Mathematics in AI

Fundamentals of ML:

In ML there are many models but initial fundamental model is linear regression which uses the prediction of the values using the deviation of the values from the line (Main curve).

For example

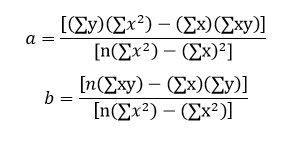
A linear regression line equation is written in the form of:

**Y = a + bX**

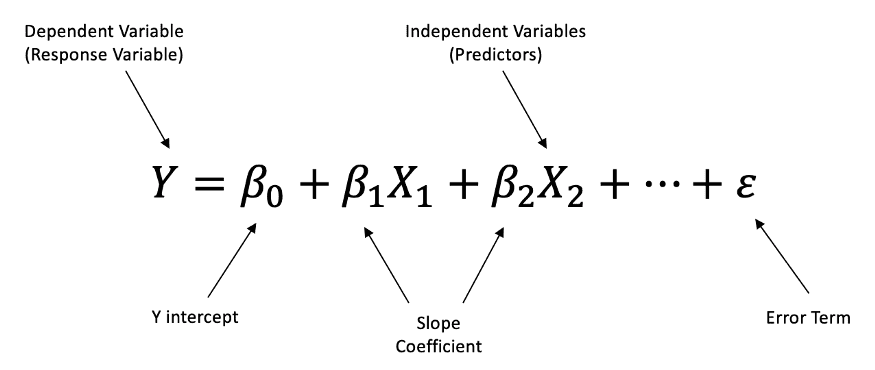
where X is the independent variable and plotted along the x-axis Y is the dependent variable and plotted along the y-axis. The slope of the line is b, and a is the intercept (the value of y when x = 0).

**Linear Regression Formula**

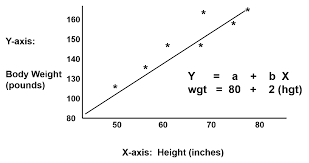
Linear regression shows the linear relationship between two variables. Now, here we need to find the value of the slope of the line, b, plotted in scatter plot and the intercept, a.



**Multi-Linear Regression Formula**



The Example of the graph is as shown



Classification works same as this it draws the line (curve) between the classes it also makes multi-class. There are many other models in real time like logistic regression

**Note : The more you dive into mathematics of ML calculation the more complex it will become**

**SOLVED EXAMPLE:**

1. Find a linear regression equation for the following two sets of data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x** | 2 | 4 | 6 | 8 |
| **y** | 3 | 7 | 5 | 10 |

**Sol:**

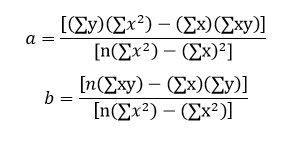
To find the linear regression equation we need to find the value of Σx, Σy, Σx and Σxy

Construct the table and find the value

|  |  |  |  |
| --- | --- | --- | --- |
| **x** | **y** | **x²** | **xy** |
| 2 | 3 | 4 | 6 |
| 4 | 7 | 16 | 28 |
| 6 | 5 | 36 | 30 |
| 8 | 10 | 64 | 80 |
| Σx = 20 | Σy = 25 | Σx² = 120 | Σxy = 144 |

The formula of the linear equation is y=a+bx. Using the formula we will find the value of a and b

Now put the values in the equation 1



a =( 25 × 120 – 20 × 144 ) / ( 4 × 120 – 400 )

a= 120 / 80

a=1.5

Put the values in the equation 2

b=( 4 × 144 – 20 × 25 ) / ( 4 × 120 – 400 )

b=76 / 80

b=0.95

Hence we got the value of a = 1.5 and b = 0.95

The linear equation is given by

Y = a + bx

Now put the value of a and b in the equation

Hence equation of linear regression is y = 1.5 + 0.95x

Let a new value (x , y) = (5 , y)

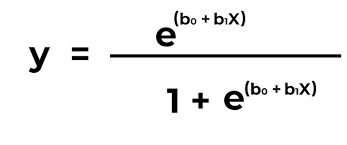
y = 1.5 + 0.95 \* (5)

y = 6.25

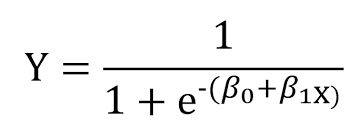
It is the new value for x = 5.

**Logistic Regression:**

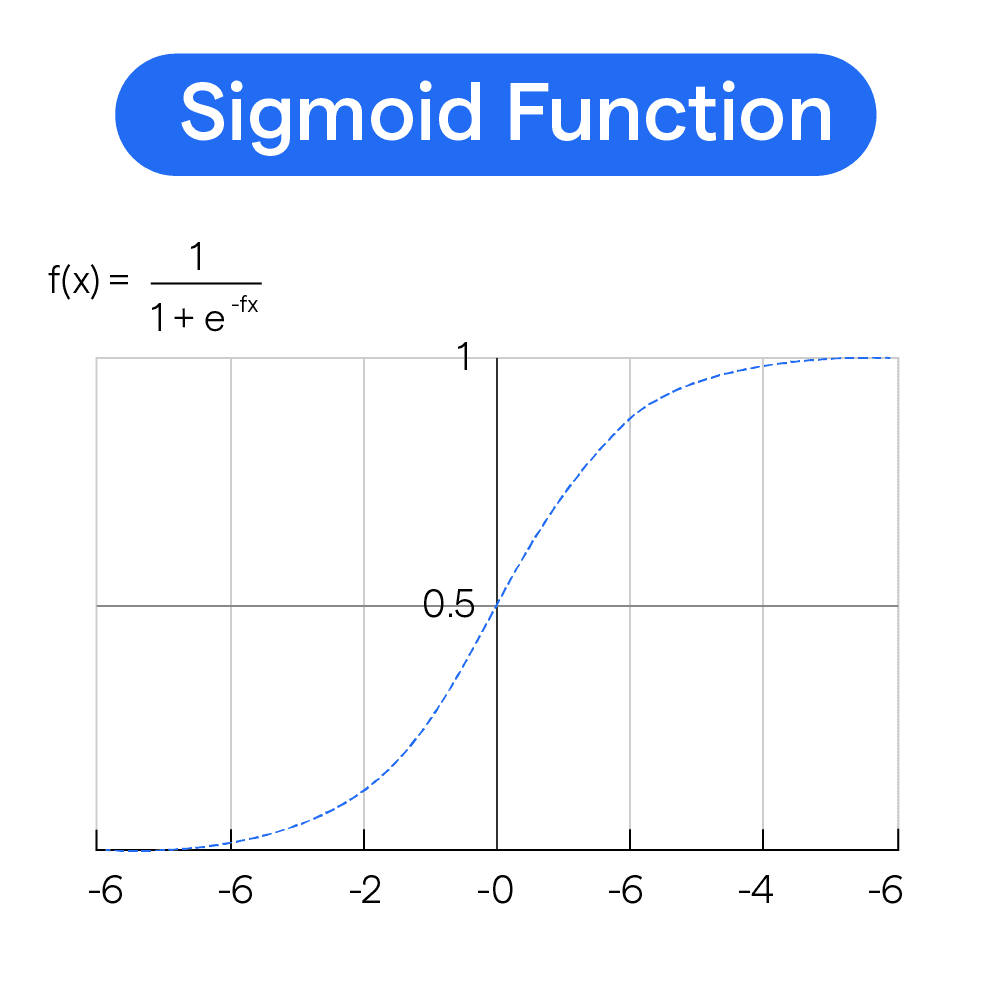
Even named regression it is a classifier used for binary classification and multi-c1ass c1ssification the main formu1as of this are as shown below



Further developed as



The output given as sigmoid curve as shown below:



**Feed - Forward Propagation:**

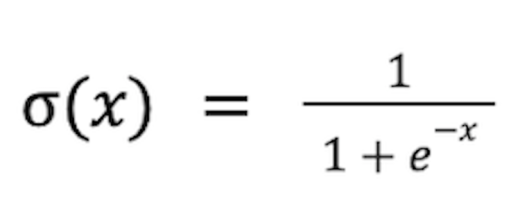
1. Initial Calculation

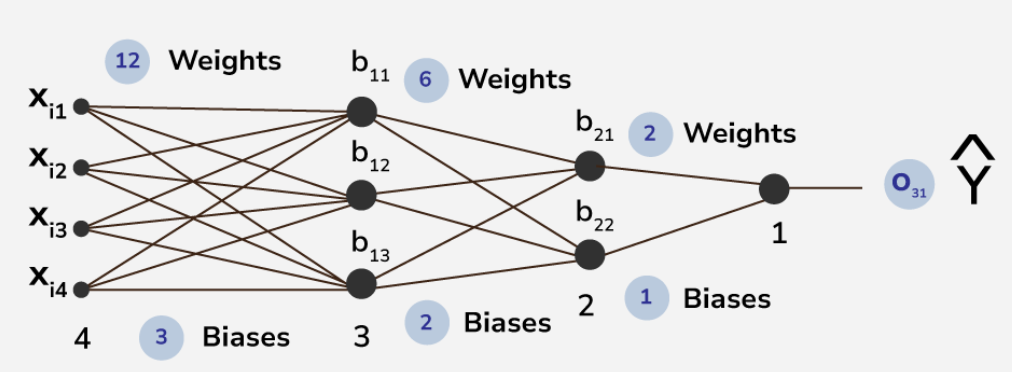
The weighted sum at each node is calculated using:

**aj​=∑(wi​,j∗xi​)**

2. Sigmoid Function

The sigmoid function returns a value between 0 and 1:





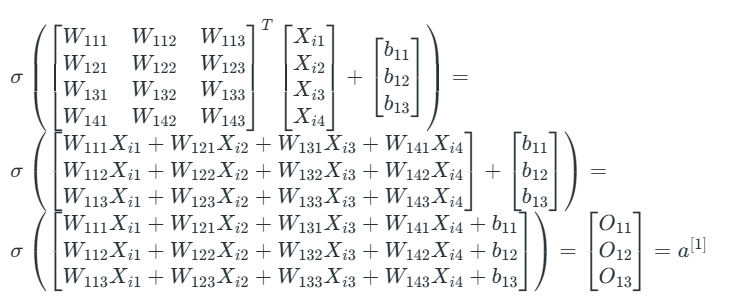
In the above picture, the first layer of Xi1, Xi2, Xi3, Xi4 are the input layer and last layer of b11, b12, *b*13​  is the output layer.

Other layers are hidden layers in this structure of ANN. This is a 4 layered deep ANN where first hidden layer consists of 3 neuron and second layer consists of 2 nodes.

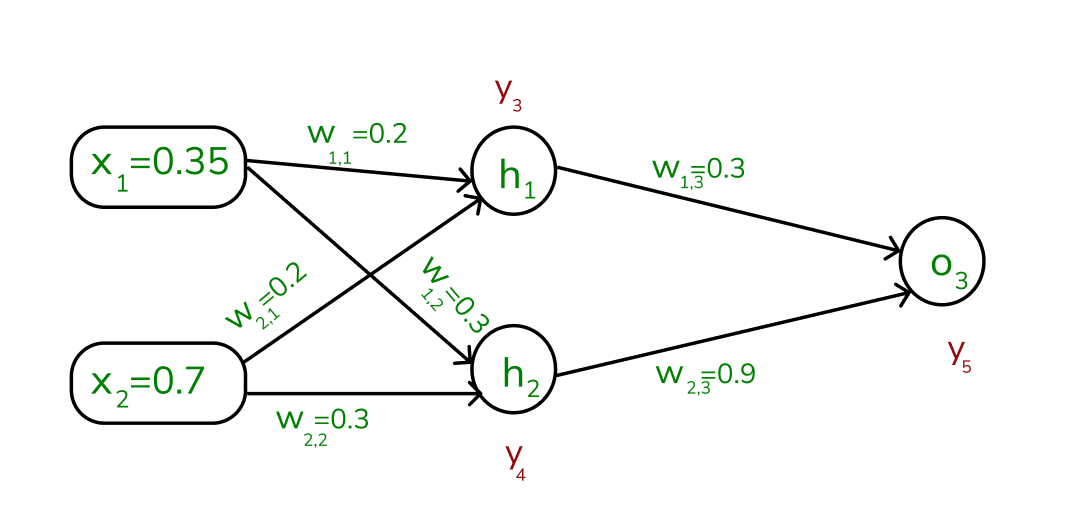
There are total **26** trainable parameters. here , in hidden layer 1 , the top to bottom biases are b11,b12,b13*b*11​,*b*12​,*b*13​ and in hidden layer 2 , the top to bottom biases are b21,b22*b*21​,*b*22​. The output layer contains the neuron having bias b31 . likewise the weights of corresponding connections are assigned like W111,W112,W113,W121,W122*W*111​,*W*112​,*W*113​,*W*121​,*W*122​etc.

### ****Inside Layer 1****

Here, from the previous output function, weights and biases are segmented as matrices and the dot product has been done along with matrix manipulation.



# Backpropagation in Neural Network



#### Error Calculation

Note that, our actual output is 0.5 but we obtained 0.67.

To calculate the error, we can use the below formula:

Error = target − y5

Error = 0.5 − 0.67 = −0.17

Using this error value, we will be backpropagating.

#### ****1. Calculating Gradients****

The change in each weight is calculated as:

Δwij = η × δj × Oj

Where:

* δj is the error term for each unit,
* η is the learning rate.

#### ****2. Output Unit Error****

For O3:

δ5 = y5 ( 1 − y5 ) ( ytarget − y5 )

=0.67(1−0.67)(−0.17)=−0.0376

#### ****3. Hidden Unit Error****

For h1:

δ3=y3(1−y3)(w1,3×δ5)

=0.56(1−0.56)(0.3×−0.0376)=−0.0027

For h2:

δ4=y4(1−y4)(w2,3×δ5)

=0.59(1−0.59)(0.9×−0.0376)=−0.0819

#### 4. Weight Updates

For the weights from hidden to output layer:

Δw2,3=1×(−0.0376)×0.59=−0.022184

New weight:

w2,3(new)=−0.22184+0.9=0.67816

For weights from input to hidden layer:

Δw1,1=1×(−0.0027)×0.35=0.000945

New weight:

w1,1(new)=0.000945+0.2=0.200945

Similarly, other weights are updated:

* w1,2(new)=0.27133
* w1,3(new)=0.08567
* w2,1(new)=0.29811
* w2,2(new) =0.24267

