Machine Perception

Project 1

Jay Paranjape and Arushi Sinha

Section 1- Classification

Data Set

This dataset consists of 60,000 RGB images of size 32x32. The images belong to objects of 10 classes such as frogs, horses, ships, trucks etc. The dataset is divided into 50,000 training images and 10,000 testing images.

We are going to build a CNN model that can classify images of various objects. We have 10 classes of images:

1. Airplane
2. Automobile
3. Bird
4. Cat
5. Deer
6. Dog
7. Frog
8. Horse
9. Ship
10. Truck

We used PyTorch library to get the pre-built CIFAR-10 dataset. We normalized this dataset to increase the computation speed.

ResNet18

The CNN model used for image classification is ResNet18

RESNET18 has 18 layers with a 7x7 kernel as 1st layer. It has four layers of ConvNets that are identical. Each layer consists of two residual blocks. Each block consists of two weight layers with a skip connection connected to the output of the second weight layer with a ReLU. If the result is equal to the input of the

ConvNet layer, then the identity connection is used. But, if the input is not similar to the output, then a convolutional pooling is done on the skip connection. [1]

We used the pre-trained model of ResNet on PyTorch and fine-tuned it on the CIFAR-10 dataset.

- Accuracy found to be 95% on training dataset.
- Accuracy found to be 79% on test dataset.

Section 2- Adversarial Attack & Defense

Attacks

The 3 perturbations used to attack our network are:

- **FGSM (Fast Gradient Sign Attack):** It is a white-box attack It is designed to attack neural networks by leveraging the way they learn, gradients. This attack adjusts the input data to maximize the loss based on the same backpropagated gradients. In other words, the attack uses the gradient of the loss w.r.t the input data, then adjusts the input data to maximize the loss. [2]

- **DeepFool Attack:** It is also an untargeted white-box attack. It misclassifies the image with minimal perturbations, which humans cannot perceive and has been shown to be better than FGSM for generating adversarial images similar to the original image. [3]

- **Additive Gaussian Noise Attack:** It is a non-targeted black-box evasion attack. Here, the image is perturbed by adding gaussian noise until the model misclassifies the image. [4]

Methodology:

- We used the Foolbox library to perform the first two attacks. [5] [6]
- We implemented the three perturbations on all 10K elements of test data set, and ran the model on this adversarial test set.
- We observed an accuracy drop from **79% to 40.5% for Deepfool Attack** and from **79% to 44.86% for FGSM attack**
- For the third attack, we introduce gaussian noise in our test set with 0 mean and 0.09 variance. This causes the test set accuracy to fall down from **79 % to 72%.** We use this modified test set for gauging the effectiveness of our adversarial defense method for this attack as described next.

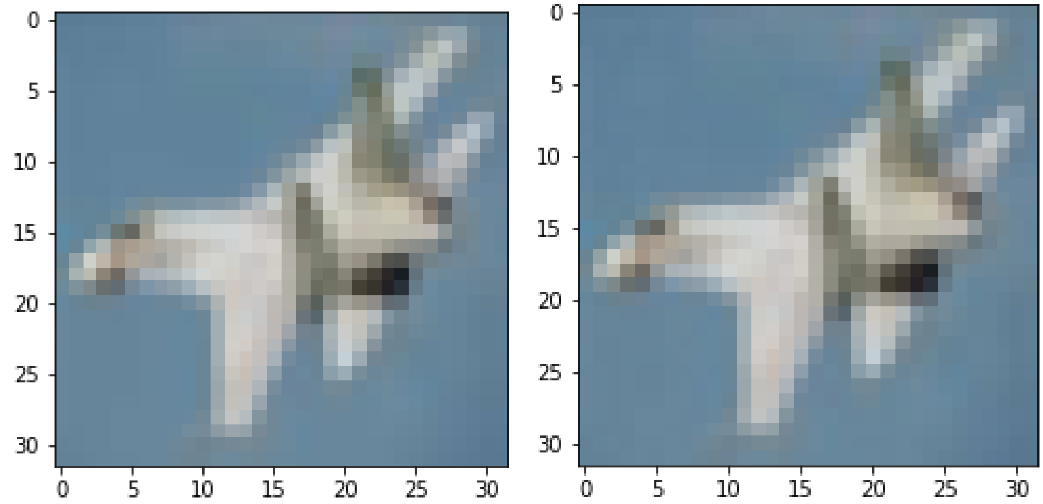
Defense:

- Adversarial defense is a technique with which we make our models robust to adversarial attacks. We show an effective technique against the Additive Gaussian Noise Attack.

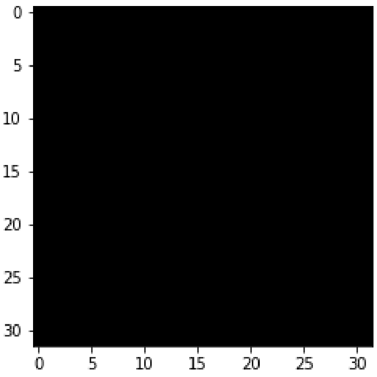
- We add gaussian noise with the same mean and variance to our training set and retrain the model on it. Thus, the model sees such noisy input during its training and learns its weights so as to minimize loss on these samples as well.
- This acts like a regularizer and improves the generalizability of the model on not only the adversarial test set, but also on the original test set.
- On the original test set, the accuracy increases from **79% to 81%**
- On the adversarial test set, the accuracy increases from **72% to 78%**

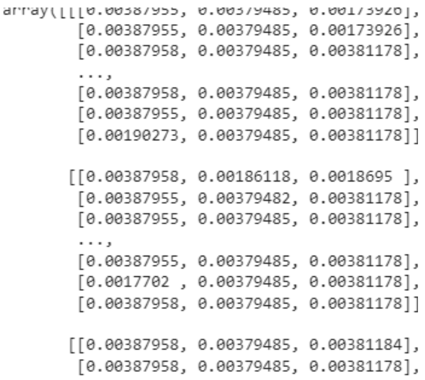
Some Visualizations:

1) Plane getting misclassified as deer after Deep fool attack



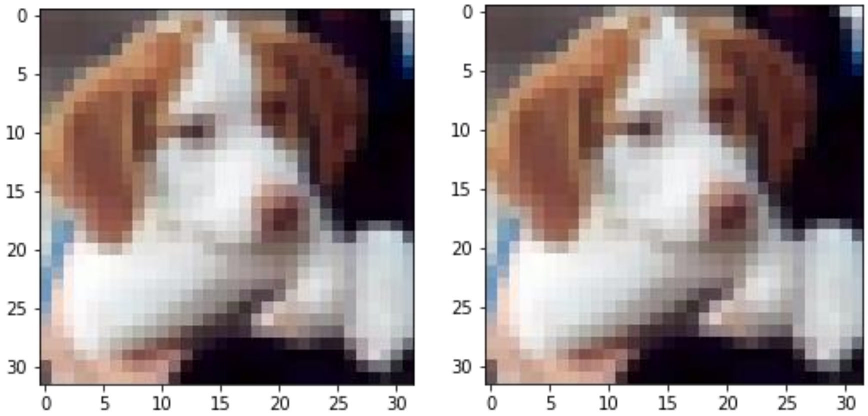
(L) Original, (R) Deepfool attack
Absolute Difference is quite small:





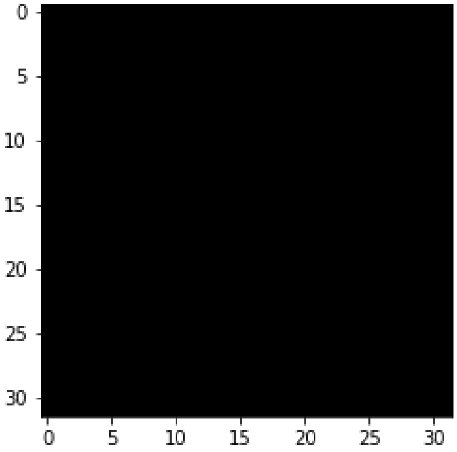
```
array([[[0.00387955, 0.00379485, 0.00173926],
        [0.00387955, 0.00379485, 0.00173926],
        [0.00387958, 0.00379485, 0.00381178],
        ...,
        [0.00387958, 0.00379485, 0.00381178],
        [0.00387955, 0.00379485, 0.00381178],
        [0.00190273, 0.00379485, 0.00381178]]

       [[0.00387958, 0.00186118, 0.0018695 ],
        [0.00387955, 0.00379482, 0.00381178],
        [0.00387955, 0.00379485, 0.00381178],
        ...,
        [0.00387955, 0.00379485, 0.00381178],
        [0.0017702 , 0.00379485, 0.00381178],
        [0.00387958, 0.00379485, 0.00381178]]

       [[0.00387958, 0.00379485, 0.00381184],
        [0.00387958, 0.00379485, 0.00381178],
```

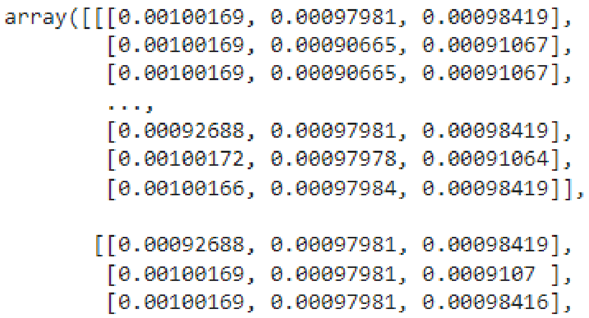
2) Dog getting misclassified as cat after FGSM attack



(L) Original, (R) FGSM Attack

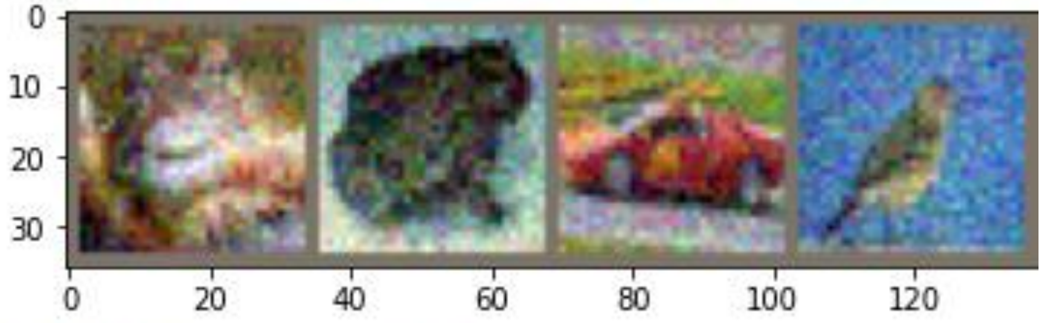
Absolute Difference is close to 0:





```
array([[[0.00100169, 0.00097981, 0.00098419],
        [0.00100169, 0.00090665, 0.00091067],
        [0.00100169, 0.00090665, 0.00091067],
        ...,
        [0.00092688, 0.00097981, 0.00098419],
        [0.00100172, 0.00097978, 0.00091064],
        [0.00100166, 0.00097984, 0.00098419]],

       [[0.00092688, 0.00097981, 0.00098419],
        [0.00100169, 0.00097981, 0.0009107 ],
        [0.00100169, 0.00097981, 0.00098416],
```

3) Black Box Additive Gaussian Noise Attacks on Some Test Set Images



```
frog  frog  car  bird
```

References

[1] S. A. Allena Venkata , D. L. Sahoo and D. V. R. Gurrala, "Resnet18 Model With Sequential Layer For Computin Accuracy on Image Classification Dataset," *IJCRT,* 2022.

[2] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and Harnessing Adversarial Examples," 2014.

[3] M. M. D. Seyed , A. Fawzi and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] B. Li, C. Chen, W. Wang and L. Carin, "Certified Adversarial Robustness," 2018.

[5] K. He, X. Zhang, . S. Ren and . J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[6] J. Rauber, R. Zimmermann, M. Bethge and W. Brendel, "Foolbox Native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, TensorFlow, and JAX," *Journal of Open Source Software,* vol. 5, p. 2607, 2020.

[7] . J. Rauber, W. Brendel and M. Bethge, "Foolbox: A Python toolbox to benchmark the robustness of machine learning models," in *Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning*, 2017.