Exercise 02

Neural Networks & Adversarial Examples

Reliable and Interpretable Artificial Intelligence
ETH Zurich

Problem 1. In this introductory exercise we will use a 3 layer toy network N. We denote a layer as $l_{\boldsymbol{A}}(\boldsymbol{x}) := \boldsymbol{A}\boldsymbol{x}$, the ReLU-activation (Rectified Linear Unit) – which applies $\max(\boldsymbol{x}_i, 0)$ elementwise – $\mathrm{ReLU}(\boldsymbol{x}) = \max(\boldsymbol{x}, \boldsymbol{0})$. We write

$$N(\boldsymbol{x}) := (l_{\boldsymbol{C}} \circ \mathrm{ReLU} \circ l_{\boldsymbol{B}} \circ \mathrm{ReLU} \circ l_{\boldsymbol{A}})(\boldsymbol{x})$$

where $\circ$ denotes function composition and $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{10\times 10}$ as well as $\boldsymbol{C} \in \mathbb{R}^{3\times 10}$ are matrices. Further, we let $\mathrm{softmax}(\boldsymbol{x})$ denote the softmax-function for $\boldsymbol{x} \in \mathbb{R}^n$,

$$\mathrm{softmax}(\boldsymbol{x})_i = \frac{e^{\boldsymbol{x}_i}}{\sum_{j=1}^{n} e^{\boldsymbol{x}_j}} \tag{1}$$

for $i \in \{1, \ldots, n\}$.

Answer the following questions:

1. For the given matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ what is the space of possible inputs to N, i.e. the domain of N?

2. For the given matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ what is the space of possible outputs of N, i.e. the codomain of N? Can the output be seen as a vector of probabilities corresponding to the class probabilities of a categorical distribution?

3. For the given matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ what is the space of possible outputs of $(\mathrm{softmax} \circ N)$, i.e. the codomain of $\mathrm{softmax} \circ N$? Can the output be seen as a vector of probabilities corresponding to the class probabilities of a categorical distribution?

4. If N is a classifier, for how many classes does it work? (Assuming the standard usage of neural networks in classification.)

5. Assume N has been trained and you want to use it to for classification on an input $\boldsymbol{x}$. Do you use $(\mathrm{softmax} \circ N)$ or N? Justify your answer.

6. For a data point $\boldsymbol{x}$ with label $y = 2$ you want to create an adversarial example $\boldsymbol{x}'$ that is close to $\boldsymbol{x}$ ($\|\boldsymbol{x} - \boldsymbol{x}'\|_\infty < \epsilon$) and that classifies to $y' = 1$. Do you use an untargeted or targeted attack?

7. You use the Fast Gradient Sign Method (FGSM) (cf. slides 26-28 of Lecture 2) to compute the perturbation in question 6. Write down the necessary equations to obtain $\boldsymbol{x}'$. Is the computation applied to N or softmax $\circ N$?

8. (**Coding**) In `fgsm.py` you are provided with an implementation for N (for some matrices $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$) in PyTorch[1]. Follow the instructions there (does not assume familiarity with PyTorch) and complete the code.[2]

Problem 2. Carlini and Wagner ([1]) (slides 32-43 in Lecture 2) state the following (notation adapted to lecture notation):

We define an objective function obj *such that* $f(\boldsymbol{x}+\boldsymbol{\eta}) = t$ *if and only if* $\text{obj}^t(\boldsymbol{x}+\boldsymbol{\eta}) \leq 0$. *There are many possible choices for* obj*:*

$$\text{obj}_1^t(\boldsymbol{x}') = -\text{loss}_t(\boldsymbol{x}') + 1 \tag{2}$$
$$\vdots$$

N is a neural network, similar to the first problem, and $f(\boldsymbol{x}) = \text{argmax}_k \left((\text{softmax} \circ N)(\boldsymbol{x})\right)_k$ denotes a neural network making a classification. $N(\boldsymbol{x})$ computes the network logits, and $f(\boldsymbol{x})$ is syntactic sugar for also making the classification. Assume that the network can classify between C classes (0-indexed) and consider $\text{loss}_t(\boldsymbol{x})$ to be the cross-entropy loss on the neural network output $N(\boldsymbol{x}')$, with t as the target label:

$$\text{loss}_t(\boldsymbol{x}) = \text{Cross-Entropy}\left((\text{softmax} \circ N)(\boldsymbol{x}), t\right) = -\sum_{c=0}^{C-1} [c = t] \log(\text{softmax}(N(\boldsymbol{x}))_c)$$

$[\phi]$ (called Iverson Brackets) evaluats to 1 if the predicate ϕ is true and to 0 otherwise.

1. Show that the statement (2) is wrong by giving a counterexample.

2. To correct this oversight you will need to adjust the definition of $\text{obj}^t(\boldsymbol{x}')$. Make these adjustments.

3. What is the additional key constraint in the Carlini-Wagner attacks in comparison to FGSM?

[1] `pytorch.org`

[2] We provide short PyTorch examples along with the rest of the materials in the first exercises, but we strongly recommend that you familiarize yourself with it ahead of the project.

4. Why do the authors introduce the proxy objective function $\text{obj}^t(\boldsymbol{x} + \boldsymbol{\eta})$?

Problem 3. In the lecture we also looked at the optimization problem phrases by [1]:

$$\begin{aligned} &\text{find } \boldsymbol{\eta} \\ &\text{minimize } \|\boldsymbol{\eta}\|_p + c \cdot \text{obj}(\boldsymbol{x} + \boldsymbol{\eta}) \\ &\text{such that } x + \boldsymbol{\eta} \in [0, 1]^n \end{aligned}$$

Optimizing the norm directly can be problematic, especially in the case of $\|\cdot\|_\infty$. In this task we will investigate this and a surrogate term. To simplify the notation we will assume that $\boldsymbol{x}$ and $\boldsymbol{\eta}$ are n-vectors here (although they are usually matrices representing images). We define $h(\boldsymbol{\eta}) = \|\boldsymbol{\eta}\|_\infty$ and $g(\boldsymbol{\eta}) = \sum_{i=0}^{n} \max(\boldsymbol{\eta}_i - \tau, 0)$ for some constant τ.

1. Calculate $\frac{\partial}{\partial \boldsymbol{\eta}} h(\boldsymbol{\eta})$.

2. Calculate $\frac{\partial}{\partial \boldsymbol{\eta}} g(\boldsymbol{\eta})$.

3. Instantiate the above derivatives for $\boldsymbol{\eta} = (1.00001, 1.0, 1.0, 1.0, 0.001, 0.001)^T$ and for $\tau = 0.9$ and $\tau = 2.0$.

4. What is a problem when minimizing $h(\boldsymbol{\eta})$ with gradient decent?

5. Does $g(\boldsymbol{\eta})$ suffer the same problem?

References

[1] Nicholas Carlini and David A. Wagner. "Towards Evaluating the Robustness of Neural Networks". In: *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. IEEE Computer Society, 2017, pp. 39–57. DOI: 10.1109/SP.2017.49. URL: https://arxiv.org/abs/1608.04644.