Adversarial Attacks to Fool Deep Neural Networks

Advanced Machine Learning - UE15CS421
Assignment

Team Details:

Reshma Bhat	01FB15ECS235
Nikitha Rao	01FB15ECS364
Rutha Prasad	01FB15ECS367

Problem statement:

To use adversarial attacks to generate images that can fool deep neural networks.

- Different types of attacks are analysed.
- Comprehensive study of defense algorithms being used to prevent adversarial attacks.
- The adversarial images generated are studied to check for variations in encodings.
- The adversarial images generated are tested against several existing trained models and their performances are compared.

Introduction to Adversarial Machine Learning:

Adversarial images are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake; they're like optical illusions for machines.

Adversarial machine learning lies at the intersection of machine learning and computer security. It aims to enable the safe adoption of machine learning techniques in adversarial settings, such as spam filtering, malware detection, and biometric recognition.

Potential approaches:

Black box attacks vs White box attacks

In white box attacks the attacker has access to the model’s parameters, while in black box attacks, the attacker has no access to these parameters, i.e., it uses a different model or no model at all to generate adversarial images with the hope that these will transfer to the target model.

Targeted vs non targeted attacks

The aim of non-targeted attacks is to enforce the model to misclassify the adversarial image, while in the targeted attacks the attacker tries to get the image classified as a specific target class, which is different from the true class.

One shot attacks vs Iterative attacks

Gradient-based methods are among the most successful attacks. Here, the attackers modify the image in the direction of the gradient of the loss function with respect to the input image. In one-shot attacks, the attacker takes a single step in the direction of the gradient. In iterative attacks, instead of a single step, several steps are taken by the attacker.

Approach used:

Types of attacks

In our approach, we deal only with white box techniques. We compare the working of four types of attacks, namely:

Non – targeted fast gradient sign method
Targeted fast gradient sign method
Non – targeted one pixel attack
Targeted one pixel attack

Non – Targeted Fast Gradient Sign Method

This method computes an adversarial image by adding a pixel-wide perturbation of magnitude in the direction of the gradient. This perturbation is computed with a single step, thus is very efficient in terms of computation time:

$$x^{adv} = x + \varepsilon \cdot \mathtt{sign}(\nabla_x J(x, y_{true}))$$

Where,

x is the input clean image

x^{adv} is the perturbed adversarial image,

J is the classification loss function

y_{true} is the true label for the input x

Targeted Fast Gradient Sign Method

Similarly to the FGSM, in this method a gradient step is computed, but in this case in the direction of the negative gradient with respect to the target class:

$$x^{adv} = x - \varepsilon \cdot \mathtt{sign}(\nabla_x J(x, y_{target}))$$

Where,

x is the input clean image

x^{adv} is the perturbed adversarial image,

J is the classification loss function

y_{target} is the true label for the input x

Non Targeted One Pixel Attack

The objective of an untargeted attack is to cause a model to misclassify an image. This means we want to perturb an image as to minimize the confidence probability of the correct classification category and maximize the sum of the probabilities of all other categories.

The success criterion can be defined as a function that returns True whenever a given perturbation is sufficient to fool a model.
The next step is to find the pixels that will result in a successful attack. We formulate this as an optimization problem where we want to minimize the confidence of the correct class.

It can be very difficult to find an efficient gradient-based optimization that will work for the problem. It would be nice to use an optimization algorithm that can find good solutions without relying on the smoothness of the function. In our case, we have discrete integer positions ranging from 0 to 31 and color intensities from 0 to 255, so the function is expected to be jagged.

We use an algorithm called differential evolution for the same.

For the one pixel attack, our input will be a flat vector of pixel values:

$$X = (x_1, y_1, r_1, g_1, b_1, x_2, y_2, r_2, g_2, b_2, \ldots)$$

These will be encoded as floating-point values, but will be floored back into integers to calculate image perturbations. First we generate a random population of n perturbations

$$\mathbf{P} = (X_1, X_2, \ldots, X_n)$$

Then, on each iteration we calculate n new mutant children using the formula

$$X_i = X_{r1} + F(X_{r2} - X_{r3})$$

such that

$$r1 \neq r2 \neq r3$$

where r1,r2,r3 are random indices into our population P, and F=0.5 is a mutation parameter.

We pick 3 random individuals from the previous generation and recombine them to make a new candidate solution. If this candidate Xi gives a lower minimum at position i (i.e., the attack is closer to success), replace the old Xi with this new one. This process repeats for several iterations until our stopping criterion, success criterion, which is when we find an image that successfully completes the attack.

Targeted One Pixel Attack

The objective of a targeted attack is to cause a model to classify an image as a given target class. We want to perturb an image as to maximize the probability of a class of our own choosing.

Like in non targeted one pixel attack, we formulate the task of finding the pixels as an optimization problem where the goal is to maximize the confidence of a target class.

The rest of the steps are the same as that in non targeted one pixel attacks.

Types of Defenses

In our approach, we choose 2 models

InceptionV3

Adversarial Inception_Resnet_V2

The adversarially trained model architecture, uses multiple algorithms to lower the attack success of the images generated from all 3 attacks.
We explore the algorithms which tackle each possible attack:

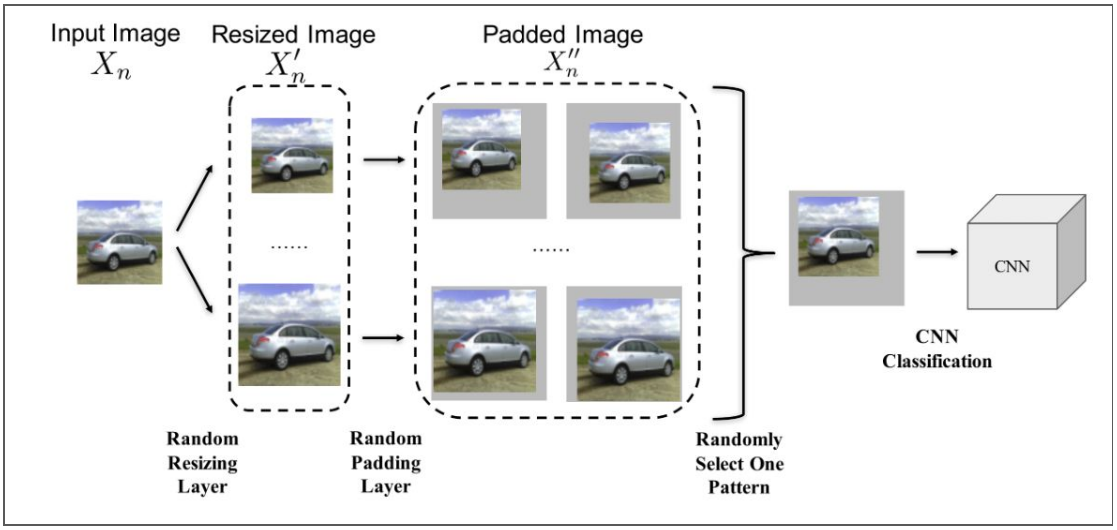
Non-Targeted fast gradient sign method

Since FGSM focuses on keeping the high level features of both the clean and adversarial images, an efficient defence would be to teach deep models to not only focus on low level features, but learn images equally on the high level representations of the same class images.

Thus in non-targeted attacks, when testing the model, the input images can be randomly resized and padded, to make the clean and adversarial images different explicitly.

This does affect the performance of the standard models on clean images, but it shown that the drop is not that drastic (about 3%). The drop in effectiveness of the attacks drops drastically (30%). Thus this method is advantageous as:

1. No additional training/ fine-tuning is required
2. Very little computation introduced
3. Compatible to different networks and different defending methods (we use randomization + ensemble adversarial training + Inception-Resnet-v2 in our submission)



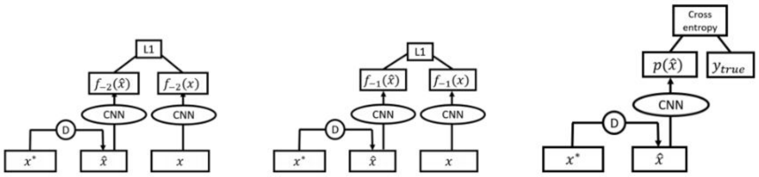
Targeted fast gradient sign method

Thus for targeted attacks, which add perturbations relating to a certain target class, we can use a denoiser/ CNN guided using a denoiser, to train the model to extract the perturbations relating to a particular class, and increase the margins between target classes.

Capsule networks and DenseNets also outperform standard models in defence. Since the perturbations are routed differently in testing, and reconstruction losses for the adversarial images will be different.

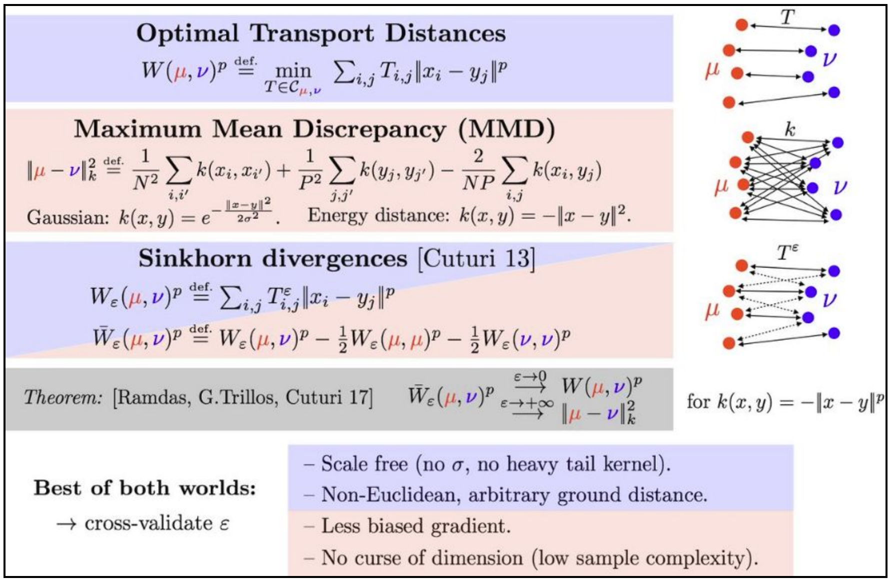
Thus the model is guided using a denoiser, to retrain the model on

- feature guided denoiser
- logits guided denoiser
- class label guided denoiser



(a) FGD (b) LGD (c) CGD

The denoiser works on MDD loss functions. The Maximum mean discrepancy technique focuses on finding how dissimilar two distributions without relying on any certain parameters (unlike KLD which uses mean and deviation).



Optimal Transport Distances

$$W(\mu,\nu)^p \overset{\text{def.}}{=} \min_{T\in\mathcal{C}_{\mu,\nu}} \sum_{i,j} T_{i,j}\|x_i - y_j\|^p$$

Maximum Mean Discrepancy (MMD)

$$\|\mu-\nu\|_k^2 \overset{\text{def.}}{=} \frac{1}{N^2}\sum_{i,i'} k(x_i,x_{i'}) + \frac{1}{P^2}\sum_{j,j'} k(y_j,y_{j'}) - \frac{2}{NP}\sum_{i,j} k(x_i,y_j)$$

Gaussian: $k(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$. Energy distance: $k(x,y) = -\|x-y\|^2$.

Sinkhorn divergences [Cuturi 13]

$$W_\varepsilon(\mu,\nu)^p \overset{\text{def.}}{=} \sum_{i,j} T^\varepsilon_{i,j}\|x_i - y_j\|^p$$

$$\bar{W}_\varepsilon(\mu,\nu)^p \overset{\text{def.}}{=} W_\varepsilon(\mu,\nu)^p - \tfrac{1}{2}W_\varepsilon(\mu,\mu)^p - \tfrac{1}{2}W_\varepsilon(\nu,\nu)^p$$

Theorem: [Ramdas, G.Trillos, Cuturi 17] $\bar{W}_\varepsilon(\mu,\nu)^p \begin{array}{l} \xrightarrow{\varepsilon\to 0} W(\mu,\nu)^p \\ \xrightarrow[\varepsilon\to+\infty]{} \|\mu-\nu\|_k^2 \end{array}$ for $k(x,y) = -\|x-y\|^p$

Best of both worlds:

→ cross-validate ε

– Scale free (no σ, no heavy tail kernel).
– Non-Euclidean, arbitrary ground distance.
– Less biased gradient.
– No curse of dimension (low sample complexity).

One pixel attack

Since one-pixel attacks are iterative, they are highly model dependant. Thus predicting possible adversarial images is tough. Thus retraining models for adversarial attacks, would require finding out all possible combinations of adversarial images that can be generated.

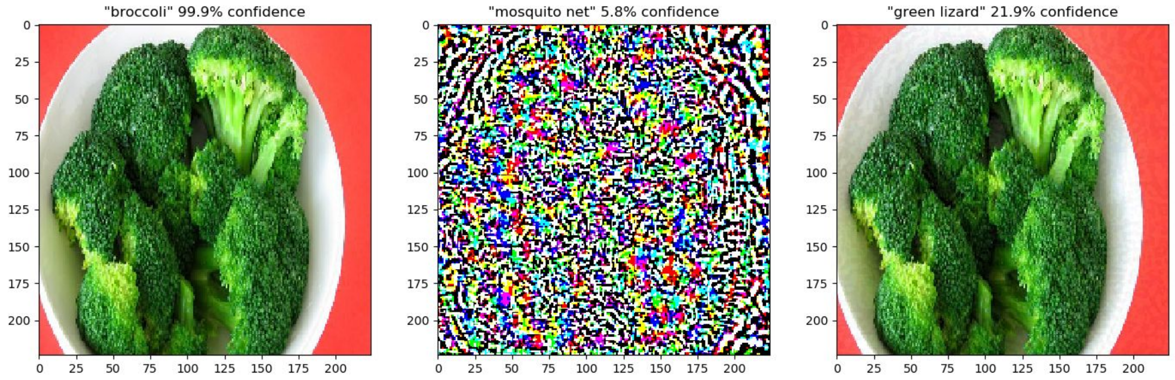
There is also a visible difference in the images, since whole pixel values are changed. Thus we can use ensemble methods to find out the best score, without retraining any of them.

For example, the model we have used, is a combination of the resnet and inception architectures and was trained on ensemble losses, using VGG-16, Inception and Resnet losses and a pre-trained adversarial network.

The basic idea is to manipulate the pixel values of the input *adversarial* images to reduce the total variation distance among the predictions of an ensemble of models. The idea mostly comes from the observation that targeted attacks are hard, and the models typically do not agree on a label on the adversarial images generated

Testing and Results:

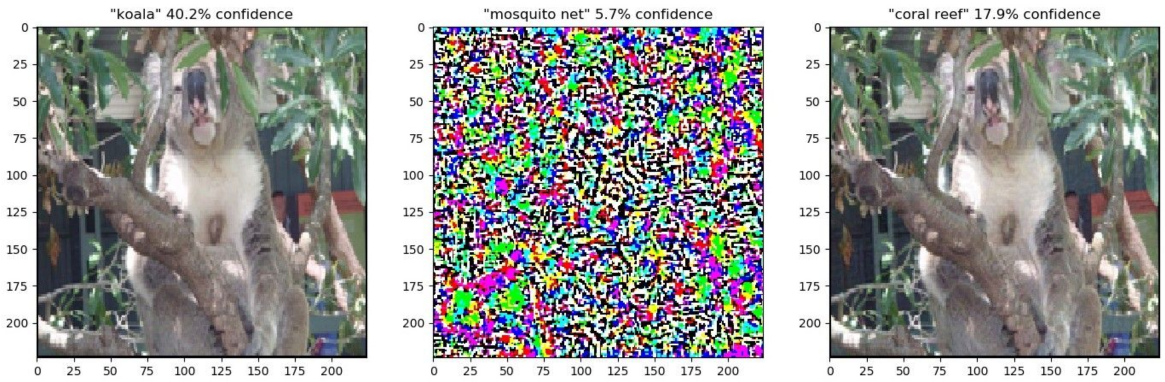
Non – Targeted Fast Gradient Sign Method


"broccoli" 99.9% confidence
"mosquito net" 5.8% confidence
"green lizard" 21.9% confidence
0
25
50
75
100
125
150
175
200

mean value of perturbation: -0.029283589

Output - Original Image	Output - Adversarial Image
('broccoli', 0.9986212)	('green_lizard', 0.21883842)
('macaque', 0.3260339)	('African_chameleon', 0.21825089)
('cauliflower', 0.00028594976)	('green_snake', 0.17226319)
('cucumber', 0.00016846452)	('American_chameleon', 0.1502423)
('mashed_potato', 7.190466e-05)	('zucchini', 0.042034246)

Targeted Fast Gradient Sign Method

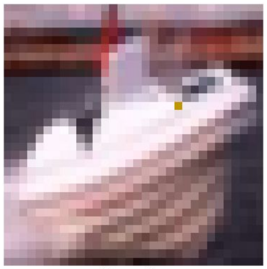


True Class: 383 Target Class: 973
mean value of perturbation: 0.006590136

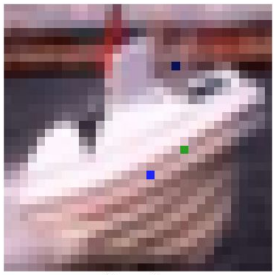
Output - Original Image	Output - Adversarial Image
('koala', 0.40233135)	('coral_reef', 0.17855316)
('macaque', 0.3260339)	('butcher_shop', 0.14763033)
('patas', 0.10155586)	('fountain', 0.09291324)
('titi', 0.037047733)	('shower_curtain', 0.060573325)
('squirrel_monkey', 0.023759123)	('carousel', 0.048611864)

Non Targeted One Pixel Attack

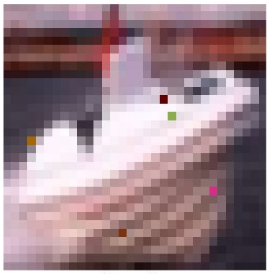
accuracy	**accuracy**	**pixels**	**attack_success_rate**
resnet	0.9231	1	0.34
resnet	0.9231	3	0.79
resnet	0.9231	5	0.79



True: ship
Predicted: cat



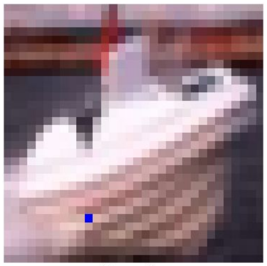
True: ship
Predicted: dog



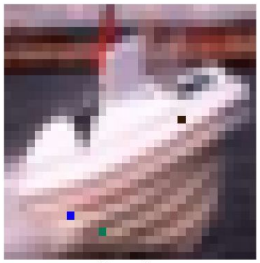
True: ship
Predicted: cat

# Pixels	True	Predicted	Accuracy (Before - After)	Steps
One	Ship	Cat	0.6866095 - 0.4855985	9
Three	Ship	Dog	0.5770358 - 0.1608321	2
Five	Ship	Cat	0.25268838	1

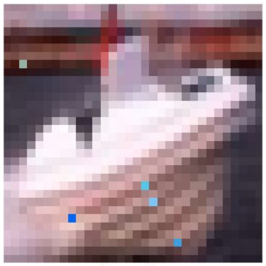
Targeted One Pixel Attack



True: ship
Predicted: ship



True: ship
Predicted: automobile



True: ship
Predicted: automobile

Target: Automobile

# Pixels	True	Predicted	Accuracy (Before - After)	Steps
One	Ship	Ship	0.0021637776 - 0.01990981	75
Three	Ship	Automobile	0.0026463128 - 0.54308754	36
Five	Ship	Automobile	0.012962655 - 0.5145811	29

accuracy	accuracy	pixels	attack_success_rate
resnet	0.9231	1	0.144444
resnet	0.9231	3	0.211111
resnet	0.9231	5	0.222222

Performance of pretrained models on adversarial images

Adversarial images constructed after FGSM and FGSM-T attacks are fed through InceptionV3 and adversarial InceptionV3. The accuracy of the models are visualised in Fig 2(a, b, c, d) that show the reduced accuracy for adversarial images.

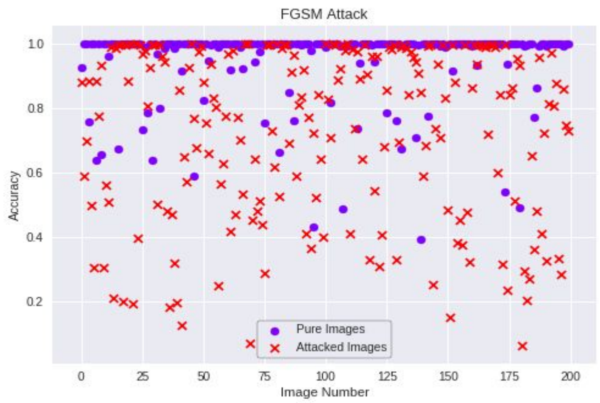


Fig 1a: FGSM Images on InceptionV3

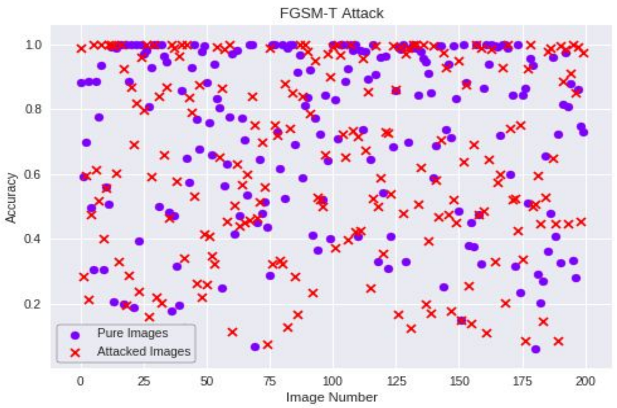


Fig 1b: FGSM-T Images on InceptionV3

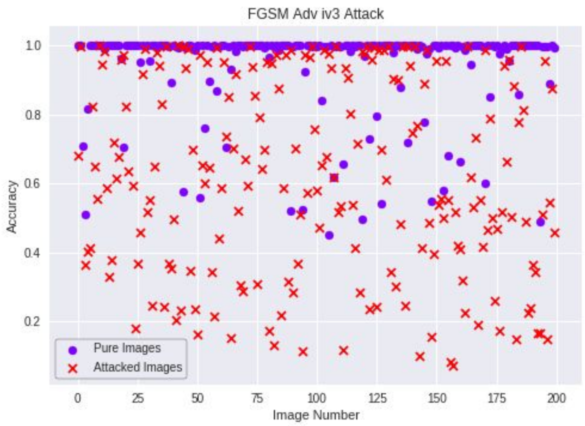


Fig 1c: FGSM Images on Adv_IncV3

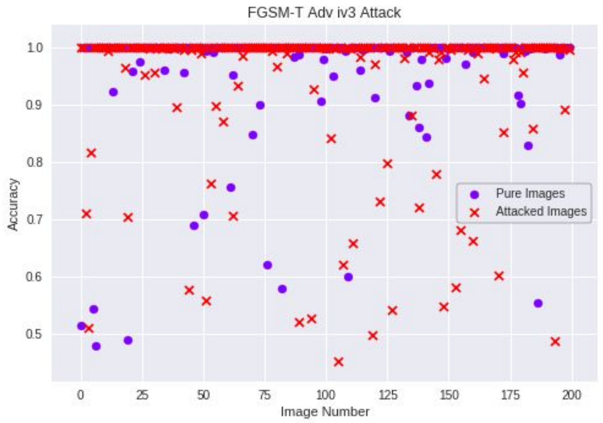


Fig 1d: FGSM-T Images on Adv_IncV3

Contrast in pure and adversarial images

The contrast between the adversarial images built from FGSM and FGSM-T attacking algorithms, and the corresponding pure images is visualised in the graph in Fig 2(a, b). The features for the images are extracted using VGG16 model. The input layer takes an image in the size of (224 x 224 x 3), and the output layer is a softmax prediction on 1000 classes. From the input layer to the last max pooling layer (labeled by 7 x 7 x 512) is regarded as the feature extraction part of the model

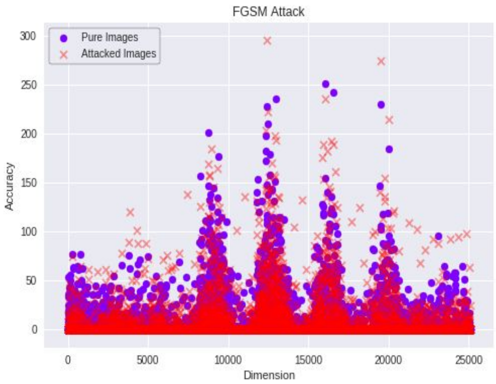


Fig 2a: FGSM attack comparison

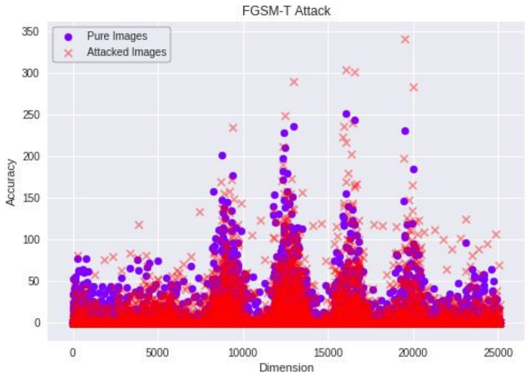


Fig 2b: FGSM-T attack comparison

References

Explaining and Harnessing Adversarial Examples

- Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy

Adversarial Examples: Attacks and Defenses for Deep Learning

- Xiaoyong Yuan, Pan He, Qile Zhu, Xiaolin Li

One pixel attack for fooling deep neural networks

- Jiawei Su, Danilo Vasconcellos Vargas, Sakurai Kouichi