Increasing robustness of deep neural network models against adversarial attacks

Pankaj D Bhagwat\(^1\) and Pratibha P Shingare\(^2\)

\(^1\)Postgraduate Student, Department of Electronics and Telecommunication, College of Engineering Pune 411005, India
\(^2\)Assistant Professor, Department of Electronics and Telecommunication, College of Engineering Pune 411005, India

E-mail: pps.extc@coe.p.ac.in

Abstract. In Autonomous driving detecting correct object is important, further studies proved that by adding small pattern above object can also lead intentional fooling of network. Small intentional changes in the input can significantly distort output of a deep neural network model. This makes the machine learned model vulnerable to these small changes in images. Hence, these models have wide scope of failure. If we are able to tackle these intentional attacks it will help to make system more robust.

In this project, we have combined multiple techniques used for defending against adversarial attacks. First technique is Adversarial training which include modifying training dataset, second technique is pre-processing input data before applying it to deep learned model and Third technique randomly selects image pre-processing technique. Third method is aimed to distract attacker who know the method used in pre-processing by randomly selecting from multiple methods in image pre-processing.

We will measure robustness of deep learned model in terms of Accuracy on Designed system and previous deep learned model. Test samples and adversarial images generated from dataset will be used for testing on deep learned model.

Among all methods which we have combined, Adversarial training proved best method to defend against white box attack. If we would have used strong defences in Random selection of image transformation then the system could have performed much better. However, Random selection have done its work of confusing the attacker by selecting random transformations.

1. Introduction

Many machine learning-based applications are life crucial e.g. COVID detection, tumour detection, autonomous driving, spam mail, surveillance etc. There are many applications which proved great success in their domain however, current studies show machine learning model are weak against properly and well-made inputs. These techniques are able to easily fool the state-of-the-art machine learning models by adding mathematically calculated noise to image which is imperceptible to human eyes.

Adversarial examples are images with noise added, these are designed to fool neural network resulting in incorrect output from the network. Ian Goodfellow et al. shown in his experiment that it was possible to fool network by generating mathematically calculated noise, that method to generate such adversarial image was named Fast Gradient Sign Method [1]. In that experiment
when the image of panda was given to network it showed 57.7% confidence on "panda" class. After attacking that image network classified image as, "gibbon" with 99.3% confidence [1][2].

In case of safety critical application such as Autonomous driving misclassification could lead to lethal accident. Using these methods, attacker could change the “Stop sign” into “Speed limit sign” using targeted attack [3]. And for detection of COVID-19 by attacking image of patient’s X-Ray scan, false report can be generated to claim insurance of any other benefits. Therefore, this type of attacks can be threat to upcoming machine learning based application as in future everything is going to be automated and machines will be able to take their own decisions. Therefore, we need to deal with this type of problem before the system advances to the next stage.

2. Attack Scenarios
Based on the requirements attacker decide the which attributes to add or remove to generate adversarial example then apply specific attack [2]. Among the Scenarios discussed below attacker can use any one of them or their combinations.

2.1. Targeted Attacks
When the attacker targets a specific class then this type of attack is called targeted attack. For example, in case of COVID detection from X-ray images, there are only two class either True or False. By attacking X-ray image, we can modify the results on well trained model for COVID detection. If we want to target the Face recognition system then we can attacker tries to disguise his/her face as authorized user [2].

2.2. Non-Targeted Attacks
In this type the attacker will misclassify the image in any other class [4]. The label does not have to be of a specific class and can be anything among the class on which machine is trained. This type of attacks is easier to generate, as it has many different options of class to choose. Advantage of this attack is that it can be given to any trained model, perturbations added are such that it generalizes and leads to misclassification of image.

2.3. Black-Box Attacks
In this type of attack, Attacker does not have complete access to the trained model or its parameter. For him the model is just as black-box. As the attacker does not know about internal trainable parameter, weights etc. However, he can still observe output probabilities of class for different input test data [4].

2.4. White-Box Attacks
In this type the attacker has complete access to the model and its parameters e.g. weights, architecture, layers etc. As pre-trained models are available so, there is high possibility of attacker to use this attack. Having full knowledge of model this is the most effective type of attack specifically application which use transfer learning.

3. Methods for Generating Adversarial Examples
There are many other techniques to generate adversarial examples. Szegedy et al. first observed that DNN models are vulnerable to adversarial perturbation [1]. In Adversarial examples perturbation are added in such a low proportion that they look close original images and are undetectable to naked eyes. And even if the strength of an attack is increased for human it will
look like corrupted image, because of compression or external noise. These Perturbations can be measured by using L_p norm.

\[ \|x\|_p = \left( \sum_{i=0}^{n} \|x_i\|^p \right)^{\frac{1}{p}} \]  

L0, L2 and L∞ are three Lp metrics used [5]. In our project to generate these adversarial images, we will be making use of IBM Adversarial Robustness toolbox which is based on python programming language for generating various attacks.

3.1. Fast Gradient Sign Method: FGSM (L∞, Untargeted)
Goodfellow et al. designed a fast method for generating adversarial examples called fast gradient sign method (FGSM) [1]. This method used only one step gradient update to generate fake images, it is based on L∞ norm [6]. This attack can be mathematically expressed as

\[ \eta = \epsilon \text{sign} \left( \nabla_x J_{\Theta}(x,l) \right) \]  

3.2. Deepfool (L2, Untargeted)
Moosavi-Dezfooli et al. used the L2 norm-based perturbation. In this method author tried to find the minimum perturbation which can be added so as to misclassify the image. Therefore, gradient change in original image is kept minimum [7][8].

3.3. Carlini/Wagner Attacks (L2, L∞ and L0, Targeted)
Carlini and Wagner designed a targeted attack, which later became one of the strongest techniques. This attack was able to break through defensive distillation method used for defending against attack. They designed three types based on L norms of attacks using L2, L∞, and L0 norms [5].

4. Methods For Defending Against Adversarial Examples
Adversarial images generated by techniques look like corrupted image to a naked human eye. Many authors have tried to implement image pre-processing technique as defence. If we are able to remove these noises from image before giving it to neural network we can defend against attacks. Therefore, variety of image pre-processing and denoising techniques were tested. Results show that applying image processing technique corrupted the image even more leading to loss in accuracy [4]. Another way was to retrain the existing model with adversarial images it proved effective only if the same attack technique on which machine was trained is used. Researchers are still working on finding techniques which will be able to defend against all kind of scenario. Both these techniques are discussed in detail below:

4.1. Adversarial Training
Increasing size of dataset by adding adversarial images generated from same dataset proves to be best technique to defend against adversarial images [1]. Goodfellow et al. experimented and concluded that adversarial training improved the robustness of model it proved effective only if the same attack technique on which machine was trained is used. However, Adversarial examples also generalizes to different model only difference is that its effect is more or less [1]. Adversarial training also helped increasing the accuracy on test data.
4.2. Input Transformation
These types of defences are recent and unexplored in this field. Most of the image transformation technique use image compression (e.g. JPEG, Wavelet etc.) and image processing (e.g. Smoothening, sharpening etc.) It was found that after image transformation adversarial examples will not affect prediction or relatively low amount. The biggest problem with transformation-based technique is that they will degrade the quality of clean (not attacked) image also leading to decrease in accuracy [8][9].

Image transformation techniques which we have selected in this project are shown below:

(i) JPEG Compression - This is image compression technique which uses Discrete cosine transform. This method can remove high frequency noise in image which is same as blurring the image. Blurring help in removing additional noise in image [10].

(ii) Spatial smoothening - Also known as blurring is widely used removing noise in an image. It uses values of neighbouring pixel to smooth each pixel [9].

(iii) Feature squeezing - This technique reduces the colour bit depth combined with spatial smoothening. This simple yet effective technique to defend against adversarial attacks [9].

5. Analysis
In the process of designing any system selection of proper procedure and efficient technique play a very important role for getting desired outcomes. In chapter we are going to discuss about conclusion, Analysis and reasons for selection of different methods used in this project.

5.1. Why Deepfool attack, not FGSM for Adversarial training?
FGSM Attack is the simplest form of attack and most of the time this attacked data is used for Adversarial training. In paper by Moosavi-Dezfooli et al. they found that training the model with overly perturbed image (e.g. training with FGSM) reduces robustness of model for adversarial images [7]. In Figure 1 we can see that by increasing the value of norm by alpha (i.e. 1, 2, 3 ...) robustness of model decreases. Therefore, we have to select the attack which has minimum norm and Deepfool have minimum norm and was also effective in misclassifying.

![Figure 1: Graph shows that training with FGSM reduces the robustness of the model and Deepfool increases robustness [7].](image)

Table 1 Shows the test results carried by Moosavi-Dezfooli et al., It can be seen that Deepfool showed the least error for Both models. Training with Deepfool increases accuracy for adversarial images and perturbation such as Deepfool are most likely to occur in real life scenarios [7]. So, Training with Deepfool will give use additional advantage.
Table 1: The test error of networks after adversarial training on adversarial examples [7].

| Classifier        | Deepfool | FGSM | Clean |
|-------------------|----------|------|-------|
| NIN (CIFAR-10)    | 11.2%    | 21.2%| 11.5% |
| LeNet (CIFAR-10)  | 20%      | 28.6%| 22.6% |

5.2. Black box attack on DL model
We have selected CIFAR 10 dataset for training and testing purpose. It includes 50,000 training images and 10,000 test datasets with 10 classes all of them are 32 x 32 coloured images. As we have to generate adversarial images which need classifier as an input parameter to generate adversary. Therefore, making use to large model would have been computationally heavy and time consuming. So, we have designed our own model with 4 convolution layer and 3 fully connected layer with max pooling and dropout. As the model will remain same for every test, we will have to check results relatively for different methods.

If we use another model rather than on which adversarial trained images were generated and used for adversarial training. Will our deep learned model will be attack work? Yes, we experimented on models and accuracy obtained by designed model on CIFAR-10 test data set was 77.29%. Then we have generated adversarial images using Deepfool attack and saved the generated images. Then these Adversarial images is to be given to the model. For this model we obtained an accuracy of 54%. As these adversarial images were generated on a different model then also test images were able to reduce the accuracy of model. This shows that adversarial perturbation generalizes across different model. If we would have used same model to generate adversary then the accuracy could have dropped even further. Then we trained model on adversarial and training data. Adding adversarial images also expand the training set with the adversarial samples which help in improving accuracy. After 10 Epoch model gave 95.37% accuracy on training dataset which was reduced to 16.4% after adversarial attack as it was white box attack. Then after adversarial training accuracy was improved, even after attack it was 47.6% on adversarially trained model [11].

6. Methodology
As Discussed earlier there are various methods for increasing robustness of a model. So as to keep it these techniques simple, let us divide it in three parts as below:

(i) Method 1 - Adversarial Training (Dealing with internal structure of model such as weights, gradients, layers etc.).

(ii) Method 2 - Image Transformation (Dealing with image data before giving it to neural network i.e. using image processing techniques).

(iii) Method 3 - Random Selection of Image Transformation Technique (Defending against white box attack).

6.1. Method 1 - Adversarial Training
Increasing size of dataset by adding adversarial images generated from same dataset proves to be best technique to defend against adversarial images [1]. So, training with any one of adversary attack is sufficient however we can even include multiple attack in different ratio. This method was proved to be effective against white box attack. Adversarial training also performs well against black box attack however, in case of white box attack it performed better. Adversarial training is properly explained in Figure 2.
6.2. Method 2 - Image Transformation

After applying image transformation adversarial examples can be converted into clean image. In this module we will pre-process the input image before giving it to the DNN. It is properly explained in Figure 3.

6.3. Method 3 - Random Selection of Image Transformation Technique

Image transformation techniques was proved ineffective against white box attack. Edward Raff et al. in their experiment combined a number of weak defences and used them in randomized fashion for pre-processing to build strong defence [3][12]. In this last module, instead of giving transformed image to DNN we will randomly select any one of the image transformation technique from multiple transformation available and send it to DNN and classifier. This minor
change introduced in software will defend against attacks which target any weak point to specific image transformation technique. It is properly explained in Figure 4.

After applying Random Transform technique, it is expected that our software will perform well against attacks. And method 3 also represent complete structure of our project and design.

![Flow explaining Random selection of image transformation before forwarding it to DNN.](image)

**Figure 4**: Flow explaining Random selection of image transformation before forwarding it to DNN.

### 7. Results
We are going to assess the robustness of DNN model in terms of accuracy. By applying adversarial images to our model and normal model we can find out which one is performing better.

As we have discussed our complete project will be divided into 3 methods. At each stage module new feature is added to the deep learned model so that at the end to method 3 we will
have our completed Robust deep learned model ready. So, let us start with method 1.

7.1. Adversarial Training

As discussed in above section 4.4, we are going to select Deepfool attack for generating fake images. Then train our CNN model with fake and training data. However, the question arises what should be the percent to Adversarial images to be included with the training data?

We experimented by training CNN with 10% of Adversarial images and 50% of Adversarial images. We found that model trained with 10% Adversarial images performed better for test set as new more images are added in training data however, performed poorly for Adversarial attacks. Model trained with 50% Adversarial images performed better for Adversarial images and poor but, Acceptable for test data. As our target is to defend against adversarial attack, we have selected model trained with 50% adversarial images. Results of test carried out on 2 models are as shown in Table no. 2.

Table 2: Results show that Accuracy of CNN trained with 50% adversarial images performed better for all attack.

| Percent of Adversarial images | Test data | Deepfool | FGSM (eps = 0.3) | C&W |
|------------------------------|-----------|----------|------------------|-----|
| 0% (Without Adversarial Training) | 80.22% | 41.16% | 17.60% | 76.2% |
| 10% | 80.81% | 46.38% | 18.10% | 77.5% |
| 50% | 76.86% | 59.78% | 29.94% | 75.78% |

7.2. Images transformation

As discussed in above section, we have selected 3 images transformation technique, before combining all of them we studied the effect of adversarial images on this technique individually. Results of the test are shown in below Table no. 3.

Table 3: Results showing Accuracy of CNN Adversarial trained model for different defence technique.

| Percent of Adversarial images | Test data | Deepfool | FGSM (eps = 0.3) | C&W |
|------------------------------|-----------|----------|------------------|-----|
| Feature Squeezing | 76.66% | **59.12%** | 28.94% | **75.2%** |
| Spatial Smoothening | 76.32% | 46.16% | 26.58% | 72.46% |
| JPEG Compression | 79.32% | **59.12%** | **30.38%** | NA |
| No Defence | 80.22% | 59.22% | 29.94% | 76.2% |

In Figure 5, we can observe that Feature Squeezing is performing good for Deepfool attack and C&W attack. And JPEG Compression is performing good against FGSM attack as shown in Figure 6.

7.3. Random Selection of Image Transformation Technique

We concluded from tests in Section 7.2 is that, if we know which technique attack is going to use, we can counter it e.g. If Attacker is using FGSM then we will use JPEG compression as image transformation technique. And this is where problem arises, if it is a Whitebox attack (Attacker knows which input transformation technique we are using) then, attacker will choose any other attack method which can invade JPEG compression.
And this is where our third technique comes into picture. If we keep on Shuffling Image transformation technique then, Attacker will not be able to decide which attack to be used [12].

Table 4: Results showing Accuracy of CNN Adversarial trained model for different attack using Random selection of transformation technique. Note - Accuracy obtained on Test data for our model is 76.86%.

| Test no. | Deepfool | FGSM (eps = 0.3) | C&W  |
|----------|-----------|------------------|------|
| 1.       | 56.8%     | 28.78%           | 73.9%|
| 2.       | 57.8%     | 28.34%           | 73.6%|
| 3.       | 57.33%    | 28.56%           | 74.18%|
| 4.       | 57.36%    | 29.45%           | 73.92%|
| 5.       | 57.14%    | 28.66%           | 73.7%|
| **Average Accuracy** | **57.286%** | **28.75%** | **73.86%** |

This module is the final stage of implementation, as discussed. we have selected 3 technique to randomly choose from. These defence techniques are:

(i) Feature Squeezing
(ii) Spatial Smoothening
(iii) JPEG Compression
Above mentioned technique are weak defences and by combining them we will try to form strong defence as explained by Edward Raff [3]. And for the same concept if we use strong defence. Then, it would prove to be a strong defence against Adversary. As this method randomly select any defence technique, we have will do 5 test iteration and average them to get approximate accuracy. Final average results in term of accuracy is as shown in below Table no.4.

8. Conclusions and Future Scope
In this project, we tried implement defence technique in three ways. So, we divided conclusion in 3 parts:

(i) Adversarial training - We were successful in applying this technique and got better Accuracy for adversarial images. However, there is still a lot can be done, like using fake images generated by different attacks for adversarial training in different ratio.

(ii) Image Transformation - Results were acceptable as we have use simple and weak defence for initial stage. It can be improved further by using complex defences e.g. Pixel deflection [8], Adversarial inverse [13], Defensive distillation [2] etc.

(iii) Random Selection of Image Transformation - If the user knows about our defence he can counter and bypass our defence. However, if we continuously keep on randomly shuffling our defences then, the attacker will surely get confused. Analysing the results, we found that, Accuracy of this method was less than that of adversarial training by 1-2%. This problem can be solved by using strong defences and increasing number of defences.

Even though there was drop in Accuracy by few 1-2% after image Transformation than that of Adversarial Training. However, we were able to defend against white-box attack by confusing the attacker by introducing Method 3 (Random Selection of Image Transformation).

9. References
[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, pp. 1–11, 2015.
[2] X. Yuan, P. He, Q. Zhu, and X. Li, “Adversarial Examples: Attacks and Defenses for Deep Learning,” IEEE transactions on neural networks and learning systems, vol. 30, no. 9, pp. 2805–2824, 2019.
[3] E. Raff, J. Sylvester, S. Forsyth, and M. McLean, “Barrage of Random Transforms for Adversarially Robust Defense,” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6528–6537, 2019.
[4] A. Kurakin, I. Goodfellow, S. Bengio, Y. Dong, F. Liao, M. Liang, T. Pang, J. Zhu, X. Hu, C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille, S. Huang, Y. Zhao, Y. Zhao, Z. Han, J. Long, Y. Berdibekov, T. Akiba, S. Tokui, and M. Abe, “Adversarial Attacks and Defences Competition,” pp. 195–231, 2018.
[5] N. Carlini and D. Wagner, “Towards Evaluating the Robustness of Neural Networks,” Proceedings - IEEE Symposium on Security and Privacy, pp. 39–57, 2017.
[6] J. S. Goodfellow, Ian J. Shlens, and Christian, “Adversarial example using fgsm.”
[7] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-Decem, pp. 2574–2582, 2016.
[8] A. Prakash, N. Moran, S. Garber, A. Dilillo, and J. Storer, “Deflecting Adversarial Attacks with Pixel Deflection,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 8571–8580, 2018.
[9] W. Xu, D. Evans, and Y. Qi, “Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks,” no. February, 2018.
[10] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, “Keeping the Bad Guys Out: Protecting and Vaccinating Deep Learning with JPEG Compression,” pp. 1–15, 2017.
[11] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016, pp. 372–387, 2016.
[12] S. Forsyth, “Combating adversarial attacks with a barrage of random transforms (bart).”
[13] N. Worzyk and O. Kramer, “Adversarials -1: Defending by Attacking,” Proceedings of the International Joint Conference on Neural Networks, vol. 2018-July, pp. 1–8, 2018.