Homework 3

Whitebox Defenses against AE

IMPORTANT: For this homework, you will be graded on how well your code works. To get credit, your code must be able to run with no errors. If your code doesn't run, we will not be debugging it during grading. Please pay careful attention to make sure your code follows the format laid out in the starter file.

Submission deadline: **April 29 (Tuesday) at 11:59 pm CDT on Gradescope**.
Late submission deadline: **April 31 (Thursday) at 11:59 pm CDT on Gradescope**.

Homework 3 is about training you to defend against adversarial attacks and evaluate the success of your defenses. In this assignment, you will be building defenses, adapting attacks to overcome these defenses, and then experimenting with more advanced defense techniques.

Part 1: Simple Defenses against PGD attack (25 points)

Your company's image classification model (a VGG network trained on CIFAR-10) has been leaked. An attacker has built a **targeted PGD attack** using white-box access to this model.

Unfortunately, your company currently lacks the resources to retrain its model and does not want to implement input filtering or adversarial detection. Instead, you are tasked with applying simple **input transformations** at inference time to disrupt the attack. The company will randomly apply **one of these transformations** during inference, so each transformation must be independently useful.

Specifications: You must implement **three input preprocessing transformations**, evaluate their impact on classification accuracy, and analyze their ability to reduce the effectiveness of a targeted PGD attack. For any given attack, you should choose the target class to be something other than the correct class label.

Transformations to Implement

1. JPEG Compression - removes high-frequency noise to disrupt adversarial perturbations

2. Image Resizing - alters pixel-level structure disrupt perturbations

3. Gaussian Blur - smooths edges and can erase some perturbations

Each transformation should be implemented in the provided function skeletons.

Evaluation Requirements:

You should evaluate the model's classification accuracy and the **attack success rate** of the targeted PGD attack under each of the following conditions:

Baseline (no transformation):

- Classification accuracy on clean images
- Classification accuracy under targeted PGD attack
- Attack success rate

After Each Transformation:

- Classification accuracy on clean images (*)
- Classification accuracy under targeted PGD attack (**)
- Attack success rate of the targeted PGD attack

Note: Even if a transformation does not prevent misclassification of adversarial inputs, it may still reduce the *attack success rate* by causing the model to misclassify the image as a class *other than the attack target*.

Testing Data Subset: Use **at least 50 random images** from the CIFAR-10 test set, with at least **5 images per class**.

Grading Breakdown (25 points total)

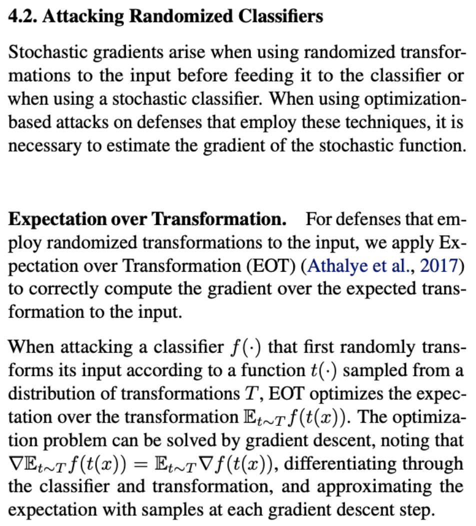
- 13 points: Implementation of the three transformation functions.
  To be considered a successful defense:
  - (*): Classification accuracy on clean images should remain above 70%
  - (**): Classification accuracy under targeted PGD attack should drop below 10%
- 12 points: Evaluation of model performance (1 point per evaluation)

Part 2: Estimation Over Transformation Attack (35 points)

You have since left the company. The attackers you previously worked against saw your Linked-In post and hired you to build a more advanced attack against the same model. You still have whitebox access to the model, but you do not know which transformation might be applied to your image at test time. Your task is to create a more advanced EOT attack that accounts for these transformations and evaluate the success of this attack.

Expectation over Transformation Attacks:

EOT attacks are similar in implementation to ensemble attacks you have previously worked with. You can reference Lecture 6, materials on 'Defense by Stochastic Gradient: randomized input transformation, Countermeasure: apply Expectation over Transformation (EoT)' to understand more about EOT attacks. Additionally, this paper provides more details on the gradient function including the passage to the right. To estimate any given transformation, you will want to include randomization in both the selection of the transformation method, as well as the parameters of the given transformation.



4.2. Attacking Randomized Classifiers

Stochastic gradients arise when using randomized transformations to the input before feeding it to the classifier or when using a stochastic classifier. When using optimization-based attacks on defenses that employ these techniques, it is necessary to estimate the gradient of the stochastic function.

Expectation over Transformation. For defenses that employ randomized transformations to the input, we apply Expectation over Transformation (EOT) (Athalye et al., 2017) to correctly compute the gradient over the expected transformation to the input.

When attacking a classifier $f(\cdot)$ that first randomly transforms its input according to a function $t(\cdot)$ sampled from a distribution of transformations T, EOT optimizes the expectation over the transformation $\mathbb{E}_{t\sim T} f(t(x))$. The optimization problem can be solved by gradient descent, noting that $\nabla \mathbb{E}_{t\sim T} f(t(x)) = \mathbb{E}_{t\sim T} \nabla f(t(x))$, differentiating through the classifier and transformation, and approximating the expectation with samples at each gradient descent step.

Attack Specifications: Implement an Expectation over Transformation (EOT) based PGD attack against the same VGG model used in Part 1. This attack should average gradients over multiple stochastic transformations (e.g., blur, compression, resizing) to be more robust against non-differentiable/randomized defenses.

Evaluation: Evaluate the attack (same evaluation format from part 1)

- Classification Accuracy under EOT attack
- Classification Accuracy under EOT attack + transformation defenses from part 1
- Write a short analysis (2-3 paragraphs) explaining why EOT attacks might be more effective than PGD attacks against the defenses from Part 1.

Format: Implement the attack in the function named "eot_attack" in the starter code. Submit your writing as a markdown or PDF file.

Grading Breakdown (35 points total). Points will be awarded as follows:

- 15 points: effective implementation of EOT (attack success + perturbation budget)
- 10 points: evaluation and reporting of accuracy under different conditions
- 10 points: written analysis

Part 3: Distillation Defense (40 points)

You have been hired by another company. They pay much more attention to their defenses and they won't release whitebox access to their base model. Instead, they want to experiment with **defensive distillation** to defend against gradient-based attacks, and they need your help implementing it.

They've provided you with the **VGG model from Part 1** to use as a **teacher model**, and your task is to train a **student model** via distillation. This student model will be released and you can assume attackers will have whitebox access to it.

Specifications:

1. Train a distilled **student model** on CIFAR-10 using softmax outputs (soft labels) from the teacher model.

2. Choose an appropriate **temperature T** for softening the outputs.
   Hint: You can experiment with different T values. Higher T may help training converge more smoothly.

3. Your student model must achieve **at least 80% accuracy on clean CIFAR-10 test data**.

4. Evaluate the student model on:

   - **Clean images**

   - **Images under PGD attack** (we provide the PGD function)

   *Important: Save your model as `"student_VGG.pth"` as this is used by our evaluation script. Do **not** modify the filename—this is already handled in the starter code.*

Writeup Requirement: Explain why the distilled model is more robust to gradient-based attacks. (1–2 short paragraphs)

Function Requirements

- `student_VGG()`: trains and saves your distilled student model.
- `evaluate_distillation()`: evaluates clean accuracy and PGD robustness.

Grading Breakdown (40 Points):

- **20 points** – Correct implementation of distillation logic
- **10 points** – Student model reaches ≥ 80% clean accuracy
- **5 points** – Evaluation
- **5 points** – Short writeup

~~Bonus Adaptive Attack (10 points)~~

~~Develop a new, stronger attack to break the distilled model. Your attack must:~~

- ~~Keep perturbations within the epsilon=8/255 L_infinity bound~~
- ~~Achieve > 50% success rate against the student model~~

  ~~**note the student model much achieve over 80% accuracy against clean data for the bonus attack to achieve points*~~

Submission

For this assignment, you will submit hw3_starter.py, student_VGG.pth, and writeup.txt to Gradescope. Do NOT change the names of these files.