International Journal of Intelligent Information Technologies
Volume 16 • Issue 1 • January-March 2020

A Comparative Study on Adversarial Noise Generation for Single Image Classification

Rishabh Saxena, VIT University, Vellore, India
Amit Sanjay Adate, VIT University, Vellore, India
Don Sasikumar, VIT University, Vellore, India

ABSTRACT

With the rise of neural network-based classifiers, it is evident that these algorithms are here to stay. Even though various algorithms have been developed, these classifiers still remain vulnerable to misclassification attacks. This article outlines a new noise layer attack based on adversarial learning and compares the proposed method to other such attacking methodologies like Fast Gradient Sign Method, Jacobian-Based Saliency Map Algorithm and DeepFool. This work deals with comparing these algorithms for the use case of single image classification and provides a detailed analysis of how each algorithm compares to each other.

KEYWORDS

Adversarial Noise, Artificial Intelligence, Convolutional Neural Network, Deep Learning, Generative Adversarial Network, Handwritten Digits, Image Classification, Machine Learning

1. INTRODUCTION

Generative models have become the dominant form of data generation tool in recent years due to their vastly superior results and optimized method. Goodfellow (2017) showed how Adversarial Learning can be used as a technique by training two networks simultaneously, by training them together under a single loss signal, in order to produce better results. This paper looks into this methodology of adversarially training samples for the use case of producing noisy images for attacking image classifiers. Several previous models using adversarial learning have shown to create images that are extremely close to their original training sample (Arjovsky & Bottou, 2017), which only helps us to use this method for creating a Deep Convolutional Generative Adversarial Network based architecture that can create the aforementioned noisy images. Previously tried and tested models exist that use Generative Adversarial networks as their base networks. These include: Deep Convolutional Generative Adversarial Networks (Radford, Metz, & Chintala, 2015) which use a convolutional neural network as it's discriminator and a deconvolutional neural network as a generator for generating images. Radford et al. (Radford, Metz, & Chintala, 2015) use various techniques for their network, including the All-Convolutional Neural Network (Springenberg, Dosovitskiy, Brox, & Riedmiller, 2014) which replaces the commonly used max-pooling layer with another convolutional layer that contains a stride of 2 that provides the same functionality on their dataset, along with the famously

DOI: 10.4018/IJIIT.2020010105

used Batch Normalization(Ioffe & Szegedy, 2015). Earth Mover's distance (Hou, Yu, & Samaras, 2016) is used in Wasserstein GAN(Arjovsky, Chintala, & Bottou, 2017) as the loss function to compare and analyse the difference between the histogram of the original dataset and the one that needs to be generated; and Bayesian GAN(Saatchi & Wilson, 2017) which takes advantage of the Bayesian function to approximate the probability density of the original dataset and the generated samples uses it as the loss function.

The aforementioned architectures produce remarkable results in their own field of image generation from the original dataset. However, these architectures fail to meet the need for adversarial image generation as the requirement for the same is that the image generated by the network must work in tandem with the original image to produce a new noisy image layer that must then be applied to the original dataset to produce a classification of that same original classifier. this intricate process involves an intermediary step for the generation of the noisy image that these legacy networks cannot make. Hence, this paper takes inspiration from Goodfellow et al. (Goodfellow, Shlens, & Szegedy, 2014), which uses a method called Fast Sign Gradient Method. This method trains the loss function of the classifier and that of the noise generator as a combined function using the following equation:

$$x^* = x + \epsilon sign(\nabla_x J(\Theta, x, y)$$

In the equation above, the noise layer is denoted by x^*, the original image is denoted by x, the magnitude of the perturbations is ϵ, is the truth label y and the noise parameter is Θ. The loss function for the same is given by $J\left(\Theta, x, y\right)$.

In another method proposed by Papernot et al. (2015), Jacobian-Based Saliency Map, the author of the paper uses an input image x in a model f that has a classification metric j and a target classification t where the difference between the probability of classification t and j is reduced and all other classification differences are increased using the following equation:

$$S(X,t)[i] = \begin{cases} 0 & \text{if} \frac{\partial F_t(X)}{\partial X_i} < 0 \text{or} \Sigma_{j \neq t} \frac{\partial F_j(X)}{\partial X_i} > 0 \\ (\frac{\partial F_t(X)}{\partial X_t} | \Sigma_{j \neq t} \frac{\partial F_j(X}{\partial X_i}| & otherwise \end{cases}$$

Another example of work related to Adversarial Attacks is the DeepFool framework that is later explained in this paper as well. Moosavi-Dezfooli et al. (Moosavi-Dezfooli, Fawzi, & Frossard, 2016) contribute to the attacking mechanism benchmarking method by creating a perturbation detector that classifies different classifiers for adversarial attack checking by analyzing their robustness for its attacking algorithm and compares their metrics to previously known classifiers along with a score.

The work done in this paper mainly focuses on the fact that a classifier can be fooled into misclassifying the label on any image when overlapped with a noise layer such that the noise layer has been adversarially crafted for that specific misclassification. This paper is organized as follows: section 2 will look at the various legacy and historical methods of misclassifying data based on different algorithms; Section 3 details the experimental setup of the network where each component has been thoroughly explained; And finally, section 4 outlines the results of a comparative study done on the algorithm proposed, along with various other algorithms that have been previously mentioned, wrapping the work with the conclusion of this work along with future prospects.