Towards Inducing Abstention in Certified Defense (Maybe)

Christian T. Covington
*AM221 - Advanced Optimization
Harvard University
christian.t.covington@gmail.com

Abstract

Recent work on classifiers that are certifiably robust to adversarial examples relies on allowing the classifier to abstain from prediction when it is not sufficiently confident. I develop a black-box attack strategy intended to cause a certifiably robust classifier to abstain from prediction with arbitrarily high probability. I train a substitute model designed to replicate the certifiably robust black-box, run two different adversarial attacks on the substitute model, and then attack the black-box with the adversarial images generated from the substitute model. My attempts were ultimately unsuccessful, so I propose potential explanations and further work.

1 Introduction

1.1 Image Classification and Adversarial Attacks

Image classification has become an important area of machine learning research, with applications in areas from medicine [1] to autonomous vehicles [2]. Researchers have had a great deal of success developing state-of-the-art methods that perform very well on benchmark data sets (e.g. 0.21% error on MNIST [3] and 3.47% error on CIFAR-10 [4]). What is currently less well-understood is how to effectively implement image classification systems in the real world; one reason for this being the existence of adversaries.

Adversarial attacks on an image classification model are attempts to cause the model in question to misidentify an image by adding to it small perturbations. These attacks can undermine the usefulness of an image classification system by causing a model to misclassify images with perturbations that may seem innocuous to humans. For example, Eykholt et al. (2017) [5] cause high-probability misclassification of a stop sign by strategically placing stickers on the sign and Finlayson et al. (2018) [6] are able to reverse diagnoses made by state-of-the-art medical imaging methods. Today, both attacking and defending image classification models are active areas of research.

1.2 Certifiable Robustness

Historically, attack and defense strategies have been judged empirically; that is, how they perform on benchmark data sets against other state-of-the-art defenses/attacks. As such, strategies are optimized and chosen for their empirical performance. One downside to this strategy is that as successful defense strategies are developed, they are often quickly outmatched by new attacks.[7] This has led some researchers to work on defense strategies with certifiable robustness guarantees.

*Code can be found at `https://gitlab.com/ctcovington/am_221_project`

A classifier is considered *certifiably robust* if its prediction for any image x can be verified to be constant for some set around x [8] [9]. One definition of such a set would be the ℓ_2 ball around x. To state the criterion more formally, a classifier $f : \mathbb{R}^n \to \mathcal{Y}$ (where $\mathcal{Y}$ is a set of labels) is certifiably robust with respect to the ℓ_2 ball if

$$\forall x, y \in \mathbb{R}^n : \|x - y\|_2 \leq r \implies f(x) = f(y)$$

where r is a parameter we can choose.

1.3 Randomized Smoothing

Lecuyer et al. (2018) [10] introduced a method for transforming arbitrary classifiers into versions that are certifiably robust in ℓ_2, Li et al. (2018) [11] gave a stronger robustness guarantee, and Cohen et al. (2019) [12] both further strengthened the guarantee and coined the term *randomized smoothing*; it will also serve as the focus of our work.

Cohen et al. describe randomized smoothing as follows. Consider a classification problem with data in $\mathbb{R}^d$ and classes (labels) $\mathcal{Y}$ and an associated classifier f. For our purposes, we will assume that f is a neural network (though randomized smoothing is applicable to arbitrary classifiers). Then, we can construct a new classifier g via randomized smoothing such that

$$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}\left(f(x + \epsilon) = c\right)$$

where $\epsilon \sim N(0, \sigma^2 I)$. It is not possible to exactly compute the distribution of a neural network's output when the input data have been permuted with Gaussian noise, so likewise it is impossible to exactly evaluate our g or determine the radius in which g is robust. Cohen et al. instead use Monte Carlo algorithms to guarantee robustness with arbitrarily high probability.

Imagine that when f classifies $N(x, \sigma^2 I)$, it returns the top two classes A, B with probabilities p_A, p_B. Then g is robust within the ℓ_2 ball with radius $r = \frac{\sigma}{2}\left(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)\right)$, where Φ^{-1} is the inverse of the Gaussian cumulative distribution function. We cannot know p_A, p_B with certainty, so we instead estimate

$$\underline{p_A} \leq p_A, \overline{p_B} \geq p_B$$

which allows us to guarantee that

$$\|\delta\|_2 < r \implies g(x + \delta) = g(x)$$

where $r = \frac{\sigma}{2}\left(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})\right)$.

Cohen et al. note in their certification strategy (computing r from above) that the smoothed classifier g will abstain from prediction under certain conditions. This helps control the rate at which the Monte Carlo sampling fails to return the true underlying value of $g(x)$. Imagine that in n samples of $f(x + \epsilon)$, there are n_A, n_B samples returning classes A and B respectively (A is the top class and B is the runner-up class). Then, the classifier abstains if we fail to reject a two-sided hypothesis test that n_A is drawn from $(\frac{1}{2}, n_A + n_B)$ at an α significance threshold. This α corresponds to our desired upper bound on the probability that $\hat{c}_A \neq g(x)$; i.e. that our Monte Carlo sampling returns a top class that is not the true most likely outcome of $g(x)$.

This abstention happens with low probability in their tests ($\approx 1\%$ on the CIFAR-10 data), but they acknowledge that an adversary could force abstention with arbitrarily high probability were they able to find a perturbation δ such that $f(x + \delta + \epsilon)$ returns c_A and c_B with nearly equal probabilities.

2 Methods and Related Work

2.1 Goal

I assume the role of an adversary, attempting to attack the classifier generated via randomized smoothing by Cohen et al. Normally, an adversary's goal is to cause the classifier to misclassify with

high probability, but the purpose of randomized smoothing is to guarantee a bound on the rate of misclassification. Thus, I will instead attempt to cause the classifier to abstain with high probability. In particular, we will be attacking the classifier Cohen et al. generated with $\sigma = 0.5$ and which was constructed to allow a misclassification rate of at most 0.001. [2] This choice of σ was chosen as optimal by Cohen et al. for radii of robustness $\in \{1, 1.5\}$. Throughout the paper, we will consider the model to be certifiably robust to adversarial noise with ℓ_2-norm ≤ 1.

2.2 Adversary Knowledge and Data Access

When constructing a hypothetical adversarial attack, one must decide the capabilities and knowledge of the adversary.

First, I chose to pursue a black-box attack strategy rather than a white-box attack strategy. In a white-box attack (traditionally the more common type of attack in the academic literature), the attacker has access to model information (architecture, parameters, etc.), whereas in a black-box attack (by my definition at least) an attacker gains knowledge of the model exlusively through queries (feeding the model data and getting predictions).

Second, for reasons explained below, the adversary will want to (as closely as possible) replicate the black-box model they are attacking. For this purpose, I assume the adversary knows general properties of the data on which the model is meant to predict but does not have access to the training data used by the black-box model or the model architecture used. Knowing general properties of data is admittedly a fuzzy concept, so I approximate it by giving the adversary access to similar data to what the black-box algorithm used to train.

Cohen et al. train their model on the standard CIFAR-10 training data [13] packaged with the PyTorch library [14], so I give my adversary access to the CIFAR-10 test data. This is not the most conservative or general approach (there is no reason to believe that an adversary would necessarily have access to data so close to the black-box training set), but it was convenient and streamlined my data preparation process. Querying the black-box model is quite slow (usually 14-15 seconds per observation), so I was not able to test performance when the adversary has access to other training data. [3]

2.3 Black-box Attack Strategy

2.3.1 Overview

As noted above, there exist both white-box and black-box attacks. Much of the early work on adversarial attacks involved white-box attacks [15] [16]. More recently, researchers have been devoting more time to studying black-box attacks.

One promising strategy (which we will implement) is training a substitute model [17] [18]. The idea, in general, is that the adversary builds a data set by querying the black-box model (the adversary provides input images and the black-box output becomes the labels) and then builds a substitute model on the queried data that is meant to resemble the black-box model. The adversary can then use the white-box attack strategy of their choice on the substitute model, the idea being that the adversarial examples learned on the substitute model should perform reasonably well on the black-box model. There is a paper, Chen et al. (2017) [19] that proposes a black-box attack strategy without the need to train a substitute model. I opted for the substitute model strategy primarily for my own edification; I had never implemented a neural network model and wanted to get some experience doing that.

[2] Note that this refers to misclassification relative to the original classifer, not misclassification relative to ground truth.

[3] In fact, I also chose the CIFAR-10 test data in part because I did not have time to query the entire CIFAR-10 training set in time to complete the project.

2.3.2 Substitute Model

For the substitute model, I am using a wide residual network [20] with depth 10. The implementation is based very heavily on that of Zijin Lou.[21] This was chosen because it has shown good performance on CIFAR10 and there were existing PyTorch implementations that required fairly minimal edits to work for my purposes.

When we originally queried the black-box model we got predicted labels in $\{-1, 0, 1, ..., 9\}$ where -1 represents the classifier abstaining from prediction. So, we train our substitute model as if there are 11 classes in our data (treating abstention as its own class).

2.3.3 Attack Strategies

I attempted adversarial attacks via two different methods.

Carlini & Wagner (2016) [CW]
The method developed by Carlini & Wagner (2016) [22] can be written formally as follows:

$$\begin{aligned} &\text{minimize } \|\delta\|_2 \\ &\text{such that } C(x+\delta) = t \\ &\qquad x + \delta \in [0,1]^n \end{aligned}$$

where x is the input image, $x + \delta$ is our adversarial image, C is a function mapping images to classes, and t is the specific class we want the adversary to produce. The optimization is performed via the Adam optimizer.[23]

For our attack, we will set $t = -1$. That is, the attack strategy will attempt to find the minimal perturbation (in terms of ℓ_2 norm) that causes the substitute model to return the *abstention* class. Note that there is no guarantee that an adversarial image produced by this method will fall within the radius for which the black-box is certifiably robust. For a radius of 1 (the smallest value for which Cohen et al. say $\sigma = 0.5$ was the optimal noise among the ones they tested), this never happened in our testing.

Tabacof & Valle (2015) [TV]
Tabacof & Valle (2015) [24] is written as follows:

$$\begin{aligned} &\text{minimize } \|\delta\|_2 \\ &\text{such that } p_{x+\delta}(t) \geq 0.95 \\ &\qquad x + \delta \in [0,1]^n \end{aligned}$$

where x is the input image, $x+\delta$ is our adversarial image, t is the specific class we want the adversary to produce, and $p_{x+\delta}(t)$ is the probability our classifier assigns to class t for our adversarial image $x + \delta$. Optimzation is performed with L-BFGS-B.[25] The ℓ_2 distances provided by this method are not necessarily ≤ 1, so some adversarial images may not fall under the robustness guarantees provided by the randomized smoothing model. For the purposes of calculating the success of an adversarial attack, I consider these examples to be failed attacks [4]. In the actual implementation of the attacks, I also stop the optimization once the constraints are met and $\|\delta\|_2 \leq 1$. I thought that it might be beneficial (in terms of causing the black-box to abstain) not to attempt to minimize the ℓ_2-norm, but rather just to ensure (when possible) that it is under the necessary threshold. The idea was that the black-box will likely be less confident in its predictions close to the edge of its certifiably robust area, so having perturbations that push images close to these edges might be effective.

Note that the primary difference between the two strategies (other than the optimization details) is how to define the constraint pertaining to the output of the adversarial example. CW requires that t is

[4] I provide a more detailed definition in the results section.

the most probable class, while TV requires that the underlying probability of t is ≥ 0.95. I thought it possible that requiring a high underlying probability (as in TV) could lead to more successful generalization from the substitute model to the black-box.

3 Results

3.1 Substitute Model

We first check how successful our substitute model seemed to be in terms of replicating the black-box model. Recall that I used CIFAR-10 test data as our training data, but with the labels replaced by results from the black-box queries. I took a random 90% sample of these data to be my substitute model training set and the remaining 10% as a test set. The substitute model achieved 64.4% accuracy on the test set.

As a point of comparison, I also checked training and set set performance for a model with the same architecture in two different scenarios; training on CIFAR-10 train data and testing on CIFAR-10 test data vs. training on CIFAR-10 test and testing on CIFAR-10 train. I thought that this could give me a sense for the extent to which our model performance is hindered by the decrease in training data size caused by using the CIFAR-10 test set as training data.

	training data: CIFAR-10 train (50,000 observations)	training data: CIFAR-10 test (10,000 observations)
Original data	82.6%	72.3%
Black-Box Queries	?	64.4%

Table 1: Test set performance on CIFAR-10 train vs. test data

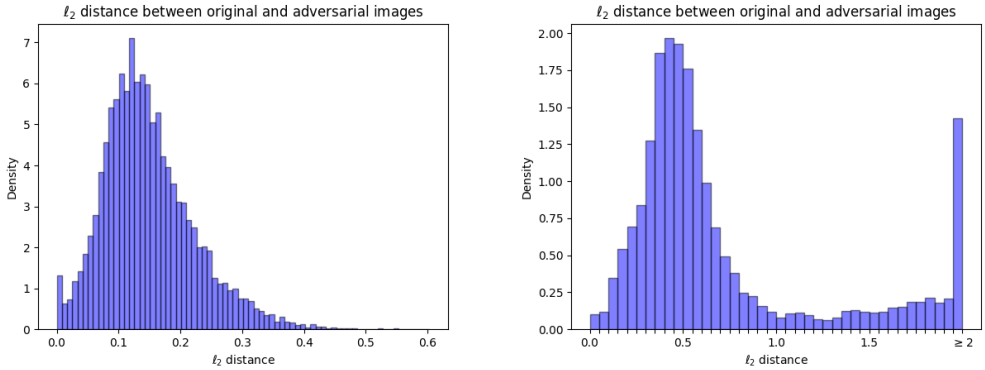
At least for the particular architecture and hyperparameters I am using, it seems like the substitute model does learn the black-box generated data less well than it does the original CIFAR-10 data and that perhaps this would be the case even if my substitute model had access to larger training data. However, it also seems that the lack of accuracy of our substitute model was at least partially a result of the relative lack of training data.

3.2 Attacks on Substitute Model

I attacked the substitute model using the Carlini & Wagner ℓ_2 attack implementation from *Foolbox* [26], a python library designed for carrying out adversarial attacks. I used the CIFAR-10 test set (which also served as the training set for my substitute model) as the base set of images to be perturbed for the attack.

The CW attack is always successful on my substitute model, in that it is able to generate an adversarial image classified as *abstention*. Additionally, the ℓ_2-norms of the perturbations used to create each adversarial image are all ≤ 1, so every adversarial image falls within the set over which our black-box algorithm is certifiably robust.

The TV attack is always able to generate an adversarial image with underlying probability of the *abstention* class ≥ 0.95. However, it is unable to do so with an ℓ_2 distance ≤ 1 with probability ≈ 0.196. So, I consider the attack to be successful on the substitute model with probability ≈ 0.804. When transferring these attacks to the black-box model, I will act as if the adversary simply did not make a change when the ℓ_2 distance is too large. Thus, the black-box will return whatever label it returned on the original data.



(a) Distribution of ℓ_2 distance of perturbations for Carlini-Wagner ℓ_2 attack

(b) Distribution of ℓ_2 distance of perturbations for L-BFGS-B attack

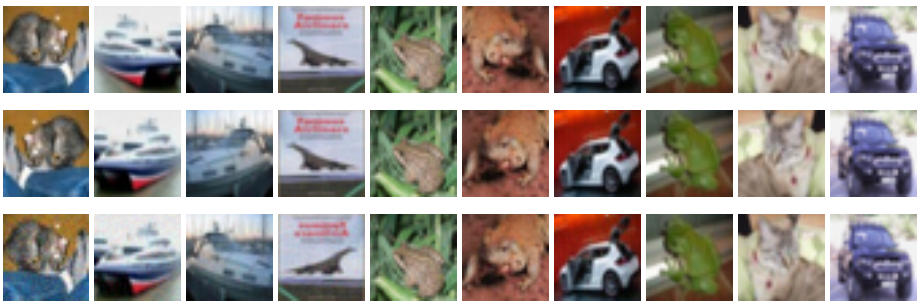


Figure 2: Sample of original images (top) vs. associated adversarial images from CW (middle) and TV (bottom). Some images were flipped horizontally (at random) in the trainng process, so the flipped images above reflect flipping of the baseline image, not that the adversarial attack resulted in an image flip.

3.3 Attacks on Black-Box Model

Though both adversarial attacks were quite effective on the substitute model, they proved to be completely ineffective on the black-box model. There is essentially no difference in the black box model's prediction performance or abstention rate between the original images and either set of adversarial images.

	Original Images	CW Adversarial Images	TV Adversarial Images
Prediction Performance	70.62%	70.97%	70.83%
Abstention Rate	0.99%	0.87%	0.99%

Table 2: Black-Box model performance on original and adversarial images

	abstain	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
abstain	36	3	5	6	12	11	7	5	4	2	8
plane	1	962	4	9	4	2	1	2	4	8	2
car	4	2	945	0	0	0	1	0	0	4	13
bird	5	8	2	853	11	17	6	11	2	2	2
cat	8	2	3	14	902	5	22	12	6	2	3
deer	13	4	0	20	13	867	5	9	7	2	0
dog	3	5	0	8	23	3	839	4	12	0	1
frog	10	1	2	7	10	8	2	1082	1	1	0
horse	2	3	1	4	4	14	16	4	913	0	6
ship	1	10	3	1	3	0	1	1	0	1046	6
truck	4	5	16	0	7	1	1	3	4	6	987

Table 3: Confusion matrix of black box output on original images (rows) and CW adversarial images (columns)

	abstain	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
abstain	37	3	2	7	9	11	6	6	5	5	8
plane	4	956	3	10	3	3	1	2	4	11	2
car	5	1	941	2	1	1	1	1	0	3	13
bird	3	3	2	861	6	17	8	12	4	1	2
cat	4	2	0	6	916	7	30	8	3	0	3
deer	6	2	0	11	3	906	1	4	6	0	1
dog	9	3	0	11	26	6	826	6	9	1	1
frog	13	1	2	7	7	4	4	1084	0	0	2
horse	7	4	1	4	5	14	10	4	911	0	7
ship	4	11	3	0	1	1	0	0	0	1052	0
truck	7	7	18	1	3	0	0	5	3	8	982

Table 4: Confusion matrix of black box output on original images (rows) and TV adversarial images (columns)

4 Discussion

My attempts to cause the certifiably robust classifier to abstain with high probability were clearly unsuccessful. I would like to take some time now to note some strategies I would try if I had more time.

4.1 Substitute Model

My first step would certainly be to look into training a better substitute model. My entire strategy is predicated on the idea that attacks on the substitute model will transfer effectively to the black-box model, which they obviously did not in my case. There is evidence of adversarial examples transferring well to new models [27], though the new models were not certifiably robust. It is possible that the certifiably robust model is particularly robust against attempts to transfer adversarial attacks, but I would not necessarily assume it. At the very least, having a substitute model that more closely replicates the black-box would help me narrow down the set of possible reasons my strategy was unsuccessful.

In terms of building a better model, I would start by giving my substitute model access to more training data. It trained on the modified CIFAR-10 test set (10,000 observations), but I think it could have really benefited from training on the modified CIFAR-10 training set (50,000 observations).[5] If this proved successful, I would also give the adversary access to other training sets (e.g. CIFAR-10.1

[5] "modified" in the sense that the labels are replaced by predictions from the black-box model.

[28] [29], MNIST, random noise, etc.) to see how the attack strategy performs when the adversary has access to data at varying levels of similarity to those on which the black-box was trained.

4.2 Attack Strategies

For each of the attack strategies I tried, I treated *abstention* as its own class. It is possible that this was not the best strategy, and that I should have tried to optimize for the exact set of circumstances that causes the black-box classifier to abstain.[6] I am not quite sure how this would work, but at a high level I think I would want to create an objective function that reflects that we want the probabilities of the top two classes to be as close as possible. Perhaps I could do something like attempting to minimize the square of the difference of the top two probabilities.

4.3 Change of Focus

In hindsight, I probably should have focused first on implementing a white-box attack on the certifiably robust model from Cohen et al. (2019). This would have avoided the need for training a substitute model (thus avoiding the associated difficulties with generating training data), and also would perhaps have been a more logical first step in a research agenda (first perform a white-box attack, then move to black-box, then test transferrability to new training sets, etc.)

[6] That is, failure to reject the hypothesis test described in the introduction.

References

[1] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, and Andrew Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *CoRR*, abs/1711.05225, 2017.

[2] Claudine Badue, Ranik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius Brito Cardoso, Avelino Forechi, Luan Ferreira Reis Jesus, Rodrigo Ferreira Berriel, Thiago Meireles Paixão, Filipe Wall Mutz, Thiago Oliveira-Santos, and Alberto Ferreira de Souza. Self-driving cars: A survey. *CoRR*, abs/1901.04407, 2019.

[3] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1058–1066, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[4] Benjamin Graham. Fractional max-pooling. *CoRR*, abs/1412.6071, 2014.

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

[6] Samuel G. Finlayson, Hyung W. Chung, Isaac S. Kohane, and Andrew L. Beam. Adversarial attacks against medical deep learning systems. *CoRR*, abs/1804.05296, 2018.

[7] Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *CoRR*, abs/1705.07263, 2017.

[8] J. Zico Kolter and Eric Wong. Provable defenses against adversarial examples via the convex outer adversarial polytope. *CoRR*, abs/1711.00851, 2017.

[9] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *CoRR*, abs/1801.09344, 2018.

[10] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy, 2018.

[11] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Second-order adversarial attack and certifiable robustness, 2018.

[12] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. *CoRR*, abs/1902.02918, 2019.

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[14] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2014.

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

[17] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016.

[18] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information, 2018.

[19] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. 2017.

[20] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[21] Zijin Lou. pytorch-cifar, 2018.

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

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[24] Pedro Tabacof and Eduardo Valle. Exploring the space of adversarial images. *CoRR*, abs/1510.05328, 2015.

[25] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, December 1997.

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

[27] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016.

[28] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? 2018. `https://arxiv.org/abs/1806.00451`.

[29] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.