A PRINCIPLED APPROACH FOR GENERATING ADVERSARIAL IMAGES UNDER NON-SMOOTH DISSIMILARITY METRICS

ARAM-ALEXANDRE POOLADIAN[1], CHRIS FINLAY[1], TIM HOHEISEL[1], AND ADAM OBERMAN[1]

ABSTRACT. Deep neural networks are vulnerable to adversarial perturbations: small changes in the input easily lead to misclassification. In this work, we propose an attack methodology catered not only for cases where the perturbations are measured by ℓ_p norms, but in fact any adversarial dissimilarity metric with a closed proximal form. This includes, but is not limited to, $\ell_1, \ell_2, \ell_\infty$ perturbations, and the ℓ_0 counting "norm", i.e. true sparseness. Our approach to generating perturbations is a natural extension of our recent work, the LogBarrier attack, which previously required the metric to be differentiable. We demonstrate our new algorithm, ProxLogBarrier, on the MNIST, CIFAR10, and ImageNet-1k datasets. We attack undefended and defended models, and show that our algorithm transfers to various datasets with little parameter tuning. In particular, in the ℓ_0 case, our algorithm finds significantly smaller perturbations compared to multiple existing methods.

1. INTRODUCTION

Deep neural networks are vulnerable to adversarial perturbations: "imperceptibly small" (measured via a dissimilarity metric) changes in the model input lead to misclassification [30, 13]. The existence of small norm adversarial attacks can be interpreted to mean that while models generalize well on natural images, and are robust to random perturbations, nevertheless they lose accuracy on worst-case perturbations. The vulnerability of DNNs is a potentially grave security risk. In developing strong perturbations, we hope that better defense mechanisms will be deployed to prevent these attacks from occurring in practice.

Adversarial attacks are often broadly categorized into one of two types: white-box attacks, where the full structure of the neural network is provided to the attacker, including gradient information, or black-box attacks, where the attacker is only given the model decision. One of the first proposed adversarial attacks is the Fast Gradient Signed Method (FGSM), which generates an adversarial image with respect to the ℓ_∞ norm, along with its iterative form, called Iterative FGSM (IFGSM) [13, 17]. A similar iterative attack was also done with respect to the ℓ_2 norm. In their purest form, the above attacks perform gradient ascent on the loss function subject to a norm constraint on the perturbation, either with one step in the case of FGSM and multiple steps in the case of IFGSM, and the ℓ_2 norm equivalent. Apart from loss maximization, attacks have been developed using loss functions that *directly* measure misclassification [5, 22]. There is also the problem of generating "realistic" attacks, such as through sparse attacks. These include for example, small stickers on a road sign, which may tamper with autonomous vehicles [10]. In the black-box setting, adversarial examples are generated using only model outputs or model decisions, which is a much more expensive endeavor. However, black-box methods can sometimes perform better, most notably by

[1] DEPARTMENT OF MATHEMATICS AND STATISTICS, MCGILL UNIVERSITY

E-mail address: aram-alexandre.pooladian@mail.mcgill.ca, christopher.finlay@mail.mcgill.com, tim.hoheisel@mcgill.ca, adam.oberman@mcgill.ca.

AO supported by AFOSR grant FA9550-18-1-0167.

avoiding gradient obfuscation, and take advantage of sampling properties near the decision boundary. Notable examples of black-box (decision-based) attacks are the Boundary Attack [4] and the recent HopSkipJumpAttack [6].

The development of new and improved adversarial attacks has occurred in parallel with various defensive training regiments to provide robustness against adversarial perturbations. The task of training a robust network is two-fold: models must be resistant to perturbations of a certain magnitude, while also maintaining classification ability on clean data. It has been argued that these two objectives are inherently "at odds" [31]. A popular method for training robust networks is *adversarial training*, where adversarial examples are added to the training data (see for example [19]). While effective, adversarial training has not scaled well with increasing network size. Two recent methods have tackled this problem for large networks on ImageNet-1k [29, 11]; this was previously not possible without a massive computational infrastructure [32, 15].

Contributions. This paper introduces an attack methodology catered for not just ℓ_p norms, but any adversarial dissimilarity metric with a closed proximal form. This includes, but is not limited to, $\ell_1, \ell_2, \ell_\infty$ perturbations and the ℓ_0 counting "norm", i.e. a true measurement of sparseness of the perturbation. Our approach adopts the relaxation structure of the recently proposed LogBarrier attack [12], which required differentiable metrics. We extend this work to include a broad class of non-smooth (non-differentiable) metrics. Our algorithm, ProxLogBarrier, uses the proximal gradient method for generating adversarial perturbations. We demonstrate our attack on MNIST, CIFAR10, and ImageNet-1k datasets. ProxLogBarrier shows significant improvement over both the LogBarrier attack, and over the other attacks we considered. In particular, in the ℓ_0 case, we achieve state-of-the-art results with respect to a suite of attacks typically used for this problem class.

2. Problem formulation of adversarial attacks and background

Let $\mathcal{X}$ be the image space, and Δ_c be the label space (the unit-simplex for c classes). An image-label pair is defined by $(x, y) \in \mathcal{X} \times \Delta_c$, with the image belonging to one of c classes. The trained model is defined by $f : \mathcal{X} \to \Delta_c$. An adversarial perturbation is supposed to be small with respect to a *dissimilarity metric* (henceforth simply called the metric) $m(\cdot; x)$, e.g. $\|\cdot - x\|_\infty$. Formally, the optimal adversarial perturbation is the minimizer of the following constrained optimization problem:

$$\min_{u \in \mathcal{X}} m(u; x) \quad \text{subject to} \quad C(f(u)) \neq y, \tag{1}$$

where $C(f(\cdot))$ is the classification function for the trained network. When using the cross-entropy classification function, problem (1) can be written as

$$\min_{u \in \mathcal{X}} m(u; x) \quad \text{subject to} \quad \max_{i \neq y} [f(u)]_i > [f(u)]_y. \tag{2}$$

DNNs might be powerful classifiers, but that does not mean their decision boundaries are well-behaved. Instead, researchers have popularized using the cross-entropy loss as a surrogate for the decision boundary: typically a model is trained until the loss is very low, which is often related to good classification performance. Thus, instead of solving (1), one can perform Projected Gradient Ascent (PGA) on the cross-entropy loss:

$$\max_{u \in \mathcal{X}} \mathcal{L}(u) \quad \text{subject to} \quad m(u; x) \leq \varepsilon, \tag{3}$$

where $m(\cdot; x)$ is typically taken to be either the ℓ_2 or ℓ_∞ norm, and ε is defines the perturbation threshold of interest.

Some adversarial attack methods try to solve the problem posed in (1) without incorporating the loss function used to train the network. For example, Carlini & Wagner attacked the logit-layer of a network and solves a different optimization problem, depending on the choice of norm [5]. With regards to adversarial defense, they demonstrated how a significant number of adversarial defense methods fail because of "gradient obfuscation", where gradients are small only locally to the image. However, this is in fact an artifact of the softmax layer [1]. Another metric of adversarial dissimilarity is the ℓ_0 "norm", which counts the number of total different pixels between the adversary and the clean image [21, 24]. This is of interest because an adversary might have to also budget the number of allowed pixels to perturb, while still remaining "imperceptible" to the human eye. For example, the sticker-attack [10] is a practical attack with real-world consequences, and does not interfere with every single part of the image.

3. OUR METHOD: PROXLOGBARRIER

We consider the formulation of adversarial attacks as in (2),

$$\min_{u \in \mathcal{X}} \; m(u; x) \quad \text{s.t.} \quad F(u) := \max_{i \neq y} [Z(u)]_i - [Z(u)]_y > 0. \tag{4}$$

Here, we abbreviate $Z(\cdot)$ is the model output *before* the softmax layer that "projects" onto Δ_c. However, since softmax is a monotonic operator, one can also use the probability outputs as well. This problem is difficult as the constraints have virtually no exploitable structure. The problem can be relaxed using a logarithmic barrier, a technique often used in traditional optimization [23],

$$\min_{u \in \mathcal{X}} \; m(u; x) - \lambda \log(F(u)). \tag{5}$$

This objective function now includes the constraint that enforces misclassification. In [12], (5) was originally solved via gradient descent, which necessarily assumes that $m(\cdot; x)$ is at least differentiable. Most dissimilarity metrics are not differentiable; for example the ℓ_∞ norm. In the original LogBarrier paper [12], a smooth approximation of this norm was used to get around this issue, but required far too many iterations to make progress.

For brevity, let $\varphi(\cdot) := -\log(\cdot)$, then optimization problem (5) becomes

$$\min_{u \in \mathcal{X}} \; m(u; x) + \lambda \varphi(F(u)). \tag{6}$$

This relaxed problem has a composite structure, with $\varphi \circ F$ being smooth provided $F(\cdot) \in \operatorname{dom}(\varphi)$. We turn to the *proximal gradient method* to efficiently solve this problem and outline this method in the following section. Instead of requiring $m(\cdot; x)$ to be differentiable, we only require a closed proximal form. This assumption is satisfied for most of the dissimilarity metrics considered in the adversarial attack literature.

3.1. **Proximal gradient method.**

Proximal algorithms are a driving force for nonsmooth optimization problems, and are receiving more attention in the deep learning community on a myriad of problems [2, 34, 20, 25]. For a full discussion on this topic, we suggest [3].

We consider the following framework for proximal algorithms, namely a composite minimization problem

$$\min_{x\in\mathcal{E}} \Phi(x) := f(x) + g(x) \tag{7}$$

where $\mathcal{E}$ is a Euclidean space. We make the following assumptions:

- g is a non-degenerate, closed, and convex function over $\mathcal{E}$
- f is non-degenerate, closed function, with $\text{dom}(f)$ convex, and has L-Lipschitz gradients over the interior of its domain
- $\text{dom}(g) \subseteq \text{int}(\text{dom}(f))$
- the solution set, S, is non-empty.

Solving this composite problem with gradient descent is not advisable, since g is not necessarily differentiable. The best one can hope for is that g has a *subgradient* at $x \in \mathcal{E}$, defined as an element $v \in \mathcal{E}$ such that

$$g(y) \geq g(x) + \langle v, y - x\rangle \quad (y \in \mathcal{E}). \tag{8}$$

The collection of subgradients of g is called the *subdifferential* of g, denoted by $\partial g(\cdot)$. When a function is differentiable, the subdifferential is a singleton, namely $\partial f(x) = \{\nabla f(x)\}$. One could turn to subgradient descent to solve the composite problem, however a subgradient might not always be helpful. For example, consider the subgradient of ℓ_∞ for an element in $\mathbb{R}^n$;

$$\partial\|\cdot\|_\infty(x) = \text{sign}(x_k)e_k,$$

where $k := \arg\max_i\{|x_i|\}$, and $\{e_i\}_{i=1}^n$ are the standard basis vectors. At each subgradient step, very little information is obtained.

Since Φ is a non-convex problem (because f is potentially not convex), our goal is to iteratively generate a sequence $\{x^{(k)}\}$ that converges to $x^* \in S$, where x^* is a *stationary point* i.e. $0 \in \partial\Phi(x^*)$. A characterization of these stationary points is the following fixed-point representation (we take $\lambda > 0$):

$$\begin{aligned} 0 \in \partial\Phi(x^*) &\iff 0 \in \nabla f(x^*) + \partial g(x^*) \\ &\iff -\lambda\nabla f(x^*) \in \lambda\partial g(x^*) \\ &\iff x^* - \lambda\nabla f(x^*) \in x^* + \lambda\partial g(x^*) = (\text{Id} + \lambda\partial g)(x^*) \\ &\iff x^* = (\text{Id} + \lambda\partial g)^{-1}\left(x^* - \lambda\nabla f(x^*)\right) \end{aligned}$$

where $(\text{Id} + \lambda\partial g)^{-1}(\cdot) =: \text{Prox}_{\lambda g}(\cdot)$ is defined as the *proximal operator of* g

$$\text{Prox}_{\lambda g}(x) := \text{argmin}_{u\in\mathcal{E}}\left\{g(u) + \frac{1}{2\lambda}\|x - u\|_2^2\right\} \quad (\lambda > 0). \tag{9}$$

The first line in the equivalence chain uses addition of subdifferentiability, which is guaranteed by our assumptions, and the rest is algebraic manipulation. Thus, to generate a stationary point, it suffices to find a fixed point of the sequence generated in the following manner:

$$x^{(k+1)} = \text{Prox}_{t_k g}(x^{(k)} - t_k\nabla f(x^{(k)})), \tag{10}$$

where $t_k > 0$ is some step size. The proximal operator exists for any convex function, but this is not a strict requirement.

Despite f not being convex, there are still convergence properties of the sequence of iterates generated in this way. The following theorem is a simplified version of what can be found in [3] (Section 10.3 with proof), and is the main motivation for our proposed method.

Theorem 1. *Given the assumptions on (7), let $\{x^k\}_{k\geq 0}$ be the sequence generated by (10), with fixed step size $t_k := d \in (\frac{L}{2}, \infty)$. Then,*

(a) the sequence $\{\Phi(x^k)\}_{k\geq 0}$ is non-increasing. In addition, $\Phi(x^{k+1}) < \Phi(x^k)$ if and only if x^k is not a stationary point of (7);

(b) $d\left(x^k - Prox_{\frac{1}{d}}(x^k - \frac{1}{d}\nabla f(x^k))\right) \to 0$ as $k \to \infty$;

(c) all limit points of the sequence $\{x^k\}_{k\geq 0}$ are stationary points of (7).

3.2. **ProxLogBarrier attack algorithm.** We iteratively find a minimizer for (6); the attack is outlined in Algorithm 1. Due to the highly non-convex nature of the decision boundary, we perform a backtracking step to ensure the proposed iterate is in fact adversarial, thus making $\varphi \circ F$ smooth. We remark that the adversarial attack problem is constrained by the image-space, and thus requires a further projection step back onto the image space (we consider pixels to be in the range [0,1]). In traditional non-convex optimization, best practice is to also record the "best iterate", as valleys are likely pervasive throughout the decision boundary. This way, even at some point our gradient sends the iterate far-off and is unable to return in the remaining iterations, we already have a better candidate.

Algorithm 1 ProxLogBarrier (PLB)

Input: image-label pair (x, y), trained model f, adversarial dissimilarity metric $m(\cdot; x)$
Intialize hyperparameters: $K_{\text{inner}}, K \in \mathbb{N}$, and $\mu, h, \lambda_0 > 0, \beta \in (0, 1)$.
Initialize $u^{(0)}$ to be misclassified, $w^{(0)} := u^{(0)}$
for $k = 0, 1, 2, \ldots, K$ **do**
  Every K_{inner} iterations: $\lambda = \lambda_0 \beta^k$
  $y^{(k)} = \text{Prox}_{\mu m}\left(u^{(k)} - h\lambda\nabla\varphi(F(u^{(k)}))\right)$
  $u^{(k+1)} = \text{Project}(y^{(k)}; \mathcal{X})$
  Backtrack along line between current and previous iterate until misclassified
  if $m(u^{(k+1)}; x) < m(w^{(k)}; x)$ **then**
    $w^{(k+1)} = u^{(k+1)}$
  else
    $w^{(k+1)} = w^{(k)}$
  end if
end for
Output: $w^{(K)}$

Proximal operators for metrics of interest. To complete the algorithm, it remains to compute the proximal operator $\text{Prox}_{\mu m}(\cdot)$ for various choices of m. One can turn to [3] for complete derivations of the proximal operators for the adversarial metrics we are considering, namely $\ell_1, \ell_2, \ell_\infty$ norms, and the ℓ_0 cardinality function. Consider measuring the ℓ_∞ distance between the clean image and our desired adversarial perturbation:

$$m(u; x) := \|u - x\|_\infty.$$

Due to the Moreau Decomposition Theorem [27], the proximal operator of this function relies on projecting onto the unit ℓ_1 ball:

$$\begin{aligned}\text{Prox}_{\mu\|\cdot - x\|_\infty}(z) &= x + \text{Prox}_{\mu\|\cdot\|_\infty}(z - x) \\ &= x + (z - x) - \mu\text{Prox}_{\mathbb{B}_1}((z - x)/\mu) \\ &= z - \mu\text{Proj}_{\|\cdot\|_1}((z - x)/\mu).\end{aligned}$$

We make use of the algorithm from [9] to perform the projection step, implemented over batches of vectors for efficiency. Similarly, one obtains the proximal operator for ℓ_1 and ℓ_2 via the same theorem,

$$\begin{aligned}\text{Prox}_{\mu\|\cdot - x\|_1}(z) &= x + \mathcal{T}_\mu(z - x), \\ \text{Prox}_{\mu\|\cdot - x\|_2}(z) &= z - \mu\text{Proj}_{\|\cdot\|_2}((z - x)/\mu),\end{aligned}$$

where $\mathcal{T}_\mu(s) := \text{sign}(s)\max\{|s| - \mu, 0\}$ is the well-known soft thresholding operator. In the case that one wants to minimize the number of perturbed pixels in the adversarial image, one can turn to the counting "norm", called ℓ_0, which counts the number of non-zero entries in a vector. While this function is non-convex, the proximal operator still has a closed form:

$$P_{\mu\|\cdot - x\|_0}(z) = x + \mathcal{H}_{\sqrt{2\mu}}(z - x)$$

where $\mathcal{H}_\alpha(s) = s\mathbf{1}_{\{|s|>\alpha\}}(s)$ is a hard-thresholding operator, and acts component-wise in the case of vector arguments.

4. Experimental methodology

Outline. We compare the ProxLogBarrier attack with several other adversarial attacks on MNIST [18], CIFAR10 [16], and ImageNet-1k [8] datasets. For MNIST, we use the network described in [24]; on CIFAR10, we use a ResNeXt network [33]; and for ImageNet-1k, ResNet50 [14, 7]. We also consider defended models for the aforementioned networks. This is to further benchmark the attack capability of the ProxLogBarrier, and to reaffirm previous work in the area. For defended models, we consider Madry-style AT for CIFAR10 and MNIST [19]. On ImageNet-1k, we use the recently proposed scaleable input gradient regularization for adversarial robustness [11]. We randomly select 1000 (test) images to evaluate performance on MNIST, CIFAR10, and 500 (test) images on ImageNet-1k. We consider the same images on their defended counterparts. We note that for ImageNet-1k, we consider the problem of Top5 misclassification, where the logrithmic barrier is with respect to the following constraint set

$$Z[u]_{(5)} - Z[u]_{(y)} > 0,$$

where (i) denotes the i^{th} largest index.

We compare the ProxLogBarrier attack with a wide range of attack algorithms that are available through the FoolBox adversarial attack library [26]. For perturbations in ℓ_0, we compare against SparseFool [21], Jacobian Saliency Map Attack (JSMA) [24], and Pointwise [28] (this latter attack is black-box). For ℓ_2 attacks, we consider Carlini-Wagner's attack (CW) [5], Projected Gradient Ascent (PGA) [17], DeepFool [22], and the original LogBarrier attack [12]. Finally, for ℓ_∞ norm perturbations, we consider PGA, DeepFool, and LogBarrier. All hyperparameters are left to their implementation defaults, with the exception of SparseFool, where we used the exact parameters indicated in the paper.

Implementation details for our algorithm. When optimizing for ℓ_2 based noise, we initialize the adversarial image with sufficiently large Gaussian noise; for ℓ_∞ and ℓ_0 based perturbations, we use uniform noise. For hyper-parameter defaults, we recommend $\lambda_0 = 0.1, \beta = 0.75, h = 0.1, \mu = 1$, with $K = 900, K_{\text{inner}} = 30$, which transfer well across all datasets and are the same parameters used in this paper. We observed some computational drawbacks for ImageNet-1k: firstly, the proximal operator for the ℓ_0 norm is far too strict. We decided to use the ℓ_1 norm to induce sparseness in our adversarial perturbation (changing both the prox parameter and the step size to 0.5). Other parameter changes for the ImageNet-1k dataset is the proximal parameter in the ℓ_∞ case, we set $\mu = 3$ and we used 2500 algorithm iterations. Finally, we found that using the softmax layer outputs helps with ImageNet-1k attacks against both the defended and undefended network.

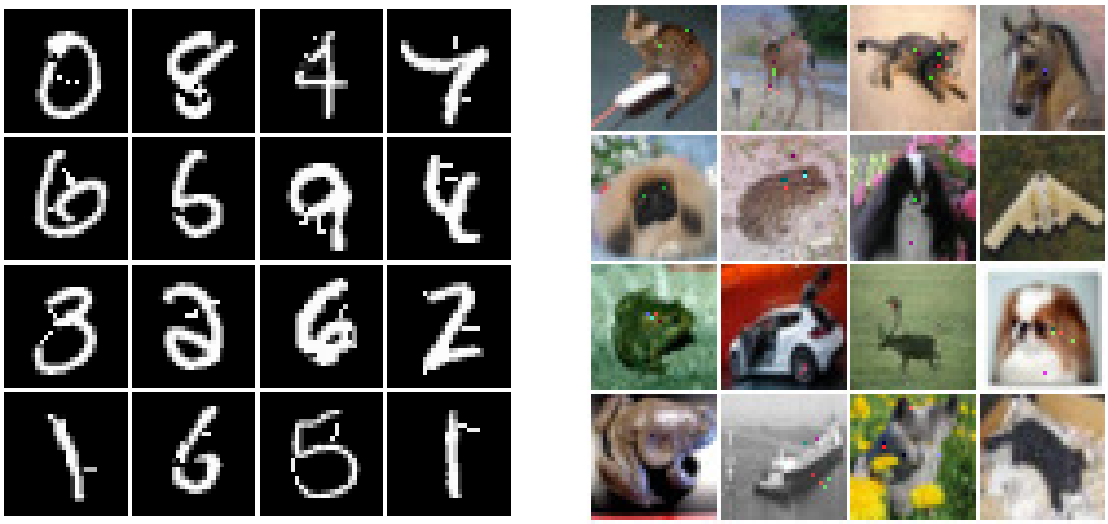
Reporting. For perturbations in ℓ_2 and ℓ_∞, we report the percent misclassification at various threshold levels that are somewhat standard [31]. Our choices for ℓ_0 distance thresholds were arbitrary, however we supplement with a median perturbation distances for all attack norms to mitigate cherry-picking. For attacks that were unable to successfully perturb at least half the sampled images, we do not report anything. If the attack was able to perturb more than half but not all, we add an asterisk to the median distance. We denote the defended models by "(D)" (recall that for MNIST and CIFAR10, we are using Madry's adversarial training, and scaleable input-gradient regularization for Imagenet-1k).

4.1. **Results.**

Perturbations in ℓ_0: Our results for ℓ_0 attacks are found in Table 1, with examples available in Figure 1 and Figure 2. Across all datasets considered, ProxLogBarrier outperforms all other attack methods, for both defended and undefended networks. It also seems undeterred from both Madry-style adversarial training on MNIST and CIFAR10. This is entirely reasonable, for the Madry-style adversarial training is targeted towards ℓ_∞ attacks. In contrast, on ImageNet-1k, the defended model trained with input-gradient regularization performs significantly better than the undefended model, even though this defence is not aimed towards ℓ_0 attacks. Neither JSMA or Pointwise scale to networks on ImageNet-1k. Pointwise exceeds at smaller images, since it takes less than 1000 iterations to cycle over every pixel and check if it can be zeroed out. We remark that SparseFool was unable to adversarially attack all images, whereas ProxLogBarrier always succeeded.

TABLE 1. Adversarial robustness statistics, measured in the ℓ_0 norm.

	MNIST			CIFAR10			ImageNet		
	% error at		median distance	% error at		median distance	% error at		median distance
	$\varepsilon = 10$	$\varepsilon = 30$		$\varepsilon = 30$	$\varepsilon = 80$		$\varepsilon = 500$	$\varepsilon = 1000$	
PLB	**86.30**	**100**	**6**	**44.10**	**68.50**	**39**	**66.00**	**80.20**	**268**
SparseFool	46.00	99.40	11	15.60	22.60	3071	30.40	46.80	1175*
JSMA	12.73	61.38	25	29.56	48.92	84	—	—	—
Pointwise	5.00	57.30	28	13.20	50.60	80	—	—	—
(D) PLB	**79.8**	**98.90**	**6**	**74.90**	**97.80**	**13**	**38.40**	**70.0**	**691**
(D) SparseFool	20.67	75.45	20	34.23	52.15	70	24.80	41.80	1310*
(D) JSMA	12.63	44.51	34	36.65	60.79	53	—	—	—
(D) Pointwise	12.50	65.80	24	23.80	43.10	102	—	—	—



(A) ℓ_0 attacks on MNIST (B) ℓ_0 attacks on CIFAR10

FIGURE 1. Adversarial images for ℓ_0 perturbations, generated by our method.

Perturbations in ℓ_∞: Results for ℓ_∞ perturbations are found in Table 2. Our attack stands out on MNIST, in both the defended and undefended case. On CIFAR10, our attack is best on the undefended network, and only slightly worse than PGA when adversarially defended. On ImageNet-1k, our method suffers dramatically. This is likely due to very poor decision boundaries on the with respect to this norm ℓ_∞, as our method will necessarily be better when the boundaries are not muddled. PGA does not focus on the decision boundaries explicitly, thus has more room to find something adversarial quickly.

TABLE 2. Adversarial robustness statistics, measured in the ℓ_∞ norm.

	MNIST			CIFAR10			ImageNet		
	% error at		median distance	% error at		median distance	% error at		median distance
	$\varepsilon = 0.1$	$\varepsilon = 0.3$		$\varepsilon = \frac{2}{255}$	$\varepsilon = \frac{8}{255}$		$\varepsilon = \frac{2}{255}$	$\varepsilon = \frac{8}{255}$	
PLB	10.30	**100**	**1.67e−1**	**95.00**	**98.60**	**2.88e−3**	20.40	33.80	6.66e−2
PGA	**10.70**	80.90	1.76e−1	54.70	87.00	5.91e−3	90.80	98.60	**2.5e−3**
DeepFool	8.12	86.55	2.25e−1	16.23	51.00	3.04e−2	**93.64**	**100**	2.8e−3
LogBarrier	5.89	73.90	2.43e−1	60.60	93.10	6.84e−3	7.60	7.70	6.16e−1
(D) PLB	**3.0**	**32.9**	**3.24e−1**	23.3	44.1	3.64e−2	11.40	18.80	1.06e−1
(D) PGA	2.8	23.6	3.37e−1	22.9	**46.1**	**3.45e−2**	**49.20**	96.60	**7.94e−3**
(D) DeepFool	2.7	10.2	6.66e−1	**23.8**	44.1	3.74e−2	43.20	**97.40**	9.31e−3
(D) LogBarrier	2.50	11.89	5.48e−1	17.6	28.3	8.01e−2	9.80	10.40	4.43e−1

Perturbations in ℓ_2: Results for perturbations measured in Euclidean distance are found in Table 3. For MNIST and ImageNet-1k, on both defended and undefended networks, our attack performs better than all other methods, both in median distance and at a given perturbation norm threshold. On CIFAR10, we are best on undefended but lose to CW in the defended case. However, the CW attack did not scale to ImageNet-1k using the implementation in the FoolBox attack library.

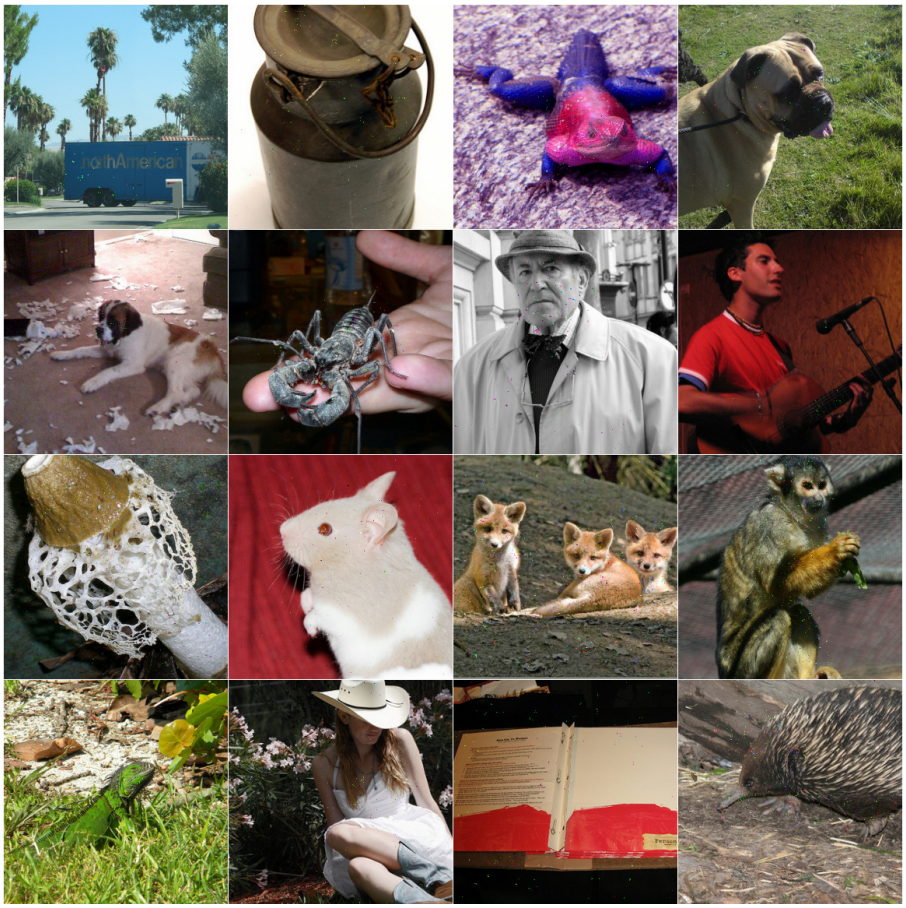


FIGURE 2. Examples of adversarial perturbations on ProxLogBarrier in the ℓ_0 adversarial metric on ImageNet-1K, where these perturbations have a pixel perturbation count of under 1000 out of 178608 total pixels.

TABLE 3. Adversarial robustness statistics, measured in the ℓ_2 norm.

	MNIST			CIFAR10			ImageNet		
	% error at		median distance	% error at		median distance	% error at		median distance
	$\varepsilon = 1.25$	$\varepsilon = 2.3$		$\varepsilon = \frac{80}{255}$	$\varepsilon = \frac{120}{255}$		$\varepsilon = 0.5$	$\varepsilon = 1$	
PLB	**38.60**	**99.40**	**1.35**	**97.70**	**99.80**	**1.15e−1**	**47.60**	**89.40**	**5.24e−1**
CW	35.10	98.30	1.41	89.94	95.97	1.32e−1	20.06	44.26	1.16
PGA	24.70	70.00	1.70	60.60	73.30	2.10e−1	37.60	70.60	6.72e−1
DeepFool	13.21	48.04	2.35	17.33	22.04	1.11	40.08	76.48	6.23e−1
LogBarrier	37.40	98.90	1.35	69.60	84.00	2.02e−1	43.70	88.30	5.68e−1
(D) PLB	**29.50**	**92.90**	**1.54**	28.7	35.4	7.26e−1	**15.80**	**28.20**	**1.74**
(D) CW	28.24	78.59	1.72	**29.6**	**38.7**	**6.60e−1**	—	—	—
(D) PGA	17.20	45.70	2.44	28.30	34.70	7.97e−1	14.60	22.60	2.20
(D) DeepFool	5.22	18.07	3.73	28.0	33.3	9.31e−1	15.60	24.40	2.14
(D) LogBarrier	25.00	89.60	1.65	28.0	34.6	7.36e−1	10.00	10.20	63.17

Algorithm runtime: We strove to implement ProxLogBarrier so that it could be run in a reasonable amount of time. For that reason, ProxLogBarrier was implemented to work over a batch of images. Our code is publicly available at the following `https://github.com/APooladian/ProxLogBarrierAttack`. Using one consumer grade GPU, we can comfortably attack several MNIST and CIFAR10 images simultaneously, but only one ImageNet-1k image at a given time. We report our algorithm runtimes in Table 4, that achieve the statistics mentioned in the previous tables. Most of the other algorithms were implemented taken from the FoolBox repository, and were not written to take advantage of the GPU. Hence we omit a comparison with the other attack algorithms based on run time. Heuristically speaking, PGA is one of the faster algorithms, whereas CW, SparseFool, and DeepFool are slower. We are not surprised that our attack in ℓ_0 takes longer than the other norms; this is likely due to the backtracking step to ensure misclassification of the iterate. On ImageNet-1k, the ProxLogBarrier attack in the ℓ_∞ metric is quite slow due to the projection step onto the ℓ_1 ball, which is $\mathcal{O}(n\log(n))$, where n is the input dimension size [9].

TABLE 4. ProxLogBarrier attack runtimes (in seconds)

	Batch Size	ℓ_0	ℓ_2	ℓ_∞
MNIST	100	8.35	6.91	6.05
CIFAR10	25	69.07	56.11	30.87
ImageNet-1k	1	35.45	29.47	75.50

5. CONCLUSION

We have presented a concise framework for generating adversarial perturbations by incorporating the proximal gradient method. We have expanded upon the LogBarrier attack, which was originally only effective in ℓ_2 and ℓ_∞ norms, by addressing the ℓ_0 norm case as well. Thus we have proposed a method unifying all three common perturbation scenarios. Our approach requires fewer hyperparameter tweaks than LogBarrier, and performs significantly better than many attack methods we compared against, both on defended and undefended models, and across all norm choices. We highlight that our method is, to our knowledge, the best choice for perturbations measured in ℓ_0, compared to all other methods available in FoolBox. We also perform better than all other attacks considered on the MNIST network with respect to median distance and commonly reported thresholds. While our paper focuses on ℓ_p adversarial metrics, it is worth noting that the proximal gradient method can open doors to potentially new adversarial metrics, provided they have closed proximal form.

REFERENCES

[1] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 274–283, 2018.

[2] Yu Bai, Yu-Xiang Wang, and Edo Liberty. Proxquant: Quantized neural networks via proximal operators. *CoRR*, abs/1810.00861, 2018.

[3] Amir Beck. *First-order methods in optimization*. 2017.

[4] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[5] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644, 2016.

[6] Jianbo Chen and Michael I. Jordan. Boundary attack++: Query-efficient decision-based adversarial attack. *CoRR*, abs/1904.02144, 2019.

[7] Cody Coleman, Daniel Kang, Deepak Narayanan, Luigi Nardi, Tian Zhao, Jian Zhang, Peter Bailis, Kunle Olukotun, Christopher Ré, and Matei Zaharia. Analysis of dawnbench, a time-to-accuracy machine learning performance benchmark. *CoRR*, abs/1806.01427, 2018.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255, 2009.

[9] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 272–279, New York, NY, USA, 2008. ACM.

[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945*, 2017.

[11] Chris Finlay and Adam M Oberman. Scaleable input gradient regularization for adversarial robustness. *arXiv preprint arXiv:1905.11468*, 2019.

[12] Chris Finlay, Aram-Alexandre Pooladian, and Adam M. Oberman. The logbarrier adversarial attack: making effective use of decision boundary information. *CoRR*, abs/1903.10396, 2019.

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

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing.

[15] Harini Kannan, Alexey Kurakin, and Ian J. Goodfellow. Adversarial logit pairing. *CoRR*, abs/1803.06373, 2018.

[16] Alex Krizhevsky and Geoffrey Hinton. *Learning multiple layers of features from tiny images*. 2009.

[17] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016.

[18] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, page 319, 1999.

[19] 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.

[20] Tim Meinhardt, Michael Möller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. *CoRR*, abs/1704.03488, 2017.

[21] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. *CoRR*, abs/1811.02248, 2018.

[22] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *CoRR*, abs/1511.04599, 2015.

[23] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[24] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. *CoRR*, abs/1511.07528, 2015.

[25] Courtney Paquette, Hongzhou Lin, Dmitriy Drusvyatskiy, Julien Mairal, and Zaid Harchaoui. Catalyst for gradient-based nonconvex optimization. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 613–622, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[26] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models. *CoRR*, abs/1707.04131, 2017.

[27] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[28] Lukas Schott, Jonas Rauber, Wieland Brendel, and Matthias Bethge. Robust perception through analysis by synthesis. *CoRR*, abs/1805.09190, 2018.

[29] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John P. Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *CoRR*, abs/1904.12843, 2019.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[31] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

[32] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *CoRR*, abs/1812.03411, 2018.

[33] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431, 2016.

[34] Pu Zhao, Kaidi Xu, Sijia Liu, Yanzhi Wang, and Xue Lin. Admm attack: An enhanced adversarial attack for deep neural networks with undetectable distortions. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, ASPDAC '19, pages 499–505, New York, NY, USA, 2019. ACM.