CONVOLUTION NEURAL NETWORKS – CNN PART I

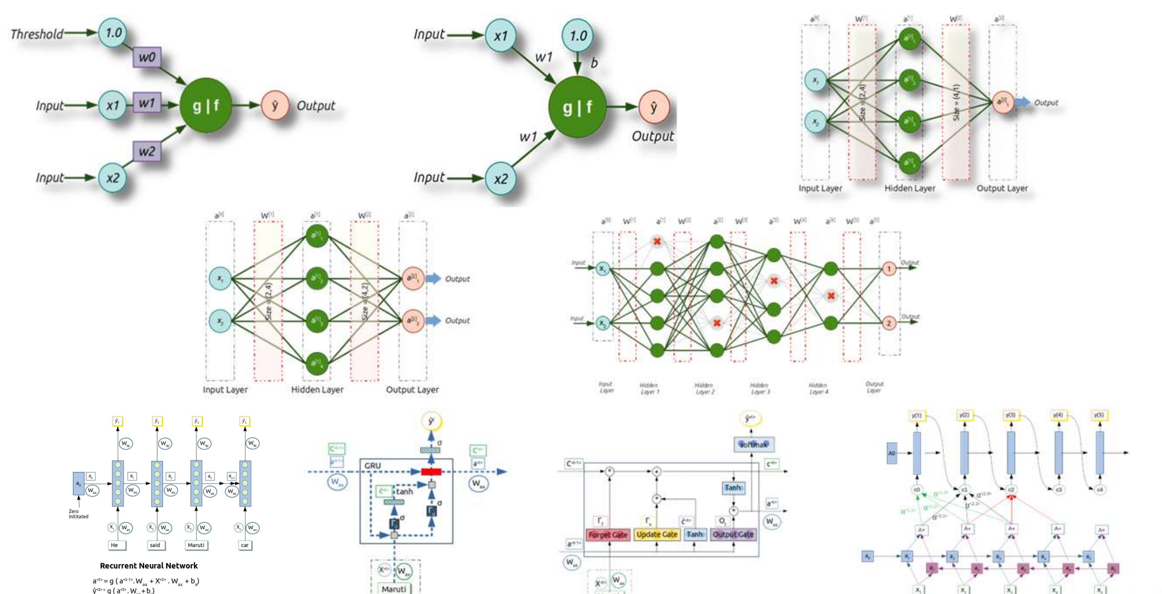
Deep Neural Networks

Session 21

Pramod Sharma
pramod.sharma@prasami.com

2

Story So Far…



5/28/2024

pra-sâmi

3

Agenda

- Computer Vision
- Convolution
- Padding
- Pooling
- Frameworks

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.

5

What is Computer Vision…

6

Ambulance given green light all through….

7

Computer vision is making progress in leaps and bounds…

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

pra-sâmi

9

Computer Vision

- 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

5/28/2024

pra-sâmi

10

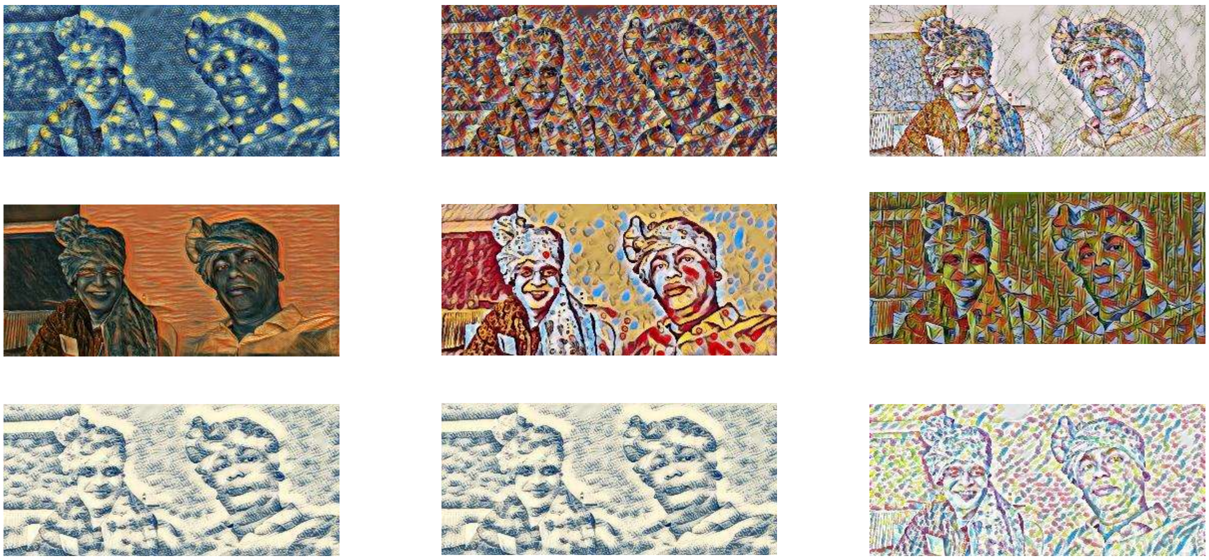
Computer Vision - Style transfer

5/28/2024

pra-sâmi

11

Computer Vision - Style transfer



5/28/2024

pra-sâmi

12

Computer Vision - Style transfer

5/28/2024

pra-sâmi

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)

5/28/2024

pra-sâmi

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

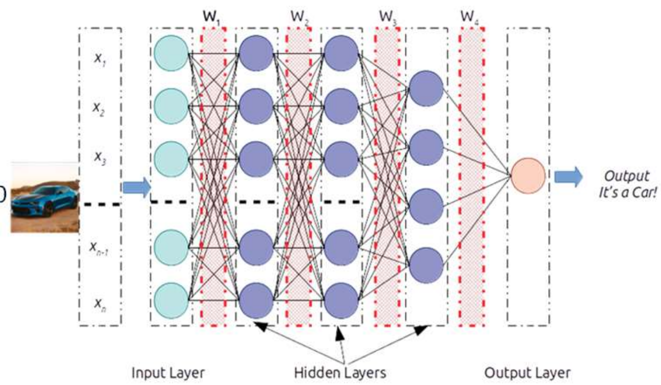
5/28/2024

pra-sâmi

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



5/28/2024

pra-sâmi

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?

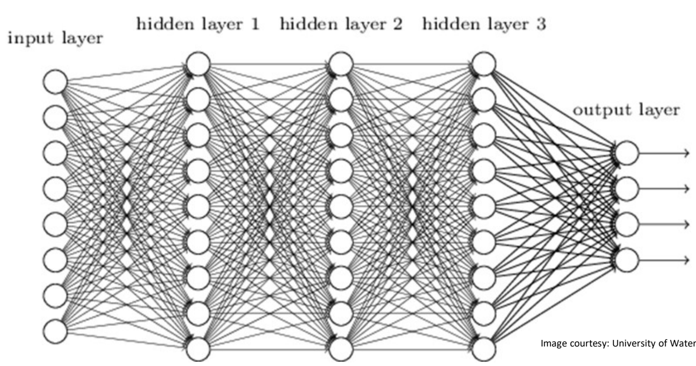


Image courtesy: University of Waterloo.

5/28/2024

pra-sâmi

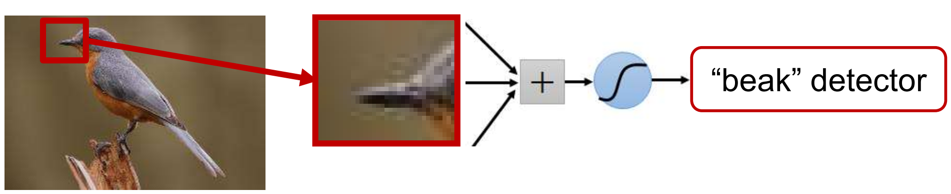
17

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

18

Learning an image…

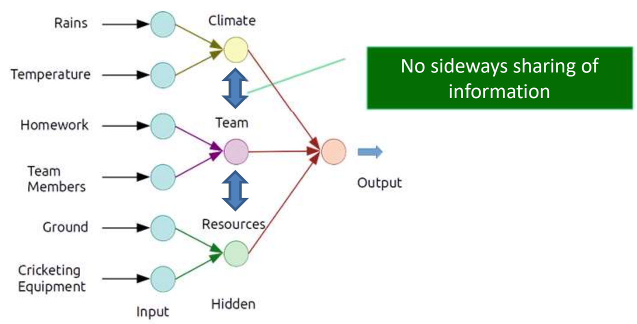
- Some patterns are much smaller than the whole image
- Can represent a small region with fewer parameters

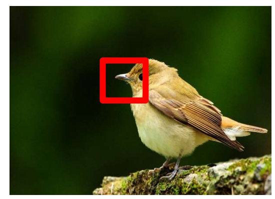


19

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”



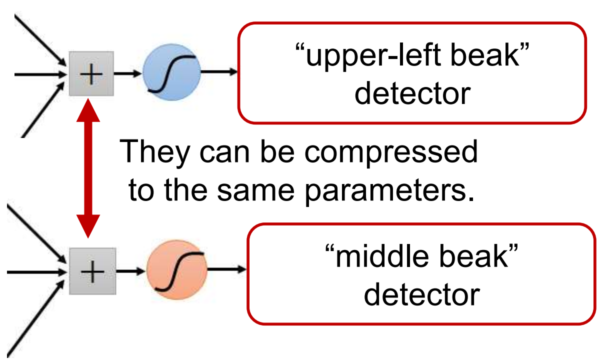


5/28/2024

pra-sâmi

20

Learning an image…



5/28/2024

pra-sâmi

21

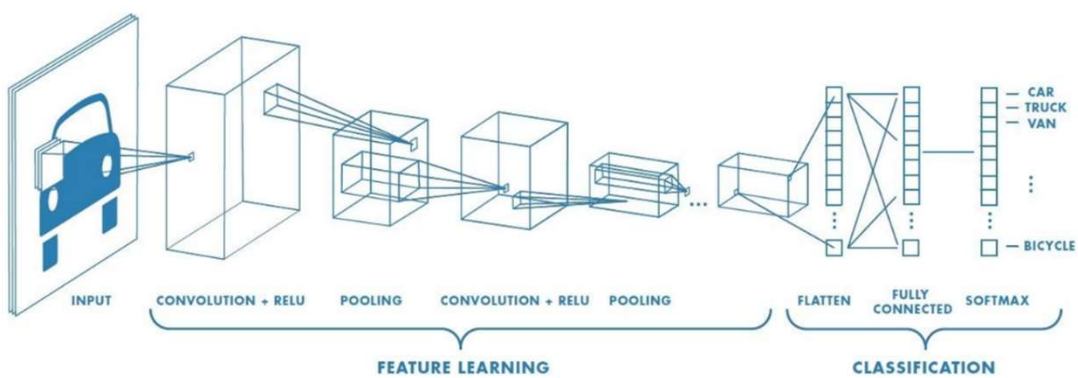
Learning an image…

- 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…

5/28/2024 pra-sâmi

22

Overall Layout



5/28/2024 pra-sâmi

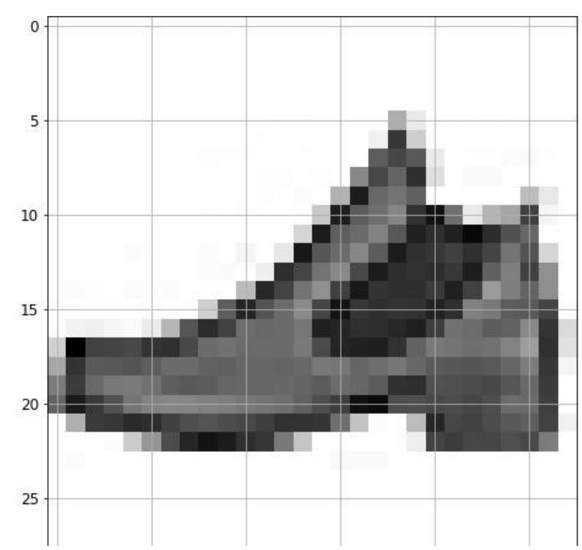
23

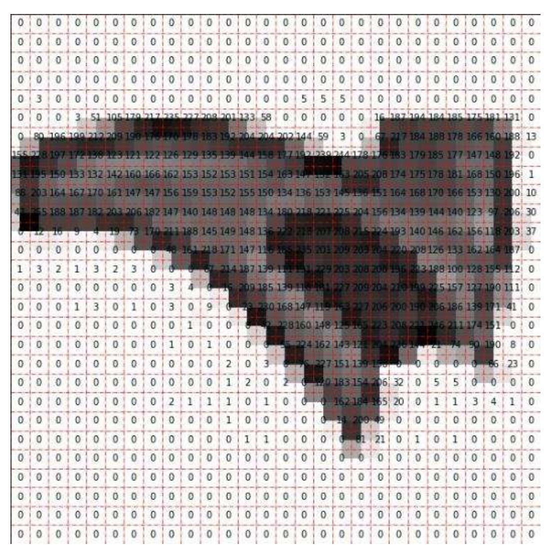
Each column of hidden units looks at a small region of the image, and the weights are shared between all image locations.

pra-sâmi

24

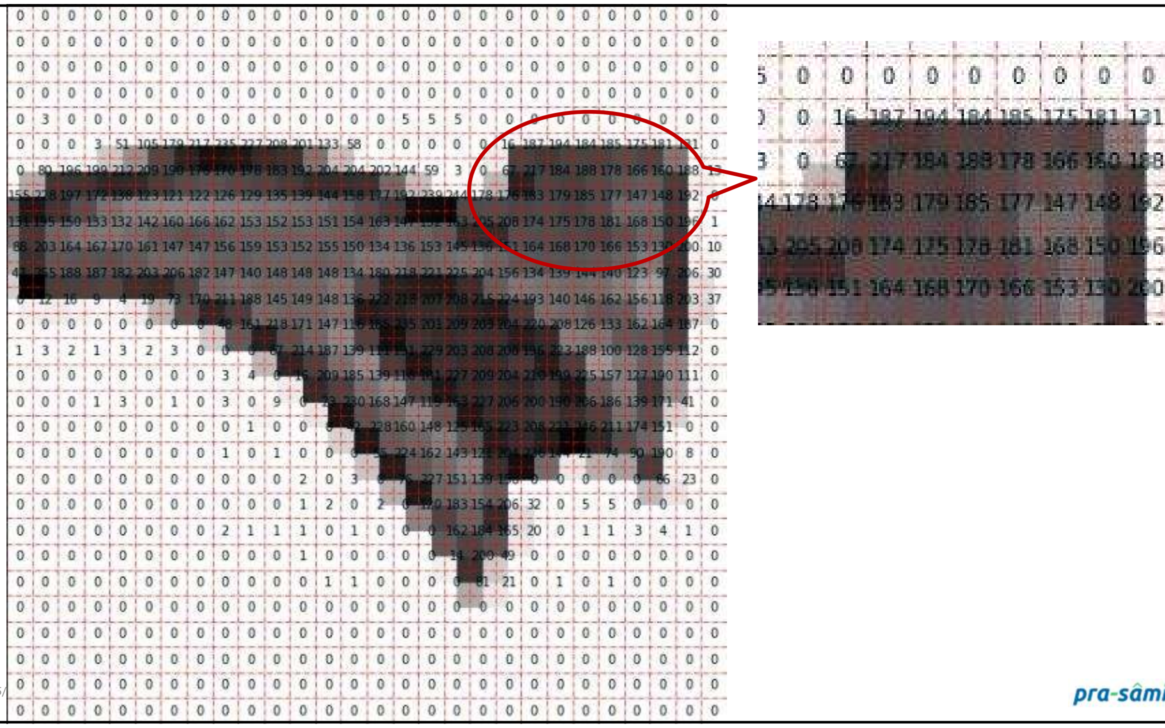
What computer sees…





pra-sâmi

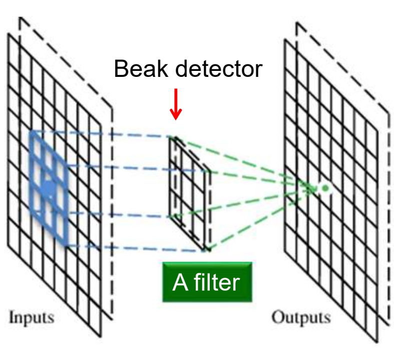
25



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

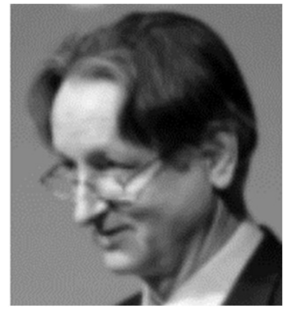


27

What does this Convolution Filter/ Kernel do?

$*$

0	1	0
1	4	1
0	1	0



28

What does this Convolution Filter/ Kernel do?

$*$

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

29

What does this Convolution Filter/ Kernel do?

$*$

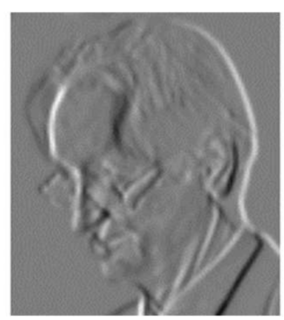
0	-1	0
-1	4	-1
0	-1	0

30

What does this Convolution Filter/ Kernel do?

$*$

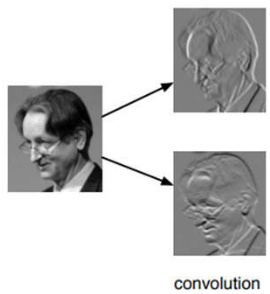
1	0	-1
2	0	-2
1	0	-1



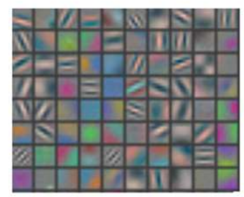
31

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.



Example first-layer filters

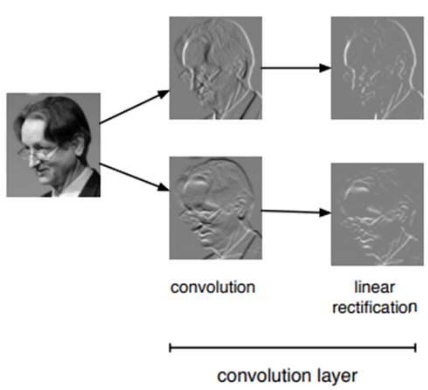


(Zeiler and Fergus, 2013, Visualizing and understanding convolutional networks)

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

Image with a edge

- Convolution is basic building block of image recognition

- Using edge detection as an example in following 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

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

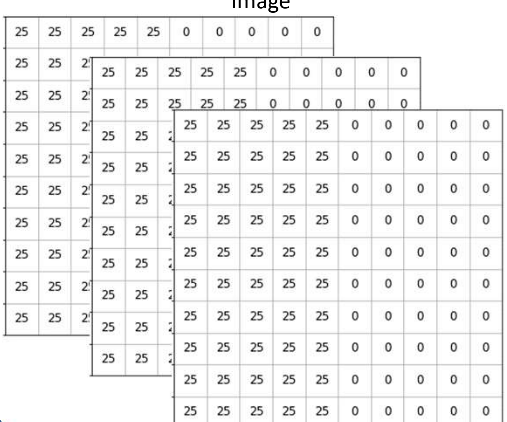
- Apply filters on the image!

5/28/2024

pra-sâmi

Convolution on 3D images (RGB)

Image



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

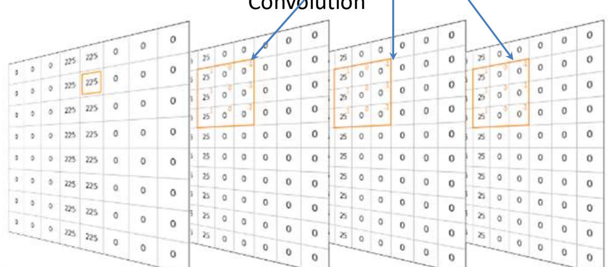
Consider this is a convolution operation

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



5/28/2024

pra-sâmi

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

5/28/2024

pra-sâmi

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

5/28/2024

pra-sâmi

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

5/28/2024

pra-sâmi

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

5/28/2024

pra-sâmi

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

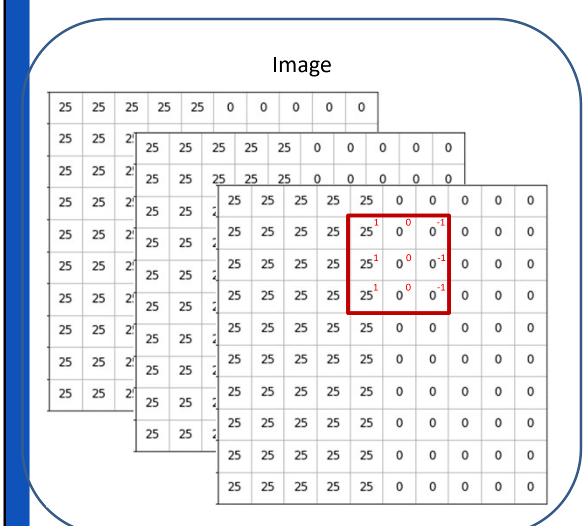
5/28/2024

pra-sâmi

40

Convolution

Image

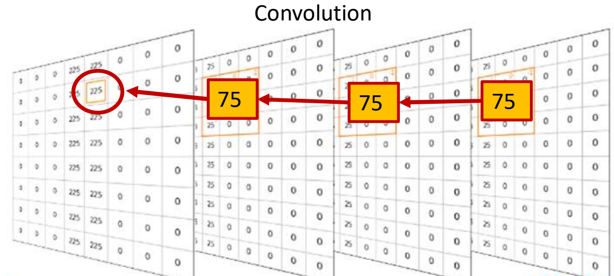


Filter

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

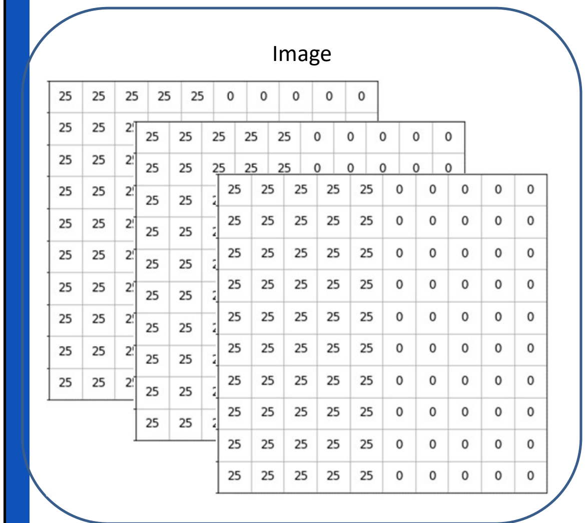


5/28/2024

pra-sâmi

41

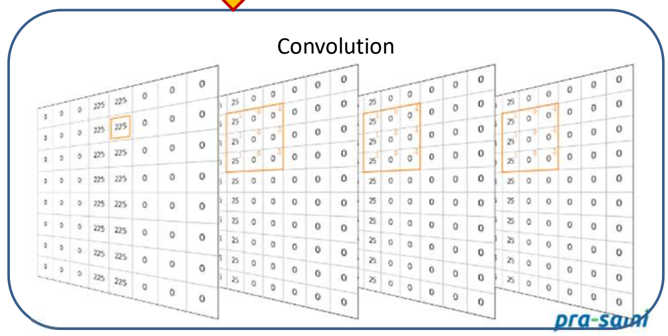
Convolution



Filter

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

- Every filter we deploy results in 2D matrix



5/28/2024

pra-sâmi

42

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

5/28/2024

pra-sâmi

43

Convolution

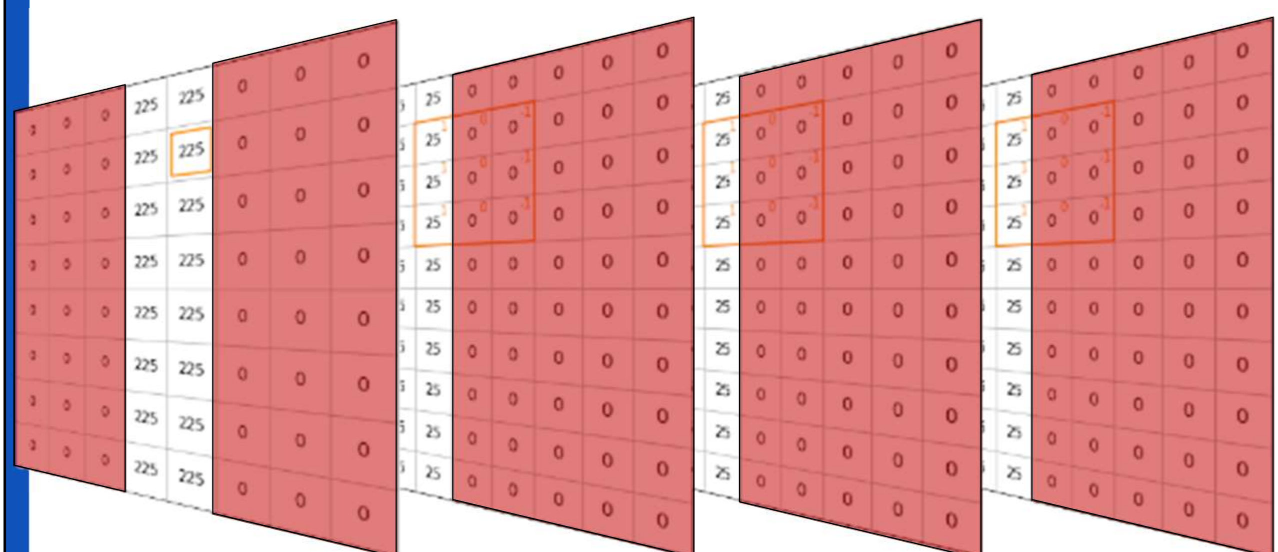
- 10 – 3 + 1 = 8 x 8

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

44

Has it detected the edge?



45

What if we move two steps at a time

pra-sâmi

46

Stride

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	[illegible]

- 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

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 =

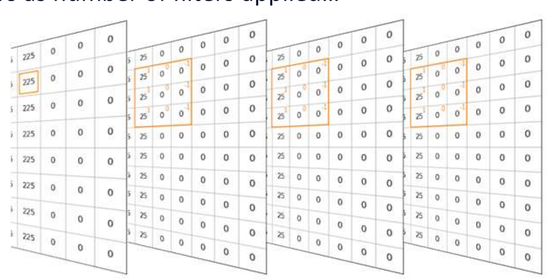
$$\{\,(\,nH_{in} - nF\,)\,/\,stride\, +\, 1\,\}$$
$$x$$
$$\{\,(\,nW_{in} - nF\,)\,/\,stride\, +\, 1\}$$

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

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…

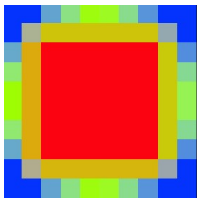


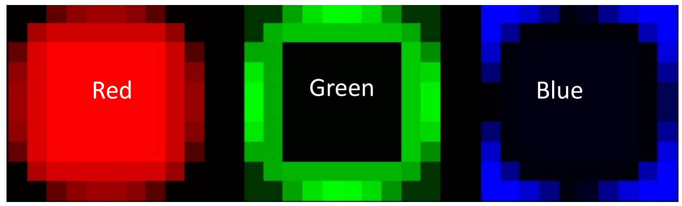
- Move “stride” steps, generally one or two
  - one in most cases…
- Strongly advisable to keep filters as odd shape (3,3) or (5,5)
- Two strong reason… We do not want asymmetric padding
  - Not good for learning features
  - It’s better to have central point of the filter

49

Another image

- This image has a few distinct edges





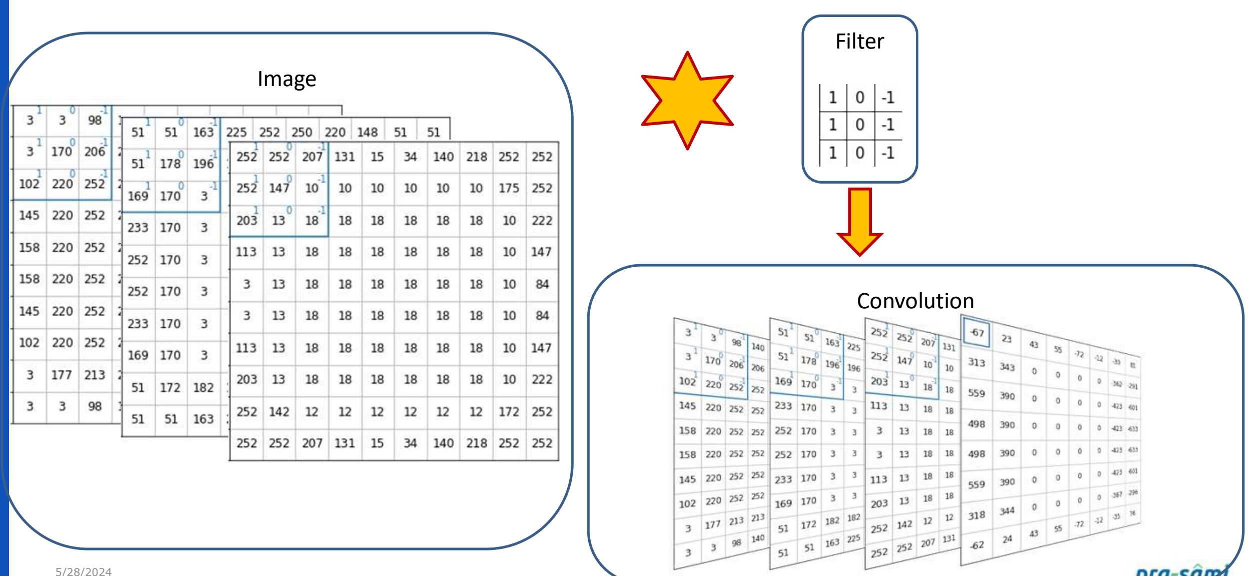
3	3	98	140	158	157	137	87	3	3
3	170	206	206	206	206	206	206	148	3
102	220	252	252	252	252	252	252	201	82
145	220	252	252	252	252	252	252	201	134
158	220	252	252	252	252	252	252	201	151
158	220	252	252	252	252	252	252	201	151
145	220	252	252	252	252	252	252	201	134
102	220	252	252	252	252	252	252	201	82
3	177	213	213	213	213	213	213	154	3
3	3	98	140	158	157	137	87	3	3

51	51	163	225	252	250	220	148	51	51
51	178	196	196	196	196	196	196	169	51
169	170	3	3	3	3	3	3	203	141
233	170	3	3	3	3	3	3	203	217
252	170	3	3	3	3	3	3	203	242
252	170	3	3	3	3	3	3	203	242
233	170	3	3	3	3	3	3	203	217
169	170	3	3	3	3	3	3	203	141
51	172	182	182	182	182	182	182	166	51
51	51	163	225	252	250	220	148	51	51

252	252	207	131	15	34	140	218	252	252
252	147	10	10	10	10	10	10	175	252
203	13	18	18	18	18	18	18	10	222
113	13	18	18	18	18	18	18	10	147
3	13	18	18	18	18	18	18	10	84
3	13	18	18	18	18	18	18	10	84
113	13	18	18	18	18	18	18	10	147
203	13	18	18	18	18	18	18	10	222
252	142	12	12	12	12	12	12	172	252
252	252	207	131	15	34	140	218	252	252

50

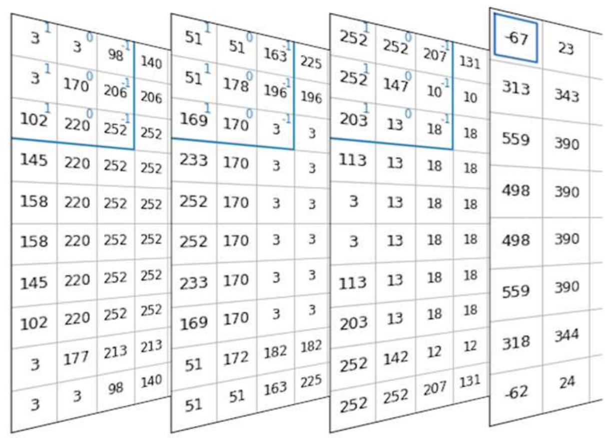
Another Convolution



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

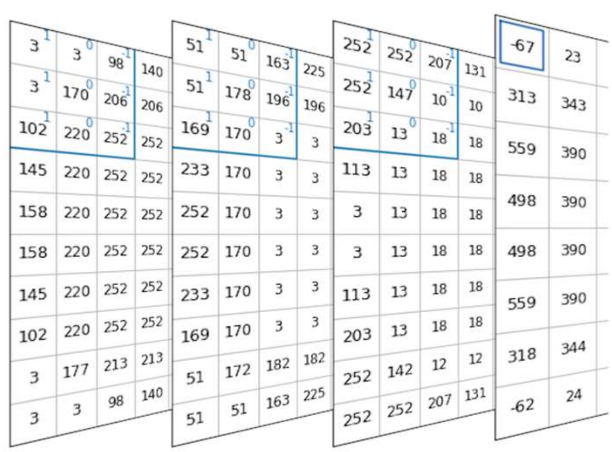


pra-sâmi

52

Another Convolution

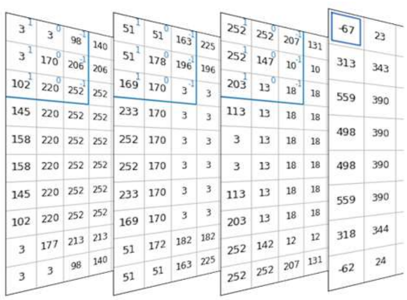
- Incoming Image shape = (10,10,3)
  - nH_{in} = 10 ; nW_{in} = 10, nC = 3
- Filter shape = (3, 3, 3)
  - nF = 3 ; nF = 3, nC = 3
- Stride = 1
- Hence the size will be 8 x 8 after convolution



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

5/28/2024 pra-sâmi

54 Other filters

- We have seen vertical filter… How about horizontal Filter….
- No surprises there….

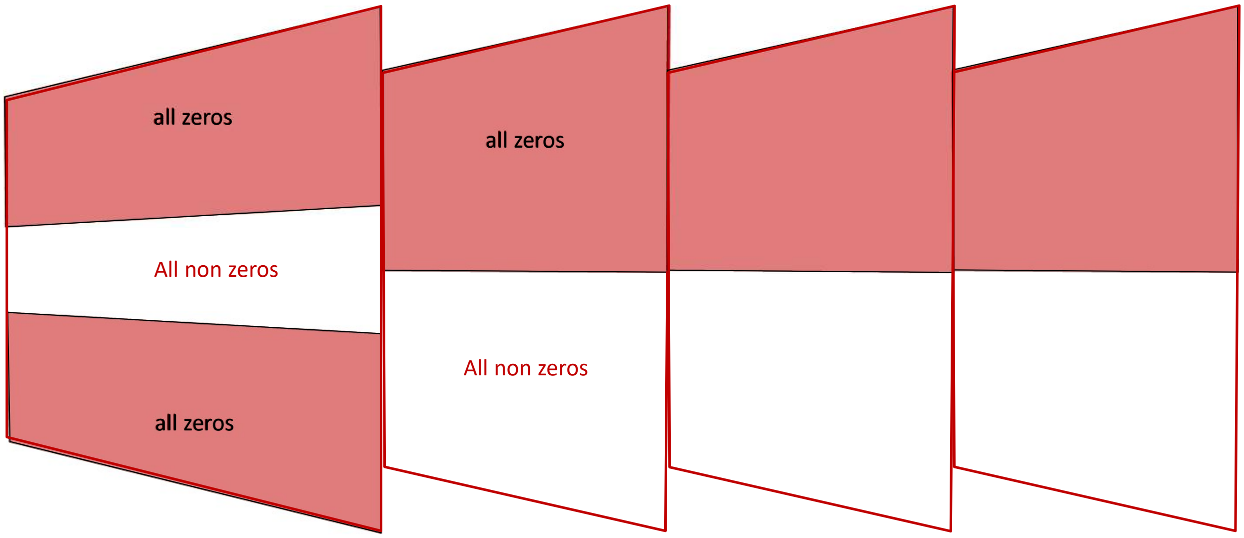
$$\boxed{\begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}}$$

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

5/28/2024 pra-sâmi

55

Horizontal Edge…



56

Other filters

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

- There was a lot of debate on filters….
- Researchers kept trying various numbers…

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

57

Shade Reversal

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

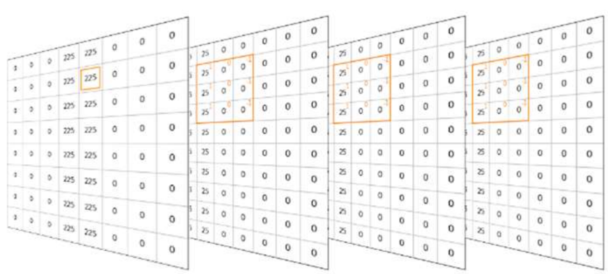
5/28/2024

pra-sâmi

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…



5/28/2024

pra-sâmi

59

Convolving a Volume

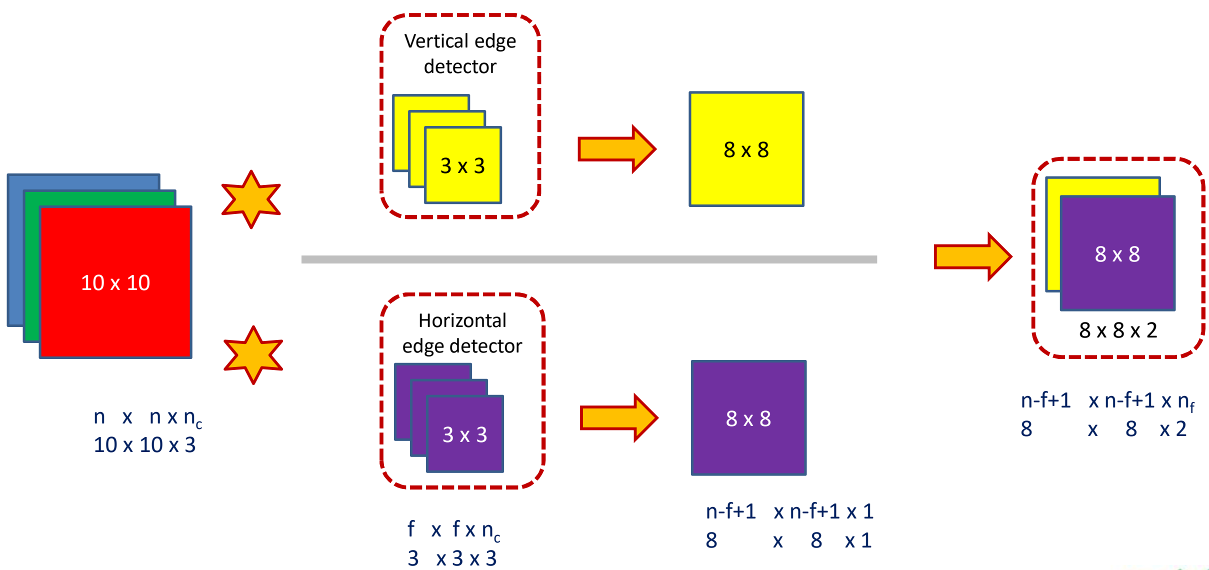
- So far we have shown that same filter is applied to all layers
- In theory, it is possible to have a filter which is looking for edges in red channel alone…

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

- 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.

60

Multiple Filters



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

5/28/2024

pra-sâmi

62

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

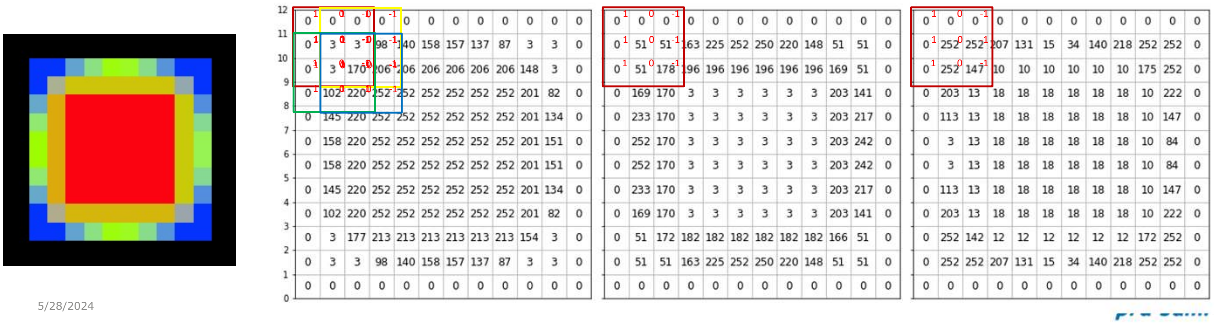
5/28/2024

pra-sâmi

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….

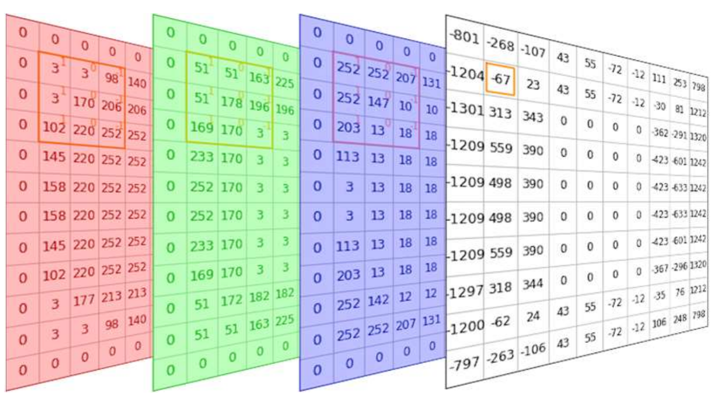


5/28/2024

64

Convolution after Padding

- The respective cell values remain same



- and we are back with original image size

5/28/2024

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 x 10

We are back to original size…

5/28/2024

pra-sâmi

How much to pad???

66

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

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

  $\{ \frac{nH_{in} - nF + 2*p}{stride} + 1 \}$ x $\{ (\frac{nW_{in} - nF + 2*p}{stride} + 1\}$ = $nH_{in}\ x\ nW_{in}$
  - or p = (nF -1)/2 for stride = 1

5/28/2024

pra-sâmi

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

68

Cross-Correlation vs. Convolution

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-sâmi

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-sâmi

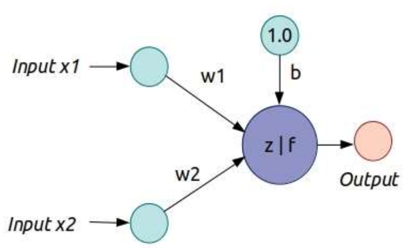
71

One layer of Convolutional Net

pra-sâmi

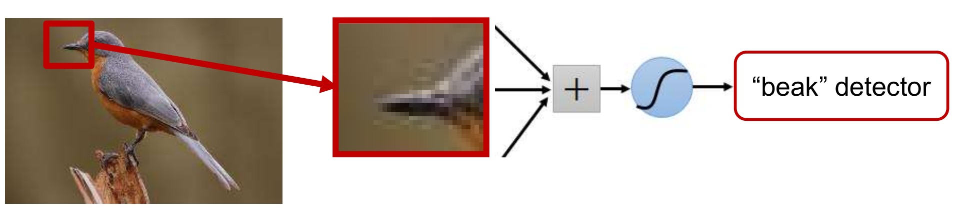
72

One layer of Conv Net



$Z_1 = a_0 . W_1 + b_1$

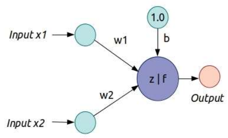
$a_1 = Relu (Z_1)$



pra-sâmi

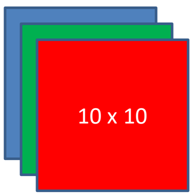
73

One Layer of Conv Net



$Z_1 = a_0 . W_1 + b_1$

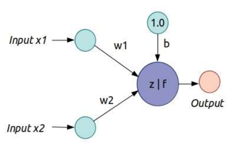
$a_1 = \text{Relu}(Z_1)$



5/28/2024

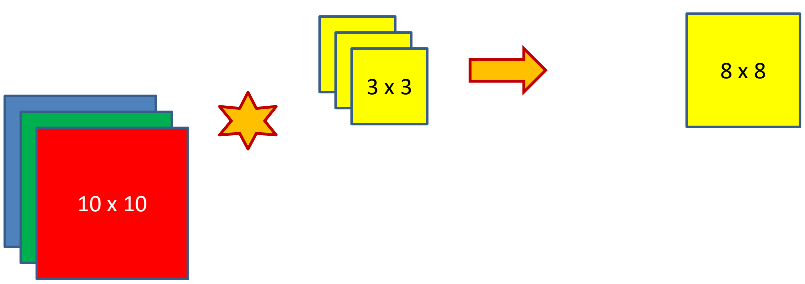
74

One Layer of Conv Net



$Z_1 = a_0 . W_1 + b_1$

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

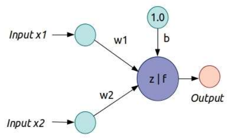


5/28/2024

pra-sami

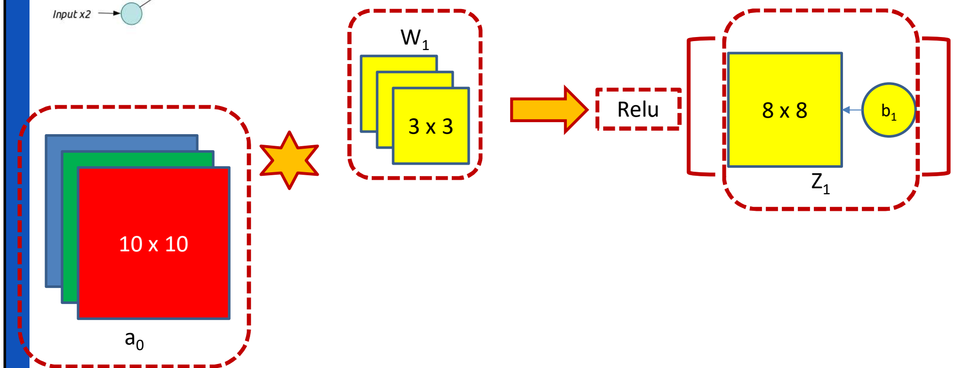
75

One Layer of Conv Net



$Z_1 = a_0 . W_1 + b_1$

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

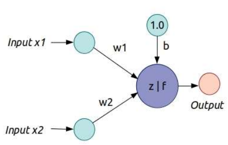


5/28/2024

pra-sami

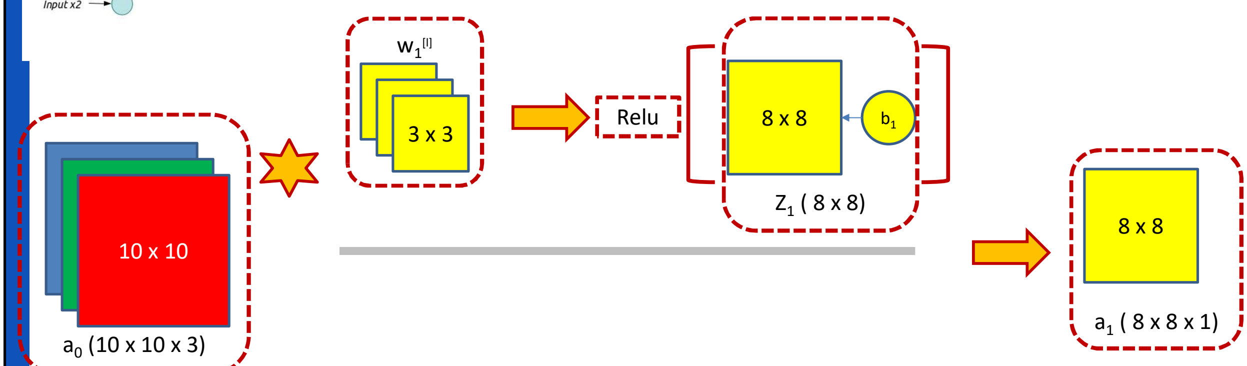
76

One Layer of Conv Net



$Z_1 = a_0 . W_1 + b_1$

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

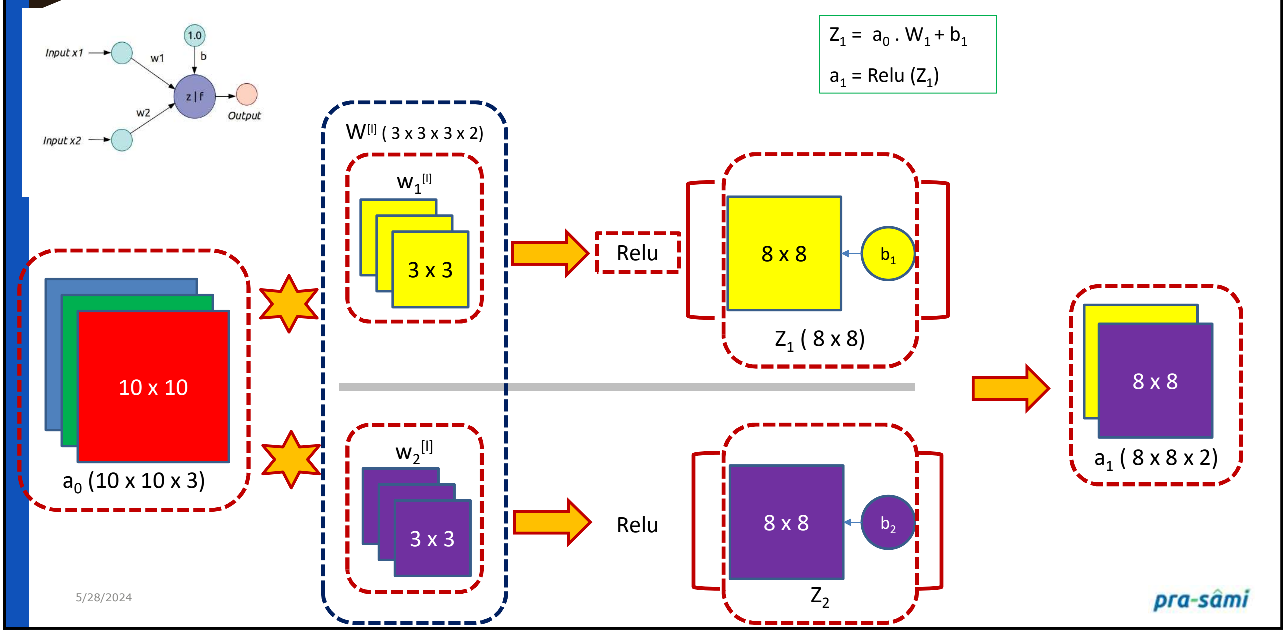


5/28/2024

pra-sâmi

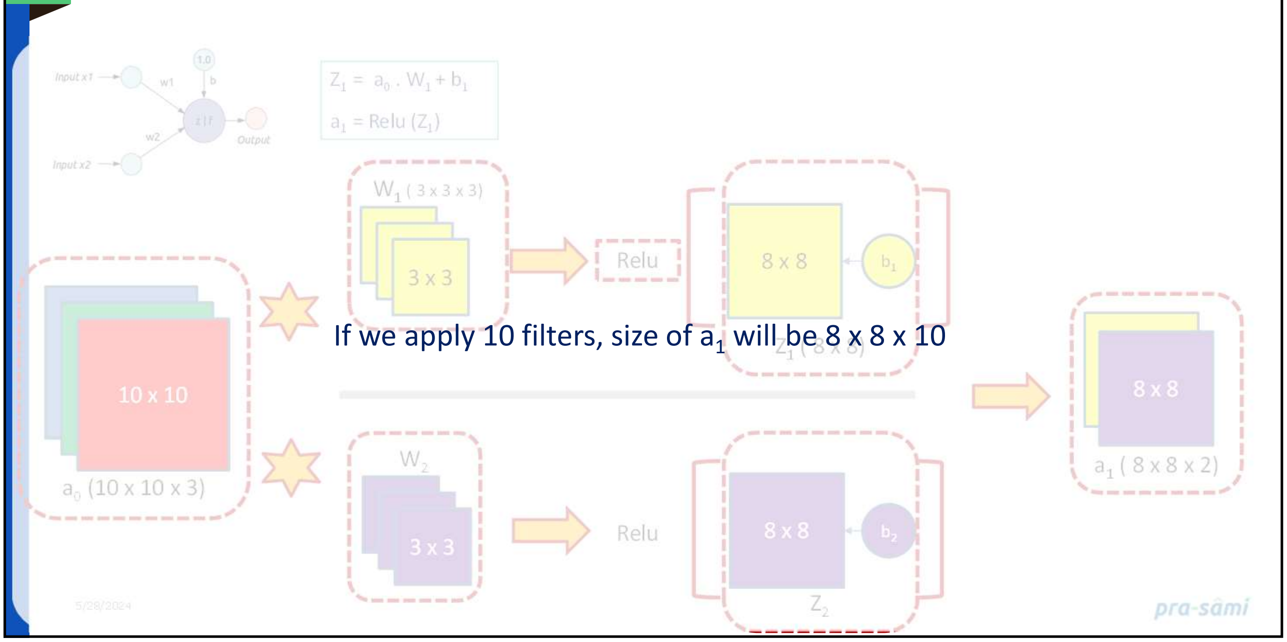
77

One Layer of Conv Net



78

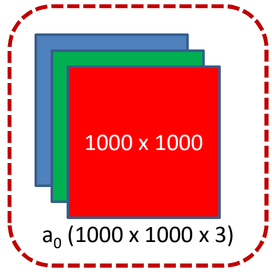
One Layer of Conv Net

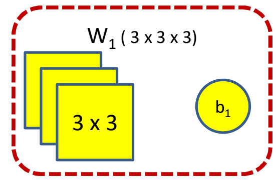


79

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





5/28/2024

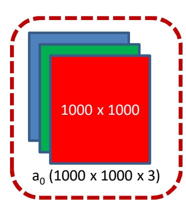
80

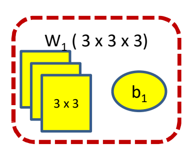
Lets Look at the Dimensions…

$f^{[l]}$:	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):	$\underbrace{f^{[l]} \times f^{[l]} \times n^{[l-1]}_c \times n^{[l]}_c}_{\text{Weights are tensors of rank 4}}$		Biases:	$n^{[l]}_c$

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

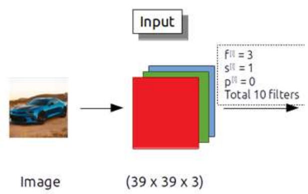




5/28/2024

81

A Simple CNN with Conv Layers

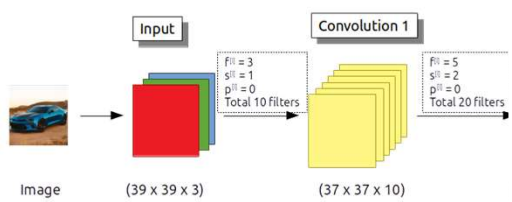


5/28/2024

pra-sâmi

82

A Simple CNN with Conv Layers

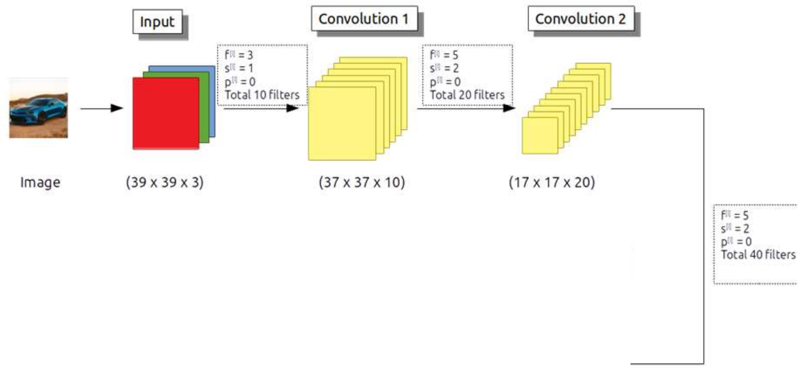


5/28/2024

pra-sâmi

83

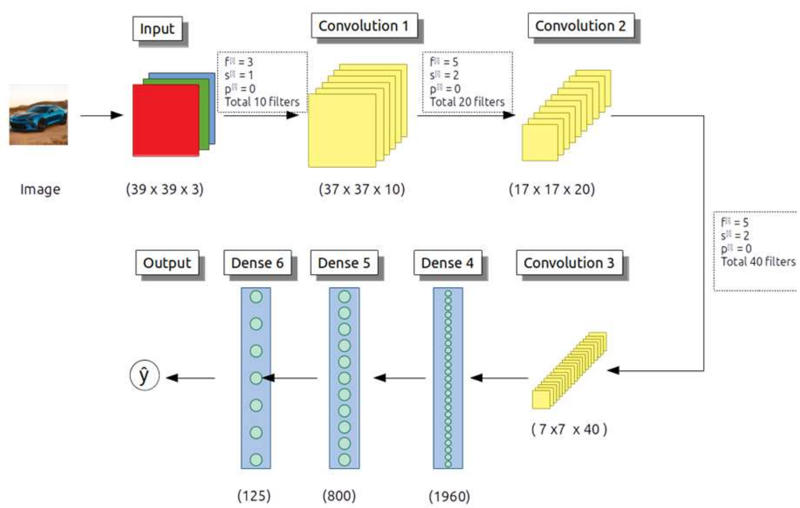
A Simple CNN with Conv Layers



pra-sâmi

84

A Simple CNN with Conv Layers

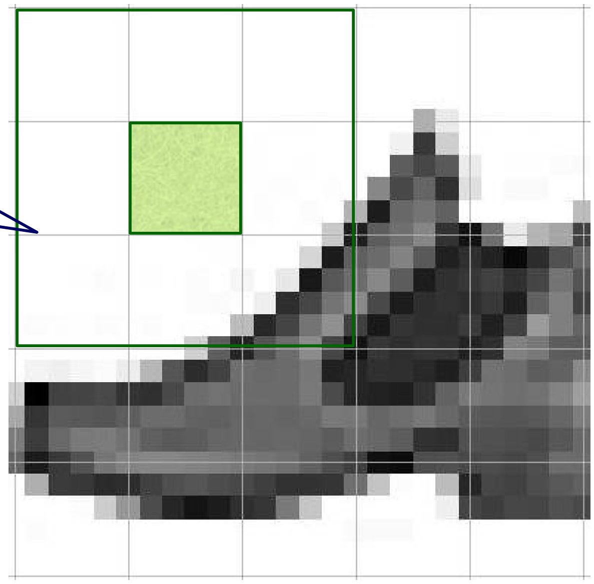


pra-sâmi

85

Convolution – Applying Filters

9 datapoints result in one



pra-sâmi

86

Pooling…
What is most significant in this area…

pra-sâmi

87

Pooling

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

2 x 2

Move by 2

0	0	0	225	225	0	0	0
0	0	0	225	225	0	0	0
[illegible]	[illegible]	[illegible]	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

88

Other image

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

2 x 2

Move by 2

-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

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

pra-sâmi

89

Pooling

- 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

5/28/2024

pra-sâmi

90

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 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…

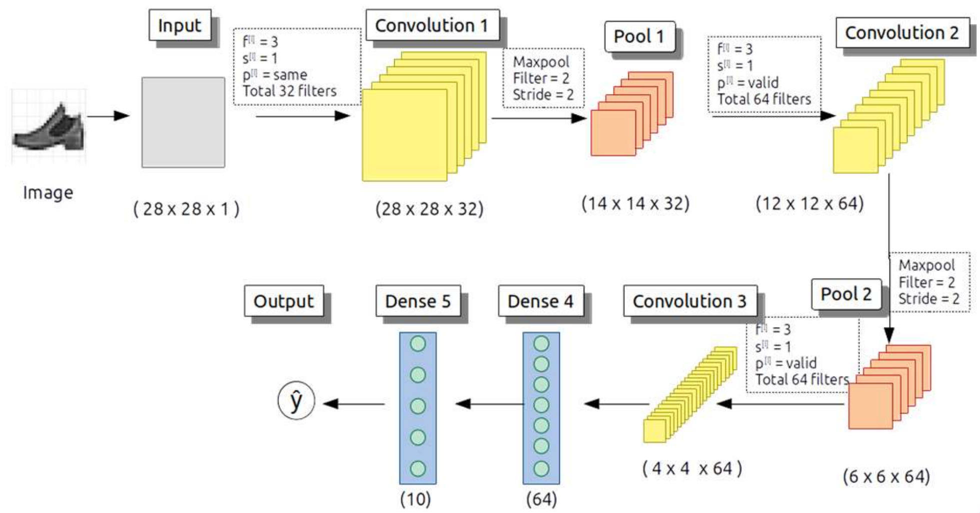
Gradient descent is not going to do anything here…. ☺

5/28/2024

pra-sâmi

91

Demo Example – Fashion MNIST

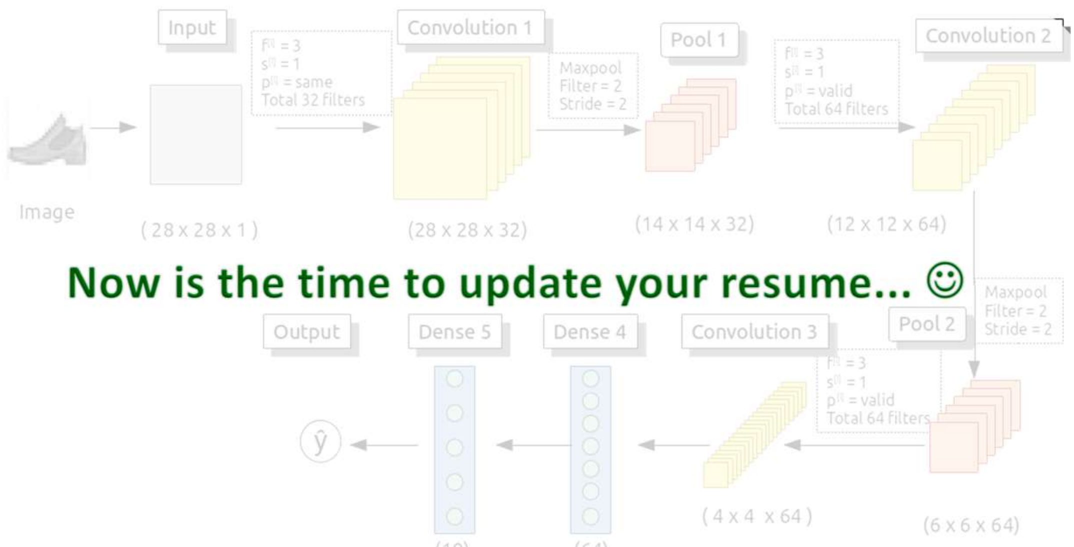


5/28/2024

pra-sâmi

92

Congratulations!!!!



5/28/2024

pra-sâmi

93

5/28/2024

pra-sâmi