Attacking Vision Transformer Based on Shampoo Optimizer

Zihao DONG

June 2022

Abstract

Vision Transformers (ViT) have been demonstrated to be yielding state-of-the-art performances in a lot of machine learning tasks, including image classification. However, studies have shown that Vision Transformers can be vulnerable to adversarial examples. In previous work, we have seen lots of adversarial attack methods, but yet a few of them successfully exploited the patch-like characteristic to attack Vision Transformer models. In this work, inspired by the Shampoo Optimizer, which views the network weights as a tensor and add structure-aware preconditioning matrices, each along a single dimension, to aid the optimization step, I will develop an attack method that exploits the patch-like characteristic of the Vision Transformer models. In the experiments, the new attack method's performance is tested on both clean trained Vision Transformer and robust trained Vision Transformer, and the performance is compared to the Projected Gradient Descent (PGD) as a baseline. The results demonstrate that the new attack method is able to achieve comparable performance to the PGD baseline.

1 Introduction

After self-attention based models (Transformers) demonstrated great success in Natural Language Processcing (NLP), lots of studies showed their state-of-the-art performances on computer vision tasks, for example, image classification and bounding box generation (Dosovitskiy et al. 2020). When pretrained on datasets like ImageNet-21K and then finetuned on a smaller dataset like CIFAR-10 and CIFAR-100, Vision Transformers (ViT) are able to achieve state-of-the-art performance while requiring substantially less computational power comparing to traditional convolution neural networks (CNN).

In previous studies, the fact that CNNs are vulnerable to adversarial attacks, the intentional manipulation of input data (image), such that the change is unrecognizable for human eyes, to perturb the prediction label of machine learning models, has been well documented (Goodfellow, Shlens, and Szegedy 2014; Madry et al. 2017). Similarly, it has been shown that Vision Transformers are just as vulnerable to adversarial attacks as CNNs (K. Mahmood, R. Mahmood, and Van Dijk 2021). Attacks designed for CNNs, for example, PGD and FGSM, are also demonstrated to be able to successfully attack Vision Transformers with a high success rates. More recently, attacks that are specifically designed for Vision Transformers are also developed. For example, Zhipeng et.al proposed the PNA that skips the backpropagation of self-attention module to generate adversarial examples, and Ameya et.al introduced a single token (patch) attack on Vision Transformers (Joshi, Jagatap, and Hegde 2021; Wei et al. 2021). However, most works do not explicitly utilize the patch-like property of Vision Transformers, i.e. the fact that ViTs process the images as patches. Inspired by the Shampoo Optimizer, which add preconditioning matrices to each layer of the weight matrix of a machine learning model to update the parameters, we abstract each image as a 3D-tensor made up of the vertical stacking of all image patches, and explicitly add preconditioning to each patch, and generate adversarial examples in a multi-step fashion (Gupta, Koren, and Singer 2018). In this way we explicitly try to utilize the patch-like processing fashion of the ViTs. We compare the performance of Shampoo Attack to a PGD baseline on a Vision Transformer pretrained on ImageNet and finetuned on CIFAR1-10, and conclude that although the new attack method does not outperform the PGD baseline, it is able to achieve results close to the PGD baseline.

2 Related Work and Background

2.1 Vision Transformer

Transformers have demonstrated great success in the Natural Language Processing field because of its unique self-attention based architecture, which is more effective on learning correlations between tokens. Leveraging this characteristic, (Dosovitskiy et al. 2020) introduced Transformer to vision tasks by dividing images into patches. They demonstrated ViT's near state-of-the-art performance on classification tasks when pretrained on larger datasets like ImageNet-21K and the finetuned on smaller datasets like CIFAR10 and CIFAR100. Further studies improved upon their work. For example, DeiT was introduced to try to overcome the necessity of pretraining on large datasets (Touvron et al. 2021).

2.2 Adversarial Attack

Adversarial examples are images that are carefully manipulated that are able to change the prediction of a neural network, and usually human eyes are incapable of telling the difference between the original image and its adversarial counterpart. Current attacking methods can be classified as White-box attack and Black-box attack. In White-box Attacks, the adversarial agent has knowledge to all necessary information of the target model, including its output logits, loss, structure, and gradient. In this setting we usually direct formulate the attack as an optimization process that updates the image toward the direction that maximizes the adversarial loss and minimizes the distance to the original image. Famous White-box attack methods include FGSM, PGD, etc. (Goodfellow, Shlens, and Szegedy 2014; Madry et al. 2017). In Black-Box attacks, the problem is usually solved by finding a direction to perturb and then perform binary search to find the decision boundary of the model along the update direction in the data space, approximating the gradient of the model, through leveraging the transferability of perturbation, etc. (Brendel, Rauber, and Bethge 2017; Chen et al. 2017; Ilyas et al. 2018; Alzantot et al. 2019).

3 Methodology

3.1 Shampoo Optimizer Matrix Form

The Shampoo Optimizer seeks to mitigate the high dimensional computation costs of other preconditioned gradient methods by viewing the network weights as a tensor and keeping preconditioning matrices for a tensor, each along a single dimension. In this report I will focus on a special case of Shampoo Optimizer in the two dimensional case because it better aligns with the nature of images. In two dimensional setting, the weights of the model we are trying to optimize can be seen as a matrix $W \in \mathbb{R}^{m \times n}$. Unlike earlier attempts of preconditioning that flattens the weights into an mn-dimensional vector and construct a $mn \times mn$-dimensional precondition matrix, Shampoo keeps two separate preconditioning matrices $L_t \in \mathbb{R}^{m \times m}$ and $R_t \in \mathbb{R}^{n \times n}$. In each iteration, these two preconditioning matrices are updated using the gradient of the model $G_t = \nabla f_t(W_t), G_t \in \mathbb{R}^{m\times n}$ and $f_t : \mathbb{R}^{m\times n} \rightarrow \mathbb{R}$ is the loss function. The algorithm pseudocode is shown below in Algorithm [1].

Algorithm 1 Shampoo Optimizer Matrix Case

```
$W_1 \leftarrow 0_{m\times n}; L_0 \leftarrow eI_m; R_0 \leftarrow eI_n$
for $t = 1, ..., T$ do
    Receive Loss $f_t$
    Compute Gradient $G_t = \nabla f_t(W_t)$
    Update Preconditioners:
        $L_t = L_{t-1} + G_t {G_t}^T$
        $R_t = R_{t-1} + {G_t}^T G_t$
    Update Step:
        $W_{t+1} = W_t - \eta {L_t}^{-\frac{1}{4}} G_t {R_t}^{-\frac{1}{4}}$
end for
```

3.2 Shampoo Attack on Vision Transformer

To apply idea of Shampoo Optimizer to the attack of Vision Transformers, we view the input image as a 3D tensor $x \in \mathbb{R}^{pp\times s\times s}$ where $p = \frac{H}{S}$ and $H = W$ is the side length of the input image and S is the side length of each individual patch. For each patch, we regard it as a 2D matrix of "weights", and apply a modified version of Algorithm [1] to this weight matrix. In this work we consider the Untargeted Attack, and the adversarial loss for target model $f(x)$ is defined as:

$$\ell_{adv} = -\ell_{CE}(f(x), y) \tag{1}$$

where y is the ground truth label of the image. Like in PGD, we add perturbations to the original image in multiple steps (by default the algorithm updates the image for 30 iterations). Let $x^i \in \mathbb{R}^{S\times S}$ denote the i-th patch, η denote the learning rate, and $\Delta \in \mathbb{R}^{S\times S}$ the perturbation added to the patch. The resulting algorithm pseudocode is shown in Algorithm [2]. Note that in Algorithm [2] we update the adversarial images using the $sign(\cdot)$ of the preconditioning matrices' products with Gradient in order to deal with the vanishing gradient. In other words, the product of the three matrices may be too small to make actual impact on the original image, therefore we take the $sign(\cdot)$ of the product. In practice, we will not iteratively go through all patches. Instead the calculation is wrapped up in a series of simplified torch tensor operations on the GPU.

Algorithm 2 Shampoo Attack

```
for $i = 0, ..., p \times p$ do
    $\Delta^i \leftarrow U(-\epsilon, \epsilon); x_1^i \leftarrow x^i + \Delta^i; L_0^i \leftarrow eI_m; R_0^i \leftarrow eI_n$
    for $t = 1, ..., T$ do
        Receive Loss $\ell_{adv,t}^i(f(x_t^i), y^i)$
        Compute Gradient $G_t^i = \nabla \ell_{adv,t}^i(f(x_t^i), y^i)$
        Update Preconditioners:
            $L_t^i = L_{t-1}^i + G_t^i {G_t^i}^T$
            $R_t^i = R_{t-1}^i + {G_t^i}^T G_t^i$
        Update Step:
            $x_t^i = x_{t-1}^i - \eta \cdot sign({L_t^i}^{-\frac{1}{4}} G_t^i {R_t^i}^{-\frac{1}{4}})$
    end for
end for
```

4 Experiments

To see the effect of Shampoo Attack on Vision Transformers, we pre-trained a Vit Base Model with image size 224 and patch size 16 on ImageNet-21K, and then finetuned the model using CIFAR10. The experiments here will be conducted on CIFAR10 dataset

4.1 Attacking Clean Vision Transformer

Before comparing the performance of Shampoo to PGD baseline, we have to empirically determine the sets of learning rates that work the best under different ϵ's. To find the best learning rate, we run 30 steps Shampoo under each ϵ with various learning rates, and find the learning rate that is able to give the highest attack success rate and at the meanwhile all perturbations close to $\pm\epsilon$.

Learning Rate	Success Rate	%0	% $\pm\epsilon$
0.25/255	0.975	**1.6**	**70**
0.5/255	**0.989**	**1.6**	**70**
0.75/255	0.984	3	65
1/255	0.980	>5	65
1.25/255	0.977	>5	65
1.5/255	0.971	2	60
2/255	0.949	30	70

Table 1: Find Optimal lr for $\epsilon = 2/255$

Learning Rate	Success Rate	%0	% $\pm\epsilon$
0.75/255	0.999	<10	<30
1/255	0.999	10	33
1.25/255	**1.000**	7	35
1.5/255	**1.000**	**1.6**	45
2/255	**1.000**	11	45
3/255	0.994	**1**	50
4/255	0.983	33.3	66.7

Table 2: Find Optimal lr for $\epsilon = 4/255$

Through experiments, we determine the best learning rates with corresponding ϵ would be (1): $lr = 0.5/255, \epsilon = 2/255$, (2) $lr = 1.5/255, \epsilon = 4/255$. Under these different sets of parameters, we compare the performance of Shampoo with PGD baseline using varying number of attack steps. We mainly pay attention to the success rate of the attack and average computation time of the attack.

Steps	Shampoo SR	Shampoo Avg. Time(s)	PGD SR	PGD Avg. Time(s)
10	**0.935**	**1.1**	0.933	1.2
20	0.978	2.1	**0.981**	**1.8**
30	**0.989**	3.1	0.985	**2.5**

Table 3: Performance Comparison with PGD with ϵ=2/255,lr=0.5/255

Steps	Shampoo SR	Shampoo Avg. Time(s)	PGD SR	PGD Avg. Time(s)
10	**0.997**	**1.1**	0.996	1.2
20	**1.0**	2.1	0.999	**1.8**
30	**1.0**	3.1	**1.0**	**2.5**

Table 4: Performance Comparison with PGD with ϵ=4/255,lr=1.5/255

From the comparison we can see that Shampoo SR is able to achieve similar or even better success rate than PGD baseline under same ϵ and learning rate, but with a small margin. Because we use lots of blocked matrix operation, singular value decomposition, and reshaping, the expected attack time for Shampoo is slightly longer than PGD. To more thoroughly test the performance of Shampoo, we further analyze Shampoo for attacking robust trained Vision Transformer

4.2 Attacking Robust Trained Vision Transformer

In this experiment we still focus on the CIFAR-10 dataset, but the target model is a robust trained Vit Base Model using code from Shao et.al (Shao et al. 2021). Shao et.al's paper analyzed the robustness of ViT for image size 32 and patch size 4 for better comparison with its counterparts, so in this experiment we will show Shampoo on 32p4 Vit and 224p16 Vit. Notice that the robust trained model intrinsically has a lower prediction accuracy so when calculating success rate we will exclude the originally wrongly predicted instances. Although Shampoo demonstrated comparable performance to PGD, it has lower success rate than PGD under all parameters we tested on, but for Vit with image size 224 and patch size 16 the performances are close numerically.

Steps	ϵ	learning rate	Shampoo SR	PGD SR
30	2	0.5	6.458	**7.451**
30	4	1.5	14.302	**16.855**
30	8	2	34.184	**40.365**

Table 5: Performance Comparison with PGD for Robust Train ViT 32p4

Steps	ϵ	learning rate	Shampoo SR	PGD SR
30	2	0.5	7.923	**8.234**
30	4	1.5	17.793	**19.487**
30	8	2	38.655	**38.778**

Table 6: Performance Comparison with PGD for Robust Train ViT 224p16

5 Conclusion

In this work, we presented the Shampoo Attack on ViT that exploits the patch feature to construct preconditioning matrices, and view each image we attack as a stack of patches and thus abstracted as a 3D tensor of "weights" we want to update in Shampoo Optimizer Matrix Case (Gupta, Koren, and Singer 2018). The model demonstrated close performance to PGD baseline when attacking a clean Vit model, and slightly worse performance when attacking robust trained ViT. Future work may extend this attack to better understand why Shampoo performs poorly on robust trained models, do experiments on the transferability of Shampoo Attack, or exploit Shampoo Optimizer Tensor Case for attack.

Reference

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

Brendel, Wieland, Jonas Rauber, and Matthias Bethge (2017). "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models". In: *arXiv preprint arXiv:1712.04248*.

Chen, Pin-Yu et al. (2017). "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models". In: *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 15–26.

Madry, Aleksander et al. (2017). "Towards deep learning models resistant to adversarial attacks". In: *arXiv preprint arXiv:1706.06083*.

Gupta, Vineet, Tomer Koren, and Yoram Singer (2018). "Shampoo: Preconditioned stochastic tensor optimization". In: *International Conference on Machine Learning*. PMLR, pp. 1842–1850.

Ilyas, Andrew et al. (2018). "Black-box adversarial attacks with limited queries and information". In: *International Conference on Machine Learning*. PMLR, pp. 2137–2146.

Alzantot, Moustafa et al. (2019). "Genattack: Practical black-box attacks with gradient-free optimization". In: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1111–1119.

Dosovitskiy, Alexey et al. (2020). "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929*.

Joshi, Ameya, Gauri Jagatap, and Chinmay Hegde (2021). "Adversarial token attacks on vision transformers". In: *arXiv preprint arXiv:2110.04337*.

Mahmood, Kaleel, Rigel Mahmood, and Marten Van Dijk (2021). "On the robustness of vision transformers to adversarial examples". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7838–7847.

Shao, Rulin et al. (2021). "On the adversarial robustness of visual transformers". In: *arXiv e-prints*, arXiv–2103.

Touvron, Hugo et al. (2021). "Training data-efficient image transformers & distillation through attention". In: *International Conference on Machine Learning*. PMLR, pp. 10347–10357.

Wei, Zhipeng et al. (2021). "Towards transferable adversarial attacks on vision transformers". In: *arXiv preprint arXiv:2109.04176*.