Michael Gates

Deep Learning Final Project

Due April 22nd 2024

Pixel Geometric Decision Based Attacks on Large Language Models and a Custom Tensorflow2 Model

1. ABSTRACT

In this paper, we will be looking at the effectiveness of two popular machine learning model attack methods, Pixel attack and Geometric Decision Based attack (GeoDA). These attacks will be evaluated against the popular Large Language Models (LLMs) ChatGPT 4 Turbo, Google Gemini 1.5 Pro, and Ollama LLaVA. A custom model will also be evaluated.

The datasets used will be two models containing deepfake images of faces and real images of faces, real-and-fake-face-detection and 140k-real-and-fake-faces. Another dataset was added to make the attacks. The third dataset was of cats from the animals10. These are all hosted on Kaggle. Originally, this dataset was meant to be a filler dataset but it worked out better than expected. It would seem that some cats and humans share similarities that can sometimes fool LLMs and the custom model.

1. INTRODUCTION

Both Pixel attack and Geometric Decision Based attack are part of the Adversarial Robustness Toolbox Python package. This toolkit is commonly used to improve a deep learning model’s robustness through the generation of adversarial examples.

Pixel attack is an evasion attack that can be either targeted or untargeted. It uses four key parameters when setting up the attack. It needs a threshold set, if one is not set then it will use 124 as the default. If you do not set a limit on the max\_iter parameter, it will run for 100 interations. “ES” is the evolutionary strategy and this determines what type of attack will be used. 0 is Covariance Matrix Adaption Evolution Strategy (CMAES). It does not rely on the weights in the model, only the outputs. This means that it would take longer to find the optimal pixel placement to fool the model. 1 is Differential Evolution (DE). DE relies on the weights in the model as well as the output. This option isn’t always available since it targets white-box models. DE is generally faster and more effective than CMAES due to this. A collage of men

Description automatically generated

Pixel Attack Examples

Geometric Decision Based attack is a target attack. GeoDA is a black-box attack. It only takes into consideration the outputs of the model to generate the adversarial examples. GeoDA uses a combination of parameters that are mathematically linked. Adjusting the number of iterations also requires adjusting the sub-dimensions, lambda parameter, and the sigma parameter. This attack is not well documented, so the default were used. Adjusting these parameters is key to making adversarial examples that are hard to detect by the human eye. GeoDA works based on the assumption that that decision boundaries of deep neural networks often have low curvature (i.e., are fairly smooth) near real data points. This means an adversarial example should exist relatively close to the original image. A collage of a person with a rainbow colored background

Description automatically generated

GeoDA Examples

1. METHOD

The custom model was trained using Tensorflow2 across 24,709 images with 80% train and 20% test split. This uses 18,967 images in the training dataset and 4,742 images in the test dataset. All images were used in datasets with less than 10,000 images and datasets with more than that, the first 10,000 images were used. The images were resized to 150 pixels by 150 pixels. They were left in color. A sequential model was used with 12 layers, four 2D convolutional layers each followed by a MaxPooling layer, a flatten layer, followed by a dense layer with 256 neurons and another dense layer with 512 neurons. Finally a dense layer was added with three neurons using the softmax activation function, for a total of 5,830,211 trainable parameters. The model was trained for 15 epochs with a batch size of 32 on an Nvidia 4090 GPU. The final training loss was 0.0337 and the final training accuracy was 99%. The accuracy on the test set was 89%. Accuracy and loss over training epochs


Training Accuracy Base Model

A graph with numbers and a number of people

Description automatically generated with medium confidence

With the baseline numbers calculated, it is time to generate the attacks. Starting with Pixel attack. For Pixel attack, the X\_test was modified with adversarial examples. The parameters used were 10 for the threshold, 10 for the max iterations, and 1 for the “es” value. Giving es=1 means that we are using the differential evolution method of attack which uses the model’s weights to generate the most optimal placement of the pixels. After generating the images, the custom model’s accuracy went slightly down, to 75% from 89%. A graph of a comparison of a number of blue squares

Description automatically generated with medium confidence

GeoDA’s performace was evaulted on the same model. The default parameters were used for setting up the attack. This attack is a targeted attack, so a new label set was created for what I want the images to be classified as. In this case, I wanted the real faces to be misclassified and I did not care about what they got classified as. This was done by creating a new label called “y\_test\_real\_face” and then setting the label to 1 for the first class. 

Creating the target labels

After running GeoDA’s adversarial examples on the base model, the accuracy dropped significantly to 51% from 89%. The attack effectively removed the majorify of the real face predictions and made the model predict as either “fake” or “cat”. A graph of a comparison of a number of blue squares

Description automatically generated with medium confidence

The model was trained fresh from 50% normal training data and 50% training data created from the attacks. For Pixel attack with the evolutionary strategy equal to DE, the accuracy did not improve. This was expected, since DE uses the weights of the model, accuracy was 73%. A graph of a comparison of a number of blue squares

Description automatically generated with medium confidence

Confusion Matrix with es=1

Switching the evolutionary strategy from DE to CMAES and testing the model again showed that the model gained robustness against the black-box version of this attack. 88% accuracy on adversairal examples with CMAES. A graph of a comparison of a number of blue squares

Description automatically generated with medium confidence

Confusion Matrix with es=0

GeoDA training was performed similarly, a training dataset was created using 50% normal training data and 50% GeoDA examples. The robustness of the model increased when tested against the attack, 89% accuracy on the GeoDA adversarial examples when the baseline model received a 51% accuracy rating. A graph showing the difference between a number of people

Description automatically generated with medium confidence

1. VS LLMs

LLMs perform well on a number of various tasks, however, detecting if a face is a real face or a deepfake generated face is not one of those tasks. A zero-shot method was used to get comparable results from all 4 LLMs tested. Five images from each category were randomly selected and given to the LLMs to determine if they were “real”, “fake”, or “cat” images. The zero-shot method was used due to the limitation of Ollama LLaVa only being able to receive one image at a time and not maintaining the history over time. The test was performed on ChatGPT 4 Turbo, Google Gemini 1.5 Pro, Ollama LLaVa 7b, and Ollama LLaVa 34b. The API was used for ChatGPT 4 Turbo and Google Gemini 1.5 Pro. See the table below for accuracy ratings on the models. Ollama LLaVa 7b and 34b were ran locally and interacted with using the API on the locally running server.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Normal Images | PixelAttack Images | GeoDA Images |
| GPT 4 Turbo | 67% accuracy | 58% accuracy | 53% accuracy |
| Gemini 1.5 Pro | 67% accuracy | 58% accuracy | 67% accuracy |
| Ollama LLaVa 7b | 60% accuracy | 42% accuracy | 53% accuracy |
| Ollama LLaVa 34b | 67% accuracy | 50% accuracy | 67% accuracy |

1. CONCLUSION

Large Language Models are good at reading files, creating accurate summaries of long documents quickly, and generating code. They are not currently capable of beating a specifically trained model at determining if an image is fake or real, however, they are fairly good at determining if an image is a cat or a face. The LLM model with the best average accuracy was Google Gemini 1.5 Pro, second was Ollama LLaVa 34b.

Training models on adversarial examples can help improve a models robustness but it should be noted to mix in the adversarial examples through the entire dataset and not only in one class of image. Also, at some point using the attacks the images become unrecognizable. Accuracy metrics alone should not be the guiding factor at determining if a model is robust or not. GeoDA will continue to add noise until all that is visible in the image is the noise as shown in the images below. The images used to be faces.A close-up of a colorful pattern

Description automatically generated

GeoDA Too Noisy

1. Sources (Dataset and the documentation on how to use the attacks)s
2. https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection
3. https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces
4. https://www.kaggle.com/datasets/alessiocorrado99/animals10
5. https://adversarial-robustness-toolbox.readthedocs.io/en/latest/modules/attacks/evasion.html