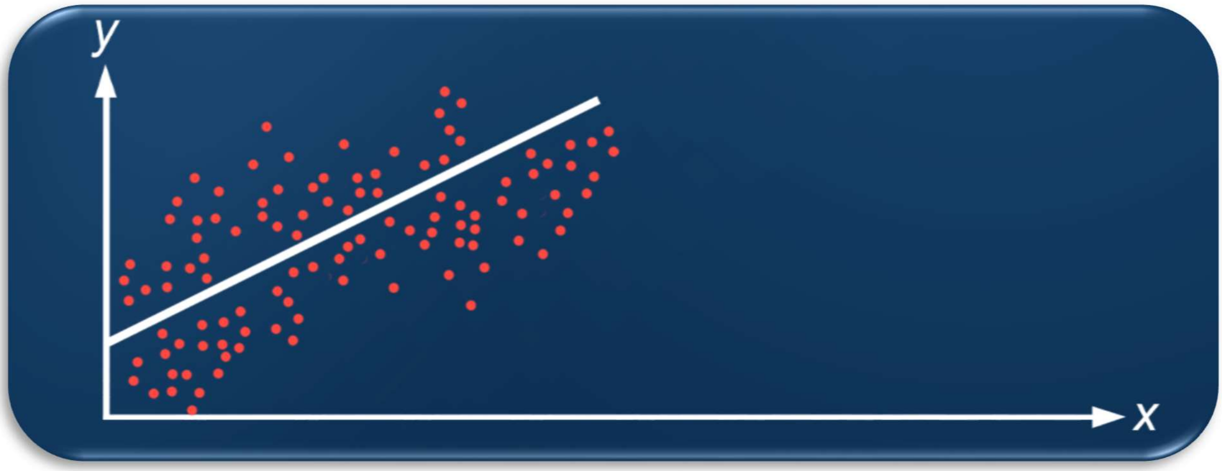
Artificial Neural Networks

Training vs Testing Error with Model Complexity

Name: Mert DUMANLI

Student ID: 160315002 – NORMAL



0. Linear_Regression_Model

RESIDUAL SUM OF SQUARES

Residual Sum Of Squares (RSS): In statistics, the residual sum of squares (RSS), also known as the sum of squared residuals (SSR) or the sum of squared estimate of errors (SSE), is the sum of the squares of residuals (deviations predicted from actual empirical values of data). It is a measure of the discrepancy between the data and an estimation model. A small RSS indicates a tight fit of the model to the data. It is used as an optimality criterion in parameter selection and model selection.

ONE EXPLANATORY VARIABLE

In a model with a single explanatory variable, RSS is given by:

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Where y_i is the i^{th} value of the variable to be predicted, x_i is the i^{th} value of the explanatory variable, and $f(x_i)$ is the predicted value of y_i (also termed $\hat{y}_i$). In a standard linear simple regression model, $y_i = a + bx_i + \varepsilon_i$ where a and b are coefficients, y and x are the regressand and the regressor, respectively, and ε is the error term. The sum of squares of residuals is the sum of squares of estimates of ε_i; that is

$$\mathrm{RSS} = \sum_{i=1}^{n} (\varepsilon_i)^2 = \sum_{i=1}^{n} (y_i - (\alpha + \beta x_i))^2$$

Where α is the estimated value of the constant term α and β is the estimated value of the slope coefficient b.

Matrix expression for the OLS residual sum of squares

The general regression model with n observations and k explanators, the first of which is a constant unit vector whose coefficient is the regression intercept, is y = Xβ + e.

Where y is an n x 1 vector of dependent variable observations, each column of the n x k matrix X is a vector of observations on one of the k explanators, β is a k x 1 vector of true coefficients, and e is an n x 1 vector of the true underlying errors. The ordinary least squares estimator for β is

$$X\hat{\beta} = y \iff X^{\mathrm{T}} X\hat{\beta} = X^{\mathrm{T}} y \iff \hat{\beta} = (X^{\mathrm{T}} X)^{-1} X^{\mathrm{T}} y.$$

The residual vector ê = $y - X\hat{\beta} = y - X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$ *; so the residual sum of squares is:*

$$\mathrm{RSS} = \hat{e}^{\mathrm{T}}\hat{e} = \|\hat{e}\|^2$$

,

(Equivalent to the square of the norm of residuals.) In full:

$$\mathrm{RSS} = y^{\mathrm{T}}y - y^{\mathrm{T}}X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y = y^{\mathrm{T}}[I - X(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}]y = y^{\mathrm{T}}[I - H]y$$

,

where H is the hat matrix, or the projection matrix in linear regression.

Relation with Pearson’s product-moment correlation

The least-squares regression line is given by y = ax + b, where

$$b = \bar{y} - a\bar{x}$$

$$a = \frac{S_{xy}}{S_{xx}}$$

$$S_{xy} = \sum_{i=1}^{n}(\bar{x} - x_i)(\bar{y} - y_i)$$

$$S_{xx} = \sum_{i=1}^{n}(\bar{x} - x_i)^2$$

.

Therefore,

$$\mathrm{RSS} = \sum_{i=1}^{n}(y_i - f(x_i))^2 = \sum_{i=1}^{n}(y_i - (ax_i + b))^2 = \sum_{i=1}^{n}(y_i - ax_i - \bar{y} + a\bar{x})^2$$

$$= \sum_{i=1}^{n}(a(\bar{x} - x_i) - (\bar{y} - y_i))^2 = a^2S_{xx} - 2aS_{xy} + S_{yy} = S_{yy} - aS_{xy} = S_{yy}(1 - \frac{S_{xy}^2}{S_{xx}S_{yy}})$$

Where $S_{yy} = \sum_{i=1}^{n}(\bar{y} - y_i)^2$.

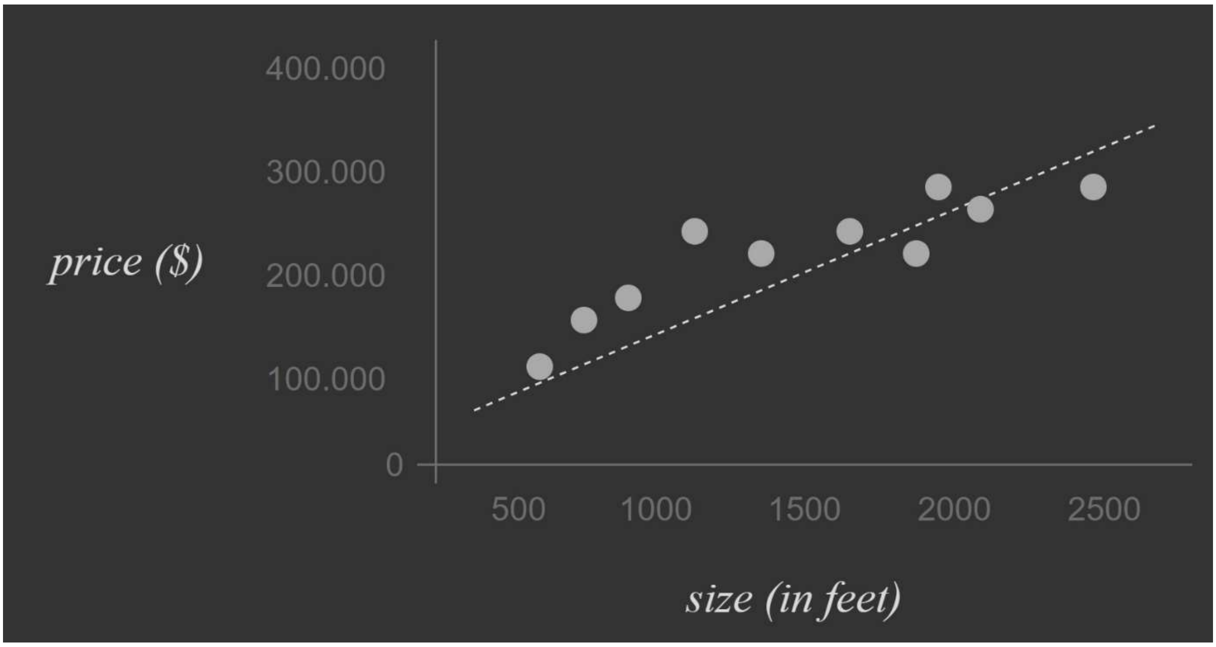
The Pearson product-moment correlation is given by $r = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}}$;

therefore, $\mathrm{RSS} = S_{yy}(1 - r^2).$

Linear Regression with One Variable

Finding the best-fitting straight line through points of a data set.

Linear regression is one of the most famous way to describe your data and make predictions on it. The picture 1. below, shows the housing prices from a fantasy country somewhere in the world. You are collecting real-estate information because you want to predict the house prices given, say, the size in square feet.



1. *House prices given their size.*

Given your input data, how can you predict any house price outside your initial data set? For example, how much a 1100 square feet house is worth? Linear regression will help answering that question: you shrink your data into a line (the dotted one in the picture above), with a corresponding mathematical equation. If you know the equation of that line, you can find any output (y) given any input (x).

Terminology and Notations

When you gathered your initial data, you actually created the so-called training set, which is the set of housing prices. The algorithm's job is to learn from those data to predict prices of new house. You are using input data to train the program, that's where the name comes from.

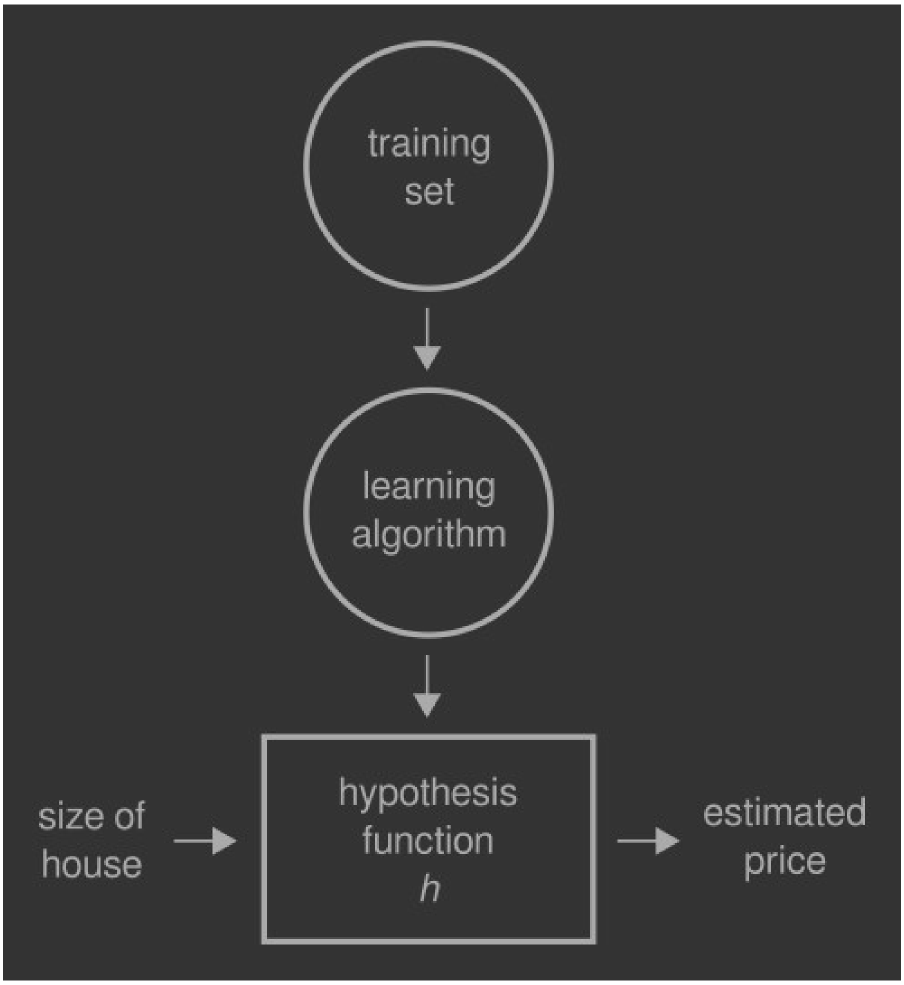
The training set can be summarized in a table, like the one you see below:

Size (feet^2) (x)	*Price (\$) (y)*
815	*165,000*
1510	*310,000*
2100	*410,000*
...	…

The number of training examples, or the number of lines in the table above, are noted as m; the input variable x is the single house size on the left column and y is the output variable, namely the price, on the right column.

The list (x,y) denotes a single, generic training example, while ($x^{(i)}$,$y^{(i)}$) represents a specific training example. So if I write ($x^{(2}$,$y^{(2)}$) I'm referring to the second row in the table above, where $x^{(2)}$ = 1500 and $y^{(2)}$ = 310,000.

Naming the algorithms parts



2. *Overview of a linear regression algorithm.*

The training set of housing prices is fed into the learning algorithm. Its main job is to produce a function, which by convention is called h (for hypothesis). You then use that hypothesis function to output the estimate house price y, by giving it the size of a house in input x.

The Hypothesis Function

The hypothesis function must have a formula, like any other function in the world. That is:

$h_\vartheta(x) = \vartheta_0 + \vartheta_1(x)$

Theta's (ϑ_i, in general) are the parameters of the function. Usually the theta subscript gets dropped and the hypothesis function is simply written as h(x).

That formula might look scary, but if you think about it it's nothing fancier than the traditional equation of a line, except that we use ϑ_0 and ϑ_1 instead of m and q. Do you remember?

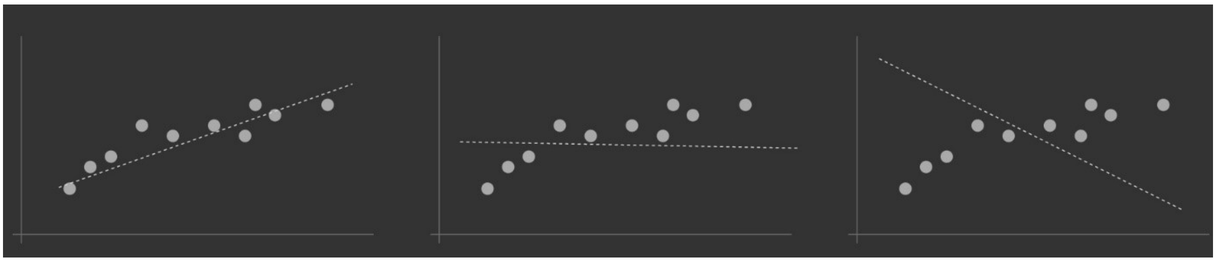
$f(x) = q + mx$

In fact the hypothesis function is just the equation of the dotted line you can see in the picture 1.

In our humble hypothesis function there is only one variable, that is x. For this reason our tasks is often called linear regression with one variable. Experts call it also univariate linear regression, where univariate means "one variable".

The Cost Function: a mathematical intuition

Well, at this point we know that there's a hypothesis function to be found. More precisely we have to find the parameters ϑ_0 and ϑ_1 so that the hypothesis function best fits the training data. If you recall how the equation of a line works, those two parameters control the slope and the height of the line. By tweaking ϑ_0 and ϑ_1 we want to find a line that represents at best our data. Picture 3. below shows that I mean:



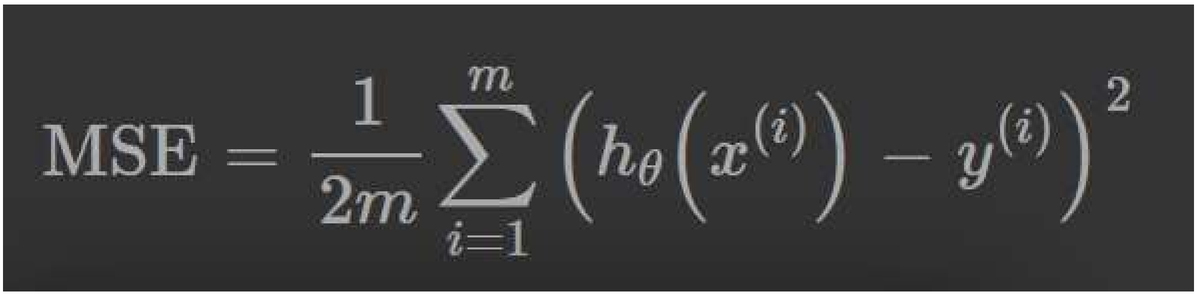
3. *Varying the value of ϑ_0 and ϑ_1 provide different outcomes: from good (left) to bad (right).*

We definitely want something like the first example in the picture above. So how to find proper values for ϑ_0 and ϑ_1? You certainly recall that in our training set we have several examples where we know the size of the house x and the actual price of the house y. We know those prices and sizes because we previously took a survey for those data. So the idea in a nutshell: let's try to choose the hypothesis function parameters so that at least in the existing training set, given the x as input parameter to the hypothesis function we make reasonable accurate predictions for the y values. Once we are satisfied, we can use the hypothesis function with its pretty parameters to make predictions on new input data.

From a mathematical point of view I want that, for each i-th point in my data set, the difference $h_0(x^{(i)}) – y^{(i)}$ is very small. Here $h_0(x^{(i)})$ is the prediction of the hypothesis when it is input the size of house number i, while $y^{(i)}$ is the actual price of the house number i. If that difference is small, it means that the hypothesis has made an accurate prediction, because it's similar to the actual data.

The operation I described so far is a part of the so-called mean squared error function (MSE), a function that does exactly that we want: it measures how close a fitted line is to some data points. The smaller the MSE, the closer the fit is to the data. Actually there are many other functions that work well for such task, but the MSE is the most commonly used one for regression problems.

If I plug our data into the MSE function, our final formula looks like that:



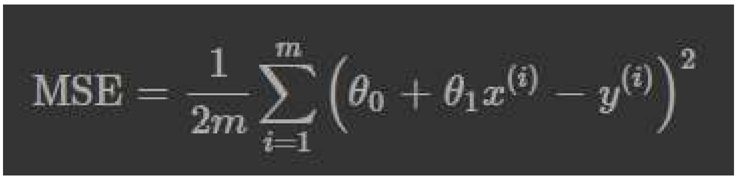
$$\mathrm{MSE} = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Note the 1 / (2m) and the summation part: we are properly computing a mean. That 2 at the denominator will ease some calculations in future steps. Also, the squaring is done so negative values do not cancel positive values.

Let me now expand the above equation. Since

$h_\vartheta(x^{(i)}) = \vartheta_0 + \vartheta_1 x^{(i)}$

Then



$$\mathrm{MSE} = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

By convention we would define a cost function (aka loss function) J that is just the above equation written more compact:

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)^2$$

Now, we want to find good values of ϑ_0 and ϑ_1, so good that the above cost function can produce the best possible values, namely the smallest ones(because small values mean less errors). This is an optimization problem: the problem of finding the best solution from all feasible solutions. It can be written as

$$\underset{\theta_0, \theta_1}{\text{minimize}} \; J(\theta_0, \theta_1)$$

Applying the Cost Function

Let's now feed our theoretical function with some real data. To better understand how the cost function works I will temporally set $\vartheta_0 = 0$ so that our hypothesis function looks like

$$h_\theta(x) = \theta_1 x$$

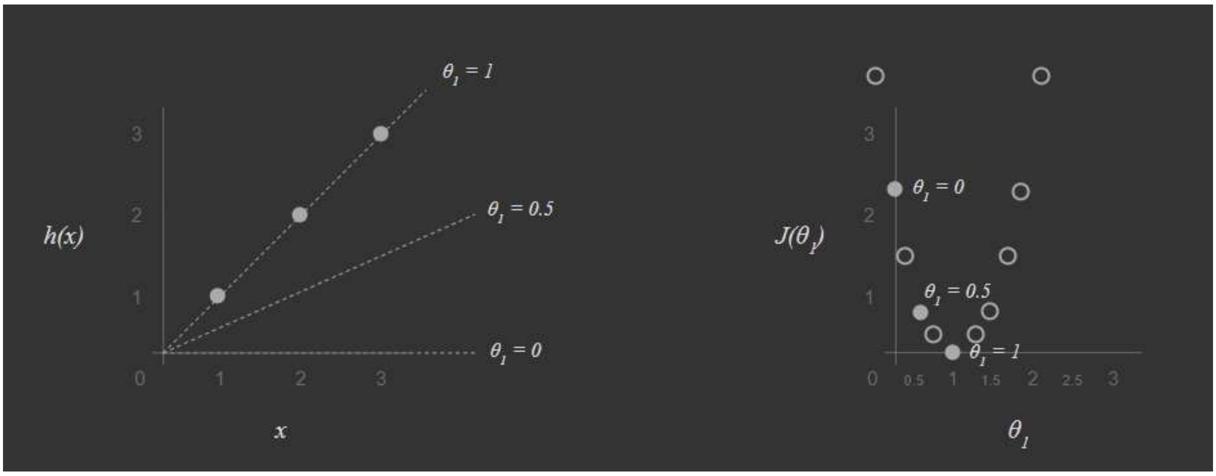
and the minimization task like

$$\underset{\theta_1}{\text{minimize}} \; J(\theta_1)$$

This will help a lot with cost function visualization: keeping $\vartheta_0 \neq 0$ would require a three-dimensional plot that initially would be a source of annoyance. Just remember: with $\vartheta_0 = 0$ the hypothesis function becomes a line passing through the origin (0, 0) while ϑ_1 controls the slope.

Cost Function with One Variable

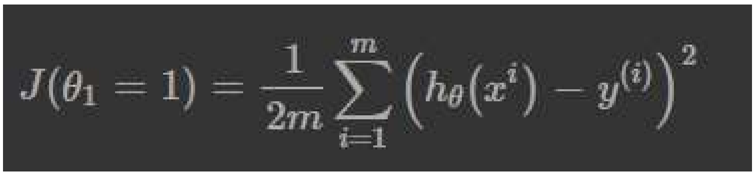
Picture 4. Shows the relationship between the hypothesis function and the cost function. Let's suppose that our data is made of three points as you may see in the leftmost plot.



4. Hypothesis function (left) and cost function (right) with one variable.

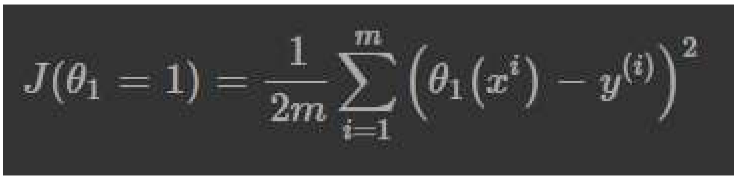
Changing the values of ϑ_1, namely changing the slope of the hypothesis function produces points in the cost function.

For example, with $\vartheta_1 = 1$:



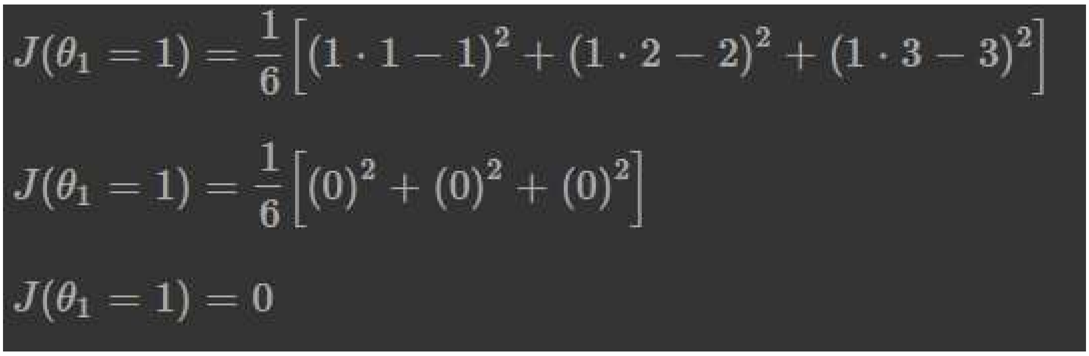
$$J(\theta_1 = 1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^i) - y^{(i)}\right)^2$$

Since $\vartheta_0 = 0$ I can write the hypothesis function like that:



$$J(\theta_1 = 1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_1(x^i) - y^{(i)}\right)^2$$

Now let's plug some numbers in:

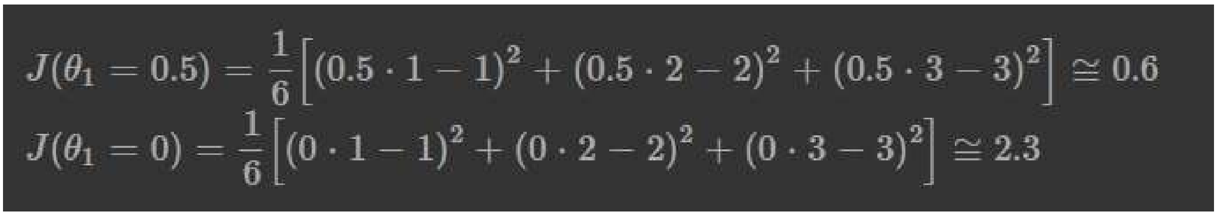


$$J(\theta_1 = 1) = \frac{1}{6}\left[(1 \cdot 1 - 1)^2 + (1 \cdot 2 - 2)^2 + (1 \cdot 3 - 3)^2\right]$$

$$J(\theta_1 = 1) = \frac{1}{6}\left[(0)^2 + (0)^2 + (0)^2\right]$$

$$J(\theta_1 = 1) = 0$$

In words: for ϑ_1 = 1, the cost function has produced a value of 0. Let's try with the other two values:



$$J(\theta_1 = 0.5) = \frac{1}{6}\left[(0.5 \cdot 1 - 1)^2 + (0.5 \cdot 2 - 2)^2 + (0.5 \cdot 3 - 3)^2\right] \cong 0.6$$
$$J(\theta_1 = 0) = \frac{1}{6}\left[(0 \cdot 1 - 1)^2 + (0 \cdot 2 - 2)^2 + (0 \cdot 3 - 3)^2\right] \cong 2.3$$

In picture 4. you may find the values 0, 0.6 and 2.3 plotted as full dots on the cost function. The empty ones are values from other theta's, not shown in the hypothesis function but computed separately: they reveal that the cost function is actually a parabola with its minimum at 0.

You can read the whole picture in reserve: every point of the cost function corresponds to a specific slope of the hypothesis function. We decided to take the best value, namely the minimum of the cost function. Looking at the curve of the cost function, the value that minimizes J(ϑ_1) is ϑ_1 = 1: that value means the best slope of the hypothesis function for our particular training set.

Python Code’s writing

I wrote the python code in Spyder (Pyhton 3.7).

1. *I set lists/arrays all data.*
2. *I divided to 1000 all data for to avoid error.*
3. *I created a list. The list’s name is wler. I set 1.0 for all values of w.*
4. *I found the expected values with predicted_y0,predicted_y1,…,predicted_y27 and set it in the predicted_e[].*
5. *I found the difference between the actual y values and the expected y values with the brackets function and set it in the u[] list.*
6. *I found the result of the summation symbol with gradient_descent function and set it in the toplamfark_gd[] list.*
7. *I found values for different w coefficients with while loop and add p, as soon as, I set the values for different coefficients w to the gd[] list.*
8. *I found RSS results with RSS function and set rss[] list. Then, I gather. And I used this result on arbiter() function.*
9. *I updated w coefficient results with arbiter function and update wler[] list.*
10. *I made 20 times.(Iteration)*
11. *After, I found RMSE with RSS function and set rmse[] list. Then, I gather.(top2)*
12. *I used “direction” to specify the direction when translating. (I used to arbiter function.)*
13. *I did “import math” to calculate RMSE.*
14. *I did “import csv” to receive data.*
15. *I did “import codecs” to change the format of the file.*
16. *I did “import random” to generating random values, used 80% of the data to randomly receive.*

NOTES

- *While importing data, I deleted the first line of the excel file and dragged the data up 1 row.*
- *I could do 30 iterations because I’ve float variables and gives an “overflow” error when more by python.*
- *I value manually for equations of different degrees because of come true the problem as above.(overflow)*

References

- https://www.internalpointers.com/post/linear-regression-one-variable
- https://en.wikipedia.org/wiki/Residual_sum_of_squares
- *©2015 Emily Fox & Carlos Guestrin- Machine Learning Specialization University of Washington*
- https://belgeler.yazbel.com/python-istihza/listelerin_ve_demetlerin_metotlari.html
- https://www.datafloyd.com/tr/pyhton-matplotlib-kutuphanesini-kullanarak-veri-gorsellestirme-temel-grafikler%C2%B6/