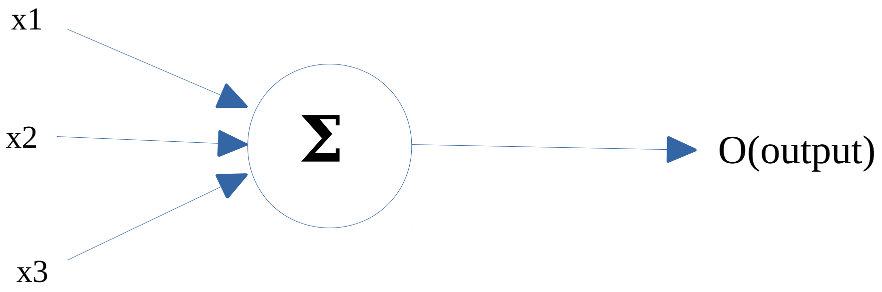
Understanding of Weight Matrix

In order to understand the ‘**weight**’ or ‘**weight matrix**’, first of all, we will have to understand the single neuron or a **perceptron**.



So for computing the output **O**, we apply a **mathematical operation** to the inputs like-

O = (x1 * x2) + x3

There are various way to compute the Output but to illustrate the weight i used this type of output.

So in the above expression, there is a problem that if x1 or x2 will be zero then we will loose one information or we can say that particular information or input is not used.

Assume x1 = 0 and x2 = 1
then   x1 * x2 = 0, so here x2 is not required or x2 is lost.

Therefor, to overcome this type of problem, we apply weight or weight matrix.

Let us take an example,

Suppose you want to make a perceptron for predicting weather it will rain today or not. Let us assume there are a binary output **O** and two binary inputs **x1** and **x2**.

Case1: Let x1 be 1 if the weather is cloudy today, 0 if not.
Case2: Let x2 be 1 if you are wearing a red shirt today and 0 if not.

So we can see here wearing the red shirt has almost no relation with the possibily of raining. So,

O = x1 + 0*x2

so here we can see that x2 will be lost

So the better solution is to give the weightage of the inputs.So

O = x1 + 0.1 * x2

0.1 is the **weight** for the x2 and 1 is the **weight** for the x1 here

So, if x2 = 0, x2*0.1 = 0 and
if x2 = 1, x2 * 0.1 = 0.1 not 1

So, it basically gives the importance of inputs for the peridicting output and now we have no problem.

So, here the weight matrix **W** can be

W = [1, 0.1]