Using Search To Produce Imperceptible Adversarial Examples

Author: Cristopher McIntyre Garcia
SN: 300025114
Email: cmcin019@uottawa.ca

Introduction

More and more computer vision **Machine Learning (ML)** models are being implemented into products such as smartphones, vehicles, and medical machinery. As such, these models need to be properly verified and tested for faults and vulnerabilities. In doing so, the models, and by extension the products, become safer and more robust to attacks. This project aims to develop a search algorithm to produces imperceptible Adversarial Examples (AE) given a dataset and model. Using a combination of explorative and exploitative methods, imperceptible image perturbations will be derived through a black-box method. Learning about these perturbations may give us more insight on how ML models learn to classify, and how we can reinforce them.

a search based approach will be used to find images perturbations and produce AE, shining light on model vulnerabilities. To make these examples imperceptible, another model will be used as an oracle...

for performing perturbations on images to produce AE. In doing so, we will hopefully develop a strategy for pointing out vulnerabilities in Machine learning models.

Multiple objective search will be used, the objectives being having one model perform no worst on the perturbed images and another model perform terribly. Images are very complex, and so we need methods to find *problems* in model implementations for real world tasks.

Related Work

The concept of AE was first introduced in [Szegedy, 2013] as an input to a ML model intentionally designed to cause misclassification. They report that in using hardly perceptible perturbations, they were able to make several networks misclassify images. Not only this, the same perturbations are effective at creating AE on different subsets of training data. That said, the only classifiers used were those using supervised methods with CNNs.

In [], a method for producing AE is presented. They use the model structure and gradients to then generate perturbations. They hypothesize that it is due to the linearity of neural networks that make it possible for them to. Though this method may be good for finding optimization based attacks, it does not protect against certain black-box attacks, as shown in [Athalye, 218].

Research Questions

Are local search algorithms able to efficiently produce adversarial examples? Maybe introducing a genetic search algorithm could increase performance?

Does introducing a second model with a different architecture to keep the perturbations minimal make it harder for humans to detect adversarial examples? Will using a self-supervised transformer falsely classify images the same way a CNN that was trained through supervised learning.

Are we able to infer any useful information about the model and its classification by seeing which noise affects it the most?

Dataset

For the purpose of this project, I will begin by using the MNIST dataset. If good results are achieved, a more complex dataset such as COCO or ImageNet will be used.

Overview

Using a simple CNN model, we will train two classifier that obtains around 99% accuracy on the MNIST dataset. These models will be treated as black-boxes, where one can only see the output of the model given an input. For each image in the testing set, we will perform perturbations using different search algorithms to hopefully generate a set of images that perform poorly with one model but not with the other, given a certain threshold. Different models might have to be used, this we will see during experiments.

The search algorithms I am interested in exploring are exploitative, explorative, and a combination of both. For a certain amount of iterations, we will compare the results with random perturbations to see if the algorithms are better.

The perturbations will be Gaussian noise that is added to the image. I will treat the search space as a plane and try adding Gaussian noise at least in certain areas (min amount everywhere like a grid)

These search algorithms will hopefully aid with the testing and validation of models by finding vulnerabilities in the models. (Think of continuous controllers)

Start form completely random image and change noise to get closer to real image, or start from image example and add subtle noise.

The search space will be all possible images. The neighbour of an image will be the image with a certain level of perturbations.

Analysis

To answer the first research question, I will develop an algorithm using