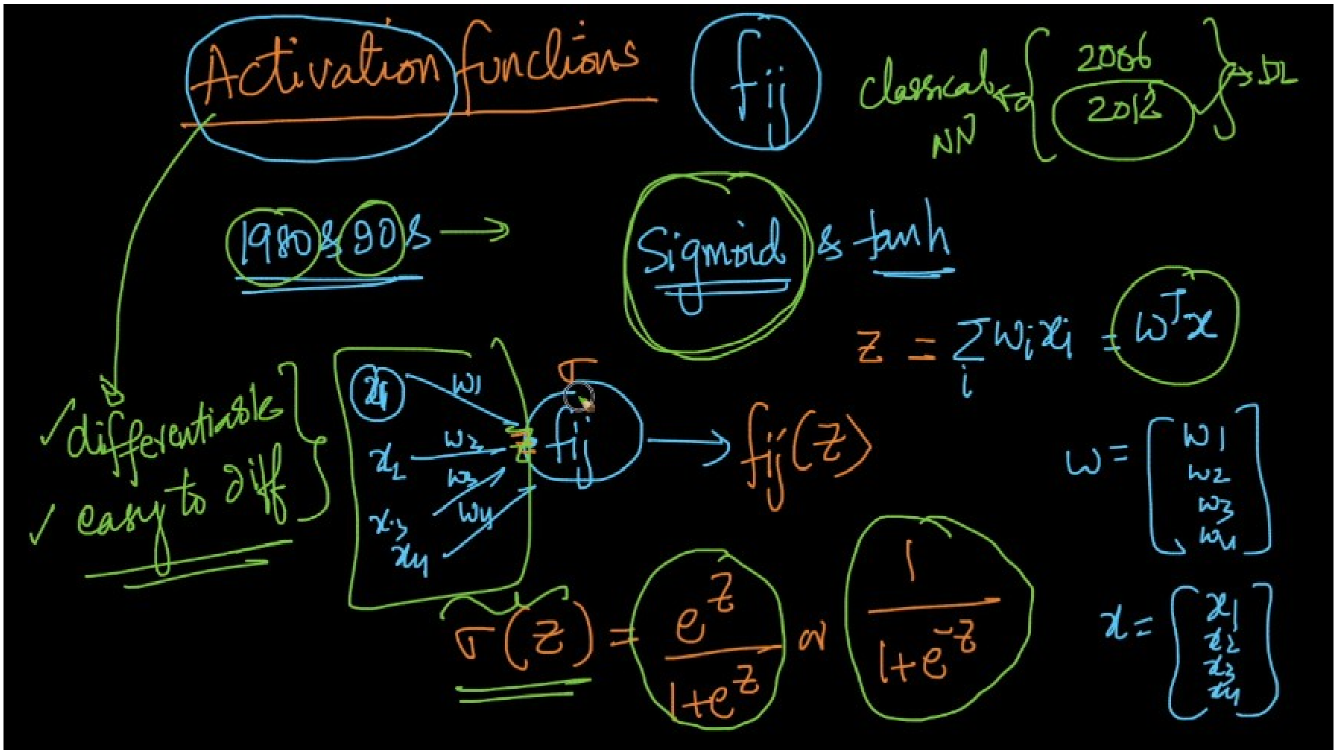
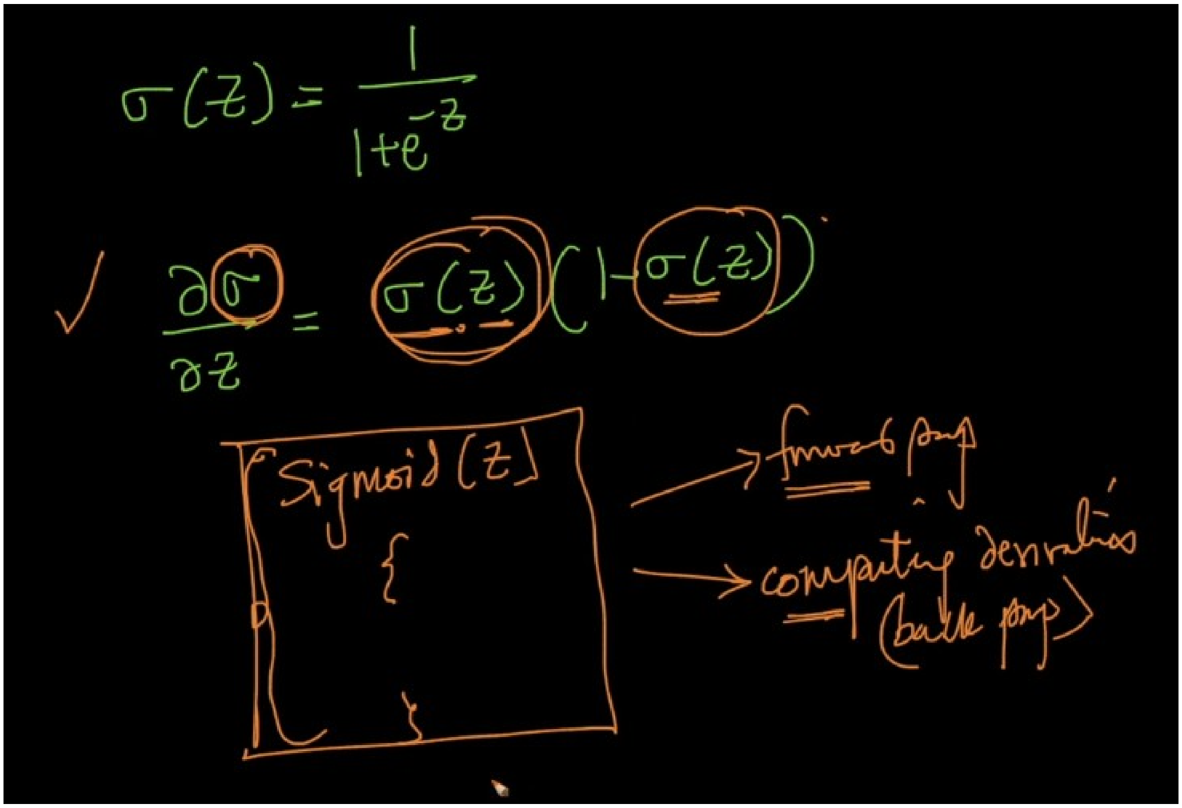
Deep Learning

Activation Functions:

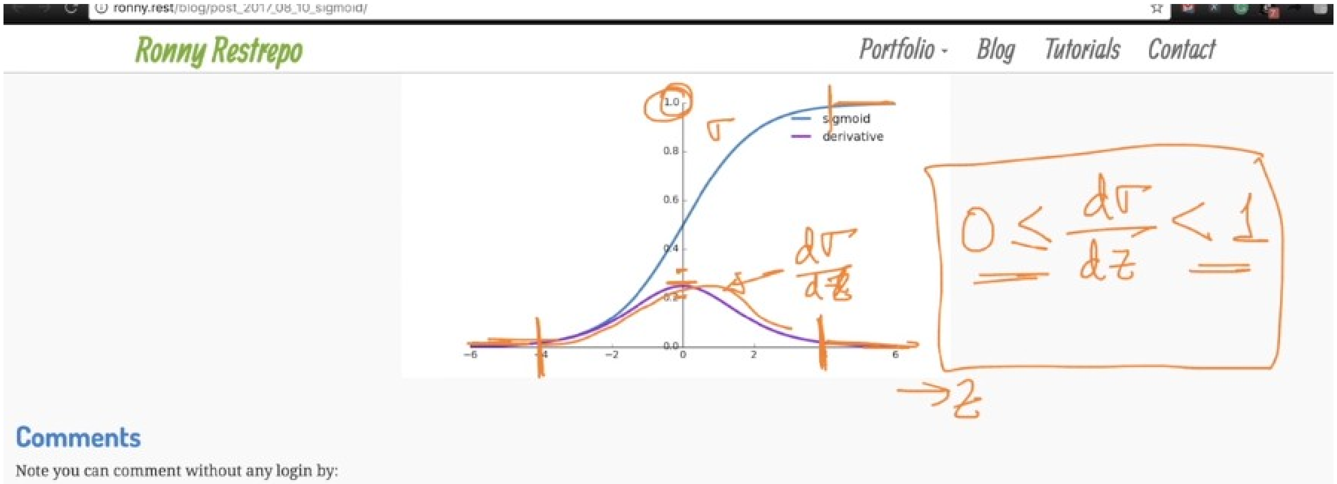


Sigmoid function derivative:

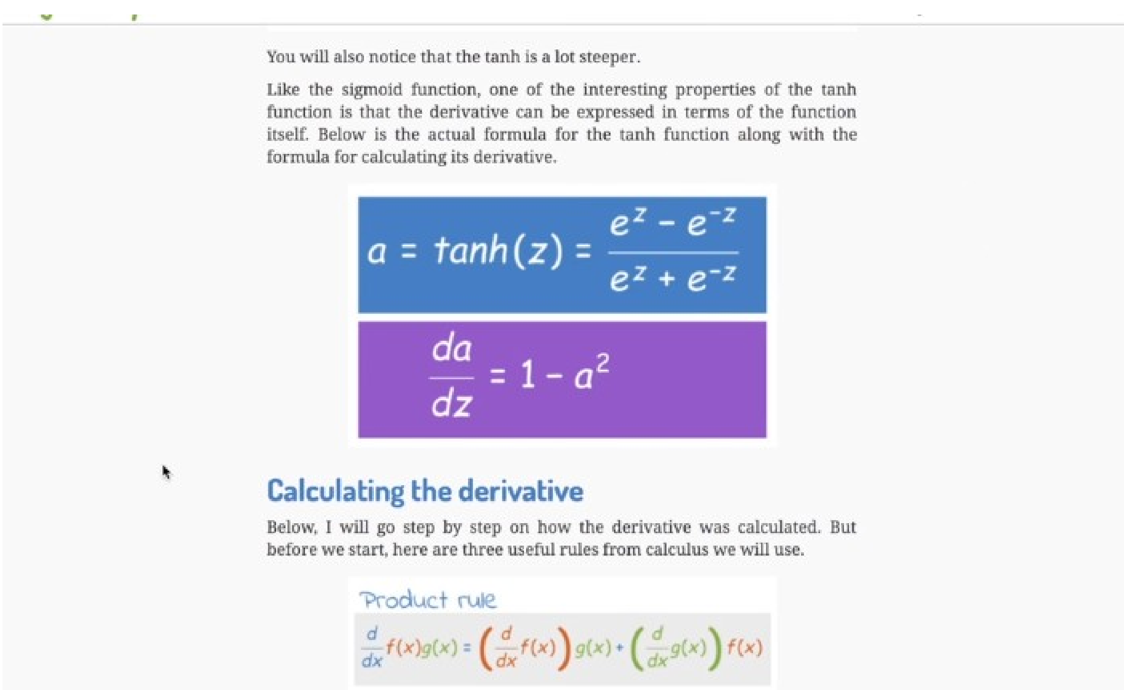


Sigmoid min. value is 0 and max. value if 1. This is sigmoid function.

Derivative of Sigmoid visualization:



Tanh Function:



You will also notice that the tanh is a lot steeper.

Like the sigmoid function, one of the interesting properties of the tanh function is that the derivative can be expressed in terms of the function itself. Below is the actual formula for the tanh function along with the formula for calculating its derivative.

$$a = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$\frac{da}{dz} = 1 - a^2$$

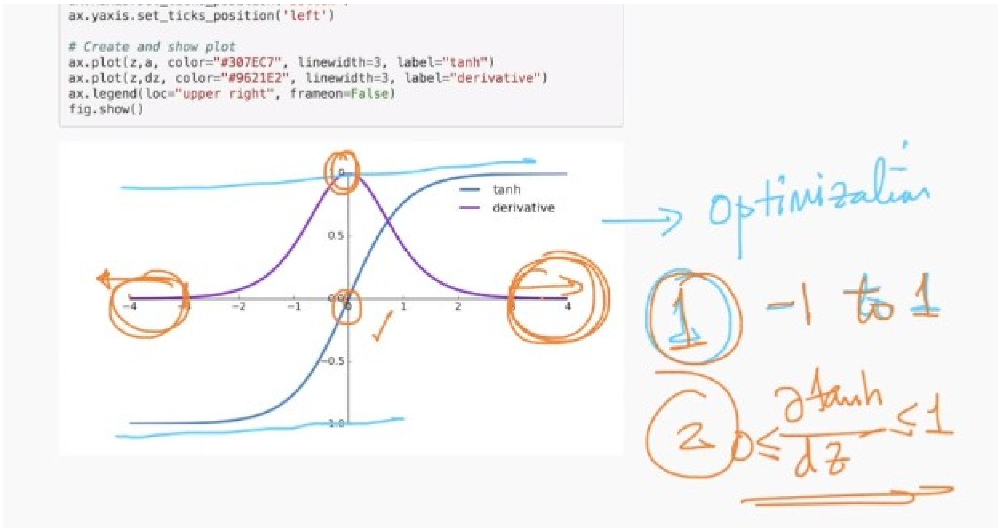
Calculating the derivative

Below, I will go step by step on how the derivative was calculated. But before we start, here are three useful rules from calculus we will use.

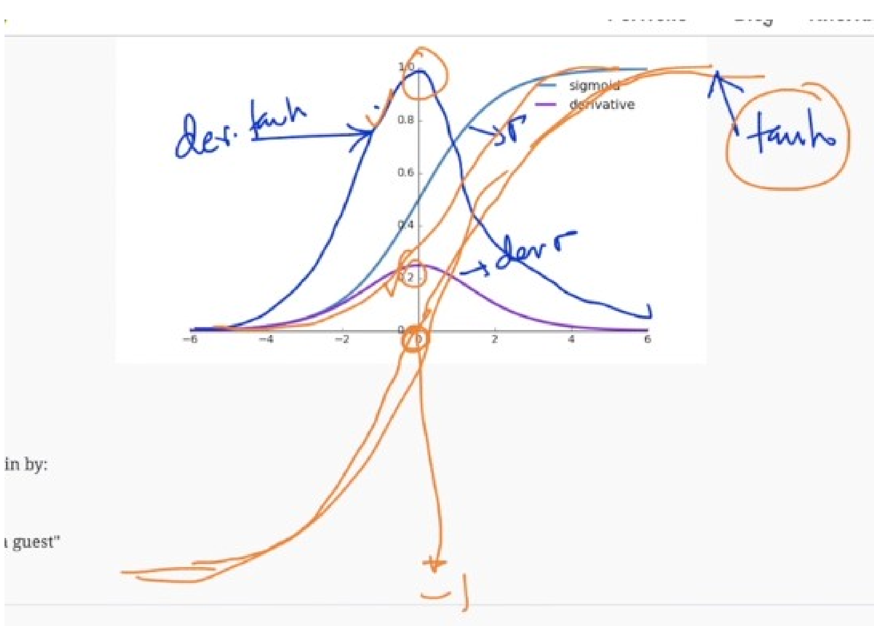
Product rule

$$\frac{d}{dx}f(x)g(x) = \left(\frac{d}{dx}f(x)\right)g(x) + \left(\frac{d}{dx}g(x)\right)f(x)$$

Tanh function and its derivative:

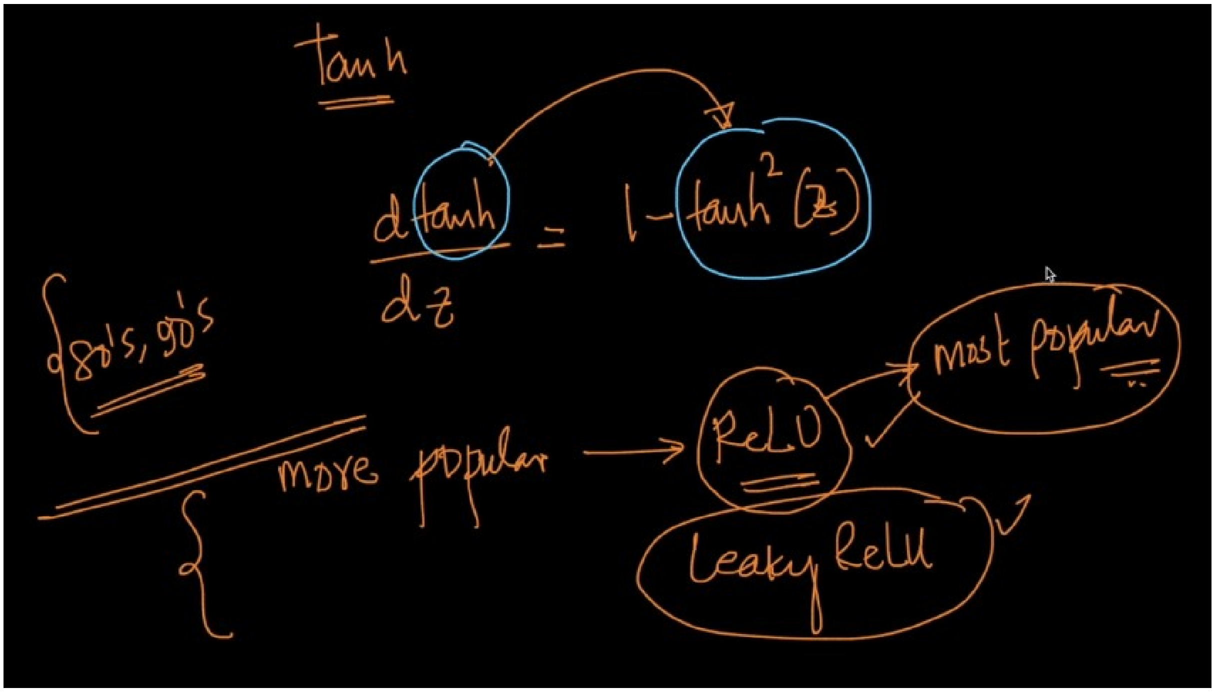


Comparison of sigmoid and tanh functions:

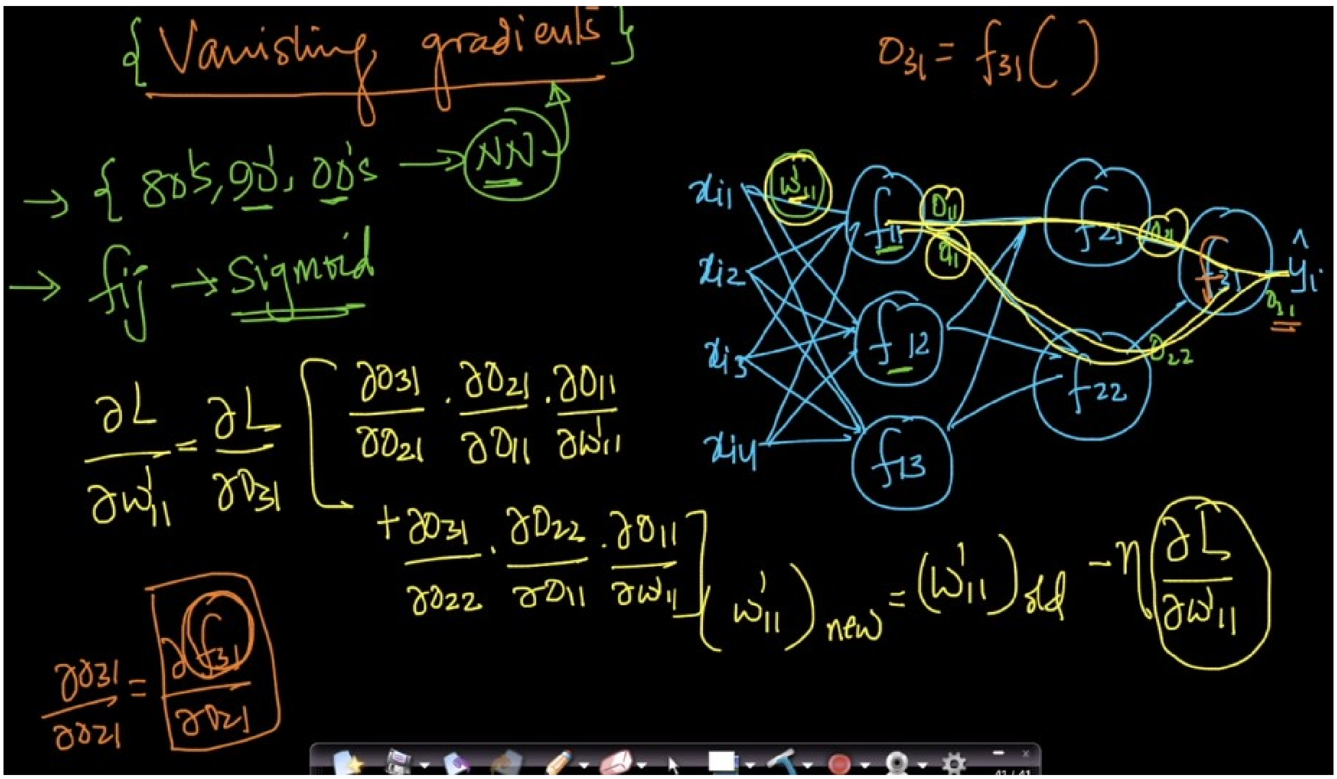


Tanh and sigmoid are the two functions that are more widely used in 80's and 90's.

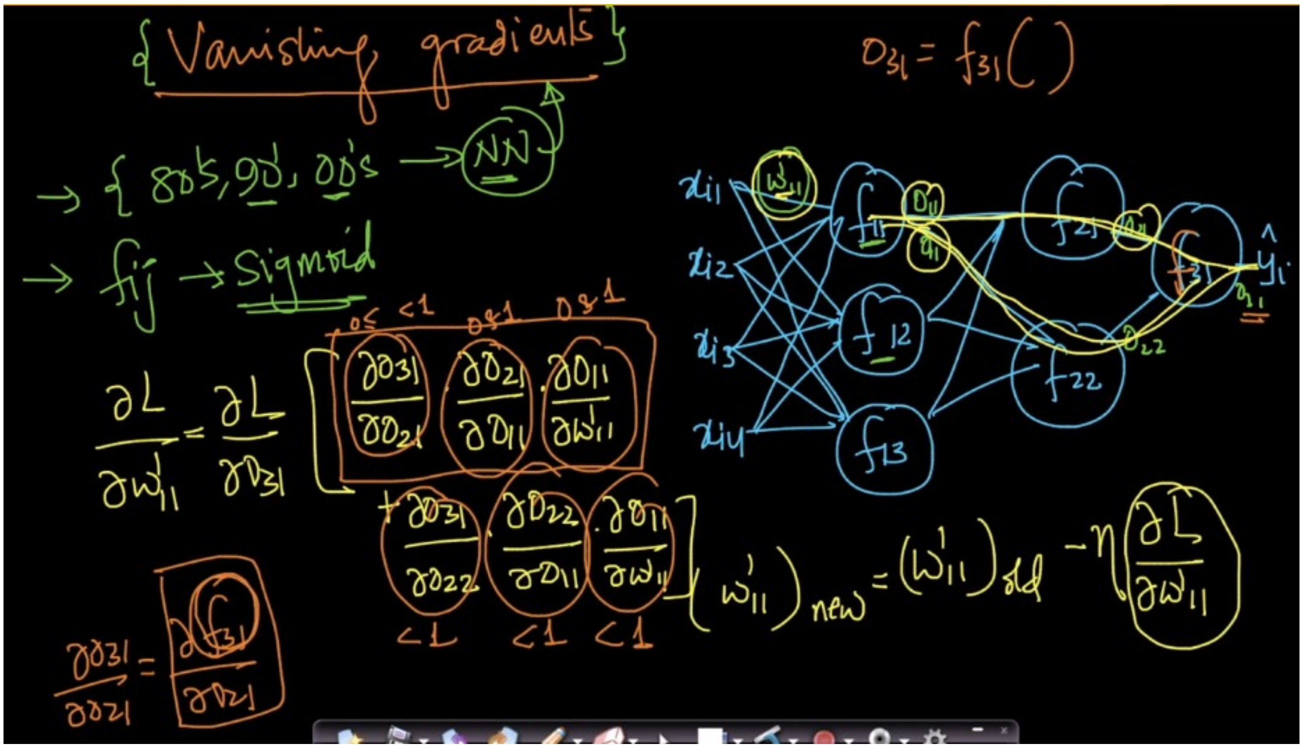
But not Relus are more widely used now.



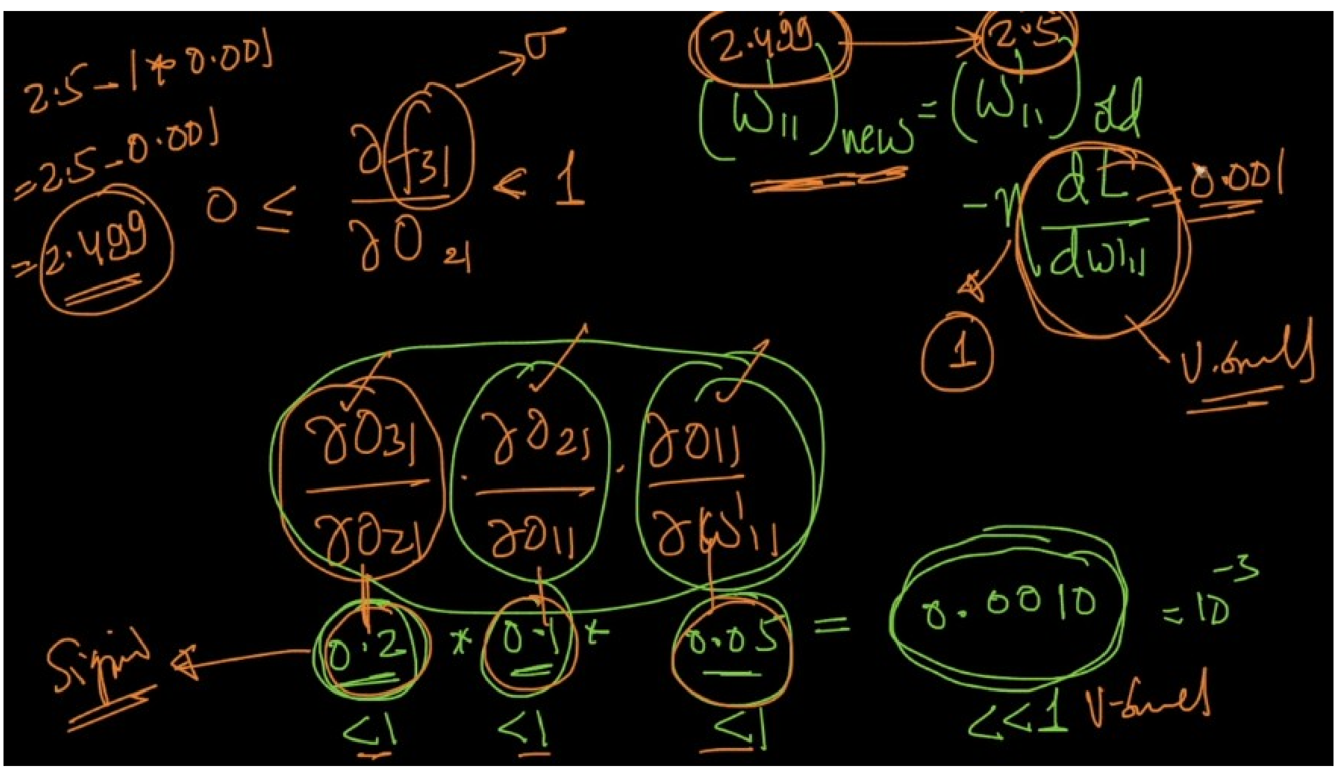
Vanishing gradients:



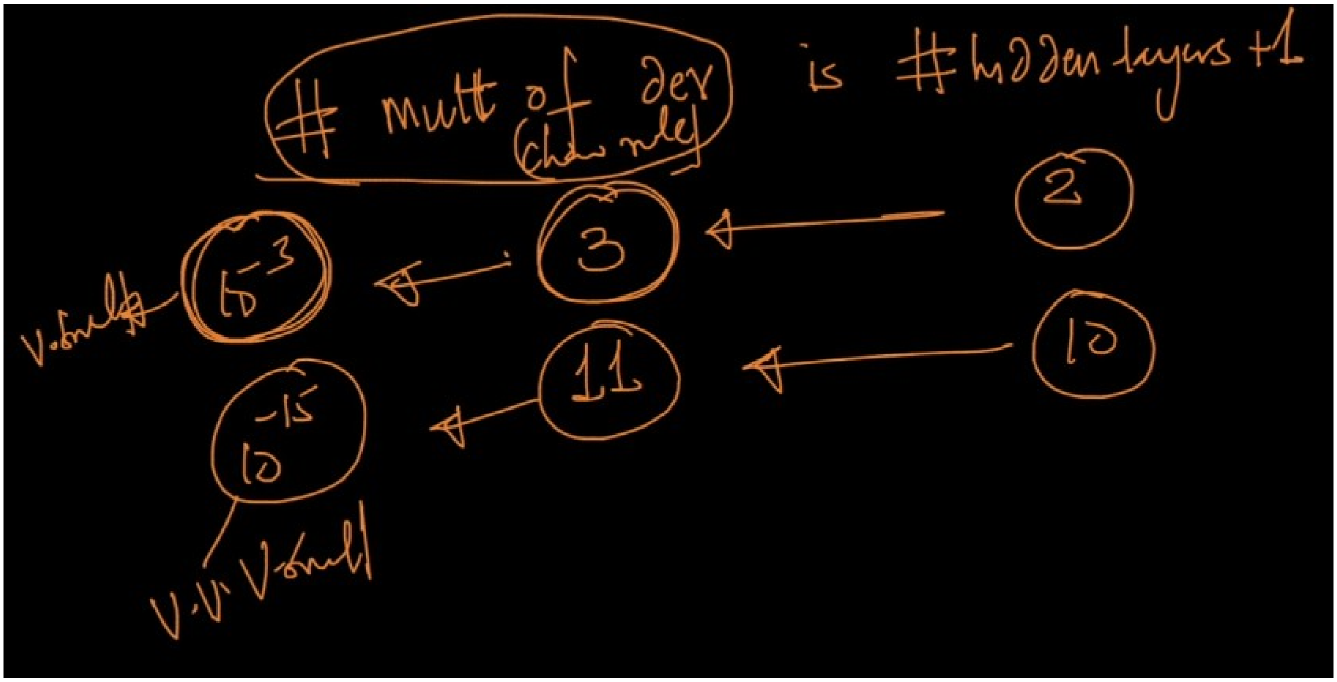
The function is sigmoid:



The multiplication of the derivatives gives very small value. Multiplying very small values gives very small value and the new derivative that is computed does not change much. This is very close to the older value.

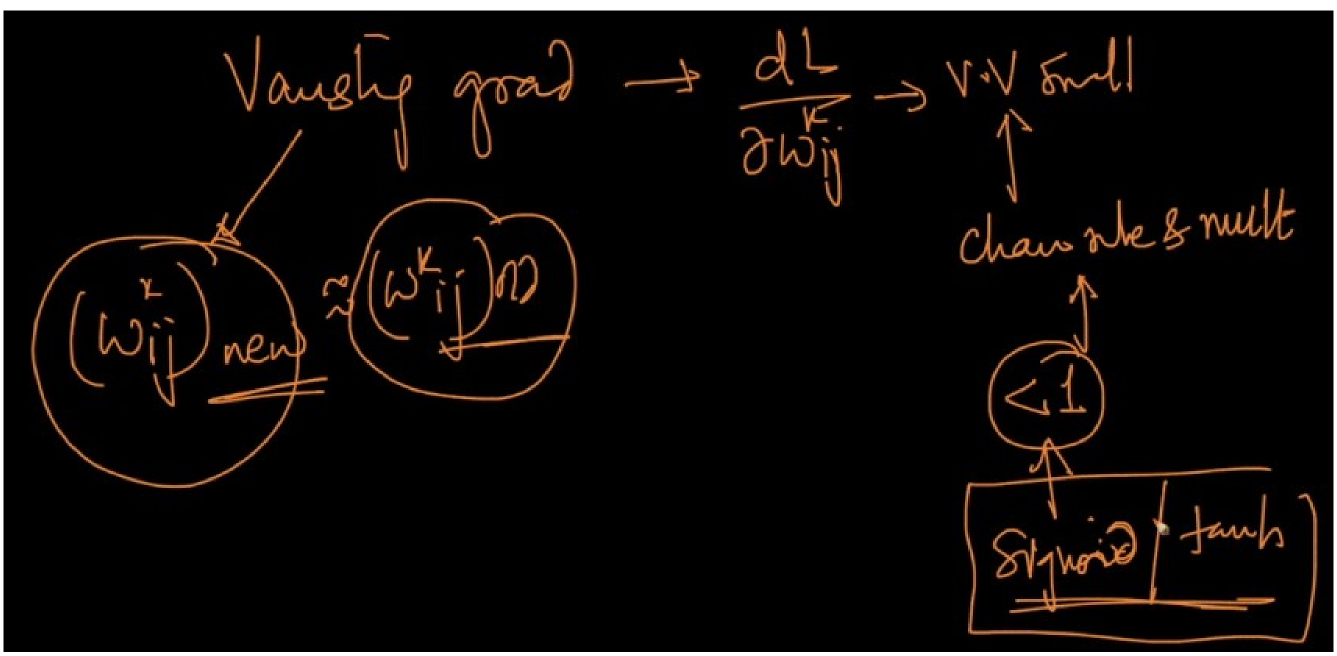


As the number of hidden layers increase then the updated values also becomes small and the new weight value will be more closer to the older value.

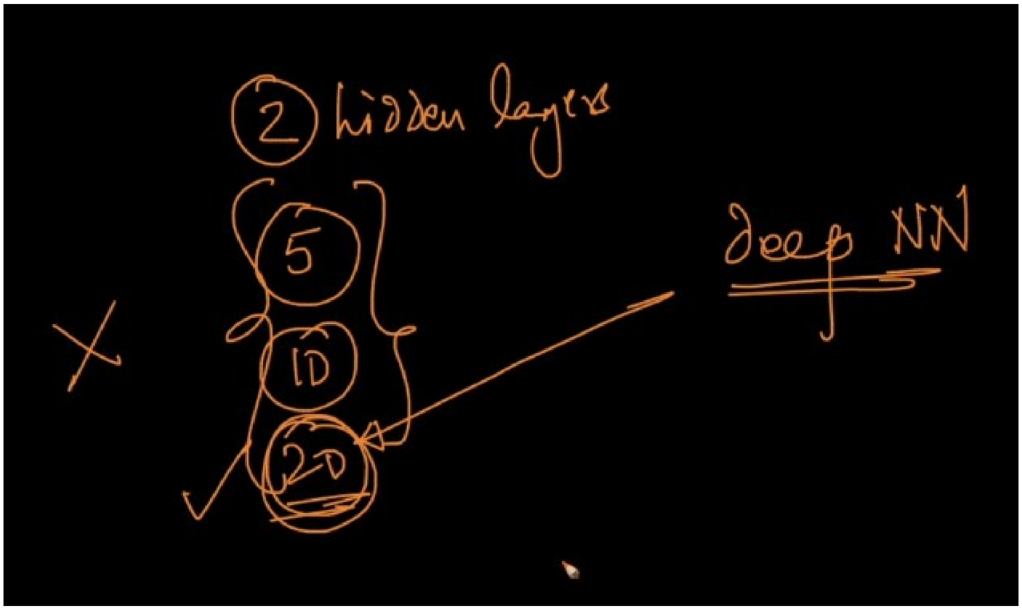


If the derivative becomes smaller and smaller there will be no difference between the new and old values. This is because of the chain rule multiplication.

This is because of the sigmoid (or) Tanh function. This is one of reasons.

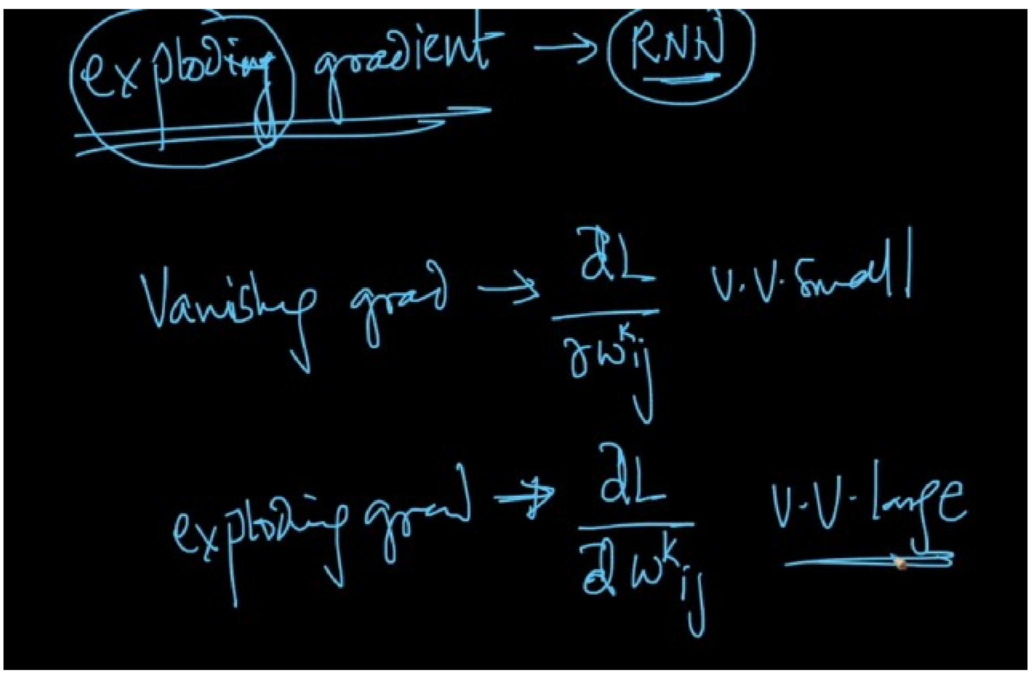


People could train 2 layers in olden days.

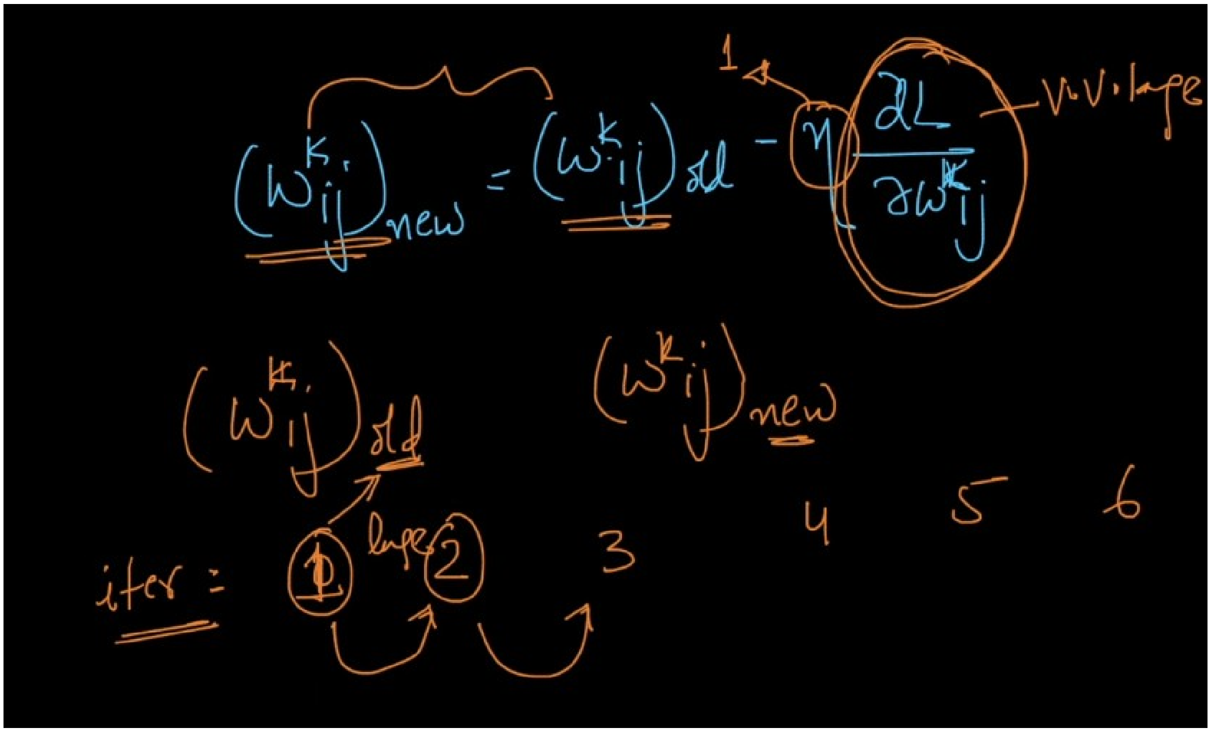


This is why ppl discovered Relu activation function.

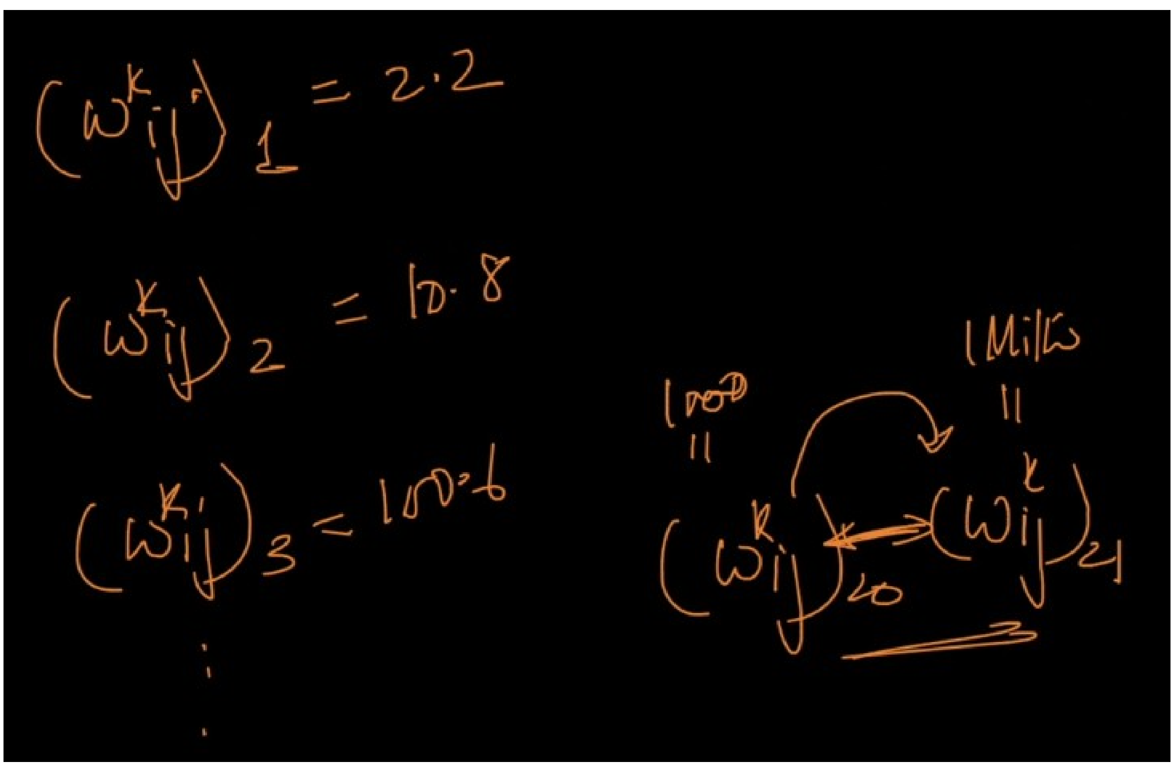
Exploding gradient problem:



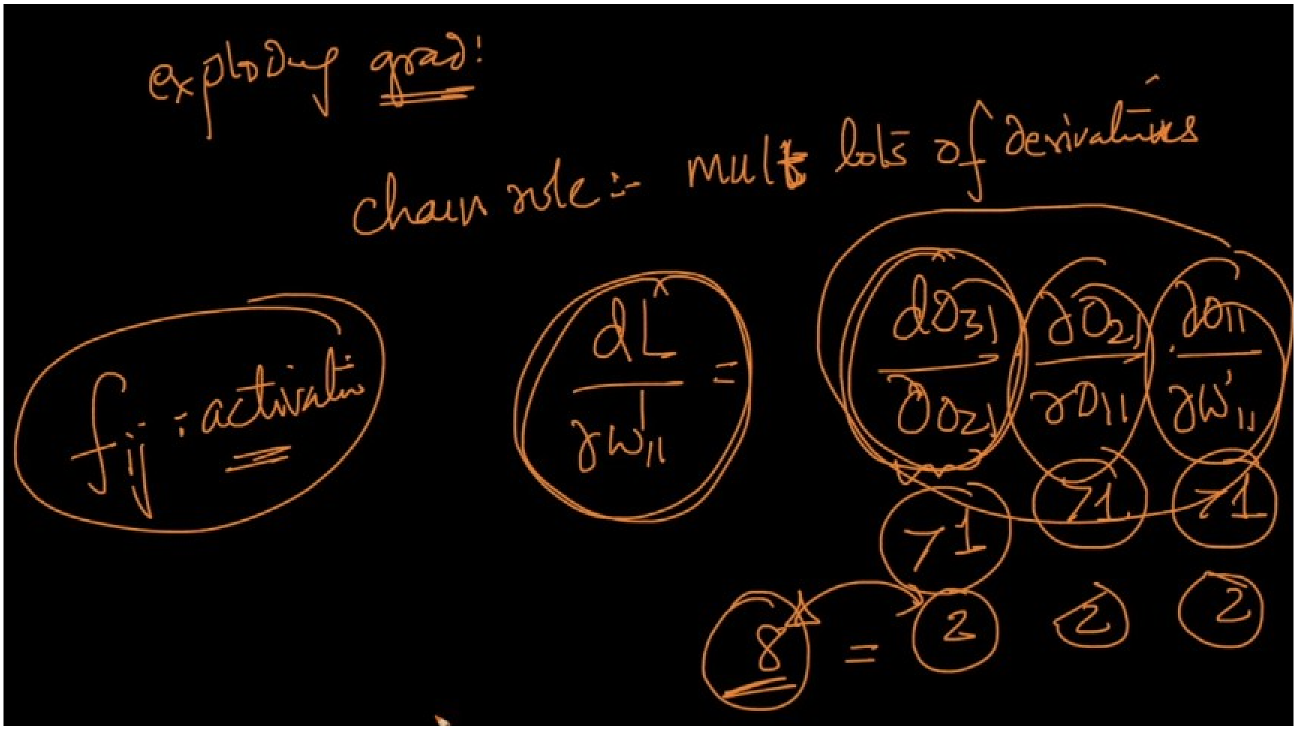
Because if the derivative of the weights are very large then the new value will change a lot. With number of iterations.



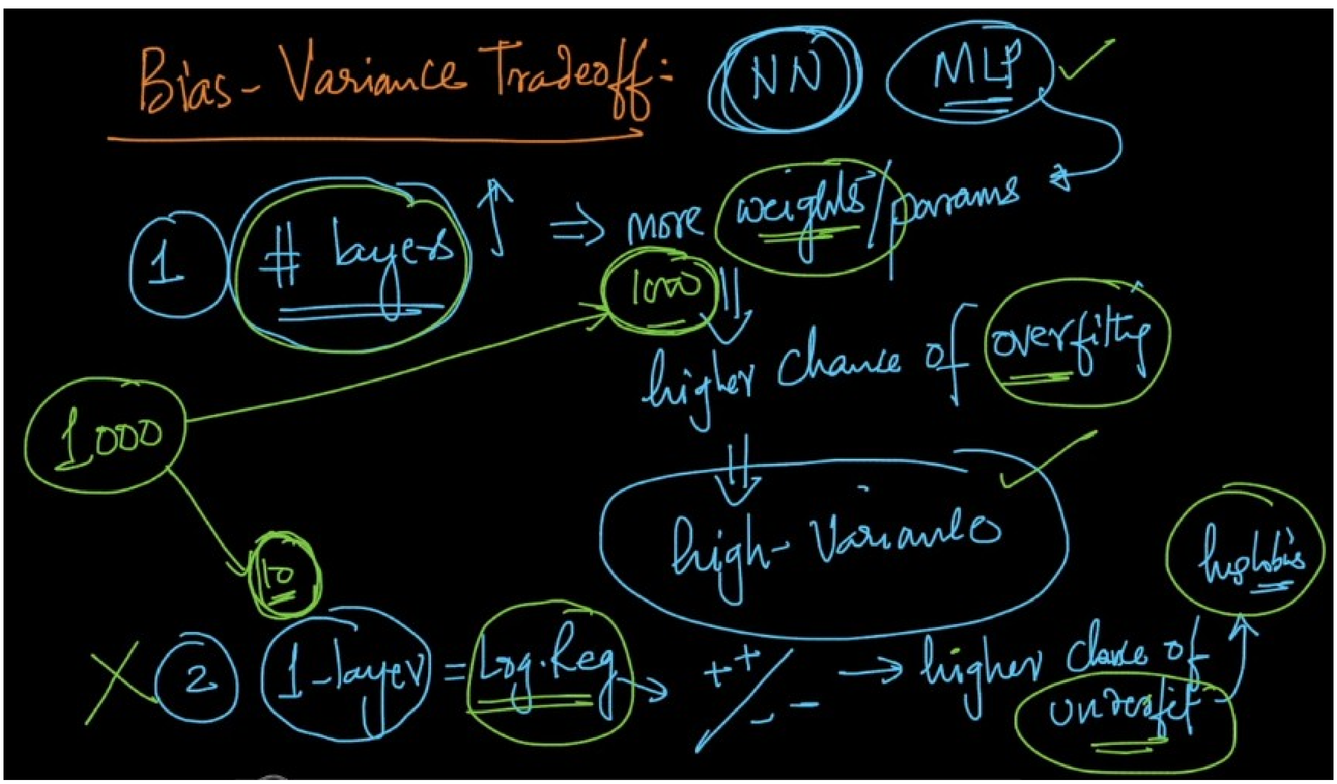
Even after 20 iterations because its derivative function is large the new weight values will change a lot we cannot decide where we can stop. There is no control on the derivative.



Why does exploding gradient occur ?
It occurs because of chain rule, we are multiplying lots of derivatives.

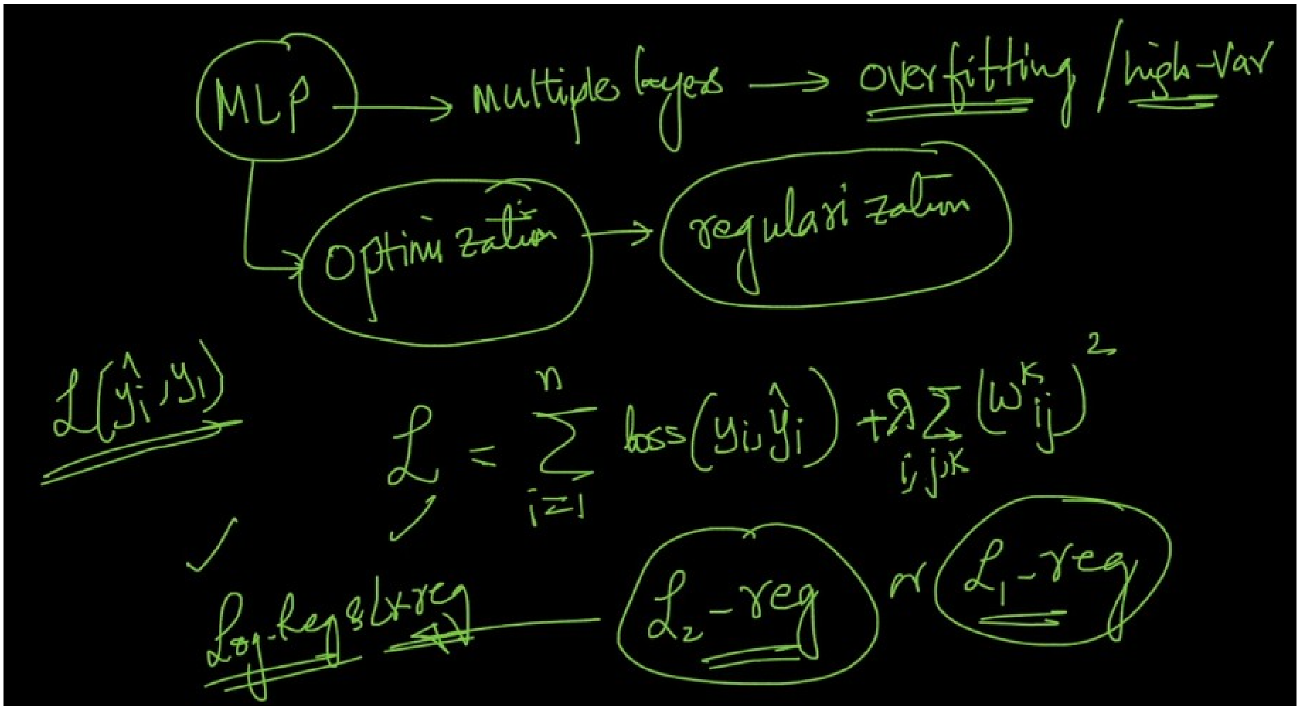


Bias – Variance trade off:

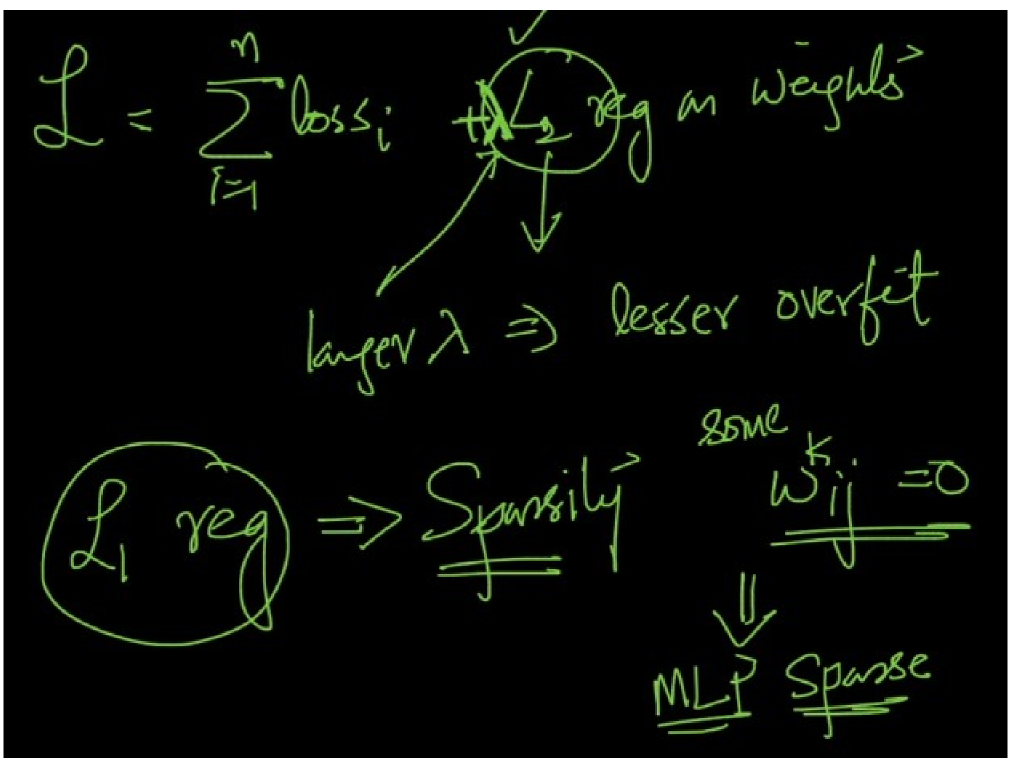


Consider we have 1000 data points and 1000 weights in the MLP. Then we have more chances of fitting to the each data point and make the MLP overfit the data vice versa.

We can optimize the under fitting and over fittting using the regularization same in case of MLP loss function.

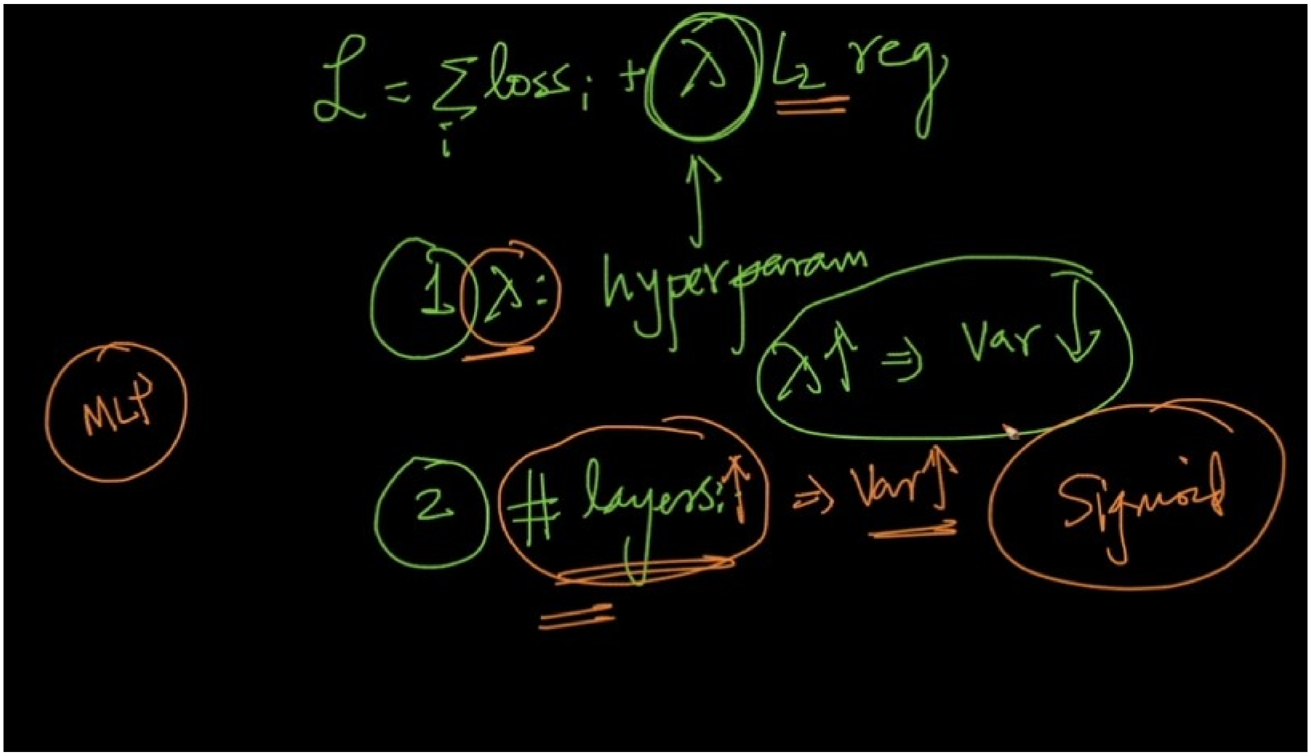


Make the loss as follows: There will be sparsity in case of L1 regularization.



L2 and L1 works for MLP's.

Hyper parameters:



Decision surfaces: Play ground: