SHIFT INVARIANCE CAN REDUCE ADVERSARIAL ROBUSTNESS

Songwei Ge, Vasu Singla *
Univeristy of Maryland, College Park
{songweig,vsingla}@umd.edu

Ronen Basri
Weizmann Institute of Science
ronen.basri@weizmann.ac.il

David Jacobs
Univeristy of Maryland, College Park
dwj@umd.edu

ABSTRACT

Shift invariance is a critical property of CNNs that improves performance on classification. However, we show that invariance to circular shifts can also lead to greater sensitivity to adversarial attacks. We first characterize the margin between classes when a shift-invariant *linear* classifier is used. We show that the margin can only depend on the DC component of the signals. Then, using results about infinitely wide networks, we show that in some simple cases, fully connected and shift-invariant neural networks produce linear decision boundaries. Using this, we prove that shift invariance in neural networks produces adversarial examples for the simple case of two classes, each consisting of a single image with a black or white dot on a gray background. This is more than a curiosity; we show empirically that with real datasets and realistic architectures, shift invariance reduces adversarial robustness. Finally, we describe initial experiments using synthetic data to probe the source of this connection.

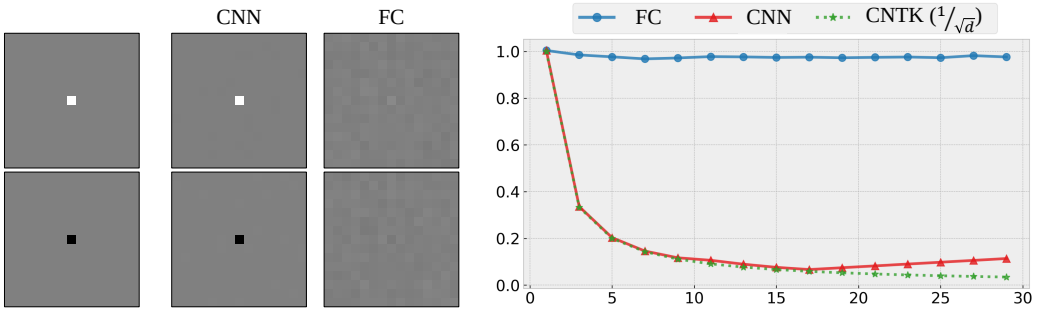
1 INTRODUCTION

In *adversarial attacks* (Szegedy et al., 2013) against classifiers, an adversary with knowledge of the trained classifier makes small perturbations to a test image (or even to objects in the world (Eykholt et al., 2018; Wu et al., 2020b)) that change the output. Vulnerability to such attacks threatens the deployment of deep learning systems in many critical applications, from spam filtering to self-driving cars. Despite a great deal of study, it remains unclear why neural networks are so susceptible to adversarial attacks. We show that invariance to circular shifts in Convolutional Neural Networks (CNNs) can be one cause of this lack of robustness. All reference to shifts will refer to circular shifts.

To motivate this conclusion we study in detail a simple example. Indeed, one of our contributions is to present perhaps the simplest possible example in which adversarial attacks can occur. Figure 1 shows a two class classification problem in which each class consists of a single image, a white or black dot on a gray background. We train either a fully connected (FC) network or a fully shift-invariant CNN to distinguish between them. Since each class contains only a single image, we measure adversarial robustness as the l_2 distance to an adversarial example produced by a DDN attack (Rony et al., 2019), using the training images as the starting point. The figure shows that the CNN is much less robust than the FC network, and that the robustness of the CNN drops precipitously with the image size.

In Sections 2 and 3 we explain this result theoretically. In Section 2 we study the effect of shift invariance on the margin of linear classifiers, for linearly separable data. Next, in Section 3, we draw on recent work that characterizes infinitely wide neural networks as kernel methods and prove that under certain assumptions, a shift invariant CNN produces a linear decision boundary for the example in Figure 1, explaining its lack of adversarial robustness. It is valuable to produce such a simple example in which a lack of adversarial robustness provably occurs, and can be fully understood. Still, it is reasonable to question whether shift invariance can affect robustness in real networks trained on real data. In Section 4 we show experimentally that it can.

*Equal contribution



(a) Clean Samples (b) Adversarial Examples (c) Distance to generate adversarial examples vs. image width $\sqrt{d}$

Figure 1: In a binary classification problem as shown in (a), with each image as a distinct class, a FC network requires an average L2 distance close to 1 to construct an adversarial example while a shift-invariant CNN only requires approximately $\frac{1}{\sqrt{d}}$ as illustrated in (c), where d is the input dimension. The curve labeled CNN shows results for a real trained network, while CNTK shows the theoretically derived prediction of $\frac{1}{\sqrt{d}}$ for an infinitely wide network. We visualized the adversarial examples for $d = 225$ of both models in (b). Note that the change is imperceptible for the CNN.

The main contribution of our paper is to show theoretically that shift invariance can undermine adversarial robustness, and to show experimentally that this is indeed the case on real-world networks.

2 Shift Invariance and the Linear Margin

In this section we consider what happens when a linear classifier is required to be shift invariant. That is, we suppose that any circularly shifted version of a training example should be labeled the same as the original sample. For convenience of notation we consider classes of 1D signals of length d, in which $\mathbf{x} = (x_1, x_2, ...x_d)$. Our results are easily extended to signals of any dimension. We denote the set of training samples with label $+1$ by X_1, and the set of samples with label -1 by X_2. Let $\mathbf{x}^s$ denote the signal $\mathbf{x}$ shifted by s, with $0 \leq s \leq d-1$. That is: $\mathbf{x}^s = (x_{s+1}, x_{s+2}, ...x_d, x_1, ...x_s)$. For a signal $\mathbf{x}$, let $\mathcal{SH}(\mathbf{x})$ denote the set containing every shifted version of $\mathbf{x}$. That is $\mathbf{x}^s \in \mathcal{SH}(\mathbf{x}),\ \forall s, 0 \leq s \leq d-1$. Let $S_i = \cup_{\mathbf{x} \in X_i} \mathcal{SH}(\mathbf{x})$ denote the set of all shifted versions of all signals in the set X_i. Further, let $f_{dc}(\mathbf{x})$ denote the DC component of a signal. That is, $f_{dc}(\mathbf{x}) = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} \mathbf{x}_i$. We denote $\bar{\mathbf{w}} = \frac{1}{\sqrt{d}} \mathbf{1}_d$. where $\mathbf{1}_d$ is a d-dimensional vector of all ones, so $f_{dc}(\mathbf{x}) = \bar{\mathbf{w}}^T \mathbf{x}$. We show the theorem:

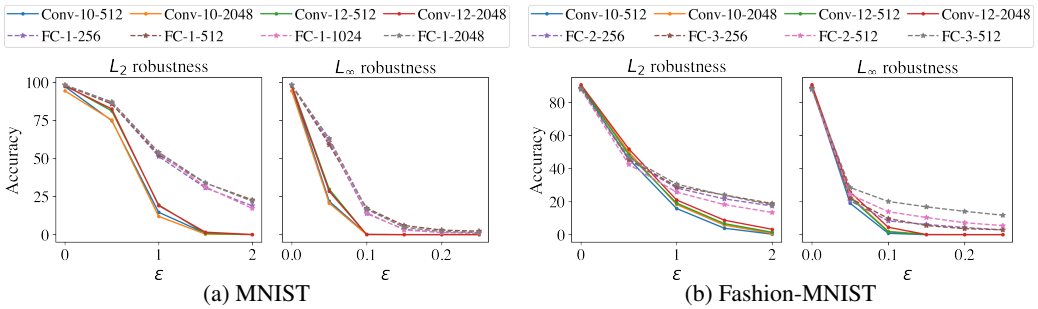
Theorem 1. *Let S_1 and S_2 denote the sets of all shifts of X_1 and X_2, as described above. They are linearly separable if and only if $\max_{\mathbf{x_1} \in S_1} f_{dc}(\mathbf{x_1}) < \min_{\mathbf{x_2} \in S_2} f_{dc}(\mathbf{x_2})$ or $\max_{\mathbf{x_2} \in S_2} f_{dc}(\mathbf{x_2}) < \min_{\mathbf{x_1} \in S_1} f_{dc}(\mathbf{x_1})$. Furthermore, if the two classes are linearly separable then, if the first inequality holds, the margin is $\min_{\mathbf{x_2} \in S_2} f_{dc}(\mathbf{x_2}) - \max_{\mathbf{x_1} \in S_1} f_{dc}(\mathbf{x_1})$, and similarly if the second inequality holds. Furthermore, the max margin separating hyperplane has a normal of $\bar{\mathbf{w}}$.*

The proof of the above theorem appears in Section B. Based on this theorem, we now consider the example shown in Figure 1. Let $\mathbf{x}_1$ denote an image consisting of a single 1 in a background of 0s, and let $\mathbf{x}_2$ denote an image containing a single -1 with 0s. Since $f_{dc}(\mathbf{x}_1) = \frac{1}{\sqrt{d}}$ and $f_{dc}(\mathbf{x}_2) = -\frac{1}{\sqrt{d}}$, if we let $X_1 = \{\mathbf{x}_1\}$ and $X_2 = \{\mathbf{x}_2\}$ it is straightforward to show:

Corollary 1. *X_1 and X_2 defined above are linearly separable with a max margin of 2. $\mathcal{SH}(X_1)$ and $\mathcal{SH}(X_2)$ are linearly separable with a max margin of $\frac{2}{\sqrt{d}}$ and a separating hyperplane with the normal vector $\bar{\mathbf{w}}$.*

3 Shift Invariance and Adversarial Attacks for Neural Networks: NTK vs. CNTK

To understand the robustness of shift invariant neural networks we examine the behavior of the neural tangent kernel (NTK) for two networks with simple inputs. It has been shown that neural networks



(a) MNIST (b) Fashion-MNIST

Figure 2: Robustness of Shift Invariant CNNs vs FC networks.

with infinite width (or convolutional networks with infinite number of channels) behave like kernel regression with the family of kernels called NTKs (Jacot et al., 2018). Here we consider the NTK for a two-layer, bias-free fully connected network, denoted FC-NTK, and for a two-layer bias-free convolutional network with global average pooling, denoted CNTK-GAP (Li et al., 2019).

In our theorem below we use both FC-NTK and CNTK-GAP to construct two-class classifiers with a training set composed of two antipodal training points, $\mathbf{x}, -\mathbf{x} \in \mathbb{R}^d$. In both cases the kernels produce linear separators, but while FC-NTK produces a separator with constant margin, independent of input dimension, due to shift invariance CNTK-GAP results in a margin that decreases with $2/\sqrt{d}$, consistent with our results in Figure 1. Due to space considerations, the definition of NTK, CNTK and their corresponding network architectures, the minimum norm interpolant $g_k(\mathbf{z})$ of the kernel regression with kernel k, along with the proof of the theorem below are deferred to Section C.

Theorem 2. *Let $\mathbf{x}, -\mathbf{x} \in \mathbb{R}^d$ be two training vectors with class labels $1, -1$ respectively.*

1. *Let $k(\mathbf{z}, \mathbf{x})$ denote NTK for the bias-free, two-layer fully connected network. Then $\forall \mathbf{z} \in \mathbb{R}^d$, the minimum norm interpolant $g_k(\mathbf{z}) \geq 0$ iff $\mathbf{z}^T\mathbf{x} \geq 0$.*

2. *Let $K(\mathbf{z}, \mathbf{x})$ denote CNTK-GAP for the bias-free, two-layer convolutional network defined in (14), and assume H_K is invertible. Then $\forall \mathbf{z} \in \mathbb{R}^d$, either $g_K(\mathbf{z}) \geq 0$ iff $\mathbf{z}^T\mathbf{1}_d \geq 0$ or $g_K(\mathbf{z}) \geq 0$ iff $\mathbf{z}^T\mathbf{1}_d \leq 0$. (I.e., $\mathbf{z}^T\mathbf{1}_d = 0$ forms a separating hyperplane.)*

The theorem tells us that NTK and CNTK produce linear classifiers. (1) tells us that NTK produces a separating hyperplane with a normal vector x, while (2) says that for CNTK the normal direction is $\mathbf{1}_d$. The following corollary follows directly from Thm 1, which explains the results in Figure 1.

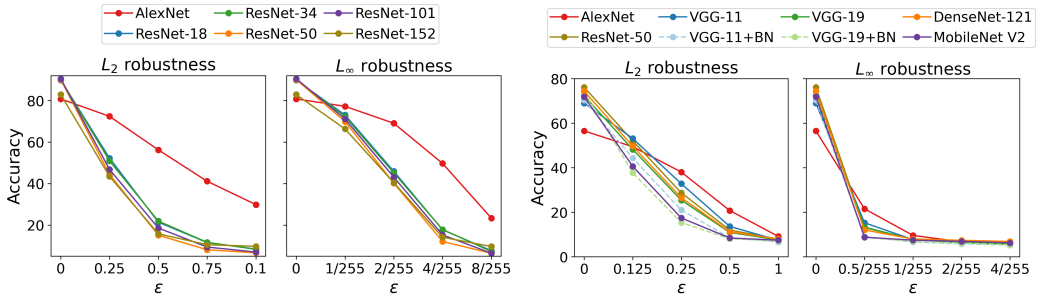
Corollary 2. *With a training set composed of an antipodal pair the margin obtained with CNTK is the difference between their DC components.*

4 Experiments

We have shown theoretically that shift invariance can reduce adversarial robustness. In this section we describe experiments that indicate that this does occur with real datasets and network architectures.

To start with, we compare the adversarial robustness of a fully shift-invariant convolutional neural network (CNN) with a fully connected (FC) network. We consider two datasets in which FC networks are able to attain reasonable performance compared to CNNs, MNIST (LeCun et al., 2010) and Fashion-MNIST (Xiao et al., 2017). We use various shift invariant CNN and FC networks for our experiments. The notation FC-X-Y denotes an FC network with X hidden layers and Y units in each hidden layer. Conv-X-Y denotes a network with a single convolutional layer with X kernel size and Y filters with stride 1 and circular padding. All networks use ReLU activation. CNNs have a penultimate layer that performs Global Average Pooling, so these networks are fully shift invariant. Finally, a fully connected layer is applied at the end of all the CNNs, which outputs the final logits. We report results for clean[1] and robust test accuracy for l_2 and l_∞ attacks at different ϵ values in Figure 2a and 2b for MNIST and Fashion-MNIST respectively. **We observe that although the clean accuracy for the models is similar on the datasets, FC networks are more robust than Shift Invariant CNNs, especially for large ϵ values.**

[1]Clean Accuracy of models is shown by $\epsilon = 0$.



(a) Cifar10 (b) ImageNet

Figure 3: Robustness of models on Cifar-10 and ImageNet datasets. "BN" indicates that the model uses batch normalization.

Next, we consider the robustness of realistic networks on large scale datasets. We do not have a firm theoretical basis for comparing the shift-invariance of these networks, since none are truly (non-) shift invariant as we will discuss in Section D.2. We start by comparing ResNets with previously introduced FC networks on SVHN (Netzer et al., 2011), where ResNets generate descent clean accuracy while FC networks do not. However, we find that ResNets practically lead to lower accuracy under adversarial attack than FC networks, especially when ϵ is large. More details are discussed in Section D.1.

On more challenging datasets such as Cifar10 and ImageNet, FC networks cannot produce comparable clean accuracy, which makes them no longer appropriate representatives of non-shift invariant classifiers for robustness comparison. (Zhang, 2019) introduces an empirical consistency measure of relative shift invariance, i.e. the percentage of the time that the model classifies two different shifts of the same image to be the same class, which we use to distinguish among models. Although it is not clear that this fully captures our notion of shift invariance, it does support the view that AlexNet is by far the least shift-invariant large-scale CNN due to violation to most of the assumptions needed for shift invariant classifier. Specifically, on ImageNet dataset, the consistency of all the models except for AlexNet are over 85, while the consistency of AlexNet is only 78.11 (Zhang, 2019). We apply l_2 and l_∞ PGD attacks to the pretrained models provided in the Pytorch model zoo on ImageNet. The accuracy of a few representative models under different attack strengths are shown in Figure 3a. **It clearly illustrates that AlexNet is indeed an outlier in terms of both shift invariance and robustness**.

To further demonstrate this, we repeat the experiments on Cifar10 with realistic architectures and report test accuracy under different attacks in Figure 3b. We also evaluate the consistency of the models to shifts and find that all the ResNet models have consistency larger than 75 while the consistency of AlexNet is only 34.79. In addition, as shown in Figure 3b, the clean accuracy of AlexNet is lower than the ResNets while it decreases much slower as the attack strength increases, **showing that the less shift invariant classifier, AlexNet, achieves better robustness**.

We refer the readers to Section D.1 for all the training details for different experiments mentioned above and results on the SVHN dataset. In Section A, we summarize previous studies on the cause of adversarial examples, shift invariance and NTK. In addition to the experiments that serve as empirical evidence that shift invariance can reduce adversarial robustness, we also provide experiments on simple, synthetic datasets in Section D.3 to begin to address the question of why this might occur.

5 Conclusion

We have shown theoretically and experimentally that shift invariance can reduce adversarial robustness in neural networks. We prove that for linear classifiers, shift invariance reduces the margin between classes to just the difference in their DC components. Using this, we construct a simple, intuitive example in which shift invariance dramatically reduces robustness. Our experiments provide evidence that this reduction in robustness shows up in real networks as well. Finally, our experiments on synthetic data provide a step towards understanding the relationship between robustness, shift invariance, dimensionality and the linear margin of the data, in neural networks.

REFERENCES

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pp. 6158–6169, 2019a.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019b.

Bilal Alsallakh, Narine Kokhlikyan, Vivek Miglani, Jun Yuan, and Orion Reblitz-Richardson. Mind the pad – cnns can develop blind spots, 2020.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pp. 8141–8150, 2019.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, (20):1–25, 2019.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems*, pp. 12893–12904, 2019.

Anadi Chaman and Ivan Dokmanić. Truly shift-invariant convolutional neural networks, 2020.

Lin Chen, Yifei Min, Mingrui Zhang, and Amin Karbasi. More data can expand the generalization gap between adversarially robust and standard models. *arXiv preprint arXiv:2002.04725*, 2020.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Amit Daniely and Hadas Schacham. Most relu networks suffer from ℓ^2 adversarial perturbations, 2020.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pp. 1675–1685. PMLR, 2019.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2018.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.

Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch normalization is a cause of adversarial vulnerability. *arXiv preprint arXiv:1905.02161*, 2019.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*, 2015.

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.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Jörn-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge. Excessive invariance causes adversarial vulnerability. *arXiv preprint arXiv:1811.00401*, 2018.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Sandesh Kamath, Amit Deshpande, and K V Subrahmanyam. Invariance vs robustness of neural networks, 2020. URL https://openreview.net/forum?id=HJxp9kBFDS.

Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14274–14285, 2020.

Marc Khoury and Dylan Hadfield-Menell. On the geometry of adversarial examples. *International Conference on Learning Representations*, 2019.

Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Zhiyuan Li, Ruosong Wang, Dingli Yu, Simon S Du, Wei Hu, Ruslan Salakhutdinov, and Sanjeev Arora. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Preetum Nakkiran. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Jerome Rony, Luiz G. Hafemann, Luiz S. Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Saburou Saitoh and Yoshihiro Sawano. *Theory of reproducing kernels and applications*. Springer, 2016.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pp. 5014–5026, 2018.

Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *International Conference on Learning Representations*, 2019.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Adi Shamir, Itay Safran, Eyal Ronen, and Orr Dunkelman. A simple explanation for the existence of adversarial examples with small hamming distance, 2019.

Carl-Johann Simon-Gabriel, Yann Ollivier, Leon Bottou, Bernhard Schölkopf, and David Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5809–5817. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/simon-gabriel19a.html.

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.

Julián Tachella, Junqi Tang, and Mike Davies. The neural tangent link between cnn denoisers and non-local filters, 2020.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8684–8694, 2020.

Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu. Does network width really help adversarial robustness? *arXiv preprint arXiv:2010.01279*, 2020a.

Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *European Conference on Computer Vision*, pp. 1–17. Springer, 2020b.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

Richard Zhang. Making convolutional networks shift-invariant again. *arXiv preprint arXiv:1904.11486*, 2019.

A Related Work

In this section we survey prior work on the origin of adversarial examples and briefly discuss work on shift-invariance in CNNs, and on the NTKs.

One common explanation for adversarial examples lies in the curse of dimensionality. (Goodfellow et al., 2015) argues that even with a linear classifier a lack of adversarial robustness can be explained by high input dimension. (Simon-Gabriel et al., 2019) expands on this by studying how gradient magnitude grows with dimension. (Gilmer et al., 2018), (Shafahi et al., 2019), (Daniely & Schacham, 2020) and (Khoury & Hadfield-Menell, 2019) provide additional insight into the role of dimension in adversarial examples, while (Shamir et al., 2019) shows that for ReLU networks, adversarial examples arise as a consequence of the geometry of $\mathbb{R}^n$.

Some studies show that adversarial examples could arise due to properties of the data, such as high frequency components (Wang et al., 2020) or biases in the features (Tsipras et al., 2018). (Schmidt et al., 2018) shows that in some cases robustness requires larger sample complexity, while (Chen et al., 2020) shows that in some cases more data can undermine robustness.

Other work suggests that susceptibility to adversarial attack may be due to aspects of the model, such as insufficient complexity (Nakkiran, 2019), width (Madry et al., 2018; Wu et al., 2020a) or batch normalization(Galloway et al., 2019). (Shah et al., 2020) shows that in many cases networks learn simple decision boundaries with small margins, rather than more complex ones with large margins, inducing adversarial vulnerability.

Perhaps most relevant to our work, (Kamath et al., 2020) examine the interplay between rotation invariance and robustness. They empirically show that when they induce more rotational invariance using data augmentation in CNNs and Group-equivariant CNNs (Cohen & Welling, 2016) adversarial robustness decreases. A quite different notion of invariance is explored in (Jacobsen et al., 2018).

Several papers have pointed out that the modern CNN architectures are not fully shift invariant for reasons such as padding mode, striding, and padding size (e.g. (Alsallakh et al., 2020)). (Azulay & Weiss, 2019) relates a lack of shift invariance to a failure to respect the sampling theorem. Several papers have suggested methods for enhancing shift invariance, including (Zhang, 2019), (Kayhan & Gemert, 2020), and (Chaman & Dokmanić, 2020)

Some of our theoretical results draw on recent work on the NTK (Jacot et al., 2018) and CNTKs (Arora et al., 2019; Li et al., 2019). These and other recent studies have shown that when networks are over-parameterized, their parameters stay close to their initialization during training, allowing a linearization of the network that makes the analysis of convergence and generalization tractable (Allen-Zhu et al., 2019a;b; Du et al., 2018; 2019). When optimizing the quadratic loss using gradient descent, the dynamics of an infinitely wide neural network can be described by kernel regression. We describe these kernels in more detail in Section 3.

B PROOF OF THEOREM 1

In this section, we prove Theorem 1 on the shift invariance and linear margin. We call a linear classifier shift invariant when it places all possible shifts of a signal in the same class. We prove that for a shift invariant linear classifier the margin will depend only on differences in the DC components of the training signals. It follows that for the two classes shown in Figure 1, the margin of a linear, shift invariant classifier will shrink in proportion to $\frac{1}{\sqrt{d}}$, where d is the number of image pixels.

Theorem 1. *Let S_1 and S_2 denote the sets of all shifts of X_1 and X_2, as described above. They are linearly separable if and only if $\max_{\mathbf{x_1}\in S_1} f_{dc}(\mathbf{x_1}) < \min_{\mathbf{x_2}\in S_2} f_{dc}(\mathbf{x_2})$ or $\max_{\mathbf{x_2}\in S_2} f_{dc}(\mathbf{x_2}) < \min_{\mathbf{x_1}\in S_1} f_{dc}(\mathbf{x_1})$. Furthermore, if the two classes are linearly separable then, if the first inequality holds, the margin is $\min_{\mathbf{x_2}\in S_2} f_{dc}(\mathbf{x}_2) - \max_{\mathbf{x_1}\in S_1} f_{dc}(\mathbf{x}_1)$, and similarly if the second inequality holds. Furthermore, the max margin separating hyperplane has a normal of $\bar{\mathbf{w}}$.*

Proof. We only consider the case in which $\max_{\mathbf{x}_1\in S_1} f_{dc}(\mathbf{x}_1) < \min_{\mathbf{x}_2\in S_2} f_{dc}(\mathbf{x}_2)$, without loss of generality. Two classes are linearly separable iff there exists a d-dimensional unit vector $\mathbf{w}$ and a threshold T such that: $\forall \mathbf{x}_1 \in S_1, \mathbf{w}^T\mathbf{x}_1 \leq T$ and $\forall \mathbf{x}_2 \in S_2, \mathbf{w}^T\mathbf{x}_2 \geq T$, and say the margin is: $\min_{\mathbf{x}_1\in S_1, \mathbf{x}_2\in S_2} \mathbf{w}^T\mathbf{x}_2 - \mathbf{w}^T\mathbf{x}_1$.

First, it is obvious that if $\max_{\mathbf{x_1}\in S_1} f_{dc}(\mathbf{x}_1) < \min_{\mathbf{x_2}\in S_2} f_{dc}(\mathbf{x}_2)$ then S_1 and S_2 are linearly separable, by letting $\mathbf{w} = \bar{\mathbf{w}}$, and

$$T = \max_{\mathbf{x_1}\in S_1} f_{dc}(\mathbf{x}_1) + \frac{\min_{\mathbf{x_2}\in S_2} f_{dc}(\mathbf{x}_2) - \max_{\mathbf{x_1}\in S_1} f_{dc}(\mathbf{x}_1)}{2}$$

The margin for $\bar{\mathbf{w}}$ is $\min_{\mathbf{x_2}\in S_2} f_{dc}(\mathbf{x}_2) - \max_{\mathbf{x_1}\in S_1} f_{dc}(\mathbf{x}_1)$.

Second, we consider the opposite direction, supposing that the two classes are linearly separable. In this case there exist some $\mathbf{w}$ and T that will separate the classes. Let $\mathbf{x}_1 \in X_1$. Then we have WLOG: $\forall s \quad \mathbf{w}^T\mathbf{x}_1^s < T$. This gives:

$$\frac{\sum_{s=0}^{d-1} \mathbf{w}^T\mathbf{x}_1^s}{d} < T \implies \frac{\mathbf{w}^T \sum_{s=0}^{d-1} \mathbf{x}_1^s}{d} < T$$

Note that $\forall \mathbf{x} \in \mathbb{R}^d$, $\sum_{s=0}^{d-1} \mathbf{x}^s$ is just a constant vector of length d with each term equal to $\sqrt{d} f_{dc}(\mathbf{x})$. Therefore:

$$\frac{\sum_{s=0}^{d-1} \mathbf{w}^T\mathbf{x}^s}{d} = f_{dc}(\mathbf{w}) f_{dc}(\mathbf{x}) \tag{1}$$

so

$$f_{dc}(\mathbf{w}) f_{dc}(\mathbf{x}_1) < T \quad \forall \mathbf{x}_1 \in X_1$$

By similar reasoning, we have:

$$f_{dc}(\mathbf{w}) f_{dc}(\mathbf{x}_2) > T \quad \forall \mathbf{x}_2 \in X_2$$

Because $f_{dc}(\mathbf{x}_1) < f_{dc}(\mathbf{x}_2)$ we have $f_{dc}(\mathbf{w}) > 0$. So:

$$\max_{\mathbf{x}_1\in X_1} f_{dc}(\mathbf{x}_1) < \frac{T}{f_{dc}(\mathbf{w})} \quad \text{and} \quad \min_{\mathbf{x}_2\in X_2} f_{dc}(\mathbf{x}_2) > \frac{T}{f_{dc}(\mathbf{w})}$$

so

$$\max_{x_1 \in X_1} f_{dc}(\mathbf{x}_1) < \min_{x_2 \in X_2} f_{dc}(\mathbf{x}_2) \tag{2}$$

This shows that S_1 and S_2 are linearly separable if and only if their DC components are separable.

We now show that if S_1 and S_2 are linearly separable, for the max margin separator we have $\mathbf{w} = \bar{\mathbf{w}}$, with a margin of $\min_{\mathbf{x}_1 \in S_2} f_{dc}(\mathbf{x}_2) - \max_{\mathbf{x}_1 \in S_1} f_{dc}(\mathbf{x}_1)$. Because $\bar{\mathbf{w}}^T \mathbf{x} = f_{dc}(\mathbf{x})$, and the DC components are separable, $\bar{\mathbf{w}}$ separates the data, and we can see that the margin will be $\min_{\mathbf{x} \in S_2} \mathbf{x}_{dc} - \max_{\mathbf{x} \in S_1} \mathbf{x}_{dc}$.

So, it remains to show that no other choice of $\mathbf{w}$ separates the data with a larger margin. Assume, WLOG that $\|\mathbf{w}\| = 1$ and that the margin is:

$$\min_{\mathbf{x}_2 \in S_2} \mathbf{w}^T \mathbf{x}_2 - \max_{\mathbf{x}_1 \in S_1} \mathbf{w}^T \mathbf{x}_1$$

which, as above, implies $f_{dc}(\mathbf{w}) > 0$. We will show that $\forall \mathbf{x}_1 \in X_1, \mathbf{x}_2 \in X_2$ and $\forall \mathbf{w}$ such that $\|\mathbf{w}\| = 1$:

$$\min_s \mathbf{w}^T \mathbf{x}_2^s - \max_s \mathbf{w}^T \mathbf{x}_1^s \leq \min_s \bar{\mathbf{w}}^T \mathbf{x}_2^s - \max_s \bar{\mathbf{w}}^T \mathbf{x}_1^s \tag{3}$$

which implies that the margin from $\bar{\mathbf{w}}$ is greater than or equal to the margin from $\mathbf{w}$. We note that $\min_s \bar{\mathbf{w}}^T \mathbf{x}_2^s - \max_s \bar{\mathbf{w}}^T \mathbf{x}_1^s = f_{dc}(\mathbf{x}_2) - f_{dc}(\mathbf{x}_1)$.

From Equation 1 we know that

$$\max_s \mathbf{w}^T \mathbf{x}^s \geq f_{dc}(\mathbf{w}) f_{dc}(\mathbf{x}), \qquad \min_s \mathbf{w}^T \mathbf{x}^s \leq f_{dc}(\mathbf{w}) f_{dc}(\mathbf{x})$$

This implies that:

$$\frac{\min_s \mathbf{w}^T \mathbf{x}_2 - \max_s \mathbf{w}^T \mathbf{x}_1}{f_{dc}(\mathbf{w})} \leq f_{dc}(\mathbf{x}_2) - f_{dc}(\mathbf{x}_1)$$

Given the constraint that $\|\mathbf{w}\| = 1$, $f_{dc}(\mathbf{w})$ is minimized by $\bar{\mathbf{w}}$, and so $f_{dc}(\mathbf{w}) \geq 1$, and Eq. 3 is shown to hold.

□

C Proof of Theorem 2 and Lemma 1

In this section, we first define the FC networks with the neural tangent kernel (NTK) (Jacot et al., 2018) and CNNs with a convolutional neural tangent kernel (CNTK) (Arora et al., 2019; Li et al., 2019) which we will use in the proof. Then we give the proofs of Theorem 2 and Lemma 1.

Let $\mathbf{x} \in \mathbb{R}^d$ denote the input to the network. A two-layer fully connected network is defined by

$$f_{\mathrm{FC}}(\mathbf{x}; W, \mathbf{v}) = \mathbf{v}^T \sigma(W\mathbf{x}), \tag{4}$$

where $W \in \mathbb{R}^{m \times d}$ and $\mathbf{v} \in \mathbb{R}^m$ are learnable parameters and $\sigma(.)$ is the ReLU function applied elementwise. Assuming W and $\mathbf{v}$ are initialized with normal distribution, the corresponding FC-NTK for inputs $\mathbf{z}, \mathbf{x} \in \mathbb{R}^d$ is given by (Bietti & Mairal, 2019)

$$k(\mathbf{z}, \mathbf{x}) = \frac{1}{\pi} \left(2\mathbf{z}^T \mathbf{x}(\pi - \phi) + \|\mathbf{z}\| \|\mathbf{x}\| \sin \phi \right), \tag{5}$$

where ϕ denotes the angle between $\mathbf{z}$ and $\mathbf{x}$, i.e., $\phi = \arccos\left(\frac{\mathbf{z}^T \mathbf{x}}{\|\mathbf{z}\| \|\mathbf{x}\|} \right)$.

Next we define the shift invariant convolutional model. Given an input $\mathbf{x} \in \mathbb{R}^d$ and filters $\{\mathbf{w}_i\}_{i=1}^m \subset \mathbb{R}^q$ we denote by $\mathbf{w}_i * \mathbf{x} \in \mathbb{R}^d$ the circular convolution of $\mathbf{x}$ with the filter $\mathbf{w}_i$ (with no bias). $W * \mathbf{x}$ denotes the results of these convolutions, represented as an $m \times d$ matrix, with the $m \times q$ matrix W denoting the collection of all filters $\{\mathbf{w}_i\}_{i=1}^m$. Finally, let $\mathbf{v} \in \mathbb{R}^m$. Then a two layer convolutional network with global average pooling is defined by

$$f_{\mathrm{Conv}}(\mathbf{x}; W, \mathbf{v}) = \frac{1}{d} \mathbf{v}^T \sigma(W * \mathbf{x})) \mathbf{1}_d, \tag{6}$$

W and $\mathbf{v}$ include the learnable parameters initialized with the standard normal distribution and $\mathbf{1}_d \in \mathbb{R}^d$ is the vector of all ones. In this model the input $\mathbf{x}$ is convolved with the rows of W. After ReLU the result undergoes a 1×1 convolution with parameters $\mathbf{v}$ followed by global average pooling, captured by the multiplication with $\frac{1}{d}\mathbf{1}_d$.

Given inputs $\mathbf{z}, \mathbf{x} \in \mathbb{R}^d$, denote by $\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_j \in \mathbb{R}^q$ their (cyclic) patches, $1 \leq i, j \leq d$, so for example $\bar{\mathbf{z}}_i = (z_i, z_{(i+1) \bmod d}, ...z_{(i+q-1) \bmod d})^T$. Then the corresponding CNTK-GAP $K(\mathbf{z}, \mathbf{x})$ is constructed as follows.

$$K(\mathbf{z}, \mathbf{x}) = \frac{1}{d^2} \sum_{i=1}^{d} \sum_{j=1}^{d} k(\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_j), \tag{7}$$

where $k(\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_j)$ is the FC-NTK given by (12) (see a related construction in (Tachella et al., 2020)).

We use FC-NTK and CNTK-GAP in kernel regression. Given training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathbb{R}$, kernel ridge regression is the solution to (Saitoh & Sawano, 2016)

$$g_k = \arg\min_{g \in \mathcal{H}_k} \sum_{i=1}^{n} (g(\mathbf{x}_i) - y_i)^2 + \lambda \|g\|_{\mathcal{H}_k}^2, \tag{8}$$

where $\mathcal{H}_k$ denotes the reproducing kernel Hilbert space associated with k. The solution of (15) is given by

$$g_k(\mathbf{z}) = (k(\mathbf{z}, \mathbf{x}_1), ..., k(\mathbf{z}, \mathbf{x}_n))(H_k + \lambda I)^{-1}\mathbf{y}, \tag{9}$$

where H_k is the $n \times n$ matrix with its i, j'th entry $k(\mathbf{x}_i, \mathbf{x}_j)$, I denotes the identity matrix, and $\mathbf{y} = (y_1, ..., y_n)^T$. Below we consider the minimum norm interpolant, i.e.,

$$g_k = \arg\min_{g \in \mathcal{H}_k} \|g\|_{\mathcal{H}_k} \quad \text{s.t.} \quad \forall i,\ g(\mathbf{x}_i) = y_i. \tag{10}$$

which is obtained when we let $\lambda \to 0$.

Let $\mathbf{x} \in \mathbb{R}^d$ denote the input to the network. A two-layer fully connected network is defined by

$$f_{\mathrm{FC}}(\mathbf{x}; W, \mathbf{v}) = \mathbf{v}^T \sigma(W\mathbf{x}), \tag{11}$$

where $W \in \mathbb{R}^{m \times d}$ and $\mathbf{v} \in \mathbb{R}^m$ are learnable parameters and $\sigma(.)$ is the ReLU function applied elementwise. Assuming W and $\mathbf{v}$ are initialized with normal distribution, the corresponding FC-NTK for inputs $\mathbf{z}, \mathbf{x} \in \mathbb{R}^d$ is given by (Bietti & Mairal, 2019)

$$k(\mathbf{z}, \mathbf{x}) = \frac{1}{\pi} \left(2\mathbf{z}^T\mathbf{x}(\pi - \phi) + \|\mathbf{z}\|\|\mathbf{x}\| \sin\phi \right), \tag{12}$$

where ϕ denotes the angle between $\mathbf{z}$ and $\mathbf{x}$, i.e., $\phi = \arccos\left(\frac{\mathbf{z}^T\mathbf{x}}{\|\mathbf{z}\|\|\mathbf{x}\|}\right)$.

Next we define the shift invariant convolutional model. Given an input $\mathbf{x} \in \mathbb{R}^d$ and filters $\{\mathbf{w}_i\}_{i=1}^m \subset \mathbb{R}^q$ we denote by $\mathbf{w}_i * \mathbf{x} \in \mathbb{R}^d$ the circular convolution of $\mathbf{x}$ with the filter $\mathbf{w}_i$ (with no bias). $W * \mathbf{x}$ denotes the results of these convolutions, represented as an $m \times d$ matrix, with the $m \times q$ matrix W denoting the collection of all filters $\{\mathbf{w}_i\}_{i=1}^m$. Finally, let $\mathbf{v} \in \mathbb{R}^m$. Then a two layer convolutional network with global average pooling is defined by

$$f_{\mathrm{Conv}}(\mathbf{x}; W, \mathbf{v}) = \frac{1}{d}\mathbf{v}^T \sigma(W * \mathbf{x}))\mathbf{1}_d, \tag{13}$$

W and $\mathbf{v}$ include the learnable parameters initialized with the standard normal distribution and $\mathbf{1}_d \in \mathbb{R}^d$ is the vector of all ones. In this model the input $\mathbf{x}$ is convolved with the rows of W. After ReLU the result undergoes a 1×1 convolution with parameters $\mathbf{v}$ followed by global average pooling, captured by the multiplication with $\frac{1}{d}\mathbf{1}_d$.

Given inputs $\mathbf{z}, \mathbf{x} \in \mathbb{R}^d$, denote by $\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_j \in \mathbb{R}^q$ their (cyclic) patches, $1 \leq i, j \leq d$, so for example $\bar{\mathbf{z}}_i = (z_i, z_{(i+1) \bmod d}, ...z_{(i+q-1) \bmod d})^T$. Then the corresponding CNTK-GAP $K(\mathbf{z}, \mathbf{x})$ is constructed as follows.

$$K(\mathbf{z}, \mathbf{x}) = \frac{1}{d^2} \sum_{i=1}^{d} \sum_{j=1}^{d} k(\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_j), \tag{14}$$

where $k(\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_j)$ is the FC-NTK given by (12) (see a related construction in (Tachella et al., 2020)).

We use FC-NTK and CNTK-GAP in kernel regression. Given training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathbb{R}$, kernel ridge regression is the solution to (Saitoh & Sawano, 2016)

$$g_k = \arg\min_{g \in \mathcal{H}_k} \sum_{i=1}^{n} (g(\mathbf{x}_i) - y_i)^2 + \lambda \|g\|^2_{\mathcal{H}_k}, \tag{15}$$

where $\mathcal{H}_k$ denotes the reproducing kernel Hilbert space associated with k. The solution of (15) is given by

$$g_k(\mathbf{z}) = (k(\mathbf{z}, \mathbf{x}_1), ..., k(\mathbf{z}, \mathbf{x}_n))(H_k + \lambda I)^{-1}\mathbf{y}, \tag{16}$$

where H_k is the $n \times n$ matrix with its i, j'th entry $k(\mathbf{x}_i, \mathbf{x}_j)$, I denotes the identity matrix, and $\mathbf{y} = (y_1, ..., y_n)^T$. Below we consider the minimum norm interpolant, i.e.,

$$g_k = \arg\min_{g \in \mathcal{H}_k} \|g\|_{\mathcal{H}_k} \quad \text{s.t.} \quad \forall i,\ g(\mathbf{x}_i) = y_i. \tag{17}$$

which is obtained when we let $\lambda \to 0$.

Theorem 2. *Let $\mathbf{x}, -\mathbf{x} \in \mathbb{R}^d$ be two training vectors with class labels $1, -1$ respectively.*

1. *Let $k(\mathbf{z}, \mathbf{x})$ denote NTK for the bias-free, two-layer fully connected network. Then $\forall \mathbf{z} \in \mathbb{R}^d$, the minimum norm interpolant $g_k(\mathbf{z}) \geq 0$ iff $\mathbf{z}^T\mathbf{x} \geq 0$.*
2. *Let $K(\mathbf{z}, \mathbf{x})$ denote CNTK-GAP for the bias-free, two-layer convolutional network, and assume H_K is invertible. Then $\forall \mathbf{z} \in \mathbb{R}^d$, either $g_K(\mathbf{z}) \geq 0$ iff $\mathbf{z}^T\mathbf{1}_d \geq 0$ or $g_K(\mathbf{z}) \geq 0$ iff $\mathbf{z}^T\mathbf{1}_d \leq 0$. (I.e., $\mathbf{z}^T\mathbf{1}_d = 0$ forms a separating hyperplane.)*

The theorem tells us that NTK and CNTK produce linear classifiers. (1) tells us that NTK produces a separating hyperplane with a normal vector x, while (2) says that for CNTK the normal direction is $\mathbf{1}_d$.

Proof. 1. Solving the regression problem (16) with $\lambda \to 0$ we have

$$H_k = \begin{pmatrix} k(\mathbf{x}, \mathbf{x}) & k(\mathbf{x}, -\mathbf{x}) \\ k(-\mathbf{x}, \mathbf{x}) & k(-\mathbf{x}, -\mathbf{x}) \end{pmatrix} = 2\mathbf{x}^T\mathbf{x} I,$$

where the latter equality is obtained from (12) by noting that $\phi = 0$ along the diagonal and $\phi = \pi$ for the off-diagonal entries. Therefore, using (16) and noting that $\mathbf{y} = (1, -1)^T$,

$$g_k(\mathbf{z}) = \frac{1}{2x^T x}(k(\mathbf{z}, \mathbf{x}) - k(\mathbf{z}, -\mathbf{x}))$$

Given a test point $\mathbf{z} \in \mathbb{R}^d$, let ϕ now denote the angle between $\mathbf{z}$ and $\mathbf{x}$ and note that the angle between $\mathbf{z}$ and $-\mathbf{x}$ is $\pi - \phi$. Therefore,

$$\begin{aligned} k(\mathbf{z}, \mathbf{x}) &= \frac{1}{\pi}\left(2\mathbf{z}^T\mathbf{x}(\pi - \phi) + \|\mathbf{z}\|\|\mathbf{x}\|\sin\phi\right) \\ k(\mathbf{z}, -\mathbf{x}) &= \frac{1}{\pi}\left(-2\mathbf{z}^T\mathbf{x}\phi + \|\mathbf{z}\|\|\mathbf{x}\|\sin\phi\right), \end{aligned}$$

implying that

$$k(\mathbf{z}, \mathbf{x}) - k(\mathbf{z}, -\mathbf{x}) = 2\mathbf{z}^T\mathbf{x}. \tag{18}$$

from which we obtain $g_k(\mathbf{z}) = \frac{\mathbf{z}^T\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$. Consequently, $g_k(\mathbf{z}) > 0$ if and only if $\mathbf{z}^T\mathbf{x} > 0$.

2. Using the definition of K (Eq. 14) it is clear that $K(\mathbf{x}, \mathbf{x}) = K(-\mathbf{x}, -\mathbf{x})$. Therefore, using Lemma 1 we need to show that $K(\mathbf{z}, \mathbf{x}) > K(\mathbf{z}, -\mathbf{x})$ on one side of the plane $\mathbf{z}^T\mathbf{1}_d = 0$. Consider the patches $\bar{\mathbf{z}}_i$ in $\mathbf{z}$ and $\bar{\mathbf{x}}_j$ in $\mathbf{x}$. From (18) we have

$$k(\bar{\mathbf{z}}_i, \bar{\mathbf{x}}_j) - k(\bar{\mathbf{z}}_i, -\bar{\mathbf{x}}_j) = 2\bar{\mathbf{z}}_i^T\bar{\mathbf{x}}_j,$$

implying that

$$K(\mathbf{z}, \mathbf{x}) - K(\mathbf{z}, -\mathbf{x}) = \frac{2}{d^2}\sum_{i=1}^{d}\sum_{j=1}^{d}\bar{\mathbf{z}}_i^T\bar{\mathbf{x}}_j.$$

Rewriting this in matrix notation we have

$$K(\mathbf{z}, \mathbf{x}) - K(\mathbf{z}, -\mathbf{x}) = \frac{2}{d^2} \mathbf{1}_d^T Z^T X \mathbf{1}_d,$$

where Z and X are $q \times d$ matrices whose columns respectively contain all the patches of $\mathbf{z}$ and $\mathbf{x}$. Since all rows of Z and X are identical up to a cyclic permutation $\hat{z} = Z\mathbf{1}_d$ and $\hat{x} = X\mathbf{1}_d$ are vectors of constants in $\mathbb{R}^q$ with the constants $\mathbf{z}^T\mathbf{1}_d$ and $\mathbf{x}^T\mathbf{1}_d$ respectively. Consequently, using Lemma 1

$$g_K(\mathbf{z}) = c(K(\mathbf{z}, \mathbf{x}) - K(\mathbf{z}, -\mathbf{x})) = \frac{2cq}{d^2}(\mathbf{z}^T\mathbf{1}_d)(\mathbf{x}^T\mathbf{1}_d).$$

where, because K is positive definite and H_K is invertible, $c = 1/(K(\mathbf{x}, \mathbf{x}) - K(\mathbf{x}, -\mathbf{x})) > 0$. Denoting $\beta = \frac{2cq}{d^2}(\mathbf{x}^T\mathbf{1}_d)$, we obtain that $g_K(\mathbf{z}) > 0$ if and only if $\operatorname{sign}(\beta)\, \mathbf{z}^T\mathbf{1}_d > 0$, proving the theorem.

□

The following lemma was used to prove Thm. 2.

Lemma 1. *Let $k(.,.)$ be a positive definite kernel with a training set $\{(\mathbf{x}_1, +1), (\mathbf{x}_2, -1)\} \subset \mathbb{R}^d \times \mathbb{R}$. If $k(\mathbf{x}_1, \mathbf{x}_1) = k(\mathbf{x}_2, \mathbf{x}_2)$ and H_k is invertible with $\lambda \to 0$ then a test point $\mathbf{z} \in \mathbb{R}^d$ is classified as +1 if and only $k(\mathbf{z}, \mathbf{x}_1) > k(\mathbf{z}, \mathbf{x}_2)$.*

Proof. Denote by $a = k(\mathbf{x}_1, \mathbf{x}_1) = k(\mathbf{x}_2, \mathbf{x}_2)$ and $b = k(\mathbf{x}_1, \mathbf{x}_2)$, then $H_k = \begin{pmatrix} a & b \\ b & a \end{pmatrix}$. Clearly, $\mathbf{y} = (1, -1)^T$ is an eigenvector of H_k with the eigenvalue $a - b > 0$, which is positive due to the positive definiteness of k. Consequently, $\mathbf{y}$ is also an eigenvector of H_k^{-1} with eigenvalue $1/(a - b) > 0$. Applying (16) we have

$$\begin{aligned} g_k(\mathbf{z}) &= (k(\mathbf{z}, \mathbf{x}_1), k(\mathbf{z}, \mathbf{x}_2))H^{-1}\mathbf{y} \\ &= \frac{1}{a - b}(k(\mathbf{z}, \mathbf{x}_1) - k(\mathbf{z}, \mathbf{x}_2)) \end{aligned}$$

Therefore, $g_k(\mathbf{z}) > 0$ if and only if $k(\mathbf{z}, \mathbf{x}_1) > k(\mathbf{z}, \mathbf{x}_2)$. □

D More Experiments

In this section, we discuss the details in training different models. We then address the question whether the realistic networks such as AlexNet and ResNets are shift invariant. Finally we discuss the relationship between the margin and margin, in order to probe the empirical observation we had on the shift invariance and adversarial robustness.

D.1 Experiment details

All the models on MNIST and FashionMNIST were trained for 20 epochs using the ADAM optimizer, with a batch size of 200 and learning rate of 0.01. The learning rate is decreased by a factor of 10 at the 10th and 15th epoch. To evaluate the robustness of these models, we use PGD l_2 and l_∞ attacks (Madry et al., 2018) with different ϵ values, a single random restart, 10 iterations and step-size of $\epsilon/5$.

On SVHN dataset, the models were trained for 100 epochs using SGD with 0.9 momentum and batch-size of 128. A learning rate of 0.1, with decay of factor 10 at the 50th and 75th epochs was used. We use the same PGD attacker as described above. The results for clean and robust accuracy with different ϵ values for l_2 and l_∞ are given in Figure 4.

On Cifar10, we train AlexNet and 5 variants of ResNet for 200 epochs using SGD with a cosine annealing schedule (Loshchilov & Hutter, 2016). We use a momentum equal to 0.9 for SGD and weight decay of $5e - 4$.

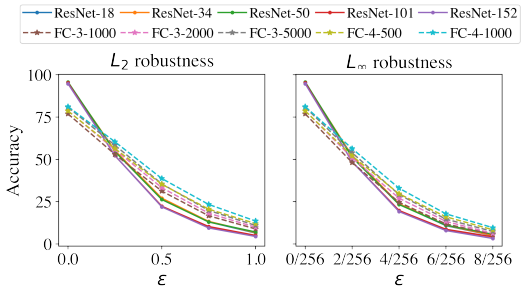


Figure 4: Robustness of ResNets vs FC networks on SVHN dataset.

D.2 Realistic architectures and shift invariance

Before describing experiments with real-world architectures, we discuss their shift invariance. With certain assumption on the padding, (Cohen & Welling, 2016) proves that the convolution operation is equivariant to shifts. The convolution operation together with global pooling leads to shift invariance. Also, it is easy to derive from (Cohen & Welling, 2016) results for models that use stride and local pooling, which we summarize in the note below.

Note 1. *A Convolutional Neural Network that meets the following assumptions has some shift invariance:*

1. *It consists of N fully convolutional or local pooling layers and 1 global pooling layer followed by any fully connected layers.*
2. *Circular padding is used in a "same" mode in the convolutional and pooling layers.*

Such a CNN with the strides $p_1, p_2, \cdots, p_N$ in the convolutional or pooling layers are invariant to a shift of P pixels, where P is any integer multiple of $\prod_{i=1}^{N} p_i$.

(Kayhan & Gemert, 2020) discusses padding modes. Architectures applied to real problems are generally not born shift invariant (Zhang, 2019; Kayhan & Gemert, 2020) due to the violations of these assumptions. The most common case is the use of zero padding. (Zhang, 2019) has shown that realistic architectures with zero paddings still preserve approximate invariance to shifts after training. Another common cause of a more severe lack of shift invariance is the use of a fully connected layer as a substitute for the global pooling layer. This is widely seen in the earlier network architectures such as LeNet (LeCun et al., 1989) and AlexNet (Krizhevsky, 2014) while more recent ones have universally adopted a global pooling layer (e.g. ResNet (He et al., 2016), DenseNet (Huang et al., 2017)) to reduce the number of parameters. Specifically, (Zhang, 2019) shows that in practice AlexNet is much less shift invariant than the other architectures. For this reason, we pay extra attention to the comparison of robustness between AlexNet and other architectures in the following sections when FC networks do not achieve decent accuracy.

Table 1: Average l_2 distance for adversarial examples over all samples for dataset 1 and 2. For orthogonal vectors, FC networks are more robust than CNNs. For orthogonal frequencies, CNNs are more robust than FC networks.

	Orth. Vectors		Orth. Frequencies	
n	FC	CNN	FC	CNN
50	0.113	0.033	0.129	0.272
100	0.083	0.029	0.126	0.251
200	0.060	0.024	0.103	0.326

D.3 Dimension and margin

Training a shift invariant network is similar to training a FC network using all shifted versions of the training set. This raises the question of why such data augmentation with shifted training examples affects adversarial robustness? Note that it has been argued that in a different context, more training data should lead to greater robustness (Schmidt et al., 2018). We examine this question with some inital experiments.

In prior work it has been suggested that larger input dimension of the data (Goodfellow et al., 2015; Gilmer et al., 2018) or larger intrinsic dimension of the data (Khoury & Hadfield-Menell, 2019) can reduce robustness. One way that shift invariance might reduce robustness is by increasing the intrinsic dimension of the training data.

To examine this question we create two synthetic datasets, one in which shift invariance increases the implicit dimension of the data, and one in which it does not. In both datasets, each class contains n, orthogonal vectors, with varying n and dimension $d = 2000$. In the first set, the vectors are randomly chosen from a uniform distribution. The vectors in each class have dimension n, but all shifted versions of them have dimension d for any n. In the second dataset, training examples are sampled by frequencies $\sin 2\pi kx$ and $\cos 2\pi kx$, where $k \in [1, n]$. The first class consists of frequencies in which k is odd, and the second class consists of frequencies with even k. In this case, the vectors in a class have dimension n, and all shifts of the vectors have dimension n.

If increasing the intrinsic dimension of the data reduces robustness, we would expect that for the FC network, increasing n would reduce robustness for both datasets. However, we would expect that for the shift invariant network, we see much lower robustness for dataset 1 than for dataset 2, especially for small n, which we do observe as shown in Table 1.

Table 2: Average l_2 distance for all samples on FC network for dataset 3. The average l_2 distance increases with and increase in p.

	p					
n	0	0.02	0.05	0.1	0.2	0.3
50	0.106	0.108	0.096	0.114	0.152	0.181
100	0.080	0.081	0.080	0.094	0.121	0.181
200	0.060	0.061	0.061	0.079	0.112	0.170

In our theoretical results, and the example in Figure 1, we see that considering all shifts of the data may not only increase its dimension but also reduce the linear margin. To tease these effects apart, we create a third data set in which as its size increases, its dimension increases but its margin does not.

In this third data set, all samples in one class have a common component in a random direction, and a second component in a random direction that is orthogonal to all other vectors. That is, let $\mathbf{c}_1, \mathbf{c}_2, \mathbf{r}_i^1, \mathbf{r}_i^2, 1 \leq i \leq n$ be a set of randomly chosen, orthogonal unit vectors. Then the i'th vector in class j is given by $\mathbf{x}_i^j = p\mathbf{c}_j + \mathbf{r}_i^j$, where p is a constant that we vary. The margin between the classes will be at least $\sqrt{2}p$. Since we are not controlling the margin for all shifts of this data, we classify this data only using an FC network. As shown in Table 2, we observe that as p increases, robustness increases. Also for large p, robustness is similar across different n values suggesting that the linear margin predicts robustness better than the dimension of the data.

The experiments in this subsection are meant to highlight some interesting questions about the connection between shift invariance and robustness. Does shift-invariance reduce robustness by increasing the implicit dimension of the data? And if so, is this because it leads to data that implicitly has a smaller margin? Our tentative answer to these questions is yes, based on experiments with simple datasets. But these are questions that surely deserve greater attention.