Exercise 04

PGD Defenses

Reliable and Interpretable Artificial Intelligence
ETH Zurich

Problem 1 (Coding)**.** This task can either be done either in `task.py` or `task.ipynb` (whichever you prefer). In Exercise 3, you have implemented the untargeted PGD attack. Here, you are going to implement a defense for that attack.

1. Using your implementation of `pgd_untargeted` from the previous exercise, implement the PGD defense as discussed in the lecture. Train the network `Net` inspected in Exercise 2 on the MNIST dataset with PGD defense to obtain a robust model. You might want to train using batches as in `mnist_train.ipynb` (Exercise 2).

2. Next, we are inspecting the robustness of the new model. First, feed 1'000 training samples to the network and record the resulting accuracy. Then, perform an untargeted PGD attack onto each sample and feed the perturbed samples to the network. What is the difference in accuracy?

3. Try to reproduce the result shown in slide number 45 of slide deck 2 (Adversarial Example Generation). For this, pick a random image x from the MNIST training dataset. Extend your algorithm such that after every k batches trained, it runs 1'000 untargeted PGD attacks for x using random starting points in $S(x)$. For each training iteration, visualize the *single data point* cross-entropy losses for the perturbed datapoints in a histogram.

Problem 2 (Projection onto ℓ_2-ball)**.** The Euclidean projection z of a point y onto the ϵ-ℓ_p-ball around x is defined as (note the ℓ_2 norm):

$$z = \text{argmin}_{x' \text{ s.t. } \|x'-x\|_p \leq \epsilon} \ \|x' - y\|_2$$

In general, this is a hard problem and closed form solutions are only known for few p. For example, we have considered projections for $p = \infty$ in the lecture. Here, we are investigating the case for $p = 2$.

1. Derive the closed form solution of projecting a point y onto the ϵ-ℓ_2-ball around a point x.

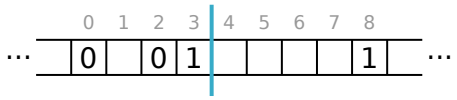
2. Prove that in 2 dimensions, your closed form solution z is correct, i.e., show that there exists no point $q \neq z$ in the ϵ-ℓ_2-ball around x that is closer to y than z. *Hint:* Assume for the sake of contradiction that there exists such a point q. Use the triangle inequality.

Problem 3 (Alternating optimization)**.** In the lecture, we have formalized defense as the optimization problem of finding the optimal parameters θ^* such that:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(x,y)\sim D}\left[\max_{x' \in S(x)} L(\theta, x', y)\right] \tag{1}$$

We stress that in general, finding the global optimum (1) is hard due to the nesting of maximization and minimization. In the lecture, we have considered a method that *approximates* the optimal parameters by alternatingly solving the outer and inner optimization problem individually. In this exercise, you are going to show that this approach can result in a *local* minimum that is not globally optimal.

Consider the basic case in one dimension where data is of the form (x, y) with discretized $x \in \mathbb{Z}$ and label $y \in \{0, 1\}$. Assume a very simple classifier ρ in one parameter $\tau \in \mathbb{Z}$, which classifies a point x as $\rho(\tau, x) = [x \geq \tau]$, where $[\cdot]$ is the indicator function. Consider a scenario where $D = \{(0, 0), (2, 0), (3, 1), (8, 1)\}$ as illustrated in the following figure:



For a datapoint (x, y), we define the loss $L(\tau, x, y) := [\rho(\tau, x) \neq y]$. Assuming all data points in D are equally likely, the expected loss of a model $\rho(\tau, \cdot)$ is the fraction of misclassified points in D. In our example, the model for $\tau = 4$ (indicated as a blue line) has expected loss $\frac{1}{4}$ as it misclassifies one out of four points. We define the adversarial region $S(x)$ around x as the set of points with distance at most 2 to x, this is: $S(x) := \{y \in \mathbb{Z} : |x - y| \leq 2\}$. For example, point $(8, 1)$ can be perturbed to $(6, 1)$.

Show how in this scenario, alternating optimization of (1) can result in a local optimum that is strictly worse than the global optimum.