CS 579 Trusworthy Machine Learning - Homework 2

Eugene Yong

April 27, 2023

Task 1: Attack Your Models

1.1 Clean Test Set

Model	Accuracy
LeNet	98.86%
ResNet18	83.51%

Table 1: Accuracy of Lenet and ResNet18 on clean test set

1.2 Adversarial Examples

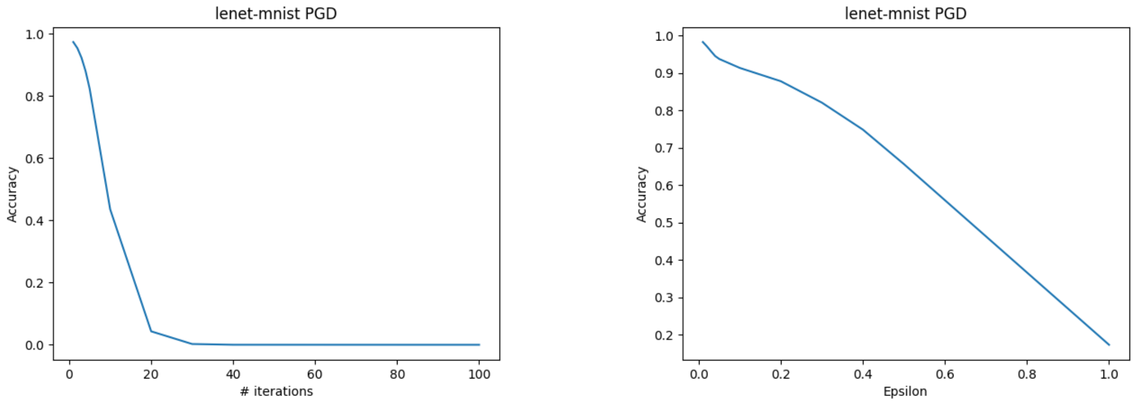
Model	Accuracy
LeNet	82.26%
ResNet18	0%

Table 2: Accuracy of Lenet and ResNet18 on their respective adversarial examples

The accuracy is of course dropped when testing with adversarial examples. It was expected since the PGD is doing FGSM iteratively with random initialization at the beginning. The PGD add a layer of noise to the original image, which the noise is in the same direction as the gradient of loss with respect to the input. Such perturbation is able to make the classifier missclassify results. However, it is unclear to me why it is so powerful against my ResNet18.

Task 2: Analyze the Impact of Several Factors on Your Attack's Success Rate

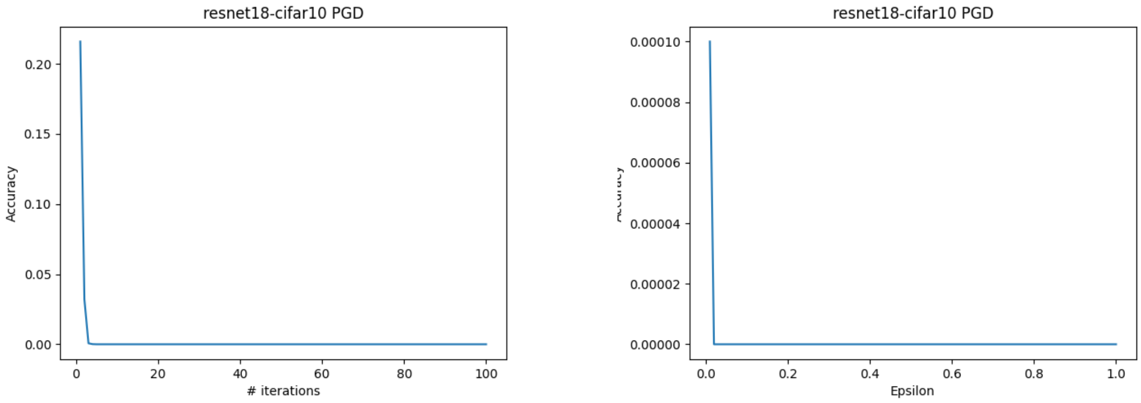
2.1 Analyze the Impact of Attack Hyper-parameters



(a) No. of Iterations vs Accuracy

(b) Epsilon vs Accuracy

Figure 1: Experimenting LeNet with different iterations and epsilon as attack hyperparameters



(a) No. of Iterations vs Accuracy

(b) Epsilon vs Accuracy

Figure 2: Experimenting ResNet18 with different iterations and epsilon as attack hyperparameters

It make sense that as the number of iteration increase, we are crafting a stronger attack, driving the perturbation towards the edge of epsilon. Thus making the model accuracy drop. With higher epsilon value, we will have a looser bound for the value we allowed to perturb, allowing the perturbation change more drastically, making noisier mask over the original image. This will also result in stronger attack and cause lower accuracy.

2.2.1 Data Augmentations

Model	Augmentation	Clean Accuracy	Adversarial Accuracy
LeNet	-	98.86%	82.26%
LeNet	Rotation	99.18%	84.17%
LeNet	Horizontal Flip	97.89%	75.93%
ResNet18	-	83.51%	0%
ResNet18	Rotation	88.27%	0.0006%
ResNet18	Horizontal Flip	87.48%	0.0003%

Table 3: Clean and adversarial accuracy of Lenet and ResNet18 on their respective data augmentation techniques

Learning with data augmentation techniques help model to better generalize and usually lead to a better clean accuracy. It also increase the model's robustness against adversarial attack because of learning from a more generalized model. Note that LeNet perform worse in clean accuracy with horizontal flip augmentation, it turns out that it did worse against adversarial attack too because the model just learn from a worse overall dataset that doesn't help generalize.

2.2.2 Regularizations

My models do exteremely bad using the default $niter = 5$ hyperparamter for PGD attack. In order to observe the impact of different regularization methods against adversarial attack, I used $niter = 4$ for this experiment.

Model	Dropout	Weight Decay	Clean Acc.	Adversarial Acc.	Acc. Diff
ResNet18	✗	-	83.51%	2.27%	-81.24%
ResNet18	✓	-	83.88%	1.48%	-82.4%
ResNet18	✗	1e-5	83.83%	2.25%	-81.58%
ResNet18	✗	1e-4	83.65%	2.66%	-80.99%
ResNet18	✗	1e-3	84.03%	1.55%	-82.48%
ResNet18	✗	1e-2	80.49%	3.58%	-76.91%
ResNet18	✗	1e-1	74.74%	7.27%	-67.47

Table 4: Clean and adversarial accuracy of ResNet18 on its respective regularization techniques

The trend here seems to be like models with higher clean accuracy will achieve lower adversarial accuracy. My guess is that as the strength of regularization increase (weight decay increases), the model tend to fit worse to the data. But at the same time, because the model is not over fitting, it is more robust to adversarial attack compare to model that fit better or overfit to the data. It is a trade off between having better clean accuracy or better adversarial accuracy. Regularization is one of the hyperparameter we can tweak to find a middle ground that works best.

Task 3: Defend Your Models with Adversarial Training

3.1 Adversarial Trained Results

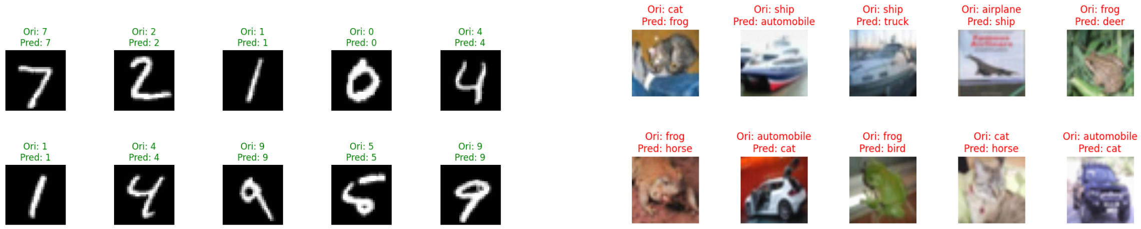
Model	PGD Iteration (Attack)	Clean Accuracy	Adversarial Accuracy
LeNet	5	99.15%	97.08%
	7		95.63%
ResNet18	5	73.35%	40.76%
	7		36.69%

Table 5: Clean and adversarial accuracy of Lenet and ResNet18 trained using adversarial training

Using adversarial examples to train the model make them more robust to adversarial attack because they learned from the perturbed examples and be able to generalize some when doing inference on test time adversarial examples. It's to my surprise that LeNet achieve a better clean accuracy by doing adversarial training, maybe it's because the dataset is simple enough that adversarial training actually help the model to generalize better. ResNet18 on the other hand did worse in clean accuracy to no surprise because it is harder to learn from perturb and noisy input data. However, it does help the model learn some of the adversarial pattern and be more robust to adversarial attack. It's to no surprise too that stronger attack still yield lower accuracy to even adversarial trained models.

Extra Credits

My Model Predictions

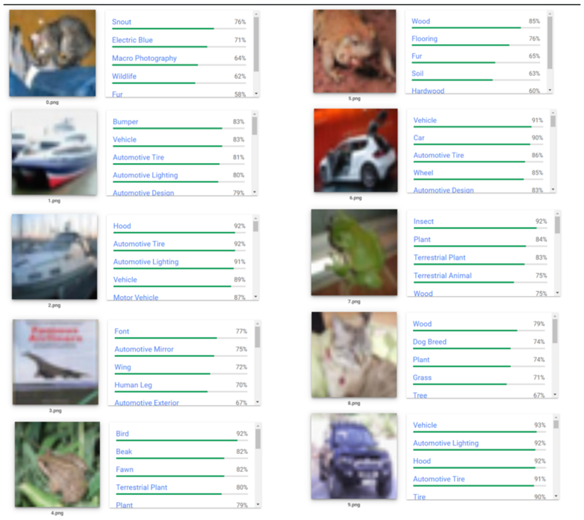


(a) LeNet on MNIST Result

(b) ResNet18 on CIFAR10 Result

Figure 3: Results of my models against my adversarial examples

Real World Model Predictions



0.png

Snout	76%
Electric Blue	71%
Macro Photography	64%
Wildlife	62%
Fur	58%

1.png

Bumper	83%
Vehicle	83%
Automotive Tire	81%
Automotive Lighting	80%
Automotive Design	79%

2.png

Hood	92%
Automotive Tire	92%
Automotive Lighting	91%
Vehicle	89%
Motor Vehicle	87%

3.png

Font	77%
Automotive Mirror	75%
Wing	72%
Human Leg	70%
Automotive Exterior	67%

4.png

Bird	92%
Beak	82%
Fawn	82%
Terrestrial Plant	80%
Plant	79%

5.png

Wood	85%
Flooring	76%
Fur	65%
Soil	63%
Hardwood	60%

6.png

Vehicle	91%
Car	90%
Automotive Tire	86%
Wheel	85%
Automotive Design	83%

7.png

Insect	92%
Plant	84%
Terrestrial Plant	83%
Terrestrial Animal	75%
Wood	75%

8.png

Wood	79%
Dog Breed	74%
Plant	74%
Grass	71%
Tree	67%

9.png

Vehicle	93%
Automotive Lighting	92%
Hood	92%
Automotive Tire	91%
Tire	90%

Figure 4: Results of google API against my adversarial examples

I searched for a very long time, way longer than I should. Still wasn't able to find any MNIST classifier online that takes in image as an input as oppose to drawing on web browser canvas.