Exploring DIVA (DIfferential eVasive Attack) attack on Edge Models

Alekhya Pyla *Vignatha Yenugutala* *Goda Devi Addanki*

Abstract

The deep learning models are generally huge in size to be deployed on edge devices. To enable this, the models are adapted to the edge devices via various techniques such as quantization, pruning, etc. The adapted models though don't affect the top-line accuracy by much and can bring a difference in predictions compared to the original mode. In this paper, we address an attack called DIVA which exploits these differences between adapted and original models without being undetectable at the original model. The attack will be performed on 3 models of MobileNet, DenseNet, ResNet models using the CIFAR/MNIST datasets under a white box setting(with the assumption that there is access to both the original and edge-adapted model) and their results will be analyzed.

1 Introduction

The Edge devices are growing exponentially in billions. With Deep learning getting popular among edge devices, it is becoming increasingly necessary for DNN inference to be performed on edge devices to avoid network latency and congestion. But DNNs are complex and require powerful hardware. Edge devices have the constraints of limited memory, bandwidth, and computing. So, DL models are tuned down to the limited resources of edge devices, known as adapted models. Adapted models are models reduced in size from the original model through methods such as quantization, pruning, and distillation. But, these adaptation methods might result in minor inaccuracies compared to the original model. Even though the inaccuracies seem insignificant, the results of the original and adapted can be different for a given input. This can become a vulnerability and be used for attacks on edge devices. In this paper, we try to address a vulnerability of a similar kind on these adaptive models.

In edge-distributed DL systems, the original model is trained on servers with hardware resources like TPUs. The trained model is converted to a reduced model and pushed to edge devices for inference. Rigorous testing is typically done on the original model, but the attacks are targeted at the reduced models. The attackers alter the input to influence the adapted models' results while the original model's output remains unaffected. These attacks are hard to tackle for two reasons. One - It goes undetected for a long time, as actual results are unaltered, so inputs are not doubted. Two - The adapted models are pushed onto millions of edge devices, so debugging them is difficult and time-consuming.

The following are the methods used for building adaptive models.

- **Quantization**: In this method, DL models use 32 or 64-bit floating point representation during the training phase. But this requires higher memory and resources. So, adapted models use 8bit representation to reduce the size and latency. Thus reducing the size can lead to some inaccuracies in inference.
- **Pruning**: In pruning, the least essential weights are discarded a sparse model is generated. This technique is widely used in vision applications.
- **Distillation**: In distillation, a smaller model known as the student model is trained to match the outputs of the original model. Even though the resultant model is smaller, it has all the required information from the original model and has similar results during inference.

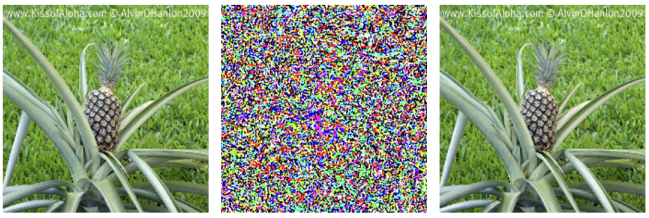
However, due to the inaccuracies that come with these adaptive models, they are very vulnerable to evasive attacks. To reduce the inaccuracies and vulnerability to evasive attacks, a few methods were discovered. Some of them are described below.

- **Quantization-aware training (QAT)**: In this method, an adapted model is trained with quantization noise to increase the robustness. In the forward pass of the training, numbers are adapted to reduced size i.e. 8bit representation. In the back-propagation, they are brought to the original representation to increase the accuracy.
- **Robust Training**: Robust Training is a minimax algorithm that tunes the model parameters to minimize the loss induced by perturbations. But these require high-performance hardware. Hence, they are applied to original models.

Despite the defensive methods mentioned above, evasive attacks can be done on the system by introducing noise and producing different results in original and adapted models. In this project, we explored one such attack called DIVA (DIfferential eVasive Attack).

2 Motivation

Generally, companies that deploy models on edge devices need to deal with the huge diversity in hardware on different phones, tablets, and cameras. For example, Facebook estimates that devices running its deep learning models comprise



(a) Original image. (b) Attack noise. (c) Attacked image.

Figure 1: Example of the attacked image

over 2000 unique SoCs running on tens of thousands of tablet and phone models. To accommodate such a large set of devices, for each one of their full-precision models, ML operators may need to create thousands of edge-adapted model versions. Most ML operators use the original version of the model for validation and debugging as both the models are similar in top-line accuracy and as there are a huge number of adapted models deployed. Thus, attacks that target a particular edge model but not the original model would be harder to detect.

In general, quantization does not significantly affect the topline average accuracy but introduces a subtle variance in prediction for individual inputs. Adversarial attacks are likely to impact other models if they are capable of impacting one. Thus, if an attacker targets a quantized model using a standard attack, it is likely to also affect the original model.

Projected Gradient Descent(PGD) is one of the state-of-the-art adversarial attacks today. The instability between the predictions of original and quantized models attacked by PGD is still not very high and hence can be validated by the original model. Thus DIVA, an attack that has higher instability in between the predictions of original and quantized models would be more challenging to detect. In this project, we plan to explore DIVA on different types of attacks and compare its performance with PGD.

3 Related Work

Attacks on DL models mainly involve generating perturbations in inputs that alter the model predictions. Attacks are divided into four types based on the attackers' level of access to the resources and threat model.

- **White box attack**: Attackers have full access to original and adapted models in white box attacks. In this attack, the attackers can alter model parameters by introducing noise. Algorithms are developed to generate perturbations that maximize the model's loss function. One such method is the Fast Gradient Sign Method (FGSM). R+FGSM Tramèr et al. (2018) is an improvement for FGSM. Another algorithm that improves FGSM further is Projected Gradient Descent PGD Madry et al. (2018).PGD is a widely known adversarial attack. The idea in PGD is to convert the one-step FGSM attack into multiple steps. PGD is determined by three parameters: the number of steps, the step size, and the perturbation strength. It is used as a baseline for comparison of DIVA attacks.

- **Black box and Semi-black Box attack** : In black box attacks, attackers have no access to the internals of the models. So, these types of attacks typically develop a surrogate model for both original and adapted models such as in Papernot et al. (2016). Generally, the model weakness is transferred to surrogate models, which helps to attack the actual model. There are also some recent works in improving black box attacks such as Feng et al. (2022). Furthermore, a semi-black box attack is where the attackers have access to one of the models i.e., the adapted model, which is used to generate a surrogate model.

- **Bit-Flip attack** :

  This attack as in Rakin et al. (2019) targets the memory of edge devices by flipping the values in memory exploiting the condition that edge devices usually have lesser sophisticated security checks. So, attackers change the model weights in the memory and also inputs.

- **Differential Testing** :

  Differential testing is one of the widely used methods in software engineering where the same input is given to two different versions of the same method and input is mutated until there is a considerable difference between both methods. Pei et al. (2017) first applied this idea to Deep Learning models and later works such as Guo et al. (2018). This idea is applied to deferentially attack the adapted model by bringing perturbations to the input keeping the original intact.

4 Design

In this project, we are implementing white box DIVA attack for the original and adapted models. The attacker has full access to both the original and adapted models in the white-box attack. the attacker can generate Adversarial samples are generated by solving an optimization problem that generates additive noise and maximizes the loss function. In case of DIVA, the loss function jointly considers both the adapted and the original models.

$$L_{DIVA}(\theta, x+A, y) = model_{orig}(x)[y] - c.model_a(x)[y] \quad (1)$$

Here, $model_{orig}(x)[y]$ is the raw probability of input x for label y in the original model's prediction, and $model_a(x)[y]$ is that for the adapted model. Hyper-parameter c balances the two probabilities, and is set by default to c = 1. c represents a trade

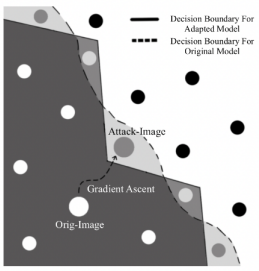


Figure 2: DIVA Attack

off between how well DIVA evades the original model and how well it attacks the adapted model. Loss function L_{DIVA} captures the difference between the raw probabilities of the two models. θ refers to the parameters of both original and adapted models.

$$A = \underset{A \leq \varepsilon}{\operatorname{argmax}} L_{DIVA}(\theta, x + A, y) \tag{2}$$

This equation can be solved by using stochastic gradient descent optimization technique as shown in 3 where α is the step size. The perturbation A is clipped to be within values of -ε and ε post the update.

$$A = A + \alpha * \frac{\partial L}{\partial \theta} \hat{x} \tag{3}$$

Hyperparameter C in equation(1) defines the relation between two loss functions in equation and has a high prominence in the equation. With lower values of C, attacks are more successfully. But, attacks get detected easily by original model. So, tradeoff between evading detection and attacking successfully is decided by value of C. We selected C value as 1 for all our experiments.

The rest of the parameters used in attack experiments are: ε value is 8. Step size α is 1 and number of maximum steps is 20.

5 Experiments

Experiments were performed using the DIVA attack for CIFAR10 and MNIST datasets. Initially, the attack was tried on ImageNet dataset but since the dataset required high memory and computation resources, CIFAR10 and MNIST is chosen. Both CIFAR10 and MNIST datasets are available in keras datasets and have images of shape - 32*32*3.

Keras provides pretrained neural network models trained on imagenet datasets. We took these as base models and finetuned them to the CIFAR10 and MNIST datasets. Quantized models with 8-bit representations were trained for creating adapted models. Now, with both the original and quantized models, we calculated the loss function following

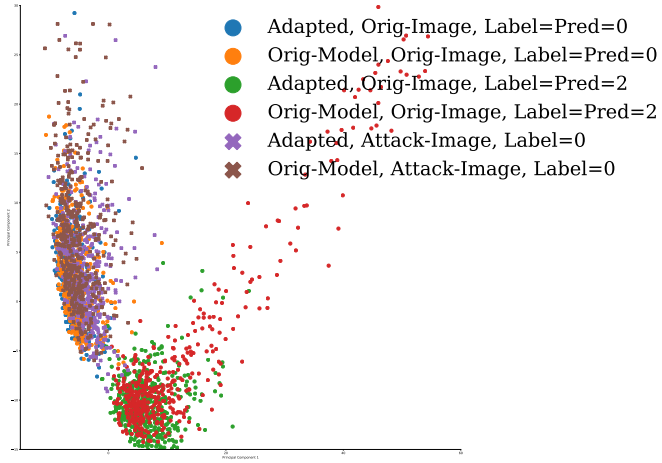


Figure 3: PCA visualization of representations learned by the original and adapted models, using the two most principal components on MNIST.

(1). Taking the upper bound (ε) of perturbation (*A*) to be added to the input image, the loss function was maximized according to (2) until the original and quantized models predictions are different. These images are called attack images and can be used to have different predictions on quantized models from that of original models.

5.1 DIVA for MNIST

The MNIST database (Modified National Institute of Standards and Technology database[1]) is a large database of handwritten digits that is commonly used for training various image processing systems. It contains 60,000 training images with 10 classes.

5.1.1 ResNet

The penultimate activation layer representations of the images in the original and adapted models were taken and their first two principal components were plotted in the 3. The plot shows how DIVA's generated adversarial noise affects both the original and adapted models. The principal component representations were taken from 1000 samples of Resnet50 where the original and adapted models classify as 0 and 2 labels for the MNIST hand-written digits data.

In the figure, Adapted, Orig Image, Label=Pred=0 shows the representations of adapted model on original images where original and predicted labels are 0. Attack image shows the representations of images generated by DIVA's adversarial noise. Even on the original images, there is a subtle difference in the representations learned by the adapted and the original model. The figure shows how DIVA shifts the representations for original and adapted models. From the purple crosses, we can observe how DIVA shifts the images belonging to 0 label

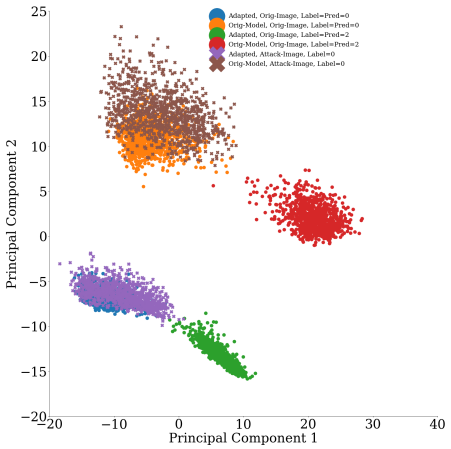


Figure 4: PCA visualization of representations of DenseNet on MNIST.

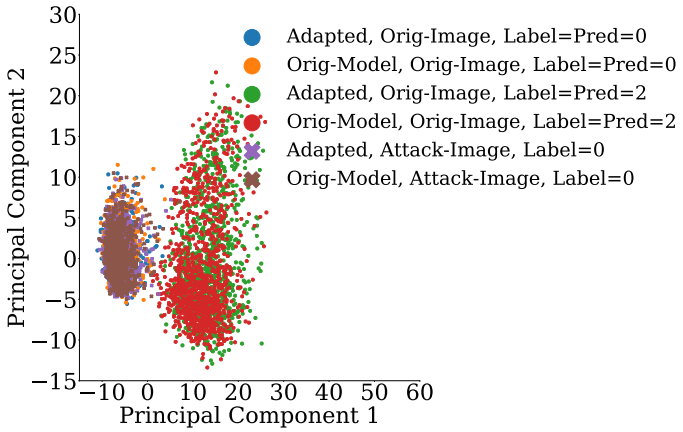


Figure 5: PCA visualization of representations learned by the original and adapted models, using the two most principal components on MNIST.

to the bottom part. For the original model, DIVA also shifts the points more towards to the bottom of the graph, but less so than it does for the adapted model, thus preserving the prediction of the correct label for most of them.

5.1.2 DenseNet

The same process followed for creating original and adapted models for Resnet were followed for Densenet except that in densenet, moving average quantization was followed. This is because the BatchNormalization layer in Densenet model has to be quantized seperately. The PCA representations learned by the original and adapted models are plotted in Figure 4.

5.1.3 MobileNet

MobileNet is a light weight CNN Model, meant for mobile and embedded devices. They have low latency. We trained MNIST data with mobilenet model with same process followed for the Resnet and Densenet models. Mobilenet models train faster, so we used more epochs to get higher accuracy for original and adapted models.

5.2 DIVA for CIFAR10

The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. Different models have been finetuned for DIVA on CIFAR10 dataset as shown below:

5.2.1 ResNet

ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks.

Figure 6: Training Accuracy for ResNet50 on CIFAR10

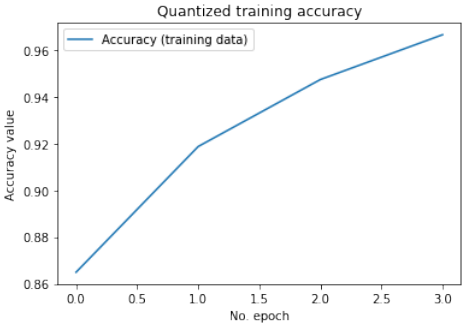


Figure 7: Training Accuracy for Quantized ResNet50 on CIFAR10

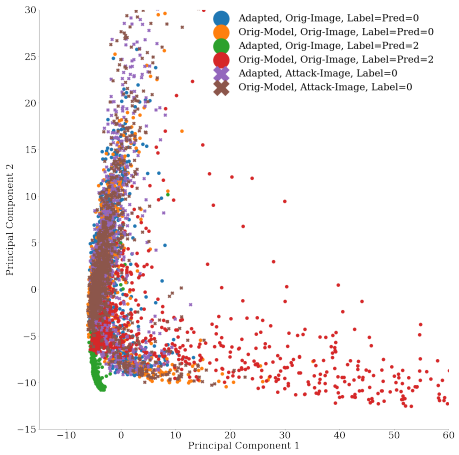


Figure 8: PCA visualization of representations learned by the original and adapted models, using the two most principal components on CIFAR10.

ResNet-50, a convolutional neural network that is 50 layers deep is used in the experiment.

Two models are finetuned, one is the original model and the other is the quantized model. The quantized model is generated by applying TensorFlow Model Optimization tfmot's quantize model on the original models using int8 quantization. Then QAT is performed to obtain the final quantized model. The training accuracy and loss for both the models can be found in 6 and 8.

Post the quantization, DIVA attack is performed on the quantized and original models with 4000 images from training set. The Principal Components for both the original and quantized models are calculated similar to MNIST as shown in

5.2.2 DenseNet

Similar quantization method as MNIST is applied to CIFAR10 dataset. The attack is done on 1000 of the input images and the principal components for the attack are shown in Figure 9.

5.2.3 MobileNet

Similar quantization method as MNIST is applied to CIFAR10 dataset. The attack is done on 2000 of the input images and the principal components for the attack are shown in Figure 10.

5.3 ImageNet Dataset Challenges

DIVA research paper is mainly based on ImageNet dataset. The original ImageNet data set has 60,000 images and it is about 150GB. We faced challenges to load the data and train the model with it. We took a subset of data , covering all the

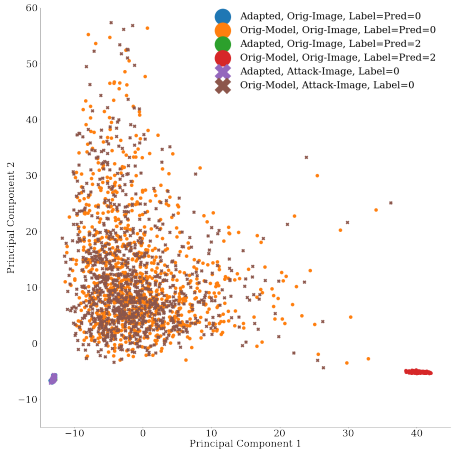


Figure 9: PCA visualization using the two most principal components on CIFAR10 for DenseNet

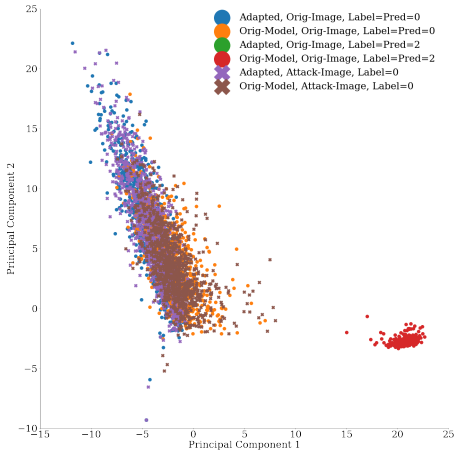


Figure 10: PCA visualization using the two most principal components on CIFAR10 for MobileNet

classes consisting of around 5000 images. But, the accuracy is degraded. So, we decided to use different datasets.

During our experiments, it was observed that support for recursive Quantization for Keras models is not yet available.

5.4 Experiment results

The accuracy results of all the models with the two datasets are as shown in the Table 1 and Table2. It is observed that the quantized models got higher accuracies for the same number of epochs, which could be attributed to the reduced complexity matching the needs of the task.

Model	Orignal model(%)	Adapted model(%)
Resnet	83.17	96.67
Densenet	81.2	88.83
Mobilenet	75.7	81.2

Table 1: Model accuracy for CIFAR10 dataset

Model	Orignal model(%)	Adapated model(%)
Resnet	97.7	97.03
Densenet	99.59	99.55
Mobilenet	98.9	99.7

Table 2: Model accuracy for MNIST dataset

6 Discussion

The t-distributed stochastic neighbor embedding, (statistical method for visualizing high-dimensional data, by giving each data point a location in a two or three-dimensional map) drawn for the last layer representations of the images in original and adapted models is shown in Figure 11. It can be observed that the TSNE representations of label-0 attack images with quantized model align with that of label-1 representations, thus getting predicted as label 1. Thus, the last layer representations of attack images gets changed for original and adapted models, thus resulting in different predictions.

7 Conclusion

Varied versions of adapted models deployed on the edge devices create new security concerns where there is a subtle difference between the edge-adapted model and the original model. This new attack called DIVA formulates an adversarial noise that maximizes the loss between the prediction of both the original and adapted models. It causes the adapted model to predict different classes while the initial model prediction remains the same. This is performed under Whitebox setting for DenseNet, MobileNet, and ResNet models for CIFAR10 and MNIST and also by constructing the quantized models

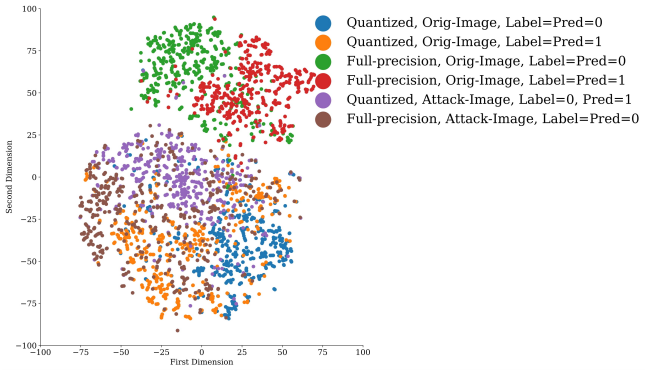


Figure 11: TSNE of MNIST images in original and adapted models

using Quantized Aware Training. It is observed that ResNet proved better among the 3 models for both datasets.

Future lines of work in this direction can be to tune the hyper-parameters for the best performance of the attack. Also, work can be expanded with varied quantization methods provided by TensorFlow. The attack can also be tested under the black box and semi-black box settings by constructing surrogate models.

References

Y. Feng, B. Wu, Y. Fan, L. Liu, Z. Li, and S. Xia. Boosting black-box attack with partially transferred conditional adversarial distribution. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15074–15083, 2022.

J. Guo, Y. Jiang, Y. Zhao, Q. Chen, and J. Sun. Dlfuzz: differential fuzzing testing of deep learning systems. *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2018.

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

K. Pei, Y. Cao, J. Yang, and S. S. Jana. Deepxplore: Automated whitebox testing of deep learning systems. *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017.

A. S. Rakin, Z. He, and D. Fan. Bit-flip attack: Crushing neural network with progressive bit search. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1211–1220, 2019.

F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. Mcdaniel. Ensemble adversarial training: Attacks and defenses. *ArXiv*, abs/1705.07204, 2018.