Improving Adversarial Transferability by Combining Momentum and Linear Back-propagation

Aanal Sonara 190070064
Hitvarth Diwanji 190100057
Krishnakant Bhatt 21q050016
Ashish Vinod Lalwaney 213050016

Abstract

Deep Learning has received huge attention in the past few decades with the applications in almost all domains. However, deep learning models are vulnerable to adversarial attacks which raises security and reliability concerns. Making neural networks more robust to black-box attacks involves coming up with more transferable adversarial examples. Here we explore various methods to improve transferability of adversarial examples and try to combine them to further improve the transferability as well as the success rate. We conduct experiments on MNIST and CIFAR-10 using a mix of official and self-written code.

1 Introduction

The field of Deep Learning is almost everywhere now and is enabling us to perform a variety of tasks. Deep Neural Nets are now used in complex applications like self driving cars, fraud detection, and many more. However, it has been shown that we can skew the predictions of a model by creating adversarial examples (adding noise to data using adversarial networks) which can not only drop the accuracy of these models, but can also lead to incorrect predictions with high confidence. This vulnerability of deep learning models towards adversarial examples is an active field of research as it not only allows us to make the models more robust but also helps us understand the working of deep networks better. The performance of a model on adversarial examples is a good metric to evaluate the robustness of a model. However, adversarial attacks for black box models are not fully-developed. Recent works in this area include momentum based and linear backpropagation based strategies, both of which solves the above mentioned problem quite effectively. Hence, we propose to combine momentum and linear backpropagation based iterative algorithm to boost adversarial attacks. We then also include ensemble based ideas to generate adversarial examples. Finally we showcase the effectiveness of our proposed method by conducting experiments on real datasets. Furthermore, we intend to design adversarial-training based defense method to defend against our proposed attack method.

2 Previous Work

Momentum based gradient update methods have been used to boost the transferability of black box adversarial attacks, (6). Here, the momentum term is added to stabilise and steer the update directions from local maxima. They have also reported that using an ensemble of models improves the transferability of adversarial examples. The authors reported that iterative methods performed better one-step gradient counterparts on black-box attacks.

Linear backpropagation has been reported to improve the transferability of adversarial examples (5). For a given model f, we use a more linear source model f'. This source model is used to calculate the gradients for black-box attacks. The authors proposed a method of ignoring the non-linear activation function (often ReLU) in gradient calculation since non-linearity arises from activation

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

functions. We calculate forward pass normally but for backward pass ignore the activation function. This representation covers both perceptron and convolutional networks. However, linearization can decrease the source model's performance because we are directly passing ReLU. Therefore, authors have proposed a method to calculate trade-off between linearity and accuracy. They use a function $z_h(x)$ = h(x) such that we are giving more weight to gradients on linear side and we are ignoring (filtering out) gradients on negative side as follows,

$$\tilde{\nabla}_x = \frac{dL(x,y)}{dz_g} W_d ... W_k \frac{dz_h}{dx} \tag{1}$$

The paper by Goodfellow et al (7) on Ensemble Adversarial Training discusses that single step adversarial training converges to a degenerate global minimum, and proposes an ensemble technique which that augments training data with perturbations transferred from other models. The paper then proposes an Adversarial Training Framework explained below.
For some target model $h \in \mathcal{H}$ and inputs (x, y_{true}) the adversary's goal is to find an adversarial example x_{adv} such that x_{adv} and x are "close" yet the model misclassifies x_{adv}. It distinguishes between white-box adversaries that have access to the target model's parameters (i.e., h), and black-box adversaries with only partial information about the model's inner workings. For Training, the following objective function is minimized using following equation (7)

$$h^* = argmin_{h \in \mathcal{H}} \; E_{(x, y_{\text{true}}) \sim \mathcal{D}} [\; max_{||x^{\text{adv}} - x||_\infty \leq e} \; L(h(x^{\text{adv}}), y_{true})] \tag{2}$$

The paper describes the three algorithms to generate adversarial examples with bounded l_∞ norm:

- Fast Gradient Sign Method (FGSM)
- Single-Step Least-Likely Class Method (Step-LL)
- Iterative Attack (I-FGSM or Iter-LL)

3 Background

Attack models add perturbation or noise to the given data which leads incorrect prediction by the model. The perturbations are created based on the level of access of the victim model. Thus, attack models are broadly classified in two categories; white-box attacks and black-box attacks. (1). Given an example x and a classifier f:x $\longrightarrow$ y, we create an adversarial example x^* in vicinity of x such that f gives wrong prediction. The adversarial example are of two types; targeted and non-targeted attacks. In targeted examples, the classifier is misled to classify it as f(x^*) = y^*. Whereas, in non-targeted examples, it is mis-classified as f(x^*) $\neq$ y.

White-Box attack: White-box attack is attempted when full access to the model parameters and architecture is given. Generally, the idea is to maximise the loss with a constraint on perturbation. In a linear perspective, FGSM (2) for a given example x $\in$ R^n, it calculates perturbation as $\epsilon sgn(\Delta_x L(x, y))$ with constrain on l_p norm for p=∞. However, this is a crude approximation and is prone to failure on small epsilon values. An improvement on this was an iterative method such as I-FGSM which has reported more powerful attack performance. Thus it confirmed that iterative methods are more suitable for attacks in white-box scenario.

Black-Box attack: Unlike the white-box setting, the maximum information black-box models have on victim models is the prediction confidence. Black-box attacks are classified in two types; query-based and transfer-based. Transfer based models create adversarial examples on some source model and attempt to attack victim (or target) models. Efforts are being made to improve transferability and performance of multi-step attacks for black box setting. Liu *et al* (3) proposed a method to improve transferability by using an ensemble of source models. In our work, we address the transfer-based attacks.

3.1 Various attack methods

There are following possible techniques in which existing approaches can be divided.

Single step approaches: This method specifies a one time perturbation in the sample by using the gradients. One such method is the fast gradient sign method (FGSM). FGSM aims to generate the adversarial example by perturbating the sample using the gradient so as to increase the overall loss $L(x^*, y)$. Here L can be the cross entropy loss.

$$x^* = x + \epsilon.sign(\nabla_x L(x, y)) \tag{3}$$

Here $\nabla_x L(x, y)$ is the gradient of the loss wrt x. The adversarial example is generated conditioned on meeting the L_∞ constraint $||x^* - x||_\infty \leq \epsilon$.

Iterative methods: These methods iteratively perturbs the given sample. Iterative fast gradient sign method (I-FGSM) is an iterative method which applies the FGSM multiple times. The I-FGSM method can be expressed as follows.

$$x^*_{t+1} = x^*_t + \alpha.sign(\nabla_x L(x^*_t, y)) \tag{4}$$

Here $x^*_0 = x$ and $\alpha = \frac{\epsilon}{T}$, and t goes from 0 to T, where T is the number of iterations. It is done so as to adhere to the L_∞ constraint.

Momentum based methods: These methods makes an advancement to the I-FGSM technique by accumulating the velocity vector in the direction of the gradient. This helps to get rid of poor local minima or maxima. The momentum iterative fast gradient sign method(MI-FGSM) is one such method. MI-FGSM works as follows.

Repeat 4 and 5 from $t = 0$ to $T - 1$:

$$g_{t+1} = \mu.g_t + \frac{\nabla_x L(x^*_t, y)}{||\nabla_x L(x^*_t, y)||_1} \tag{5}$$

$$x^*_{t+1} = x^*_t + \alpha.sign(g_{t+1}) \tag{6}$$

Here x is a real example, y is the corresponding ground truth, L is the loss function, ϵ is the size of perturbation, T is the number of iterations, μ is the decay factor. Here also the adversarial example is generated conditioned on meeting the L_∞ constraint $||x^* - x||_\infty \leq \epsilon$.

4 Our approach

We have used a linear backpropagation to calculate the gradients and MI-FGSM approach to update the gradients on an ensemble of models. Thus for m number of models we define g_{t+1} as

$$g_{t+1} = \sum_{i=1}^{m} \mu.g^i_t + \frac{\nabla_x L((x^*_t)^i, y)}{||\nabla_x L((x^*_t)^i, y)||_1} \tag{7}$$

And then use (6) to update the adversarial example for T iterations.

5 Datasets and Code

The dataset used for training and testing is MNIST (4). MNIST is a small dataset of handwritten digits 0 to 9. It has training examples of 60k and test examples of 10k. We trained our models on google colab and therefore used MNIST dataset. It requires minimal pre-processing and thus was a good fit based on our available resources.
Cifar10 dataset contains 10 classes and 6000 images per class. They are of size 32x32. There are 50k training images and 10k testing images. It has the same size as MNIST and classes are very different from one another (8).
The authors of (5) provided their official code here. We implemented their backpropagation

6 Experiments and Results

We trained three models on mnist dataset for 20 epochs each. The training and validation accuracy are given in 4. Using ensemble models 1, 2 and 3, we attack our victim model. We keep the victim model to be stronger than our source models. It can be seen from 4 that the victim model has been trained to lower loss values.

Table 1 enlists the success rates for different models when attacked using methods MI-FGSM, LinBP, MI-FGSM + LinBP respectively with cifar10 dataset. The results clearly shows the improvement achieved by the combination of MI-FGSM and LinBP over the individual methods as reported in (5).

	MIFGSM	LinBP	LinBP+MIFGSM
vgg19	0.9992	1.0000	**1.0000**
WRN	0.7804	0.9417	**0.9478**
ResNeXT	7813	0.9468	**0.9530**
DenseNet	0.7524	0.9306	**0.9391**
pyramidnet	0.2806	**0.4950**	0.4917
gdas	0.6355	0.8217	**0.8301**

Table 1: Success rates for different models using MIFGSM, LinBP, LinBP+MIFGSM attack methods on CIFAR-10 dataset

The value of the perturbation hyper-parameter affect the success rate of attacks. We performed experiments with different values of epsilon to observe the effect of epsilon on success rate. 1 represents the change in success rate for different model architectures with the change in epsilon on CIFAR-10. We evaluated the approach of (5) along with momentum (MI-FGSM) on mnist dataset.

Figure 2 shows success rate vs epsilon for our method on MNIST. We observe that beyond certain epsilon, we have 100% success in fooling the victim model

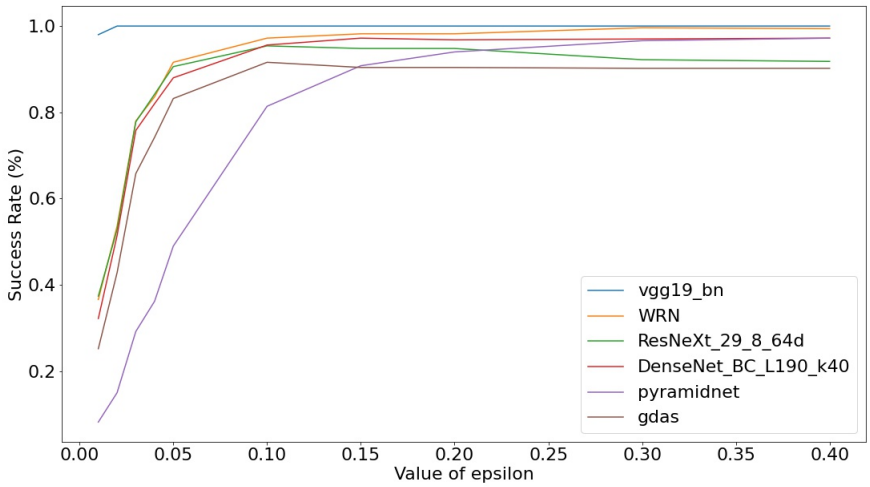


Figure 1: The variation in the success rates of the adversarial examples for the models vgg19, WRN, ResNet, DenseNet, pyramidnet, gdas with the variation in the value of ϵ.

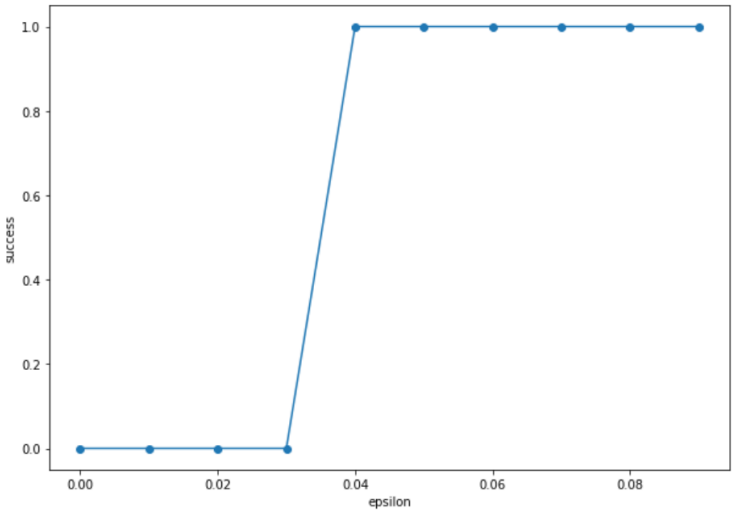
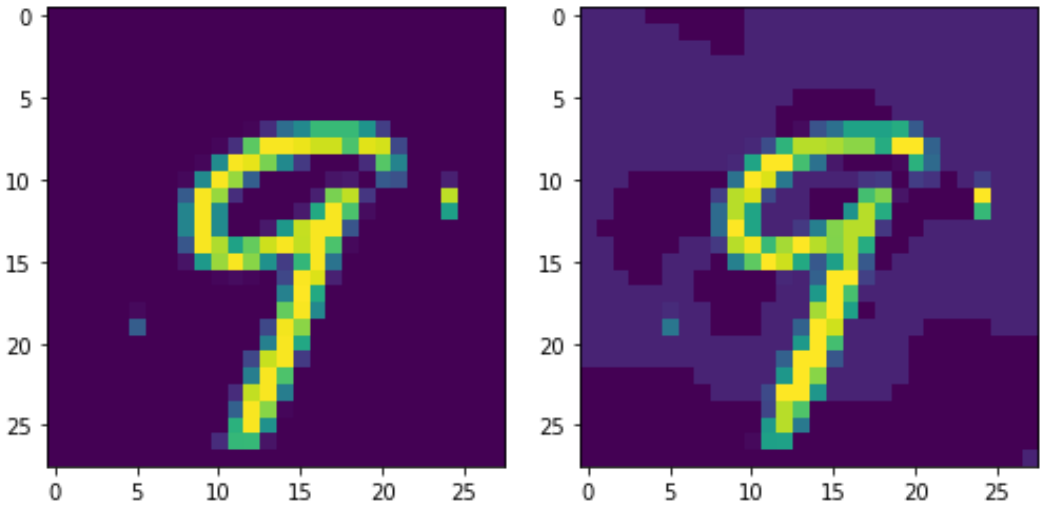


Figure 2: success rate vs epsilon for Our method on MNIST



(a) Original Image (b) Adversarial Image

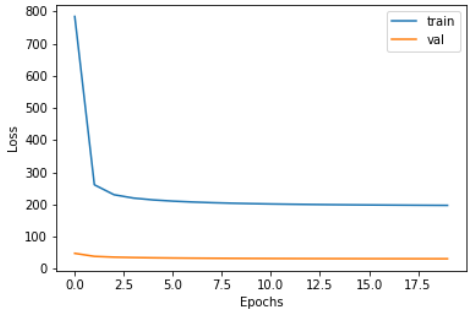
Figure 3: An example of adversarial sample generated using our method. Here $\epsilon = 0.1$ and the method was run for 1000 iterations

7 Conclusion

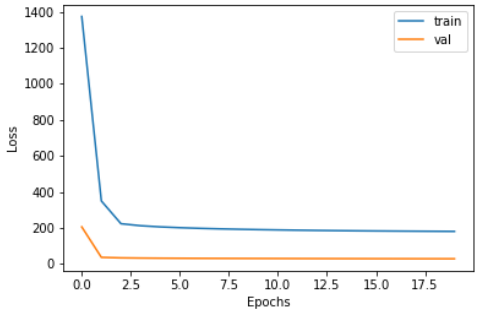
In this work, we have presented an approach that uses an ensemble of models and combines Linear Backpropagation and Momentum updates to attack a black box model. Not only our approach was successful in fooling black box models, experimental results show its effectiveness against its parent methods. Thus, we can use this approach to extensively scrutinize our models and evaluate it's robustness to adversarial attacks.
We can use these stronger attacks to train our models and make them more robust. As a direction of future work, we can combine the loss of adversarial example (7) and normal example & train our models.

References

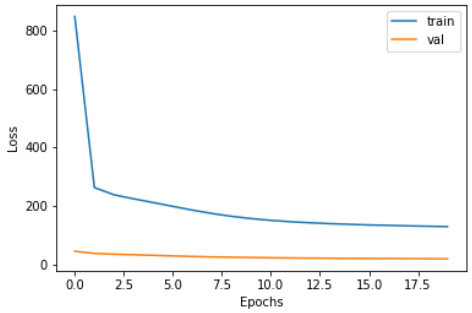
[1] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Asia CCS, 2017.



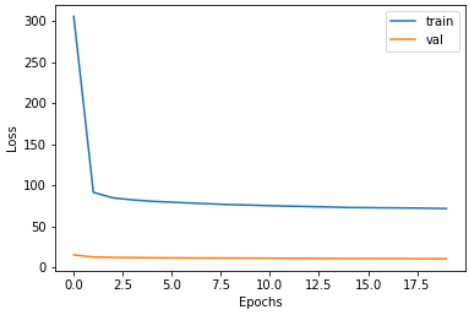
(a) *Validation loss for model 1 of ensemble models*



(b) *Validation loss of model 2 of ensemble models*



(c) *Validation loss of model 3 of ensemble models*



(d) *Validation loss of victim model*

Figure 4: *Validation loss of our models*

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In ICLR, 2015.

[3] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In ICLR, 2017.

[4] Deng, L., 2012. The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6), pp.141–142.

[5] Guo, Y., Li, Q. and Chen, H., 2020. Backpropagating linearly improves transferability of adversarial examples. Advances in Neural Information Processing Systems, 33, pp.85-95.

[6] Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X. and Li, J., 2018. Boosting adversarial attacks with momentum. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9185-9193).

[7] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D. and McDaniel, P., 2017. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204.

[8] Alex Krizhevsky, Vinod Nair and Geoffrey Hinton., CIFAR-10 (Canadian Institute for Advanced Research)

Appendix A: Test 3

Appendix B: Test 3