Identifying Layers Susceptible to Adversarial Attacks

Preprint · July 2021

CITATIONS
0

READS
9

2 authors, including:

Shoaib Ahmed Siddiqui
Deutsches Forschungszentrum für Künstliche Intelligenz
32 PUBLICATIONS **887** CITATIONS

Identifying Layers Susceptible to Adversarial Attacks

Shoaib Ahmed Siddiqui
German Research Center for Artificial Intelligence (DFKI)
TU Kaiserslautern
shoaib_ahmed.siddiqui@dfki.de

Thomas Breuel
NVIDIA Research
tbreuel@nvidia.com

Abstract

Common neural network architectures are susceptible to attack by adversarial samples. Neural network architectures are commonly thought of as divided into low-level feature extraction layers and high-level classification layers; susceptibility of networks to adversarial samples is often thought of as a problem related to classification rather than feature extraction. We test this idea by selectively retraining different portions of VGG and ResNet architectures on CIFAR-10, Imagenette and ImageNet using non-adversarial and adversarial data. Our experimental results show that susceptibility to adversarial samples is associated with low-level feature extraction layers. Therefore, retraining high-level layers is insufficient for achieving robustness. This phenomenon could have two explanations: either, adversarial attacks yield outputs from early layers that are indistinguishable from features found in the attack classes, or adversarial attacks yield outputs from early layers that differ statistically from features for non-adversarial samples and do not permit consistent classification by subsequent layers. We test this question by large-scale non-linear dimensionality reduction and density modeling on distributions of feature vectors in hidden layers and find that the feature distributions between non-adversarial and adversarial samples differ substantially. Our results provide new insights into the statistical origins of adversarial samples and possible defenses.

1 Introduction

Deep neural networks often yield performance on test sets comparable to human performance [9]. However, at the same time, they have been found to be susceptible to imperceptible perturbations of inputs [23, 8, 16, 29]. These new samples crafted by an adversary with the aim to fool the classifier are termed adversarial examples [23]. There has been a plethora of research in developing stronger defenses as well as stronger adversarial attacks to circumvent these defenses [8, 16, 29, 32, 27, 1, 18, 6, 14]. However, the reasons for their existence are still poorly understood [8, 10, 26]. Understanding these differences between deep neural networks and human perception is important both in order to understand the mathematical and statistical structure of such networks, as well as to protect systems against attacks.

Deep neural networks automate the task of feature extraction, obviating the need for hand-engineering features. Such networks are thought of as consisting of initial feature extraction layers and high-level layers responsible for learning decision boundaries. In fact, in many cases, initial feature extraction layers are often reused in practice between different datasets and tasks to speed up convergence (commonly known as transfer learning [33]). In the context of adversarial samples, if we could reuse feature extraction layers, it would greatly speed up research in adversarial samples, since adversarial samples could be studied on pre-extracted data. If susceptibility to adversarial samples is associated with high-level layers, it would also give us insights into the nature of adversarial phenomena and suggest that adversarial samples might be related primarily to the formation of decision boundaries by

Preprint. Under review.

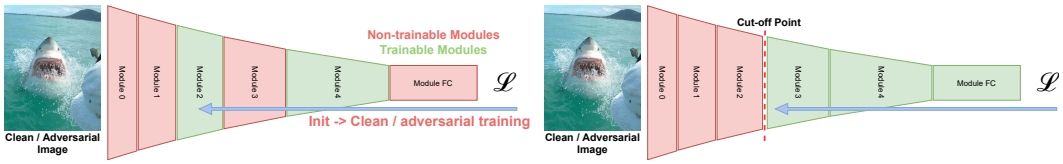


Figure 1: Overview of the training methodology where a set of of blocks is reinitialized and retrained (using either clean or adversarial samples) while others are kept frozen after loading the pretrained model. One special case is that of a single cut-off point (right) where the network from the beginning to the cut-off point or from the cut-off point to the end is trained while keeping the other part fixed.

the high-level layers. This idea has been leveraged in techniques such as large-margin classification to achieve adversarial robustness [4].

To test these ideas, we use a novel block-wise retraining protocol. In particular, considering a pretrained model, either trained non-adversarially or adversarially, we reinitialize and retrain a particular set of blocks either conventionally for an adversarially pretrained model or adversarially for a conventionally pretrained model. The key findings of this paper can be summarized as follows:

- Adversarial retraining of just low-level/early layers is associated with strong reductions in susceptibility to adversarial samples.
- Adversarial retraining of just high-level/late layers fails to result in robustness to adversarial samples.
- The distributions of feature vectors from non-adversarial and adversarial inputs differ substantially at all levels; therefore, susceptibility to adversarial attacks is associated with the early generation of feature vectors that do not occur in non-adversarial images.
- Adversarial training results in weights for early layers that bring the distribution of feature vectors for adversarial samples back to the distribution of feature vectors for non-adversarial samples.

Overall, these results show that susceptibility to adversarial samples is primarily a phenomenon associated with early layers and low-level feature extraction. Adversarial samples do not merely transform the appearance of one image into that of another, but instead generate novel feature vectors that result in novel activation patterns in late layers and high-level, class-specific feature vectors.

2 Related Work

Origins of Adversarial Examples. Adversarial robustness has become an important area of research in computer vision [8, 16, 29, 32, 27, 1, 18, 6, 14]. There has been a range of sophisticated hypothesis developed to explain the phenomenon of adversarial examples which includes high reliance on texture [7], excessive invariance [11], over-reliance on high-frequencies [26], piece-wise linear nature of deep networks [8], or even a bias present in the dataset itself [10]. However, their statistical origins are still poorly understood. Therefore, our work provides detailed insights into how adversarial examples change the prediction of a classifier.

Robust Optimization. Robust optimization is one of the most powerful adversarial defense methods to date [8, 16, 29, 32, 27] where the model is trained on adversarial samples rather than clean samples. We use robust optimization as a tool within our analysis. Our analysis highlights the importance of the initial layers of the network, which can aid in developing more powerful defenses in the future.

Image-Denoising. Our work naturally connects to image-denoising-based approaches for adversarial robustness. Image-denoising-based defenses introduce a purification/denoising network at the beginning of the model, which attempts to cancel out any perturbation added to the input [1, 18, 6, 14]. Since it is attached at the beginning of the network, this purification/denoising network can be considered analogous to the initial blocks of the network in our framework. Therefore, our findings shed light on the consistency between robust optimization and image-denoising-based defenses.

Adversarial Example Detection. Prior works have also attempted to identify clean samples from adversarial ones and reject them before classification based on different criterion [20]. Our results also highlight the fact that it is possible to identify adversarial examples as they form highly distinct feature vectors. However, based on our analysis, the recovery of the original class is not possible.

Activation Perspective of Adversarial Robustness. Activations of the network under adversarial attack have been analyzed in prior works [17, 15, 2]. Recently, Bai et al. (2021) [2] have also analyzed the discrepancy in the activations of conventionally trained models, and show that the magnitude of activations for adversarial samples is substantially higher than natural samples. However, the analysis in all these cases was limited to the differences in the final feature vector represented by the penultimate layer of the network. On the other hand, our analysis aims to understand the differences as they originate through the initial layers, and amplified while propagating through successive layers, ultimately forming out-of-distribution (OOD) samples for the classifier.

3 Methods

Layers present in a network are often thought of as operating at different semantic levels, where initial layers respond to basic features such as edges or gradients, while higher layers represents complete objects or some prominent parts of it [30]. In order to analyze the role of different layers of the network in terms of their susceptibility to adversarial noise, we use a block-wise retraining protocol. Given a model (ResNet or VGG in our case), we split the model into different modules. Both ResNet and VGG models are naturally dissected into six modules by the down-sampling layers within the network (max-pooling in the case of VGG and convolutional layer with a stride of 2 in the case of ResNets). An overview of the method is presented in Fig. 1.

We first pretrain the complete network either conventionally or adversarially. Conventional training refers to training on clean images while adversarially training refers to training on adversarial images computed using a particular attack method. We follow the adversarial training recipe from Madry et al. (2017) [16] where we train the model on adversarial images computed using Projected-Gradient Descent (PGD) attack. Once the model is pretrained, we reinitialize and retrain a set of modules of the network adversarially for the conventionally pretrained model or conventionally for the adversarially pretrained model, while keeping the weights for the rest of the modules fixed. Our main experiments rely on a single splitting point for the network, where we only retrain all the layers before or after the cut-off point.

3.1 Datasets

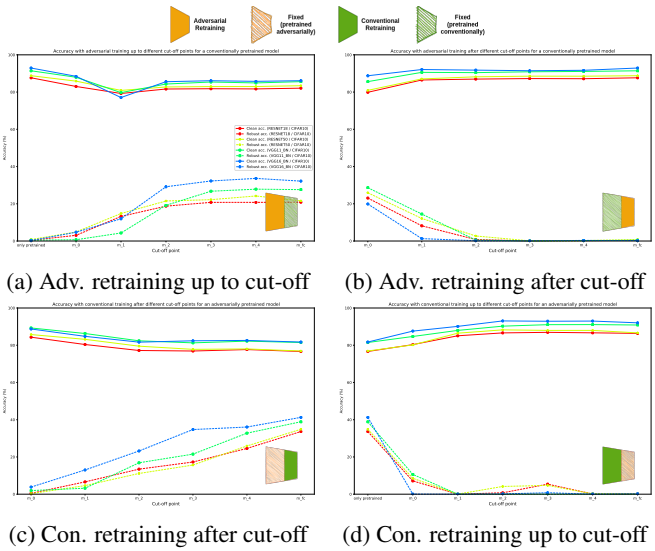
We validate our findings by testing on three different datasets including CIFAR-10 [12], ImageNet [21] and Imagenette [5]. CIFAR-10 [12] is a 10 class dataset comprising of low-resolution images (32×32) with 50000 training and 10000 test samples. We also include the large-scale high-resolution ImageNet dataset with 1.28M training and 50000 validation samples to evaluate our hypothesis[1]. Finally, we include a small subset of ImageNet called Imagenette with only 10 classes but high-resolution images (9469 training and 3925 test samples).

3.2 Experimental Protocol

All the CIFAR models were trained on a single GPU (NVIDIA RTXA6000) with Adam optimizer with an initial learning rate of 0.001 and a batch size of 128. The models were trained for 300 epochs, with a cosine decay in the learning rate after every epoch. For adversarial training, we used an epsilon of 8/255 and epsilon per iteration of 2/255. We used PGD-based adversarial training [16, 29] where the number of iterations was fixed to 7. We use identical training settings for Imagenette.

For the ImageNet dataset, we use fast adversarial training [27] to speed up the training process. Fast adversarial training uses a random start followed by a single step in the direction of the gradient which makes it equivalent to PGD-1. We use an epsilon of 4/255 for ImageNet as per the common practice [27, 28]. The model was trained using 8 GPUs (NVIDIA RTXA6000) with synchronized batch-norm using SGD with an initial learning rate of 0.256, a momentum of 0.875, and a batch size

[1]The validation set serves the purpose of the test set in our case as direct access to the test set is not available. Therefore, no hyperparameters are tuned directly on the validation set.



(a) Adv. retraining up to cut-off (b) Adv. retraining after cut-off

(c) Con. retraining after cut-off (d) Con. retraining up to cut-off

Figure 2: Partial retraining of VGG and ResNet architectures on CIFAR-10 shows that robustness to adversarial samples is achieved if and only if the weights for early layers were either pretrained or retrained with adversarial samples. *Only pretrained* refers to the performance of the original pretrained network without any partial retraining.

of 256, where the learning rate was reduced by a factor of 10 after the 30^{th}, 60^{th} and the 90^{th} epoch. We used a weight decay of 0.0001 to train all our models.

For both Imagenette and CIFAR-10, we add clean samples to the batch during adversarial retraining. This inclusion of clean examples helps maintain the clean accuracy of the model. However, we do not include any clean samples when retraining ResNet-50 on ImageNet.

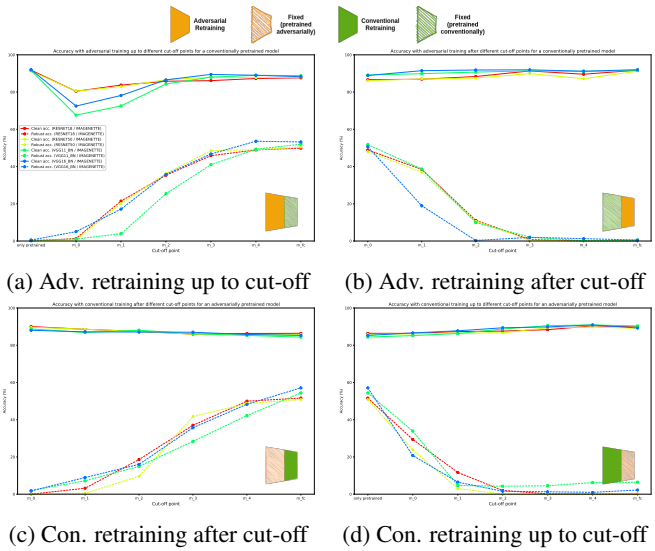
For model evaluation, we use PGD-200 [29, 28, 16] with a single restart. Stronger attacks do exist [3], but the objective of this paper is to determine the relative susceptibility of layers. Therefore, absolute numbers are not particularly important in our case. It is important to mention that we attack the model using the actual target during evaluation rather than the prediction. This ensures that the robust accuracy does not accidentally inflate due to the attack moving the class from wrong to the correct label. However, we still attack the prediction of the model during adversarial training to avoid the *label-leaking* effect [13].

3.3 Model Architectures

We evaluated the VGG [22] and ResNet [9] model families imported from the TorchVision [24] model repository. We specifically evaluated VGG-11 and VGG-16, where both of these models were equipped with batch-norm. In order for these architectures to work on CIFAR, we replace the average pooling layer before the classification head (which outputs a 7×7 tensor) with Global Average Pooling (GAP) layer, which reduces the dimensionality to 1×1. This is similar to the residual architecture [9]. We include ResNet-18 and ResNet-50 within the residual family [9] for our experiments with identical architecture across datasets.

4 Key Results

We evaluate the block-wise susceptibility of four models (ResNet-18, ResNet-50, VGG-11, VGG-16) belonging to two major model families (ResNet [9] and VGG [22]) on three image recognition datasets (CIFAR-10 [12], Imagenette [5] and ImageNet [21]) using our block-wise retraining protocol. The primary results are divided into four different training settings that we evaluate which include (i) adversarial retraining before the cut-off, (ii) adversarial retraining after the cut-off, (iii) conventional retraining before the cut-off, and (iv) conventional retraining after the cut-off.



(a) Adv. retraining up to cut-off (b) Adv. retraining after cut-off

(c) Con. retraining after cut-off (d) Con. retraining up to cut-off

Figure 3: Results on Imagenette are primarily consistent with all the findings on the CIFAR-10 dataset despite a larger input size of 224×224. However, we see that since the size of the objects is larger in this case, the importance of the higher-level modules increases.

4.1 CIFAR-10

We see that the adversarial performance plateaus after just adversarially retraining the first three modules of the network as shown in Fig. 2. Furthermore, all four different settings provide a consistent picture, where adversarial robustness is consistently associated with adversarially trained early layers. This also indicates that the main discrepancy between conventionally and adversarially trained models lies at the lower-level features. For conventional retraining, just retraining the initial two modules evades the robustness of the network, while having a marginal effect on clean accuracy. This highlights the fact that the initial modules are the most distinct ones between conventionally and adversarially trained models.

4.2 Imagenette

Despite a larger image size of 224×224 for Imagenette, we see a similar trend in terms of retraining as CIFAR-10 where adversarial retraining of initial modules is important for obtaining robustness (Fig. 3). However, we see a relative increase in the importance of higher-level modules as compared to CIFAR-10. This can be attributed to the larger image size, where the object occupies a larger fraction of the image, requiring a larger effective receptive field of the network.

4.3 ImageNet

The results on ImageNet are again consistent with our prior results on CIFAR-10 and Imagenette. However, since ImageNet is significantly larger in both image size as well as the number of images, we see a shift in the cut-off point where mid-level modules (m_2, m_3, and m_4) also play a dominant role for robustness as evident from Fig. 4. Our evaluation is limited to ResNet-50 trained using fast adversarial training [27] on ImageNet [21]. As the distribution of parameters is not the same in every module, a small number of layers is insufficient for robustness on ImageNet. Therefore, this shift in the cut-off point is not surprising. Looking at the performance in the case where the model is retrained after the cut-off point, we see the same trend where excluding the first two modules results in poor robustness of the model.

(a) Adv. retraining up to cut-off (b) Adv. retraining after cut-off

(c) Con. retraining after cut-off (d) Con. retraining up to cut-off

Figure 4: ImageNet results with ResNet-50 architecture are consistent with the results on Imagenette or CIFAR-10, but we observe a shift in the cut-off point resulting in a rise in the importance of the higher-level modules. This can be partly explained by the larger size of ImageNet where initial modules themselves are insufficient to provide robustness since most of the parameters lie in the higher-level modules.

5 Analysis

Based on our preliminary findings, we analyze particular aspects of the models in more detail. This includes extending our analysis to every layer, evaluating all possible combinations of modules as well as evaluating layer robustness to reinitialization.

5.1 Per-Layer Analysis

We perform a more fine-grained layer-wise retraining evaluation for ResNet-18 on CIFAR-10 where we move away from complete modules which comprise of a different number of layers at each level and focus on the individual layers themselves. This analysis decouples the aggregation artifacts and highlights if there are other layers that are equally important as the initial layers in the network. The results for the per-layer cut-off experiment are visualized in Fig. 5. The analysis show that the gains are flattened out after the inclusion of the first seven layers, which is consistent with the results from the module-based experiment as these layers form the initial modules of the network. We observe the most significant gain for the first layer in the network, which is referred to as *m_0* in our previous results.

5.2 Module Combinations

In our first set of experiments, we considered a single split in the network where we either train the network before the cut-off or after the cut-off while keeping the other part fixed. However, this does not preclude the possibility that a combination of lower-level and higher-level modules provides better robustness as compared to just a single split. In order to test this, we trained ResNet-18 (CIFAR-10) on all the different possible combinations of modules. These results are summarized in Fig. 6 where we plot the median accuracy for all possible combinations of the modules with or without a particular module. These results are qualitatively consistent with single cut-off experiments, where adversarial training of the initial modules is essential for robustness to adversarial samples. This indicates that there are no specific combinations of lower-level and high-level modules that are robust, but rather, just the initial set of layers are important for this purpose.

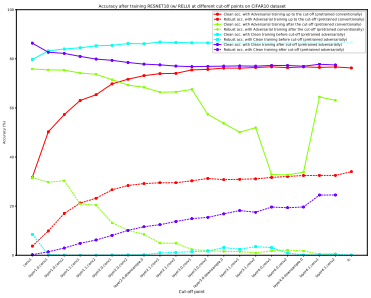
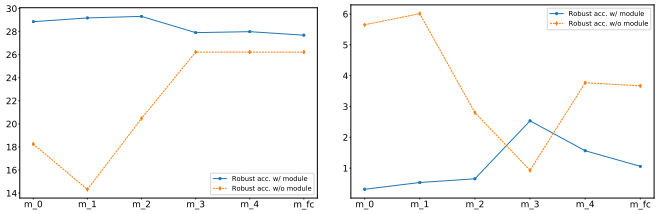


Figure 5: Results decomposed at the level of layers rather than modules for the ResNet-18 architecture on CIFAR-10. These results are consistent with the module-based results as the initial layers are part of the initial modules of the network. We do not observe any significant contribution from a single layer, but rather, the contribution from different layers accumulates.



(a) Adversarial retraining (b) Conventional retraining

Figure 6: Median accuracy when considering the distribution of accuracies (ResNet-18 on CIFAR-10) when either including or excluding a particular module. Median robust performance drops whenever the initial layers are not adversarially trained in the end.

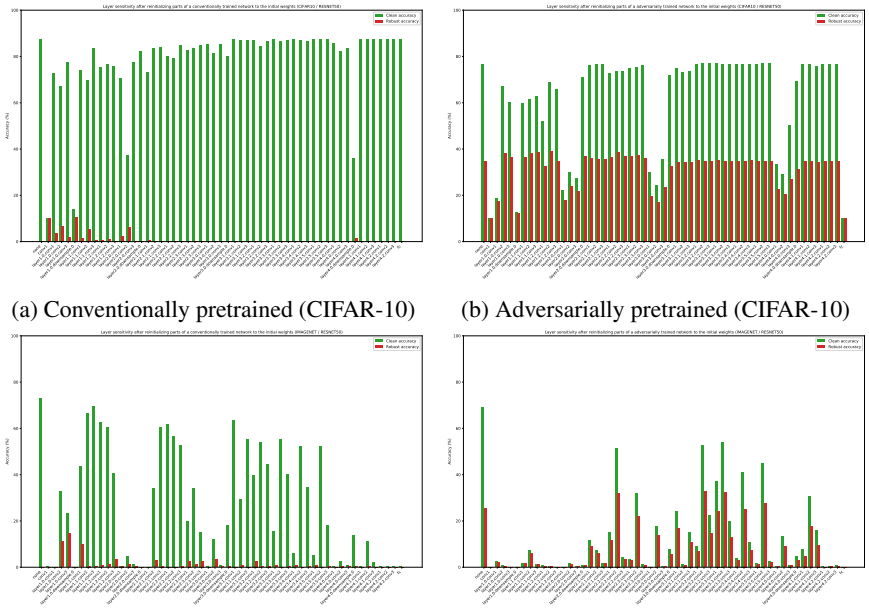
5.3 Layer-wise Reinitialization Robustness

We attempt to understand the behavior described by Zhang et al. (2019) [31] for adversarial samples, where they found some layers to be much more important than others for overall classification performance. Although we reproduce their results on clean samples, from the point of view of adversarial training, we find (Fig. 7a) that reinitialization of low-level layers tends to induce significant adversarial robustness. This is consistent with our other findings, namely that adversarial samples are a phenomenon associated with low-level layers. In addition, it suggests that susceptibility to adversarial samples is associated with training [10]. Conversely, susceptibility to adversarial samples is never significantly increased due to layer reinitialization (Fig. 7b). Since ResNet-50 models trained on full ImageNet are much more susceptible to layer reinitialization, the results are more difficult to interpret. However, we still observe that reinitialization of initial layers tends to induce higher robustness to adversarial samples (Fig. 7c). Furthermore, while non-adversarial accuracy is usually strongly affected by reinitialization, adversarial accuracy is usually less affected in comparison (Fig. 7d).

6 Feature Distributions

Above, we have seen the differential effects of early and late layers on adversarial robustness. Adversarial attacks on a network might operate by changing one type of feature into another, leaving the overall distribution of feature vectors the same, or by producing novel feature vectors that do not occur in non-adversarial samples. These changes might occur only in late layers or both in early and late layers. This distinction is important both for understanding the nature of adversarial attacks and to devise possible defenses.

Prior work visualizing the activities in hidden convolutional layers has primarily focused on visualizing the aggregate or per-filter activity [19]. In contrast, we visualize the distribution of activations for all the filters simultaneously across many images and layers, under both adversarial and non-adversarial conditions, using nonlinear dimensionality reduction by picking a single vector across



(a) Conventionally pretrained (CIFAR-10) (b) Adversarially pretrained (CIFAR-10)

(c) Conventionally pretrained (ImageNet) (d) Adversarially pretrained (ImageNet)

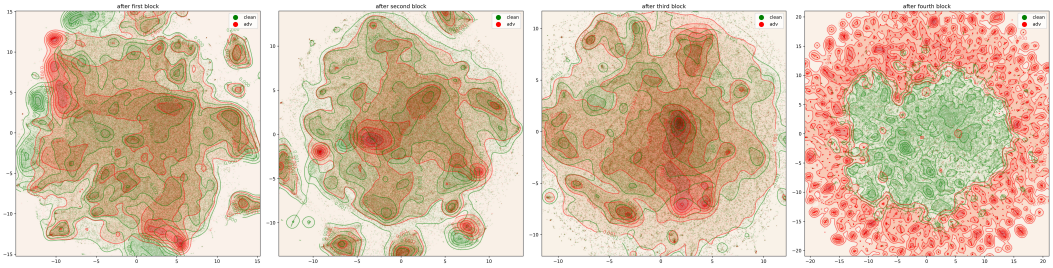
Figure 7: Layer-wise reinitialization robustness for ResNet-50 on CIFAR and ImageNet with either conventional or adversarial training. *none* refers to the performance of the original model. Adversarially pretrained ResNet-50 demonstrates the high sensitivity of the initial layers in contrast to the conventionally pretrained model, which indicates that the initial layers change significantly to cater for the adversarial noise, highlighting their importance in obtaining robust models.

spatial dimensions of the activation ($\mathbf{z} \in \mathbb{R}^C$ where C is the number of filters in a layer). Representative results are shown in Figure 8 using t-SNE dimensionality reduction. Images represent random samples after blocks one through four in a ResNet 50 network, choosing 100 random samples from each of 1000 different images. Qualitative equivalent results can be seen with other non-linear dimensionality techniques (UMAP, TriMap) and are not shown here. Each scatterplot is overlayed with a kernel density estimation in the dimensionality-reduced space, with green regions corresponding to non-adversarial samples and red regions corresponding to adversarial samples.

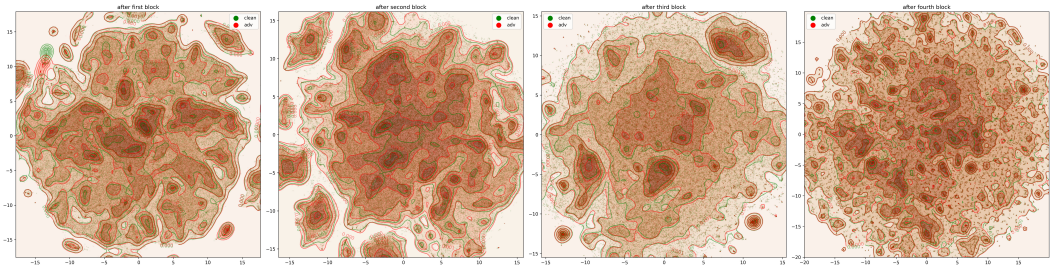
Fig. 8a shows that the distribution of feature vectors differs substantially between non-adversarial and adversarial samples. In fact, after block 4 (high-level features), the distribution of adversarial and non-adversarial sample are almost completely non-overlapping, showing that adversarial samples do not imitate non-adversarial activation patterns or outputs, but generate very different collections of high-level features. This difference is particularly striking since there is little overlap even in the feature vectors that might correspond to background regions in the image. The second striking phenomenon observable in Fig. 8a is that substantial distributional differences are present even after the first block, i.e., in low-level features. This is consistent with our findings above, namely that differences between non-adversarial and adversarial samples must occur already in the early layers of the network and are responsible for susceptibility to adversarial samples.

Fig. 8b illustrates the same distributions for adversarially trained networks. What we find here is that the distributions of feature vectors associated with non-adversarial and adversarial samples are much closer, not just overlapping in general, but reproducing peaks and regions of high density in substantial detail. That is, adversarial training has to adjust not just the high-level convolutional layers to match distributions between adversarial and non-adversarial samples, but also the low-level feature extraction layers (and from our above experiments, we already know that this change to the distribution of low-level feature vectors is both necessary and sufficient).

State-of-the-art reductions in susceptibility to adversarial samples through adversarial training results in feature vector distributions for non-adversarial and adversarial samples that closely match each other. Nevertheless, the resulting models still have substantial susceptibility to adversarial samples. This implies that the remaining successful adversarial attacks probably work by transforming feature



(a) Conventionally pretrained ResNet-50 (ImageNet)



(b) Adversarially pretrained ResNet-50 (ImageNet)

Figure 8: Dimensionality reduced activation vectors using t-SNE [25] of four (*m_1* to *m_4*) modules in the network. These plots highlight the substantial differences between clean and adversarial activations, which are amplified upon propagation through higher-level modules. Adversarial training minimizes these differences between activations.

vectors into each other while staying within the distribution of non-adversarial feature vectors. In other words, adversarial attacks on adversarially trained networks appear to be qualitatively different from adversarial attacks on undefended networks.

7 Conclusion

We have described partial retraining of networks as a technique for localizing susceptibility to adversarial samples in deep neural networks. Furthermore, we have demonstrated that dimensionality reduction of sets of activation vectors in different layers can be a useful tool for understanding the statistics and relations of adversarial and non-adversarial vectors. Our experimental results demonstrate that susceptibility of deep neural networks to adversarial samples is associated with the early, non-specific layers of such networks. That is, we have shown that adversarial samples generate differences in feature distributions in those layers and that training networks to be robust to adversarial samples largely eliminates those distributional differences. Practically, this means that in order to achieve robustness to adversarial samples, it is both necessary and sufficient to retrain only the early layers where feature vectors are not yet highly class specific. Our experiments also show that adversarial samples can be detected and visualized easily as anomalies or outliers. A substantial gap between human performance and deep neural networks however remains even after adversarial training. Our results show that these differences are not merely quantitative in nature. Rather, in the absence of adversarial training, adversarial samples succeed via generating novel feature vectors, while after adversarial training, adversarial samples mimic the feature distribution of non-adversarial samples, suggesting that different defense mechanisms may be required. The techniques described should prove useful in future work on understanding the statistical origins of adversarial samples, as well as devising practical techniques for defending against adversarial samples.

Broader Impact

Our investigation aims to help understand the causes of the existence of adversarial examples, which is a major failure mode of current deep learning models. Deep learning-based visual recognition systems have been deployed in a range of different areas, including self-driving cars and security

systems. Improving the robustness of these systems is critical. Furthermore, a better understanding of robustness can help us achieve robustness in an efficient way without going through the compute-intensive process of adversarial training. However, on the flip side, these robust systems can potentially be used in a negative context such as mass surveillance.

Acknowledgements

The authors would like to acknowledge useful discussions with Iuri Frosio on adversarial robustness. This work is in part supported by the BMBF project DeFuseNN (Grant 01IW17002) and the NVIDIA AI Lab (NVAIL) program.

References

[1] Naveed Akhtar, Jian Liu, and Ajmal Mian. Defense against universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3389–3398, 2018.

[2] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing. *arXiv preprint arXiv:2103.08307*, 2021.

[3] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[4] Gamaleldin F Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. *arXiv preprint arXiv:1803.05598*, 2018.

[5] FastAI. Imagenette. `https://github.com/fastai/imagenette`, 2019.

[6] J. Folz, S. Palacio, J. Hees, and A. Dengel. Adversarial defense based on structure-to-signal autoencoders. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3568–3577, 2020.

[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

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

[9] 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*, pages 770–778, 2016.

[10] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

[11] Joern-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. In *International Conference on Learning Representations*, 2019.

[12] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55, 2014.

[13] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[14] Guanlin Li, Shuya Ding, Jun Luo, and Chang Liu. Enhancing intrinsic adversarial robustness via feature pyramid decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 800–808, 2020.

[15] Zhuorong Li, Chao Feng, Jianwei Zheng, Minghui Wu, and Hongchuan Yu. Towards adversarial robustness via feature matching. *IEEE Access*, 8:88594–88603, 2020.

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[17] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. *arXiv preprint arXiv:1909.00900*, 2019.

[18] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 262–271, 2020.

[19] Paulo E Rauber, Samuel G Fadel, Alexandre X Falcao, and Alexandru C Telea. Visualizing the hidden activity of artificial neural networks. *IEEE transactions on visualization and computer graphics*, 23(1):101–110, 2016.

[20] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In *International Conference on Machine Learning*, pages 5498–5507. PMLR, 2019.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

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

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[24] PyTorch Team. torchvision. `https://github.com/pytorch/vision/`, 2021.

[25] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[26] Haohan Wang, Xindi Wu, Pengcheng Yin, and Eric P Xing. High frequency component helps explain the generalization of convolutional neural networks. *arXiv preprint arXiv:1905.13545*, 2019.

[27] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[28] Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

[29] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 501–509, 2019.

[30] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015.

[31] Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *arXiv preprint arXiv:1902.01996*, 2019.

[32] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.

[33] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.