Deep Learning (83882) - Ex 1

Due: 07.12.2024, 11:59pm

Q1

Consider a multivariate logistic regression problem. Recall that given a vector $\mathrm{x} \in \mathbb{R}^d$, and a parameter matrix $W \in \mathbb{R}^{d \times c}$, where c is the number of classes, the model can be written in the following manner using the Softmax function:

$$softmax(\mathrm{z})_{[i]} = \frac{exp(\mathrm{z_i})}{\sum_j exp(\mathrm{z_j})},$$

where, $\mathrm{z} = W^T\mathrm{x}$.

1. Show that $softmax(\mathrm{z}) = softmax(\mathrm{z} + m \cdot \mathbf{1})$ for every scalar m, where $\mathbf{1}$ is a vector of ones of appropriate size.
2. For $c = 2$, show that the Sigmoid function is equivalent to the Softmax function.
3. Present an alternative to the Sigmoid function which also maps from the real line to the $[0, 1]$ interval (any valid solution will be acceptable here).

Q2

Let $\mathrm{x} \in \{0, 1\}^2$ be an input vector. Consider the following model (scalar function):

$$f(\mathrm{x}) = \mathrm{w}^T\mathrm{h} + b_2$$
$$\mathrm{h} = max(U^T\mathrm{x} + \mathrm{b}_1, 0)$$

Where $U \in \mathbb{R}^{2\times2}, \mathrm{b}_1 \in \mathbb{R}^2, \mathrm{w} \in \mathbb{R}^2, b_2 \in \mathbb{R}$, and the max is taken element-wise.

Suppose we would like to represent with $f(\mathrm{x})$ the XOR function, defined as:

$$XOR(0, 0) = 0,$$
$$XOR(0, 1) = 1,$$
$$XOR(1, 0) = 1,$$
$$XOR(1, 1) = 0,$$

using the rule $sign(f(\mathrm{x}))$, that is, the answer is 1 if $f(\mathrm{x}) \geq 0$ and 0 if $f(\mathrm{x}) < 0$.

1. Find a suitable set of parameters for this task. A guess is fine, but show that indeed it solves the above task.
2. Will it be possible to represent the XOR function if we replace the max function with the identity function (i.e., $h = U^T\mathrm{x}$)? If so, show how. If not, explain why not.
3. What happens if we use $ReLU(U^T\mathrm{x})$ instead of max? Analyze whether the XOR function can still be represented.

Q3

Using Numpy only, implement the model described in Q2 and learn an optimal set of parameters using the Gradient Descent (GD) algorithm.

1. Create a dataset consisting of the following 4 examples $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$, and assign them the following labels $\{-1, 1, 1, -1\}$ correspondingly.
2. We will use the squared loss to optimize the parameters: $\mathcal{L}(y, f(\mathrm{x})) = (y - f(\mathrm{x}))^2$.

3. Plot the loss value as a function of the number of epochs.
4. Additionally, randomly choose two trainable parameters, and plot their evolution epochs.

Note that you may need several random initializations to converge to the optimal solution (the one that correctly classifies all examples).

Submission Instructions

Please submit a report with the answers to Q1 and Q2, the plots for Q3, and your code. Add to the report clear instructions how to run your code. Additionally:

1. Code should include comments explaining key steps of the implementation.
2. The report should include a short section on the challenges you faced during the implementation.
3. Code should be submitted in .py or .ipynb file format only.
4. Add to the report your name and ID.
5. Pack your submission files with your favorite file archiver (e.g., .rar, .zip).
6. The archive name should be your ID. If the exercise is done in pairs, the name of the file should be in the following format ID1_ID2. Only one member needs to submit the solution.

Good Luck
Ran