Adversarial Robustness

Ekaterina Mozhegova

7th April

The study is based on the tutorial "Adversarial Robustness - Theory and Practice".

Abstract

Adversarial attacks are small yet malicious perturbations applied to input datasets. Often invisible to the human eye, they can damage the internal structure of the model and consequently its accuracy and reliability. These attacks pose a threat to security-critical tasks, including computer vision and natural language processing, among others.

Contents

1 Chapter 1 - Introduction to adversarial robustness

Let's define the model for a classification problem with the hypothesis function h_Θ. The loss function (softmax loss) of the model is the following: $l(h_\theta(x_i), y_i)$.

The common approach for the classification problem is to solve the optimization problem:

$$\min_\theta \frac{1}{m} \sum_{i=1}^{m} l(h_\theta(x_i), y_i)$$

It is typically solved by calculating the gradient of our loss function:

$$\Theta := \Theta - \frac{\alpha}{\beta} \sum_{i \in \beta} \nabla_\Theta l(h_\theta(x_i), y_i)$$

In standard optimization tasks, we aim to minimize the loss function. However, for creating adversarial attacks, our objective is to maximize it:

$$\max_{\hat{x}} l(h_\Theta(\hat{x}, y))$$

$\hat{x}$ here describes the adversarial example.

What we are specifically interested in is the gradient:

$$\nabla_\Theta l(h_\theta(x_i), y_i)$$

Adversarial example $\hat{x}$ needs to be similar to the original input x to stay meaningful, so we optimize over the pertrubation to x.

$$\max_{\delta \in \Delta} l(h_\Theta(x + \delta), y)$$

For intuition, Δ should be the range in which the input is the same as the original x.

1.1 Targeted attacks

Using this technique we can deceive the model into predicting an incorrect class. To achieve this, we aim to maximize the loss function for the correct label while minimizing the loss function for the targeted class.

$$\text{maximize} \left(l(h_\theta(x + \delta), y) - l(h_\Theta(x + \delta), y_{\text{target}}) \right)$$

1.2 Adversarial risks

The concept of adversarial risk is introduced to enhance the robustness of the model. We can evaluate adversarial risks alongside traditional empirical risks. They are estimated using a finite set of data.

$$R_{adv}(h\theta) = E_{(x,y)\sim D} \left[\max_{\delta \in \Delta(x)} \mathcal{L}(h\theta(x + \delta), y) \right]$$

Evaluating adversarial risks allows us to assess the accuracy of the model even in cases where it is subjected to attacks or adversarial manipulation. This evaluation can help us predict the behaviour of our model in case of attacks.

$$\Theta := \Theta - \frac{\alpha}{|\beta|} \sum_{(x,y)\in\beta} \nabla_\Theta \max_{\delta\in\Delta(x)} l(h_\theta(x_i+\delta), y_i)$$

The gradient of the inner term, which involves a maximization problem, is computed as follows, taking into account Danskin's theorem.

$$\delta^* = \text{argmax}_{\delta\in\Delta(x)} \mathcal{L}(h_\Theta(x+\delta^*), y)$$

$$\Delta_\Theta \max_{\delta\in\Delta(x)} l(h_\Theta(x+\delta), y) = \Delta_\Theta l(h_\Theta(x+\delta^*), y)$$

2 Chapter 2 - Linear models

2.1 Linear models

Let's define the model for a binary classification problem with the hypothesis function h_Θ. The loss function (logistic loss) of the model is the following:

The hypothesis function is:

$$h_\Theta(x) = w^T x + b$$

And the loss function is:

$$l(h_\Theta(x), y) = \log(1+\exp(-y\cdot h_\Theta(x))) \equiv L(y\cdot h_\Theta(x)),$$

where $L(z) = \log(1+\exp(-z))$, a decreasing function.

$$\max_{\|\delta\|\le\epsilon} l(w^T(x+\delta), y) \equiv \max_{\|\delta\|\le\epsilon} L(y\cdot(w^T(x+\delta))+b)$$

Due to L being a decreasing function, its argument must be minimized in order for the function to reach its maximum.

$$\max_{\|\delta\|\le\epsilon} L(y\cdot(w^T(x+\delta))+b) = L \min_{\|\delta\|\le\epsilon} (y\cdot(w^T(x+\delta))+b) \equiv y\cdot(w^T(x+b)) + \min_{\|\delta\|\le\epsilon} (y\cdot w^T\cdot\delta)$$

For solving this optimization problem:

$$\min_{\|\delta\|\le\epsilon} (y\cdot w^T\cdot\delta),$$

let's first assume that $y=1$.

Since $|\delta|_\infty \le \epsilon$, where $\epsilon > 0$, the obvious approach to solve the given minimization problem is to assign $\delta = -\epsilon$ for $w^T \ge 0$ and $\delta = \epsilon$ for $w^T \le 0$.

$$\delta^* = -y\epsilon\cdot\text{sign}(w)$$

Let's denote the optimized δ^* as

$$\delta^* = -\epsilon||w||_1$$

and substitute it in the initial equation, so that:

$$\max_{\|\delta\|_\infty \leq \epsilon} L(y \cdot (w^T(x+\delta)+b)) = L(y \cdot (w^T x + b) - \epsilon\|w\|_1)$$

2.2 Practical results

I observed a tradeoff while testing the robustness of the model on MNIST dataset for images labeled as 0 and 1. Surprisingly, the model made more mistakes on non-adversarial data, despite previously performing better on this type of data.