Towards Certifying ℓ_∞ Robustness using Neural Networks with ℓ_∞-dist Neurons

Bohang Zhang [1] **Tianle Cai** [2 3] **Zhou Lu** [4] **Di He** [5] **Liwei Wang** [1 6]

Abstract

It is well-known that standard neural networks, even with a high classification accuracy, are vulnerable to small ℓ_∞-norm bounded adversarial perturbations. Although many attempts have been made, most previous works either can only provide empirical verification of the defense to a particular attack method, or can only develop a certified guarantee of the model robustness in limited scenarios. In this paper, we seek for a new approach to develop a theoretically principled neural network that inherently resists ℓ_∞ perturbations. In particular, we design a novel neuron that uses ℓ_∞-distance as its basic operation (which we call ℓ_∞-dist neuron), and show that any neural network constructed with ℓ_∞-dist neurons (called ℓ_∞-dist net) is naturally a 1-Lipschitz function with respect to ℓ_∞-norm. This directly provides a rigorous guarantee of the certified robustness based on the margin of prediction outputs. We then prove that such networks have enough expressive power to approximate any 1-Lipschitz function with robust generalization guarantee. We further provide a holistic training strategy that can greatly alleviate optimization difficulties. Experimental results show that using ℓ_∞-dist nets as basic building blocks, we consistently achieve state-of-the-art performance on commonly used datasets: 93.09% certified accuracy on MNIST ($\epsilon = 0.3$), 35.42% on CIFAR-10 ($\epsilon = 8/255$) and 16.31% on TinyImageNet ($\epsilon = 1/255$).

[1]Key Laboratory of Machine Perception, MOE, School of EECS, Peking University [2]Department of Electrical and Computer Engineering, Princeton University [3]Zhongguancun Haihua Institute for Frontier Information Technology [4]Department of Computer Science, Princeton University [5]Microsoft Research [6]Center for Data Science, Peking University. Correspondence to: Liwei Wang <wanglw@cis.pku.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

1. Introduction

Modern neural networks are usually sensitive to small, adversarially chosen perturbations to the inputs (Szegedy et al., 2013; Biggio et al., 2013). Given an image $\boldsymbol{x}$ that is correctly classified by a neural network, a malicious attacker may find a small adversarial perturbation $\boldsymbol{\delta}$ such that the perturbed image $\boldsymbol{x} + \boldsymbol{\delta}$, though visually indistinguishable from the original image, is assigned to a wrong class with high confidence by the network. Such vulnerability creates security concerns in many real-world applications.

Developing a model that can resist small ℓ_∞ perturbations has been extensively studied in the literature. Adversarial training methods (Szegedy et al., 2013; Goodfellow et al., 2015; Madry et al., 2017; Zhang et al., 2019; Ding et al., 2020) first generate on-the-fly adversarial examples of the inputs, then update model parameters using these perturbed samples together with the original labels. While such approaches can achieve decent empirical robustness, the evaluation is restricted to a particular (class of) attack method, and there are no formal guarantees whether the resulting model is robust against other attacks (Athalye et al., 2018; Tramer et al., 2020; Tjeng et al., 2019).

Another line of algorithms train provably robust models for standard networks by maximizing the certified radius provided by robust certification methods, typically using linear relaxation (Wong & Kolter, 2018; Weng et al., 2018; Mirman et al., 2018; Zhang et al., 2018; Wang et al., 2018; Singh et al., 2018), semidefinite relaxation (Raghunathan et al., 2018; Dvijotham et al., 2020), interval bound relaxation (Mirman et al., 2018; Gowal et al., 2018) or their combinations (Zhang et al., 2020b). However, most of these methods are sophisticated to implement and computationally expensive. Besides these approaches, Cohen et al. (2019a); Salman et al. (2019); Zhai et al. (2020) study the certified guarantee on ℓ_2 perturbations for Gaussian smoothed classifiers. However, recent works suggest that such methods are hard to extend to the ℓ_∞-perturbation scenario if the input dimension is large.

In this work, we propose a new approach by introducing a novel type of neural network that naturally resists local adversarial attacks, and can be easily certified under ℓ_∞ per-

turbation. In particular, we propose a novel neuron called ℓ_∞-dist neuron. Unlike the standard neuron design that uses a linear transformation followed by a non-linear activation, the ℓ_∞-dist neuron is purely based on computing the ℓ_∞-distance between the inputs and the parameters. It is straightforward to see that such a neuron is 1-Lipschitz with respect to ℓ_∞-norm, and the neural networks constructed with ℓ_∞-dist neurons (called ℓ_∞-dist nets) enjoy the same property. Based on such a property, we can efficiently obtain the certified robustness for any ℓ_∞-dist net using the margin of the prediction outputs.

Theoretically, we investigate the expressive power of ℓ_∞-dist nets and their robust generalization ability. We first prove a Lipschitz-universal approximation theorem which shows that ℓ_∞-dist nets can approximate any 1-Lipschitz function (with respect to ℓ_∞-norm) arbitrarily well. We then give upper bounds of robust test error, which would be small if the ℓ_∞-dist net learns a large margin classifier on the training data. These results demonstrate the excellent expressivity and generalization ability of the ℓ_∞-dist net function class.

While ℓ_∞-dist nets have nice theoretical guarantees, training such a network is still challenging. For example, the gradient of the parameters for ℓ_∞-norm distance is sparse, which makes the optimization difficult. In addition, we find that commonly used tricks and techniques in conventional network training cannot be taken for granted for this fundamentally different architecture. We address these challenges by proposing a holistic strategy for ℓ_∞-dist net training. Specifically, we show how to initialize the model parameters, apply proper normalization, design suitable weight decay mechanism, and overcome the sparse gradient problem via smoothed approximated gradients. Using the above methods, training an ℓ_∞-dist net is just as easy as training a standard network without any adversarial training, even though the resulting model is already provably robust.

Furthermore, the ℓ_∞-dist net has wide adaptability by serving as a robust feature extractor and combining itself with conventional networks for practical applications. After building a simple 2-layer perceptron on top of an ℓ_∞-dist net, we show that the model allows fast training and certification, and consistently achieves state-of-the-art certified robustness on a wide range of classification tasks. Concretely, we reach **93.09%** certified accuracy on MNIST under perturbation $\epsilon = 0.3$, **79.23%** on FashionMNIST under $\epsilon = 0.1$, **35.42%** on CIFAR-10 under $\epsilon = 8/255$, and **16.31%** on TinyImageNet under $\epsilon = 1/255$. As a comparison, these results outperform the previous best-known results (Xu et al., 2020a), in which they achieve 33.38% certified accuracy on CIFAR-10 dataset, and achieve 15.86% certified accuracy on TinyImageNet using a WideResNet model which is *33 times larger* than the ℓ_∞-dist net.

Our contributions are summarized as follows:

- We propose a novel neural network using ℓ_∞-dist neurons, called ℓ_∞-dist net. We show that any ℓ_∞-dist net is 1-Lipschitz with respect to ℓ_∞-norm, which directly guarantees the certified robustness (Section 3).
- In the theoretical part, we prove that ℓ_∞-dist nets can approximate any 1-Lipschitz function with respect to ℓ_∞-norm. We also prove that ℓ_∞-dist nets have a good robust generalization ability (Section 4).
- In the algorithmic part, we provide a holistic training strategy for ℓ_∞-dist nets, including parameter initialization, normalization, weight decay and smoothed approximated gradients (Section 5).
- We show how to combine ℓ_∞-dist nets with standard networks and obtain robust models more effectively (Section 6). Experimental results show that we can consistently achieve state-of-the-art certified accuracy on MNIST, Fashion-MNIST, CIFAR-10 and TinyImageNet dataset (Section 7).
- Finally, we provide all the implementation details and codes at https://github.com/zbh2047/L_inf-dist-net.

2. Related Work

Robust Training Approaches. Adversarial training is the most successful method against adversarial attacks. By adding adversarial examples to the training set on the fly, adversarial training methods (Szegedy et al., 2013; Goodfellow et al., 2015; Madry et al., 2017; Huang et al., 2015; Zhang et al., 2019; Wong et al., 2020; Ding et al., 2020) can significantly improve the robustness of conventional neural networks. However, all the methods above are evaluated according to the empirical robust accuracy against pre-defined adversarial attack algorithms, such as projected gradient decent. These methods cannot guarantee whether the resulting model is also robust against other attacks.

Certified Robustness for Conventional Networks. Many recent works focus on certifying the robustness of learned neural networks under *any* attack. These approaches are mainly based on bounding the certified radius layer by layer using some convex relaxation methods (Wong & Kolter, 2018; Wong et al., 2018; Weng et al., 2018; Mirman et al., 2018; Dvijotham et al., 2018b; Zhang et al., 2018; Wang et al., 2018; Singh et al., 2018; Xiao et al., 2019; Balunovic & Vechev, 2020; Raghunathan et al., 2018; Dvijotham et al., 2020). However, such approaches are usually complicated, computationally expensive and have difficulties in applying to deep and large models. To overcome these drawbacks, Mirman et al. (2018); Gowal et al. (2018) considered interval bound propagation (IBP), a special form of convex relaxation which is much simpler

and computationally cheaper. However, the produced bound is loose which results in unstable training. Zhang et al. (2020b); Xu et al. (2020a) took a further step to combine IBP with linear relaxation to make the bound tighter, which achieves current state-of-the-art performance. Fundamentally different from all these approaches that target to certify conventional networks, we proposed a novel network that provides robustness guarantee by its nature.

Certified Robustness for Smoothed Classifiers. Randomized smoothing can provide a (probabilistic) certified robustness guarantee for general models. Lecuyer et al. (2018); Li et al. (2018); Cohen et al. (2019a); Salman et al. (2019); Zhai et al. (2020); Zhang et al. (2020a) showed that if a Gaussian random noise is added to the input, a certified guarantee on small ℓ_2 perturbation can be computed for Gaussian smoothed classifiers. However, Yang et al. (2020); Blum et al. (2020); Kumar et al. (2020) showed that randomized smoothing cannot achieve nontrivial certified accuracy against larger than $\Omega\left(\min\left(1, d^{1/p-1/2}\right)\right)$ radius for ℓ_p perturbations, where d is the input dimension. Therefore it cannot provide meaningful results for a relatively large ℓ_∞ perturbation due to the curse of dimensionality.

Lipschitz Networks. Another line of approaches sought to bound the global Lipschitz constant of the neural network. Lipschitz networks can be very useful in certifying adversarial robustness (Tsuzuku et al., 2018), proving generalization bounds (Sokolić et al., 2017), or estimating Wasserstein distance (Arjovsky et al., 2017). Previous works trained Lipschitz ReLU networks by either directly constraining the spectral norm of each weight matrix to be less than one, or optimizing a loss constructed using the global Lipschitz constant which is upper bounded by these spectral norms (Cisse et al., 2017; Yoshida & Miyato, 2017; Gouk et al., 2018; Tsuzuku et al., 2018; Qian & Wegman, 2019). However, as pointed out by Huster et al. (2018) and Anil et al. (2019), such Lipschitz networks lack expressivity to some simple Lipschitz functions and the global Lipschitz bound is not tight. Recently, Anil et al. (2019) proposed a new Lipschitz network that is a Lipschitz-universal approximator. (Li et al., 2019) extended their work to convolutional architectures. However, the robustness performances are still not as good as other certification methods, and none of these methods can provide good certified results for ℓ_∞ robustness. In this work, we show the proposed ℓ_∞-dist net is also a Lipschitz-universal approximator (under ℓ_∞-norm), and the Lipschitz ℓ_∞-dist net can substantially outperform other Lipschitz networks in term of certified accuracy.

ℓ_p-dist Neurons. We notice that recently Chen et al. (2020); Xu et al. (2020b) proposed a new network called AdderNet, which leverages ℓ_1-norm operation to build the network for sake of efficient inference. Wang et al. (2019) also considered replacing dot-product neurons by ℓ_1-dist or ℓ_2-dist neurons in order to enhance the model's nonlinearity and expressivity. Although ℓ_∞-dist net looks similar to these networks on the surface, they are designed for fundamentally different problems, and in fact ℓ_1-dist neurons (or ℓ_2-dist neurons) can not give any robust guarantee for norm-bounded perturbations (see Appendix C for more discussions).

3. ℓ_∞-dist Network and its Robustness Guarantee

3.1. Preliminaries

Consider a standard classification task. Suppose we have an underlying data distribution $\mathcal{D}$ over pairs of examples $\boldsymbol{x} \in \mathcal{X}$ and corresponding labels $y \in \mathcal{Y} = \{1, 2, \cdots, M\}$ where M is the number of classes. Usually $\mathcal{D}$ is unknown and we can only access a training set $\mathcal{T} = \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$ in which $(\boldsymbol{x}_i, y_i)$ is *i.i.d.* drawn from $\mathcal{D}$. Let $f \in \mathcal{F}$ be the classifier of interest that maps any $\boldsymbol{x} \in \mathcal{X}$ to $\mathcal{Y}$. We call $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{\delta}$ an *adversarial example* of $\boldsymbol{x}$ to classifier f if f can correctly classify $\boldsymbol{x}$ but assigns a different label to $\boldsymbol{x}'$. In real practice, the most commonly used setting is to consider the attack under ϵ-bounded ℓ_∞-norm constraint, i.e., $\boldsymbol{\delta}$ satisfies $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$, which is also called ℓ_∞ perturbations.

Our goal is to learn a model from $\mathcal{T}$ that can resist attacks at $(\boldsymbol{x}, y)$ over $(\boldsymbol{x}, y) \sim \mathcal{D}$ for any small ℓ_∞ perturbation. It relates to compute the radius of the largest ℓ_∞ ball centered at $\boldsymbol{x}$ in which f does not change its prediction. This radius is called the *robust radius*, which is defined as (Zhai et al., 2020; Zhang et al., 2019):

$$R(f; \boldsymbol{x}, y) = \begin{cases} \inf\limits_{f(\boldsymbol{x}') \neq f(\boldsymbol{x})} \|\boldsymbol{x}' - \boldsymbol{x}\|_\infty, & f(\boldsymbol{x}) = y \\ 0, & f(\boldsymbol{x}) \neq y \end{cases} \tag{1}$$

Unfortunately, exactly computing the robust radius of a classifier induced by a standard deep neural network is very difficult. For example, Katz et al. (2017) showed that calculating such radius for a DNN with ReLU activation is NP-hard. Researchers then seek to derive a tight *lower bound* of $R(f; \boldsymbol{x}, y)$ for general f. Such lower bound is called *certified radius* and we denote it as $CR(f; \boldsymbol{x}, y)$. It follows that $CR(f; \boldsymbol{x}, y) \leq R(f; \boldsymbol{x}, y)$ for any $f, \boldsymbol{x}, y$.

3.2. Networks with ℓ_∞-dist Neurons

In this subsection, we propose a novel neuron called the ℓ_∞-dist neuron, which is inherently robust with respect to ℓ_∞-norm perturbations. Using these neurons as building blocks, we then show how to obtain robust neural networks dubbed ℓ_∞-dist nets.

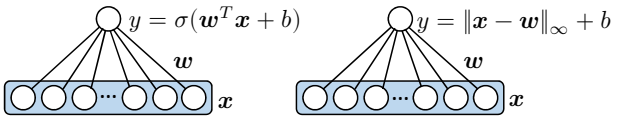

Figure 1. Illustration of the conventional neuron (left) and the ℓ_∞-dist neuron (right).

Denote $\boldsymbol{x}$ as the input vector to a neuron. A standard neuron processes the input by first projecting $\boldsymbol{x}$ to a scalar value using a linear transformation, then applying a non-linear activation function σ on it, i.e., $\sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$ with $\boldsymbol{w}$ and b as parameters and function σ being sigmoid or ReLU activation. Unlike the previous design paradigm, we introduce a new type of neuron using ℓ_∞ distance as the basic operation, called ℓ_∞-dist neuron:

$$u(\boldsymbol{x}, \theta) = \|\boldsymbol{x} - \boldsymbol{w}\|_\infty + b, \tag{2}$$

where $\theta = \{\boldsymbol{w}, b\}$ is the parameter set (see Figure 1 for an illustration). From Eqn. 2 we can see that the ℓ_∞-dist neuron is non-linear as it calculates the ℓ_∞-distance between input $\boldsymbol{x}$ and parameter $\boldsymbol{w}$ with a bias term b. As a result, there is no need to further apply a non-linear activation function.

Remark 3.1. *Conventional neurons use dot-product to represent the similarity between input $\boldsymbol{x}$ and weight $\boldsymbol{w}$. Likewise, ℓ_∞-distance is also a similarity measure. Note that ℓ_∞-distance is always non-negative, and a smaller ℓ_∞-distance indicates a stronger similarity.*

Without loss of generality, we study the properties of multi-layer perceptron (MLP) networks constructed using ℓ_∞-dist neurons. All theoretical results can be easily extended to other neural network architectures, such as convolutional networks. We use $\boldsymbol{x} \in \mathbb{R}^{d_{\text{input}}}$ to denote the input vector of an MLP network. An MLP network using ℓ_∞-dist neurons can be formally defined as follows.

Definition 3.2. *(ℓ_∞-dist Net) Define an L layer ℓ_∞-dist net as follows. Assume the l-th hidden layer contains d_l hidden units. The network takes $\boldsymbol{x}^{(0)} \triangleq \boldsymbol{x} \in \mathbb{R}^{d_{\text{input}}}$ as input, and the k-th unit in the l-th hidden layer $x_k^{(l)}$ is computed by*

$$x_k^{(l)} = u(\boldsymbol{x}^{(l-1)}, \theta^{(l,k)}) = \|\boldsymbol{x}^{(l-1)} - \boldsymbol{w}^{(l,k)}\|_\infty + b^{(l,k)}, \quad 1 \le l \le L, 1 \le k \le d_l \tag{3}$$

where $\boldsymbol{x}^{(l)} = (x_1^{(l)}, x_2^{(l)}, \cdots, x_{d_l}^{(l)})$ is the output of the l-th layer.

For classification tasks, the dimension of the final outputs of an ℓ_∞-dist net matches the number of categories, i.e., M. Based on Remark 3.1, we use the negative of the final layer to be the outputs of the ℓ_∞-dist net $\boldsymbol{g}$, i.e. $\boldsymbol{g}(\boldsymbol{x}) = (-x_1^{(L)}, -x_2^{(L)}, \cdots, -x_M^{(L)})$ and define the predictor $f(\boldsymbol{x}) = \arg\max_{i \in [M]} g_i(\boldsymbol{x})$. Similar to conventional networks, we can apply any standard loss function on the ℓ_∞-dist net, such as the cross-entropy loss or hinge loss.

3.3. Lipschitz and Robustness Facts about ℓ_∞-dist Nets

In this subsection, we will show that the ℓ_∞-dist neurons and the neural networks constructed using them have nice theoretical properties in controlling the robustness of the model. We first show that ℓ_∞-dist nets are 1-Lipschitz with respect to ℓ_∞-norm, then derive the certified robustness of the model based on such property.

Definition 3.3. *(Lipschitz Function) A function $\boldsymbol{g}(\boldsymbol{z}) : \mathbb{R}^m \to \mathbb{R}^n$ is called λ-Lipschitz with respect to ℓ_p-norm $\|\cdot\|_p$, if for any $\boldsymbol{z}_1, \boldsymbol{z}_2$, the following holds:*

$$\|\boldsymbol{g}(\boldsymbol{z}_1) - \boldsymbol{g}(\boldsymbol{z}_2)\|_p \le \lambda \|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_p$$

Fact 3.4. *Any ℓ_∞-dist net $\boldsymbol{g}(\cdot)$ is 1-Lipschitz with respect to ℓ_∞-norm, i.e., for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{d_{\text{input}}}$, we have $\|\boldsymbol{g}(\boldsymbol{x}_1) - \boldsymbol{g}(\boldsymbol{x}_2)\|_\infty \le \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_\infty$.*

Proof. It's easy to check that every basic operation $u(\boldsymbol{x}^{(l-1)}, \theta^{(l,k)})$ is 1-Lipschitz, and therefore the mapping from one layer to the next $\boldsymbol{x}^{(l)} \to \boldsymbol{x}^{(l+1)}$ is 1-Lipschitz. Finally by composition we have for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{d_{input}}$, $\|\boldsymbol{g}(\boldsymbol{x}_1) - \boldsymbol{g}(\boldsymbol{x}_2)\|_\infty \le \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_\infty$. □

Remark 3.5. *Fact 3.4 is only true when the basic neuron uses* infinity-norm *distance. For a network constructed using ℓ_p-dist neurons where $p < \infty$ (e.g. 2-norm), such network will not be 1-Lipschitz (even with respect to ℓ_p-norm) because the mapping from one layer to the next $\boldsymbol{x}^{(l)} \to \boldsymbol{x}^{(l+1)}$ is not 1-Lipschitz.*

Since $\boldsymbol{g}$ is 1-Lipschitz with respect to ℓ_∞-norm, if the perturbation over $\boldsymbol{x}$ is rather small, the change of the output can be bounded and the prediction of the perturbed data $\boldsymbol{x}'$ will not change as long as $\arg\max_{i \in [M]} g_i(\boldsymbol{x}) = \arg\max_{i \in [M]} g_i(\boldsymbol{x}')$, which directly bounds the certified radius.

Fact 3.6. *Given model $f(\boldsymbol{x}) = \arg\max_{i \in [M]} g_i(\boldsymbol{x})$ defined above, and $\boldsymbol{x}$ being correctly classified, we define $\text{margin}(\boldsymbol{x}; \boldsymbol{g})$ as the difference between the largest and second-largest elements of $\boldsymbol{g}(\boldsymbol{x})$. Then for any $\boldsymbol{x}'$ satisfying $\|\boldsymbol{x} - \boldsymbol{x}'\|_\infty < \text{margin}(\boldsymbol{x}; \boldsymbol{g})/2$, we have that $f(\boldsymbol{x}) = f(\boldsymbol{x}')$. In other words,*

$$CR(f, \boldsymbol{x}, y) \ge \text{margin}(\boldsymbol{x}; \boldsymbol{g})/2 \tag{4}$$

Proof. Since $\boldsymbol{g}(\boldsymbol{x})$ is 1-Lipschitz, each element of $\boldsymbol{g}(\boldsymbol{x})$ can move at most $\text{margin}(\boldsymbol{x}; \boldsymbol{g})/2$ when $\boldsymbol{x}$ changes to $\boldsymbol{x}'$, therefore the largest element will remain the same. □

Using this bound, we can certify the robustness of an ℓ_∞-dist net *of any size* under ℓ_∞-norm perturbations with little computational cost (*only a forward pass*). In contrast, existing certified methods may suffer from either poor scalability (methods based on linear relaxation) or curse of dimensionality (randomized smoothing).

4. Theoretical Properties of ℓ_∞-dist Nets

The expressive power of a model family and its generalization are two central topics in machine learning. Since we have shown that ℓ_∞-dist nets are 1-Lipschitz with respect to ℓ_∞-norm, it's natural to ask whether ℓ_∞-dist nets can approximate *any* 1-Lipschitz function (with respect to ℓ_∞-norm) and whether we can give generalization guarantee on the robust test error based on the Lipschitz property. In this section, we give affirmative answers to both questions. Without loss of generality, we consider *binary classification* problems and assume the output dimension is 1. All the omitted proofs in this section can be found in Appendix A.

4.1. Lipschitz-Universal Approximation of ℓ_∞-dist Nets

It is well-known that the conventional network is a universal approximator, in that it can approximate any continuous function arbitrarily well (Cybenko, 1989). Similarly, in this section we will prove a Lipschitz-universal approximation theorem for ℓ_∞-dist nets, formalized in the following:

Theorem 4.1. *For any 1-Lipschitz function $\tilde{g}(\boldsymbol{x})$ (with respect to ℓ_∞-norm) on a bounded domain $\mathbb{K} \in \mathbb{R}^{d_{input}}$ and any $\epsilon > 0$, there exists an ℓ_∞-dist net $g(\boldsymbol{x})$ with width no more than $d_{input} + 2$, such that for all $\boldsymbol{x} \in \mathbb{K}$, we have $\|g(\boldsymbol{x}) - \tilde{g}(\boldsymbol{x})\|_\infty \leq \epsilon$.*

We briefly present a proof sketch of Theorem 4.1. The proof has the same structure to Lu et al. (2017), who first proved such universal approximation theorem for width-bounded ReLU networks, by constructing a special network that approximates a target function $\tilde{g}(\boldsymbol{x})$ by the sum of indicator functions of grid points, i.e. $\tilde{g}(\boldsymbol{x}) \approx \sum_{\boldsymbol{z}\in\mathbb{S}} \tilde{g}(\boldsymbol{z}) \mathbf{1}_{\{\boldsymbol{x}:\|\boldsymbol{x}-\boldsymbol{z}\|_\infty \leq \epsilon/2\}}(\boldsymbol{x})$ where $\mathbb{S} = \{\boldsymbol{z} \in \mathbb{K} : z_i = C_i\epsilon, C_i \in \mathbb{Z}\}$. For ℓ_∞-dist nets, such an approach cannot be directly applied as the summation will break the Lipschitz property. We employ a novel "max of pyramids" construction to overcome the issue. The key idea is to approximate the target function using the maximum of many "pyramid-like" basic 1-Lipschitz functions, i.e. $\tilde{g}(\boldsymbol{x}) \approx g(x) := \max_{\boldsymbol{z}\in\mathbb{S}}(\tilde{g}(\boldsymbol{z}) - \|\boldsymbol{x} - \boldsymbol{z}\|_\infty)$. To represent such a function, we first show that the ℓ_∞-dist neuron can express the following basic functions: $f(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{z}\|_\infty$, $f(\boldsymbol{x}) = x_i + b$, $f(\boldsymbol{x}) = -x_i + b$ and $f(\boldsymbol{x}) = \max(x_i, x_j)$; Then $g(x)$ can be constructed using these functions as building blocks. Finally, we carefully design a computation pattern for a width-bounded ℓ_∞-dist net to perform such max-reduction layer by layer.

Theorem 4.1 implies that an ℓ_∞-dist net can approximate any 1-Lipschitz function with respect to ℓ_∞-norm on a compact set, using width barely larger than the input dimension. Combining it with Fact 3.4, we conclude that ℓ_∞-dist nets are a good class of models to approximate 1-Lipschitz functions.

4.2. Bounding Robust Test Error of ℓ_∞-dist Nets

In this subsection, we give a generalization bound for the *robust test error* of ℓ_∞-dist nets. Let $(\boldsymbol{x}, y)$ be an instance-label couple where $\boldsymbol{x} \in \mathbb{K}$ and $y \in \{1, -1\}$ and denote $\mathcal{D}$ as the distribution of $(\boldsymbol{x}, y)$. For a function $g(\boldsymbol{x}) : \mathbb{R}^{d_{\text{input}}} \to \mathbb{R}$, we use $\text{sign}(g(\boldsymbol{x}))$ as the classifier. The r-robust test error γ_r of a classifier g is defined as

$$\gamma_r = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\sup_{\|\boldsymbol{x}'-\boldsymbol{x}\|_\infty \leq r} \mathbb{I}_{yg(\boldsymbol{x}')\leq 0} \right].$$

Then γ_r can be upper bounded by the margin error on training data and the size of the network, as stated in the following theorem:

Theorem 4.2. *Let $\mathbb{F}$ denote the set of all g represented by an ℓ_∞-dist net with width $W (W \geq d_{input})$ and depth L. For every $t > 0$, with probability at least $1 - 2e^{-2t^2}$ over the random drawing of n samples, for all $r > 0$ and $g \in \mathbb{F}$ we have that*

$$\gamma_r \leq \inf_{\delta\in(0,1]} \left[\frac{1}{n} \sum_{i=1}^{n} \underbrace{\mathbb{I}_{y_i g(\boldsymbol{x}_i)\leq \delta + r}}_{\text{large training margin}} + \underbrace{\tilde{O}\left(\frac{LW^2}{\delta\sqrt{n}}\right)}_{\text{network size}} + \left(\frac{\log\log_2(\frac{2}{\delta})}{n}\right)^{\frac{1}{2}} \right] + \frac{t}{\sqrt{n}}. \tag{5}$$

Theorem 4.2 demonstrates that when a large margin classifier is found on training data, and the size of the ℓ_∞-dist net is not too large, then with high probability, the model can generalize well in terms of *adversarial robustness*. Note that our bound does not depend on the input dimension, while previous generalization bounds for the general Lipschitz model class (i.e. Luxburg & Bousquet (2004)) suffer from the curse of dimensionality (Neyshabur et al., 2017).

5. Training ℓ_∞-dist Nets

In this section we will focus on how to train an ℓ_∞-dist net successfully. Motivated by the theoretical analysis in previous sections, it suffices to find a large margin solution on the training data. Therefore we can simply use the standard multi-class hinge loss to obtain a large training margin, similar to Anil et al. (2019). Since the loss is differentiable almost everywhere, any gradient-based optimization method can be used to train ℓ_∞-dist nets.

However, we empirically find that the optimization is challenging and directly training the network usually *fails* to obtain a good performance. Moreover, conventional wisdom like batch normalization cannot be taken as a grant in the ℓ_∞-dist net since it will hurt the model's robustness. In the following we will dig into the optimization difficulties and provide a holistic training strategy to overcome them.

5.1. Normalization

One important difference between ℓ_∞-dist nets and conventional networks is that for conventional networks, the output of a linear layer is unbiased (zero mean in expectation) under random initialization over parameters (Glorot & Bengio, 2010; He et al., 2015), while the output of an ℓ_∞-dist neuron is biased (always being non-negative, assuming no bias term). Indeed, for a weight vector initialized using a standard Gaussian distribution, assume a zero input $\boldsymbol{x}^{(0)} = \mathbf{0} \in \mathbb{R}^d$ is fed into an ℓ_∞-dist net, the expected output of the first layer can be approximated by $x_j^{(1)} = \mathbb{E}_{\boldsymbol{w}^{(1,j)}} \|\boldsymbol{x}^{(0)} - \boldsymbol{w}^{(1,j)}\|_\infty \approx \sqrt{2 \log d}$. The output vector $\boldsymbol{x}^{(1)}$ is then fed into subsequent layers, making the outputs in upper layers linearly increase.

Normalization is a useful way to control the scale of the layer's outputs to a standard range. Batch Normalization (Ioffe & Szegedy, 2015), which shifts and scales feature values in each layer, is shown to be one of the most important components in training deep neural networks. However, if we directly apply batch normalization in ℓ_∞-dist nets, the Lipschitz constant will change due to the scaling operation, and the robustness of the model cannot be guaranteed.

Fortunately, we find using the shift operation alone already helps the optimization. Therefore we apply the shift operation in all intermediate layers after calculating the ℓ_∞ distance. As a result, we remove the bias terms in the corresponding ℓ_∞-dist neurons as they are redundant. We do not use normalization in the final layer. Similar to BatchNorm, we use the running mean during inference, which serves as additional bias terms in ℓ_∞-dist neurons and does not affect the Lipschitz constant of the model. We do not use affine transformation, which is typically used in BatchNorm.

5.2. Smoothed Approximated Gradients

We find that training an ℓ_∞-dist net from scratch is usually inefficient, and the optimization can easily be stuck at some bad solution. One important reason is that the gradients of the ℓ_∞-dist operation (i.e., $\nabla_{\boldsymbol{w}} \|\boldsymbol{z} - \boldsymbol{w}\|_\infty$ and $\nabla_{\boldsymbol{z}} \|\boldsymbol{z} - \boldsymbol{w}\|_\infty$) are very sparse which typically contain only one non-zero element. In practice, we observe that there are less than 1% parameters updated in an epoch if we directly train the ℓ_∞-dist net using SGD/Adam from random initialization.

To improve the optimization, we relax the ℓ_∞-dist neuron by using the ℓ_p-dist neuron for the whole network to get an approximate and non-sparse gradient of the model parameters. During training, we set p to be a small value in the beginning and increase it in each iteration until it approaches infinity. For the last few epochs, we set p to infinity and train the model to the end. Empirically, using smoothed approximated gradients significantly boosts the performance, as will be shown in Section 7.3.

5.3. Parameter Initialization

We find that there are still optimization difficulties in training *deep* models using normalization and the smoothed gradient method. In particular, we find that a deeper model performs worse than its shallow counterpart in term of *training accuracy* (see Appendix E), a phenomenon similar to He et al. (2016). To fix the problem, He et al. (2016) proposed ResNet architecture by modifying the network using identity mapping as skip connections. According to Proposition A.2 (see Appendix A), an ℓ_∞-dist layer can also perform identity mapping by assigning proper weights and biases at initialization, and a deeper ℓ_∞-dist net then can act as a shallow one by these identity mappings.

Given such findings, we can directly construct identity mappings at initialization. Concretely, for an ℓ_∞-dist layer with the same input-output dimension, we first initialize the weights randomly from a standard Gaussian distribution as common, then modify the diagonal elements (i.e. $w_j^{(l,j)}$ in Definition 3.2) to be a large negative number C_0. Throughout all experiments, we set $C_0 = -10$. We do not need the bias in ℓ_∞-dist neurons after applying mean shift normalization, and the running mean automatically makes an identity mapping.

5.4. Weight Decay

Weight decay is a commonly used trick in training deep neural networks. It is equivalent to adding an ℓ_2 regularization term to the loss function. However, we empirically found that using weight decay in the ℓ_∞-dist net gives inferior performance (see Section 7.3). The problem might be the incompatibility of weight decay (ℓ_2 regularization) with ℓ_∞-norm used in ℓ_∞-dist nets, as we will explain below.

Conventional networks use dot-product as the basic operation, therefore an ℓ_2-norm constraint on the weight vectors directly controls the scale of the output magnitude. However, it is straightforward to see that the ℓ_2-norm of the weight vector does not correspond to the output scale of the ℓ_∞ distance operation. A more reasonable choice is to use ℓ_∞-norm regularization instead of ℓ_2-norm. In fact, we have $\|\boldsymbol{x} - \boldsymbol{w}\|_\infty \leq \|\boldsymbol{x}\|_\infty + \|\boldsymbol{w}\|_\infty$, analogous to $\langle \boldsymbol{x}, \boldsymbol{w} \rangle \leq \|\boldsymbol{x}\|_2 \|\boldsymbol{w}\|_2$.

For general ℓ_p-dist neurons during training, we can use ℓ_p-norm regularization analogously. By taking derivative with respect to the weight $\boldsymbol{w}$, we derive the corresponding weight decay formula:

$$\Delta_{w_i} = -\lambda \nabla_{w_i} \|\boldsymbol{w}\|_p^2 = -\lambda \left(\frac{|w_i|}{\|\boldsymbol{w}\|_p} \right)^{p-2} w_i \qquad (6)$$

where λ is the weight decay coefficient. Note that Eqn. 6 reduces to commonly used weight decay if $p = 2$. When $p \to \infty$, the weight decay tends to take effects only on the element w_i with the largest absolute value.

6. Certified Robustness by Using ℓ_∞-dist Nets as Robust Feature Extractors

We have shown that the ℓ_∞-dist net is globally Lipschitz in the sense that the function is 1-Lipschitz everywhere over the input space. This constraint is pretty strong when we only require function's Lipschitzness on a specific manifold, such as real image data manifold. To make the model more flexible and fit the practical tasks better, we can build a lightweight conventional network on top of an ℓ_∞-dist net. The ℓ_∞-dist net $\boldsymbol{g}$ will serve as a robust feature extractor, and the lightweight network $\boldsymbol{h}$ (a shallow MLP in our experiments) will focus on task-specific goals, such as classification. We denote the composite network as $\boldsymbol{h} \circ \boldsymbol{g}$.

We first show how to certify the robustness for the composite network $\boldsymbol{h} \circ \boldsymbol{g}$. Given any input $\boldsymbol{x}$, consider a perturbation set $\mathcal{B}_\infty^\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}' : \|\boldsymbol{x}' - \boldsymbol{x}\|_\infty \leq \epsilon\}$, where ϵ is the predefined perturbation level. If $(\boldsymbol{h} \circ \boldsymbol{g})(\boldsymbol{x}')$ predicts the correct label y for all $\boldsymbol{x}' \in \mathcal{B}_\infty^\epsilon(\boldsymbol{x})$, we can guarantee robustness of the network $\boldsymbol{h} \circ \boldsymbol{g}$ for input $\boldsymbol{x}$. Since $\boldsymbol{g}$ is 1-Lipschitz with respect to ℓ_∞-norm, we have $\boldsymbol{g}(\boldsymbol{x}') \in \mathcal{B}_\infty^\epsilon(\boldsymbol{g}(\boldsymbol{x}))$ by definition. Based on this property, to guarantee robustness of the network $\boldsymbol{h} \circ \boldsymbol{g}$ for input $\boldsymbol{x}$, it suffices to check whether for all $\boldsymbol{z}' \in \mathcal{B}_\infty^\epsilon(\boldsymbol{g}(\boldsymbol{x}))$, $\boldsymbol{h}(\boldsymbol{z}')$ predicts the correct label y. This is equivalent to certifying the robustness for a conventional network $\boldsymbol{h}$ given input $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x})$, which can be calculated using any previous certification method such as convex relaxation. We describe one of the simplest convex relaxation method named IBP (Gowal et al., 2018) in Appendix B, which is used in our experiments. Other more advanced approaches such as CROWN-IBP (Zhang et al., 2020b; Xu et al., 2020a) can also be considered.

After obtaining the bound, we can set it as the training objective function to train the neural network parameters, similar to Gowal et al. (2018); Zhang et al. (2020b). All calculations are differentiable, and gradient-based optimization methods can be applied. Note that since $\boldsymbol{h}$ and $\boldsymbol{g}$ have entirely different architectures, we apply the training strategy introduced in Section 5 to the ℓ_∞-dist net $\boldsymbol{g}$ only. More details will be presented in Section 7.1.

7. Experiments & Results

In this section, we conduct extensive experiments for the proposed network. We train our models on four popular benchmark datasets: MNIST, Fashion-MNIST, CIFAR-10 and TinyImagenet.

7.1. Experimental Setting

Model details. We mainly study two types of models. The first type is denoted as ℓ_∞-dist Net, i.e., a network consists of ℓ_∞-dist neurons only. The second type is a composition of ℓ_∞-dist Net and a shallow MLP (denoted as ℓ_∞-dist Net+MLP). That is, using ℓ_∞-dist Net as a robust feature extractor as described in Section 6. We use a 5-layer ℓ_∞-dist Net for MNIST and Fashion-MNIST, and a 6-layer ℓ_∞-dist Net for CIFAR-10 and TinyImageNet. Each hidden layer has 5120 neurons, and the top layer has 10 neurons (or 200 neurons for TinyImageNet) for classification. Normalization is applied between each intermediate layer. For ℓ_∞-dist Net+MLP, we remove the top layer and add a 2-layer fully connected conventional network on top of it. The hidden layer has 512 neurons with tanh activation. See Table 5 for a complete demonstration of the models on each dataset.

Training configurations. In all experiments, we train ℓ_∞-dist Net and ℓ_∞-dist Net+MLP using Adam optimizer with hyper-parameters $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-10}$. The batch size is set to 512. For data augmentation, we use random crop (padding=1) for MNIST and Fashion-MNIST, and use random crop (padding=4) and random horizontal flip for CIFAR-10, following the common practice. For TinyImageNet dataset, we use random horizontal flip and crop each image to 56×56 pixels for training, and use a center crop for testing, which is the same as Xu et al. (2020a). As for the loss function, we use multi-class hinge loss for ℓ_∞-dist Net and the IBP loss (Gowal et al., 2018) for ℓ_∞-dist Net+MLP. The training procedure is as follows. First, we relax the ℓ_∞-dist net to ℓ_p-dist net by setting $p = 8$ and train the network for e_1 epochs. Then we gradually increase p from 8 to 1000 exponentially in the next e_2 epochs. Finally, we set $p = \infty$ and train the last e_3 epochs. Here e_1, e_2 and e_3 are hyper-parameters varying from the dataset. We use $lr = 0.02$ in the first e_1 epochs and decease the learning rate using cosine annealing for the next $e_2 + e_3$ epochs. We use ℓ_p-norm weight decay for ℓ_∞-dist nets and ℓ_2-norm weight decay for the MLP with coefficient $\lambda = 0.005$. All these explicitly specified hyper-parameters are kept fixed across different architectures and datasets. For ℓ_∞-dist Net+MLP training, we use the same linear warmup strategy for hyper-parameter ϵ_{train} in Gowal et al. (2018); Zhang et al. (2020b). See Appendix D (Table 6) for details of training configuration and hyper-parameters.

Evaluation. Following common practice, we test the robustness of the trained models under ℓ_∞-perturbation $\epsilon = 0.3$ on MNIST, $\epsilon = 0.1$ on Fashion-MNIST, $\epsilon = 8/255$ on CIFAR-10 and $\epsilon = 1/255$ on TinyImageNet. We use two evaluation metrics to measure the robustness of the model. We first evaluate the robust test accuracy under the Projected Gradient Descent (PGD) attack (Madry et al., 2017). Following standard practice, we set the number of steps of the PGD attack to be 20. We also calculate the certified radius for each sample, and check the percentage of test samples that can be certified to be robust within the chosen radius. Note that the second metric is always lower than the first.

Table 1. Comparison of our results with existing methods[1].

Dataset	Method	FLOPs	Test	Robust	Certified
MNIST ($\epsilon = 0.3$)	Group Sort (Anil et al., 2019)	2.9M	97.0	34.0	2.0
	COLT (Balunovic & Vechev, 2020)	4.9M	97.3	-	85.7
	IBP (Gowal et al., 2018)	114M	97.88	93.22	91.79
	CROWN-IBP (Zhang et al., 2020b)	114M	98.18	93.95	92.98
	ℓ_∞-dist Net	82.7M	98.54	94.71	92.64
	ℓ_∞-dist Net+MLP	85.3M	**98.56**	**95.28**	**93.09**
Fashion MNIST ($\epsilon = 0.1$)	CAP (Wong & Kolter, 2018)	0.41M	78.27	68.37	65.47
	IBP (Gowal et al., 2018)	114M	84.12	80.58	77.67
	CROWN-IBP (Zhang et al., 2020b)	114M	84.31	80.22	78.01
	ℓ_∞-dist Net	82.7M	**87.91**	79.64	77.48
	ℓ_∞-dist Net+MLP	85.3M	**87.91**	**80.89**	**79.23**
CIFAR-10 ($\epsilon = 8/255$)	PVT (Dvijotham et al., 2018a)	2.4M	48.64	32.72	26.67
	DiffAI (Mirman et al., 2019)	96.3M	40.2	-	23.2
	COLT (Balunovic & Vechev, 2020)	6.9M	51.7	-	27.5
	IBP (Gowal et al., 2018)	151M	50.99	31.27	29.19
	CROWN-IBP (Zhang et al., 2020b)	151M	45.98	34.58	33.06
	CROWN-IBP (loss fusion) (Xu et al., 2020a)	151M	46.29	35.69	33.38
	ℓ_∞-dist Net	121M	**56.80**	**37.46**	33.30
	ℓ_∞-dist Net+MLP	123M	50.80	37.06	**35.42**

Baselines. We compare our proposed models with state-of-the-art methods for each dataset, including relaxation methods: CAP (Wong & Kolter, 2018), PVT (Dvijotham et al., 2018a), DiffAI (Mirman et al., 2019), IBP (Gowal et al., 2018), CROWN-IBP (Zhang et al., 2020b), CROWN-IBP with loss fusion (Xu et al., 2020a), COLT (Balunovic & Vechev, 2020), and Lipschitz networks: GroupSort (Anil et al., 2019). We do not compare with randomized smoothing methods (Cohen et al., 2019a; Salman et al., 2019; Zhai et al., 2020) since it cannot obtain good certification under s relative large ℓ_∞-norm perturbation as discussed in Section 2. We report the performances picked from the original papers if not specified otherwise.

7.2. Experimental Results

We list our results in Table 1 and Table 2. We use "Standard", "Robust" and "Certified" as abbreviations of standard (clean) test accuracy, robust test accuracy under PGD attack and certified robust test accuracy. All the numbers are reported in percentage. We use "FLOPs" to denote the number of basic floating-point operations needed (i.e., multiplication-add in conventional networks or subtraction in ℓ_∞-dist nets) in forward propagation. Note that FLOPs in linear relaxation methods are typically small because, in the original papers, they are implemented on small networks due to high costs.

General Performance of ℓ_∞-dist Net. From Table 1 we can see that using ℓ_∞-dist Net alone already achieves decent certified accuracy on all datasets. Notably, ℓ_∞-dist Net reaches the start-of-the-art certified accuracy on CIFAR-10 dataset while achieving a significantly higher standard accuracy than all previous methods. Note that we just use a standard loss function to train the ℓ_∞-dist Net *without any adversarial training*.

General Performance of ℓ_∞-dist Net+MLP. For all these datasets, ℓ_∞-dist Net+MLP achieves better certified accuracy than ℓ_∞-dist Net, establishing new state-of-the-art results. For example, on the MNIST dataset, the model can reach 93.09% certified accuracy and 98.56% standard accuracy; on CIFAR-10 dataset, the model can reach 35.42% certified accuracy, which is 6.23% higher than IBP and 2.04% higher than the previous best result; on the TinyImageNet dataset (see Table 2), the simple ℓ_∞-dist Net+MLP model (156M FLOPs computational cost) already beats the previous best result in Xu et al. (2020a) although their model is 33 times larger than ours (5.22G FLOPs). Furthermore, the clean test accuracy of ℓ_∞-dist Net+MLP is also better than CROWN-IBP on MNIST, Fashion-MNIST and CIFAR-10 dataset.

Efficiency. Both the training and the certification of ℓ_∞-dist Net are very fast. As stated in previous sections, the computational cost per iteration for training ℓ_∞-dist Net is roughly the same as training a conventional network of the same size, and the certification process only requires a forward pass. In Table 4, we quantitatively compare the per-epoch training speed of our method with previous methods such as IBP or CROWN-IBP on CIFAR-10 dataset. All these experiments are run on a single NVIDIA-RTX 3090 GPU. As we can see, for both ℓ_∞-dist Net and ℓ_∞-dist Net+MLP, the per-epoch training time is less than 20 seconds, which is significantly faster than CROWN-IBP and is comparable to IBP.

[1]For GroupSort, the results are obtained from Anil et al. (2019), Fig. 8,9. For COLT, the certified result uses MILP (mixed integer linear programming) solver which is much slower than other methods. For IBP, the results are obtained from Zhang et al. (2020b). For PVT, the certified accuracy are under perturbation $\epsilon = 0.03$ which is smaller than 8/255.

Table 2. Comparison of our results with Xu et al. (2020a) on TinyImageNet dataset ($\epsilon = 1/255$).

Method	Model	FLOPs	Test	Robust	Certified
CROWN-IBP (loss fusion) (Xu et al., 2020a)	CNN7+BN	458M	21.58	19.04	12.69
	ResNeXt	64M	21.42	20.20	13.05
	DenseNet	575M	22.04	19.48	14.56
	WideResNet	5.22G	27.86	20.52	15.86
ℓ_∞-dist net	ℓ_∞-dist Net+MLP	156M	21.82	18.09	**16.31**

Table 3. Ablation studies for Section 5 on CIFAR-10 dataset.

	Smooth gradient	Identity init	Weight decay	ℓ_∞-dist Net			ℓ_∞-dist Net+MLP		
				Test	Robust	Certified	Test	Robust	Certified
A	✗	✗	✗	28.21	8.42	7.02	37.68	28.72	27.76
B	✓	✗	✗	55.63	35.28	32.56	37.61	30.99	29.71
C	✓	✓	✗	56.15	35.96	32.71	48.97	36.21	35.02
D	✓	✓	ℓ_2-norm	53.64	32.12	29.01	45.34	33.48	32.49
E	✓	✓	✓	56.80	37.46	33.30	50.80	37.06	35.42

Table 4. Comparison of per-epoch training speed for different methods on CIFAR-10 dataset.

Method	Per-epoch Time (seconds)
IBP	17.4
CROWN-IBP	112.4
CROWN-IBP (loss fusion)	43.3
ℓ_∞-dist Net	19.7
ℓ_∞-dist Net+MLP	19.7

Discussions with GroupSort Network. We finally make a special comparison with GroupSort, because both GroupSort network and our proposed ℓ_∞-dist Net design 1-Lipschitz networks explicitly and can be directly trained using a standard loss function. However, in the GroupSort network, all weight matrices $\mathbf{W}$ are constrained to have bounded ℓ_∞-norm, i.e., $\|\mathbf{W}\|_\infty \leq 1$, leading to a time-consuming projection operation (see Appendix C in Anil et al. (2019)). This operation brings optimization difficulty (Cohen et al., 2019b) and further limit the scalability of the network structure. We hypothesis that this is the major reason why ℓ_∞-dist Net substantially outperforms GroupSort on the MNIST dataset, as shown in Table 1.

7.3. Ablation Studies

In this section, we conduct ablation experiments to see the effect of smoothed approximated gradients, parameter initialization using identity map construction, and ℓ_p-norm weight decay. The results are shown in Table 3. We use the model described in Section 7.1 on CIFAR-10 dataset with hyper-parameters provided in Table 6. From Table 3 we can clearly see that:

- Smoothed approximated gradient technique is crucial to train a good model for ℓ_∞-dist Net. After applying smoothed approximated gradient only, we can already achieve 32.56% certified accuracy;
- Both smoothed approximated gradient and identity-map initialization are crucial to train a good model for ℓ_∞-dist Net+MLP. Combining the two techniques results in 35.02% certified accuracy;
- ℓ_p-norm weight decay can further boost the results, although the effect may be marginal (0.59% and 0.4% improvement of certified accuracy for the two models).
- Conventional ℓ_2-norm weight decay does harm to the performance of ℓ_∞-dist nets.

In summary, these optimization strategies in Section 5 all contributes to the final performance of the model.

8. Conclusion

In this paper, we design a novel neuron that uses ℓ_∞ distance as its basic operation. We show that the neural network constructed with ℓ_∞-dist neuron is naturally a 1-Lipschitz function with respect to ℓ_∞ norm. This directly provides a theoretical guarantee of the certified robustness based on the margin of the prediction outputs. We further formally analyze the expressive power and the robust generalization ability of the network, and provide a holistic training strategy to handle optimization difficulties encountered in training ℓ_∞-dist nets. Experiments show promising results on MNIST, Fashion-MNIST, CIFAR-10 and TinyImageNet datasets. As this structure is entirely new, plenty of aspects are needed to investigate, such as how to further handle the optimization difficulties for this network. For future work we will study these aspects and extend our model to more challenging tasks like ImageNet.

Acknowledgements

This work was supported by National Key R&D Program of China (2018YFB1402600), Key-Area Research and Development Program of Guangdong Province (No. 2019B121204008), BJNSF (L172037) and Beijing Academy of Artificial Intelligence. Project 2020BD006 supported by PKU-Baidu Fund.

References

Anil, C., Lucas, J., and Grosse, R. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pp. 291–301, 2019.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *CoRR*, abs/1802.00420, 2018.

Balunovic, M. and Vechev, M. Adversarial training and provable defenses: Bridging the gap. In *International Conference on Learning Representations*, 2020.

Bartlett, P. L., Harvey, N., Liaw, C., and Mehrabian, A. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Blum, A., Dick, T., Manoj, N., and Zhang, H. Random smoothing might be unable to certify ℓ_∞ robustness for high-dimensional images. *arXiv preprint arXiv:2002.03517*, 2020.

Chen, H., Wang, Y., Xu, C., Shi, B., Xu, C., Tian, Q., and Xu, C. Addernet: Do we really need multiplications in deep learning? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1468–1477, 2020.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pp. 854–863, 2017.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320, Long Beach, California, USA, 09–15 Jun 2019a. PMLR.

Cohen, J. E., Huster, T., and Cohen, R. Universal lipschitz approximation in bounded depth neural networks. *arXiv preprint arXiv:1904.04861*, 2019b.

Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Ding, G. W., Sharma, Y., Lui, K. Y. C., and Huang, R. Mma training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations*, 2020.

Dvijotham, K., Gowal, S., Stanforth, R., Arandjelovic, R., O'Donoghue, B., Uesato, J., and Kohli, P. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265*, 2018a.

Dvijotham, K., Stanforth, R., Gowal, S., Mann, T. A., and Kohli, P. A dual approach to scalable verification of deep networks. In *UAI*, volume 1, pp. 2, 2018b.

Dvijotham, K. D., Stanforth, R., Gowal, S., Qin, C., De, S., and Kohli, P. Efficient neural network verification with exactness characterization. In Adams, R. P. and Gogate, V. (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 497–507, Tel Aviv, Israel, 22–25 Jul 2020. PMLR.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Huang, R., Xu, B., Schuurmans, D., and Szepesvári, C. Learning with a strong adversary. *arXiv preprint arXiv:1511.03034*, 2015.

Huster, T., Chiang, C.-Y. J., and Chadha, R. Limitations of the lipschitz constant as a defense against adversarial examples, 2018.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pp. 97–117. Springer, 2017.

Koltchinskii, V., Panchenko, D., et al. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1): 1–50, 2002.

Kumar, A., Levine, A., Goldstein, T., and Feizi, S. Curse of dimensionality on randomized smoothing for certifiable robustness. *arXiv preprint arXiv:2002.03239*, 2020.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. *arXiv preprint arXiv:1802.03471*, 2018.

Li, B., Chen, C., Wang, W., and Carin, L. Second-order adversarial attack and certifiable robustness. *CoRR*, abs/1809.03113, 2018.

Li, Q., Haque, S., Anil, C., Lucas, J., Grosse, R. B., and Jacobsen, J.-H. Preventing gradient attenuation in lipschitz constrained convolutional networks. In *Advances in Neural Information Processing Systems*, volume 32, pp. 15390–15402. Curran Associates, Inc., 2019.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6231–6239. Curran Associates, Inc., 2017.

Luxburg, U. v. and Bousquet, O. Distance-based classification with lipschitz functions. *Journal of Machine Learning Research*, 5(Jun):669–695, 2004.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Mirman, M., Gehr, T., and Vechev, M. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pp. 3578–3586, 2018.

Mirman, M., Singh, G., and Vechev, M. A provable defense for deep residual networks. *arXiv preprint arXiv:1903.12519*, 2019.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

Qian, H. and Wegman, M. N. L2-nonexpansive neural networks. In *International Conference on Learning Representations*, 2019.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pp. 11292–11303, 2019.

Singh, G., Gehr, T., Mirman, M., Püschel, M., and Vechev, M. Fast and effective robustness certification. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 10802–10813. Curran Associates, Inc., 2018.

Sokolić, 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, 2017. doi: 10.1109/TSP.2017.2708039.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

Tjeng, V., Xiao, K. Y., and Tedrake, R. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyGIdiRqtm.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *Advances in neural information processing systems*, pp. 6541–6550, 2018.

von Luxburg, U. and Bousquet, O. Distance-based classification with lipschitz functions. *J. Mach. Learn. Res.*, 5: 669–695, 2004.

Wang, C., Yang, J., Xie, L., and Yuan, J. Kervolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31–40, 2019.

Wang, S., Chen, Y., Abdou, A., and Jana, S. Mixtrain: Scalable training of formally robust neural networks. *arXiv preprint arXiv:1811.02625*, 2018.

Weng, L., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Daniel, L., Boning, D., and Dhillon, I. Towards fast computation of certified robustness for ReLU networks. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5276–5285, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2018.

Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8400–8409. Curran Associates, Inc., 2018.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

Xiao, K. Y., Tjeng, V., Shafiullah, N. M. M., and Madry, A. Training for faster adversarial robustness verification via inducing reLU stability. In *International Conference on Learning Representations*, 2019.

Xu, K., Shi, Z., Zhang, H., Wang, Y., Chang, K.-W., Huang, M., Kailkhura, B., Lin, X., and Hsieh, C.-J. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33, 2020a.

Xu, Y., Xu, C., Chen, X., Zhang, W., Xu, C., and Wang, Y. Kernel based progressive distillation for adder neural networks, 2020b.

Yang, G., Duan, T., Hu, E., Salman, H., Razenshteyn, I., and Li, J. Randomized smoothing of all shapes and sizes. *arXiv preprint arXiv:2002.08118*, 2020.

Yoshida, Y. and Miyato, T. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C.-J., and Wang, L. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378*, 2020.

Zhang, D., Ye, M., Gong, C., Zhu, Z., and Liu, Q. Black-box certification with randomized smoothing: A functional optimization based framework. *arXiv preprint arXiv:2002.09169*, 2020a.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*, pp. 4939–4948, 2018.

Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L. E., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D., and Hsieh, C.-J. Towards stable and efficient training of verifiably robust neural networks. In *International Conference on Learning Representations*, 2020b.