Class Retrieval of Adversarial Attacks

Jalal Al-Afandi, András Horváth
Peter Pazmany Catholic University Faculty of Information Technology and Bionics
Budapest, Práter u. 50/A, 1083
alafandi.mohammad.jalal, horvath.andras @itk.ppke.hu

Abstract

Adversarial attacks pose a genuine threat in practical machine learning applications. There are existing methods to detect these attacks, which can only prevent the systems from making erroneous decisions, but not helping them in any way. Here we will present a new, additional and required element to ameliorate adversarial attacks: the recovery of the original class after a detected attack. Recovering the original class of an adversarial sample without taking any precautions is an uncharted concept which we would like to introduce with our retrieval algorithm based on counter attacks. As case studies, we demonstrate the validity of our approach on MNIST, CIFAR10 and ImageNet datasets where recovery rates were 88%*,* 53% *and* 62% *accordingly.*

1. Introduction

Adversarial attacks which were first introduced in [23] pose a significant challenge in the practical application of deep learning. These methods exploit that the high dimensional inputs can be perturbed slightly pushing the samples through the border of the high dimensional geometrical manifolds of the classifiers [9].

Minor perturbations over the entire image were the first introduced adversarial attacks which has been demonstrated by Goodfellow [10]. Many threatening results have surfaced such as [17] demonstrating a universal perturbation fooling a classifier on any image and [1] showing the possibility of fooling a classifier with a 3D printed object, urging the research community to find defense mechanisms to ensure the safe application of neural networks in real-world computer vision systems.

Most commonly applied defenses against adversarial attacks belong to one of the three following approaches: adversarial training [22], modifying the network architecture [20] or detection approaches [13]. The first two methods can be considered passive defenses, which make the network more resilient against attacks and has to be applied during network training. The third method, adversarial attack detection, is the most viable in case of real-world applications, since in this approach the defense mechanism is separated from the weights and architectures of the original network, hence the detector can be changed or updated without retraining the original classifier.

Detection of adversarial attacks is a great initial step in real world applications and provides the highest accuracy from the previously mentioned approaches, but on its own, can not be enough to ensure safety, since detection of attacks will still leave the autonomous system in complete doubt preventing it from making sound and reliable decisions. Imagine a self-driving car which detects an object and can correctly identify that it was malevolently attacked. This information is not enough to make an action and can leave the agent in complete doubt regarding its actions. The ultimate safe solution against adversarial attacks requires an additional step which is presented in this paper: the retrieval of the original class of the attacked samples. We will not only introduce this new problem, but also introduce a novel algorithm, which retrieves the original class of attacked images and might serve as a baseline in future experiments. We will demonstrate the effectiveness of our algorithm using four different attack mechanisms on three different datasets MNIST, CIFAR10 and 10 randomly selected classes from ImageNet.

1.1. Adversarial Attack Detection

Defenses against adversarial attacks are required to prevent security threats in real-world application of neural networks. Most defenses rely on one of the following three main approaches:

- Modifying the training process e.g adding adversarial samples [19] or perturbing the input before inference [4] [7] [14].

- Modifying the network architecture, e.g adding extra masking layer(s) before the last layer [8] or changing the loss function by penalizing the degree of variety per class in the output[20].
- Using external models as detectors eg. SafetyNet[13] and Convolutional Filter Statistics detector [11].

Although in certain cases adversarial examples can be hard to detect, this approach is still a viable defense mechanism, providing the highest accuracy reaching $\sim 85\%$ [11] preventing most security breaches. There are a lot of other detectors [16] [12] separating the clean image from the adversarial one by finding some distinguishable features and properties e.g. convolution filter statistics [11] and manifolds [16]. Although detectors are considered strong defenses against adversarial attacks, an attack can cause a halt in the system hindering the achievement of any task. In a time sensitive task e.g self-driving cars, where an on-the-spot decision has to be drawn, detectors are not sufficient and a retrieving approach has to be installed recovering the original class of the input.

2. Class retrieval

Most non-detection defenses are vulnerable to counter-counter attacks [3] rendering a potential exposure keeping the system without any functioning protective shield. Detection based defenses on the other other hand can be continuously updated, but lack the ability to steer the decision making process obstructing the installation of any safety measure. Thus, a recovery algorithm has to be employed after the detection of adversarial attacks providing robustness and resilience.

[21] hypothesized that adversarial attacks exploit the edge of the decision boundary between classes pushing the adversarial sample to the targeted class. Their idea stemmed from the speculation that training data will be pushed to the edge of the decision boundary once they are classified correctly. In [18] and [24], the authors assume that the reason behind the adversarial vulnerability of neural networks is the highly positively curved decision boundary where the curvature is very intricate near the classes borders. The high dimensionality of neural networks creates convoluted borders between all the classes making a targeted adversarial attack highly possible. Taking into account the complexity of the curvature of the decision boundary, we hypothesise that the distance between the adversarial sample and the original class's manifold in the feature space of the decision boundary is smaller than the distance between the adversarial sample and any other classes' manifolds, hence, all the adversarial samples and their counter attacks are in the vicinity of the original class manifold. We have implemented our idea, an adversarial retrieving algorithm, on the notion of our former hypothesis to predict the original class by counter attacking the adversarial samples.

What we can derive from our hypothesis is that during the counter attack it would be the easiest to transform back the attacked image to its original class, since the attacked sample still contains the required features in majority, and the manifold of the attacked class is the closest to the decision boundary of the original class. The counter attack can return the attacked sample to its original class easily since the adversarial sample is on the edge of the original class decision boundary. Due to the high dimensionality of the decision boundary curvature, there exist an intricate border between the manifold of each two randomly selected classes. To illustrate this hypothesis we made experiments on the MNIST dataset, where 100 randomly selected samples from the same class were attacked (we considered them an 11th class) and depicted the projected two-dimensional position of their manifolds using the UMAP algorithm. An example image can be seen on Fig. 1 and as it can be seen it confirms our assumptions that going back to the original class manifold can be easier (done in less iterations) than turning the sample to any other class. The cross entropy loss between the counter attack's output for a targeted class will be the smallest when targeting the original class.

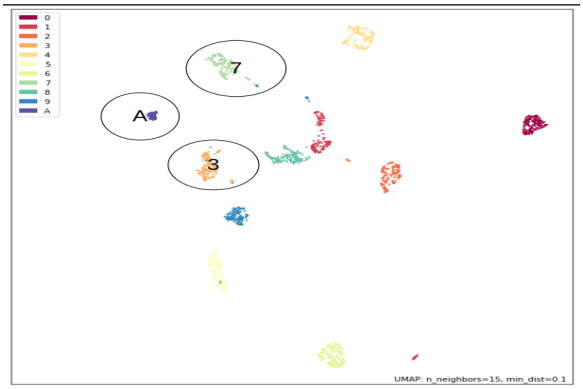

Figure 1. This figure displays a two dimensional UMAP projection of the MNIST digits ins the sklearn package with an additional 100 attacked samples which originally belonged to class 7 and were transformed to class 3 with the PGD algorithm. The classes are marked and circled on the figure 'A' denoting the attacked samples. We generated similar figures for other classes as well and the results were qualitatively the same in all cases.

Our adversarial attack class retrieval algorithm is presented in algorithm 1 as a pseudo-code. *AdvImg* is the adversarial image which has been selected by an

Algorithm 1: Class retrieval algorithm of adversarial attacks

Parameters: $NbIter, NbClass, AdvImg, AdvLab$
Result: $OrigClass$
$Losses = 0, Losses[AdvLab] = \infty$
for $Target : 0$ **to** $NbClass$ **do**
  if $Target! = AdvLab$ **then**
    $ContAdvImg = Attack(AdvImg, NbIter, Target)$
    $Losses[Target] = loss(ContAdvImg, Target)$
  end
end
$OrigClass = argmin(Losses)$

adversarial attack detector. The neural network prediction for the adversarial image is $AdvLab$ which is a misclassification according to our detector. $NbClass$ is a fixed parameter representing the number of classes in our classification problem. We apply a targeted counter attack using $Attack()$ function where $NbIter$ is the number of iterations and $Target$ is the targeted label. The loss function, $loss()$ calculates the cross entropy loss of the counter adversarial image ($ContAdvImg$) having $Target$ as a label. We exterminate the possibility of the adversarial label, $AdvLab$, being the original class by setting its loss to infinity. The original label, $OrigClass$, is the class with the minimum loss excluding the adversarial label where we used $argmin$ function to return the index of the smallest loss.

To prove the validity of our work, we assumed the existence of an optimal detector which can identify all adversarial attacks. We investigated four different adversarial attacks (projected gradient descent attack (PGD) [15] , iterative basic method with momentum [6], Deepfool [19] and patch based attack [2]) [1] which were briefly explained in the previous section. Deepfool is not a targeted attack, thus we only used the first two previously mentioned attacks, PGD attack and iterative basic method with momentum, as a counter adversarial attack. All the investigated adversarial attacks are white box attacks which relies on the gradients to calculate the small perturbations fooling the classifier.

3. Experiments

3.1. MNIST

To validate our hypotheses, detailed experiments have been made over the MNIST and other datasets, as we will see in the next paragraphs. We investigated four different adversarial attacks (PGD attack, iterative basic method with momentum, Deepfool and patch based attack) to create a matrix with another two counter attacks (PGD attack, iterative basic method with momentum). In each case, we attempted to retrieve the class of 1000 successfully attacked samples which means altogether 8000 experiments were made. Adversarial attack's maximum distortion were set to 0.8 which is irrelevant for Patch based attacks due to their unlimited perturbations (only limited by the range of the intensity values). Number of iterations, $NbIter$, is 1000 for the adversarial attacks insuring a successful attack, but we set it to 3 for the counter adversarial attack illustrating the fast convergence to the original class. AlexNet architecture was used throughout our experiments providing a good baseline network where the average accuracy on clean samples is 96% (the original 28x28 images were rescaled to 224x224 ensuring the required input size). Figure 2 demonstrates the high accuracy of the class retrieval algorithm investigating the usage of two counter attacks against the adversarial samples of four different attacks. In average, 94% of the attacked samples were recovered correctly predicting their original class averting misclassification.

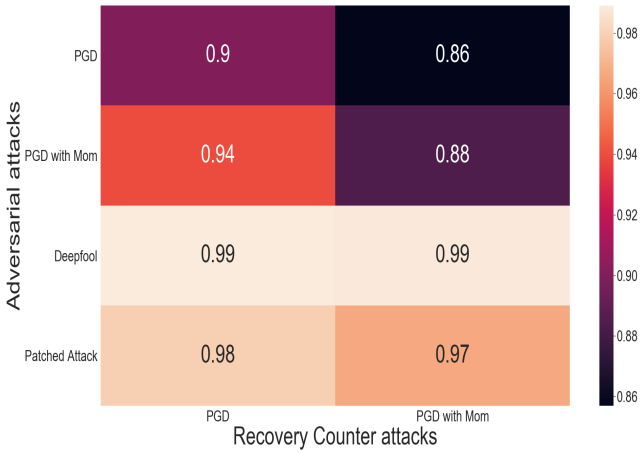

Figure 2. The figure illustrates the success rate of our class retrieval algorithm on the MNIST dataset, where each cell represent the accuracy of the retrieval in a specific setup e.i the algorithm used for the attack can be see in the rows and the algorithm used for the counter attack can be find in the columns.

3.2. CIFAR10

We investigated another simple dataset, CIFAR10, to show the effectiveness of our approach. The same setup which was described in detail in the previous

[1] The first two attacks were adopted from the Advertorch library [5] while we used the codes of the original papers for the other two attacks

paragraph was used. The only parameters which have been modified significantly are Adversarial attack's maximum distortion and adversarial attacks number of iterations $NbIter$, we set the former to 0.3 and the later to 11. We opted to use these smaller values in comparison to our setup with MNIST due to the faster and easier conversion of adversarial samples. Figure 3 shows the success rates using our class retrieval algorithm over CIFAR10 dataset engulfing eight different setups. We used the AlexNet architecture throughout our experiments providing a good baseline with 88% accuracy classifying clean samples. The overall averaged retrieval accuracy is 53%, which demonstrates the viability of our approach.

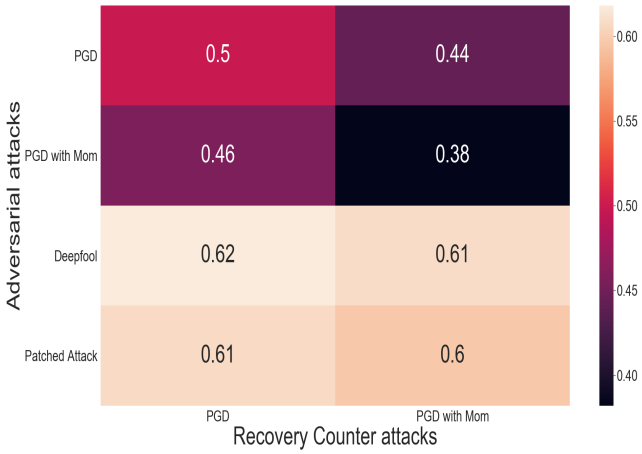

Figure 3. The figure illustrates class recovery success rates on the CIFAR10 dataset, where each cell represents the accuracy of the retrieval in a specific setup e.i the algorithm used for the attack and the algorithm used for the counter attack.

3.3. ImageNet

Our algorithm can be applied in practice with datasets which contain a limited number of classes N, and because of the nature of the algorithm ($N-1$ number of counter attacks have to be made).To investigate complex and more practical dataset with high resolution images, we have randomly selected 10 classes from ImageNet to execute similar experiments as in case of MNIST and CIFAR10. Altogether 6000 attacks and retrievals were made and to balance the effect of random class selection we selected ten different classes for each 100 attacks. High resolution images are easier to attack because of their high dimensionality rendering a complicated but vulnerable decision boundary curvature which can be compromised by slightly modifying the high number of pixels. Due to this we decreased the adversarial attack's number of iteration to six. Throughout our investigation, We used the pretrained version of $Inception_{v3}$ architecture, from the torchvision models library. Inception model has one thousand possible output classes, but our adversarial and counter attacks were targeting the randomly selected ten classes only. We can see the success rate on ImageNet in figure 4 where each cell represents the accuracy of a specific attack and counter attack investigating thousand cases with 10 different random classes for each hundred trials. 62% is the average accuracy of recovering six thousand attacked images from ImagNet using our adversarial retrieval.

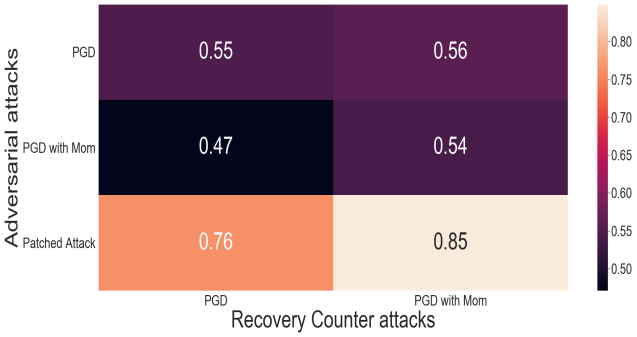

Figure 4. The figure depicts the success rate of our adversarial retrieval on ImageNet dataset, where each cell represent the accuracy of the retrieval in a specific setup investigating 1000 cases selecting 10 random classes for each hundred trials.

4. Conclusion

We presented a novel problem, the class retrieval and recovery of adversarial attacks along with a proposed solution, which can be used as a baseline approach in further experiments. Our retriever is a self-evident addition to adversarial attack detectors and the combination of these two methods can enable the practical applicability of deep network even in case of attacks. We investigated four different adversarial attacks (PGD attack , iterative basic method with momentum, Deepfool and patch based attack) on three different datasets (MNIST, CIFAR10 and ImageNet). The results are promising and consistent across all attacks and datasets where the average accuracy is 88%, 53% and 62% respectively. Although the retrieval algorithm was not able to recover the original class in all cases, but, as a preliminary concept, it clearly shows that the original class can be retrieved. We hope this can open the way for further development and fine-tuning of class retrievals of adversarial attacks which can increase the robustness of deep neural networks in real-world applications.

Acknowledgements

This research has been partially supported by the Hungarian Government by the following grant: 2018-1.2.1-NKP-00008 Exploring the Mathematical Foundations of Artificial Intelligence and the support of the grant EFOP-3.6.2-16-2017-00013 is also gratefully acknowledged.

References

[1] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. *CoRR*, abs/1707.07397, 2017. 1

[2] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch, 2017. 3

[3] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods, 2017. 2

[4] S. Dathathri, S. Zheng, T. Yin, R. M. Murray, and Y. Yue. Detecting adversarial examples via neural fingerprinting. *arXiv preprint arXiv:1803.03870*, 2018. 1

[5] G. W. Ding, L. Wang, and X. Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019. 3

[6] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018. 3

[7] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy. A study of the effect of jpg compression on adversarial images, 2016. 1

[8] J. Gao, B. Wang, Z. Lin, W. Xu, and Y. Qi. Deepcloak: Masking deep neural network models for robustness against adversarial samples, 2017. 2

[9] J. Gilmer, L. Metz, F. Faghri, S. S. Schoenholz, M. Raghu, M. Wattenberg, and I. Goodfellow. Adversarial spheres, 2018. 1

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples, 2014. 1

[11] X. Li and F. Li. Adversarial examples detection in deep networks with convolutional filter statistics, 2017. 2

[12] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang. Detecting adversarial image examples in deep neural networks with adaptive noise reduction, 2019. 2

[13] J. Lu, T. Issaranon, and D. Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly, 2017. 1, 2

[14] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao. Foveation-based mechanisms alleviate adversarial examples, 2016. 1

[15] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks, 2017. 3

[16] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples, 2017. 2

[17] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. *CoRR*, abs/1610.08401, 2016. 1

[18] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, and S. Soatto. Analysis of universal adversarial perturbations, 2017. 2

[19] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016. 1, 3

[20] A. S. Ross and F. Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients, 2017. 1, 2

[21] A. Rozsa, M. Gunther, and T. E. Boult. Towards robust deep neural networks with bang, 2018. 2

[22] S. Sankaranarayanan, A. Jain, R. Chellappa, and S. N. Lim. Regularizing deep networks using efficient layerwise adversarial training, 2018. 1

[23] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks, 2013. 1

[24] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses, 2020. 2