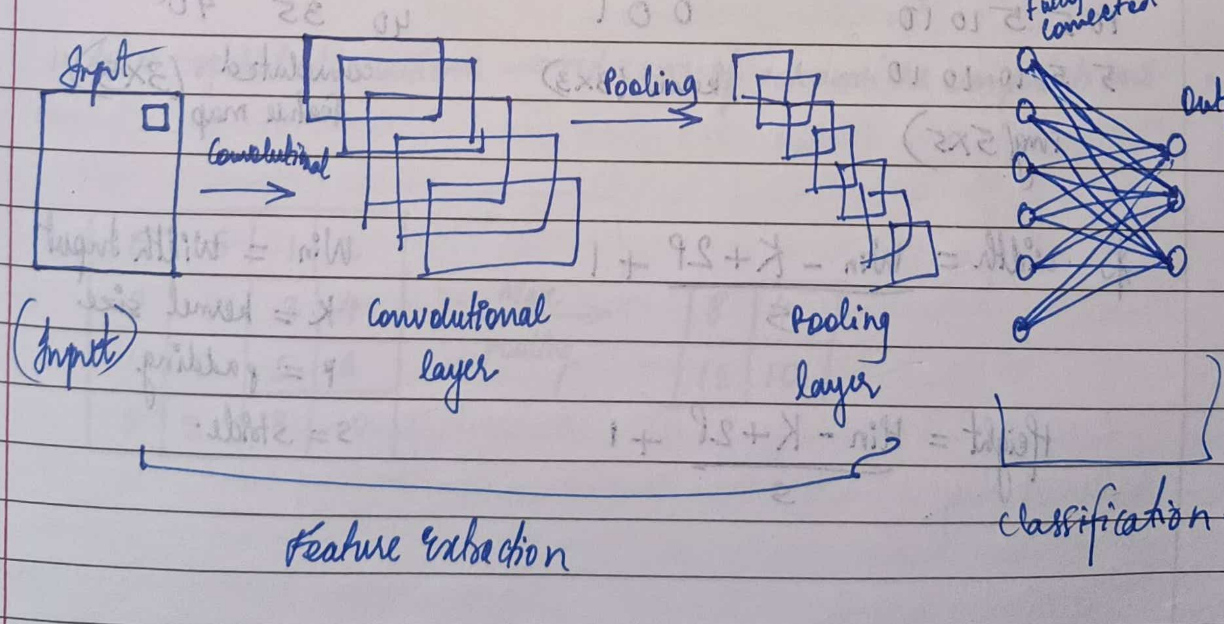
U-3 (DL)

Page No.
Date

Convolution Neural Network.

- Network Architecture for deep learning.
- learns directly from imgs. CNN is madeup of several layers [→ process, → transform]
- CNN → img analysis tasks, scence classification, object detection & segmentation
- Adv → • CNN automatically detects imp features without any human supervision.
  - computationally efficient
  - higher accuracy
  - weight sharing
  - minimize computation
  - uses same knowledge across all img locations.
- Disadv → • adversarial attacks → case of feeding network bad eg to cause misclassification.
  - tend to be much slower because of like maxpool.
- Appl^n → • object detection • Voice synthesis • Voice synthesis • Astrophysis.
  - image segmentation • recognition of handwritting.

CNN Architecture -



CNN 4 layers → Convolution
Pooling
Relu
Fully connected (classification

1) Convolutional layer -

- ~~Foundation~~ of CNN,
- responsible ~~for~~ executing convolution operations.
- Kernel/Filter component → performs convolution.
- feature extraction
- detects patterns & features using learnable kernels or filters
- detects diff types of filters.
- whole img scanned → kernels makes adjustments horizontally & vertically according to stride rate.

stride rate → no. of steps taken by kernel.

$$\begin{matrix} 5_{\times1} & 10_{\times1} & 5_{\times1} & 10 & 5 \\ 10_{\times1} & 5_{\times1} & 10_{\times1} & 5 & 10 \\ 5_{\times0} & 10_{\times0} & 10_{\times1} & 5 & 5 \\ 10 & 5 & 5 & 10 & 10 \\ 5 & 5 & 10 & 10 & 10 \end{matrix} \times \begin{matrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \Rightarrow \begin{matrix} 35 & 40 & 30 \\ 40 & 40 & 45 \\ 40 & 35 & 40 \end{matrix}$$

img(5x5) kernel (3x3) convoluted feature map (3x3)

$$\text{width} = \frac{W_{in} - K + 2P}{S} + 1$$

$$\text{Height} = \frac{H_{in} - K + 2P}{S} + 1$$

Win = Width Input
K = kernel size
P = padding
S = stride.

* 2) Pooling Layer –

- helps reduce dimensionality, saves computational complexity.
- Pooling can be max pooling or avg pooling.
- Max pooling → Takes max value from a kernel
- Averge Pooling → Takes avg of all values from a kernel.

$$\begin{bmatrix} 35 & 40 & 30 \\ 40 & 40 & 45 \\ 40 & 35 & 40 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 40 & 45 \\ 40 & 45 \end{bmatrix}$$

input feature map — 2×2 pooling window — downsampled feature map (max pooling)

* Pooling layer include following parameters –

1) Pooling window size – size of pooling region 2×2 or 3×3.
2) Stride – step by which window moves horizontally & vertically across feature map.
3) Padding – allows to control size of pooling layer maps by adding y zeros around input features.

- reduces spatial dimensions → reduce no. of parameters computations.

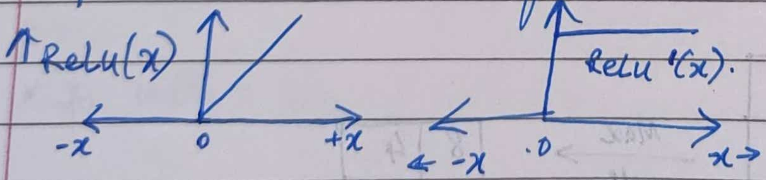
$$\begin{bmatrix} 5 & 6 & 1 & 2 \\ 7 & 8 & 3 & 4 \\ 11 & 6 & 5 & 6 \\ 3 & 2 & 8 & 10 \end{bmatrix} \xrightarrow{\text{Max pooling}} \begin{bmatrix} 8 & 4 \\ 12 & 10 \end{bmatrix}$$

3) Fully Connected Layer -

- every input connected → neuron → called flattened input.
- multilayer preception or neural network where each neuron is connected to all other neurons in previous layer.
- connec" b/w neuron have associated weight which is learned using training process
- output of neuron → applying a activation fun" to weighted sum of inputs
- allows network → complex non-linear re" b/w extracted features.
- large no. of parameters can lead to overfitting esp for high dimensional data.
- To mitigate we apply regularization, drop out batch normalization
- max pooling → noise reduction with dimensionality reduction
- To avoid loosing pixels → padding → add extra pixels edges of input matrix
- amt of sliding → input tensor → stride.

4) ReLU

- activation fun" decides if neuron should be fired or not by weighted sum & input bias
- simplicity & performance → used alot
- Rectified Linear Unit ReLU = max(x, 0)
- provides non-linear transformation • keeps all +ve elements & discards all -ve elements



- Relu avoids vanishing gradient descent problem.
- Dropout layers help avoid overfitting on training data.

adv →
- efficient to compute
- doesn't ~~saturate~~ saturate
- less likely to suffer vanishing gradient problem.

• non linear activation fun^c used in CNN

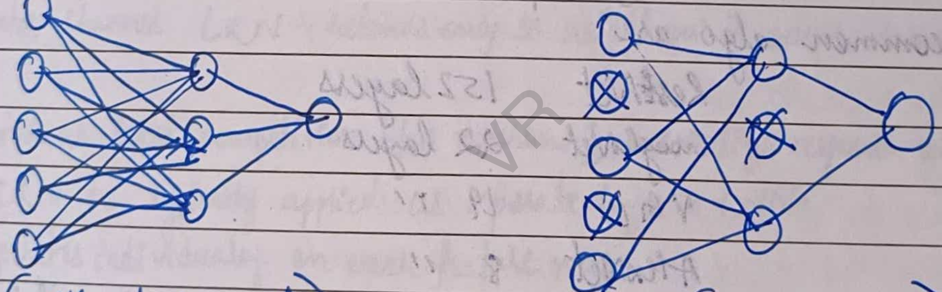
$$f(x) = \begin{cases} 0 & x < 0 \\ x & \end{cases}$$

max(0, x)

Page No.
Date

5) Droop out layer –

- Regularization technique.
- randomly dropping out or setting zero neurons in fully conected layer.
- done for preventing overfitting.
- purpose is to force network to learn more robust & redundant representation of & encourages diverge representation.
- forward pass → dropout layer scales active neuron by factor of $\frac{1}{1-\text{dropoutrate}}$ to component for dropped at in backward ~~pass~~ pass, dropped out neurons don't contribute to gradient update



(without dropout) (with dropout)

Padding –

- pixels on corners & edge are less used compared to those in middle.
- info on borders of img is not preserved as well as info from center.
- tackle this we add padding. adds rows & columns of zeros on outline of input image. adding padding → new dimensions $(W+2P)\times(W+2P)$

x	y	z
a	b	c
m	n	p

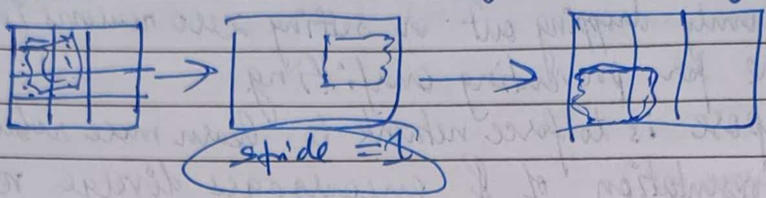
$P=2$ →

```
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 x y z 0 0
0 0 a b c 0 0
0 0 m n p 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

- adding padding
  ↳ increase contribution of corner pixels to learning process.

Page No.
Date

* Stride – • in convolutional layers, filter slides from left to right & top to bottom until it passes through entire input ime.
• steps of filter it takes while study over the image.



* Interleaving b/w layers –

- diff layers in CNN can be mixed or repeated any no. of time, interleaving of layers.
- allows research & experimentation.
- common algo are

ResNet	152 layers
GoogleNet	22 layers
VGL	19 "
AlexNet	8 "

- specific interleaving patterns depend on network archi & task at hand
- designed to leverage strengths of diff layer types & facilitate effective feature extraction, non-linearity down sampling & regularization.
- convolution, pooling & Relu layers are interleaved in order to express power.
- Relu follows Convolutional layers.
- Interleaved means they are alternated in b/w
- Let C → Conv P → Pooling R → Relu
  one interleaving → CRCRP or CRCRCRP.

* Training CNN → 1) Input 2) Convolution → define kernel size, padding & stride
3) Pooling layer → define type of pooling, padding, stride, pooling window size.
4) Additional Conv layers of Pooling layers 5) Fully connected → activation func
6) Train → forward pass – 1, backward pass – back propagation initial weights & bias learning loss
7) Test & evaluate.

Page No.
Date

* Local Response Normalization -

- unbounded activation funⁿ require normalization.
- normalization is applied before activation func.
- used in CNN for normalization
- was first introduced in AlexNet.
- Introduced lateral inhibition increases competition among neurons.
- helps to normalize & scale them for a particular neuron based on activities of neighbouring neuron.
- Improve robustness of networks to noise & learn more features.
- works by normalizing activation of convolution layer within local region.
- prevent activations from too large or too small to prevent model from being oversensitive to noise.

1) Inter Channel LRN - normalization is done in depth dimens. for each (x,y) pair.
2) Intra Channel LRN - extends only to neighbour in same channel.

- Introduces local competition b/w neurons & normalize response within specific neighbourhood
- LRN is typically applied as seperate layer in CNN.
- operates individualy on each feature map
- inserted to convolution & before pooling to normalize activation before downsampling.

Adv → • Improved generalization
- better discriminative power.
- resisten to variation in illuminator or constrast
- amplify diff b/w strong & weak activation.

- Replaced by Batch Normalization in modern CNN architechre to as it operates on batch of sample rather than neighbourhood makeing it more efficient & easier to train.