Batch-wise Logit-Similarity: Generalizing Logit-Squeezing and Label-Smoothing

Ali Shafahi
ashafahi@cs.umd.edu
Amin Ghiasi
amin@cs.umd.edu
Mahyar Najibi
najibi@cs.umd.edu
Furong Huang
furongh@cs.umd.edu
John Dickerson
john@cs.umd.edu
Tom Goldstein
tomg@cs.umd.edu

Department of Computer Science,
University of Maryland,
College Park,
Maryland, USA

Abstract

We study how cheap regularization methods can increase adversarial robustness. In particular, we introduce logit-similarity which can be seen as a generalization of label-smoothing and logit-squeezing. Our version of logit-squeezing applies a batch-wise penalty and allows penalizing the logits aggressively. By measuring the robustness of our models against various gradient-based and gradient-free attacks, we experimentally show that, with the correct choice of hyper-parameters, regularized models can be as robust as adversarially trained models on the CIFAR-10 and CIFAR-100 datasets when robustness is measured in terms of ℓ_∞ attacks. Unlike conventional adversarial training, regularization methods keep training time short and become robust against ℓ_2 attacks in addition to ℓ_∞.

1 Introduction

Deep Neural Networks (DNNs) have been instrumental to the success of many computer vision related tasks such as classification [16], and object detection [10]. While machine learning models have achieved human-level performance on some computer vision benchmarks, they have been shown to perform poorly on images that have been even slightly perturbed [3, 34]. When slightly perturbed images get misclassified with high confidence, we call them *adversarial examples*. The susceptibility of models to adversarial examples has raised many security concerns for safety-critical tasks such as traffic sign detection and classification in autonomous vehicles [19].

Various methods have been proposed for defending against adversarial examples. Some work by removing the adversarial noise using Generative Adversarial Networks (GANs) [29,

[31], by training a noise removal autoencoder [24], or by squeezing the input image features [38]. In [2], many simple defenses were broken, including thermometer encoding [5], using local intrinsic dimensionality as a method for detecting adversarial examples [20], input transformations such as compression and image quilting [12], stochastic activation pruning [8], adding randomization during inference time [37], purifying the adversarial image by modifying the example to increase it's probability of belonging to the training distribution [32], and defending with the help of generative models [29]. These defenses were broken using Expectation Over Transformation (EOT) [1] for randomized defenses or Backward Pass Differentiable Approximation (BPDA).

BPDA is a new attack tailored for defenses that work by "obfuscating" the gradients, a concept similar to "gradient masking." A popular defense mechanism that was shown by [2] to withhold its claims against such revised attacks is adversarial training [21]. Adversarial training is the process of training the model on generated adversarial examples. Some earlier versions of adversarial training were on a mixture of clean and adversarial examples [11], while some only train on adversarial examples [21]. Most adversarial training schemes are based on the robust optimization min-max formulation eq. (1).

$$\begin{aligned} &\min_{\theta} \max_{\delta} \quad l(x+\delta, y, \theta) \\ &\text{subject to } \ ||\delta||_p \leq \varepsilon, \end{aligned} \tag{1}$$

where θ represents the trainable parameters of the model, δ is the perturbation which is controlled by the adversary, ε is the bound on the perturbation, l is usually the regular training loss function (*i.e.* cross-entropy for classification), and $||.||_p$ is some p-norm[1].

While adversarial training has been successful in increasing robustness, it is computationally expensive. Very recently, a computationally cheap defense named logit-squeezing was introduced in [13]. Logit-squeezing is proposed as a result of experiments showing the effectiveness of another defense method called clean logit pairing and suggesting that its robustness may be due to the logits taking on small values. Therefore they proposed training with

$$\underset{\theta}{\text{minimize}} \quad l(x, y, \theta) + \beta ||z(x)||_2, \tag{2}$$

where, $z(x)$ is the logit vector for example x. This formulation yields promising results for logit-squeezing on MNIST when combined with random Gaussian data augmentation.

Logit-squeezing has been studied in detail, it has been combined with other regularizers [33]. Some previous studies suggested that logit-squeezing is ineffective by showing that it does not stand up against scrutiny. They show that for their models, making the attack stronger (*i.e.*, increasing the number of Projected Gradient Descent (PGD) iterations or random restarts) breaks the defense [26]. Some argue that defenses which regularize the logits work by making the loss-landscape jagged and more difficult for gradient-based attacks [9] which again can be countered by performing attacks with more iterations or using gradient-free attacks.

In this study, we dig deeper into logit-squeezing and show that if correctly applied, it indeed does increase the robustness against attacks even with many iterations. This is in contrast to recent studies, which failed to observe these benefits when sub-optimal parameter settings were used. We take a step further and explain why we believe logit-squeezing

[1] Usually ℓ_∞

works. Using our insights, we show that we can, to some extent, mimic the benefits of logit-squeezing using other methods such as label smoothing. We also generalize logit-squeezing, and consider cases where the logits are forced towards a common value other than zero.

2 Adversarial attacks

Adversarial attacks/perturbations come in various forms. A perturbation can be universal [25, 30] or tailored for particular instances. Per-instance attacks can be localized as patches [4, 14, 27]. This work focuses on per-instance perturbations where their pixel values are only bounded (i.e., ℓ_p-bounded adversarial examples). In this section, we briefly review some of the famous adversarial *per-instance* attacks, and describe how they will be used to evaluate defense methods. Adversarial examples can be crafted by finding the minimum perturbation expressed in any p-norm that is required to change the category of the example. Optimization-based formulations exist that are geared towards finding a bounded perturbation that minimizes accuracy by maximizing the cross-entropy loss (Eq. 3).

$$\begin{aligned} \underset{\delta_i}{\text{maximize}} \quad & l(x_i+\delta_i, y_i) \\ s.t. \quad & ||\delta_i||_p \leq \varepsilon \end{aligned} \tag{3}$$

The loss function, l, for the PGD attack can be any meaningful loss such as the cross-entropy loss (xent) or the Carlini-Wagner (CW) [6] loss which aims to cause an incorrect logit (z_w) to be larger than the correct logit (z_y) by solving Eq. 4. PGD attacks are among the strongest experimental methods for generating bounded adversarial examples [2]. The PGD algorithm from [21] is essentially the Basic Iterative Method (BIM) attack [17] that starts from a random initialization. BIM itself is an iterative version of the FGSM attack [11].

$$\begin{aligned} \underset{\delta_i}{\text{maximize}} \quad & -RELU(z_y(x_i+\delta_i) - z_w(x_i+\delta_i)) \\ s.t. \quad & ||\delta_i||_p \leq \varepsilon \end{aligned} \tag{4}$$

3 Batch-wise logit squeezing

As briefly mentioned before, logit-squeezing was argued to be ineffective when tested against PGD attacks with many iterations. However, the true robustness of logit-squeezing depends on the choice of hyper-parameters – and, unlike other regularization methods, the hyper-parameters must be *large* before effects are realized. In our version (Eq. 5), we penalize the logit matrix for the entire mini-batch instead of penalizing the logit vector for every example as done in eq. (2). This leads the penalty to focus on outliers of the entire mini-batch and smooths training behavior when β assumes large values.

$$\underset{\theta}{\text{minimize}} \quad \sum_b l_b(x_b, y_b, \theta) + \frac{\beta}{b_n} ||Z(x_b)||_F, \tag{5}$$

Here l_b is the average cross-entropy loss for the mini-batch b, and b_n is the mini-batch size.[2]

[2] In all our experiments, the batch-size was set to 128.

3.1 Ablation study on logit squeezing parameters

We use the MNIST dataset [18] and the architecture and hyper-parameters from [23] to perform an ablation study showing the impact of logit-squeezing hyper-parameters, which include: (a) number of training iterations (k - Fig. 4); (b) logit-squeezing coefficient (β - Fig. 2); and (c) the standard deviation of the Gaussian augmentation (σ - Fig. 3). We also compare the robustness of our batch-wise logit squeezing formulation (Eq. 5) with the robustness of the original method (Eq. 2) in Fig. 1. We measure robustness taking the minimum accuracy of the examples made using 40-step PGD attacks on both the cross-entopy loss (xent) and the Carlini-Wagner loss (cw). We use a standard ℓ_∞ $\varepsilon = 0.3$. With our batch-wise logit-squeezing formulation, we are able to build robust models using a wide-range of β-values more easily (Fig. 1). Also, using $\sigma \geq \varepsilon$ helps boost robustness (Fig. 3). Finally, increasing the number of training iterations k helps improve robustness (Fig. 4). The last may be simply because we are sampling more points and enforcing the logits to be squeezed at many points sampled around the input. Based on our MNIST experiments, we opted on using $\sigma > \varepsilon$ and doubling the number of training iterations for the CIFAR experiments in the following sections.

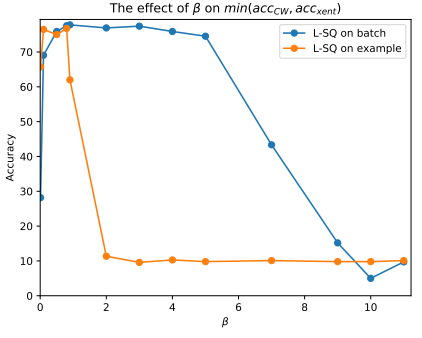

Fig. 1: Batching effect: Per-example logit squeezing versus batch-wise logit squeezing.

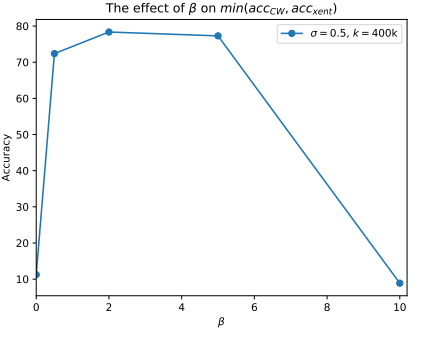

Fig. 2: Effect of coefficient of batch-wise logit squeezing (β) on robustness.

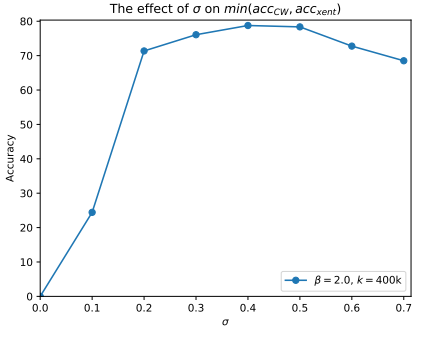

Fig. 3: Effect of additive Gaussian standard deviation (σ) used during training.

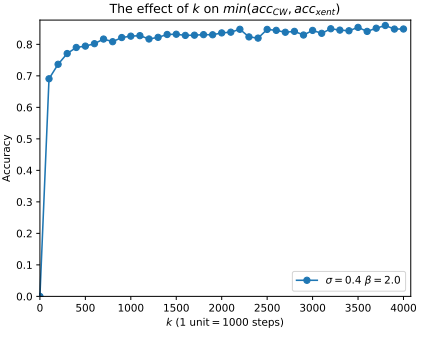

Fig. 4: Effect of number of training iterations (k) on robustness.

3.2 Robustness against white-box attacks on CIFAR-10

To see the effects of our batch-wise logit-squeezing on CIFAR-10, we train various CIFAR-10 [15] models[3] by varying the logit-squeeze parameter β. During all experiments, we set the standard deviation of the Gaussian noise added during training to be equal to 20 and train the models for 160,000 iterations. We study the robustness of the logit-squeezed CIFAR-10 models against white-box ℓ_∞-bounded PGD attacks under various number of attack iterations i. In these experiments, we use a step-size of $\varepsilon_s = 2$[4] and bound the perturbation by $\varepsilon = 8$. For comparison to well-known state-of-the-art defenses, we also report the robustness of the 7-step adversarially trained model from [21] which has similar hyper-parameters and network architecture as our models. While the 7-step PGD adversarially trained model is trained for fewer iterations (160,000), every iteration takes roughly 8 times more computation since it does training on adversarial examples made using a $i = 7$ PGD attack.

We report the result of PGD attacks on the cross-entropy loss and CW loss in table 1. We see that an increase in the squeezing parameter (β) causes the robustness (*i.e.* accuracy on adversarial examples) to increase and the accuracy on clean test examples to slightly worsen. Increasing the number of iterations (i) of the PGD attack degrades the robustness further (fig. 5). Therefore, the robustness against weak adversaries with limited i is not reflective of the true robustness against white-box attacks. However, the robustness drop plateaus after 100 iterations as shown in fig. 5. As a result, throughout the paper, we report robustness against PGD attacks with $i = 200$ iterations.

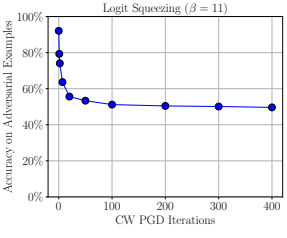

Fig. 5: Accuracy on adversarial examples generated by the PGD attack on the CW loss (worst case for logit-squeezed models) and different numbers of iterations. The logit-squeeze parameter used during training is $\beta = 11$. The robustness drop almost plateaus after 100 iterations. We report the robustness of our models on $i = 200$-iteration PGD attacks.

With fewer attack iterations (FGSM), the robustness of the logit-squeezed models is very high compared to the 7-step PGD adversarially trained model. Our best logit-squeezed model is more robust than the adversarially trained model even against attacks with 200 iterations. Also, notice that the PGD attack on the CW loss is more powerful compared to the PGD attack on the xent loss for the logit-squeezed models. Therefore, in section 5.1.2, when we are going to evaluate the robustness of our models on attacks which take days, we use the worst-case-loss (*i.e.* CW).

We also notice that increasing the number of iterations hurts the models with smaller β coefficients more in comparison to those which have larger β values. For this reason, the conclusions drawn in earlier studies using small β coefficients do not necessarily characterize the performance of this method. Due to our batch-wise equations, we can increase the β coefficient. The conclusions of [26] were drawn based on very different hyper-parameter choices and a different objective (Eq. 2).

When comparing the performance of aggressive batch-wise logit-squeezing (eq. (5)) and adversarial training, we notice that even with an increased number of PGD-iterations, the

[3] We use the wide-resnet WRN-32-10 architecture and apply standard data-augmentation techniques. Our choice of hyper-parameters is similar to [21].

[4] We experimented with different values of ε_s but the results did not change significantly.

defense	Test	PGD attacks on the xent			PGD attacks on the CW		
		20-xent	50-xent	200-xent	20-CW	50-CW	200-CW
$\beta = 5$	92.45%	43.26%	43.25%	38.86%	45.50%	37.82%	33.91%
$\beta = 10$	**92.68%**	52.55%	45.18%	40.83%	47.48%	41.39%	37.87%
$\beta = 11$	92.08%	**58.51%**	**55.91%**	**53.87%**	**55.63%**	**53.56%**	**50.44%**
7step-AdvT	87.25%	45.84%	45.39%	45.32%	46.90%	46.66%	46.48%

Table 1: White-box PGD attacks on robust CIFAR-10 models. Increasing the β parameter increases robustness at the cost of accuracy.

model with $\beta = 11$ is more robust compared to adversarial training. Also, its accuracy on clean examples is about 5% more. If we keep on increasing the value of β, after some threshold, the model's classification accuracy and robustness both drop as it was also seen for the MNIST experiments in Fig. 2.

4 So, why is logit squeezing working?

One naive hypothesis based on the competing terms in Eq. 5 is that the logit regularizer term is preventing the xent loss from taking on very small values (*i.e.*, overfitting). Therefore, over-fitting might be the reason for vulnerability. While true, preventing over-fitting is not the major reason but just a side-effect. To verify this, we prevented the cross-entropy loss from taking on very small values by training on a clipped version of the xent loss

$$\underset{\theta}{\text{minimize}} \sum_i \text{clip}(l_i(x_i, y_i, \theta), \omega), \tag{6}$$

where ω is the minimum cross-entropy loss parameter, l_i is the cross entropy loss of example i, and $\text{clip}(., \omega) = \max(., \omega)$. Training with eq. (6) does not yield training accuracy better than 50% when $\omega \geq 1.5$, which is even slightly less than the cross-entropy we would get for the case where $\beta = 10$. Clipping the average cross-entropy loss of every mini-batch instead of just clipping the cross-entropy loss per-example does not work either. Consequently, it is not possible to get accurate and robust models solely by enforcing high cross-entropy values. Higher cross-entropy values and robustness against over-fitting are rather the side-effects of logit-squeezing.

By observing the loss plots projected on random and adversarial directions, the main reason for robustness becomes more clear. Robustness is achieved since the regularization find minima that are not only accurate but at the same time flat. Here, the flatness is measured with respect to the input image. So, the question is how can we make the xent loss with respect to the images flat? This is probably achievable at the cost of increasing the xent loss for the clean training examples. One possible strategy is to force all logits to remain close to each other. If the logits are close to each other for both the clean training image and many sampled points surrounding the clean image (*i.e.* clean image + Gaussian noise), it is not possible to change the cross-entropy loss much. This can correspond to flatness in the cross-entropy loss metric. We believe that this is the driving force behind the effectiveness of our "aggressive" logit-squeezing.

5 How else can we imitate the desirable properties of logit-squeezing?

If our hypothesis about "aggressive" logit-squeezing is correct, we should be able to get similar behavior by enforcing the desired similarity property on the logits. In this section, we describe two possible ways of doing this.

5.1 logit-similarity: generalizing logit-squeezing and focusing on its core

If all that matters is having the logits close to each other, then we do not need to force the logits to be small. Being small can be interpreted as being similar and close to zero. We can alternatively force the logits to cluster together around any fixed value. Therefore we define "logit similarity" which is simply a generalization of our batch-wise logit-squeezing based on our hypothesis. Logit similarity can be imposed by training with the regularized loss function

$$\underset{\theta}{\text{minimize}} \quad \sum_b l_b(x_b, y_b, \theta) + \beta' / b_n ||Z(x_b) - \gamma||_F. \tag{7}$$

where, γ can take on any value, β' is the similarity penalty, and b_n is the training mini-batch size which we set to be 128. We use $\gamma = 1$ in our experiments.

5.1.1 Aggressive logit similarity results on CIFAR-10

We apply the same augmentation and hyper-parameters, and deploy the same network architecture used in section 3.2. The results for different similarity penalty parameter β' are summarized in table 2. As expected, the robustness of these models are relatively high and comparable to that of adversarial training which experimentally validates our hypothesis.

	PGD attacks on the xent and CW loss			
defense	20-xent	200-xent	20-cw	200-cw
$\beta' = 5$	44.00%	30.59%	40.77%	28.65%
$\beta' = 10$	47.28%	35.67%	45.52%	34.56%
$\beta' = 11$	**56.36%**	**49.79%**	**56.74%**	**50.33%**
7step-AdvT	45.84%	45.32%	46.90%	46.48%

Table 2: White-box PGD attacks on logit-similar CIFAR-10 models. We use $\varepsilon = 8$ ℓ_∞ attacks.

5.1.2 Other types of attacks: multiple restart PGD, ℓ_2, and gradient-free attacks

Recently, using multiple random restarts is shown to sometimes be more effective than increasing the number of attack iterations [26]. Attacking with more random restarts has decreased the accuracy of adversarially trained models [22]. We therefore attempt attacking our model with $\beta' = 11$ using 10 random restarts each with 100 iterations. We attack the CW loss because we find it results in stronger attacks. Under this strong attack, the robustness of our model is 45.27% which is comparable to that of the 7-PGD adversarially trained model.

Furthermore, while robustness against ℓ_∞ attacks is the main measure used in many studies, we also investigate the robustness of our models against ℓ_2 attacks. Note that it has been shown that models adversarially trained against ℓ_∞ attacks are susceptible to ℓ_2 attacks [28]. Given that we do not use adversarial examples of any kind during training, we speculate whether our models are robust against other types of attacks as well. Following [28], we attack the models using an ℓ_2 perturbation budget of $\varepsilon = 1.5 \times 255$. Under a 10-restart ℓ_2 attack with $i = 100$ iterations, the 7-PGD trained model from [21] achieves only 15.36% while our logit-similar model with $\beta' = 11$ achieves **54.99%**.

Finally, we also evaluate the performance of our model against SPSA, which is considered one of the most effective gradient-free attacks [35]. Gradient-free attacks are useful in situations where the loss function is noisy or the gradients are not accurate. This attack is very expensive and running 20 iterations of it on 2048 instances from the validation set took 3 days and performed worst (88.13%) than the gradient-based PGD attacks. This illustrates that our methods do not work by masking the gradients [7].

5.2 Label smoothing

Another way to promote clustering of logits during training is to smooth the ground-truth labels. Label smoothing refers to making the "one-hot" label vectors into "one-warm" vector. Given a smoothing factor $0 \leq \lambda \leq 1$, we can smooth the labels using the update

$$\mathbf{y}_{warm} = \mathbf{y}_{hot} - \lambda \times (\mathbf{y}_{hot} - \frac{1}{N_c}), \tag{8}$$

where $\mathbf{y}_{hot}$ is the one-hot vector label and N_c is the number of classes.

Label smoothing is known to increasing model robustness to some degree [36]. Like logit-squeezing, we suggest applying label smoothing very aggressively in conjunction with random data augmentation. Once the labels are transformed into warm labels, we train on the cross-entropy loss function using the warm vectors as the true labels. Here, we propose that the aggressive application of label smoothing plus random Gaussian augmentation can yield the desired properties for achieving high degrees of robustness at a cheap cost. We propose using very large values of λ. Note that, for CIFAR-10 where $N_c = 10$, $\lambda = 0.95$ means that the correct class will take on the warm probability of 0.145 and the other classes will each have 0.095 probability. Table 3 summarizes the results for two aggressive choices of λ.

	PGD attacks on the xent and CW loss		
defense	Test	20-xent	20-cw
$\lambda = 0.9$	92.60%	43.30%	39.76%
$\lambda = 0.95$	**92.88%**	43.00%	41.29%
7step-AdvT	87.25%	**45.84%**	**46.90%**

Table 3: White-box PGD attacks on an aggressive label-smoothed CIFAR-10 models. We use ℓ_∞ attacks with $\varepsilon = 8$.

6 Results on tasks with more classes: CIFAR-100

Given the promising performance of these cheap regularizers on CIFAR-10, one could wonder whether these aggressive methods for enforcing logit-similarity can increase robustness

in scenarios where the logit-vector is higher-dimensional and more sensitive? To answer this question empirically, we train robust classifiers for the CIFAR-100 dataset, which has 100 classes [15]. The network architecture and hyper-parameters were similar to that used for CIFAR-10 in section 3.2. To build a better sense of the robustness of our hardened classifiers, we adversarially trained two robust classifiers for the CIFAR-100 dataset. One was trained with 2-step PGD adversaries and the other with 7-step PGD adversaries. They both require considerably more computation than our aggressive logit-squeezing. The results are summarized in table 4.

	PGD attacks on the xent and CW loss			
defense	20-xent	200-xent	20-cw	200-cw
$\beta = 1$	23.89%	18.99%	11.91%	9.00%
$\beta = 5$	30.91%	26.00%	19.80%	15.79%
$\beta = 7$	**31.99%**	**30.05%**	**25.92%**	**23.87%**
2step-AdvT	17.08%	16.49%	17.80%	17.52%
7step-AdvT	22.76%	22.42%	23.12%	22.95%

Table 4: White-box PGD attacks on our logit-squeezed and adversarially trained CIFAR-100 models. We use ℓ_∞ attacks with $\varepsilon = 8$.

Given the correct choice of hyper-parameters, our CIFAR-100 robust model can get comparable and slightly better results than adversarial training, even against strong adversaries with many iterations.

7 Conclusions

We make an argument in defense of logit-squeezing; it does indeed increase robustness when used in conjunction with large hyper-parameters and our batch-wise objective. We explored the mechanisms by which logit-squeezing acts. After identifying that logit-squeezing works by making the logits similar to each other across data samples, we proposed and validated logit-similarity as a generalization of logit-squeezing. We also illustrated that label smoothing can be used to get similar benefits using similar mechanisms – they all cluster the logits. We show that the logit-similarity regularized CIFAR-10 and CIFAR-100 models acheive robustness comparable to PGD adversarially trained models against ℓ_∞ attacks with similar hyper-parameters and architectures.

We also show that since these regularized defenses are not trained on adversarial examples generated using any particular attack, they can outperform adversarially trained models when confronted with attacks in other p-norms. In particular, we show that our logit-similar trained model achieves 54.99% robustness against ℓ_2 attacks while the ℓ_∞ adversarially trained model only achieves 15.36%. Our hope is that such simple regularization methods can provide fast and efficient ways to achieve robustness without the complexity and expense of adversarial training.

References

[1] Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*, 2017.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[3] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[4] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

[5] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018.

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.

[7] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, and Aleksander Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[8] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

[9] Logan Engstrom, Andrew Ilyas, and Anish Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

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

[12] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[13] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[14] Danny Karmon, Daniel Zoran, and Yoav Goldberg. Lavan: Localized and visible adversarial noise. *arXiv preprint arXiv:1801.02608*, 2018.

[15] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

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

[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. No need to worry about adversarial examples in object detection in autonomous vehicles. *arXiv preprint arXiv:1707.03501*, 2017.

[20] Xingjun Ma, Bo Li, Yisen Wang, Sarah M Erfani, Sudanthi Wijewickrema, Michael E Houle, Grant Schoenebeck, Dawn Song, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*, 2018.

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[22] MadryLab. Cifar-10 adversarial examples challenge, 2018. URL https://github.com/MadryLab/cifar10_challenge.

[23] MadryLab. Mnist adversarial examples challenge, 2018. URL https://github.com/MadryLab/mnist_challenge.

[24] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.

[25] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[26] Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. Logit pairing methods can fool gradient-based attacks. *arXiv preprint arXiv:1810.12042*, 2018.

[27] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1300–1307. IEEE, 2019.

[28] Haifeng Qian and Mark N Wegman. L2-nonexpansive neural networks. *arXiv preprint arXiv:1802.07896*, 2018.

[29] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[30] Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. *arXiv preprint arXiv:1811.11304*, 2018.

[31] Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. *ICLR Submission, available on OpenReview*, 4, 2017.

[32] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

[33] Cecilia Summers and Michael J Dinneen. Logit regularization methods for adversarial robustness. 2018.

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

[35] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

[36] David Warde-Farley and Ian Goodfellow. 11 adversarial perturbations of deep neural networks. *Perturbations, Optimization, and Statistics*, page 311, 2016.

[37] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[38] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.