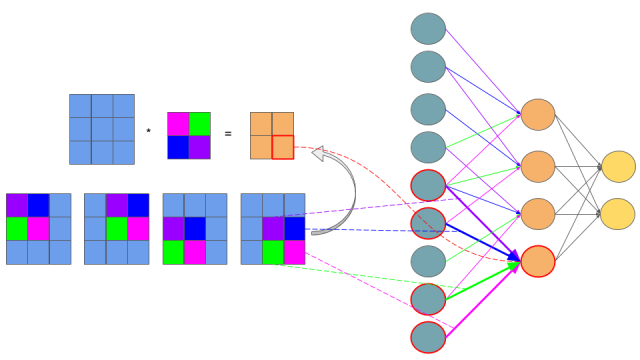
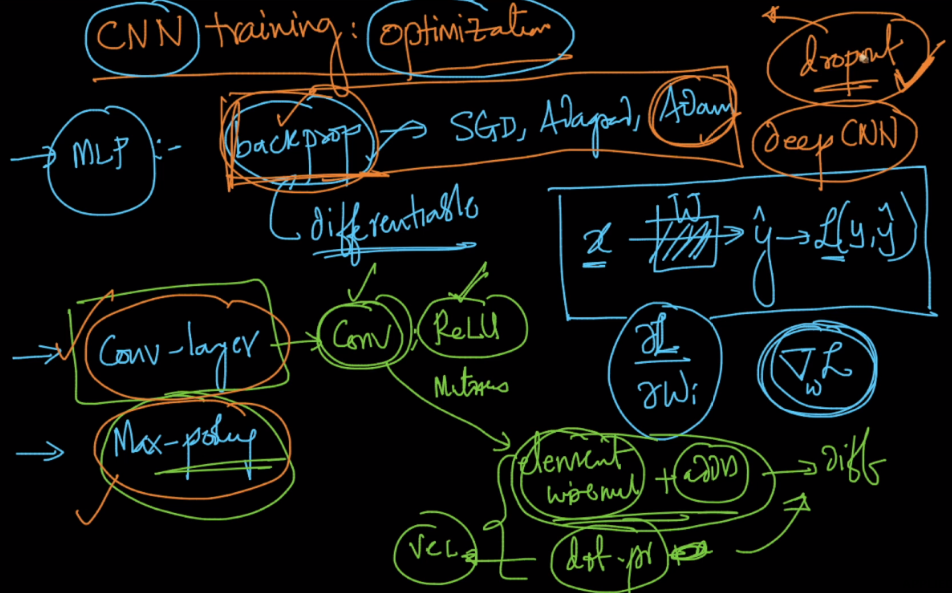
Training CNN is similar to MLP because the values of kernel can be think of as weight and hence they can be trained like them. And since it’s same we can apply all the concepts we have in MLP like dropout, optimizers, initializers etc. But training for Max pooling is different

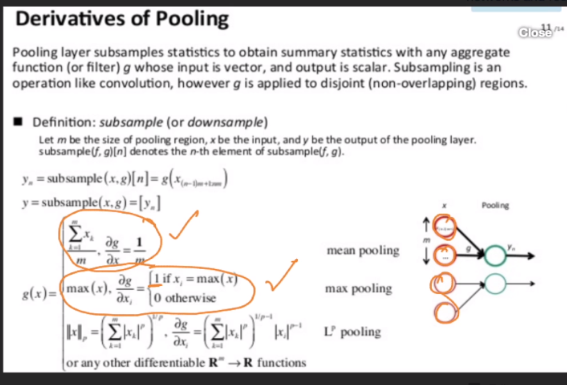
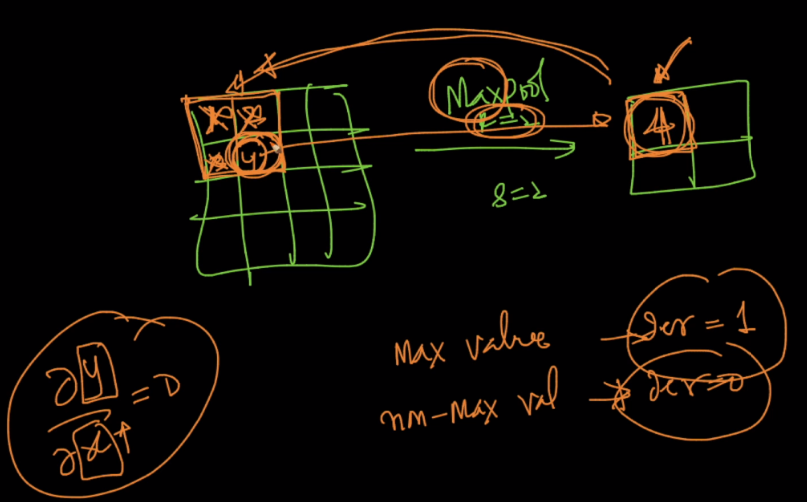


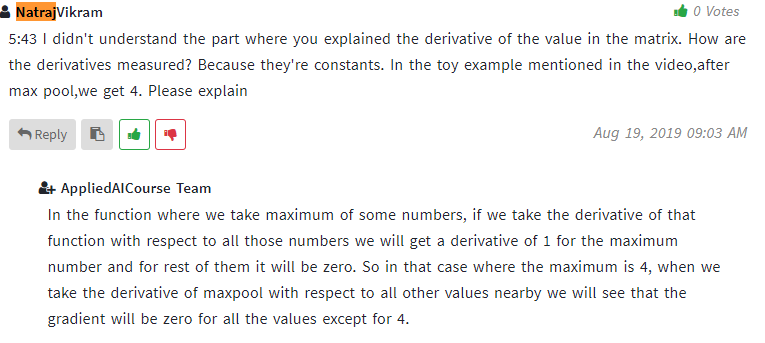
While doing backpropogation for training of max pooling layer the derivative will be calculated differently, why?

Because we can see that for below image and kernel, the any value in kernel will depend on 1 of the 4 values of image which is max, let’s say first value in kernel we get is 4, now differentiation of 4 w.r.t all other 3 values will be 0(because wkt differentiation of y wrt x is 0 when y is not changing when x changes).

And differention of kernel value 4 wrt image value 4 will be 1.

So differentiation for max pooling can be defined as 1 if kernel value == max value, else 0





Read below links at your own risk, I haven’t understood this

<https://becominghuman.ai/back-propagation-in-convolutional-neural-networks-intuition-and-code-714ef1c38199>

<https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/>

<https://grzegorzgwardys.wordpress.com/2016/04/22/8/>