ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Adversarial Training without Hard Labels

Ammar Al-Najjar[1], István Megyeri[1], and Márk Jelasity[1,2] *

1- University of Szeged, Hungary
2- HUN-REN–SZTE Research Group on Artificial Intelligence, Hungary

Abstract. Adversarial training is widely used to enhance classifier robustness. Several improvements have been proposed including different forms of distillation and self-alignment. Here, we propose a novel loss function combining these two approaches, while *not using the hard ground truth labels* directly. Our new loss function is demonstrated to simultaneously improve both the robustness and the accuracy of some well-known competing solutions. This is a step towards combatting the robustness-accuracy tradeoff, a crucial issue in adversarial training. Our method also reduces the variance of the accuracy over the classes in the experimental scenarios we examined, leading to a more balanced model.

1 Introduction

It has long been known that tiny adversarial input perturbations might lead to large output error in most machine learning models [7]. Adversarial training [2], where adversarially perturbed training samples are used as augmentation, is the most widely adopted approach to improve adversarial robustness.

Several improvements have been proposed, including the application of soft labels. One source of soft labels is the trained model itself. Logit pairing [8] and later on, TRADES [9], both use this *self-alignment* approach, where the outputs of the network on the clean and perturbed example are aligned with the help of the loss function.

Distillation [1] methods have also been proposed that use the soft output of a teacher network as regularization. RSLAD [11] and IAD [10] are examples where a robust teacher network is used. HAT [4], however, uses a standard teacher (trained on clean samples only), an approach that we also follow, because training a robust teacher is about an order of magnitude more expensive than training a regular network. Also, our goal is not compressing a network that is already robust, but instead, to improve adversarial training.

Here, as sources of soft labels, we combine self-alignment and distillation using a standard teacher. We propose to align the logit layer to that of a standard teacher network, and to align normal and adversarial input logits, minimizing

*Supported by the AI Competence Centre of the Cluster of Science and Mathematics of the Centre of Excellence for Interdisciplinary Research, Development and Innovation of the University of Szeged. We also thank the support by the the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory, and by project TKP2021-NVA-09 that has been implemented with the support provided by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme. We acknowledge KIFÜ (Governmental Agency for IT Development, Hungary, `https://ror.org/01s0v4q65`) for awarding us access to the Komondor HPC facility based in Hungary.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

their mean squared error (MSE). At the same time, we do not rely on the hard labels of the training set directly. Note that TRADES [9] and HAT [4] both rely on hard labels as ground truth, while our method relies solely on the guidance of a standard teacher network.

Our contributions are the following:

- We propose a novel loss function using both self-alignment and distillation, but no hard labels
- We show empirically on two benchmark datasets that our loss function simultaneously achieves a better clean and robust accuracy than competing solutions
- We observe that our loss function achieves a more even distribution of accuracy (both clean and robust) over the classes than competing methods

2 Background

Here, we first describe the basics of adversarial training. Let $f(x;\theta)$ denote a neural network with parameters θ. Assuming a classification problem, f outputs a probability distribution over the classes: $f : D \to [0,1]^C$, where C is the number of classes and D is the data domain.

Assuming that the distribution of labeled data (x, y) is $p(x, y)$, the goal of adversarial training is to solve the optimization problem

$$\arg\min_{\theta} E_{p(x,y)}[\mathcal{L}(x_{adv}, y, \theta)], \quad \text{where} \quad x_{adv} = \underset{x' \colon \|x'-x\|_p \leq \epsilon}{\arg\max} \mathcal{L}_{adv}(x', y, \theta). \quad (1)$$

Here, x_{adv} is the adversarial example based on attacking the natural example (x, y) with a perturbation of a bounded L_p distance ϵ. Here, we will use the commonly adopted L_∞ norm. The inner maximization problem is normally approximated with a basic iterative method using projected gradient descent (PGD) [2].

There are several proposals for defining adversarial training that differ in their choice of losses $\mathcal{L}$ and $\mathcal{L}_{adv}$. The original choice was simply using the cross-entropy $CE(f(x;\theta), y)$ for both losses [2], thus using only the hard labels y. More complex loss functions have also been proposed for both $\mathcal{L}$ and $\mathcal{L}_{adv}$ that use combinations of the softmax output and the logit layer of f, using different distance functions including, for example, mean squared error and Kullback-Leibler (KL) divergence. For example, TRADES [9] defines

$$\mathcal{L}(x, y, \theta) \coloneqq CE(f(x;\theta), y) + \beta KL(f(x;\theta), f(x_{adv};\theta)), \quad (2)$$

$$\mathcal{L}_{adv}(x', y, \theta) \coloneqq KL(f(x;\theta), f(x';\theta)), \quad (3)$$

where β is a hyperparameter. Note that TRADES applies self-alignment in the second term of the loss $\mathcal{L}$, where the clean and adversarial inputs are required to result in the same *soft* output.

Another source of guidance based on soft labels is knowledge distillation [1, 4, 11, 10] where the soft output of a teacher model is aligned to that of the student model. Our proposed loss function will use both self-alignment and distillation.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

3 Our Loss Function

For our loss function, we require a pre-trained standard model $g(x)$ that we use as a teacher model. The teacher is frozen throughout the adversarial training. Since training a standard model is much cheaper than adversarial training, it is feasible to train our own teacher model as part of the initialization. Here, model g will have the same architecture as the student model (that is, the model that we train adversarially), but this is not a strict requirement.

Given an example (x, y), we define our loss function as

$$\begin{aligned}\mathcal{L}(x, y, \theta) \coloneqq \beta[KL(f(x;\theta), g(x)) + KL(f(x_{adv};\theta), g(x))] + \\ KL(f(x;\theta), f(x_{adv};\theta))\end{aligned} \tag{4}$$

and $\mathcal{L}_{adv}$ is identical to that of TRADES given in (3). Note that none of the functions depend directly on the hard label y, the teacher network is responsible for guiding the training alone. This is achieved in the first term (multiplied by hyperparameter β) where both the regular and adversarial softmax outputs of the student are aligned with the teacher network. This solution promotes both clean and adversarial accuracy, based on informed soft labels, which will allow us to preserve more clean accuracy while achieving robustness. The self-alignment term is also identical to that of TRADES (see (2)).

The same ideas can be implemented based on the logit layers as well, instead of the softmax output. Assuming that our networks have a logit layer that precedes the final softmax output, we denote the logit layer of the student network $f(x;\theta)$ by $z(x;\theta)$, that is, $f(x;\theta) = \text{softmax}(z(x;\theta))$. The logit layer of the teacher $g(x)$ is denoted by $v(x)$. The logit version of our loss is implemented using mean squared error (MSE):

$$\begin{aligned}\mathcal{L}(x, y, \theta) \coloneqq \beta[MSE(z(x;\theta), v(x)) + MSE(z(x_{adv};\theta), v(x))] + \\ MSE(z(x;\theta), z(x_{adv};\theta)),\end{aligned} \tag{5}$$

$$\mathcal{L}_{adv}(x', y, \theta) \coloneqq MSE(z(x;\theta), z(x';\theta)). \tag{6}$$

Here, the self-alignment term is called logit pairing [8].

4 Experimental Results

In this section, we empirically evaluate the performance of our method on two benchmark datasets: Cifar10 and Cifar100 [12]. First, we investigate the effect of our hyperparameters, then we compare our method with four baselines: SAT [2], TRADES [9], HAT [4] and normal training with cross-entropy on hard labels.

Training Setup. We train ResNet-18 [5] networks using the SGD optimizer with Nesterov momentum 0.9 [15] and weight decay 0.0005. We use cyclic learning rates [6] with cosine annealing and a maximum learning rate of 0.2. We train all the models for 100 epochs with a batch size of 128. For computing adversarial examples during training, we apply the L_∞-PGD with the following parameters:

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Table 1: The effect of hyperparameters on accuracy and robust accuracy. The results of our logit-based loss are shown. Δ represents the performance improvement over our softmax-based loss (positive values mean that the logit-based variant is better).

	Cifar10				Cifar100			
β	Acc.	Acc. Δ	AA	AA Δ	Acc.	Acc. Δ	AA	AA Δ
0.05	81.12	1.16	50.90	0.99	52.88	*-2.85*	22.39	*-1.94*
0.1	84.06	2.44	50.76	1.13	59.08	1.40	23.93	*-1.16*
0.2	86.33	1.62	50.56	1.10	62.25	3.51	24.20	0.64
0.3	86.92	0.99	49.84	1.18	63.96	4.39	23.82	0.99
0.4	87.71	3.03	48.97	1.71	65.12	4.98	23.32	1.15

Table 2: Comparison with baseline methods.

		Cifar10			Cifar100		
Epoch	Method	β	Acc. (%)	AA (%)	β	Acc. (%)	AA (%)
Early Stopping	Standard	-	95.11	0	-	76.49	0
	SAT	-	84.96	47.17	-	56.41	22.49
	TRADES	5.0	84.09	50.26	5.0	57.23	23.83
	HAT	2.5	85.45	49.68	3.5	58.39	23.78
	Ours	0.2	**86.33**	**50.56**	0.2	**62.25**	**24.20**
Last (100.)	SAT	-	85.71	46.26	-	58.93	22.63
	TRADES	5.0	84.05	49.97	5.0	57.24	23.72
	HAT	2.5	85.9	49.14	3.5	58.39	23.77
	Ours	0.2	**86.07**	**50.63**	0.2	**62.68**	**24.16**

$\epsilon = 8/255$, step size $\alpha = 2/255$, and $K = 10$ iterations. We trained all the methods with the same settings (note that for our baselines we got better results with these settings than with those reported in the original publications).

Evaluation. We used early stopping [14]: we selected the best model based on the robustness of the model to the 20 iteration PGD attack on the validation set. The approximate robust accuracy, we used the full attack set of AutoAttack (AA) [13]. We report the average scores over three runs.

Discussion. **Table 1** compares the two variants of our approach: the logit-based (4) and the softmax-based (5,6) variant, under several settings of β. It is clear that β controls the tradeoff between robustness and accuracy, and $\beta = 0.2$ represents a good compromise for both datasets. Also, the logit variant is better in most settings, however, we do note that on CIFAR100, the best robust accuracy is achieved with the softmax loss and $\beta = 0.1$, namely 25.09 according to AA. **Table 2** compares our logit-based variant with methods from related work. The parameter (β) of the baseline methods is adopted from the recommendations in the corresponding publications. Note that we slightly improved the previously published baseline results due to our different training setup. Our method *improves both clean and robust accuracy simultaneously*, compared to the baselines.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

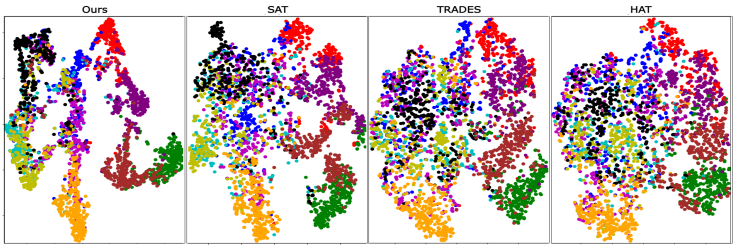

Fig. 1: t-SNE visualizations of the clustering of CIFAR10.

Table 3: Class-wise standard deviation of clean and robust accuracy (%).

	Cifar10		Cifar100	
Model	Clean	Robust	Clean	Robust
Ours	**6.9**	**16.0**	**15.3**	**17.9**
HAT	8.8	16.5	15.7	**17.9**
TRADES	9.0	17.0	16.3	18.3
SAT	9.4	17.8	19.0	18.2

5 More Balanced Class-wise Robustness

A main advantage of applying only soft labels (and no hard labels at all) is that we can account for the possibility that some—otherwise correctly predicted—classes might be less confident in general due to, for example, being similar to other classes. This way, we can avoid biasing the training by overconfident targets represented by the one-hot encoded hard labels.

To support this observation, we examined the standard deviation of the accuracy over the classes. That is, we compute the clean and robust accuracy (using AA) within each class over the test set and compute the class-wise standard deviation, as shown in Table 3. Indeed, the table confirms that our method consistently achieves the most uniform distribution in all the metrics.

To further illustrate the better balance between the classes, we visualize the clustering of each of the models we discussed in Section 4 using t-SNE [16]. We randomly selected 300 samples from each class in the Cifar10 test set, and we computed adversarial examples based on these samples using the setup described in Section 4. Then, for each model, the logit layers are computed and visualized, as shown in Figure 1. Colors correspond to class labels.

6 Conclusions

We have argued that using both distillation and self-alignment, while avoiding the direct application of hard labels, is a promising approach towards combatting the robustness-accuracy tradeoff. Indeed, we could *simultaneously* improve both the clean and robust accuracy of some well known methods such as TRADES and HAT, while achieving a well-balanced accuracy distribution over the classes.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

While a larger-scale empirical evaluation would further strengthen our results, the measurements we presented provide sufficient validation to warrant the exploration of the idea further.

References

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the Knowledge in a Neural Network." In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[2] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. "Towards Deep Learning Models Resistant to Adversarial Attacks." In *International Conference on Learning Representations*, 2018.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Networks." *Commun. ACM*, vol. 63, pp. 139–144, 2020.

[4] Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. "Helper-based Adversarial Training: Reducing Excessive Margin to Achieve a Better Accuracy vs. Robustness Trade-off." In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021.

[5] 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*, pp. 770–778, 2016.

[6] Leslie N. Smith and Nicholas Topin. "Super-convergence: Very Fast Training of Neural Networks Using Large Learning Rates." In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, SPIE, May 2019, pp. 369–386. doi: 10.1117/12.2520589.

[7] 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*, 2014.

[8] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. "Adversarial Logit Pairing." *arXiv preprint*, vol. arXiv:1803.06373, 2018.

[9] Hongyang Zhang, Yizhe Yu, Jian Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. "Theoretically Principled Trade-off Between Robustness and Accuracy." In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[10] Jianing Zhu, Jiangchao Yao, Bo Han, Jingfeng Zhang, Tongliang Liu, Gang Niu, Jingren Zhou, Jianliang Xu, and Hongxia Yang. "Reliable Adversarial Distillation with Unreliable Teachers." *arXiv preprint*, vol. arXiv:2106.04928, 2021.

[11] Bojia Zi, Shihao Zhao, Xingjun Ma, and Yu-Gang Jiang. "Revisiting Adversarial Robustness Distillation: Robust Soft Labels Make Student Better." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16443–16452, 2021.

[12] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images." Technical report, 2009.

[13] Francesco Croce and Matthias Hein. "Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks." In *ICML*, 2020.

[14] Liam Rice, Eric Wong, and Zico Kolter. "Overfitting in Adversarially Robust Deep Learning." In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[15] Yurii E. Nesterov. "A Method for Solving the Convex Programming Problem with Convergence Rate $O(1/k^2)$." *Doklady Akademii Nauk SSSR*, 1983.

[16] L. van der Maaten and G. Hinton. "Visualizing data using t-SNE". *Journal of Machine Learning Research*, 9(86): 2579–2605, 2008.