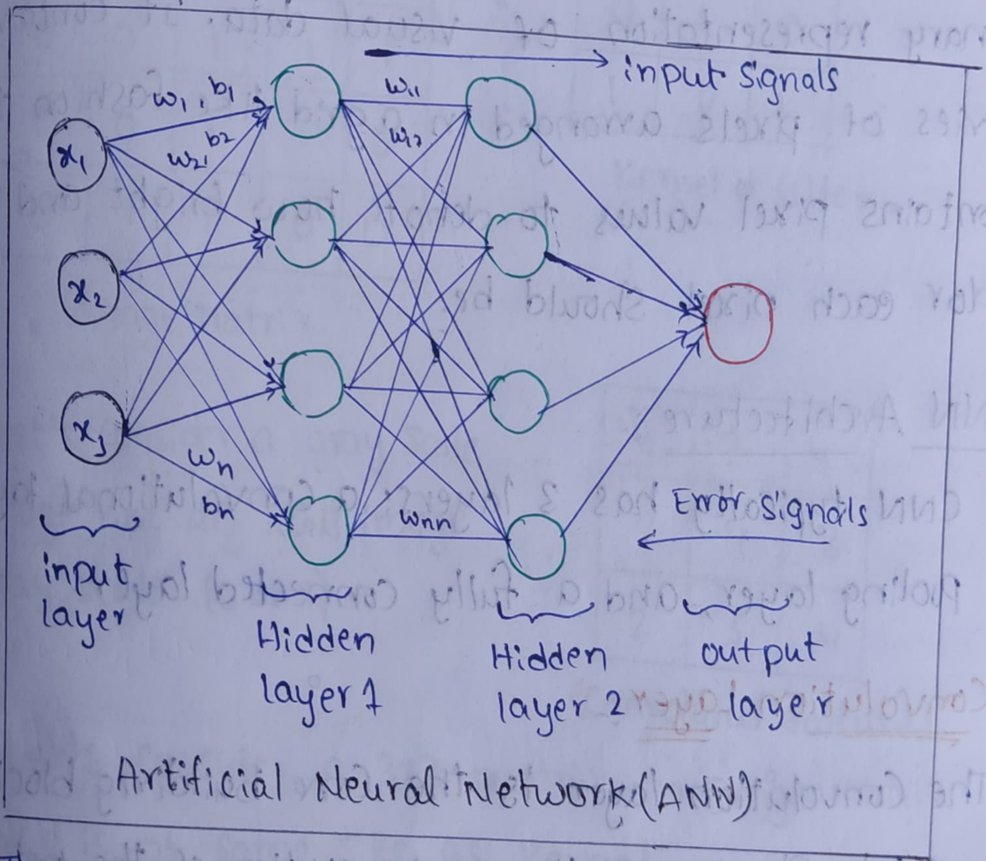
Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) are multi-layer fully-connected neural networks with an input layer, multiple hidden layers and an output layer. Every node in one layer is connected to every other node in the next layer.

→ We make the network deeper by increasing the number of hidden layers.



Artificial Neural Network (ANN)

→ The Artificial neural network takes input and computes the weighted sum of the inputs and includes a bias.

→ It determines weighted total is passed as an input to an activation function to produce the output.

→ Activation functions choose whether a node should fire or not. only those who are fired (ON) make it to the output layer. ~~there are~~

Convolutional Neural Network (CNN):-

A Convolutional Neural Network (CNN), also known as ConvNet, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid like fashion that contains pixel values to denote how bright and what color each pixel should be.

CNN Architecture:-

A CNN typically has 3 layers: a convolutional layer, a pooling layer, and a fully connected layer.

1. Convolution Layer:-

The Convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load.

→ This layer performs a dot product between 2 matrices where one matrix is the set of learnable parameters known as kernel or filter and other one is Image matrix.

→ The kernel is spatially smaller than an image but is more in-depth. This means that, if the image is composed of three (RGB) channels, the kernel height and width will be spatially small, but the depth extends up to all 3 channels.

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} =$$

Image Matrix

Kernel or filter

→ Let's consider a gray scale image not an RGB image
gray scale - (0 or 1)

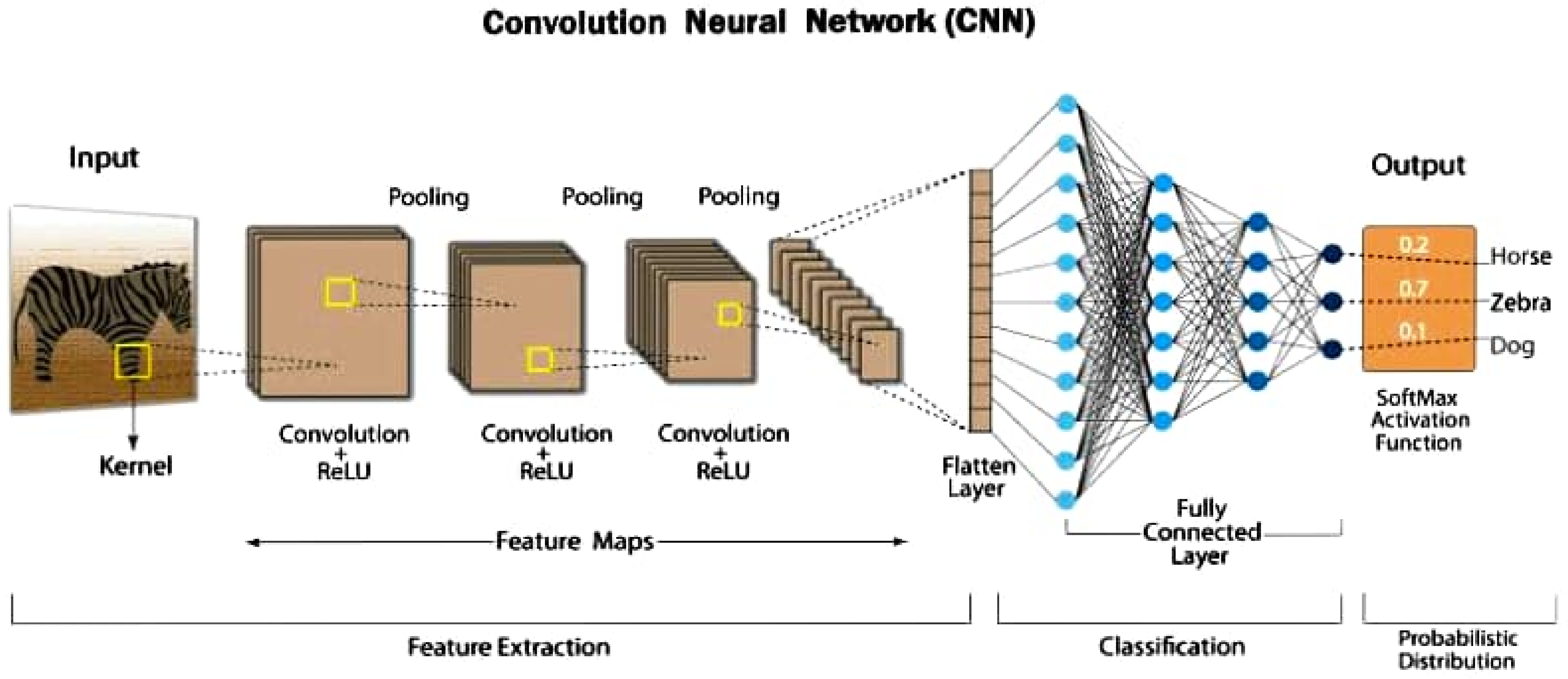
$$\begin{bmatrix} 4 & 3 & 4 \\ 2 & 4 & 3 \\ 2 & 3 & 4 \end{bmatrix}$$

(Convoluted feature)

1. Let's consider a 3x3 matrix in image matrix which is of same size as kernal.
2. Multiply this part with kernal and enter it in convoluted matrix. Let's take 1st part Red box

→ $(1\times1)+(1\times0)+(1\times1)+(0\times0)+(1\times1)+(1\times0)+(0\times1)+(0\times0)+(1\times1) = 4$

⇒ So, 4 has entered in first part of conv. matrix.


Convolution Neural Network (CNN)
Input
Kernel
Pooling
Pooling
Pooling
Convolution + ReLU
Convolution + ReLU
Convolution + ReLU
Feature Maps
Flatten Layer
Fully Connected Layer
Output
0.2
0.7
0.1
Horse
Zebra
Dog
SoftMax Activation Function
Feature Extraction
Classification
Probabilistic Distribution

Stride :- How many points or colums we have to shift in Image matrix for doing next Convolution part.

3. Now, move or go to next part Green box, this moving can be done by stride value. And Calculate Convolution.

4. Follow the same procedure for entire Convolution Matrix.

→ During the forward pass, the kernel slides across the height and weight of the image-producing the image representation of that receptive region.

→ This produces a 2-dimensional representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image. The sliding size of the kernel is called stride.

$$\boxed{W_{out} = \frac{W - F + 2P}{S} + 1}$$

W_{out} - dimensions of o/p Conv layer

W - dimensions of input image matrix

F - Filter/Kernal dimensions

P - Amount of Padding

S - Stride value.

Motivation behind Convolution:-

Trivial neural network layers use matrix multiplication by a matrix of parameters describing the interaction between the input & output unit. This means that every output unit interacts with every input unit.

→ However, CNN have sparse interaction. This is achieved by making kernal smaller than the input e.g., image can have millions or thousands of pixels, but while processing it using kernel we can detect meaningful information that is of ten's or hundred's of pixels.

→ This means that we need to store few parameters that not only reduces the memory requirement of the model but also improves the statistical efficiency of the model.

→ The layers of convolution neural network will have a property of equivariance to translation. It says that if we changed the input in a way, the output will also get changed in the same way.

2. Pooling layer:-

Pooling layers are used to reduce the dimensions of the feature maps. Thus it reduces the number of parameters to learn and 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.

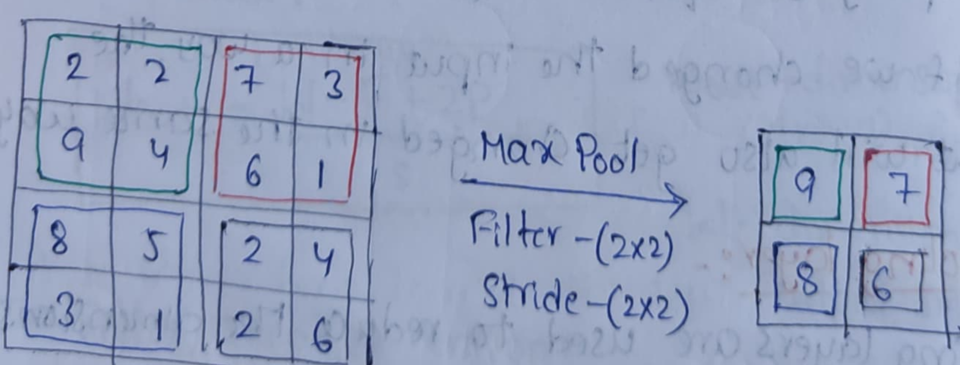
→ So, further operations are performed on summarised features instead of precisely positioned features generated by conv. layer.

→ This makes the model more robust to variations in the position of the features in the i/p image.

Types of pooling layers:-

1. Max Pooling:-

Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. So, the output would be feature map containing most prominent features of the previous feature map.



2. Average Pooling:-

Average pooling computes average of the elements present in the region of feature map covered by the filter.

$$\begin{bmatrix} 2 & 2 & 7 & 3 \\ 9 & 4 & 6 & 1 \\ 8 & 5 & 2 & 4 \\ 3 & 1 & 2 & 6 \end{bmatrix} \xrightarrow[\text{Filter-(2x2)}\ \text{Stride-(2,2)}]{\text{Average Pool}} \begin{bmatrix} 4.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}$$

3. Flattening:

Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into single long continuous linear vector.

$$\begin{bmatrix} 9 & 7 \\ 8 & 6 \end{bmatrix} \xrightarrow{\text{Flatening}} \begin{bmatrix} 9 \\ 7 \\ 8 \\ 6 \end{bmatrix}$$

Pooled feature map

→ The flattened matrix is fed as input to the fully connected layer (Neural Network) to classify the image.

Summary of CNN:-

→ Here's how exactly CNN recognises a image.

1. The pixels from the image are fed to the convolutional layer that performs the convolution operation.
2. It results in convolved map.
3. The convolved map is applied to a ReLU function to generate a rectified feature map.
4. The image is processed with multiple convolutions

and ReLU layers for locating the features.

5. Different pooling layers with various filters are used to identify specific parts of the image.

6. The pooled feature map is flattened and fed to a fully connected layer.

7. The output from fully connected layer is passed through a softmax function to get the final output.

Recurrent Neural Network (RNN)