A principled approach for generating adversarial images under non-smooth dissimilarity metrics

Aram-Alexandre Pooladian*, **Chris Finlay, Tim Hoheisel, and Adam M Oberman**
McGill University, Department of Mathematics and Statistics

Abstract

Deep neural networks perform well on real world data but are prone to adversarial perturbations: small changes in the input easily lead to misclassification. In this work, we propose an attack methodology 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, ℓ_1, ℓ_2, and ℓ_∞ perturbations; the ℓ_0 counting "norm" (i.e. true sparseness); and the total variation seminorm, which is a (non-ℓ_p) convolutional dissimilarity measuring local pixel changes. Our approach is a natural extension of a recent adversarial attack method, and eliminates the differentiability requirement of the metric. We demonstrate our algorithm, ProxLogBarrier, on the MNIST, CIFAR10, and ImageNet-1k datasets. We consider undefended and defended models, and show that our algorithm easily transfers to various datasets. We observe that ProxLogBarrier outperforms a host of modern adversarial attacks specialized for the ℓ_0 case. Moreover, by altering images in the total variation seminorm, we shed light on a new class of perturbations that exploit neighboring pixel information.

1 Introduction

Deep neural networks (DNNs) have strong classification abilities on training and validation datasets. However, they are vulnerable to adversarial images, which are formally defined as imperceptibly small changes (in a given dissimilarity metric) to model input that lead to misclassification (Szegedy et al., 2014; Goodfellow et al., 2014). This behavior could mean several things: the model is overfitting on some level; the model is under-regularized; or this is simply due to complex nonlinearities in the model. This has lead to several lines of work in the deep learning community: the generation of adversarial images, defending against these adversarial attacks, and lastly determining *which* dissimilarity metric to consider.

Regarding the latter, it is not obvious what "imperceptibly small" means, and recent work has demonstrated adversarial image generation beyond ℓ_p norms by considering *deformations* instead of perturbations (Alaifari et al., 2018). There is also the problem of generating "realistic" attacks, such as through sparse attacks. For example these include small stickers on a road sign, which may tamper with autonomous vehicles (Eykholt et al., 2017). The purpose of this work is adversarial image generation for a broad class of (possibly non-differentiable) dissimilarity metrics for both undefended and defended networks. We do not make judgment regarding which metric is "best"; instead we are interested in an attack framework that works well for a broad class of metrics.

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, dubbed Iterative FGSM (IFGSM) (Goodfellow et al., 2014; Kurakin et al., 2016). 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 training loss function subject to a norm constraint on the perturbation, either with one step in the case of FGSM, or multiple steps in the case of IFGSM, and their ℓ_2 norm equivalents. Apart from training loss maximization,

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

attacks have been developed using loss functions that *directly* measure misclassification (Carlini and Wagner, 2016; Moosavi-Dezfooli et al., 2015). Others have considered the ℓ_1 and ℓ_0 norms; these both induce sparsity in the perturbations (Modas et al., 2018). In the black-box setting, adversarial examples are generated using only model decisions, which is a much more expensive endeavor. However, black-box methods often perform better, most notably by avoiding gradient obfuscation, since they take advantage of sampling properties near the decision boundary of the model. Notable examples of black-box (decision-based) attacks are the Boundary Attack (Brendel et al., 2017) and the recent HopSkipJumpAttack (Chen and Jordan, 2019).

The development of new and improved adversarial attacks has occurred in parallel with various defensive training regimes 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" (Tsipras et al., 2018). A popular method for training robust networks is *adversarial training*, where adversarial examples are added to the training data (see for example (Madry et al., 2017)).

Contributions

This paper introduces an attack methodology for not just ℓ_p norms, but any adversarial dissimilarity metric with a closed proximal form. This includes, but is not limited to, ℓ_1, ℓ_2, ℓ_∞, the ℓ_0 counting "norm", i.e. a true measurement of sparseness of the perturbation, and total variation, a non-ℓ_p dissimilarity. Our approach adopts the relaxation structure of the recently proposed LogBarrier attack (Finlay et al., 2019), 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. Finally, by using the total variation dissimiliarity, we shed light on a new class of imperceptible adversaries that incorporates neighboring pixel information, which can be viewed as an adversarial attack measured in a convolutional norm.

2 Background material

2.1 Adversarial attacks

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 should 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 optimization problem:

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

DNNs might be powerful classifiers, but that does not mean their decision boundaries are well-behaved. Instead, researchers have popularized using the training loss, often 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 Descent (PGD) on the cross-entropy loss:

$$\max_{u \in \mathcal{X}} \; \mathcal{L}(u) \quad \text{subject to} \quad m(u; x) \leq \varepsilon, \tag{2}$$

where $m(\cdot; x)$ is typically taken to be either the ℓ_2 or ℓ_∞ norm, and ε 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 and Wagner (2016) attack the logit-layer of a network and solve a different optimization problem, which depends on the choice of norm. Regarding adversarial defense methods, they demonstrated how a significant number of prior defense methods fail because of "gradient obfuscation", where gradients are small only locally to the image (Athalye et al., 2018). 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 (Modas et al., 2018; Papernot et al., 2015). This is of interest because an adversary might be required to also budget the number of allowed pixels to perturb, while still remaining "imperceptible" to the human eye. For example, the sticker-attack (Eykholt et al., 2017) is a practical attack with real-world consequences, and does not interfere with every single part of the image.

2.2 Proximal gradient method

Our adversarial attack amounts to a 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 (Bai et al., 2018; Zhao et al., 2019; Meinhardt et al., 2017; Paquette et al., 2018). For a full discussion on this topic, we suggest (Beck, 2017).

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

$$\min_{x\in\mathcal{E}} \Phi(x) := f(x) + g(x) \tag{3}$$

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

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

Generating a stationary point of (3) amounts to finding a fixed point of the following sequence:

$$x^{(k+1)} = \mathrm{Prox}_{\tau g}(x^{(k)} - \tau\nabla f(x^{(k)})), \tag{4}$$

where $\tau > 0$ is some step size, and $\mathrm{Prox}_{\lambda g}(\cdot)$ is defined as

$$\mathrm{Prox}_{\lambda g}(x) := \arg\min_{u\in\mathcal{E}} g(u) + \frac{1}{2\lambda}\|u - x\|_2^2.$$

Despite f not being convex, there are still convergence properties we can get from a sequence of iterates generated in this way. The following theorem is a simplified version of what can be found in (Beck, 2017) (Section 10.3 with proof), and is the main motivation for our proposed method.

Theorem 1 *Given the assumptions on (3), let $\{x^k\}_{k\geq 0}$ be the sequence generated by (4), with fixed step size $\tau \in (\frac{L}{2}, \infty)$. 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 (3);

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

(c) all limit points of the sequence $\{x^k\}_{k\geq 0}$ are stationary points of (3).

3 Our method: ProxLogBarrier

Following the previous theoretical ideas, we reformulate (1) in the following way:

$$\min_{u\in\mathcal{X}} m(u;x) \quad \text{s.t.} \quad z_{\max} - z_y > 0. \tag{5}$$

Here, $Z(\cdot)$ is the model output before the softmax layer that "projects" onto Δ_c, and so $z_{\max} := \max_{i\neq y}[Z(u)]_i$ and $z_y := [Z(u)]_y$. In other words, we want to perturb the clean image minimally in such a way that the model misclassifies it. This problem is difficult as the decision boundary has virtually no exploitable structure. Thus the problem can be relaxed using a logarithmic barrier, a technique often used in traditional optimization (Nocedal and Wright, 2006),

$$\min_{u\in\mathcal{X}} m(u;x) - \lambda\log(z_{\max} - z_y). \tag{6}$$

This objective function now includes the constraint that enforces misclassification. In (Finlay et al., 2019), (6) was originally solved via gradient descent, which necessarily assumes that $m(\cdot;x)$ is at least differentiable. The assumption of differentiability is not a given, and may be impracticable. For example, consider the subgradient of ℓ_∞ for an element in $\mathbb{R}^n$;

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

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. Indeed, in the original LogBarrier paper, a smooth approximation of this norm was used to get around this issue. We shall see that this does not occur with our proposed ProxLogBarrier method.

For brevity, let $\varphi(\cdot) := -\log(\cdot)$ and

$$F(u) := \left(\max_{i\neq y} Z(u)\right) - [Z(u)]_y.$$

The optimization problem (6) becomes

$$\min_{u\in\mathcal{X}} m(u;x) + \lambda\varphi(F(u)). \tag{7}$$

One can draw several similarities between (7) and (3). As before, we have no guarantees of convexity on $\varphi\circ F$, which is a representation of f in the composite problem, but it is smooth provided $F(\cdot) \in \mathrm{dom}(\varphi)$ (that is, F is smooth from a computational perspective). Our dissimilarity metric $m(u;x)$ represents g, as it usually has a closed-form proximal operator. Thus, we simply turn to the proximal gradient method to solve the minimization problem in (7).

We iteratively find a minimizer for the problem; 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. 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 (pixels must be in the range [0,1]). In traditional non-convex optimization, best practice is

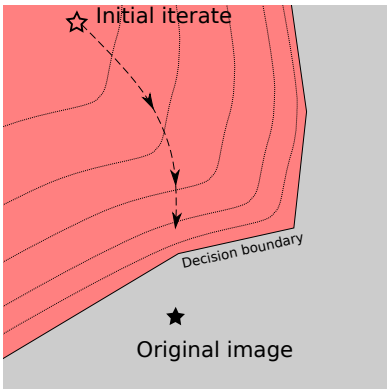


Figure 1: Illustration of the ProxLogBarrier attack: the attack is initialized with a misclassified image, which is then moved towards the original image.

to also record the "best iterate", as valleys are likely pervasive throughout the decision boundary. This way, even at some point our gradient sends our image far-off and is unable to return in the remaining iterations, we already have a better candidate.

The algorithm begins with a misclassified image, and moves the iterates towards the original image by minimizing the dissimilarity metric. Misclassification is maintained by the log barrier function, which prevents the iterates from crossing the decision boundary. Refer to Figure 1. Contrast this with PGD based algorithms, which begin at or near the original image, and iterate away from the original image.

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 $\mu, h, \lambda > 0, \beta \in (0, 1)$.
Initialize $u^{(0)}$ to be misclassified, $w^{(0)} := u^{(0)}$
for $k = 0, 1, 2, \ldots, K$ **do**
  Every K_{inner} iterations: $\lambda \leftarrow \lambda\beta$
  $y^{(k)} = \text{Prox}_{\mu m}\left(u^{(k)} - h\lambda\nabla\varphi(F(u^{(k)}))\right)$
  $u^{(k+1)} = \text{Project}(y^{(k)}; \mathcal{X})$
  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 ℓ_p dissimilarities

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 (Beck, 2017) 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 (Rockafellar and Wets, 2009), 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 (Duchi et al., 2008) 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 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.

Example of non-ℓ_p dissimilarity: Total variation

We let $\mathcal{X}$ denote the image space, and for the time being assume the images are grayscale, and let M denote the finite-difference operator on the grid-space defined by the image. Then $M : \mathcal{X} \to \mathcal{X} \times \mathcal{X}$, where

$$(Mv)_{i,j} = \begin{pmatrix} D_x v \\ D_y v \end{pmatrix}_{i,j} := \begin{pmatrix} v_{i+1,j} - v_{i,j} \\ v_{i,j+1} - v_{i,j} \end{pmatrix}_{i,j}, \quad (8)$$

where (i, j) are the pixel indices of the image in row-column notation. The *anisotropic total variation seminorm* is defined by

$$\|v\|_{\text{TV}} := \|Mv\|_{1,1} = \sum_{i,j} |(D_x v)_{i,j}| + |(D_y v)_{i,j}|, \quad (9)$$

where $\|\cdot\|_{1,1}$ is an induced matrix norm. Heuristically, this is a measure of large deviations between adjacent pixel values. In practice Mv can be implemented via a convolution. In the case of color images, we aggregate the total variation for each channel. Total variation (TV) is not a true norm, in that non-zero images v can have zero TV. In what follows, we omit the distinction and write TV-norm to mean the total variation seminorm. Traditionally, TV has been used in the context of image denoising (Rudin et al., 1992).

What does this mean in the context of adversarial perturbations? The TV-norm of the perturbation will be small when the perturbation has few jumps between pixels. That is, small TV-norm perturbations have *locally flat regions*. This is primarily because TV-norm is *convolutional* in nature: the finite-difference gradient operator incorporates neighboring pixel information. We note that this is not the first instance of TV being used as a disimillarity metric (Xiao et al., 2018); however our approach is quite different and is not derived from a flow. An outline for the proximal operator can be found in (Beck, 2017); we use a standard package for efficient computation (Barbero and Sra, 2011, 2018).

4 Experimental methodology

Outline

We compare the ProxLogBarrier attack with several other adversarial attacks on MNIST (LeCun et al., 1999), CIFAR10 (Krizhevsky and Hinton, 2009), and ImageNet-1k (Deng et al., 2009). For MNIST, we use the network described in (Papernot et al., 2015); on CIFAR10, we use a ResNeXt network (Xie et al., 2016); and for ImageNet-1k, ResNet50 (He et al., 2016; Coleman et al., 2018). 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 adversarial training for CIFAR10 and MNIST (Madry et al., 2017). On ImageNet-1k, we use the recently proposed scaleable input gradient regularization for adversarial robustness (Finlay and Oberman, 2019). We randomly select 1000 (test) images to evaluate performance on MNIST and 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 log barrier is with respect to the following constraint set

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

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

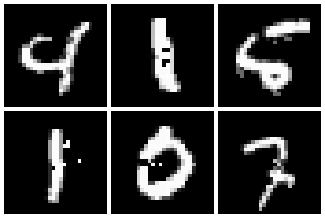
We compare the ProxLogBarrier attack with a wide range of attack algorithms that are available through the FoolBox adversarial attack library (Rauber et al., 2017). For perturbations in ℓ_0, we compare against SparseFool (Modas et al., 2018), Jacobian Saliency Map Attack (JSMA) (Papernot et al., 2015), and Pointwise (Schott et al., 2018) (this latter attack is black-box). For ℓ_2 attacks, we consider Carlini-Wagner's attack (CW) (Carlini and Wagner, 2016), Projected Gradient Descent (PGD) (Kurakin et al., 2016), DeepFool (Moosavi-Dezfooli et al., 2015), and the original LogBarrier attack (Finlay et al., 2019). Finally, for ℓ_∞ norm perturbations, we consider PGD, 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. We omit the One-Pixel attack (Su et al., 2017), as (Modas et al., 2018) showed that this attack is quite weak on MNIST, CIFAR10, and not tractable on ImageNet-1k.

Implementation details for our algorithm

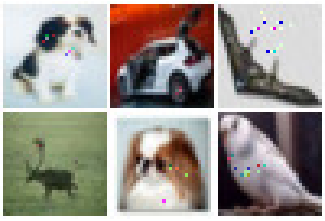
When optimizing for ℓ_2 or TV based noise, we initialize the adversarial image with sufficiently large Gaussian noise; for ℓ_∞ and ℓ_0 based perturbations, we use uniform noise. In the event that an initial perturbation cannot be generated, we start randomly at another image with different class. For hyper-parameters, we used $\lambda_0 = 0.1, \beta = 0.75, h = 0.1, \mu = 1$, with $K = 900, K_{\text{inner}} = 30$. 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 are that for 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. For TV-norm, perturbations, we set the proximal parameter $\mu = 5$, and $K = 200$ with $K_{\text{inner}} = 20$ (far less than before).

Reporting

For perturbations in ℓ_2 and ℓ_∞, we report the percent misclassification at various threshold levels that are somewhat standard (Tsipras et al., 2018). Our choices for ℓ_0 distance thresholds were arbitrary, however we supplement with a median perturbation distances on 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



(a) ℓ_0 attacks on MNIST



(b) ℓ_0 attacks on CIFAR10

Figure 2: Adversarial images for ℓ_0 perturbations, generated by our method.

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).

Perturbations in ℓ_0

Result for ℓ_0 perturbations are found in Table 1, with examples available in Figure 2 and Figure 4b. Across all datasets considered, ProxLogBarrier outperforms all other attack methods, for both defended and undefended networks. It also appears immune to Madry-style adversarial training on both 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 zero'd out. We remark that SparseFool was unable to adversarially attack all images, whereas ProxLogBarrier always succeeded.

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 PGD when adversarially defended. On ImageNet-1k, our method suffers dramatically. This is likely due to very poor decision boundaries with respect to this norm ℓ_∞, as our method will necessarily be better when the boundaries are not muddled. PGD does not focus on the decision boundaries explicitly, thus has more room to find something adversarial quickly.

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.

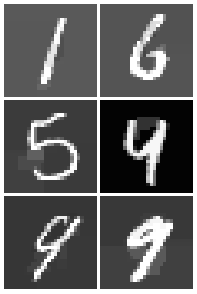
[1] We believe this is an implementation error on behalf of the repository. To accurately compare, we attacked an 18-layer ResNet for CIFAR10 that achieves slightly worse clean error as reported in (Modas et al., 2018). Our median percent pixels perturbed was 1.4%, and they reported 1.27%.

Table 1: Adversarial robustness statistics, measured in the ℓ_0 norm.

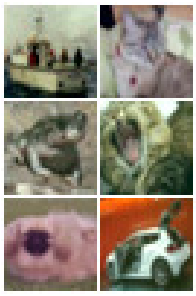
	MNIST			CIFAR10			ImageNet		
	% error at		median distance	% error at		median distance	% error at		median distance
	$\varepsilon = 10$	$\varepsilon = 30$		$\varepsilon = 30$	$\varepsilon = 80$		$\varepsilon = 500$	$\varepsilon = 1000$	
PLB	**86.30**	**100**	**6**	**44.10**	**68.50**	**39**	**66.00**	**80.20**	**268**
SparseFool	46.00	99.40	11	15.60	22.60	3071[1]	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	—	—	—

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
PGD	**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) PGD	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



(a) TV-norm attacks on MNIST



(b) TV-norm attacks on CIFAR10

Figure 3: Adversarial images for TV-norm perturbations, generated by our method.

Perturbations in the TV-norm

To our knowledge, there are no other TV-norm atacks against which to compare our methods. However, we present the median total variation across the data in question, and a handful of pictures for illustration. On MNIST, adversarial images with minimal total variation are often as expected: near-flat perturbations or very few pixels perturbed (see Figure 3a). For CIFAR10 and ImageNet-1k, we have found that adversarial images with small TV-norm have an adversarial "tint" on the image: they appear nearly identical to the original, with a small color shift. When the adversary is not a tint, perturbations are highly localized or localized in several regions. See for example Figures 3b and 4a.

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

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
PGD	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) PGD	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

Table 4: Statistics for perturbations in TV-norm

	median TV-norm	max TV-norm
MNIST	2.52	11.0
CIFAR10	1.36	11.0
ImageNet-1k	13.4	149.6

Table 5: 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

a batch of images. 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 5. Algorithms implemented from the FoolBox repository were not written to take advantage of the GPU, hence we omit run-time comparisons. Heuristically speaking, PGD is one of the faster algorithms, whereas CW, SparseFool, and DeepFool are slower. We omit the computational complexity for minimizing total variation since the proximal operator is coded in C, and not Python. Our official implementation can be found here.

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 (Duchi et al., 2008).

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 and the total variation seminorm. 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 in the median distance and in commonly reported thresholds. The proximal gradient method points towards new forms of adversarial attacks, such as those measured in the TV-norm, provided the attack's dissimilarity metric has a closed proximal form.

(a) TV-norm attacks

(b) ℓ_0 attacks, with fewer than 1000 pixels perturbed

Figure 4: Adversarial images for ImageNet-1k. Note that ℓ_0 attacks are only visible when the image is magnified. The TV-norm perturbations are visible as either a tint of the full image, or as a set of local tints.

References

Alaifari, R., Alberti, G. S., and Gauksson, T. (2018). Adef: an iterative algorithm to construct adversarial deformations. *CoRR*, abs/1804.07729.

Athalye, A., Carlini, N., and Wagner, D. A. (2018). 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.

Bai, Y., Wang, Y., and Liberty, E. (2018). Proxquant: Quantized neural networks via proximal operators. *CoRR*, abs/1810.00861.

Barbero, A. and Sra, S. (2011). Fast newton-type methods for total variation regularization. In Getoor, L. and Scheffer, T., editors, *ICML*, pages 313–320. Omnipress.

Barbero, A. and Sra, S. (2018). Modular proximal optimization for multidimensional total-variation regularization. *Journal of Machine Learning Research*, 19(56):1–82.

Beck, A. (2017). *First-order methods in optimization*.

Brendel, W., Rauber, J., and Bethge, M. (2017). Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*.

Carlini, N. and Wagner, D. A. (2016). Towards evaluating the robustness of neural networks. *CoRR*, abs/1608.04644.

Chen, J. and Jordan, M. I. (2019). Boundary attack++: Query-efficient decision-based adversarial attack. *CoRR*, abs/1904.02144.

Coleman, C., Kang, D., Narayanan, D., Nardi, L., Zhao, T., Zhang, J., Bailis, P., Olukotun, K., Ré, C., and Zaharia, M. (2018). Analysis of dawnbench, a time-to-accuracy machine learning performance benchmark. *CoRR*, abs/1806.01427.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Li, F. (2009). 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.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. (2008). 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. ACM.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. (2017). Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945*.

Finlay, C. and Oberman, A. M. (2019). Scaleable input gradient regularization for adversarial robustness. *arXiv preprint arXiv:1905.11468*.

Finlay, C., Pooladian, A., and Oberman, A. M. (2019). The logbarrier adversarial attack: making effective use of decision boundary information. *IEEE International Conference on Computer Vision (ICCV)*.

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

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham. Springer International Publishing.

Krizhevsky, A. and Hinton, G. (2009). *Learning multiple layers of features from tiny images*.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2016). Adversarial examples in the physical world. *CoRR*, abs/1607.02533.

LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, page 319.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Meinhardt, T., Möller, M., Hazirbas, C., and Cremers, D. (2017). Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. *CoRR*, abs/1704.03488.

Modas, A., Moosavi-Dezfooli, S., and Frossard, P. (2018). Sparsefool: a few pixels make a big difference. *CoRR*, abs/1811.02248.

Moosavi-Dezfooli, S., Fawzi, A., and Frossard, P. (2015). Deepfool: a simple and accurate method to fool deep neural networks. *CoRR*, abs/1511.04599.

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Papernot, N., McDaniel, P. D., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2015). The limitations of deep learning in adversarial settings. *CoRR*, abs/1511.07528.

Paquette, C., Lin, H., Drusvyatskiy, D., Mairal, J., and Harchaoui, Z. (2018). Catalyst for gradient-based nonconvex optimization. In Storkey, A. and Perez-Cruz, F., 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. PMLR.

Rauber, J., Brendel, W., and Bethge, M. (2017). Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models. *CoRR*, abs/1707.04131.

Rockafellar, R. T. and Wets, R. J.-B. (2009). *Variational analysis*, volume 317. Springer Science & Business Media.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268.

Schott, L., Rauber, J., Brendel, W., and Bethge, M. (2018). Robust perception through analysis by synthesis. *CoRR*, abs/1805.09190.

Su, J., Vargas, D. V., and Sakurai, K. (2017). One pixel attack for fooling deep neural networks. *CoRR*, abs/:1710.08864.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. (2014). Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.

Xiao, C., Zhu, J., Li, B., He, W., Liu, M., and Song, D. (2018). Spatially transformed adversarial examples. *CoRR*, abs/1801.02612.

Xie, S., Girshick, R. B., Dollár, P., Tu, Z., and He, K. (2016). Aggregated residual transformations for deep neural networks. *CoRR*, abs/1611.05431.

Zhao, P., Xu, K., Liu, S., Wang, Y., and Lin, X. (2019). 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. ACM.