Robust Defense Against L_p-Norm-Based Attacks by Learning Robust Representations

CSCI 699: Adversarial Machine Learning
Spring 2020 Final Project Report

Iordanis Fostiropoulos
University of Southern California
fostirop@usc.edu

Basel Shbita
University of Southern California
shbita@usc.edu

Myrl Marmarelis
University of Southern California
mmarmare@usc.edu

Abstract

Image classifiers are vulnerable to adversarial perturbations. Previous work [8, 9] proposed restrictions of the feature space as a defense mechanism against L_p-Norm attacks. In this work we evaluate the claim that adversarial examples are caused by the difficulty of DNN to learn robust representations. We propose a regularization objective based on the Euclidean distance between classes in the hypothetical latent space found in the features of the penultimate layer. Our novel objective increases the size of perturbation epsilon ε required to generate the aforementioned adversarial examples. We evaluate our approach and conclude that successful attacks require an exceedingly large perturbation budget ε that puts into question the true class of the adversarial image. Our approach is significantly more robust than that of the previous state of the art, and entails an insignificant fine-tuning of existing models.

1 Introduction

In recent years, deep neural networks (DNNs) have exhibited their superiority over traditional models in an abundance of machine learning tasks such as image classification. Unfortunately, as DNN models become more widely deployed, the incentive for defeating and exploiting them by adversaries increases. Various kinds of attacks can easily manipulate the behavior of these models and impose great difficulty for users. Some adversarial attacks can purposely sway the output of the network while being imperceptible to humans [16]. Such adversarial samples pose a serious threat for security- and safety-critical applications. There is broad evidence in certain domains that adversaries deliberately alter their methods in order to prevent detection [17, 14].

The inability of existing state of the art models to detect those samples reflects that the best performing models do not interpret the basic visual principles in a distinctive manner, which in turn can lead the model to fail in its most critical tasks. Therefore, we require new methods to allow DNNs to learn robust representations and design systems that are reliable and resilient.

1.1 Adversarial Attacks

Much effort has been put into crafting adversarial attacks [13, 12] and on creating defenses that are more useful against a specific type of attacks [15]. The rise of such attacks has outpaced the development of robust defenses. The adversary's observations of the learning paradigm are what constitutes the inputs for the attack. There are two settings in which attacks are performed by adversaries. A black-box setting, in which the adversary's observation is limited to the output of the model on arbitrary inputs, and a white-box setting, in which the attacker has access to the full model, notably its architecture and parameters, thus,

Preprint. Under review.

he can also observe the intermediate computations at hidden layers and can use all of those to generate its attacks. The white-box adversary is considered the strongest adversary.

The degree to which an adversarial example is imperceptible from its benign original is usually measured using L_p-norms, e.g., L_0 (e.g., [10]), L_2 (e.g., [16]), or L_∞ (e.g., [2]). These measures of imperceptibility are critical for creating adversarial examples and defending against them. Using these measures for the evaluation of distance between images is not sufficient by itself. Many defenses frequently rely on L_p-norm-based data sanitization methods, aiming to remove poisoned or otherwise anomalous data [11], are still vulnerable to adaptive attacks, being unable to generalize well. Thus, they cannot provide a robust solution against the continuously-adapting adversaries.

1.2 Representation Learning

The performance of most machine learning models depends heavily on the choice of the representation of the data on which they are applied. A good representation is one that makes a subsequent learning task easier. Representation learning offers one way of doing unsupervised and semi-supervised learning. Specifically, we can learn good representations for unlabeled data, and then use those representations in the same setting to solve the supervised learning problem. The main idea behind this approach is the notion that learning about the distribution of inputs will aid in learning the mapping from inputs to outputs and thus enable higher robustness and performance in models. Most representation learning problems face a trade-off between retaining as much input information as possible and achieving nice properties in the latent space. Ultimately, we want to incorporate this idea in our task to achieve a representation in latent space which in turn will be used as an effective input to a final classifier and make the image classification task easier and more robust against L_p-norm-based attacks.

1.3 Contribution

We propose a learning objective on image classifiers that directly improves robustness. By altering the composition of the final feature space, we control the kind of information encoded in this representation that drives the image's classification. Our added constraint is a triplet loss with a margin hyper-paramter: each training sample is encouraged to reside nearby other samples of the same class, separated in Euclidean distance by those of other classes by a fixed minimal threshold.

2 Background

2.1 Triplet Loss

The triplet loss, endowed with a distance function, pushes the samples of each class into their own disjoint clusters. Mao et al. [8] recently proposed a similar line of inquiry that they fashion as metric learning, since the triplet loss operates on the notion of a metric. They chose the angular distance in the penultimate feature space; we employed the Euclidean norm. Our justification is as follows: the Euclidean clustering has information-theoretic benefits that sprout from the VAE [3] literature. In harnessing those same results, we can exploit an existing framework that justifies the minimization between the Euclidean norm of same-class samples in the latent space.

2.2 Adversarial Attacks

The main goal of an attack algorithm is to force a trained DNN to make wrong predictions using a minimal perturbation budget. The attacker's objective can be formalized as follows:

$$\underset{\delta}{\operatorname{argmax}} \mathcal{L}(x+\delta, y), \qquad \text{s.t., } \|\delta\|_p < \varepsilon \tag{1}$$

Where y is the ground-truth label for an input sample x, δ denotes the adversarial perturbation, $\mathcal{L}$ denotes the loss function, $\|\ \|_p$ denotes the p-norm, and ε is the available perturbation budget. In this work, we focus on four recently proposed state-of-the-art attacks, which will be used to evaluate our model's robustness. Figure 1 shows some of the perturbed adversarial samples we generated using the methods presented below.

2.2.1 Fast Gradient Sign Method (FGSM)

One of the first successful attack methods is the fast gradient sign method [2]. The FGSM is a one-step attacking method. The generated adversarial sample $\tilde{x}$ corresponding to a source input x and true label y is $\tilde{x} = x + \varepsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x, y))$. Figure 1a shows some samples we generated using this method.

2.2.2 Projected Gradient Descent (PGD)

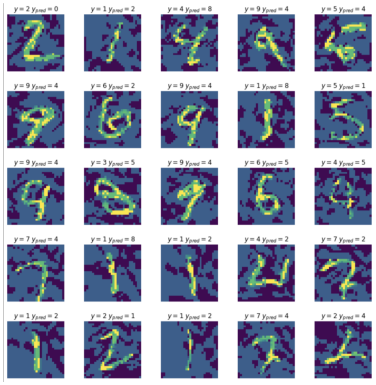
The PGD method [7] applies FGSM for several iterations with a step size of γ as $x_i = x_{i-1} + \gamma \cdot \text{sign}(\nabla_{x_{i-1}} \mathcal{L}(x_{i-1}, y))$, and then computes $\tilde{x}_i = \text{clip}(x_i, x_i - \varepsilon, x_i + \varepsilon)$. It proves to be a strong iterative attack, relying on the first order information of the target model. Figure 1b shows some samples we generated using this method.

2.2.3 Basic Iterative Method (BIM)

Kurakin et al. [4] propose an iterative variant of FGSM. This method generates an adversarial sample as $\tilde{x}_i = \text{clip}_\varepsilon(\tilde{x}_{i-1} + \frac{\varepsilon}{i} \cdot \text{sign}(\nabla_{\tilde{x}_{i-1}} \mathcal{L}(\tilde{x}_{i-1}, y)))$, where $\tilde{x}_0$ is the clean image x, i is the iteration number and $\text{clip}_\varepsilon(\cdot)$ is a clipping function to keep x in its domain. Figure 1c shows some samples we generated using this method.

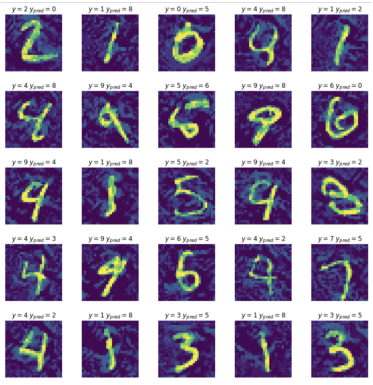
2.2.4 Momentum Iterative Method (MIM)

MIM [1] is a variant of BIM with an additional momentum term introduced to stabilize the direction of the gradient. This method generates an adversarial sample as $\tilde{x}_i = \text{clip}_\varepsilon(x_{i-1} + \frac{\varepsilon}{i} \cdot \text{sign}(g_i))$, where x_0, i and $\text{clip}_\varepsilon(\cdot)$ are the same as mentioned in section 2.2.3 and $g_i = \mu \cdot g_{i-1} + \frac{\nabla_{x_{i-1}} \mathcal{L}(x_{i-1}, y)}{\|\nabla_{x_{i-1}} \mathcal{L}(x_{i-1}, y)\|_1}$, where μ is the decay factor. Figure 1d shows some samples we generated using this method.

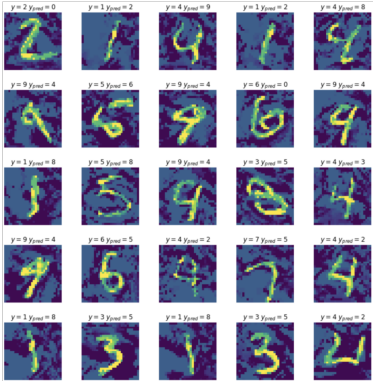


(a) Sample images generated by FGSM

(b) Sample images generated by PGD



(c) Sample images generated by BIM



(d) Sample images generated by MIM

Figure 1: Sample images generated by different attack methods

3 Experiments

3.1 Dataset

We evaluate our approach over MNIST [6], a popular dataset that is widely used for developing both attacks on DNNs and defenses against them. MNIST is a dataset of 28×28 pixel images of digits. The dataset contains 60, 000 training images and 10, 000 testing images. Our task is to classify images of handwritten digits into one of ten classes, each representing a single digit.

3.2 Model

We evaluate our method using a convolutional deep neural network architecture. We use a 6 layer ConvNet similar to [9]. Our model consists of three convolutional blocks with 32, 64 and 128 filters. Each convolutional block, consists of two identical convolutional layers, with ReLu activation function. We use three fully connected layers after the last convolutional layer. The triplet loss function is applied on the second to last layer. Last layer being the softmax. We implemented our model using Tensorflow 2.0, which has built-in functionality to make customizing models easier and a flexible ecosystem of tools and libraries.

Previous work [9, 8] train the model jointly on the Cross-Entropy objective as well as the loss function they introduce. We found this to be impractical on our experimental results. Thus we train our model's triplet loss independently of the softmax cross-entropy loss. We do this by using the `stop_gradients` operation in Tensorflow. During training time only, we prevent the softmax layer to update the weights of the remaining neural network based on the cross-entropy loss.

In practice for larger models that are more difficult to train, we can use a pre-trained model and extract all layers except the last one. We can use the `stop_gradients` operation on the softmax layer and fine-tune the model on the triplet-loss objective with a sufficiently small learning rate as to avoid over-learning.

Block #	1	2	3
Layers	Conv(32, 5×5, ReLu) $\times 2$	Conv(64, 5×5, ReLu) $\times 2$	Conv(64, 5×5, ReLu) $\times 2$
Block #	4	5	6 (softmax layer)
Layers	Dense(512, ReLu)	Dense(64)	Dense(10)

Setup. We tune an existing network for 10 epochs, using a learning rate of 10^{-4} and batch size of 128.

3.3 Adversarial Training

It has been shown that adversarial training can enhance the robustness of DNNs, as recently proposed in some defense methods [5]. We also evaluate the impact of adversarial training in conjunction with our proposed defense. For this, we jointly train our model on clean and attacked samples, which are generated using FGSM (see section 2.2.1) and PGD (see section 2.2.2) by uniformly sampling with an ε value of 0.3.

3.4 Results

The results in Table 1 show that our method significantly outperforms existing schemes by a large margin. The results also indicate that adversarial training further complements our method and provides an enhanced robustness.

Table 1: The robustness results of our models.

	White-Box Setting				
Training	No-Attack	FGSM	BIM	MIM	PGD
			$\varepsilon_{MNIST} = 0.3$		
Softmax	99.2	34.48	0	0	0
Ours	**99.58**	69.45	26.52	44.93	95.49
Ours$_{\textbf{PGD}}$	99.46	80.7	50.08	65.73	**97.23**
Ours$_{\textbf{FGSM}}$	99.5	**82.46**	**50.68**	**66.77**	97.04

Figure 2 shows our latent space with the adversarial examples shown in it (see key inside the figure).

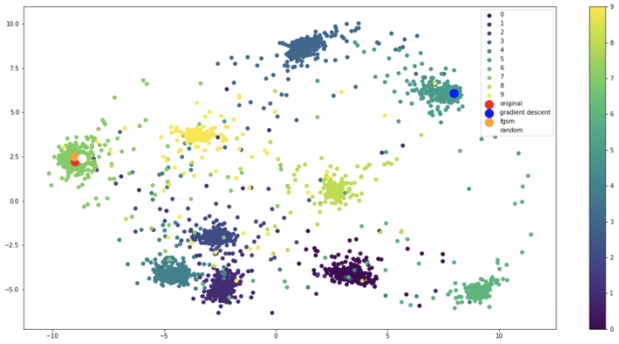


Figure 2: Our model's latent space with the adversarial examples shown

4 Discussion & Future Work

We demonstrated the efficacy of the triplet loss in enhancing adversarial robustness. Our empirical investigation yielded an efficient training technique involving the "splicing" of the gradients between a classification loss and the aforementioned triplet loss.

Observe the layout of the feature space in Figure 2. By means of our objective function, adversaries need to traverse a greater distance in order to make it from one cluster all the way to another. Hindered by the adversarial budget ε that places a constraint on proximity to the original input, large enough adversarial perturbations are rendered unaffordable.

Moving forward, it would be of utmost significance to test our framework on a most difficult data set as considered by the state of the art: namely ImageNet. Decent results in that domain would validate our technique beyond reasonable scrutiny. We are in the process of evaluating the procedure on a ResNet110 model for the CIFAR-10 dataset. The task in this case is to classify images into one of ten object categories: airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Our aspiration is to obtain commensurate results on specifically a ResNet56 model for ImageNet, wherein the task is to classify images into one of a thousand categories including animals, food, tools, scenes, and activities. With regards to both of these applications, the baselines and adversarial algorithms have been implemented already. All that remains is to finalize the models' training.

5 Conclusion

Evidently and as demonstrated, our triplet-loss objective increases both robustness *and* accuracy, a rarity in any adversarial training technique. In order for attacks to be successful in this new regime, the necessary perturbation norm becomes untenable. The proposed approach is simple and may even be applied to existing architecture-objective couplings almost ad hoc.

References

[1] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

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

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

[5] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 195–231. Springer, 2018.

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

[7] 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.

[8] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 478–489, 2019.

[9] Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial defense by restricting the hidden space of deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3385–3394, 2019.

[10] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[11] Andrea Paudice, Luis Muñoz-González, Andras Gyorgy, and Emil C Lupu. Detection of adversarial training examples in poisoning attacks through anomaly detection. *arXiv preprint arXiv:1802.03041*, 2018.

[12] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems*, pages 6103–6113, 2018.

[13] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

[14] Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

[15] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in neural information processing systems*, pages 3517–3529, 2017.

[16] 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.

[17] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples. *arXiv preprint arXiv:1704.03453*, 2017.