Understanding Adversarial Robustness: The Trade-off between Minimum and Average Margin

Kaiwen Wu and Yaoliang Yu
University of Waterloo, Waterloo AI Institute, Vector Institute
{kaiwen.wu, yaoliang.yu}@uwaterloo.ca

Abstract

Deep models, while being extremely versatile and accurate, are vulnerable to adversarial attacks: slight perturbations that are imperceptible to humans can completely flip the prediction of deep models. Many attack and defense mechanisms have been proposed, although a satisfying solution still largely remains elusive. In this work, we give strong evidence that during training, deep models maximize the *minimum* margin in order to achieve high accuracy, but at the same time decrease the *average* margin hence hurting robustness. Our empirical results highlight an intrinsic trade-off between accuracy and robustness for current deep model training. We also show that such trade-off can be broken by robust training methods.

1 Introduction

The seminal work of Szegedy et al. [1] demonstrated surprisingly that it is possible to craft very minimal changes to an input image that (a) is clearly not perceivable by humans but (b) can completely flip the predictions of state-of-the-art deep models. Such detrimental perturbations are called adversarial examples and have raised serious concerns on the safety of deep models.

A sequence of works have proposed new attack algorithms to generate adversarial examples [*e.g.* 2–4], as well as defensive techniques [*e.g.* 5–7] to train more robust models. In addition, a line of research focusing on provable certification of (non)robustness has emerged, providing proof of non-existence of adversarial examples in a neighbourhood. For example, one can use the Lipschitz constant of the network to provide a lower bound on the adversarial perturbation [*e.g.* 8–11]. More recently, certification methods base on mixed integer programming (MIP) [12, 13] have been proposed to provide more accurate lower bound. However, due to NP-hardness of MIP, one need convex relaxations [14–17] in order to scale the certification method to larger models.

In this paper, we reveal a surprising trade-off between minimum and average margin in standard deep model training. In particular, the recent result of [18] implies that on a linearly separable dataset a linear classifier converges to the SVM solution, under mild conditions on the loss function. Since SVM explicitly maximizes the minimum margin, any linear classifier would eventually achieve the same. However, our experiments reveal that the average margin is greatly reduced during training. In other words, models maximize the minimum margin at the expense of reducing the average margin hence becoming more susceptible to adversarial attacks. This surprising phenomenon was also confirmed on linearly nonseparable datasets and a variety of nonlinear classifiers.

2 Background and Notations

We consider the multi-category classification problem. Denote a set of n training instances as $\{(\mathbf{x}_i, y_i) \in \mathcal{X} \times [c] : i = 1, \dots, n\}$, where the feature domain $\mathcal{X} \subseteq \mathbb{R}^d$ and $[c] := \{1, \dots, c\}$ with $c \geq 2$ the number of categories. A classifier $\mathbf{f} : \mathcal{X} \to \mathbb{R}^c$, predicting $\hat{y}$ of a test sample $\mathbf{x}$, corresponds

NeurIPS 2019 Workshop on Machine Learning with Guarantees, Vancouver, Canada.

to a set partition $\mathsf{F}_1, \ldots, \mathsf{F}_c$ of the domain $\mathcal{X}$ such that $\cup_k \mathsf{F}_k = \mathcal{X}$ and for all $k \neq l$, $\mathsf{F}_k \cap \mathsf{F}_l = \emptyset$. We define the distance from a point $\mathbf{x} \in \mathcal{X}$ to a set $\mathsf{F} \subseteq \mathcal{X}$ as:

$$\mathsf{d}(\mathbf{x}, \mathsf{F}) := \inf\{\|\mathbf{x} - \mathbf{z}\| : \mathbf{z} \in \mathsf{F}\}. \tag{1}$$

Thus, $\mathsf{d}(\mathbf{x}, \operatorname{bd} \mathsf{F}_{\hat{y}})$ is the distance to the decision boundary , corresponding to the minimum perturbation to flip the prediction. We define the *individual* margin of $(\mathbf{x}, y)$ w.r.t. the classifier $\{\mathsf{F}_k\}$ as:

$$\mathsf{m}(\mathbf{x}, y) := \mathsf{m}(\mathbf{x}, y; \{\mathsf{F}_k\}) := \operatorname{sign}(\hat{y}(\mathbf{x}), y) \cdot \mathsf{d}(\mathbf{x}, \operatorname{bd} \mathsf{F}_{\hat{y}}),$$

where $\operatorname{sign}(\hat{y}, y) = 1$ if $\hat{y} = y$ and $\operatorname{sign}(\hat{y}, y) = -1$ otherwise. Namely, when predicting correctly, the margin is the distance to the decision boundary, measuring the minimum perturbation to change the prediction; when predicting wrongly, it is the negation of distance to the decision boundary, measuring the distance to correct the prediction. With the above notations, we define

- **minimum margin:** $\min_{1 \leq i \leq n} \mathsf{m}(\mathbf{x}_i, y_i)$
- **average margin:** $\frac{1}{n} \sum_{i=1}^{n} \mathsf{m}(\mathbf{x}_i, y_i)$

In the next section, we will reveal an intrinsic trade-off between these two notions.

3 Minimum *vs*. Average Margin: A Case Study

In statistical learning theory, it is well-known that we can bound the generalization error of a classifier using empirical margins [19]. In particular, the support vector machines (SVM) explicitly maximize the minimum margin. In adversarial learning, however, we argue that average margin is more indicative of the robustness of a classifier.

Our main observations in this section are: (a) current machine learning models, especially when they become deep and powerful, implicitly maximize the minimum margin to achieve high accuracy; (b) there appears to be an inherent trade-off between minimum margin and average margin. In particular, by maximizing the minimum margin the model also (unconsciously) minimizes the average margin, hence becomes susceptible to adversarial attacks. Note that we do not claim minimum margin always contradicts average margin. It does not, as can be easily shown through carefully constructed toy datasets (*e.g.* a highly symmetric dataset). Our empirical observation is that there does appear to be some tension between the two notions of margin on real datasets.

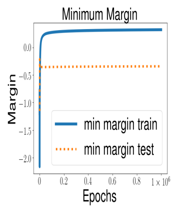
To begin with, we first demonstrate that minimum margin is maximized during standard training. In fact, this can be formally established through the implicit bias of optimization algorithms, such as gradient descent which is ubiquitously used in training deep models. Recall that on a (linearly) separable dataset, (hard-margin) linear SVM explicitly maximizes the minimum margin. The following result, due to Soudry et al. [18], confirms the same for most models currently used in machine learning, including logistic regression.

Theorem 1 (Soudry et al. 18)**.** *Consider binary classification. For almost all linearly separable datasets and any smooth decreasing loss with an exponential tail, gradient descent with small constant step size and any starting point* $\mathbf{w}_0$ *converges to the (unique) solution* $\widehat{\mathbf{w}}$ *of hard-margin SVM:*

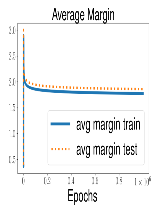
$$\lim_{t \to \infty} \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|} = \frac{\widehat{\mathbf{w}}}{\|\widehat{\mathbf{w}}\|}. \tag{2}$$

Thus, Theorem 1 implies in particular that if we optimize logistic regression by gradient descent on a linearly separable dataset, then we implicitly maximize the minimum margin, just as in SVM. In addition to linear classifiers considered in Theorem 1, the implicit maximizing minimum margin effect for deep neural networks has also been discovered [20, 21].

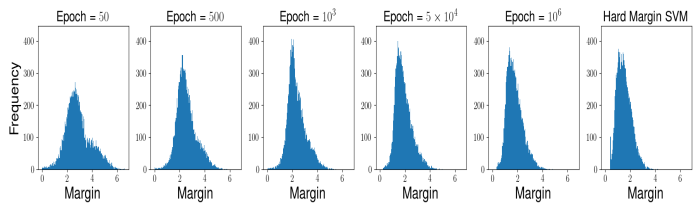
Next, we ask the natural question: **How does average margin change during training?** To answer this question, we train a binary logistic regression (LR) using gradient descent on MNIST [22] to classify 0's and 1's. Note that the subset of MNIST consisting only of 0's and 1's is indeed linearly separable, as LR achieves zero training error (see Figure 3 in appendix). All conditions of Theorem 1 are thus satisfied, and we expect LR to maximize minimum margin. Indeed, Figures 1a and 1b confirms that the minimum margin continues to increase during training until approaches that of SVM. Meanwhile, the average margin decreases drastically after a few epochs at the very beginning, and then keeps decreasing. To gain further insight, we plot the margin histograms at different training



(a) Min margin



(b) Avg margin



(c) **First 5 plots:** Margins of LR during training. **Last plot:** Margins of SVM. During training, the histogram continues shifting towards the left.

epochs in Figure 1c. It is clear that the distribution of margins shifts towards the left during training and eventually approaches that of SVM. In other words, during training the majority of data are pushed towards the decision boundary, leading LR to become more and more vulnerable.

We summarize the finding from this toy example. Among all zero-risk classifiers, standard training has bias towards the max-minimum-margin classifier. Unfortunately, the classifier with the largest minimum margin does not have large average margin, hence suffering adversarial vulnerability. Thus, to make models robust, one has to break the margin trade-off, by introducing bias towards average margin. Notice that several robust training methods based on margin maximization have been proposed [23, 24], which should be understood as average margin instead of minimum margin.

4 Experiments

In this section, we provide further extensive evidence to minimum-average margin trade-off discussed in §3, by investigating the following six models on MNIST [22] and CIFAR10 [25]:

- **MNIST-LR/CIFAR-LR:** Multiclass logistic regression.
- **MNIST-MLP/CIFAR-MLP:** Shallow neural networks with one hidden layer.
- **MNIST-CNN/CIFAR-CNN:** Deep convolutional neural networks.

4.1 Approximating Distance

In general, exactly computing the distance to the decision boundary of deep models is intractable. Instead, we use the following Lipschitz lower bound as an approximation:

Theorem 2 (Hein and Andriushchenko 8)**.** *Suppose* $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^c$ *is a multiclass classifier (with argmax prediction rule). Then, for any* $r > 0$*:*

$$\mathsf{d}(\mathbf{x}, \mathrm{bd}\,\mathsf{F}_{\hat{y}}) \geq \min\left\{\min_{k \neq \hat{y}} \frac{f_{\hat{y}}(\mathbf{x}) - f_k(\mathbf{x})}{L_{\mathbf{x}}^k}, r\right\}, \tag{3}$$

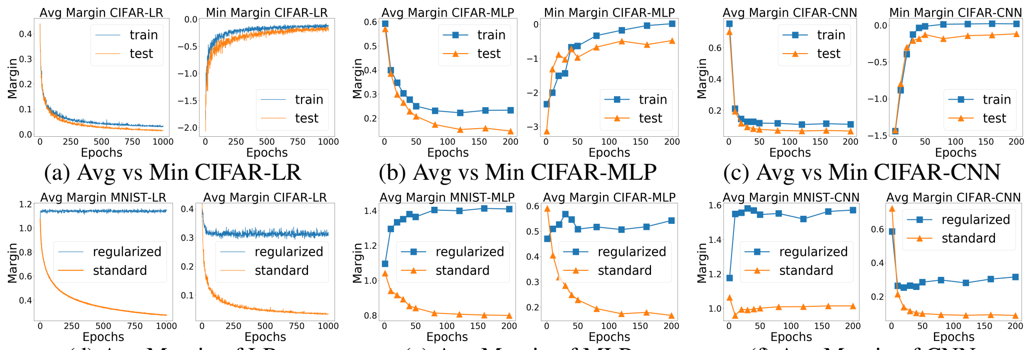
where $L_{\mathbf{x}}^k$ *is the local Lipschitz constant of* $f_{\hat{y}}(\mathbf{x}) - f_k(\mathbf{x})$ *over the ball* $B(\mathbf{x}, r)$*.*

The lower bound (3) is tight when $\mathbf{f}$ is linear, and for more general nonlinear $\mathbf{f}$, it is reasonably close to the true distance, as shown empirically in [9]. Here, our estimation for the Lipschitz constant $L_{\mathbf{x}}^k$ is simply the maximum norm of the gradient (difference) of many random samples in the ball $B(\mathbf{x}, r)$, as we found this simple strategy is reasonably accurate and efficient. Throughout the experiments, we set $r = 5$ and the sampling size equal to 1024×50. We find that doubling the sample size do not change the estimation results much. All experiments can be run on a single GPU.

4.2 Margin Trade-off for Nonlinear Models

We plot the margins on both training and test sets in Figures 2a to 2c. It is clear that similar phenomenon as those in §3 can be observed: the minimum margin continues increasing while the average margin keeps decreasing. This again highlights an intrinsic trade-off between minimum margin and average margin.

To investigate how the margin distribution changes during training, we plot margin histograms (see Figures 5a to 5f in appendix). As training proceeds, we observe again the distribution of margins shifts towards left, meaning that the majority of data points are pushed closer to the boundary. To some extent, this provides compelling explanation of the non-robustness of deep models: although achieving high accuracy by maximizing the minimum margin, the average margin, which is a better indicator of robustness, is at serious jeopardy.



(a) Avg vs Min CIFAR-LR (b) Avg vs Min CIFAR-MLP (c) Avg vs Min CIFAR-CNN

(d) Avg Margin of LR (e) Avg Margin of MLP (f) Avg Margin of CNN

Figure 2: **Top:** Average and minimum margin trade-off on CIFAR. **Bottom:** Average margin of regularized and standard training.

4.3 Explicit Optimization of Margin

Since standard training will bias towards the max-minimum-margin classifier, we propose a regularizer to explicitly promote average margin, hence break the margin trade-off. For deep neural networks, even computing the margin is already NP-hard [11], let alone optimizing it. However, the margin in the feature space provides a lower bound of the margin in the input space. Let $\mathbf{f}$ be a deep neural network with L layers, $\mathbf{f}(\mathbf{x}) = W_L \cdot \sigma(W_{L-1} \cdot \sigma(\cdots \sigma(W_1 \cdot \mathbf{x})))$, where σ is the activation function. Let $\Phi(\mathbf{x})$ be the output of the second last fully connected layer, then $\|\Phi(\mathbf{x}_1) - \Phi(\mathbf{x}_2)\| \leq Lip(\Phi)\,\|\mathbf{x}_1 - \mathbf{x}_2\|$, where $Lip(\Phi)$ is the Lipschitz constant of the feature transformation map Φ. In particular, it suggests a natural way to maximize input space margin: we control the Lipschitz constant of $\mathbf{f}$ and maximize the feature space margin simultaneously. Our regularized objective is

$$\sum_{i=1}^{n} \phi\left(y_i, \mathbf{f}(\mathbf{x}_i)\right) - \lambda \left[\min_{k \neq y_i} (\mathbf{w}_{y_i} - \mathbf{w}_k)^\top \Phi(\mathbf{x}_i)\right]_0^\tau + \beta \sum_{1 \leq l \leq L} \|W_l W_l^\top - I\|_{\mathrm{F}}^2, \tag{4}$$

where the first term is the standard training loss (such as cross-entropy), the third term is the orthogonal constraint for controlling the Lipschitz constant of the network, and the second term is the average margin penalty in the feature space. Notice that τ is a truncation parameter. Namely, we only maximize the average margin up to τ, to avoid outliers dominating the average margin.

We retrain the models in §4.2 using our average margin regularizer. We find that models trained by (4) have much larger average margin than models trained by standard training (see Figures 2d to 2f). In addition, we evaluate the models under projected gradient descent attack (PGD) [7], and observe that our regularizer has higher robust accuracy, compared to standard training, which indicates our regularizer can break the margin trade-off and increase the robustness. Notice that our regularizer is different from the Lipschitz constant regularization method (LCR) [26], as we are maximizing the margin explicitly. From Table 1, we can see that our regularizer has both higher average margin as well as higher robust accuracy, especially for large ϵ.

5 Conclusion

In this work, we discovered the intrinsic trade-off between minimum and average margin, which appears across different models and datasets. We gave strong empirical evidence that deep models maximize the minimum margin during training to achieve high accuracy, but at the expense of decreasing the average margin significantly, hence becoming more susceptible to adversarial attacks. In the future, we plan to theoretically analyze the training dynamics to provide further insights to the phenomenon.

Table 1: Comparison between different training methods. **Std:** standard training **LCR:** Lipschitz constant regularization **AMR:** average margin regularizer **Column 4:** clean accuracy **Column 5-8:** robust accuracy under l_2 PGD attack **The last column:** average margin

	Models	Method	Clean	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 1.5$	$\epsilon = 2.0$	Avg Margin
MNIST	MLP	Std	**98.24**	89.26	49.23	15.78	5.06	0.80
		LCR	95.99	91.12	80.62	60.56	34.07	1.36
		AMR	96.01	**91.18**	**81.01**	**62.93**	**38.44**	**1.41**
	CNN	Std	99.14	95.95	90.48	88.72	87.52	1.01
		LCR	**99.29**	97.21	87.83	58.30	26.96	1.34
		AMR	99.25	**97.90**	**97.60**	**97.51**	**97.33**	**1.40**
	Models	Method	Clean	$\epsilon = 0.1$	$\epsilon = 0.2$	$\epsilon = 0.3$	$\epsilon = 0.4$	Avg Margin
CIFAR	MLP	Std	**54.04**	39.77	26.67	16.77	9.95	0.17
		LCR	50.09	46.23	41.65	37.09	32.62	0.45
		AMR	50.36	**46.28**	**42.81**	**39.06**	**35.67**	**0.54**
	CNN	Std	78.77	54.32	39.81	37.14	36.27	0.09
		LCR	**80.54**	**70.84**	58.72	44.82	32.54	0.26
		AMR	78.12	69.59	**59.84**	**49.02**	**38.65**	**0.31**

Acknowledgement

We thank the reviewers for their comments. This work is supported by NSERC and a grant from Waterloo-Huawei Joint Innovation Lab.

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[3] 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.

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2017. arXiv:1608.

[5] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy*, 2016.

[6] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, 2017.

[7] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning model resisstant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[8] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2266–2276, 2017.

[9] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations (ICLR)*, 2018.

[10] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems*, pages 4944–4953, 2018.

[11] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Duane Boning, Inderjit S. Dhillon, and Luca Daniel. Towards fast computation of certified robustness for relu networks. In *(ICML)*, 2018.

[12] Vincent Tjeng, Kai Y. Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2019.

[13] Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. Robustness certification with refinement. In *International Conference on Learning Representations*, 2019.

[14] Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, pages 8400–8409, 2018.

[15] Eric Wong and J Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. 2018.

[16] Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pages 10900–10910, 2018.

[17] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

[18] Daniel Soudry, Elad Hoffer, and Nathan Srebro. The Implicit Bias of Gradient Descent on Separable Data. In *International Conference on Learning Representations*, 2018.

[19] V. Koltchinskii and D. Panchenko. Empirical Margin Distributions and Bounding the Generalization Error of Combined Classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

[20] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pages 9461–9471, 2018.

[21] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019.

[22] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. URL `http://yann.lecun.com/exdb/mnist/`.

[23] Ziang Yan, Yiwen Guo, and Changshui Zhang. Deep defense: Training dnns with improved adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 419–428, 2018.

[24] Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In *AISTATS*, 2019.

[25] Geoffrey Hinton Alex Krizhevsky, Vinod Nair. The cifar-10 dataset. URL `https://www.cs.toronto.edu/~kriz/cifar.html`.

[26] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *Proceedings of the 34th International Conference on Machine Learning*, pages 854–863, 2017.

A Detailed Experiment Setting

Model Architectures:

- **MNIST-LR and CIFAR-LR:** Multiclass logistic regression.
- **MNIST-MLP and CIFAR-MLP:** Neural networks with one hidden layer. The hidden layer has 1024 neurons and ReLU activation.
- **MNIST-CNN and MNIST-CNN:** Two AlexNet-like convolutional neural networks with a slight difference. They are exactly the same as the models used by Weng et al. [9], Carlini and Wagner [4].

Margin Estimation: For linear logistic regression, we perform estimation in every epoch; for nonlinear classifiers, we only perform estimation in epochs 1, 10, 20, 30, 40, 50, 80, 120, 160 and 200. We perform more estimations at the beginning, as margins may change drastically in initial stages. The estimation is performed on a subset of size 500, which is randomly chosen from the original training and test sets.

Standard Training: We use SGD (learning rate 0.01) with momentum (0.9) and nestorv to train all six models. Batch size is set to 128. Linear models (LR) are trained for 1000 epochs. Nonlinear models (MLP, CNN) are trained for 200 epochs. No data augmentation is used.

Lipschitz Constant Regularizer: We follow the method proposed by Cisse et al. [26]. We set $\beta = 10^{-3}$ as suggested in [26].

Average Margin Regularizer: We set $\lambda = 0.1$ and $\beta = 10^{-3}$ for all models. While different τ's are used for different models. τ is tuned such that it is on the same order as the Lipschitz constant of the model. MNIST-LR and MNIST-MLP use $\tau = 5$. MNIST-CNN use $\tau = 10$. CIFAR-LR use $\tau = 5$. CIFAR-MLP and CIFAR-CNN use $\tau = 10$. The idea is using larger τ for deeper models.

Attack Method: We use projected gradient descent (PGD) attack to evaluate the robust accuracy. We set step size 0.01 and number of iterations 1000. Increasing the number of iterations does not change the robust accuracy much, thus 1000 iterations is sufficient to generate strong adversarial examples.

B Training Curves

Training curves of the binary logistic regression in Section 3 are shown in Figure 3. Training curves of models in Section 4 are shown in Figure 4.

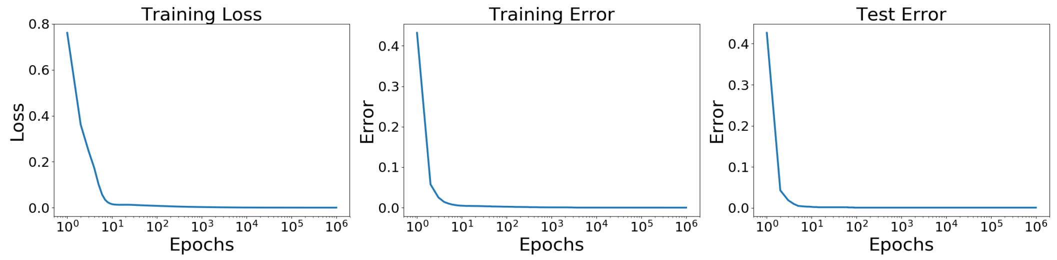


Figure 3: Training curves of logistic regression classifying 0 and 1 on MNIST. For the ease of illustration, x-axis is in log scale. **Left:** Training loss w.r.t. epochs. **Middle:** Training error w.r.t. epochs. **Right:** Test error w.r.t. epochs.

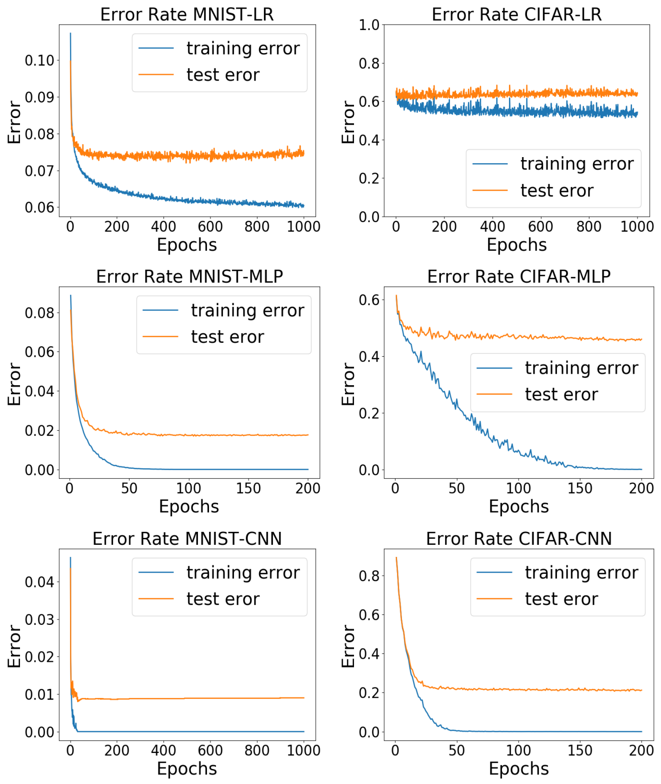
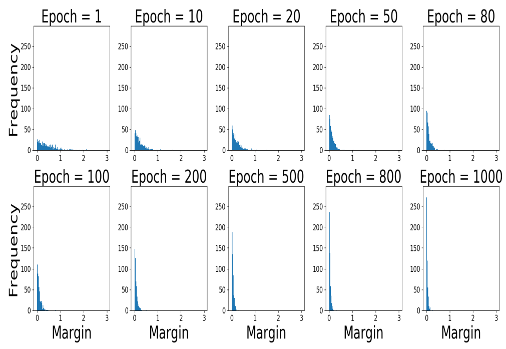


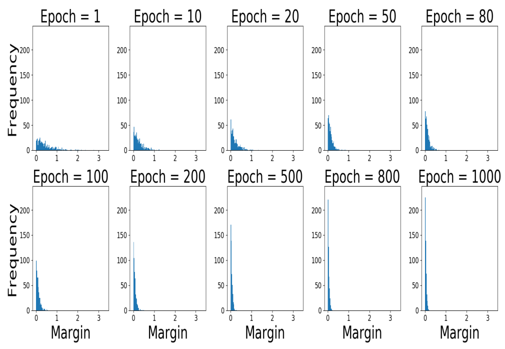
Figure 4: Training curves of 3 models on MNIST and 3 models on CIFAR10. **First Column:** MNIST models. **Second Column:** CIFAR10 models.

C Margin Histograms of CIFAR Models

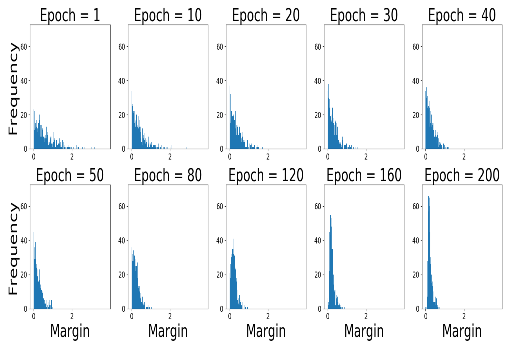
Margin histograms on training set and test set for CIFAR models are shown in Figures 5a to 5f.



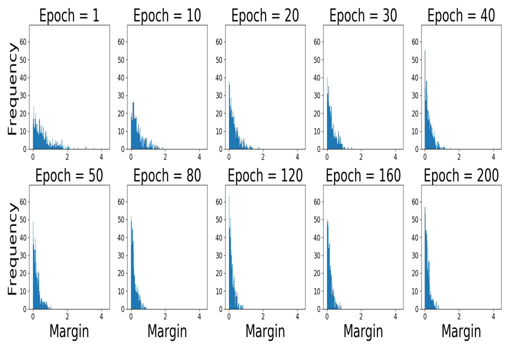
(a) Margin histogram of CIFAR-LR on training set



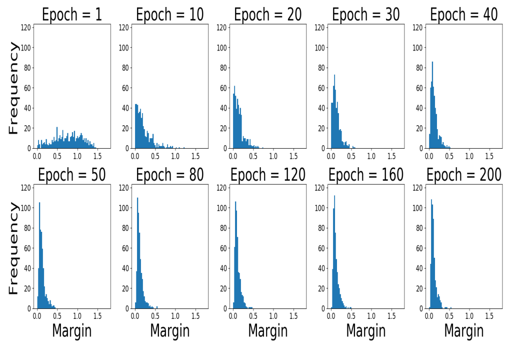
(b) Margin histogram of CIFAR-LR on test set



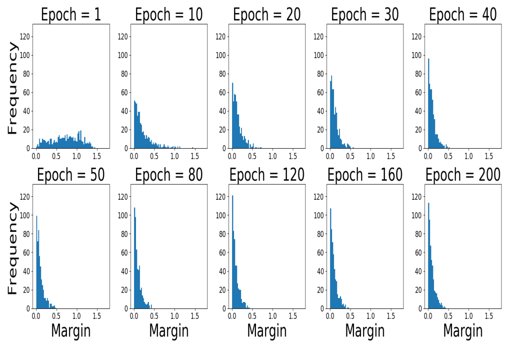
(c) Margin histogram of CIFAR-MLP on training set



(d) Margin histogram of CIFAR-MLP on test set



(e) Margin histogram of CIFAR-CNN on training set



(f) Margin histogram of CIFAR-CNN on test set

Figure 5: Margin histograms for CIFAR models

D Margin Curves of MNIST Models

Minimum and average margin trade-off for MNIST models is shown in Figure 6.

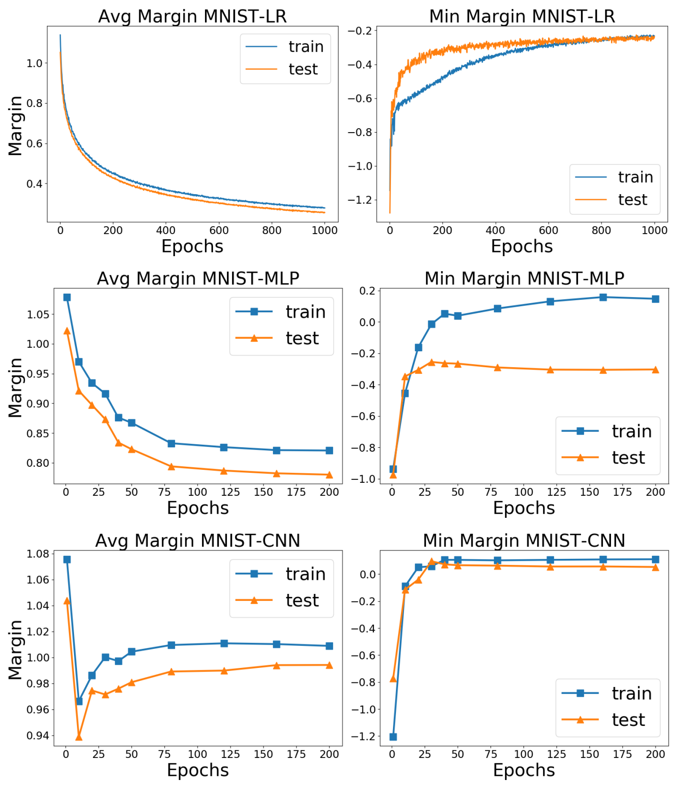
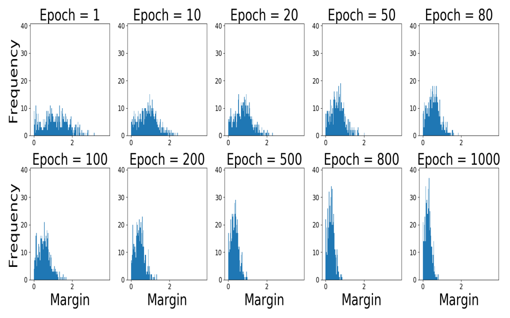


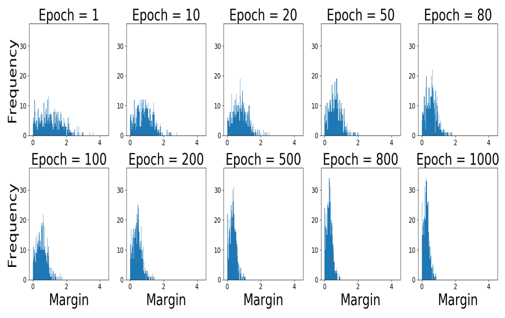
Figure 6: Average margin and minimum margin during training of 3 MNIST models. In the top left figure, we shift the curve of average margin on training set by a small constant 0.001, to avoid overlapping of two curves. This is only for illustration purpose.

E Margin Histograms of MNIST Models

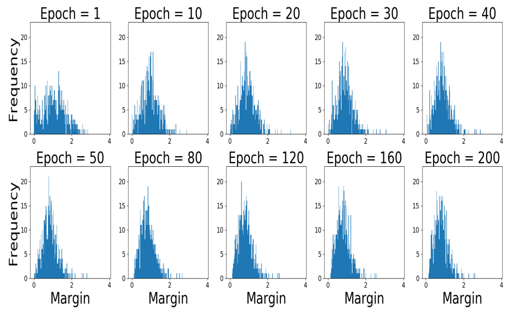
We plot margin histograms for MNIST models in Figures 7a to 7f.



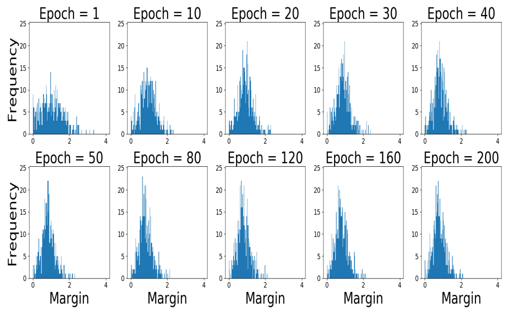
(a) Margin histogram for MNIST-LR on training set



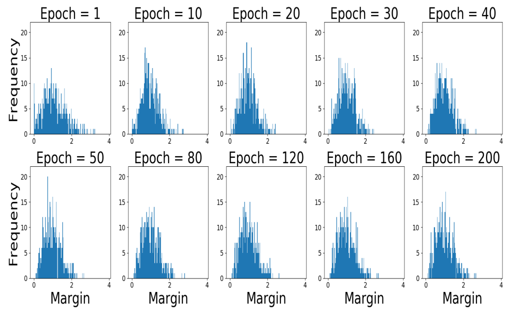
(b) Margin histogram for MNIST-LR on test set



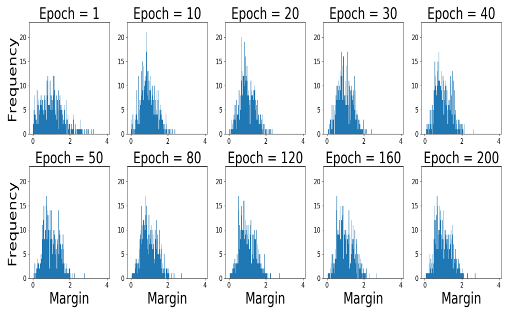
(c) Margin histogram for MNIST-MLP on training set



(d) Margin histogram for MNIST-MLP on test set



(e) Margin histogram for MNIST-CNN on training set



(f) Margin histogram for MNIST-CNN on test set

Figure 7: Margin histograms for MNIST models