Deep Learning for Image Analysis

DL4IA – Report for Assignment 1

Student Linus Falk

March 22, 2023

1 Introduction

First assignment in the course Deep learning for image analysis

2 Mathematical exercises

Given the linear regression model:

$$z_i = \sum_{j=1}^{p} w_j x_{ij} + b \tag{1}$$

with the cost function

$$\begin{aligned} J &= \frac{1}{n} \sum_{i=1}^{n} L_i \\ \text{where } L_i &= (y_i - z_i)^2 \end{aligned} \tag{2}$$

Exercise 1.

$$\frac{\partial J}{\partial w_j} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial J}{\partial z_i} \frac{\partial z_i}{\partial w_j}$$

$$\frac{\partial J}{\partial b} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial J}{\partial z_i} \frac{\partial z_i}{\partial b}$$

Exercise 2.

$$\frac{\partial J}{\partial z_i} = \frac{1}{n} \sum_{i=1}^{n} 2(L_i)$$

$$\frac{\partial z_i}{\partial b} = \frac{\partial}{\partial b} \sum_{j=1}^{p} (w_j x_{ij} + b) = 1$$

$$\frac{\partial z_i}{\partial w_j} = \frac{\partial}{\partial w_j} \sum_{j=1}^{p} (w_j x_{ij} + b) = \sum_{j=1}^{p} x_{ij}$$

3 Code exercises

For exercises where you are asked to write code, include the important functions and steps that you have implemented. For functions it is good practise to always have a small description in the beginning of the function, explaining what the function does, what the arguments are and what is returned. In LaTeX you can for instance use the *minted* package to include code (read more about minted here).

Exercise 3. *Implement a gradient descent algorithm ...*

Based on the derivations given in exercise 1 and 2, the following functions are implemented to perform gradient descent for linear regression.

- The *initialize* function is used to initialize the weights and offsets by method X.
- The *cost* function takes ... and ... as input and calculates the cost according to eq. (2).

```
def initialize(arg1, arg2, ...):
    """ param arg1: description of argument 1
        param arg2: description of argument 2
        ...s

        return: description of what the function returns
    """

    w = ...
    b = ...


    return w, b

def cost(arg1, arg2, ...):
    """ param arg1: description of argument 1
        param arg2: description of argument 2
        ...

        return: description of what the function returns
    """

    J =

    return J
```

The weights and offsets are optimized to minimize the cost as follows:

```
w,b = initialize(*args) #Initialize weights and offsets
n_iterations = 1000 #Number of gradient descent iterations

for it in range(n_iterations):

    ...

    J = cost(*args)
```

4 Results

When presenting results, include plots of the optimization process, quantitative scores for your models and other relevant stuff to show the performance of your models. An easy to use and well documented library in python is *Matplotlib* (link). For plots, **always** write a caption explaining what is shown in the plot, and also what can be observed/concluded from the plot. Remember to always have labels on the axes, and when suitable a legend (if multiple plots in same figure). Make the plots of a suitable size to be visible (if printed). See Figure 1 for an example.

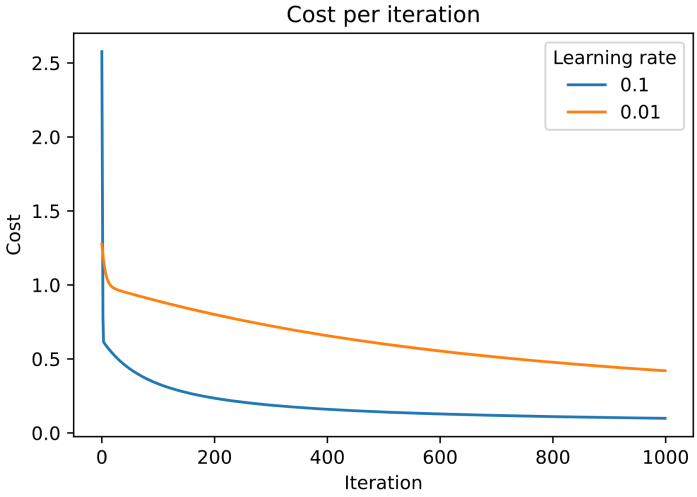

Figure 1: Plot showing how the cost (2), evaluated on the training sets, is changing with with number of iterations. It can be seen that...

References

[1] Adams, Douglas (1979). The Hitchhiker's Guide to the Galaxy. Pocket Books. p. 3.