

Generating Adversarial Examples In Deep Neural Networks




Using FGSM to Fool ML Models

Supervisor: Dr. Farshid Abdollahi

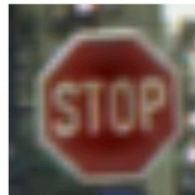
Author: Mohammad Ahmadi

May 16, 2022

Adversarial examples are inputs to a neural network that result in an incorrect output from the network; they're like optical illusions for machines. As an example from "Explaining and Harnessing Adversarial Examples ^[1]" paper, fast adversarial example generation applied to GoogLeNet (customized neural network architecture) on ImageNet (Dataset). By adding an imperceptibly small vector whose elements are equal to the sign of the elements of the gradient of the cost function with respect to the input, we can change GoogLeNet's classification of the image.

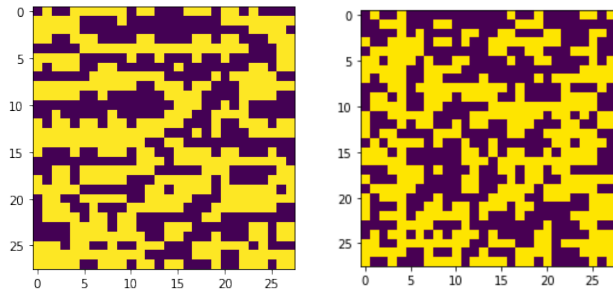
	$+ .007 \times$		$=$	
x		$\text{sign}(\nabla_x J(\theta, x, y))$		$x + \epsilon \text{sign}(\nabla_x J(\theta, x, y))$
"panda"		"nematode"		"gibbon"
57.7% confidence		8.2% confidence		99.3 % confidence

Adversarial examples have the potential to be dangerous. To illustrate ^[2] , consider the following images, potentially consumed by an autonomous vehicle: To humans, these images appear to be the same: our bio- logical classifiers (vision) identify each image as a stop sign. The image on the left is indeed an ordinary image of a stop sign. We produced the image on the right by adding a precise perturbation that forces a particular DNN to classify it as a yield sign. Here, an adversary could potentially use the altered image to cause a car without failsafes to behave dangerously.

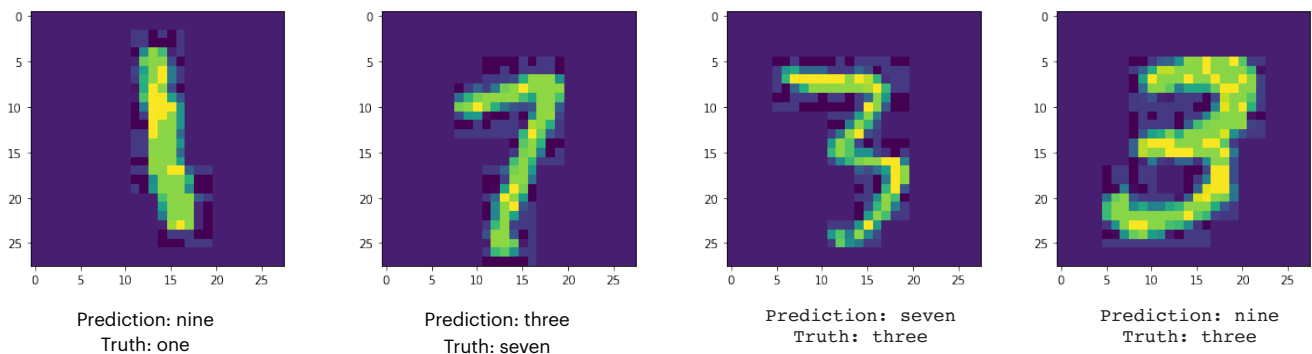


In my project I used labeled MNIST^[3] dataset (handwritten numbers from 0 to 9) 60000 training set and 10000 test set and applied them to a convolutional neural network with 3 convolutional layer, maxpooling, MSE loss function and softmax activation function. I used 3 Dropout layers in the network and Adam optimizer in order to make it more robust. My model obtained 0.98% accuracy on the dataset.

We need perturbation to combine it with the real images therefore we should use adversarial pattern function. To make an adversarial pattern I used Tensorflow Gradient Tape Method and MSE loss function to accurate the formula ($sign(\nabla_x J(\theta, x, y))$). Here are some examples of the patterns with 28 * 28 * 1 channels.



When the patterns got ready we can multiply it by ϵ which I set to 0.1 and add it to the original image. After evaluating with network accuracy decreased to 4%.



Traditional techniques for making machine learning models more robust, such as weight decay and dropout, generally do not provide a practical defense against adversarial examples. So far, I used combination of two methods in my project to defenses against adversarial examples. I attempted adversarial training^[4] which I trained network with 20000 generated adversarial example and Defensive

distillation^[5] where we train the model to output probabilities of different classes, rather than hard decisions about which class to output. The probabilities are supplied by an earlier model, trained on the same task using hard class labels. This creates a model whose surface is smoothed in the directions an adversary will typically try to exploit, making it difficult for them to discover adversarial input tweaks that lead to incorrect categorization. After applying these two methods accuracy of the model increased to 84%.

In conclusion adversarial examples show that many modern machine learning algorithms can be broken in surprising ways. Both strategies I have tested increases accuracy but so far fails because it is not adaptive, it may block one kind of attack, but it leaves another vulnerability open to an attacker who knows about the defense being used. As a result we need extra researches on AI security and robustness.

References:

- [1] Ian J Goodfellow, et al. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations, 2015.
- [2] Nicolas Papernot, et al. Practical Black-Box Attacks against Machine Learning. In Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security, Abu Dhabi, UAE, 2017.
- [3] Yann LeCun et al. The mnist database of handwritten digits, 1998.
- [4] Tao Bai, et al. Recent Advances in Adversarial Training for Adversarial Robustness. Accepted by International Joint Conference on Artificial Intelligence (IJCAI-21), 2021
- [5] Nicolas Papernot, et al. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. arXiv preprint arXiv: 1511.04508, 2017