DETECTING ADVERSARIAL PERTURBATIONS WITH SALIENCY

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper we propose novel method for detecting adversarial examples by training a binary classifier with both origin data and saliency data. In the case of image classification model, saliency simply explain how the model make decisions by identifying significant pixels for prediction. Perturbing origin image is essentially perturbing saliency of right output w.r.t. origin image. Our approach shows good performance on detecting adversarial perturbations. We quantitatively evaluate generalization ability of the detector where detector trained with strong adversaries and its' saliency perform well on weak adversaries. In addition, we further discuss relationship between solving adversary problem and model interpretation, which helps us understand how convolutional neural networks making wrong decisions.

1 INTRODUCTION

Deep neural networks (DNNs) have made significant progress in classification problems Krizhevsky et al. (2012); Simonyan & Zisserman (2014); Szegedy et al. (2015); He et al. (2016), which have shown to generate good results when provided sufficient data. However, DNNs are found to be easily fooled by adversarial examples generated by adding small and visually imperceptible modifications on normal samples, leading to wrong classification. The existence of adversarial examples reminds us rethinking differences between human visual system and computer vision system based on DNNs.

Many defense methods Bastani et al. (2016); Gu & Rigazio (2014); Huang et al. (2015); Jin et al. (2015); Krizhevsky & Hinton (2009); Kurakin et al. (2016); Shaham et al. (2015); Zheng et al. (2016) are proposed to make neural networks more robust to adversarial examples.

Recently, improving DNNs' robustness to adversarial examples has attracted the attention of many researchers. Several defense methods are proposed to classify adversarial examples correctly, while most of these methods are easily to be attacked as well.

Detection on adversarial examples is another defense task focusing on distinguish between clean samples and adversarial samples Feinman et al. (2017); Bhagoji et al. (2017); Gong et al. (2017); Grosse et al. (2017); Metzen et al. (2017a); Dan & Gimpel (2017); Li & Li (2016). By assuming that adversarial dataset and origin dataset are intrinsically different, classifiers are trained to determine if a sample is clean or adversarial. However, these detection are can be easily destroyed by constructing a differentiable function that is minimized for fooling both classifier and detector with strong iterative attacks.

In this work, we adopt saliency, explaining how a classification DNN can be queried about the spatial support of a particular class in a given image, to tackle with detecting adversarial examples. To calculate saliency for an output w.r.t. input image, we use calculations with gradients to figure out importance of each individual pixels which is meant to reflect their influence on the final classification. Notice that a model learns wrong classification output always learns wrong features and wrong saliency as well. Using the DNN's intrinsic quality that adversarial samples don't completely match it's saliency guides us training a binary classifier to know whether a given sample is real or fake.

2 BACKGROUND

In this section, we introduce notations that are used for analyzing adversarial detection problem, introduce 4 attack methods and 3 defense methods, and introduce image-specific class saliency.

2.1 NOTATION

Formally, given an image x with ground truth $y = f_\theta(x)$, non-targeted adversarial example x^* targeted adversarial example x_t^* for target t are suppose to satisfy the following constraints:

$$f_\theta(x^*) \neq y \tag{1}$$

$$f_\theta(x_t^*) = t \tag{2}$$

$$d(x, x^*) \leq B \tag{3}$$

where function d denote distance metric to quantify similarity and B denote upper bound of allowed perturbation ϵ to origin image.

In the case of DNNs, the classification model f_θ is a highly non-linear function. To seek out which pixels leading to wrong classification when given adversarial sample, f_θ is usually approximated as a linear function:

$$f_\theta(x) = \theta_w x + \theta_b \tag{4}$$

The image-specific class saliency can be calculated as the derivative of f_θ w.r.t. input at the image x.

$$\theta_w = \frac{\partial f_\theta(x)}{\partial x} \tag{5}$$

The computation of the image-specific saliency map for a single class is extremely quick, since it only requires a single back-propagation pass.

2.2 CRAFTING ADVERSARIAL EXAMPLES

Fast Gradient Sign Methods (FGSM) and Iterated Fast Gradient Sign Methods. Goodfellow et al. (2014) proposed a simple gradient based algorithm to generate adversarial examples. With a hyper-parameter step-width ε, adversarial example can be generated by performing one step in the direction of the gradients sign:

$$x^* = x + \varepsilon \cdot sign(\frac{\partial f_\theta(x)}{\partial x}) \tag{6}$$

FGSM is a weak attack which is not designed for generating the minimal adversarial perturbations. Kurakin et al. (2016) introduced an iterative version of the fast gradient sign methods, where replace step-width ε with multiple smaller steps α and setting clip value ε for accumulated perturbations in all iterations. Iterated FGSM start by setting $x_0^* = x$, and for each iteration i computing x_0^* with:

$$x_i^* = clip_\varepsilon(x_{i-1}^* + \alpha \cdot sign(\frac{\partial f_\theta(x)}{\partial x})) \tag{7}$$

Jacobian-based Saliency Map Approach (JSMA). Papernot et al. (2015) proposed a greedy algorithm using the Jacobian to determine choosing which pixel to be perturbed.

$$s_t = \frac{\partial t}{\partial x_i}; s_o = \sum_{j \neq t} \frac{\partial j}{\partial x_i}; s(x_i) = s_t \left| s_o \right| \cdot (s_t < 0) \cdot (s_o > 0) \tag{8}$$

In Equation 8, s_t represents the Jacobian of target class t w.r.t. input image and s_o represents sum of Jacobian values of all non-target class. Changing the selected pixel will significantly increase the likelihood of the model labeling the image as the target class. Clearly, JSMA attack works towards optimizing the L_0 distance metric.

C&W's Attack. Carlini & Wagner (2017b) proposed an attack by approximating the solution to the following optimization problem:

$$argmin(d(s, x+\delta) + c \cdot l(x+\delta)) \tag{9}$$

where l is objective function for solving $f(x+\delta) = t$. In this work, we choose $l(x^* = max(max(Z(x^*)_i : i \neq t) - Z(x^*)_t, -\kappa)$, where κ is the hyper-parameter controlling the confidence of misclassification.

2.3 Detecting Adversarial Examples

Grosse et al. (2017) train a "$N+1$" classification model D to detect adversarial examples with the method of adding these samples to the training set, assigning a new $N+1st$ label for them. However, experiments in Carlini & Wagner (2017a) shows that this detection failed distinguishing adversarial examples at nearly 0% accuracy under a second round attack. Experiment in in Carlini & Wagner (2017a) also shows that this detection methods cannot resist black-box attack where attackers have no access to D. By splitting training set in half for individually training two models, D and imitated D, C&W's Attack succeed 98% when fooling D using parameters for attacking imitated D.

Gong et al. (2017) construct a "$1+1$" classification model by means of regarding real data and fake data as two completely different datasets despite being visually similar. Because of the intrinsic similarity between"$N+1$" detection model and "$1+1$" detection model, this method also failed at second round attack in nearly 0% accuracy for detecting adversarial examples. Black-box attack on "$1+1$" doesn't show significant difference with "$N+1$" .

Metzen et al. (2017b) augment the base network by adding subnetworks as branches at some layers and produce an output $p_{adv} \in [0, 1]$ representing the probability of the input being adversarial. By training the subnetworks with a balanced binary classification dataset consist of clean data and fake data generated by attacking freezed base network, the subnetwork can detect adversarial examples at the inner convolutional layers of the network. Similar to above two second round attacking methods, Metzen et al. (2017b) propose an iterative calculating methods:

$$x_0^{adv} = x$$

$$x_{n+1}^{adv} = clip_x^{\epsilon}\left\{x_n^{adv} + \alpha[(1-\sigma) \cdot sgn(\nabla_x J_f(x_n^{adv}, y_{true}(x))) + \sigma \cdot sgn(\nabla_x J_d(x_n^{adv}, 1))]\right\}$$

Parameter σ is used for trading off objective for base classifier f and objective for detection classifier.

2.4 Gradients as Saliencys

A common approach to understanding the decisions of image classification systems is to find regions of an image that were particularly influential to the final classification Baehrens et al. (2010); Zeiler & Fergus (2014); Springenberg et al. (2014); Zhou et al. (2016); Selvaraju et al. (2016); Zintgraf et al. (2016). At visual level, saliency represents discriminative pixels for model making decisions and Karen Simonyan Simonyan et al. (2013) launch weakly supervised object segmentation experiment only rely on saliency map. Saliency of wrong decision caused by fake sample always visually different from Saliency derived from right sample.

3 Methodology

As is shown in 4, when an image is perturbed by attacking method, saliency of classification output w.r.t. adjusted image is perturbed as well. Accordingly, We follow the steps below building our detection system.

Step1. Train a classifier f with origin training dataset X_{train}, then craft adversarial dataset X_{train}^{adv} and X_{test}^{adv} by attacking f using FGSM/Iterated FGSM/JSMA.

Step2. By calculating saliency for each image in X_{train}, X_{test}, X_{train}^{adv} and X_{test}^{adv} based on the attacked classifier f, we create saliency dataset S_{train}, S_{test}, S_{train}^{adv} and S_{test}^{adv}.

Step3. We apply both raw data and saliency data as input for training binary classifier D. Raw data and saliency data are concatenated on channel axis in our experiment.

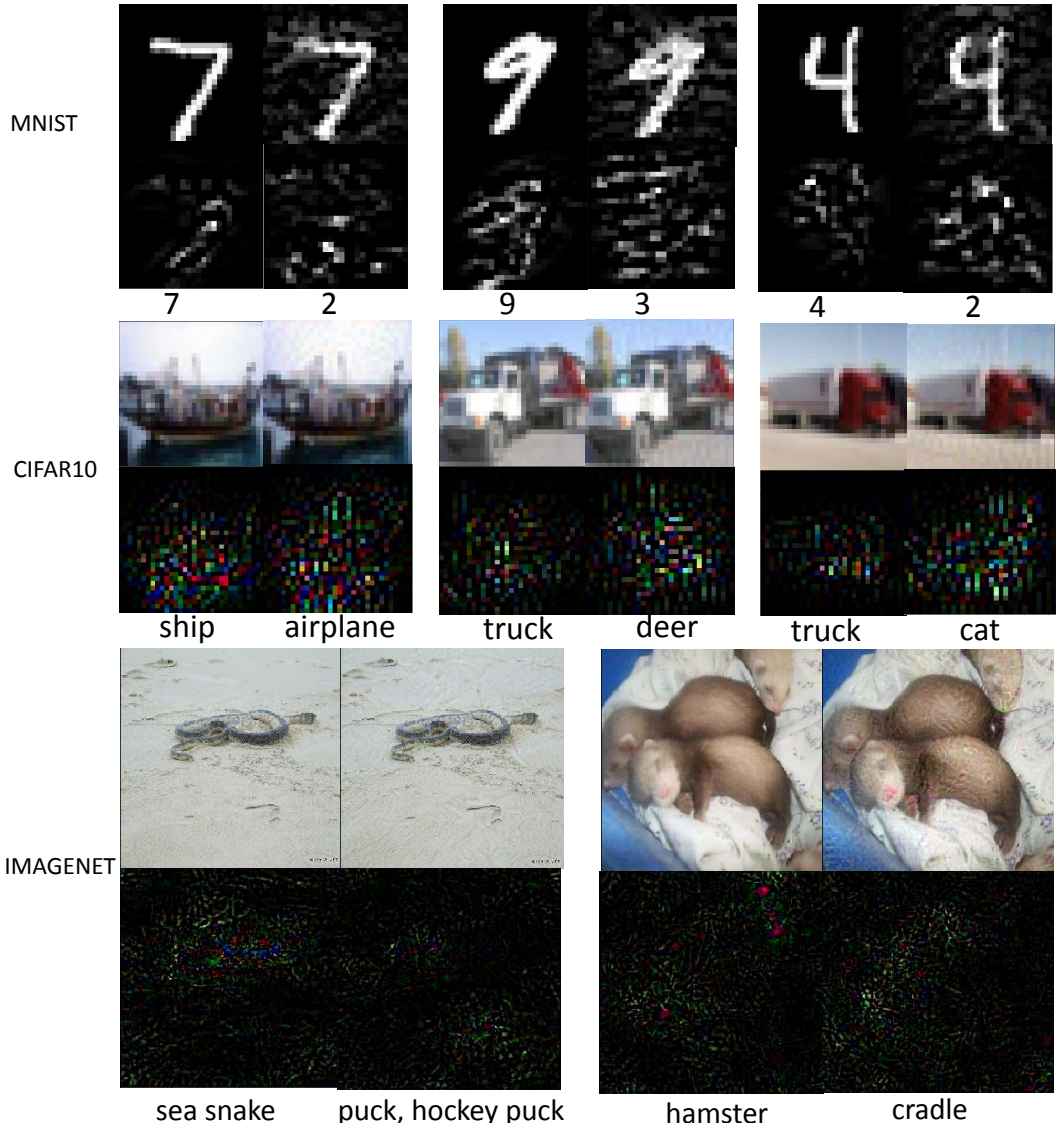

Figure 1: Origin image from MNIST,CIFAR10,IMAGENET dataset and their corresponding saliency. For each four-grids sample, left parts display clean data and right parts display fake data attacked by FGSM. Lower half in four-grids sample represent corresponding saliency for upper half images.

Table 1: Accuracy on adversarial samples generated with FGSM, Iterative l_1 and Iterative l_∞

Dataset	FGSM/ Iterative l_∞/ Iterative l_2			
	$f(x_{test})$	$f(x_{test}^{adv(f)})$	$D(x_{test})$	$D(f(x_{test}^{adv(f)}))$
MNIST	0.99/ 0.99/ 0.99	0.12/ 0.05/ 0.04	1.00 / 1.00 / 0.99	0.99/ 1.00/ 1.00
CIFAR10	0.81/ 0.81/ 0.81	0.13/ 0.07/ 0.07	0.98/ 0.98/ 0.91	0.98/ 0.98/ 0.91
10-IMAGENET	0.90/ 0.90/ 0.90	0.17/ 0.09/ 0.12	0.92/ 0.91/ 0.93	0.90/ 0.91/ 0.94

We evaluate false positive and true positive rates of detector. Furthermore, we evaluate two kinds of generalizability of D: 1) Attacked by the same adversary with different ϵ and 2) Attacked by one adversary when tested on data from other adversaries when fixing ϵ.

4 EXPERIMENT

In this section, we present result of accuracy on detecting adversarial samples generated with FGSM, Iterative l_1 and Iterative l_∞ with 3 dataset: MNIST, CIFAR10, IMAGENET subset Russakovsky et al. (2015). We evaluate generalizability of D for the same attack on f with different choices of ϵ. We also evaluate generalizability of D for the same perturbation extent ϵ with different attacking methods on f.

4.1 IMPLEMENTATION DETAILS

Our experiment is implemented in Keras 2.0 and tensorflow 1.0Abadi et al. (2016). Deep neural networks we adopt for Classifer f network and Detector D network are showed in Figure.2. For MNIST/CIFAR10 dataset, Detector(D) network is smaller than Classifier network since intuitively adversarial binary classification task extract less features. Besides, all DNNs for MNIST/CIFAR10 datasets are trained from scratch. We follow Metzen et al. (2017b) dataset collecting method, randomly selecting 10 classes from Imagenet training set and validation set, The random selected classes are: mongoose; plant, flora, plant life; Yawl; timber wolf, grey wolf, gray wolf, Canis lupus; dugong, Dugong dugon; hammer; sunglasses, dark glasses, shades; typewriter keyboard; triumphal arch; mushroom. Therefore, We have 10000 images in train set, 3035 images in validation set and 500 images(from ImageNets validation data) in test set. The motivation of using subset instead of full-dataset is of two-fold: 1) to reduce computation cost of crafting adversarial dataset, 2) to avoid adversarial conversion between similar classes, eg. perturbing image recogonized as sea snake to image recognized as water snake is not constructive. We employ VGG16 and its parameters from Caffe model zoo on initializing f and D for 10-CLASSES IMAGENET.

We employ 4 typical attacking algorithms in this paper: FGSM, Iterative FGSM with l_2 distance, Iterative FGSM with l_∞ distance and JSMA. We revise origin FGSM to avoid label leaking problemKurakin et al. (2016). JSMA is not applied to Imagenet subset for its' low efficiency on pixel searching when attacking images of size 224*224*3.

4.2 MNIST/CIFAR10

We train MNIST-NET-f shown in Figure 2 for 10 epochs with Adam optimizerKingma & Ba (2014) and learning rate was set to 0.001. MNIST-NET-f run up to 99.73% and 99.32% accuracy on training data and test data respectively. Afterwards, adversarial dataset was generated with 4 attacks. With clean data and adversarial data, we calculate saliency maps for all images. MNIST-NET-D are trained for 10 epochs with Adam optimizer where learning rate was set to 0.0001. CIFAR10-NET-f are trained for 100 epochs with Adam optimizer where learning rate was set to 0.0001, CIFAR10-NET-f run up to 83.89% and 81.32% accuracy on training data and test data respectively. CIFAR10-NET-D are trained for 5 epochs with Adam optimizer where setting learning rate as 0.0001. False positive and True positive rates of MNIST-NET-D and CIFAR10-NET-D are shown in table 1.

Results in Figure 3 show similar performance of generalizability where a D trained with large ϵ cannot reach a good effect on adversarial samples generated with small ϵ. Meanwhile, D trained with adversarial samples crafted with small ϵ generalized acceptably well to all adversarial samples.

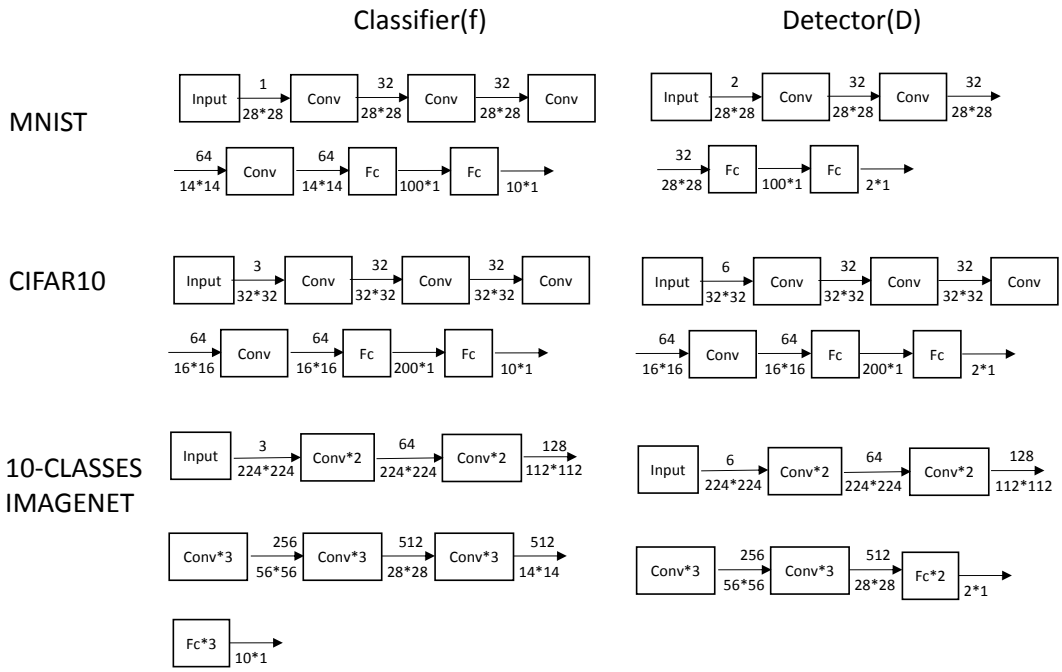

Figure 2: Deep neural network used in our implementation for different datasets, called MNIST-NET-f, MNIST-NET-D, CIFAR10-NET-f, CIFAR10-NET-D, VGG16-f and VGG16-D in following passage. MNIST-NET-f, MNIST-NET-D, CIFAR10-NET-f, CIFAR10-NET-D are trained from scratch, and left two are finetuned with VGG parameters from Caffe Model Zoo. All pooling operations and activations are set to maxpooling and relu respectively, which are not shown in this figure

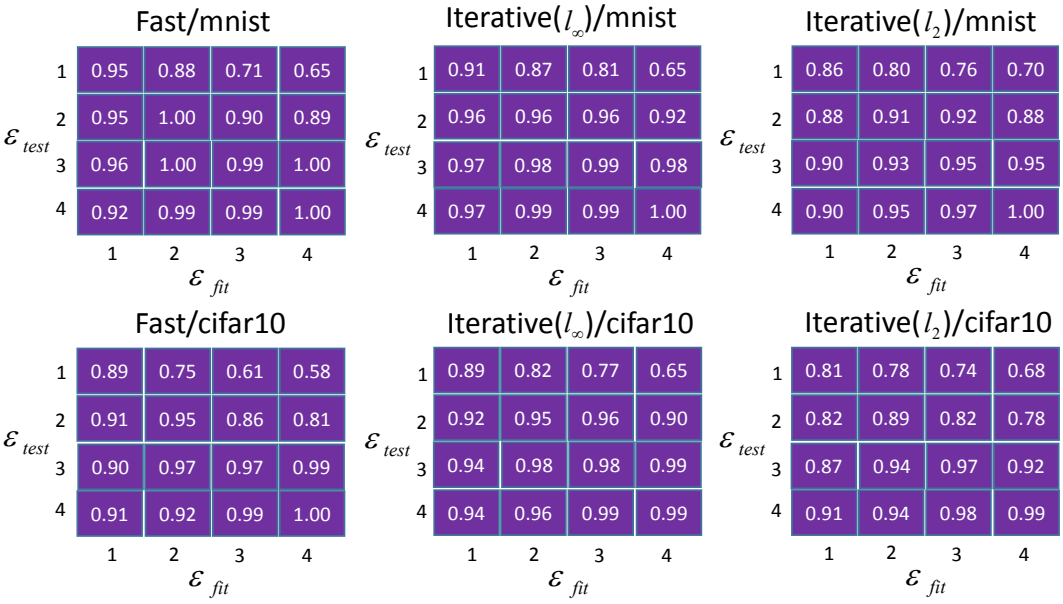

Figure 3: Accuracy metric on MNIST/CIFAR10 of detector trained for adversary with maximal distortion ϵ_{fit} when tested on the same adversary with distortion ϵ_{test}.

Following Metzen et al. (2017b), We set ϵ as minimal under the constraint that the classification accuracy is below 30%. Result in Figure 4 shows that FGSM and JSMA generalized not good enough with detector trained with iterative(l_2) and detector trained iterative(l_∞) , but iterative(l_2) based detector and iterative(l_∞) based detector perform well to FGSM-based adversaries and JSMA-based

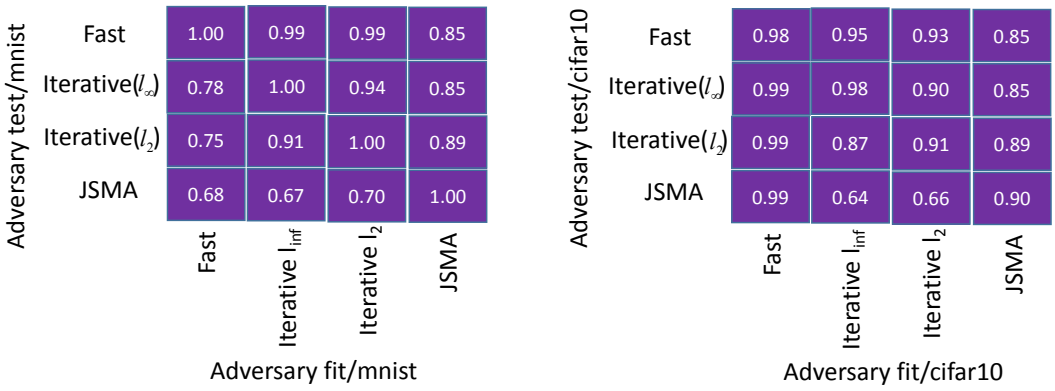

Figure 4: Accuracy metric on MNIST/CIFAR10 of detector trained for one adversary when tested on other adversaries. The maximal distortion of the adversary (when applicable) has been chosen minimally such that the predictive accuracy of the classifier is below 30%. Numbers correspond to the accuracy of the detector on unseen test data.

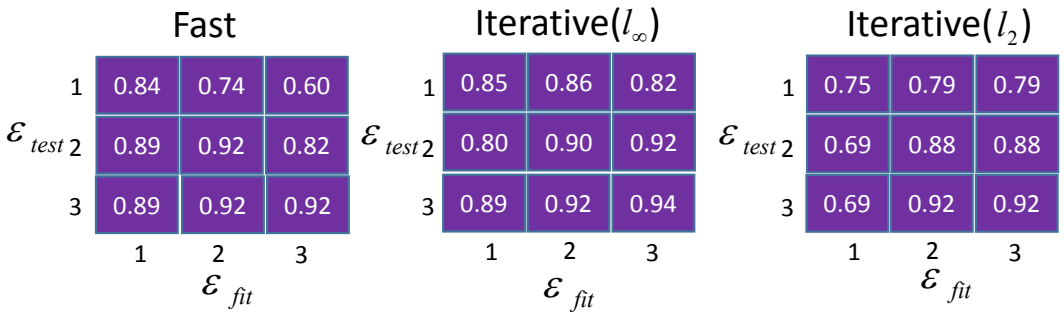

Figure 5: Accuracy metric on IMAGENET subset of detector trained for adversary with maximal distortion when tested on the same adversary with distortion test. Evaluation method is the same as MNIST/CIFAR10 evaluation settings.

adversaries. CIFAR10 dataset show similar character with MNIST experiment except that JSMA and FGSM cannot generalized well to each other. Therefore, we draw the conclusion for our detection approach that stronger adversary generalize well to the weak adversary since iterated method is stronger than fast method to some extent and JSMA optimize loss function under l_0 distance metrics where concentrated on perturbing small group of pixels severely, leading to incompatible results with other three adversary.

4.3 IMAGENET SUBSET

In this section, we concentrated on studying one question: if our detection approach could perform well on eye-level images. Empirically, adversarial examples on MNIST/CIFAR10 usually show visually distinguishable perturbation even texture and structure of origin image are changed. Therefore, many researches on defending MNIST/CIFAR10-level adversary helps little to find out the extrinsic difference between human visual system and deep neural networks. Take MNIST adversary for example, saliency of wrong output w.r.t. adversarial example seems visually approximate to its' perturbation. However, in Imagenet-level images, these unreasonable properties found on MNIST/CIFAR10-level no longer appear.

In this experiment, we use only 3 attacking methods: FGSM, Iterative(l_1) and Iterative(l_∞) for their suitable demand for computation recourses. We fine-tuning VGG16-f shown in Figure 2 for 10000 epochs with Adam optimizerKingma & Ba (2014). Initial learning rate was set to 0.001, reduced to 0.0001 after 100 epochs, and further reduced to 0.00001 after 1000 epochs. VGG16-f run up to 91.82% and 89.83% accuracy on training data and test data respectively. VGG16-D are trained

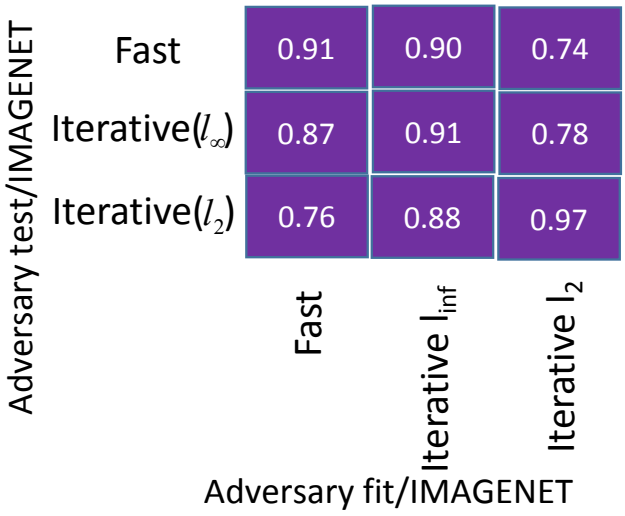

Figure 6: Accuracy metric on IMAGENET subset of detector trained for one adversary when tested on other adversaries. Evaluation method is the same as MNIST/CIFAR10 evaluation settings.

for 100 epochs with Adam optimizer where learning rate was set to 0.0001.False positive and True positive rates of VGG16-D are shown in table 1.

Results in Figure 5 shows similar direction with MNIST/CIFAR10 experiment: detectors trained with smaller perturbation upper-bound generally perform well on higher ones but not vice versa. Results in Figure 6 shows that detector trained with stronger adversaries generalize well to detector trained with weaker adversaries, which is identical to MNIST/CIFAR10 evaluations.

5 Discussion

When we dive into the feature extraction procedure of deep convolutional neural networks, saliency seems to be semantic enough but not express enough features. We dissect deep convolutional network with a revised version of Grad-CAM to find out how adversarial examples contributing to wrong output classification.

We generalize interpretability of saliency maps to each layers by computing gradient of output w.r.t. feature maps in certain layer as feature map weights α. Intuitively, α represents influence of a feature map for the final decision. Weighted summation feature maps in Figure 7 are generated referring to Selvaraju et al. (2016). Weighted summation feature maps in 'relu2-1', 'relu4-1' and 'pooling5' in VGG16 model. These three Weighted feature maps roughly represent feature extracted by shallow layers, middle layers and deep layers. Visualization shows that shallow layers are robust enough to adversarial examples while middle layers start to extract wrong features, leading to deep layers' failure.

6 Conclusion

We have proposed a approach for detecting adversarial examples by training a binary classifier by taking saliency perturbation information into consideration. Our approach shows 100% accuracy on detecting adversarial perturbations on MNIST dataset and show above 90% accuracy on CIFAR10, IMAGENET subset under FGSM/Iterative(l_2)/Iterative(l_∞),JSMA attack. By quantitatively evaluate generalization ability of the detector, we conclude that our detector trained with strong adversaries performs well on weak adversaries, proving its' generalizability and transferability. Afterwards,, we further discuss relationship between solving adversary problem and model interpretation, claiming that shallow layers are robust to adversarial attack and middle layers start calculating wrong decisions.

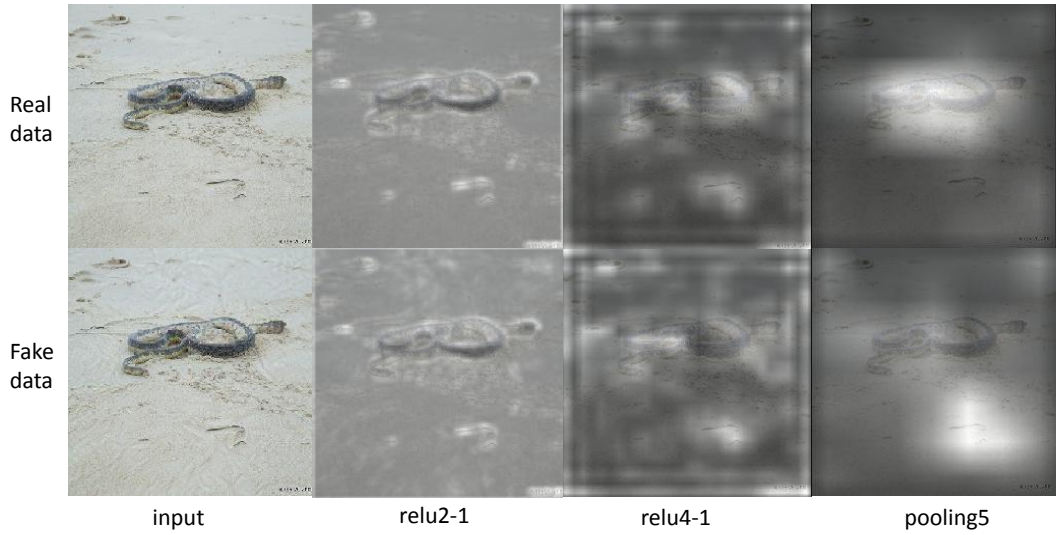

Figure 7: Weighted summation of feature maps in 'relu2-1', 'relu4-1' and 'pooling5' in VGG16 model. These three Weighted summation of feature maps roughly represent feature extracted by shallow layers, middle layers and deep layers. Visualization shows that shallow layers are robust enough to adversarial examples while middle layers start to extract wrong features, leading to deep layers' failure.

REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. Measuring neural net robustness with constraints. 2016.

Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. 2017.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *arXiv preprint arXiv:1705.07263*, 2017a.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017b.

Hendrycks Dan and Kevin Gimpel. Early methods for detecting adversarial images. 2017.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*, 2017.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017.

Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *Computer Science*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ruitong Huang, Bing Xu, Dale Schuurmans, and Csaba Szepesvari. Learning with a strong adversary. *Computer Science*, 2015.

Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello. Robust convolutional neural networks under adversarial noise. *Computer Science*, 2015.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. 2016.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. 2017a.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *stat*, 1050:21, 2017b.

Nicolas Papernot, Patrick Mcdaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. pp. 372–387, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. *Computer Science*, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Computer Vision and Pattern Recognition*, pp. 4480–4488, 2016.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.

Luisa M Zintgraf, Taco S Cohen, and Max Welling. A new method to visualize deep neural networks. *arXiv preprint arXiv:1603.02518*, 2016.