Towards Robust Neural Networks

1st Senad Beadini
Dept. of Computer Science
Sapienza University of Rome
Rome, Italy
beadini.1754617@studenti.uniroma1.it

2nd Edoardo U. Proverbio
Dept. Computer Engineering
Sapienza Univiersity of Rome
Rome, Italy
proverbio.1658119@studenti.uniroma1.it

3nd Daniele D'antoni
Dept. Computer Engineering
Sapienza Univiersity of Rome
Rome, Italy
dantoni.1752328@studenti.uniroma1.it

***Abstract*—Deep neural networks have achieved amazing results in several tasks. However, recent works have shown how these models are vulnerable to adversarial examples, which pose questions about their safety in critical applications. In this paper we want to show which are the basic techniques used to generate an attack and the methods to handle them. Then we are going to test empirically how these defenses approach work on a sort of real application like Traffic Signs Classification. Our main goal consists in drawing the picture of the methods regarding model defenses through adversarial training, with a comparison between FGSM, FastFGSM and TRADES showing which one makes the model more robust to PGD attacks. Furthermore, we include in this analysis a new recent approach of adversarial training using contrastive learning, in which a pre-training step has shown to increase neural networks' robustness.**

***Index Terms*—Deep Learning, Traffic Sign Classification, Adversarial Learning, Robust Machine Learning**

I. Introduction

Deep learning models have achieved amazing results (higher than any other approach) in tasks like image classification, object detection and recognition, language translation, voice synthesis and even face recognition [1], [2]. Despite these great achievements, Szegedy et al. [3] found that Deep Neural Networks (DNNs), and in particular classifiers, are vulnerable to small input perturbations. These kind of data can fool easily even a state-of-the-art model; more alarming is the fact that, even though the images are very similar so that differences to the original ones are in most cases imperceptible to the human eye, such models report high confidence in the wrong predictions Fig. 1. These perturbed samples are called adversarial examples. For the consequences that these inputs could have in actual systems, the study of adversarial attacks and robustness of DNNs has become crucial in the research community. In fact nowadays, we have large number of research papers concerning methods to identify adversarial attacks and develop new defenses [4], [5].

In this paper, we want to test experimentally and analyze some methods which has been recently developed in order to make a DNN robust. This work is focused on traffic sign classification systems, a relevant task for several practical applications, like self-driving cars. Indeed, an adversarial attack on traffic sign systems is not acceptable, leading potentially to security problems. Yet, summarizing, our contributions are :

- Briefly discuss the robustness problem for a DNN.
- Testing the robustness of ResNet18 on traffic signs classification.
- Showing the performance of different adversarial training on aforementioned model.
- Testing the supervised contrastive learning as adversarial strategy and proving that a pre-training step using contrastive learning increases robustness even with a small number of epochs.

II. Adversarial Generation

In this section, we are going to introduce the adversarial example generation task, then the formal definition of adversarial training.

Formally, given a trained (deep learning) classifier C and a data sample (image) X we can define the generation of an adversarial example as an optimization problem [9]:

$$\min_{X'} ||X - X'|| \tag{1a}$$

$$\text{s.t.} \quad C(X) = l, \quad C(X') = l', \quad l \neq l'. \tag{1b}$$

where l and l' denote the label of X and X', respectively. We can define $\tau = X' - X$ the perturbation added on the data sample X and write our adversarial example as: $X' = X + \tau$. In other words, the adversarial example is an image with a minimal perturbation that changes the model's prediction. In Fig. 1, there's an example of adversarial example for a state-of-the-art DNN [10].

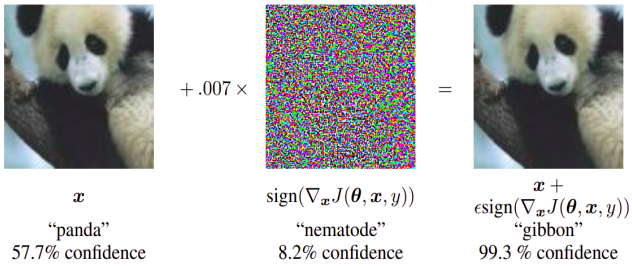


Fig. 1: Adversarial image generated by FGSM algorithm [10]. From the left: clean image, perturbation and adversarial image. As we can see the two images are identical to the human eye.

III. Adversarial Attack algorithms

Here we introduce the algorithms which we are going to use to generate adversarial attacks. The following algorithms "solves" the problem (1a) and its constraints and can produce adversarial examples. All the methods discussed on this paper refer only to adversarial white-box attacks (which means having a *total* access to the model).

A. Fast gradient sign method (FSGM) attack

The simplest way to generate adversarial data has been introduced by Goodfellow et al. in [10]. The algorithms is called fast gradient sign method (FGSM) and it's extremely effective from a computational complexity point of view. It essentially perform a one step gradient update towards the direction of the sign of the gradient at each pixel. The formula (2) below shows how the adversarial example is generated from the original image:

$$X' = X + \epsilon sign(\nabla_x L(\theta, X, l) \tag{2}$$

where ϵ is the magnitude of the perturbation. The higher the latter, the more perturbed will be the adversarial image.

B. Projected gradient descent (PGD) attack

Projected gradient descent (PGD) can be seen as the iterative generalization of FGSM algorithm. It was introduced in [11]. The iterative procedure is shown in formula 3:

$$X_{i+1} = clip_\epsilon(X_i + \alpha sign(\nabla_x(L(\theta, X_i, y))) \tag{3}$$

where *Clip* denotes the function that projects its argument to the surface of x's neighbor ball while α is the step-size. An attack like this one (heuristically) searches the example X' which has the largest loss value in the l_∞ (or l_2) ball around the X original sample. Due to its iterative nature, this algorithm is able to produce aggressive adversarial data that has high probability to fool a model. This algorithm usually is used as a benchmark for neural networks' robustness. The drawback of PGD is its high computational complexity, since for each iteration we need to compute the gradients of the network w.r.t. the adversarial image. This leads a huge cost for high number of steps. An improvement of this algorithms is YOPO that has been proposed in [12]. The advantage of YOPO is that it requires fewer resources than PGD although it produces aggressive adversarial examples such as PGD.

IV. Defense methods

The most common approach to build a robust DNN consists in using adversarial training (AT). The core idea is to train the model not only with training data, but with adversarial examples too [5]. There are a huge number of papers that studies deeply AT [6], [7], [8]. A high level view of how a model is trained using AT is summarized by the algorithm 1. In general, AT methods differ from how steps 1 and 2 are performed. For example, step 1 may differ from what kind of algorithm you use to generate the new batch, as well as step 2 can be strongly different for what kind of loss you use.

Algorithm 1: High level AT for a DNN

Set the network F with randomly weights
repeat
  0) Read mini-batch $B_i = \{x_1, .., x_b\}$
  1) Generate adv. images B_i^* from B_i with current F
  2) Do one training step with F using B_i^*
until *F converges*;

A. FGSM training

FGSM training is the simplest method of AT. It essentially apply the algorithm 1 using FGSM function for generating adversarial images. This method however has been outperformed by more advanced techniques since theoretical results have shown its weaknesses.

B. TRADES

TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization (TRADES) is a recent method published in 2019 [19]. Other AT increases the robustness of a model but this result often comes at the cost of a reduction in the overall accuracy on clean images. TRADES aims to balance adversarial robustness against natural accuracy through a regularized loss to be optimized, formula 4:

$$\min_f E\left\{ L(f(X), Y) + \beta \max_{X' \in B(X,\epsilon)} L(f(X), f(X')) \right\} \tag{4}$$

The first term encourages the natural accuracy to be optimized by minimizing the difference between $f(X)$, prediction on clean image, and Y (true label). The second regularization term pushes the decision boundary of classifier away from the sample instances via minimizing the difference between the prediction of natural example $f(X)$ and the adversarial one $f(X)$. The tuning parameter β balances the importance of natural and robust accuracy.

C. Contrastive Adversarial Learning as Pre-Training

Contrastive learning (CL) is a new learning paradigm which try to cope with self-supervised representation learning. New developments like in [20] propose to face adversarial learning with CL. In particular, [20] introduces how CL adversarial training can be used as a pre-training step for downstream task and possibly increasing model's robustness.

1) Standard Framework of CL: This is the basic implementation of CL, built upon SimCLR [21], with the idea of learning by maximizing agreements of differently augmented views of the same image. This way a sample x is augmented through two different transformation $\tau : (t, t')$ creating a pair $\tilde{x}_i, \tilde{x}_j$ which is processed by network backbone. In the end outputted features are optimized under the CL loss called NT-Xent:

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

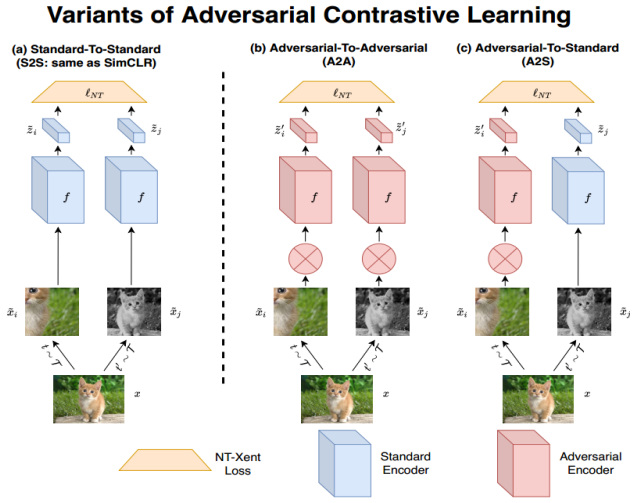
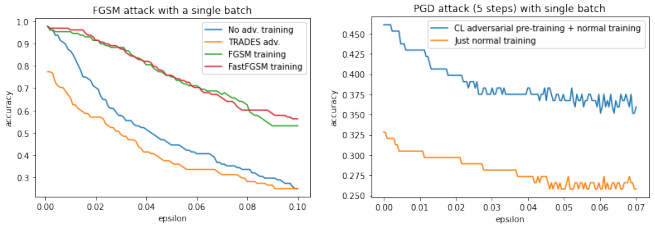


Fig. 2: Contrastive Adversarial Learning, picture from [20].

(a) figure 3 (b) figure 4



(a) figure 5 (b) figure 6

2) Adversarial Framework A2S and A2A: In this framework adversarial attacks is integrated with the CL process previously showed. Adversarial perturbations are generated using PGD or YOPO [12] and corresponding augmented samples are injected into one (Adversarial to Standard), or both (Adversarial to Adversarial) siamese (backbone) network. We have to point out that in every framework each backbone (encoder) shares the weights. In Fig 2, we can see clearly the frameworks' representation. This procedure can be used as a pre-training step and we could use the obtained model's encoder (backbone) as the initialization of a new model for a downstream task.

V. Traffic Signs Classification Analysis

Traffic Signs dataset has been obtained from Kaggle [22]. The dataset contains roughly 73k images (size 32x32) of 43 different classes. We split the dataset in 50k-10k-13k for training, validation and testing. Our experiments present the following points of interests for what concerns the various features and methods:

- Random aggressive augmentations, essential for effective adversarial training.[1]
- ϵ, the strength perturbation, small enough to not kill training, but big enough to provide good adversarial images.

VI. Experiments

ResNet18 (pre-trained on imagenet, PyTorch) has been fine-tuned on our dataset using three AT methods (FGSM, FastFGSM and TRADES). FastFGSM is essentially FGSM AT with a random initialization. FGSM and FastFGSM required just 10 epochs to reach a significant accuracy (clean testset) while TRADES needs a bigger number of epochs, we stopped it at 15 epochs. As optimization algorithm was used Adam with learning rate of 0.0005 and batch size of 128. For TRADES the value of ϵ was $3/255$ and for FGSM and FastFGSM was $8/255$. In Table I the accuracies (on clean testset) we obtained. In figure a[3,4,5] we show the trend accuracy using a stress test attack using different parameters (either different ϵ or number of steps). Remembering that, ϵ usually controls the strength of the perturbation. From the plots it's clear that TRADES allows neural networks being robust for some value of ϵ using PGD attacks (the more aggressive). FGSM or FastFGSM dramatically fail and seems they don't increase robustness. However, they perform well for FGSM attack better than TRADES. The fact that TRADES is less performing is not surprising because it is due to the fact that a long training has not been done (just 80% clean accuracy). However it's interesting that FGSM and FastFGSM training perform well on FGSM type attacks but terribly bad on PGD; this is a phenomenon known as *catastrophic overfitting* [24] and is one of the weaknesses of FGSM AT. Even more interesting is the result we have obtained with the adversarial contrastive learning. By running the pre-training phase, so performing supervised contrastive learning using the A2S framework (Fig 2), we achieved an initialization of the network weights. Subsequently, by carrying out the following normal training (without adversarial training) on the traffic signs dataset, we obtained a greater robustness to PGD attacks than the normal model (without pre-training). In figure a6 we can see the comparison. This result is quite surprising because pre-training has only been done for 10 epochs, but nevertheless the model with pre-training has a higher robustness (almost 10 %) for a wide ϵ range.

TABLE I: Test clean accuracy

Models	Adv. Training	Clean Accuracy
ResNet18	None	96.64 %
ResNet18	FGSM	97.49%
ResNet18	FastFGSM	97.7 %
ResNet18	TRADES	80.5 %

[1] We couldn't use crop augmentation due to the already small size of images

VII. Conclusion Future Work

In this paper, we analyzed the most common approach to tackle the problem of adversarial attacks for neural networks, specifically focusing on traffic signs classification. We showed that AT can be an effective technique to prevent adversarial attacks even though it restricted to the strength of attack's perturbation. We have seen that TRADES ensures robustness against PGD attacks better than all other approaches. In the end, confirming the result in [20], we argue how contrastive learning can be used with adversarial attacks as a pre-training phase to increase performances and robustness, and interestingly we saw the effectiveness of pre-training even for a small number of epochs. Some words about the programming part, the code has been developed from scratch for the training, testing and validation for FGSM. Instead the generation of adversarial examples during training has been done with DeepRobust framework [18]. TRADES and FastFGSM has been done using the aforementioned library while the part of contrastive learning has been implemented from scratch using the losses implemented in PyTorch-metric-learning [23]. AT is currently a hot topic in deep learning. New results and papers are published every month. In this work, we have been limited by computational power using just Google Colab, hence our experimental possibilities were quietly limited. Further investigations of contrastive learning in the scenario of adversarial robustness could be worth it, especially using other types of losses and hyper-parameter tuning.

References

[1] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.

[2] Mehdipour Ghazi, Mostafa, and Hazim Kemal Ekenel. "A comprehensive analysis of deep learning based representation for face recognition." Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2016.

[3] Szegedy, Christian, et al. "Intriguing properties of neural networks." arXiv preprint arXiv:1312.6199 (2013).

[4] Silva, Samuel Henrique, and Peyman Najafirad. "Opportunities and challenges in deep learning adversarial robustness: A survey." arXiv preprint arXiv:2007.00753 (2020).

[5] Xu, Han, et al. "Adversarial attacks and defenses in images, graphs and text: A review." International Journal of Automation and Computing 17.2 (2020): 151-178.

[6] Shafahi, Ali, et al. "Adversarial training for free!." arXiv preprint arXiv:1904.12843 (2019).

[7] Tramèr, Florian, et al. "Ensemble adversarial training: Attacks and defenses." arXiv preprint arXiv:1705.07204 (2017).

[8] Naseer, Muzammal, et al. "A self-supervised approach for adversarial robustness." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[9] Yuan, Xiaoyong, et al. "Adversarial examples: Attacks and defenses for deep learning." IEEE transactions on neural networks and learning systems 30.9 (2019): 2805-2824.

[10] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[11] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial examples in the physical world." (2016).

[12] Zhang, Dinghuai, et al. "You only propagate once: Accelerating adversarial training via maximal principle." arXiv preprint arXiv:1905.00877 (2019).

[13] Bai, Tao, et al. "Recent Advances in Adversarial Training for Adversarial Robustness." arXiv preprint arXiv:2102.01356 (2021).

[14] Bai, Tao, et al. "Recent Advances in Adversarial Training for Adversarial Robustness." arXiv preprint arXiv:2102.01356 (2021).

[15] Jiang, Ziyu, et al. "Robust Pre-Training by Adversarial Contrastive Learning." NeurIPS. 2020.

[16] Ho, Chih-Hui, and Nuno Vasconcelos. "Contrastive learning with adversarial examples." arXiv preprint arXiv:2010.12050 (2020).

[17] Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[18] Li, Yaxin, et al. "Deeprobust: A pytorch library for adversarial attacks and defenses." arXiv preprint arXiv:2005.06149 (2020).

[19] Zhang, Yu, Jiao et al. "Theoretically Principled Trade-off between Robustness and Accuracy" arXiv preprint arXiv:1901.08573v3 (2019).

[20] Jiang, Ziyu, et al. "Robust Pre-Training by Adversarial Contrastive Learning." NeurIPS. 2020.

[21] Chen, Kornblith, Norouzi, Hinton. "A Simple Framework for Contrastive Learning of Visual Representations" arXiv preprint arXiv:2002.05709v3 (2020).

[22] https://www.kaggle.com/flo2607/traffic-signs-classification

[23] Musgrave, Kevin, Serge Belongie, and Ser-Nam Lim. "Pytorch metric learning." arXiv preprint arXiv:2008.09164 (2020).

[24] Kim, Hoki, Woojin Lee, and Jaewook Lee. "Understanding catastrophic overfitting in single-step adversarial training." arXiv preprint arXiv:2010.01799 (2020).