INVESTIGATE THE EFFECT OF NEURAL NETWORK REPRESENTATIONS ON THE TRANSFERABILITY OF ADVERSARIAL ATTACKS

By

HAI HUYNH

Bachelor of Science in Computer Science

Oklahoma State University

Stillwater, OK

2018

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2020

INVESTIGATE THE EFFECT OF NEURAL

NETWORK REPRESENTATIONS ON THE

TRANSFERABILITY OF ADVERSARIAL ATTACKS

Thesis Approved:

Dr. Christopher Crick

Thesis Adviser

Dr. Johnson Thomas

Dr. Blayne Mayfield

Name: HAI HUYNH

Date of Degree: MAY, 2020

Title of Study: INVESTIGATE THE EFFECT OF NEURAL NETWORK REPRESENTATIONS ON THE TRANSFERABILITY OF ADVERSARIAL ATTACKS

Major Field: COMPUTER SCIENCE

Abstract: Deep neural networks have been widely applied in various fields of many industries such as medical, security, and self-driving cars. They even surpass human performance in image recognition tasks; however, they have a worrying property. Neural networks are vulnerable to extremely small and human-imperceptible perturbations in images that lead them to provide wrong results with high confidence. Moreover, adversarial images that fool one model can fool another even with different architecture as well. Many studies suggested that a reason for this transferability of adversarial samples is the similar features that different neural networks learn; however, this is just an assumption and remains a gap in our knowledge of adversarial attacks. Our research attempted to validate this assumption and provide better insight into the field of adversarial attacks. We hypothesize that if a neural network representation in one model is highly correlated to the neural network representations of other models, an attack on that network representation would yield better transferability. We tested this hypothesis through experiments with different network architectures as well as datasets. The results were sometimes consistent and sometimes inconsistent with the hypothesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Deep neural networks have been widely applied in various fields of industries such as medical, security, and self-driving cars, and surpassed human performance in image recognition tasks; however, a research by Szegedy et al. [6] discovered that neural networks had a worrying property. They were vulnerable to extremely small perturbation and wrongly label perturbed images with high confidence (Figure 1). Additional studies [11, 12, 13] also confirmed the vulnerability of deep neural networks.

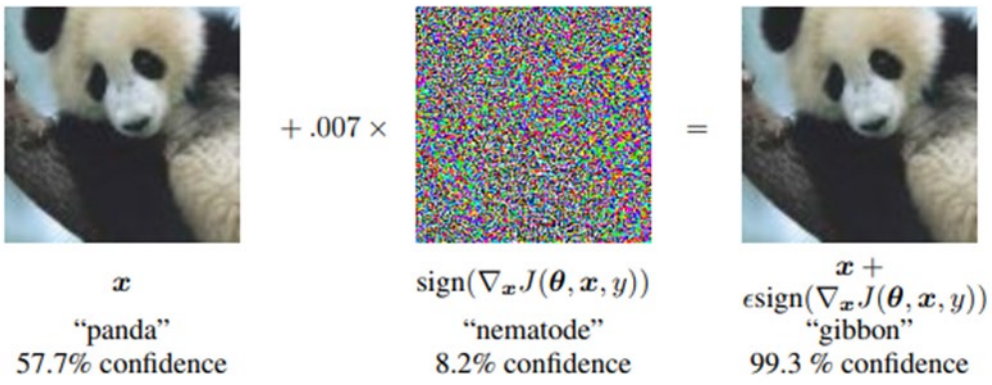

Figure 1. Adversarial attack on an image using FSGM method

An adversary could exploit this property to cause disastrous consequences. A self-driving car could potentially interpret a stop sign as an advertisement banner and crash into other vehicles. Studies in [1, 10] have found that printed adversarial samples were still able to fool deep learning models as in Figure 2.

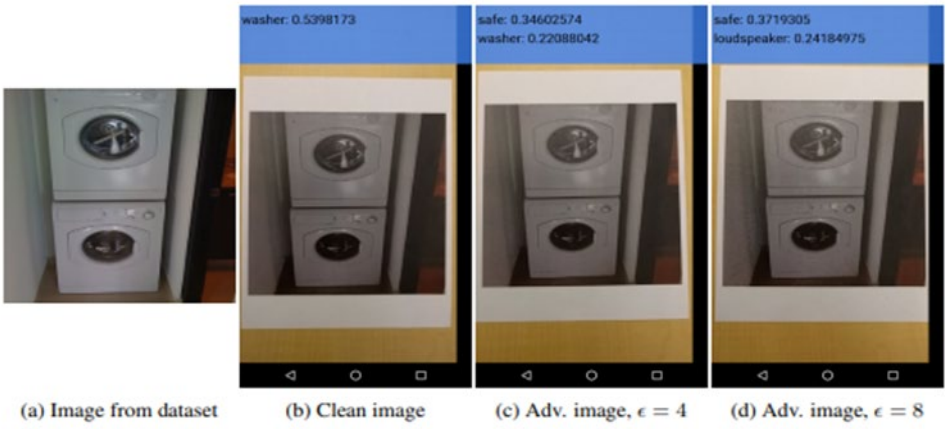

Figure 2. Printed adversarial image can still fool deep learning models

Recently, a security team from McAfee [8] shared their research with the public showing that a modified speed limit 35 sign as in Figure 3 could trick the Tesla into interpreting it as a speed limit 85 sign. Also, another worry property of neural networks is that adversarial samples for one model can fool the other models of the same input dataset [5, 14, 15].

Figure 3. Attack on a speed limit sign in an experiment conducted by McAfee. Tesla car interpreted speed limit 35 as speed limit 85

In the current stage of adversarial attack literature, there is limited knowledge on the transferability of the attack as to why it happens or how it happens. One suggestion is that

different neural networks learn similar features on the same input space; therefore, they are vulnerable [3]. However, this is just an assumption. In this project, the main objective is to study the effect of network representations on the transferability of adversarial attacks as follows:

- Obtain the similarity matrix of the models
- Attack each layer of the simpler models and transfer to more complex models
- Explore different network architectures and dataset

This research is an initial attempt to validating the assumption of shared features leading to transferability as well as narrowing the gap of knowledge in the adversarial attack literature.

CHAPTER II

REVIEW OF LITERATURE

In 2013, Szegedy et al. [6] published a study showing that neural networks had some intriguing properties. They found out that neural networks did not generalize as well as we thought. Imperceptible engineered noise would cause neural networks to fail at their tasks (Figure 4) despite being able to resist random decently. Szegedy's research has opened up a whole new gap in our understanding of neural networks. Even though many efforts have been put into the literature, our knowledge remains limited in this area. Our study explores the unknown area and contributes to narrowing this gap.

(a) Even columns: adversarial examples for a linear (FC) classifier (stddev=0.06)

(b) Even columns: adversarial examples for a 200-200-10 sigmoid network (stddev=0.063)

(c) Randomly distorted samples by Gaussian noise with stddev=1. Accuracy: 51%.

Figure 4. Neural networks do not generalize well with adversarial examples (a) and (b), even though they are robust to random noise (c)

Based on Szegedy's work, Goodfellow et al. [2] introduced the fast gradient sign method (FGSM) that inspired many other attacks including the one we will utilize in our research. The idea of this attack is simple. Instead of minimizing the value of the loss function by using gradient descent, the attack performs a small perturbation in the directions that will maximize the loss function (gradient ascent) (Figure 1). This work acted as the basis for the activation attack that we used in our research to further explore the field of adversarial attacks.

Papernot et al. [5] dug deeper into the black box scenario and provided a practical attack on real-world image classifiers (Figure 5). The authors used a substitute model that mimics the decision boundary of the oracle model and then attack the substitute. The generated adversarial samples were then used on the oracle model and achieved a great success rate. Our project extended Papernot's research by applying the idea of using a substitute model to fool the target model to perform further experiments on the transferability.

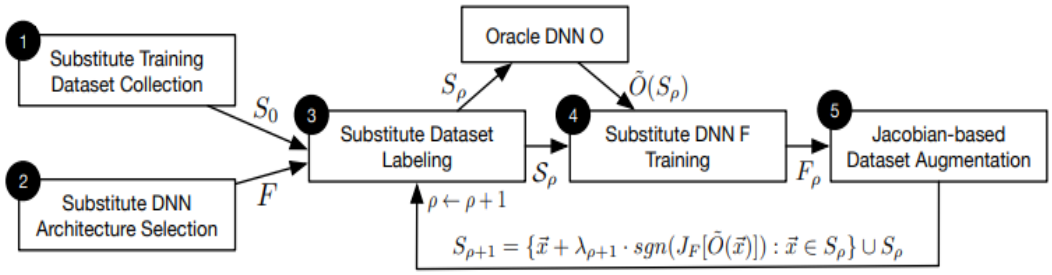

Figure 5. Substitute method on attacking a black-box model. The adversary train a substitute model with the use of both normal and synthesis input (via Jacobian-based Dataset Augmentation), perform attack on the this model to generate adversarial samples

With transferability, as mentioned before, a suggestion is the similarity of learned features of the models. The idea that sparked such a suggestion was the work of Yosinski et al. [7]. Their research inspires not only the transfer learning area but also the study in black-box adversarial attacks. In [7], they substituted a portion of the un-trained models with the parts from pre-trained

models to test their hypothesis that neural networks learn similar features given the same input dataset (Figure 6). The results did confirm that using a portion of the pre-trained models helped speed up the learning process of the new models. Many transferable adversarial attack methods are based on Yosinki et al.'s work including [3].

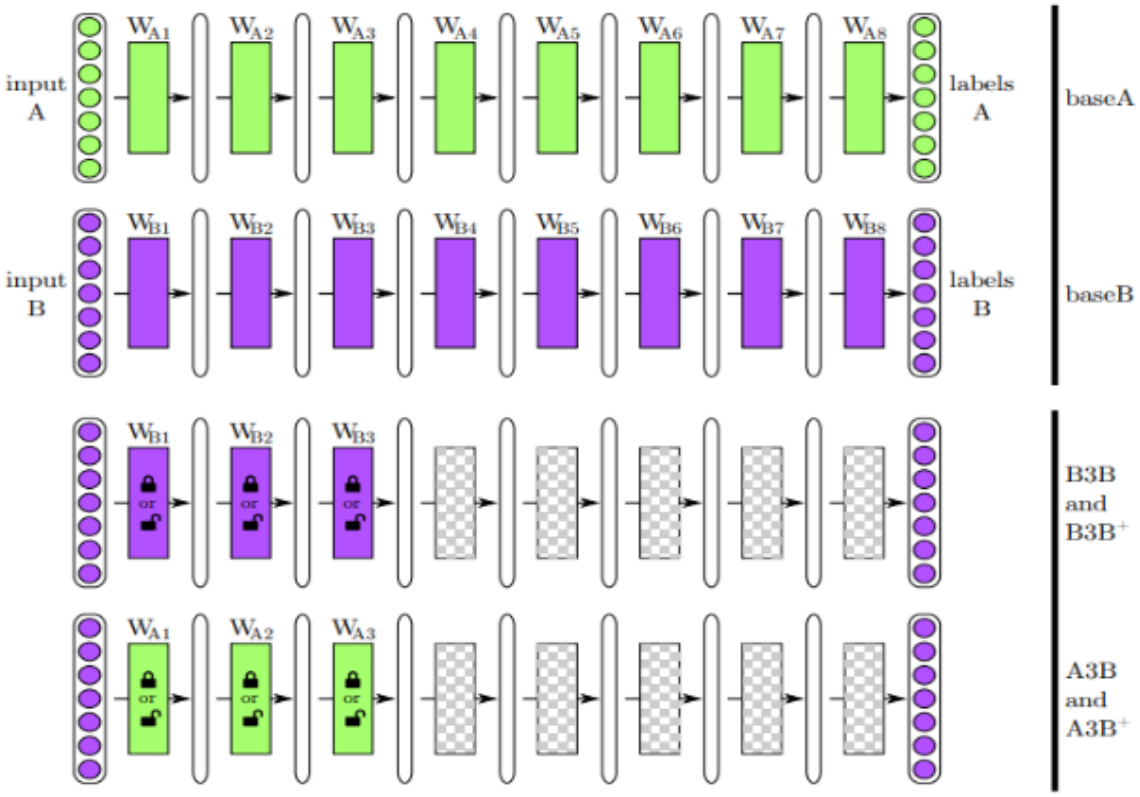

Figure 6. Transfer learning process. Using pre-trained layers, the learning time is reduced significantly for the untrained models. This behavior suggests that models learn similar features.

The idea that neural networks learn similar features on the same input inspired Inkawhich et al. [3] to create a black-box attack with highly effective transferability. As mentioned before, this attack (activation attack) assumed that neural networks of the same input dataset learn similar features, and thus attacking features would yield a better transfer rate. Activation attack's objective is to modify the activation of layers of an input image so that they will become similar to the activation of another image, thus fooling the model (Figure 7).

Kornblith et al. [4] provided us a valuable tool to generate a similarity matrix (Figure 8) of two neural networks through layer-wise comparison and aggregation, details will be discussed in the following sections.

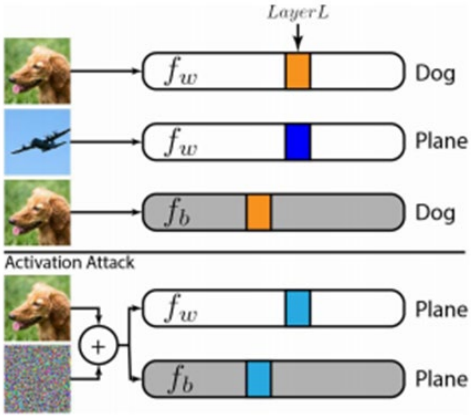

Figure 7 . Activation Attack. Modifying the input image so that the activations of it becomes more similar to a target image

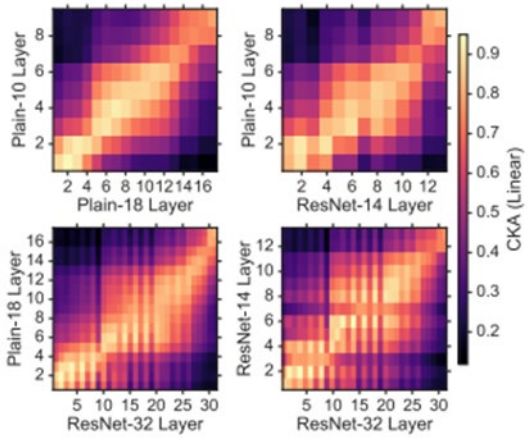

Figure 8. Similarity Matrices between layers of one model to layers of other models even with different architectures

Our research complements both [3] and [4] by utilizing their methods to perform experiments on the effect of similar features on the transferability of adversarial attack, thus, it provides a better insight for this inexplicit area of the literature.

CHAPTER III

METHODOLOGY

This section explores the technical details of the tools we used in our experiments including the work of [3] and [4]. We will first discuss the metrics that were used to measure the similarity as well as the effectiveness of the transfer attack. Then, the activation attack of [3] will be discussed and followed by the experiment setup.

A. Metrics

1. Effectiveness of the transferability

The measurement of the effectiveness of a transfer attack is considered the error that the attack introduces into the target model:

$$\mathrm{E_i} = \mathrm{E_a} - \mathrm{E_n}$$

Where E_i is the introduced error, E_a is the error of the model on the adversarial samples, and $E_n$is the error of the model on the clean input. By using this measurement, we could completely focus on the effect of the attack that is carried on the model without having to pay attention to other details.

2. Similarity

An intuitive approach to measure the similarity between two vectors is the dot product.

$$\langle a, b \rangle = \sum_{i=0}^{n} a_i b_i$$

Higher value indicates that the two vectors are more related. Similarly, the dot product can be applied to comparing the similarity between two feature matrices as in Figure 9. X represents the activation of layer A for all of the input of one model and Y represents the activation of a layer in model Y. Each row is the flatten array of activated features of a model.

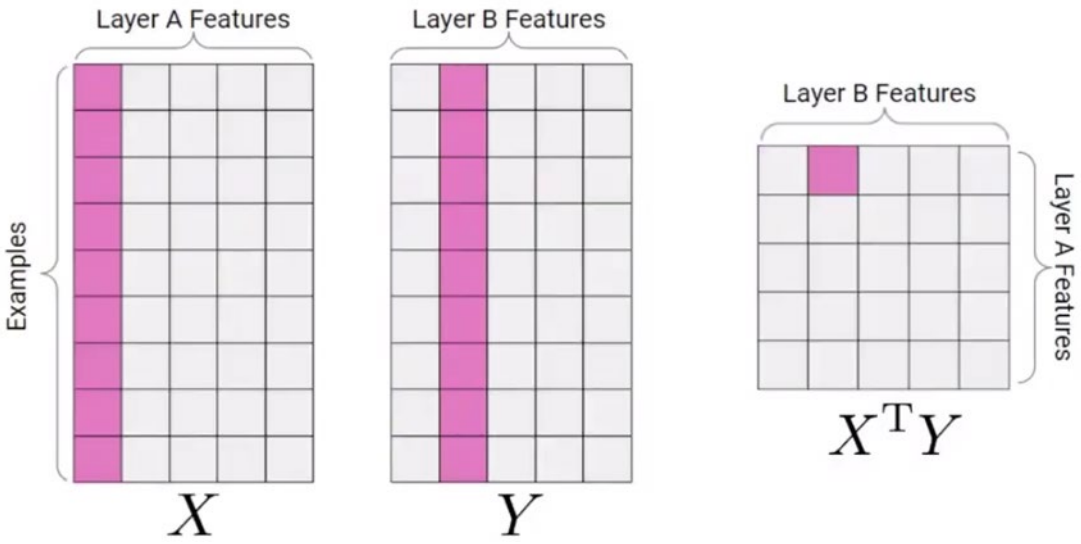

Figure 9. Using the dot product to obtain the similarity between features of 2 different model

When we calculate the norm on $X^T Y$, we could retrieve a single value for the comparison score. As mentioned in [4], instead of using feature comparison as above, we could use example comparison Figure 10 instead.

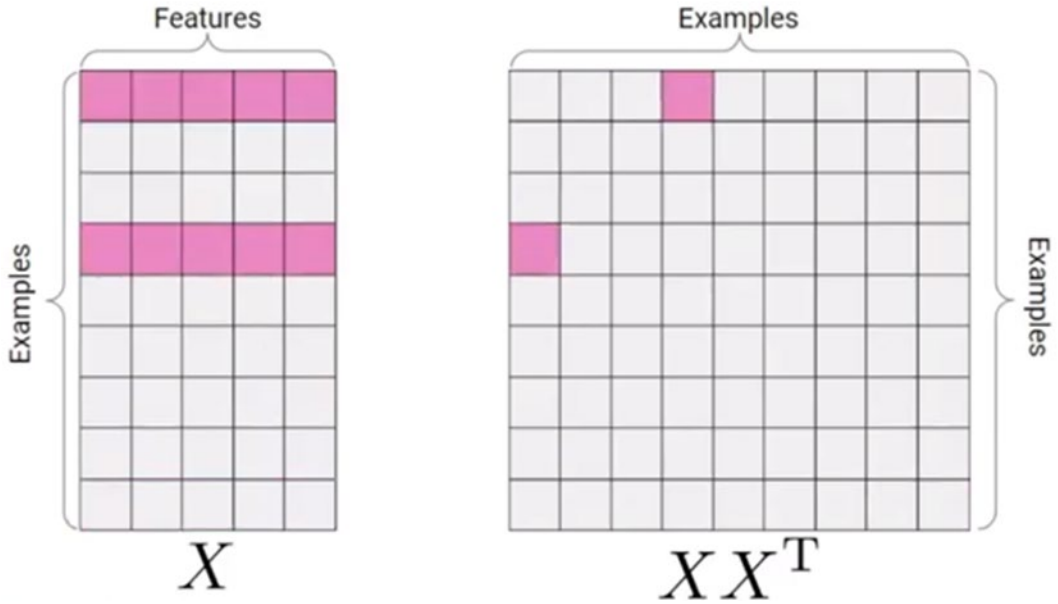

Figure 10. Inter-example similarity matrix

We could then turn XX^T and YY^T into vectors and take the dot product of them. According to [4], it turns out that the "sum of squared dot products between features is the same as the dot product between the reshaped inter-example similarity matrices"

$$\|\mathrm{X}^{\mathrm{T}}\mathrm{Y}\|_2^2 = \langle \mathrm{vec}(\mathrm{XX}^{\mathrm{T}}), \mathrm{vec}(\mathrm{YY}^{\mathrm{T}}) \rangle$$

However, this similarity score is not usable without normalization. Layers of each model have different sizes. If a layer is large, it will lead to a higher score; therefore, a layer of model A might be less similar to a layer of model B but will have a higher similarity score if either layer of the models is large enough. To normalize this score, the geometric definition of the dot product is utilized.

$$\mathrm{A} \cdot \mathrm{B} = |\mathrm{A}||\mathrm{B}| \cos(\theta) \Rightarrow \cos(\theta) = \frac{\mathrm{A} \cdot \mathrm{B}}{|\mathrm{A}||\mathrm{B}|}$$

The cosine of the angle θ between vector A and B will provide us a value in the range of [-1, 1]; however, since the vectors in our research are positive (we do not have negative pixels), $\cos(\theta)$

returns a value between [0, 1]. Therefore, the normalized similarity score could be calculated as follows:

$$\frac{\|\mathrm{X}^{\mathrm{T}}\mathrm{Y}\|^2}{\|\mathrm{X}\mathrm{X}^{\mathrm{T}}\|\|\mathrm{Y}\mathrm{Y}^{\mathrm{T}}\|} = \frac{\langle \mathrm{vec}(\mathrm{X}\mathrm{X}^{\mathrm{T}}), \mathrm{vec}(\mathrm{Y}\mathrm{Y}^{\mathrm{T}}) \rangle}{\|\mathrm{X}\mathrm{X}^{\mathrm{T}}\|\|\mathrm{Y}\mathrm{Y}^{\mathrm{T}}\|}$$

B. Attack Algorithm

1. Loss Function

$$J_{AA}(I_t, I_s) = \|f_L(I_t) - f_L(I_s)\|_2 = \|A_t^L - A_s^L\|_2$$

The target of activation attack is to minimize the difference between the layers of the source and the target images, thus minimizing the loss function J_{AA}. It is simply the Euclidean distance between the activation of the source and target images at a particular layer. For source image I_s, the activation at layer L is $f_L(I_s) = A_s^L$. Similarly, the target image I_t is $f_L(I_t) = A_t^L$.

2. Attack Algorithm

Activation Attack utilizes a variant of the iterative momentum attack. With the use of the signs of momentum, it iteratively modifies each pixel in a small amount until some K steps are achieved. The formula for momentum is as follows:

$$m_{k+1} = m_k + \frac{\nabla_{I_k} J_{AA}(I_t, I_k)}{\left\|\nabla_{I_k} J_{AA}(I_t, I_k)\right\|_1}$$

The momentum is considered the weighted accumulation of gradients, In here, we have $m_0 = 0$ and for iteration k, I_k is the modified source image. Once we have the momentum, the next step is to perturb the source image in the direction of this momentum:

$$I_{k+1} = Clip(I_k - \alpha * sign(m_{k+1}), 0,1)$$

3. Target Selection

The activation attack relies on modifying an image so that its activation becomes similar to the target image. The steps to obtain the targets for the attack are as follows:

- For each class of the input, select 10 samples and save them to a dictionary
- For each input image, randomly select a target class, and return the target sample with the largest distance to the input image of the selected target class

C. Experiment Setup

We conducted our experiments on MNIST and CIFAR-10 datasets. In both datasets, we attacked the pre-trained traditional 2-layer CNN (CNN-2) and 3-layer CNN (CNN-3) (Figure 11), and observed the effect they would have on the other more complex models. We used default parameters as mentioned in [4] except for $\alpha = 0.25$.

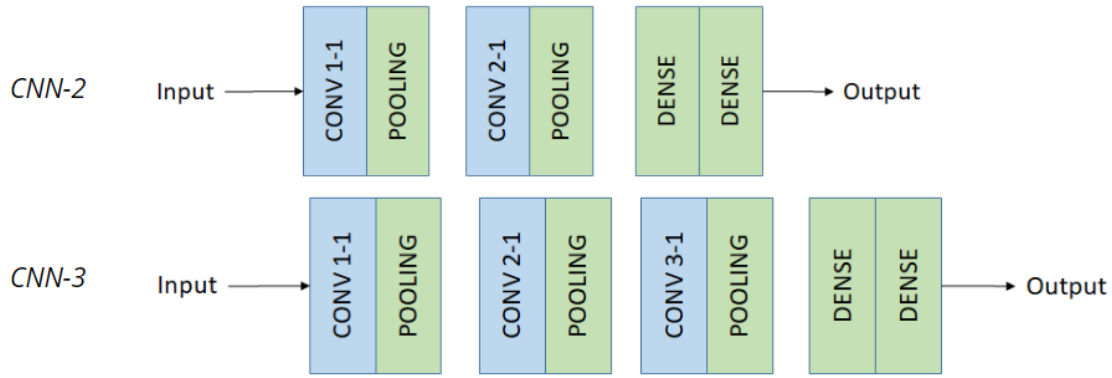

Figure 11. Network architectures used for performing the adversarial attack

The more complex models we used in this experiment were based on VGG (Visual Geometry Group) architecture [9]. This architecture utilizes the stacking of smaller convolutional filters, such as 3x3 instead of the traditional 5x5. Using this stacking nature, we created and pre-trained

CNN-5 for MNIST and VGG-2 and VGG-3 for CIFAR-10 as in Figure 12. For MNIST, we performed transferability test from CNN-2 to CNN-3, and CNN-5, and from CNN-3 to CNN-5. For CIFAR-10, we performed the test from CNN-2 to CNN-3, VGG-2, and VGG-3, and from CNN-3 to VGG-2 and VGG-3.

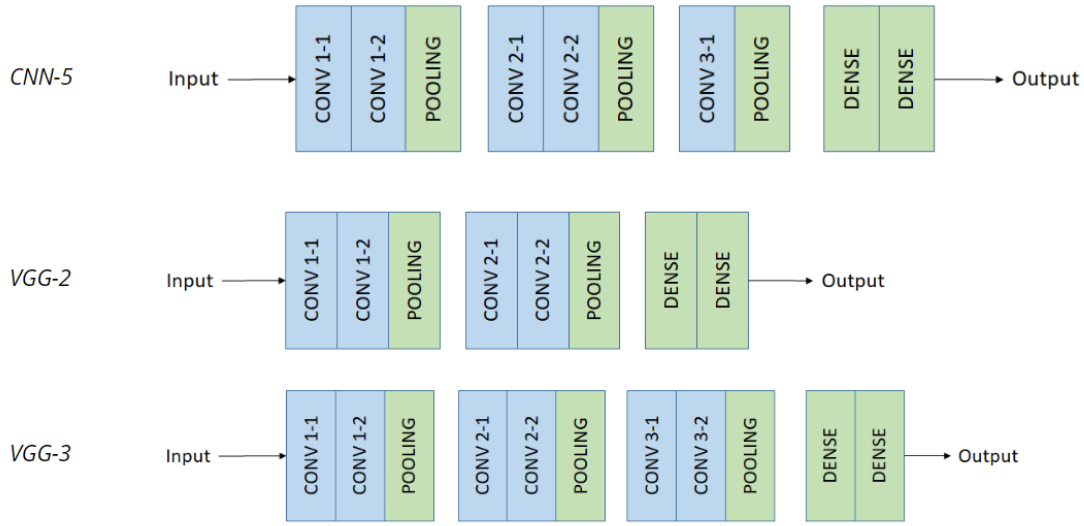

Figure 12. More complex architectures used to observe the effect of transfer attack

We chose to experiment on VGG-based models because they were more complex than the traditional CNN but not as complex as models such as ResNet or GoogleNet; therefore, we could better observe the nature of transfer attack without having to worry about additional factors from the highly complex models that might affect the attack.

CHAPTER IV

FINDINGS

Before the experiment, we predicted that if a layer of model A was more correlated the layers of model B, it would yield better transferability if we attacked that layer comparing to the other layers of model A. In Figure 13, when we summed up the total similarity score for each layer, we obtained a score of **0.3845** for 1st layer, **0.4197** for the 2nd layer, and **4.315** for the 3rd layer. By our prediction, attacking the 3rd layer should yield the highest E_i, then 2nd layer, and 1st layer should come last.

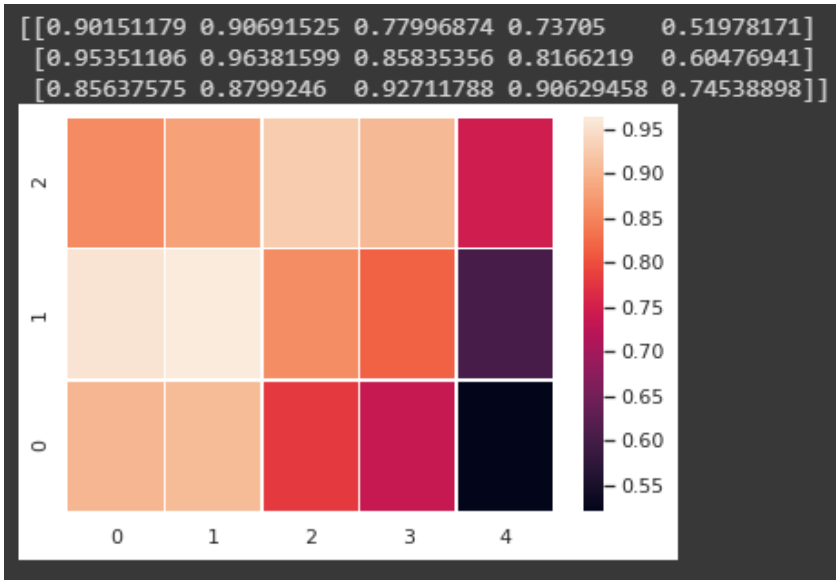

Figure 13. Similarity matrix of CNN-3 and CNN-5

Indeed, in the case of transferring adversarial samples from CNN-3 to CNN-5, the results in Table 1 were consistent with our prediction.

		Layer 1			Layer 2			Layer 3		
	$\mathbf{E_n}$	E_a	E_i	Similarity	E_a	E_i	Similarity	E_a	E_i	Similarity
CNN-3	0.008	0.204	0.196		0.358	0.35		0.534	0.526	
CNN-5	0.027	0.27	0.243	3.845	0.328	0.301	4.197	0.421	0.394	4.315

Table 1. Transferability of CNN-3 to CNN-5

However, that was not the case for all of the scenarios. Results from Table 2 suggested that when transferring from CNN-2 to CNN-3, despite having a larger similarity score (2.745 > 2.557), the attack on layer 1 (0.122) did not perform as well as the attack on layer 2 (0.155).

		Layer 1			Layer 2		
	$\mathbf{E_n}$	E_a	E_i	Similarity	E_a	E_i	Similarity
CNN-2	0.009	0.246	0.237		0.428	0.419	
CNN-3	0.008	0.13	0.122	2.745	0.163	0.155	2.557
CNN-5	0.027	0.294	0.267	4.164	0.304	0.277	4.306

Table 2. Transferability of CNN-2 to CNN-3 and CNN-5

We observed the same behavior on the CIFAR-10 dataset. The results from Table 3 and 4 followed our hypothesis in some cases such as CNN-2 to VGG-2 and VGG3, and did not in the other cases such as CNN-2 to CNN-3, CNN-3 to VGG-2 and VGG-3.

		Layer 1		Layer 2			
	$\mathbf{E_n}$	E_a	E_i	Similarity	E_a	E_i	Similarity
CNN-2	0.449	0.588	0.139		0.68		
CNN-3	0.443	0.554	0.111	2.682	0.613	0.17	2.677
VGG-2	0.424	0.533	0.109	3.281	0.568	0.144	3.629
VGG-3	0.403	0.559	0.156	4.454	0.566	0.163	5.211

Table 3. Transferability of CNN-2 to CNN-3, VGG-2, and VGG-3

		Layer 1			Layer 2			Layer 3		
	$\mathbf{E_n}$	E_a	E_i	Similarity	E_a	E_i	Similarity	E_a	E_i	Similarity
CNN-3	0.443	0.563	0.12		0.652	0.209		0.725	0.282	
VGG-2	0.424	0.557	0.133	3.161	0.538	0.114	3.591	0.521	0.097	3.539
VGG-3	0.403	0.545	0.142	4.242	0.55	0.147	5.186	0.53	0.127	5.269

Table 4. Transferability of CNN-3 to VGG-2 and VGG-3

One possible reason for this inconsistent behavior that we believe is the significance of individual layers. In our experiments, we considered the similarity of a layer of model A to all of the layers in model B. However, the correlation of individual layers from model B might play a role in the transferability of the attack. For example, considering a 2-layer CNN (CNN-2) and a 3-layer CNN (CNN-3). Assume that the total similarity of layer 1 of CNN-2 to all of the layers in CNN-3 is higher. If the correlation of layer 1 of CNN-2 to layer K of CNN-3 is less than the correlation of layer 2 to layer K, the attack on layer 2 may yield better results because the correlation to layer k is more significant compared to the other layers. Further research is necessary to study this behavior.

Another interesting behavior we observed in our results is that deeper layers tend to provide more error for the attack. We believe a reason for this is the nature of deep neural networks, where class-specific features extracted in the deeper layers are more sensitive compared to the general features extracted in the early layers.

CHAPTER V

CONCLUSION

Our experiment contributed a deeper view into the transferability of adversarial attacks by extending the work of [3] and [4]. We explored the assumption that similar features lead to the transferability of adversarial attacks. The hypothesis that the higher similarity would yield a more efficient attack was used as the basis for our experiments. We found that the results were sometimes consistent and sometimes not consistent with our guess. Further research in the significance of individual layers is necessary to complete the view of the picture of this assumption. We also noticed that attacking the deeper layers tends to yield a more efficient attack.

REFERENCES

1. Naveed Akhtar and Ajmal Mian, Threat of adversarial attacks on deep learning in computer vision: A survey, 01 2018.

2. Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, Explaining and harnessing adversarial examples, CoRR abs/1412.6572 (2014).

3. Nathan Inkawhich, Wei Wen, Hai (Helen) Li, and Yiran Chen, Feature space perturbations yield more transferable adversarial examples, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

4. Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton, Similarity of neural network representations revisited, Proceedings of the 36th International Conference on Machine Learning (Long Beach, California, USA) (Kamalika Chaudhuri and Ruslan Salakhutdinov, eds.), Proceedings of Machine Learning Research, vol. 97, PMLR, 09–15 Jun 2019, pp. 3519– 3529.

5. Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami, Practical black-box attacks against machine learning, Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (New York, NY, USA), ASIA CCS 17, Association for Computing Machinery, 2017, p. 506519.

6. Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, Intriguing properties of neural networks, International Conference on Learning Representations, 2014.

7. Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, How transferable are features in deep neural networks?, Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (Cambridge, MA, USA), NIPS14, MIT Press, 2014, p. 33203328.

8. Povolny, Steve, and Shivangee Trivedi . "Model Hacking ADAS to Pave Safer Roads for Autonomous Vehicles." McAfee Blogs, 19 Feb. 2020, www.mcafee.com/blogs/other-blogs/mcafee-labs/model-hacking-adas-to-pave-safer-roads-for-autonomous-vehicles.

9. X. Zhang, J. Zou, K. He and J. Sun, "Accelerating Very Deep Convolutional Networks for Classification and Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 10, pp. 1943-1955, 1 Oct. 2016.

10. Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, & Justin Gilmer. (2017). Adversarial Patch.

11. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 427–436

12. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. arXiv preprint arXiv:1610.08401 (2016)

13. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. arXiv: Computer Vision and Pattern Recognition (2016)

14. F. Tramer, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. ` McDaniel. The space of transferable adversarial examples. CoRR, abs/1704.03453, 2017

15. N. Papernot, P. D. McDaniel, and I. J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. CoRR, abs/1605.07277, 2016

VITA

Hai Huynh

Candidate for the Degree of

Master of Science

Thesis: INVESTIGATE THE EFFECT OF NEURAL NETWORK REPRESENTATIONS ON THE TRANSFERABILITY OF ADVERSARIAL ATTACKS

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in May, 2020.

Completed the requirements for the Bachelor of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in 2018.

Publications:

J. Cecil, P. Ramanathan, and H. Huynh. “A Cloud-Based Cyber-Physical Framework for Collaborative Manufacturing, Accepted for Presentation and Publication,.” the Proceedings of the 47th SME North American Manufacturing Research Conference, NAMRC 47, , Erie, PA, (2019): n. pag.

J. Cecil, R. Krishnamurthy, H. Huynh, O. Tapia, T. Ahmad and A. Gupta, "Simulation Based Design Approaches to Study Transportation and Habitat Alternatives for Deep Space Missions," 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 2018, pp. 1439-1444.