura, the latest operating system at the time of the experiments. The M1 chip, known for its power efficiency and performance, provided a solid foundation for conducting the research experiments effectively and efficiently.

Google Colab

Google Colab is an open-source, interactive web tool used to run code and make annotations about it and allows you to run and save the notebook on your Google Drive, facilitating easy access and sharing with others.

PyTorch

This platform facilitates the development of machine learning algorithms by making it possible for participants to train models and dynamic computational graphs.

ALGORITHM

PROCESS OF ADVERSARIAL TRAINING

In the context of visual attacks, adversarial training could be employed to improve the robustness of the ResNet50 model. This involves generating adversarial examples, which are inputs designed to cause the model to make a mistake, and then training the model on these examples so that it can better resist these types of attacks in the future. The model used for the Resnet50 dataset has a 100% accuracy. The process of adversarial training goes as follows. First, the model must be trained on batches of data photos. The size of the batch used in the Resnet50 model consists of 64. After then, the training began with 50 epochs. Each epoch consisted of 256 iterations. The dataset consists of 4 different training folders, for Top-k, Center-mass, Location fooling and Clean images. Folders each consist of 900 images compressed into a .npz file format ready to go for the classifier to be trained. Testing folders alike, and they consist of 100 images each.

Process & Results

A machine learning model is used to process a picture, which can be a digital representation of any item, person, or scene, in the context in question. This model has filters, which are tiny matrices that change the input data (in this example, the picture) into a form that the model can learn from.

The idea of a 'gradient' comes into play in this approach, which in the context of machine learning refers to the slope of a function at a certain location. The function in this case is the model's loss function, which measures the model's performance. The gradient essentially serves as a guide for us to alter our parameters (filters) in order to improve the model's performance.

When 'ranking filters,' the process begins by sorting these filters in a certain order, frequently depending on their contribution to the model's final output. These filters are then modified in '10 increments,' implying that the filter values are altered incrementally to track the subsequent changes in the model's output.

Grad-CAM (Gradient-weighted Class Activation Mapping) is a method used to depict the model's attention or the regions in the picture that it thinks most relevant. An image's 'initial Grad-CAM' refers to the Grad-CAM created before to any filter alterations and serves as a baseline to compare with future Grad-CAMs generated after each filter modification.

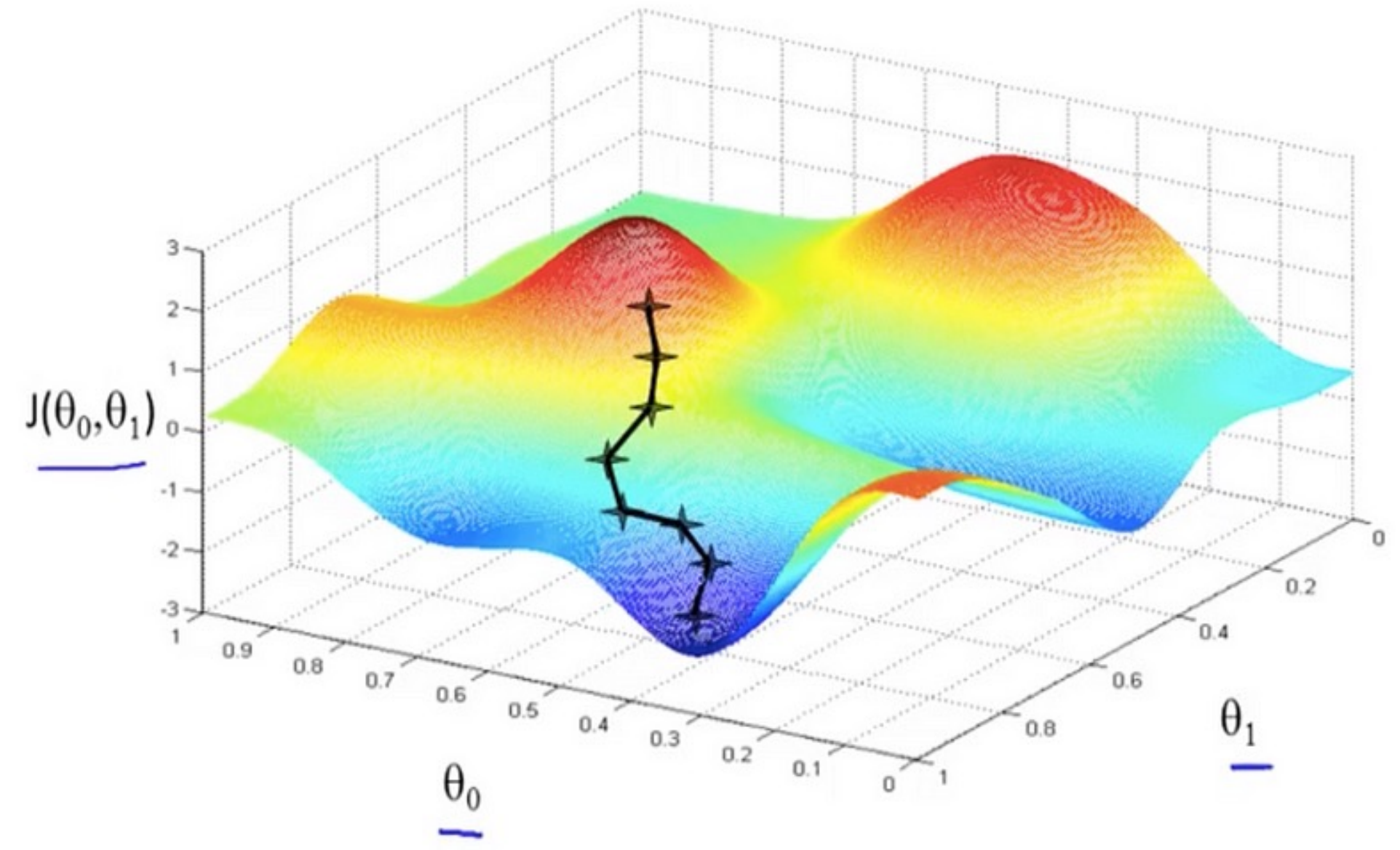


Figure 2: Gradient Calculation

In machine learning, the gradient helps us determine the direction to adjust our parameters. It's like a compass pointing us to the direction of the 'steepest descent' that we should take to minimize our loss function.

Our investigation produced compelling results, demonstrating the effects of training classifiers in a combination of Top-l or Center-mass or Location fooling attacks and clean data. Despite the integration of these fooling attacks, the classifiers exhibited 100% training accuracy, indicating the robustness and resilience of our models. Next step would be extensively testing the trained classifier with validation dataset.

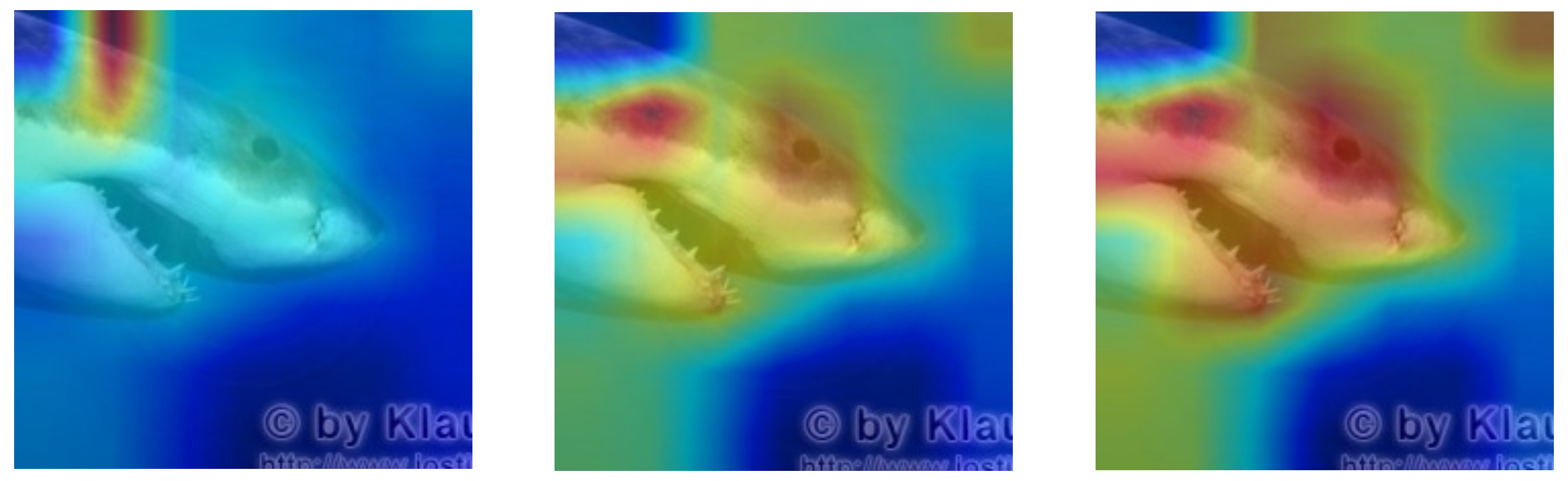
Fooling Type	Training Classifier Accuracy
Top-k	100%
Center-mass	100%
Location	100%

Figure 3: Value of Accuracy

Discussion

The study reveals that Passive fooling, which includes Location Fooling, Top-k Fooling, and Center-mass Fooling, can generate uninformative interpretations while maintaining the model's accuracy. This implies that the model, while appearing to perform well, is actually focusing on irrelevant locations or pixels within the image, thus leading to potentially misleading interpretations.

On the other hand, Active fooling manipulates the model to produce completely incorrect interpretations that switch explanations between target classes. Despite the misleading interpretations, the model's accuracy remains intact, demonstrating the deceptive nature of these attacks. These attacks are particularly effective against interpretation methods such as Layer-wise Relevance Propagation (LRP), Grad-CAM, and Simple-Grad, which are typically employed to understand the decision-making process of neural networks.



Figures 4-5-6: Different Grad-CAM visualizations that highlight different sections of the picture

References

- Juyeon Heo, Sunghwan Joo, Taesup Moon, 2019. Fooling Neural Network Interpretations via Adversarial Model Manipulation. https://arxiv.org/abs/1902.02041
- Suvaditya Mukherjee, 2022. The Annotated ResNet-50, Explaining how ResNet-50 works and why it is so popular. Published in Towards Data Science. https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758
- Renu Khandelwal, 2018. Machine learning : Gradient Descent. Published in Medium. https://arshren.medium.com/gradient-descent-5a13f385d403
- Google Colab. https://colab.research.google.com
- Pytorch. https://pytorch.org
- Grad-CAM Reveals the Why Behind Deep Learning Decisions. https://www.mathworks.com/help/deeplearning/ug/gradcam-explains-why.html

Acknowledgments

- This research experience was sponsored by the National Science Foundation and hosted by The Old Dominion University.
- Special thanks to my mentor Dr. Jiang Li.
- Special thanks to the directors of the program Dr. Chunsheng Xin, Dr. Peng Jian and Dr. Khan Iftekharuddin for the opportunity to participate in the research program.
- Special thanks to PhD student (Shahab Uddin) that guided me throughout this research.