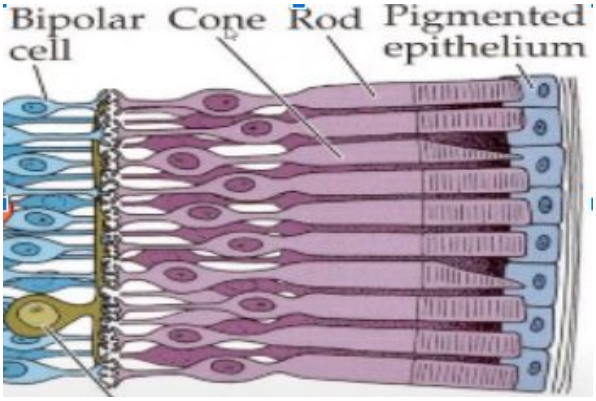
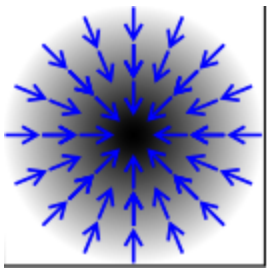
Q1>What are Channels and Kernels (according to EVA)?

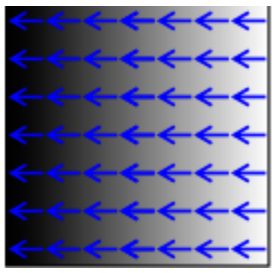
Channels:

To understand what channels are, we need to understand the mechanism of eyes.Internally our eyes see things by Rods and Cones shown in the figure . Rods are responsible for black and white (intensity) and Cones are responsible for Colors like R, G ,B. These are called channels. So human eyes have 4 channels i.e Black/white (intensity) channel ,Red channel,Green channel,Blue channel.The combination of all channels and their intensity produce the final image in the back of our head .



In general images are having 3 channels i.e RGB.Given a colored image of 100 pixels will have the shape of image as (10X10X3) , where 3 indicates number of channels.Each channel will have the intensity of that specific channel and combination of all the channels makes an image coloured .The channel here is a collection features and gredients of an image .Where feature is a piece of information about gradients and edges in the image . And gradients are basically the direction of change in color intensity .





Kernels:

In Convolution neural network kernel is a matrix ,which is used to extract the features from the images.The kernel moves over the image by performing dot product on a sub-region of the image pixel values and and sum then to get the output.This processes is known as convolution . The kernel matrix is randomly initiated and responsible for detecting edges and gredients in the image. Different kernels detect different edges and gredients.It moves over the image by stride values.

Q2>Why should we (nearly) always use 3x3 kernels?

To understand this let's think we have an image of hape (5 X 5).To get the 1st layer of convolution on a 3X3 kernel we have to move the kernel. Let's say we move it 2 times so we got (3x3+3x3) =18 and got more layers and lower number weights. So we got a smaller receptive field per convolution .So it will be able to get the small and complex feature of an image.Also as the wights are lower it will not be computationally heavy .

But If the kernel size is large then we will get a large receptive field and the convolution image will be generic. Hence it might not be able to capture the complex patterns in the image. Also because we have a higher number of weights we are doing high computation and get less information as layers will be less .Let's say we move a 5x5 kernel twice , we get (5x5 +5x5) = 50 which is a bigger number .
Hence we use a 3x3 kernel .

Q3>How many times do we need to perform 3x3 convolutions operations to reach close to 1x1 from 199x199 ?

With a 3x3 matrix on one convolution we lose 2 pixels . it means if we have a 5x5 metric to reach to 1x1 we need to go by 5x5 -> 3x3 -> 1x1 . A 2 time we performed convolution to reach .If we have 199x 199 then we need to have 100 convolutions to reach a 1x1 image. These convolution will be

199x199 =>, 197x197 =>, 195x195 =>, 193x193 =>, 191x191 =>, 189x189 =>, 187x187 =>, 185x185 =>, 183x183 =>, 181x181 =>, 179x179 =>, 177x177 =>, 175x175 =>, 173x173 =>, 171x171 =>, 169x169 =>, 167x167 =>, 165x165 =>, 163x163 =>, 161x161 =>, 159x159 =>, 157x157 =>, 155x155 =>, 153x153 =>, 151x151 =>, 149x149 =>, 147x147 =>, 145x145 =>, 143x143 =>, 141x141 =>, 139x139 =>, 137x137 =>, 135x135 =>, 133x133 =>, 131x131 =>, 129x129 =>, 127x127 =>, 125x125 =>, 123x123 =>, 121x121 =>, 119x119 =>, 117x117 =>, 115x115 =>, 113x113 =>, 111x111 =>, 109x109 =>, 107x107 =>, 105x105 =>, 103x103 =>, 101x101 =>, 99x99 =>, 97x97 =>, 95x95 =>, 93x93 =>, 91x91 =>, 89x89 =>, 87x87 =>, 85x85 =>, 83x83 =>, 81x81 =>, 79x79 =>, 77x77 =>, 75x75 =>, 73x73 =>, 71x71 =>, 69x69 =>, 67x67 =>, 65x65 =>, 63x63 =>, 61x61 =>, 59x59 =>, 57x57 =>, 55x55 =>, 53x53 =>,

51x51 =>, 49x49 =>, 47x47 =>, 45x45 =>, 43x43 =>, 41x41 =>, 39x39 =>, 37x37 =>, 35x35 =>, 33x33 =>, 31x31 =>, 29x29 =>, 27x27 =>, 25x25 =>, 23x23 =>, 21x21 =>, 19x19 =>, 17x17 =>, 15x15 =>, 13x13 =>, 11x11 =>, 9x9 =>, 7x7 =>, 5x5 =>, 3x3 =>, 1x1

Q4>How are kernels initialized?

The kernels in CNN are initialized with some normally distributed random numbers.Later these weights are learned or trained through back propagation .

Q5>What happens during the training of a CNN?

Initially the kernel is first initialized with some normally distributed random numbers.Then the kernel slides over the input image and does a summation on dot products in each slide. After every layer the kernel learns new edges.From new edges learns patterns and from patterns it learns the image.

The training involves images along with their labels. For example the label for 4 will be [0,0,0,0,1,0,0,0,0,0]. The output in the first run might vary .After that loss value is calculated which is basically mean squared error.The objective is to minimize the error . To do so it tries to adjust the weights to reduce error by back propagation .