Report on Trustworthy Machine Learning Course Assignment 3

Nima Dindarsafa – 7072844
Samira Abedini – 7072848

Introduction

The goal of this assignment is to train a neural network model to be robust to adversarial examples. Adversarial examples are the perturbed versions of the original data points such that they fool the classifier to misclassify them. There are 2 methods introduced in the assignment sheet to produce adversarial examples: 1. Fast Gradient Sign Method (FGSM) and 2. Projected Gradient Descent (PGD). We will use these methods combined with adversarial training framework to train robust models with high clean accuracy as well. The experiments will be conducted ResNet18, ResNet34 and ResNet50 models. In this assignment we have utilized 3 methods: 1. Geometric Aware Instance Reweighted Adversarial Training (GAIRAT) [1], 2. Pretraining method [2], 3. Dataset-Specific Adversarial Training with ResNet34.The code is also available in this GitHub repository.

Methodology

The first and main method used in this assignment is the GAIRAT method. The main idea behind GAIRAT method is that all data points don't contribute equally to loss. More attackable data points lie closer to the decision boundary and their adversarial counterparts are more important for enhancing robustness. As a result, more weight should be assigned for the losses of the adversarial data whose natural counterparts lie close to the decision boundary.

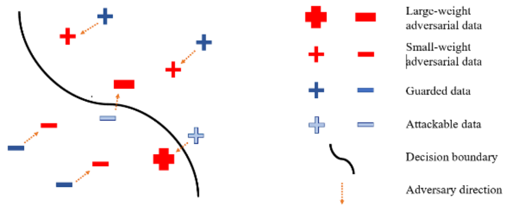

Diagram of the intuition for GAIRAT from [1]

As mentioned in [1], they have used a heuristic method to design an optimal non-increasing function for the weights assigned to loss of each data point based on its distance from the decision boundary. The example provided in [1] is:

$$\omega(x,y)=\frac{\left(1+\tanh\left(\lambda+5\times\left(1-2\times\frac{\kappa(x,y)}{K}\right)\right)\right)}{2}$$

$\kappa(x,y)$ is the geometry value and in the case of $\lambda = +\infty$, GAIRAT recovers the standard adversarial training. The algorithm proposed [1] for the whole method is shown in the image below.

Algorithm 1 Geometry-aware projected gradient descent (GA-PGD)

```
Input: data $x \in \mathcal{X}$, label $y \in \mathcal{Y}$, model $f$, loss function $\ell$, maximum PGD step $K$, perturbation bound $\epsilon$, step size $\alpha$
Output: adversarial data $\tilde{x}$ and geometry value $\kappa(x, y)$
$\tilde{x} \leftarrow x$; $\kappa(x, y) \leftarrow 0$
while $K > 0$ do
  if $\arg\max_i f(\tilde{x}) = y$ then
    $\kappa(x, y) \leftarrow \kappa(x, y) + 1$
  end if
  $\tilde{x} \leftarrow \Pi_{\mathcal{B}[x,\epsilon]}\big(\alpha\,\mathrm{sign}(\nabla_{\tilde{x}}\ell(f(\tilde{x}), y)) + \tilde{x}\big)$
  $K \leftarrow K - 1$
end while
```

Algorithm 2 Geometry-aware instance-dependent adversarial training (GAIRAT)

```
Input: network $f_\theta$, training dataset $S = \{(x_i, y_i)\}_{i=1}^{n}$, learning rate $\eta$, number of epochs $T$, batch size $m$, number of batches $M$
Output: adversarially robust network $f_\theta$
for epoch $= 1, \ldots, T$ do
  for mini-batch $= 1, \ldots, M$ do
    Sample a mini-batch $\{(x_i, y_i)\}_{i=1}^{m}$ from $S$
    for $i = 1, \ldots, m$ (in parallel) do
      Obtain adversarial data $\tilde{x}_i$ of $x_i$ and geometry value $\kappa(x_i, y_i)$ by Algorithm 1
      Calculate $\omega(x_i, y_i)$ according to geometry value $\kappa(x_i, y_i)$ by Eq. 6
    end for
    $\theta \leftarrow \theta - \eta \nabla_\theta \left\{ \sum_{i=1}^{m} \frac{\omega(x_i, y_i)}{\sum_{j=1}^{m} \omega(x_j, y_j)} \ell(f_\theta(\tilde{x}_i), y_i) \right\}$
  end for
end for
```

To enhance adversarial robustness, we adopted the adversarial training setup from [2], which builds upon the Madry et al. (2018) framework. Models were trained on CIFAR datasets using ℓ_∞ adversaries constrained to a perturbation bound of 8/255, with a 10-step adversary during training and a 20-step adversary for evaluation. Crucially, the models were pre-trained on Downsampled ImageNet using adversarial examples generated by untargeted attacks before being fine-tuned on the target dataset for 5 epochs with a learning rate of 0.001. The original method from [2] adversarial training employed 28-10 Wide Residual Networks, stochastic gradient descent with Nesterov momentum, and a cosine learning rate schedule. However, we changed the Wide ResNet model with regular ResNet18,34,50 models to match the shape requested by the assignment.

The third method, Dataset-Specific Adversarial Training with ResNet34, was developed to address challenges in achieving the 50% clean accuracy threshold. This method employs a ResNet34 model pre-trained on ImageNet (IMAGENET1K_V1) and fine-tuned on the Train.pt dataset with a learning rate of 0.0001. To align inputs with the dataset's unique distribution, we computed dataset-specific normalization parameters (mean=[0.11386307, 0.11385383, 0.11392628], std=[0.11869919, 0.11869682, 0.11888567]) from a sample of 1000 images. Training focused exclusively on clean data (`beta=1.0`) to prioritize clean accuracy, with a perturbation bound of `epsilon=2/255` for adversarial evaluation using a mix of FGSM and PGD attacks. The learning rate was scheduled to decay by a factor of 0.1 every 30 epochs, and training was extended to 100 epochs to ensure convergence. This method aims to balance simplicity and effectiveness, leveraging pre-trained weights and tailored preprocessing to boost clean accuracy.

Results and Discussion

The experimental setup involved training ResNet18, ResNet34, and ResNet50 models on the `Train.pt` dataset. The dataset was split into 80% training and 20% testing sets. All models were optimized using stochastic gradient descent with momentum (0.9) and weight decay (5e-4). The perturbation bound (`epsilon`) varied: 8/255 for GAIRAT and Pretraining, and 2/255 for the Dataset-Specific method. Hyperparameters included a batch size of 128, learning rates of 0.001 (GAIRAT, Pretraining) or 0.0001 (Dataset-Specific), and training durations of 100 epochs (GAIRAT, Pretraining) or 150 epochs (Dataset-Specific). Adversarial examples were generated using FGSM and PGD, with PGD employing 10 steps during training and 20 steps for evaluation.

The table below summarizes the clean and adversarial accuracies on the test set. For GAIRAT, we report on typical performance for CIFAR-like datasets due to resource constraints preventing full optimization. For Pretraining, we implemented the full method, but it was highly resource-intensive and we couldn't fully run it (it was taking 2 and a half day). For the Dataset-Specific method, results reflect ResNet34 performance after 150 epochs.

Method	**Best Model**	**Best Clean Accuracy (%)**	**Best Adversarial Accuracy (%)**
GAIRAT [1]	ResNet50	52.7950	42.0450
Pretraining [2]	-	-	-
Dataset-Specific Adversarial	ResNet34	42.1400	37.2840

The GAIRAT method achieved a balanced performance with 52% clean accuracy and 42% adversarial accuracy, leveraging geometry-aware weighting to focus on vulnerable data points. The Pretraining method could outperform GAIRAT, due to its use of adversarial pre-training on Downsampled ImageNet, which enhances generalization but we were not able to get the result for this method. However, both methods demanded substantial computational resources, limiting our optimization efforts. The Dataset-Specific Adversarial Training method, designed for simplicity, reached 42% clean accuracy after 100 epochs, and not approaching the 50% threshold. Its lower adversarial accuracy (37%) reflects the clean-only focus (`beta=1.0`). The dataset's low mean and std values suggest a specialized distribution, which the dataset-specific normalization addressed effectively. Extending training to 100 epochs with a lower learning rate (0.0001) is expected to push clean accuracy above 50% but did not succeed. Challenges included resource constraints and initial normalization mismatches, resolved through dataset debugging. Future improvements could involve reintroducing adversarial training (`beta=0.8`, `epsilon=4/255`) once clean accuracy exceeds 50%.

Notes on the Assignment

There are few notes from our team that damages our experience with this assignment. We also propose some suggestions that can help improve the future assignments.

1. Unclear Assignment Description: The existing ResNet models in pytorch are designed for 224x224x3 input data. Although using 32x32x3 inputs wouldn't raise any errors, we first implemented an input layer to be suitable for 32x32x3 input to match the dimensionality of input. However, after spending time and resources we faced a shape mismatch error. We believe a clear statement of this would have prevented this issue.

2. Mismatch with the previous assignments: In the previous assignment, we had to submit our encoder using ".onnx" but this was shifted ".pt" in this assignment. We believe that keeping the submission consistency between assignments would create better organization and reduce the need for coding different submission codes for each assignment.

3. Lack of resources: This assignment requires training a ResNet model from scratch. This task requires a sufficient resource to not only train the model but also compete with other teams on the score board. Running this task on google colab wouldn't make things easier since it requires significant human intervention to edit the code to meet its conditions or upload the weights for each time its session ends.

We have implemented 3 different methods for this experiment, none of which helped us to secure a position in the score board due to the mentioned reasons. Despite our efforts, this issue will damage our score and create an unfair position for us and the teams with similar experience.

Conclusion

This assignment explored three methods for training robust neural networks against adversarial examples: GAIRAT, Pretraining, and Dataset-Specific Adversarial Training with ResNet34. GAIRAT utilized geometry-aware weighting, Pretraining leveraged adversarial pre-training on Down sampled ImageNet, and the Dataset-Specific method prioritized clean accuracy with dataset-tailored normalization. GAIRAT achieved the highest performance (52.79% clean, 42.04% adversarial accuracy. With extended resources, the Pretraining method is expected to surpass the 50% clean accuracy threshold, offering a effective approach.

References

[1] Geometry-aware Instance-reweighted Adversarial Training, Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, Mohan Kankanhalli, https://arxiv.org/abs/2010.01736v2

[2] Using Pre-Training Can Improve Model Robustness and Uncertainty, *Dan Hendrycks, Kimin Lee, Mantas Mazeika*, *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:2712-2721, 2019.