Constraining Logits by Bounded Function for Adversarial Robustness

Hiroshi Takahashi  
NTT  
Tokyo, Japan  
hiroshi.takahashi.bm@hco.ntt.co.jp

Masanori Yamada  
NTT  
Tokyo, Japan  
masanori.yamada.cm@hco.ntt.co.jp

Shin’ya Yamaguchi  
NTT  
Tokyo, Japan  
shinya.yamaguchi.mw@hco.ntt.co.jp

Sekitoshi Kanai  
NTT  
Tokyo, Japan  
sekitoshi.kanai.fu@hco.ntt.co.jp

Masanori Yamada  
NTT  
Tokyo, Japan  
masanori.yamada.cm@hco.ntt.co.jp

Yasutoshi Ida  
NTT  
Tokyo, Japan  
yasutoshi.ida@ieee.org

Abstract—We propose a method for improving adversarial robustness by addition of a new bounded function just before softmax. Several studies hypothesize that small logits (inputs of softmax) by logit regularization contributes to adversarial robustness of deep learning. Following this hypothesis, we analyze norms of logit vectors at the optimal point under the assumption of universal approximation and explore new methods for constraining logits by addition of a bounded function before softmax. We theoretically and empirically reveal that small logits by addition of a common activation function, e.g., hyperbolic tangent, do not improve robustness since input vectors of the function (pre-logit vectors) can have large norms. From the theoretical findings, we develop the new bounded function. The addition of our function contributes to adversarial robustness because it makes logit and pre-logit vectors have small norms. Since our method only adds one activation function before softmax, it is easy to combine our method with adversarial training. Our experiments demonstrate that our method is comparable to logit regularization methods in terms of robustness against untargeted attacks without adversarial training. Furthermore, it is superior or comparable to logit regularization methods and a recent defense method (TRADES) when using adversarial training.

I. INTRODUCTION

Deep neural networks (DNNs) are used in many applications, e.g., image recognition [14], and have achieved great success. Although DNNs can handle data accurately, they are vulnerable to adversarial examples, which are imperceptibly perturbed data to make DNNs misclassify data [27]. To investigate vulnerabilities of DNNs and improve adversarial robustness, many adversarial attack and defense methods have been presented and evaluated [4], [17], [19], [22].

Among defense methods, adversarial training is regarded as a promising method [17], [19]. Adversarial training generates adversarial examples of training data and trains DNNs on these adversarial examples. On the other hand, since the generation of adversarial examples requires high computation cost, several studies introduced defense methods not using adversarial examples [15], [21], [30]. [30] showed that label smoothing can be used as the efficient defense methods. Similarly, [15] presented logit squeezing that imposes a penalty of norms of logit vectors, which are input vectors of softmax. Since recent studies have shown that label smoothing also induces small logits, they regard label smoothing and logit squeezing as logit regularization methods and experimentally investigate the relation between robustness and logit norms [20], [23–25]. [23] showed that adversarial training actually reduces the norms of logits along with the increase in the strength of the attacks in training. According to these studies, constraining of logit norms to be small values is one approach to improve adversarial robustness.

In this paper, we propose a method of constraining the logit norms that uses a bounded activation function just before softmax on the basis of our theoretical findings about logit regularization. To understand the effect of logit regularization, we analyze the optimal logits for training of a neural network that has universal approximation. In this analysis, we first prove that (i) norms of the optimal logit vectors of cross-entropy are infinitely large values, and (ii) logit squeezing and label smoothing make norms of the optimal logit vectors be finite values. Infinitely large logits mean that the function from data points to logits does not have finite Lipschitz constants, which are constrained for adversarial robustness [5], [11], [28]. Next, to verify the effect of small logit norms, we evaluate robustness of models the logit vectors of which have various norms. From the experiments, we confirm that models can be robust against untargeted adversarial attacks if norms of logits are below a certain value. This observation suggests that the addition of a bounded function just before softmax contributes to robustness. However, we also reveal that adding common bounded activation functions, e.g., hyperbolic tangent, does not improve adversarial robustness. This is because these functions are monotonically increasing, and the optimal inputs (hereinafter, we call pre-logits) of these functions become infinitely large values; i.e., models from data points to pre-logits do not have finite Lipschitz constants. To overcome this drawback, we develop a new function called bounded logit function (BLF). BLF is bounded by finite values, and its pre-logits at the maximum and minimum points are also finite...
values. As a result, the addition of BLF just before softmax makes the optimal logits and pre-logits have finite values. Since our method only adds one activation function before softmax, it is easy to combine BLF with adversarial training. We confirmed that BLF outperforms logit squeezing without adversarial training, and, when using adversarial training, BLF is superior or comparable to logit squeezing, label smoothing, and TRADES, which is a recent strong defense method [32], in classification experiments on MNIST and CIFAR10 perturbed by the untargeted gradient-based attack (PGD [17], [19]) and gradient-free attacks (SPSA [29] and Square Attacks [1]).

II. PRELIMINARIES

A. Softmax Cross-entropy and Logit Regularization

To classify the i-th data point \( x^{(i)} \in \mathcal{X} \) where \( \mathcal{X} \) is a data space, neural networks learn a probability distribution over \( M \) classes conditioned on the data point \( x \) as \( P_\theta(k|x^{(i)}) \) for \( k = 1, \ldots, M \) where \( \theta \) is a parameter vector. Let \( z_\theta(x^{(i)}) = [z_{\theta,1}(x^{(i)}), z_{\theta,2}(x^{(i)}), \ldots, z_{\theta,M}(x^{(i)})]^T \) be a logit vector composed of a logit \( z_{\theta,k}(x^{(i)}) \), which is an input vector of softmax \( f_s(\cdot) \). The k-th element of softmax \( [f_s(\cdot)]_k \) represents the conditional probability of the k-th class as

\[
P_\theta(k|x^{(i)}) = [f_s(z_\theta(x^{(i)})))]_k = \frac{\exp(z_{\theta,k}(x^{(i)}))}{\sum_{m=1}^{M} \exp(z_{\theta,m}(x^{(i)))),}
\]

(1)

To train the model \( f_s(z_\theta(x)) \), we minimize cross-entropy loss \( \mathcal{L}_{CE} \). The objective function \( J \) is

\[
\mathcal{L}_{CE}(z_\theta(x^{(i)}), p^{(i)}) = -\sum_{k=1}^{M} p_k^{(i)} \log[f_s(z_\theta(x^{(i)}))]_k,
\]

(2)

where \( p^{(i)} \in \mathbb{R}^M \) is a target vector for \( x^{(i)} \). Target vectors are generally one-hot vectors as \( p_k = 1 \) for \( k = t \) and \( p_k = 0 \) for \( k \neq t \).

III. ANALYSIS OF LOGIT REGULARIZATION METHODS FOR ADVERSARIAL ROBUSTNESS

We first investigate logits obtained by softmax cross-entropy and logit regularization methods. Next, we experimentally show the relation between logit norms and robustness.

A. Optimal Logs for Minimization of Training Loss

To clarify the background of our results, we first show the assumption of our analysis inspired by the universal approximation properties of neural networks.

**Assumption.** We assume that (a) if data points have the same values as \( x^{(i)} = x^{(j)} \), they have the same labels as \( p^{(i)} = p^{(j)} \), (b) the logit vector \( z_\theta(x) \) can be an arbitrary vector for each data point, and (c) the optimal point \( \theta^* = \text{argmin} \sum_{i=1}^{N} \mathcal{L}(z_\theta(x^{(i)}), p^{(i)}) \) achieves \( \mathcal{L}(z_\theta(x^{(i)}), p^{(i)}) = \text{min}_{\theta} \mathcal{L}(z_\theta(x^{(i)}), p^{(i)}) \) for all \( i \).

This assumption ignores generalization performance and takes into account deterministic labels. This assumption is satisfied if we use the models that can be arbitrary functions and minimize cross-entropy on the dataset where the same data points have the same labels. Though it is a strong assumption, our analysis is valuable for understanding the behavior of the

\[\text{Since several papers have shown that over-parameterized network with least squares loss can achieve zero training loss [8], [9], it might not be a very strong assumption.}\]
logits since DNNs have large representation capacity and we assign a label for each data point with no duplication.

From the assumption, we can regard the optimal logits $z_\theta^*(x^{(i)}) = \arg\min_{z_\theta} \mathcal{L}_{CE}(z_\theta(x^{(i)}), p^{(i)})$ for the $i$-th data point as the logits obtained by minimization of the training objective functions. Next, we show the property of the optimal logits for softmax cross-entropy.

**Theorem 1.** If we use softmax cross-entropy and one-hot vectors as target values, at least one element of the optimal logits $z_\theta^*(x^{(i)}) = \arg\min_{z_\theta} \mathcal{L}_{CE}(z_\theta(x^{(i)}), p^{(i)})$ does not have a finite value.

This theorem indicates that softmax cross-entropy enlarges logit norms. Since this result has been mentioned in several studies [26], [30], we omit the proof. From this theorem, we can derive the following corollary:

**Corollary 1.** If all elements of inputs $x^{(i)}_k$ are normalized as $0 \leq x^{(i)}_k \leq 1$ and training dataset has at least two different labels, the optimal logit function $z_\theta^*(x)$ for softmax cross-entropy is not globally Lipschitz continuous function, i.e., there is not a finite constant $C \geq 0$ as

$$\|z_\theta^*(x^{(i)}) - z_\theta^*(x^{(j)})\|_\infty \leq C\|x^{(i)} - x^{(j)}\|_\infty, \quad \forall x^{(i)}, x^{(j)} \in \mathcal{X}. \quad (4)$$

**Proof.** If Corollary 1 does not hold, there is a finite constant $C$ satisfying

$$\|z_\theta^*(x^{(i)}) - z_\theta^*(x^{(j)})\|_\infty \leq C\|x^{(i)} - x^{(j)}\|_\infty \quad (5)$$

for $\forall x^{(i)}, x^{(j)} \in \mathcal{X}$ and $0 \leq C < \infty$. Let $t$ and $t'$ be labels of $x^{(i)}$ and $x^{(j)}$ $(t \neq t')$, respectively. From Theorem 1, $z_{\theta,m}(x^{(i)})$ for $m \neq t$ does not have finite values and $-\infty < z_{\theta,t}(x^{(i)}) \leq \infty$. On the other hand, $z_{\theta,m}(x^{(j)})$ for $m \neq t'$ does not have finite values. Thus, $|z_{\theta,m}(x^{(i)})| = z_{\theta,t}(x^{(i)}) - z_{\theta,t}(x^{(j)})$ does not have finite values. Therefore, the left-hand side of (5) is not finite values. On the other hand, we have $\|x^{(i)} - x^{(j)}\|_\infty \leq 1$ because we assume $0 \leq x^{(i)}_k \leq 1$. As a result, $\infty \leq C\|x^{(i)} - x^{(j)}\|_\infty \leq C$, and it contradicts the statement $0 \leq C < \infty$, which completes the proof.

This corollary indicates that the optimal logit function for softmax cross-entropy does not have a Lipschitz constant; logit vectors can be drastically changed by small perturbations. This fact does not immediately mean that the models are vulnerable since the logit gaps between the correct label and other labels on given data are also infinite in this case. Even so, since adversarial examples are outside of the training data, it is difficult to expect the outputs for adversarial examples. In fact, constraints of Lipschitz constants are used for improving adversarial robustness [5], [11], [28]. This corollary also indicates that finite logit values are necessary conditions for a Lipschitz constant. Thus, robust models have finite logit values, which is in agreement with the empirical observation that adversarial training reduces the norms of logits along with robustness [23].

Next, we consider the optimal logits for logit regularization methods. In the same way as Theorem 1, we can show the following propositions:

**Proposition 1.** The optimal logits $z_\theta^*(x^{(i)}) = \arg\min_{z_\theta} \mathcal{L}_{CE}(z_\theta(x^{(i)}), p^{(i)})$ satisfy $z_{\theta,t,k}(x^{(i)}) = \begin{cases} \log(\frac{\sum_{m \neq t} \exp((\theta_{\cdot,m}(x^{(i)})))}{\sum_{m \neq t} \exp((\theta_{\cdot,m}(x^{(i))}))}) & k = t \\ \log(\frac{\sum_{m \neq k} \exp((\theta_{\cdot,m}(x^{(i))}))}{\sum_{m \neq k} \exp((\theta_{\cdot,m}(x^{(i))})}) & k \neq t. \end{cases}$

If an element of $\exp(z_\theta^*(x^{(i)}))$ has a finite value, all elements of $z_\theta^*(x^{(i)})$ have finite values.

**Proof.** The objective function for $x^{(i)}$ is $J = -\sum_m p_m \log f_s(z_\theta(x^{(i)}))_m = -(1-\alpha) \log f_s(z_\theta(x^{(i)}))_t - \frac{\alpha}{M-1} \sum_m \sum_{m \neq t} \log f_s(z_\theta(x^{(i)}))_m$ since $p_t = 1 - \alpha$ and $p_m = \frac{\alpha}{M-1}$ for $m \neq t$. By differentiating $J$, we obtain

$$\frac{\partial J}{\partial z_{\theta,k}} = \begin{cases} \left[ f_s(z_\theta(x^{(i)}))_k \right] & k = t \\ \left[ f_s(z_\theta(x^{(i)}))_k \right] - \frac{\alpha}{M-1} & \text{otherwise}. \end{cases} \quad (6)$$

Since we assume that one of the elements of $z_\theta^*$ has a finite value, $\sum_m \exp z_{\theta,m} > 0$. Thus, (6) for $k = t$ becomes

$$|f_s(z_\theta(x^{(i)}))_t = \frac{\exp z_{\theta,t}}{\sum_m \exp z_{\theta,m}} = 1 - \alpha, \quad (7)$$

and thus, we have

$$z_{\theta,t} = \log(\frac{1-\alpha}{\sum_m \exp z_{\theta,m}}). \quad (8)$$

In the same manner, we have

$$|f_s(z_\theta(x^{(i)}))_k = \frac{\alpha}{M-1}, \quad (9)$$

$$z_{\theta,k} = \log(\frac{\alpha}{M-1 - \alpha} \sum_m \exp z_{\theta,m}(x^{(i)})), \quad (10)$$

for $k \neq t$. It is difficult to solve (8) and (10) in closed form, but they have finite values as follows: If $z_{\theta,k}(x^{(i)}) \rightarrow -\infty$ where $k' \neq t$, (9) does not hold since the left side of (9) becomes $0$ and $0 < \alpha < M - 1$. If $z_{\theta,t}(x^{(i)}) \rightarrow \infty$, (7) does not hold since the left side of (7) becomes $1$. Thus, all elements of the logits have finite values.

**Proposition 2.** The optimal logits for logit squeezing $z_{\theta^*}(x^{(i)}) = \arg\min_{z_\theta} \mathcal{L}_{CE}(z_\theta(x^{(i)}), p^{(i)} + \frac{\lambda}{2} \|z_\theta(x^{(i)})\|_2)$ satisfy

$$z_{\theta,k}(x^{(i)}) = \begin{cases} -\frac{1}{\lambda} f_s(z_\theta^*(x^{(i)}))_k + 1/\lambda & k = t \\ -\frac{1}{\lambda} f_s(z_\theta^*(x^{(i)}))_k + \lambda z_{\theta,k} & k \neq t \end{cases} \quad (11)$$

Namely, all elements of the optimal logit vector $z_\theta^*(x^{(i)})$ have finite values.

**Proof.** The objective function of logit squeezing for $x^{(i)}$ is $J = -\log |f_s(z_\theta(x^{(i)}))|_t + \frac{\lambda}{2} \|z_\theta(x^{(i)})\|_2$. Since we assume $z_\theta(x^{(i)})$ can be an arbitrary vector, $\frac{\partial J}{\partial z_{\theta,t}} = 0$ at the minimum points. By differentiating $J$, we obtain

$$\frac{\partial J}{\partial z_{\theta,k}} = \begin{cases} \frac{1}{f_s(z_{\theta,t}(x^{(i)}))} & k = t \\ \frac{1}{f_s(z_{\theta,t}(x^{(i)}))} + \frac{\lambda}{f_s(z_{\theta,t}(x^{(i)))}} & k \neq t \end{cases} \quad (12)$$

Thus, we have $\frac{1}{f_s(z_{\theta,t}(x^{(i)}))} + \frac{\lambda}{f_s(z_{\theta,t}(x^{(i)))}} = 0$. Therefore, the following equation holds: $z_{\theta,t} = \frac{1}{\lambda} f_s(z_{\theta,t}(x^{(i)}))$. Since each element of softmax functions is bounded as $0 \leq f_s(z_{\theta,t}(x^{(i)})) \leq 1$, all elements of $z_{\theta^*}(x^{(i)})$ are finite values.
\[ f_k(\gamma g(\mathbf{z}_\theta(x^{(i)}))) | k \leq 1, \text{ we have } 0 \leq z_{\theta^*,k} \leq 1. \] In the same manner, we have \( -\frac{1}{\gamma} \leq z_{\theta^*,k} \leq 0 \) for \( k \neq t \). Therefore, all elements of the optimal logits have finite values.

These propositions indicate that logit regularization methods enable the optimal logit values to have finite values; i.e., these methods satisfy the necessary condition of Lipschitz continuous. This property might improve robustness since the logit functions do not tend to change drastically by small perturbation on input data points. If logit regularization methods induce small Lipschitz constants, the models can be robust against adversarial examples.

From the hypothesis that small logits can improve adversarial robustness, we consider approaches to bound logits other than logit regularization methods. As an alternative to logit regularization, we can constrain logits by addition of a bounded activation function just before softmax. As such functions, hyperbolic tangent (tanh) and sigmoid functions are common functions in neural networks. If we use these monotonically increasing functions just before softmax, we have the following theorem:

**Theorem 2.** Let \( g(z) \) be tanh or sigmoid function and \( \gamma \) be a hyper-parameter satisfying \( 0 < \gamma < \infty \). If we use \( g(z) \) before softmax as \( f_k(\gamma g(\mathbf{z}_\theta(x^{(i)}))) \), all elements of the optimal pre-logit vector \( z_{\theta^*}(x^{(i)}) = \text{argmin}_{\mathbf{z}_\theta} L_{\text{CE}}(\gamma g(\mathbf{z}_\theta(x^{(i)})), \mathbf{p}^{(i)}) \) do not have finite values while all elements of the optimal logit vector \( \gamma g(\mathbf{z}_{\theta^*}) \) have finite values.

**Proof.** First, we show the case using tanh. The objective function for \( x^{(i)} \) is \( J = L_{\text{CE}}(\gamma g(\mathbf{z}_\theta(x^{(i)})), \mathbf{p}^{(i)}) = -\log[f_k(\gamma \tanh(\mathbf{z}_\theta(x^{(i)})))]. \) Since we assume that \( z_{\theta}(x^{(i)}) \) can be an arbitrary vector, \( \frac{\partial J}{\partial \mathbf{z}_\theta} = 0 \) at the minimum point \( z_{\theta^*}(x^{(i)}) \). Since tanh is an element-wise function, \( \frac{\partial J}{\partial \mathbf{z}_\theta} \) is written as \( \frac{\partial J}{\partial \mathbf{z}_\theta} = \gamma \frac{\partial}{\partial \mathbf{z}_\theta} \tanh(z_{\theta,k}) \frac{\partial z_{\theta,k}}{\partial z_{\theta,k}} \). \( \frac{\partial}{\partial \mathbf{z}_\theta} \tanh(z_{\theta,k}) \) does not become 0 since \(-1 \leq \tanh(z_{\theta,k}) \leq 1 \) and \( 0 < \gamma < \infty \). Thus, \( \frac{\partial}{\partial \mathbf{z}_\theta} \tanh(z_{\theta,k}) = 0 \) corresponds to \( \frac{\partial}{\partial z_{\theta,k}} \tanh(z_{\theta,k}) = 0 \). We have \( \frac{\partial}{\partial z_{\theta,k}} \tanh(z_{\theta,k}) = 1 - \tanh^2(z_{\theta,k}). \) Only if \( z_{\theta^*,k}(x^{(i)}) \rightarrow \pm \infty \), the following equation holds: \( \frac{\partial}{\partial z_{\theta,k}} \tanh(z_{\theta,k}) = 0 \). Therefore, all elements of the optimal input vector \( \mathbf{z}_\theta \) do not have finite values. The case using sigmoid function \( \sigma(z) \) can be shown in the same manner. Since the derivative of sigmoid becomes \( \frac{\partial}{\partial z_{\theta,k}} \sigma(z_{\theta,k})(1 - \sigma(z_{\theta,k})) \), the equation \( \frac{\partial}{\partial z_{\theta,k}} = 0 \) holds when \( z_{\theta^*,k}(x^{(i)}) \rightarrow \pm \infty \).

This theorem indicates that though optimal logits become finite values by addition of tanh or sigmoid, the pre-logit functions \( z_{\theta^*}(x) \) do not have finite Lipschitz constants. Therefore, the pre-logit can be changed by small perturbation. Thus, this theorem indicates that bounded logits might not be sufficient for adversarial robustness.

**B. Empirical Evaluation of Logit Regularization**

As shown above, logit regularization can induce the finite logit values, and tanh and sigmoid functions cannot keep pre-logits small. In this section, we empirically investigate the relation between logit norms and adversarial robustness. We evaluated the average norms of logits on clean data of CIFAR10 and accuracies on adversarial examples of CIFAR10 (untargeted PGD \( \epsilon = 4/255 \) and 100 iterations) for various logit regularization methods. Note that we normalized CIFAR10 such that their pixel values are in \([0,1]\). We used logit squeezing (LSQ), label smoothing (LSM), and bounded logits by tanh with various \( \alpha \), \( \lambda \), and \( \gamma \).

Fig 1 shows adversarial robustness against the average norms of logit vectors. Note that results of BLF in Fig. 1 are discussed in the next section. In this figure, each point corresponds to each hyper-parameter. For tanh, we also plot adversarial robustness against average norms of pre-logit vectors \( z \) as well as logit vectors \( \gamma g(z) \). In Fig. 1, models learned using logit regularization methods have various norms and robustness depending on \( \alpha \) and \( \lambda \), and the robust accuracies become almost zero when norms exceed about seven. The results of tanh indicate that even if models have small logits by adding bounded functions, they are vulnerable when their pre-logit norms are large. In addition, the results of tanh imply that just scaling logits \( \gamma z \) does not improve the robustness. From the observation, we need a new function that has bounded outputs and inputs.

**IV. PROPOSED METHOD**

We propose constraining logits by the addition of a new activation function called bounded logit function (BLF) just before softmax. BLF is defined as follows:

**Definition 1.** Bounded logit function (BLF) is defined as

\[
g(z) = 2 \{ z\sigma(z) + \sigma(z) - z\sigma^2(z) \} - 1, \tag{12}
\]

where \( \sigma \) is sigmoid function. When \( z \) is a vector, BLF becomes element-wise operation.

BLF is similar to tanh as shown in Fig. 2 and has the same properties of tanh, e.g. \( \lim_{z \rightarrow +\infty} g(z) = 1 \), \( \lim_{z \rightarrow -\infty} g(z) = -1 \), and \( \partial g(z)/\partial z|_{z=0} = z \). However, this function has the...
maximum and minimum points in $-\sqrt{5} - 1 < z < -2$ and $2 < z < \sqrt{5} + 1$ while tanh does not have the finite maximum and minimum points. Thus, we have the following theorem:

**Theorem 3.** Let $g(z)$ be BLF and $\gamma$ be a hyper-parameter satisfying $0 < \gamma < \infty$. If we use $g(z)$ before softmax as $f_z(\gamma g(z_0(x)))$, all elements of the optimal pre-logit vector $z_{\theta^*}(x^{(i)}) = \arg\min_x \mathcal{L}_{CE}(\gamma g(z_0(x^{(i)})), p^{(i)})$ have finite values, and all elements of the optimal logit vector $\gamma g(z_{\theta^*})$ also have finite values. Specifically, we have the following equalities and inequalities:

$$
\gamma g(z_{\theta^*,k}(x^{(i)})) = \begin{cases} 
\gamma \max_z g(z) & k = t \\
\gamma \min_z g(z) & \text{otherwise,}
\end{cases}
$$

$$
\gamma < |\gamma g(z_{\theta^*,k}(x^{(i)}))| < \frac{\gamma + 1}{\sqrt{2}},
$$

$$
z_{\theta^*,k}(x^{(i)}) = \begin{cases} 
\arg\max_{\theta} g(z) & k = t \\
\arg\min_{\theta} g(z) & \text{otherwise,}
\end{cases}
$$

$$
2 < |z_{\theta^*,k}(x^{(i)})| < \sqrt{5} + 1.
$$

**Proof.** The objective function for $x^{(i)}$ is $J = \mathcal{L}_{CE}(\gamma g(z_0(x^{(i)})), p^{(i)}) = -\log[f_z(\gamma g(z_0(x^{(i)})))]$. where $g(z) = 2 \left\{ \sigma(z) + \sigma(z) - \sigma^2(z) \right\} - 1$. Since we assume that $z_0(x^{(i)})$ can be an arbitrary vector, $\partial J / \partial z = 0$ at the minimum point $z_{\theta^*}(x^{(i)})$. Similarly to tanh, BLF $g(z)$ is an element-wise function and bounded by finite values. Therefore, $\gamma \partial g(z_{\theta^*,k})$ does not become 0. As a result, $\partial J / \partial z_{\theta^*,k} = 0$ holds when $\partial g(z_{\theta^*,k}) = 0$ for all $k$. We have $\partial J / \partial z_{\theta^*,k} = 2\sigma(z_{\theta^*,k})(1 - \sigma(z_{\theta^*,k}))(2 + z_{\theta^*,k} - 2\sigma z_{\theta^*,k})$, thus, candidates of the optimal points satisfy one of the following conditions: (a) $\sigma(z) = 0$, (b) $\sigma(z) = 1$, and (c) $2 + z - 2\sigma(z) = 0$. First, inputs satisfying (a) and (b) correspond to $z \rightarrow \pm \infty$. Their outputs $g(z)$ become $\lim_{z \rightarrow \infty} g(z) = -1$ and $\lim_{z \rightarrow -\infty} g(z) = 1$, respectively. Next, to investigate inputs satisfying (c), we define $f(z) = 2 + z - 2\sigma(z)$, which is a continuous function. This function can be written as $f(z) = -\frac{z - 1}{e^z + 1} = -\frac{z - 1}{e^z + 1}$ for $z < 0$. Since $\tanh(z) < z$ for $z > 0$ and $\tanh(z) > z$ for $z < 0$, we have $f(z) = 2 - z\tanh(z/2) > 2 - z^2$ for $z \neq 0$. By using this equation, we have $f(z) > 0$ for $-2 < z < 2$. In addition, since $\frac{1}{e^z - 1} < 1 - e^{-z}$ for $z > -1$, we have $f(z) = \frac{2(1+z^2-e^{-z})}{1+e^{-z}} < \frac{(-z+1)^2}{(1+e^{-z})(z+1)} < 0$ for $z > \sqrt{5} + 1$. On the other hand, since we have $e^z < \frac{1}{1-z}$ for $z < 1$, we have the following inequality: $f(z) = \frac{2(1+z^2-e^{-z})}{1+e^{-z}} < \frac{(-z+1)^2}{(1+e^{-z})(z+1)} < 0$ for $z < -\sqrt{5} - 1$. Therefore, the points satisfying $\partial g(z_{\theta^*,k}) = 0$ are included in $-\sqrt{5} - 1 < z < -2$ and $2 < z < \sqrt{5} + 1$ from intermediate value theorem. Since $|g(z)|$ where $z$ satisfying (c) ($-\sqrt{5} - 1 < z < -2$ and $2 < z < \sqrt{5} + 1$) are greater than $|g(z)| = 1$ where $z$ satisfying (a) and (b), $z$ satisfying (c) are minimum and maximum points of $g(z)$. Thus, the objective function $J$ achieves the smallest when we have $\gamma g(z_{\theta^*,t}(x^{(i)})) = \gamma \max_z g(z)$ and $\gamma g(z_{\theta^*,k}(x^{(i)})) = \gamma \min_z g(z)$ for $k \neq t$ and the optimal inputs are $z_{\theta^*,t}(x^{(i)}) = \arg\max_{\theta} g(z)$ and $z_{\theta^*,k}(x^{(i)}) = \arg\min_{\theta} g(z)$ for $k \neq t$. And thus, we have $\gamma < |\gamma g(z_{\theta^*,k}(x^{(i)}))| < \frac{\gamma + 1}{\sqrt{2}}$ and $2 < |z_{\theta^*,k}(x^{(i)})| < \sqrt{5} + 1$. □

Theorem 3 indicates that this function gives finite logits and pre-logits unlike tanh. Therefore, we can keep both norms of logit and pre-logit vectors small when we add BLF before softmax. The scale of logits is controlled by using the hyper-parameter $\gamma$. We conducted experiments using various $\gamma$ in the same manner as mentioned in the previous section, and evaluated adversarial robustness against norms of logits and pre-logits (Fig. 1). From this figure, BLF keeps the logits and pre-logits small, and its robust accuracy is higher than that of logit squeezing. Furthermore, since the optimal pre-logits $z_{\theta^*,j}(x^{(i)})$ do not depend on $\gamma$, pre-logits of BLF on some $\gamma$ can have almost the same norms; the $L_{\infty}$ norms are vertically aligned on about 2.4 despite the difference in $\gamma$. The result indicates that the empirical optimal pre-logits follow Theorem 3, though our theoretical results are based on the Assumption in Section III-A.

$\gamma$ can be used as a learnable parameter. However, the optimal $\gamma$ becomes infinitely large by minimization of softmax cross-entropy, and the logit norms become infinite values. Thus, the learnable $\gamma$ does not improve adversarial robustness. To confirm this, we also evaluated a learnable version of BLF (L-BLF) in the next section. In this setting, we used $\gamma = \text{softplus}(\hat{\gamma})$ and optimized $\hat{\gamma}$ to keep $\gamma$ non-negative.

Note that one of the reasons why we use BLF is that BLF is similar to tanh, so it is easy to verify the effect of the finite optimal points. We can use other bounded functions, which are not monotonically increasing, instead of BLF.

V. Experiments

We conducted experiments to evaluate the proposed method in terms of robustness against gradient-based attacks, robustness against gradient-free attacks, and operator norms of models. In the experiments, we evaluated the models using only clean training data (standard training) and using adversarially perturbed training data (adversarial training), respectively.

A. Experimental Conditions

Datasets of the experiments were MNIST [18] and CIFAR10 [16]. Our method was compared with a model trained without any logit regularization methods (Baseline), logit squeezing
(LSQ), and label smoothing (LSM). In our method, we evaluated BLF with fixed $\gamma$ (BLF) and BLF with learnable $\gamma$ (L-BLF). We also compared them with TRADES, which is a strong defense method using adversarial examples [32], in the adversarial training setting.

For MNIST, we used convolutional neural networks (CNNs) composed of two convolutional layers and two fully connected layers. We trained CNNs by using SGD with momentum of 0.5 and learning rate of 0.01 for 100 epochs and used weight decay of 0.01. The minibatch size was set to 64. For CIFAR10, we used ResNet-18 (RN18) [14] and WideResNet-34-10 (WRN) [31] also following [32]. We trained RN18 and WRN by using SGD with momentum of 0.9 for 350 epochs in the standard training setting and for 120 epochs in the adversarial training setting. In both settings, we used weight decay of 0.0005, and the initial learning rate was 0.1. After the 150-th and the 250-th epoch, we divided the learning rate by 10 in the standard training setting. In the adversarial training setting, we divided the learning rate by 10 after the 50-th and 100-th epoch for RN18, and after the 75-th and 100-th epoch for WRN. The minibatch size was set to 128.

We used untargeted projected gradient descent (PGD), which is the most popular white box attack, as a gradient-based attack and used SPSA and Square attacks as gradient-free attacks. The hyper-parameters for PGD were based on [19]. The $L_\infty$ norm of the perturbation $\epsilon$ was set to 0.3 for MNIST and $8/255$ for CIFAR10 at training time. For PGD, we randomly initialized the perturbation and updated it for 40 iterations with a step size of 0.01 on MNIST at training and evaluation times, and on CIFAR10 for 7 iterations with a step size of $2/255$ at training time and 100 iterations with the same step size at evaluation time. At evaluation time, we use $\epsilon \in \{0, 0.05, \ldots, 0.3\}$ on MNIST and $\epsilon \in \{0, 2/255, \ldots, 20/255\}$ on CIFAR10. $\epsilon = 0$ corresponds to clean data. For TRADES, we set hyper-parameters of adversarial examples based on the code provided by the authors.2 On MNIST, step size was 0.01, and the number of steps and $\epsilon$ were 40 and 0.3, respectively. On CIFAR10, step size was 2/255, and the number of steps and $\epsilon$ were 10 and 8/255, respectively. We selected the best hyper-parameters of our method $\gamma$, logit squeezing $\lambda$, label smoothing $\alpha$, and TRADES $\beta$ among five parameters. The selected hyper-parameters are shown in Fig. 3. For WRN, we used the same hyper-parameters as those of RN18. We trained models for five times for MNIST and three times for CIFAR10 and show the average and standard deviation of test accuracies. To generate adversarial examples, we used advertorch [7].

B. Robustness against Untargeted Gradient-based Methods

1) Accuracy against PGD: Figs 3(a) and (b) show accuracies on MNIST attacked by untargeted PGD. In the standard training setting (Fig. 3(a)), BLF is the most robust against PGD when $\epsilon > 0.15$. In adversarial setting (Fig. 3(b)), our proposed function improves robustness the most. Figs 3(c)-(f) show the results on CIFAR10. In the standard training setting, label smoothing improves robustness the most for RN18, and our proposed method improves robustness the most for WRN. In the adversarial training setting, our method improves the robustness the most. It is more robust than TRADES even though the number of iteration for TRADES is larger than that of our method. L-BLF fails to improve robustness. This is because $\gamma$ becomes large to minimize the loss function, and norms of logits have large values.

The reason label smoothing and logit squeezing are not so effective in the adversarial training might be due to the complexity of the objective function: regularization terms might disturb the mini-max problem for adversarial robustness. On the other hand, our method does not change the objective function. Thus, it is more suitable for adversarial training.

C. Robustness against Untargeted Gradient-free Attacks

Since [20] pointed out that logit regularization only masks or obfuscates gradient to improve robustness, we evaluated the robustness against gradient-free attacks. In the experiments, we tuned hyperparameters ($\lambda$, $\alpha$, $\gamma$, $\beta$) for each attack and train the model for one time for each hyper-parameter.

1) Accuracy against SPSA attacks: We used SPSA because it can operate when the loss surface is difficult to optimize [3], [29]. We set the hyper-parameters of SPSA as epsilons of 0.15 for MNIST and $8/255$ for CIFAR10, perturbation size of 0.01, Adam learning rate of 0.01, maximum iterations of 40 for

---

1https://github.com/yaodongyu/TRADES
are in agreement with the results of gradient-based attacks. Robustness the most in a majority of settings. These results and logit squeezing improve robustness more than BLF. In Fig. 3(a). On CIFAR10, label smoothing against PGD of BLF is lower than those of label smoothing SPSA. In the standard training setting, BLF improves robustness relatively.

high computation costs [24]. Even so, we could evaluate the robustness of our method relatively.

TABLE I

|               | Baseline | LSQ | LSM | BLF | TRADES |
|---------------|----------|-----|-----|-----|--------|
| MNIST ST      | 53.5     | 60.8| 67.4| 68.4| N/A    |
| MNIST AT      | 90.2     | 92.3| 90.0| 92.7| 86.4   |
| RN18 ST       | 0.3      | 23.5| 24.9| 23.4| N/A    |
| RN18 AT       | 56.6     | 56.8| 56.6| 58.7| 57.4   |
| WRN ST        | 0.1      | 42.1| 16.3| 13.9| N/A    |
| WRN AT        | 60.6     | 61.5| 60.0| 62.3| 61.3   |

TABLE II

|               | Baseline | LSQ | LSM | BLF | TRADES |
|---------------|----------|-----|-----|-----|--------|
| MNIST ST      | 51.5     | 53.0| 64.0| 59.4| N/A    |
| MNIST AT      | 91.4     | 90.3| 91.0| 92.1| 85.3   |
| RN18 ST       | 0.3      | 20.1| 27.6| 31.7| N/A    |
| RN18 AT       | 54.5     | 55.4| 55.6| 53.8| 55.3   |
| WRN ST        | 0.2      | 30.2| 38.4| 30.8| N/A    |
| WRN AT        | 59.7     | 59.7| 60.0| 60.9| 59.4   |

MNIST and 10 for CIFAR10, and batchsize of 2048.3 Table I lists the robust accuracies of each method against SPSA. In the standard training setting, BLF improves robustness against SPSA the most on MNIST even though robustness against PGD of BLF is lower than those of label smoothing and logit squeezing in Fig. 3(a). On CIFAR10, label smoothing and logit squeezing improve robustness more than BLF in adversarial training settings, however, our method improves robustness the most in a majority of settings. These results are in agreement with the results of gradient-based attacks.

2) Accuracy against Square Attacks: Though SPSA does not use exact gradients, it still approximates gradients. Thus, obfuscating gradients might be still effective for SPSA. To investigate whether logit regularization just obfuscates gradients, we evaluate the robustness against Square Attacks [1], which are query-based black box attacks. Since the Square Attacks use random search to generate attacks, obfuscating gradients are ineffective. To generate Square Attacks, we set the number of queries to 5000 and use the code in [6]. Note that Square Attacks use a margin loss instead of cross entropy loss.

Robust accuracies against Square Attacks are listed in Tab. II. In this table, BLF achieves the highest or the second highest accuracies on almost all settings. In addition, all logit regularization methods without adversarial training can improve the robust accuracies though Square Attacks do not use gradients. Therefore, logit constraints do not only just obfuscate gradients for improving robustness.

3We could not use the original hyper-parameters [29] since SPSA requires high computation costs [24]. Even so, we could evaluate the robustness of our method relatively.

D. Evaluation of Operator Norms

As discussed in Section III, cross-entropy can cause large Lipschitz constants, which might be a cause of vulnerabilities. To investigate Lipschitz constants of models, we computed averages of $L_\infty$ operator norms of convolution layers of RN18 (Tab. III) by following [12]. The $L_\infty$ operator norms of convolution layers can be a criterion of Lipschitz constants since one of Lipschitz constants of composite functions is the product of Lipschitz constants of composing functions and $L_\infty$ operator norm is a Lipschitz constant for a linear function. Table III shows that logit regularization methods induce small $L_\infty$ operator norms of convolution layers compared with Baseline even though they do not explicitly impose the penalty of parameter values. This table indicates that BLF can outperform other methods when using adversarial training because it effectively induces small Lipschitz constants. On the other hand, the $L_\infty$ norm of L-BLF is almost the same as that of Baseline. Thus, BLF with learnable $\gamma$ does not improve the robustness. Note that the $L_\infty$ norm of Baseline does not become extremely large because we applied some regularization methods, e.g., weight decay and early stopping, into all methods to obtain good generalization performance.

VI. LIMITATION OF LOGIT REGULARIZATION METHODS

[10] has shown that adversarial logit pairing, which is similar to logit regularization, is sensitive to targeted attacks, and [20] have shown that logit squeezing is sensitive to PGD attacks with multi-restart. In this section, we evaluate logit regularization methods with strong untargeted and targeted PGD attacks with multi-start. As strong PGD attacks, we used AutoPGD (APGD) with cross-entropy and targeted AutoPGD with difference of logits Ratio Loss (TAPGD) in [6]. APGD is more sophisticated than PGD; APGD uses momentum and adaptively selects the step size. We restarted APGD and TAPGD five times. In the experiments, we tuned hyperparameters ($\lambda$, $\alpha$, $\gamma$, $\beta$) for each attack and train the model for one time for each hyper-parameter.

Results are listed in Tab. IV. In this table, logit regularization methods are superior or comparable to Baseline. In particular, logit regularization methods outperform TRADES on MNIST. On CIFAR10, robust accuracies against TAPGD become zero when standard training. However, when using adversarial training, logit regularization methods can outperform Baseline; i.e., logit regularization contributes to general adversarial robustness. Robust accuracies against APGD of BLF become the highest on a majority of settings. Thus, BLF is effective in defending against untargeted strong at-
tacks while BLF might neither improve nor deteriorate the robustness against targeted attacks.

Indeed, logit regularization methods without adversarial training are not robust enough for the targeted attacks. However, they are still effective against practical threat models, e.g., untargeted attacks or black box attacks. Furthermore, when combining adversarial training with these methods, they can be comparable to strong defense methods.

VII. CONCLUSION

We proposed a method of constraining the logits by adding a new bounded function just before softmax following the hypothesis that small logits improve the adversarial robustness. The added function has finite maximum and minimum points so that logits and pre-logits have small values. Compared with other logit regularization methods, our method effectively improves robustness in adversarial training despite its simplicity. Though we provided insights into the vulnerabilities of softmax cross-entropy and empirical evidence of the effectiveness of logit regularization, it is still an open question why small logits can improve robustness. Even so, our experiments showed that our method is comparable to the recent defense method in terms of adversarial robustness against untargeted gradient-based and gradient-free attacks in adversarial training. Thus, our results indicate that the investigation into the relation between logit regularization and robustness is still an important research direction to reveal the cause of vulnerabilities.

REFERENCES

[1] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, “Square attack: a query-efficient black-box adversarial attack via random search,” in Proc. ECCV, 2020.

[2] A. Athalye, N. Carlini, and D. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in Proc. ICMIL, 2018, pp. 274–283.

[3] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, “On evaluating adversarial robustness,” arXiv:1902.06705, 2019.

[4] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, “Unlabeled data improves adversarial robustness,” in Proc. NeurIPS, 2019, pp. 11190–11201.

[5] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, “Parseval networks: Improving robustness to adversarial examples,” in Proc. ICMIL, 2017, pp. 854–863.

[6] F. Croce and M. Hein, “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,” in Proc. ICML, 2020. [Online]. Available: https://github.com/fra31/auto-attack.

[7] G. W. Ding, L. Wang, and X. Jin, “AdVerTorch v0.1: An adversarial robustness toolbox based on pytorch,” arXiv:1902.07622, 2019.

[8] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, “Gradient descent finds global minima of deep neural networks,” in Proc. ICMIL, 2019, pp. 1675–1685.

[9] S. S. Du, X. Zhai, B. Poczos, and A. Singh, “Gradient descent probably optimizes over-parameterized neural networks,” arXiv:1810.02054, 2018.

[10] L. Engstrom, A. Ilyas, and A. Athalye, “Evaluating and understanding the robustness of adversarial logit pairing,” arXiv:1807.10272, 2018.

[11] F. Farnia, J. Zhang, and D. Tse, “Generalizable adversarial training via spectral normalization,” in Proc. ICLR, 2019.

[12] H. Gouk, E. Frank, B. Pfahringer, and M. Cre, “Regularisation of neural networks by enforcing lipschitz continuity,” arXiv:1804.04368, 2018.

[13] S. Goyal, C. Qin, J. Uesato, T. Mann, and P. Kohli, “Uncovering the limits of adversarial training against norm-bounded adversarial examples,” arXiv:2010.03593, 2020.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. CVPR, 2016, pp. 770–778.

[15] H. Kannan, A. Kurakin, and I. Goodfellow, “Adversarial logit pairing,” arXiv:1803.06373, 2018.

[16] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Tech. Rep., 2009.

[17] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial machine learning at scale,” arXiv:1611.01236, 2016.

[18] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner et al., “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in Proc. ICLR, 2018.

[20] M. Mosbach, M. Andriushchenko, T. Trost, M. Hein, and D. Klakow, “Logit pairing methods can fool gradient-based attacks,” arXiv:1810.12042, 2018.

[21] T. Pang, K. Xu, Y. Dong, C. Du, N. Chen, and J. Zhu, “Rethinking softmax cross-entropy loss for adversarial robustness,” in Proc. ICLR, 2019.

[22] A. S. Ross and F. Doshi-Velez, “Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients,” in Proc. AAAI, 2018.

[23] A. Shafahi, A. Ghasi, F. Huang, and T. Goldstein, “Label smoothing and logit squeezing: A replacement for adversarial training?” arXiv:1910.11585, 2019.

[24] A. Shafahi, A. Ghasi, M. Najibi, F. Huang, J. Dickerson, and T. Goldstein, “Batch-wise logit-similarity: Generalizing logit-squeezing and label-smoothing,” in Proc. BMVC, 2019.

[25] C. Summers and M. J. Dinneen, “Improved adversarial robustness via logit regularization methods,” arXiv:1906.03749, 2019.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[27] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” arXiv:1312.6199, 2013.

[28] Y. Tszuzuki, I. Sato, and M. Sugiyama, “Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks,” in Proc. NeurIPS, 2018, pp. 6524–6531.

[29] J. Uesato, B. O’Donoghue, P. Kohli, and A. van den Oord, “Adversarial risk and the dangers of evaluating against weak attacks,” in Proc. ICMIL, vol. 80. PMLR, 2018, pp. 5025–5034.

[30] D. Warde-Farley and I. Goodfellow, “11 adversarial perturbations of deep neural networks,” Perturbations, Optimization, and Statistics, vol. 311, 2016.

[31] S. Zagoruyko and N. Komodakis, “Wide residual networks,” arXiv:1605.07146, 2016.