Generating Adversarial Samples in Mini-Batches May Be Detrimental To Adversarial Robustness

Timothy Redgrave, Colton Crum
Department of Computer Science and Engineering University of Notre Dame
Notre Dame IN 46556, USA
tredgrav@nd.edu,ccrum@nd.edu

Abstract

Neural networks have been proven to be both highly effective within computer vision, and highly vulnerable to adversarial attacks. Consequently, as the use of neural networks increases due to their unrivaled performance, so too does the threat posed by adversarial attacks. In this work, we build towards addressing the challenge of adversarial robustness by exploring the relationship between the mini-batch size used during adversarial sample generation and the strength of the adversarial samples produced. We demonstrate that an increase in mini-batch size results in a decrease in the efficacy of the samples produced, and we draw connections between these observations and the phenomenon of vanishing gradients. Next, we formulate loss functions such that adversarial sample strength is not degraded by mini-batch size. Our findings highlight a potential risk for underestimating the true (practical) strength of adversarial attacks, and a risk of overestimating a model's robustness. We share our codes to let others replicate our experiments and to facilitate further exploration of the connections between batch size and adversarial sample strength.

1. Introduction

Advances within neural networks have allowed them to achieve unprecedented levels of success across a number of computer vision tasks [12, 20]. This in turn has led to their adoption across a wide range of applications, including within safety-critical systems such as facial recognition [10] and autonomous vehicles [15]. However, despite the impressive performance and remarkable potential displayed by neural networks, researchers have shown that these systems are highly susceptible to adversarial attacks - inputs which have been specifically modified in order to fool or mislead their targets into making incorrect decisions [4,23]. Following this discovery, there has been a tremendous amount of research focused on developing methods which can be used to effectively train and assess adversarially robust models, leading to a number of developments on both fronts.

On the side of adversarial assessment, many advances have come from the development of newer and stronger attacks which represent a greater range of threat models. Early adversarial attacks focused on performing white-box attacks which leveraged gradient-based information in a relatively straightforward manner [11], but more recent works have led to the development of stronger white-box attacks through iterative methods [5, 17], strong and efficient black-box attacks [1, 7], and even attacks which remain effective when deployed in a physical setting [3, 16]. Through the creation of additional and stronger attacks, it is now possible to test models against a larger amount of theoretical threat models, thereby allowing for a more thorough assessment of the vulnerabilities of these neural networks.

Within the realm of adversarial robustness training, there have also been a number of significant developments. These include the development of principled formulations for adversarial training [17], methods which allow for more efficient and easier implementations of adversarial training [21, 25], and techniques which incorporate strategies such as smoothing in order to boost the efficacy of existing training methods [6].

However, despite the significant advances displayed in both adversarial training and testing, gaps still remain on both fronts. Often existing training techniques are evaluated against iterative white-box attacks which make use of the gradient, which can lead to an overestimation of model robustness through obfuscated gradients [2, 18]. Likewise, it is difficult to assess the strength of novel adversarial attacks as this requires examining their efficacy against both baseline and adversarially trained models. These difficulties highlight the continued need to ensure not only the efficiency, but also the validity of claims made with regards to adversarial robustness.

To this end, we identify a potential obstruction on the path to adversarial robustness - the generation of adver-

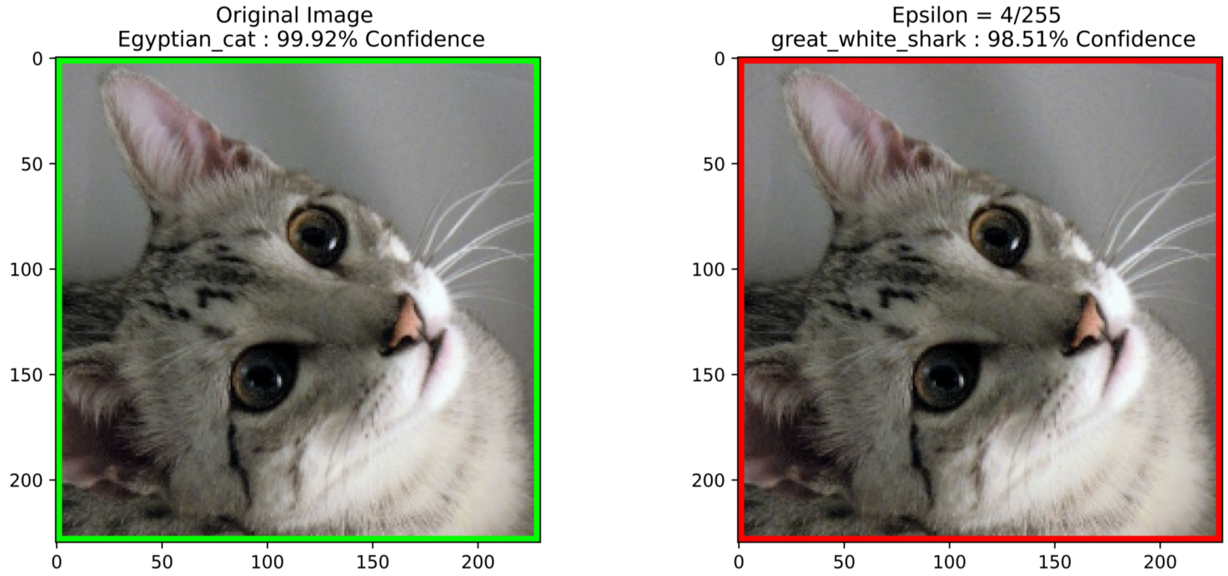

Original Image
Egyptian_cat : 99.92% Confidence
Epsilon = 4/255
great_white_shark : 98.51% Confidence
0
50
100
150
200

Figure 1. The image above shows the vulnerability of neural networks to adversarial attacks. Perturbing the input image by an ϵ of just four pixel intensity levels (out of 255) within the L_∞ norm, an amount practically imperceptible to humans, is enough to cause the model to change the classification of the image from Egyptian cat to great white shark.

sarial samples within large mini-batches. In this work we conduct experiments which demonstrate that the creation of adversarial samples within a batched setting leads to a degradation in the strength of the samples created. We attribute this effect to the reduction of the loss over a mini-batch to the mean loss, a process which results in vanishing gradients. We show that without this reduction, the negative effect of increased batch size is removed, and we discuss how this information can be leveraged moving forward. While our findings are relatively straightforward and align with an intuitive understanding of the issue, we are unaware of any previous works which explicitly address these concerns, and we argue that it is essential to highlight any potential mechanisms which may lead to a reduction in adversarial strength which could in turn lead to an underdevelopment or over-assessment of model robustness. The code used to generate the results for our project may be found here: https://github.com/AdvCVPR23/Adversarial-Mini-Batching.

2. Preliminaries

2.1. Cross-Entropy Loss

Cross-entropy loss is the most commonly used loss function when it comes to the supervised training of neural networks for classification tasks, and it plays a prominent role within a number of white-box adversarial attacks. For our purposes, we will consider the formulation for the cross-entropy loss for a mini-batch to be

$$CE = -\frac{1}{N}\sum_{i=1}^{N} c_n \log\left(\frac{e^{f(x_n)_{c_n}}}{\sum_{c=1}^{C} e^{f(x_n)_{c_j}}}\right) \tag{1}$$

where N represents the number of samples in the mini-batch, C represents the number of classes within the given classification problem, c_n represents the class label for a given sample, and $f(x_n)$ represents the unnormalized logits of a given classifier f and a sample x_n (thereby making the $\frac{e^{f(x_n)_{c_n}}}{\sum_{c=1}^{C} e^{f(x_n)_{c_j}}}$ term represent the softmax score for a given class c_n). This formulation aligns with the default formulations given for cross-entropy loss within both TensorFlow and PyTorch, and for our purposes, it is most important to note the $\frac{1}{N}$ term which gives us the mean loss rather than the total loss.

2.2. Fast Gradient Method

Introduced by Goodfellow *et al.* [11], the Fast Gradient Method (FGM) is one of the earliest and simplest methods that can be used to generate adversarial samples (within the original paper [11] the attack is applied only within the L_∞ norm and is called the Fast Gradient Sign Method (FGSM), however, when applied to other norms the sign operation is no longer used). FGM creates adversarial images in the following manner: first, a base image that is to be perturbed is fed through the model that is being attacked. The model's outputs are then used in conjunction with a given loss function in order to calculate the model's loss with respect to that particular image. This loss is then back-propagated to the input, wherein the resultant gradients are added to the image to create the adversarial image. In the final step of FGM, this adversarial image is clipped such that it is within some fixed distance ϵ of the original image (the distance ϵ will be dependent on what norm the target is operating within). Mathematically, this idea can be captured in the following equations:

$$x_{adv} = x + \epsilon \cdot \frac{\nabla_x L(f(x), y)}{||\nabla_x L(f(x), y)||_2} \tag{2}$$

for the L_2 norm and

$$x_{adv} = x + \epsilon \cdot \text{Sign}(\nabla_x L(f(x), y)) \tag{3}$$

for the L_∞ norm respectively. Here x represents a clean or unperturbed sample, $\nabla_x L(f(x), y)$ represents the gradient of the loss with to the input x and some given set of labels y (for our purposes, we consider cross-entropy loss as described in equation 1), and x_{adv} represents the newly constructed adversarial sample.

By modern standards, FGM is considered a simple or weak adversarial attack as it is a single-step method and only calculates the gradients with respect to the base image once and does not perform any randomization. However, despite the simplicity of this method, it remains highly effective against models that are trained in a non-robust manner, can be executed relatively quickly due to its single-step nature, and has seen surprising levels of success in some modern adversarial training regimes [25]. For these reasons we use FGM to represent a weak attack to test the effect of mini-batch size against.

2.3. Projected Gradient Descent

The second attack method we use within our assessment is Projected Gradient Descent (PGD) [17, 24]. PGD operates by first injecting some level of noise into a given base image. PGD then iteratively performs the FGM attack using this modified base image in order to produce a stronger adversarial sample. This can be understood mathematically as

$$x_{adv}^{t+1} = \text{FGM}(x_{adv}^{t}), x_{adv}^{1} = \text{FGM}(x + noise) \tag{4}$$

with $noise$ representing any type of noise bounded within the given ϵ and x_{adv}^t representing the adversarial sample after t iterations of the FGM attack. The iterative nature of PGD combined with the random initialization allows for the construction of far stronger adversarial samples than single-step methods such as FGM, and PGD has been claimed to be the strongest first-order attack as a result [17]. Based on this, we use PGD to test the effect of mini-batch size on a strong adversary.

3. Experimental Design

3.1. Experiments

Within our work we perform four sets of experiments in order to elucidate the effect of mini-batch size within adversarial sample generation.

- **Baseline:** Our first set of experiments involves generating samples using the loss formulation given within equation 1. The results from these runs demonstrate the effect of mini-batch size on adversarial strength when using the default formulations for loss accumulation over mini-batches (as reflected in the implementations for both TensorFlow and Pytorch).

- **Batch Correction:** The second set of experiments performed removes the term $\frac{1}{N}$ from equation 1, resulting in:

  $$CE = -\sum_{i=1}^{N} c_n \log(\frac{e^{f(x_n)_{c_n}}}{\sum_{c=1}^{C} e^{f(x_n)_{c_j}}}) \tag{5}$$

  These results allow us to directly attribute the cause of adversarial strength degradation to the commonly used reduction over mini-batch size (i.e., the use of the mean loss rather than total loss). It is worth noting that the computational complexities of equations 1 and 5 are equivalent and thus this formulation does not incur any additional overhead in terms of memory usage or computation time.

- **Mixed Precision:** Our third set of experiments repeat our experiments using our original loss formulation given in equation 1 while using 16-bit floating point precision instead of 32-bit floating point precision for our tensors. This allows us to more concretely connect the degradation in adversarial attack strength to the phenomenon of vanishing gradients.

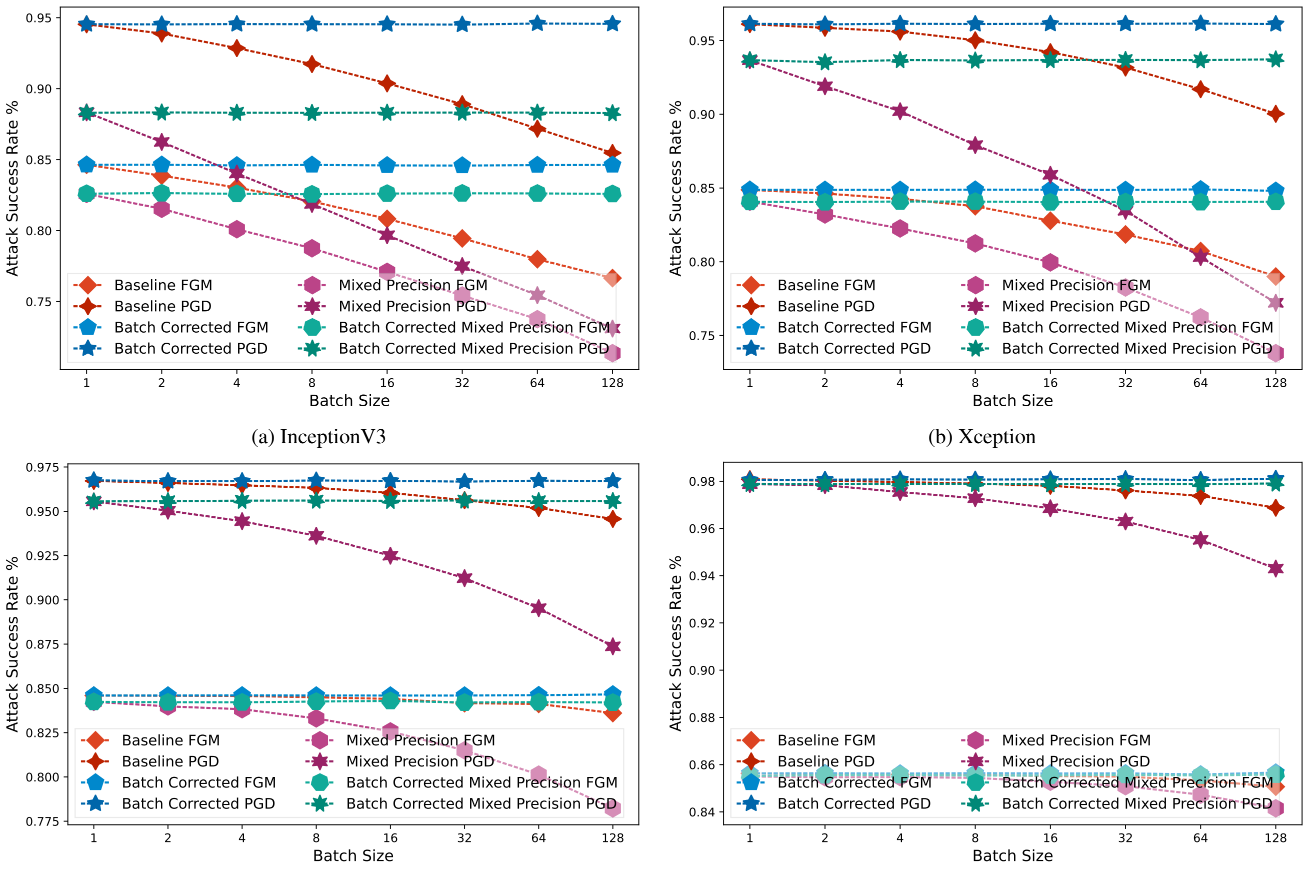
- **Batch Corrected Mixed Precision:** Our final set of experiments focuses on performing attacks using equation 5 and 16-bit precision floats. These experiments allow us to further confirm our understanding of the mechanisms behind our observations

All experiments are run eight times with mini-batches of 1, 2, 4, 8, 16, 32, 64, and 128 images.

3.2. Attack Parameters

For each experiment we perform FGM and PGD within the Euclidean (L_2) and infinite (L_∞) norms using the parameters given in table 1.

Performing both FGM and PGD allows us to assess the effect of mini-batch size on weak attacks and strong attacks respectively, and performing attacks across both the L_2 and L_∞ norms allows us to assess the effect of mini-batch size across different measures of visual distance.



(a) InceptionV3

(b) Xception

(c) ResNet50

(d) DenseNet121

Figure 2. The graphs above show the detrimental effect of large mini-batch sizes when generating adversarial attacks bounded within the Euclidean norm (L_2). Lines with convex polygons represent FGM attacks, and lines with concave polygons show PGD attacks.

Attack Type	Epsilon	Steps	Step Size
FGM L_2	128/256	1	ϵ
FGM L_∞	8/256	1	ϵ
PGD L_2	128/256	32	$2 \cdot \epsilon/32$
PGD L_∞	8/256	32	$2 \cdot \epsilon/32$

Table 1. Attacks with a strength of $\epsilon = 8/256$ correspond to a strength of 0.031 in the normalized $[0, 1]$ space as used in works such as [2].

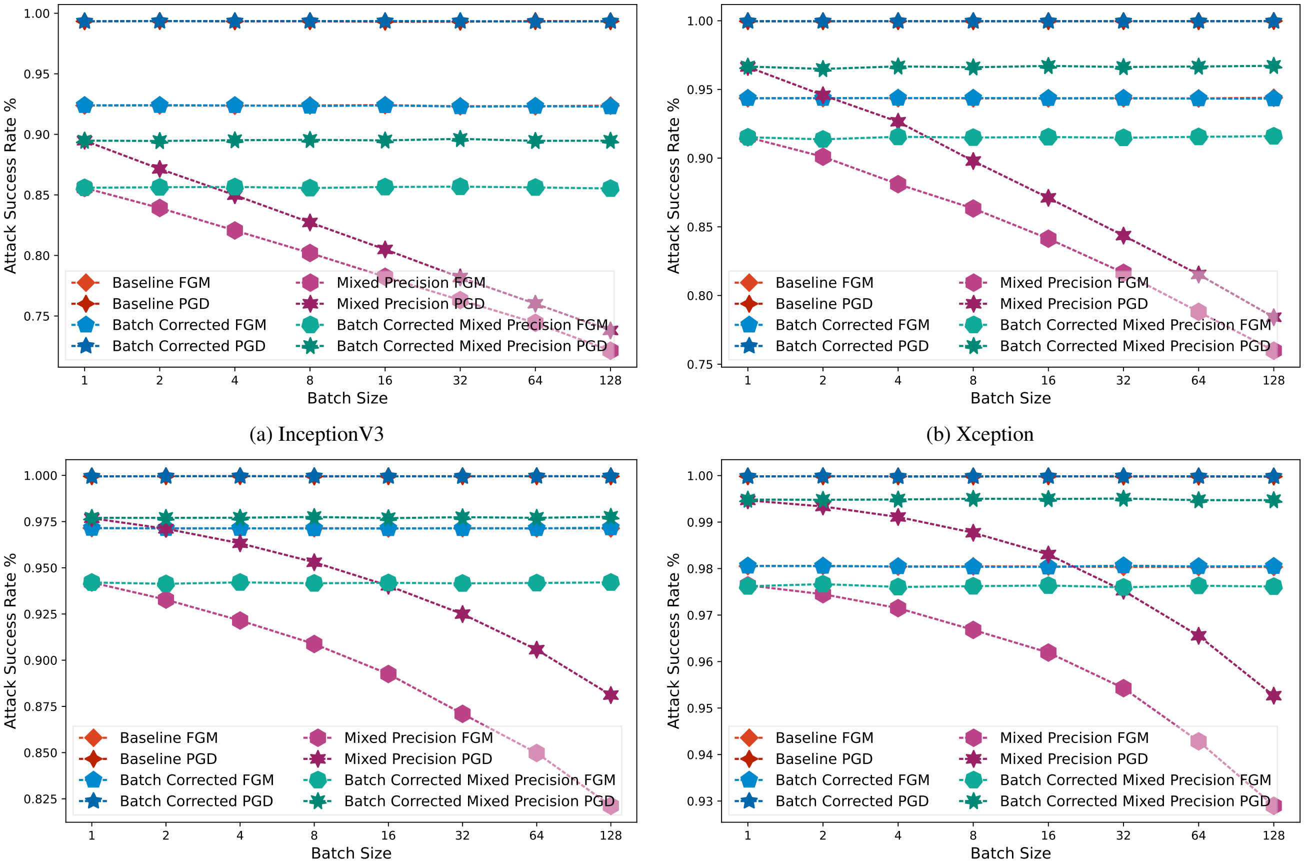
3.3. Dataset and Models

In our experiments we make use of ImageNetV2 [19]. ImageNetV2 is a collection of 10,000 images consisting of 10 sample images for each of the 1000 different classes found within the original ImageNet [9]. While ImageNetV2 was primarily designed as an additional test set for models trained on the original ImageNet, ImageNetV2 [19] forms an ideal basis for testing adversarial image generation as it properly mirrors a real-world threat model wherein attackers may be able to manipulate genuine data which was not available to the model designers during the training or testing of a given model. For our experiments all images are resized to be 224x224.

We ran all experimental setups using four separate model architectures: InceptionV3 [22], Xception [8], ResNet50 [13], and DenseNet121 [14]. Each model was pre-trained on ImageNet [9] and made use of the default TensorFlow weights and preprocessing functions (specific to each model).

3.4. Additional Considerations

In order to limit potential confounding factors within our analysis, we use an Intexl Xeon Gold 6226 CPU paired with an NVIDIA Quadro RTX 6000 GPU for all of our experiments. Additionally, in order to minimize the effect of the random initialization within PGD attacks, we repeat all runs five times and take the average result across the runs.



(a) InceptionV3

(b) Xception

(c) ResNet50

(d) DenseNet121

Figure 3. This figure shows the same set of results as fig. 2 but for attacks bounded within the infinite norm (L_∞).

4. Evaluation and Analysis

4.1. Baseline vs. Batch Correction

The first thing which may be observed when examining the results of our experiments is the universal degradation of adversarial sample strength across attacks and models when using large mini-batches in conjunction with the proposed baseline cross-entropy loss formulation given in equation 1. While this effect is much more pronounced within attacks bounded within the Euclidean norm and for the InceptionV3 and Xception architectures (in contrast with the infinite norm and the ResNet50 and DenseNet121 architectures respectively), the use of larger mini-batch sizes during the generation of adversarial samples consistently leads to a decrease in the number of samples able to successfully fool the given classifier. The most drastic example of this can be seen when we look at the InceptionV3 model. When generating adversarial samples via PGD bound within the Euclidean norm (L_2), the rate of successful attacks against InceptionV3 drops from 94.54% when samples are generated individually, to 85.47% when samples are generated within batches of size 128.

In contrast, when we use our formulation given within equation 5 which calculates the total loss rather than the mean loss, we can see that the penalty to adversarial strength induced by larger batch sizes is alleviated. Under this corrected formulation attacks are able to retain their strength whether they are generating individual samples or they are generating samples in large batches. These findings align well with an intuitive view of the problem; if samples are generated based on the mean loss rather than the total loss, the contribution of each individual sample will decrease as the number of samples considered increases. This in turn will lead to gradients which are smaller in magnitude, and thus will incidentally result in vanishing gradients as larger and larger batch sizes are explored.

4.2. Confirmation through Mixed Precision

These results regarding the relationship between batch size and adversarial sample strength are further supported when we examine our mixed precision results. As can be observed from our findings, regardless of batch size, a decrease in precision leads to a drop in adversarial strength. This is to be expected as by halving our precision, we have a less granular view of our loss surface, and thus the descent direction given by the gradient will be less exact. The more interesting observation, however, arises when we examine our baseline mixed precision results and compare them against our batch corrected results. If we do this, we can obverse that while adversarial samples generated in a mixed precision manner may result in a lower degree of adversarial strength for small batch sizes, as batch size increases, batch corrected samples generated in a mixed precision manner often reach a point where they are more effective on average than samples generated with full precision but without batch correction. This highlights the importance of using the sum of the loss over the mini-batch rather than the mean of the loss over the mini-batch when generating adversarial samples.

4.3. Discussion

Given the relative simplicity of our results and how well they align with what may be expected from an intuitive examination of the problem, it is worth asking whether these results are important. To that end, we present a two-fold argument as to the relevance of this work. The first argument comes from a desire to standardize robustness testing and verification. While a number of previous works (such as [2, 5]) have discussed the need to build towards a shared set of baseline testing standards, to our knowledge our work is the first which explicitly discusses how to formulate loss functions such that sample strength is not degraded by mini-batch size during testing. This allows us to take greater advantage of parallelized environments without sacrificing any potential adversarial strength (and as a consequence, over estimating adversarial robustness).

Our second argument stems from a slightly less obvious use case - the generation of adversarial samples during adversarial training. While switching from using the mean loss over a mini-batch to using the total loss over a mini-batch may be a relatively painless adjustment to make during model assessment, this switch is not as straightforward during model training. Mean loss reduction allows for a more direct comparison of models trained using different batch sizes, a smoother approximation of the loss surface, and generally more convenient numerical properties. In contrast, if the reduction of adversarial samples is done by the sum of the losses over a mini-batch, a number of hyper-parameters may need to be adjusted based on the batch size used including step-size, momentum decay, and more. Therefore, we can observe that even while the challenge of correcting for the effect of large batch sizes may be relatively easily addressed within model testing, it is a much more complicated issue to address during model training. Furthermore, while an incorrect formulation of the adversarial loss may lead to an over estimation of robustness if done during model testing, this same mistake may lead to a reduction in true model robustness if made during model training. As such, we assert that it is crucial to explicitly establish the potential consequences of incorrect adversarial training and assessment, and to build our training and evaluation tools such in such a way that these risks are mitigated.

5. Conclusion

In this work we demonstrated the link between large mini-batch sizes and the reduction of adversarial strength within the adversarial samples created. Through testing formulations which used mean loss, total loss, and mixed precision, we were able to isolate the effect of batching within our analysis and demonstrate that while its magnitude may change based on the model or attack used, its negative impact remains present regardless. We then briefly discussed how this may affect models during both training and assessment, and we further affirm the need for standard testing tool and parameters for evaluating model robustness.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII*, 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. 1, 4, 6

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International Conference on Machine Learning*, pages 284–293. PMLR, 2018. 1

[4] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 387–402. Springer, 2013. 1

[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. 1, 6

[6] Jinghui Chen, Yu Cheng, Zhe Gan, Quanquan Gu, and Jingjing Liu. Efficient robust training via backward smoothing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6222–6230, 2022. 1

[7] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1277–1294. IEEE, 2020. 1

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017. 4

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 4

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 1

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015. 1

[13] 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. 4

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 4

[15] Parham M Kebria, Abbas Khosravi, Syed Moshfeq Salaken, and Saeid Nahavandi. Deep imitation learning for autonomous vehicles based on convolutional neural networks. *IEEE/CAA Journal of Automatica Sinica*, 7(1):82–95, 2019. 1

[16] Stepan Komkov and Aleksandr Petiushko. Advhat: Real-world adversarial attack on arcface face id system. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 819–826. IEEE, 2021. 1

[17] 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. 1, 3

[18] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017. 1

[19] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019. 4

[20] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 1

[21] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1

[22] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 4

[23] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014. 1

[24] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2018. 3

[25] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2019. 1, 3