AVT:Attack on Vision Transformer

Md Nazmul Karim, Umar Khalid
University of Central Florida
(nazmul.karim18,umarkhalid)@knights.ucf.edu

Abstract

With the advancement of deep learning (DL) models in fields like computer vision and natural language processing (NLP), there have been a growing concern about the safety of these models against attacks. One such attack is adversarial attack where one can make imperceptible changes to the input such that the DL model misclassify the modified input. The modified input, also known as adversarial examples, can be generated either accessing the parameters of the neural network or in a black box manner. Efforts have been made to generate such adversarial examples in different ways to fool different type neural networks. One such type of network is vision transformer. Attacking a vision transformer is hard since it does not work like conventional CNN. Other reasons coould be that a tarnsformer does not have inductive biases like convolutional neural network and it is also permutation invariant. In this work, we present an adaptive attack mechanism that fools a transformer to misclassify. We have conducted a study to find the right type of attack for a transformer.

1. Introduction

Machine learning (ML) technique is known to learn meaningful patterns when exposed to a lot of data. Due to this power of learning, it can be used to solve problems such as prediction and classification etc. One of the branches of machine learning is deep neural network (DNN) where neurons with activation are used for leaning the representation of the underlying data distribution. For tasks like image classification and video recognition, convolution neural networks (ConvNets) have proven to be highly successful [5, 10, 11, 13]. The reason for the advancement of the computer vision field can be attributed to the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [9]. This dataset has served as the resource for testing large scale classification and recognition systems along with many feature learning based applications.

In recent times, transformer [3] has become essential in solving problems in computer vision. In a transformer, there is no convolution layer and one has to feed sequential data to it instead of grid structured data. A transformer first split an image into patches followed by flattening of those patches. After getting the linear embeddings, the positional embeddings of is learned in a conventional manner. These positional embeddings are then fed to the trandformer encoder. This whole process is then optimized using a huge dataset and used as a pretrained model for the downstream task like image classification or object detection. Transformers unconventional way of replacing convolution layer makes us rethink some of the concerns and applications in computer vision. One such concern is the security of the vision transformer against adversarial attacks. Until now, most of the attacks are developed convolution based learning systems.

The large scale classifier seems to be vulnerable to adversarial examples [4, 12]. It has been shown that classifiers gets fooled by the examples that are slightly different from the samples that are correctly classified. This is a huge concern since perturbation generated for one particular model can fool other models trained on the same training data. Since the revelation of such vulnerability of neural networks, researchers have developed different type of attacks, e.g. iterative FGSM [6], Carlini and Wagner attack [2], adversarial patch [1], PGD attack [7] etc.Deepfool[8] is another technique that generates a small perturbation iteratively to misclassify images. A linear model approximation was employed for this with a first order approximation to calculate the perturbation. Another strong attack, PGD is designed to generate the perturbation that increases and maximizes the loss of a neural network for a particular input while the perturbation size is constrained to a small specific constant referred to as epsilon. This constraint can be expressed as the $l2$ or $l\infty$ norm of the perturbation.

In this project, we evaluate the robustness of vision transformer against differnt type of attacks. We then develop an adaptive attack based simple fast gradient sign method and and basic iterative attacks. We present the case that strong white box attacks like PGD, carlini and wagner attack or deepfool does not work well on the transformer. The perturbation can be found in one or two steps. Iterative based approaches does not work well for the transformer. Our ex-

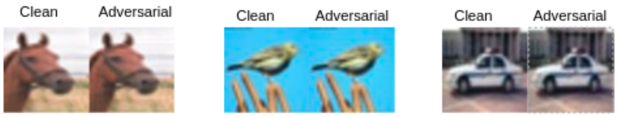

Figure 1. Some of the clean and adversarial images generated by our attack. The model we attacked is without any defense and the norm of the perturbation is bounded to 6.

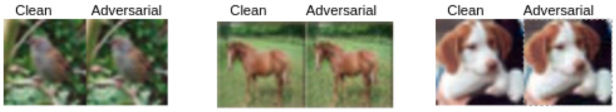

Figure 2. Some of the clean and adversarial images generated by our attack. The model we attacked is defended using image resizing technique and the norm of the perturbation is bounded to 10.

perimental results shows that the proposed attack achieves 98 % success rate in terms of fooling the normally trained transformer.On the other hand, when we use a defense technique like resizing we have a fooling rate of 45 %.

2. Adversarial Attack Generation

We used 3 different attack where PGD and FGSM are quite similar. The projected gradient descent method is primarily based on Fast Gradient Sign Method (FGSM). In FGSM, for some given example x, an adversarial example is generated base on the sign of the gradient descent which is first computed using back propagation. In order to, maximize the loss, a perturbation is generated in the direction of the gradient.

$$x_{adv} = x + \delta,\ where\ \delta = \epsilon \times sgn(\nabla_x L(\theta, x, y)) \quad (1)$$

FGSM is an attack for an $l\infty$ bounded adversary, where the perturbation δ is bounded by epsilon value such that $||\delta||_\infty \leq \epsilon$. FGSM is a one step method for maximizing the loss between adversarial example and an original sample. Projected gradient descent (PGD) is essentially an extension of FGSM where multiple steps can be taken to generate an adversarial example and given as,

$$x^{t+1} = \Pi_{x+S}(x^t + \alpha \times sgn(\nabla_x L(\theta, x, y))) \quad (2)$$

where Π denotes the projection onto the norm ball of interest. The third method we used is basic iterative method to generate perturbations. All of these attacks are generated using the python foolbox module.

We have observed that strong iterative attacks like PGD alone does not decrease classification accuracy. We have tried carlini and wagner attack, deepfool attack too. All of these attacks are executed in at least 10 different settings. Iterative methods project the avd images to an l-p ball in each iteration which distorts perturabtion direction and magnitude. We are hypothesizing that this puts the adversarial examples in unfavorable position in the input space. ”Unfavorable” is used as the chance to be miscalssified being decreased. The decision boundary of a vision transformer might be significantly different than the normally trained convolutional neural networks. So, whenever we calculate the perturabtion in multiple steps, the position of the adverasrial examples end up being closer to the clean sample. This may be because of the random loss gradient sign at each step. This motivates us to use less steps to produce perturbation. But, even for step size of 5, the fooling rate does not increase that much.

On the other hand, one step method like FGSM performs well. If we just use FGSM alone, it does not work for all the samples. Since, it is an one step method, it has one chance to misclassify that sample. In one iteration, the loss gradient with respect to image may not give us the proper perturbation for all the samples. If we use FGSM one time, we classify M number of samples out of 100 sampels, where M is far less than 100. We then use FGSM again for rest of the $(100 - M)$ samples. This time, the gradient sign and magnitude is enough to fool N of these $(100 - M)$ samples. We then run FGSM again for $(100 - M - N)$ samples. We do this 4-5 times. This is still not enough for misclassifying all the samples. We then use basic iterative method one time followed by several PGD attacks with different settings.

3. Experiments and Results

After fixing the attack settings, we run our experiment for different perturbations norms. Figure 3 shows the attack success rate for normally trained model. It can be seen that the success rate is saturated after the a certain valued norm. We used a value of 4 for the ϵ and got an attack success rate of 98%. In figure 4, we show the success rate for defended model. The success rate increases linearly with ϵ. We use a value of 10 for ϵ and get a success rate of 45%. Figure 1 and 2 show the clean and adversarial samples for normally trained and defended models respectively. Since we used a small perturbation norm, the images are practically similar, at least visually. The resolution is low and one could find it hard to observe the differences between clean and adversarial images.

4. Summary

In summary, Transformer are vulnerable to one step attack than multi-step attacks. Our results supports this claim. However, this might not be completely true as we did explore other direction of attacks such as query based black box attacks. We also could not explore other combination of white box attacks because of time constraints.

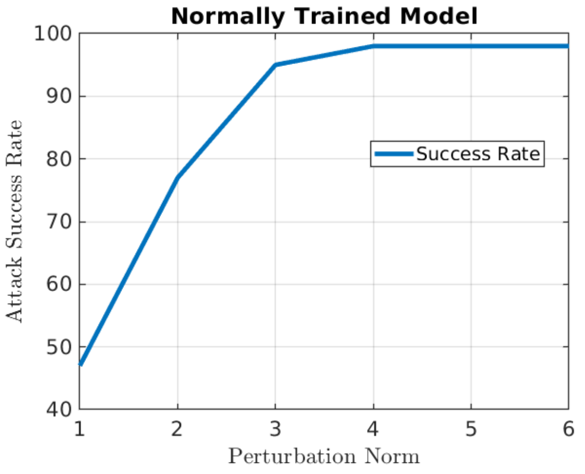

Figure 3. Attack success rate of the normally trained model for different perturbation sizes. After a ϵ value of 4, we have a fixed attack success rate.

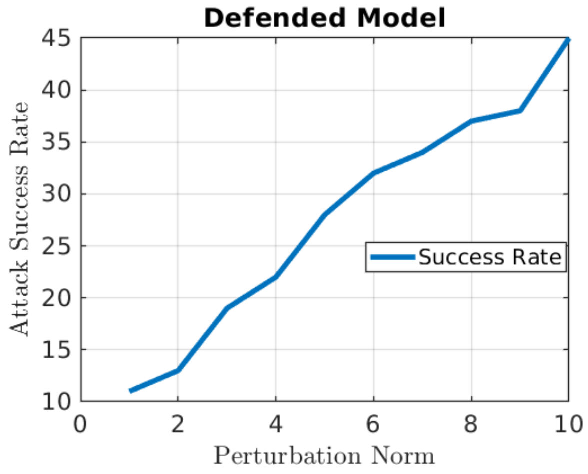

Figure 4. Attack success rate of the defended model for different perturbation sizes. The attack success rate is almost linear with the values of ϵ. But we are bounded to use a maximum value of 10.

References

[1] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 1

[2] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 1

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1

[6] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 1

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1

[8] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 1

[10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. 1

[11] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014. 1

[12] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[13] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1