IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.107100

Defence of CNN against the Adversarial Attacks

Sagar Biswari[1], A. Sai Praneeth[2], Prof. Merin Meleet[3], Prof. Sharadadevi[4], S Kaganurmath[5]

Department of Information Science & Engineering, R.V. College Of Engineering, Bangalore, India[1-5]

Abstract—The rapid advancement of artificial intelligence (AI) and deep learning (DL) approaches, it's more important than ever to assure the security and robustness of the algorithms being used. The vulnerability of DL algorithms to hostile samples has recently been extensively recognised as a security concern. The faked samples can cause numerous DL model misbehaviors while being seen as harmless by humans. Adversarial attacks have been successfully implemented in real-world circumstances, demonstrating their utility. As a result, adversarial attack and defensive strategies have gotten a lot of interest from the machine learning and security sectors, and turned into a hot area of study.The theoretical foundations, methods, and applications of adversarial attack strategies are originally introduced in this study. Following that, we examine a number of outstanding topics and concerns in the hopes of spurring more study in this important area.

Keywords—Machine learning, Deep neural network, FGSM, Adversarial attack, Adversarial defense,CNN

I. INTRODUCTION

The use of deep learning (DL) to handle a range of machine learning (ML) problems, including image classification,[1] natural languages processing[2] and game theory,[3] have been promoted for a trillion-fold boost in computer capacity. However, the research community has found a significant security vulnerability to existing DL algorithms: Adversaries can readily mislead DL models without being noticed by humans by disturbing innocuous samples. Confidence undetectable perturbations for human vision/audition are enough to trigger the model to produce a misprediction.

The vulnerability in this case happens with anything or anyone which uses the image classification as a tool in classification and differentiating between the different images as per the use case of their own. Be it from the medical division to the automation of the machines or vehicles, the need of image classification is highly required in the scenarios.

Taking into account the implementation , the classification works very fine until the point where the m machines are not confused or are made confused in order to gain some personal benefits or to make harm to the others who are in the same field or competition.Thus in order to safeguard the interests , some defences are to be there to cancel out the harms or damages caused or in the process.

This phenomenon is regarded to be an important barrier to the broad use of DL models in production, called the adverse sample. This open topic has been examined with substantial research efforts.In this work, the adversarial assaults and responses representing state-of-the art efforts in this field are investigated and summarised. We next remark on the efficacy of the assault and defence tactics given and debate them.

II. LITERATURE SURVEY

With the increase in the adversarial attacks on the deep neural networks various studies have been made to defend against these attacks and retain the accuracy and precision of the deep neural networks , one of the authors discovered that the adversarial assault can alter the accuracy of lung nodule malignancy prediction.

To mitigate the effects of an adversarial attack, an ensemble-based defence approach was devised. A CNN ensemble with several initializations was used[1].Another author of the paper discusses numerous security flaws in DL algorithms that can be exploited by adversarial perturbations.

The author used relevant adversarial cases to test a variety of COVID-19 diagnostic techniques that depend on DL algorithms in this work.[2],similarly author talks about how deep learning systems are vulnerable to adversarial assaults, according to the research, stressing the critical need for defensive approaches that can identify and neutralise these attacks before they happen.

UnMask, an adversarial detection and protection system based on robust feature alignment, was designed to counter these adversarial attacks.UnMask's main goal is to safeguard these models by ensuring that each image's projected class ("bird") has the required robust characteristics (e.g., beak, wings, eyes).The model might be under assault if a picture is categorised as "bird," yet the extracted characteristics are wheel, saddle, and frame.[3]

The paper we surveyed went through biased training data or vulnerable underlying models, imperceptible modifications on inputs may result in devastating consequences. Although existing approaches show promise in protecting against such malicious assaults, most of them can only cope with a limited number of attack types, making the deployment of large-scale IIoT devices a difficult task. The authors propose a federated defence strategy to solve this issue, which may aggregate defensive knowledge against hostile instances from many sources. [4]

III. DESIGN AND IMPLEMENTATION

1. Dataset

The Modified National Institute of Standards and Technology dataset is an abbreviation for Modified National Institute of Standards and Technology dataset.

It's a collection of 60,000 tiny square grayscale pictures of handwritten single numerals ranging from 0 to 9.The goal is to sort a handwritten digit picture into one of ten groups that represent integer values ranging from 0 to 9, inclusively.It is a commonly used and well-understood dataset that has been "solved" for the most part.

Deep learning convolutional neural networks are the best-performing models, with a classification accuracy of over 99 percent and an error rate of between 0.4 percent and 0.2 percent on the hold out test dataset.Despite the fact that the MNIST dataset has been successfully solved, it may serve as a helpful starting point for creating and implementing a technique for tackling image classification tasks using convolutional neural networks.

We can create a new model from scratch rather than reading the literature on well-performing models on the dataset.We can utilise the dataset because it already contains a well-defined train and test dataset.We may further divide the training set into a train and validation dataset in order to estimate a model's performance for a specific training run.Over the course of each run, performance on the train and validation datasets may be displayed to offer learning curves and insight into how well a model is performing.

2. Adversarial Attacks

We provide a few typical adversarial attack techniques and approaches in this section.These approaches are aimed at image classification DL models, although they may be used on any DL model.

A. L-BFGS Method

Adversarial examples are inputs that, according to a distance metric (e.g. L2 distance a.k.a. Euclidean distance or mean squared error), appear extremely similar to their actual counterparts, but lead a classifier to misclassify them.

The Broyden-Fletcher-Goldfarb-Shanno(L-BFGS) [5] algorithm is a non-linear gradient based numerical optimization approach with limited memory.The L-BFGS approach attempts to solve this optimization issue, where r is the perturbation factor.

B. Fast Gradient Sign Method

FGSM looks for the direction in which a target machine learning model's loss function grows the fastest.Because back propagation requires knowledge of the model's design and parameters, FGSM is an example of a whitebox assault.Once the gradient has been computed, a tiny quantity of input can be pushed towards the adversarial gradient.

Formulation of the FGSM.

Here, x' represents the adversarial example, which should resemble x when it is small, and y represents the model's output.

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y))$$

Eqn.1 Depicting the equation to calculate gradient

J indicates the model's loss function, and is a tiny constant that determines the amount of the perturbation.There's no assurance that the created hostile instances will actually be adversarial.

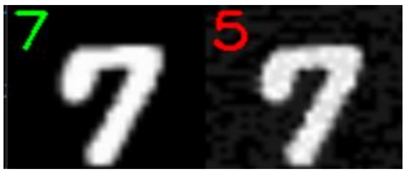

Fig 1. Fgsm attack on the mnist dataset

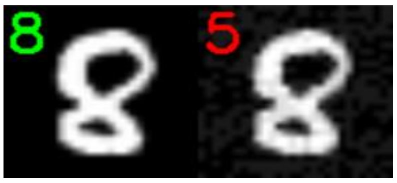

Fig 2. Fgsm attack on the mnist dataset

The fact that the gradients are taken with regard to the input picture is a fascinating feature.Because the goal is to generate a picture that maximises the loss, this is done.Finding how much each pixel in the image contributes to the loss value and adding a perturbation correspondingly is one way to do this.

This works quickly because utilising the chain rule and determining the needed gradients makes it simple to figure out how each input pixel contributes to the loss.As a result, the gradients are measured in relation to the picture.Furthermore, because the model is no longer being trained, the model parameters are unchanged.The only objective is to deceive a model that has already been trained.

C. Basic Iterative Method

BIM is an extension of FGSM in which the same step size is used numerous times.BIM is also known as the Iterative FGSM in some studies (I-FGSM).

$$\boldsymbol{X}_0^{adv} = \boldsymbol{X}, \quad \boldsymbol{X}_{N+1}^{adv} = Clip_{X,\epsilon}\Big\{\boldsymbol{X}_N^{adv} + \alpha\,\mathrm{sign}\big(\nabla_X J(\boldsymbol{X}_N^{adv}, y_{true})\big)\Big\}$$

Eqn.2 Basic Iterative Method

J represents the model's loss function, N denotes the number of iterations, and is a constant that determines the size of the perturbations in the BIM formulation .The Clip function guarantees that the created adversarial example remains within the ball's (i.e. [x-, x+]) and input space's ranges (i.e. [0, 255] for pixel values)

3. Defence Against Attacks

Training your model on these sorts of pictures is one of the simplest methods to guard against adversarial assaults.For example, if we're concerned about malicious users using FGSM assaults on our model, we may "inoculate" our neural network by training it on our own FSGM pictures.This form of adversarial inoculation is often administered by either:

Using a given dataset to train our model, generate a collection of adversarial pictures, and then fine-tune the model using the adversarial imagesCreating mixed batches of both training and adversarial pictures, then fine-tuning our neural network on these mixed bunches

The first technique is easier to use and involves fewer calculations (since we need to generate only one set of adversarial images).On the downside, because we only fine-tune the model on hostile instances at the end of training, this technique is less resilient.

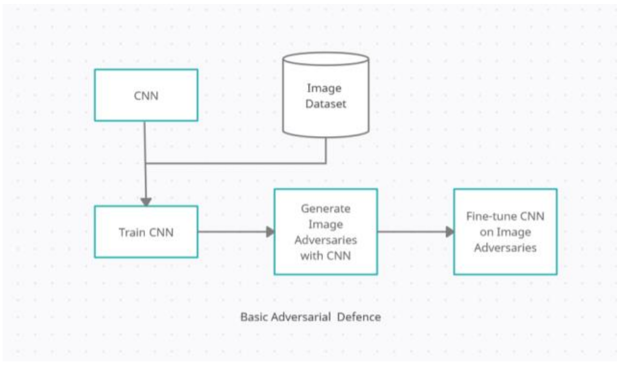
CNN
Image Dataset
Train CNN
Generate Image Adversaries with CNN
Fine-tune CNN on Image Adversaries
Basic Adversarial Defence

Fig 3. Basic Adversarial Defence

The second technique is considerably more difficult and time-consuming to implement.For each batch when the network is trained, we must utilise the model to create adversarial pictures.The benefit of the second technique is that the model is more resilient because it sees both original training and adversarial pictures after each batch update during training.

Furthermore, for each batch, the model is utilised to create the adversarial pictures.As the model improves at deceiving itself, it will be able to learn from its failures, culminating in a model that is more resistant to adversarial attacks.We may change the batch creation procedure instead of fine-tuning the network on a collection of hostile samples.

We train neural networks in batches of data when we do so.Each batch is a subset of the training data, and it's usually in powers of two in size (8, 16, 32, 64, 128, etc.).We conduct a forward pass of the network for each batch, compute the loss, backpropagate, and then update the network's weights.

This is essentially any neural network's typical training process.The model has improved by two factors after each batch update.First, in the training data, the model should have acquired more discriminating patterns.Second, the model has learnt to protect itself against hostile instances created by the model.

IJARCCE
ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 7, July 2021
DOI 10.17148/IJARCCE.2021.107100

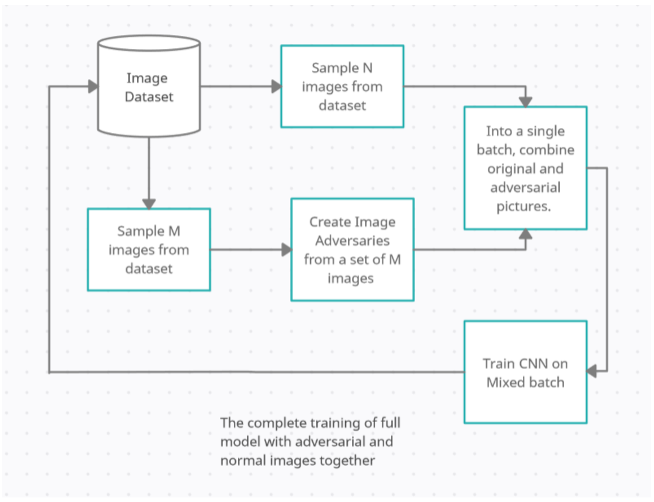
Image Dataset
Sample N images from dataset
Into a single batch, combine original and adversarial pictures.
Sample M images from dataset
Create Image Adversaries from a set of M images
Train CNN on Mixed batch
The complete training of full model with adversarial and normal images together

Fig 4. Novel Method to train and defend against attacks

The model learns to defend itself against adversarial attacks throughout the course of a training procedure (tens to hundreds of epochs with tens of thousands to hundreds of thousands of batch updates). Although this strategy is more complicated than the simple fine-tuning approach, the advantages much exceed the drawbacks.

Tables

Placed figure below shows the different use cases for the model to be trained and tested. When the model is trained traditionally by taking into account the normal training dataset , it works quite well onto the normal image testing. But when moving onto the testing with the adversarial images the accuracy drops significantly. So this paper proposes the training method in which the training dataset is generated by mixing the batches which ramp up the accuracy in case of adversarial testing as well.

TABLE I. ACCURACY/LOSSES FOR TESTING

Condition	**Type of Images**	**accuracy**	**loss**
Normal Training Dataset	Normal Testing Images	0.9895	0.0419
Normal Training Dataset	Adversarial Testing Images	0.0087	15.7830
Dynamic Mixed Dataset	Normal Testing Images	0.9911	0.0273
Dynamic Mixed Dataset	Adversarial Testing Images	0.9753	0.0815

TABLE II. HYPERPARAMETERS FOR CNN CLASSIFIER

Epochs	50
Batch_size	[364,227]
Number of transform layers	2
Model_confidence	softmax

The CNN classifier has a lot of parameters that may be tweaked to fine-tune the model. The number of epochs, batch size, number of transform layers, and activation function had the most influence. The parameters provided in TABLE II

DOI 10.17148/IJARCCE.2021.107100

were picked after numerous trials since they performed the best for our test tales. These criteria were set to prevent underfitting and overfitting.

IV. RESULTS AND ANALYSIS

We trained our CNN for 50 epochs on the MNIST dataset. On the training set, we got 99.70 percent accuracy, and on the testing set, we got 98.92 percent accuracy, indicating that our CNN is good at making digit predictions.

When we produce a collection of 10,000 hostile pictures and ask CNN to identify them, our "high accuracy" model is grossly insufficient and incorrect.Our CNN achieved 98.92 percent accuracy on our testing set after fine-tuning it on a collection of 10,000 hostile pictures.The testing set's accuracy has fallen by about 0.5 percent, but the good news is that we're now identifying our hostile pictures with 99 percent accuracy, suggesting that:

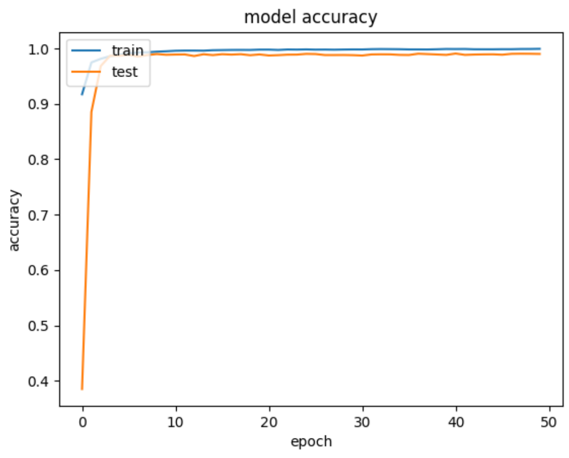

Fig 5. Graph depicting accuracy with normal training

On the original, unaltered pictures from the MNIST dataset, our model can make accurate predictions. On the produced adversarial pictures, we can also make correct predictions (indicating that we've effectively guarded against them). During the training phase, a better technique of mixing and integrating hostile pictures with the original images.

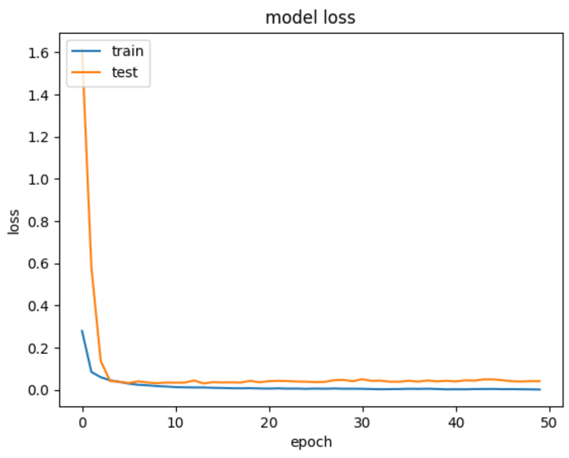

Fig 6. Graph depicting loss with normal training

It is assured that the quantitative loss measure for a particular epoch will be derived using the loss function across all data items. As a result, only a subset of the full dataset can be determined. Because the model produces its own hostile pictures in each batch, rather than depending on a single round of fine-tuning after training, the outcome is a more robust model capable of fighting against adversarial attacks.

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 7, July 2021

DOI 10.17148/IJARCCE.2021.107100

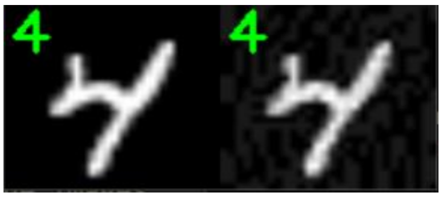

Fig 7. Results from network trained on mixed batch of adversarial images and original images

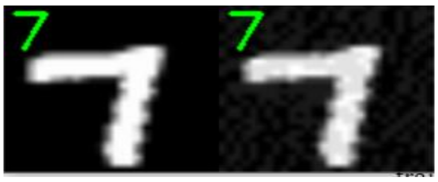

Fig 8. Images were correctly labeled after the model trained on both adversarial and original images with high accuracy.

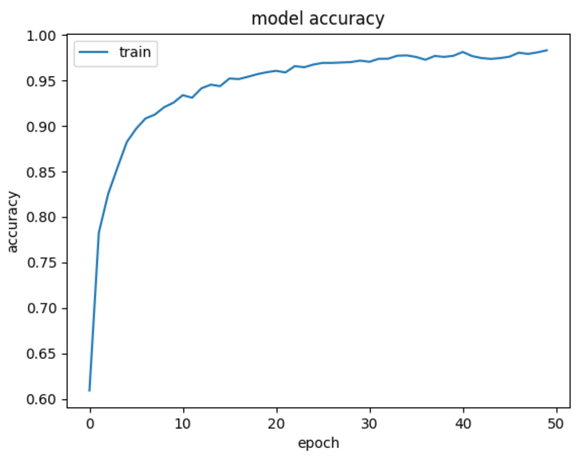

Fig 5. Graph depicting accuracy with mixed training

After training against both normal pictures and adversarial images, the model's accuracy is shown against each epoch across a range of 50 epochs. After training against both normal pictures and adversarial images to defend and forecast properly for any sort of image, here is the loss plot for each epoch across a range of 50 epochs for the model after training.

In order to fine-tune the CNN classifier, it is necessary to change a number of its parameters. There was a significant influence on the number of epochs, batch size, number of transform layers, and activation function of the transformation. These criteria were selected to prevent the issue of either under- or over-fitting the garment.

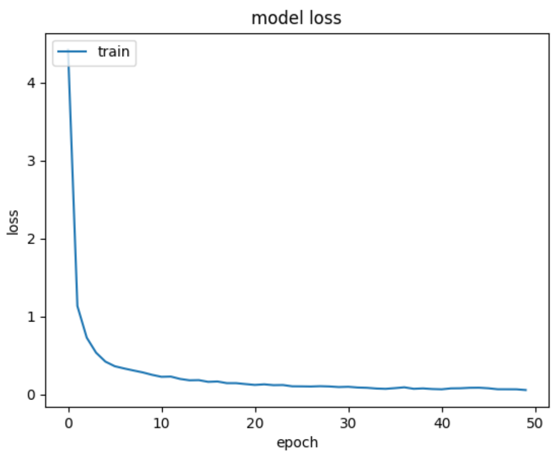

Fig 5. Graph depicting loss with mixed training

V. CONCLUSION

We study the intrinsic weaknesses of contemporary CNNs. We are presenting techniques to find a limited group of pixels efficiently, without having any network parameter information that leads to misdiagnosis by a disruption. Deep network of neurons. Our experimental results are comprehensive.

Surprisingly enough, the efficiency of our simple ways to generate opponent examples. Defenses are an intriguing avenue for study against these threats.This indicates that using adversarial training, a technique of training (or finalisation) networks to construct more robust classifiers for adversarial pictures is not particularly efficient for those adversarial images.

We really observed that even with opponent training the networks can only somewhat enhance their capacity to withstand fresh local search attacks.Only by thorough examination of oracle inquiries can avert efforts to build an adverse image can we predict that one possible countermeasure against those limited adversarial attacks.

The benefit of this technique is that the CNN can better defend itself from adverse instances via learning patterns from the original training examples and learning patterns from adverse fly pictures. Since with each batch of training the model may create its own opponent example, it can constantly learn from itself.

REFERENCES

[1] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Proceedings of the 26th Conference on Neural Information Processing Systems; 2012 Dec 3–6; Lake Tahoe, NV, USA; 2012. p. 1097–105.
[2] Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014. arXiv:1406.1078.
[3] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, et al.Mastering the game of Go with deep neural networks and tree search Nature, 529 (7587) (2016), pp. 484-489
[4] CrossRefView Record in ScopusGoogle Scholar Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, et al. Intriguing properties of neural networks. 2013. arXiv:1312.6199.
[5] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. 2014. arXiv:1412.6572.
[6] Kurakin A, Goodfellow I, Bengio S. Adversarial examples in the physical world. 2016. arXiv:1607.02533.
[7] Zheng T, Chen C, Ren K. Distributionally adversarial attack. 2018. arXiv:1808.05537.
[8] Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: Proceedings of the 2017 IEEE Symposium on Security and Privacy; 2017 May 22–26; San Jose, CA, USA; 2017. p. 39–57.
[9] Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: Proceedings of the 2016 IEEE European Symposium on Security and Privacy; 2016 Mar 21–24; Saarbrucken, Germany; 2016. p. 372–87.
[10] Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: a simple and accurate method to fool deep neural networks. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA; 2016. p. 2574–82.
[11] Papernot N, McDaniel P, Goodfellow I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. 2016. arXiv:1605.07277.
[12] Liu Y, Chen X, Liu C, Song D. Delving into transferable adversarial examples and black-box attacks. 2016. arXiv:1611.02770.
[13] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. 2017. arXiv: 1706.06083.
[14] Xie C, Wu Y, van der Maaten L, Yuille A, He K. Feature denoising for improving adversarial robustness. 2018. arXiv:1812.03411.
[15] Zheng T, Chen C, Yuan J, Li B, Ren K. PointCloud saliency maps. 2018. arXiv:1812.01687.
[16] Li J, Ji S, Du T, Li B, Wang T. TextBugger: generating adversarial text against real-world applications. 2018. arXiv:1812.05271.
[17] Athalye A, Carlini N, Wagner D. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. 2018. arXiv:1802.00420.