K. Monisha
1817132
B.E. CSE IV Yr

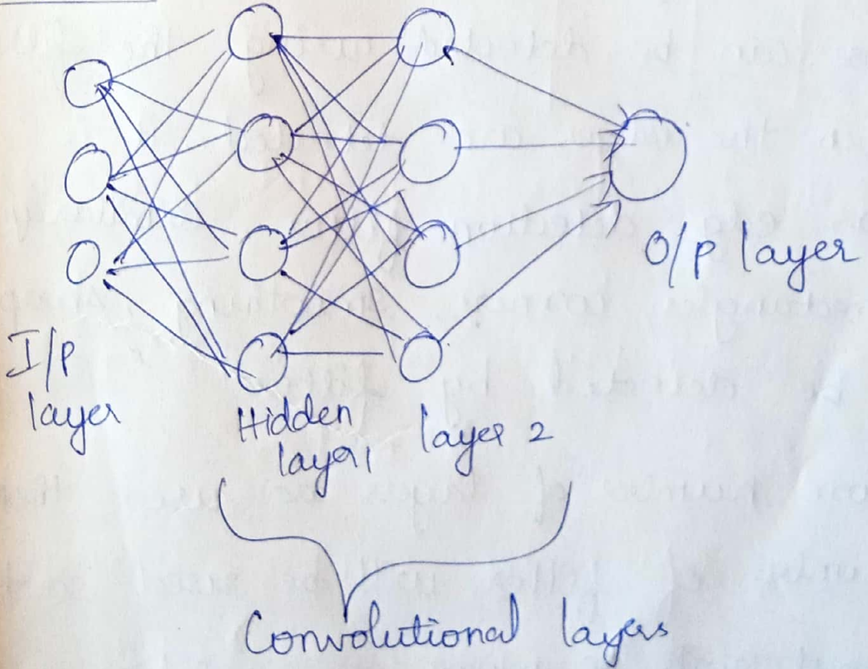
CONVOLUTIONAL NEURAL NETWORKS :-

→ CNN is an ANN - Artificial Neural Network Prominent application of CNN is in image analysis

→ CNN can be seen as an ANN has some specialization and detects patterns in the images. CNN has hidden layers called convolutional layers. There can be one or more hidden layers.

→ It has non convolutional layers too. But, basis is convolutional layers. It does the operation.

Diagram



Here, when i/p passed to hidden layer, it does some mathematical operation and it is pushed to next hidden layer. after this multiple filtering occurs, finally, the result is pushed to o/p layer.

Convolution

→ In simple terms, convolution is a mathematical operation on two functions (ie) f and g that produces a third function expressing how the shape of one is modified by the other.

→ CNN is all about finding patterns. This is fundamentally connected the number of filters

→ Filters are the ones which helps totally in detecting the patterns. When an image is given, their edges, objects, shapes, textures etc all these can be detected using the filters.

→ When the edges are detected, it is called as edge detection filter. Similarly, squares, rectangles, corners, smoothing, sharpenin etc can be detected by filter

→ More number of layers are used, then more number of filter will be used so that the output will be more appropriate.

→ First hidden layer is convolution layer. A filter is a matrix with rows and column.

Now, 3 X 3 matrix

$$\begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix}$$

⇒ The filter shown ~~above~~ shall slide over i/p block (matrix)

Sliding is called convolving. Filter is going to convolve and is the crux.

Eg. ⇒ 5 x 5 matrix to be convolved with the filter chosen.

I/p matrix

$$\begin{vmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix} \xrightarrow[\text{convolved}]{\text{To be}} \begin{vmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{vmatrix}$$

Step 1

The filter should slide in the first 3 x3 matrix 5X5 matrix

$$\begin{bmatrix} 1\times 1 & 1\times 0 & 1\times -1 & 0 & 1 \\ 0\times 1 & 1\times 0 & 0\times -1 & 0 & 1 \\ 0\times 1 & 1\times 0 & 1\times -1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

⇒ add the results and put in 1st conv field.

0		

Step ②

Now, the sliding happens.

$$\begin{bmatrix} 1 & 1(1) & 1(0) & 0(-1) & 1 \\ 0 & 1(1) & 0(0) & 0(-1) & 1 \\ 1 & 1(1) & 1(0) & 0(-1) & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{array}{|c|c|c|}\hline 0 & 3 & \\ \hline & & \\ \hline & & \\ \hline \end{array}$$

Step ③

$$\begin{bmatrix} 1 & 1 & 1\times1 & 0\times0 & 1\times-1 \\ 0 & 1 & 0\times1 & 0\times0 & 1\times-1 \\ 1 & 1 & 1\times1 & 0\times0 & 1\times-1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{array}{|c|c|c|}\hline 0 & 3 & -1 \\ \hline & & \\ \hline & & \\ \hline \end{array}$$

. Similary, all the remaining rows is calculated. The final ~~co~~ result is now convoluted.

0	3	-1
0	2	-1
0	0	0

⇒ This filtered matrix may undergo one more round of convolution.

→ No. of hidden layers and convolving shall get one final convolved matrix.

⇒ Filters is selected ~~by~~ as -1, 1, 0 whereas -1 represents black, 1 represents white, 0 represents grey. It can have many forms such as $\begin{bmatrix} -1 & -1 & -1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$, $\begin{bmatrix} -1 & 1 & 0 \\ -1 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$ etc.

Three Important parameters that decides the size of an image :-

i) Depth

ii) Stride

iii) Zero Padding.

→ Zero Padding is very important. If 5x5 matrix is convolved with 3x3 filter, the output will be of 3x3. (ie) it shrinks. When the important data present in the edges, it may Case the information. So, to avoid this, Zero Padding is defined.

⇒ When we convolved with any filter, the dimension of the input image is calculated by the following formula.

$$\boxed{O/P\ size = (n-f+1) \times (n-f+1)}$$

→ n - input size

→ f ~~x f~~ - filter size

(eg) $n = 5$

$f = 3$

$O.S = (5-3+1)(5-3+1)$

$= (3)(3)$

(ie) 3x3 matrix

→ Zero Padding is nothing but, it padded the zeros at the edges .

(eg)
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

⇒ **Depth** → It is based on number of filters used. When n no. of different filters are used, then the depth of feature map is also n.

⇒ **Stride** → Sliding is important here (ie) moving 1 pixel at a time corresponds to stride 1. if 2 pixel moves, then it is stride 2.

Max Pooling

After the every convolutional layer in CNN, Max Pooling is added which is pure math and it reduce the dimension of the image

(eg) when 8x8 matrix is input imag, after max pooling with 2*2 filter, and stride = 2 the output is down sampled to 4*4.

$$\begin{bmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow[\substack{2\times2 \text{ filter} \\ \text{stride}=2}]{\text{Max Pooling}} \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$$

⇒ o/p is derived from the max of all 4 pixels

→ Here, strong pixels are retained, low values are not.

→ This reduces the computational load (ie) lesser the pixels to handle, easier the computation.

→ This also helps in avoiding overfitting

Average Pooling ⇒ It is same as max pool but here, it takes average of all the values in the region. But it is not preferred over the max pooling as it fails with the detection of sharp edges and other complex features.

Sum Pooling ⇒ Instead of taking average or maximum value, the sum of all pixels in chosen region is calculated. Sum pooling also is preferred next to the Max pooling in applications.

Activation Function

→ Activation function typically follows some layer.

→ Activation function of a neuron defines output of the neuron given set of inputs

→ AF operates on the value which can then be transformed to anything between lower limit and upper limit (say 0 and 1).

Sigmoid. → It is an activation function

For sigmoid, zero is the lower limit, one is the Upper limit.

i) If i/p is negative, transforms this number close to '0'

ii) If i/p is positive, transforms this number close to '1'.

iii) If the input is close to 0, ~~t~~transforms this number close to 0 and 1.

Role of activation function :-

Firing is - One

No firing is - zero.

Closer to 'one is more activated the neuron. Lesser the activation, when is closer to '0'.

ReLu → Rectified Linear Unit .

-> It actually transforms the input to 0 or input value itself .

=> When i/p value is negative (ie) less than 0 or equal to 0, it will make it as 0 or 0 is the o/p.

=> If i/p is greater than 0, then o/p shall be nothing else than the given i/p.

```
if (x<=0)
 { return 0;
  } else
   { return x;
   }.
```

$f(x) = \max(0, x)$.