A Comparative Study of Autoencoders against Adversarial Attacks

I-Ting Chen and Birsen Sirkeci-Mergen
Electrical Engineering
San Jose State University
San Jose, CA 95192
{ i-ting.chen@sjsu.edu, birsen.sirkeci@sjsu.edu }

Abstract—Many machine learning models can be fooled by adversarial examples. The adversarial examples are the subtly perturbed inputs crafted to fool the machine learning models. While the models correctly classify the original inputs, the perturbed adversarial inputs are misclassified to other classes. The perturbations added to the original inputs are usually not easily perceived by human eyes. Therefore the malicious attacks with the adversarial examples are serious security risks because the attacks can remain stealth. This paper compares various denoising autoencoders applied to inputs before the classifiers to filter the fast gradient sign method (FGSM) adversarial perturbations in the inputs. Both the capacity of removing the attacks and performance change on the original un-attacked inputs are analyzed.

Keywords: adversarial examples, autoencoders, machine learning models, security

1 Introduction

Machine learning is a field that applies statistics to the computer programs and endows the computer programs the ability to solve tasks that cannot be handled by other fixed programs. With the statistical characteristic, the programs are capable of learning from the data set fed to them. After learning from the large amounts of data set, the programs can interpret the new data they receive.

One important objective of the machine learning algorithms is generalization. A good machine learning algorithm should be generalized enough to perform well on unseen data sets, based on the assumption that the unseen data sets are identically distributed as the training data set. Generalization is essential to the success of applications of machine learning algorithms. A generalized machine learning algorithm should predict new data sets with high accuracy, thus further instructions from human is not required. Nevertheless, researches have shown that adversarial examples, a collection of unnatural inputs that are slightly perturbed from the original clean ones, breaks the generalization of many machine learning algorithms and causes erroneous predictions [1][4].

The machine learning models are vulnerable to the adversarial attacks. The transferability of adversarial examples among models allows the attackers to create the adversarial examples without the knowledge of the model to be attacked [1][3]. The transferability of the adversarial examples to the physical world [5] could lead to damaging attacks to the machine learning systems, especially to the security-critical areas such as the traffic sign detection systems of autonomous cars [6].

Many works had been done to detect or remove the input adversarial perturbations with unsupervised learning. For example, it was demonstrated that adversarial examples could be detected by monitoring the coefficients of the principal components analysis (PCA) corresponding to the lower variances, or the reconstructed inputs by an auxiliary decoder at the logit units before the softmax layer [7]. The classifiers trained with the inputs being pre-processed with PCA transformation were more robust to adversarial examples, since the adversarial attacks would utilize the principal components with lower variances [8]. An autoencoder trained to reconstruct the adversarial examples to the corresponding clean ones could defend against the adversarial attacks, and a denoising autoencoder trained to clean Gaussian noise from the original inputs could also denoise the adversarial examples without the knowledge of the attacks [9].

This paper examines the application of denoising autoencoders to pre-filter the fast gradient sign method (FGSM) [4] adversarial perturbations out from the inputs. The denoising autoencoder was applied to find the more robust intermediate representation of inputs in the deep network pre-trainings [10] [11]. In our analysis, different autoencoder structures and different MNIST [2] digit subsets were tested, and an additional comparison was made to a set of basic autoencoders. The basic autoencoders have the correspondingly same architectures but were simply trained to compress and decompress the original clean inputs. This comparison gives clue whether the perturbation removal is mainly related to the denoising property of the denoising autoencoders or to the general characteristics of the autoencoders.
2 Performance of Machine Learning Classifiers under FSGM Adversarial Attacks

The FGSM [4] applies a perturbation vector in the input space that was along the direction of the maximal change of the cost function. An input perturbation vector along the direction of the model weight vector can be highly accumulative in the linear models, and the underlying linearity of non-linear models also suggests that the similar perturbations to a linear model could also be damaging to non-linear models. The FGSM offers an analytical form to generate adversarial examples from clean inputs using the formula:

\[ \bar{x} = x + \epsilon \text{sign}(\nabla J(x)) \]

where \( J \) is the cost to train the model.

In this section, we study the performance of two classifiers in handling FGSM adversarial attacks without any prefiltering. In the first classifier model, we utilized the logistic regression classifier [4] with the benchmark FGSM adversarial attack on a MNIST data set including 3s and 7s only. The adversarial perturbation was in the form of \( \epsilon \text{sign}(w) \), where \( \epsilon \) was the adversarial perturbation factor and \( w \) the learned model weight vector. An example of the sign of the weight vector is displayed in Figure 1. The logistic regression classifier accuracy is displayed in Table 1 for different perturbations. Without any attack (\( \epsilon = 0 \)), the logistic regression classifier has comparable accuracy 98.38% on the clean 3s and 7s test set. On the other hand, the accuracy decreases considerably with stronger attacks (larger \( \epsilon \)). For \( \epsilon = 0.25 \), the accuracy is 4.12% on the corresponding clipped adversarial examples.

Table 1: The logistic regression classifier accuracy (using digits 3 and 7 only) on the clean and adversarial examples with different \( \epsilon \)

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>0</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
<th>0.20</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>98.38</td>
<td>90.58</td>
<td>53.04</td>
<td>29.74</td>
<td>13.79</td>
<td>4.12</td>
</tr>
</tbody>
</table>

The second classifier model utilized is a fully-connected neural network with two 100-unit hidden-layers with relu activation function and a 10-unit softmax output layer, abbreviated as FC-100-100-10. We trained the FC-100-100-10 model to classify the MNIST all 10 digits. The accuracy of FC-100-100-10 was 97.90% on clean examples. The Cleverhans python library [12] was then applied for FGSM adversarial examples generation. Table 2 summarizes the classification accuracy on the adversarial examples with different values of \( \epsilon \). Similar to the logistic regression classifier performance, the performance of FC-100-100-10 degrades considerably as the attacks get stronger. For \( \epsilon = 0.25 \), the accuracy decreases to 2.92% on adversarial examples.

Table 2: The FC-100-100-10 accuracy (of 10 digits) on the clean and adversarial examples with different \( \epsilon \)

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>0</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
<th>0.20</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>97.90</td>
<td>88.67</td>
<td>52.20</td>
<td>18.66</td>
<td>6.46</td>
<td>2.92</td>
</tr>
</tbody>
</table>

3 Analysis of Autoencoders for Removal of FSGM Adversarial Attacks

3.1 Considered Autoencoder Architectures

An autoencoder is an artificial neural network generally used for unsupervised learning. In this work, several autoencoder architectures were used to pre-filter the adversarial perturbations of the inputs. All the considered autoencoders had relu activation function at hidden layers and sigmoid at the output layer. The number of training epochs was set as 10 for the logistic regression model and 50 for the FC-100-100-10 model, and the training batch size was set as 20. Two different sets of autoencoders are considered. Set-A of autoencoders have the dimension of the bottleneck layer (middle layer) as 128. Set-B of autoencoders have different bottleneck layer dimensions. The main purpose of using different autoencoders architectures is examining if the adversarial perturbation filtering effect is a general property of the autoencoder instead of a result caused by specific autoencoder structure or encoding dimensions. Table 3 summarizes the tested autoencoder architectures.

<table>
<thead>
<tr>
<th>( \epsilon )</th>
<th>0</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
<th>0.20</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>98.38</td>
<td>90.58</td>
<td>53.04</td>
<td>29.74</td>
<td>13.79</td>
<td>4.12</td>
</tr>
</tbody>
</table>

The second classifier model utilized is a fully-connected neural network with two 100-unit hidden-layers with relu activation function and a 10-unit softmax output layer,
Autoencoder Model | Architecture Description
---|---
Set-A | NN-128 Fully-connected neural network with one 128-unit hidden layer
      | NN-256-128-256 Fully-connected neural network with 256, 128, 256-unit hidden layers
      | NN-512-256-128-256-512 Fully-connected neural network with 512, 256, 128, 256, 512-unit hidden layers
      | CNN-128 Convolution network shown in Figure 2
Set-B | NN-32 Fully-connected neural network with one 32-unit hidden layer
      | NN-128-64-32-64-128 Fully-connected neural network with 128, 64, 32, 64, 128-unit hidden layers
      | CNN-256 Convolution network shown in Figure 3

Table 3: Architectures of considered autoencoders

Each autoencoder architecture was implemented as a basic autoencoder and as a denoising autoencoder. The training inputs and targets of the basic autoencoders were the same images from the clean MNIST training set digits to be classified. The training inputs of the denoising autoencoders were the MNIST training set digits to be classified added with a Gaussian noise with mean 0 and standard deviation 0.5, and the targets were the corresponding clean training set digits.

3.2 Performance of Autoencoders on Adversarial Attacks

Figure 4 and 5 show the logistic regression classifier accuracy with the basic and denoising autoencoder implementations. Both the basic and denoising autoencoders prior to the logistic regression model have improved the classification accuracy compared to no implementation of any autoencoder when perturbation increased. The implementation of basic autoencoders has less accuracy improvement compared to the denoising autoencoders when the adversarial perturbation increased. The best accuracy improvement by the denoising autoencoders is from 4.12% to 93.57% when $\varepsilon = 0.25$.

Figure 6 and 7 show the FC-100-100-10 accuracy with the basic and denoising autoencoder implementations. It also indicates that both basic and denoising autoencoders alleviate the FGSM adversarial attacks when perturbation increases, and the denoising autoencoders outperform the basic autoencoders. The best accuracy is improved by the denoising autoencoder from 2.92% to 75.52% when $\varepsilon = 0.25$.

Figures 8-13 show the clean and adversarial examples beforehand and after the basic and denoising autoencoders. The adversarial perturbation become visible to human eyes when $\varepsilon$ reaches 0.15. Figure 8-13 indicate that both basic and denoising autoencoders have effect on removing the FGSM adversarial attacks.
adversarial perturbations added to the clean inputs. Notice that Figure 8-13 are the results corresponding to the basic and denoising autoencoders with architecture NN-512-256-128-256-512, but the rest of the autoencoders perform similarly.

Figure 8: The pre-filtering result on MNIST 3s of the basic autoencoder with architecture NN-512-256-128-256-512, and the adversarial perturbation factor $\epsilon$ from top to bottom are $0$ (clean), $0.05$, $0.10$, $0.15$, $0.20$, $0.25$. (a) The inputs before being processed by the autoencoder. (b) The outcomes after the inputs were processed by the autoencoder. (c) The differences between (a) and (b), which indicated the portion that were removed by the autoencoder.

Figure 9: The pre-filtering result on MNIST 3s of the denoising autoencoder with architecture NN-512-256-128-256-512, and from top to bottom are the cases of input images with Gaussian noise (mean 0, standard deviation 0.5), $\epsilon = 0$ (clean), $\epsilon = 0.05$, $\epsilon = 0.10$, $\epsilon = 0.15$, $\epsilon = 0.20$, $\epsilon = 0.25$. The first row presents the denoising capability of the denoising autoencoder. (a) The inputs before being processed by the autoencoder. (b) The outcome after the inputs were processed by the autoencoder. (c) The differences between (a) and (b).

Figure 8(c) and Figure 9(c) show the differences between the inputs and outputs of the basic and denoising autoencoders respectively, which represents the portion that was filtered out by the autoencoders. The FGSM adversarial perturbations, which are in the form of the sign of logistic regression model weight in Figure 1, are included in the portion that was filtered out by both the basic and denoising autoencoders. Notice that similar results occurred for Figure 10(c) and Figure 11(c) corresponding to the digit 7s of logistic regression classifier, and Figure 12(c) and Figure 13(c) for the FC-100-100-10 network.

The outperformance of the denoising autoencoders over the basic ones might be related to the more authentic recovery of the digits. For example, in the Figure 8(b) and Figure 9(b), the recovered digits of the adversarial examples by the basic autoencoders are more irregular when the adversarial perturbations were stronger, even though they could remove the FGSM perturbations. On the contrary, the recovered digits by the denoising autoencoders were more authentic and less affected by the adversarial perturbations.

The result indicated that besides the removal of the FGSM adversarial perturbations, the robust input recovery capability
of the denoising autoencoders under the adversarial perturbations influence was also critical. Since the denoising autoencoders could learn more robust representations from the noised inputs [10], it might imply the more stable recovery ability of the denoising autoencoders, which was important to the application of the autoencoders to remove the FGSM adversarial perturbations.

**Figure 12:** The pre-filtering result on MNIST 10 digits of the basic autoencoder with architecture NN-512-256-128-256-512, and the adversarial perturbation factor ϵ from top to bottom are 0 (clean), 0.05, 0.10, 0.15, 0.20, 0.25. (a) The inputs before being processed by the autoencoder. (b) The outcome when the inputs were processed by the autoencoder. (c) The differences between (a) and (b).

The denoising autoencoder architectures, different encoding dimensions at bottleneck layer, and different MNIST digit subsets. All these configurations demonstrated the effective removal of the adversarial perturbations from the inputs. The best results of classification accuracy improvement were from 4.12% to 93.57% for the logistic regression classifier on digit 3s and 7s, and from 2.92% to 75.52% for the neural network classifier on the 10 digits. Although new adversarial examples of lower distortion with respect to the stacked autoencoder and classifier could be generated [9], the pre-filtering denoising autoencoder might still be a feasible approach against the black-box FGSM adversarial attacks.

Our work also studied the components that made the denoising autoencoder a feasible module to remove the adversarial perturbations. In our comparison experiment, the phenomenon of adversarial perturbations filtering happened both at the basic and denoising autoencoders. The capability of robustly recovering the inputs from the adversarial perturbations critically determined the outperformance of the denoising autoencoders over the basic ones on defending the FGSM adversarial attacks.

5 References


