CNN: Convolution Neural Network

A Convolution Neural Network also known as ConvNet or CNN are a special kind of neural networks for processing data that has a known grid-like topology. It is used in object detection and image classification. It is one of the most successful neural networks that is inspired from the human visual processing system. It works in similar way as the brain's visual cortex processes images and recognize patterns.

CNN was introduced in 1988 as LeNet by Yann LeCun. It was used for digit and zip code recognition. Since then, it has gained huge popularity. Today, almost in every sector CNN is used. From self-driving cars, facial recognition to disease recognition using visual image CNN plays a vital role. It extracts important features from the input data using convolution, a mathematical operation that slides filters over the input data. The architecture of CNN is basically divided into three main layers.

1. Convolution Layer
2. Pooling Layer
3. Fully Connected Layer

Convolution Layer

In Convolution layer, we use small grids which are also called filters or kernels that slides over the image. It is the first building block, and the main mathematical task performed in CNN operation. Each small grids that slides over the image looks for specific patterns from the image like lines, curves, and shapes. As it moves, it creates new grid which highlights where it found these patterns. The new grids are also called feature map.

For Greyscale image of size (M * M) and we apply (N * N) grid/filter on it. P is padding (default 0) and S is Stride (default 1) then we get our feature map of size:

$$\left(\frac{M-N+2P}{S}\right)+1$$

The output size of the convolution would be (M * N * 1), the depth remains 1 for greyscale image.

For RGB image, M is the input image size, N is the filter size, P is Padding (default 0) and S is Stride (default 1). Then, the output feature would be:

$$\left(\frac{M-N+2P}{S}\right)+1$$

The output size of the convolution would be (M * N * 3), the depth remains 3 for RGB image.

Applying Convolution layer in image size (28 * 28 * 1): -

*layers.Conv2D(filters = 2, kernel_size = (3,3), activation = 'relu', input_shape = (28 * 28 * 1)*

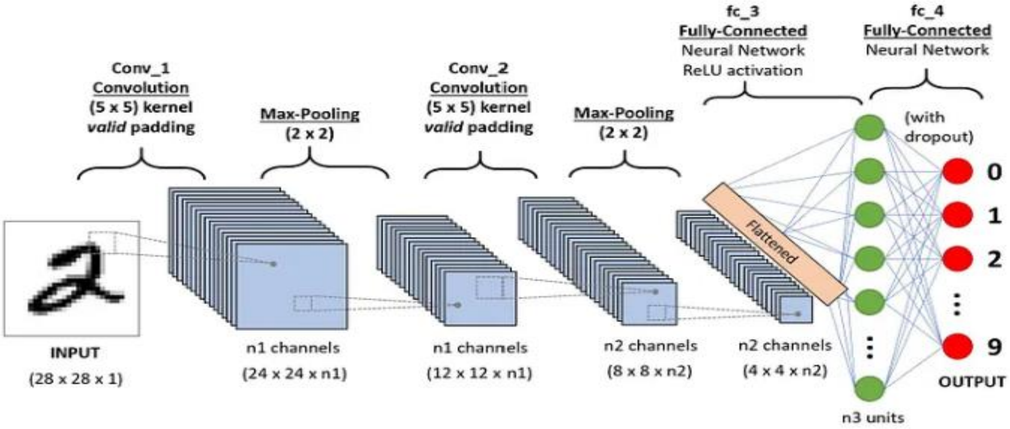
*Here, 2 filters of size 3 * 3 are applied. So, the final output would be (26 * 26 * 2). Depth 2 for 2 filters used.*

Pooling Layer:

The Pooling Layer is responsible for reducing the size of the convolved feature. This is to decrease the computational power required to process the data through dimensionality reduction. There are different types of pooling like Max Pooling, Average Pooling, Global Pooling, Stochastic Pooling, and Lp pooling. The two mainly used are Max and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Fully Connected Layer (FC Layer):

The fully connected layer also known as the dense layer plays critical role in convolutional neural networks. It is essential component of the network architecture. It captures global patterns and connections in the input data by connecting every neuron from the previous layer to every neuron in the fully connected layer. This layer is placed at the end of the CNN, following the convolutional and pooling layers. The primary function of the fully connected layer is to perform high-level reasoning and decision-making based on the features extracted by the preceding layers. It accomplishes this by learning complex non-linear mappings between the input and output data. Each neuron in the fully connected layer receives inputs from all the neurons in the previous layer and produces an output by applying a set of weights and biases, followed by an activation function.



A CNN sequence to classify handwritten digits