Exploring Adversarial Images in Deep Neural Networks

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

***Abstract*—Adversarial examples have raised questions regarding the robustness and security of deep neural networks. In this work we formalize the problem of adversarial images given a pre-trained classifier, showing that even in the linear case the resulting optimization problem is nonconvex. We generate adversarial images using deep classifiers on the ImageNet dataset. We probe the pixel space of adversarial images using noise of varying intensity and distribution. We bring novel visualizations that showcase the phenomenon and its high variability. We show that adversarial images appear in large regions in the pixel space, and that it is hard to leave those regions by adding noise to the images, even with high intensity.**

I. Introduction

After the huge empirical success of deep neural networks, Szegedy *et al.* surprised the community, showing that small but purposeful pixel distortions can easily fool the best convolutional networks for image classification [1]. Szegedy *et al.* used the gradient of the network output with respect to its input to find the minimal pixel distortion that leads to misclassification — small distortions which are hardly visible to humans. We present some examples in Figure 2 for two different datasets (MNIST and ImageNet) and three different network architectures.

Adversarial images even generalize across different network architectures [2]. Nguyen *et al.* showed how adversarial images can be generated using evolutionary approaches [3]. Goodfellow *et al.* demonstrated that only one gradient evaluation is necessary to arrive at an adversarial image [4]. Papernot *et al.* showed that even a black-boxes can be adversarially attacked, given an oracle to provide labels for input images by training a local substitute model [5]. That black-box attack can be used in a more general manner to steal machine learning models that are available as prediction APIs [6]. Sara Sabour *et al.* [7] show that adversarial attacks can not only lead to mislabeling, but also manipulate the internal representations of the network. Adversarial examples can also attack neural network policies in the context of reinforcement learning [8].

The problem of adversarial images has divided the Machine Learning community, with some hailing it as a "deep flaw" of deep neural networks [9]; and others promoting a more cautious interpretation, and showing, for example, that most classifiers are susceptible to adversarial examples [4], [10].

Despite the controversy, adversarial images surely suggest a lack of robustness, since they are (for humans) essentially equal to correctly classified images. Immunizing a network against those perturbations increases its ability to generalize, a form of regularization [4] whose statistical nature deserves further investigation. Even the traditional backpropagation training procedure can be improved with adversarial gradient [11]. The idea behind adversarial regularization is to make the input space smooth, therefore small changes in the input will not lead to large changes on the output, which is one explanation of the existence of adversarial images. This idea is formalized in the Virtual Adversarial Training, where a regularization term of input space smoothness is added to the training procedure, allowing even unsupervised networks to benefit from this concept [12].

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 [4], [13] 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. We propose a framework (Figure 1) that allows us to ask interesting questions about adversarial images.

M.Sc. dissertation.

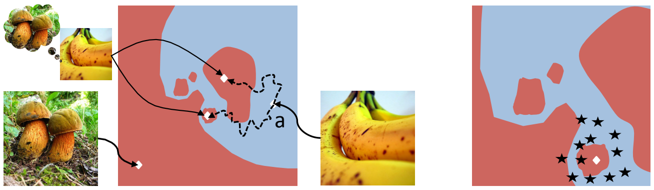

Fig. 1. Fixed-sized images occupy a high-dimensional space spanned by their pixels (one pixel = one dimension), here depicted as a 2D colormap. **Left:** classifiers associate points of the input pixel space to output class labels, here 'banana' (blue) and 'mushroom' (red). From a correctly classified 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, the classifier will be consistent (even if wrong). In isolated, unstable regions, as depicted, the classifier will be erratic.

II. Creating Adversarial Images

Assume we have a pre-trained classifier $\boldsymbol{p} = f(\boldsymbol{x})$ that, for each input $\boldsymbol{x} \in \mathcal{I}$, corresponding to the pixels of a fixed-sized image, outputs a vector of probabilities $\boldsymbol{p} = [p_1 \cdots p_i \cdots p_n]$ of the image belonging to the class label i. We will be rather lax in what we accept as output: most uncertainties behaving like probabilities will do (i.e., ranging from 0 to 1, additive, normalized, etc.). 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 $\boldsymbol{d}$ to $\boldsymbol{x}$, 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 $\boldsymbol{x} + \boldsymbol{d} \in \mathcal{I}$. In other words, the input is box-constrained. Thus, we have the following optimization:

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

That formulation is more general than the one presented by [1], for it ignores non-essential details, such as the choice of the adversarial label. It also showcases the non-convexity: since $\max(x) < 0$ is convex, the inequality is clearly concave [14], making the problem non-trivial even if the model $\boldsymbol{p} = f(\boldsymbol{x})$ were linear in $\boldsymbol{x}$. Deep networks, of course, exacerbate the non-convexity due to their highly non-linear model. For example, a simple multi-layer perceptron for binary classification could have $f(\boldsymbol{x}) = \text{logit}^{-1}(W_2 \cdot \tanh(W_1 \cdot \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2)$, which is neither convex nor concave due to the hyperbolic tangent.

A. Procedure

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. Because any two images can be directly switched with a large-enough distortion, the problem is only interesting for small distortions, preferably those imperceptible to humans.

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

$$\begin{aligned} \underset{\boldsymbol{d}}{\text{minimize}} \quad & \|\boldsymbol{d}\|_2^2 + C \cdot \mathrm{H}(\boldsymbol{p}, \boldsymbol{p}^A) \\ \text{subject to} \quad & L \leq \boldsymbol{x} + \boldsymbol{d} \leq U \\ & \boldsymbol{p} = f(\boldsymbol{x} + \boldsymbol{d}) \end{aligned} \tag{2}$$

where we introduce the adversarial probability target $\boldsymbol{p}^A = [\mathbb{1}_{i=a}]$, which assigns zero probability to all but a chosen adversarial label a. We use the square of the ℓ_2-distance as the penalty term to the adversarial distortion. We experimented with the ℓ_1-distance penalty, but we did not find any improvements in the adversarial attack. The formulation in equation 2 is essentially the same of 1, picking an explicit (but arbitrary) adversary label. We stipulate the loss function: the cross-entropy (H) between the probability assignments; while 1 keep that choice open.

The constant C balances the importance of the two objectives. The lower the constant, the more we will minimize the distortion norm. Values too low, however, may turn the optimization unfeasible. We want the lowest, but still feasible, value for C.

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, in our case $221 \times 221 \times 3 = 146\,523$ for the OverFeat network [15]. Such input sizes make numeric differentiation (e.g. finite differences) impractical to compute the huge number of gradients required to find an adversarial image: a single gradient of the output with respect to the input of a 256×256 pixels×3 channels image requires almost 200 thousand feedforward evaluations. We must, thus, resort to backpropagation, just as if we were training the network.

In contrast to current neural network training, that reaches hundreds of millions of weights, those sizes are small enough to allow second-order procedures, which converge faster and with better guarantees [16]. We chose L-BFGS-B, a box-constrained version of the popular L-BFGS second-order optimizer [17]. 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 [18].

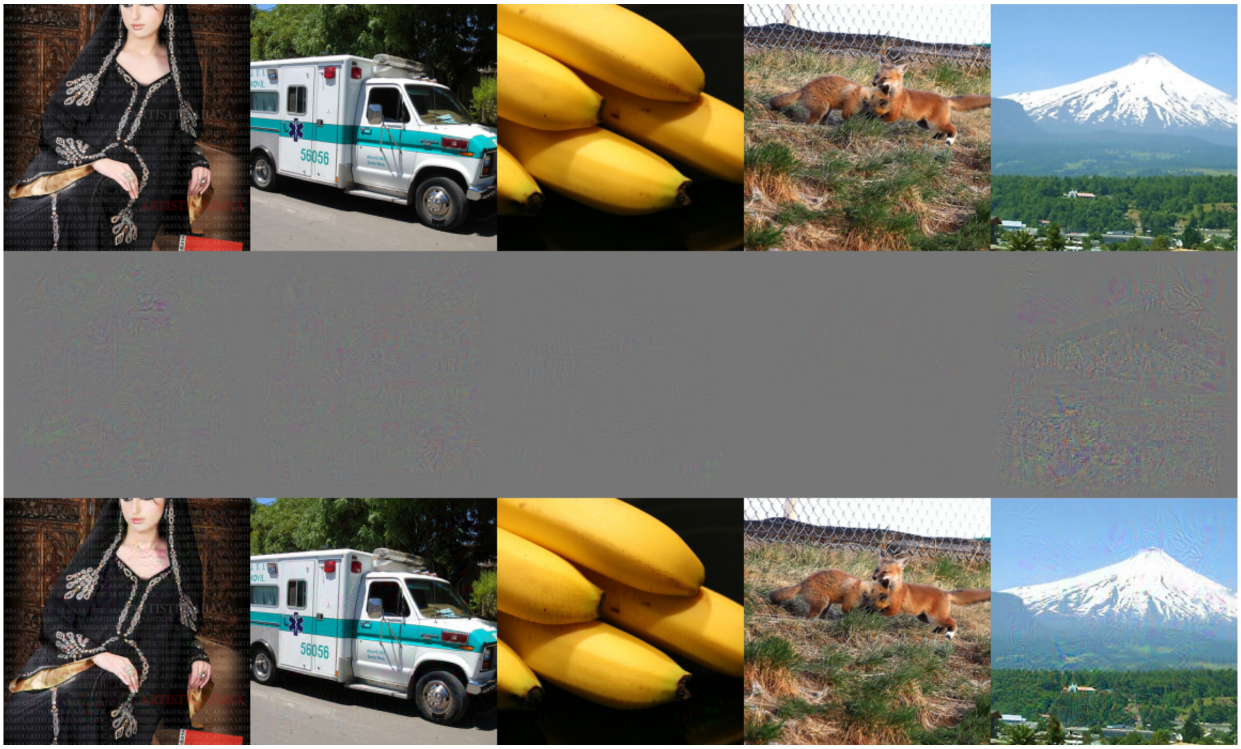

Fig. 2. Original images on the top row, adversarial images on the bottom row, distortions (difference between original and adversarial images) on the middle row. OverFeat on ImageNet. From left to right, correct labels: ‘Abaya’, ‘Ambulance’, ‘Banana’, ‘Kit Fox’, ‘Volcano’. Adversarial labels for all: ‘Bolete’ (a type of mushroom).

Finally, we implemented a bisection search to determine the optimal value for C [19]. The algorithm is explained in detail in the next section.

B. Algorithm

Algorithm 1 implements the optimization procedure used to find the adversarial images. The algorithm is essentially a bisection search for the constant C, where in each step we solve a problem equivalent to equation 2. 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 $\boldsymbol{d}$ is used as initial value for the next attempt.

To achieve the general formalism of equation 1 we would have to find the adversarial label leading to minimal distortion. However, in datasets like ImageNet [20], with hundreds of classes, that search would be too costly. Instead, in our experiments, we opt to consider the adversarial label as one of the sources of random variability. As we will show, that does not upset the analyses.

Algorithm 1 Adversarial image generation algorithm

```
Require: A small positive value $\epsilon$
Ensure: $L\text{-}BFGS\text{-}B(\boldsymbol{x}, \boldsymbol{p}^A, C)$ solves optimization 2
{Finding initial $C$}
$C \leftarrow \epsilon$
repeat
    $C \leftarrow 2 \times C$
    $\boldsymbol{d}, \boldsymbol{p} \leftarrow L\text{-}BFGS\text{-}B(\boldsymbol{x}, \boldsymbol{p}^A, C)$
until $\max(p_i)$ in $\boldsymbol{p}$ is $p_a$
{Bisection search}
$C_{low} \leftarrow 0$, $C_{high} \leftarrow C$
repeat
    $C_{half} \leftarrow (C_{high} + C_{low})/2$
    $\boldsymbol{d}', \boldsymbol{p} \leftarrow L\text{-}BFGS\text{-}B(\boldsymbol{x}, \boldsymbol{p}^A, C_{half})$
    if $\max(p_i)$ in $\boldsymbol{p}$ is $p_a$ then
        $\boldsymbol{d} \leftarrow \boldsymbol{d}'$
        $C_{high} \leftarrow C_{half}$
    else
        $C_{low} \leftarrow C_{half}$
    end if
until $(C_{high} - C_{low}) < \epsilon$
return $\boldsymbol{d}$
```

III. 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 [21]. However, those manifolds are exceedingly convoluted, discouraging direct geometric approaches to investigate the pixel space.

Thus, our approach is indirect, 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.

A. Datasets and Models

We used the pre-trained OverFeat network [15], 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. [1] employed AlexNet [22], which achieved 1st place at the ImageNet competition in 2012, with 15.3% top-5 error.

We preprocess the inputs by standardizing each pixel with the global mean and standard deviation of all pixels in the training set images. We used Torch7 [18] for the implementation[1].

Figure 2 illustrates examples generated by the procedure above. Original and adversarial images are virtually indistinguishable. The pixel differences (middle row) do not show any obvious form.

B. Methods

We sampled 5 classes (Abaya, Ambulance, Banana, Kit Fox, and Volcano), 5 correctly classified examples from each class, and sampled 5 adversarial labels (Schooner, Bolete, Hook, Lemur, Safe), totaling 125 adversarial images. All random sampling was made with uniform probability. To sample only correctly classified examples, we rejected the misclassified ones 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. We initially tested a Gaussian i.i.d. model for the noise. For each image $\boldsymbol{x} = \{x_i\}$, our procedure creates an image $\boldsymbol{x}' = \{\text{clamp}(x_i + \epsilon)\}$ 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 $\boldsymbol{x}'$ within the original range $[L - U]$ we employ $\text{clamp}(x) = \min(\max(x, L), U)$. In practice, we observed that clamping has little effect on the noise statistics.

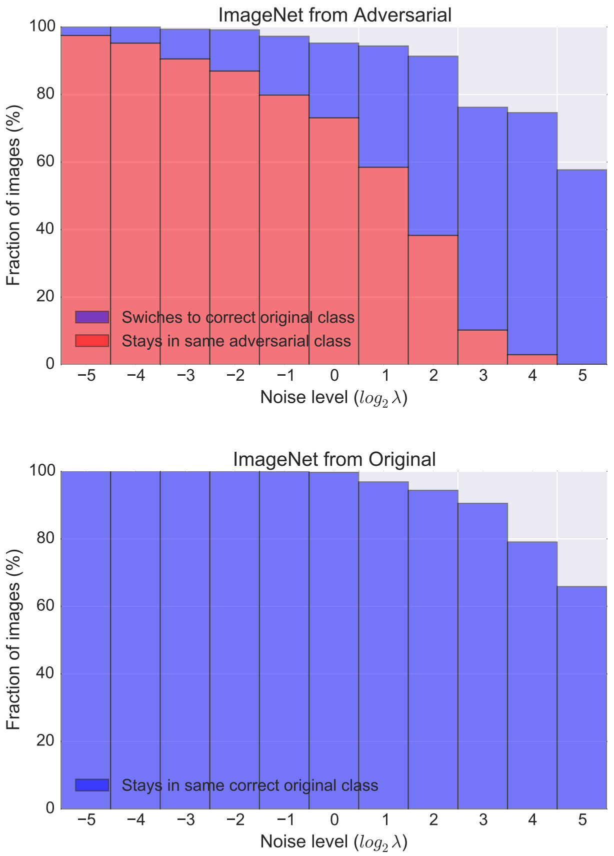

Fig. 3. Adding Gaussian noise to the images. We perform the probing procedure explained in Section III-B to measure the stability of the classifier boundaries at different points of the pixel space. To escape the adversarial pockets completely we have to add a noise considerably stronger than the original distortion used to reach them in the first place: adversarial regions are not isolated. That 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.

C. 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.

The original images display a remarkable robustness against Gaussian noise (Figure 3(b)), confirming that robustness to random noise does not imply robustness to

[1] The source code for adversarial image generation and pixel space analysis can be found at https://github.com/tabacof/adversarial.

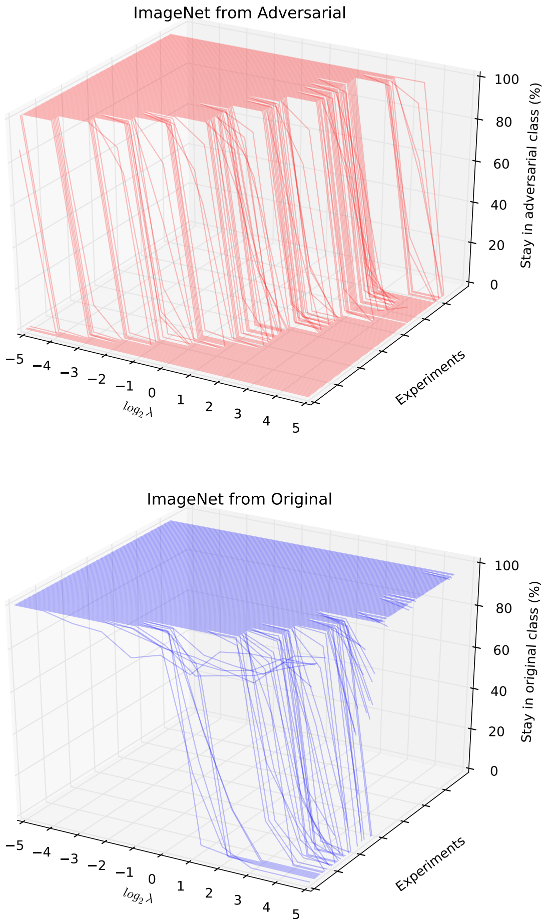

Fig. 4. Adding Gaussian noise to the images. Another view of the probing procedure explained in Section III-B. Contrarily to the averaged view of Figure 3, here each one of the 125 experiments appears as an independent curve along the *Experiments* axis (their order is arbitrary, chosen to reduce occlusions).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

adversarial examples [10]. That shows that while the adversarial pockets are not exactly isolated, neither are they as well-behaved as the zones that contain the correctly classified samples.

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 that 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.

IV. Conclusion

Our analysis reinforces previous claims found in the literature [4], [13]: adversarial images are not necessarily isolated, spurious points: many of them inhabit relatively dense regions of the pixel space. That helps to explain why adversarial images tend to stay adversarial across classifiers of different architectures, or trained on different sets [1]: slightly different classification boundaries stay confounded by the dense adversarial regions.

Are 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.

V. Publications

During his Master studies, the author has participated in four scholarly publications:

- First author of 23, published at the IJCNN 2016. The author received a Travel Grant from IEEE-CIS to attend the conference. The article was the subject of two media pieces 24], [25.
- First author of 26, presented at the Workshop on Adversarial Training at the NIPS 2016 conference. The paper was selected as one of the spotlight presentations.
- Second author (with equal contribution to the first) of 27, presented at the Workshop on Bayesian Deep Learning at NIPS 2016.
- Second author of 28, published at the journal PLOS ONE.

The first two papers listed above are mostly incorporated to the dissertation, while the last two are not. As of June 2017, those papers have been cited 23 times according to Google Scholar.

Acknowledgment

We thank Brazilian agencies CAPES, CNPq and FAPESP for financial support. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research. Eduardo Valle is partially supported by a Google Awards LatAm 2016 grant, and by a CNPq PQ-2 grant (311486/2014-2).

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014, arXiv:1312.6199. [Online]. Available: http://arxiv.org/abs/1312.6199

[2] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017. [Online]. Available: http://arxiv.org/abs/1704.03453

[3] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 427–436.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015, arXiv:1412.6572.

[5] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, 2017, pp. 506–519.

[6] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *USENIX Security*, 2016.

[7] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial manipulation of deep representations," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016, arXiv:1511.05122. [Online]. Available: http://arxiv.org/abs/1511.05122

[8] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," in *International Conference on Learning Representations (ICLR) Workshop*, 2017, arXiv:1702.02284. [Online]. Available: http://arxiv.org/abs/1702.02284

[9] R. Bi, "Does deep learning have deep flaws?" 2014, accessed: 2015-09-08. [Online]. Available: http://www.kdnuggets.com/2014/06/deep-learning-deep-flaws.html

[10] A. Fawzi, S. Moosavi-Dezfooli, and P. Frossard, "Robustness of classifiers: from adversarial to random noise," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 1624–1632.

[11] A. Nøkland, "Improving back-propagation by adding an adversarial gradient," *arXiv preprint arXiv:1510.04189*, 2015. [Online]. Available: http://arxiv.org/abs/1510.04189

[12] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii, "Distributional smoothing with virtual adversarial training," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016, arXiv:1507.00677. [Online]. Available: http://arxiv.org/abs/1507.00677

[13] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," in *International Conference on Learning Representations (ICLR) Workshop*, 2015, arXiv:1412.5068. [Online]. Available: http://arxiv.org/abs/1412.5068

[14] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[15] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013. [Online]. Available: http://arxiv.org/abs/1312.6229

[16] J. Nocedal and S. Wright, *Numerical optimization.* Springer Science & Business Media, 2006.

[17] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[18] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[19] R. L. Burden and J. D. Faires, "2.1 the bisection algorithm," *Numerical Analysis. Prindle, Weber & Schmidt, Boston, MA., pp. x*, vol. 676, 1985.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 248–255.

[21] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[23] P. Tabacof and E. Valle, "Exploring the space of adversarial images," in *2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016*, 2016, pp. 426–433.

[24] S. Winiger, "Adversarial machines: Fooling a.is (and turn everyone into a manga)," 2015, accessed: 2016-05-31. [Online]. Available: https://medium.com/@samim/adversarial-machines-998d8362e996

[25] M. James, "The flaw in every neural network just got a little worse," 2015, accessed: 2016-05-31. [Online]. Available: http://www.i-programmer.info/news/105-artificial-intelligence/9090-the-flaw-in-every-neural-network-just-got-a-little-worse.html

[26] P. Tabacof, J. Tavares, and E. Valle, "Adversarial images for variational autoencoders," in *NIPS 2016 Workshop on Adversarial Training*, 2016, arXiv:1612.00155. [Online]. Available: http://arxiv.org/abs/1612.00155

[27] R. Oliveira, P. Tabacof, and E. Valle, "Known unknowns: Uncertainty quality in bayesian neural networks," in *Bayesian Deep Learning Workshop NIPS 2016*, 2016, arXiv:1612.01251. [Online]. Available: http://arxiv.org/abs/1612.01251

[28] A. Godoy, P. Tabacof, and F. J. Von Zuben, "The role of the interaction network in the emergence of diversity of behavior," *PloS one*, vol. 12, no. 2, p. e0172073, 2017.