Adversarial Robustness

Ekaterina Mozhegova

7th April

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

Abstract

Adversarial robustness is essential in security-critical tasks, such as computer vision or natural language processing. For example, in cybersecurity and healthcare domains.

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

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

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

2.2

References

[1] Author, *Title*, Journal/Website, Year.