On Fragile Features and Batch Normalization in Adversarial Training

Nils Philipp Walter David Stutz Bernt Schiele
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken
{nwalter,david.stutz,schiele}@mpi-inf.mpg.de

Abstract

Modern deep learning architecture utilize batch normalization (BN) *to stabilize training and improve accuracy. It has been shown that the BN layers alone are surprisingly expressive. In the context of robustness against adversarial examples, however, BN is argued to increase vulnerability. That is, BN helps to learn* fragile *features. Nevertheless, BN is still used in adversarial training, which is the de-facto standard to learn* robust *features. In order to shed light on the role of BN in adversarial training, we investigate to what extent the expressiveness of BN can be used to "robustify" fragile features in comparison to random features. On CIFAR10, we find that adversarially fine-tuning just the BN layers can result in non-trivial adversarial robustness. Adversarially training* only *the BN layers from scratch, in contrast, is not able to convey meaningful adversarial robustness. Our results indicate that fragile features can be used to learn models with moderate adversarial robustness, while random features cannot.*

1. Introduction

Deep Neural Networks (DNNs) have set the state-of-the-art for many tasks in computer vision. Almost all DNN architectures make use of *batch normalization (BN)* [8] to stabilize the training procedure. In particular, BN plays an important role in being able to train particularly deep architectures such as ResNets [5]. Besides that, BN layers have been shown to be surprisingly expressive compared to convolution and fully connected layers: [3] showed that only training the BN layers is sufficient to achieve non-trivial accuracy for very deep ResNets. The fact that BN-layers are equipped with significantly fewer parameters makes the results particularly surprising.

On the other hand, BN has also been argued to increase vulnerability against adversarial examples [4], imperceptibly perturbed images causing mis-classification [15]. The work in [17] suggests that the statistics of the BN layers could be responsible not only for adversarial vulnerability but also for the reduced (clean) accuracy frequently observed, due to the fact that the activation statistics of clean examples and adversarial examples strongly differ [10, 18]. Beyond BN, it is argued [7] that adversarial examples are a consequence of so called *fragile* features. A fragile feature is believed to pick up spurious correlations, that are exploited by the attacker to craft adversarial examples. The most common way to tackle adversarial vulnerability is adversarial training, which learns *robust* features and has been shown to ignore spurious correlations. However, [13] also shows that it is possible to extract robust networks from normally trained ones. This indicates that there are robust features "hidden" among fragile ones.

Complementary to these works, we investigate how adversarial *fine-tuning* of the BN layers can be used to improve adversarial robustness. For that we adversarially tune the BN parameters (and statistics) starting with a normally trained, non-robust base model. On CIFAR10 [9], fine-tuning statistics *and* parameters leads to non-trivial adversarial robustness at the cost of reduced clean accuracy. These experiments indicate that training just the BN parameters allows to utilize fragile features for adversarial robustness – at least to some extent. As no robustness can be achieved when training *only* the BN layers (from scratch), random feature are shown to be useless for robustness. With these experiments, our work supports existing evidence that there exist robust features among fragile ones [13]. However, adversarial training seems to learn *additional* robust features not learned using standard training.

2. Related Work

Adversarial Training: Following [11], adversarial examples can be obtained by maximizing cross-entropy loss, i.e., $\max_{\delta\in\mathcal{S}} L(f(x+\delta), y)$ where L computes the loss between the classifier's output $f(x+\delta)$ and the ground truth label y. Then, adversarial training can be formulated as a min-max learning problem $\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\delta\in\mathcal{S}} L(f_\theta(x+\delta), y)\right]$. For evaluation, ensembles of different attacks has become standard [1].

Fragile and Robust Features: [16] suggests that recent classifiers have to rely on so called *fragile* features to obtain high accuracy. However, these spurious correlations can be exploited by adversarial examples. Thus, improving

Table 1. Overview of training configurations: We consider fine-tuning only the BN statistics (BN STATS), only the BN parameters (BN ONLY PARAMS), or the statistics *and* parameters (BN PARAMS). Additionally, we consider training logits and the first convolutional layer. We also consider normally/adversarially training *only* the BN layers from scratch (BN ONLY).

	Model	all	BN params	BN stats	%params
From scratch	NORMAL/ ADV	✓			100.00
	+BN ONLY	✗	✗	✓	0.15
Fine-tune	BN STATS	✗	✗	✓	(0.15)
	BN PARAMS	✗	✓	✓	0.15

adversarial robustness naturally leads to reduced (clean) accuracy [7, 14]. While [13] shows that robust sub-networks exist within normal networks, it remains largely unclear to *what extent* robust features replace or built-upon fragile features to improve adversarial robustness.

Batch Normalization (BN): It is still poorly understood how exactly BN helps training [12]. Nevertheless, BN is known to be very expressive itself [3]: training *only* the BN parameters (and updating the statistics) is sufficient to obtain non-trivial accuracy. Moreover, units/channels with small BN parameters can be removed without affecting accuracy, indicating that BN learns to "turn off" specific (random) features. In the context of adversarial robustness, BN is argued to increase vulnerability [4] and the statistics are known to be very different when training on adversarial examples [17]. We study how fragile features are utilized in adversarial fine-tuning in comparison to random features.

3. Fragile Features and Batch Normalization in Adversarial Training

We aim to understand to what extent non-trivial robustness is possible based on fragile features compared to random features. To this end, we adversarially fine-tune normally trained models and evaluate robustness compared to adversarially training the BN (and only the BN) layers from scratch. We briefly include the high-level idea of our experiments in Section 3.1, before discussing experimental setup in Section 3.2 and presenting our results in Section 3.3

3.1. Methodology

This study is mostly based on two observations: First, [3] showed that training only the BN parameters (and statistics) is sufficient to achieve decent accuracy. Furthermore they argued that BN learns to identify and "turn off" useless random features. Second, it is argued that adversarial vulnerability is caused by *fragile features* – spurious correlations that are useful for accuracy but can be exploited by an adversary. Note that fragile features are not restricted to the image domain as suggested in [16] because they propagate into deeper layers when used.

Table 2. Main quantitative results for adversarial fine-tuning of BN layers: following the naming of Table 1, we report clean error (in %) and robust error (against AutoAttack, in %) on test examples for L_∞ attacks with $\epsilon = {}^{8}\!/_{255}$. The baseline ADV model improves robust error significantly, at the cost of increased clean error. While BN STATS and ADV BN ONLY are unable to obtain non-trivial robustness, BN PARAMS reduces robust error significantly, even if not matching ADV. This indicates that robustness with fragile features is possible, but limited.

Model	Test error	Robust test error
NORMAL	4.11	99.8
+BN ONLY	46.11	99.7
+BN ONLY + log + conv1	35.89	99.6
ADV	17.67	58.4
+BN ONLY	43.33	98.0
+BN ONLY + log + conv1	35.22	98.1
BN STATS (no reset)	6.44	99.9
BN STATS (reset)	6.33	99.6
BN ONLY PARAMS	77.11	85.3
BN PARAMS	29.67	70.5
BN PARAMS + log + conv1	26.89	68.2

In order to quantify to what degree these fragile features prevent adversarial robustness, we conduct two main experiments: We *fine-tune* a normally trained model using adversarial training [11] by only updating the BN parameters (and statistics). This means all parameters except the BN parameters are frozen. The motivation is that adversarial training will try to avoid or deactivate non-robust, i.e., fragile, features in order to improve adversarial robustness. We emphasize that for, e.g., a ResNet-20, the BN parameters only make up 0.15% of the parameters. Then, we also consider training only the BN parameters (and statistics) *from scratch* using adversarial training. For standard accuracy, random features are shown to be helpful. Thus, this acts as a baseline to check whether adversarial training of only the BN parameters on top of fragile features works better than on random ones.

3.2. Experimental Setup

For simplicity, we use ResNet-20 [5] as a base model. Similar to [3], we place the BN layers before the activation layer, which leads to better performance according to [6]. All our experiments are conducted on the CIFAR-10 dataset [9]. For the (normally trained) baseline model, we also employ AutoAugment [2] to improve accuracy.

For adversarial fine-tuning, we use 10 iterations of PGD with a L_∞ perturbation budget of $\epsilon = {}^{8}\!/_{255}$. In contrast to the baseline model, we do not use AutoAugment, but rather follow related work and employ random crops and flips only [11]. We note that accurate classifiers are argued to rely on spurious correlations, justifying our use of AutoAugment for the baseline model. For adversarial training, however, AutoAugment is not reported to improve robustness. At test

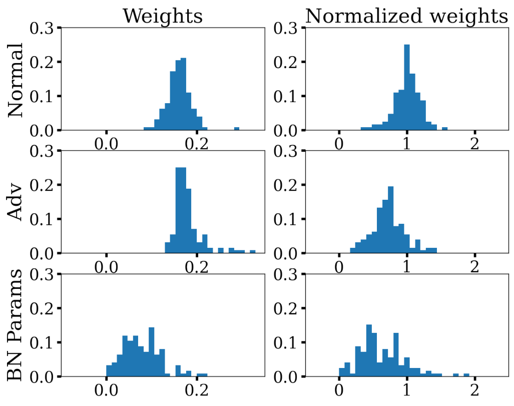

Figure 1. We plot histograms of the BN parameters (8th layer) and their normalized versions according to Equation 1 for NORMAL, ADV and BN PARAMS. On average, BN PARAMS reduces the normalized weights m compared to the baseline models. This indicates that robustness is achieved by "turning off" a non-significant portion of the fragile features.

time, we use Autoattack [1] for robustness evaluation on the first 1000 test examples.

We report results for several training configurations, see Table 1: Given a normally trained baseline model (NORMAL), we adversarially fine-tune it by only updating the BN statistics (BN STATS) or both the statistics and parameters (BN PARAMS). In the latter case, we might also learn the logit or first convolutional layer. As baseline, we consider a adversarially trained model (ADV). Finally, we follow [3] and consider adversarially training only the BN layers (from scratch, ADV + BN ONLY).

3.3. Results

We report our main results in Table 2. We first consider only fitting the BN statistics *or* parameters, before considering to update both, potentially with the logit and first convolutional layer. This is then compared to adversarially training BN (and only BN) from scratch. This allows us to draw conclusions about the usefulness of fragile and random features for adversarial robustness. As shown in Table 2, our normally trained baseline (NORMAL) achieves a clean error of 4.11%, but is not robust against adversarial examples (robust error 99.8%). The adversarially trained baseline (ADV) achieves 17.67% clean error and improves robustness with 58.4% robust error. This reflects the commonly observed robustness-accuracy trade-off.

Importance of BN Statistics: As a first experiment we (adversarially) fine-tune *only* the BN-statistics (i.e., μ and σ). Note that there are as many statistics as BN parameters. The statistics are updated by simply forwarding adversarial examples. While clean test error increases slightly, robust test error does *not* improve. We confirmed this for smaller values of ϵ, as well. Also, the cross-entropy loss on adversarial examples does not change significantly (measured using PGD as AutoAttack does *not* maximize loss). *Not* updating the BN statistics and just learning the BN parameters, in contrast, improves adversarial robustness slightly, from 99.8 to 85.3% robust test error. However, clean test error is nearly as high. These results show that neither statistics nor parameters alone are capable of obtaining meaningful, non-trivial adversarial robustness. We also did not find any recognizable difference in the statistics or parameter distributions across layers. Moreover, learning the logit layer, to account for the updated statistics, did not help and results do not change when resetting the statistics before forwarding adversarial examples.

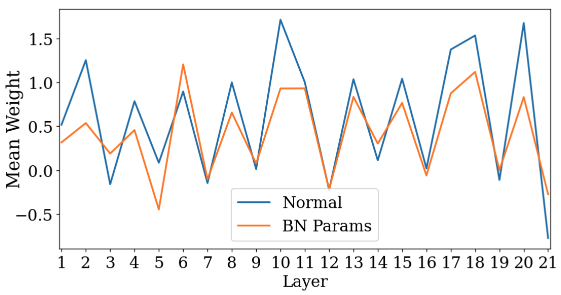

Figure 2. We plot the mean of the normalized weights per layer. On average, the yellow line is (BN PARAMS) below the blue line (NORMAL), which indicates that adversarially fine-tuning the BN-layers tilts the normalized weights to be more negative.

Non-Trivial Robustness: When adversarially fine-tune BN statistics *and* parameters, in contrast, we are able to obtain non-trivial robust test error of 70.5%. Again, the statistics are automatically updated through forward passes, while the parameters are trained using stochastic gradient descent. At the same time, the clean error increases significantly from 4.11% to 30.78%. This indicates that the combination of updated statistics and adversarially learned parameters makes the difference. To understand how the BN statistics and parameters change, we examined histogram plots as depicted in Figure 1. The distribution of the BN parameters, for examples, becomes slightly more concentrated around zero across most layers. However, considering the BN parameters only, without the statistics can be misleading, as the histogram of the variance collapses also. Therefore, when analyzing the BN transformation, we view BN as *one* affine tranformation of the form $f(z) = mz + b$. A simple rearrangement yields:

$$\mathbf{BN}(z_i) = \left(\frac{\gamma_i}{\sqrt{\sigma_i^2 + \epsilon}}\right) z_i + \left(-\frac{\gamma_i \mu_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta_i\right). \quad (1)$$

In the following we refer to $m = \gamma_i / \sqrt{\sigma_i^2 + \epsilon}$ as *normalized* weight and $b = -\gamma_i \mu_i / \sqrt{\sigma_i^2 + \epsilon} + \beta_i$ as *normalized* bias. The histograms of m for NORMAL, ADV and BN PARAMS can be found in Figure 1. The plots indicate that, compared to ADV, the normalized weights of BN PARAMS have a lower mean in most layers. In order to make this result crisp, we explicitly computed the mean of the normalized

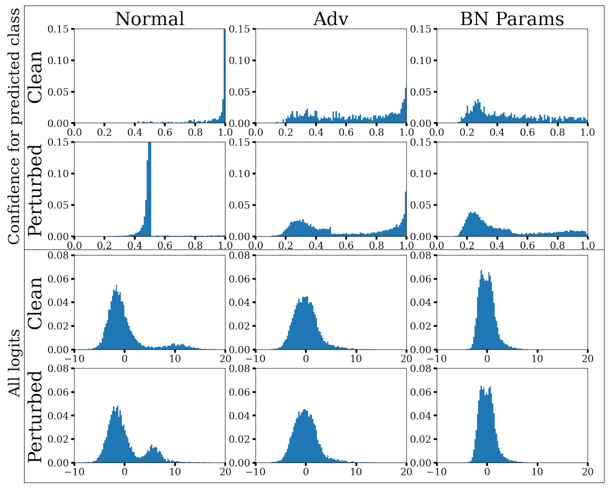

Figure 3. We plot confidence and logit histograms for NORMAL, ADV and BN PARAMS showing that BN PARAMS is able to follow the behavior of ADV to a large extent, both on clean and adversarial examples.

weights per layer, cf. Figure 2. The average difference between the mean of the normalized weights is -0.177. This is a strong indicator that BN PARAMS has to "turn off" part of the normally trained and thus fragile features in order to improve robustness. While the BN statistics alone are not capable of achieving this, the BN parameters can be learned to remove specific features (i.e., channels for convolutional layers). Additionally training the logit and first convolutional layer does not improve adversarial robustness significantly. Finally, Figure 3 shows that the logit distributions on clean and adversarial examples also starts to resemble those observed for ADV.

Adversarially Training BN Only: Following the idea of [3], we also conducted experiments with only training the BN layers from scratch, but adversarially (ADV + BN ONLY). This means that the convolutional and fully-connected layers are initialized with random features and the BN layers attempt to utilize these features to obtain adversarial robustness. Table 2 shows that we are unable to obtain non-trivial adversarial robustness based on random features. Specifically, ADV + BN ONLY achieves a clean error of 43.44% (significantly worse than BN PARAMS), which leads to the conclusion that random features do not contribute to robustness.

4. Discussion and Conclusion

Overall, our experiments allow to draw an interesting conclusion: while random features are not useful to learn robust classifiers, fragile features allow to achieve non-trivial robustness. We suspect this to be possible by ignoring some fragile features. Even though the adversarial robustness does not match the adversarially trained baseline, this result is significant because it was largely believed that fragile features are inherently not robust. However, the remaining robustness gap also shows that new (robust) features are necessary for proper adversarial robustness. These can only be learned using adversarial training from scratch.

References

[1] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, 2020. 1, 3

[2] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data, 2019. 2

[3] Jonathan Frankle, David J. Schwab, and Ari S. Morcos. Training batchnorm and only batchnorm: On the expressive power of random features in cnns, 2021. 1, 2, 3, 4

[4] Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W. Taylor. Batch normalization is a cause of adversarial vulnerability. 2019. 1, 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks, 2016. 2

[7] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features, 2019. 1, 2

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015. 1

[9] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 1, 2

[10] Aishan Liu, Shiyu Tang, Xianglong Liu, Xinyun Chen, Lei Huang, Zhuozhuo Tu, Dawn Song, and Dacheng Tao. Towards defending multiple adversarial perturbations via gated batch normalization. *CoRR*, abs/2012.01654, 2020. 1

[11] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018. 1, 2

[12] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *NeurIPS*, 2018. 2

[13] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Hydra: Pruning adversarially robust neural networks, 2020. 1, 2

[14] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. *CVPR*, 2019. 2

[15] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. 1

[16] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy, 2019. 1, 2

[17] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L. Yuille, and Quoc V. Le. Adversarial examples improve image recognition. *arXiv.org*, abs/1911.09665. 1, 2

[18] Cihang Xie and Alan L. Yuille. Intriguing properties of adversarial training. *CoRR*, abs/1906.03787. 1