On grey-box adversarial attacks and transfer learning

Vinay Uday Prabhu*
UnifyID Inc
San Francisco, 94107
vinay@unify.id

John Whaley
UnifyID Inc
San Francisco, 94107
john@unify.id

ABSTRACT

In this short paper, we disseminate some initial results on experimentally validating the potency of adversarial attacks in the context of transfer learning. In this "grey-box" framework, we assume that while the attacker indeed has complete knowledge about the pre-trained DNN-Deep Neural Network (the white-box aspect), this DNN is being used not to classify but to only generate features which are then used to train a possibly shallow *local* classifier by the target user that is not accessible to the attacker (the black-box aspect). Specifically, we investigate the efficacy of the Carlini-Wagner-l2 attack on a Tshirt-versus-Trouser image binary SVM classifier that uses a pre-trained DNN trained on the MNIST dataset to generate the features which were then used to train the local SVM. We used the CleverHans library to generate the adversarial images and have duly open-sourced the experiments to facilitate reproducibility.

Keywords

Deep learning, adversarial attacks, transfer learning

1. INTRODUCTION

In this short paper, we would like to report some initial results from our investigations on the effectiveness of adversarial perturbations in the context of transfer learning. While our initial investigations did reveal that the attack that we chose to experiment with did not succeed in the specific transfer-learning scenario considered, our narrative is not to extrapolate this result as a proof of failure of the attack model, but to invite the community to closely scrutinize this specific transfer-learning regime which we believe is very relevant and rather ubiquitous.

In the sub-sections below, we begin by motivating the ubiquity of transfer learning and provide some background on adversarial attacks.

1.1 Background on transfer learning

The ubiquity of *model zoos* [3, 2, 4] that host pre-trained Deep Neural Network (DNN) models that achieve state-of-the-art (SotA) accuracy in their respective domains and the steep expenses associated with original large-scale dataset curation has resulted in fewer instances where the entire DNN is trained from scratch. The strategy of harnessing a pre-trained DNN that was trained on a very large dataset (e.g. ImageNet, which contains 1.2 million images sampled from 1000 categories) and then using this DNN as a feature extractor is a technique that has achieved considerable success [15, 1]. In fact, most Deep Learning (DL) frameworks come preloaded with many such models. For example, Keras [6] comes pre-loaded with Xception, VGG16, VGG19, ResNet50, InceptionV3, InceptionResNetV2 and MobileNet models.[1]

In the security domain, deep learning applications that seek to identify a person on the basis of passive biometrics such as gait rely on training a k-user classifier DNN using data from a curated test-user cohort and then deploying the thus-trained DNN as a feature extractor. The extracted features are then used to locally train a local shallow classifier using standard outlier detecting typicality modeling tools such as Iso-Forests, one-class SVM and Gaussian Mixture Models [9, 7].

In Figure 1, we provide one such example which will be used in the rest of the paper. The goal of this classifier is to classify images as those containing *T-shirts* or *Trousers* with transfer learning using a DNN pre-trained on the MNIST dataset. The 28×28 images for T-shirts and Trousers are extracted from the Fashion-MNIST [16] dataset.[2]

1.2 Background on adversarial attacks

The study of adversarial attacks on Machine Learning (ML) systems, especially those incorporating DNNs, has attracted a lot of recent interest.

An *adversarial attack* entails generating adversarial inputs by strategically introducing small perturbations that make the resultant adversarial input example only slightly different from correctly classified examples drawn from the data distribution, but at the same time resulting in a potentially controlled misclassification [14]. In the context of computer vision (CV) models, the adversarial perturbations are often imperceptible to the human eye.

Of much concern is also the revelation that a variety of

*Primary author

[1] https://keras.io/applications/#usage-examples-for-image-classification-models

[2] https://github.com/zalandoresearch/fashion-mnist

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

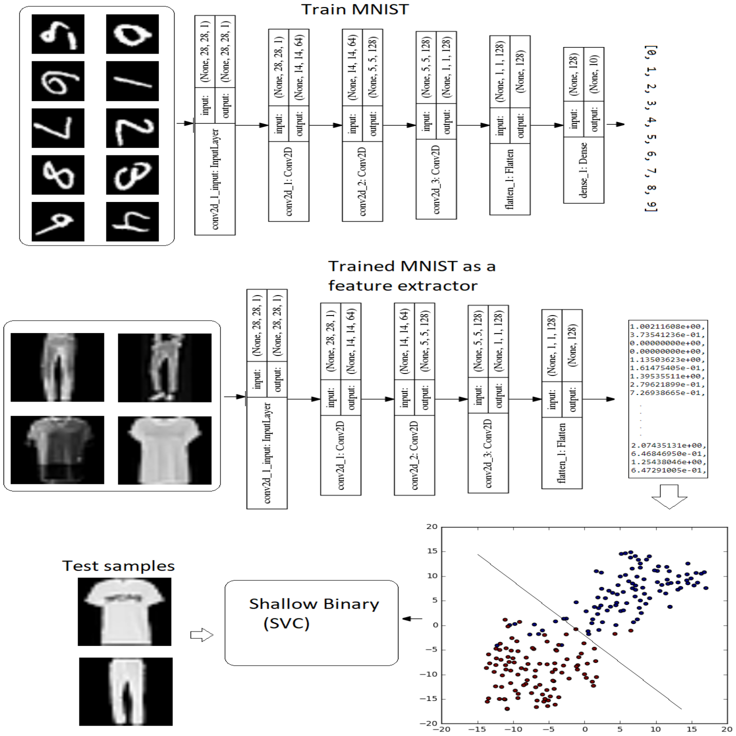

Figure 1: Classifying T-shirts and Trousers with transfer learning using a DNN pre-trained on the MNIST dataset

the state-of-the-art (SotA) DNN models with disparate architectures, trained on different subsets of the training data, have been found to misclassify the same adversarial example, uncovering the presence of *fundamental blind spots* in our DNN frameworks. After this discovery, several works have emerged ([13, 10]), addressing both a means of *defense* against adversarial examples, as well as novel attacks.

1.3 Premise of the paper

The premise of the paper is as follows. As evinced in [8], we see that there are two kinds of adversarial attacks studied in this field. First, the black-box attacks ([12, 11]) where the adversary possesses no knowledge of the model type or any distribution information about training data. (Note that the adversary does have the ability to observe the labels output by the DNN to the chosen inputs.) The second kind are the white-box attacks where the adversary has complete knowledge of the model (the architecture and the weights) being targeted.

Keeping in mind the ubiquity of using transfer learning as characterized in the first subsection, we'd like to explore a third kind, which we term as *grey-box attacks*, where the attacker has knowledge of the *parent* DNN (both the architecture and weights — the white-box component) but the deployed system uses the parent DNN as a feature extractor which are then fed into a *local* and oft-shallow classifier which is not accessible to the attacker (the black-box component). This can be extended to other related scenarios where the *parent* DNN's first few layers are *frozen* and only the last few layers are retrained using a *local* dataset.

Figure 2 showcases the grey-box-attack scenario for the

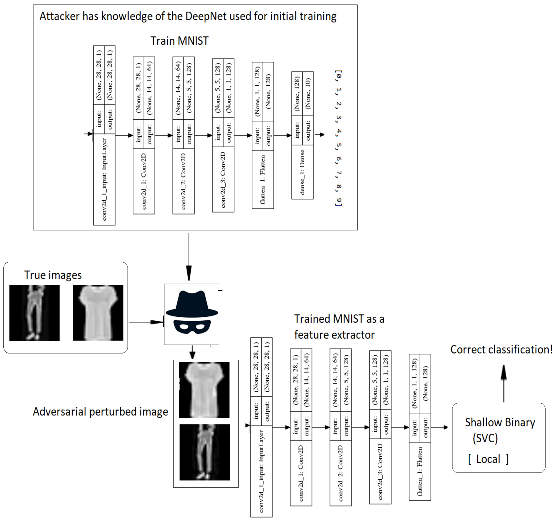
Attacker has knowledge of the DeepNet used for initial training
Train MNIST
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
True images
Trained MNIST as a feature extractor
Correct classification!
Adversarial perturbed image
Shallow Binary
(SVC)
[Local]

Figure 2: grey-box-attack for the T-shirts/Trouser image classification problem that uses a parent DNN trained on the MNIST dataset

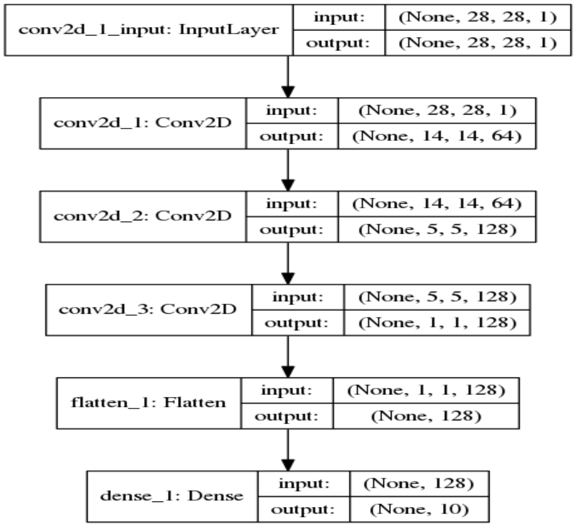

Figure 3: The DNN trained with the MNIST dataset

T-shirts/Trouser image classification problem that uses a parent DNN trained on the MNIST dataset that is as introduced in the subsection above. In this paper, we launch a preliminary investigation trying to seek an answer to the following query: *What happens when adversarial perturbations generated by an attacker are tailored for a specific (parent) DNN architecture, but the parent DNN architecture is being used only as a feature extractor and the extracted features are fed into a local (possibly) shallow classifier that is unknown to the attacker?* In the following section, we describe the specifics of the procedure followed and the results obtained.

2. PROCEDURE AND RESULTS

In this section, we will describe the five step procedure that we followed and the results.

2.1 Training the parent CNN with the MNIST dataset

We trained a simple 3-layer CNN as shown in Figure 3 with the Categorical crossentropy cost function using the Adam optimizer with batch size=128 for 6 epochs. We obtained 98.7% accuracy of the test dataset.

2.2 T-shirt/Trouser dataset extraction

The Fashion-MNIST dataset has 70000 28×28 grayscale images of clothes that belong to the following 10 categories: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag and Ankle boot. In this paper, we formulated a binary classification to differentiate between T-shirts and trousers by extracting 600 random samples each of images belonging to the first 2 categories in the Fashion-MNIST datasets. The train-test split is 500-to-100.

2.3 Binary classification by transfer learning

Our transfer learning based binary classifier is described in Figure 1. As seen, the images are fed into the CNN trained on the MNIST dataset and the features are extracted by extracting the 128-dimensional features from the pre-softmax layer of the CNN. A natural question that arises now is: Are these features discriminative enough? This is answered in Figure 4 which showcases the 2-D t-SNE (perplexity=25) scatterplot visualization of the 128-dimensional features extracted from the CNN for 200 random samples from the Fashion-MNIST dataset. The red points pertain to the T-shirt images while the blue ones belong to the Trouser class. The *deep* features thus extracted are then used to learn a *local* SVM model with a linear kernel that achieves class-wise accuracies of 97% and 98% respectively.

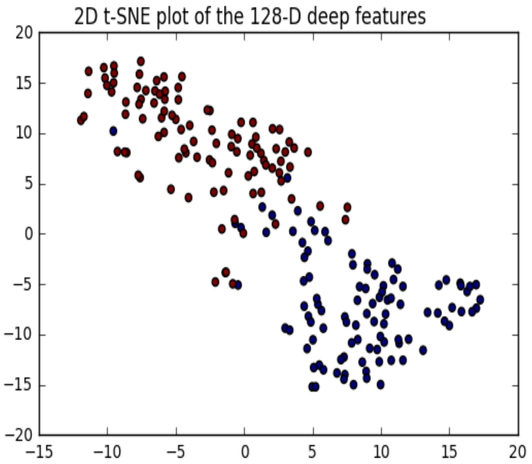

Figure 4: 2D t-SNE visualization of the 128-dimensional features extracted from the CNN trained on the MNIST dataset.(perplexity=25)

2.4 Generating adversarial perturbations using the Carlini-Wagner-l_2 procedure implemented in the CleverHans library

In order to ensure repeatability of the results, we use the `cleverhans` software library [10] that provides standardized reference implementations of adversarial example-construction techniques and adversarial training as well as providing standardized benchmarks of model performance in the adversarial setting explained above. Specifically, we chose the `CarliniWagnerL2` class (see [5] for the attack details) in the attacks module with the following configuration parameters:

```
{cw_params = {'binary_search_steps': 1,
              'y_target': None,
              'max_iterations':10000,
              'learning_rate': 0.9,
              'batch_size':200,
              'initial_const':0.5,
             'clip_min' :0,
             'clip_max':1}}
```

This resulted in successful generation of adversarial examples on 200 of 200 test images for the 2-class Fashion-MNIST dataset with a mean successful distortion of 0.9314.

2.5 Effect of the adversarial perturbations as seen at the output of the feature extractor

As seen in Figure 5, the two sets of examples pertaining to the true image and the adversarial image look visually indiscriminate with respect to each other. However, when

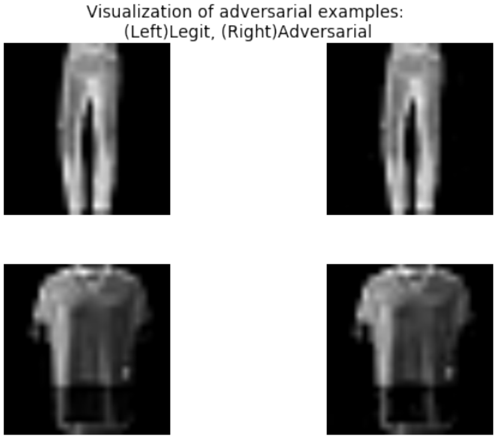
Visualization of adversarial examples:
(Left)Legit, (Right)Adversarial

Figure 5: Visualization of the true and adversarial examples

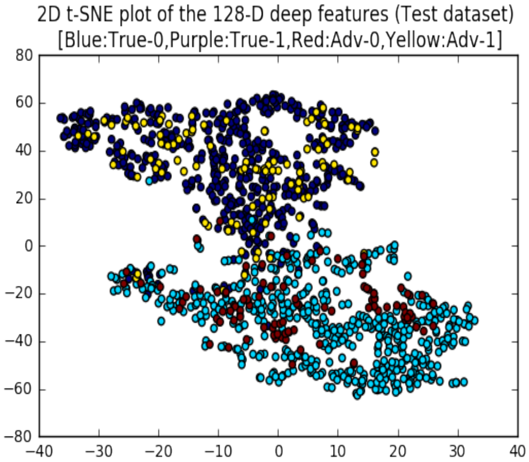

Figure 6: 2D t-SNE plot of the 128-D deep features for the true and adversarial test samples

these images are passed through the CNN and the 128-D features are processed via a 2D-tSNE visualizer, the scatterplot that emerges paints a completely different picture. In Figure 6, we see the 2D-tSNE scatterplot of 1200 features of which 1000 belong to the true training samples and 200 belong to the adversarial test samples. We clearly see that the points pertaining to the adversarial perturbations are mostly co-embedded clustered next to the ones belonging to the same class that they were supposed to misclassify.

When the features belonging to the adversarial inputs are now tested with the SVM that was trained using the clean input samples, we get class-wise accuracies of 96% and 97% respectively. That is, precisely one test sample from each class that was crafted to be adversarial retained its potency.

3. CONCLUSION AND FUTURE WORK

In this paper, we introduce the grey-box attack framework to understand the potency of adversarial perturbations in the context of transfer learning. Specifically, we investigated the Carlini-Wagner-l_2 un-targeted attack on a binary classifier of images where the attacker has access to the CNN used, except that the CNN is being used as a feature extractor and the extracted features are in turn being used to train a (shallow) SVM which is inaccessible to the attacker. We discovered minimal loss of accuracy with regards to a naïve test accuracy benchmark. This dissemination is clearly a work-in-progress and we are currently investigating whether this seeming loss of adversarial potency is repeated under different transfer learning methodologies, attack algorithms and datasets. We have duly open-sourced the implementation of this paper at the following link: https://github.com/vinayprabhu/Gainsboro-box-attacks-[3]

4. REFERENCES

[1] http://cs231n.github.io/transfer-learning/. 2017.

[2] https://deeplearning4j.org/model-zoo. 2017.

[3] https://github.com/bvlc/caffe/wiki/model-zoo. 2017.

[4] https://mxnet.incubator.apache.org/model$_{z}oo$/.2017.

[5] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.

[6] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[7] M. Gadaleta and M. Rossi. Idnet: Smartphone-based gait recognition with convolutional neural networks. *Pattern Recognition*, 74:25–37, 2018.

[8] A. Kumar and S. Mehta. A survey on resilient machine learning. *arXiv preprint arXiv:1707.03184*, 2017.

[9] N. Neverova, C. Wolf, G. Lacey, L. Fridman, D. Chandra, B. Barbello, and G. Taylor. Learning human identity from motion patterns. *IEEE Access*, 4:1810–1820, 2016.

[10] I. G. e. a. Nicolas Papernot, Nicholas Carlini. cleverhans v2.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2017.

[11] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[12] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint arXiv:1602.02697*, 2016.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[14] 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.

[15] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016.

[16] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. 2017.

[3] Gainsboro is a light shade of grey with RGB values of (220,220,220)