11-785: PERMUTATION PHASE DEFENSE

Oyku Han
oykuh@andrew.cmu.edu

Harlin Lee
harlinl@andrew.cmu.edu

Stanley Nnamdi Adom
sadom@andrew.cmu.edu

ABSTRACT

Deep neural networks have shown the capacity to perform classification tasks better than humans in some instances. However, they are susceptible to fatal adversarial attacks. Permutation Phase Defense (PPD), is a method purposed at the ICLR conference to combat adversarial attacks. It amalgamates random permutation of image pixel and phase components of its Fourier transform. In this report, the PPD method was implemented and results were compared to original paper. Testing on MNIST and CIFAR-10 dataset, the perturbation l_∞ norm produced similar outcome while l_2 diverged.

1 INTRODUCTION

Deep Neural Networks (DNNs) are among the most preferred methods in image classification tasks, and their performance can match human's on simple tasks such as classification of numbers on the MNIST dataset. [Ciresan et al. (2012)] However, DNNs are also vulnerable to adversarial attacks, which perturb the input deliberately such that the network is fooled into making false predictions. In the field of image recognition, adversarial examples are images that are very close to the decision boundary of a classification network, and their difference from clean, unperturbed images is undetectable by eye. [Goodfellow et al. (2015)]

Successful adversarial attacks on DNNs are a serious source of concern for many applications including autonomous driving and medical image diagnostics. Finding a robust defense mechanism against attacks is an important open question. Permutation Phase Defense (PPD), which is submitted to the 2019 International Conference on Learning Representations (ICLR), proposes a method to mitigate the effects of the adversary on the network by encrypting and transforming the input to the frequency domain. [Anonymous (2019)] As part of the 2019 ICLR Reproducibility Challenge, this project aims to 1) reproduce the experimental results presented in the PPD paper, 2) analyze the paper's experimental design and results and their shortcomings, and 3) propose concrete methods to improve their system.

2 BASELINE: PERMUTATION PHASE DEFENSE (PPD) MODEL

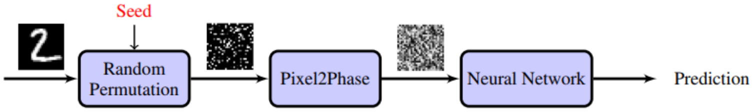


Figure 1: Architecture of the Permutation Phase Defense model, reproduced from the original paper.

As a baseline model, Permutation Phase Defense (PPD) is implemented to defend the neural network against adversarial attacks during classification of MNIST and CIFAR-10. The first line of defense in PPD is to permute the pixels of the input image using a fixed random seed that is hidden from the adversary. The permuted images then undergo Fourier transformation, after which only the phase information is kept to train a DNN for image classification. This architecture is depicted in Figure 1. The preferred type of neural network for PPD is a Multi-Layer Perceptron (MLP). Thus, the shuffling of the pixels provides encryption, and does not disrupt the prediction accuracy, while

	MNIST					CIFAR-10				
l_∞ **perturbation**	0.03	0.1	0.2	0.3	0.4	0.03	0.1	0.2	0.3	0.4
replicated FGSM	98.04%	97.76%	97.10%	**92.71%**	83.73%	52.65%	**44.59%**	**38.44%**	34.12%	**28.87%**
original FGSM	97.8%	97.6%	97.1%	95.4%	91%	48.2%	47.9%	45.3%	39.7%	31.1%
replicated BIM	97.99%	97.80%	97.49%	96.72%	93.84%	**52.55%**	47.47%	42.45%	38.32%	34.61%
original BIM	97.8%	97.6%	96.7%	95.2%	91%	48.2%	47.7%	45.2%	39%	30.1%
replicated PGD	98.06%	97.86%	97.26%	96.25%	95.73%	52.42%	48.01%	42.27%	39.06%	38.78%
original PGD	97.8%	97.7%	97%	95.2%	91.3%	48.3%	47.6%	45.4%	41.6%	37.1%
replicated MIM	**97.98%**	**97.71%**	**96.95%**	93.91%	**85.65%**	52.96%	45.38%	39.61%	**33.90%**	29.53%
original MIM	**97.8%**	**97.6%**	**95.8%**	**87.4%**	**67.5%**	**48.2%**	**47.5%**	**40.4%**	**26.1%**	**15.6%**
l_2 **perturbation**	0.1	0.7	1.1	3.2	4	0.3	2	4	6.5	10.5
replicated FGM	**8.23%**	**7.92%**	**7.85%**	**7.76%**	**7.73%**	**48.66%**	**48.38%**	**48.53%**	**48.53%**	**48.51%**
original FGM	97.8%	97.8%	97.8%	97.3%	97.1%	48.1%	48.1%	48%	47.3%	45.4%
replicated PGD	97.95%	97.87%	98.06%	98.04%	97.98%	55.19%	54.97%	54.68%	54.60%	54.38%
original PGD	97.8%	97.8%	97.8%	**97.3%**	**96.7%**	48.3%	48.3%	**47.5%**	**45.7%**	**39.3%**
replicated CW	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD	TBD
original CW	**97.7%**	**97.4%**	**97.7%**	97.7%	97.8%	**48%**	**48%**	47.9%	46.5%	39.5%

Table 1: Comparison of the prediction accuracies reported in the original paper and our replicated network. Accuracies shown in green indicate that our ensemble of 10 PPD models was able to perform better than their counterpart reported in the original paper. The lowest accuracies are also highlighted in bold.

working in the phase domain ensures that the perturbation gets distributed over pixels, and its effect is limited to that of random noise.

The original PPD paper trains an ensemble of PPD models, each with a different fixed random seed. However, the adversary is assumed to have access to only a single model. The prediction accuracies on clean data are reported as 96% and 45% on MNIST and CIFAR-10 when a single PPD model is used. The accuracy increases to 97.75% and 48% on MNIST and CIFAR-10 when an ensemble of 10 PPD models are used. The authors claim that 10 models ensure good enough performance.

This ensemble model is tested against 7 adversarial attacks in two norms, l_2 and l_∞. Attacks on l_∞ norm are Fast Gradient Sign Method (FGSM), Basic Iterative Method (BIM), Projected Gradient Descent ($l_\infty-$PGD), Momentum Iterative Method (MIM) whereas l_2 attacks include Fast Gradient Method (l_2-FGM), (l_2-PGD), Carlini and Wagner Method (CW). They were implemented using the Cleverhans API developed in Papernot et al. (2018). Performance of ensemble of 50 PPD models were reported in the original PPD paper and is also shown in Table 1 against these attacks.

2.1 Initial Results: Reproducing PPD Results

The experiments described in the PPD were replicated to the best of our capabilities. Identical to the original paper, our PPD model is also a simple 3 layered MLP (800→300→10), where the input image is shuffled and its phase component is passed to the network. No dropouts or batch normalization layers were used, since the paper did not mention any. As the PPD paper reports that using an ensemble of 10 models is sufficient enough for a defense mechanism, we replicated their approach using an ensemble of 10 PPD models. However, the original PPD paper does not define their ensemble method, so we chose to use majority voting rule.

In Table 1, the accuracies of our replicated ensemble using 10 PPD models are compared to the accuracies for an ensemble of 50 models reported in the original PPD paper. Figure 2 visualizes some of our results as a function of perturbation. For the MNIST dataset, when l_∞ attacks are used, the replicated experiments back the results of the original paper as the reported accuracy differences are under 1.1% change for perturbations (ϵ) under 0.3. For higher perturbation levels, the difference between the replicated and original work is most noticeable in MIM attack where for ϵ equal to 0.3

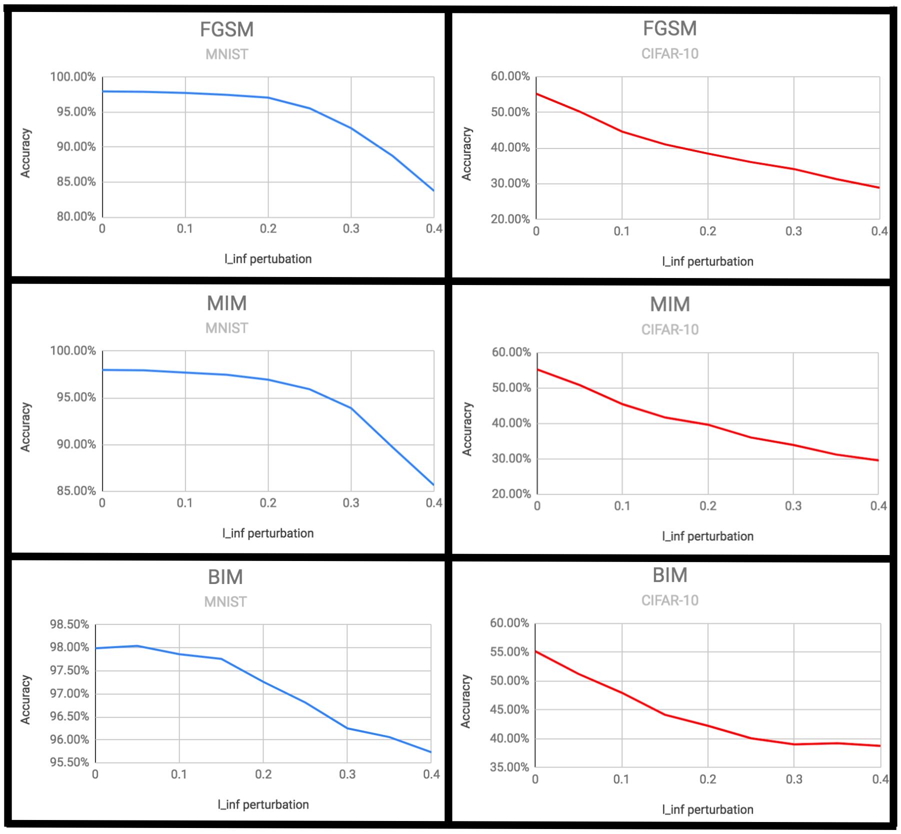


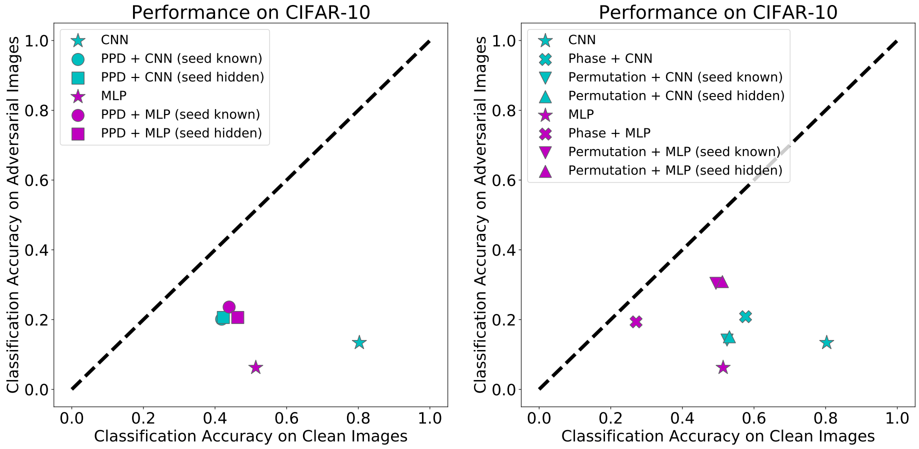
Figure 2: Classification accuracy under FGSM, MIM, and BIM attacks for our ensemble of 10 models. ℓ_∞ PGD and all ℓ_2 attacks were not plotted, simply because the change in accuracy was minimal. These values correspond to those in Table 1.

and 0.4, our implementation achieves a better accuracy with 6.5% and 18.1% increase respectively. For the more complicated dataset of CIFAR-10, the original PPD paper accuracies differ from our own by at most 5% for most of the attacks for various ϵ values. The exceptions are FGSM with ϵ values of 0.2 and 0.3, MIM application with a perturbations of 0.3 and 0.4 whose accuracies differed by 6.9%, 5.6%, 7.8% and 13.93% respectively. It is important to note that for MIM, our implementation performance was better than the original with the aforementioned difference.

However, a comparison on the l_2 attack, FGM, implementations for MNIST uncover a discrepancy where the highest accuracy according to our initial experiment is 8.23% whereas the lowest accuracy listed in the original paper was 97.1%. This might be due to several factors including the possible difference in ensemble method or the number of models used in the ensemble or implementation issues in our experiment. On the other hand, rest of the experiments regarding l_2 attacks on both MNIST and CIFAR-10 proved that the original accuracies mentioned are valid as all of our implementation was able to achieve higher accuracies using PPD as the defense.

3 Error Analysis and Proof of Concept

In this section we discuss our concerns about the PPD model as described in the ICLR paper, and provide justification for our final experiments planned in Section 4. For simplicity, let us analyze the PPD model's performance on the CIFAR-10 dataset, and focus on the ℓ_∞ FGSM attack with $\epsilon = 0.4$. The paper reports that a single PPD + 3 layer-MLP model was able to achieve 45% classification accuracy on clean CIFAR-10, and the 50 ensemble models were able to achieve 31.1% accuracy under attack. They did not report the performance of a single PPD + MLP model under attack, but our experiment yielded 24% for *both* cases when the single model had its permutation seed known *and* hidden to the adversary.



(a) With and without PPD.

(b) With and without permutation/phase transformation

Figure 3: MLP and CNN performance on CIFAR-10

Importance of hidden permutation seed and the effect of ensemble methods: According to our experiments, whether the permutation seed was hidden to the adversary or not had negligible effect on the defense of a single PPD model. This raises questions as to whether the paper is making fair claims, i.e. whether their experimental design properly decouples the effect of ensemble learning from the effect of PPD. Therefore, for our experiments outside of Table 1, which is simply reproducing efforts, we will report single model results. In order to properly consider ensemble methods, we believe that the adversary should be either 1) given an ensemble model and their 50 seeds, or 2) allowed to create smart adversarial examples that are "averaged" over many different random permutations.

Accuracy on clean images: The PPD model is a DNN designed and trained to be robust in image classification tasks under adversarial attack. When evaluating a defensive model like PPD, it is important to consider its performance in two aspects: robustness against adversarial attacks, and classification accuracy on clean images. In that regard, the authors seem to gloss over the fact that the classification accuracy on clean CIFAR-10 is very poor, when Benenson (2016) and Graham (2014) has reported accuracy of 96.53% on the same dataset already in 2016.

Sacrificing the accuracy on clean images in order to gain robustness in an adversarial setting is expected to some degree, and is often a conscious design choice. However, an accuracy as low as 45% on CIFAR-10 casts doubt into whether the PPD method would actually be useful in practice. Hence, we plan to explore in the final experiment how the PPD model performs in even harder tasks. Our hypothesis is that the drop in performance from adding in PPD will only be greater in the difficult datasets and classification tasks that are described in Section 4.1.

Or, perhaps this lackluster performance is due to the MLP part of PPD, and not due to the pixel permutation and phase transform. Maybe we can extend PPD to CNNs and improve the classification accuracy on clean images, since the state-of-the-art image classification DNNs are CNNs. So we have tested this.

Figure 3a summarizes the performance of the following six models on CIFAR-10: (DNN type, PPD specifications) $\sim$ $\{$MLP, CNN$\}$ $\times$ $\{$no PPD, PPD (seed known to adversary), PPD (seed hidden to adversary)$\}$. Since this is just for proof of concept, we made a simple CNN with just four convolutional layers that is used as an example in Keras (2018). First, note that models with seed unknown to the adversary (squares) perform almost identically to the models with seed known to the adversary (circles), as mentioned before. Furthermore, note that without PPD (stars), there is a huge gain in the classification accuracy on clean images (horizontal axis) from using CNN over MLP, as expected. However, this gain is clearly absent in the PPD models, which affirms our concerns about the practicality of PPD. We believe that this is because permuting the pixels invalidates the shift-invariance assumption exploited by CNNs, and because we are only using the phase information.

This point is reinforced by Figure 3b, in the comparison between naive models (stars) and models with just permutation (triangle), and between naive models (stars) and models with just phase (crosses). For both MLP and CNN, working with just phase information increases the adversarial accuracy while decreasing performance in clean images. This may be because the perturbation is distributed to nearby pixels, while disregarding magnitude leads to the loss of color information in CIFAR-10. It is possible that this decrease in performance will not be observed in the black and white MNIST images, but we have not tested this. The effect of permuting pixels on CNN is similar to that of pixel2phase, but the decrease in clean image accuracy is almost negligible in MLP. This is as expected, since we concatenate the pixels into a vector in MLP anyways. In CNN, however, the position of pixels are essential to detection of meaningful local features.

4 Final Experiment

4.1 Further Testing of PPD

The authors have shown that their 3-layer MLP model achieves decent classification accuracy on those datasets, especially for MNIST. They have also reported that the omission of the magnitude information from Fourier Transform does not hurt the classification on MNIST and CIFAR-10. However, we suspect that this might not be the case on other larger and more complex datasets such as MESSIDOR, The Street View House Numbers, and VisualQA. We will test PPD model on these datasets and report their accuracies under different magnitudes of l_2 and l_∞ attacks.

4.2 Improving PPD Model

Based on our error analysis in Section 3 and Figure 3a, it is evident that applying the PPD model *as is* to a simple MLP and a simple CNN sacrifices the baseline accuracy too much to be a useful system. Therefore, in our final experiment, we will try to extend PPD to CNNs in a way that produces good baseline accuracy on clean images while maintaining the robustness against adversarial attacks.

We will mainly experiment with changing the type and order of permutation, as the analysis on Figure 3b suggests. First, we will replace shuffling by pixel with shuffling by relevant features, such as Harris corner detectors, or features detected by convolutional layers. This can yield meaningful results, because each descriptor has a meaning, and we would not lose information from neighboring pixels. Also, we plan to create a stronger defense system by putting the random permutation between the phase and neural network. This can help the network better understand the orientation of the phase value. We intend to experiment with denoising the images as well.

5 Related Work

After the introduction of adversarial examples in Szegedy et al. (2014), several methods have been proposed as defense mechanism. Goodfellow et al. (2015) proposed an adversarial training method where the network is trained using the adversarial examples created using FGSM to make the network robust to small perturbations. Madry et al. (2018) has devised the Projected Gradient Descent attack and has utilized the attack to train the network adversarially with the guarantee that the network would be resistant to a pre-defined set of attacks. Although the adversarial training helps immensely in the defense against the attacks that was used in the training, they still leave the network open to different types of attacks. On the other hand, PPD was desgined to be more of a general defense against all adversarial attacks, according to its authors. Another defense solution was to hide the classifier weights or gradients from the adversary. But it was discovered by transfering adversarial examples from one model to an adversarially trained substitute, an adversarial attack that would not succeed on the initial network was proven to work on the substitute. [Papernot et al. (2017)] Last but not least, hiding parts of the input via replacing segments of image or recovering initially randomly dropped pixels by total variance minimization was proposed as a defense in Guo et al. (2018). However, to maintain good prediction performance most of the image has to be maintained close to its original form which does not help in defending against adversaries.

References

Anonymous. Ppd: Permutation phase defense against adversarial examples in deep learning. In *Submitted to International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkElFj0qYQ. under review.

Rodrigo Benenson. What is the class of this image? discover the current state of the art in objects classification. http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#43494641522d3130, 2016.

Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *IN PROCEEDINGS OF THE 25TH IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR 2012*, pp. 3642–3649, 2012.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

Benjamin Graham. Fractional max-pooling, 2014.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

Keras. cifar10 cnn example. https://github.com/keras-team/keras/blob/master/examples/cifar10_cnn.py, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pp. 506–519, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4944-4. doi: 10.1145/3052973.3053009. URL http://doi.acm.org/10.1145/3052973.3053009.

Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6199.