END3971 – Artificial Intelligence and Expert Systems

Mehmet G. Güler -
Ebru Geçici

Due 13.11.2022 @23.45 (Sharp)

Assignment 1 (10%)

You can work on the project with groups of 2-4 students (not more, not less). This is the first assignment of END3971 AI & ES lecture. You should submit it through online.yildiz.edu.tr by zipping your code, otherwise probably you will not be able to upload it. Every group must submit only one assignment and all of the students name should be written at the top of the jupiter file. Otherwise those who do not have their names cannot get any points.

Problem Definition:

In this assignment you will find the parameters of a linear regression with gradient descent. In particular, you will find the parameters of the following linear regression:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5 \qquad (1)$$

The data is given in an excel sheet attached to this assignment. You should not use the built in regression functions of python. You should write your own code to find the parameters.

Recall that we have covered (not all of them but some portion of it) the gradient descent in finding the parameters of a linear regression. In particular, for any function f(x), you can find the minimum of the function using gradient descent by the following algorithm:

$$x^k = x^{k-1} - \alpha f'(x^{k-1})$$

Assume you want to find the minimum point of $f(x) = x^2 - 4x + 4$. You start at initial point x^0 and assign to a random number, say $x^0 = 0$. Then take the derivative of the function and find its value at this point. Here, the derivative is $f'(x) = 2x - 4$. Hence $f'(0) = -4$. Then find the next point x^1 by the formula above, i.e., $x^1 = x^0 - (\alpha)(-4)$. Here alpha is the stepsize and it should be a parameter to your function, i.e., the user must be able to play with it. Here let's assume $\alpha = 0.1$. Then we have $x^1 = 0 - (0.1)(-4) = 0.4$.

For the linear regression, our aim is to minimize the cost function, i.e., $J(\theta)$. In regression, the cost function is given by:

$$J(\Theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i\right)^2$$

The algorithm for updating θ values using gradient descent algorithm is given as follows:

$$\theta_j^k := \theta_j^{k-1} - \alpha \frac{\partial J(\Theta^{k-1})}{\partial \theta_j}$$

Which reduces to following:

$$\theta_0^k := \theta_0^{k-1} - \alpha \frac{1}{m} \sum_{i=1}^{m} (1)\left(h_{\theta^{k-1}}(x^i) - y^i\right)$$

$$\theta_j^k := \theta_j^{k-1} - \alpha \frac{1}{m} \sum_{i=1}^{m} (x^i)\left(h_{\theta^{k-1}}(x^i) - y^i\right)$$

The yellow part is nothing but the derivative. Note that since the θ_0 it is shown separately since its derivative is different than other theta parameters. From now on I will drop the superscript k.

Well, you may not be able to understand it at your first look, but believe it or not, it's extremely easy. Let's try to find it for our case. For our case, the cost function is equal to:

$$J(\theta) = \frac{1}{(2)(100)}\sum_{i=1}^{100}\left(\theta_0 + \theta_1 x_1^i + \theta_2 x_2^i + \theta_3 x_3^i + \theta_4 x_4^i + \theta_5 x_5^i - y^i\right)^2$$

We have 100 data, hence m=100. Well, if you are patient and like mathematical derivations, you can take the derivative of the above function with respect to θ_j and get the following:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{100}\sum_{i=1}^{100} x_j^i\left(\theta_0 + \theta_1 x_1^i + \theta_2 x_2^i + \theta_3 x_3^i + \theta_4 x_4^i + \theta_5 x_5^i - y^i\right)$$

where, as a special case, we have $x_0^i = 1$ for every i. Note that, the part written in red is nothing but the prediction given by the regression line. Hence the value in the paranthesis is simply the error that you are making (*i.e., the prediction minus the real value*). What you are doing is: sum *weighted* errors where the weight of each error is x_j^i. In order to understand the value of x_j^i, consider the following partial derivative for θ_2:

$$\frac{\partial J(\theta)}{\partial \theta_2} = \frac{1}{100}\sum_{i=1}^{100} (x_2^i)\left(\theta_0 + \theta_1 x_1^i + \theta_2 x_2^i + \theta_3 x_3^i + \theta_4 x_4^i + \theta_5 x_5^i - y^i\right)$$

Note that here x_2^i stands for the second feature value of the i^th^ data point (recall we have 100 data points). The value of x_2^i changes for each of 100 data points in the summation.

For θ_0, on the other hand, we have the following

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{100}\sum_{i=1}^{100} (1)\left(\theta_0 + \theta_1 x_1^i + \theta_2 x_2^i + \theta_3 x_3^i + \theta_4 x_4^i + \theta_5 x_5^i - y^i\right)$$

Now, the algorithm for updating θ values using gradient descent algorithm is given as follows:

$$\theta_0 := \theta_0 - \alpha\frac{1}{100}\sum_{i=1}^{100} (1)\left(\theta_0 + \theta_1 x_1^i + \theta_2 x_2^i + \theta_3 x_3^i + \theta_4 x_4^i + \theta_5 x_5^i - y^i\right) \tag{2}$$

$$\theta_j := \theta_j - \alpha\frac{1}{100}\sum_{i=1}^{100} (x_j^i)\left(\theta_0 + \theta_1 x_1^i + \theta_2 x_2^i + \theta_3 x_3^i + \theta_4 x_4^i + \theta_5 x_5^i - y^i\right) \tag{3}$$

For example, for θ_1 we have:

$$\theta_1 := \theta_1 - \alpha\frac{1}{100}\sum_{i=1}^{100} x_1^i\left(\theta_0 + \theta_1 x_1^i + \theta_2 x_2^i + \theta_3 x_3^i + \theta_4 x_4^i + \theta_5 x_5^i - y^i\right)$$

For example, for θ_2 we have:

$$\theta_2 := \theta_2 - \alpha\frac{1}{100}\sum_{i=1}^{100} x_2^i\left(\theta_0 + \theta_1 x_1^i + \theta_2 x_2^i + \theta_3 x_3^i + \theta_4 x_4^i + \theta_5 x_5^i - y^i\right)$$

In order to get an intuition:

Step 0: You can start at a point where all theta values are equal to some arbitrary value, say 1.

Step 1: Then you make predictions for 100 data points using the theta values that you have.

Step2: For each θ_i

Step2.i: Calculate the weighted sum of these errors

Step3.i: Multiply it with $\frac{\alpha}{100}$ and subtract it from the current value of θ_i to find the new θ_i

Step4: Go to step 1

One critical point is, you should do your updates for the theta using the current values of ALL theta. That is, if you are in iteration 15 for example and trying to find θ_3, then don't use the values of θ_0, θ_1 and θ_2 that you found in step 15, but use the values that you found in the previous step, i.e. step 14. In another words, you should update theta values **simultaneously**.

To Do:

1. (70p) For the given dataset, find the optimal parameters of the regression given in equation (1). Use the first 70 data as your training set, and the remaining 30 data as your test set. Report the theta values and the corresponding cost function values.

2. (30p) We have covered the regularized regression in our lecture which penalizes the theta values with a parameter called lambda. In particular, its cost function is given as follows:

$$J(\Theta) = \frac{1}{(2)(100)} \sum_{i=1}^{100} \left(h_\theta\left(x^i\right) - y^i\right)^2 + \lambda \sum_{j=1}^{5} \theta_j^2$$

Use the gradient descent method to find the optimal parameters for $\lambda = 0.01$, $\lambda = 0.1$, and $\lambda = 1$ and pick the best model. Use first 60 data as your training set, the next 20 data as your CV set and the final 20 as your test set. Note that you should modify the gradient descent update equations in (2) and (3) to find the optimal parameters.

Please take α as a parameter, i.e., you should be able to take it as an input from the user and use a default value of 0.1 if no value is given.

Good luck.