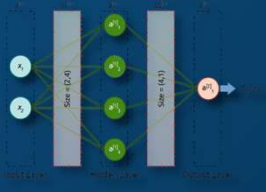
Convolution Neural Networks – CNN
Part I

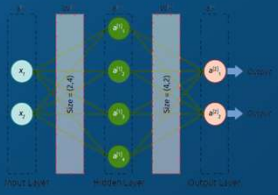
Deep Neural Network

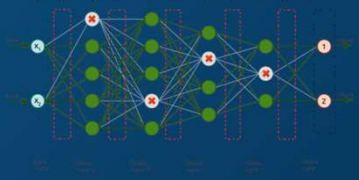
Session 21

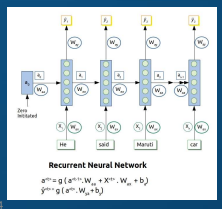
Pramod Sharma
pramod.sharma@prasami.com

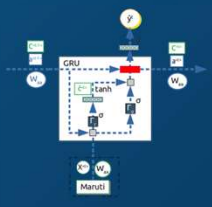
2 Story So Far…

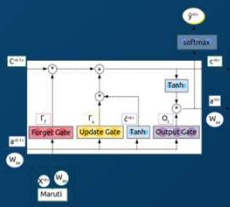


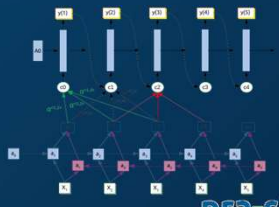












pra-sami

3

Agenda

- Computer Vision
- Convolution
- Padding
- Pooling
- Frameworks

pra-sami

4

Acknowledgement…

Geoffrey Everest Hinton CC FRS FRSC

- 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-sami

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…

pra-sami

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

pra-sami

9 Computer Vision

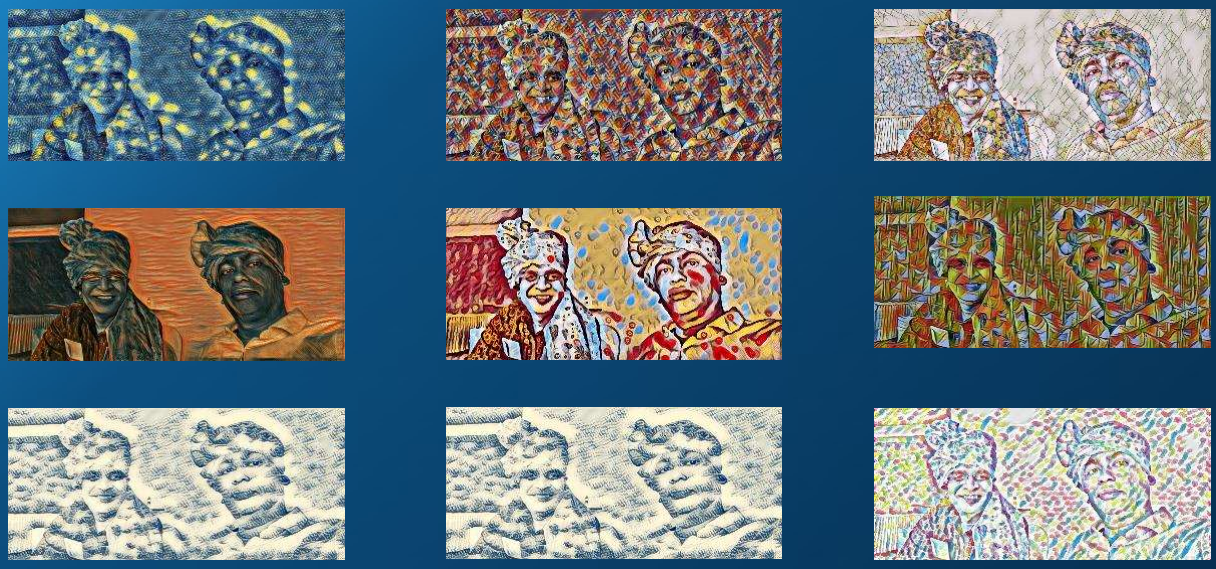
- Self driving car
- Fully automated warehouse and ports
  - https://youtu.be/REV8IkY52iY
- Image search services,
- Unlock phone
- Provide access to secure area
  - Open your house
  - Enter office without your access card
- Object identification Apps
  - Garment
  - Food,
  - Nature
- Natural style transfer
- Automatic video classification systems

pra-sâmi

10 Style Transfer

pra-sâmi

11 Style transfer



pra-sami

12 Style transfer

pra-sami

13 Computer Vision

- 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)

pra-sami

14 Computer Vision

- 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)

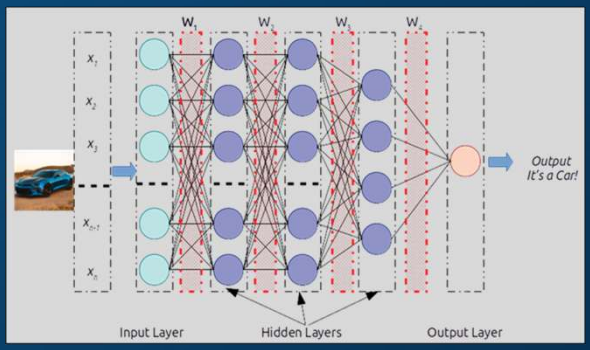
"Imaging a medical database in which the age of the patient sometimes hops to the input dimension which normally codes for weight!" - Geoff Hinton

pra-sami

15

Why?

- 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 (W1) will be 67,500 x 1000
- Do your math…… that size is huge

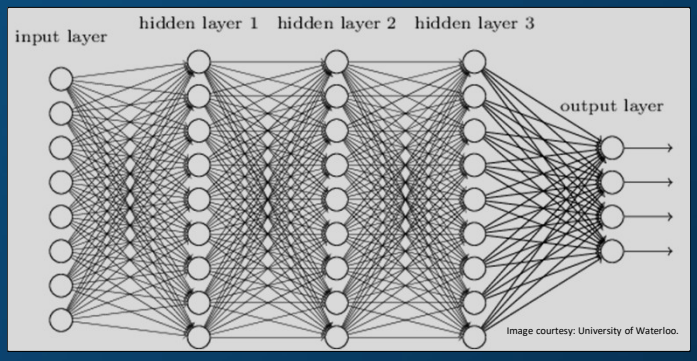


pra-sami

16

Smaller Network: CNN

- 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?



pra-sami

17

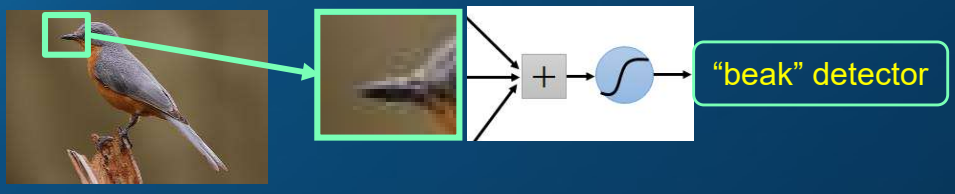
Images are high-dimensional vectors. It would take a huge amount of parameters to characterize the network.

pra-sami

18

Learning an image…

- Some patterns are much smaller than the whole image
- Can represent a small region with fewer parameters

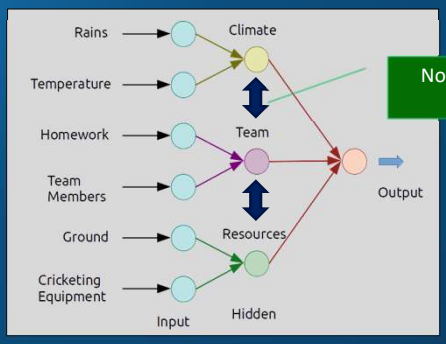


pra-sami

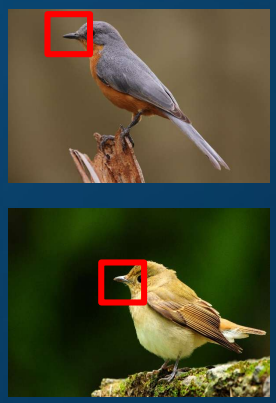
19

Learning an image…

- 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"



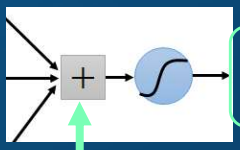
No sideways sharing of information



pra-sami

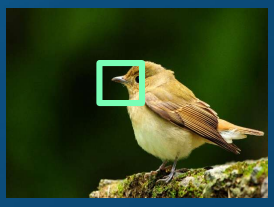
20

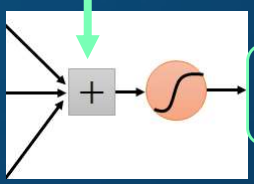
Learning an image…



"upper-left beak" detector

They can be compressed to the same parameters.





"middle beak" detector

pra-sami

21

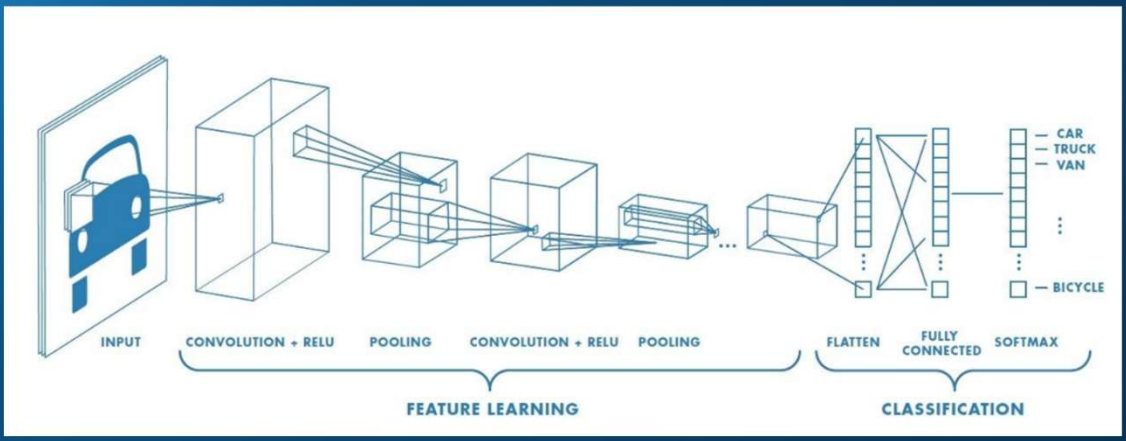
Learning an image…

- 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 set of layers: the convolution layer, pooling layer…

pra-sami

22

Overall Layout



pra-sami

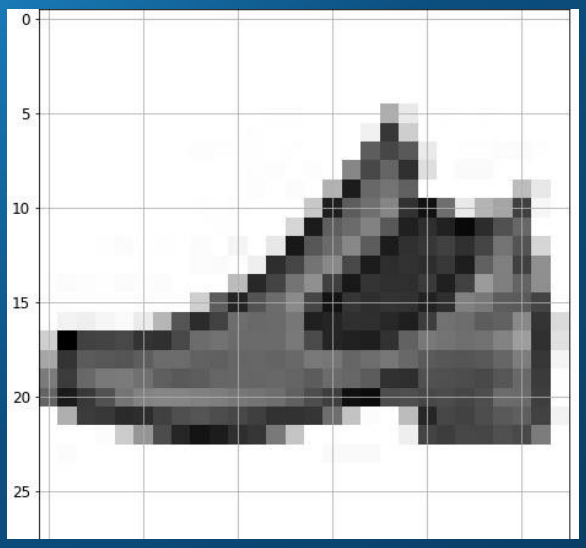
23

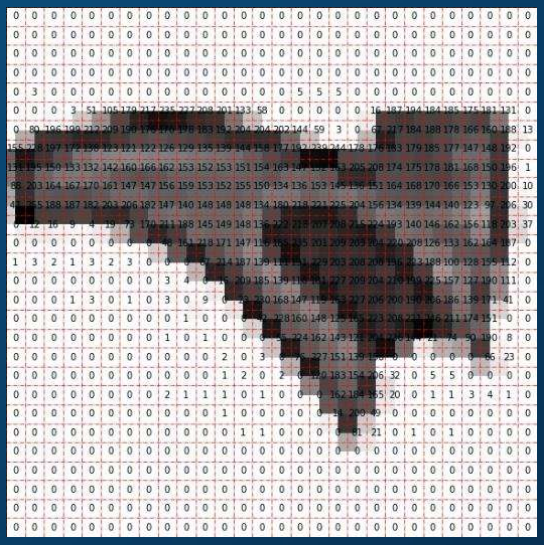
Each column of hidden units looks at a small region of the image, and the weights are shared between all image locations.

pra-sami

24

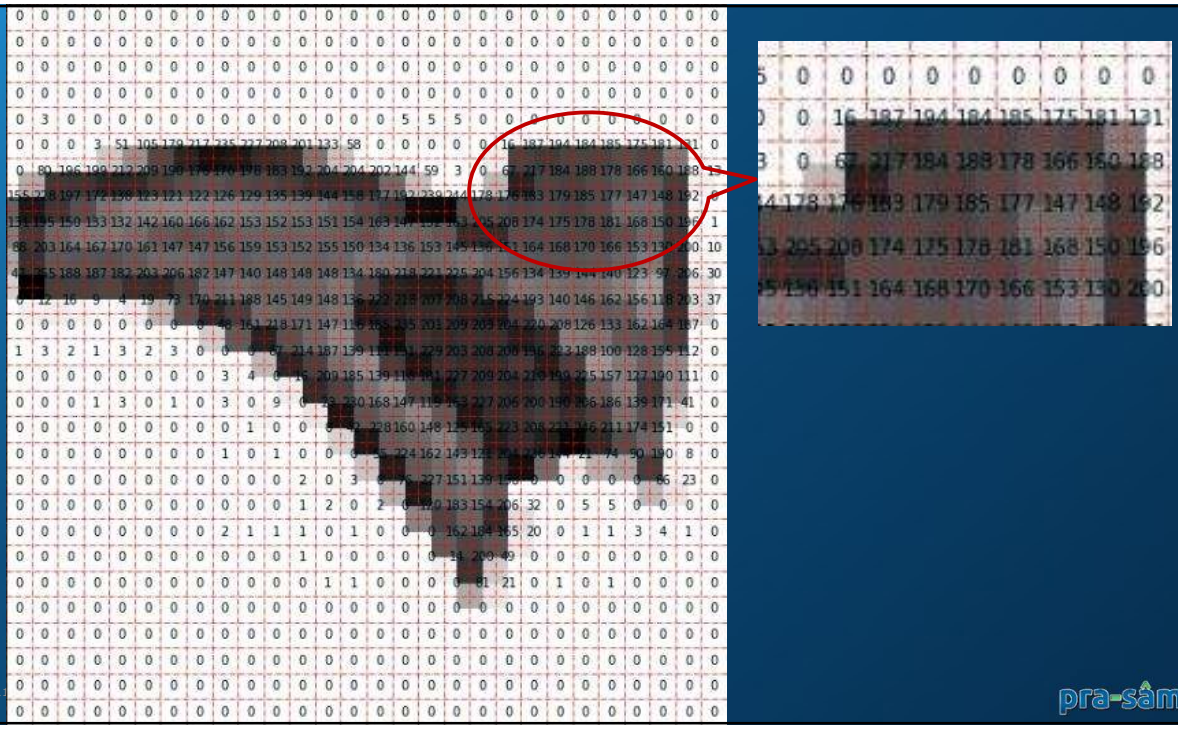
What computer sees…





pra-sami

25

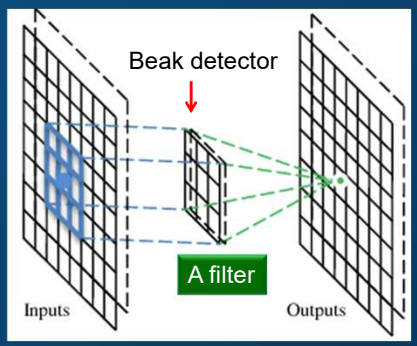


pra-sami

26

A Convolutional Layer

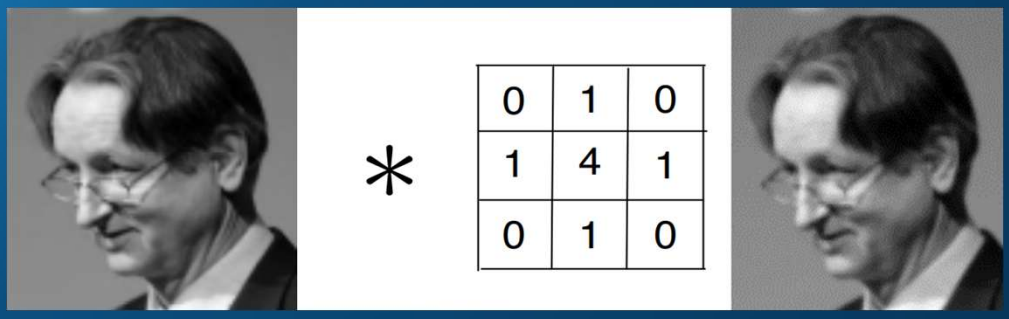
- A CNN is a neural network with some convolutional layers
  - And, of course, a few other layers
- A convolutional layer has a number of filters that does convolutional operation
  - Some of the literature would call it Kernel



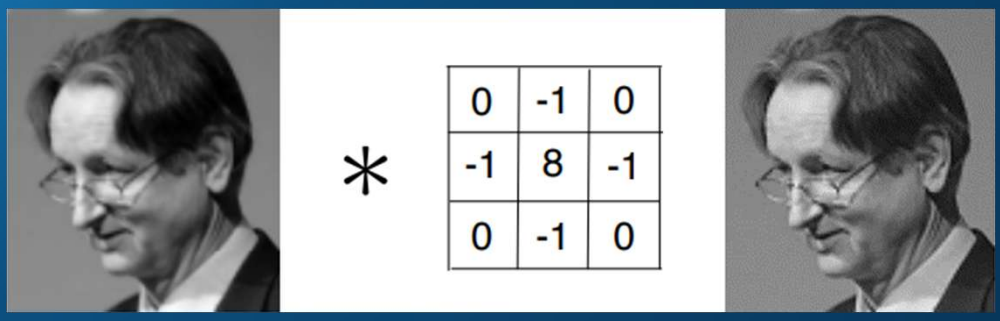
11/30/2024

pra-sami

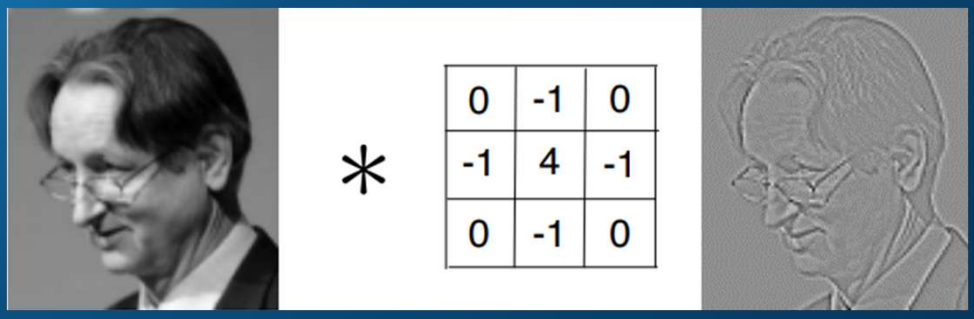
27 What does this Convolution Filter/ Kernel do?



28 What does this Convolution Filter/ Kernel do?

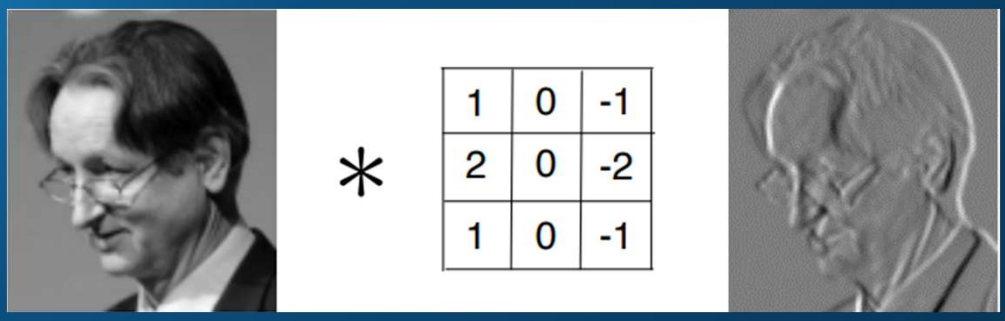


29 What does this Convolution Filter/ Kernel do?



pra-sâmi

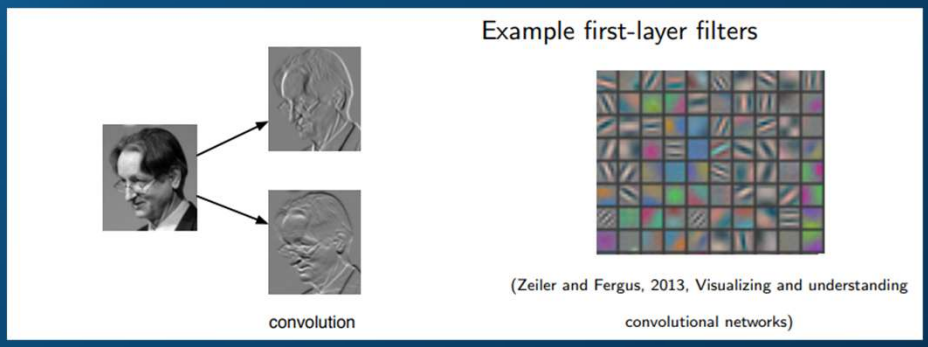
30 What does this Convolution Filter/ Kernel do?



pra-sâmi

31 Convolutional Networks

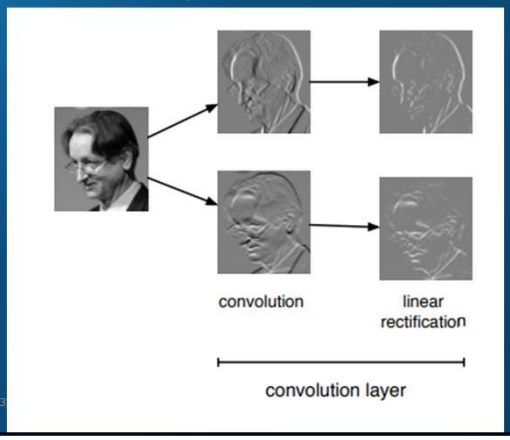
- Two kinds of layers:
  - Detection layers (or convolution layers)
  - Pooling layers
- The convolution layer has a set of filters.
  - Output is a set of feature maps, each one obtained by convolving the image with a filter.



pra-sami

32 Convolutional Networks

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



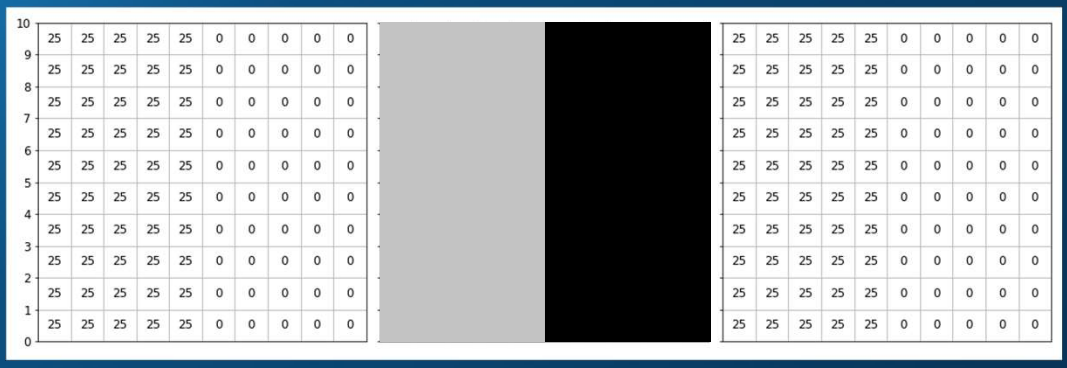
- Convolution is a linear operation
- Therefore, we need a nonlinearity:
  - Otherwise two convolution layers would be no more powerful than one
- Two edges in opposite directions shouldn't cancel
- Non-linearity makes the gradients sparse, which helps optimization

pra-sami

33

Image with a edge

- Convolution is basic building block of image recognition
- Using edge detection as an example in following image…

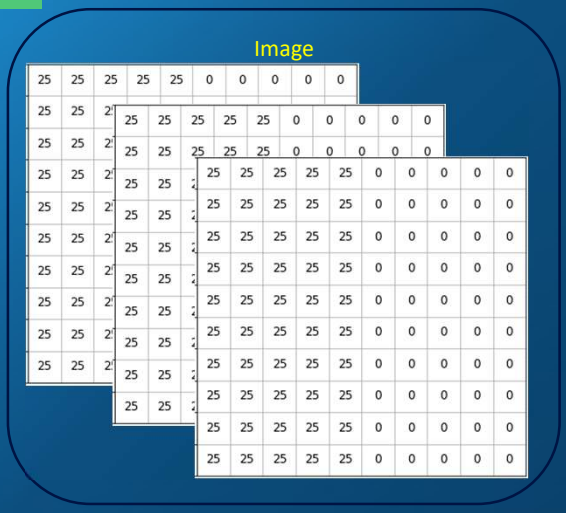


- Apply filters on the image!

pra-sami

34

Convolution on 3D images (RGB)



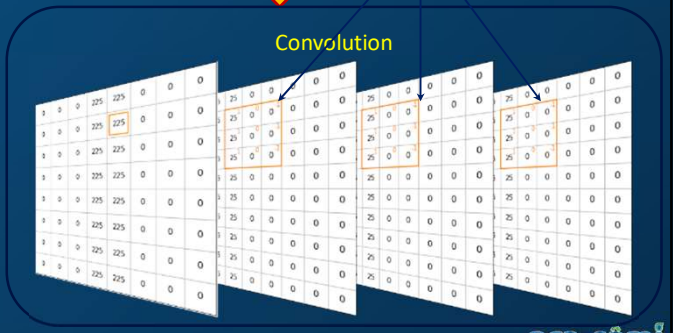
If you are looking for edge in one channel only, make rest of them as zeros.

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



pra-sami

35

Convolution on 3D images (RGB)

- 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

pra-sami

36

Convolution on 3D images (RGB)

- 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

pra-sami

37

Convolution on 3D images (RGB)

- 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-sami

38

Convolution on 3D images (RGB)

- 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-sami

39

Convolution on 3D images (RGB)

- 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

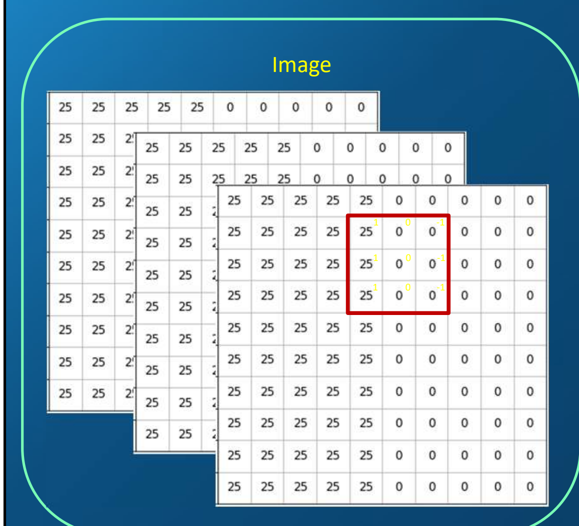
- Layer R
  - = 0 + 0 + 0 + 0 + 0 + 0 – 0 – 0 – 0
  - = 0
- Layer G
  - = 0 + 0 + 0 + 0 + 0 + 0 – 0 – 0 – 0
  - = 0
- Layer B
  - = 0 + 0 + 0 + 0 + 0 + 0 – 0 – 0 – 0
  - = 0
- Total = 0 + 0 + 0 = 0

11/30/2024

pra-sâmi

40

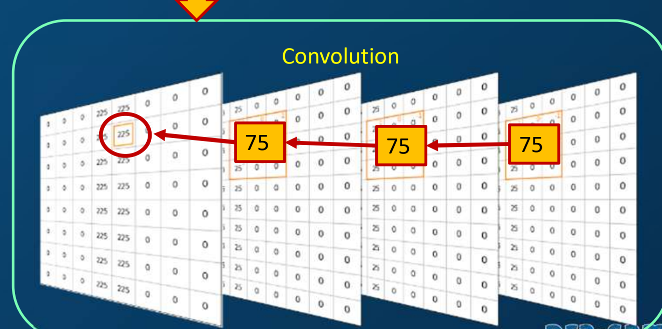
Convolution



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

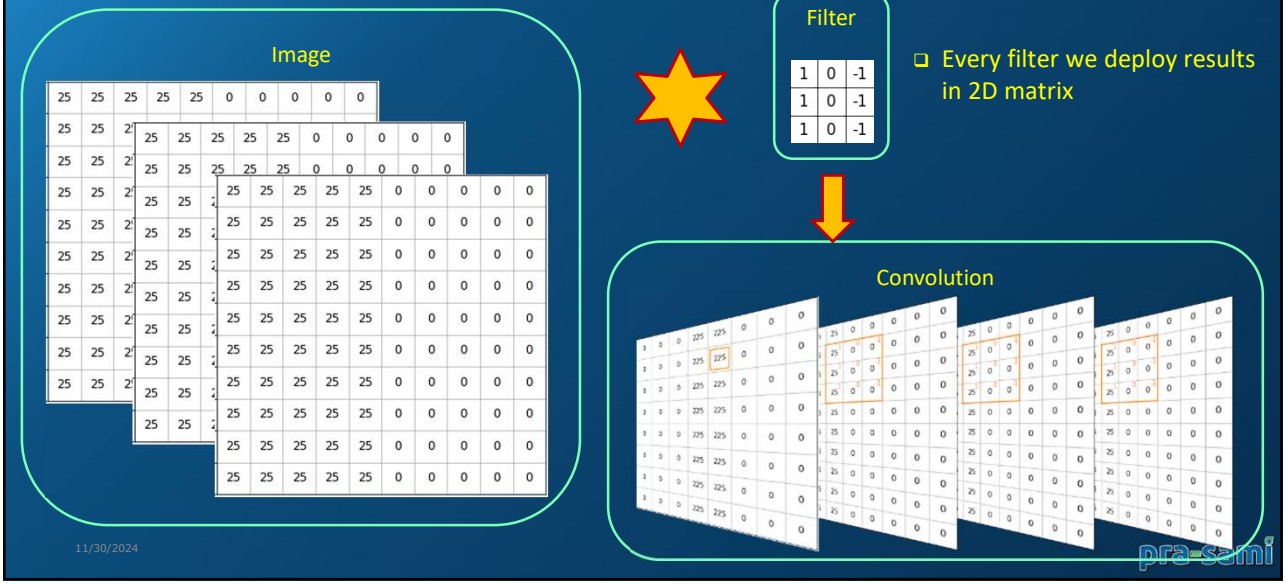


11/30/2024

pra-sâmi

41

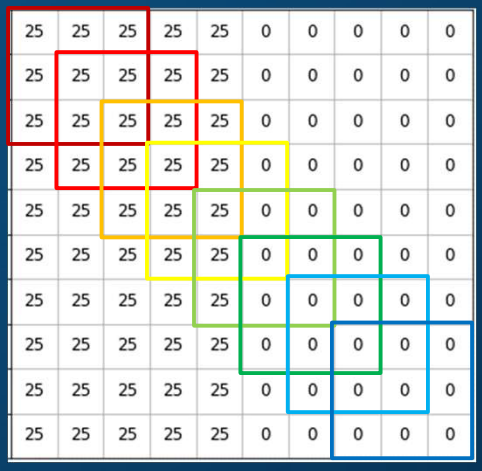
Convolution



42

Convolution

- How many steps filter can take before it goes out of image?
- 10 – 3 + 1 = 8 in either direction…



11/30/2024

pra-sami

43

Convolution

- 10 – 3 + 1 = 8 x 8

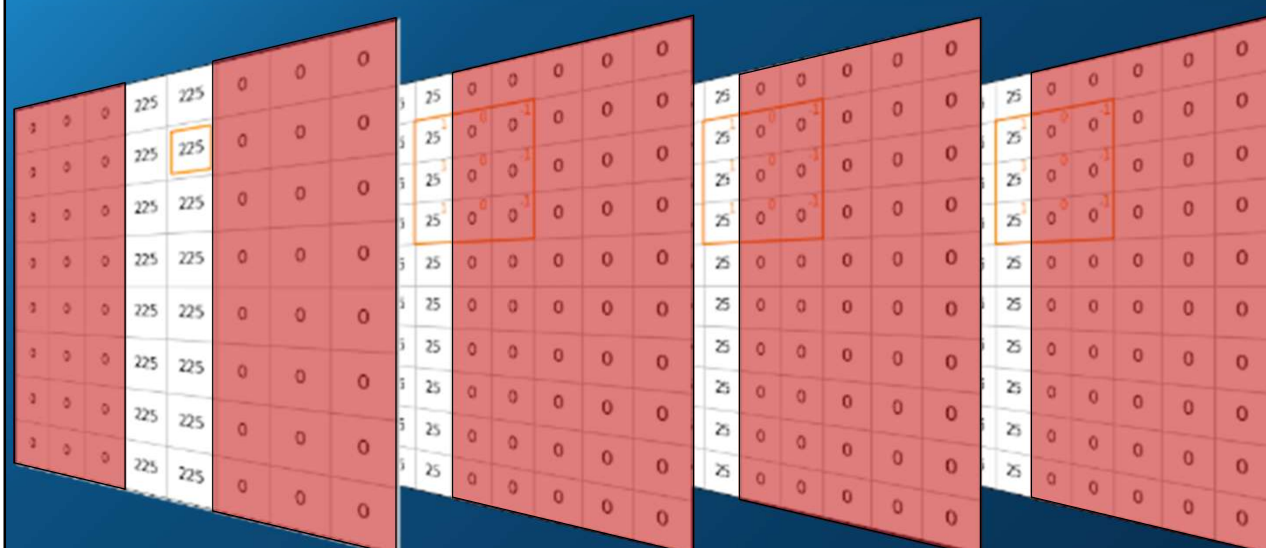
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0
25	25	25	25	25	0	0	0	0	0

- 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
  $= \{ nH_{in} - nF + 1 \} x \{ nW_{in} - nF + 1 \}$

11/30/2024

44

Has it detected the edge?



11/30/2024

45

What if we move two steps at a time

pra-sami

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-sami

47

Stride

- (10 – 3)/2 + 1 = 4 x 4

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

- 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.

11/30/2024

pra-sâmi

48

Convolution

- 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

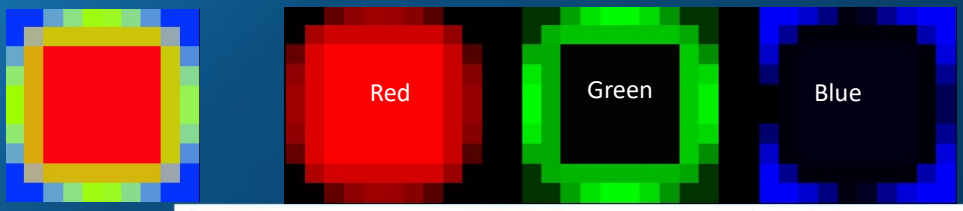
11/30/2024

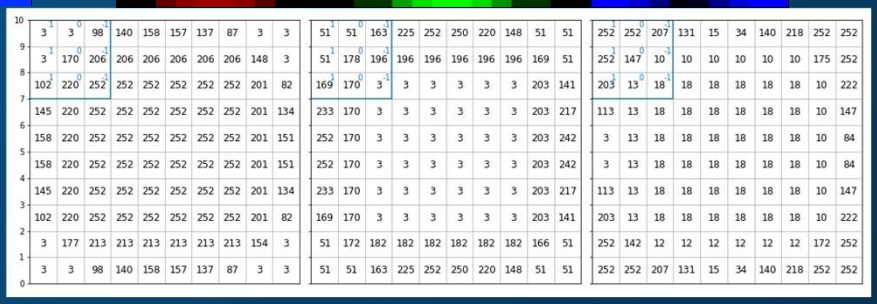
pra-sâmi

49

Another image

- This image has a few distinct edges



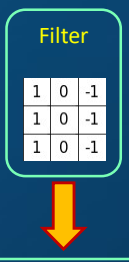


11/30/2024

pra-sami

50

Another Convolution



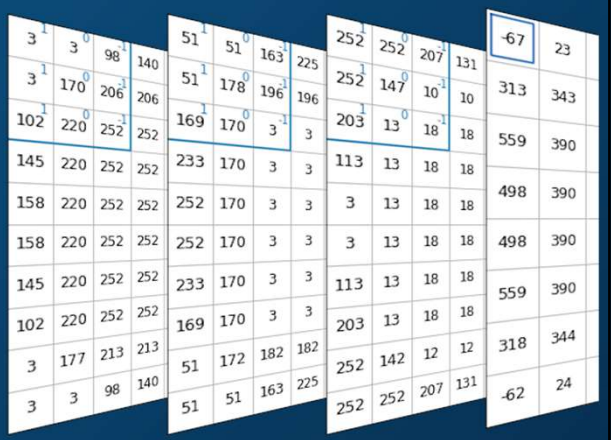
11/30/2024

pra-sami

51

Another 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

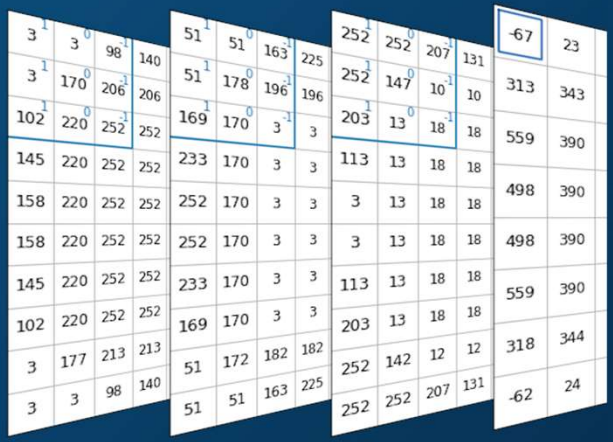


pra-sâmi

52

Another Convolution

- 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

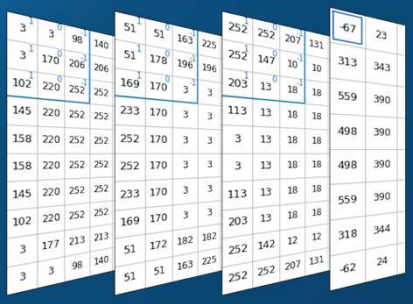


pra-sâmi

53

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

pra-sami

54

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

pra-sami

55 Horizontal Edge…

pra-sami

56 Other filters

- Sobel Filter…

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

- There was a lot of debate on filters….
- Researchers kept trying various numbers…
- Why not learn these parameters…

$$\begin{bmatrix} w_1 & w_4 & w_7 \\ w_2 & w_5 & w_8 \\ w_3 & w_6 & w_9 \end{bmatrix}$$

pra-sami

57

Shade Reversal

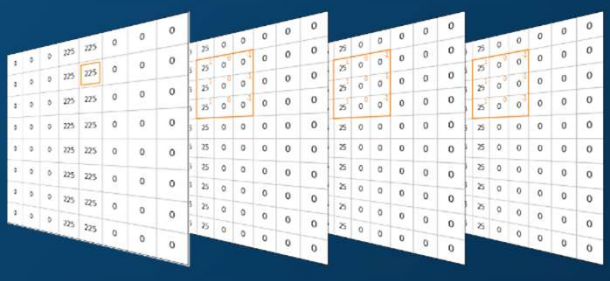
- So far we have seen lighter to darker shade filters…
- What happens if we move from darker shade to lighter

pra-sami

58

Shade Reversal

- 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-sami

59

Convolving a Volume

- 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…

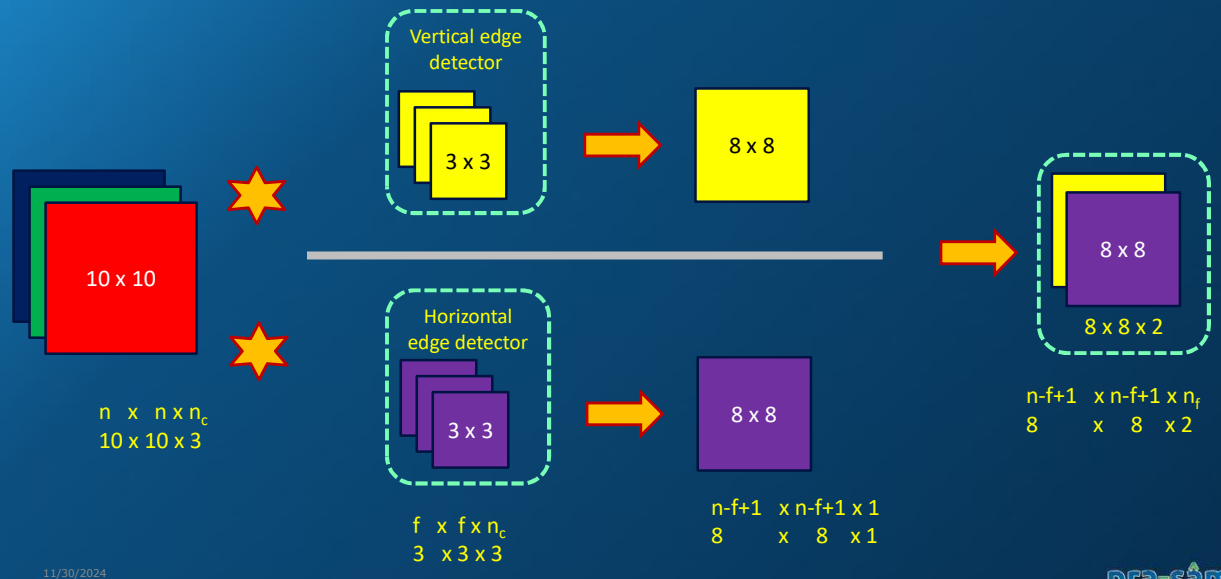
$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- 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.

pra-sami

60

Multiple Filters



pra-sami

61

Two Issues with the convolution…

- 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-sami

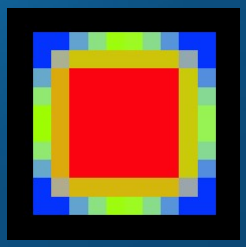
62

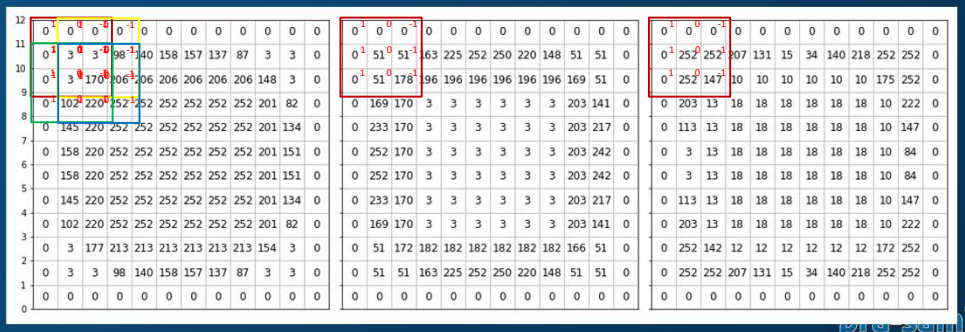
What if we zero pad the image all around… will it help?

pra-sami

63 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….

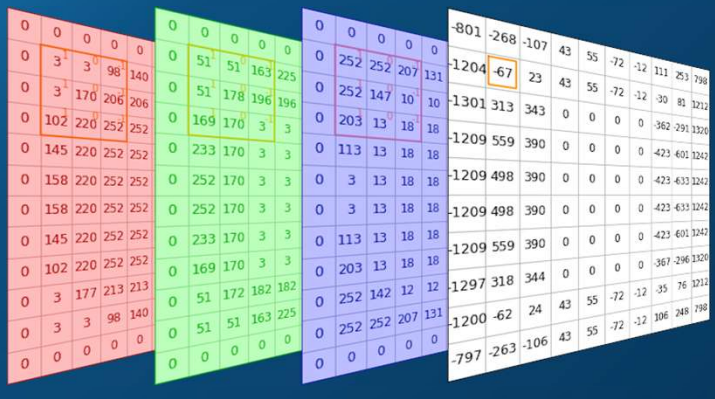




pra-sâmi

64 Convolution after Padding

- The respective cell values remain same



- and we are back with original image size

pra-sâmi

65 Convolution after Padding

- 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 x 10

We are back to original size…

pra-sami

66 How much to pad???

- There are two recommended mechanism
- **Valid** : output is calculated as

$$\{ \frac{nH_{in} - nF + 2*p}{stride} + 1 \} \text{ x } \{ \frac{nW_{in} - nF + 2*p}{stride} + 1 \}$$

- 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

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

  - or p = (nF -1)/2 for stride = 1

pra-sami

67

How much to pad???

- 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-sami

68

Cross-Correlation vs. Convolution

pra-sami

69

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

pra-sami

70

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... ☺

pra-sami

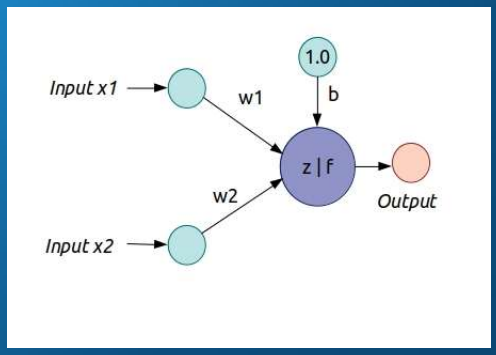
71

One layer of Convolutional Net

pra-sami

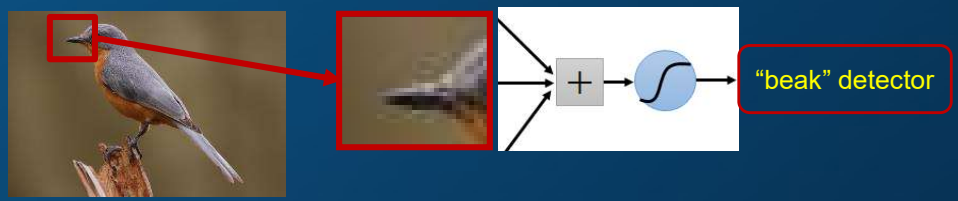
72

One layer of Conv Net



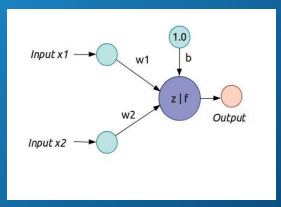
$Z_1 = a_0 . W_1 + b_1$

$a_1 = \text{Relu}(Z_1)$



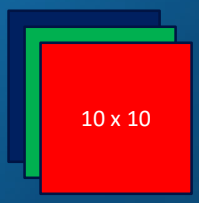
pra-sami

73 One Layer of Conv Net



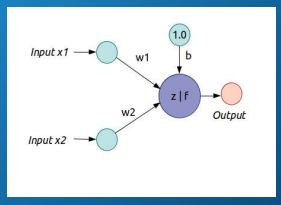
$Z_1 = a_0 . W_1 + b_1$

$a_1 = \text{Relu}(Z_1)$



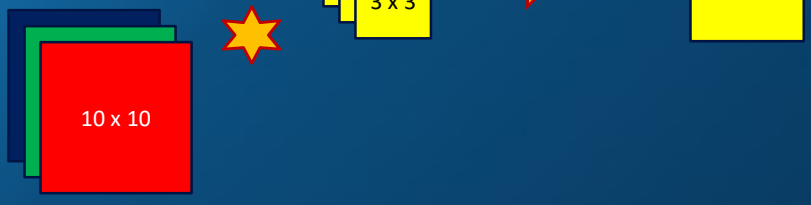
pra-sami

74 One Layer of Conv Net

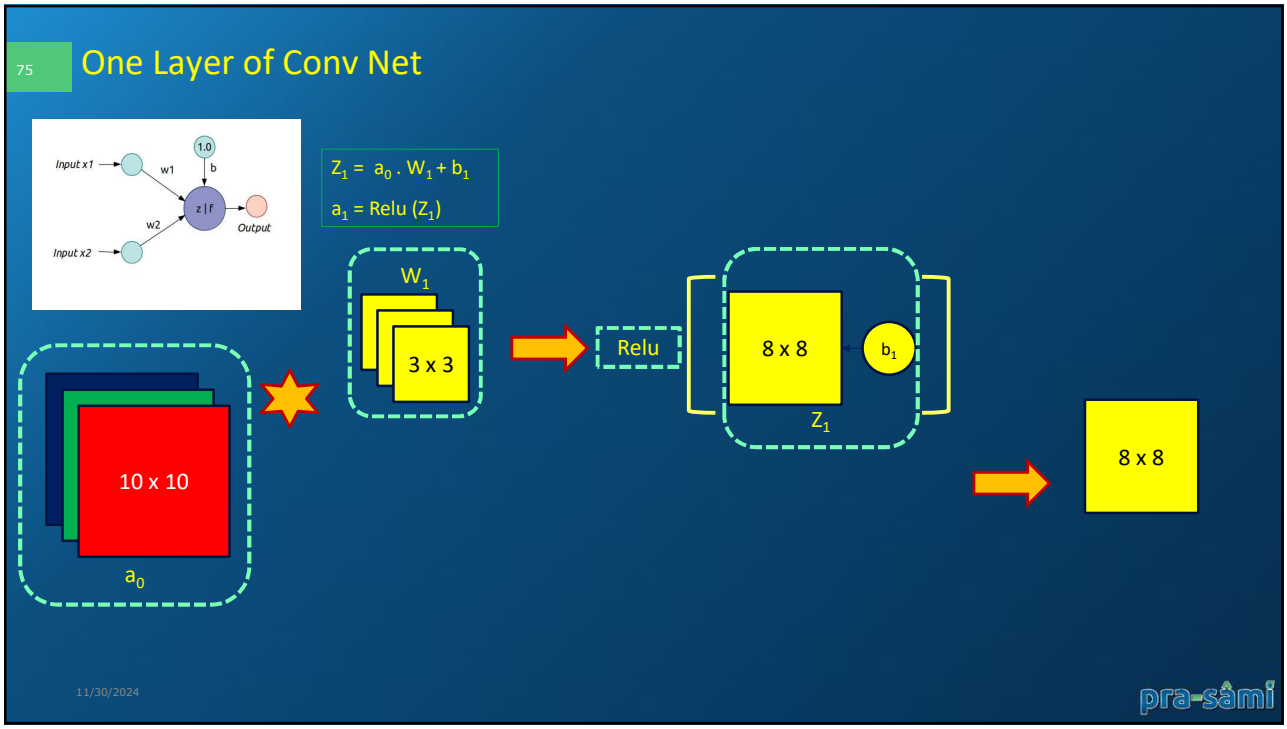


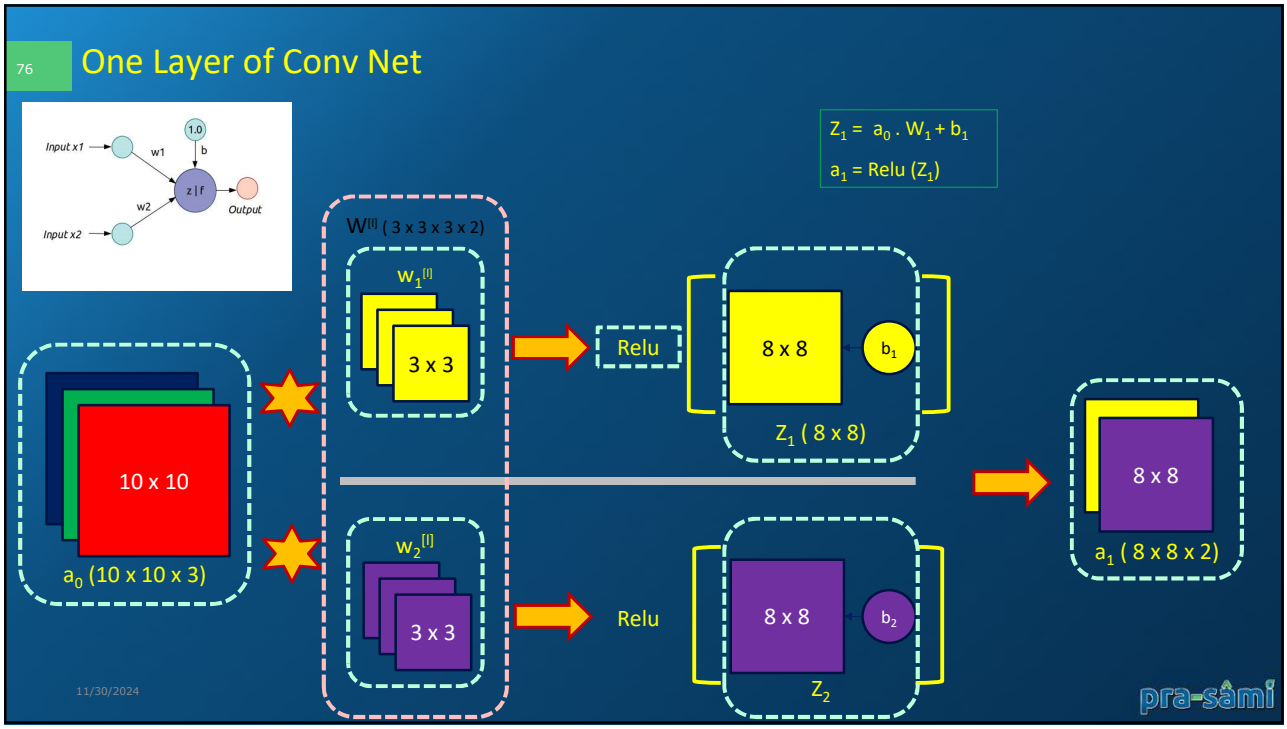
$Z_1 = a_0 . W_1 + b_1$

$a_1 = \text{Relu}(Z_1)$



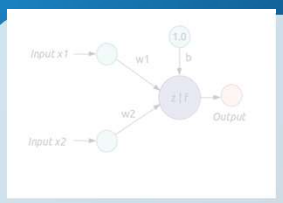
pra-sami

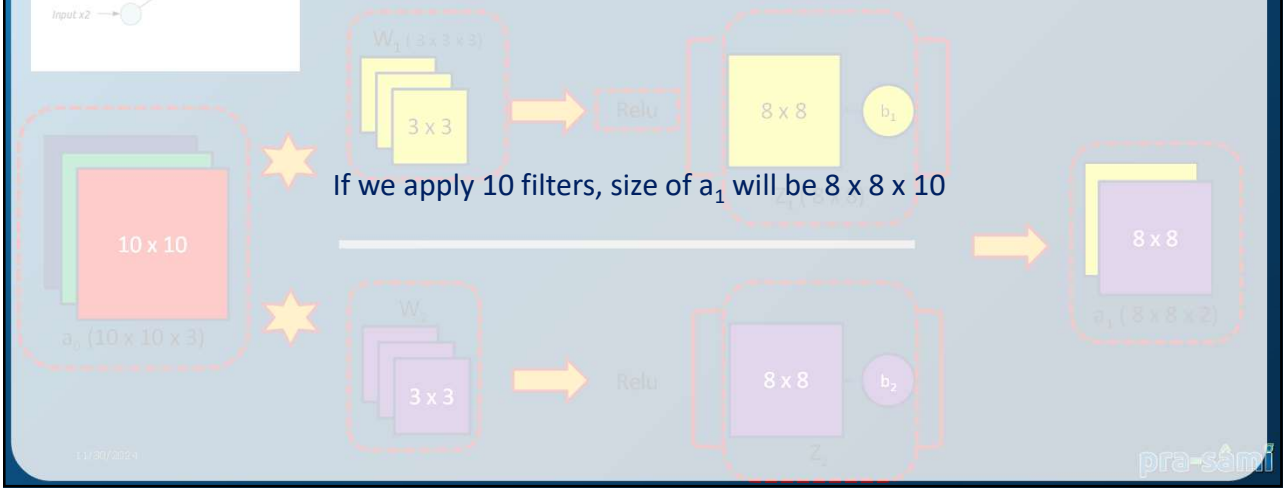

75
One Layer of Conv Net
Input x1
Input x2
w1
w2
b
1.0
z | f
Output
$Z_1 = a_0 . W_1 + b_1$
$a_1 = Relu (Z_1)$
W_1
3 x 3
Relu
8 x 8
b_1
Z_1
8 x 8
10 x 10
a_0
11/30/2024
pra-sami


76
One Layer of Conv Net
Input x1
Input x2
w1
w2
b
1.0
z | f
Output
$Z_1 = a_0 . W_1 + b_1$
$a_1 = Relu (Z_1)$
$W^{[l]}$ (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)
11/30/2024
pra-sami

77

One Layer of Conv Net



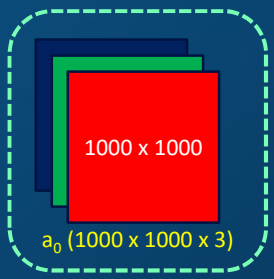


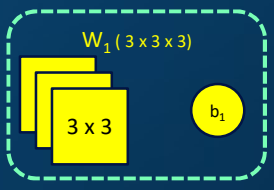
pra-sami

78

How Many Parameters…

- 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





pra-sami

79

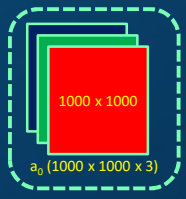
Lets Look at the Dimensions…

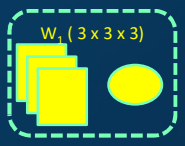
$f^{[l]}$:	Filter Size		Input:	$n^{[l-1]}_H \times n^{[l-1]}_W \times n^{[l-1]}_c$
$p^{[l]}$:	Padding size		Output:	$n^{[l]}_H \times n^{[l]}_W \times n^{[l]}_c$
$s^{[l]}$:	Stride		$n^{[l]}_H$:	$(n^{[l-1]}_H + 2\,p^{[l]} - f^{[l]}) / s^{[l]} + 1$
$n^{[l]}_c$:	Number of filters		$n^{[l]}_w$:	$(n^{[l-1]}_w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





pra-sami

80

A Simple CNN with Conv Layers

pra-sami

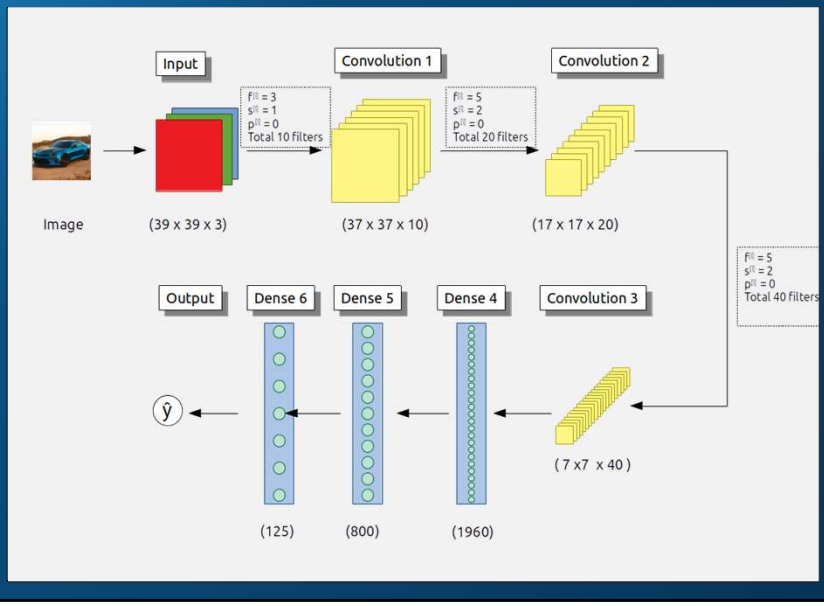
81 A Simple CNN with Conv Layers

pra-sami

82 A Simple CNN with Conv Layers

pra-sami

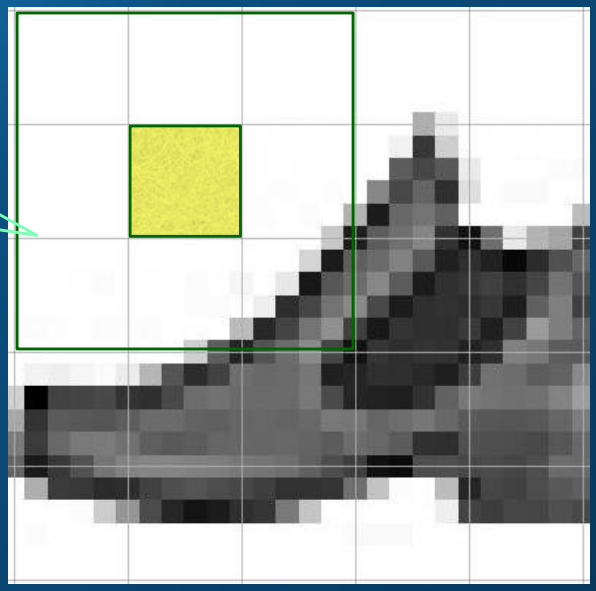
83 A Simple CNN with Conv Layers



pra-sami

84 Convolution – Applying Filters

9 datapoints result in one



pra-sami

85

Pooling…
What is most significant in this area…

pra-sâmi

86

Pooling

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

0	0	0	225	225	0	0	0
0	0	0	225	225	0	0	0
0	0	0	225	225	0	0	0
0	0	0	225	225	0	0	0
0	0	0	225	225	0	0	0
0	0	0	225	225	0	0	0
0	0	0	225	225	0	0	0
0	0	0	225	225	0	0	0

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

0	225	225	0
0	225	225	0
0	225	225	0
0	225	225	0

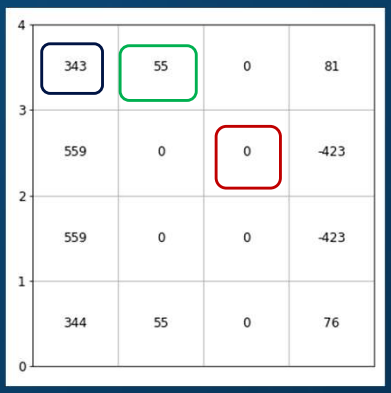
pra-sâmi

87

Other image

- 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



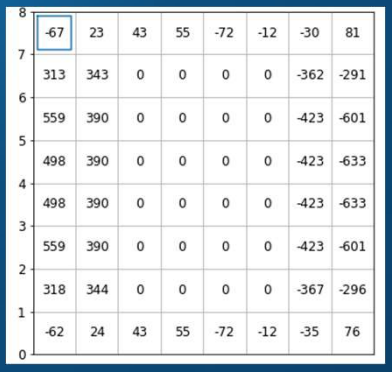
11/30/2024

pra-sami

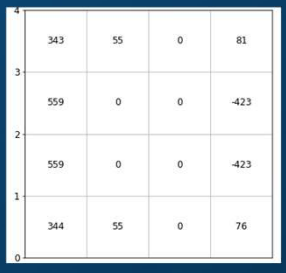
88

Pooling

- 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 option is to use Average instead of Max
  - But not used frequently.



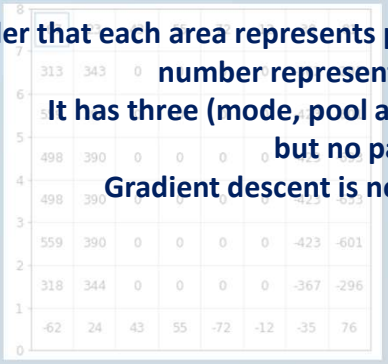
11/30/2024

pra-sami

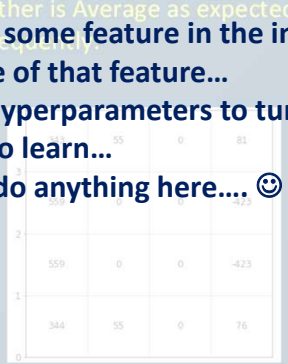
89

Pooling

- 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

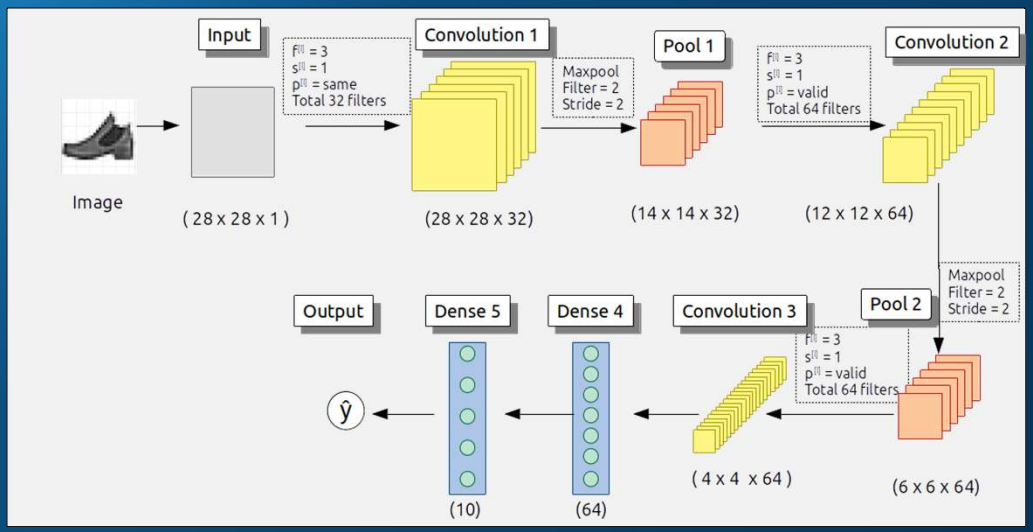


Consider that each area represents presence of some feature in the image and high number represents, presence of that feature…
It has three (mode, pool and stride) hyperparameters to tune…
but no parameters to learn…
Gradient descent is not going to do anything here…. ☺

pra-sâmi

90

Demo Example – Fashion MNIST



pra-sâmi

91

Congratulations!!!!

pra-sami

92

Reflect...

- What is the main purpose of Convolutional Neural Networks (CNNs)?
  - A) Text generation
  - B) Image and video recognition
  - C) Time series analysis
  - D) Natural language processing
- Answer: B) Image and video recognition
- Which of the following layers is a key component of CNNs?
  - A) Recurrent layer
  - B) Convolutional layer
  - C) Dropout layer
  - D) Activation layer
- Answer: B) Convolutional layer
- What is the role of the pooling layer in a CNN?
  - A) To increase the dimensions of the input
  - B) To extract features from the input data
  - C) To reduce the spatial dimensions (width and height) of the input
  - D) To combine multiple input channels
- Answer: C) To reduce the spatial dimensions (width and height) of the input
- Which operation is performed by a convolutional layer in a CNN?
  - A) Element-wise addition of the input matrix
  - B) Cross Correlation between a filter (kernel) and portions of the input matrix
  - C) Subtraction of one input channel from another
  - D) Matrix inversion
- Answer: B) Cross Correlation between a filter (kernel) and portions of the input matrix

pra-sami

93 Reflect…

- What is the function of a kernel (filter) in a CNN?
  - A) To resize images
  - B) To detect specific features like edges or textures in the input
  - C) To add noise to the image
  - D) To combine multiple images into one
- Answer: B) To detect specific features like edges or textures in the input
- What does stride refer to in a CNN?
  - A) The number of filters used
  - B) The number of steps the filter moves across the input matrix
  - C) The size of the input image
  - D) The number of output channels
- Answer: B) The number of steps the filter moves across the input matrix
- Which of the following is a common activation function used in CNNs?
  - A) Sigmoid
  - B) Tanh
  - C) ReLU (Rectified Linear Unit)
  - D) SoftMax
- Answer: C) ReLU (Rectified Linear Unit)
- In CNNs, what is the effect of padding?
  - A) To increase the number of filters
  - B) To prevent the reduction of spatial dimensions by adding zeros around the input matrix
  - C) To reduce the memory footprint of the model
  - D) To change the size of the kernel
- Answer: B) To prevent the reduction of spatial dimensions by adding zeros around the input matrix

pra-sami

94 Reflect…

- Why do deeper CNNs typically perform better than shallow CNNs?
  - A) Deeper CNNs have more parameters and can memorize the training data better
  - B) Deeper CNNs can learn hierarchical representations, capturing complex features
  - C) Deeper CNNs require fewer data points for training
  - D) Deeper CNNs prevent overfitting
- Answer: B) Deeper CNNs can learn hierarchical representations, capturing complex features
- What is the vanishing gradient problem in the context of CNNs?
  - A) The gradients become too small to update the weights effectively in deeper networks
  - B) The model loses information about small details in images
  - C) The network stops learning after a certain number of epochs
  - D) The gradients become too large, leading to unstable training
- Answer: A) The gradients become too small to update the weights effectively in deeper networks
- What is the purpose of the fully connected (dense) layer at the end of a CNN?
  - A) To downsample the input
  - B) To map the learned features to the final output classes
  - C) To combine the pooling layers into a single layer
  - D) To prevent overfitting by regularizing the model
- Answer: B) To map the learned features to the final output classes
- What is a common method to reduce overfitting in a CNN?
  - A) Use a very large filter size
  - B) Add more fully connected layers
  - C) Use techniques like dropout, data augmentation, or early stopping
  - D) Increase the learning rate
- Answer: C) Use techniques like dropout, data augmentation, or early stopping

pra-sami

95

pra-sami