Deep Learning PhD course: Pre-course assignment

Niklas Wahlström

January 7, 2021

Due: 1st of March 2021, 23:59

In this pre-course assignment, you will implement the linear regression model by using gradient descent.[1] In the course, we will extend this assignment to an implementation of a full neural network. Hence, this initial code will serve as a good base for the first hand-in assignment in the course.

1 Linear regression with gradient descent

Consider a dataset $\{\mathbf{x}_i, y_i\}_{i=1}^n$. Each input is a vector $\mathbf{x}_i = [x_{i1}, \dots, x_{ip}]^\mathsf{T}$ and each output $y_i \in \mathbb{R}$ is a scalar. We want to find a model for the output using linear regression. For one data point $i \in \{1, \dots, n\}$, the linear regression model can be described as

$$z_i = \sum_{j=1}^{p} w_j x_{ij} + b = \mathbf{w}^\mathsf{T}\mathbf{x}_i + b, \tag{1a}$$

where the weight vector $\mathbf{w} = [w_1, \dots, w_p]^\mathsf{T}$ and the offset b are the parameters. The cost J is computed by summing the following loss over all training data points

$$L_i = (y_i - z_i)^2, \tag{1b}$$

$$J = \frac{1}{n}\sum_{i=1}^{n} L_i. \tag{1c}$$

To train this model, we need access to the gradient of the cost function with respect to the model parameters, i.e. $\frac{\partial J}{\partial w_1}, \dots, \frac{\partial J}{\partial w_p}$, and $\frac{\partial J}{\partial b}$, which you will derive below.

Exercise 1.1 *Based on the model in* (1)*, derive expressions for*

$$\frac{dJ}{db} \quad \textit{and} \quad \frac{dJ}{dw_j} \quad \textit{in terms of} \quad \frac{\partial J}{\partial z_i}, \quad \frac{dz_i}{db}, \quad \textit{and} \quad \frac{dz_i}{dw_j}. \tag{2}$$

Exercise 1.2 *Based on the model in* (1)*, derive expressions for*

$$\frac{\partial J}{\partial z_i}, \quad \frac{dz_i}{db}, \quad \textit{and} \quad \frac{dz_i}{dw_j}. \tag{3}$$

Exercise 1.3 *Implement a gradient descent algorithm (Lindholm et al. 2020, Chp 5.2) that minimizes the cost J with respect to the parameters $\mathbf{w}, b$ based on a training data set $\{\mathbf{x}_i, y_i\}_{i=1}^n$. The implementation should involve the following functionalities:*

- **Initialize** *Write a function that initializes the parameters $\mathbf{w}, b$. For linear regression it is sufficient to initialize all parameters with zeros.*
- **Compute cost and gradients** *Compute the cost J and the gradient of the cost function with respect to all parameters $\frac{\partial J}{\partial b}$ and $\frac{\partial J}{\partial \mathbf{w}}$. For this, you need the mathematical expression in* (1), (2) *and* (3).
- **Optimize** *Update the parameters iteratively with gradient descent.*

[1] A linear regression problem you can also solve analytically using the normal equations. However, for the purpose of it being an initial code that you can extend in following assignments, you should solve it here using an gradient descent.

- **Predict** *Using the trained model, predict* z_i *based on the corresponding input* $\mathbf{x}_i$.

Exercise 1.4 *Evaluate the model on the* `Auto.csv` *dataset*[2]. *Treat* `mpg` *(miles per gallon) as your output* y_i *and consider two different choices for your input* $\mathbf{x}_i$

1. *Only the input variable* `horsepower`
2. *All input variables except* `name`

As pre-processing steps to construct your dataset $\{\mathbf{x}_i, y_i\}_{i=1}^n$ *you need to*

1. *Remove all rows with* `NA` *values.*
2. *Normalize the input variables, for example by subtracting the mean and dividing with the standard deviation.*

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}, \qquad \bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}, \quad s_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}$$

Choose a suitable step length for the optimization. If you did the normalization of the inputs as suggested above, many step lengths will work.[3]

Fit two linear regression models for the two choices of inputs. For both of them, produce a plot of the cost J *versus iterations. The final cost should be approximately 23.9 and 11.6 for the two models, respectively.*

Also, include a plot with `horsepower` *vs* `mpg` *both for the data together with the line representing the linear regression model you got when only using* `horsepower` *as your input.*

Submit a pdf with answers to Exercise 1.1 and Exercise 1.2 and the requested plots in Exercise 1.4 as well as code attached to the pdf document.

A The auto.csv data

Origin: The dataset was used in the 1983 American Statistical Association Exposition.

Description: Gas mileage, horsepower, and other information for 392 vehicles.

Format: A data frame with 392 observations on the following 9 variables.

`mpg`: miles per gallon
`cylinders`: Number of cylinders between 4 and 8
`displacement`: Engine displacement (cu. inches)
`horsepower`: Engine horsepower
`weight`: Vehicle weight (lbs.)
`acceleration`: Time to accelerate from 0 to 60 mph (sec.)
`year`: Model year (modulo 100)
`origin`: Origin of car (1. American, 2. European, 3. Japanese)
`name`: Vehicle name

References

Lindholm, Andreas, Niklas Wahlström, Fredrik Lindsten, and Thomas B. Schön (2020). *Supervised Machine Learning*. URL: `https://smlbook.org`.

[2] Download: `https://statlearning.com/Auto.csv`
[3] However, if you didn't normalize the inputs, you will have a difficult time finding a suitable step length.