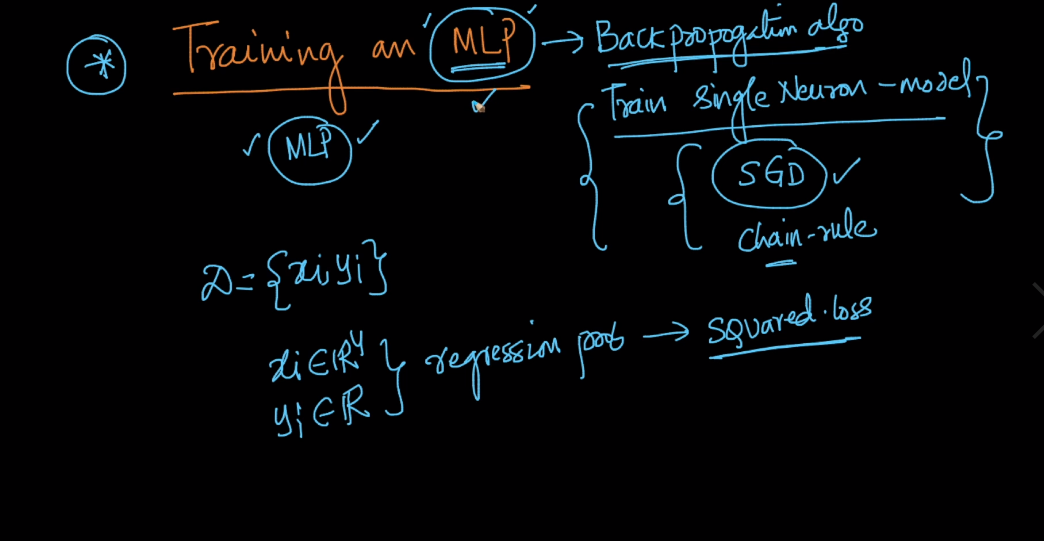
**Training an MLP: Chain Rule**

For training a multi layer perceptron we take a dataset xi, yi where xi is real 4-dim and yi is real 1-dim.

Now here we solve it for regression problem and for regression problem loss is squared loss.



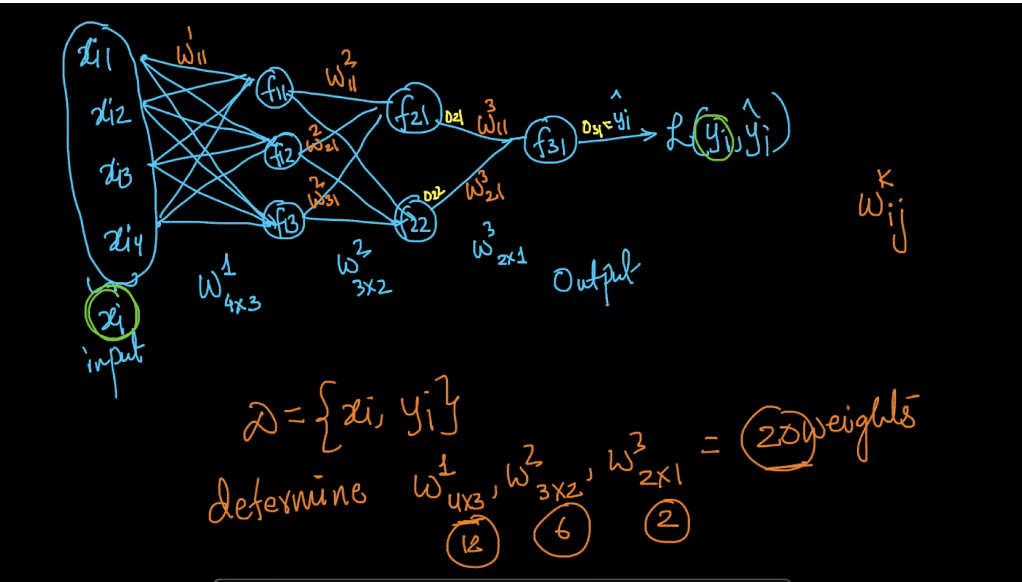
Now on training part our task is to determine weight matrices which minimizes loss.

As we see in below image Multi layer neural network and three weight matices .

1. W14\*3 weight matrices for input layer to first layer, its dimension is 4\*3 because vertices are from 4 input neurons to 3 first layer neurons i.e 4\*3 vertices.
2. W23\*2 weight matrices for first layer to second layer
3. W32\*1 weight matrices for second layer to output layer.

Therefore there are total 20 weights we have to determine.

And output i.e y\_hat from output layer goes to Loss function.



Now steps for training

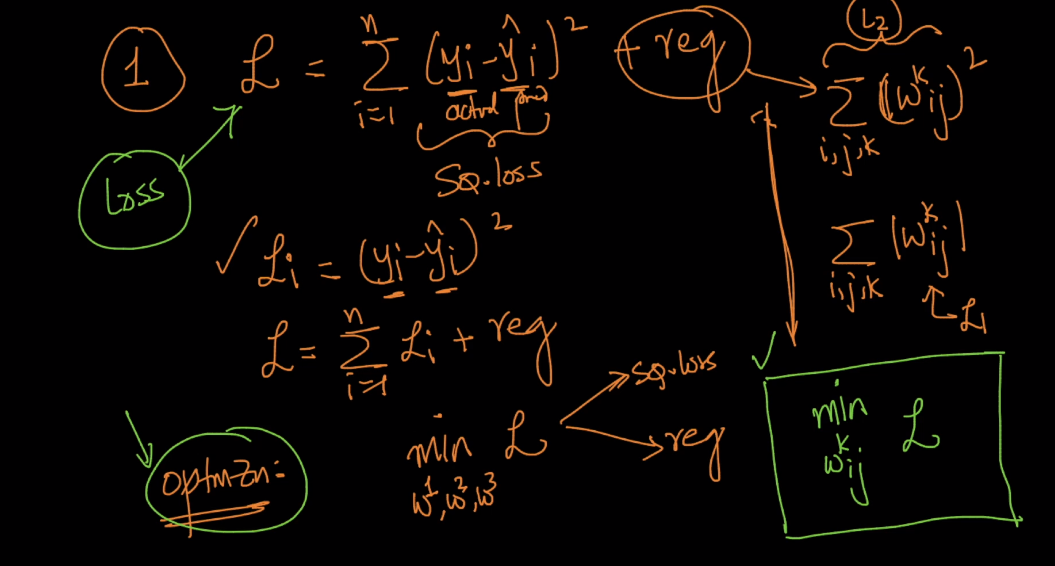
1. Defining loss-function :

In loss function there are two parts first is squared loss and second is regularization,

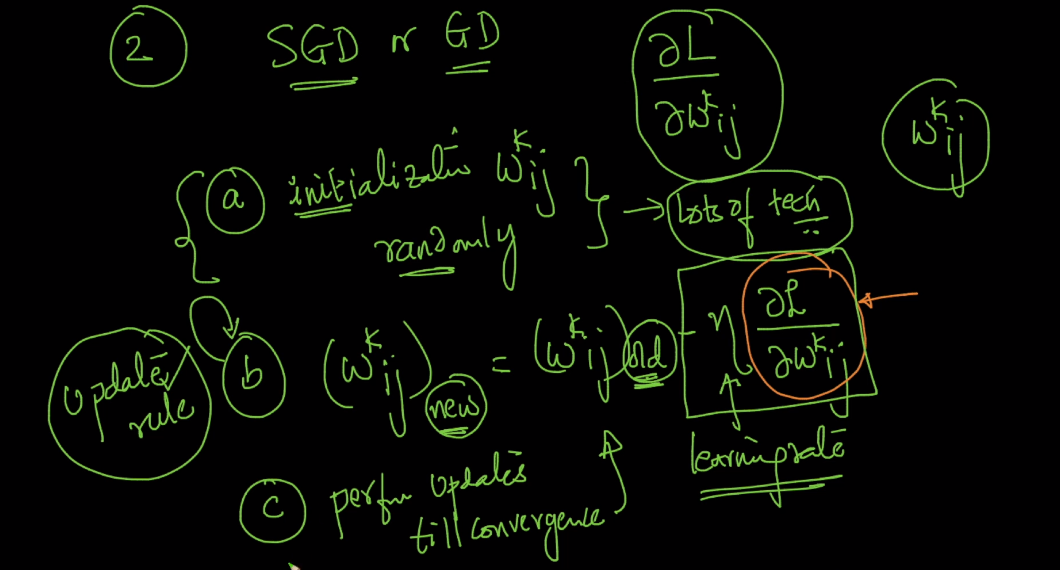
And in regularization we have to take all values of I,j,k which is shown below.

1. Optimizations :

In this we have to define optimization in which we select w1,w2,w3 such that it minimizes loss



1. Solving optimization problem by SGD or GD:
   1. Initialize wki,j  and eta(learning rate) randomly
   2. Found new w by using old w and differentiation of loss by w and eta as shown below
   3. Perform updates till convergence (i.e w\_old and w\_new is very very close.)



Now main task is to differentiation of loss by w which is shown below.

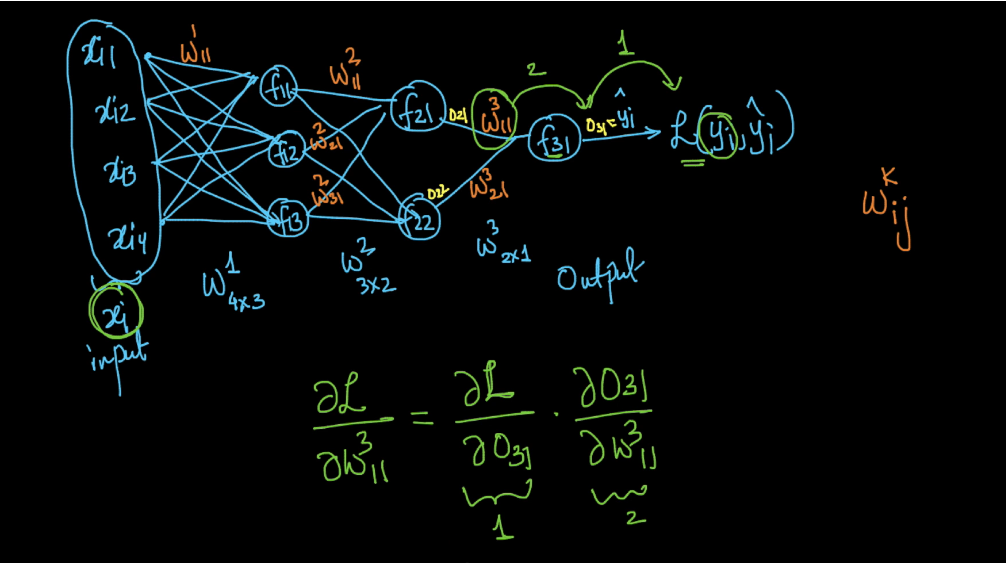
So first we do W3 weights :

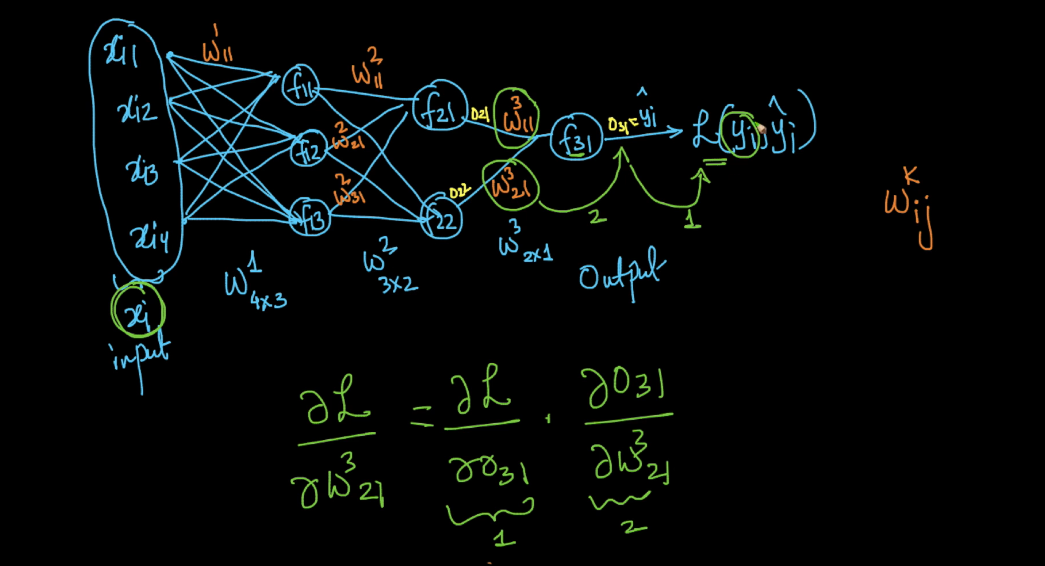
1. Del\_l/del\_ W311 :

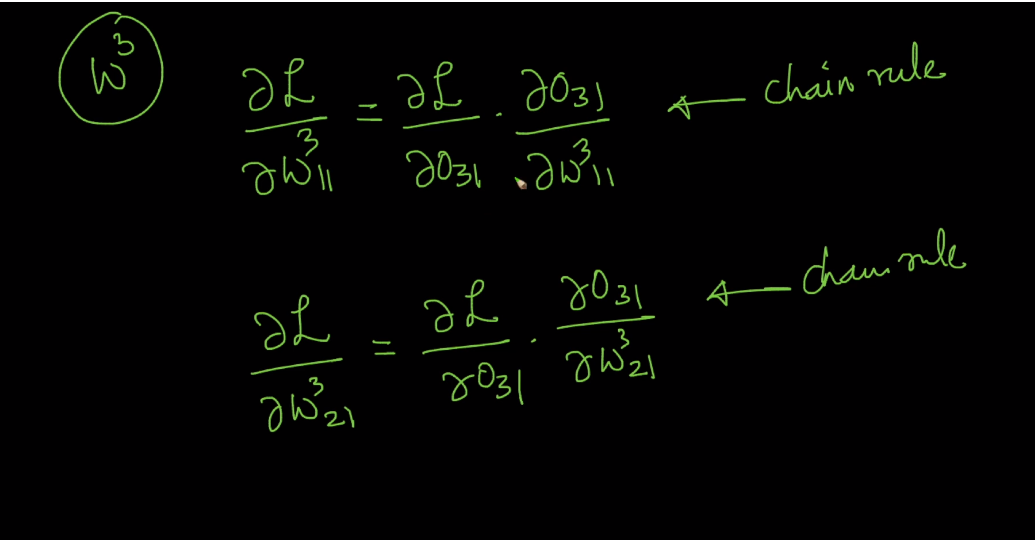
Now for this what happens is W311 ­goes into f31 and impacts its output i.e O31 which is nothing but yi\_hat which goes into loss function and impacts loss function.

Diff. is shown in below pic.

Similarly we do for W3 weights







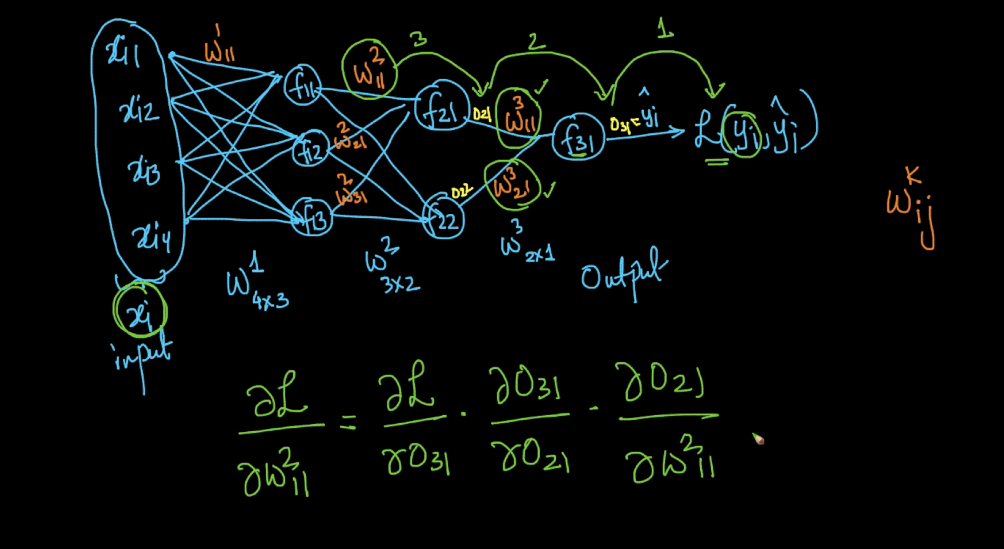
Now we do for W2 weights :

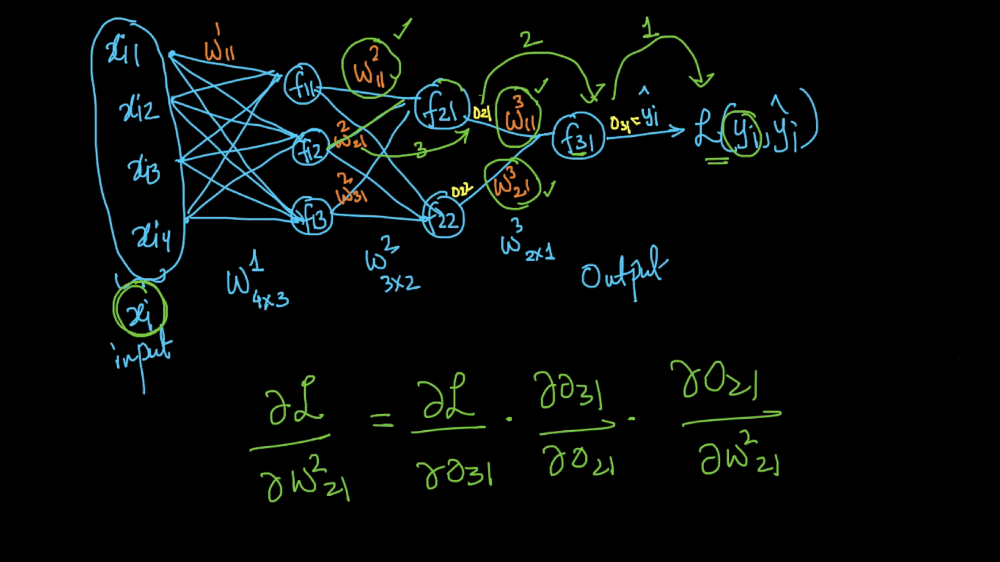
1. Del\_l/del\_ W211 :

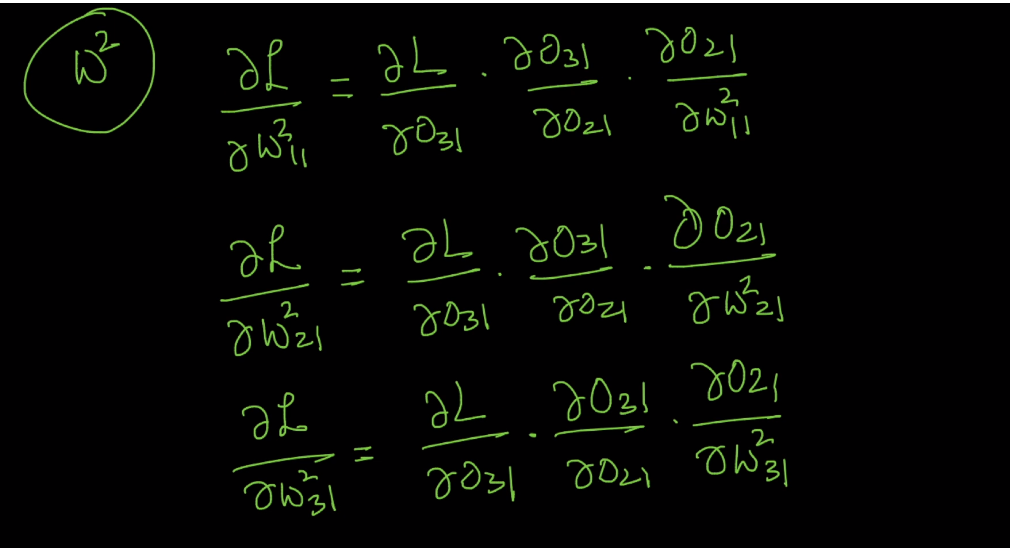
In this W211  goes into f21 and impacts O21 , this O21 goes into f31 and impacts O31 which is nothing but yi\_hat which goes into loss function and impacts loss function.

Diff. is shown below

Similarly we do for all w­2 weights.

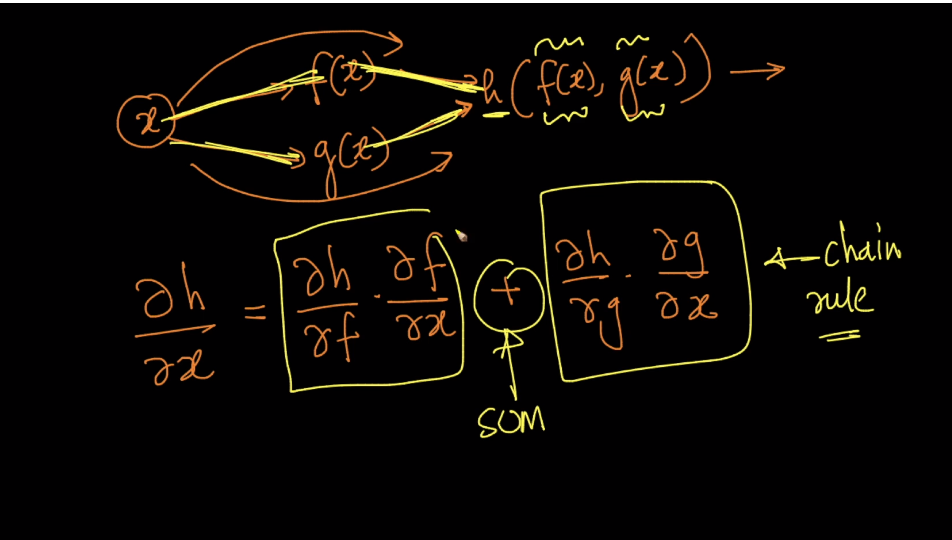






Now for W1 weights : this is tricky one therefore it require special chain rule which is shown below.

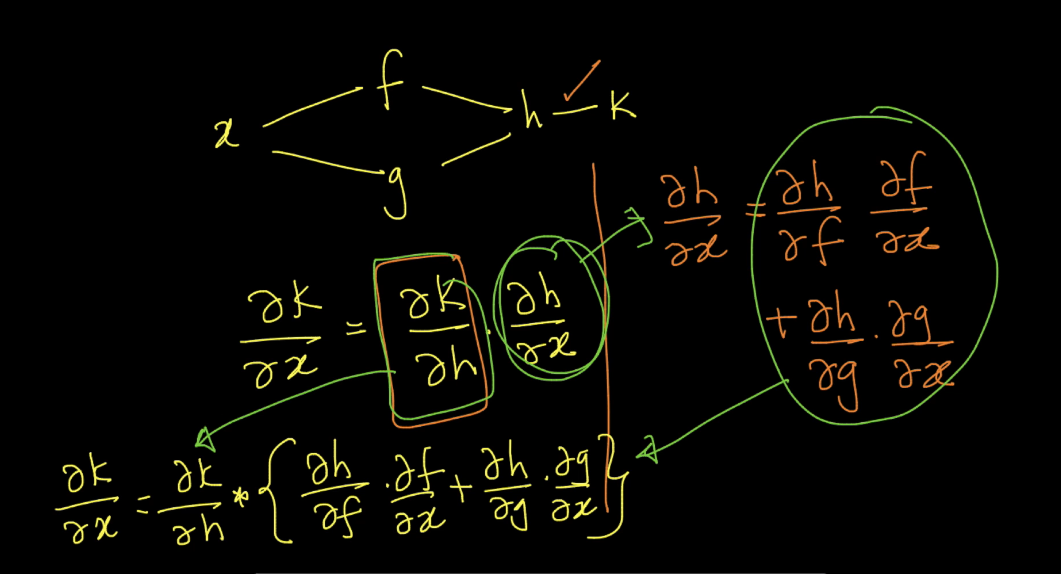
Here x is going into f(x) and g(x) and their output goes into h therefore now h is a function of f(x) and g(x) therefore in this we need to take the derivative of both the path and sum them up as shown below.



Now this is similar to above only diff. is output from f is going into k therefore here k (diff. of k) is same for both paths therefore its diff is diff. which is shown below.

Here del\_k/del\_h is same for both path therefore it remains common and as it is but

Del\_h/del\_x is diff for both path therefore we took derivative of both path and sum it



Now in below it shows derivative of W1 weights :

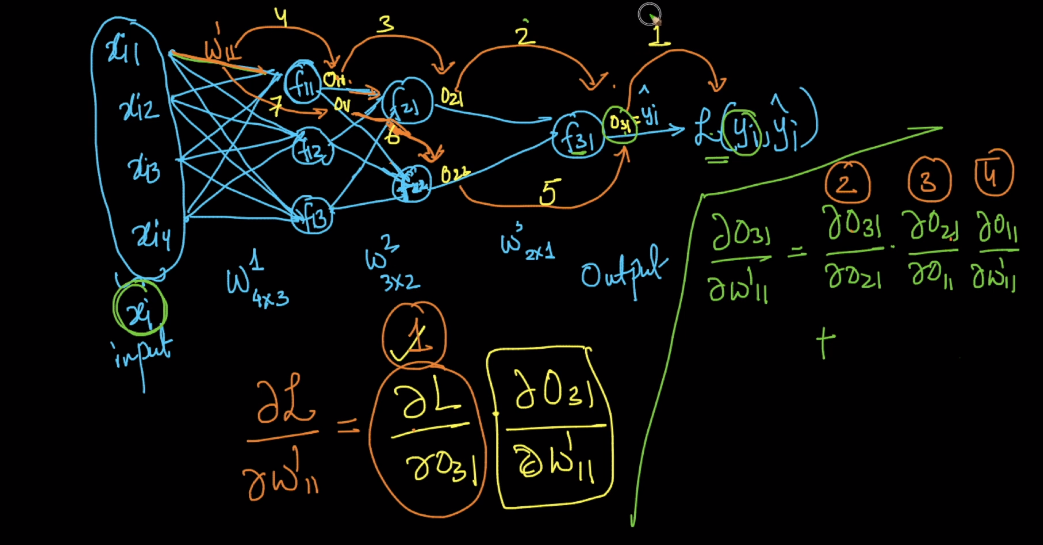
1. Del\_l/del\_w111 :

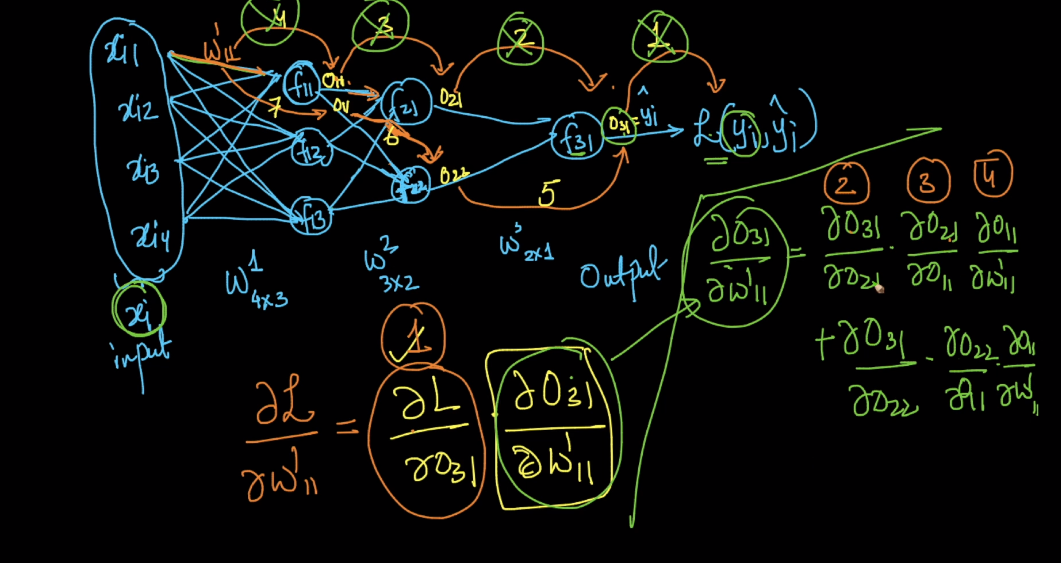
Here w111 ­impacts O11 but O11 instead of going only towards f21 it is also going towards f22.

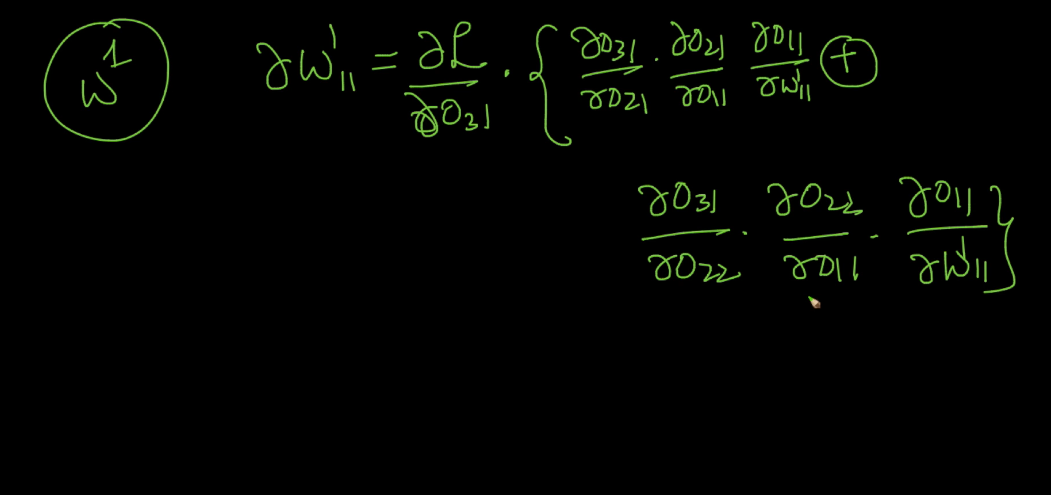
O11 which is going towards f21 impacts O21, O21 impacts O31, and O31 to loss

O11 which is going towards f22 impacts O22, O22 impacts O31, and O31 to loss

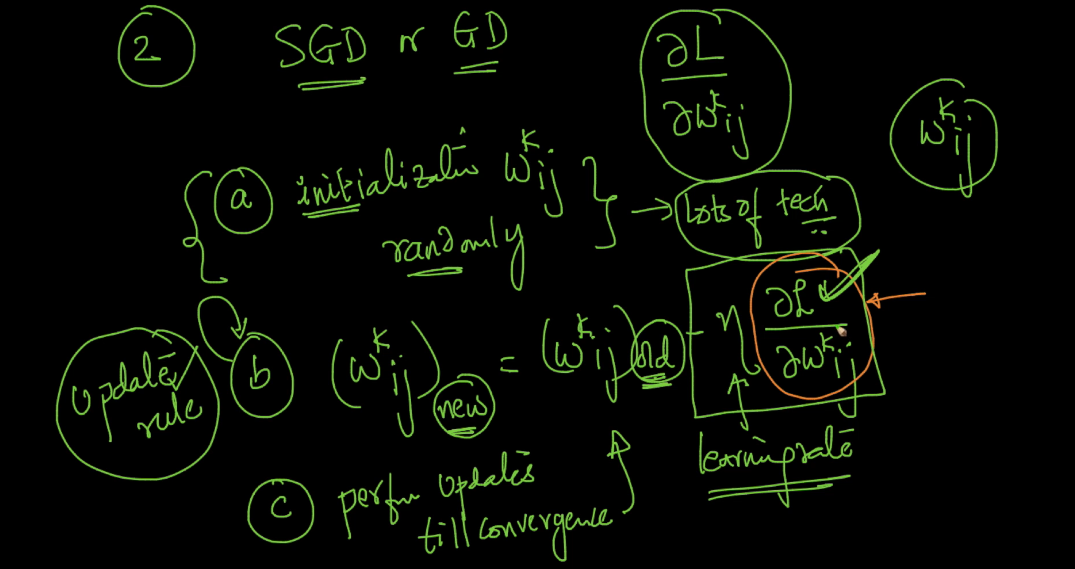
So here we have 2 paths therefore we do same as we seen above.







So in this way we calculate derivatives and therefore by this we can get new\_w



Comments :

