arXiv:1510.05328v2 [cs.NE] 25 Oct 2015

EXPLORING THE SPACE OF ADVERSARIAL IMAGES

Pedro Tabacof, Eduardo Valle
School of Electrical and Computing Engineering (FEEC)
University of Campinas (Unicamp)
Campinas, SP, Brazil
{tabacof,dovalle}@dca.fee.unicamp.br

ABSTRACT

We formalize and show an algorithm for finding adversarial images given a probabilistic classifier (e.g. deep convolutional net). We probe the pixel space of adversarial images using random Gaussian noise with varying standard deviation, and a heavy-tailed noise given by a non-parametric model of the empirical distribution of the adversarial distortion pixels. We show that adversarial images appear in large regions in the pixel space.

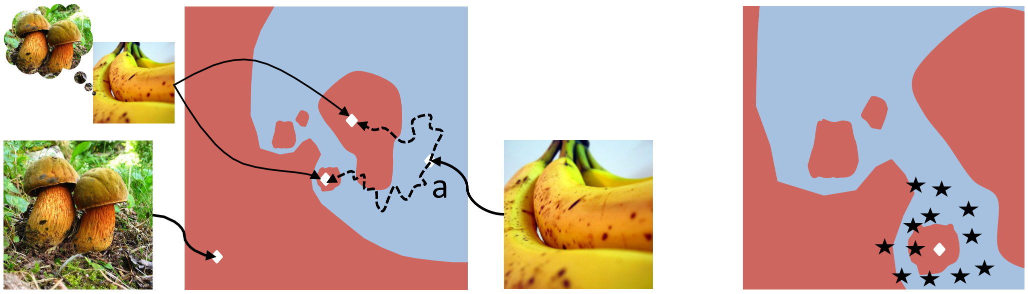


Figure 1: Fixed-sized images exist in a high-dimensional space spanned by their pixels (one pixel = one dimension), here depicted as a two dimensional colormap. **Left:** classifiers associate points of the input pixel space to output class labels, here ‘banana’ (blue) and ‘mushroom’ (red). From a correctly classified original image (a), an optimization procedure (dashed arrows) can find adversarial examples that are, for humans, essentially equal to the original, but that will fool the classifier. **Right:** we probe the pixel space by taking a departing image (white diamond), adding random noise to it (black stars), and asking the classifier for the label. In compact, stable regions we expect the classifier to be consistent, even if wrong. In isolated, unstable regions, as depicted, the classifier will be erratic.

1 INTRODUCTION

Small but purposeful pixel distortions can easily fool the best deep convolutional networks for image classification (Szegedy et al., 2013; Nguyen et al., 2014). The small distortions are hardly visible by humans, but still can mislead most neural networks. Those confounding distortions have divided the Machine Learning community, with some hailing their existence as a “deep flaw” of neural networks (Bi, 2014). On the other hand, recent theoretical analysis has shown that classes that are not easily distinguishable will make *all* classifiers susceptible to adversarial examples (Fawzi et al., 2015).

Despite the controversy, adversarial images surely suggest a lack of robustness, since they are (for humans) essentially equal to the correctly classified images. Immunizing a network against those perturbations increases its ability to generalize, a form of regularization (Goodfellow et al., 2014) whose statistical nature deserves further investigation.

In this paper, we extend previous works on adversarial images for deep neural networks (Szegedy et al., 2013), by exploring the pixel space of such images using random perturbations. That framework (Figure 1) allows us to ask interesting questions about adversarial images. Initial skepticism about the relevance of adversarial images suggested they existed as isolated points in the pixel space, reachable only by a guided procedure with complete access to the model. More recent works (Goodfellow et al., 2014; Gu & Rigazio, 2014) claim that they inhabit large and contiguous regions in the space. The correct answer has practical implications: if adversarial images are isolated or inhabit very thin pockets, they deserve much less worry than if they form large, compact regions. In this work we intend to shed light to the issue with an in-depth analysis of adversarial image space.

2 Creating adversarial images

Assume we have a pre-trained classifier $p = f(X)$ that, for each input $X \in \mathcal{I}$, corresponding to the pixels of a fixed-sized image, outputs a vector of probabilities $p = [p_1 \cdots p_i \cdots p_n]$ of the image belonging to the class label i. We can assign h to the label corresponding to the highest probability p_h. Assume further that $\mathcal{I} = [L - U]$, for grayscale images, or $\mathcal{I} = [L - U]^3$ for RGB images, where L and U are the lower and upper limits of the pixel scale. In most cases L is 0, and U is either 1 or 255.

Assume that c is the correct label and that we start with $h = c$, otherwise there is no point in fooling the classifier. We want to add the smallest distortion D to X, such that the highest probability will no longer be assigned to h. The distortions must keep the input inside its space, i.e., we must ensure that $X + D \in \mathcal{I}$. In other words, the input is box-constrained. Thus, we have the following optimization problem:

$$\begin{aligned} \underset{D}{\text{minimize}} \quad & \|D\| \\ \text{subject to} \quad & L \leq X + D \leq U \\ & p = f(X + D) \\ & \max(p_1 - p_c, ..., p_n - p_c) > 0 \end{aligned} \tag{1}$$

This formulation is more general than the one presented by Szegedy et al. (2013), for it ignores non-essential details, such as the choice of the adversarial label. It also showcases the problem non-convexity: since $\max(x) < 0$ is convex, the inequality is clearly concave (Boyd & Vandenberghe, 2004), making the problem non-trivial even if the model $p = f(X)$ were linear in X. Deep networks, of course, hinder the problem further by making the model highly non-convex due to its nonlinearities.

2.1 Algorithm

Training a classifier usually means minimizing the classification error by changing the model weights. To generate adversarial images, however, we hold the weights fixed, and find the minimal distortion that still fools the network.

We can simplify the optimization problem of eq. 1 by exchanging the $\max$ inequality for a term in the loss function that measures how adversarial the probability output is:

$$\begin{aligned} \underset{D}{\text{minimize}} \quad & \|D\| + C \cdot \mathrm{H}(p, p^A) \\ \text{subject to} \quad & L \leq X + D \leq U \\ & p = f(X + D) \end{aligned} \tag{2}$$

where we introduce the adversarial probability target $p^A = [\mathbb{1}_{i=a}]$, which assigns zero probability to all but a chosen adversarial class label a. This formulation is essentially the same of Szegedy et al. (2013), picking an explicit (but arbitrary) adversary label. We make the loss function explicit: the cross-entropy (H) between the probability assignments, while Szegedy et al. (2013) keep the choice open.

The constant C balances the importance of the two objectives. The lower the constant, the more we will emphasize minimizing the distortion norm; but too low and the adversarial search may fail. Thus, we have to find the lowest possible C that will still lead to an adversarial image.

We can solve the new formulation with any local search compatible with box-constraints. Since the optimization variables are the pixel distortions, the problem size is exactly the size of the network input, which in our case varies from $28 \times 28 = 784$ for MNIST (LeCun et al., 1998) to $221 \times 221 \times 3 = 146\,523$ for OverFeat (Sermanet et al., 2013)). That is small enough to allow second-order procedures, which converge faster and with better guarantees (Nocedal & Wright, 2006). We chose L-BFGS-B, a box-constrained version of the popular L-BFGS second-order optimizer (Zhu et al., 1997). We set the number of corrections in the limited-memory matrix to 15, and the maximum number of iterations to 150. We used Torch7 to model the networks and extract their gradient with respect to the inputs (Collobert et al., 2011).

Finally, we implemented a bisection search to determine the optimal value for C (Kaw et al., 2009). Bisection requires initial lower and upper bounds for C, such that the upper bound succeeds in finding an adversarial image, and the lower bound fails. It will then search the transition point from failure to success (the "zero" in a root-finding sense): that will be the best C. We can use $C = 0$ as lower bound, as it always leads to failure (the distortion will go to zero). To find an upper bound leading to success, we start from a very low value, and exponentially increase it until we succeed. During the search for the optimal C we use warm-starting in L-BFGS-B to speed up convergence: the previous optimal value found for D is used as initial value for the next attempt.

```
Data: Input image $X$; Trained classifier $f(X)$; adversarial label $a$
Result: Adversarial image distortion $D$
$L\text{-}BFGS\text{-}B(X, p^A, C)$ solves optimization 2
$\epsilon$ is a small positive value
// Finding initial $C$
$C \leftarrow \epsilon$
repeat
    $C \leftarrow 2 \times C$
    $D, p \leftarrow L\text{-}BFGS\text{-}B(X, p^A, C)$
until $\max(p_i)$ in $p$ is $p_a$
// Bisection search
$C_{low} \leftarrow 0, C_{high} \leftarrow C$
repeat
    $C_{half} \leftarrow (C_{high} + C_{low})/2$
    $D', p \leftarrow L\text{-}BFGS\text{-}B(X, p^A, C_{half})$
    if $\max(p_i)$ in $p$ is $p_a$ then
        $D \leftarrow D'$
        $C_{high} \leftarrow C_{half}$
    else
        $C_{low} \leftarrow C_{half}$
    end
until $(C_{high} - C_{low}) < \epsilon$
return $D$
```

Algorithm 1: Adversarial image generation algorithm

To achieve the general formalism of eq. 1 we would have to find the adversarial label leading to minimal distortion. However, in datasets like ImageNet (Deng et al., 2009), with hundreds of classes, this search would be too costly. Instead, in our experiments, we opt to consider the adversarial label as one of the sources of random variability.

3 Adversarial space exploration

In this section we explore the vector space spanned by the pixels of the images to investigate the "geometry" of adversarial images: are they isolated, or do they exist in dense, compact regions? Most researchers currently believe that images of a certain appearance (and even meaning) are contained into relatively low-dimensional manifolds inside the whole space (Bengio, 2009). However, those manifolds are probably exceedingly convoluted, discouraging direct geometric approaches to investigate the pixel space.

Here, thus, we approach the problem indirectly, by probing the space around the images with small random perturbations. In regions where the manifold is nice — round, compact, occupying most of the space — the classifier will be consistent (even if wrong). In the regions where the manifold is problematic — sparse, discontinuous, occupying small fluctuating subspaces — the classifier will be erratic.

3.1 Datasets and models

To allow comparison with the results of Szegedy et al. (2013), we employ the MNIST handwritten digits database (10 classes, 60k training and 10k testing images), and the 2012 ImageNet Large Visual Recognition Challenge Dataset (1000 classes, 1.2M+ training and 150k testing images).

For MNIST, Szegedy et al. (2013) tested convolutional networks and autoencoders, while we employ a logistic linear classifier. While logistic classifiers have limited accuracy (∼7.5% error), their training procedure is convex (Boyd & Vandenberghe, 2004)). They also allowed us to complement Szegedy et al.'s results by investigating adversarial images in a shallow classifier.

For ImageNet, we used the pre-trained OverFeat network (Sermanet et al., 2013), which achieved 4th place at the ImageNet competition in 2013, with 14.2% top-5 error in the classification task, and won the localization competition the same year. Szegedy et al. (2013) employed AlexNet (Krizhevsky et al., 2012), which achieved 1st place at the ImageNet competition in 2012, with 15.3% top-5 error.

Figure 2 illustrates the application of Algorithm 1 to both datasets. Original and adversarial images are virtually indistinguishable. The pixel differences (middle row) do not show any obvious form — although a faint "erasing-and-rewriting" effect can be observed for MNIST. Figure 2a also suggests that simple, shallow classifiers are more robust to adversarial images, since the distortions are larger and more visible (both in comparison to OverFeat's in Figure 2b, and to the ones that appears originally in the work of Szegedy et al. (2013)).

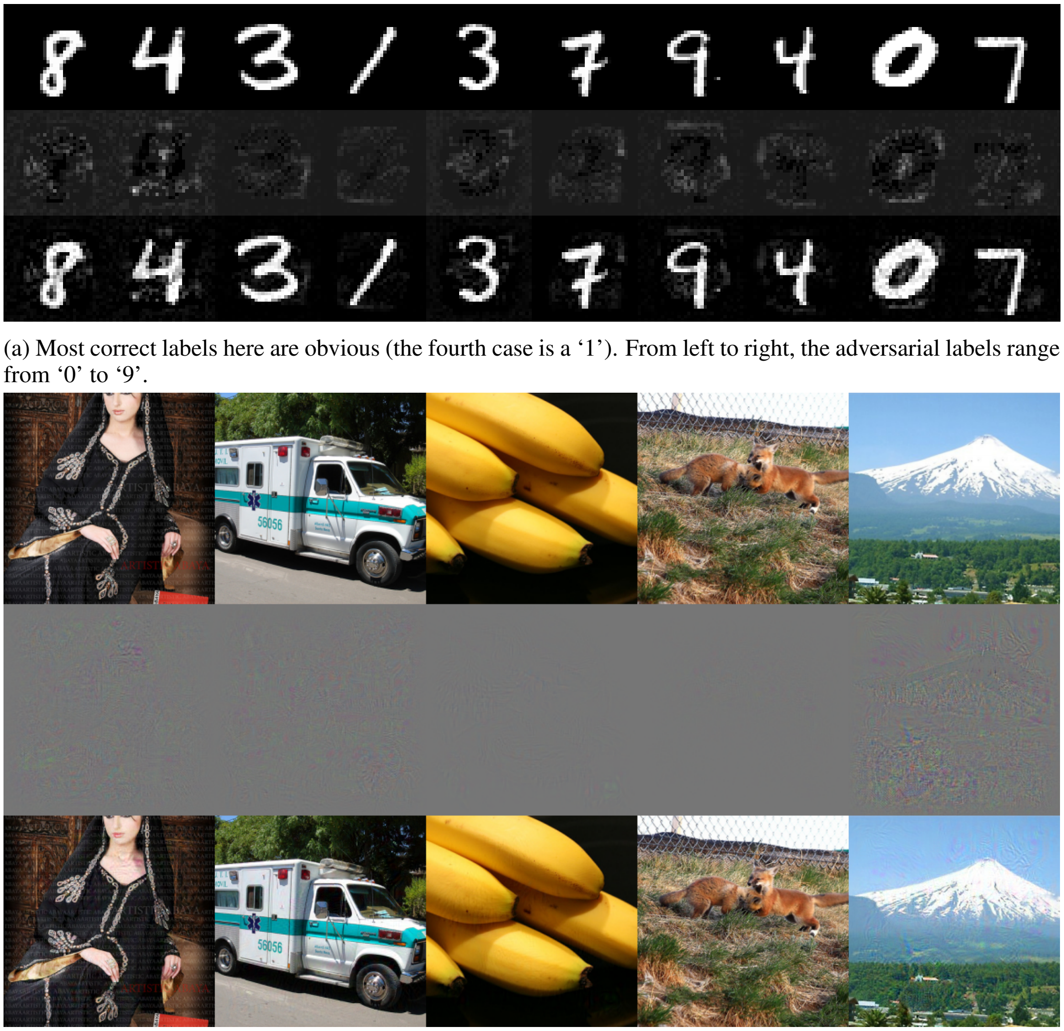
3.2 Methods

Each dataset (MNIST, ImageNet) was investigated independently, by applying Algorithm 1. For ImageNet we chose 5 classes (Abaya, Ambulance, Banana, Kit Fox, and Volcano), 5 correctly classified examples from each class, and 5 adversarial labels (schooner, bolete, hook, lemur, safe), totaling 125 adversarial images. For MNIST, we just picked 125 correctly classified examples from the 10K examples in the test set, and chose an adversarial label (from 9 possibilities) for each one. All choices that are not explicit were made at random, with uniform probability. To pick correctly classified examples at random, we rejected those originally misclassified until we accumulated the needed amount. We call, in the following sections, those correctly classified images *originals*, since the adversarial images are created from them.

The probing procedure consisted in picking an image pair (an adversarial image and its original), adding varying levels of noise to their pixels, resubmitting both to the classifier, and observing if the newly assigned labels corresponded to the original class, to the adversarial class, or to some other class.

We measured the *levels of noise* (λ) relative to the difference between each image pair (i.e., the distortion D found by Algorithm 1). We initially tested a Gaussian i.i.d. model for the noise. For each image $X = \{x_i\}$, our procedure creates an image $X' = \{\mathrm{clamp}(x_i + \epsilon)\}$ where $\epsilon \sim \mathcal{N}(\mu, \lambda\sigma^2)$, and μ and σ^2 are the sample mean and variance of the distortion pixels. In the experiments we ranged λ from 2^{-5} to 2^5. To keep the pixel values of X' within the original range $[L - U]$ we employ $\mathrm{clamp}(x) = \min(\max(x, L), U)$. In practice, we observed that clamping has little effect on the noise statistics.

An i.i.d. Gaussian model discards two important attributes of the distortions: the spatial correlations, and the higher-order momenta. We were interested to evaluate the relative importance of those two aspects, and thus performed an extra round of experiments that, while still discarding all spatial correlations by keeping the noise i.i.d., adds higher momenta information by modeling non-parametrically the distribution of distortion pixels. Indeed, a study of those higher momenta (Table 1) suggests that the adversarial distortions has a much heavier tail than the Gaussians, and we wanted to investigate how this affects the probing. The procedure is exactly the same as before,



(a) Most correct labels here are obvious (the fourth case is a ‘1’). From left to right, the adversarial labels range from ‘0’ to ‘9’.

(b) From left to right, correct labels: ‘Abaya’, ‘Ambulance’, ‘Banana’, ‘Kit Fox’, ‘Volcano’. Adversarial labels for all: ‘Bolete’ (a type of mushroom).

Figure 2: Adversarial examples for (a) MNIST with logistic regression; and (b) ImageNet with OverFeat. For both datasets: original images on the top row, adversarial images on the bottom row, distortions (difference between original and adversarial images) on the middle row.

but with $\epsilon \sim \mathcal{M}$, where $\mathcal{M}$ is an empirical distribution induced by a non-parametric observation of the distortion pixels. In those experiments we cannot control the level: the variance of the noise is essentially the same as the variance of the distortion pixels.

The main metric throughout the paper is the fraction of images (in %) that keep or switch labels when noise is added to a departing image, which we use as a measure of the stability of the classifier at the departing image in the pixel space. The fraction is computed over a sample of 100 probes, each probe being a repetition of the experiment with all factors held fixed but the sampling of the random noise.

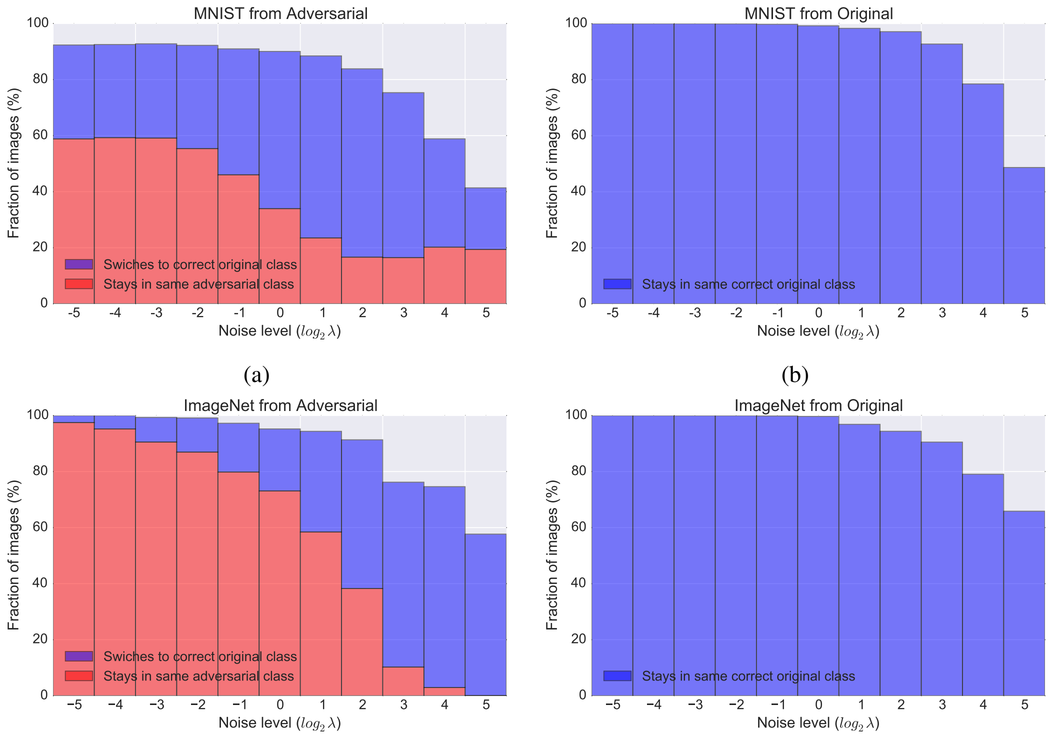
3.3 Results

Figure 3 shows that adversarial images do not appear isolated. On the contrary, to completely escape the adversarial pocket we need to add a noise with much higher variance — notice that the horizontal axis is logarithmic — than the distortion used to reach the adversarial image in the first place.

Table 1: Descriptive statistics of the adversarial distortions for the two datasets averaged over the 125 adversarial examples. Pixels values in $[0 - 255]$.

	Mean	Variance	Skewness	Ex. Kurtosis
MNIST	29.3 ± 1.5	269.6 ± 301.4	0.06 ± 0.96	7.8 ± 3.0
ImageNet	118.38 ± 0.07	7.6 ± 17.2	0.00 ± 0.09	6.5 ± 4.1

In both networks, the original images display a remarkable robustness against Gaussian noise (Figures 3b and 3d), confirming that robustness to random noise does not imply robustness to adversarial examples (Fawzi et al., 2015). This shows that while the adversarial pockets are not exactly isolated, neither are they as compact and well-behaved as the zones where the correctly classified samples exist.



(a)

(b)

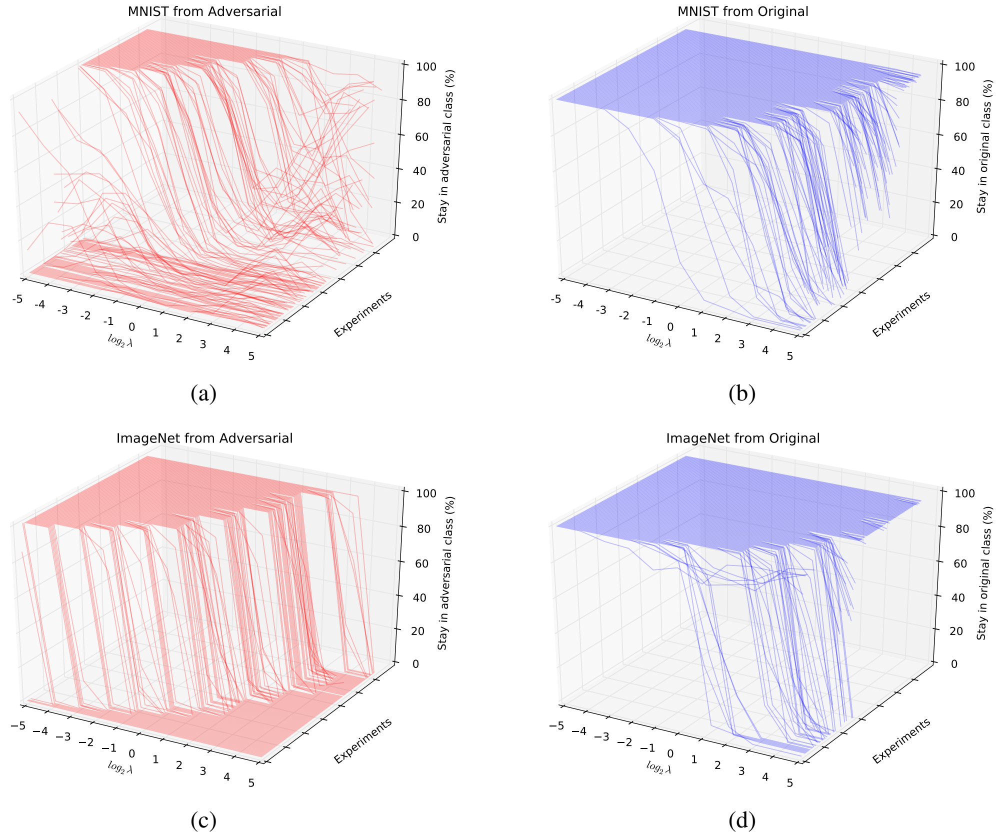
(c)

(d)

Figure 3: Adding Gaussian noise to the images. We perform the probing procedure explained in Section 3.2 to measure the stability of the classifier boundaries at different points of the pixel space. To escape the adversarial pockets completely we have to add a noise considerably stronger than the original distortion used to reach them in the first place: adversarial regions are not isolated. This is especially true for ImageNet / OverFeat. Still, the region around the correctly classified original image is much more stable. This graph is heavily averaged: each stacked column along the horizontal axis summarizes 125 experiments × 100 random probes.

The results in Figure 3 are strongly averaged, each data point summarizing, for a given level of noise, the result of 125 experiments: the fraction of images that fall in each label for *all* five original class labels, *all* five original samples from each label, and *all* five adversarial class labels. In reality there is a lot of variability that can be better appreciated in Figure 4. Here each curve alongside the axis *experiments* represents a *single* choice of original class label, original sample, and adversarial class label, thus there are 125 curves. (The order of the curves along this axis is arbitrary and chosen to minimize occlusions and make the visualization easier). The graphs show that depending on a specific configuration, the label may be very stable and hard to switch (curves that fall later or do not fall at all), or very unstable (curves that fall early). Those 3D graphs also reinforce the point about the stability of the correctly classified original images.

Those results reinforce our previous observation that the relatively shallow MNIST / logistic classifier seems more resilient against adversarial images than ImageNet / OverFeat. On the former, not only are the adversarial distortions more obvious (Figure 2a); but they are also more fragile: a small push is enough to throw many adversarial examples back to the correct space (contrast Figure 4a to Figure 4c).



(a)

(b)

(c)

(d)

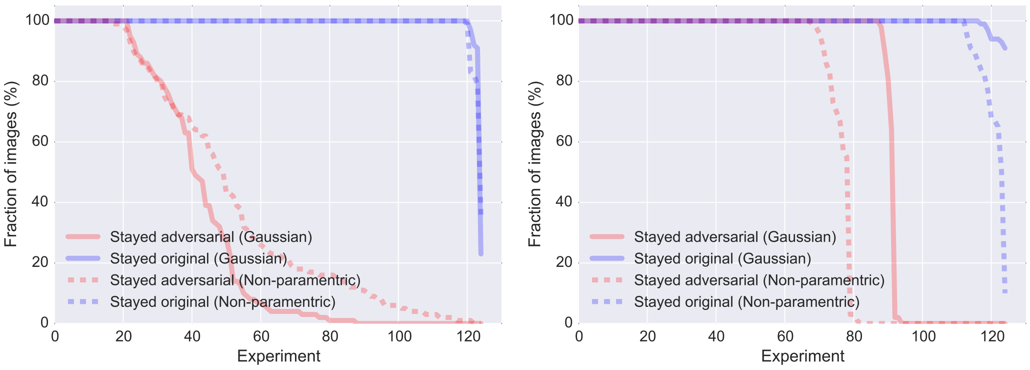
Figure 4: Adding Gaussian noise to the images. Another view of the probing procedure explained in Section 3.2. Contrarily to the averaged view of Figure 3, here each one of the 125 experiments appears as an independent curve along the *Experiments* axis (their order is arbitrary, chosen to reduce occlusions). Each point of the curve is the fraction of probes (out of a hundred performed) that keeps their class label.

Finally, we wanted to investigate how the nature of the noise added affected the experiments. Recall that our i.i.d. Gaussian noise differs from the original optimized distortion in two important aspects: no spatial correlations, and no important higher-order momenta. To explore the influence of those two aspects, we introduced a noise modeled after the empirical distribution of the distortion pixels. This still ignores spatial correlations, but captures higher-order momenta. The statistics of the distortion pixels are summarized in Table 1, and reveal a distribution that is considerably heavier-tailed than the Gaussians we have employed so far.

Figure 5 contrasts the effect of this noise modeled non-parametrically after the distortions with the effect of the comparable Gaussian noise ($\lambda = 1$). Each point in the curves is one of the 125 experiments, and represents the fraction of the 100 probe images that stays in the same class as the departing — adversarial or original — image. The experiments where ordered by this value in each curve (thus the order of the experiments in the curves is not necessarily the same). Here what is important are not the individual experiments, but the shape of the curve: how early and how quickly it falls.

For ImageNet, the curves for the non-parametric noise (dotted lines) fall before the curves for the Gaussian noise (continuous line), showing that, indeed, the heavier tailed noise affects the images more, even without the spatial correlation. On the other hand, for MNIST this effect seems to be the opposite, although the curves are more muddled together. Again the shallow classifier behaves differently: the effect of the heavy-tailed noise is less distinct.

In addition, all curves fall rather sharply. This shows that in almost all experiments, either all probes stay in the same label as the original, either all of them switch. Few experiments present intermediate results. This rather bimodal behavior was already present in the curves of Figure 4.



(a) MNIST / logistic regression

(b) ImageNet / OverFeat

Figure 5: For each of the 125 experiments we measure the fraction of the probe images (i.e., departing image + random noise) that stayed in the same class label. Those fractions are then sorted from biggest to lowest along the *Experiments* axis. The area under the curves indicates the entire fraction of probes among all experiments that stayed in the same class.

4 Discussion

Adversarial images are not necessarily isolated, spurious points: many of them inhabit relatively dense regions of the pixel space. Our in-depth analysis reinforce previous claims found in the literature (Goodfellow et al., 2014; Gu & Rigazio, 2014), explaining why adversarial images generalize between different training sets and architectures (Szegedy et al., 2013): if the adversarial manifold occupies a significant amount of space, different separation boundaries may be susceptible to the same confounding examples.

The adversarial image resilience to noise is extremely variable. Still, in general, you need more variance in the noise to leave an adversarial pocket, than the variance in the optimized distortion required to get there.

The nature of the noise affects the resilience of both adversarial and original images. In ImageNet, Gaussian noise affects the images less than a heavy-tailed noise modeled after the empirical distribution of the distortions used to reach the adversarial images in the first place. An important next step in the exploration, in our view, is to understand the spatial nature of the adversarial distortions, i.e., the role spatial correlations play.

Curiously, a weak, shallow classifier (logistic regression), in a simple task (MNIST), seems less susceptible to adversarial images than a strong, deep classifier (OverFeat), in a complex task (ImageNet). The adversarial distortion for MNIST/logistic regression is more evident and humanly discernible. It is also more fragile, with more adversarial images reverting to the original at relatively lower noise levels. Is susceptibility to adversarial images an inevitable Achilles' heel of powerful complex classifiers? Speculative analogies with the illusions of the Human Visual System are tempting, but the most honest answer is that we still know too little. Our hope is that this article will keep the conversation about adversarial images ongoing and help further explore those intriguing properties.

The source code for adversarial image generation and pixel space analysis is at `https://github.com/tabacof/adversarial`.

ACKNOWLEDGMENTS

We would like to thank Micael Carvalho for his helpful hints and revision; Jonghoon Jin for open access to his OverFeat wrapper[1]; and Soumith Chintala, and the rest of the Torch7 team for all of their help on the mailing list.

REFERENCES

Bengio, Yoshua. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

Bi, Ran. Does deep learning have deep flaws? `http://www.kdnuggets.com/2014/06/deep-learning-deep-flaws.html`, 2014. Accessed: 2015-09-08.

Boyd, Stephen and Vandenberghe, Lieven. *Convex optimization*. Cambridge university press, 2004.

Collobert, Ronan, Kavukcuoglu, Koray, and Farabet, Clément. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009.

Fawzi, Alhussein, Fawzi, Omar, and Frossard, Pascal. Analysis of classifiers' robustness to adversarial perturbations. *arXiv preprint arXiv:1502.02590*, 2015.

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

Gu, Shixiang and Rigazio, Luca. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*, 2014.

Kaw, Autar K, Kalu, Egwu K, and Nguyen, Duc. Numerical methods with applications. 2009.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Yann, Cortes, Corinna, and Burges, Christopher JC. The mnist database of handwritten digits, 1998.

Nguyen, Anh, Yosinski, Jason, and Clune, Jeff. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *arXiv preprint arXiv:1412.1897*, 2014.

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

Sermanet, Pierre, Eigen, David, Zhang, Xiang, Mathieu, Michaël, Fergus, Rob, and LeCun, Yann. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

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

Zhu, Ciyou, Byrd, Richard H, Lu, Peihuang, and Nocedal, Jorge. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[1] `https://github.com/jhjin/overfeat-torch`