Outrunning Them All: Fast Adversarial Training Against Multiple Perturbations

Divyam Madaan [1] **Fabian Kögel** [1] **Risto Koverola** [1] **Zineb Lahrichi** [1]

Abstract

With the real-world application of deep learning in safety-critical domains, it is required to make models robust against multiple adversarial attacks. However, defending against multiple attacks increases training time, as it requires adversarial examples against each attack. In this work, we replicate the FGSM adversarial training (Wong et al., 2020) which has shown to be an effective method for reducing the training cost of adversarial training against ℓ_∞ attacks. Furthermore, we extend the method to ℓ_1 and ℓ_2 attacks to utilize it in training against multiple attacks. To further lower the training time, we propose our novel *random adversarial training* and *split-batch training* methods. We validate the robustness of our proposed method on various datasets and architectures, demonstrating that it achieves comparable performance ($\pm 1\%$ robustness) to previous techniques with up to 80% lower training cost.

1. Introduction

Deep learning has demonstrated remarkable success on various benchmark applications (He et al., 2016a; Amodei et al., 2016; Devlin et al., 2018). This has lead to the rise of deep learning in safety-critical applications. However Szegedy et al. (2013) found these networks extremely susceptible to adversarial examples, carefully crafted imperceptible perturbations introduced into the input to cause misclassification.

Since then, many defense strategies have been developed (Ross & Doshi-Velez, 2018; Madry et al., 2017; Tramèr & Boneh, 2019; Zhang et al., 2019) and eventually broken by stronger adversarial attacks (Carlini & Wagner, 2017; Athalye et al., 2018; Uesato et al., 2018). A method (Madry et al., 2017) that has yet to be broken uses projected gradient descent (PGD). However its robustness comes at the cost of large computational cost and longer training time. Shafahi et al. (2019) formulate "Free" Adversarial Training which improves the training performance by merging the adversarial generation and the defense training backpropagation. Recently, Wong et al. (2020) discovered that under the right circumstances even weak adversarial examples can train a model to robustness against strong attacks. They show why researches previously failed to apply Fast Gradient Sign Method (FGSM), the original predecessor of PGD, and that with simply using random initialization FGSM achieves comparable robustness to PGD and Free training, while being significantly faster than both.

Most of the existing defense techniques have been tailored to defend against single adversarial attack, which is rarely the case in real-life scenarios. Additionally it has been shown that training against one attack can make the model vulnerable against other type of attacks (Tramèr & Boneh, 2019; Schott et al., 2018). Tramèr & Boneh (2019) therefore propose training methods against multiple perturbations. However, these come again at an additional cost in training time.

In this work we tackle the problem of computational cost for achieving robustness against multiple attacks. The contributions of our project can be summarized as follows:

- We replicate the ICLR 2020 paper "Fast is better than Free: Revisiting Adversarial training" (Wong et al., 2020), demonstrating that FGSM adversarial training can be as effective as the costly PGD training, while being magnitudes faster for ℓ_∞ norm attacks.
- We extend their approach to multiple ℓ_p attacks. Furthermore, we propose our own novel *random adversarial training* and *split-batch adversarial training* to significantly lower the training time for multiple perturbations adversarial training.
- We experimentally evaluate our method on various datasets including MNIST, CIFAR-10 and CIFAR-100 where it achieves comparable performance to existing techniques for multiple perturbations while significantly lowering the training time.

[1]School of Computing, KAIST, South Korea. Correspondence to: Divyam Madaan <dmadaan@kaist.ac.kr>, Fabian Kögel <f.koegel@kaist.ac.kr>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

2. Related Work

Adversarial attacks. Szegedy et al. (2013) proposed adversarial perturbations for the first time using L-BFGS optimization procedure, since then several studies have demonstrated the fragility of convolution networks with a set of much stronger attacks (Carlini & Wagner, 2017; Schott et al., 2018; Athalye et al., 2018; Brendel et al., 2018). Projected Gradient Descent Attack (Madry et al., 2017) takes maximum loss increments within a specific ℓ_p norm ball. These attacks fall in the 'white-box attack' category where the network is assumed to be transparent to the attacker. Adversarial examples can also be crafted using a 'black box' or 'oblivious' attack where there is no information of network architecture or parameters available to the model. Black box attacks can be done using a substitute model (Papernot et al., 2017), using an ensemble approach (Liu et al., 2018) or using zeroth-order optimization attacks (Zhang et al., 2019).

Adversarial defenses. Since the literature on adversarial robustness of neural networks is vast, we only discuss some of the most relevant studies. There have been a wide range of proposed empirical defenses (Song et al., 2017; Madry et al., 2017; Tramèr et al., 2018) based on heuristic training strategies. One of the most successful defense, which has proven to be robust against various attacks, is adversarial training (Madry et al., 2017), in which the neural network is trained to optimize the maximum loss obtainable uses projected gradient descent over the region of allowable perturbations.

However, all this works claim empirical robustness against the perturbation-type which was used for the training of the defense. With the continuous advent of stronger attacks, it is essential to evaluate a defense on various adversaries which might posses a threat for any defense. Recently, Tramèr & Boneh (2019); Maini et al. (2019); Schott et al. (2018) show that most of these defenses overfit to the attack used in training and adversarial training (Madry et al., 2017) can be broken with stronger decision based and gradient-free attacks. In this work, we thus investigate the robustness against various attacks and show that our proposed method can attain higher robustness against various attacks.

3. Replication Methodology

We replicate the paper "Fast is better than Free: Revisiting Adversarial Training" (Wong et al., 2020). It re-investigates the Fast Gradient Sign Method (Goodfellow et al., 2014), which is one of the earliest methods and often neglected as ineffective. They show that FGSM with random initialization can be as effective as the much more computationally expensive PGD, while being as fast as standard training. The general idea of adversarial training is to minimize the maximal loss that can be provoked by perturbing the input.

$$\min_{\theta} \sum_{i} \max_{\delta \in \Delta} L(f_\theta(x_i + \delta), y_i)$$

Algorithm 1 FGSM adversarial training

input T epochs, given some radius ϵ, N PGD steps, step size α, and a dataset of size M for a network f_θ
output Network f_θ

for $t = 1 \dots T$ **do**
  for $i = 1 \dots M$ **do**
    // Perform FGSM adversarial attack
    $\delta = \text{Uniform}(-\epsilon, \epsilon)$
    $\delta = \delta + \alpha \cdot (\nabla_\delta \ell(f_\theta(x_i + \delta), y_i))$
    $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
    $\theta = \theta - \nabla_\theta \ell(f_\theta(x_i + \delta), y_i)$
  end for
  Update model weights on generated examples.
end for

Specifically, gradient-based approaches first run the model with an initialized perturbation, obtain the gradient with respect to the perturbation δ and then advance the perturbation one step α in direction of the gradient. If the perturbation steps outside the norm-ball $\mathcal{B}(x, \epsilon)$ it is projected back into it. FGSM runs only one iteration (see Algorithm 1), while PGD runs multiple iterations (to better approximate the non-convex loss function). After the perturbation has been obtained the model is trained on the perturbed samples to learn a more robust decision boundary.

FGSM is a weaker attack, that approximates the inner maximization in a single step, and many authors have found it ineffective (Wong et al., 2020). Our replication paper shows that the step size α and the initialization of δ are critical for its success. They identify two failure modes referred as "Catastrophic Overfitting" which is caused by insufficient initialization or large step-sizes. Both modes produce perturbations at the boundary leading to a very restricted attack model to which the defense overfits. Randomly initializing the perturbation and adding steps smaller than 2ϵ (the diameter of the norm ball) makes sure that during training adversary examples spanning the whole range are generated.

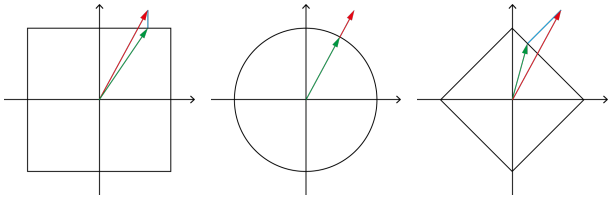
In summary, our replication paper shows:

- FGSM can achieve comparable robustness to PGD training on ℓ_∞ attacks across various datasets.
- FGSM training can be accelerated using DAWNBench training techniques (Smith, 2017).
- Catastrophic Overfitting could be the failure mode of FGSM that previously resulted in poor robustness for many researchers.

Algorithm 2 Random Adversarial Training

input Dataset $\mathcal{D}$, training iterations T, trained model f_θ with parameters θ, batch size m.
output Network f_θ.

for $t = \{1, \dots, T\}$ **do**
  Sample mini-batch $B = \{x_1, \dots, x_m\} \subset \mathcal{D}$
  Sample an attack $\mathcal{A}_i$ with epsilon ε_i and step-size α_i from a distribution of attacks $p(\mathcal{A})$
  for $i = \{1, \dots, m\}$ **do**
    // Run PGD adversary for $\mathcal{A}_i$
    for $k = \{1, \dots, K\}$ **do**
      $\tilde{x_i} = \underset{\mathcal{B}(x_i, \varepsilon)}{\text{proj}} \left(\tilde{x_i} + \alpha \cdot \text{sgn}(\nabla_x \mathcal{L}(\theta, \tilde{x_i}, y))\right)$
    end for
  end for
  Optimize θ with the generated examples.
end for



(a) l_∞ clamping (b) l_2 scaling (c) l_1 simplex proj.

Figure 1. Shapes of 2D norm balls and mechanisms for projecting perturbations outside the range onto their surface

4. Improvement Methodology

To extend our replicated approach of FGSM ℓ_∞ adversarial training to ℓ_1 and ℓ_2, we exchange the projections on the norm balls used in the perturbation generation step and the corresponding initialization method (see Figure 1). Instead of initializing the vector components uniformly, we distribute the vector angles and magnitudes uniformly within the norm ball. If taking a step into the direction of the gradient puts the norm of the perturbation outside the epsilon range, the vector is projected back on the surface of the norm ball. Since ℓ_2 is the vector magnitude we can use a simple scaling approach with factor $\frac{\epsilon}{\min(\|\delta\|_2, \epsilon)}$ that scales the values outside the norm ball. For ℓ_1 we project back on the simplex represented by the norm ball.

Recently, Tramèr & Boneh (2019) proposed a training method to train a single network to be robust against multiple ℓ_p attacks using following strategies:

Max strategy: In every iteration it runs a full PGD attack for every attack type and then uses the attack that produced maximal loss.

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\text{argmax} \max_{\delta_{k_{1:n}} \in \mathcal{B}_{k_{1:n}}(x,\varepsilon)} \mathcal{L}\left(f_\theta\left(x + \delta_k\right), y\right) \right] \tag{1}$$

Average strategy: In every iteration it runs a full PGD attack for every attack type and then uses the model on all the attacks and backpropagates the average loss.

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D})} \frac{1}{n} \sum_{i=1}^{i=n} \left[\max_{\delta_i \in \mathcal{B}_i(x,\varepsilon)} \mathcal{L}_i\left(f_\theta\left(x + \delta_i\right), y\right) \right] \tag{2}$$

It is easy to observe that this approach would benefit greatly from a faster method than PGD. We use these strategies but replace PGD with our FGSM for all norms. To further reduce the computation, we hypothesize that it is not necessary to run all attacks for training with multiple perturbations and propose our random sampling strategy.

Random Adversarial Training Motivated by the fact that the norm balls overlap and (Wong et al., 2020) find that training against weak attacks, if done right, can make the network also robust against strong attacks, we propose our novel random adversarial training. Specifically, we sample a random attack from a given attack distribution and optimize the model parameters on the adversarial examples generated by FGSM attack on the sampled attack. More formally, we summarize our method in algorithm 2 and we can define our objective as follows:

$$\min_\theta \mathbb{E}_{\substack{(x,y)\sim\mathcal{D} \\ \delta_k \sim p(\delta)}} \left[\max_{\delta_k \in \mathcal{B}_k(x,\varepsilon)} \mathcal{L}\left(f_\theta\left(x + \delta_k\right), y\right) \right] \tag{3}$$

Split-batch Adversarial Training Another way of reducing the computation is to run all attacks, but reduce the amount of data used to estimate the gradient. Instead of running on a full batch, we propose Split-batch adversarial training that distributes a given batch to n/k portions where n is the size of the batch and k is the number of attacks and runs every attack only on k-th portion of the batch and then apply the max/average strategies.

- After replicating the results, we extend the‘ approach to ℓ_1 and ℓ_2 attacks and show that similar speedups can be reached without compromising accuracy.
- With a fast method of training against three single attacks at hand, we show that instead of running all attacks and picking the maximum or average for the training step, randomly choosing an attack is also effective and saves computation time.
- We show that it is not necessary to run with a full batch of adversarial examples for every norm attack. Instead we split the batch and run multiple attacks in one go with each only using a subset of the samples.

5. Experiments

5.1. Experimental Setup

Datasets

1. **MNIST** The MNIST database of handwritten digits of size (28x28) has a training set of 60,000 examples, and a test set of 10,000 examples. We use Lenet5-Caffe as the baseline model for this dataset.
2. **CIFAR-10** CIFAR-10 is a collection of 60,000 (32x32) color images in 10 different classes with 6,000 images of each class. We use the pre-activation version of the ResNet18 architecture (He et al., 2016b) as a base network for this dataset.
3. **CIFAR-100** CIFAR-100 is a collection of 60,000 (32x32) color images where each class has 500 images for training and 100 images for test. Similar to CIFAR-10, we use pre-activation version of the ResNet18 as the base model for this dataset.

5.2. Replication Results

Table 1. Replicated robust performance on MNIST

Method	Std. Acc.	PGD (ϵ = 0.1)	PGD (ϵ = 0.3)
PGD	97.01	93.58	91.13
FGSM	98.36	96.68	85.35

Robust Test Performance We first replicate the robust test performance on MNIST of PGD training and FGSM (+random initialization) training against PGD l_∞ attacks at typical strength. We train with $\epsilon = 0.3$ and $\alpha = 0.01$ for PGD and $\alpha = 0.375$ for FGSM. Table 1 shows the results in comparison to the original results. In comparison to the original paper we obtain slightly lower absolute values due to differences in our experimental setup (see appendix A). In relative terms we confirm the author's findings: the FGSM training with random initialization produces comparable robustness to PGD training.

Table 2. Replicated training times on CIFAR10

Method	Epochs	Sec./Epoch	Time (min)
DAWNBench + PGD-7	10	114	19.00
DAWNBench + Free	80	98.9	131.86
DAWNBench + FGSM	10	30	7.5
PGD-7	205	1584	5418
Free	205	200.1	683

Training Time We replicate the experiments on the time to train a model to 45% robust accuracy against a PGD attack with $\epsilon = 8/255$ on CIFAR-10. Our results in Table 2 show PGD and Free take considerably longer to train than the FGSM method, as also reported by the original authors.

Table 3. Catastrophic overfitting on CIFAR10 and MNIST

	Method	Std. Acc.	Adv. Acc.
MNIST	FGSM		
	+ zero init	95.56 %	0.00 %
	+ previous init	98.55 %	87.48 %
	+ random init		
	+ $\alpha = 0.3$	98.60%	85.35%
	+ $\alpha = 0.6$	95.81%	0.00%
	Free	-	-
	PGD	99.00%	92.81%
CIFAR-10	FGSM		
	+ zero init	83.97%	0.00%
	+ random init		
	+ $\alpha = 10/255$	79.63%	45.21%
	+ $\alpha = 16/255$	83.21%	0.02%
	Free	83.90%	45.40%
	PGD	81.11%	48.07%

The training time for Free in combination with DAWN-Bench is suspiciously high, but we are not considering Free for our work and we reproduce the main point about FGSM, so we chose not to investigate further.

Catastrophic Overfitting We replicate the two failure modes that the authors identified. Zero initialization or too large step sizes cause FGSM training to abruptly overfit to the adversarial examples, resulting in zero percent robustness. We train multiple models with FGSM using different initialization and step sizes on CIFAR-10 and obtain the same results as the original paper (see Table 3) when evaluating against a PGD attack at $\epsilon = 8/255$. We additionally conduct similar experiments on the MNIST dataset against a PGD attack at $\epsilon = 0.3$, confirming that the phenomenon is not an artifact of training on CIFAR-10 dataset.

5.3. Quantitative Results

Results on ℓ_1 and ℓ_2 FGSM extension We investigate the performance of FGSM adversarial training with our extension to ℓ_1 and ℓ_2. We train with PGD and FGSM at $\epsilon = \{6.5, 180.0\}$ respective for ℓ_2 and ℓ_1 attack on MNIST, and $\epsilon = \{100, 4000\}$ for ℓ_2 and ℓ_1 attack on CIFAR-10. The results for MNIST are summarized in Table 4. Full results including the other datasets can be found in the appendix. We can observe that, FGSM adversarial training gives comparable performance to PGD adversarial training for ℓ_2 adversarial attack with 87.72% reduction in training time. However, they do not achieve robustness for the full ϵ the have been trained on. After inspection of the generated perturbations, we conclude this is due to the projection method we use and the method failing. On ℓ_1, FGSM is un-

Table 4. ℓ_p Extension of FGSM Training against PGD Attacks on MNIST

	ℓ_2					ℓ_1				
Model	Acc.	$\epsilon = 2.0$	$\epsilon = 3.0$	$\epsilon = 6.5$	Time (min)	Acc.	$\epsilon = 0.3$	$\epsilon = 180.0$	$\epsilon = 235.2$	Time (min)
No Attack	99.06	27.22	16.86	1.98	1.5	98.96	98.91	0.00	0.00	0.8
PGD	96.85	74.77	**53.19**	0.18	28.5	11.35	11.35	11.35	11.35	10
FGSM	99.01	**81.35**	45.64	0.0	**3.5**	99.01	99.01	11.93	2.90	2.2

Table 5. Single and Multiple Perturbation FGSM Training on MNIST and CIFAR10

	MNIST						CIFAR-10					
Model	Acc.	ℓ_∞	ℓ_1	ℓ_2	Avg.	Time(min)	Acc.	ℓ_∞	ℓ_1	ℓ_2	Avg.	Time(min)
Single Attack Training												
+ ℓ_∞ FGSM	99.13	**87.99**	7.08	**92.51**	**62.53**	1	81.59	**46.31**	16.95	76.09	46.45	15
+ ℓ_1 PGD	98.94	0.00	5.34	39.46	14.94	1.5	89.86	0.49	**72.54**	76.64	49.89	207
+ ℓ_2 FGSM	99.27	0.01	0.02	86.35	28.80	1	91.52	9.73	11.70	83.20	34.88	15
Multiple Attack Training	using PGD ℓ_1						using PGD ℓ_1					
+ Max FGSM	98.92	64.69	9.55	92.16	55.47	8	81.03	**43.57**	43.34	76.56	**54.49**	228
+ Avg FGSM	99.07	**83.11**	**14.53**	**93.40**	**63.68**	7.9	85.18	31.57	**50.36**	**79.97**	53.97	239
+ Random FGSM (Ours)	98.97	82.54	9.62	93.14	61.77	**3.5**	84.60	33.26	**50.10**	**79.59**	**54.32**	**79**

able to defend against adversarial perturbations. We believe that this is due to the higher sparsity of ℓ_1 perturbations, which requires more than one step to estimate the gradient and converge the model. The results are consistent with those on CIFAR-10, which we report in the appendix.

FGSM in Multiple Perturbation Training We run multiple sets of experiments using PGD and FGSM training against all three norm attacks on MNIST, CIFAR-10 and CIFAR-100 datasets. The summarized results in Table 5 show, single attack adversarial training is only robust to the respective attack used for training, highlighting the problem of generalization across multiple adversarial perturbations. FGSM works well as replacement for PGD in max and average training strategies, reaching the same average robustness with about 40% less training time. Again further results on CIFAR-10 and CIFAR-100 confirm the findings and are reported in the appendix.

Multiple Perturbation Training with Random Sampling We compare our proposed random sampling strategy for training against multiple simultaneous perturbations against the max and average strategy of Tramèr & Boneh (2019). In particular, we train against ℓ_∞, ℓ_1, and ℓ_2-norm adversarial attack. The results in Table 5 show that our proposed random strategy outperforms the max strategy by 6.30% with 56.25% reduction in training time on MNIST dataset. We can also observe that it leads to 65.35% reduction in training time for CIFAR-10 dataset while achieving comparable performance across multiple perturbations. In addition, we provide results for random sampling with CIFAR-100 dataset in the appendix.

Table 6. Split Batch PGD average strategy training on CIFAR-10

Model	Acc.	ℓ_∞	ℓ_1	ℓ_2	Avg.	Time
Full Batch	84.09	**36.11**	**48.58**	**78.84**	**54.51**	592
Split Batch	84.02	34.24	47.02	**78.69**	53.32	**358**

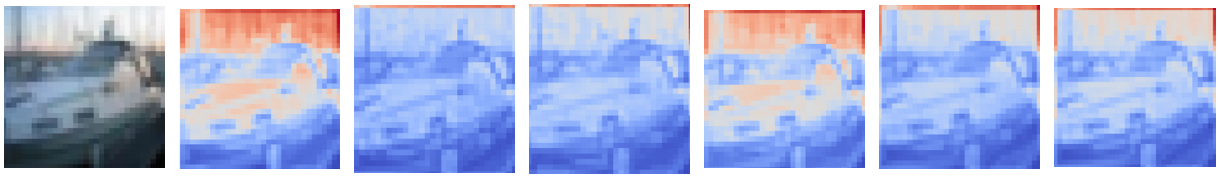
Multiple Perturbation Training with Split Batch We evaluate our proposed split batch adversarial training with the average strategy in Table 6. When replacing the full batch with only a third of the samples per attack we can observe a 39.53% reduction in training time while achieving comparable performance. In comparison, our random strategy with full batch (Table 5) already shows a much more significant reduction in the training time, demonstrating the greater efficiency of this method.

5.4. Qualitative Results

Correlation between Attack Types In order to gain insights into the different attack types, we visualize the correlation between ℓ_p attacks. Figure 3 showshigh correlation in perturbations generated under ℓ_1 and ℓ_2. In contrast, ℓ_∞ generates significantly more perturbed pixels and shows little correlation with others. We investigate if we can exploit correlations in our proposed random sampling strategy by using different sampling percentages for the norms. The results in Table 7 show that the sampling percentage has a significant impact on the method's performance. In terms of robustness, higher sampling for a norm leads to an increase against the corresponding attack, but a consistent decrease against other attack types. This demonstrates an inherent trade-off also observed in previous works on multiple perturbations (Tramèr & Boneh, 2019; Schott et al., 2018).

Table 7. Effect of Sampling Percentage. Random Sampling using PGD Training against PGD attacks on CIFAR-10

Sampling in % $(\ell_\infty, \ell_1, \ell_2)$	Standard Acc.	ℓ_∞ PGD $\epsilon = 8/255$	ℓ_1 PGD $\epsilon = 2000/255$	ℓ_2 PGD $\epsilon = 80/255$	Average Adv.
Uniform	79.7	38.3	46.7	**64.4**	49.8
(75, 25, 0)	75.6	**45.3**	43.7	62.7	50.6
(50, 25, 25)	79.8	41.0	42.9	64.1	49.3
(25, 75, 0)	78.5	38.5	**51.1**	63.9	**51.2**
(25, 50, 25)	82.2	32.1	47.7	**64.4**	48.1
(25, 25, 50)	**84.2**	30.8	39.6	63.5	44.6



(a) Original (b) ℓ_∞ (c) ℓ_1 (d) $\ell 2$ (e) Max (f) Average (g) Random

Figure 2. Visualization of convolutional features of first layer of various adversarial trained networks.

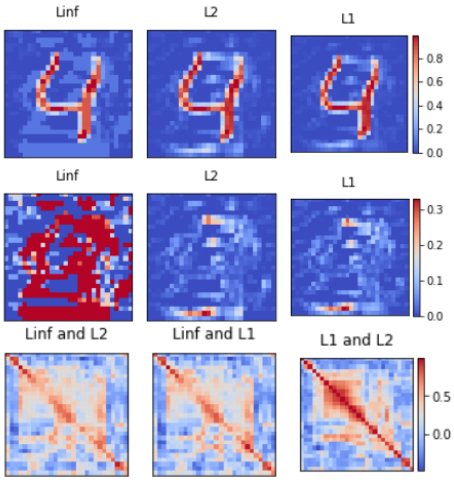


Figure 3. Adversarial Example, Perturbation and Correlation between different ℓ_p norm attacks for an MNIST example.

Note that, 0% ℓ_2 sampling with high ℓ_1 sampling leads to a model that is still robust against ℓ_2 attack, supporting our understanding of correlation between attack types.

Visualization of Latent Features To further gain insight into multiple perturbations training, we visualize the first layer across various networks. Figure 2 shows similarities between ℓ_1 and ℓ_2 models and between ℓ_∞ and max strategy models. This indicates that the max strategy learns maximally from the ℓ_∞ attacks, which is reasonable since ℓ_∞ is the most potent attack. Notice that our random strategy leads to similar features as the average strategy showing the reason for their comparable performance in Table 5.

6. Conclusion

In this work we tackle the problem of computational cost for achieving robustness against multiple attacks. We successfully replicate the ICLR 2020 paper "Fast is better than Free: Revisiting Adversarial training" (Wong et al., 2020) which shows that given the right circumstances FGSM adversarial training can be as effective as the costly PGD training, while being magnitudes faster. We extend their method to ℓ_p-norm attacks to make it usable in multiple perturbations training. In addition to the max and average training strategies of Tramèr & Boneh (2019) we propose our novel *random adversarial training* that trains on a random attack. Our experimental results demonstrate a successful extension to ℓ_2 and application in multiple perturbations training. Our random training method leads to significant reduction in computation cost (up to 80% on PGD with CIFAR-10), while achieving comparable average performance ($\pm$ 1%) for CIFAR-10).

Limitations & Future Work The extension of FGSM to ℓ_1 norm attacks proved difficult and convergence problems lead to low robustness. This might be a general issue as convergence in a single iteration with very sparse gradients is difficult. On the other hand we found the correct choice of hyper parameters to be critical and methods to estimate them are needed. Furthermore we want to better understand the relationship between attack types and define strength quality to estimate equivalent parameters better. Concrete approaches for future work are random sampling with oversampling of non-intersecting areas of norms and split batch with intelligent partitioning.

References

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML*, 2016.

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *ICLR*, 2018.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 2017.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *ECCV*, 2016b.

Liu, X., Li, Y., Wu, C., and Hsieh, C.-J. Adv-bnn: Improved adversarial defense through robust bayesian neural network. *arXiv preprint arXiv:1810.01279*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2017.

Maini, P., Wong, E., and Kolter, J. Z. Adversarial robustness against the union of multiple perturbation models. *arXiv preprint arXiv:1909.04068*, 2019.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017.

Ross, A. S. and Doshi-Velez, F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *AAAI*, 2018.

Schott, L., Rauber, J., Bethge, M., and Brendel, W. Towards the first adversarially robust neural network model on mnist. In *ICLR*, 2018.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! In *Neurips*, 2019.

Smith, L. N. Cyclical learning rates for training neural networks. In *WACV*, 2017.

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tramèr, F. and Boneh, D. Adversarial training and robustness for multiple perturbations. In *Neurips*, 2019.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018.

Uesato, J., O'Donoghue, B., Oord, A. v. d., and Kohli, P. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

A. Training and Evaluation Setup

We trained the models for CIFAR-10 on RTX GeForce 2080Ti similar to the authors. For consistency of results we also use LeNet5 for MNIST and RestNet18 for CIFAR-10. For the replication of the robust test performance on MNIST, we reduced the number of epochs from 100 to 10 to be able to run on CPU within reasonable time limits. All PGD adversaries used at evaluation are run with 10 random restarts for 40 iterations. All code for reproducing the experiments in this paper as well as pretrained model weights can be found online [1].

B. Additional Results

See next page.

[1] https://github.com/faibk/fast_adversarial

Table 8. Single and Multiple Perturbation PGD Training on MNIST and CIFAR10

	MNIST						CIFAR10					
Model	Acc.	ℓ_∞	ℓ_1	ℓ_2	Avg.	Time (min)	Acc.	ℓ_∞	ℓ_1	ℓ_2	Avg.	Time (min)
No Attack (Standard)	99.14	0.02	0.04	37.56	12.54	0.9	92.95	0.00	0.00	22.34	7.45	9
Single Attack Training												
+ ℓ_∞ PGD	98.98	88.41	10.64	91.47	63.51	6.5	81.11	**48.07**	16.22	75.66	46.65	78
+ ℓ_1 PGD	**98.84**	92.44	**17.22**	**95.31**	68.32	17	89.86	0.49	**72.54**	76.64	49.89	207
+ ℓ_2 PGD	99.32	0.68	1.50	87.46	29.88	7.2	89.06	22.81	26.08	**82.63**	43.84	81
Multiple Attack Training												
+ Max PGD	98.86	**91.56**	32.03	94.07	72.55	29.1	80.90	**46.26**	39.55	76.48	54.10	357
+ Avg PGD	98.93	90.10	32.58	94.01	72.23	29	**83.90**	37.11	48.58	**78.84**	54.84	384
+ Random PGD (Ours)	**98.98**	90.92	**35.04**	**94.71**	**73.56**	**10.6**	83.16	38.07	**49.87**	78.39	**55.44**	**123**

Table 9. Single and Multiple Perturbation FGSM and PGD Training on CIFAR100.

	FGSM						PGD					
Model	Acc.	ℓ_∞	ℓ_1	ℓ_2	Avg.	Time (min)	Acc.	ℓ_∞	ℓ_1	ℓ_2	Avg.	Time (min)
Single Attack Training												
+ ℓ_∞	55.34	25.27	11.38	49.27	28.64	15	54.58	25.74	11.74	49.11	28.86	78
+ ℓ_1	-	-	-	-	-	-	65.89	1.61	48.90	46.31	32.27	207
+ ℓ_2	69.32	5.28	7.34	56.58	23.07	15	64.81	10.81	15.01	56.56	27.46	81
Multiple Attack Training	using PGD ℓ_1											
+ Max	54.95	22.67	21.54	49.57	31.26	228	58.33	24.34	19.70	49.13	31.06	357
+ Avg	59.98	22.67	21.54	49.57	31.26	228	54.09	18.02	29.74	52.28	33.35	384
+ Random (Ours)	59.66	15.43	29.68	53.32	32.81	**80**	57.92	18.57	29.85	52.13	**33.52**	**123**

Table 10. FGSM Training under ℓ_2 on MNIST

Model	Standard Acc.	PGD ($\epsilon = 0.55$)	PGD ($\epsilon = 2.0$)	PGD ($\epsilon = 3.0$)	PGD ($\epsilon = 6.5$)	PGD ($\epsilon = 8.4$)	Time (min)
No Attack (Standard)	99.06	92.45	27.22	16.86	1.98	0.4	1.5
PGD ($\epsilon = 6.5, \alpha = 0.22$)	96.85	94.02	74.77	**53.19**	0.18	0.01	28.5
FGSM ($\epsilon = 6.5, \alpha = 2.5$)	99.01	**97.77**	**81.35**	45.64	0.01	0.0	**3.5**

Table 11. FGSM Training under ℓ_1 on MNIST

Model	Standard Acc.	PGD ($\epsilon = 0.3$)	PGD ($\epsilon = 180.0$)	PGD ($\epsilon = 235.2$)	Time (min)
No Attack (Standard)	98.96	98.91	0.00	0.00	0.8
PGD ($\epsilon = 0.3, \alpha = 0.01$)	99.00	99.00	5.74	0.14	10
PGD ($\epsilon = 180.0, \alpha = 6.0$)	11.35	11.35	11.35	11.35	10
PGD ($\epsilon = 235.2, \alpha = 7.84$)	11.35	11.35	11.35	11.35	10
FGSM ($\epsilon = 0.3, \alpha = 0.375$)	99.14	99.12	0.04	0.00	2
FGSM ($\epsilon = 180.0, \alpha = 225.0$)	99.01	99.01	11.93	2.90	2.2
FGSM ($\epsilon = 235.2, \alpha = 294.0$)	11.35	11.35	11.35	11.35	1