Reproducing "Toward Robust Image Classification[1]"

Lu Wang (lw491), Shuo Zhang (sz514)

Abstract

Neural networks is very useful for image classification, but is vulnerable when tested on adversarial images. Previously, methods of making image classification more robust vary from preprocessing (cropping, applying noise, blurring), adversarial training, to dropout randomization. In this paper, the author proposed a model to detect adversarial images first and then reclassify images based on images declared as clean by the model. The adversarial image detection model is based on a combination of two techniques: dropout randomization with preprocessing applied to images within a given Bayesian uncertainty. Then, the model is evaluated on mixtures of original MNIST dataset, and an adversarial MNIST dataset generated using Fast Gradient Sign Method (FGSM), Jacobian-based Saliency Map Attack (JSMA) and Basic Iterative Method (BIM) attacks, respectively. The authors achieved an average adversarial image detection accuracy of 97%, with an average image reclassification accuracy, after discarding images flagged as adversarial, of 99%. We reproduce the processes from scratch and achieved an average adversarial image detection accuracy of 96% and an average image classification accuracy of 98%. Except FGSM attack, under which our performance is a little bit weaker than the authors', we were able to achieve similar, or even better performance in terms of detection and reclassification accuracy.

1 Introduction

Deep neural networks (DNN) work well on image classification and can produce state-of-the-art results. However, they are shown to be vulnerable to attacks by adversarial examples, for example, adversarial images altered by the introduction of small perturbations. These perturbations can result in misclassification of images [2].

There have been many defenses against adversarial image attacks on neural network classification. A typical method is to utilize image preprocessing defenses, in which the images are altered in some way before being classified (blurring, cropping, noise) in order to disrupt any adversarial perturbations[5, 6]. Another typical method is to use dropout randomization defenses, in which the neural network adds randomization which supports the use of Bayesian uncertainty measurements to assess the likelihood of an image being adversarial[3, 12].

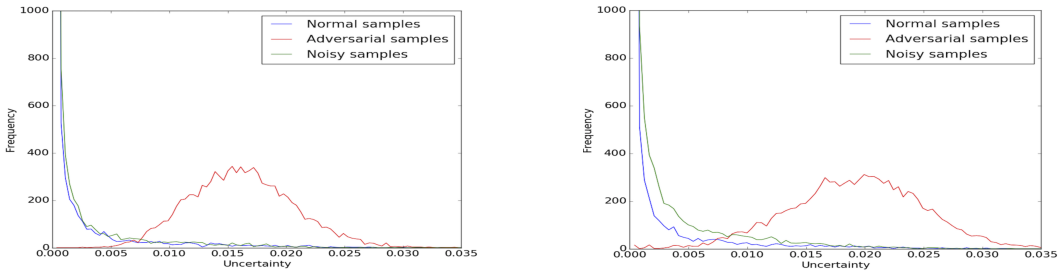
The author proposed a combination of defense: Firstly, to classify the image on Bayesian uncertainty in a dropout neural network[15], and then to use a secondary defense, preprocessing to double-check edge cases. In this regard, the model can not only achieve high clean image detection accuracy, but also identify adversarial images near uncertainty thresholds.

We reproduced this paper from scratch, and were able to get similar results as compared with the paper.

2 Related Works

There are mainly two types of defense in the literature. One is uncertainty-based defense[3, 12] and the other one is preprocessing-based defense[5].

As for uncertainty-based defense, on one hand, Feinman et al[3] used the uncertainty estimates on dropout networks, and the results are evaluated on MNIST[9], CIFAR-10[7], and SVHN[11] datasets. For every dataset, they achieved good results with adversarial images generated via Fast Gradient Sign Method (FGSM)[4], Jacobian-based Saliency Map Attack (JSMA)[13] and Basic Iterative Method (BIM) attacks[8]. The approach took advantage of the uncertainty estimates possible with dropout networks by assuming that the Bayesian uncertainty will be greater for adversarial examples than for clean data, as shown in Figure 1, because of the effect of the randomization on the necessarily precise perturbations. In addition, they used a Gaussian Mixture Model to analyze the outputs of the last hidden layer of their neural network, claiming that adversarial images will have a different distribution than clean ones. They also incorporated a kernel density estimate defense. While on the other hand, Papernot and McDaniel[12] estimated the uncertainty relying on the predictions of a second, separate neural network which is used to train the classification network. This defense also showed good results for the MNIST dataset.



(a) BIM (b) JSMA

Figure 1: Model uncertainty distributions for MNIST. Distributions are based on a histogram with 100 bins.

Also, several researchers used preprocessing-based defenses to detect adversarial examples. Graese, Rozsa, and Boult [5] explored several preprocessing techniques with the MNIST dataset and found that cropping-resizing was robust to different kinds of attack such as FGSM and Fast Gradient Value (FGV)[14] attacks. Besides, Guo et al.[6] tested cropping-resizing as well as other image transformations (image quilting, JPEG compression, etc.) and found that cropping-resizing was "very efficient" for different methods of attack.

3 Main Contribution of the Paper

The model in this paper uses a combination of the two defenses aforementioned in Section 2. Firstly, to classify the image on Bayesian uncertainty in a dropout neural network[15], and then to use a secondary defense, preprocessing to double-check edge cases. This hybrid method achieves both accuracy of adversarial images detection and accuracy of reclassification of images detected as clean, exceeding those of recent papers using similar techniques.

4 Methodology

4.1 Detection algorithm

The detection algorithm was shown in Figure 2. As can be seen clearly, when computed uncertainty is smaller than threshold L, the image is labeled clean and when the value is greater than H, the image is labeled as adversarial. The model outputs the provisional classification as the final classification, without reference to the later secondary method. However, if the uncertainty lies within L and H, the model employs the crop-resize secondary method. The final label is determined by the count of agreed predictions with the original provisional classification by neural networks, compared to C.

Algorithm 1: Detecting adversarial images during inferring time.

input : an image *img*, a high threshold H, a low threshold L, and a count of agreed predictions C.
output: image label (clean or adversarial).

```
calculate img prediction;
calculate img uncertainty;
if uncertainty > H then
    declare adversarial;
else
    if uncertainty < L then
        declare clean;
    else
        for 1:5 do
            crop & resize img;
            classify img;
            if class = prediction then
                increment counter;
            end
        end
        if counter > C then
            declare clean;
        else
            declare adversarial;
        end
    end
end
```

Figure 2: Detection Algorithm

4.2 Calculating Model Bayesian Uncertainty

The model uses dropout layers in the inference stage: For each image, we made N stochastic passes through the network. Each pass produced a probability for each available class by applying Softmax to the resulting logit vectors $z_1(x), ..., z_N(x)$. We then take the mean of the logit vector $z(x)$ for each class to obtain the stochastic prediction. The image is provisionally classified as the class with the highest mean. We can also obtain the stochastic uncertainty by computing the mean of the standard deviation of the predictions over the N stochastic passes.

4.3 Image processing

Images with middle uncertainties are then split into five overlapping crops (top left, bottom left, top right, bottom right, and center), and then resized back to the original size. Each of these crops is reclassified by the network, and if number C out of the five new classifications agree with the original, the image is labeled clean. Otherwise, the image is declared adversarial and then we filtered out of the test sample. This method is based on an assumption that class predictions of resized crops are likely to be different for adversarial images.

5 Experiment

We firstly used the LeNet[10], which is a convolutional neural network architecture to train all the parameters using the MNIST dataset with the dropout ratio set to 0.5. On clean, non-adversarial samples, this network achieved an accuracy of 98%.

Then we used three techniques to generate a set of adversarial images for testing, namely, Fast Gradient Sign Method (FGSM) [4],Basic Iterative Method(BIM) [8] and Jacobian-based Saliency Map Attack (JSMA) [13]. When the adversarial images are ready, the model is tested on 50/50 mixed clean and adversarial images for each of three attacks.

In the model, we adjusted the uncertainty levels based on the information gain to improve accuracy and then used these thresholds as the high and low level of Bayesian uncertainty. In other words, H and L are chosen to maximise the information gain firstly and then tuned by hyperparameter search to acquire the best performance. We also set the count of agreed prediction C to 5, which means that all

of 5 classifications from cropped and resized images need to be the same as the provisional original classification. Such H, L and C are chosen because they produced the highest ultimate accuracy. The model is tested on a hybrid set of 10,000 images comprised of 50% clean and 50% adversarial images.

After testing with the basic, undefended LeNet, we removed clean images which were misclassified; in the stage of testing the defended model, we also only used adversarial images generated from clean images which were correctly classified by the basic LeNet.

6 Results

All results are shown in Table 1, 2 and 3, with our reproducing results on the left and paper results in parenthesis.

For the FGSM attack, as seen in Table 1, the basic LeNet without defense had an accuracy of 58.5% on the 50/50 mixed test set; most of adversarial images were misclassified. By using the hybrid model, we were able to detect adversarial images with an significantly high accuracy of 92.9% and increased reclassification accuracy for the images flagged as clean to 96.1%, as seen from Table 2 and 3. However, our results are a little bit worse than the authors', because our hyperparameters might be different from theirs, which were not shown explicitly in the original paper.

While our FGSM performance is slightly weaker than the authors', we were able to reach a higher accuracy in terms of BIM attack. Compared with unprotected model, which was only 49.6% accurate on the BIM-mixed test set, our model detected adversarial images from the same test set with an accuracy of 96.5% and reclassified clean-flagged images 99.4% accurately. Both detection accuracy and reclassification accuracy slightly beat those of the paper.

As for the mixed test set generated via the JSMA attack, our undefended LeNet had an accuracy of 51.1%. With our defense, our model achieved 99.8% reclassification accuracy with 98.2% detection accuracy. These two results are very similar to the authors', which got 99.9% reclassification accuracy and 98.3% detection accuracy.

In addition, similar to the author's results, our results also shown better performance than the uncertainty-based defense and preprocessing-based defense, proposed by [3] and [5], respectively.

Table 1: Classification accuracy of undefended model (Reproducing results on the left and paper results inside parenthesis)

	FGSM	**BIM**	**JSMA**
Classification Acc.	58.5% (59.0%)	49.6% (50.0%)	51.1% (52.0%)

Table 2: Results for adversarial image detection (Reproducing results on the left and paper results inside parenthesis)

	FGSM	**BIM**	**JSMA**
False Negative	515 (252)	323 (249)	116 (85)
False Positive	194 (134)	28 (104)	66 (74)
True Negative	4766 (4796)	4928 (4826)	4893 (4856)
True Positive	4445 (4725)	4633 (4728)	4843 (4892)
Detection Acc.	92.9% (96.1%)	96.5% (96.4%)	98.2% (98.3%)

Table 3: Results of applying the defense with different attacks using MNIST dataset (Reproducing results on the left and paper results inside parenthesis)

Attack	**Clean images labeled clean**		**Adv. images labeled clean**		**Classification Accuracy**
	Correct	**Incorrect**	**Correct**	**Incorrect**	
FGSM	95.8% (97.3%)	0.0% (0.0%)	0.2% (1.7%)	3.9% (1.0%)	96.1% (99.0%)
BIM	99.3% (97.8%)	0.0% (0.0%)	0.0% (1.2%)	0.6% (0.1%)	99.4% (99.0%)
JSMA	98.6% (98.5%)	0.0% (0.0%)	1.1% (1.4%)	0.2% (0.1%)	99.8% (99.9%)

7 Conclusion

In conclusion, the proposed defense model using the combination of Bayesian uncertainty and image preprocessing proved to be effective, with accuracy in the high 90s, at detecting adversarial examples in a mixed MNIST test set. This allowed for considerably higher accuracy in classifying the remaining, primarily clean, images. The results are shown to be better than those of the previous works.

As for our reproducing work from scratch, it also works well. Other than FGSM attack, under which our performance is slightly weaker than the authors', we were able to achieve similar, even better performance in terms of detection and reclassification accuracy when tested on hybrid set with adversarial images generated by BIM and JSMA attacks.

8 Our Contribution

This reproducing work is finished by two people, Lu and Shuo. Lu is responsible for using LeNet convolutional neural network to train the clean images and writing the whole report. Shuo is responsible for generating adversarial images using FGSM, BIM and JSMA attacks and for coding the hybrid defense model.

We then calculated the reclassification accuracy and detection accuracy together and merged the code. It took us days to do hyper-parameter search and tuning, on which we worked together.

References

[1] Basemah Alshemali, Alta Graham, and Jugal Kalita. "Toward robust image classification". In: *Proceedings of SAI Intelligent Systems Conference*. Springer. 2019, pp. 483–489.

[2] Nicholas Carlini and David Wagner. "Adversarial examples are not easily detected: Bypassing ten detection methods". In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 2017, pp. 3–14.

[3] Reuben Feinman et al. "Detecting adversarial samples from artifacts". In: *arXiv preprint arXiv:1703.00410* (2017).

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[5] Abigail Graese, Andras Rozsa, and Terrance E Boult. "Assessing threat of adversarial examples on deep neural networks". In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2016, pp. 69–74.

[6] Chuan Guo et al. "Countering adversarial images using input transformations". In: *arXiv preprint arXiv:1711.00117* (2017).

[7] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[8] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. "Adversarial machine learning at scale". In: *arXiv preprint arXiv:1611.01236* (2016).

[9] Yann LeCun. "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/* (1998).

[10] Yann LeCun et al. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[11] Yuval Netzer et al. "Reading digits in natural images with unsupervised feature learning". In: (2011).

[12] Nicolas Papernot and Patrick McDaniel. "Extending defensive distillation". In: *arXiv preprint arXiv:1705.05264* (2017).

[13] Nicolas Papernot et al. "The limitations of deep learning in adversarial settings". In: *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE. 2016, pp. 372–387.

[14] Andras Rozsa, Ethan M Rudd, and Terrance E Boult. "Adversarial diversity and hard positive generation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 25–32.

[15] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.