CONVOLUTION NEURAL NETWORKS – CNN

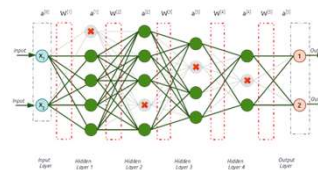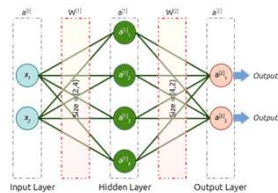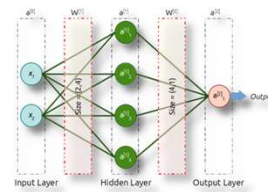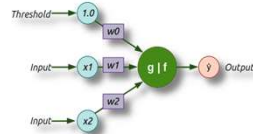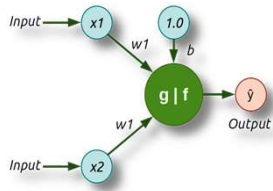Introduction to Machine Learning, Artificial Intelligence and Deep Learning

Session 15

Pramod Sharma
pramod.sharma@prasami.com

Story So Far…

2

pra-sâmi

Agenda

3

Computer Vision

Convolution

Padding

Pooling

Frameworks

pra-sâmi

Acknowledgement…

4

Geoffrey Everest Hinton CC FRS FRSC

❑ is an English Canadian cognitive psychologist and computer scientist, most noted for his work on artificial neural networks.

❑ Since 2013, he divides his time working for Google (Google Brain) and the University of Toronto. In 2017, he cofounded and became the Chief Scientific Advisor of the Vector Institute in Toronto.

❑ With David Rumelhart and Ronald J. Williams, Hinton was co-author of a highly cited paper published in 1986 that popularized the backpropagation algorithm for training multi-layer neural networks, although they were not the first to propose the approach.

❑ Hinton is viewed as a leading figure in the deep learning community.

❑ The dramatic image-recognition milestone of the AlexNet designed in collaboration with his students Alex Krizhevsky and Ilya Sutskever for the ImageNet challenge 2012 was a breakthrough in the field of computer vision.

pra-sâmi

5

What is Computer Vision…

pra-sâmi

6

Ambulance given green light all through….

pra-sâmi

7

Computer vision is making progress in leaps and bounds…

7/12/2023

pra-sâmi

8

Convolutional neural networks (CNN, ConvNet) is a class of deep, feed-forward (not recurrent) artificial neural networks that are applied to analyzing visual imagery.

7/12/2023

pra-sâmi

Computer Vision

9

- ❑ Self driving car

- ❑ Fully automated warehouse and ports
  - ❖ https://youtu.be/RFV8IkY52iY

- ❑ Image search services,

- ❑ Unlock phone

- ❑ Provide access to secure area
  - ❖ Open your house
  - ❖ Enter office without your access card

- ❑ Object identification Apps
  - ❖ Garment
  - ❖ Food,
  - ❖ Nature

- ❑ Natural style transfer

- ❑ Automatic video classification systems
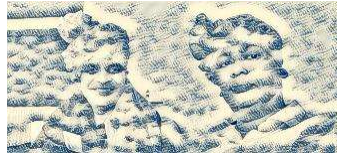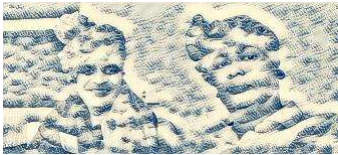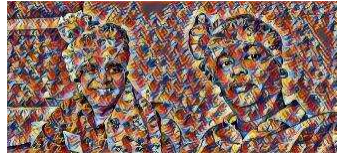
7/12/2023

pra-sâmi

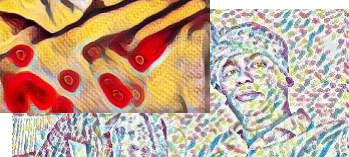## Style Transfer

10

7/12/2023

pra-sâmi

Style transfer

pra-sâmi

Style transfer

pra-sami

Computer Vision

13

- ❑ Have been used in image recognition since the 1980s

- ❑ Increase in computational power, the amount of available training data, CNNs have managed to achieve better performance

- ❑ Rapid advancement
  - ❖ Newer and Newer products and applications are coming up
  - ❖ Some of you will get a chance to directly work on these advance applications

- ❑ The development community is also very kind in sharing their success stories

- ❑ The ideas can be borrowed in other applications:
  - ❖ Voice recognition
  - ❖ Natural language processing (NLP)

7/12/2023

pra-sâmi

Computer Vision

14

- ❑ What makes vision hard?

- ❑ Vision needs to be robust to a lot of transformations or distortions:
  - ❖ Change in pose/viewpoint
  - ❖ Change in illumination
  - ❖ Deformation
  - ❖ Occlusion (some objects are hidden behind others)

- ❑ Many object categories can vary wildly in appearance (e.g. chairs)

"Imaging a medical database in which the age of the patient sometimes hops to the input dimension which normally codes for weight!" - Geoff Hinton
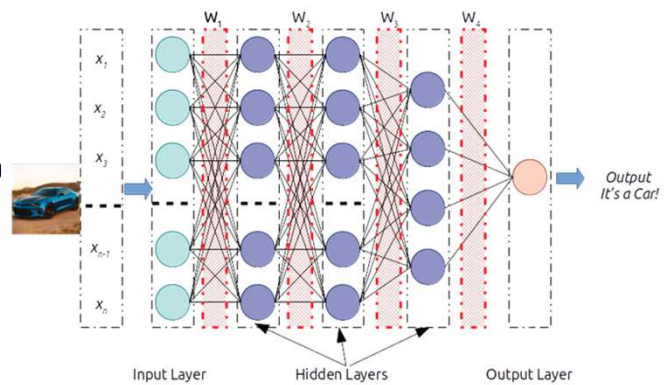
7/12/2023

pra-sâmi

Why?

15

- ❑ Enough of sales pitch…

- ❑ Why not simply use a regular deep neural network with fully connected layers?

- ❑ Small (150 x 150 x 3) image has 67,500 pixels

- ❑ If we consider first hidden layer as 1000,

- ❑ First weight matrix $W^{[1]}$ will be 67,500 x 1000

- ❑ Do your math…… that size is huge

7/12/2023

pra-sâmi

Smaller Network: CNN

16

- ❑ We know it is good to learn a small model

- ❑ Fully connected model, each hidden unit is processing every input
  - ❖ Do we really need all the edges?

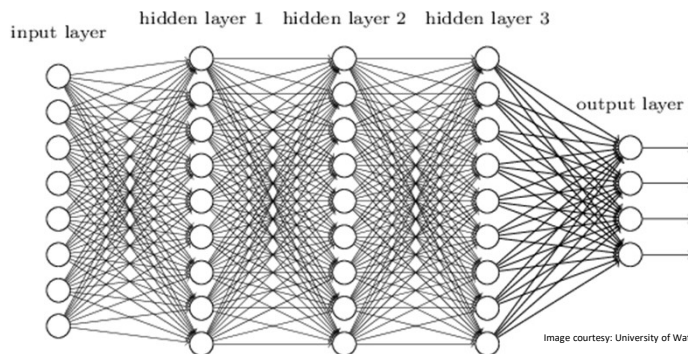- ❑ Can some of these be shared?

Image courtesy: University of Waterloo.

7/12/2023

pra-sâmi

17

Images are high-dimensional vectors. It would take a huge amount of parameters to characterize the network.
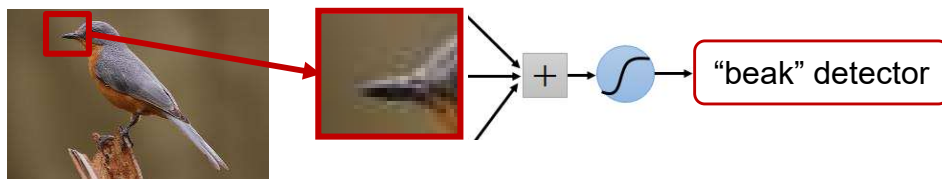
7/12/2023

pra-sâmi

18

Learning an image…

❑ Some patterns are much smaller than the whole image

❑ Can represent a small region with fewer parameters

"beak" detector

7/12/2023

pra-sâmi

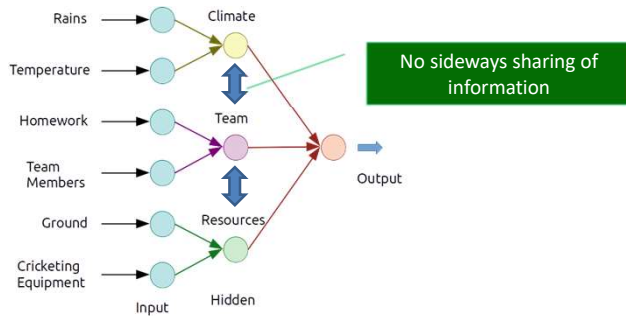## Learning an image…

19

- ❑ Same pattern appears in different places
  - ❖ Can they be compressed!

- ❑ What about training a lot of such "small" detectors and each detector must "move around"
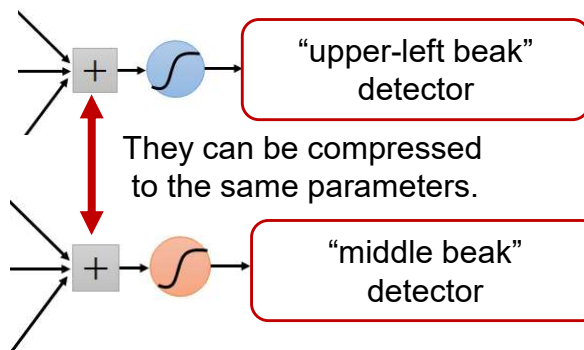

No sideways sharing of information

7/12/2023

pra-sâmi

Learning an image…

20

"upper-left beak" detector

They can be compressed to the same parameters.

"middle beak" detector

7/12/2023

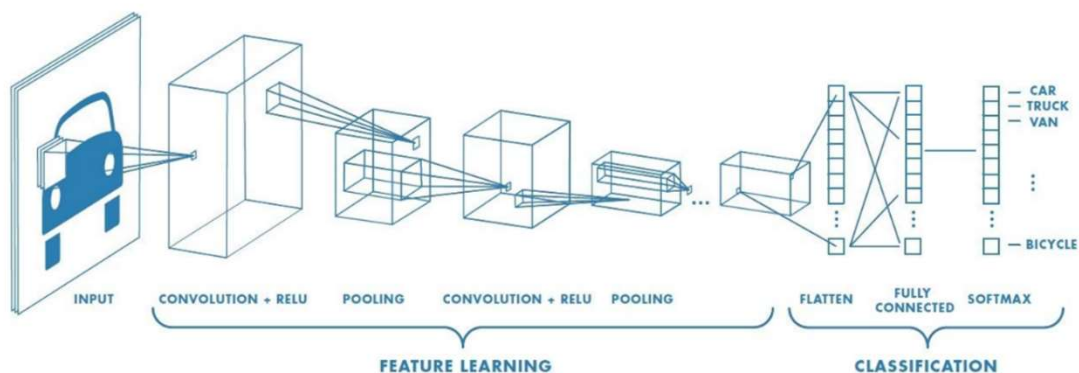pra-sâmi

Learning an image…

21

- ❑ The same sorts of features that are useful in analyzing one part of the image will probably be useful for analyzing other parts as well.
  - ❖ E.g., edges, corners, contours, object parts

- ❑ We want a neural net architecture that lets us learn a set of feature detectors that are applied at all image locations

- ❑ So far, we've seen a bunch of types of layers
  - ❖ Fully connected layers (dense)
  - ❖ Embedding layers (i.e. lookup tables)
  - ❖ A few more in RNNs (GRU, LSTMs, etc.)

- ❑ Different layers could be stacked together to build powerful models

- ❑ Let's add another layer type: the convolution layer

7/12/2023

pra-sâmi

Overall Layout

22

7/12/2023

pra-sâmi

23

Each column of hidden units looks at a small region of the image, and the weights are shared between all image locations.

pra-sâmi

24

What computer sees…

pra-sâmi

pra-sâmi

A Convolutional Layer

❑ A CNN is a neural network with some convolutional layers
  ❖ And, of course, a few other layers

❑ A convolutional layer has a number of filters that does convolutional operation
  ❖ Some of the literature would call it Kernel

pra-sâmi

What does this Convolution Filter/ Kernel do?

27

$*$

0	1	0
1	4	1
0	1	0

pra-sâmi

What does this Convolution Filter/ Kernel do?

28

$*$

0	-1	0
-1	8	-1
0	-1	0

pra-sâmi

What does this Convolution Filter/ Kernel do?

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

7/12/2023

pra-sâmi

What does this Convolution Filter/ Kernel do?

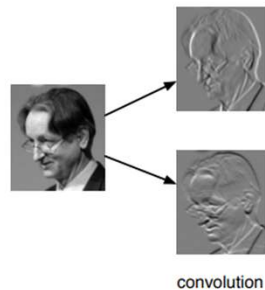$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$
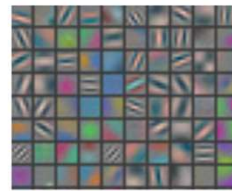
7/12/2023

pra-sâmi

Convolutional Networks

31

- ❑ Two kinds of layers:
  - ❖ Detection layers (or convolution layers)
  - ❖ Pooling layers

- ❑ The convolution layer has a set of filters.
  - ❖ Output is a set of feature maps, each one obtained by convolving the image with a filter.

Example first-layer filters

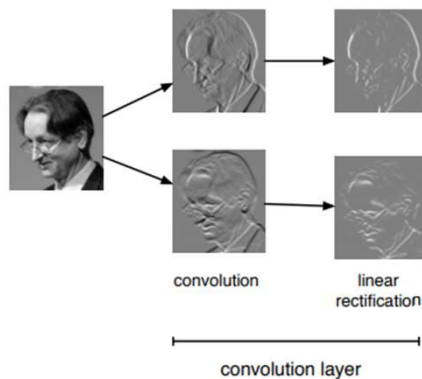(Zeiler and Fergus, 2013, Visualizing and understanding convolutional networks)

convolution

7/12/2023

pra-sâmi

Convolutional Networks

32

- ❑ It's common to apply a linear rectification (activations) nonlinearity or something else:
  - ❖ y_i = Relu(z_i), Tanh(z_i), etc.

- ❑ Convolution is a linear operation.

- ❑ Therefore, we need a nonlinearity, otherwise 2 convolution layers would be no more powerful than 1

- ❑ Two edges in opposite directions shouldn't cancel

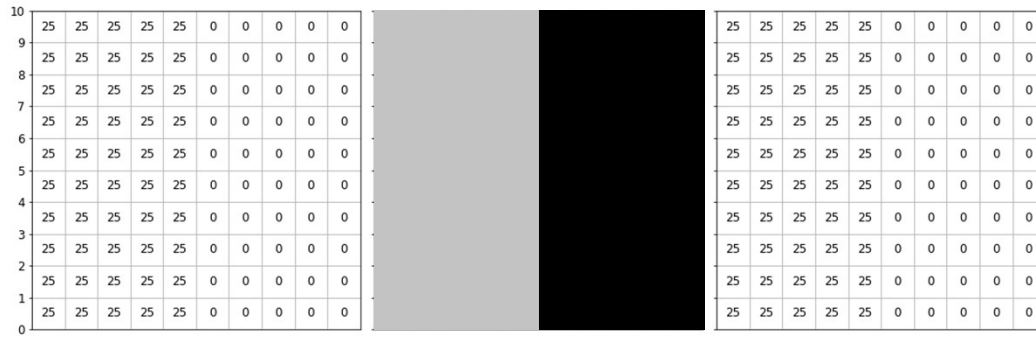- ❑ Non-linearity makes the gradients sparse, which helps optimization

convolution

linear rectification

convolution layer

7/12

pra-sâmi

16

Image with a edge

33

- ❑ Convolution is basic building block of image recognition

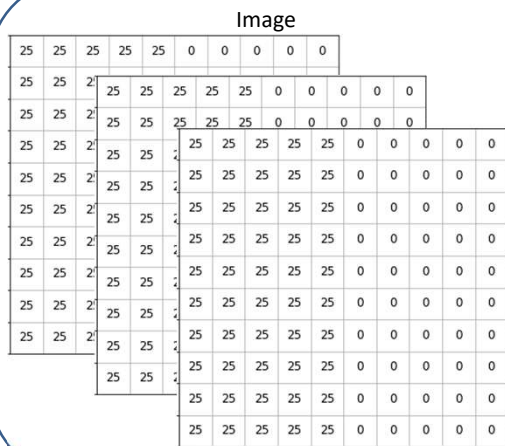- ❑ Using edge detection as an example in following image…
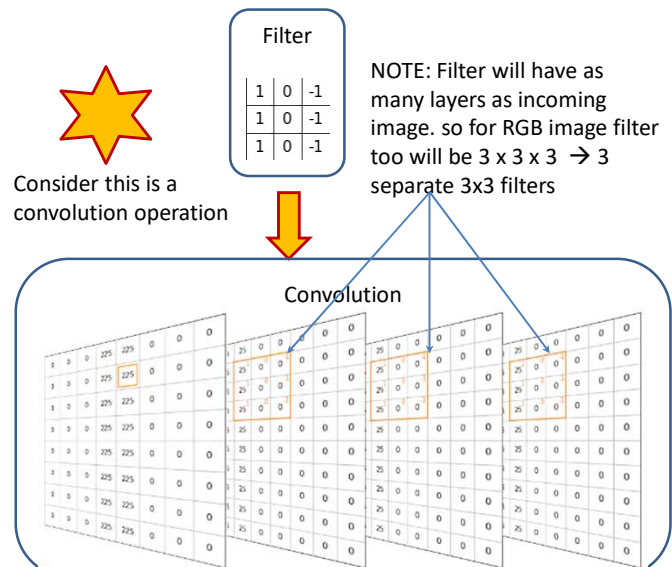

- ❑ Apply filters on the image!

7/12/2023

pra-sâmi

Convolution on 3D images (RGB)

34

Image

If you are looking for edge in one channel only, make rest of them as zeros.

Consider this is a convolution operation

Filter

1	0	-1
1	0	-1
1	0	-1

NOTE: Filter will have as many layers as incoming image. so for RGB image filter too will be 3 x 3 x 3 → 3 separate 3x3 filters

Convolution

7/12/2023

pra-sâmi

17

Convolution on 3D images (RGB)

35

❑ First convolution

25^1	25^0	25^{-1}	25	25	0	0	0	0	0
25^1	25^0	25^{-1}	25	25	0	0	0	0	0
25^1	25^0	25^{-1}	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0

❑ Layer R
- ❖ = 25*1 + 25*1 + 25*1 + 0 + 0 + 0 − 1*25 − 1*25 − 1*25
- ❖ = 0

❑ Layer G
- ❖ = 25*1 + 25*1 + 25*1 + 0 + 0 + 0 − 1*25 − 1*25 − 1*25
- ❖ = 0

❑ Layer B
- ❖ = 25*1 + 25*1 + 25*1 + 0 + 0 + 0 − 1*25 − 1*25 − 1*25
- ❖ = 0

❑ Total = 0 + 0 + 0 = 0

7/12/2023

pra-sâmi

Convolution on 3D images (RGB)

36

❑ Second convolution
- ❖ It will be identical to First

25	25^1	25^0	25^{-1}	25	0	0	0	0	0
25	25^1	25^0	25^{-1}	25	0	0	0	0	0
25	25^1	25^0	25^{-1}	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0

❑ Layer R
- ❖ = 25 + 25 + 25 + 0 + 0 + 0 − 25 − 25 − 25
- ❖ = 0

❑ Layer G
- ❖ = 25 + 25 + 25 + 0 + 0 + 0 − 25 − 25 − 25
- ❖ = 0

❑ Layer B
- ❖ = 25 + 25 + 25 + 0 + 0 + 0 − 25 − 25 − 25
- ❖ = 0

❑ Total = 0 + 0 + 0 = 0

7/12/2023

pra-sâmi

Convolution on 3D images (RGB)

37

❑ What happens 4th step

25	25	25	25^1	25^0	0^{-1}	0	0	0	0
25	25	25	25^1	25^0	0^{-1}	0	0	0	0
25	25	25	25^1	25^0	0^{-1}	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0

❑ Layer R
- ❖ = 25 + 25 + 25 + 0 + 0 + 0 − 0 − 0 − 0
- ❖ = 75

❑ Layer G
- ❖ = 25 + 25 + 25 + 0 + 0 + 0 − 0 − 0 − 0
- ❖ = 75

❑ Layer B
- ❖ = 25 + 25 + 25 + 0 + 0 + 0 − 0 − 0 − 0
- ❖ = 75

❑ Total = 75 + 75 + 75 = 225

pra-sâmi

Convolution on 3D images (RGB)

38

❑ And for 5th Step

25	25	25	25	25^1	0^0	0^{-1}	0	0	0
25	25	25	25	25^1	0^0	0^{-1}	0	0	0
25	25	25	25	25^1	0^0	0^{-1}	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0

❑ Layer R
- ❖ = 25 + 25 + 25 + 0 + 0 + 0 − 0 − 0 − 0
- ❖ = 75

❑ Layer G
- ❖ = 25 + 25 + 25 + 0 + 0 + 0 − 0 − 0 − 0
- ❖ = 75

❑ Layer B
- ❖ = 25 + 25 + 25 + 0 + 0 + 0 − 0 − 0 − 0
- ❖ = 75

❑ Total = 75 + 75 + 75 = 225

pra-sâmi

Convolution on 3D images (RGB)

39

❑ 6th Step onwards again all values are 0

25	25	25	25	25	0^1	0^0	0^{-1}	0	0
25	25	25	25	25	0^1	0^0	0^{-1}	0	0
25	25	25	25	25	0^1	0^0	0^{-1}	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0

❑ Layer R
- ❖ $= 0 + 0 + 0 + 0 + 0 + 0 - 0 - 0 - 0$
- ❖ $= 0$

❑ Layer G
- ❖ $= 0 + 0 + 0 + 0 + 0 + 0 - 0 - 0 - 0$
- ❖ $= 0$

❑ Layer B
- ❖ $= 0 + 0 + 0 + 0 + 0 + 0 - 0 - 0 - 0$
- ❖ $= 0$

❑ Total $= 0 + 0 + 0 = 0$

7/12/2023

pra-sâmi

Convolution

40

Image

Filter

1	0	-1
1	0	-1
1	0	-1

❑ $25 * 1 + 25 * 1 + 25 * 1$
$+ 0 * 0 + 0 * 0 + 0 * 0$
$+ 0 *(-1) + 0*(-1) + 0 *(-1)$
$= 75$

Convolution

75   75   75

7/12/2023

pra-sami

Convolution

Image

25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0

Filter

1	0	-1
1	0	-1
1	0	-1

❑ Every filter we deploy results in 2D matrix

Convolution

Convolution

❑ How many steps filter can take before it goes out of image?

❑ $10 - 3 + 1 = 8$ in either direction…

25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0

Convolution

43

□ $10 - 3 + 1 = 8 \times 8$

25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0

□ Output will be a 2D Matrix

□ Assuming it moves by one step

□ Given that size of the image is 10 and size of the filter is 3

□ Taking : 10- 3 +1 = 8 steps

□ Output image will have 8 cells

□ Hence, Output image size
  $= \{ nH_{in} - nF + 1 \} x \{ nW_{in} - nF + 1 \}$

pra-sâmi

Has it detected the edge?

44
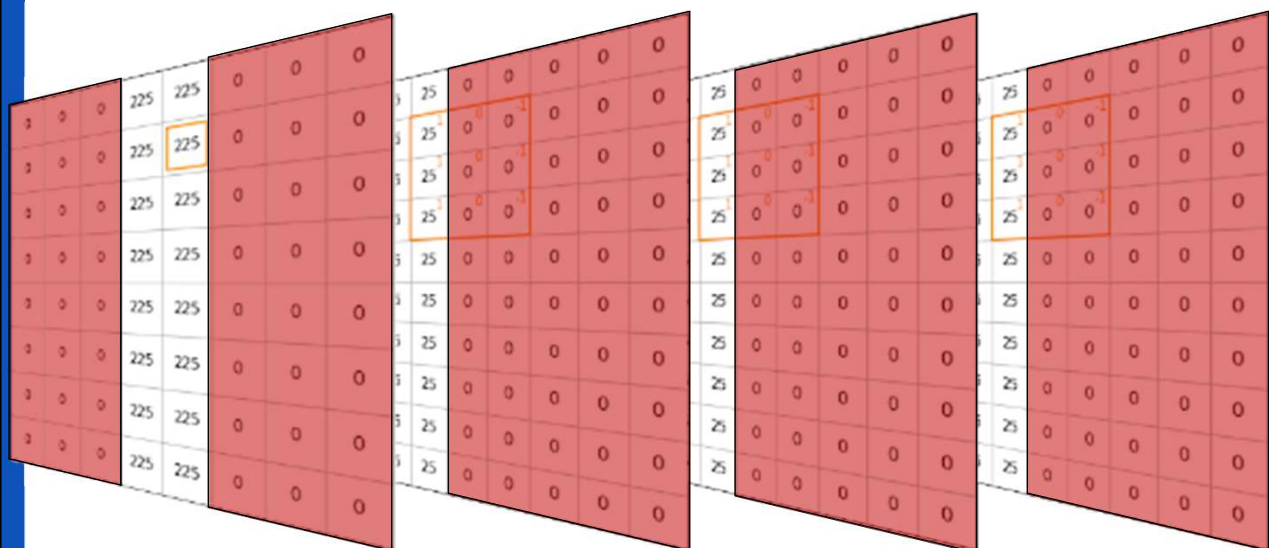
pra-sâmi

45

What if we move two steps at a time

pra-sâmi

46

Stride

- ❑ Steps are called stride

- ❑ If we move 2 steps at a time, we can move 4 steps only

- ❑ In other word with stride of 2
  - ❖ Given that size of the image is 10 and size of the filter is 3

- ❑ Output image size will be : $(10 - 3)/2 + 1 = 4.5$ or 4

- ❑ For fractions pick lower integer

- ❑ Over flow not permitted

pra-sâmi

Stride

47

□ $(10 - 3)/2 + 1 = 4 \times 4$

□ Steps are called stride

□ Hence, Output image size =
$$\{ (nH_{in} - nF) / stride + 1 \}$$
$$x$$
$$\{ (nW_{in} - nF) / stride + 1\}$$

□ If we apply multiple filters ➔ this layer will have 3D matrix.
  ❖ Each layer corresponding to one filter.

pra-sâmi

Convolution

48

□ Apply filters and stack filtered layers together to make a 3D matrix

□ Hence from 3 layer RGB, we can construct as many layers as number of filters applied…

□ Move "stride" steps, generally one or two
  ❖ one in most cases…

□ Strongly advisable to keep filters as odd shape (3,3) or (5,5)

□ Two strong reason..

□ We do not want asymmetric padding
  ❖ Not good for learning features
  ❖ It's better to have central point of the filter
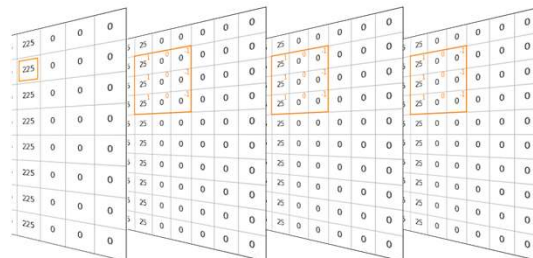
pra-sâmi

Another image

❑ This image has a few distinct edges

Red Green Blue

pra-sâmi

Another Convolution

Image

Filter

1	0	-1
1	0	-1
1	0	-1

Convolution

pra-sâmi

Another Convolution

51

- Layer R =   3 +   3 + 102 - 98 - 206 – 252 = -448
- Layer G = 51 + 51 + 169 - 163 - 196 - 3 = -91
- Layer B = 252 + 252 + 203 - 207 - 10 - 18 = 472
- Total = -448 + -91 + 472 = -67

pra-sâmi

Another Convolution

52

- Incoming Image shape = (10,10,3)
  - nH_{in} = 10 ; nW_{in} = 10, nC = 3
- Filter shape = (3, 3, 3)
  - nF = 3 ; nF = 3, nC = 3
- Stride = 1
- Hence the size will be 8 x 8 after convolution

pra-sâmi

Another Convolution

- Single filter convolution:
- Layer R = 3 + 3 + 102 - 98 - 206 – 252 = -448
- Layer G = 51 + 51 + 169 - 163 - 196 - 3 = -91
- Layer B = 252 + 252 + 203 - 207 - 10 - 18 = 472
- Total = -448 + -91 + 472 = -67

- Incoming Image shape = (10,10,3) ➔ nHin = 10 ; nWin = 10, nC = 3
- Filter shape = (3, 3, 3) ➔ nF = 3 ; nF = 3, nC = 3
- Stride = 1
- Hence the size will be 8 x 8 after convolution

In convolution,:
- With every convolution image is shrinking
- Corners and edges of image are used less frequently than the middle

7/12/2023

pra-sâmi

Other filters

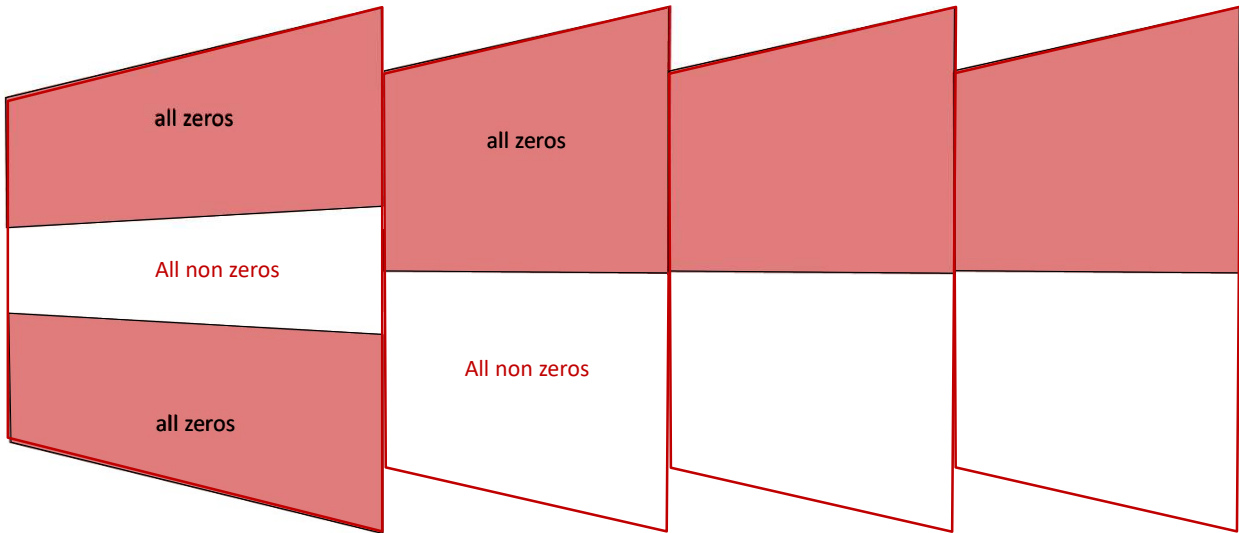- We have seen vertical filter… How about horizontal Filter….

- No surprises there….

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

- The math will be exactly the same and we would get horizontal edge

7/12/2023

pra-sâmi

Horizontal Edge…

55

all zeros

All non zeros

all zeros

all zeros

All non zeros

pra-sâmi

Other filters

56

□ Sobel Filter…
$$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$$

□ There was a lot of debate on filters….

□ Researchers kept trying various numbers…

□ Why not learn these parameters…
$$\begin{matrix} w_1 & w_4 & w_7 \\ w_2 & w_5 & w_8 \\ w_3 & w_6 & w_9 \end{matrix}$$

pra-sâmi

Shade Reversal

57

❑ So far we have seen lighter to darker shade filters…

❑ What happens if we move from darker shade to lighter

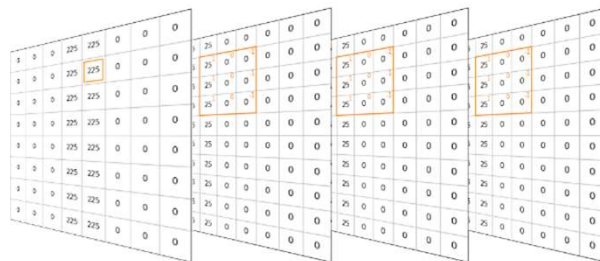pra-sâmi

Shade Reversal

58

❑ So far we have seen lighter to darker shade filters…

❑ What happens if we move from darker shade to lighter

❑ We will again get the edge only it will be negative this time…

pra-sâmi

Convolving a Volume

59

- So far we have shown that same filter is applied to all layers

- In theory, it is possible to have a filter which is looking for edges in red channel alone...

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
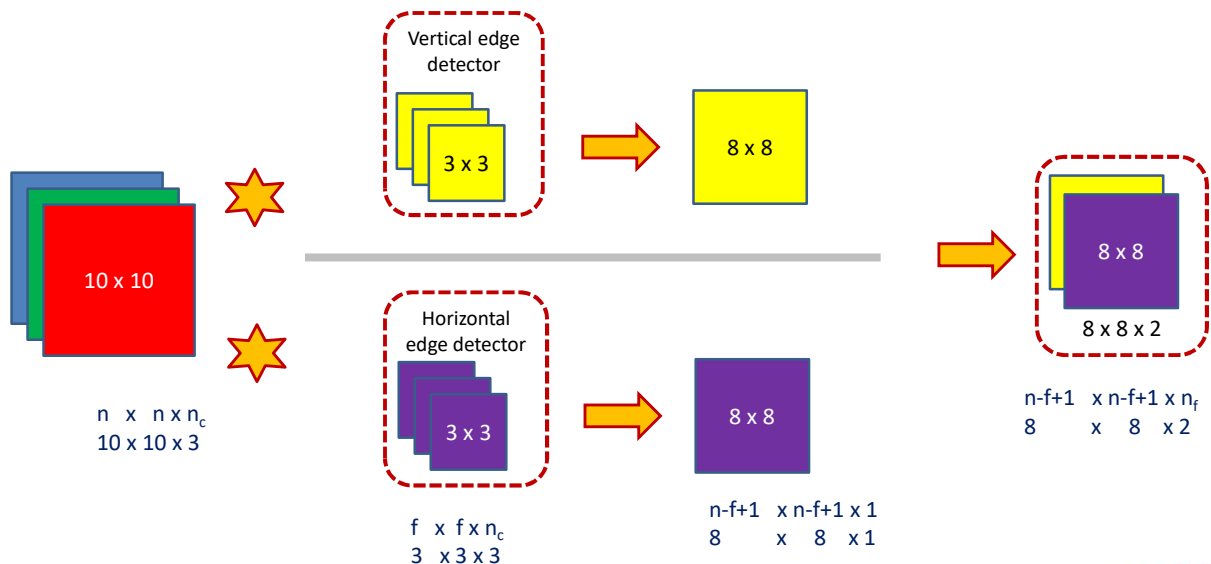
- So far we have been showing that 3D image converts to 2D image when we apply filter

- By applying a number of filters to detect different edges, we can have 3d Convolutional Volumes.

7/12/2023

pra-sâmi

Multiple Filters

60

Vertical edge detector — 3 x 3 → 8 x 8

Horizontal edge detector — 3 x 3 → 8 x 8

10 x 10

$n \times n \times n_c$
10 x 10 x 3

$f \times f \times n_c$
3 x 3 x 3

$n\text{-}f\text{+}1 \times n\text{-}f\text{+}1 \times 1$
8 $\times$ 8 $\times 1$

8 x 8 x 2

$n\text{-}f\text{+}1 \times n\text{-}f\text{+}1 \times n_f$
8 $\times$ 8 $\times 2$

7/12/2023

pra-sâmi

30

Two Issues with the convolution…

61

- ❑ With every convolution image is shrinking
  - ❖ Knowing that 100s of layer is not uncommon in the architecture
  - ❖ Image can soon become 1px X 1px

- ❑ Corners and edges of image are used less frequently than the middle

pra-sâmi

62

What if we zero pad the image all around… will it help?

pra-sâmi

Zero Padding

❑ If we do not want our images to shrink during convolution and to take maximum advantage of edges,

❑ Pad the image by adding zeros all around.

❑ Now we will be using corner cell 4 times at least...

❑ Edge cells will be used more....

7/12/2023

Convolution after Padding

❑ The respective cell values remain same

❑ and we are back with original image size

7/12/2023

pra-sâmi

Convolution after Padding

65

- Incoming image shape = (10, 10, 3)
  - i.e $nH_{in} = 10$; $nW_{in} = 10$; $nC = 3$

- Padding p = 1

- Padded image shape = (12, 12, 3)
  - i.e $nH_{in} = 12$; $nW_{in} = 12$; $nC = 3$

- Filter shape = (3, 3, 3)
  - i.e. nF = 3 ; nF = 3, nC = 3

- Assuming we move "stride" steps at any time
  - i.e. stride = 1

- Output image size:

$$= \{ \frac{nH_{in} - nF + 2*p}{stride} + 1 \}$$

x

$$\{ (\frac{nW_{in} - nF + 2*p}{stride} + 1\}$$

- Image Size $= \{ \frac{10 - 3 + 2*1}{1} + 1 \}$

x

$$\{\frac{10 - 3 + 2*1}{1} + 1 \}$$

$$= 10 \times 10$$

We are back to original size...

pra-sâmi

How much to pad???

66

- There are two recommended mechanism

- **Valid** : output is calculated as
$$\{ \frac{nH_{in} - nF + 2*p}{stride} + 1 \} \times \{ (\frac{nW_{in} - nF + 2*p}{stride} + 1\}$$

- So for 10 x 10 image a 5 x 5 filter with 1 px padding, image size will be 8 x 8

- **Same** : do the padding in such a way so that resultant image is of same size
$$\{ \frac{nH_{in} - nF + 2*p}{stride} + 1 \} \times \{ (\frac{nW_{in} - nF + 2*p}{stride} + 1\} = nH_{in} \times nW_{in}$$
  - or p = $(nF - 1)/2$ for stride = 1

pra-sâmi

How much to pad???

67

- With $p = (nF - 1)/2$ for stride $= 1$;

- We want p to be an integer and hence
  - ❖ Need nF to be odd

- For even value of nF we would end up in asymmetric padding.

- Unless we feel one edge of the image is more important than other, there is no need to have asymmetric padding

pra-sâmi

68

Cross-Correlation vs. Convolution

pra-sâmi

Cross-Correlation vs. Convolution

❑ In Signal Theory and Maths

❑ Convolution involves multiplying the filter after mirroring on both axis

❑ i.e. for filter $\begin{bmatrix} 3 & 4 & 5 \\ 1 & 0 & 2 \\ -1 & 9 & 7 \end{bmatrix}$

❑ It will be mirrored along both axis... $\begin{bmatrix} 7 & 9 & -1 \\ 2 & 0 & 1 \\ 5 & 4 & 3 \end{bmatrix}$

❑ Then we do element wise multiplication.

❑ Signal Engineers will agree with me... ☺

❑ Such correlations have properties like associative (a*b)*c = a*(b*c) and all other properties

7/12/2023

pra-sâmi

Cross-Correlation vs. Convolution

❑ So that we are correct semantically...

❑ What we are doing is called Cross-Correlation....

❑ However, Data Scientists across the world have been using filters without reversing it and still call it Convolution...

Now you know... don't write home about it... ☺
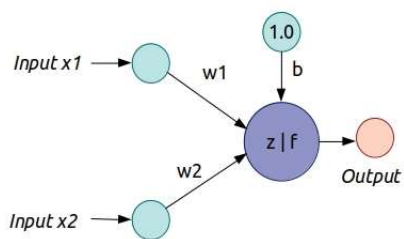
7/12/2023
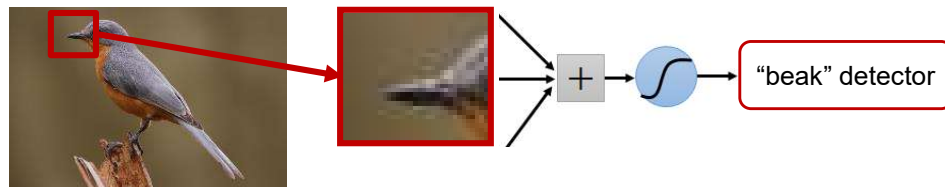
pra-sâmi

71

One layer to Convolutional Net

pra-sâmi

---

72 **One layer of Conv Net**
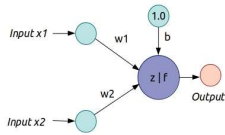
Input x1 → (w1)

1.0

b

z | f → Output

w2

Input x2 →

$Z_1 = a_0 \cdot W_1 + b_1$

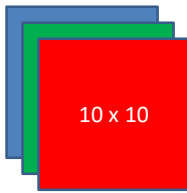$a_1 = Relu (Z_1)$

+ → "beak" detector

pra-sâmi

One Layer of Conv Net

73

$$Z_1 = a_0 . W_1 + b_1$$

$$a_1 = Relu (Z_1)$$

10 x 10

7/12/2023

One Layer of Conv Net

74

$$Z_1 = a_0 . W_1 + b_1$$

$$a_1 = Relu (Z_1)$$

3 x 3

8 x 8

10 x 10

7/12/2023

pra-sami

One Layer of Conv Net

75

$$Z_1 = a_0 . W_1 + b_1$$
$$a_1 = Relu(Z_1)$$

Input x1 → w1
1.0
b
w2
z | f
Output
Input x2 →

W_1

3 x 3

→ Relu

8 x 8 b_1

Z_1

10 x 10

a_0

7/12/2023

pra-sami

One Layer of Conv Net

76

$$Z_1 = a_0 . W_1 + b_1$$
$$a_1 = Relu(Z_1)$$

Input x1 → w1
1.0
b
w2
z | f
Output
Input x2 →

$W^{[l]}$ (3 x 3 x 3 x 2)

$w_1^{[l]}$

3 x 3

→ Relu

8 x 8 b_1

Z_1 (8 x 8)

10 x 10

a_0 (10 x 10 x 3)

$w_2^{[l]}$

3 x 3

→ Relu

8 x 8 b_2

Z_2

8 x 8

a_1 (8 x 8 x 2)

7/12/2023

pra-sâmi

38

One Layer of Conv Net

77

$Z_1 = a_0 . W_1 + b_1$

$a_1 = Relu(Z_1)$

$W_1 (3 \times 3 \times 3)$

3 x 3

Relu

8 x 8 ← b_1

$Z_1 (8 \times 8)$

If we apply 10 filters, size of a_1 will be 8 x 8 x 10

Input x1 → w1 b 1.0

Σ|F Output

Input x2 → w2

10 x 10

$a_0 (10 \times 10 \times 3)$

W_2

3 x 3

Relu

8 x 8 ← b_2

Z_2

8 x 8

$a_1 (8 \times 8 \times 2)$

7/12/2023

pra-sâmi

How Many Parameters…

78
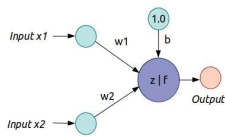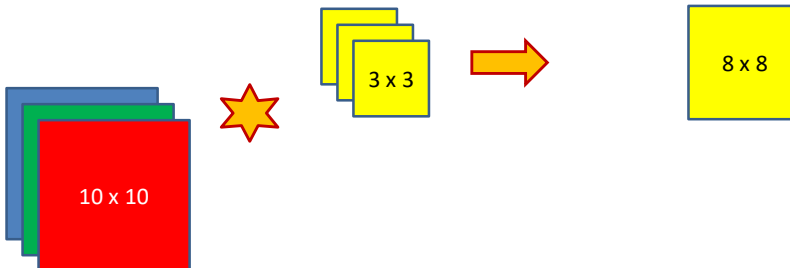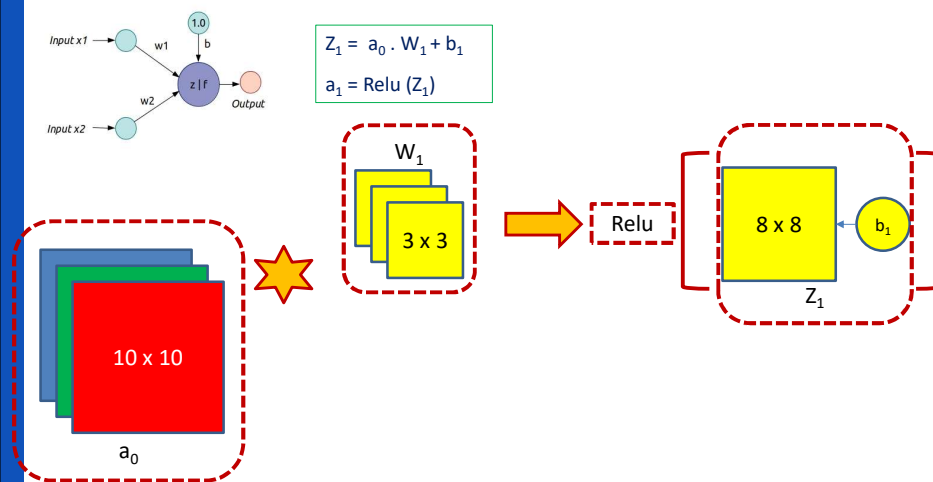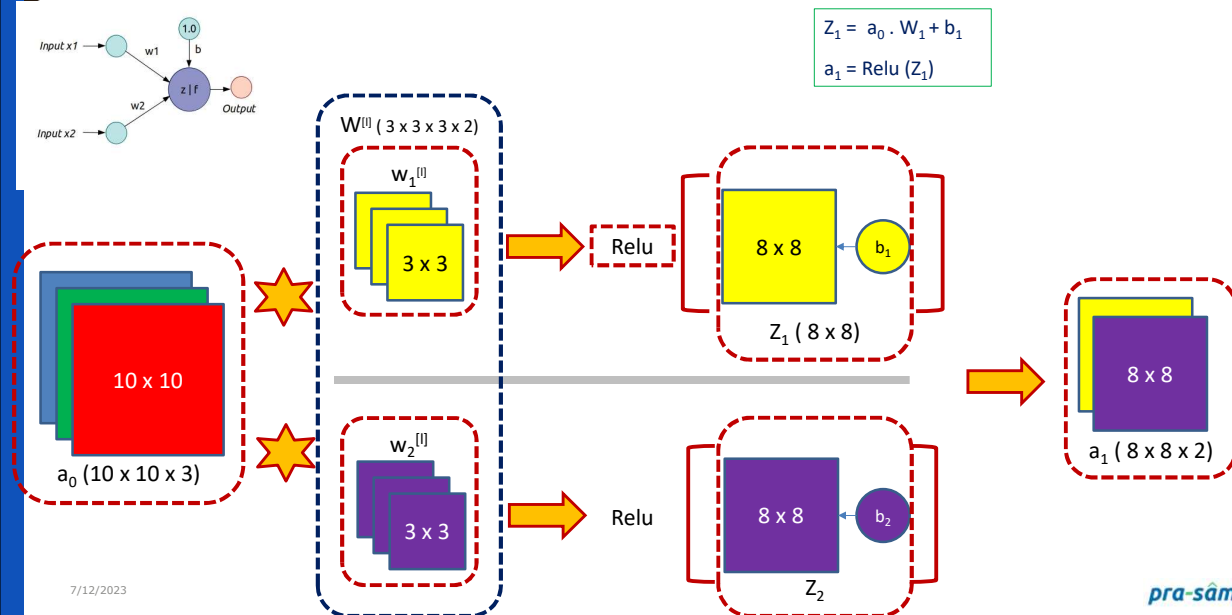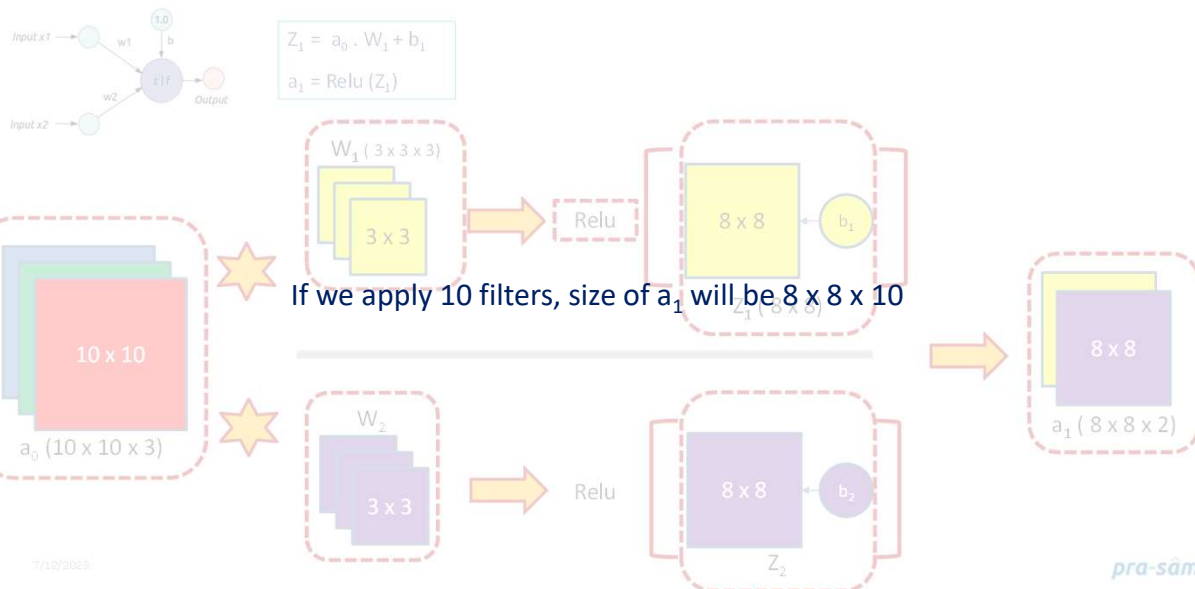
❑ Imagine using 10 filters in a layer, how many parameters in the layer

❑ Each filter is 3 x 3 x 3 + a bias = 28 parameters

❑ For 10 filters = total 280 parameters

❑ Hence, irrespective of size of input image /activation, we still have 280 filters learning what we need to know…. Yay!!!!!

❑ Helps in prevention of over-fitting

1000 x 1000

$a_0 (1000 \times 1000 \times 3)$

$W_1 (3 \times 3 \times 3)$

3 x 3

b_1
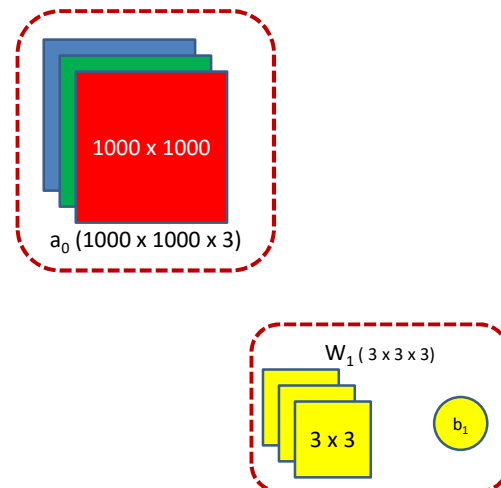
7/12/2023

pra-sâmi

Lets Look at the Dimensions…

79

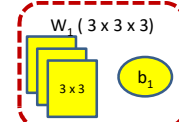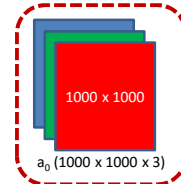|			
$f^{[l]}$:	Filter Size	Input:	$n^{[l-1]}_H \times n^{[l-1]}_W \times n^{[l-1]}_c$
$p^{[l]}$:	Padding size	Output:	$n^{[l]}_H \times n^{[l]}_W \times n^{[l]}_c$
$s^{[l]}$:	Stride	$n^{[l]}_H$:	$(n^{[l-1]}_H + 2\,p^{[l]} - f^{[l]}) / s^{[l]} + 1$
$n^{[l]}_c$:	Number of filters	$n^{[l]}_w$:	$(n^{[l-1]}_w + 2\,p^{[l]} - f^{[l]}) / s^{[l]} + 1$
Filter size:	$f^{[l]} \times f^{[l]} \times n^{[l-1]}_c$	Activations $a^{[l]}$:	$n^{[l]}_H \times n^{[l]}_W \times n^{[l]}_c$
Weights (all filters):	$f^{[l]} \times f^{[l]} \times n^{[l-1]}_c \times n^{[l]}_c$	Biases:	$n^{[l]}_c$

Weights are tensors of rank 4

Activation for all m training examples m
$m \times n^{[l]}_H \times n^{[l]}_W \times n^{[l]}_c$
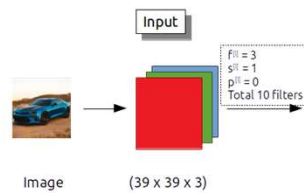
Don't be surprised if you see Filter number first

1000 x 1000

a_0 (1000 x 1000 x 3)

W_1 (3 x 3 x 3)

3 x 3

b_1

7/12/2023

pra-sâmi

A Simple CNN with Conv Layers

80

Input

$f^{[l]} = 3$
$s^{[l]} = 1$
$p^{[l]} = 0$
Total 10 filters

Image

(39 x 39 x 3)
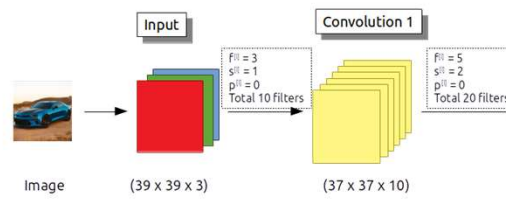
7/12/2023

pra-sâmi

A Simple CNN with Conv Layers
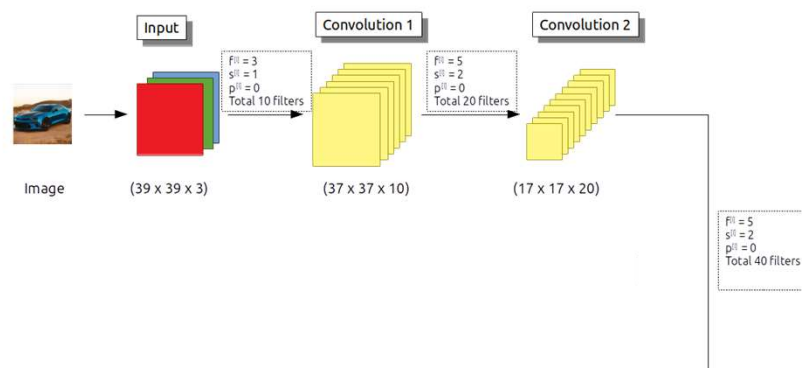
A Simple CNN with Conv Layers

A Simple CNN with Conv Layers
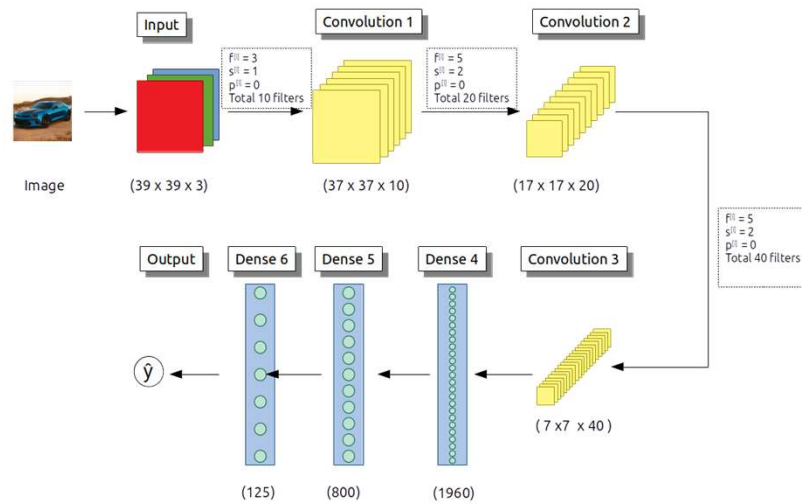
pra-sâmi

Pooling…
What is most significant in this area…

pra-sâmi

Pooling

85

- Two methods of Pooling – 'Max' and 'Average'
- Max : maximum value of the from the cells being filtered
- Average : Average Values from the cells

- Mode = 'max'; pool = 2; stride = 2

7/12/2023

pra-sâmi

Other image

86

- Image Size = 10,10,3; filter Size = 3,3,3; stride = 1
- After Convolution = 8, 8, 1

- Input size = 8,8,1; pool = 2; Stride = 2
- After pooling = 4,4,1

7/12/2023

pra-sâmi

43

Pooling

87

- Image Size = 10,10,3; filter Size = 3,3,3; stride = 1
- After Convolution = 8, 8, 1

-67	23	43	55	-72	-12	-30	81
313	343	0	0	0	0	-362	-291
559	390	0	0	0	0	-423	-601
498	390	0	0	0	0	-423	-633
498	390	0	0	0	0	-423	-633
559	390	0	0	0	0	-423	-601
318	344	0	0	0	0	-367	-296
-62	24	43	55	-72	-12	-35	76

- Input size = 8,8,1; pool = 2; Stride = 2
- After pooling = 4,4,1
- Formula for size are still applicable,
- Its independently done on each channels
- Other option is to use Average instead of Max
  - But not used frequently.

343	55	0	81
559	0	0	-423
559	0	0	-423
344	55	0	76

7/12/2023

pra-sâmi

Pooling

88

- Image Size = 10,10,3; filter Size = 3,3,3; stride = 1
- After Convolution = 8, 8, 1

- Input size = 8,8,1; pool = 2; Stride = 2
- After pooling = 4,4,1
- Formula for size are still applicable,
- Its independently done on each channels
- Other is Average as expected but not used frequently.

Consider that each area represents presence of some feature in the image and high number represents, presence of that feature…
It has three (mode, pool and stride) hyperparameters to tune…
but no parameters to learn…
Gradient descent is not going to do anything here…. ☺

7/12/2023

pra-sâmi

Demo Example – Fashion MNIST

89

pra-sâmi

Congratulations!!!!

90

Now is the time to update your resume... ☺

pra-sâmi

91

pra-sâmi