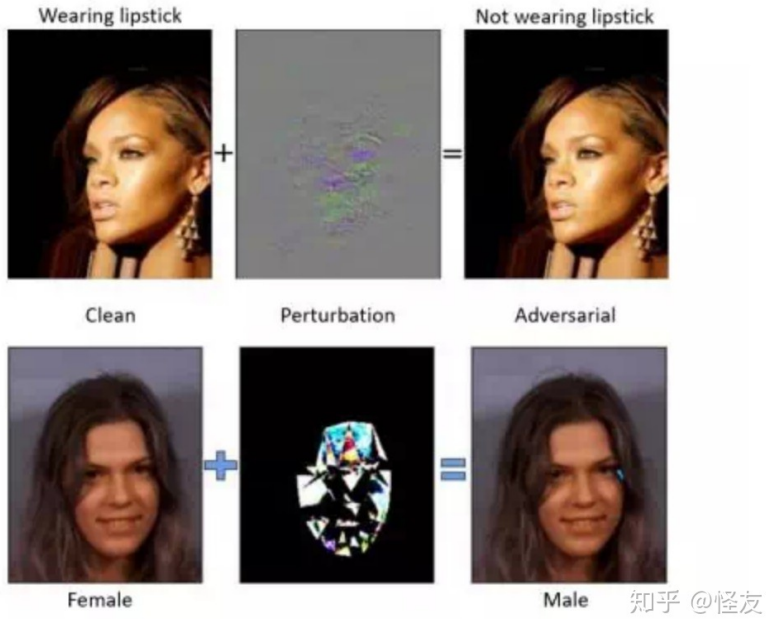
Dropen: A new approach to training robust networks

Zhouyu Zhang
Tsinghua University
工物82
zhangzho18@mails.tsinghua.edu.cn

1 Introduction

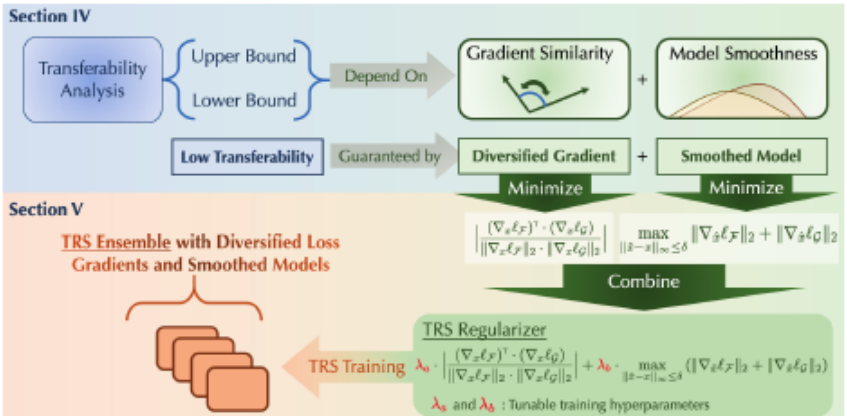
This report is intended to briefly introduce our work on robust neural networks. A robust classifier is one that correctly labels adversarially perturbed images, while adversarial examples are images intentionally perturbed for causing misunderstandings in the intelligent classifier agents. The term *intentionally* means that the adversarial examples are generated based on the idea to **maximize** the classification loss (for example, crossentropy loss) within a variable scope as small as possible. In our experiment settings, we mainly address the situation of untargeted attacks, where maximized loss brought by adversarial examples will misconduct the classifier to go astray from the original ground truth labels. The vulnerability of neural networks towards adversarial attacks can cause catastrophic consequences, since a small change in the input data might be able to induce sharp shifts in the classification outputs.



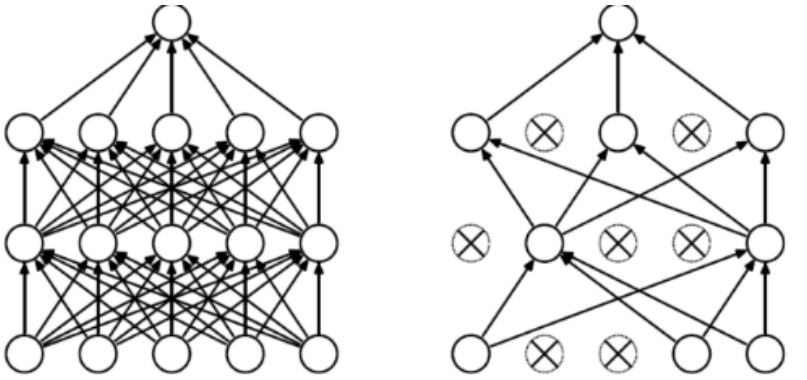
Classical methods to training a robust model involves 'informing' the model about future possible adversarial examples during the training process. The idea is quite simple. For each batch of inputs, first calculate the loss forward with original inputs. Then, based on the calculation graph of model loss, try to maximize the loss within the predefined limited input scope (the perturbation should be limited). The maximization tools fall typically into

PGD (projected gradient descent) or FGSM (fast gradient sign method). Another clue is to use ensembles of models. The idea is based on the belief that different sub-models or weak classifiers will possibly give different feedback upon the same perturbed input, thus rendering better performance on adversarial attacks through a wholistic decision procedure (for example, voting or calculaing the average score). However, recent research has shown that adversarial examples are able to transfer between different models. Intriguingly, though most of the attack strategies require access to the information of target machine learning models (whitebox attacks), it has been found that even without knowledge about the exact target model, adversarial examples generated against another model can transferably attack the target victim model, giving rise to blackbox attacks. This property of adversarial transferability poses great threat to the real-world DNN-based systems since the attacker could succeed even without intrusion to the target ML system. Also, the method of using ensembles for adversarial robustness seems to be doomed, since sub-models can possibly be fooled by the same adversarial example.

In April, Yang et al.[1] proposed a new method named Transferability Reduced Smooth-ensemble (TRS) to limit transferability between base models within an ensemble and therefore improve its robustness. In particular, they enforced the smoothness of models as well as reduced the loss gradient similarity between models to introduce global model orthogonality. The specific method involves imposing a penalty on gradient cosine similarities between sub-models and on model gradient magnitudes.



Inspired by the contribution of Yang et al., we wanted to try the TRS penalty in single model situations. However, their work were based on ensemble models, how should we adapt these methods to a single model? We took the idea of dropout networks.The computing logic of dropout layers make them the perfect counterparts of the final stage for ensemble models, where the scores and results of previous sub-networks were processed or averaged. **A dropout network can be viewed as an ensemble of enormous randomly generated sub-networks** . The methods for ensembles can be intuitively adapted for single models.



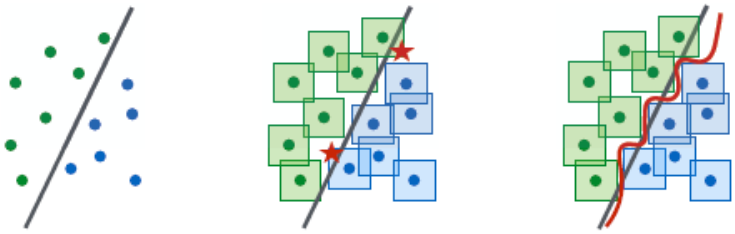
(a) Standard Neural Net (b) After applying dropout.

In our experiment, we successfully implemented an adaption of TRS regularizer for dropout networks and test the robustness performance of our models upon PGD and FGSM adversarial attacks. We name our method Dropen (combination of dropout and ensemble). In terms of adversarial robustness, models trained with Dropen methods outperform models trained with vanilla methods and models not adversarially trained .

2 Related Work

The phenomenon of adversarial attacks was initially discovered by Szegedy et al.[2] in 2013.

In 2019 Shafahi et al. [3]came up with the method of free PGD adversarial training, and Madry et al.[4] in 2017 pointed out that the reason of neural networks performing poorly upon adversarial examples was that the decision boudaries of model were too sharp.



In 2020 Wong et al. [5] refined the work of Shafahi et al. and proposed a method for fast and energy efficient to train robust networks, mainly utilizing the FGSM method.

We got our inspiration from [1]. The work conducted in [6] helped consolidate our belief that there is a link between dropout networks and ensemble models.

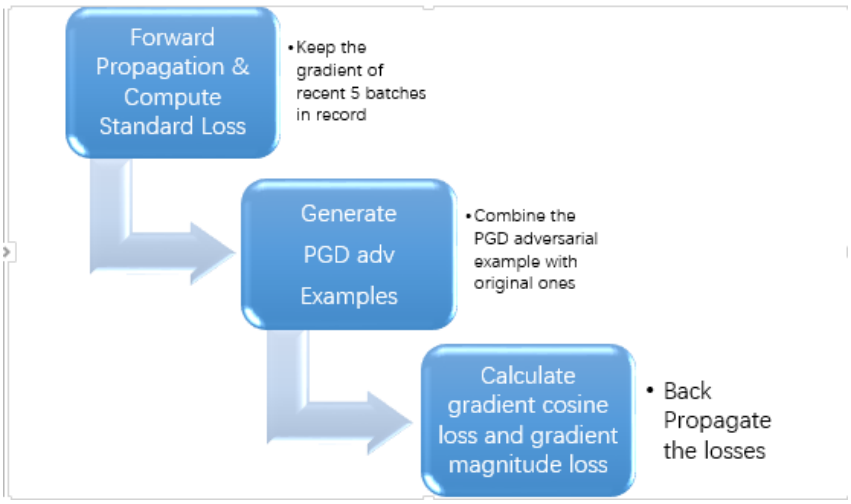
3 Method

Our TRS Dropen method can be summarized into three steps, mainly implemented in dropen_trainer.py. The following list and figure describe how each batch of data is processed for training and updating weight, note that the workflow is repeated for each batch so it may take quite long to train an epoch:

- Calculate forward the standard losses with a batch of original inputs. Record the gradient of previous 5 batches in GPU memory and use them as detached variables to compute cosine similarity with the gradient of the current batch.

- Use projected gradient descent method to generate adversarial examples, calculate again the forward losses with adversarial examples and record the magnitude of gradients for loss calculation that follows.

- Accumulate the cosine similarity losses, model smoothness losses (adversarial gradient magnitude losses) and standard losses in a weighted method. The weight put on each loss can be modified for better performances. Propagate backward the losses to update the model weights.



4 Experment

4.1 Experiment Settings

Datasets used:

- MNIST[1].The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.The database is also widely used for training and testing in the field of machine learning. It was created by "re-mixing" the samples from NIST's original datasets.The creators felt that since NIST's training dataset was taken from American Census Bureau employees, while the testing dataset was taken from American high school students, it was not well-suited for machine learning experiments. Furthermore, the black and white images from NIST were normalized to fit into a 28x28 pixel bounding box and anti-aliased, which introduced grayscale levels.
- CIFAR10[2].The CIFAR-10 dataset (Canadian Institute For Advanced Research) is a collection of images that are commonly used to train machine learning and computer vision algorithms. It is one of the most widely used datasets for machine learning research. The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes.The 10 different classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6,000 images of each class.

For MNIST, we use LeNet with a dropout layer as base model. For CIFAR10, we use a simple two convolution layered model as the base model. The network configurations can be found in dropen_models.py. We will compare TRS Dropen trained base models with PGD adversarially trained ones. Base models not adversarially trained will also be introduced as an illustration of how vulnerable models can be to adversarial attacks.

The hyper-parameters can be easily found in the default argparser of dropen_train.py.

4.2 Experiment Results

The higher the perturbation radius goes, the fiercer the attack will be. Large perturbation radius means that the adversarial example goes quite far in the direction of maximizing losses, thus resulting in accuracy decrease. We gradually increase the perturbation radius and keep record of the accuracy descent. The attack form is PGD50 (50 epochs of gradient descent) with adjustable perturbation radius.

[1]http://yann.lecun.com/exdb/mnist/
[2]https://www.cs.toronto.edu/ kriz/cifar.html

	MNIST	CIFAR10
PGD Training	75.39%	74.89%
Direct Training	80.27%	86.91%
TRS Training	**62.9%**	**65.36%**

Table 1: Accuracy descent when CIFAR10 perturbation radius eps = 0.02, MNIST pertubation radius eps = 0.7

	MNIST	CIFAR10
PGD Training	84.55%	85.89%
Direct Training	88.02%	93.22%
TRS Training	**74.73%**	**72.9%**

Table 2: Accuracy descent when perturbation radius eps = 0.025, MNIST pertubation radius eps = 0.75

The TRS Dropen trained base models sufferred less accuracy loss when facing adversarial attacks with the same strength

5 Conclusion

Our work on TRS Dropen trained models brought about a new training strategy for obtaining robust neural networks whose performance surpassed neural networks trained with vanilla PGD adversary. Due to time limitations, we did not compare TRS method with more recent and advanced robust training methods. And also due to GPU memeory limits, we were only able to maintain 5 batches for gradient cosine similarity calculation, in order to control the training time in a reasonable amount. Still, the performance of TRS Dropen model proves to be promising and worth more intensive investigation. If possible, we will conduct experiments with bigger models and different loss weights. We believe these will result in better performances and more pervasive methods.

6.Refs

References

[1] Yang, Z., L. Li, X. Xu, et al. Trs: Transferability reduced ensemble via encouraging gradient diversity and model smoothness. *arXiv preprint arXiv:2104.00671*, 2021.

[2] Szegedy, C., W. Zaremba, I. Sutskever, et al. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[3] Shafahi, A., M. Najibi, A. Ghiasi, et al. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

[4] Madry, A., A. Makelov, L. Schmidt, et al. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[5] Wong, E., L. Rice, J. Z. Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[6] Hara, K., D. Saitoh, H. Shouno. Analysis of dropout learning regarded as ensemble learning. *CoRR*, abs/1706.06859, 2017.