Homework 2

1. Robustness-Accuracy Tradeoff In this exercise we will investigate the robustness-accuracy tradeoff. This toy model might give you some further intuition about the idea of useful vs robust features that we talked about in class.

Consider a binary classification task with the following data generating model:

$$y \overset{u.a.r}{\sim} \{-1, +1\}, \qquad x_1 = \begin{cases} +y, & \text{w.p. } p \\ -y, & \text{w.p. } 1-p \end{cases}, \qquad x_2, \ldots, x_{d+1} \overset{i.i.d}{\sim} \mathcal{N}(\eta y, 1).$$

For this distribution there is a tension between standard and ℓ_∞-robust accuracy. In the robust setting, we have an adversary that has ε budget to corrupt each input, that is, they can perturb each x to x' such that $\|x - x'\|_\infty < \varepsilon$.

For concreteness, think of a large d and $p = 0.95$, $\eta = \frac{10}{\sqrt{d}}$ and $\varepsilon = 2\eta$.

1. Suggest a classifier that has $1 - \delta$ accuracy for any small $\delta > 0$ when d is large enough. What is the robust accuracy ($\mathbb{P}[f(x') = y]$, where x' is perturbed by the adversary) of the classifier? What is a simple classifier that achieves $\geq p$ robust accuracy?

2. Show that for any classifier that achieves $1 - \delta$ accuracy for $\delta \to 0$ and d large enough, will have robust accuracy $O(\delta)$.
   Hint: First note that in order to achieve arbitrarily good standard accuracy you have to utilize at least some of the features $x_2, \ldots, x_{d+1}$. Then, consider the probability $\mathbb{P}[f(x) = 1 \,|\, x_1, y]$, and express both the standard ($\mathbb{P}[f(x) = y]$) and robust ($\mathbb{P}[f(x') = y]$) accuracies in terms of it.

2. Robustness through random noise In class we talked about adversarial training as a way to defend against adversarial examples. In this problem, you will investigate the potential of a different "paradoxical" way to increase robustness: adding extra noise to the input image during testing. In particular, for each pixel of an image (and all 3 channels), we generate noise $\eta \sim \mathcal{N}(0,\ \sigma^2)$ for some fixed σ^2. During testing, before feeding the image to the classifier we apply this noise and we classify $x + \eta_x$ instead.

Use the colab provided to investigate this idea for an attacker employing targeted Projected Gradient Descent (PGD) with ℓ_∞ perturbations (targeted attack means that they try to convert an image to a specific class instead of just

causing misclassification to the model). You will use the ImageNet v2 dataset together with Inception, a pretrained model for the original ImageNet dataset. In the colab, there is code to download and load ImageNet v2 and Inception.

In the following you can use between 100 and 200 samples (or a reasonable number of samples that works for you) out of 10,000 images of ImageNet V2 to create the required plots since PGD is computationally expensive.

1. Implement the noise addition step for various values of σ^2 and create a plot to show the clean vs robust accuracy tradeoff as σ^2 increases. Also plot the success rate of the adversary, i.e. the probability that the image will be classified as the adversarial target. Choose values of σ^2 that you think provide a reasonable tradeoff, since if σ^2 becomes too large then both the standard and the robust accuracy will plummet. Does the defense seem to be working? What is the intuition behind it?

2. Propose a PGD-based attack that is specifically tailored for this defense. What do you need to do to bypass it? Implement your new attack and compare the results by again creating the clean vs robust accuracy plot for the same values of σ^2 you chose before.

3. Open-ended: Try to enrich your defense: instead of using 1 random noise profile generate m of them, each with possibly different value for σ^2. The final prediction will be a (weighted) average of the predictions of the individual noisy images. What is the robustness/accuracy tradeoff here and do you think the defense is strong enough? If the attacker knows your defense parameters (m, the various σ^2s and your, possibly randomized, final classication rule) can they still bypass it?

Relevant citations for further reading

- Towards Evaluating the Robustness of Neural Networks: https://arxiv.org/abs/1608.04644
- Certified Adversarial Robustness via Randomized Smoothing: https://arxiv.org/pdf/1902.02918.pdf