1)why don't we start all of the weights with zeros ?

Ans : If all the weights are initialized to zeros, the derivatives will remain the same for every w in W[l]. As a result, neurons will learn the same features in each iteration. This problem is known as the network failing to break symmetry. And not only zero, any constant initialization will produce a poor result.

2)why is it beneficial to starts weights with a mean zero distribution ?

Ans ,: Initializing all the weights with zeros leads the neurons to learn the same features during training. In fact, any constant initialization scheme will perform very poorly. Consider a neural network with two hidden units, and assume we initialize all the biases to 0 and the weights with some constant α.all the weights are initialized with 0, the derivative with respect to loss function is the same for every w in W[l], thus all weights have the same value in subsequent iterations.

3)what is dilated Convolution and how does it works ?

Ans : Dilated Convolution it is a technique that expands the kernel by inserting holes between its consecutive elements. In simpler terms, it is the same as convolution but it involves pixel skipping, so as to cover a larger area of the input.

Dilated convolution helps expand the area of the input image covered without pooling. The objective is to cover more information from the output obtained with every convolution operation. This method offers a wider field of view at the same computational cost. We determine the value of the dilation factor (l) by seeing how much information is obtained with each convolution on varying values of l.

Formula :

(F\_{\*l}k)(p) = \sum\_{(s +lt = p)} F(s) k(t)

where,

F(s) = Input

k(t) = Applied Filter

\*l = l-dilated convolution

(F\*lk)(p) = Output

Advantages of Dilated Convolution:

1) Larger receptive field.

2) Computationally efficient (as it provides a larger coverage on the same computation cost)

3)Lesser Memory consumption (as it skips the pooling step) implementation

* 4)No loss of resolution of the output image.

5)Structure of this convolution helps in maintaining the order of the data.

4)what is TRANSPOSED CONVOLUTION & how does it works ?

Ans : Transposed convolutions are standard convolutions but with a modified input feature map. The stride and padding do not correspond to the number of zeros added around the image and the amount of shift in the kernel when sliding it across the input, as they would in a standard convolution operation.

A transposed convolutional layer, on the other hand, is usually carried out for upsampling i.e. to generate an output feature map that has a spatial dimension greater than that of the input feature map. Just like the standard convolutional layer, the transposed convolutional layer is also defined by the padding and stride. These values of padding and stride are the one that hypothetically was carried out on the output to generate the input. i.e. if you take the output, and carry out a standard convolution with stride and padding defined, it will generate the spatial dimension same as that of the input.

Implementing a transposed convolutional layer can be better explained as a 4 step process

Step 1: Calculate new parameters z and p’

Step 2: Between each row and columns of the input, insert z number of zeros. This increases the size of the input to (2\*i-1)x(2\*i-1)

Step 3: Pad the modified input image with p’ number of zeros

Step 4: Carry out standard convolution on the image generated from step 3 with a stride length of 1.

5)explain the separable convolution.

Ans : Separable Convolution is a process in which a single convolution can be divided into two or more convolutions to produce the same output. A single process is divided into two or more sub-processes to achieve the same effect.

Mainly there are two types of Separable Convolutions :

•Spatially Separable Convolutions.

•Depth-wise Separable Convolutions.

6)what is depthwise convolution. & how does it works.

Ans :.Depthwise Convolution is a type of convolution where we apply a single convolutional filter for each input channel. In the regular 2D convolution performed over multiple input channels, the filter is as deep as the input and lets us freely mix channels to generate each element in the output.

depthwise convolutions keep each channel separate. To summarize the steps, we:

•Split the input and filter into channels.

•We convolve each input with the respective filter.

•We stack the convolved outputs together

7)what is depthwise separable convolution. & how does it works.

Ans : In depth-wise operation, convolution is applied to a single channel at a time unlike standard CNN's in which it is done for all the M channels. So here the filters/kernels will be of size Dk x Dk x 1. Given there are M channels in the input data, then M such filters are required.

depth-wise separable convolutions.

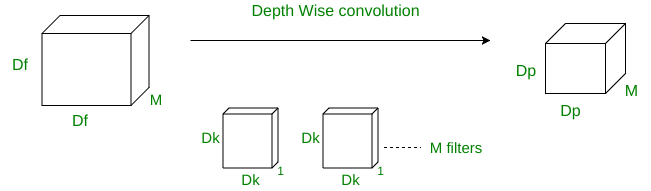
This process is broken down into 2 operations –

1.Depth-wise convolutions

2.Point-wise convolutions

1.DEPTH WISE CONVOLUTION :

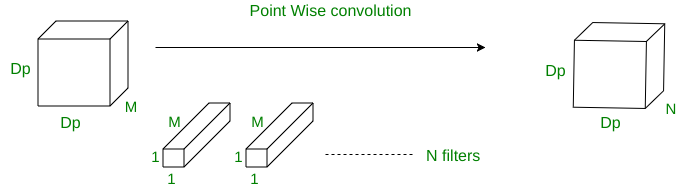
In depth-wise operation, convolution is applied to a single channel at a time unlike standard CNN’s in which it is done for all the M channels. So here the filters/kernels will be of size Dk x Dk x 1. Given there are M channels in the input data, then M such filters are required. Output will be of size Dp x Dp x M.



Total no of multiplications = M x Dk2 x Dp2

2.POINT WISE CONVOLUTION :

In point-wise operation, a 1×1 convolution operation is applied on the M channels. So the filter size for this operation will be 1 x 1 x M. Say we use N such filters, the output size becomes Dp x Dp x N.



Total no of multiplications = M x Dp2 x N

Total multiplications = Depth wise conv. multiplications + Point wise conv. multiplications

Total multiplications = M \* Dk2 \* Dp2 + M \* Dp2 \* N = M \* Dp2 \* (Dk2 + n)

So for depth wise separable convolution operation

Total no of multiplications = M x Dp2 x (Dk2 + N)

8)capsule networks are what they sound like.

Ans : A Capsule Neural Network (CapsNet) is a machine learning system that is a type of artificial neural network (ANN) that can be used to better model hierarchical relationships. The approach is an attempt to more closely mimic biological neural organization.capsnets address the "Picasso problem" in image recognition images that have all the right parts but that are not in the correct spatial relationship For image recognition, capsnets exploit the fact that while viewpoint changes have nonlinear effects at the pixel level, they have linear effects at the part level. This can be compared to inverting the rendering of an object of multiple parts.

9)why is a pooling such an Important operation in CNNs.

Ans : The pooling operation involves sliding a two-dimensional filter over each channel of feature map and summarising the features lying within the region covered by the filter.

For a feature map having dimensions nh x nw x nc, the dimensions of output obtained after a pooling layer is.

(nh - f + 1) / s x (nw - f + 1)/s x nc

where,

-> nh - height of feature map

-> nw - width of feature map

-> nc - number of channels in the feature map

-> f - size of filter

-> s - stride length

Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network.

The pooling layer summarises the features present in a region of the feature map generated by a convolution layer. So, further operations are performed on summarised features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image.

Types of pooling :

1)max pooling.

2)Average pooling.

3)global pooling.

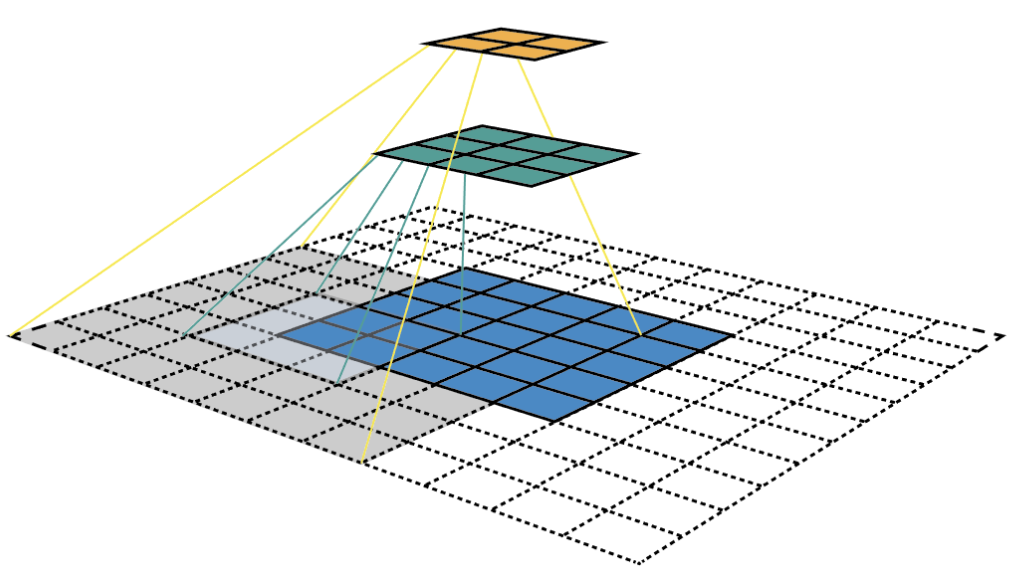
Convolutional layers are the basic building blocks of a convolutional neural network used for computer vision applications such as image recognition. A convolutional layer slides a filter over the image and extracts features resulting in a feature map that can be fed to the next convolutional layer to extract higher-level features. Thus, stacking multiple convolutional layers allows CNNs to recognize increasingly complex structures and objects in an image.

a major problem with convolutional layers is that the feature map produced by the filter is location-dependent. This means that during training, convolutional neural networks learn to associate the presence of a certain feature with a specific location in the input image. This can severely depress performance. Instead, we want the feature map and the network to be translation invariant.In a convolutional neural network, pooling is usually applied on the feature map produced by a preceding convolutional layer and a non-linear activation function.

10)what are Respective fields and how do they work ?

Ans : Receptive field is defined as the region in the input space that a particular CNN’s feature.We will use "feature" and "activation" interchangeably here and treat them as the linear combination (sometimes applying an activation function after that to increase non-linearity) of the previous layer at the corresponding location [3]. Because of the the convolution operation, features at different layers represent different sizes of region in the input image. As it goes deeper, the size represented by a feature gets larger. In this example below, we start with the bottom layer (5x5) and then apply a convolution that results in the middle layer (3x3) where one feature (green pixel) represents a 3x3 region of the input layer (bottom layer). And then apply the convolution to middle layer and get the top layer (2x2) where each feature corresponds to a 7x7 region on the input image. These kind of green and orange 2D array are also called feature maps which

refer to a set of features created by applying the same feature extractor at different locations of the input map in a sliding window fastion. Features in the same feature map have the same receptive field and look for the same pattern but at different locations. This creates the spatial invariance of ConvNet



This approach can actually work to some extent and is exatcly the idea of YOLO (You Only Look Once). The extra step taken by SSD is that it applies more convolutional layers to the backbone feature map and has each of these convolution layers output a object detection results. As earlier layers bearing smaller receptive field can represent smaller sized objects, predictions from earlier layers help in dealing with smaller sized objects Because of this, SSD allows us to define a hierarchy of grid cells at different layers. For example, we could use a 4x4 grid to find smaller objects, a 2x2 grid to find mid sized objects and a 1x1 grid to find objects that cover the entire image.