ADVERSARIAL DEFENSE VIA DATA DEPENDENT ACTIVATION FUNCTION AND TOTAL VARIATION MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We improve the robustness of deep neural nets to adversarial attacks by using an interpolating function as the output activation. This data-dependent activation function remarkably improves both classification accuracy and stability to adversarial perturbations. Together with the total variation minimization of adversarial images and augmented training, under the strongest attack, we achieve up to 20.6%, 50.7%, and 68.7% accuracy improvement with respect to the fast gradient sign method, iterative fast gradient sign method, and Carlini-Wagner L_2 attacks, respectively. Our defense strategy can be added to many of the existing methods. We give an intuitive explanation of our defense strategy via analyzing the geometry of the feature space. For reproducibility, the code will be available on GitHub.

1 INTRODUCTION

The adversarial vulnerability (Szegedy et al., 2013) of deep neural nets (DNNs) threatens their applicability in security critical tasks, e.g., autonomous cars (Akhtar & Mian, 2018), robotics (Guisti et al., 2016), DNN-based malware detection systems (Papernot et al., 2016b; Grosse et al., 2016). Since the pioneering work by Szegedy et al. (2013), many advanced adversarial attack schemes have been devised to generate imperceptible perturbations to sufficiently fool the DNNs (Goodfellow et al., 2014; Papernot et al., 2016a; Carlini & Wagner, 2016; Wu et al., 2018; Ilyas et al., 2018; Athalye et al., 2018b). And not only are adversarial attacks successful in white-box attacks, i.e. when the adversary has access to the DNN parameters, but they are also successful in black-box attacks, i.e. it has no access to the parameters. Black-box attacks are successful because one can perturb an image so it misclassifies on one DNN, and the same perturbed image also has a significant chance to be misclassified by another DNN; this is known as transferability of adversarial examples (Papernot et al. (2016d)). Due to this transferability, it is very easy to attack neural nets in a black-box fashion (Liu et al., 2016; Brendel et al., 2017). In fact, there exist universal perturbations that can imperceptibly perturb any image and cause misclassification for any given network (Moosavi-Dezfooli et al. (2017)). There is much recent research on designing advanced adversarial attacks and defending against adversarial perturbation.

In this work, we propose to defend against adversarial attacks by changing the DNNs' output activation function to a manifold-interpolating function, in order to seamlessly utilize the training data's information when performing inference. Together with the total variation minimization (TVM) and augmented training, we show state-of-the-art defense results on the CIFAR-10 benchmark. Moreover, we show that adversarial images generated from attacking the DNNs with an interpolating function are more transferable to other DNNs, than those resulting from attacking standard DNNs.

2 RELATED WORK

Defensive distillation was recently proposed to increase the stability of DNNs which dramatically reduces the success rate of adversarial attacks (Papernot et al., 2016c), and a related approach (Tramr et al. (2018)) cleverly modifies the training data to increase robustness against black-box attacks, and adversarial attacks in general. To counter the adversarial perturbations, Guo et al. (2018) proposed to use image transformation, e.g., bit-depth reduction, JPEG compression, TVM, and image quilting. A similar idea of denoising the input was later explored by Moosavi-Dezfooli et al. (2018), where they divide the input into patches, denoise each patch, and then reconstruct the image. These input transformations are intended to be non-differentiable, thus making adversarial attacks more difficult, especially for gradient-based attacks. Song et al. (2018) noticed that small adversarial perturbations shift the distribution of adversarial images far from the distribution of clean images. Therefore they proposed to purify the adversarial images by PixelDefend. Adversarial training is another family of defense methods to improve the stability of DNNs (Goodfellow et al., 2014; Mardy et al., 2018; Na et al., 2018). And GANs are also employed for adversarial defense (Samangouei et al., 2018). In (Athalye et al., 2018a), the authors proposed a straight-through estimation of the gradient to attack the defense methods that is based on the obfuscated gradient. Meanwhile, many advanced attack methods have been proposed to attack the DNNs (Wu et al., 2018; Ilyas et al., 2018).

Instead of using softmax functions as the DNNs' output activation, Wang et al. (2018) utilized a class of non-parametric interpolating functions. This is a combination of both deep and manifold learning which causes the DNNs to sufficiently utilize the geometric information of the training data. The authors show a significant amount of generalization accuracy improvement, and the results are more stable when one only has a limited amount of training data.

3 Deep Neural Nets with Data-Dependent Activation Function

In this section, we summarize the architecture, training, and testing procedures of the DNNs with the data-dependent activation (Wang et al., 2018). An overview of training and testing of the standard DNNs with softmax output activation is shown in Fig. 1 (a) and (b), respectively. In the kth iteration of training, given a mini-batch of training data $\mathbf{X}$, $\mathbf{Y}$, the procedure is:

Forward propagation: Transform $\mathbf{X}$ into features by a DNN block (ensemble of convolutional layers, nonlinearities and others), and then pass this output through the softmax activation to obtain the predictions $\tilde{\mathbf{Y}}$:

$$\tilde{\mathbf{Y}} = \mathrm{Softmax}(\mathrm{DNN}(\mathbf{X}, \Theta^{k-1}), \mathbf{W}^{k-1}).$$

Then the loss is computed (e.g., cross entropy) between $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$: $\mathcal{L} = \mathrm{Loss}(\mathbf{Y}, \tilde{\mathbf{Y}})$.

Backpropagation: Update weights (Θ^{k-1}, $\mathbf{W}^{k-1}$) by gradient descent (learning rate γ):

$$\mathbf{W}^k = \mathbf{W}^{k-1} - \gamma \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{Y}}} \cdot \frac{\partial \tilde{\mathbf{Y}}}{\partial \mathbf{W}}, \quad \Theta^k = \Theta^{k-1} - \gamma \frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{Y}}} \cdot \frac{\partial \tilde{\mathbf{Y}}}{\partial \tilde{\mathbf{X}}} \cdot \frac{\partial \tilde{\mathbf{X}}}{\partial \Theta}.$$

Once the model is optimized, the predicted labels for testing data $\mathbf{X}$ are:

$$\tilde{\mathbf{Y}} = \mathrm{Softmax}(\mathrm{DNN}(\mathbf{X}, \Theta), \mathbf{W}).$$

Wang et al. (2018) proposed to replace the data-agnostic softmax activation by a data-dependent interpolating function, defined in the next section.

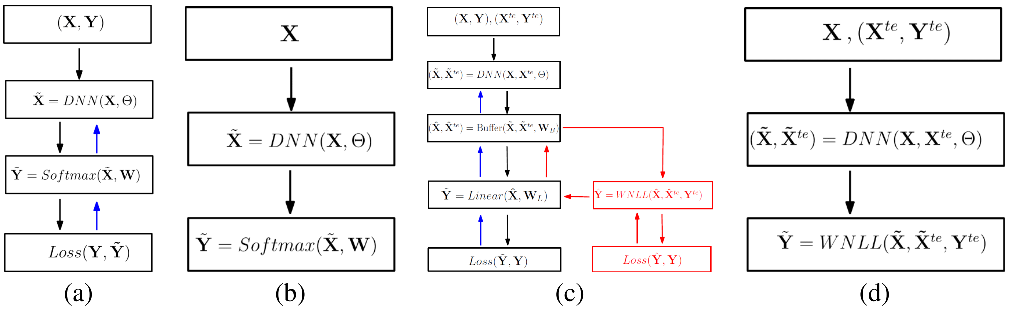

$(\mathbf{X}, \mathbf{Y})$
$\tilde{\mathbf{X}} = DNN(\mathbf{X}, \Theta)$
$\tilde{\mathbf{Y}} = Softmax(\tilde{\mathbf{X}}, \mathbf{W})$
$Loss(\mathbf{Y}, \tilde{\mathbf{Y}})$
(a)
$\mathbf{X}$
$\tilde{\mathbf{X}} = DNN(\mathbf{X}, \Theta)$
$\tilde{\mathbf{Y}} = Softmax(\tilde{\mathbf{X}}, \mathbf{W})$
(b)
$(\mathbf{X}, \mathbf{Y}), (\mathbf{X}^{te}, \mathbf{Y}^{te})$
$(\tilde{\mathbf{X}}, \tilde{\mathbf{X}}^{te}) = DNN(\mathbf{X}, \mathbf{X}^{te}, \Theta)$
$(\hat{\mathbf{X}}, \hat{\mathbf{X}}^{te}) = \mathrm{Buffer}(\tilde{\mathbf{X}}, \tilde{\mathbf{X}}^{te}, \mathbf{W}_B)$
$\tilde{\mathbf{Y}} = Linear(\hat{\mathbf{X}}, \mathbf{W}_L)$
$\hat{\mathbf{Y}} = WNLL(\hat{\mathbf{X}}, \hat{\mathbf{X}}^{te}, \mathbf{Y}^{te})$
$Loss(\tilde{\mathbf{Y}}, \mathbf{Y})$
$Loss(\hat{\mathbf{Y}}, \mathbf{Y})$
(c)
$\mathbf{X}, (\mathbf{X}^{te}, \mathbf{Y}^{te})$
$(\tilde{\mathbf{X}}, \tilde{\mathbf{X}}^{te}) = DNN(\mathbf{X}, \mathbf{X}^{te}, \Theta)$
$\hat{\mathbf{Y}} = WNLL(\tilde{\mathbf{X}}, \tilde{\mathbf{X}}^{te}, \mathbf{Y}^{te})$
(d)

Figure 1: Training and testing procedures of the DNNs with softmax and WNLL functions as the last activation layer. (a) and (b) show the training and testing steps for the standard DNNs, respectively; (c) and (d) illustrate the training and testing procedure of the WNLL activated DNNs, respectively.

3.1 Manifold Interpolation - A Harmonic Extension Approach

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ be a set of points in a high dimensional manifold $\mathcal{M} \subset \mathbb{R}^d$ and $\mathbf{X}^{\rm te} = \{\mathbf{x}_1^{\rm te}, \mathbf{x}_2^{\rm te}, \cdots, \mathbf{x}_m^{\rm te}\}$ be a subset of $\mathbf{X}$ which are labeled with label function $g(\mathbf{x})$. We want to interpolate a function u that is defined on the entire manifold and can be used to label the entire dataset $\mathbf{X}$. The harmonic extension is a natural and elegant approach to find such an interpolating function, which is defined by minimizing the Dirichlet energy functional:

$$\mathcal{E}(u) = \frac{1}{2}\sum_{\mathbf{x},\mathbf{y}\in\mathbf{X}} w(\mathbf{x},\mathbf{y})\left(u(\mathbf{x}) - u(\mathbf{y})\right)^2, \tag{1}$$

with the boundary condition:

$$u(\mathbf{x}) = g(\mathbf{x}),\ \mathbf{x} \in \mathbf{X}^{\rm te},$$

where $w(\mathbf{x},\mathbf{y})$ is a weight function, typically chosen to be Gaussian: $w(\mathbf{x},\mathbf{y}) = \exp(-\frac{||\mathbf{x}-\mathbf{y}||^2}{\sigma^2})$ with σ being a scaling parameter. The Euler-Lagrange equation for Eq. (1) is:

$$\begin{cases} \sum_{\mathbf{y}\in\mathbf{X}} \left(w(\mathbf{x},\mathbf{y}) + w(\mathbf{y},\mathbf{x})\right)\left(u(\mathbf{x}) - u(\mathbf{y})\right) = 0 & \mathbf{x} \in \mathbf{X}/\mathbf{X}^{\rm te} \\ u(\mathbf{x}) = g(\mathbf{x}) & \mathbf{x} \in \mathbf{X}^{\rm te}. \end{cases} \tag{2}$$

By solving the linear system (Eq. (2)), we obtain labels $u(\mathbf{x})$ for unlabeled data $\mathbf{x} \in \mathbf{X}/\mathbf{X}^{\rm te}$. This interpolation becomes invalid when the labeled data is tiny, i.e., $|\mathbf{X}^{\rm te}| \ll |\mathbf{X}/\mathbf{X}^{\rm te}|$. To resolve this issue, the weights of the labeled data is increased in the Euler-Lagrange equation, which gives:

$$\begin{cases} \sum_{\mathbf{y}\in\mathbf{X}} \left(w(\mathbf{x},\mathbf{y}) + w(\mathbf{y},\mathbf{x})\right)\left(u(\mathbf{x}) - u(\mathbf{y})\right) + \\ \left(\frac{|\mathbf{X}|}{|\mathbf{X}^{\rm te}|} - 1\right)\sum_{\mathbf{y}\in\mathbf{X}^{\rm te}} w(\mathbf{y},\mathbf{x})\left(u(\mathbf{x}) - u(\mathbf{y})\right) = 0 & \mathbf{x} \in \mathbf{X}/\mathbf{X}^{\rm te} \\ u(\mathbf{x}) = g(\mathbf{x}) & \mathbf{x} \in \mathbf{X}^{\rm te}. \end{cases} \tag{3}$$

The solution $u(\mathbf{x})$ to Eq. (3) is named weighted nonlocal Laplacian (WNLL), denoted as $\mathrm{WNLL}(\mathbf{X}, \mathbf{X}^{\rm te}, \mathbf{Y}^{\rm te})$. For classification tasks, $g(\mathbf{x})$ is the one-hot labels for the example $\mathbf{x}$.

3.2 Training and Testing the DNNs with Data-Dependent Activation Function

In both training and testing of the WNLL-activated DNNs, we need to reserve a small portion of data/label pairs, denoted as $(\mathbf{X}^{\rm te}, \mathbf{Y}^{\rm te})$, to interpolate the label for new data $\mathbf{Y}$. We name the reserved data $(\mathbf{X}^{\rm te}, \mathbf{Y}^{\rm te})$ as the template. Directly replacing softmax by WNLL has difficulties in back propagation, namely the true gradient $\frac{\partial \mathcal{L}}{\partial \Theta}$ is difficult to compute since WNLL defines a very complex implicit function. Instead, to train WNLL-activated DNNs, a proxy via an auxiliary neural net (Fig.1(c)) is employed. On top of the original DNNs, we add a buffer block (a fully connected layer followed by a ReLU), and followed by two parallel branches, WNLL and the linear (fully connected) layers. The auxiliary DNNs can be trained by alternating between training DNNs with linear and WNLL activations, respectively. The training loss of the WNLL activation function is backpropped via a straight-through estimation approach (Athalye et al., 2018a; Bengio et al., 2013). At test time, we remove the linear classifier from the neural nets and use the DNN and buffer blocks together with WNLL to predict new data (Fig. 1 (d)); here for simplicity, we merge the buffer block to the DNN block. For a given set of testing data $\mathbf{X}$, and the labeled template $\{(\mathbf{X}^{\rm te}, \mathbf{Y}^{\rm te})\}$, the predicted labels for $\mathbf{X}$ is given by

$$\tilde{\mathbf{Y}} = \mathrm{WNLL}(\mathrm{DNN}(\mathbf{X}, \mathbf{X}^{\rm te}, \Theta), \mathbf{Y}^{\rm te}).$$

4 Adversarial Attacks

We consider three benchmark attack methods in this work, namely, the fast gradient sign method (FGSM) (Goodfellow et al., 2014), iterative FGSM (IFGSM) (Kurakin et al., 2016), and Carlini-Wagner's L_2 (CW-L2) (Carlini & Wagner, 2016) attacks. We denote the classifier defined by the DNNs with softmax activation as $\tilde{y} = f(\theta, \mathbf{x})$ for a given instance $(\mathbf{x}, y)$. FGSM finds the adversarial image $\mathbf{x}'$ by maximizing the loss $\mathcal{L}(\mathbf{x}', y)$, subject to the l_∞ perturbation $||\mathbf{x}'-\mathbf{x}||_\infty \leq \epsilon$ with ϵ as the attack strength. Under the first order approximation i.e., $\mathcal{L}(\mathbf{x}', y) = \mathcal{L}(\mathbf{x}, y) + \nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}, y)^T \cdot (\mathbf{x}' - \mathbf{x})$, the optimal perturbation is given by

$$\mathbf{x}' = \mathbf{x} + \epsilon\,\mathrm{sign} \cdot (\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}, \theta)) \,. \tag{4}$$

IFGSM iterates FGSM to generate enhanced adversarial images, i.e.,

$$\mathbf{x}^{(m)} = \mathbf{x}^{(m-1)} + \epsilon \cdot \mathrm{sign}\left(\nabla_{\mathbf{x}^{(m-1)}}\mathcal{L}(\mathbf{x}^{(m-1)}, y)\right), \tag{5}$$

where $m = 1, \cdots, M$, $\mathbf{x}^{(0)} = \mathbf{x}$ and $\mathbf{x}' = \mathbf{x}^{(M)}$, with M be the number of iterations.

The CW-L2 attack is proposed to circumvent defensive distillation. For a given image-label pair $(\mathbf{x}, y)$, and $\forall t \neq y$, CW-L2 searches the adversarial image that will be classified to class t by solving the optimization problem:

$$\min_{\delta} ||\delta||_2^2, \quad \text{subject to } f(\mathbf{x} + \delta) = t,\ \mathbf{x} + \delta \in [0, 1]^n, \tag{6}$$

where δ is the adversarial perturbation (for simplicity, we ignore the dependence of θ in f).

The equality constraint in Eq. (6) is hard to satisfy, so instead Carlini et al. consider the surrogate

$$g(\mathbf{x}) = \max\left(\max_{i \neq y}(Z(\mathbf{x})_i) - Z(\mathbf{x})_y, 0\right), \tag{7}$$

where $Z(\mathbf{x})$ is the logit vector for an input $\mathbf{x}$, i.e., output of the neural net before the softmax layer. $Z(\mathbf{x})_i$ is the logit value corresponding to class i. It is easy to see that $f(\mathbf{x} + \delta) = t$ is equivalent to $g(\mathbf{x} + \delta) \leq 0$. Therefore, the problem in Eq. (6) can be reformulated as

$$\min_{\delta} ||\delta||_2^2 + c \cdot g(\mathbf{x} + \delta) \quad \text{subject to } \mathbf{x} + \delta \in [0, 1]^n, \tag{8}$$

where $c \geq 0$ is the Lagrangian multiplier.

By letting $\delta = \frac{1}{2}(\tanh(\mathbf{w}) + 1) - \mathbf{x}$, Eq. (8) can be converted to an unconstrained optimization problem. Moreover, Carlini et al. introduce the confidence parameter κ into the above formulation. Above all, CW-L2 attacks seek adversarial images by solving the following problem

$$\min_{\mathbf{w}} ||\frac{1}{2}(\tanh(\mathbf{w}) + 1) - \mathbf{x}||_2^2 + c \cdot \max\left(-\kappa, \max_{i \neq y}(Z(\frac{1}{2}(\tanh(\mathbf{w})) + 1)_i) - Z(\frac{1}{2}(\tanh(\mathbf{w})) + 1)_y\right). \tag{9}$$

This unconstrained optimization problem can be solved efficiently by the Adam optimizer (Kingma & Ba, 2014). All three of the attacks clip the values of the adversarial image $\mathbf{x}'$ to between 0 and 1.

4.1 Adversarial Attack for DNNs with WNLL Activation Function

In this work, we focus on untargeted attacks and defend against them. For a given small batch of testing images $(\mathbf{X}, \mathbf{Y})$ and template $(\mathbf{X}^{\mathrm{te}}, \mathbf{Y}^{\mathrm{te}})$, we denote the DNNs modified with WNLL as output activation as $\tilde{\mathbf{Y}} = \mathrm{WNLL}(Z(\{\mathbf{X}, \mathbf{X}^{\mathrm{te}}\}), \mathbf{Y}^{\mathrm{te}})$, where $Z(\{\mathbf{X}, \mathbf{X}^{\mathrm{te}}\})$ is the composition of the DNN and buffer blocks as shown in Fig. 1 (c). By ignoring dependence of the loss function on the parameters, the loss function for DNNs with WNLL activation can be written as $\tilde{\mathcal{L}}(\mathbf{X}, \mathbf{Y}, \mathbf{X}^{\mathrm{te}}, \mathbf{Y}^{\mathrm{te}})$. The above attacks for DNNs with WNLL activation on the batch of images, $\mathbf{X}$, are formulated below.

- **FGSM**

$$\mathbf{X}' = \mathbf{X} + \epsilon \cdot \mathrm{sign}\left(\nabla_{\mathbf{X}}\tilde{\mathcal{L}}(\mathbf{X}, \mathbf{Y}, \mathbf{X}^{\mathrm{te}}, \mathbf{Y}^{\mathrm{te}})\right). \tag{10}$$

- **IFGSM**

$$\mathbf{X}^{(m)} = \mathbf{X}^{(m-1)} + \epsilon \cdot \mathrm{sign}\left(\nabla_{\mathbf{X}^{(m-1)}}\tilde{\mathcal{L}}(\mathbf{X}^{(m-1)}, \mathbf{Y}, \mathbf{X}^{\mathrm{te}}, \mathbf{Y}^{\mathrm{te}})\right), \tag{11}$$

  where $m = 1, 2, \cdots, N$; $\mathbf{X}^{(0)} = \mathbf{X}$ and $\mathbf{X}' = \mathbf{X}^{(M)}$.

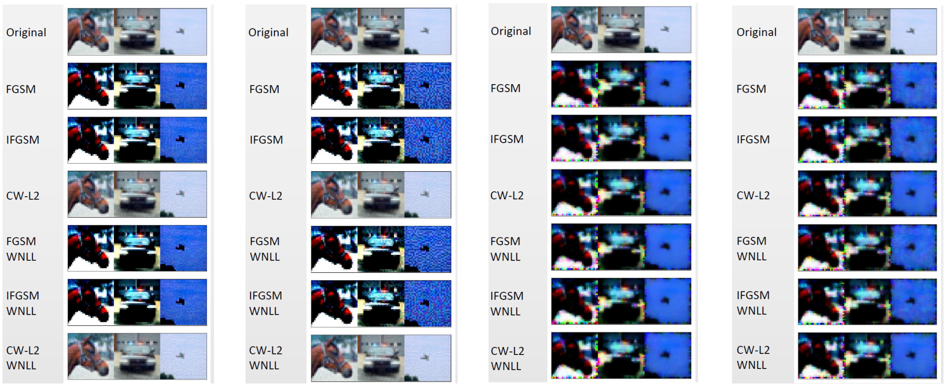
- **CW-L2**

$$\begin{aligned} \min_{\mathbf{W}} ||\frac{1}{2}(\tanh(\mathbf{W}) + 1) - \mathbf{X}||_2^2 + & \\ c \cdot \max\left(-\kappa, \max_{\mathbf{i} \neq \mathbf{Y}}(Z(\frac{1}{2}(\tanh(\mathbf{W})) + 1)_{\mathbf{i}}) - Z(\frac{1}{2}(\tanh(\mathbf{W})) + 1)_{\mathbf{Y}}\right), & \end{aligned} \tag{12}$$

  where $\mathbf{i}$ is the logit values of the input images $\mathbf{X}$.

Based on our numerical experiments, the batch size of $\mathbf{X}$ has minimal influence on the adversarial attack and defense. In all of our experiments we choose the batch size of $\mathbf{X}$ to be 500. Similar to Wang et al. (2018), we choose the size of the template to be 500.

We apply the above attack methods to ResNet-56 (He et al., 2016) with either softmax or WNLL as the output activation function. For IFGSM, we run 10 iterations of Eqs. (5) and (11) to attack DNNs with two different output activations, respectively. For CW-L2 attacks (Eqs. (9, 12)) in both scenarios, we set the parameters $c = 10$ and $\kappa = 0$. Figure 2 depicts three randomly selected images (horse, automobile, airplane) from the CIFAR-10 dataset, their adversarial versions by different attack methods on ResNet-56 with two kinds of activation functions, and the TV minimized images. All attacks successfully fool the classifiers to classify any of them correctly. Figure 2 (a) shows that FGSM and IFGSM with perturbation $\epsilon = 0.02$ changes the contrast of the images, while it is still easy for humans to correctly classify them. The adversarial images of the CW-L2 attacks are imperceptible, however they are extremely strong in fooling DNNs. Figure 2 (b) shows the images of (a) with a stronger attack, $\epsilon = 0.08$. With a larger ϵ, the adversarial images become more noisy. The TV minimized images of Fig. 2 (a) and (b) are shown in Fig. 2 (c) and (d), respectively. The TVM removes a significant amount of detailed information from the original and adversarial images, meanwhile it also makes it harder for humans to classify both the TV-minimized version of the original and adversarial images. Visually, it is hard to discern the adversarial images resulting from attacking the DNNs with two types of output layers.



(a) (b) (c) (d)

Figure 2: Samples from CIFAR-10. Panel (a): from the top to the last rows show the original, adversarial images by attacking ResNet-56 with FGSM, IFGSM, CW-L2 ($\epsilon = 0.02$); and attacking the ResNet-56 with WNLL as output activation. Panel (b) corresponding to those in panel (a) with $\epsilon = 0.08$. Charts (c) and (d) corresponding to the TV minimized images in (a) and (b), respectively.

5 Analysis of the Geometry of Features

We consider the geometry of features of the original and adversarial images. We randomly select 1000 training and 100 testing images from the airplane and automobile classes, respectively. We consider two visualization strategies for ResNet-56 with softmax activation: (1) extract the original 64D features output from the layer before the softmax, and (2) apply the principle component analysis (PCA) to reduce them to 2D. However, the principle components (PCs) do not encode the entire geometric information of the features. Alternatively, we add a 2 by 2 fully connected (FC) layer before the softmax, then utilize the 2D features output from this newly added layer. We verify that the newly added layer does not change the performance of ResNet-56 as shown in Fig. 3, and that the training and testing performance remains essentially the same for these two cases.

Figure 4 (a) and (b) show the 2D features generated by ResNet-56 with additional FC layer for the original and adversarial testing images, respectively, where we generate the adversarial images by using FGSM ($\epsilon = 0.02$). Before adversarial perturbation (Fig. 4 (a)), there is a straight line that can easily separate the two classes. The small perturbation causes the features to overlap and there is no linear classifier that can easily separate these two classes (Fig. 4 (b)). The first two PCs of the 64D features of the clean and adversarial images are shown in Fig. 4 (c) and (d), respectively.

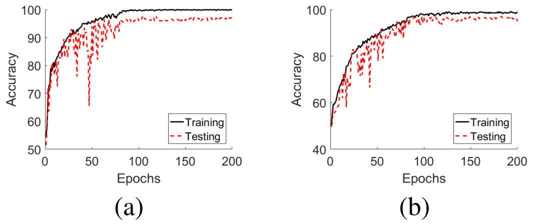


Figure 3: Training and testing epochs v.s. accuracy of ResNet-56 on CIFAR-10. (a): without the additional FC layer; (b): with the additional FC layer.

Again, the PCs are well separated for clean images, while adversarial perturbation causes overlap and concentration.

The bottom charts of Fig. 4 depict the first two PCs of the 64D features output from the layer before the WNLL. The distributions of the unperturbed training and testing data are the same, as illustrated in panels (e) and (f). The new features are better separated which indicates that DNNs with WNLL are more robust to small random perturbation. Panels (g) and (h) plot the features of the adversarial and TV minimized adversarial images in the test set. The adversarial attacks move the automobiles' features to the airplanes' region and TVM helps to eliminate the outliers. Based on our computation, most of the adversarial images of the airplane classes can be correctly classified with the interpolating function. The training data guides the interpolating function to classify adversarial images correctly. The fact that the adversarial perturbations change the features' distribution was also noticed in (Song et al., 2018).

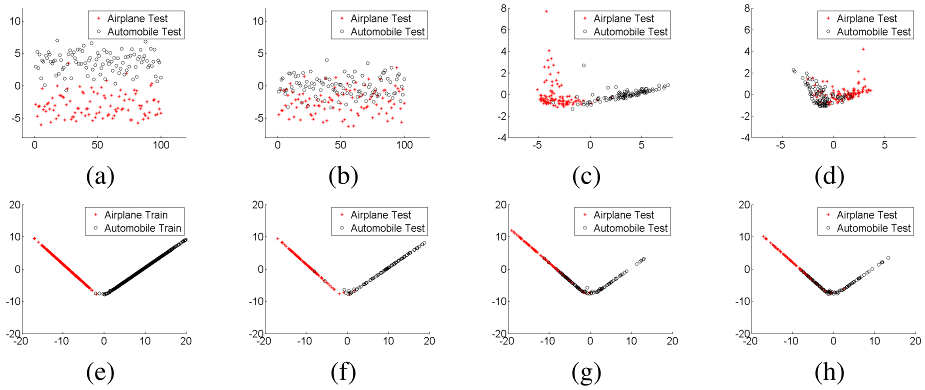


Figure 4: Visualization of the features learned by DNNs with softmax ((a), (b), (c), (d)) and WNLL ((e), (f), (g), (h)) activation functions. (a) and (b) plot the 2D features of the original and adversarial testing images; (c) and (d) are the first two principle components of the 64D features for the original and adversarial testing images, respectively. Charts (e), (f) plot the first two components of the training and testing features learned by ResNet-56 with WNLL activation; (g) and (h) show the two principle components of the adversarial images and TV minimized adversarial images for the test set.

6 Adversarial Defense by Interpolating Function and TVM

To defend against adversarials, we combine the ideas of data-dependent activation, input transformation, and training data augmentation. We train ResNet-56, respectively, on the original training data, the TV minimized training data, and a combination of the previous two. On top of the data-dependent activation output and augmented training, we further apply the TVM (Rudin et al., 1992) used by Guo et al. (2018) to transform the adversarial images to boost defensive performance. The basic idea is to reconstruct the simplest image $\mathbf{z}$ from the sub-sampled image, $X \odot \mathbf{x}$, with X the mask filled by a Bernoulli binary random variable, by solving the following TVM problem

$$\min_{\mathbf{z}} ||(1-X) \odot (\mathbf{z}-\mathbf{x})||_2 + \lambda_{TV} \cdot TV_2(\mathbf{z}),$$

Table 1: Mutual classification accuracy on the adversarial images resulting from attacking ResNet-56 with the softmax and the WNLL activation functions.

Attack Method	Training data	$\epsilon = 0.02$	$\epsilon = 0.04$	$\epsilon = 0.06$	$\epsilon = 0.08$	$\epsilon = 0.1$
Classification accuracy of ResNet-56 with softmax on adversarial images produced by attacking ResNet-56 with WNLL						
FGSM	Original data	59.6	59.5	58.0	56.3	54.3
FGSM	TVM data	50.7	40.6	41.2	37.4	34.5
FGSM	Original + TVM data	62.9	61.7	60.6	59.4	58.9
IFGSM	Original data	49.1	43.6	40.4	36.8	34.8
IFGSM	TVM data	30.3	23.7	20.1	18.0	17.3
IFGSM	Original + TVM data	53.9	49.2	44.7	41.9	39.9
CW-L2	Original data	54.7	54.2	54.4	53.8	54.0
CW-L2	TVM data	59.8	59.5	58.7	59.8	59.1
CW-L2	Original + TVM data	81.5	81.5	81.8	81.2	81.5
Classification accuracy of ResNet-56 with WNLL on adversarial images produced by attacking ResNet-56 with softmax						
FGSM	Original data	65.4	65.9	63.6	61.7	60.5
FGSM	TVM data	61.5	56.7	50.8	44.7	41.0
FGSM	Original + TVM data	69.7	67.6	65.5	64.8	63.4
IFGSM	Original data	51.9	43.9	38.9	35.4	34.2
IFGSM	TVM data	32.1	22.8	19.5	17.8	16.1
IFGSM	Original + TVM data	60.0	53.0	47.5	41.6	38.4
CW-L2	Original data	81.5	81.4	81.5	81.6	81.4
CW-L2	TVM data	57.6	58.4	57.8	58.4	58.4
CW-L2	Original + TVM data	90.6	90.6	90.5	90.1	90.4

where $\lambda_{TV} > 0$ is the regularization constant.

7 Numerical Results

7.1 Transferability of the Adversarial Images

To verify the efficacy of attack methods for DNNs with WNLL output activation, we consider the transferability of adversarial images. We train ResNet-56 on the aforementioned three types of training data with either softmax or WNLL activation. After the DNNs are trained, we attack them by FGSM, IFGSM, and CW-L2 with different ϵ. Finally, we classify the adversarial images by using ResNet-56 with the opponent activation. We list the mutual classification accuracy on adversarial images in Table. 1. The adversarial images resulting from attacking DNNs with two types of activation functions are both transferable, as the mutual classification accuracy is significantly lower than testing on the clean images. Overall we see a remarkably higher accuracy when applying ResNet-56 with WNLL activation to classify the adversarial images resulting from attacking ResNet-56 with softmax activation. For instance, for DNNs that are trained on the original images and attacked by FGSM, DNNs with the WNLL classifier have at least 5.4% higher accuracy (56.3% v.s. 61.7% ($\epsilon = 0.08$)). The accuracy improvement is more significant in many other scenarios.

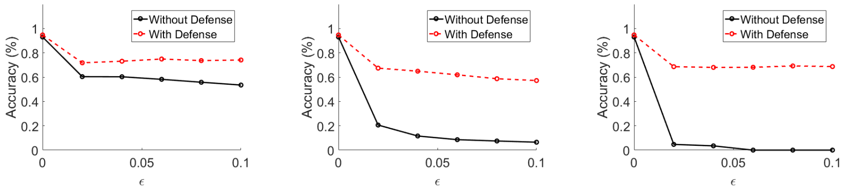
7.2 Adversarial Defense

Figure 5 plots the result of adversarial defense by combining the WNLL activation, TVM, and training data augmentation. Panels (a), (b) and (c) show the testing accuracy of ResNet-56 with and without defense on CIFAR-10 data for FGSM, IFGSM, and CW-L2, respectively. It can be observed that with increasing attack strength, ϵ, the testing accuracy decreases rapidly. FGSM is a relatively weak attack method, as the accuracy remains above 53.5% ($\epsilon = 0.1$) even with the strongest attack. Meanwhile, the defense maintains accuracy above 71.8% ($\epsilon = 0.02$). Figure 5 (b) and (c) show that both IFGSM and CW-L2 can fool ResNet-56 near completely even with small ϵ. The defense maintains the accuracy above 68.0%, 57.2%, respectively, under the CW-L2 and IFGSM attacks. Compared to state-of-the-art defensive methods on CIFAR-10, PixelDefend, our method is much simpler and faster. Without adversarial training, we have shown our defense is more

Table 2: Testing accuracy of ResNet-56 on the adversarial/TVM CIFAR-10 dataset. The testing accuracy without any defense are in red italic; and the results with all three defenses are in boldface.

Attack Method	Training data	$\epsilon = 0$	$\epsilon = 0.02$	$\epsilon = 0.04$	$\epsilon = 0.06$	$\epsilon = 0.08$	$\epsilon = 0.1$
				Vanilla ResNet-56			
FGSM	Original data	93.0	*60.4*/39.4	*60.3*/39.4	*58.2*/40.2	*55.8*/30.9	*53.5*/40.1
FGSM	TVM data	88.3	54.1/39.6	49.5/41.6	43.6/44.3	39.5/45.1	35.9/45.0
FGSM	Original + TVM data	93.1	63.2/66.6	62.7/67.8	62.4/68.7	62.0/68.1	61.3/68.7
IFGSM	Original data	93.0	*20.6*/35.0	*11.6*/32.3	*8.6*/31.0	*7.5*/28.8	*6.5*/27.6
IFGSM	TVM data	88.3	10.3/32.9	6.7/31.1	6.1/31.7	6.1/30.8	6.0/29.2
IFGSM	Original + TVM data	93.1	32.1/61.5	24.5/57.4	20.1/54.1	17.1/51.3	15.9/48.9
CW-L2	Original data	93.0	*4.7*/36.8	*3.5*/36.4	*0*/36.8	*0*/36.8	*0*/35.9
CW-L2	TVM data	88.3	8.2/36.5	8.1/36.0	8.0/35.9	8.0/35.8	8.0/36.3
CW-L2	Original + TVM data	93.1	13.6/62.2	13.6/62.2	13.0/62.1	12.0/62.1	12.0/61.9
				Data-Dependent Activated ResNet-56			
FGSM	Original data	94.5	71.1/49.9	72.1/51.1	71.3/51.7	70.6/52.2	67.3/51.8
FGSM	TVM data	90.6	62.6/49.3	56.8/54.1	52.1/56.2	46.0/56.6	41.0/57.1
FGSM	Original + TVM data	94.7	70.6/**71.8**	68.8/**73.1**	67.2/**74.9**	66.9/**73.6**	63.7/**74.1**
IFGSM	Original data	94.5	43.7/44.7	35.3/42.1	31.3/39.5	28.2/37.8	27.0/35.5
IFGSM	TVM data	90.6	12.1/44.3	7.1/41.1	7.2/37.4	6.9/37.2	6.8/35.3
IFGSM	Original + TVM data	94.7	35.0/**67.4**	25.1/**64.9**	20.5/**61.9**	17.5/**58.7**	16.3/**57.2**
CW-L2	Original data	94.5	11.9/40.1	11.7/40.8	11.0/40.8	10.8/41.2	10.8/40.5
CW-L2	TVM data	90.6	52.6/48.5	52.7/48.4	52.2/45.8	52.8/47.7	51.9/44.8
CW-L2	Original + TVM data	94.7	61.6/**68.6**	61.1/**68.0**	61.9/**68.1**	61.2/**69.2**	61.5/**68.7**

stable to IFGSM, and more stable to all three attacks under the strongest attack than PixelDefend Song et al. (2018). Moreover, our defense strategy is additive to adversarial training and many other defenses including PixelDefend.



(a) (b) (c)

Figure 5: Attack strength ϵ v.s. accuracy without defense, and defending by WNLL activation, TVM and augmented training. (a), (b), (c) plot results for FGSM, IFGSM, and CW-L2 attack, respectively.

To analyze the defensive contribution from each component of the defensive strategy, we separate the three parts and list the testing accuracy in Table. 2. Simple TVM cannot defend FGSM attacks except when the DNNs are trained on the augmented data, as shown in the first and fourth horizontal blocks of the table. WNLL activation improves the testing accuracy of adversarial attacks significantly and persistently. Augmented training can improve the stability consistently as well.

8 Concluding Remarks

In this paper, by analyzing the influence of adversarial perturbations on the geometric structure of the DNNs' features, we propose to defend against adversarial attack by applying a data-dependent activation function, total variation minimization on the adversarial images, and training data augmentation. Results on ResNet-56 with CIFAR-10 benchmark reveal that the defense improves robustness to adversarial perturbation significantly. Total variation minimization simplifies the adversarial images, which is very useful in removing adversarial perturbation. Another interesting direction to explore is to apply other denoising methods to remove adversarial perturbation.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

References

N. Akhtar and A. Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. arXiv preprint arXiv:1801.00553, 2018.

A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. International Conference on Machine Learning, 2018a.

A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. International Conference on Machine Learning, 2018b.

Y. Bengio, N. Leonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432, 2013.

W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv preprint arXiv:1712.04248, 2017.

N. Carlini and D.A. Wagner. Towards evaluating the robustness of neural networks. IEEE European Symposium on Security and Privacy, pp. 39–57, 2016.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6275, 2014.

K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel. Adversarial perturbations against deep neural networks for malware classification. arXiv preprint arXiv:1606.04435, 2016.

A. Guisti, J. Guzzi, D.C. Ciresan, F.L. He, J.P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Carlo, and et al. A machine learning approach to visual perception of forecast trails for mobile robots. IEEE Robotics and Automation Letters, pp. 661–667, 2016.

C. Guo, M. Cisse, and L. van der Maaten. Countering adversarial images using input transformation. International Conference on Learning Representations, 2018.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, pp. 770–778, 2016.

A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. Black-box adversarial attacks with limited queries and information. International Conference on Machine Learning, 2018.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

A. Kurakin, I. J. Goodfellow, and S. Bengio. Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533, 2016.

Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770, 2016.

A. Mardy, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. International Conference on Learning Representations, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

Seyed-Mohsen Moosavi-Dezfooli, Ashish Shrivastava, and Oncel Tuzel. Divide, denoise, and defend against adversarial attacks. CoRR, abs/1802.06806, 2018. URL http://arxiv.org/abs/1802.06806.

T. Na, J. H. Ko, and S. Mukhopadhyay. Cascade adversarial machine learning regularized with a unified embedding. International Conference on Learning Representations, 2018.

N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. IEEE European Symposium on Security and Privacy, pp. 372–387, 2016a.

N. Papernot, P. McDaniel, A. Sinha, and M. Wellman. Sok: Towards the science of security and privacy in machien learning. arXiv preprint arXiv:1611.03814, 2016b.

N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. IEEE European Symposium on Security and Privacy, 2016c.

Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. CoRR, abs/1605.07277, 2016d. URL http://arxiv.org/abs/1605.07277.

L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena, pp. 259–268, 1992.

P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversaial attacks using generative models. International Conference on Learning Representations, 2018.

Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. International Conference on Learning Representations, 2018.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, and I. Goodfellow. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.

Florian Tramr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=rkZvSe-RZ.

B. Wang, X. Luo, Z. Li, W. Zhu, Z. Shi, and S. Osher. Deep neural nets with interpolating function as output activation. Advances in Neural Information Processing Systems, 2018.

X. Wu, U. Jang, J. Chen, L. Chen, and S. Jha. Reinforcing adversarial robustness using model confidence induced by adversarial training. International Conference on Machine Learning, 2018.