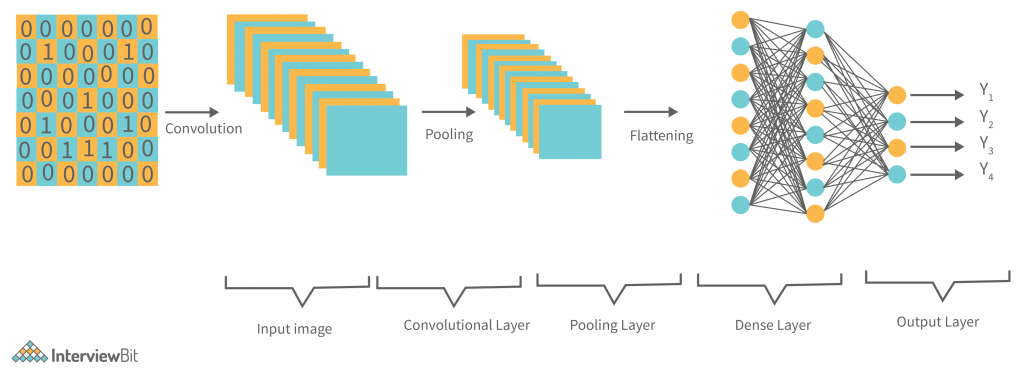
**DAY 09**

**28.06.2023**

**Convolutional Neural Networks**

Convolutional Neural Networks (CNN) are a type of multi-layer neural network that is meant to discern visual patterns from pixel images. In CNN, ‘convolution’ is referred to as the mathematical function. It’s a type of linear operation in which you can multiply two functions to create a third function that expresses how one function’s shape can be changed by the other.

In simple terms, two images that are represented in the form of two matrices, are multiplied to provide an output that is used to extract information from the image. CNN is similar to other neural networks, but because they use a sequence of convolutional layers, they add a layer of complexity to the equation. CNN cannot function without convolutional layers.





***Typical CNN Architecture***

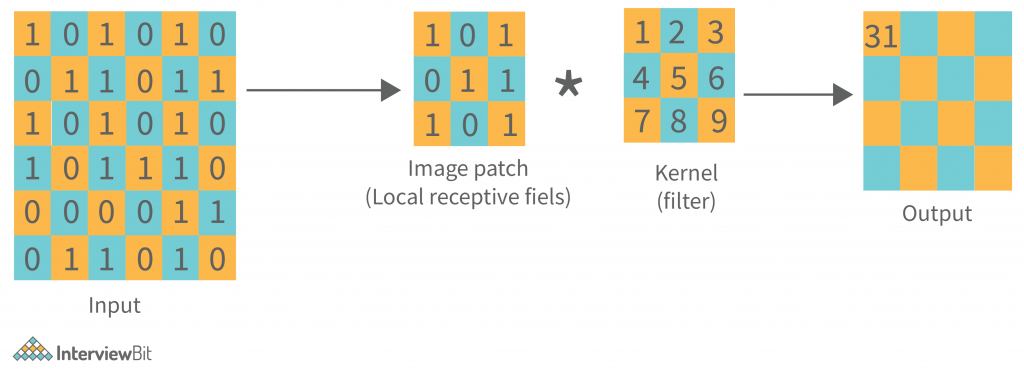
A convolutional neural network has following layers which are its building blocks

They are,

* Convolution Layer
* Pooling Layer
* Fully Connected Layer
* Activation Function
* Dropout Layer

***Convolutional Layer***

They are the foundation of CNN, and they are in charge of executing convolution operations. The Kernel/Filter is the component in this layer that performs the convolution operation (matrix). Until the complete image is scanned, the kernel makes horizontal and vertical adjustments dependent on the stride rate. The kernel is less in size than a picture, but it has more depth. This means that if the image has three (RGB) channels, the kernel height and width will be modest spatially, but the depth will span all three.



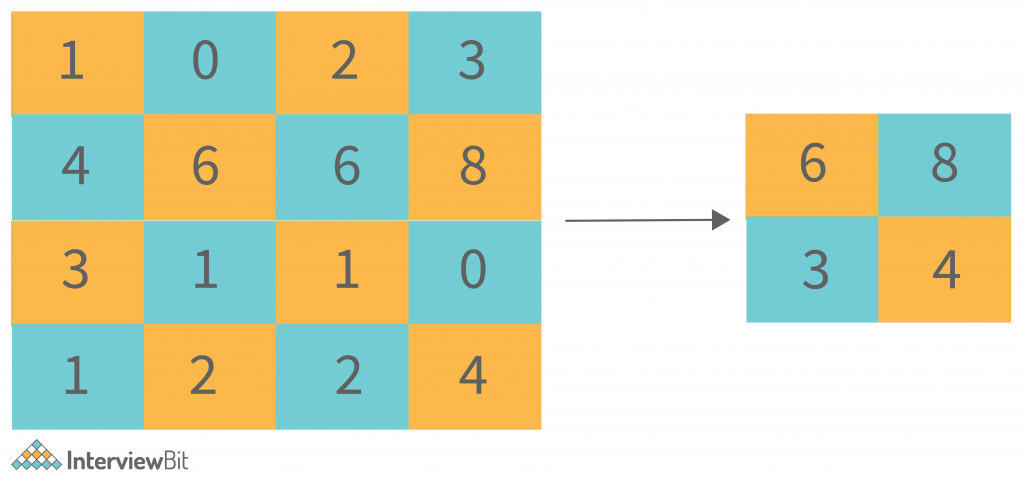


Other than convolution, there is another important part of convolutional layers, known as the Non-linear activation function. The outputs of the linear operations like convolution are passed through a non-linear activation function. Although smooth nonlinear functions such as the sigmoid or hyperbolic tangent (tanh) function were formerly utilized because they are mathematical representations of biological neuron actions. The rectified linear unit (ReLU) is now the most commonly used non-linear activation function.

**f(x) = max (0, x).**

***Pooling Layer******(POOL)***

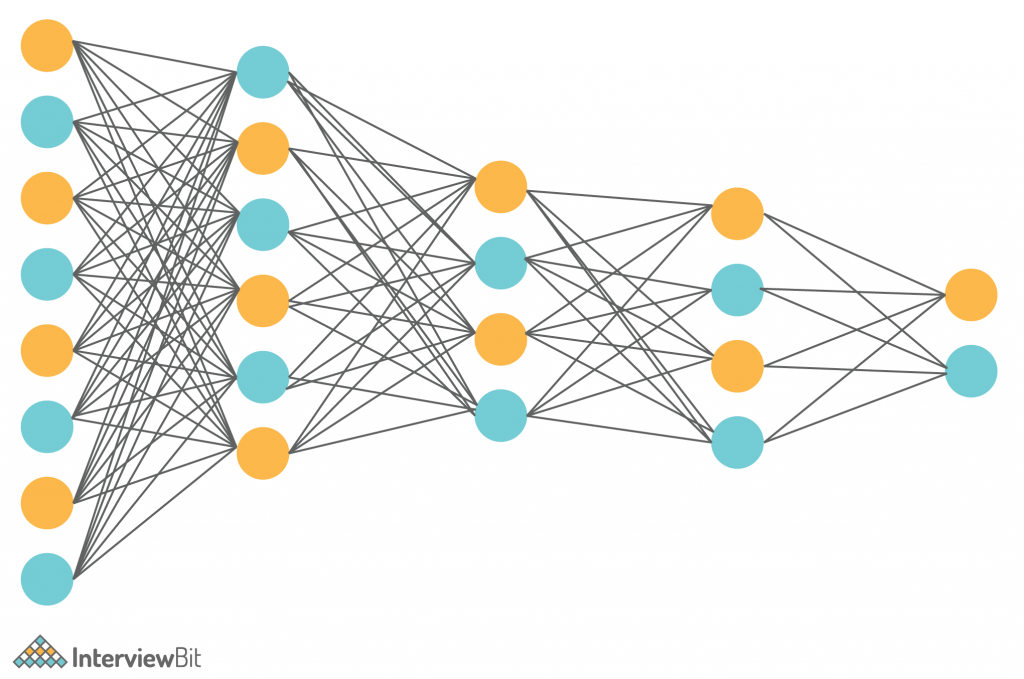
This layer is in charge of reducing dimensionality. It aids in reducing the amount of computing power required to process the data. Pooling can be divided into two types: maximum pooling and average pooling.





The maximum value from the area covered by the kernel on the image is returned by max pooling. The average of all the values in the part of the image covered by the kernel is returned by average pooling.

***Fully Connected Layer (FC)***





The fully connected layer (FC) works with a flattened input, which means that each input is coupled to every neuron. After that, the flattened vector is sent via a few additional FC layers, where the mathematical functional operations are normally performed. The classification procedure gets started at this point. FC layers are frequently found near the end of CNN architectures if they are present.

***Activation Function***

The last fully connected layer’s activation function is frequently distinct from the others. Each activity necessitates the selection of an appropriate activation function. The SoftMax function, which normalizes output real values from the last fully connected layer to target class probabilities, where each value ranges between 0 and 1 and all values total to 1, is an activation function used in the multiclass classification problem.

***Dropout Layers***

The Dropout layer is a mask that nullifies some neurons’ contributions to the following layer while leaving all others unchanged. A Dropout layer can be applied to the input vector, nullifying some of its properties; however, it can also be applied to a hidden layer, nullifying some hidden neurons. Dropout layers are critical in CNN training because they prevent the training data from overfitting. If they aren’t there, the first batch of training data has a disproportionately large impact on learning.

**References**

<https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243>

<https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/>

<https://www.upgrad.com/blog/basic-cnn-architecture/>

<https://www.tutorialspoint.com/microsoft_cognitive_toolkit/microsoft_cognitive_toolkit_convolution_neural_network.htm>