Universal Adversarial Training

Ali Shafahi,* Mahyar Najibi,* Zheng Xu,* John Dickerson, Larry S. Davis, Tom Goldstein
Deparment of Computer Science
University of Maryland
College Park, Maryland 20742
{ashafahi, najibi, xuzh, john, lsd,tomg}@cs.umd.edu

Abstract

Standard adversarial attacks change the predicted class label of a selected image by adding specially tailored small perturbations to its pixels. In contrast, a *universal* perturbation is an update that can be added to *any* image in a broad class of images, while still changing the predicted class label. We study the efficient generation of universal adversarial perturbations, and also efficient methods for hardening networks to these attacks. We propose a simple optimization-based universal attack that reduces the top-1 accuracy of various network architectures on ImageNet to less than 20%, while learning the universal perturbation $13\times$ faster than the standard method.

To defend against these perturbations, we propose *universal adversarial training*, which models the problem of robust classifier generation as a two-player min-max game, and produces robust models with only $2\times$ the cost of natural training. We also propose a simultaneous stochastic gradient method that is almost free of extra computation, which allows us to do universal adversarial training on ImageNet.

1 Introduction

Deep neural networks (DNNs) are vulnerable to adversarial examples, in which small and often imperceptible perturbations change the class label of an image (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014; Nguyen, Yosinski, and Clune 2015; Papernot et al. 2016). Many works have shown that these vulnerabilities can be exploited by showing real world attacks on face detection (Sharif et al. 2016), object detection (Wu et al. 2019), and copyright detection (Saadatpanah, Shafahi, and Goldstein 2019).

Adversarial examples were originally formed by selecting a single example, and sneaking it into a different class using a small perturbation (Carlini and Wagner 2017b). This is done most effectively using (potentially expensive) iterative optimization procedures (Dong et al. 2017; Madry et al. 2018; Athalye, Carlini, and Wagner 2018).

Different from per-instance perturbation attacks, (Moosavi-Dezfooli et al. 2017b; 2017a) show there exists "universal" perturbations that can be added to any image to change its class label (fig. 1) with high probability. Universal perturbations empower attackers who cannot generate per-instance adversarial examples on the go, or who want to change the identity of an object to be selected later in the field. Also, universal perturbations have good cross-model transferability, which facilitates black-box attacks.

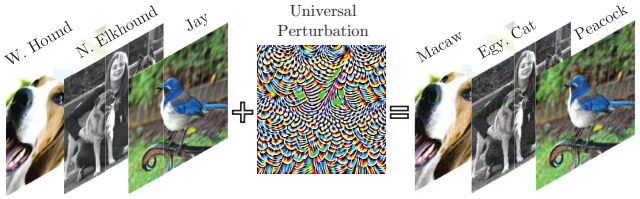


Figure 1: A universal perturbation made using a subset of ImageNet and the VGG-16 architecture. When added to the validation images, their labels usually change. The perturbation was generated using the proposed algorithm 2. Perturbation pixel values lie in $[-10, 10]$ (i.e. $\epsilon = 10$).

Among various methods for hardening networks to per-instance attacks, *adversarial training* (Madry et al. 2018) is known to dramatically increase robustness (Athalye, Carlini, and Wagner 2018). In this process, adversarial examples are produced for each mini-batch during training, and injected into the training data. While effective at increasing robustness against small perturbations, it is not effective for larger perturbations which are often the case for universal perturbations. Also, the high cost of this process precludes its use on large and complex datasets.

Contributions This paper studies effective methods for producing and deflecting universal adversarial attacks. First, we pose the creation of universal perturbations as an optimization problem that can be effectively solved by stochastic gradient methods. This method dramatically reduces the time needed to produce attacks as compared to (Moosavi-Dezfooli et al. 2017b). The efficiency of this formulation empowers us to consider universal adversarial training. We formulate the adversarial training problem as a min-max op-

*Equal contribution.

timization where the minimization is over the network parameters and the maximization is over the universal perturbation. This problem can be solved quickly using alternating stochastic gradient methods with no inner loops, making it far more efficient than per-instance adversarial training with a strong adversary (which requires a PGD inner loop to generate adversarial perturbations). Prior to our work, it was argued that adversarial training on universal perturbations is infeasible because the inner optimization requires the generation of a universal perturbation from scratch using many expensive iterations (Perolat et al. 2018). We further improve the defense efficiency by providing a "low-cost" algorithm for defending against universal perturbations. Through experiments on CIFAR-10 and ImageNet, we show that this "low-cost" method works well in practice. Note, the approach first introduced here has been expanded on to create a range of "free" adversarial training strategies with the same cost as standard training (Shafahi et al. 2019).

2 Related work

We briefly review *per-instance* perturbation attack techniques that are closely related to our paper and can be used during the universal perturbation update step of universal adversarial training. The Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014) is one of the most popular one-step gradient-based approaches for ℓ_∞-bounded attacks. FGSM applies one step of gradient ascent in the direction of the sign of the gradient of the loss function with respect to the input image. When a model is FGSM adversarially trained, the gradient of the loss function may be very small near unmodified images. In this case, the R-FGSM method remains effective by first using a random perturbation to step off the image manifold, and then applying FGSM (Tramèr et al. 2018). Projected Gradient Descent (PGD) (Madry et al. 2018) iteratively applies FGSM multiple times, and is one of the strongest per-instance attacks (Athalye, Carlini, and Wagner 2018). The PGD version of (Madry et al. 2018) applies an initial random perturbation before multiple steps of gradient ascent and projection of perturbation onto the norm-ball of interest, and in a standard projected gradient method (Goldstein, Studer, and Baraniuk 2014). Finally, DeepFool (Moosavi-Dezfooli, Fawzi, and Frossard 2016) is an iterative method based on a linear approximation of the training loss objective.

Adversarial training, in which adversarial examples are injected into the dataset during training, is an effective method to learn a robust model resistant to per-instance attacks (Madry et al. 2018; Huang et al. 2015; Shaham, Yamada, and Negahban 2015; Sinha, Namkoong, and Duchi 2018). Robust models adversarially trained with FGSM can resist FGSM attacks (Kurakin, Goodfellow, and Bengio 2017), but can be vulnerable to PGD attacks (Madry et al. 2018). (Madry et al. 2018) suggest strong attacks are important, and they use the iterative PGD method in the inner loop for generating adversarial examples when optimizing the min-max problem. PGD adversarial training is effective but time-consuming when the perturbation is small. The cost of the inner PGD loop is high, although this can sometimes be replaced with neural models for attack generation (Baluja and Fischer 2018; Poursaeed et al. 2018; Xiao et al. 2018). These robust models are adversarially trained to fend off per-instance perturbations and have not been designed for, or tested against, universal perturbations.

Unlike per-instance perturbations, *universal* perturbations can be directly added to any test image to fool the classifier. In (Moosavi-Dezfooli et al. 2017b), universal perturbations for image classification are generated by iteratively optimizing the per-instance adversarial loss for training samples using DeepFool. In addition to classification tasks, universal perturbations are also shown to exist for semantic segmentation (Metzen et al. 2017). Robust universal adversarial examples are generated as a universal targeted adversarial patch in (Brown et al. 2017). They are targeted since they cause misclassification of the images to a given target class. (Moosavi-Dezfooli et al. 2017a) prove the existence of small universal perturbations under certain curvature conditions of decision boundaries. Data-independent universal perturbations are also shown to exist and can be generated by maximizing spurious activations at each layer. These universal perturbations are slightly weaker than the data dependent approaches (Mopuri, Garg, and Babu 2017; Mopuri, Ganeshan, and Radhakrishnan 2018). As a variant of universal perturbation, unconditional generators are trained to create perturbations from random noises for attack (Reddy Mopuri, Krishna Uppala, and Venkatesh Babu 2018; Reddy Mopuri et al. 2018). Universal perturbations are often larger than per-instance perturbation. For example on ImageNet, universal perturbations generated in prior works have ℓ_∞ perturbations of size $\epsilon = 10$ while non-targeted per-instance perturbations as small as $\epsilon = 2$ are often enough to considerably degrade the performance of conventionally trained classifiers possibly due to the fact that ImageNet is a complex dataset and is fundamentally susceptible to per-instance perturbations (Shafahi et al. 2018). As a consequence, from a defense perspective, per-instance defenses on ImageNet focus on smaller perturbations compared to universal perturbations.

There has been very little work on defending against universal attacks. To the best of our knowledge, the only dedicated study is by (Akhtar, Liu, and Mian 2018), who propose a perturbation rectifying network that pre-processes input images to remove the universal perturbation. The rectifying network is trained on universal perturbations that are built for the downstream classifier. While other methods of data sanitization exist (Samangouei, Kabkab, and Chellappa 2018; Meng and Chen 2017) , it has been shown (at least for per-instance adversarial examples) that this type of defense is easily subverted by an attacker who is aware that a defense network is being used (Carlini and Wagner 2017a).

Two recent preprints (Perolat et al. 2018; Mummadi, Brox, and Metzen 2018) model the problem of defending against universal perturbations as a two-player min-max game. However, unlike us, and similar to per-instance adversarial training, after each gradient descent iteration for updating the DNN parameters, they generate a universal adversarial example in an iterative fashion. Since the generation of universal adversarial perturbations can be very time-consuming, this makes their approach slow and prevents

Algorithm 1 Iterative solver for universal perturbations (Moosavi-Dezfooli et al. 2017b)

Initialize $\delta \leftarrow 0$
while Prob$(X, \delta) < 1 - \xi$ **do**
  for x_i in X **do**
    if $f(w, x_i + \delta) \neq f(w, x_i)$ **then**
      Solve $\min_r \|r\|_2$ s.t. $f(w, x_i + \delta + r) \neq f(w, x_i)$ by DeepFool
      Update $\delta \leftarrow \delta + r$, then project δ to ℓ_p ball
    end if
  end for
end while

them from training the DNN parameters for many iterations.

3 Optimization for universal perturbation

Given a set of training samples $X = \{x_i, i = 1, \ldots, N\}$ and a network $f(w, \cdot)$ with frozen parameter w that maps images onto labels, (Moosavi-Dezfooli et al. 2017b) propose to find universal perturbations δ that satisfy,

$$\|\delta\|_p \leq \epsilon \text{ and Prob}(X, \delta) \geq 1 - \xi, \tag{1}$$

Prob(X, δ) represents the "fooling ratio," which is the fraction of images x whose perturbed class label $f(w, x + \delta)$ differs from the original label $f(w, x)$. The parameter ϵ controls the ℓ_p diameter of the bounded perturbation, and ξ is a small tolerance hyperparameter. Problem (1) is solved by the iterative method in algorithm 1. This solver relies on an inner loop to apply DeepFool to each training instance, which makes the solver slow. Moreover, the outer loop of algorithm 1 is not guaranteed to converge. Different from (Moosavi-Dezfooli et al. 2017b), we consider the following optimization problem for building universal perturbations,

$$\max_{\delta} \mathcal{L}(w, \delta) = \frac{1}{N} \sum_{i=1}^{N} l(w, x_i + \delta) \text{ s.t. } \|\delta\|_p \leq \epsilon, \tag{2}$$

where $l(w, \cdot)$ represents the loss used for training DNNs. This simple formulation (2) searches for a universal perturbation that maximizes the training loss, and thus forces images into the wrong class.

The naive formulation (2) suffers from a potentially significant drawback; the cross-entropy loss is unbounded from above, and can be arbitrarily large when evaluated on a single image. In the worst-case, a perturbation that causes misclassification of just a single image can maximize (2) by forcing the average loss to infinity. To force the optimizer to find a perturbation that fools many instances, we propose a "clipped" version of the cross entropy loss,

$$\hat{l}(w, x_i + \delta) = \min\{l(w, x_i + \delta), \beta\}. \tag{3}$$

We cap the loss function at β to prevent any single image from dominating the objective in (2), and giving us a better surrogate of misclassification accuracy. In section 5, we investigate the effect of clipping with different β.

We directly solve eq. (2) by a stochastic gradient method described in algorithm 2. Each iteration begins by using gradient ascent to update the universal perturbation δ so that it maximizes the loss. Then, δ is projected onto the ℓ_p-norm ball to prevent it from growing too large. We experiment with various optimizers for this ascent step, including Stochastic Gradient Descent (SGD), Momentum SGD (MSGD), Projected Gradient Descent (PGD), and ADAM (Kingma and Ba 2014).

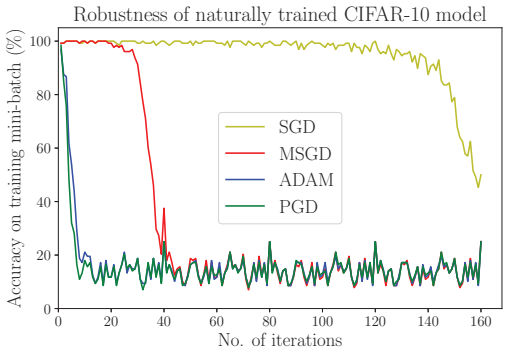


Figure 2: Classification accuracy on adversarial examples of universal perturbations generated by increasing the cross-entropy loss. PGD and ADAM converge faster. We use 5000 training samples from CIFAR-10 for constructing the universal adversarial perturbation for naturally trained WRN model from (Madry et al. 2018). The batch-size is 128, ϵ=8, and the learning-rate/step-size is 1.

Algorithm 2 Stochastic gradient for universal perturbation

for epoch $= 1 \ldots N_{ep}$ **do**
  for minibatch $B \subset X$ **do**
    Update δ with gradient variant $\delta \leftarrow \delta + g$
    Project δ to ℓ_p ball
  end for
end for

We test this method by attacking a naturally trained WRN 32-10 architecture on the CIFAR-10 dataset. We use $\epsilon = 8$ for the ℓ_∞ constraint for CIFAR. Stochastic gradient methods that use "normalized" gradients (ADAM and PGD) are less sensitive to learning rate and converge faster, as shown in fig. 2. We visualize the generated universal perturbation from different optimizers in fig. 3. Compared to the noisy perturbation generated by SGD, normalized gradient methods produced stronger attacks with more well-defined geometric structures and checkerboard patterns. The final evaluation accuracies (on test-examples) after adding universal perturbations with $\epsilon = 8$ were 42.56% for the SGD perturbation, 13.08% for MSGD, 13.30% for ADAM, and 13.79% for PGD. The clean test accuracy of the WRN is 95.2%.

Our proposed method of universal attack using a clipped loss function has several advantages. It is based on a standard stochastic gradient method that comes with convergence guarantees when a decreasing learning rate is used (Bottou, Curtis, and Nocedal 2018). Also, each iteration is based on a minibatch of samples instead of one instance,

(a) SGD (b) MSGD (c) ADAM (d) PGD

Figure 3: Visualizations of universal perturbations after 160 iterations of the optimizers depicted in fig. 2.

which accelerates computation on a GPU. Finally, each iteration requires a simple gradient update instead of the complex DeepFool inner loop; we empirically verify fast convergence and good performance of the proposed method (see section 5).

4 Universal adversarial training

We now consider training robust classifiers that are resistant to universal perturbations. Similar to (Madry et al. 2018), we borrow ideas from robust optimization. We use robust optimization to build robust models that can resist universal perturbations. In particular, we consider universal adversarial training, and formulate this problem as a min-max optimization problem,

$$\min_{w} \max_{\|\delta\|_p \leq \epsilon} \mathcal{L}(w, \delta) = \frac{1}{N} \sum_{i=1}^{N} l(w, x_i + \delta) \qquad (4)$$

where w represents the neural network weights, $X = \{x_i, i = 1, \ldots, N\}$ represents training samples, δ represents universal perturbation noise, and $l(\cdot)$ is the loss function. Here, unlike conventional adversarial training, our δ is a universal perturbation (or, more accurately, mini-batch universal). Previously, solving this optimization problem directly was deemed computationally infeasible due to the large cost associated with generating a universal perturbation (Perolat et al. 2018), but we show that eq. (4) is efficiently solvable by alternating stochastic gradient methods shown in algorithm 3. We show that unlike (Madry et al. 2018), updating the universal perturbation only using a simple step is enough for building universally hardened networks. Each iteration alternatively updates the neural network weights w using gradient descent, and then updates the universal perturbation δ using ascent.

We compare our formulation (4) and algorithm 3 with PGD-based adversarial training, which trains a robust model by optimizing the following min-max problem,

$$\min_{w} \max_{Z} \frac{1}{N} \sum_{i=1}^{N} l(w, z_i) \quad \text{s.t. } \|Z - X\|_p \leq \epsilon. \qquad (5)$$

The standard formulation (5) searches for per-instance perturbed images Z, while our formulation in (4) maximizes using a universal perturbation δ. (Madry et al. 2018) solve (5) by a stochastic method. In each iteration, an adversarial example z_i is generated for an input instance by the PGD iterative method, and the DNN parameter w is updated once. Our formulation (algorithm 3) only maintains one single perturbation that is used and refined across all iterations. For this reason, we need only update w and δ once per step (there is no expensive inner loop), and these updates accumulate for both w and δ through training.

Algorithm 3 Alternating stochastic gradient method for adversarial training against universal perturbation

Input: Training samples X, perturbation bound ϵ, learning rate τ, momentum μ
  for epoch $= 1 \ldots N_{ep}$ **do**
    for minibatch $B \subset X$ **do**
      Update w with momentum stochastic gradient
        $g_w \leftarrow \mu g_w - \mathbb{E}_{x \in B}[\nabla_w \, l(w, x + \delta)]$
        $w \leftarrow w + \tau g_w$
      Update δ with stochastic gradient ascent
        $\delta \leftarrow \delta + \epsilon \, \text{sign}(\mathbb{E}_{x \in B}[\nabla_\delta \, l(w, x + \delta)])$
      Project δ to ℓ_p ball
    end for
  end for

We consider different rules for updating δ during universal adversarial training,

$$\text{FGSM } \delta \leftarrow \delta + \epsilon \cdot \text{sign}(\mathbb{E}_{x \in B}[\nabla_\delta l(w, x + \delta)]), \qquad (6)$$

$$\text{SGD } \delta \leftarrow \delta + \tau_\delta \cdot \mathbb{E}_{x \in B}[\nabla_\delta l(w, x + \delta)], \qquad (7)$$

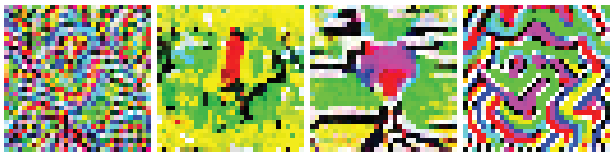
and ADAM. We found that the FGSM update rule was most effective when combined with the SGD optimizer for updating DNN weights w.

We use fairly standard training parameters in our experiments. In our CIFAR experiments, we use $\epsilon = 8$, batchsize of 128, and we train for 80,000 steps. For the optimizer, we use Momentum SGD with an initial learning rate of 0.1 which drops to 0.01 at iteration 40,000 and drops further down to 0.001 at iteration 60,000. One way to assess the update rule is to plot the model accuracy before and after the ascent step (i.e., the perturbation update). It is well-known that adversarial training is more effective when stronger attacks are used. In the extreme case of a do-nothing adversary, the adversarial training method degenerates to natural training. As illustrated in the supplementary, we see a gap between the accuracy curves plotted before and after gradient ascent. We find that the FGSM update rule leads to a larger gap, indicating a stronger adversary. Correspondingly, we find that the FGSM update rule yields networks that are more robust to attacks as compared to SGD update.

Attacking hardened models

We evaluate the robustness of different models by applying algorithm 2 to find universal perturbations. We attack universally adversarial trained models (produced by eq. (4)) using the FGSM universal update rule (uFGSM), or the SGD universal update rule (uSGD). We also consider robust models from per-instance adversarial training (eq. (5)) with adversarial steps of the FGSM and PGD type.

Robust models adversarially trained with weaker attackers such as uSGD and (per-instance) FGSM are relatively vulnerable to universal perturbations, while robust models



(a) FGSM (b) PGD (c) uFGSM (d) uSGD

Figure 4: Universal perturbations made (with 400 iterations) for 4 different CIFAR-10 robust models: adversarially trained with FGSM or PGD, and universally adversarially trained with FGSM (uFGSM) or SGD (uSGD).

Table 1: Validation accuracy of hardened WideResnet models trained on CIFAR-10. Note that Madry's PGD training is significantly slower than the other training methods.

		Validation Accuracy on	
		UnivPert	Natural
(Robust) models trained with	Natural	9.2%	95.2%
	FGSM	51.0%	91.42%
	PGD	86.1%	87.25%
	uADAM (ours)	91.6%	**94.28%**
	uFGSM (ours)	**91.8%**	93.50%

from (per-instance) PGD and uFGSM can resist universal perturbations. We plot the universal perturbations generated using algorithm 3 in fig. 4. When we visually compare the universal perturbations of robust models (fig. 4) and those of a naturally trained model (fig. 3), we can see a drastic change in structure. Similarly, even among hardened models, universal perturbations generated for weaker robust models have more geometric textures, as shown in fig. 4 (a,d).

While an ϵ-bounded per-instance robust model is also robust against ϵ-bounded universal attacks since the universal attack is a more constrained version of the per-instance attack, training robust per-instance models is only possible for small datasets like CIFAR and for small ϵ. For larger datasets like ImageNet, we cannot train per-instance robust models with such large ϵ's common for universal attacks. However, we include the per-instance adversarially trained model as a candidate universally robust model for CIFAR-10 in our comparisons to allow comparisons in settings where it is possible to train per-instance robust models.

We apply the strongest attack to validation images of the natural model and various universal adversarially trained models using different update steps. The results are summarized in table 1. Our models become robust against universal perturbations and have higher accuracies on natural validation examples compared to per-instance adversarially trained models. Their robustness is even more if attacked with iDeepFool (93.29%). Compared to the per-instance FGSM trained model which has the same computational cost as ours, our universal model is more robust. Note that the PGD trained model is trained on a 7-step per-instance adversary and requires about $4\times$ more computation than ours.

Low-cost universal adversarial training

As shown in table 1, our proposed algorithm 3 was able to harden the CIFAR-10 classification network. This comes at the cost of doubling the training time. Adversarial training in general should have some cost since it requires the generation or update of the adversarial perturbation of the mini-batch *before* each minimization step on the network's parameters. However, since universal perturbations are approximately image-agnostic, results should be fairly invariant to the order of updates. For this reason, we propose to compute the image gradient needed for the perturbation update during the same backward pass used to compute the parameter gradients. This results in a simultaneous update for network weights and the universal perturbation in algorithm 4, which backprops only once per iteration and produces approximately universally robust models at almost no cost in comparison to natural training. The "low-cost universal adversarially trained" model of CIFAR-10 is 86.1% robust against universal perturbations and has 93.5% accuracy on the clean validation examples. When compared to the original version in table 1, the robustness has only slightly decreased. However, the training time is cut by half. This is a huge improvement in efficiency, in particular for large datasets like ImageNet with long training times.

Algorithm 4 Simultaneous stochastic gradient method for adversarial training against universal perturbation

Input: Training samples X, perturbation bound ϵ, learning rate τ, momentum μ
Initialize w, δ
for epoch $= 1 \dots N_{ep}$ **do**
  for minibatch $B \subset X$ **do**
    Compute gradient of loss with respect to w and δ
      $d_w \leftarrow \mathbb{E}_{x \in B}[\nabla_w \, l(w, x + \delta)]$
      $d_\delta \leftarrow \mathbb{E}_{x \in B}[\nabla_\delta \, l(w, x + \delta)]$
    Update w with momentum stochastic gradient
      $g_w \leftarrow \mu g_w - d_w$
      $w \leftarrow w + \tau g_w$
    Update δ with stochastic gradient ascent
      $\delta \leftarrow \delta + \epsilon \text{sign}(d_\delta)$
    Project δ to ℓ_p ball
  end for
end for

5 Universal perturbations for ImageNet

To validate the performance of our proposed optimization on different architectures and more complex datasets, we apply algorithm 2 to various popular architectures designed for classification on the ImageNet dataset (Russakovsky et al. 2015). We compare our method of universal perturbation generation with the current state-of-the-art method, iterative DeepFool (iDeepFool for short – alg. 1). We use the authors' code to run the iDeepFool attack on these classification networks. We execute both our method and iDeepFool on the exact same 5000 training data points and terminate both methods after 10 epochs. We use $\epsilon = 10$ for ℓ_∞ constraint following (Moosavi-Dezfooli et al. 2017b), use a step-size of

1.0 for our method, and use suggested parameters for iDeepFool. Similar conclusions could be drawn when we use ℓ_2 bounded attacks. We independently execute iDeepFool since we are interested in the accuracy of the classifier on attacked images – a metric not reported in (Moosavi-Dezfooli et al. 2017b) [1].

Benefits of the proposed method We compare the performance of our stochastic gradient method for eq. (2) and the iDeepFool method for eq. (1). We generate universal perturbations for Inception (Szegedy et al. 2016) and VGG (Simonyan and Zisserman 2014) networks trained on ImageNet, and report the top-1 accuracy in table 2. Universal perturbations generated by both iDeepFool and our method fool networks and degrade the classification accuracy. Universal perturbations generated for the training samples generalize well and cause the accuracy of validation samples to drop. However, when given a fixed computation budget such as number of passes on the training data, our method outperforms iDeepFool by a large margin. Our stochastic gradient method generates the universal perturbations at a much faster pace than iDeepFool. About 20× faster on InceptionV1 and 6× on VGG16 (13× on average).

After verifying the effectiveness and efficiency of our proposed stochastic gradient method[2], we use our algorithm 2 to generate universal perturbations for ResNet-V1 152 (He et al. 2016) and Inception-V3. Our attacks degrade the validation accuracy of ResNet-V1 152 and Inception-V3 from 76.8% and 78% to 16.4% and 20.1%, respectively. The final universal perturbations used for the results presented are illustrated in the supplementary.

Table 2: Top-1 accuracy on ImageNet for natural images, and adversarial images with universal perturbation.

		InceptionV1	VGG16
Natural	Train	76.9%	81.4 %
	Val	69.7%	70.9%
iDeepFool	Train	43.5%	39.5%
	Val	40.7%	36.0%
Ours	Train	**17.2%**	**23.1%**
	Val	**19.8%**	**22.5%**
iDeepFool time (s)		9856	6076
our time (s)		**482**	**953**

The effect of clipping Here, we analyze the effect of the "clipping" loss parameter β in eq. (2). For this purpose, similar to our other ablations, we generate universal perturbations by solving eq. (2) using PGD for Inception-V3 on ImageNet. We run each experiment with 5 random subsets of training data. The accuracy reported is the classification accuracy on the entire validation set of ImageNet after adding the universal perturbation. The results are summarized in fig. 5. The results showcase the value of our proposed loss function for finding universal adversarial perturbations.

[1]They report "fooling ratio" which is the ratio of examples who's label prediction changes after applying the universal perturbation. This has become an uncommon metric since the fooling ratio can increase if the universal perturbation causes an example that was originally miss-classified to become correctly classified.

[2]Unless otherwise specified, we use the sign-of-gradient PGD for our stochastic gradient optimizer in algorithm 2.

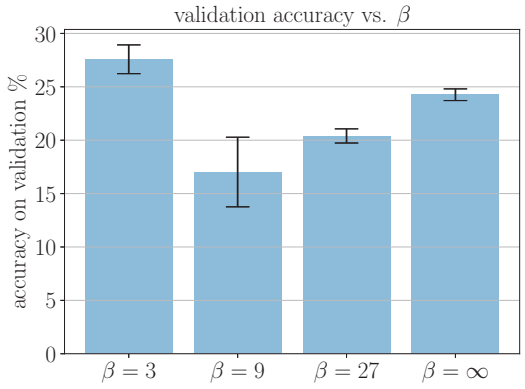


Figure 5: Attack performance varies with clipping parameter β in eq. (2). Attacking Inception-V3 is more successful with clipping ($\beta = 9$) than without clipping ($\beta = \infty$).

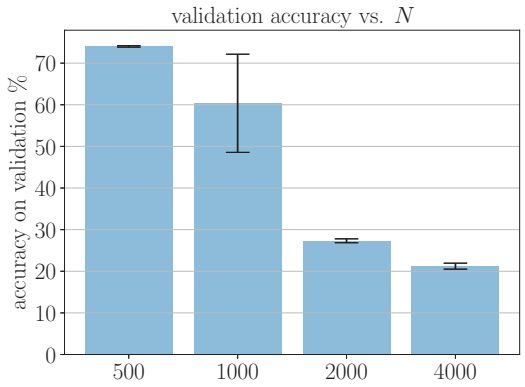
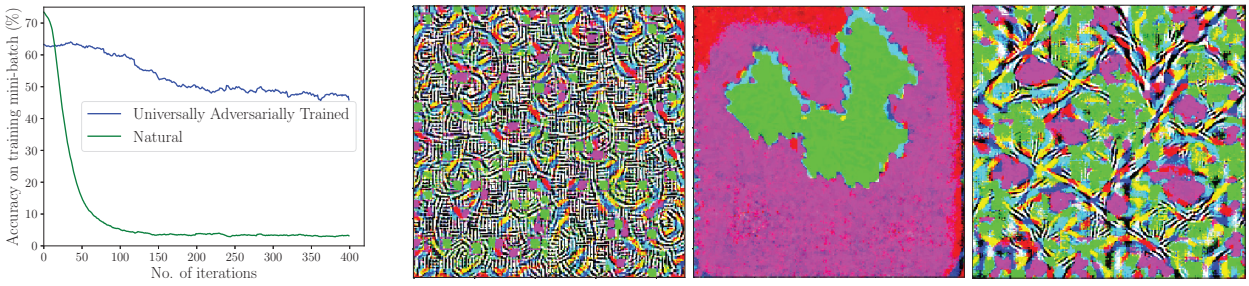


Figure 6: Attack success significantly improves when the number of training points is larger than number of classes. For reference, Inception-V3's top-1 accuracy is 78%. N_{ep} in algorithm 2 was set to 100 epochs for 500 data samples, 40 for 1000 and 2000 samples, and 10 for more.

How much training data does the attack need? As in (Moosavi-Dezfooli et al. 2017b), we analyze how the number of training points ($|X|$) affects the strength of universal perturbations in fig. 6. We build δ using varying amounts of training data. For each experiment, we report the accuracy on the entire validation set after we add the perturbation δ.

6 Universal adversarial training ImageNet

In this section, we analyze our robust models that are universal adversarially trained by solving the min-max problem (section 4) using algorithm 3. We use $\epsilon = 10$ for ImageNet. Note that unlike CIFAR where we were able to train



(a) Generating universal perturbations (b) Natural AlexNet (c) Universal Training (d) Low-cost Universal Tr.

Figure 7: Universal perturbations fool naturally trained AlexNet on ImageNet, but fail to fool our robust AlexNets. The universal perturbations generated for the universal adversarial trained AlexNets have little geometric structure compared to that of naturally trained nets. (b) Universal perturbation of natural model. The accuracy of the validation images + noise is only **3.9**% (c) Perturbation for our universally trained model using algorithm 3. The accuracy of the validation images + noise is **42.0%**. (d) Perturbation for the model trained with low-cost universal training variant (algorithm 4). The accuracy of the validation images + noise is **28.3**%. While the universal noise for the low-cost variant of universal adversarial training has some structure compared to the original, it is less structured than an attack on the natural model (b). Curves smoothed for better visualization.

a per-instance PGD-7 robust model with $\epsilon = 8$, for ImageNet, there exists no model which can resist per-instance non-targeted perturbations with such large ϵ. For ImageNet, we again use fairly standard training parameters (90 epochs, batch-size 256).

Since our universal adversarial training algorithm (algorithm 3) is cheap, it scales to large datasets such as ImageNet. We first train an AlexNet model on ImageNet. We use the natural training hyper-parameters for universal adversarially training our AlexNet model. Also, we separately use our "low-cost universal training" algorithm to train a robust AlexNet with no overhead cost. We then attack the natural, universally trained, and no-cost universally trained versions of AlexNet using universal attacks.

As seen in fig. 7 (a), the AlexNet trained using our universal adversarial training algorithm (algorithm 3) is robust against universal attacks generated using both algorithm 1 and algorithm 2. The naturally trained AlexNet is susceptible to universal attacks. The final attacks generated for the robust and natural models are presented in fig. 7 (b,c,d). The universal perturbation generated for the robust AlexNet model has little structure compared to the universal perturbation built for the naturally trained AlexNet. This is similar to the trend we observed in fig. 3 and fig. 4 for the WRN models trained on CIFAR-10.

The accuracy of the universal perturbations on the validation examples are summarized in table 3. Similar to CIFAR-10, the low-cost version of universal adversarial training is robust but not as robust as the main method. We also train a universally robust ResNet-101 ImageNet model. While a naturally trained ResNet-101 achieves only 7.23% accuracy on universal perturbations, our ResNet-101 achieves 74.43% top1 and 92.00% top5 accuracies.

Table 3: Accuracy on ImageNet for nat and robust models.

Training	Evaluated Against			
	Natural Images		Universal Attack	
	Top-1	Top-5	Top-1	Top-5
Natural	56.4%	79.0 %	3.9 %	9.4 %
Universal	49.5%	72.7%	42.0%	65.8 %
Low-cost U.	48.4%	72.4%	28.3%	48.3 %

7 Conclusion

We proposed using stochastic gradient methods and a "clipped" loss function as an effective universal attack that generates universal perturbations much faster than previous methods. To defend against universal perturbations, we proposed to train robust models by optimizing a min-max problem using alternating or simultaneous stochastic gradient methods. We show that this is possible using certain universal noise update rules that use "normalized" gradients. The simultaneous stochastic gradient method comes at almost no extra cost compared to natural training and has almost no additional cost compared to conventional training.

Acknowledgements Goldstein and his students were supported by the DARPA GARD, DARPA QED for RML, and DAPRA YFA programs. Further support came from the AFOSR MURI program. Davis and his students were supported by the DARPA MediFor program under cooperative agreement FA87501620191, "Physical and Semantic Integrity Measures for Media Forensics". Dickerson was supported by NSF CAREER Award IIS-1846237 and DARPA SI3-CMD Award S4761. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

Akhtar, N.; Liu, J.; and Mian, A. 2018. Defense against universal adversarial perturbations. *CVPR*.

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

Baluja, S., and Fischer, I. 2018. Adversarial transformation networks: Learning to generate adversarial examples. *AAAI*.

Bottou, L.; Curtis, F. E.; and Nocedal, J. 2018. Optimization methods for large-scale machine learning. *SIAM Review* 60(2):223–311.

Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665*.

Carlini, N., and Wagner, D. 2017a. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security*, 3–14. ACM.

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

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Hu, X.; Li, J.; and Zhu, J. 2017. Boosting adversarial attacks with momentum. *CVPR*.

Goldstein, T.; Studer, C.; and Baraniuk, R. 2014. A field guide to forward-backward splitting with a fasta implementation. *arXiv preprint arXiv:1411.3406*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Huang, R.; Xu, B.; Schuurmans, D.; and Szepesvári, C. 2015. Learning with a strong adversary. *arXiv preprint arXiv: 1511.03034*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *ICLR*.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2017. Adversarial machine learning at scale. *ICLR*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. *ICLR*.

Meng, D., and Chen, H. 2017. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 135–147. ACM.

Metzen, J. H.; Kumar, M. C.; Brox, T.; and Fischer, V. 2017. Universal adversarial perturbations against semantic image segmentation. In *ICCV*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; Frossard, P.; and Soatto, S. 2017a. Analysis of universal adversarial perturbations. *arXiv preprint arXiv:1705.09554*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017b. Universal adversarial perturbations. In *CVPR*, 1765–1773.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2574–2582.

Mopuri, K. R.; Ganeshan, A.; and Radhakrishnan, V. B. 2018. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE TPAMI*.

Mopuri, K. R.; Garg, U.; and Babu, R. V. 2017. Fast feature fool: A data independent approach to universal adversarial perturbations. *BMVC*.

Mummadi, C. K.; Brox, T.; and Metzen, J. H. 2018. Defending against universal perturbations with shared adversarial training. *arXiv preprint arXiv:1812.03705*.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 427–436.

Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, 372–387. IEEE.

Perolat, J.; Malinowski, M.; Piot, B.; and Pietquin, O. 2018. Playing the game of universal adversarial perturbations. *arXiv preprint arXiv:1809.07802*.

Poursaeed, O.; Katsman, I.; Gao, B.; and Belongie, S. 2018. Generative adversarial perturbations. *CVPR*.

Reddy Mopuri, K.; Ojha, U.; Garg, U.; and Venkatesh Babu, R. 2018. Nag: Network for adversary generation. In *CVPR*, 742–751.

Reddy Mopuri, K.; Krishna Uppala, P.; and Venkatesh Babu, R. 2018. Ask, acquire, and attack: Data-free uap generation using class impressions. In *ECCV*, 19–34.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *IJCV*.

Saadatpanah, P.; Shafahi, A.; and Goldstein, T. 2019. Adversarial attacks on copyright detection systems. *arXiv preprint arXiv:1906.07153*.

Samangouei, P.; Kabkab, M.; and Chellappa, R. 2018. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *ICLR*.

Shafahi, A.; Huang, W. R.; Studer, C.; Feizi, S.; and Goldstein, T. 2018. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*.

Shafahi, A.; Najibi, M.; Ghiasi, A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial training for free! *Conference on Neural Information Processing Systems (NeurIPS)*.

Shaham, U.; Yamada, Y.; and Negahban, S. 2015. Understanding adversarial training: Increasing local stability of neural nets through robust optimization. *arXiv preprint arXiv:1511.05432*.

Sharif, M.; Bhagavatula, S.; Bauer, L.; and Reiter, M. K. 2016. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 1528–1540. ACM.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*.

Sinha, A.; Namkoong, H.; and Duchi, J. 2018. Certifying some distributional robustness with principled adversarial training. *ICLR*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *ICLR*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *CVPR*, 2818–2826.

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

Wu, Z.; Lim, S.-N.; Davis, L.; and Goldstein, T. 2019. Making an invisibility cloak: Real world adversarial attacks on object detectors. *arXiv preprint arXiv:1910.14667*.

Xiao, C.; Li, B.; Zhu, J.-Y.; He, W.; Liu, M.; and Song, D. 2018. Generating adversarial examples with adversarial networks. *IJCAI*.