A Meta Learning Approach for Adversarial Attacks

Shaik Mohammed Sayeed CS19BTECH11004

Gantasala Naga Aneesh Ajaroy CS19BTECH11010

Peddi Naga Hari Teja CS19BTECH11021

Vemulapalli Aditya CS19BTECH11025

Appanagari Sathwik Chakravarthi ES19BTECH11008

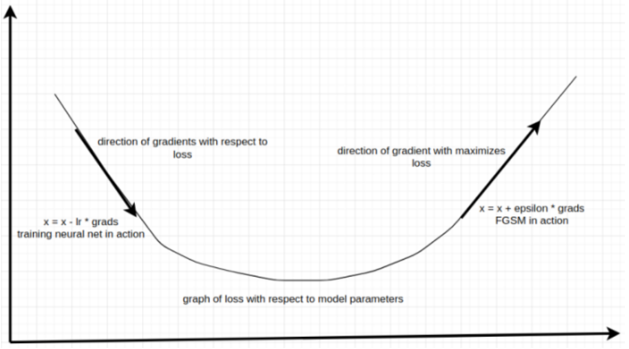
Abstract

In this final report we elaborated upon the adversarial attack method FGSM and the basic procedure of training an adversarial model agnostic meta learning model. We have provided details about the three implemented models, compared their performances on clean and adversarial samples. We presented our analysis of results obtained from these models. Finally, we presented our difficulties encountered while writing the code for adversarial MAML model.

1. FGSM Attack

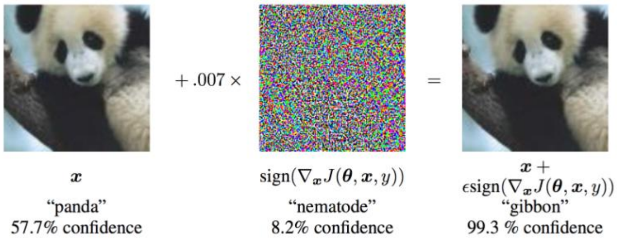
1.1 Introduction

It is most popular and one of the first adversarial attacks. It is intuitive and remarkably powerful. It is a white box attack. It assumes the attacker has full knowledge about the model such as inputs, weights, outputs and also has access to the model. It is done based on the architecture of the given network. It attacks the neural network based on the idea that gradient descent is used to find the lowest point of loss, hence going in the opposite direction of it, the loss can be maximized by adding a small amount of perturbation. The gradient of loss is calculated w.r.t to the input data and the input data is adjusted to maximize the loss.


direction of gradients with respect to loss
direction of gradient with maximizes loss
x = x - lr * grads
training neural net in action
x = x + epsilon * grads
FGSM in action
graph of loss with respect to model parameters

1.2 Example

Consider the following example,


$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

The original input image x belongs to the class panda. represents the model parameters. y represents the ground truth label. The network is trained using the loss equation J(θ,x,y). The gradient of loss is calculated w.r.t the input data as ∇x J(θ,x,y). In the direction of sign(∇x J(θ,x,y)), the input data is adjusted by a small step ε (in this case 0.007). The final perturbed image is misclassified by the network as gibbon.

2. Adversarial Training

2.1 Introduction

One of the basic adversarial defense methods is to train the model with adversarial samples generated from the training set. Both the clean and perturbed training samples are feeded into the architecture at the same time so as to prevent the loss in accuracy on the original dataset.

2.2 Drawback

The main drawback of the method is that it is effective only on the specific type of attack the model is trained on. When different attacks are used to generate adversarial examples, the model needs more and more adversarial training data.

3. Adversarial Meta Learning

3.1 Introduction

It is a variant of MAML. The meta learning model is trained with both clean and adversarial samples to find robust and better model parameters. They are used in both inner and outer loops in the MAML algorithm and they contribute equally in updating the model parameters. The correlation between the clean and adversarial samples is used by ADML, to make the model parameters robust to adversarial samples and generalize well to new samples.

3.2 Brief Procedure

Suppose we have a task distribution $p(T)$, we then sample a batch of tasks T_i, from the task distribution. Then for each task we sample a batch of k data points in train and test sets.

Suppose for each task we sample clean and adversarial samples for both train and test sets as $D_{clean_i}^{train}$, $D_{clean_i}^{test}$, $D_{adv_i}^{train}$ and $D_{adv_i}^{test}$.

We then find the optimal parameter θ'. We minimize the loss by gradient descent on both clean and adversarial training sets and find optimal parameters both the sets as θ'_{clean_i} and θ'_{adv_i}

$$\theta'_{clean_i} = \theta - \alpha_1 \nabla_\theta L_{T_i}(f_\theta, D_{clean_i}^{train})$$

$$\theta'_{adv_i} = \theta - \alpha_2 \nabla_\theta L_{T_i}(f_\theta, D_{adv_i}^{train})$$

After this we enter into the meta training phase where the optimal parameter θ' is calculated. We update it by minimizing the loss w.r.t to optimal parameters θ'_{clean_i} and θ'_{adv_i} found in the previous step.

$$\theta = \theta - \beta_1 \nabla_\theta \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta'_{clean_i}}, D_{clean_i}^{test})$$

$$\theta = \theta - \beta_2 \nabla_\theta \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta'_{adv_i}}, D_{adv_i}^{test})$$

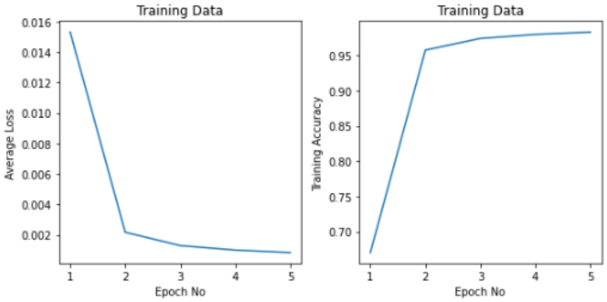
4. Code Repository

GitHub Link: Deep-Learning-2022-Project

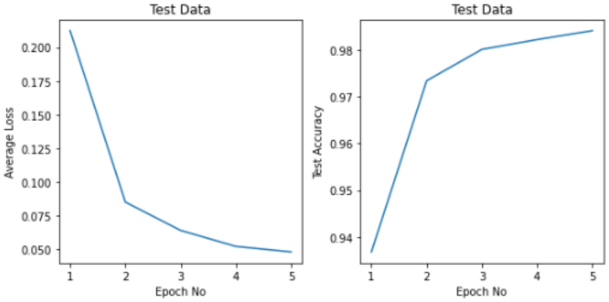
5. Results

Normal Model

Training Loss and Training Accuracy for each epoch on clean samples dataset



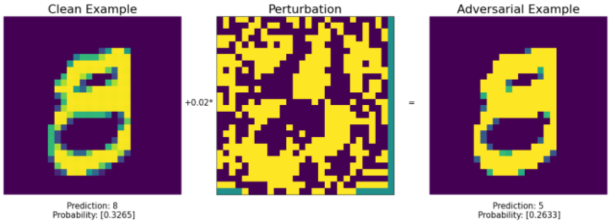
Test Loss and Test Accuracy for each epoch on clean samples dataset



Test Accuracy obtained on adversarial sample dataset: 0.1046

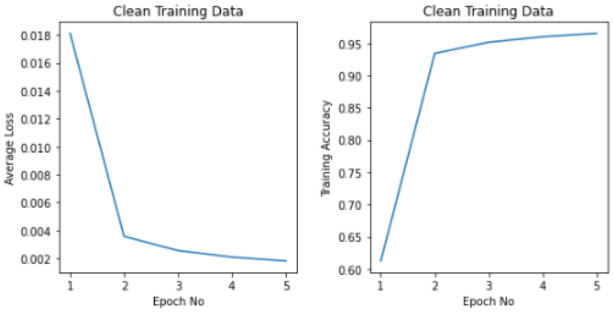
We can observe that the accuracy has drastically dropped down.

Generated Adversarial Samples using FGSM Attack

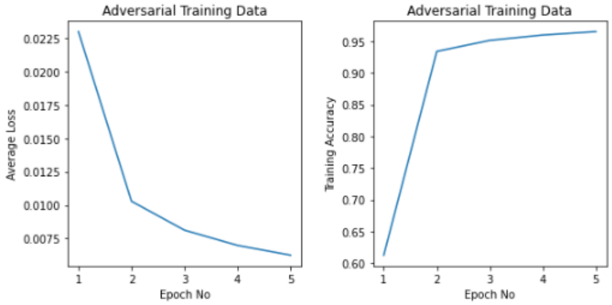


Adversarial Model

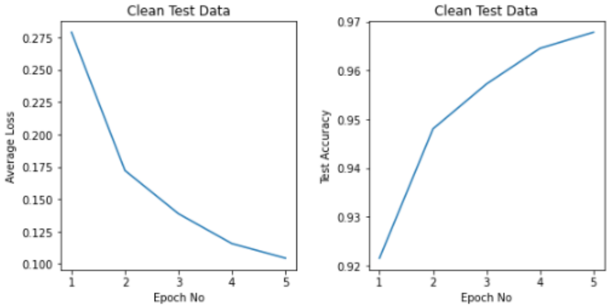
Training Loss and Training Accuracy for each epoch on clean samples dataset



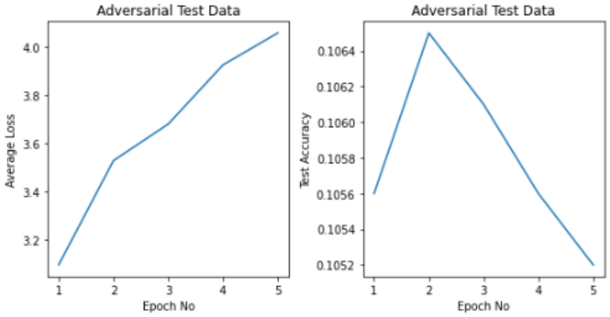
Training Loss and Training Accuracy for each epoch on adversarial samples dataset



Test Loss and Test Accuracy for each epoch on clean samples dataset



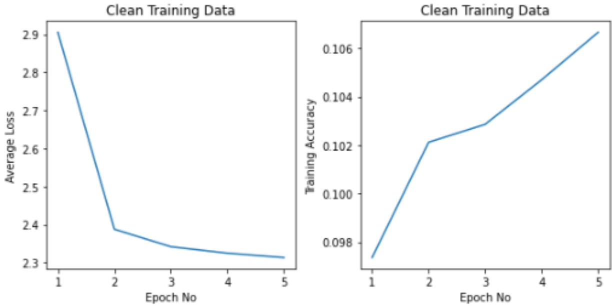
Test Loss and Test Accuracy for each epoch on adversarial samples dataset



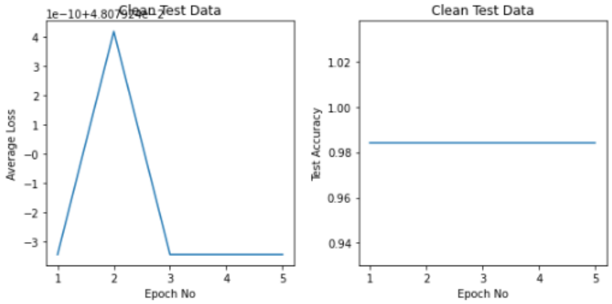
Comparing this and previous model we can observe that there is only slight difference in test accuracy on test sets of clean and adversarial samples.

Adversarial MAML Model

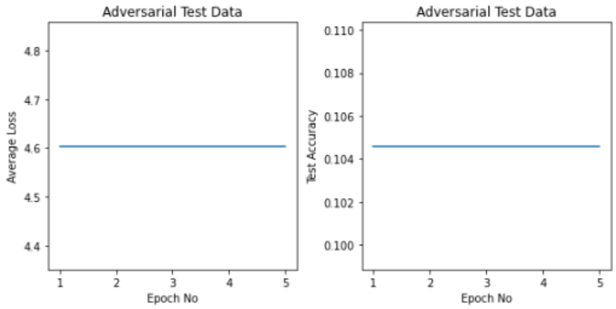
Training Loss and Training Accuracy for each epoch on clean samples dataset



Test Loss and Test Accuracy for each epoch on clean samples dataset



Test Loss and Test Accuracy for each epoch on adversarial samples dataset



Even though we didn’t train with the adversarial samples for this model, the accuracy obtained on test dataset of adversarial samples is on par with the previous two models.

Problems Encountered

We didn’t get the expected results from the adversarial MAM.L model. We are still working on improving the code for this model.

6. References

[1] https://arxiv.org/pdf/1412.6572.pdf
[2] https://arxiv.org/pdf/1712.09665.pdf
[3] https://arxiv.org/pdf/1801.02610.pdf
[4] https://arxiv.org/pdf/1703.03400.pdf
[5] https://proceedings.neurips.cc/paper/2020/file/cfee398643cbc3dc5eefc89334cacdc1-Paper.pdf