Boosting Adversarial Robustness From The Perspective of Effective Margin Regularization

Ziquan Liu
https://sites.google.com/view/ziquanliu
Antoni B. Chan
https://www.cs.cityu.edu.hk/~abchan/

Department of Computer Science
City University of Hong Kong
Hong Kong SAR, China

Abstract

The adversarial vulnerability of deep neural networks (DNNs) has been actively investigated in the past several years. This paper investigates the scale-variant property of cross-entropy loss, which is the most commonly used loss function in classification tasks, and its impact on the effective margin and adversarial robustness of deep neural networks. Since the loss function is not invariant to logit scaling, increasing the effective weight norm will make the loss approach zero and its gradient vanish while the effective margin is not adequately maximized. On typical DNNs, we demonstrate that, if not properly regularized, the standard training does not learn large effective margins and leads to adversarial vulnerability. To maximize the effective margins and learn a robust DNN, we propose to regularize the effective weight norm during training. Our empirical study on feedforward DNNs demonstrates that the proposed effective margin regularization (EMR) learns large effective margins and boosts the adversarial robustness in both standard and adversarial training. On large-scale models, we show that EMR outperforms basic adversarial training, TRADES and two regularization baselines with substantial improvement. Moreover, when combined with several strong adversarial defense methods (MART [48] and MAIL [26]), our EMR further boosts the robustness.

1 Introduction

One major challenge to the security of computer vision systems is that deep neural networks (DNNs) often fail to achieve a satisfactory performance under adversarial attacks [45]. Since the phenomenon is observed, various adversarial attacks [5, 9, 15] and defense methods [23, 33, 53] have been proposed and the understanding into the adversarial vulnerability of DNNs is improved [3, 20, 46]. Denote the DNN as $f_{\boldsymbol{\theta}} : \boldsymbol{x} \mapsto \boldsymbol{l}$, with $\boldsymbol{x} \in \mathbb{R}^D$ and $\boldsymbol{l} \in \mathbb{R}^K$. The model is optimized by algorithm $\mathcal{A}$ that minimizes empirical risk $\mathcal{L}$ over training set $\mathcal{D}_{tr}$,

$$\boldsymbol{\theta}^* = \mathcal{A}(f_{\boldsymbol{\theta}}, \mathcal{D}_{tr}, \mathcal{L}). \tag{1}$$

There are generally four *direct* methods to improve the robustness of DNNs. First, the function space $f_{\boldsymbol{\theta}}$ can be designed to accommodate the need for adversarial robustness. For

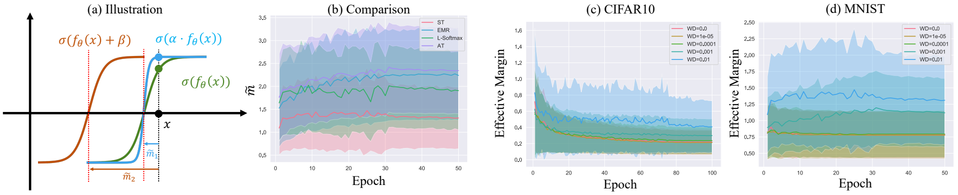

Figure 1: The problem of cross-entropy loss in maximizing effective margins and the proposed ERM's performance in terms of increasing the effective margins on MNIST test set. **(a)** The gradient of the sigmoid function $\sigma(\cdot)$ vanishes if we scale up the input logit $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ by $\alpha > 1$. However, its distance to the decision boundary (dashed red vertical line), i.e., the effective margin $\tilde{m}$ defined in (3) remains the same. This property shows that only training with cross-entropy loss does not effectively increase the actual margin. Our work aims to push the decision boundary away from the sample so that the $\tilde{m}$ is increased, e.g., $f_{\boldsymbol{\theta}}(\boldsymbol{x})+\beta$ with $\beta > 0$. **(b)** The means (solid lines) and standard deviations (background shadow) of the effective margins of an MLP on the MNIST test set. The proposed EMR achieves an $\tilde{m}$ and adversarial robustness that is comparable with adversarial training, and outperforms standard training with weight decay or L-Softmax. See Table 1 for details. **(c)** and **(d)** Effective margin on CIFAR10 and MNIST when different λ_{WD} are used. Training without weight decay (WD) or with small λ_{WD} leads to smaller effective margins.

example, replacing the piecewise linear activation with a smooth activation function improves the performance of adversarial training [50], and some architectural configurations are better than others in terms of adversarial robustness [19]. Second, the algorithm $\mathcal{A}$ can be incorporated with inductive biases to learn a function with some specific properties, such as low model complexity [22], local linearization [38] and feature alignment [25]. Third, the training set $\mathcal{D}_{tr}$ can be shifted by adversarial perturbation [33, 53] or other data augmentation [16, 39] to enhance the robustness. Finally, a carefully designed loss $\mathcal{L}$ can be used to improve the robustness, such as Max Mahalanobis center loss [36]. Besides the direct adversarial defenses, *indirect* adversarial defenses are also investigated, e.g., adversarial examples detection [30, 35, 42, 51] and obfuscated gradient defenses [2, 4, 12, 17, 32, 43, 44, 49].

This work falls into the second category (inductive bias) where the neural network is trained with regularization to boost the adversarial robustness. We consider the most popular loss function for the classification task, the cross-entropy (XE) loss,

$$XE_i = -\sum_{k=1}^{K} y_{ik} \log \frac{\exp(l_{ik})}{\sum_j \exp(l_{ij})}, \tag{2}$$

where the logit $l_{ik} = f_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{x}_i)$ is the k-th output of the neural network for the i-th sample. One property of the network is that the prediction for the i-th sample, i.e., $\hat{y}_i = \arg\max_{k \in [K]} l_{ik}$, is invariant to scaling the logit vector $\boldsymbol{l}_i = [l_{ik}]_k$ by a positive constant α. In other words, the classification accuracy will not change if we scale $\boldsymbol{l}_i$ up to $\alpha \boldsymbol{l}_i$ where $\alpha > 1$. However, the XE loss will vanish if we scale up the logit.

This phenomenon brings a problem in optimization with XE loss, since the training only aims to minimize the loss without maximizing the *effective* margin, which is defined as the normalized logit difference (see Equation 3) and is invariant to the weight magnitudes. Once a sample is correctly classified, the scale-variant property of XE loss can be exploited by

SGD to minimize the loss while the distance to the decision boundary (in the input space) remains small (see Fig. 1a). In a homogeneous NN [14, 29, 31], such as multi-layer perceptron (MLP) and convolutional neural network (CNN) without residual connections or normalization layers, the logit magnitude scales with weight norms so the training algorithm can minimize the loss by increasing weight norms. In a ResNet [18], the final classification layer can be scaled up to minimize the cross-entropy loss. Weight decay [22] is a common strategy to increase the effective margin by controlling the squared l_2 norm of weights in the DNN. However, it is known that adversarial robustness cannot be achieved by only using weight decay (WD), especially in deeper networks [45]. The most popular way to robustify DNNs is adversarial training (AT) [33], which *explicitly* perturbs samples to be on the desired margin from the original samples, and then trains on the perturbed samples. However, AT incurs increased computational cost for training due to the generation of the adversarial training samples in each iteration.

In this paper, we propose *effective margin regularization* (EMR) to push the decision boundary away from the samples by controlling the effective weight norms of the samples. We first show that traditional regularization such as weight decay and large-margin loss (e.g., [27]) cannot train a DNN with satisfactory robustness. Then the proposed method is compared with WD, large-margin softmax and adversarial training, where we show its strength at maximizing the effective margin and thus improving adversarial robustness. Finally, on large-scale DNNs, we propose an approximation to EMR and demonstrate that when combined with adversarial training, EMR achieves competitive results compared with basic adversarial training and two recent regularization methods for improving adversarial training, i.e., Input Gradient Regularization (IGR) [41] and Hypersphere Embedding (HE) [37]. Note that our EMR is complementary to adversarial training (AT) – EMR pushes the decision boundary away from the training samples so as to increase the effective margin, while AT generates training samples on the desired margin. Thus EMR and AT can be combined to further improve adversarial robustness.

2 Related Work

Adversarial Defense. The standard way to train an adversarially robust DNN is to use adversarial training [33]. The clean examples are deliberately perturbed to approach the desired margin distance, so that the effective margin is produced during training. Based on adversarial training, regularization approaches are proposed to learn a DNN with desired properties. [41] proposes to regularize the norm of the loss gradient with respect to input (IGR). In contrast, our work proposes to regularize the gradient of *logit* with respect to input to maximize the effective margin. Locally Linear Regularization (LLR) [38] is proposed to learn a more linear loss function at each training sample, while our paper controls the local logit function's weight norm for training samples. Hypersphere embedding (HE) [37] proposes to normalize the features and classification layer to alleviate the influence of weight norms. In our empirical study, we demonstrate that EMR achieves better robustness than IGR and HE on large-scale neural networks.

Margin Regularization. The hinge loss [7] is a classical loss to induce a large margin in SVM [7]. On DNNs, several losses are proposed to induce large margins, such as Large-Margin Softmax [27], A-Softmax [28] and AM-Softmax [47]. These large-margin losses still have problems to learn large effective margins since the scale of features and weights affects the loss values. On both MLP and CNN, we demonstrate that training with L-Softmax

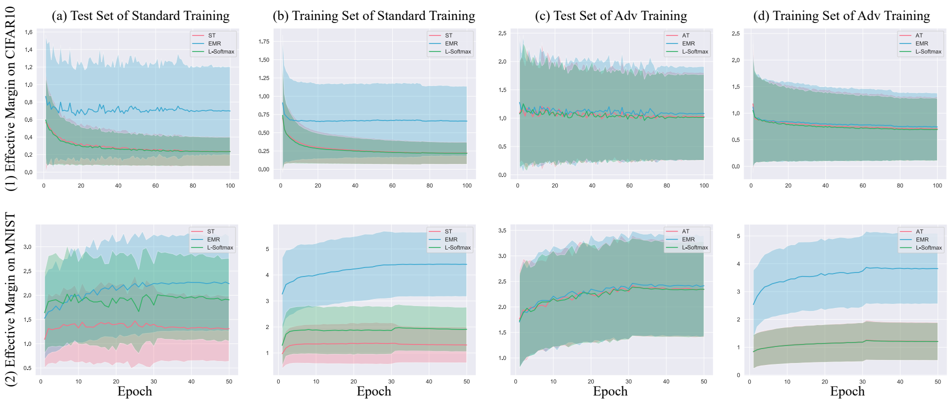

Figure 2: Effective margins on the training and test sets of CIFAR10 and MNIST. In normal training, EMR has a significant advantage over L-Softmax and weight decay (WD). In adversarial training, the improvement of EMR is still observable.

loss improves the effective margin compared with the standard cross-entropy loss, while EMR learns a larger effective margin than L-Softmax since EMR considers the scale problem in the loss function. [6, 31] study the normalized margin of homogeneous DNNs trained with gradient descent from a theoretical perspective and prove that the normalized margin is maximized by the gradient descent. Our work empirically investigates the normalized margin in DNNs trained with *stochastic* gradient descent and its influence on the adversarial robustness. We show that by controlling the effective weight norm and increasing the effective margin, the adversarial robustness can be improved over vanilla training with SGD and WD. The attack method DeepFool [34] moves an input sample to cross its decision boundary by treating the model as a linear classifier at each optimization step, which is related to the margin of a classifier. In contrast, our paper proposes to defend against adversarial attacks by increase the effective margin during training. We did not evaluate the DeepFool since it is not a standard attack method in adversarial defense literature [10] and our experiment shows that DeepFool is not as effective as PGD at attacking large-scale models (see supplemental). Max-Margin Adversarial training (MMA) [13] proposes to approximate the margin by pushing input samples to cross the classification boundary with PGD and recording the moved distance. In contrast, EMR proposed to maximize the effective margin by regularizing the effective weight matrix norm, and can boost the adversarial robustness of both standard training and adversarial training. Since the performance of MMA is worse than vanilla PGD and TRADES according to AutoAttack benchmark [9], we do not include the comparison with MMA in the experiment.

3 Regularizing Effective Weight Norm Improves Effective Margin and Adversarial Robustness

We use the notation for a DNN in Section 1. A general DNN, such as MLP, CNN and evaluation-mode Resnet with piece-wise linear activation functions (e.g., ReLU and LeakyReLU), can be expressed as a linear function for *each* input sample, i.e., $f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \boldsymbol{W}(\boldsymbol{x}_i)\boldsymbol{x}_i + \boldsymbol{b}(\boldsymbol{x}_i)$.

The weight matrix $\boldsymbol{W}_i \triangleq \boldsymbol{W}(\boldsymbol{x}_i)$ and bias $\boldsymbol{b}_i \triangleq \boldsymbol{b}(\boldsymbol{x}_i)$ are determined by input samples since the activations will change with the input. Similar to the normalized margin in [31], we define the *effective margin* $\tilde{m}_i$ as the normalized logit difference between the ground-truth class and the closest other class,

$$\tilde{m}_i = \min_{j \neq y_i} \frac{f_{\boldsymbol{\theta}}^{(y_i)}(\boldsymbol{x}_i) - f_{\boldsymbol{\theta}}^{(j)}(\boldsymbol{x}_i)}{\|\boldsymbol{w}_i^{(y_i)} - \boldsymbol{w}_i^{(j)}\|_2}, \tag{3}$$

where y_i is the ground-truth class for $\boldsymbol{x}_i$ and $\boldsymbol{w}_i^{(j)}$ is the jth row vector of $\boldsymbol{W}_i$. The quantity is relevant to adversarial robustness, since it describes the actual distance of a sample to the decision boundary in the input space. Thus, to improve adversarial robustness, it is desirable to maximize the effective margin, so that adversarial examples will fall inside the margin but still correctly classified. However, during training a DNN, once a training image is correctly classified by the network, the scale of $f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ can be increased to minimize the loss without effectively maximizing the actual margin. We aim to alleviate this problem by regularizing the effective weight norm during training.

For simplicity, we first consider a DNN without residual connections and assume that the bias term is appended as an additional dimension of the weight and $\boldsymbol{x}_i := [\boldsymbol{x}_i; 1]$; residual DNNs are considered in the experiment section. The logit output of the network is determined by the angle ϕ_{ij} between $\boldsymbol{x}_i$ and $\boldsymbol{w}_i^{(j)}$, and the lengths $\|\boldsymbol{x}_i\|$ and $\|\boldsymbol{w}_i^{(j)}\|$,

$$l_{ij} = \boldsymbol{x}_i^T \boldsymbol{w}_i^{(j)} = \|\boldsymbol{x}_i\| \|\boldsymbol{w}_i^{(j)}\| \cos(\phi_{ij}). \tag{4}$$

Training	λ_{WD}	Clean Acc.	PGD20	$\tilde{m}_{\text{train}}$	$\tilde{m}_{\text{test}}$
ST	0.1	77.88	34.06	1.00±0.69	1.02±0.70
ST	0.01	96.54	48.41	1.31±0.69	1.31±0.67
ST	0.001	98.41	24.41	1.11±0.51	1.12±0.53
ST	0.0001	98.33	4.69	0.79±0.34	0.79±0.36
ST+LSoftmax	0.1	17.83	17.27	3.13±1.11	3.21±1.16
ST+LSoftmax	0.01	98.00	60.59	1.91±0.86	1.91±0.85
ST+LSoftmax	0.001	**98.63**	51.07	1.82±0.63	1.81±0.66
ST+LSoftmax	0.0001	98.50	28.91	1.34±0.44	1.34±0.47
ST+EMR$_{0.1}$	0.001	97.50	**87.56**	4.41±1.24	2.24±0.98
AT	0.01	97.66	90.11	1.18±0.69	2.27±1.07
AT	0.001	98.68	92.62	1.21±0.69	2.34±0.94
AT	0.0001	**98.98**	92.24	1.09±0.61	2.13±0.77
AT+LSoftmax	0.01	97.57	89.88	1.18±0.69	2.28±1.07
AT+LSoftmax	0.001	98.72	92.60	1.21±0.68	2.35±0.94
AT+LSoftmax	0.0001	99.02	92.50	1.10±0.63	2.12±0.78
AT+EMR$_{3e-4}$	0.001	98.68	**92.78**	3.83±1.27	2.42±0.99

Table 1: Adversarial robustness and effective margins of MLP on MNIST for standard training (ST) and adversarial training (AT). PGD20 attack has an l_∞ bound of $\varepsilon = 0.1$ and the step size α is 0.01. The mean and standard deviation of effective margins defined in Equation 3 of training ($\tilde{m}_{\text{train}}$) and test ($\tilde{m}_{\text{test}}$) sets are shown.

The range of input magnitudes $\|\boldsymbol{x}_i\|$ is often fixed in the training/test stages, e.g., normalizing pixel values to $[0.0, 1.0]$. Thus, in order to minimize the loss, the training aims to increase l_{iy} and decrease $l_{ij}, \forall j \neq y$, by updating $\|\boldsymbol{w}_i^{(j)}\|$ and $\cos(\phi_{ij})$. As we want to learn a large effective margin, it is beneficial to constrain the weight norm $\|\boldsymbol{w}_i^{(j)}\|$ during training and let the optimization focus on the angular distance. Weight decay is a common method to control the weight norm of individual layers, while other works have proposed large margin losses [27]. In contrast, here we propose effective margin regularization (EMR) to directly penalize the *effective* weight norm for training samples, i.e.,

$$\mathcal{L} = \frac{1}{B} \sum_i^B XE(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i) + \lambda_{EMR} \frac{1}{B} \sum_i^B \sum_j^K \|\boldsymbol{w}_i^{(j)}\|^2. \tag{5}$$

Different from WD, which regularizes all parameters of a DNN, our EMR regularizes the *local* weight norms of the training samples. In our implementation, to compute the effective weight matrix for each output dimension, we loop over the object categories and take the gradient of the sum of the j-th logit over a batch, i.e., $\sum_i l_{ij}$, with respect to the input batch, and obtain $\boldsymbol{w}_i^{(j)}, \forall i$ as a result of computational independence of samples. The λ_{WD} is the coefficient for the WD regularization, where the squared L2 norm of all parameters is added to the loss. In contrast, we regularize a weighted sum of L2 norm of the logit's gradient (effective weight norm) as in Equation 5.

Empirical Results for Standard Training. We test the performance of EMR on two common feedforward neural networks, MLP and CNN. The MLP consists of 4 hidden layers and one output layer with the hidden dimensions as 1,024. The model is trained using SGD+Momentum on MNIST [24] for 50 epochs and with a batch size of 100. The initial learning rate is 0.01 and we divide it by 10.0 at the 30th epoch. The CNN consists of 4 convolution layers and the detailed architecture is in supplemental. The model is trained using SGD+Momentum on CIFAR10 [21] for 100 epochs and with a batch size of 100. The initial learning rate is 0.01 and we divide it by 10.0 at 75th and 90th epochs. We compare standard WD, L-Softmax and EMR on the CNN and MLP. On MLP, the margin parameter of L-Softmax loss is set as 4, while on CNN, the margin parameter is 1, otherwise the training will fail to converge. To evaluate the robustness, PGD attack [33] with l_∞ norm bound is used. For CNN, we use PGD10 with step size $\alpha = 0.0078$ and norm bound $\varepsilon = 0.031$. For MLP, we use PGD20 with step size $\alpha = 0.01$ and norm bound $\varepsilon = 0.1$.

Training	λ_{WD}	Clean Acc.	PGD10	$\tilde{m}_{\text{train}}$	$\tilde{m}_{\text{test}}$
ST	0.01	55.77	4.19	0.39±0.31	0.41±0.32
ST	0.001	82.10	0.68	0.28±0.19	0.30±0.21
ST	0.0001	**83.00**	0.50	0.22±0.15	0.23±0.17
ST+LSoftmax	0.01	55.06	4.90	0.41±0.33	0.43±0.34
ST+LSoftmax	0.001	82.39	0.81	0.27±0.18	0.28±0.19
ST+LSoftmax	0.0001	82.73	0.29	0.22±0.15	0.23±0.16
ST+EMR$_{0.01}$	0.0005	69.78	**16.37**	0.66±0.48	0.70±0.50
AT	0.01	34.73	22.98	0.90±0.82	1.22±1.04
AT	0.001	60.27	32.47	0.76±0.65	1.09±0.84
AT	0.0001	**63.59**	32.58	0.71±0.60	1.03±0.78
AT+LSoftmax	0.01	34.62	22.87	0.91±0.82	1.21±1.02
AT+LSoftmax	0.001	60.47	32.29	0.76±0.65	1.08±0.84
AT+LSoftmax	0.0001	63.09	32.96	0.71±0.61	1.03±0.78
AT+EMR$_{0.001}$	0.0005	62.79	**33.41**	0.74±0.64	1.08±0.83

Table 2: Adversarial robustness and effective margins of CNNs on CIFAR10. PGD10 attack has an l_∞ bound of $\varepsilon = 0.031$ and the step size α is 0.0078.

In Fig. 1c-1d, we first plot the effective margin of test samples during model training when standard WD is used with different hyperparameters λ_{WD}. Note that we only plot $\tilde{m}_i$ of correctly classified images since those images are the targets of adversarial attacks. The effective margin when WD is not used is clearly smaller than imposing a large weight decay, which validates our argument that penalizing large weight norms helps increase the effective margin in XE loss optimization. Fig. 2a-b show the effective margin of training and test samples for standard training (ST). L-Softmax has a benefit on the effective margins for MLP, but not CNN. In contrast, on both architectures, EMR achieves the highest effective margin. We show the clean and robust test accuracy curves during CNN and MLP training in the supplemental.

Table 1 (top) shows the evaluation results of MLP when using standard training with three methods. Two key observations are that: a) increasing λ_{WD} cannot achieve adversarial robustness that is comparable with L-Softmax or EMR, indicating that more advanced approaches are needed to maximize the effective margin; b) the adversarial robustness of training with EMR is substantially higher than training with L-Softmax, demonstrating the importance of regularizing effective weight norms. Table 2 (top) shows the robust accuracy of CNN. L-Softmax does not have a benefit over ST in this case, while EMR still substantially improves the robustness. Note that EMR still needs WD to achieve a satisfactory performance, since the weight decay handles the overall model complexity, while EMR constrains the local weight norms.

Empirical Results for Adversarial Training. The experiment with standard training on clean examples suggests that EMR may also have a benefit when adversarial training (AT) is used. Therefore, we compare WD, EMR and L-Softmax using AT [33]. In AT, the XE loss has adversarially perturbed input,

$$\mathcal{L} = XE(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\delta}_i), y_i), \quad \|\boldsymbol{\delta}_i\|_\infty \leq \varepsilon. \tag{6}$$

The perturbation is often searched by the PGD attack [33]. Here the same PGD used in the robustness evaluation is adopted in adversarial training for both CNN and MLP. Fig. 2c-d show the effective margin during AT. The advantage of EMR is decreased when AT is used, but we still observe an increase of effective margin compared with the two baselines. Tables 1 and 2 (bottom) also show the robustness evaluation for AT. Although the performance gap between EMR and the baseline is decreased compared to ST, we still observe improved robustness for both models using EMR. See the supplemental for the clean and robust test accuracy curve during AT of CNN and MLP.

4 Effective Margin Regularization for Large-Scale DNNs

The success of EMR on CNN and MLP provides a motivation to adopt EMR for training large-scale DNNs. One major issue of applying EMR to large-scale models is that the computation of effective weight matrices needs a loop over image categories, which is not scalable to large DNNs with many classes. Thus, we propose an approximation of EMR that does not need a loop, which is more amenable to large-scale DNNs. Define $\boldsymbol{l}_i$ as the logit vector for the ith sample, and $h_i = \sum_j p_{ij} l_{ij}(\boldsymbol{x}_i)$ as the weighted logit mean, where $\sum_j p_{ij} = 1$ is a constant weight vector in a $(K-1)$-dim simplex. The gradient of h_i with respect to $\boldsymbol{x}_i$ is $\nabla_{\boldsymbol{x}} h_i = \sum_j p_{ij} \boldsymbol{w}_i^{(j)}$, and its squared l_2 norm is

$$\hat{\mathcal{L}}_{EMR}(\boldsymbol{x}_i) = \|\nabla_{\boldsymbol{x}} \sum_j p_{ij} l_{ij}(\boldsymbol{x}_i)\|_2^2 = \sum_j \sum_k p_{ij} p_{ik} \langle \boldsymbol{w}_i^{(j)}, \boldsymbol{w}_i^{(k)} \rangle. \tag{7}$$

We take this quantity as an approximation to EMR in (5), which implicitly computes the effective weight matrix via the gradient of the logits. The original EMR regularizes the self-product term in the summation of (7), i.e., $\mathcal{L}_{EMR}(\boldsymbol{x}_i) = \sum_j \langle \boldsymbol{w}_i^{(j)}, \boldsymbol{w}_i^{(j)} \rangle$. Thus the difference between $\hat{\mathcal{L}}_{EMR}$ and $\mathcal{L}_{EMR}$ is the cross product term between $\boldsymbol{w}_i^{(j)}$ and $\boldsymbol{w}_i^{(k)}$. Minimizing the cross product is helpful for the classification task because it decreases the cosine similarity between different categories' weights.

In (7), p_{ij} will control the weight for the summation in $\hat{\mathcal{L}}_{EMR}$, and intuitively higher weights should be applied to the larger logits. Thus, we compute the p_{ij} by a softmax function whose input is $\boldsymbol{l}_i$ divided by a temperature t. Note that this computation is detached from the gradient computation graph, since we require that p_{ij} is constant. The temperature parameter controls the weight of product terms in $\hat{\mathcal{L}}_{EMR}$: if $t \to 0$, we only have the prediction's squared weight norm, i.e., $\hat{\mathcal{L}}_{EMR} = \max_{j \in K} \|\boldsymbol{w}_i^{(j)}\|_2^2$; if $t \to \infty$, we have a summation of $\langle \boldsymbol{w}_i^{(j)}, \boldsymbol{w}_i^{(k)} \rangle$ for all i, j in $\hat{\mathcal{L}}_{EMR}$. In the empirical study, we find that selecting an appropriate temperature parameter is able to improve the performance of EMR. In the supplemental we show the performance of the approximate EMR (Approx-EMR) on CNN and MLP with ST and AT. The approximation achieves a comparable performance in both models and even better robustness in the CNN trained with AT.

Another crucial problem with EMR is that the "training mode" of a DNN with batch normalization will incur correlations among the batch of samples, since the batch normalization is used with the current batch's mean and variance. To avoid the intertwined gradients, *we use the evaluation mode in the forward propagation when computing EMR*, where we normalize the input batch with the running mean and variance. In this way, we treat the DNN as a locally linear function, and EMR still has the meaning of regularizing local weight norms. Moreover, using evaluation mode in the effective norm computation is faster than using training mode. See the pseudo-code is in the supplemental.

5 Large-Scale Experiments

We evaluate the proposed EMR on large-scale DNNs and show that EMR improves upon AT, TRADES, two recent baselines [37, 41] that also aim to optimize the effective margin and strong adversarial defense methods [26, 48].

5.1 Experimental Setting

Architectures and Datasets. We evaluate the effectiveness of EMR with ResNet18 [18] and WideResNet-34-10 [52] following existing work on adversarial robustness [33]. The experiment is run on CIFAR10 [21], consisting of 50k training and 10k test images. There are 10 object categories in CIFAR10 and each category has 5000 training and 1000 test images. Both adversarial training with PGD [33] and TRADES PGD [53] are used in the experiment.

Training Setting. We use a standard SGD-Momentum optimizer and XE loss for training. All experiments use an initial learning rate of 0.1 and batch size of 128. For AT-PGD, we train the network for 100 epochs and the learning rate is divided by 10 at epochs 60 and 90. For TRADES, we train the network for 130 epochs and the learning rate is divided by 10 at epochs 60 and 120. Note that for EMR, when the learning rate is decayed, λ_{EMR} is also divided by 10. To avoid robust overfitting [40], we split the original training set into a training and validation set with 48k and 2k images respectively, and select a model with the best robust test accuracy on the validation set. If not mentioned, all methods are evaluated using the model selection based on the validation set for a fair comparison.

Evaluation. FGSM [15], PGD [33] and AutoAttack [10] are used to evaluate the adversarial robustness of a DNN. In FGSM, PGD and AutoAttack, we use the l_∞ norm as the metric to bound the adversarial perturbation. The l_∞ norm PGD attack uses gradient ascent to increase the loss by updating the input image and projecting it to an ε-bounded l_∞ ball. FGSM is a special case of PGD using only 1 iteration. AutoAttack is an ensemble of parameter-free attacks consisting of Auto-PGD$_{CE}$, Auto-PGD with Difference of Logits Ratio loss, FAB [8] and Squared Attack [1]. In our experiment, we use $\varepsilon = 8/255 = 0.031$ and $\alpha = 2/255 = 0.0078$ in FGSM and PGD, and $\varepsilon = 0.031$ for AutoAttack. We notice that PGD without random start is more effective than PGD with random start so our PGD evaluation starts from the input image without random noise.

Baselines. To make a fair comparison between EMR and baseline with WD, we search the λ_{WD} from 2e-4 to 1e-3, so that the improvement of EMR upon WD is not a result of weak WD regularization. Besides AT and TRADES, we compare EMR with Input Gradient Regularization (IGR) [41] and Hypersphere Embedding (HE) [37]. IGR regularizes the squared l_2 norm of the *loss*, instead of the network output as with EMR, with respect to the input. HE applies a normalization function to the feature, i.e., output of the penultimate layer, and the classification layer's weight, so that the XE loss is only determined by the cosine similarity. We compare the performance of HE and our EMR in the WideResNet experiment using

	Clean Acc.	FGSM	PGD10	PGD100	AutoAttack
AT	87.35	59.97	53.55	52.30	50.31
AT+IGR[41]	**87.49**	60.32	53.49	52.42	50.42
AT+HE[37]	84.53	64.07	60.36	59.80	51.88
AT+EMR (ours)	85.74	60.67	55.43	54.62	**52.20**
TRADES	82.95	60.65	56.71	56.17	52.19
TRADES+IGR[41]	**84.18**	61.16	56.67	55.90	52.41
TRADES+HE[37]	79.61	60.95	58.23	57.99	51.49
TRADES+EMR (ours)	83.03	60.89	57.27	56.89	**52.73**
AT+MAIL [26]	86.96	60.90	55.42	54.53	45.07
AT+MAIL+EMR (ours)	**87.33**	61.32	56.77	56.00	**46.25**
TRADES+MAIL [26]	84.82	60.44	55.35	54.69	51.97
TRADES+MAIL+EMR (ours)	**85.37**	61.67	56.82	56.21	**53.29**
MART [48]	**83.62**	61.83	57.32	56.43	51.40
MART+EMR (ours)	83.55	62.87	58.09	57.43	**52.16**

Table 3: Evaluation of adversarial robustness using WideResNet-34-10 on CIFAR10. The best result under the strongest attack is emphasized with bold text.

their official implementation. For IGR and HE, we use their default training parameters. For EMR, we select the hyperparameters with grid search. The specific hyperparameter settings are reported in the supplemental.

5.2 Experimental Results

Table 3 compares the result of vanilla AT, IGR, HE and our EMR using WideResNet-34-10 [33]. HE and EMR improve the performance over the vanilla AT, while EMR achieves the best result. For TRADES, HE and IGR do not improve upon the baseline by a large margin, while EMR still achieves the best result. Note that HE has the best robust accuracy under PGD attack, but it is easily attacked by AutoAttack, a stronger and more reliable attack than PGD so it has become a more important evaluation method than PGD for adversarial robustness in recent years [11, 26]. We also find that HE does not work well with TRADES adversarial training, which is consistent with the claim in their official code repository. See the result of using ResNet18 in the supplemental.

Probabilistic margin-aware instance re-weighting learning (MAIL) [26] is proposed to weight samples based on the probabilistic margin $p_{iy} - \max_{j \neq y} p_{ij}$, where p_{ij} is the output of softmax function in the XE loss. However, MAIL uses the unnormalized margin in the re-weighting and does not consider the effective margin maximization. Thus, we can apply our EMR in MAIL loss training to control the effective weight norm so that the re-weighting is based on the effective margin instead of the unnormalized margin. We use the default settings in the MAIL loss for MAIL and MAIL+EMR. All training images of CIFAR10 are used for training and the evaluation is done on the model of the final epoch. Table 3 shows the result of using EMR in MAIL, which demonstrates that for both AT and TRADES, there is a substantial improvement in adversarial robustness. Fig. 3 shows the robust accuracy under AutoAttack versus attack budget ε and compares MAIL with MAIL-EMR. EMR always improves the performance in this attack range. In addition, we combine EMR with Misclassification Aware adveRsarial Training (MART) [48], to demonstrate the effectiveness of our EMR, shown in Tab. 3, where EMR also substantially improves the robustness under PGD and AutoAttack.

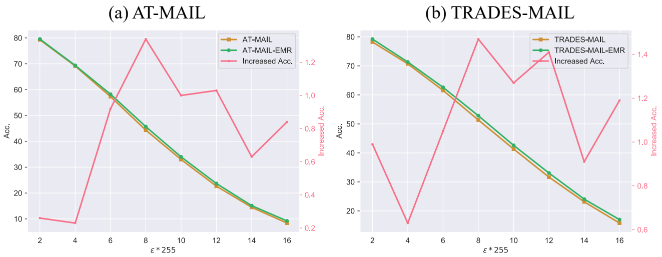

Figure 3: Robust test accuracy of WideResNet under AutoAttack when ε increases. (a) and (b) compare the original MAIL and MAIL+EMR using AT and TRADES, where left y-axis is the absolute accuracy and right y-axis is the relative accuracy of EMR with respect to the MAIL baseline. Our EMR always improves the robustness of MAIL across a large ε range.

	Model	Clean Acc.	FGSM	PGD10	PGD100	AA
TRADES+EWR-clean	ResNet18	**79.88**	57.57	53.86	53.18	48.66
TRADES+EWR-adv	ResNet18	79.59	57.43	53.53	53.01	**48.88**
TRADES+EWR-clean	WRN-34-10	**83.97**	61.26	56.51	55.76	52.55
TRADES+EWR-adv	WRN-34-10	83.03	60.89	57.27	56.89	**52.73**

Table 4: Comparison between TRADES+EMR using clean and adversarial images.

5.3 Ablation Studies

For TRADES+EMR, we study the difference between computing EMR using clean and adversarial examples, since both images are used in TRADES training. Table 4 shows the results. EMR with clean examples has a better clean accuracy, while EMR with adversarial examples has a better robust accuracy. Since our target is the adversarial robustness, we use latter setting of EMR. To evaluate the parameter sensitivity of EMR, we evaluate the robustness of ResNet18 when selecting the hyperparameters with grid search. In the supplemental, we show the robust accuracy of different combinations of temperature t and λ_{EMR}. In AT, selecting a large temperature generally improves the performance. We also report the computational time of EMR combined with IGR and HE in the supplemental.

6 Conclusion

This paper investigates the effective margin in training DNNs with the XE loss. Our experiment shows that existing methods do not adequately maximize the effective margin. Therefore, we propose EMR to maximize the effective margin and learn an adversarially robust DNN. On both MLP and CNN, EMR shows a clear strength over WD and L-Softmax in terms of both effective margin and adversarial robustness. On large-scale models, we demonstrate the efficacy of EMR by comparing with 10 strong baselines. We will explore a fast and general EMR in future work so that our method can be applied to larger models.

Acknowledgement

This work was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11215820).

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[3] Gregor Bachmann, Seyed-Mohsen Moosavi-Dezfooli, and Thomas Hofmann. Uniform convergence, adversarial spheres and a simple remedy. In *International Conference on Machine Learning*, pages 490–499. PMLR, 2021.

[4] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations*, 2018.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[6] Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

[7] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[8] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020.

[9] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[10] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[11] Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15721–15730, 2021.

[12] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

[13] Gavin Weiguang Ding, Yash Sharma, Kry Yik-Chau Lui, and Ruitong Huang. Max-margin adversarial (mma) training: Direct input space margin maximization through adversarial training. *ArXiv*, abs/1812.02637, 2020.

[14] Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Advances in Neural Information Processing Systems*, 31, 2018.

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

[16] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34, 2021.

[17] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Hanxun Huang, Yisen Wang, Sarah Erfani, Quanquan Gu, James Bailey, and Xingjun Ma. Exploring architectural ingredients of adversarially robust deep neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[20] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[22] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

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

[24] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[25] Yao Li, Martin Renqiang Min, Thomas Lee, Wenchao Yu, Erik Kruus, Wei Wang, and Cho-Jui Hsieh. Towards robustness of deep neural networks via regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7496–7505, 2021.

[26] Feng Liu, Bo Han, Tongliang Liu, Chen Gong, Gang Niu, Mingyuan Zhou, Masashi Sugiyama, et al. Probabilistic margins for instance reweighting in adversarial training. *Advances in Neural Information Processing Systems*, 34, 2021.

[27] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016.

[28] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[29] Ziquan Liu, Yufei Cui, and Antoni B Chan. Improve generalization and robustness of neural networks via weight scale shifting invariant regularizations. *arXiv preprint arXiv:2008.02965*, 2020.

[30] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE international conference on computer vision*, pages 446–454, 2017.

[31] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

[32] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

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

[34] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[35] Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. Towards robust detection of adversarial examples. *Advances in Neural Information Processing Systems*, 31, 2018.

[36] Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *arXiv preprint arXiv:1905.10626*, 2019.

[37] Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Jun Zhu, and Hang Su. Boosting adversarial training with hypersphere embedding. *Advances in Neural Information Processing Systems*, 33:7779–7792, 2020.

[38] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. *Advances in Neural Information Processing Systems*, 32, 2019.

[39] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34, 2021.

[40] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

[41] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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

[43] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[44] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

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

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

[47] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[48] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

[49] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

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

[51] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

[52] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[53] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.