**Machine Learning Algorithms(1) — Simple Linear Regression**

[[Kasun Dissanayake](https://kasunprageethdissanayake.medium.com/?source=post_page-----4791764f5b2d--------------------------------)](https://kasunprageethdissanayake.medium.com/?source=post_page-----4791764f5b2d--------------------------------)

[[Towards Dev](https://towardsdev.com/?source=post_page-----4791764f5b2d--------------------------------)](https://towardsdev.com/?source=post_page-----4791764f5b2d--------------------------------)

[Kasun Dissanayake](https://kasunprageethdissanayake.medium.com/?source=post_page-----4791764f5b2d--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2F4a3ca0851dfe&operation=register&redirect=https%3A%2F%2Ftowardsdev.com%2Fmachine-learning-algorithms-1-simple-linear-regression-4791764f5b2d&user=Kasun+Dissanayake&userId=4a3ca0851dfe&source=post_page-4a3ca0851dfe----4791764f5b2d---------------------post_header-----------)

Published in

[Towards Dev](https://towardsdev.com/?source=post_page-----4791764f5b2d--------------------------------)

·

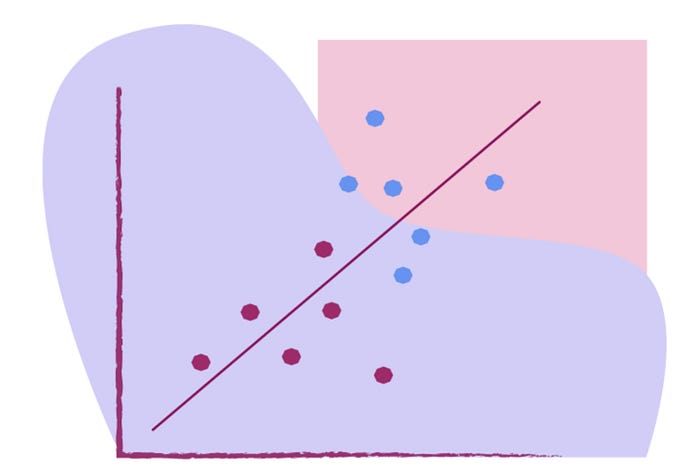
11 min read

·

Oct 16, 2023

622

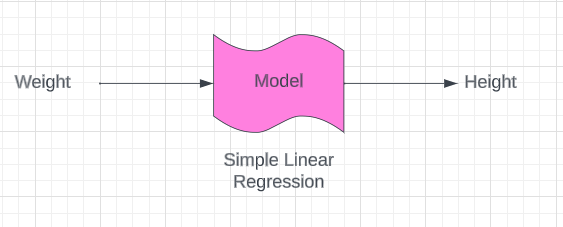
3



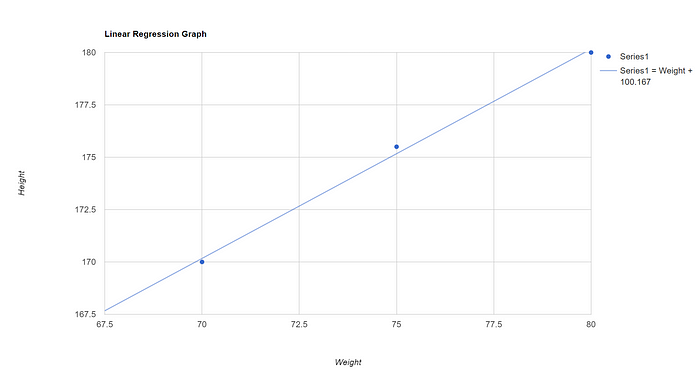
Inthis article we will learn our first machine learning algorithm called **Simple Linear Regression.** This is an important algorithm because the techniques learned in this algorithm also be applicable in deep learning when you are probably learning the first neural network which is called an[**Artificial neural network**](https://www.sciencedirect.com/topics/engineering/artificial-neural-network)**.**I will try to break down this into separate modules so that you can understand it in a better way(So this is the **part 1** of Machine Learning series). **Linear Regression** as the name suggests in supervised machine learning, regression problem statements can definitely be solved with the help of similar linear regression. Now what exactly simple linear regression algorithm is, let’s say that I have a dataset and this particular data set has features like **Weight and Height.**



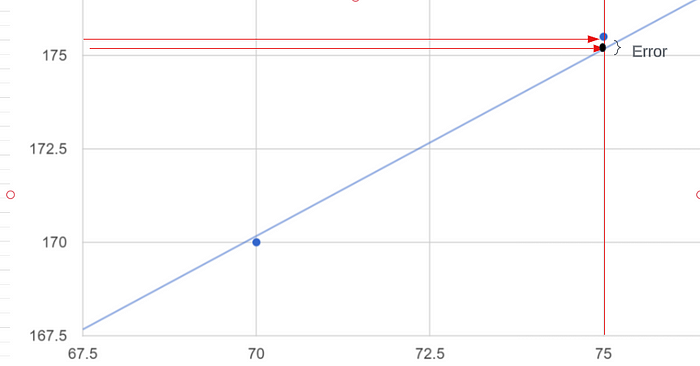
Let’s say if the weight is somewhere around **74 kg** my height may be **170 cm** if the weight is**80 kg** the height may be**180 cm** if the weight is around **75 kg** the height may be **175.5 cm**. Let’s say in this kind of data set our main aim is to train a model whenever we give our new weight, The **model should be able to predict the height** and now you just see this feature is basically our independent feature(weight) and this feature is specifically about **output or dependent feature**. So this is what we are planning to do we are going to train with the help of simple linear regression. So why do we say it as simple linear regression so that by looking at this you can understand how many input features are over here? We have **one input feature and one output feature**. Whenever we have this **one input feature then we say it is simple linear regression**. If we have **multiple input features then we can call it multiple linear regression**. So in this tutorial, we trying to specify a model and we will train it with specific data, and later on this model should be able to predict the height.



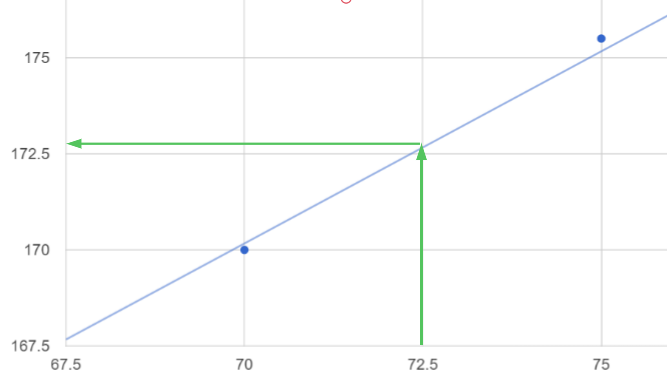
So concerning this particular data set let me plot some points. Let’s say there are some points that are plotted like this so with the help of regression what we are trying to do is to create a **best-fit line** and this best-fit line will actually help to make the prediction for the weight. So let’s say how the prediction happens once we get this best-fit line and this best-fit line should be created in such a way that you know the distance between the true points.



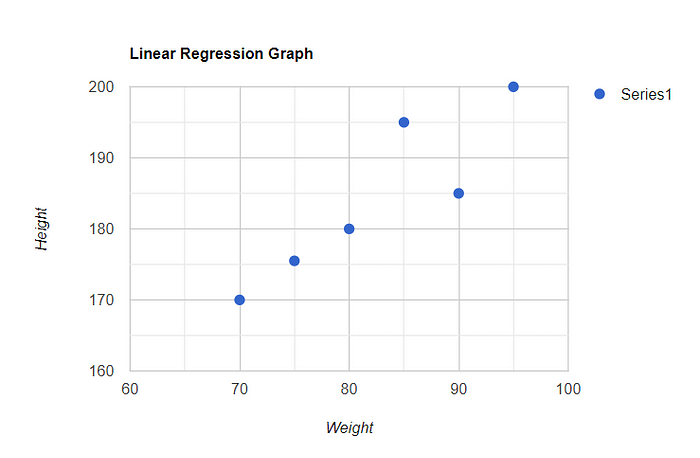
The true points mean the output of my data which is 170 cm, 180 cm, and 175.5 cm. The distances between these predicted points are basically the **error**.



So these are my true points(**blue color points**) and the blue color line is a **predicted line** using these points. We are just creating a predicted line which is the **best-fit line** whenever we get a new data point let’s say a particular weight of 72.5kg how do we predict our **output**which means **what should be the height**? We just draw a line from the x-axis to the predicted line which is my best-fit line and from the best-fit line I am going to draw another line to my y-axis So this line meets my height in the y-axis which is the output for my given input.



This is what we do in simple linear regression. Let’s try to understand what is the **mathematical equation** of this particular best-fit line and what is the specific error with the best-fit line. Before that, we need to understand some of the notation that I am actually going to use while explaining this entire machine-learning algorithm. So I am going to draw another graph let’s say this is the x-axis and I have my **weight**in the **x-axis** and **height**in the **y-axis** and I have just randomly created some points and these points basically mention our data set.



I am going to train our specific model. And again we are planning to create a best-fit line here. To create this best-fit line we just need some equation which is,

y = mx + c

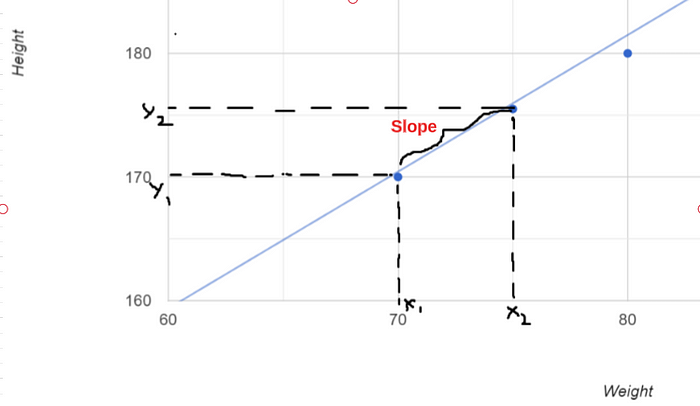
If you have seen some of the research papers they may also use something like,

Y=β0+β1x

You may have also seen an equation like,

hθ(x)=θ0+θ1x

I will be using the specific notation which is nothing but **hθ(x)=θ0+θ1x**. This equation is known as the **hypothesis**and is represented by the form**h(x)=θ0 +θ1x**(principally the same as y = mx + c) where θ0 (or c) and θ1 are (or m) parameters. We want to find the values of the parameters that will allow our hypothesis to best match the data. Now here X means my **independent feature**which is the **weight**. Please try to understand what is **θ0**and what is **θ1**. First of all, what exactly so **θ0, We** say it is an **Intercept**. why do we say it as Intercept? That is from simple mathematical calculation. Let’s assume my **X** value is **zero**, then what will happen is **hθ(x)=θ0**. As you can see in my graph the best-fit line meets the y-axis somewhere. So the point at which my best-fit line meets the y-axis and gets it as an **interceptor**. It means when the x-axis is zero which is the value of **θ0**. Now we know what is the meaning of **θ0**. It’s nothing but an **interceptor**. When we talk about the **slope or coefficient** what it indicates is the unique movement in the x-axis and what is the movement concerning the y-axis.



That is indicated by**θ1** in the equation. Suppose if I have many independent features then this equation becomes something like **hθ(x)=θ0 + θ1x1 + θ2x2 + .. + θnxn**.

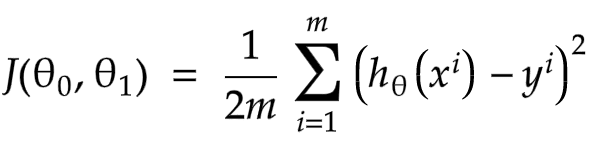
So finally we know we can predict a **y value** for a **given x value** using this equation. We denote this predicted point as **ŷ.**You know that y is our actual value of output. Now we can get an equation for the error using these 2 values.

Error = y - ŷ

Now we are going to come up with a best-fit line wherein when I try to calculate or do the summation of all these errors it should be minimal. Suppose there are multiple best-fit lines with different error summation values. You have to pick the best-fit line which has a **minimal error** summation value.

**Regression Cost Function**

Here we are going to find the optimized way to select the best-fit line. For this, we will be using the **cost function**. This cost function gives in a notation.



We have to create the best-fit line so that we can get the summation of all the specific errors and it should be minimal. So that is why we are taking this cost function in this specific way. **hθ(x)^i** is my **predicted point**. And **y^i** is my **truth point/true output**. When we do the subtraction we can get the error value here.



The reason we are squaring is because the technique of the cost function that we are using is a **mean squared error**. Are there different kinds of cost functions that exist? Yes, there are **mean absolute error(MAE) and root mean square error(RMSE)**.

the **θ0** denotes the **intercept**and **θ1** denotes the **slope**. You just need to continuously change the θ0 and θ1 values and try to find out the best-fit line for minimal error.

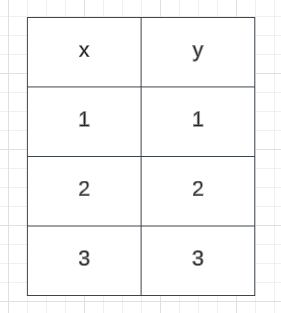
So what is the equation of the straight line,

hθ(x)=θ0+θ1x

From this, I am going to explain all this in a **2D diagram**to give a better idea of this theory. So I am going to assume that my **θ0 = 0**. Then my intercept will be zero and the best-fit line will pass through the origin. Now I can create my equation like this.

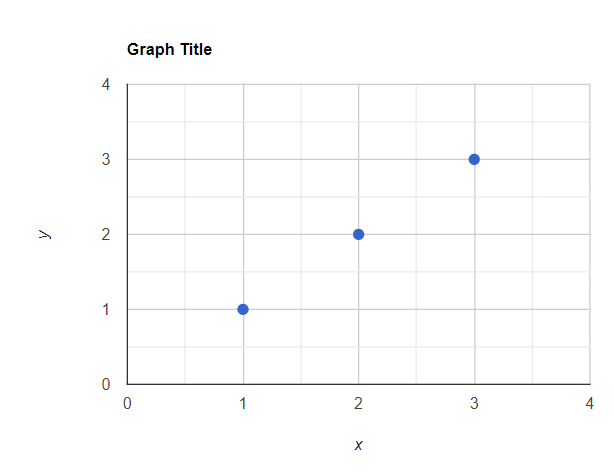
hθ(x)= θ1x : because my θ0 = 0

I am going to use this equation to get values to the hθ(x). Let’s consider this is my entire data set and I am trying to create a best-fit line and find the minimum error for this line.



Example Dataset

Let's plot a graph of these data. Now I am going to use the above equation to draw my best-fit line.

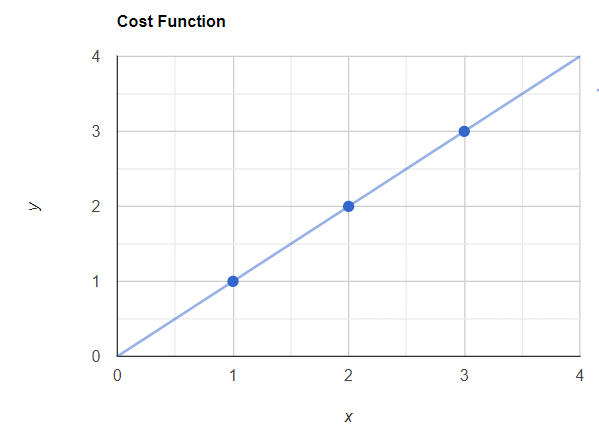


Graph with actual values

Now my slope is θ1 and let’s assume our **θ1= 1**. Later on, we will change the slope to get the different different best-fit lines to minimize the error.

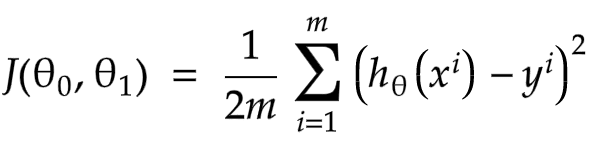
hθ(x)= θ1x  
Let θ1 = 1 (This is my slope value. Assumption this value equals to 1)  
Now according to the x values in the data hθ(x) values should be like this,  
x = 1 -> hθ(x) = 1   
x = 2-> hθ(x) = 2  
x = 3-> hθ(x) = 3

Now we can draw the best-fit line with these values and this line will pass through the origin(x = 0, y= 0).



Graph with actual values and best-fit line

Now you can see my predicted points and the truth points are overlapped. Now let's apply this cost function to this.



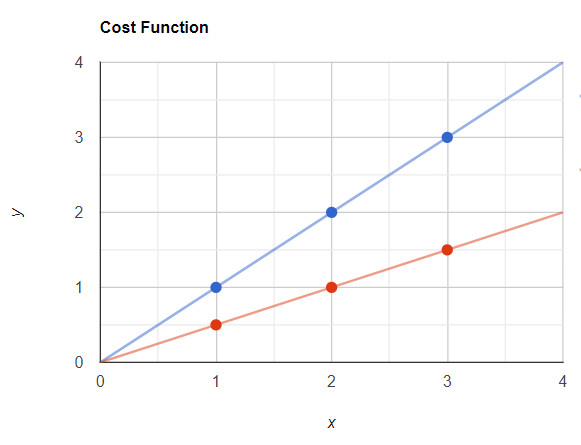
Here we assume like θ0 = 0 and the J(θ0, θ1) will be J(θ1)   
Now in the dataset I have 3 points. So that m will be 3. m=3  
Now the summation of i =1 to 3 means the entire summation of 1,2 and 3 value.  
So I will get the value by expanding this values like this,  
  
J(θ1) = 1/2\*3 [All 3 dataset Sum (predicted value - actual true value)^2]  
J(θ1) = 1/2\*3 [ (1 -1)^2 + (2 - 2)^2 + (3 - 3)^2}  
J(θ1) = 0

Now you know J(θ1) = 0 which means there is no error. This is correct because obviously there is no error because the best-fit line passes all true points.

Let's change our slope value to 0.5.

hθ(x)= θ1x  
Let θ1 = 0.5(This is my slope value. Assumption this value equals to 0.5)  
Now according to the x values in the data hθ(x) values should be like this,  
x = 1 -> hθ(x) = 0.5  
x = 2-> hθ(x) = 1  
x = 3-> hθ(x) = 1.5

Now we can draw the best-fit line with these values and this line will pass through the origin(x = 0, y= 0).



So here red points are my predicted points and blue points are my actual points. Now let’s calculate my error value using J(θ1).

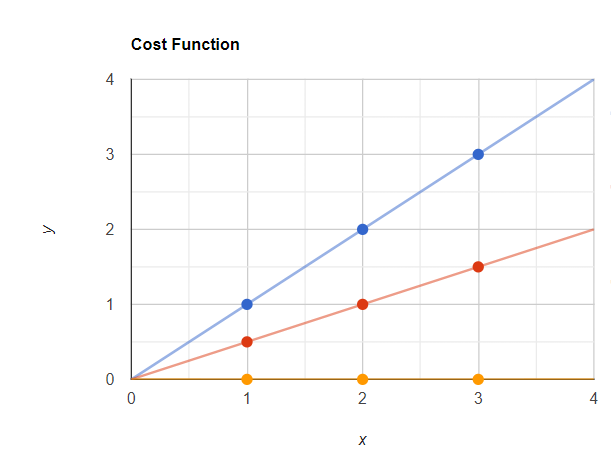
Here we assume like θ0 = 0 and the J(θ0, θ1) will be J(θ1)   
Now in the dataset I have 3 points. So that m will be 3. m=3  
Now the summation of i =1 to 3 means the entire summation of 1,2 and 3 value.  
So I will get the value by expanding this values like this,  
  
J(θ1) = 1/2\*3 [All 3 dataset Sum (predicted value - actual true value)^2]  
J(θ1) = 1/2\*3 [ (0.5 -1)^2 + (1 - 2)^2 + (1.5 - 3)^2}  
J(θ1) = 1/2\*3 [ (-0.5)^2 + (-1)^2 + (-1.5)^2}  
J(θ1) = 0.58

Now my J(θ1) value which is the error is 0.58. It is a larger value compared to the previous value.

Let’s change our slope value to **0**.

hθ(x)= θ1x  
Let θ1 = 0(This is my slope value. Assumption this value equals to 0)  
Now according to the x values in the data hθ(x) values should be like this,  
x = 1 -> hθ(x) = 0  
x = 2-> hθ(x) = 0  
x = 3-> hθ(x) = 0

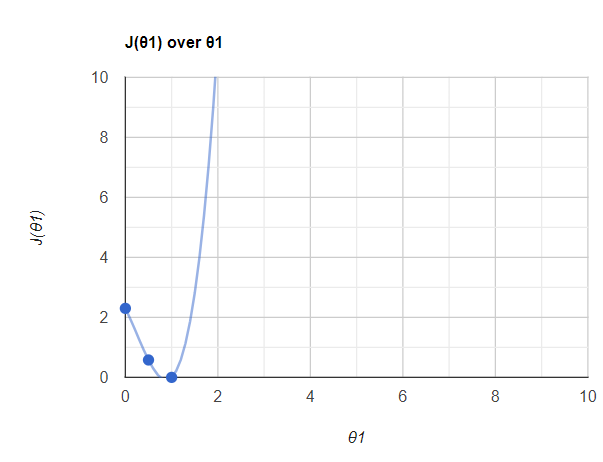
Now we can draw the best-fit line with these values and this line will pass through the origin(x = 0, y= 0).



Here now my predicted points are in yellow color and the actual points are in blue color. Now let’s calculate my error value using J(θ1).

Here we assume like θ0 = 0 and the J(θ0, θ1) will be J(θ1)   
Now in the dataset I have 3 points. So that m will be 3. m=3  
Now the summation of i =1 to 3 means the entire summation of 1,2 and 3 value.  
So I will get the value by expanding this values like this,  
  
J(θ1) = 1/2\*3 [All 3 dataset Sum (predicted value - actual true value)^2]  
J(θ1) = 1/2\*3 [ (0 -1)^2 + (0 - 2)^2 + (0 - 3)^2}  
J(θ1) = 1/2\*3 [ (-1)^2 + (-2)^2 + (-3)^2}  
J(θ1) = 2.3

Now let’s plot these J(θ1) values in a graph. It looks like this,



So here we have used 3 points and when we will use more J(θ1) and θ1 points to draw this you will get a graph like this. You know that at the point of **θ1 =1 my error is very low**. Actually, it is **zero**. So we can say that my **error is minimized** when we find this **θ1**value **θ1= 1**. We are calling this value as [**Global Minima**](https://wngaw.github.io/linear-regression/). The overall objective is to minimize the cost function by iterating through different values of **θ**. The lowest possible value of the cost function is also known as the global minimum. The final linear regression model will hold the values of θ that yield the lowest cost function. In the Global Minima, I will get my best-fit line now.

So this entire curve is called [**Gradient descent**](https://towardsdatascience.com/minimizing-the-cost-function-gradient-descent-a5dd6b5350e1). This will be important in deep learning techniques.

So I hope you will get a better understanding of Simple Linear Regression and Regression Cost Function. You will learn more about Convergence Algorithms in the next article.

**[Machine Learning Algorithms(2) — Convergence Algorithm and Multiple Linear Regression](https://towardsdev.com/machine-learning-algorithms-2-convergence-algorithm-and-multiple-linear-regression-858f37d4e94c?source=post_page-----4791764f5b2d--------------------------------" \t "_blank)**

[The previous article taught us about Simple Linear Regression and Cost Function. Here we are learning about the…](https://towardsdev.com/machine-learning-algorithms-2-convergence-algorithm-and-multiple-linear-regression-858f37d4e94c?source=post_page-----4791764f5b2d--------------------------------" \t "_blank)

[towardsdev.com](https://towardsdev.com/machine-learning-algorithms-2-convergence-algorithm-and-multiple-linear-regression-858f37d4e94c?source=post_page-----4791764f5b2d--------------------------------" \t "_blank)

Thank You!

[Simple Linear Regression](https://medium.com/tag/simple-linear-regression?source=post_page-----4791764f5b2d---------------simple_linear_regression-----------------)

[Regression Cost Function](https://medium.com/tag/regression-cost-function?source=post_page-----4791764f5b2d---------------regression_cost_function-----------------)

[Cost Function](https://medium.com/tag/cost-function?source=post_page-----4791764f5b2d---------------cost_function-----------------)

[Machine Learning](https://medium.com/tag/machine-learning?source=post_page-----4791764f5b2d---------------machine_learning-----------------)