Sample-wise Adversarial Robustness Regularizes Neural Networks

Anonymous Authors[1]

Abstract

The problem of adversarial samples has shown that modern Neural Network (NN) models could be rather fragile. Among the most promising techniques to solve the problem, one is to require the model to be ϵ-*adversarially robust*; that is, to require the model not to change predicted labels when any given input samples are perturbed within a certain range. However, it is widely observed that such methods would lead to standard performance degradation. We study why the degradation occurs from the statistical perspective. More specifically, with a mix of theory and experiments, we show that adversarial robustness is a form of drastic *regularization* that hurts the capacity of NNs and results in performance degradation, when adversarial noises are prevented from being amplified too much through spectral normalization. We believe the result is a first step to understand and alleviate the degradation.

1. Introduction

Deep Neural Networks (NNs) have achieved remarkable performance among a range of tasks, e.g., image recognition (Krizhevsky et al., 2012). Yet, such models are rather fragile in some cases. It is widely observed that a perturbation added to a natural image that is imperceptible to humans can fool state-of-the-art NN models such that they predict wrongly. To overcome the shortcoming, methods are developed to make the models more robust. Among the methods, one class of the promising methods requires NNs to be *sample-wise adversarially robust*; that is, to require the model not to change predicted labels when any given input samples are perturbed within a certain range. The methods include adversarial training (Madry et al., 2018), Lipschitz-Margin Training (Tsuzuku et al., 2018) etc.

However, it is widely observed that such methods would lead to degradation of *standard* performance. We study why the degradation occurs from the *statistical perspective*. Previously, many works have analyzed *adversarial risk* from the statistic perspective (Attias et al., 2018; Schmidt et al., 2018; Cullina et al., 2018; Yin & Bartlett, 2018; Khim & Loh, 2018); that is, a classifier needs to generalize well both on normal samples and on adversarial samples. Few works have analyzed the influence of adversarial robustness on generalization behaviors of empirical risk of normal data. Tsipras et al. (2019) point out that the accuracy on normal samples provably conflicts with the accuracy on adversarial samples for linear models. In this work, we study the trade-

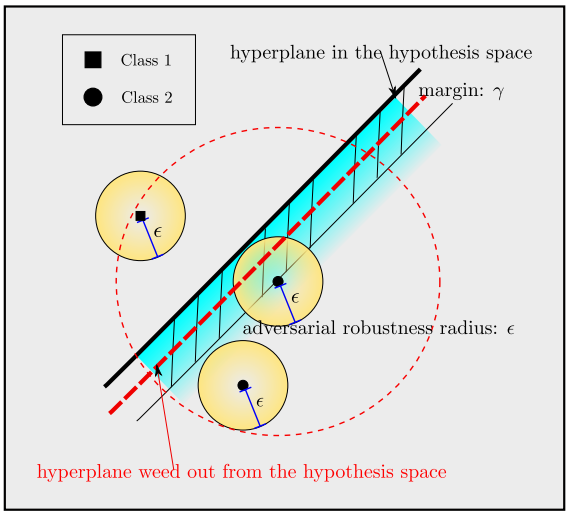

Figure 1. Illustration of the regularization effect of adversarial robustness. If a NN T is ϵ-adversarial robust, for a given sample $\boldsymbol{x}$ (drawn as filled squares or circles) and points $\boldsymbol{x}'$ in the yellow ball $\{\boldsymbol{x}' \mid \rho(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon\}$ around $\boldsymbol{x}$, the predict labels of $\boldsymbol{x}, \boldsymbol{x}'$ should be the same, and the loss variation is potentially bigger as $\boldsymbol{x}'$ moves from the center to the edge, as shown as intenser yellow color at the edge of a ball. Collectively, the adversarial robustness of each sample requires a margin exists for the decision hyperplane, shown as the shaded cyan margin. As normally known, margin is related to generalization ability that shrinks the size of the hypothesis space. In this case, the margin required by adversarial robustness would weed out hypotheses that do not have an adequate margin, such as the red dashed line shown in the illustration. If the adversarial perturbation is amplified too much by a NN, it could even severely hurt the capacity of the network, since it may require an unreasonable margin, shown as the red dashed circle.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

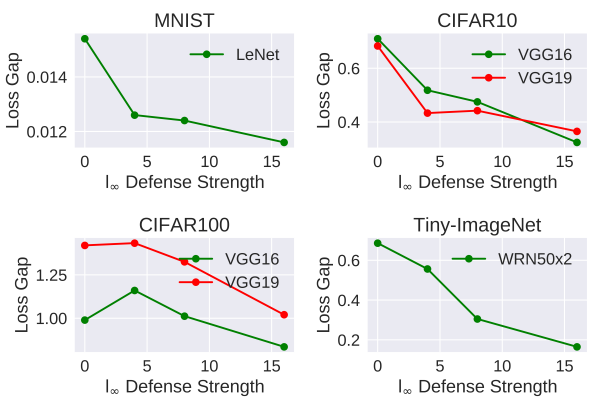

Figure 2. The plot of the *loss gap* between training loss and test loss v.s. the *adversarial robustness* of NNs on MNIST, CIFAR10/100, Tiny ImageNet. It shows that the loss gap (generalization error) decreases as adversarial robustness grows. The adversarial robustness strength is characterized by the maximally allowed l_∞ norm of adversarial samples that are used to train the NNs — we use adversarial training (Madry et al., 2018) to build in adversarial robustness in NNs. For CIFAR dataset, we train NNs with different depth, thus the sub-plots have two lines. For the details of the experiments, refer to section 5.

off of adversarial robustness in NNs on real datasets. More specifically, we show with a mix of theory and experiments that adversarial robustness is a form of drastic *regularization* that hurts the capacity of NNs and results in performance degradation, when adversarial noises are prevented from being amplified too much through spectral normalization. The regularization effect is demonstrated in fig. 1.

Adversarial robustness requires for a given NN T, each sample that is *natural*, e.g., a natural image, to be some distance from the decision boundary, shown as the balls in fig. 1. Conversely, it asks overall, a margin should exist around the decision boundary, also shown in the shaded margin around a hyperplane in fig. 1. As well known, margin relates with generalization ability in Support Vector Machine. Intuitively, when we require for each natural image to has an adversarially robust ball, we have weed out the hypotheses in the hypothesis space that is too "close" to training samples, such as the dashed red hyperplane in fig. 1. Thus, it is reasonable to suggest the margin required by adversarial robustness may also serve as a form of regularization.

However, a NN is a complex nonlinear system, the margin required in the data space has a complicated nonlinear relationship with the *classification margin*, which is defined as the score difference between the logit (the pre-activation of the last layer of a NN that is fed to a softmax activation function) of the class that the label dictates, and the largest logit of classes that is not of the label class, similar with the score margin used in Bartlett et al. (2017). As a worst case analysis, suppose that a perturabtion $\delta \boldsymbol{x}$ is amplified way large, as the dashed red ball shown in fig. 1. It would require the NN to have an unreasonable margin, and degrade the performance of the NN in a disruptive and unpredictable way. Consequently, the relationship is hard to characterize analytically, since we cannot esimate their relationship reliably.

To make the problem amenable to analysis, we apply spectral normalization (Miyato et al., 2018) to NNs to require the adversarial perturbation not to be amplified. The restriction enables us to analyze the regularization effects of adversarial robustness through *ramp loss*, as a linear approximation to cross entropy loss in section 4.1. With ramp loss, we prove in theorem 4.1 that the generalization error decreases as adversarial robustness increases. We corroborate the theoretical results with experiments on networks that use cross entropy loss, and show that in MNIST, CIFAR10/CIFAR100 and Tiny ImageNet (a subset of ImageNet the detail of which is described in the appendix), as we increase the adversarial robustness built into the NNs, the generalization gap decreases, as shown in fig. 2. We believe the result is a first step to understand and alleviate the degradation.

Contributions. We study the regimen where the perturbation induced by adversarial noise is prevented from being amplified by applying spectral normalization on NNs.

- Under ramp loss, we rigorously derive a generalization bound in theorem 4.1 that characterizes the regularization effect adversarial robustness has on NNs. It predicts that GE shrinks as adversarial robustness grows.
- We show that generalization error characterizes by ramp loss is a linear approximation of cross entropy loss in the regimen studied. We corroborate theoretical analysis with with experiments that confirm that the approximation rather reliably predicts the generalization behaviors of NNs, as shown in fig. 2.

2. Related Works

Robustness in machine learning models is a large field. We focus on the works that analyze robustness from the statistic perspective.

Tsipras et al. (2019) point out that the accuracy on normal samples provably conflicts with the accuracy on adversarial samples for linear models when assuming a certain Gaussian distribution, while we study the trade-off of adversarial robustness in NNs on real datasets.

The majority of works that study adversarial robustness from the statistic perspective study the generalization behaviors of machine learning models under *adversarial risk*. However, adversarial risk is not our focus. In this paper, we

study when a classifier needs to accommodate adversarial samples, what is the influence that the accommodation has on generalization behaviors of empirical risk of normal data. The works that study adversarial risk include Attias et al. (2018), Schmidt et al. (2018), Cullina et al. (2018), Yin & Bartlett (2018), Khim & Loh (2018), and Sinha et al. (2018). It may be tempting to think that the standard risk is just the case where the perturbation ϵ in the adversarial risk is zero. However, such intuition is not correct. The bounds obtained under the setting of adversarial risk characterize the risk gap introduced by adversarial examples, thus, it is intuitive that we get a larger risk gap for a larger allowed perturbation limit ϵ, which is roughly among the conclusions obtained in those bounds. That is to say, the conclusion normally leads to a larger generalization error as an algorithm is asked to handle more adversarial examples, for that it focuses on characterizing the error of adversarial samples, not that of normal samples.

3. Preliminaries

3.1. Statistical Learning

Assume a sample space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the space of input data, and $\mathcal{Y}$ is the label space. $Z := (X, Y)$ are the random variables with an unknown distribution μ, from which we draw samples. We use $S_m = \{z_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^m$ to denote the training set of size m whose samples are drawn independently and identically distributed (i.i.d.) by sampling Z. Given a loss function l, the goal of learning is to identify a function $T : \mathcal{X} \mapsto \mathcal{Y}$ in a hypothesis space (a class $\mathcal{T}$ of functions) that minimizes the expected risk

$$R(l \circ T) = \mathbb{E}_{Z\sim\mu}\left[l\left(T(X), Y\right)\right],$$

Since μ is unknown, the observable quantity serving as the proxy to the true risk $R(f)$ is the empirical risk

$$R_m(l \circ T) = \frac{1}{m}\sum_{i=1}^{m} l\left(T(\boldsymbol{x}_i), y_i\right).$$

Our goal is to study the discrepancy between R and R_m, which is termed as *generalization error* — it is sometimes termed as generalization gap in the literature

$$\mathrm{GE}(l \circ T) = |R(l \circ T) - R_m(l \circ T)|. \tag{1}$$

3.2. Adversarial Robustness

Though adversarial robustness has been implicitly discussed in many existing works, we formalize its definition in the following to clarify the concept.

Definition 1 ((ρ, ϵ)-adversarial Robustness)**.** *Given a multi-class classifier $f : \mathcal{X} \to \mathbb{R}^L$, and a metric ρ on $\mathcal{X}$, where L is the number of classes, f is said to be adversarially robust w.r.t. adversarial perturbation of strength ϵ, if there exists a $\epsilon > 0$ such that for $\delta\boldsymbol{x} \in \{\rho(\delta\boldsymbol{x}) \leq \epsilon\}$, and $\forall z = (x, y) \in \mathcal{Z}, i \neq y \in \mathcal{Y}$, we have*

$$f_y(\boldsymbol{x} + \delta\boldsymbol{x}) - f_i(\boldsymbol{x} + \delta\boldsymbol{x}) \geq 0.$$

ϵ is called **adversarial robustness radius***. When the metric used is clear, we also refer (ρ, ϵ)-adversarial robustness as ϵ-adversarial robustness.*

Note that the definition is a *sample-wise* one; that is, it requires for each sample to have a guiding area, in which all samples are of the same class.

3.3. Algorithmic Robustness Framework

In order to characterize the bound to the GE, we build on the *algorithmic robustness* framework (Xu & Mannor, 2012). We introduce the framework below.

Definition 2 ($(K, \epsilon(\cdot))$-robust)**.** *An algorithm is $(K, \epsilon(\cdot))$ robust, for $K \in \mathbb{N}$ and $\epsilon(\cdot) : \mathcal{Z}^m \mapsto \mathbb{R}$, if $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted by $\mathcal{C} = \{C_k\}_{k=1}^K$, such that the following holds for all $s_i = (\boldsymbol{x}_i, y_i) \in S_m, z = (\boldsymbol{x}, y) \in \mathcal{Z}, C_k \in \mathcal{C}$:*

$$\begin{aligned} &\forall s_i = (\boldsymbol{x}_i, y_i) \in C_k, \forall z = (\boldsymbol{x}, y) \in C_k \\ \implies &|l(f(\boldsymbol{x}_i), y_i) - l(f(\boldsymbol{x}), y)| \leq \epsilon(S_m). \end{aligned}$$

The gist of the definition is to constrain the variation of loss values on test examples w.r.t. those of training ones through local property of the algorithmically learned function f_{S_m}. Intuitively, if $s \in S_m$ and $z \in \mathcal{Z}$ are "close" (e.g., in the same partition C_k), their loss should also be close, due to the intrinsic constraint imposed by f.

For any algorithm that is robust, Xu & Mannor (2012) proves

Theorem 3.1 (Xu & Mannor (2012))**.** *If a learning algorithm is $(K, \epsilon(\cdot))$-robust and $\mathcal{L}$ is bounded, a.k.a. $\mathcal{L}(f(\boldsymbol{x}), y) \leq M\ \forall z \in \mathcal{Z}$, for any $\eta > 0$, with probability at least $1 - \eta$ we have*

$$GE(f_{S_m}) \leq \epsilon(S_m) + M\sqrt{\frac{2K\log(2) + 2\log(1/\eta)}{m}}. \tag{2}$$

To control the first term, an approach is to constrain the variation of the loss function. Covering number (Shalev-Shwartz & Ben-David, (2014), Chapter 27) provides a way to characterize the variation of the loss function, and conceptually realizes the actual number K of disjoint partitions.

Definition 3 (Covering number)**.** *Given a metric space $(\mathcal{S}, \rho)$, and a subset $\tilde{\mathcal{S}} \subset \mathcal{S}$, we say that a subset $\hat{\mathcal{S}}$ of $\tilde{\mathcal{S}}$ is a ϵ-cover of $\tilde{\mathcal{S}}$, if $\forall s \in \tilde{\mathcal{S}}, \exists \hat{s} \in \hat{\mathcal{S}}$ such that $\rho(s, \hat{s}) \leq \epsilon$. The ϵ-covering number of $\tilde{\mathcal{S}}$ is*

$$\mathcal{N}_\epsilon(\tilde{\mathcal{S}}, \rho) = \min\{|\hat{\mathcal{S}}| : \hat{\mathcal{S}} \text{ is a } \epsilon\text{-covering of } \tilde{\mathcal{S}}\}.$$

For any regular k-dimensional manifold embedded in space equipped with a metric ρ, e.g., the image data embedded in $L^2(\mathbb{R}^2)$, the square integrable function space defined on $\mathbb{R}^2$, it has a covering number $\mathcal{N}(\mathcal{X};\rho,\epsilon)$ of $(C_{\mathcal{X}}/\epsilon)^k$ (Verma, 2013), where $C_{\mathcal{X}}$ is a constant that captures its "intrinsic" properties, and ϵ is the radius of the covering ball. When we calculate the GE bound of NNs, we would assume the data space is a k-dimensional regular manifold that accepts a covering.

3.4. Neural Networks

We study the map T defined in section 3.1 as a NN (though technically, T now is a map from $\mathcal{X}$ to $\mathbb{R}^L$, instead of to $\mathcal{Y}$, such a abuse of notation should be clear in the context).

We present the definition of Multi-Layer Perceptron (MLP) here, which captures all ingredients for theoretical analysis and enables us to convey the analysis without unnecessary complications, though we note that the analysis extends to Convolutional Neural Networks (CNNs) almost equally.

A MLP is a map that takes an input x from the space $\mathcal{X}$, and builds its output by recursively applying a linear map W_i followed by a pointwise non-linearity g:

$$x_i = g(\boldsymbol{W}_i \boldsymbol{x}_{i-1}),$$

where i indexes the times of recursion, which is denoted as a layer in the community, $i = 1,\dots,L$, $x_0 = x$, and g denotes the activation function, which could be Rectifier Linear Unit (ReLU) (Glorot et al., 2011), Sigmoid, Tanh, or max pooling operator (Bécigneul, 2017). To compactly summarize the operation of T, we denote

$$Tx = g(\boldsymbol{W}_L g(\boldsymbol{W}_{L-1} \dots g(\boldsymbol{W}_1 \boldsymbol{x}))). \tag{3}$$

4. Generalization Bound Guaranteed by Adversarial Robustness

In this section, we rigorously establish the bound outlined in the introduction. As illustrated in fig. 1, adversarial robustness requires that for any given sample, the loss within the adversarial robustness radius should vary smoothly w.r.t. the loss of its reference training sample at the center. However, in a real-world NN, the relationship between margin and loss is nonlinear, thus hard to characterize analytically. In the following, we study a restricted case, where the loss variation w.r.t. to perturbation is almost linearized.

4.1. Problem Restriction

We use the mostly widely used cross entropy loss as a running example to motivate our restriction. Specifically, we establish that the loss variation in term of ramp loss of samples in a small covering ball, such as the one of which the radius is the adversarial robustness radius, is a linear approximation to the loss variation of cross entropy loss, when the network is restricted by spectral normalization not to amplify the adversarial perturbation.

4.1.1. Ramp Loss as A Linear Approximation to Cross Entropy Loss

Given a sample $z = (\boldsymbol{x}, y) \in \mathcal{Z}$, we have its cross entropy loss as

$$l(\boldsymbol{x}, y) = -\mathbf{1}_y \log \frac{e^{\boldsymbol{w}_y^T \boldsymbol{x}}}{\sum_{i=1}^{|\mathcal{Y}|} e^{\boldsymbol{w}_i^T \boldsymbol{x}}}, \tag{4}$$

where $\mathbf{1}_y$ is the indicator function. We ignore NN for now, and feed $\boldsymbol{x}$ directly to linear model. Let $v_i = \boldsymbol{w}_i^T \boldsymbol{x}, i = 1,\dots,|\mathcal{Y}|$. Though roughly the larger v_y is, the smaller the loss would be, the loss variation w.r.t. to the change of $\boldsymbol{x}$ has many dependent factors, and hard to characterize. However, it is reasonably to suggest that the change rate is linear locally, which is the classic insight of calculus. Let $u_i(\boldsymbol{x}) = (\boldsymbol{w}_y - \boldsymbol{w}_i)^T \boldsymbol{x}$. Note that u_i is the margin to the class boundary of class i and y. Locally, we have

$$\frac{\partial l}{\partial u_i} = \mathbf{1}_y (1 + \sum_{i \neq y}^{|\mathcal{Y}|} e^{-u_i(\boldsymbol{x})}) du_i \tag{5}$$

From eq. (5), we can see that for a small change $d\boldsymbol{x}$ around $\boldsymbol{x}$, the rate of change of l w.r.t. $u(d\boldsymbol{x})$, is given by $(1 + \sum_{i \neq y}^{\mathcal{Y}} e^{-u_i(\boldsymbol{x})})$. The larger the margin $u_i(\boldsymbol{x})$ is, the smaller the rate the loss changes, when assuming the sample is properly classified, so that all margins are positive. It suggests that locally, we can use a linear model to approximate the relationship between classification margin and loss variation.

To further introduce the restriction of the problem, we introduce the *ramp loss*, *margin operator*, and *ramp risk*.

Definition 4 (Margin Operator)**.** *A margin operator* $\mathcal{M} : \mathbb{R}^L \times \{1,\dots,L\} \to \mathbb{R}$ *is defined as*

$$\mathcal{M}(\boldsymbol{v}, y) := v_y - \max_{i \neq y} v_i$$

We have informally referred to margin previously. Here we formally state that the margin in this paper refers to the output of the margin operator, where v_i could be the logits of NNs, or the logit of a linear model.

Definition 5 (Ramp Loss)**.** *The ramp loss* $l_\gamma : \mathbb{R} \to \mathbb{R}^+$ *is defined as*

$$l_\gamma(r) := \begin{cases} 0 & r < -\gamma \\ 1 + r/\gamma & r \in [-\gamma, 0] \\ 1 & r > 0 \end{cases}$$

Definition 6 (Ramp Risk). *Given a classifier f, ramp risk is the risk defined as*

$$R_\gamma(f) := \mathbb{E}(l_\gamma(-\mathcal{M}(f(X), Y))),$$

where X, Y are random variables in the instance space $\mathcal{Z}$ previously.

Write the ramp loss in full, we have

$$l_\gamma(\boldsymbol{x}, y) = \max\{0, 1 - \frac{1}{\gamma}(\boldsymbol{w}_y - \boldsymbol{w}_{\hat{y}})^T \boldsymbol{x}\},$$

where $\hat{y} = \arg\max_{i \neq y} v_i$.

Now we show how ramp loss approximates eq. (4). Given a ϵ-covering of instance space $\mathcal{Z}$, and a sample $\boldsymbol{x}$ is classified correctly, let $C := \{\boldsymbol{x'} \mid \rho(\boldsymbol{x}, \boldsymbol{x'}) \leq \epsilon\}$, and

$$\boldsymbol{x}_a = \arg\min_{\boldsymbol{x'} \in C} \sum_{i=1}^{|\mathcal{Y}|} e^{-u_i(\boldsymbol{x'})}, \boldsymbol{x}_b = \arg\max_{\boldsymbol{x'} \in C} \sum_{i=1}^{|\mathcal{Y}|} e^{-u_i(\boldsymbol{x'})}.$$

Note that since all margins are positive, the smaller each margin gets, the larger the value is. It leads to the extreme values of loss as following.

$$l_{\min} = \mathbf{1}_y \log(\sum_{i=1}^{|\mathcal{Y}|} e^{-u_i(\boldsymbol{x}_a)}), l_{\max} = \mathbf{1}_y \log(\sum_{i=1}^{|\mathcal{Y}|} e^{-u_i(\boldsymbol{x}_b)}).$$

By mean value theorem, we have

$$l_{\max} - l_{\min} = \frac{dl}{d\boldsymbol{u}}(\boldsymbol{x}_c)^T \left(\boldsymbol{u}(\boldsymbol{x}_b) - \boldsymbol{u}(\boldsymbol{x}_a)\right),$$

where $(dl/d\boldsymbol{u})_i$ is calculated earlier in eq. (5). Notice that if we let

$$\frac{1}{\gamma} = (1 + \sum_{i \neq y}^{\mathcal{Y}} e^{-u_i(\boldsymbol{x}_c)}),$$

and only consider the contribution to loss by the margin that is the largest among non-label classes, we have a linear approximation of the loss as

$$l(\boldsymbol{x}, y) = \max\{l_{\min}, l_{\max} - \frac{1}{\gamma} u_{\hat{y}}(\boldsymbol{x})\}.$$

We have established the connection between ramp loss and cross entropy loss in the setting where a sample is classified correctly. The case a sample is classified wrongly is similar, and we do not repeat here.

Thus, the ramp loss can be viewed as a loss calibrated from a nonlinear loss such as cross entropy that linearizes the classification errors of eq. (4) in its critical regions, a.k.a. the region in the margin of the decision boundary. *Thus, we can see that the margin γ in the ramp risk can be viewed as a dynamic scaling factor that translates the perturbation of samples in the region within a certain margin of the decision boundary to loss variation linearly.* It enables an analytic analysis of the loss variation in theorem 4.1. Similar simplification was used before in Bartlett et al. (2017), in which they show that NNs trained with random labels have smaller normalized margins than the ones trained with real labels.

4.1.2. Constraining Perturbation Amplification via Spectral Normalization

However, to apply the simplification to study adversarial samples in practical NNs, we need to ensure that the approximation is valid. In the following, we isolate such a case.

Notice that adversarial robustness essentially characterizes a "margin" in the input data space; that is, there exists a margin for any decision boundaries, so that for any perturbation less than ϵ applied to a training sample, it does not change the predicted label. In addition, an adversarial sample is a sample that is imperceptibly different from the reference training sample that it is perturbed from. So the perturbation is a small change that already satisfies our approximation.

What reminds is to ensure after the sample is being propagated through a NN T, the perturbation still keeps as a small change. To achieve it, we restrict the spectral norm of weight $\boldsymbol{W}_i$ of a NN at each layer i, as explained in the following.

Let $T' := g(\boldsymbol{W}_{L-1} \ldots g(\boldsymbol{W}_1 \boldsymbol{x}))$, be the NN that excludes the last layer L. By theorem 3 in Sokolic et al. (2017), we have

$$T'\boldsymbol{x} - T'\boldsymbol{x}' = \boldsymbol{J}_{\boldsymbol{x},\boldsymbol{x}'}(\boldsymbol{x}' - \boldsymbol{x})$$

where $\boldsymbol{J}_{\boldsymbol{x},\boldsymbol{x}'}$ is defined as the average Jacobian on the line segment between $\boldsymbol{x}$ and $\boldsymbol{x}'$.

Let $\boldsymbol{x}$ be a sample, and $\delta\boldsymbol{x}$ be the adversarial perturbation applied on $\boldsymbol{x}$. We can obtain the upper bound of the norm change by

$$||T'(\boldsymbol{x} + \delta\boldsymbol{x}) - T'\boldsymbol{x}||_2 \leq \prod_{i=1}^{L} ||\boldsymbol{W}_i||_2 ||\delta\boldsymbol{x}||_2,$$

where $||\boldsymbol{W}_i||_2$ is the spectral norm of $\boldsymbol{W}_i$. For space limit, we omit the upper bound calculation. It can be found in Sokolic et al. (2017), which is straight forward. As the upper bound suggests, by restricting the spectral norm of $\boldsymbol{W}_i$ to be 1, we can keep $T'\delta\boldsymbol{x}$ still as a small change.

By connecting the two margins — the adversarial robustness radius, and the classification decision boundary margin — we can characterize the regularization effects of adversarial robustness in the restricted setting.

4.2. Generalization Bound

With the previous problem restriction, we can cleanly characterize the influence of adversarial robustness on generalization.

We recall some notations that are used in slightly different contexts previously. Recall that T' denotes the NN without the last layer; let $\{\boldsymbol{w}_i\}_{i=1,\dots,|\mathcal{Y}|}$ be the set of vectors made up with ith rows of $\boldsymbol{W}_L$ (the last layer's weight matrix); given a sample $z = (\boldsymbol{x}, y) \in \mathcal{Z}$, let $u(\boldsymbol{x}) := (\boldsymbol{w}_y - \boldsymbol{w}_{\hat{y}})^T T'\boldsymbol{x}$, where $\hat{y} = \arg\max_{i \neq y} \boldsymbol{w}_i^T T'\boldsymbol{x}$; given a ϵ-covering of $\mathcal{Z}$, let

$$u_{\min} = \min_{\boldsymbol{x} \in \{\boldsymbol{x} \mid \rho(\boldsymbol{x}, \boldsymbol{x}_i) \leq \epsilon, \forall \boldsymbol{x}_i \in S_m\}} u(\boldsymbol{x}) \tag{6}$$

where ρ is the metric on $\mathcal{X}$. *$u_{\min}$ is the smallest positive margin of samples in the covering balls that contain training samples*.

We state the generalization error bound guaranteed by adversarial robustness in the following theorem.

Theorem 4.1. *Let T denote a NN defined at eq. (3), l_γ the ramp loss defined at definition 5, and $\mathcal{Z}$ the instance space assumed at section 3.1. Assume that $\mathcal{Z}$ is a k-dimensional regular manifold that accepts an ϵ-covering with covering number $(\frac{C_{\mathcal{X}}}{\epsilon})^k$. If T is ϵ_0-adversarially robust (defined at definition 1), $\epsilon \leq \epsilon_0$, and denote $u_{\min}$ the smallest positive margin in the covering balls that contain training samples (formally defined at eq. (6)), then given a set of i.i.d. training samples $S_m = \{z_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^m$ drawn from $\mathcal{Z}$, its generalization error GE$(l \circ T)$ (defined at eq. (1)) satisfies that, for any $\eta > 0$, with probability at least $1 - \eta$*

$$GE(l_\gamma \circ T) \leq \max\{0, 2(1 - \frac{u_{\min}}{\gamma})\} \tag{7}$$

$$+ \sqrt{\frac{2\log(2) C_{\mathcal{X}}^k}{\epsilon^k m} + \frac{2\log(1/\eta)}{m}}. \tag{8}$$

We analyze the influence of adversarial robustness on the generalization error. From the bound, we can see that GE is a function of two variables $u_{\min}, \epsilon$, besides the number of training samples m. We analyze the influence of $\epsilon, u_{\min}$ respectively. ϵ_0 is the upper bound of the norm of adversarial perturbation that a NN T is robust to. Adversarial robustness of a NN T grows with ϵ_0. As the theorem shows, eq. (8) decreases as ϵ increases, thus a more adversarially robust NN allows a larger ϵ as $\epsilon \leq \epsilon_0$, and would lead to a smaller value of eq. (8). $u_{\min}$ is the smallest positive margin of samples in covering balls that contain training samples. The margin from samples to decision boundary grows with adversarial robustness, so is the smallest margin among the margins $u_{\min}$ of all samples in the vicinity of training samples. It implies as ϵ_0 increases, $u_{\min}$ increases, and eq. (7) decreases. Overall, combining the influence of $\epsilon, u_{\min}$, **we conclude that as a more adversarially robust NN should have a smaller generalization gap.**

We present the proof of theorem 4.1 in the following.

Proof. Similar with the proof of theorem 3.1, we partition space $\mathcal{Z}$ into the ϵ-cover of $\mathcal{Z}$, which by assumption is a k-dimension manifold. Its covering number is upper bounded by $C_{\mathcal{X}}^k/\epsilon^k$, denoting $K = C_{\mathcal{X}}^k/\epsilon^k$, and C_i the ith covering ball. We study the constraint/regularization that adversarial robustness imposes on the variation of the loss function. Since we only have ϵ-adversarial robustness, the radius of the covering balls is at most ϵ — this is why we use the same symbol. Beyond ϵ, adversarial robustness does not give information on the possible variation anymore.

First, we analyze the risk change in a covering ball C_i. The analysis is divided into two cases: 1) all training samples in C_i are classified correctly; 2) all training samples in C_i are classified wrong. Note that no other cases exist, for that the radius of C_i is restricted to be ϵ. It guarantees that all samples in the ball is classified as the same class. Thus, either all training samples are all classified correctly, or wrongly.

We first study case 1). Given any sample $z = (\boldsymbol{x}, y) \in C_i$, let $\hat{y} = \arg\max_{i \neq y} \boldsymbol{w}_i^T T'\boldsymbol{x}$. Its ramp loss is

$$l_\gamma(\boldsymbol{x}, y) = \max\{0, 1 - \frac{1}{\gamma}(\boldsymbol{w}_y - \boldsymbol{w}_{\hat{y}})^T T'\boldsymbol{x}\}.$$

Note that within C_i, $(\boldsymbol{w}_y - \boldsymbol{w}_{\hat{y}})^T T'\boldsymbol{x} \geq 0$, thus $l_\gamma(\boldsymbol{x}, y)$ is mostly 1, and we would not reach the region where $r > 0$ in definition 5. Let $u(\boldsymbol{x}) := (\boldsymbol{w}_y - \boldsymbol{w}_{\hat{y}})^T T'\boldsymbol{x}$, and $u_{\min}^i = \min_{\forall \boldsymbol{x} \in C_i} u(\boldsymbol{x})$. We have

$$l_\gamma(\boldsymbol{x}, y) \leq \max\{0, 1 - \frac{u_{\min}^i}{\gamma}\} \leq \max\{0, 1 - \frac{u_{\min}}{\gamma}\},$$

We drop the i index on partition since $u_{\min}$ is the smallest margin among all partitions. However, we note that the right most inequality is to provide a cleaner surrogate to represent the loss variation later. Recall that for each ball, γ is a dynamical scaling factor discussed in section 4.1. Thus, for different partitions, likely γ is not the same for cross entropy loss, and the comparison of smallest margins between partitions may not be meaningful — though the inequality holds exactly in ramp loss.

The inequality above shows adversarial robustness requires that $T'\boldsymbol{x}$ should vary slowly enough, so that in the worst case, the loss variation within the adversarial radius should satisfy the above inequality. The observation leads to the constraint on the loss difference $\epsilon(\cdot)$ defined earlier in definition 2 in the following.

Given any training sample $z := (\boldsymbol{x}, y)$ and $z' := (\boldsymbol{x}', y') \in$

C_i, where C_i is the covering ball that covers $\boldsymbol{x}$, we have

$$\begin{aligned}&|l_\gamma(\boldsymbol{x},y)-l_\gamma(\boldsymbol{x}',y')|\\=&|\max\{0,1-\frac{u(\boldsymbol{x})}{\gamma}\}-\max\{0,1-\frac{u(\boldsymbol{x}')}{\gamma}\}|\\\leq&\max\{0,2(1-\frac{u_{\min}}{\gamma})\}.\end{aligned}\tag{9}$$

Then, we study case 2), in which all training samples $z\in C_i$ are classified wrong. In this case, for all $z\in C_i$, the $\hat{y}$ given by $\hat{y}=\arg\max_{i\neq y}\boldsymbol{w}_i^TT'\boldsymbol{x}$ in the margin operator is the same, for that $\hat{y}$ is the wrongly classified class. Its ramp loss is

$$l_\gamma(\boldsymbol{x},y)=\max\{0,1-\frac{1}{\gamma}(\boldsymbol{w}_y-\boldsymbol{w}_{\hat{y}})^TT'\boldsymbol{x}\}.$$

Note that in the case 1), it is the y that stays fixed, while $\hat{y}$ may differ from sample to sample; while in the case 2), it is the $\hat{y}$ stays fixed, while y may differ.

Similarly, within C_i as required by adversarial robustness, $(\boldsymbol{w}_y-\boldsymbol{w}_{\hat{y}})^TT'\boldsymbol{x}\leq 0$, thus we always have $1-\frac{1}{\gamma}(\boldsymbol{w}_y-\boldsymbol{w}_{\hat{y}})^TT'\boldsymbol{x}\geq 1$, implying

$$l_\gamma(\boldsymbol{x},y)=1.$$

Thus, $\forall z=(\boldsymbol{x},y),z'=(\boldsymbol{x}',y')\in C_i$

$$|l_\gamma(\boldsymbol{x},y)-l_\gamma(\boldsymbol{x}',y')|=0.\tag{10}$$

Since only these two cases are possible, by eq. (9) and eq. (10), we have $\forall z,z'\in C_i$

$$|l_\gamma(\boldsymbol{x},y)-l_\gamma(\boldsymbol{x}',y')|\leq\max\{0,2(1-\frac{u_{\min}}{\gamma})\}.\tag{11}$$

The rest follows the standard proof in algorithmic robust framework.

Let N_i be the set of index of points of samples that fall into C_i. Note that $(|N_i|)_{i=1\dots K}$ is an IDD multimonial random variable with parameters m and $(|\mu(C_i)|)_{i=1\dots K}$. Then

$$\begin{aligned}&|R(l\circ T)-R_m(l\circ T)|\\=&|\sum_{i=1}^K\mathbb{E}_{Z\sim\mu}[l(TX,Y)]\mu(C_i)-\frac{1}{m}\sum_{i=1}^m l(T\boldsymbol{x}_i,y_i)|\\\leq&|\sum_{i=1}^K\mathbb{E}_{Z\sim\mu}[l(TX,Y)]\frac{|N_i|}{m}-\frac{1}{m}\sum_{i=1}^m l(T\boldsymbol{x}_i,y_i)|\\&+|\sum_{i=1}^K\mathbb{E}_{Z\sim\mu}[l(TX,Y)]\mu(C_i)-\sum_{i=1}^K\mathbb{E}_{Z\sim\mu}[l(TX,Y)]\frac{|N_i|}{m}|\\\leq&|\frac{1}{m}\sum_{i=1}^K\sum_{j\in N_i}\max_{z\in C_i}|l(T\boldsymbol{x},y)-l(T\boldsymbol{x}_j,y_j)|\end{aligned}\tag{12}$$

$$+|\max_{z\in\mathcal{Z}}|l(T\boldsymbol{x},y)|\sum_{i=1}^K|\frac{|N_i|}{m}-\mu(C_i)||.\tag{13}$$

Remember that $z=(\boldsymbol{x},y)$.

By eq. (11) we have eq. (12) is equal or less than $\max\{0,2(1-\frac{u_{\min}}{\gamma})\}$. By Breteganolle-Huber-Carol inequality, eq. (13) is less or equal to $\sqrt{\frac{\log(2)2^{k+1}C_{\mathcal{X}}^k}{\gamma^k m}+\frac{2\log(1/\eta)}{m}}$.

The proof is finished. □

4.3. Practicality of the Restriction

We have established that ramp loss is an approximation to the problem where the influence of adversarial robustness is restricted in a linear regimen, and prove that adversarial robustness will lead to a decrease of generalization gap for NNs with ramp loss. However, the phenomenon we are interested in is how adversarial robustness influences NNs used in practice. Thus, in this section, we show the theorem proved in the restricted setting rather reliably predicts the behaviors of NNs trained with cross entropy loss, when the weights of NNs are spectrally normalized as discussed in section 4.1.

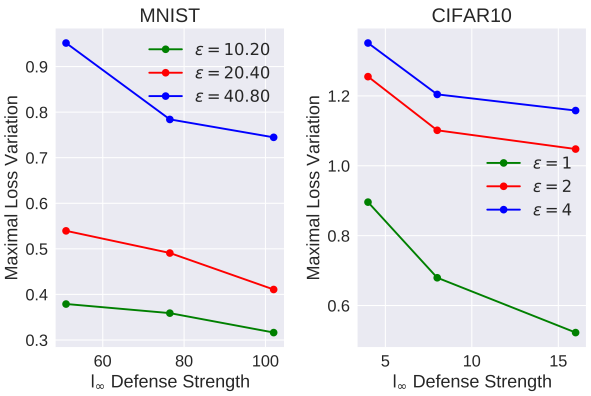

Figure 3. The plot of maximal loss variation within a covering ball v.s. NNs with different adversarial robustness. The maximal loss variation is calculated by generating adversarial samples from a reference training sample that is of at most ϵ perturbation (measured in l_∞ norm) from the reference training sample, but are still classified correctly (we will describe how adversarial robustness is built in NNs in section 5, when we systematically validate the prediction given by bound theorem 4.1). We use all correctly classified training samples to generate the adversarial samples to get the maximal value.

The first term eq. (7) in theorem 4.1 predicts that the loss variation within a covering ball decreases as adversarial robustness grows — the second term is a standard term in algorithmic robust framework, and is not an approximation, thus we do not verify it effectiveness. We verify that given a covering ball with fixed radius, how the loss variations in

the ball change w.r.t. adversarial robustness. Using network trained on MNIST and CIFAR10, we numerically calculate the loss variation between samples within the covering ball by computing the maximal loss variation induced by adversarial samples generated from training samples. We graph the results as following in fig. 3. It can be shown from fig. 3 that the max loss variation decreases as adversarial robustness grows as predicted.

5. Experiment Validation

We compute the generalization error in NNs that are used in practice to corroborate the GE bound proved in section 4 in this section. The details of experiments setting in this section are put to appendix A.

As discussed in the introduction, the adversarial training method (Madry et al., 2018) is a technique to increase the adversarial robustness of NNs, argubly the most well received one. Thus, we choose it to test the regularization effect of adversarial robustness on NNs. We use the spectral normalization technique in Miyato et al. (2018) to constrain the spectral norms of weights of NNs.

We train NNs with increasingly stronger adversarial robustness, and explore their loss gap. The adversarial strength of adversarial samples are measured in the l_∞ norm of the perturbation applied on samples. In this paper, we rescale all perturbation to the range of $0 - 255$; that means in fig. 2 fig. 3 fig. 4, the adversarial robustness is measured in this range, so is adversarial attack strength. A NN with adversarial robustness 8 is a NN trained with adversarial samples generated by perturbations that are at most 8 in l_∞ norm. The adversarial samples generated to attack a NN are generated in the same way.

5.1. Experiments on MNIST and CIFAR10/100

Given that we need to rule out the influence of batch normalization (BN) (Ioffe, Sergey and Szegedy, 2015), we test on best networks that can function without BN, i.e., VGG type network (Simonyan & Zisserman, 2015). We compute the gap of training loss and test loss of NNs trained with increasingly stronger adversarial robustness on datasets MNIST, CIFAR10/100. We use LeNet (Madry et al., 2018) for MNIST, VGGNet (Simonyan & Zisserman, 2015) for CIFAR10/100. We experiment VGGNet with two types of depth — one with 16 layers and one with 19 layers. The result is shown in **fig. 2 in the introduction**, and is consistent with our prediction.

We note that though the generalization error decreases as adversarial robustness increases, it does not imply we have a better performance on test accuracy on standard datasets. This is because the training loss decreases as well, thus, test loss is worse compared with vanilla network without adversarial training.

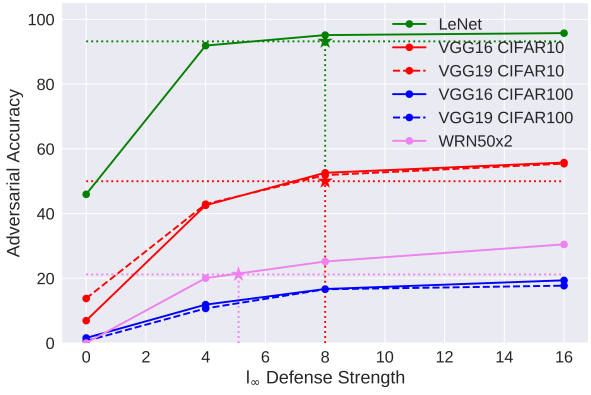

Figure 4. The plot of accuracy on adversarial samples v.s. adversarial defense strength built in NNs. The dotted line of which the intersections are marked by stars are adversarial accuracy in (Madry et al., 2018) (MNIST and CIFAR10), in (Li et al., 2018) (Tiny ImageNet) under similar adversarial attack strength.

We demonstrate that the adversarial robustness training reproduced is relevant by showing that its defense effect is comparable with existing works in fig. 4. We can see from it that similar adversarial robustness to (Madry et al., 2018; Li et al., 2018) is achieved for MNIST, CIFAR10, Tiny ImageNet in the NNs we reproduce.

5.2. Experiments on Tiny ImageNet

We would also like to explore the effect adversarial robustness on a larger dataset. We note that most of methods to build adversarial robustness in NNs do not experiment on ImageNet due to its scale. Since training ImageNet is very computational expensive with adversarial training, we run experiments on a subset of ImageNet, named Tiny ImageNet (ImageNet, 2018). We use Wide ResNet (Zagoruyko & Komodakis, 2016) with BN, in that it is rather hard to train ImageNet with VGG type network without BN. We note that in this case, spectral norms of weights of NNs are not strictly 1 due to the influence of BN. We observe similar phenomenon that loss gaps decreases as adversarial robustness grows, shown in fig. 2. The relevance of the adversarial defense effect is shown in fig. 4 as well.

References

Attias, I., Kontorovich, A., and Mansour, Y. Improved generalization bounds for robust learning. Technical report, 2018. URL https://arxiv.org/pdf/1810.02180.pdf.

Bartlett, P., Foster, D. J., and Telgarsky, M. Spectrally-

normalized margin bounds for neural networks. In *NIPS*, pp. 1–24, 2017.

Bécigneul, G. On the effect of pooling on the geometry of representations. Technical report, mar 2017. URL http://arxiv.org/abs/1703.06726.

Cullina, D., Bhagoji, A. N., and Mittal, P. PAC-learning in the presence of evasion adversaries. In *NIPS*, 2018.

Deng, J. D. J., Dong, W. D. W., Socher, R., Li, L.-J. L. L.-J., Li, K. L. K., and Fei-Fei, L. F.-F. L. ImageNet: A large-scale hierarchical image database. *CVPR*, 2009.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Glorot, X., Bordes, A., and Bengio, Y. Deep Sparse Rectifier Neural Networks. In *AISTATS*, 2011.

ImageNet, T. Tiny imagenet, 2018. URL https://tiny-imagenet.herokuapp.com/.

Ioffe, Sergey and Szegedy, C. Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*, 2015.

Khim, J. and Loh, P.-L. Adversarial Risk Bounds for Binary Classification via Function Transformation. Technical report, 2018. URL http://arxiv.org/abs/1810.09519.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

Lee, C., Xie, S., Gallagher, P. W., Zhang, Z., and Tu, Z. Deeply-supervised nets. In *AISTATS*, 2015.

Li, Y., Min, M. R., Yu, W., Hsieh, C.-J., Lee, T., and Kruus, E. Optimal transport classifier: Defending against adversarial attacks by regularized deep embedding. *arXiv preprint arXiv:1811.07950*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral Normalization for Generative Adversarial Networks. In *ICLR*, 2018.

Neyshabur, B., Bhojanapalli, S., and Srebro, N. A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks. In *ICLR*, 2018.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *ICLR*, 2014.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and adry, A. M. Adversarially Robust Generalization Requires More Data. In *NIPS*, 2018.

Shalev-Shwartz, S. and Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms*. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014. ISBN 9781107057135.

Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

Sinha, A., Namkoong, H., and Duchi, J. Certifying Some Distributional Robustness with Principled Adversarial Training. In *ICLR*, 2018.

Sokolic, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. D. Robust Large Margin Deep Neural Networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, aug 2017. ISSN 1053-587X. doi: 10.1109/TSP.2017.2708039.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Mądry, A. Robustness May Be at Odds with Accuracy. In *ICLR*, 2019.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Lipschitz-Margin Training : Scalable Certification of Perturbation Invariance for Deep Neural Networks. In *NIPS*, 2018.

Verma, N. Distance Preserving Embeddings for General n-Dimensional Manifolds. *Journal of Machine Learning Research*, 14:2415–2448, 2013.

Xu, H. and Mannor, S. Robustness and generalization. *Machine Learning*, 86(3):391–423, 2012. ISSN 08856125. doi: 10.1007/s10994-011-5268-1.

Yin, D. and Bartlett, P. Rademacher Complexity for Adversarially Robust. Technical report, 2018.

Zagoruyko, S. and Komodakis, N. Wide Residual Networks. In *BMVC*, 2016.

A. Appendices

We summarize the details of the experiments in this section.

A.1. Datasets

MNIST. The MNIST dataset consists of $70,000$ 28×28 images of handwritten digits, where $60,000$ are training data and $10,000$ are test data. We use raw images without pre-processing.

CIFAR10/100. Each CIFAR dataset consists of $50,000$ training data and $10,000$ test data. CIFAR-10 has 10 classes and CIFAR-100 has 100 classes respectively. We do data augmentation as in (Lee et al., 2015): during training, we zero-pad 4 pixels along each image side, and sample a 32×32 region cropping from the padded image or its horizontal flip; during testing, we use the original non-padded image.

Tiny ImageNet. Tiny ImageNet (Deng et al., 2009; ImageNet, 2018) is a subset of ImageNet dataset. It has 200 classes. Each class has 500 training images and 50 validation images. The images on the Tiny ImageNet dataset are 64×64 pixels, as opposed to the 256×256 pixel images on the full ImageNet set. The data augmentation is straightforward. An input image is 56×56 randomly cropped from a resized image using the scale and aspect ratio augmentation as well as scale jittering. We adopt a single 56×56 cropped image for testing.

A.2. Models and Implementation Details

MNIST. For MNIST dataset, we use the LeNet model as mentioned in (Madry et al., 2018).

CIFAR. For the CIFAR dataset, we use two different models, i.e., VGG-16 and VGG-19 as mentioned in (**?**). We replace the fully connected layer with 4096 channels in the paper with one with 512 channels. We adopt VGG-Net without additional batch normalization layers. Dropout layers are omitted and all the weights of VGG-net are initialized orthogonally (Saxe et al., 2014).

Tiny ImageNet. For Tiny ImageNet dataset, we used 50-layered wide residual networks with 4 groups of residual layers and $[3, 4, 6, 3]$ bottleneck residual units for each group respectively. The 3×3 filters of the bottleneck residual units have $[64 \times k, 128 \times k, 256 \times k, 512 \times k]$ feature maps with the widen factor $k = 2$. We replace the first convolutional layer that originally consists of 7×7 filters with stride 2 and padding 3 with 3×3 filters with stride 1 and padding 1. The max pooling layer after the first convolutional layer is removed. We retain batch normalization layers in this task. The weights of convolution layers for Tiny ImageNet are initialized with Xavier uniform (Glorot & Bengio, 2010). Again, all dropout layers are omitted.

Training. Experiments on MNIST dataset uses a mini-batch size of 256 and is trained for 160 epochs. The learning rate starts at 0.1 while being multiplied by 0.01 at epochs 80 and 120 respectively. Experiments on CIFAR dataset uses a mini-batch size of 512 and is trained for 160 epochs. The learning rate starts at 0.05, decay every 30 epoches by 2. The experiments on Tiny ImageNet dataset uses a mini-batch size of 256 and is trained 90 epochs. The initial learning rate is set to be 0.1 and decays by 10 at 30 and 60 epochs respectively. All experiments are trained on training set with stochastic gradient descent uses the momentum value of 0.9 without using additional regularizer such as weight decay. We constrain the spectral norm of the convolutional layers and linear layers of neural network following (Miyato et al., 2018).

A.3. Adversarial Robustness Defense Method

Specifically, we use l_∞-PGD (Madry et al., 2018) untargeted attack adversary for our adversarial training, where the adversary is created by performing projected gradient descent starting from a random perturbation around the natural example. Our adversarial defense method is depends on Madry et al. (2018), but we employ more varied adversarial defense defense than they have employed, so to analyze the different regularization effects different adversarial defense strength has on NNs. They only consider the strength of $76.5/255$ for MNIST, and $8/255$ for CIFAR-10. We extend larger and smaller strength to study the behavior of deep neural networks. For MNIST dataset, we use total adversarial perturbation of $\epsilon = [51.0/255.0, 76.5/255, 102.0/255]$, perturbation per step of $2.55/255$, and 40 total attack steps with 1 random restart. For all the other datasets, we use 10 steps of size $2/255$ and maximum of $\epsilon = [4/255, 8/255, 16/255]$ respectively for different defensive strengths. We replace all the natural training images with adversarial examples every iteration during training.

A.4. Adversarial Attack Methods

We evaluate the defense methods against l_∞-PGD (Madry et al., 2018) untargeted attack adversary, which is one of the strongest white-box attack methods. When considering adversarial attack, they usually train and evaluate against the same perturbation. And for our tasks, we only use the moderate adversaries that generated by 10 iterations with steps of size $||\gamma||_\infty = 2/255$ or $||\gamma||_\infty = 2.55/255$ and maximum of $||\gamma||_\infty = 8/255$ or $||\gamma||_\infty = 76.5/255$. When evaluating adversarial robustness, we only consider clean examples classified correctly originally, and calculate the accuracy of the adversarial samples generated from them that are still correctly classified.

When evaluating adversarial robustness, we consider all the examples; that is, for samples that are classified correctly

originally, we generated adversarial samples from it, and check whether the generated samples are classified correctly, and for samples that are classified wrongly originally, we directly classify them wrongly — yet we count them in the overall samples when we calculating the adversarial accuracy.

A.5. Results

The numerical results on MNIST, CIFAR10/100 are summarized in table 1.

Table 1. Top-1 accuracy rate comparison on the networks without batch normalization, including LeNet for MNIST dataset, VGG-Net for CIFAR datasets. All the NNs are tranined with spectral normalization. The three tables are for MNIST, CIFAR10, and CIFAR100 respectively. Different defensive strengths are adopted. Defensive strength denotes the strength of adversarial samples we adopted during training. We evaluate on clean images and adversarial images. The adversarial samples to attack are generated with the max perturbation norm of 76.5 for MNIST and 8 for CIFAR dataset. Adv Trn and SP denotes adversarial trainig and spectral normalization respectively. GE denotes the loss gap between training loss and test loss. ACC denotes the accuracy on standard samples, while PGD denotes the accuracy on adversarial samples.

MNIST		Defensive Strength			
		0	51.0	76.5	102.0
LeNet + SP	Acc.	99.42	99.51	99.38	99.17
	GE	0.0154	0.0126	0.0124	0.0116
	PGD	45.93	91.88	95.12	95.73

CIFAR10		Defensive Strength			
		0	4	8	16
VGG16 + Adv Trn + SP	Acc.	86.50	86.25	84.56	80.46
	GE	0.710	0.518	0.475	0.324
	PGD	6.91	42.53	52.58	55.75
VGG19 + Adv Trn + SP	Acc.	84.77	84.56	83.44	78.96
	GE	0.683	0.433	0.442	0.365
	PGD	13.77	42.89	51.85	55.40

CIFAR100		Defensive Strength			
		0	4	8	16
VGG16 + Adv Trn + SP	Acc.	49.27	49.94	48.08	46.44
	GE	0.989	1.160	1.012	0.835
	PGD	1.55	11.85	16.67	19.36
VGG19 + Adv Trn + SP	Acc.	43.26	43.35	42.83	42.31
	GE	1.420	1.433	1.325	1.020
	PGD	0.61	10.68	16.59	17.71

The numerical results on Tiny ImageNet are summarized in table 2.

Table 2. Top-1 accuracy on Tiny-ImageNet dataset with spectral normalization on wide residual networks. Different adversarial defensive strengths are adopted. We evaluate on clean images and adversarial images with the perturbation of $\|\gamma\|_\infty = 8$ from PGD attack. Defensive strength denotes the strength of adversarial examples we adpoted in training. The results of symbols are the same as table 1.

Tiny ImageNet		Defensive Strength			
		0	4	8	16
WideResNet + SP	Acc.	63.43	62.09	61.09	57.36
	GE	0.686	0.556	0.305	0.165
	PGD	0.00	20.03	25.17	30.47