Azerbaijan Journal of High Performance Computing, Vol 3, Issue 1, 2020, pp. 46-53
https://doi.org/10.32010/26166127.2020.3.1.46.53

Defending Strategies Against Adversarial Attacks in Retrieval Systems

Suleyman Suleymanzade

Institute of Information Technology, Azerbaijan National Academy of Sciences, Baku, Azerbaijan, suleyman.suleymanzade@socar.az

*Correspondence: Suleyman Suleymanzade, Institute of Information Technology, Azerbaijan National Academy of Sciences, Baku, Azerbaijan, suleyman.suleymanzade@socar.az

Abstract

During history, retrieval systems become more complicated in their architecture design and work principles. The system that gathers text and visual data from the internet must classify the data and store it as the set of metadata. The modern AI classifiers that are used in retrieval systems might be tricked by skilled intruders who use adversarial attacks on the retrieval system. The goal of this paper is to review different strategies of attacks and defenses, describe state-of-the-art methods from both sides, and show how important the development of HPC is in protecting systems.

Keyword: adversarial attacks, retrieval systems, FGSM, PGD, HPC

1. Introduction

The main goal of adversarial attacks is to trick the classifier in such a way to cause machine learning to make a mistake. In the lousily coupled infrastructures such as retrieval systems, adversarial attacks, as well as the other types of attacks (Hydara, I., Sultan, A. B. M., Zulzalil, H., & Admodisastro, N., 2015; Voitovych, O. P., Yuvkovetskyi, O. S., & Kupershtein, L. M., 2016; Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W., 2017; Bai, Y., & Chen, Z., 2015), create multiple threats for system's authenticity, possession, availability, and integrity

An attack strategy is designed by hackers depending on their goals and intrusion capabilities to a victim system. From the other side, the defender tactics must be modeled concerning thread modeling techniques with possible attack vectors. That causes to the conclusion that the defender must not ignore any possible threats to a system by any skilled hacker. With regarding accessibility to the system by the intruder, there are two main attack classes:

- Black Box Attack
- White Box Attack

The black box attacks strategies usually selected by the hackers if there is no data about the victim model (Papernot, N., McDaniel, P., Goodfellow, I., et al., 2017). In contrast, the white box attacks represent the attacking mechanism of the system's integrity through changing the model's parameters (gradients) (Zhang, Y., & Liang, P.,

2019).

Both strategies can also be classified as Targeted and Un-targeted attacks (Kwon, H., Kim, Y., Park, K. W., et al., 2018). In a targeted attack, the attacker disturbs the input data to predict the wrong but specific target class. An untargeted attack defines the attack techniques where the target label can be anything except the correct label.

2. Attack strategies

Fast Gradient Sign Method FGSM

Google introduced this method (Goodfellow, I. J., Shlens, J., & Szegedy, C., 2014). The attack is classified as a white box attack because the attacker Initially must have access to the training set. The Idea of FGSM based on the manipulation of the gradients of the loss for the input data in order to create the new data that can maximize the loss. The loss maximization must not be random but calculated in such a way to change the gradients onto the direction to misclassify a model. This kind of attack works well for images because it is hard to detect the little changes in images for the human eye, especially when only a few pixels are replaced.

In famous work, the researchers show that adding small noises to the original image of the panda causes the model to tag the image as a gibbon, with high accuracy (Goodfellow, I. J., Shlens, J., & Szegedy, C., 2014).

In this article, the classifier defined by the deep neural network (DNN) with the SoftMax output activation as $\tilde{y} = f(\theta, x)$ for a given data-label pair (x, y). FGSM identifies the adversarial data $\tilde{x}$ by maximizing the loss $L(\tilde{x}, y) = L(f(\theta, x'), y)$ subject to the l_∞ perturbation constraint $||x' - x||_\infty \leq \varepsilon$ with ε to be the attack strength. Under the first-order approximation, i.e., $L(\tilde{x}, y) \approx L(x, y) + \nabla_x L(x, y)^T \cdot (\tilde{x} - x)$ the adversarial data can be presented as

$$\tilde{x} = x + \varepsilon * sign(\nabla_x J(\theta, x, y)),$$

where

- $\tilde{x}$: Adversarial data,
- x: Original input data,
- y: Original input label,
- ε: Multiplier,
- θ: Model parameters,
- J: Loss.

There is also an extension of FGSM with additionally enhanced iterations IFGSM that can be defined as

$$x^m = x^{m-1} + \varepsilon * sign(\nabla_x J(\theta, x^{m-1}, y)),$$

where $m = 1, \dots, M$, $x^{(0)} = x$ and $x' = x^{(M)}$, with M being the number of iterations, the targeted FGSM can mislead any CNN with ReLU activation to classify. This formula can be extended with an elementwise clipping function which clips each element x^m of the input x into the range of $[\max(0, x^m - \varepsilon), \min(1, x^m + \varepsilon)]$,

$$x^m = clamp\left(x^{m-1} + a * sign\left(\nabla_x J(\theta, x^{m-1}, y)\right)\right), x^0 = x,$$

where the $a = \frac{\varepsilon}{N}$. Typically, each component of the input vector, e.g., a pixel, is normalized within [0, 1].

Project Gradient Descent (PGD) attack is another type of IFGM (Wu, F., Gazo, R., Haviarova, E., & Benes, B., 2019). The idea based on randomly picking a point within a confined wrapper around each clean input and then applying the multi-step IFGM in order to model adversarial data for the right input.

There are some powerful gradient-based attacks known for today:

Elastic-Net attacks EAD based on the L_2 and L_∞ distortion, where $an\ L_1$-oriented adversarial example includes the state-of-the-art L_2 attack in the particular case (Chang, T. J., He, Y., & Li, P., 2018).

Another example based on the idea to us e C&W algorithm on L_2 shows that the adversarial example can be generated as an interactive attack (Chen, P. Y., Sharma, Y., Zhang, H., Yi, J., & Hsieh, C. J., 2018; Carlini, N., & Wagner, D., 2017). The loss function was defined as $l(x') = max\ (max\{Z(x')_i : i \neq t\} - Z(x')_t, -k)$. As κ increases, the model classifies the adversarial example as increasingly more likely; this example was high-confidence adversarial.

The third powerful gradient-based attack was described (Carlini, N., & Wagner, D., 2017). The method based on PGD adversarial example with the "first-order adversary" represented as most vigorous attack utilizing the local first-order information about the network

In the conclusion of the gradient-based attacks, the research shows that if an intruder can access to the model's gradients, they can craft a new fake-set of adversarial examples to trick the model. It proves that adversarial example is hard to be detected with classic methods – the type of security through obscurity aside is hard to defend against them.

Backward Pass Differentiable approximation.

Uses against the defender that mask the gradients, so approximation of $f(x)$ becomes a hard problem. In that case, attackers build the neural network from scratch by using the same train data as the classifier where he uses his gradients. The idea is simple: if the attacker cannot use the gradients, he creates the gradients. BPDA allows for attacking non-differentiable networks by approximating the gradients of the non-differentiable layer. The gradient is estimated by computing the forward pass normally but replacing a non-differentiable layer $f(\cdot)$ with a differentiable approximation $h(\cdot) \approx f(\cdot)$ on the backward pass.

Assume that the transform $f(x)$ is simple; if data were altered too much, it would be hard to predict the correct label correctly. If a gradient from $f(x)$ is unattainable, then the attacker defines the network $g(x)$ and train the neural network to approximate $f(x)$. If $g(x)$ approximates $f(x)$, then there is a problem to getting it is gradient and using it to replace the one that would come from $f(x)$ when running their optimizer.

3. Defending strategies

Defending strategies must rely on research methodologies, such as formal methods or empirical defenses (Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A., 2017; Chong, S., Guttman, J., Datta, A., Myers, A., et al., 2016; Voas, J., & Schaffer, K., 2016; Smith, W., 2019). Formal methods are a mathematical technique used to guarantee the robustness of software/ hardware systems. As applied to neural networks, state of the art formal method techniques today cannot verify a network more than a few layers deep. Unlike the formal methods, empirical defenses are relying on

experiments to evaluate the effectiveness of a defense. There are some well-known strategies against adversarial attacks based on empirical methods: Adversarial training, Gradient Masking, Extra class, Input modification, Detection, and the Barrage of Random Transform BaRT.

Adversarial Training

This technique based on the idea that the defender trains the adversarial examples with the rest of the dataset. This method teaches the model to ignore noises and only learn it from the robust feature.

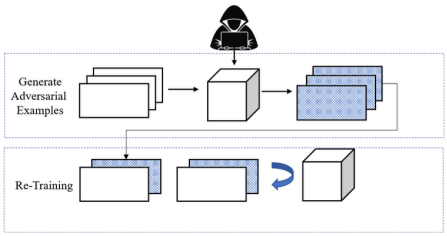

Fig. 1: Adversarial Attack

The adversarial training has a disadvantage – it can only defend the model against the same attack used to craft examples included initially in the training pool. The increasing number of adversarial examples that defender adds to the training set creates another problem with the model's underfitting.

In general, to describe the adversarial training the risk minimization (ERM) must be defined, where the aim is to minimize the risk over adversarial example

$$h* = argmin_{h\in\mathcal{H}} \mathbb{H}_{(x,y_{true})\sim D}\left[\max_{||x^{adv}-x||_\infty\leq\epsilon} L(h(x^{adv}), y_{true})\right],$$

- $h \in \mathcal{H}$: target model,
- $budget\ \epsilon\ where\ ||x^{adv}-x||_\infty \leq \epsilon.$

There is some variation of the adversarial training method: Ensemble adversarial training, where the augments training data with perturbations transferred from other models (Oltramari, A., & Kott, A., 2018).

Another method was proposed in by training the dataset via Spectral Normalization (Tramèr, F., Kurakin, A., Papernot, N., et al., 2017)

The Barrage of Random Transform BaRT

The Idea of BaRT technique based on modifying the image at the inference time (Farnia, F., Zhang, J. M., & Tse, D., 2018). This modifying includes the transformation such as blurring, noise adding, FFT Alteration, Gaussian blur effects (Raff, E., Sylvester, J., Forsyth, S., & McLean, M., 2019). Moreover, this chain of transformation is done randomly. The following algorithm shows how BaRT is applied to the data:

- Select a large number of transformations;
- Tune each of the transformations randomly;
- Select a subset of transformations to apply them for the input;
- Produce the transforming in random order.

Experience showed that, even after recalculating the adversarial gradients, the Barrage of Random Transforms (BaRT) is one of the most powerful defense methods, even the most severe attacks, such as PGD.

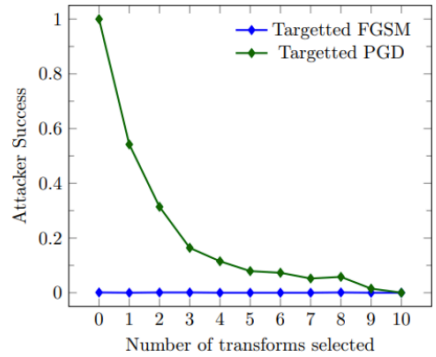

Fig. 2. BaRT performance

The figure above shows that the number of transformations dramatically drops the success of the adversarial attack.

.4. Combined technique with parallel computation and detection process on hpc

The usage of HPC for face detection in crowded places where all detection processes are done in parallel is at risk of adversarial attacks. The data is streamed continuously, and the training processes must be handled continuously. In such complicated systems, detection and adversarial training processes on HPC are done in partitioned forms. Two strategies must be balanced in order to achieve efficient work.

Distributed detection strategy:

The distributed approach based on the separation of concerns conception with the space of multiple HPC processes where each process uses one entity of trained neuron network in order to detect only one adversarial attack. The advantage of the distribution strategy is that detection agents (processes) use lightweight neural networks that detect and trained fast. However, detection is done in a round-robin way where each agent that responsible for the different adversarial attacks must check the same object.

Clone agent strategy:

This approach is based on an overloaded neural network that can detect multiple adversarial attacks. The advantage of this approach is to use the copies of the same process scan objects only once, without repeating. The disadvantage of such an approach is the risk of the model's overfitting.

Another advantage of HPC in adversarial learning is the ability to parallelize the process of training. Some adversarial methods such as MAT (A Multi-strength Adversarial Training Method to Mitigate Adversarial Attacks) combines the effect of multiple adversarial strength has parallel computation version which called a parallel MAT it consists of multiple neural networks and summarized in upper-boundary

decision unit (Naidu, V. P. S., 2011). Each of these neural networks can be trained in parallel on HPC. One approach is to use the parallelizing backpropagation of neural network with MapReduce and cascading model (Song, C., Cheng, H. P., Yang, H., Li, S., et al., 2018).

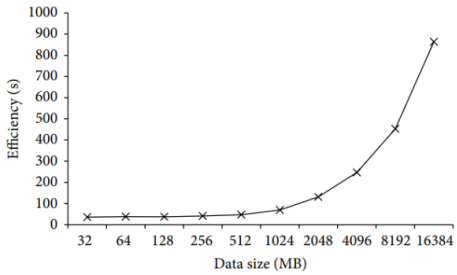

Fig. 3: Parallel training efficiency on HPC

The efficiency of parallelism of the neural network, as well as any other parallel computation process, is calculated by Amdahl's law and High-Performance Conjugate Gradients (HPCG) (Liu, Y., Jing, W., & Xu, L., 2016; Amdahl, G. M., 1967). The efficiency of parallel backpropagated neural network CBPNN against standalone, standalone BPNN shown in the figure above.

5. Conclusion

This paper attempted to describe the state-of-the-art studies for adversarial examples from the attacker and defender perspective in the deep learning domain. The threat of adversarial attacks is increasing, but there are few defense methods. The defending strategies against adversarial attacks, which were mentioned above, include training phases that take time to adapt the system for new adversarial attacks. The attacker might use not only one, but the set of attacking models to trick the system while adversarial training runs at the background. One way to solve this problem is to parallelize the adversarial training process against a single attacking model; another way is to use ensemble methods, including horizontal and vertical strategy. These issues force the use of powerful server-based HPC systems to respond to the next attack quickly. The impact of HPC so solve computational problems is growing in many fields (Ismayilova, N., & Ismayilov, E., 2018). According to, the growth of computational power and AI-based systems causes an increase in the number of adversarial based attack techniques. This trend will continue to grow. Defenders must react to this problem before and take action at the time before the attacker damages a system.

References

Amdahl, G. M. (1967, April). Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference* (pp. 483-485).

Bai, Y., & Chen, Z. (2015, November). Analysis and Exploit of Directory Traversal

Vulnerability on VMware. In *International Conference on Applications and Techniques in Information Security* (pp. 238-244). Springer, Berlin, Heidelberg.

Carlini, N., & Wagner, D. (2017, May). Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)* (pp. 39-57). IEEE.

Carlini, N., & Wagner, D. (2017, November). Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security* (pp. 3-14).

Chang, T. J., He, Y., & Li, P. (2018). Efficient two-step adversarial defense for deep neural networks. *arXiv preprint arXiv:1810.03739.*

Chen, P. Y., Sharma, Y., Zhang, H., Yi, J., & Hsieh, C. J. (2018, April). Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-second AAAI conference on artificial intelligence.*

Chong, S., Guttman, J., Datta, A., Myers, A., et al. (2016). Report on the NSF workshop on formal methods for security. *arXiv preprint arXiv:1608.00678.*

Farnia, F., Zhang, J. M., & Tse, D. (2018). Generalizable adversarial training via spectral normalization. *arXiv preprint arXiv:1811.07457.*

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572.*

Hydara, I., Sultan, A. B. M., Zulzalil, H., & Admodisastro, N. (2015). Current state of research on cross-site scripting (XSS) – A systematic literature review. *Information and Software Technology, 58,* 170-186.

Ismayilova, N., & Ismayilov, E. (2018) Convergence of HPC and AI: Two Directions of Connection. Azerbaijan Journal of High Performance Computing, 1(2), 179-184.

Kwon, H., Kim, Y., Park, K. W., et al. (2018). Friend-safe evasion attack: An adversarial example that is correctly recognized by a friendly classifier. *computers & security, 78,* 380-397.

Liu, Y., Jing, W., & Xu, L. (2016). Parallelizing backpropagation neural network using MapReduce and cascading model. *Computational intelligence and neuroscience, 2016.*

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083.*

Mahjabin, T., Xiao, Y., Sun, G., & Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *International Journal of Distributed Sensor Networks, 13*(12), 1550147717741463.

Naidu, V. P. S. (2011, November). Multi-resolution image fusion by FFT. In *2011 International Conference on Image Information Processing* (pp. 1-6). IEEE.

Oltramari, A., & Kott, A. (2018). Towards a reconceptualisation of cyber risk: an empirical and ontological study. *Journal of Information Warfare, 17*(1), 49-73.

Papernot, N., McDaniel, P., Goodfellow, I., et al. (2017, April). Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security* (pp. 506-519).

Qiu, S., Liu, Q., Zhou, S., & Wu, C. (2019). Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences, 9*(5), 909.

Raff, E., Sylvester, J., Forsyth, S., & McLean, M. (2019). Barrage of random transforms for adversarially robust defense. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6528-6537).

Smith, W. (2019). A Comprehensive Cybersecurity Defense Framework for Large Organizations.

Song, C., Cheng, H. P., Yang, H., Li, S., et al. (2018, July). MAT: A multi-strength adversarial training method to mitigate adversarial attacks. In *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)* (pp. 476-481). IEEE.

Thuraisingham, B. (1993). Multilevel security for information retrieval systems. *Information & management, 24*(2), 93-103.

Tramèr, F., Kurakin, A., Papernot, N., et al. (2017). Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204.*

Végh, J. (2019). How Amdahl's Law limits the performance of large artificial neural networks. *Brain informatics, 6*(1), 4.

Voas, J., & Schaffer, K. (2016). Whatever happened to formal methods for security? *Computer, 49*(8), 70.

Voitovych, O. P., Yuvkovetskyi, O. S., & Kupershtein, L. M. (2016, September). SQL injection prevention system. In *2016 International Conference Radio Electronics & Info Communications (UkrMiCo)* (pp. 1-4). IEEE.

Wu, F., Gazo, R., Haviarova, E., & Benes, B. (2019). Efficient Project Gradient Descent for Ensemble Adversarial Attack. *arXiv preprint arXiv:1906.03333.*

Xie, J., Xu, B., & Chuang, Z. (2013). Horizontal and vertical ensemble with deep representation for classification. *arXiv preprint arXiv:1306.2759.*

Zhang, Y., & Liang, P. (2019). Defending against whitebox adversarial attacks via randomized discretization. *arXiv preprint arXiv:1903.10586.*

Submitted 12.01.2020
Accepted 20.05.2020