FAST IS BETTER THAN FREE: REVISITING ADVERSARIAL TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial training, a method for learning robust deep networks, is typically assumed to be more expensive than traditional training due to the necessity of constructing adversarial examples via a first-order method like projected gradient descent (PGD). In this paper, we make the surprising discovery that it is possible to train empirically robust models using a much weaker and cheaper adversary, an approach that was previously believed to be ineffective, rendering the method no more costly than standard training in practice. Specifically, we show that adversarial training with the fast gradient sign method (FGSM), when combined with random initialization, is as effective as PGD-based training but has significantly lower cost. Furthermore we show that FGSM adversarial training can be further accelerated by using standard techniques for efficient training of deep networks, allowing us to learn a robust CIFAR10 classifier with 45% robust accuracy at $\epsilon = 8/255$ in 6 minutes, and a robust ImageNet classifier with 43% robust accuracy at $\epsilon = 2/255$ in 12 hours, in comparison to past work based on "free" adversarial training which took 10 and 50 hours to reach the same respective thresholds.

1 INTRODUCTION

Although deep network architectures continue to be successful in a wide range of applications, the problem of learning *robust* deep networks remains an active area of research. In particular, safety and security focused applications are concerned about robustness to adversarial examples, data points which have been adversarially perturbed to fool a model (Szegedy et al., 2013). The goal here is to learn a model which is not only accurate on the data, but also accurate on adversarially perturbed versions of the data. To this end, a number of defenses have been proposed to mitigate the problem and improve the robustness of deep networks, with some of the most reliable being certified defenses and adversarial training. However, both of these approaches come at a non-trivial, additional computational cost, often increasing training time by an order of magnitude over standard training. This has slowed progress in researching robustness in deep networks, due to the computational difficulty in scaling to much larger networks and the inability to rapidly train models when experimenting with new ideas. In response to this difficulty, there has been a recent surge in work that tries to to reduce the complexity of generating an adversarial example, which forms the bulk of the additional computation in adversarial training (Zhang et al., 2019; Shafahi et al., 2019). While these works present reasonable improvements to the runtime of adversarial training, they are still significantly slower than standard training, which has been greatly accelerated due to competitions for optimizing both the speed and cost of training (Coleman et al., 2017).

In this work, we argue that adversarial training, in fact, is not as hard as has been suggested by this past line of work. In particular, we revisit one of the the *first* proposed methods for adversarial training, using the Fast Gradient Sign Method (FGSM) to add adversarial examples to the training process (Goodfellow et al., 2014). Although this approach has long been dismissed as ineffective, we show that by simply introducing random initialization points, FGSM-based training is *as effective as projected gradient descent based training* while being an order of magnitude more efficient. Moreover, FGSM adversarial training (and to a lesser extent, other adversarial training methods) can be drastically accelerated using standard techniques for efficient training of deep networks, including e.g. cyclic learning rates (Smith & Topin, 2018), mixed-precision training (Micikevicius et al., 2017), and other similar techniques. The method has extremely few free parameters to tune, and can be easily adapted to most training procedures.

The end result is that, with these approaches, we are able to train (empirically) robust classifiers far faster than in previous work. Specifically, we train an ℓ_∞ robust CIFAR10 model to 45% accuracy at $\epsilon = 8/255$ (the same level attained in previous work) in *6 minutes*; previous papers reported times of 80 hours for PGD-based training (Madry et al., 2017) and 10 hours for the more recent "free" adversarial training method (Shafahi et al., 2019). Similarly, we train an ℓ_∞ robust ImageNet classifier to 43% top-1 accuracy at $\epsilon = 2/255$ (again matching previous results) in 12 hours of training (compared to 50 hours in the best reported previous work that we are aware of (Shafahi et al., 2019)). Both of these times roughly match the comparable time for quickly training a standard *non-robust* model to reasonable accuracy. We extensively evaluate these results against *strong PGD-based attacks*, and show that they obtain the same empirical performance as the slower, PGD-based training. Thus, we argue that despite the conventional wisdom, adversarially robust training is not actually more challenging than standard training of deep networks, and can be accomplished with the notoriously weak FGSM attack.

2 Related work

After the discovery of adversarial examples by Szegedy et al. (2013), Goodfellow et al. (2014) proposed the Fast Gradient Sign Method (FGSM) to generate adversarial examples with a single gradient step. This method was used to perturb the inputs to the model before performing backpropagation as an early form of adversarial training. Later, the Basic Iterative Method improved upon FGSM by taking multiple, smaller FGSM steps, ultimately rendering FGSM adversarial training ineffective (Kurakin et al., 2016). This iterative adversarial attack was further strengthened by adding multiple random restarts, and was also incorporated into the adversarial training procedure. These improvements form the basis of what is widely understood today as adversarial training against a projected gradient descent (PGD) adversary, and the resulting method is recognized as an effective approach to learning robust networks (Madry et al., 2017). Since then, the PGD attack and its corresponding adversarial training defense have been augmented with various techniques, such as optimization tricks like momentum to improve the adversary (Dong et al., 2018), combination with other heuristic defenses like matrix estimation (Yang et al., 2019) or logit pairing (Mosbach et al., 2018), and generalization to multiple types of adversarial attacks (Tramèr & Boneh, 2019; Maini et al., 2019).

In addition to adversarial training, a number of other defenses against adversarial attacks have also been proposed. Adversarial defenses span a wide range of methods, such as preprocessing techniques (Guo et al., 2017; Buckman et al., 2018; Song et al., 2017), detection algorithms (Metzen et al., 2017; Feinman et al., 2017; Carlini & Wagner, 2017a), verification and provable defenses (Katz et al., 2017; Sinha et al., 2017; Wong & Kolter, 2017; Raghunathan et al., 2018), and various theoretically motivated heuristics (Xiao et al., 2018; Croce et al., 2018). While certified defenses have been scaled to reasonably sized networks (Wong et al., 2018; Mirman et al., 2018; Gowal et al., 2018; Cohen et al., 2019; Salman et al., 2019), the guarantees don't match the empirical robustness obtained through adversarial training.

With the proposal of many new defense mechanisms, of great concern in the community is the use of strong attacks for evaluating robustness: weak attacks can give a misleading sense of security, and the history of adversarial examples is littered with adversarial defenses (Papernot et al., 2016; Lu et al., 2017; Kannan et al., 2018; Tao et al., 2018) which were ultimately defeated by stronger attacks (Carlini & Wagner, 2016; 2017b; Athalye et al., 2017; Engstrom et al., 2018; Carlini, 2019). This highlights the difficulty of evaluating adversarial robustness, as pointed out by other work which began to defeat proposed defenses en masse (Uesato et al., 2018; Athalye et al., 2018). Since then, several best practices have been proposed to mitigate this problem (Carlini et al., 2019).

Despite the eventual defeat of other adversarial defenses, adversarial training with a PGD adversary remains empirically robust to this day. However, running a strong PGD adversary within an inner loop of training is expensive, and some earlier work in this topic found that taking larger but fewer steps did not always significantly change the resulting robustness of a network (Wang, 2018). To combat the increased computational overhead of the PGD defense, some recent work has looked at regressing the k-step PGD adversary to a variation of its single-step FGSM predecessor called "free" adversarial training, which can be computed with little overhead over standard training by using a single backwards pass to simultaneously update both the model weights and also the input pertur-

Algorithm 1 PGD adversarial training for T epochs, given some radius ϵ, adversarial step size α and N PGD steps and a dataset of size M for a network f_θ

```
for $t = 1 \dots T$ do
  for $i = 1 \dots M$ do
    // Perform PGD adversarial attack
    $\delta = 0$ // or randomly initialized
    for $j = 1 \dots N$ do
      $\delta = \delta + \alpha \cdot \mathrm{sign}(\nabla_\delta \ell(f_\theta(x_i + \delta), y_i))$
      $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
    end for
    $\theta = \theta - \nabla_\theta \ell(f_\theta(x_i + \delta), y_i)$ // Update model weights with some optimizer, e.g. SGD
  end for
end for
```

bation (Shafahi et al., 2019). Finally, when performing a multi-step PGD adversary, it is possible to cut out redundant calculations during backpropagation when computing adversarial examples for additional speedup (Zhang et al., 2019).

Although these improvements are certainly faster than the standard adversarial training procedure, they are not much faster than traditional training methods, and can still take hours to days to compute. On the other hand, top performing training methods from the DAWNBench competition (Coleman et al., 2017) are able to train CIFAR10 and ImageNet architectures to standard benchmark metrics in mere minutes and hours respectively, using only a modest amount of computational resources. Although some of the techniques can be quite problem specific for achieving bleeding-edge performance, more general techniques such as cyclic learning rates (Smith & Topin, 2018) and half-precision computations (Micikevicius et al., 2017) have been quite successful in the top ranking submissions, and can also be useful for adversarial training.

3 Adversarial training overview

Adversarial training is a method for learning networks which are robust to adversarial attacks. Given a network f_θ parameterized by θ, a dataset (x_i, y_i), a loss function ℓ and a threat model Δ, the learning problem is typically cast as the following robust optimization problem,

$$\min_\theta \sum_i \max_{\delta \in \Delta} \ell(f_\theta(x_i + \delta), y_i). \tag{1}$$

A typical choice for a threat model is to take $\Delta = \{\delta : \|\delta\|_\infty \leq \epsilon\}$ for some $\epsilon > 0$. This is the ℓ_∞ threat model used by Madry et al. (2017) and what we consider in this paper. The procedure for adversarial training is to use some adversarial attack to approximate the inner maximization over Δ, followed by some variation of gradient descent on the model parameters θ. For example, one of the earliest versions of adversarial training used the Fast Gradient Sign Method to approximate the inner maximization. This could be seen as a relatively inaccurate approximation of the inner maximization for ℓ_∞ perturbations, and has the following closed form (Goodfellow et al., 2014):

$$\delta^\star = \epsilon \cdot \mathrm{sign}(\nabla_x \ell(f(x), y)). \tag{2}$$

A better approximation of the inner maximization is to take multiple, smaller FGSM steps of size α instead. When the iterate leaves the threat model, it is projected back to the set Δ (for ℓ_∞ perturbations. This is equivalent to clipping δ to the interval $[-\epsilon, \epsilon]$). Since this is only a local approximation of a non-convex function, multiple random restarts within the threat model Δ typically improve the approximation of the inner maximization even further. A combination of all these techniques is known as the PGD adversary (Madry et al., 2017), and its usage in adversarial training is summarized in Algorithm 1.

Note that the number of gradient computations here is proportional to $O(MN)$ in a single epoch, where M is the size of the dataset and N is the number of steps taken by the PGD adversary. This

Algorithm 2 "Free" adversarial training for T epochs, given some radius ϵ, N minibatch replays, and a dataset of size M for a network f_θ

```
$\delta = 0$
// Iterate T/N times to account for minibatch replays and run for T total epochs
for $t = 1 \dots T/N$ do
  for $i = 1 \dots M$ do
    // Perform simultaneous FGSM adversarial attack and model weight updates $T$ times
    for $j = 1 \dots N$ do
      // Compute gradients for perturbation and model weights simultaneously
      $\nabla_\delta, \nabla_\theta = \nabla \ell(f_\theta(x_i + \delta), y_i)$
      $\delta = \delta + \epsilon \cdot \text{sign}(\nabla_\delta)$
      $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
      $\theta = \theta - \nabla_\theta$ // Update model weights with some optimizer, e.g. SGD
    end for
  end for
end for
```

Algorithm 3 FGSM adversarial training for T epochs, given some radius ϵ, N PGD steps, step size α, and a dataset of size M for a network f_θ

```
for $t = 1 \dots T$ do
  for $i = 1 \dots M$ do
    // Perform FGSM adversarial attack
    $\delta = \text{Uniform}(-\epsilon, \epsilon)$
    $\delta = \delta + \alpha \cdot \text{sign}(\nabla_\delta \ell(f_\theta(x_i + \delta), y_i))$
    $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
    $\theta = \theta - \nabla_\theta \ell(f_\theta(x_i + \delta), y_i)$ // Update model weights with some optimizer, e.g. SGD
  end for
end for
```

is N times greater than standard training (which has $O(M)$ gradient computations per epoch), and so adversarial training is typically N times slower than standard training.

3.1 "Free" adversarial training

To get around this slowdown of a factor of N, Shafahi et al. (2019) instead propose "free" adversarial training. This method takes FGSM steps with full step sizes $\alpha = \epsilon$ followed by updating the model weights for N iterations on the same minibatch (also referred to as "minibatch replays"). The algorithm is summarized in Algorithm 2. Note that perturbations are not reset between minibatches. To account for the additional computational cost of minibatch replay, the total number of epochs is reduced by a factor of N to make the total cost equivalent to T epochs of standard training. Although "free" adversarial training is faster than the standard PGD adversarial training, it is not as fast as we'd like: Shafahi et al. (2019) need to run over 200 epochs in over 10 hours to learn a robust CIFAR10 classifier and two days to learn a robust ImageNet classifier.

4 Fast adversarial training

To speed up adversarial training and move towards the state of the art in fast standard training methods, we first highlight the main empirical contribution of the paper: that FGSM adversarial training combined with random initialization is just as effective a defense as PGD-based training. Following this, we discuss several techniques from the DAWNBench competition (Coleman et al., 2017) that are applicable to all adversarial training methods, which reduce the total number of epochs needed for convergence with cyclic learning rates and further speed up computations with mixed-precision arithmetic.

Table 1: Performance of various adversarial training methods on CIFAR10 for $\epsilon = 8/255$

Method	Standard accuracy	PGD ($\epsilon = 8/255$)
FGSM adversarial training		
+ zero init	81.48%	0.00%
+ previous init	85.60%	42.32%
+ random init	84.38%	43.78%
+ $\alpha = 10/255$ step size	83.25%	46.25%
+ $\alpha = 16/255$ step size	85.39%	0.00%
“Free” adversarial training ($m = 8$)	83.80%	46.78%
PGD adversarial training	83.04%	44.56%

4.1 Revisiting FGSM adversarial training

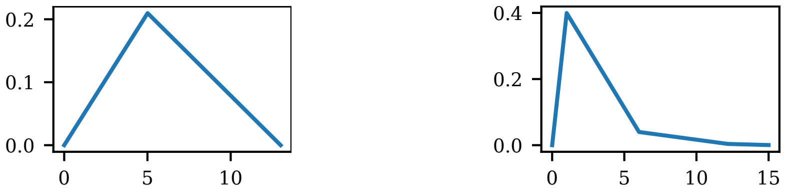
Despite being quite similar to FGSM adversarial training, free adversarial training is empirically robust against PGD attacks whereas the FGSM adversarial training method is believed not to be robust. To analyze why, we identify a key difference between the methods: a property of free adversarial training is that the perturbation from the previous iteration is used as the initial starting point for the next iteration. However, there is little reason to believe that an adversarial perturbation for a previous minibatch is a reasonable starting point for the next minibatch. As a result, we hypothesize that the main benefit comes from simply starting from a non-zero initial perturbation.

In light of this difference, our approach is to use FGSM adversarial training with random initialization for the perturbation, as shown in Algorithm 3. We find that, in contrast to what was previously believed, this simple adjustment to FGSM adversarial training can be used as an effective defense on par with PGD adversarial training. Crucially, we find that starting from a non-zero initial perturbation is the primary driver for success, regardless of the actual initialization. In fact, both starting with the previous minibatch's perturbation or initializing from a uniformly random perturbation allow FGSM adversarial training to succeed at being robust to full-strength PGD adversarial attacks.

To test the effect of initialization in FGSM adversarial training, we train several models to be robust at a radius $\epsilon = 8/255$ on CIFAR10, starting with the most “pure” form of FGSM, which takes steps of size $\alpha = \epsilon$ from a zero-initialized perturbation. The results, given in Table 1, are consistent with the literature, and show that the model trained with zero-initialization is not robust against a PGD adversary. However, surprisingly, simply using a random or previous-minibatch initialization instead of a zero initialization actually results in reasonable robustness levels (with random initialization performing slightly better) that are comparable to both free and PGD adversarial training methods. The adversarial accuracies in Table 1 are calculated using a PGD adversary with 50 iterations, step size $\alpha = 2/255$, and 10 random restarts. Specific optimization parameters used for training these models can be found in Appendix A.

FGSM step size Note that an FGSM step with size $\alpha = \epsilon$ from a non-zero initialization is not guaranteed to lie on the boundary of the ℓ_∞ ball, and so this defense could potentially be seen as too weak. We find that increasing the step size by a factor of 1.25 to $\alpha = 10/255$ further improved the robustness of the model so that it is on par with the best reported result from free adversarial training. However, we also found that forcing the resulting perturbation to lie on the boundary with a step size of $\alpha = 2\epsilon$ resulted in catastrophic overfitting: it does not produce a model robust to adversarial attacks. These two failure modes (starting from a zero-initialized perturbation and generating perturbations at the boundary) may explain why previous attempts at FGSM adversarial training failed.

Computational complexity A second key difference between FGSM and free adversarial training is that the latter uses a single backwards pass to compute gradients for both the perturbation and the model weights while repeating the same minibatch m times in a row, called “minibatch replay”. In comparison, the FGSM adversarial training does not need to repeat minibatches, but needs two backwards passes to compute gradients separately for the perturbation and the model weights. As a



(a) CIFAR10

(b) ImageNet

Figure 1: Cyclic learning rates used for FGSM adversarial training on CIFAR10 and ImageNet over epochs. The ImageNet cyclic schedule is decayed further by a factor of 10 in the second and third phases.

result, the computational complexity for an epoch of FGSM adversarial training is not truly free and is equivalent to two epochs of standard training.

4.2 DAWNBench improvements

Although free adversarial training is of comparable cost per iteration to traditional standard training methods, it is not quite comparable in total cost to more recent advancements in fast methods for standard training. Notably, top submissions to the DAWNBench competition have shown that CIFAR10 and ImageNet classifiers can be trained at significantly quicker times and at much lower cost than traditional training methods. Although some of the submissions can be quite unique in their approaches, we identify two generally applicable techniques which have a significant impact on the convergence rate and computational speed of standard training.

Cyclic learning rate Introduced by Smith (2017) for improving convergence and reducing the amount of tuning required when training networks, a cyclic schedule for a learning rate can drastically reduce the number of epochs required for training deep networks (Smith & Topin, 2018). A simple cyclic learning rate schedules the learning rate linearly from zero, to a maximum learning rate, and back down to zero (examples can be found in Figure 1). Using a cyclic learning rate allows CIFAR10 architectures to converge to benchmark accuracies in tens of epochs instead of hundreds, and is a crucial component of some of the top DAWNBench submissions.

Mixed-precision arithmetic With newer GPU architectures coming with tensor cores specifically built for rapid half-precision calculations, using mixed-precision arithmetic when training deep networks can also provide significant speedups for standard training (Micikevicius et al., 2017). This can drastically reduce the memory utilization, and when tensor cores are available, also reduce runtime. In some DAWNBench submissions, switching to mixed-precision computations was key to achieving fast training while keeping costs low.

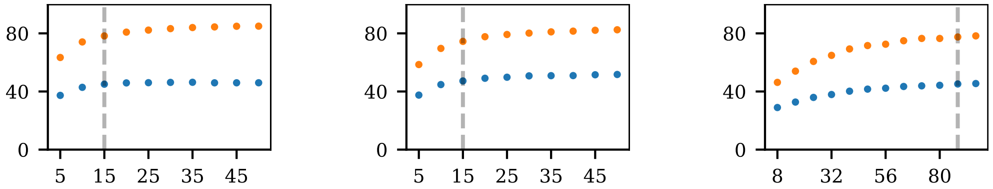
We adopt these two techniques for use in adversarial training, which allows us to drastically reduce the number of training epochs as well as the runtime on GPU infrastructure with tensor cores, while using modest amounts of computational resources. Notably, both of these improvements can be easily applied to existing implementations of adversarial training by adding a few lines of code with very little additional engineering effort, and so are easily accessible by the general research community.

5 Experiments

To demonstrate the effectiveness of FGSM adversarial training with fast training methods, we run a number of experiments on MNIST, CIFAR10, and ImageNet benchmarks. All CIFAR10 experiments in this paper are run on a single GeForce RTX 2080ti using the ResNet18 architecture, and all ImageNet experiments are run on a single machine with four GeForce RTX 2080tis using the ResNet50 architecture (He et al., 2016). Repositories for reproducing the CIFAR10 and ImageNet experiments and the corresponding trained model weights are available at the following anonymous GitHub user account: `https://github.com/anonymous-sushi-armadillo`.

Table 2: Robustness of FGSM and PGD adversarial training on MNIST

Method	Standard accuracy	PGD ($\epsilon = 0.1$)	PGD ($\epsilon = 0.3$)	Verified ($\epsilon = 0.1$)
PGD	99.20%	97.66%	89.90%	96.7%
FGSM	99.20%	97.53%	88.77%	96.8%



(a) FGSM (b) PGD (c) "Free" ($m = 8$)

Figure 2: Performance of models trained on CIFAR10 at $\epsilon = 8/255$ with cyclic learning rates and half precision, given varying numbers of epochs across different adversarial training methods. Each point denotes the average model performance over 3 independent runs, where the x axis denotes the number of epochs N the model was trained for, and the y axis denotes the resulting accuracy. The orange dots measure accuracy on natural images and the blue dots plot the empirical robust accuracy on adversarial images. The vertical dotted line indicates the minimum number of epochs needed to train a model to 45% robust accuracy.

All experiments using FGSM adversarial training are carried out with random initial starting points and step size $\alpha = 1.25\epsilon$ as described in Section 4.1. All PGD adversaries used at evaluation are run with 10 random restarts for 50 iterations (with the same hyperparameters as those used by Shafahi et al. (2019) but further strengthened with random restarts). Speedup with mixed-precision was incorporated with the Apex `amp` package at the `O1` optimization level for ImageNet experiments and `O2` without loss scaling for CIFAR10 experiments.[1]

5.1 Verified performance on MNIST

Since the FGSM attack is known to be significantly weaker than the PGD attack, it is understandable if the reader is still skeptical of the true robustness of the models trained using this method. To demonstrate that FGSM adversarial training confers real robustness to the model, in addition to evaluating against a PGD adversary, we leverage mixed-integer linear programming (MILP) methods from formal verification to calculate the exact robustness of small, but verifiable models (Tjeng et al., 2017). We train two convolutional networks with 16 and 32 convolutional filters followed by a fully connected layer of 100 units, the same architecture used by Tjeng et al. (2017). We use both PGD and FGSM adversarial training at $\epsilon = 0.3$, where the PGD adversary for training has 40 iterations with step size 0.01 as done by Madry et al. (2017). The exact verification results can be seen in Table 2, where we find that FGSM adversarial training confers nearly indistinguishable empirical and verified robustness to PGD adversarial training on MNIST.[2]

5.2 Fast CIFAR10

We begin our CIFAR10 experiments by combining the DAWNBench improvements from Section 4.2 with various forms of adversarial training. For N epochs, we use a cyclic learning rate that increases linearly from 0 to λ over the first $2N/5$ epochs, then decreases linearly from λ to 0 for the remaining epochs, where λ is the maximum learning rate. For each method, we individually tune

[1] Since CIFAR10 did not suffer from loss scaling problems, we found using the `O2` optimization level without loss scaling for mixed-precision arithmetic to be slightly faster.

[2] Exact verification results at $\epsilon = 0.3$ for both the FGSM and PGD trained models are not possible since the size of the resulting MILP is too large to be solved in a reasonable amount of time. The same issue also prevents us from verifying networks trained on datasets larger than MNIST, which have to rely on empirical tests for evaluating robustness.

Table 3: Time to train a robust CIFAR10 classifier to 45% robust accuracy using various adversarial training methods with the DAWNBench techniques of cyclic learning rates and mixed-precision arithmetic, showing significant speedups for all forms of adversarial training.

Method	Epochs	Seconds/epoch	Total time (minutes)
DAWNBench + PGD	15	91.59	23.14
DAWNBench + Free ($m = 8$)	88	13.46	19.88
DAWNBench + FGSM	15	23.41	6.02
PGD-7 (Madry et al., 2017)[3]	205	1456.22	4965.71
Free ($m = 8$) (Shafahi et al., 2019)[4]	205	197.77	674.39

Table 4: Imagenet classifiers trained with adversarial training methods at $\epsilon = 2/255$ and $\epsilon = 4/255$

Method	ϵ	Standard accuracy	PGD+1 restart	PGD+10 restarts
FGSM	$2/255$	60.90%	43.46%	43.43%
Free ($m = 4$)	$2/255$	64.37%	43.31%	43.28%
FGSM	$4/255$	55.45%	30.28%	30.18%
Free ($m = 4$)	$4/255$	60.42%	31.22%	31.08%

λ to be as large as possible without causing the training loss to diverge, which is the recommended learning rate test from Smith & Topin (2018).

To identify the minimum number of epochs needed for each adversarial training method, we repeatedly run each method over a range of maximum epochs N, and then plot the final robustness of each trained model in Figure 2. While all the adversarial training methods benefit greatly from the cyclic learning rate schedule, we find that both FGSM and PGD adversarial training require much fewer epochs than free adversarial training, and consequently reap the greatest speedups.

Using the minimum number of epochs needed for each training method to reach a baseline of 45% robust accuracy, we report the total training time in Table 3. We find that while all adversarial training methods benefit from the DAWNBench improvements, FGSM adversarial training is the fastest, capable of learning a robust CIFAR10 classifier in 6 minutes using only 15 epochs. Interestingly, we also find that PGD and free adversarial training take comparable amounts of time, largely because free adversarial training does not benefit from the cyclic learning rate as much as PGD or FGSM adversarial training.

5.3 Fast ImageNet

Finally, we apply all of the same techniques (FGSM adversarial training, mixed-precision, and cyclic learning rate) on the ImageNet benchmark. In addition, the top submissions from the DAWNBench competition for ImageNet utilize two more improvements on top of this, the first of which is the removal of weight decay regularization from batch normalization layers. The second addition is to progressively resize images during training, starting with larger batches of smaller images in the beginning and moving on to smaller batches of larger images later. Specifically, training is divided into three phases, where phases 1 and 2 use images resized to 160 and 352 pixels respectively, and phase 3 uses the entire image. We train models to be robust at $\epsilon = 2/255$ and $\epsilon = 4/255$ and compare to free adversarial training in Table 4, showing similar levels of robustness. In addition to using ten restarts, we also report the PGD accuracy with one restart to reproduce the evaluation done by Shafahi et al. (2019).

[3] Runtimes calculated on our hardware using the publicly available training code at `https://github.com/MadryLab/cifar10_challenge`.

[4] Runtimes calculated on our hardware using the publicly available training code at `https://github.com/ashafahi/free_adv_train`.

Table 5: Time to train a robust ImageNet classifier using various fast adversarial training methods

Method	Precision	Epochs	Min/epoch	Total time (hrs)
FGSM (phase 1)	single	6	22.65	2.27
FGSM (phase 2)	single	6	65.97	6.60
FGSM (phase 3)	single	3	114.45	5.72
FGSM	single	15	-	14.59
Free $(m = 4)$	single	92	34.04	52.20
FGSM (phase 1)	mixed	6	20.07	2.01
FGSM (phase 2)	mixed	6	53.39	5.34
FGSM (phase 3)	mixed	3	95.93	4.80
FGSM	mixed	15	-	12.14
Free $(m = 4)$	mixed	92	25.28	38.76

With these techniques, we can train an ImageNet classifier using 15 epochs in 12 hours using FGSM adversarial training, taking a fraction of the cost of free adversarial training as shown in Table 5.[5] We compare to the best performing variation of free adversarial training which which uses $m = 4$ minibatch replays over 92 epochs of training (scaled down accordingly to 23 passes over the data). Note that free adversarial training can also be enhanced with mixed-precision arithmetic, which reduces the runtime by 25%, but is still slower than FGSM-based training.

6 Conclusion

Our findings show that FGSM adversarial training, when used with random initialization, can in fact be just as effective as the more costly PGD adversarial training. While a single iteration of FGSM adversarial training is double the cost of free adversarial training, it converges significantly faster, especially with a cyclic learning rate schedule. As a result, we are able to learn adversarially robust classifiers for CIFAR10 in minutes and for ImageNet in hours, even faster than free adversarial training but with comparable levels of robustness. We believe that leveraging these significant reductions in time to train robust models will allow future work to iterate even faster, and accelerate research in learning models which are resistant to adversarial attacks.

References

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018.

Nicholas Carlini. Is ami (attacks meet interpretability) robust to adversarial examples? *arXiv preprint arXiv:1902.02322*, 2019.

Nicholas Carlini and David Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

[5] We use the implementation of free adversarial training for ImageNet publicly available at `https://github.com/mahyarnajibi/FreeAdversarialTraining` and reran it on the our machines to account for any timing discrepancies due to differences in hardware

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017a.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57. IEEE, 2017b.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

Cody Coleman, Deepak Narayanan, Daniel Kang, Tian Zhao, Jian Zhang, Luigi Nardi, Peter Bailis, Kunle Olukotun, Chris Ré, and Matei Zaharia. Dawnbench: An end-to-end deep learning benchmark and competition. *Training*, 100(101):102, 2017.

Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. *arXiv preprint arXiv:1810.07481*, 2018.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

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

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

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.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pp. 97–117. Springer, 2017.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.

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

Pratyush Maini, Eric Wong, and J Zico Kolter. Adversarial robustness against the union of multiple perturbation models. *arXiv preprint arXiv:1909.04068*, 2019.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pp. 3575–3583, 2018.

Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. Logit pairing methods can fool gradient-based attacks. *arXiv preprint arXiv:1810.12042*, 2018.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597. IEEE, 2016.

Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pp. 10877–10887, 2018.

Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sebastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. *arXiv preprint arXiv:1906.04584*, 2019.

Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 464–472. IEEE, 2017.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of residual networks using large learning rates. 2018.

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

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

Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *Advances in Neural Information Processing Systems*, pp. 7717–7728, 2018.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. *arXiv preprint arXiv:1904.13000*, 2019.

Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

Jianyu Wang. Bilateral adversarial training: Towards fast training of more robust models against adversarial attacks. *arXiv preprint arXiv:1811.10716*, 2018.

Eric Wong and J Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017.

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

Kai Y Xiao, Vincent Tjeng, Nur Muhammad Shafiullah, and Aleksander Madry. Training for faster adversarial robustness verification via inducing relu stability. *arXiv preprint arXiv:1809.03008*, 2018.

Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. Me-net: Towards effective adversarial robustness with matrix estimation. *arXiv preprint arXiv:1905.11971*, 2019.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Painless adversarial training using maximal principle. *arXiv preprint arXiv:1905.00877*, 2019.

A Training parameters for Table 1

Parameter	FGSM	PGD	Free
Epochs	30	400	400
Learning rate schedule	Cyclic	Piecewise	Piecewise
Max learning rate	0.05	0.1	0.05
Attack iterations	N/A	8	8

For all methods, we use a training batch size of 128 and a test batch size of 100, and SGD optimizer with momentum 0.9 and weight decay $5 * 10^{-4}$.

B Training parameters for Figure 2

Parameter	FGSM*	PGD	Free
Max learning rate	0.21	0.21	0.04
Attack iterations	N/A	8	8

For all methods, we use a training batch size of 128 and a test batch size of 100, and SGD optimizer with momentum 0.9 and weight decay $5 * 10^{-4}$.
*When training FGSM on a larger number of epochs, we use a 5-step PGD adversary with 1 restart on 1 minibatch to detect overfitting to the FGSM adversaries, and early stop if necessary.