UNIT-II

# Convolution Neural Network

Page No.
Date

Q] What is Convolution Neural Network CNN Discuss applications of CNN.

→ ~~Grove~~ Convolution Neural Network (CNN) is type of deep learning model specially designed for processing data with grid-like structure such as image.

- CNN represents the input data in the form of multidimensional data it works well with large number of labeled data.

- The Goal of CNN is reduce the image so that it would be easier to process without loosing features that are valuable for accurate prediction.

- Lets take the example of human brain as is context we see the image we classify the image by its color, shape, size and the message it is upto ~~to~~ to convey. Similarly for machine image is just the array of pixels. Machine tries to learn the unique pattern in the image & computer tries to find these pattern & information from the image.

- Most of the companies have adopted the CNN for image recognition.

- Machine can be trained by giving tons of images to learn & find patterns.

Page No.
Date

Application of CNN:

1) Image classification

2) Face Recognition

3) Medical image

4) Self-driving cars

5) Object detection (Yolo, R-CNN)

6) Video analysis

7) Gesture Recognition

Q) Explain convolution operation in CNN with example. Explain descrete & circular convolution operation. Take input 5*5 input data, 3*3 Kernel data & calculate Convoluted features.

→ • Convolution operation focus on extracting the feature from input. Convolution operation allows the network to detect horizontal & vertical edges of an image & based on those edges high-level features are build.

• Convolution operation uses three parameters i) Input image ii) feature detector iii) Feature map

Page No.
Date

- Convolution operation uses input matrix & filter i.e., also known as Kernel. Input matrix are pixel value of grayscale image and filter is small matrix that detect the edges of image.

- Input image is converted to binary 1 & 0 format.

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |

Input

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

Kernel

Step by Step o/p

$$[0,0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \begin{matrix} 1+0+0 \\ +0+1+0 \\ +1+0+1 \end{matrix} = 4$$

$$+[0,1] - \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 3$$

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} K \end{bmatrix} = 3$$

⋮

Final Output ~~feature~~ map (3x3)

$$\begin{bmatrix} 4 & 3 & 3' \\ 3 & 3 & 2 \\ 4 & 2 & 5 \end{bmatrix}$$

1] Descrete convolution →

- It is method of combining two descrete signals to produce a third signal.

- In CNN it is used to apply kernels (filters) ~~to~~ for feature extraction.

- Usually zero-padding is used in kernel goes outside the boundary.

$$y[n] = \sum_{m=-\infty}^{\infty} x[m] \cdot h[n-m]$$

- Op^ size depends on padding and stride.

2] Circular convolution →

- Works like discrete convolution but wraps around instead of filling with 0.

- Output size is always same as input.

$$\left[\left(\text{Descrete convo}\right) \bmod N\right]$$

$$\sum_{m=0}^{N-1}$$

Page No.
Date

Q] What are pooling layers in CNN? Explain its types with example

→ • pooling is technique in CNN to reduce the spatial size of the representation. Its main purpose is to decrease the amount of parameters and computation in the network.

• pooling layer takes the output of previous layer & compute summary of neighbourhood around specific location.

• The main purpose of pooling is to.
- reduce computation by lowering number of value
- Control overfitting by reducing parameters
- Reduce memory requirement for storage.

• The pooling layer will always reduce the size by factor of 2.
Suppose you have image of (6x6) pixel 36px
it will be pooled (3x3) 9px

• Pooling operation is also called as subsampling

• Pooling layer is also called as downsampling It reduces the number of parameters in the input.

There are two types of pooling
i] Max pooling
ii] Avg pooling

A] max pooling
- Takes maximum value in the window
- Keeps strongest activations

1 3
2 4
o/p = 4

B) Average pooling
- Takes average of values in the window.
- produce smoother output

1 3
2 4

(1+2+3+4)/4 = 2.5

Q] Explain the following hyper parameters for the convolution layer:
i] filter size
ii] Output depth
iii] Stride
iv] zero-padding

→ i] Filter size.
- filter size is the height x width of the matrix sliding over the input image.
- common filter sizes are:
  - 3x3, 7x7, 5x5
- Two types of filters are
  1) small filter → fine detail (eg edges)
  2) Large filter → broad patterns (eg texture)

- Each filter has values (weights) learned during training.
- multiple filters detect different shapes.
- Example:

3x3 filter $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$

ii] Output depth

- Number of feature maps created by ~~Output~~ convolutional layer.
- Each feature map = result of 1 filter
- more filters → more feature detected.
- Different filter detects diff color, corner, edges.
- Output depth increase with number of filters.
- more depth = more computation.

eg → 28x28x3 → RGB image.

filer : 8 ~~filters~~ filters (3x3x3 each)

o/p → 26x26x8 (assume no padding, stride1)

iii] Stride.

~~stride~~ : number of pixels filter moves horizontally and vertically over the pixels of the input during convolution.

- Stride is component for compression of images & video data. ex if NN stride = 1 the pixel will move by 1unit at time.
- Stride is smaller if we want fine/high details in our output.
- Stride is higher if we want macro level details.

eg → Input = 6x6
filter = 3x3 no padding
stride1 => 4x4 } outputs.
stride2 => 2x2

## 4] Zero-padding

- padding refers to adding extra pixels around the boundary to an image before applying the convolution.
- without padding the output gets ~~zero~~ reduce after each convolution.
- Zero padding helps making the output dimension & kernel size dependent.
- Three common zero padding are-

1) Valid convolution →
- No zero-padding used
- filters only processes areas fully inside the image
- Output size is smaller than input.

Input = 5x5, filter = 3x3, o/p = 3x3

2) Same convolution

- padding added so output size = i/p size.
- kernel size is F: i/p is padded by F-1 zeros

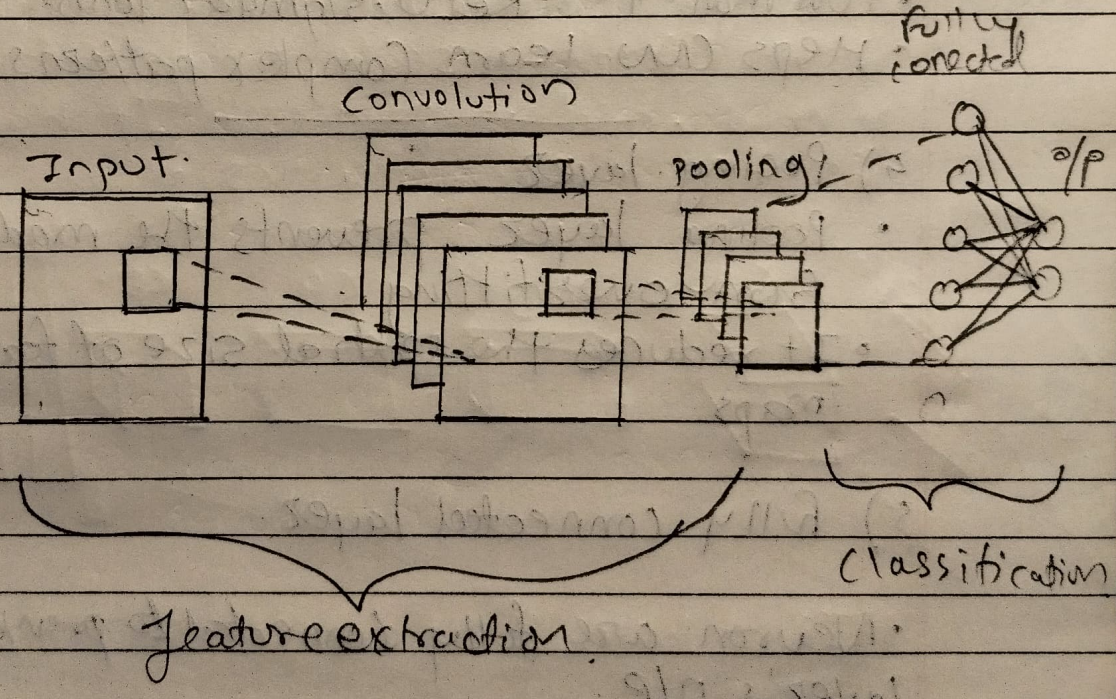
eg I/p = 5X5 ~~filter~~, filter = 3X3 → Add 1 layer of zeros → O/p = 5X5

3) Full convolution

- padding added so filter can process all possible overlaps : including partial ones.
- O/p > than i/p

eg 5X5 ; filter = 3X3 Add 2 layers of zeros → O/p = 7X7.

8] Explain Basic architecture of CNN




1] Input layer: This layer accepts the raw data such as image.
dimensions → height x width x depth.
eg → 12X12X4

## 2) Convolution layer.

- Core layers performing convolution using filters / Kernels.
- Detects local features like edges, corners & textures
- filters are learned during training.

## 3) Activation function layer

- This layer applies activation f^n to each element in the output of the convolution layer
- Comman f^n : ReLU, sigmoid, Tanh.
- Helps CNN learn complex patterns.

## 4) Pooling layer

- Pooling layer prevents the model from overfitting.
- It reduces the spatial size of feature maps

## 5) Fully connected layer.

- Neuron are fully connected to previous layer's o/p
- Converts extracted features into a final classification output.

Page No.
Date

Function of hidden layers in CNN.

1] Learning features.

2] Dimensionality Reduction.
- Pooling reduces feature map size.

3] Translation invariance.
- pooling ensures detection even if featur. slightly shift position.

4] Hierarchical Representation.
- Layer-by-Layer CNN builts a hierarchy.
- low level → mid-level → high level Features.

Page No.
Date

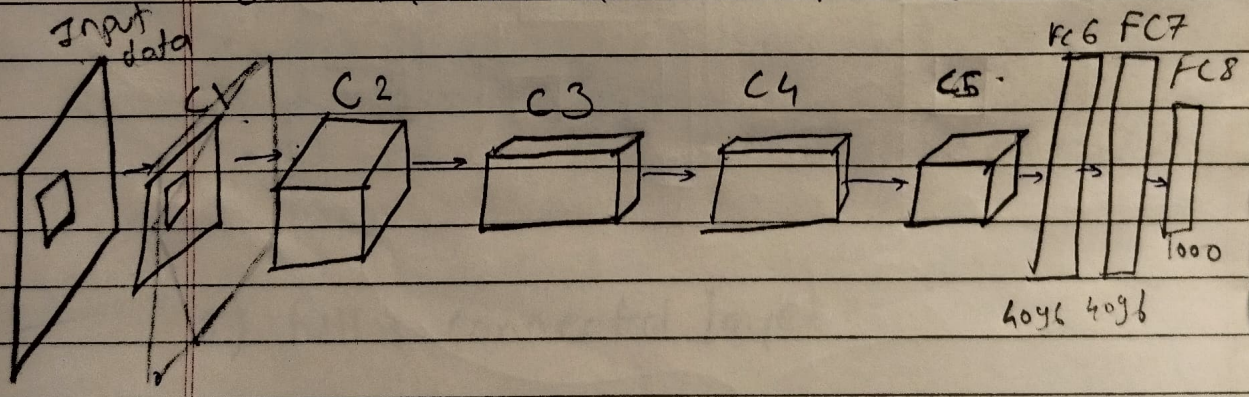
9] Draw & explain architecture of Alexnet

→ • Alexnet is Deep convolution NN for image Classification

• Alexnet compose of 5 Convolution layer with combination of max-pooling layer. 3 fully connected layers, & 2 dropout layers

• The Activation function used in all layer is ReLU. & Activation f^n used in output layer is softmax.

• Alexnet architecture was created with large-scale image datasets in mind, ~~and it produced~~ It has 60 million characteristics & ab




1) Input layer I → accepts 227x227x3 image.

2) Conv layers → There are five convolution layers in total. These five layers uses convolution Kernels to generate feature maps.

3) Max-pooling layers → Max pooling layer is applied betⁿ every two convolutional layers

4) Fully connected layers → After the convolutional & pooling layers the tensors are flattened and fed into three fully connected layers

5) Dropout layers: Two dropout layers are used to reduce overfitting which was a major issue due to model have 60M parameters

6) Output layer → Last layer is softmax layer which computes the loss fⁿ for learning & provides the classification output.