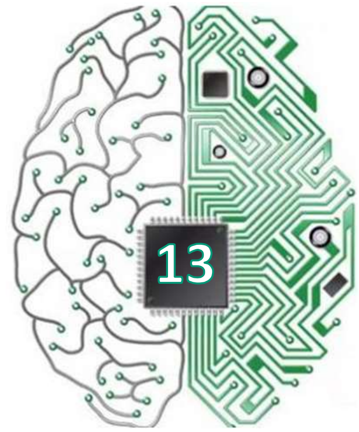
Open Elective Course [OE]
Course Code: CSO507
Winter 2023-24

Lecture#

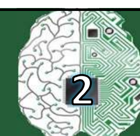
Deep Learning

Unit-4: Convolutional Neural Networks (Part-I)



Course Instructor:

Dr. Monidipa Das
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology (Indian School of Mines) Dhanbad, Jharkhand 826004, India



Convolutional Neural Network: Motivation [contd. from previous lecture]

Statistics of local images

- Many properties of the image patches are the same independent of their position
- Intuitively, objects can appear at different locations in the image
- They can also appear at different distance away from the camera

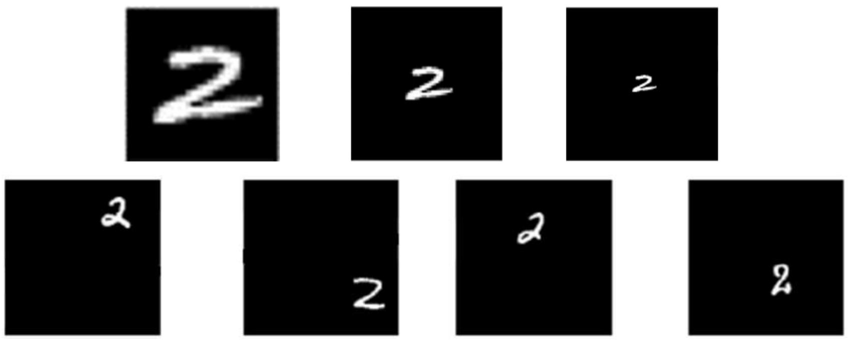
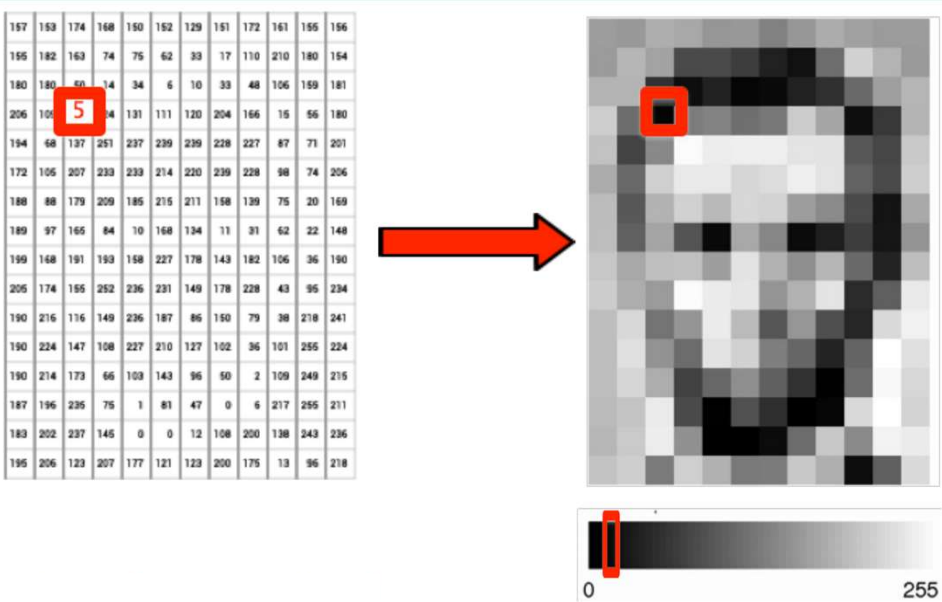
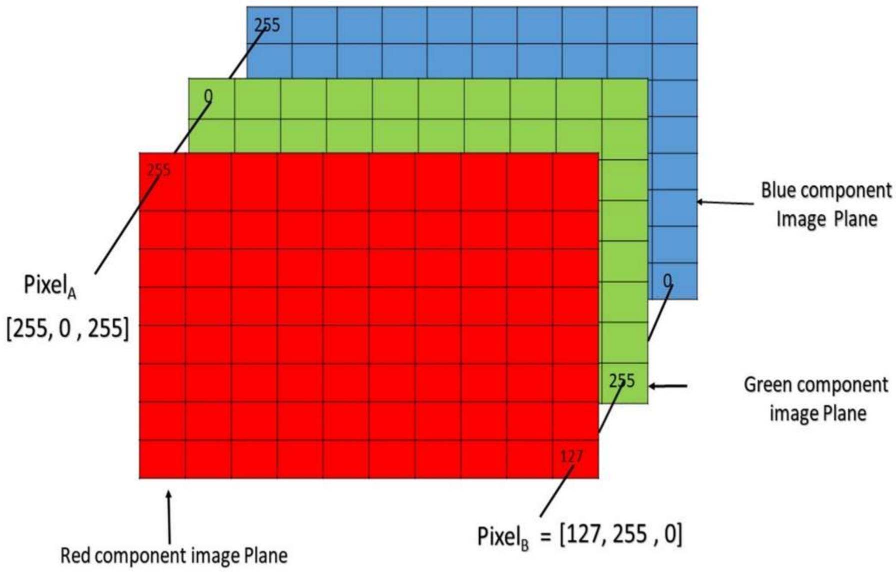


Image Representation (8-bit Grayscale)



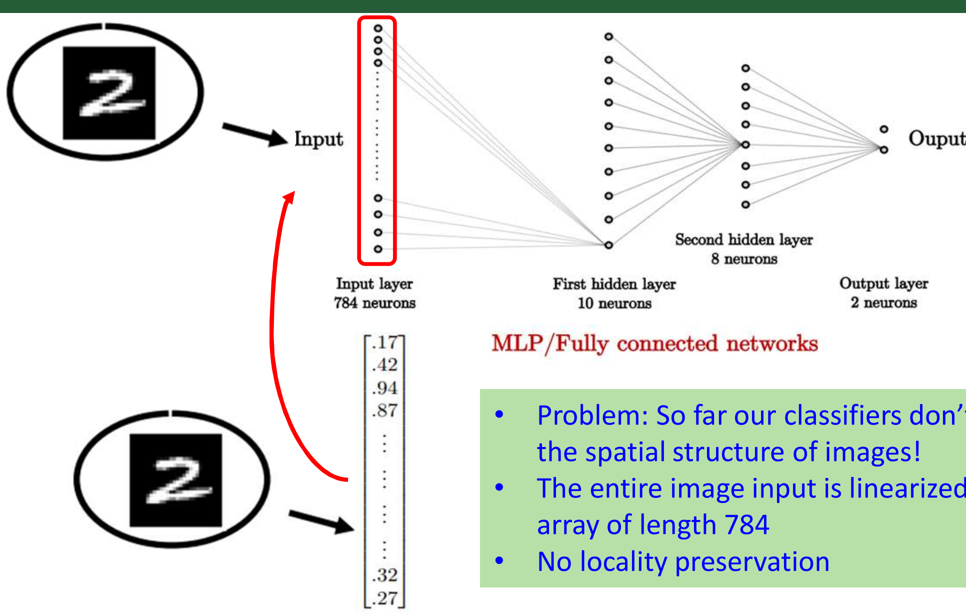
Color Images (e.g., 24-bit RGB image)

5



Ignorance of Spatial Structure

6



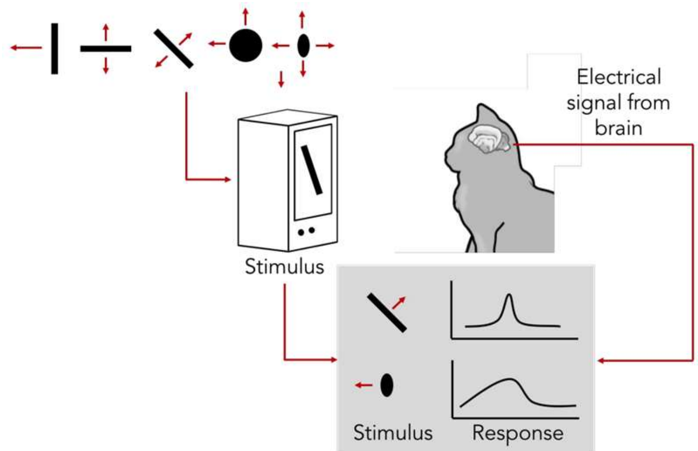
MLP/Fully connected networks

- Problem: So far our classifiers don't respect the spatial structure of images!
- The entire image input is linearized into an array of length 784
- No locality preservation

Biological Inspirations

7

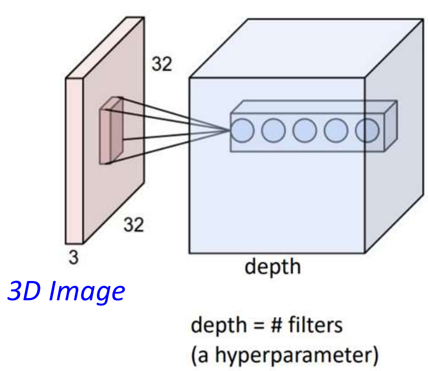
- **Work of Hubel and Wiesel in cat visual cortex (Nobel Prize)**
  - Discovery of neurons that respond maximally to specific stimulus patterns within their receptive field (simple cells)
  - Discovery of neurons that are more location invariant (complex cells)

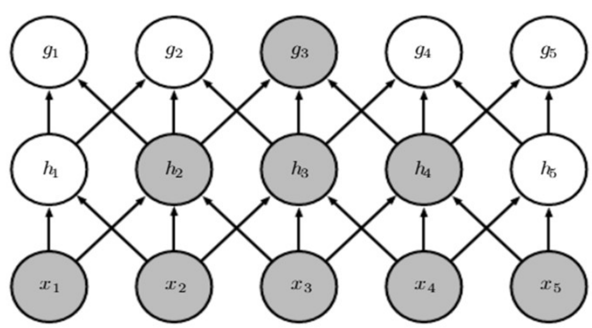


Convolutional Neural Networks: Key Idea 1

8

- **Features have local receptive fields**
  - Each hidden unit is connected to a patch of the input image.
  - Units are connected to all 3 colour channels.

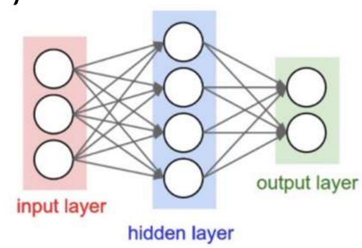




Fully-Connected Layers Are Limited

9

- **Issue: many model parameters in fully connected networks**
- Each node provides input to each node in the next layer
  - Many model parameters…
  - increases chance of overfitting
  - requires more training data
  - increases training time

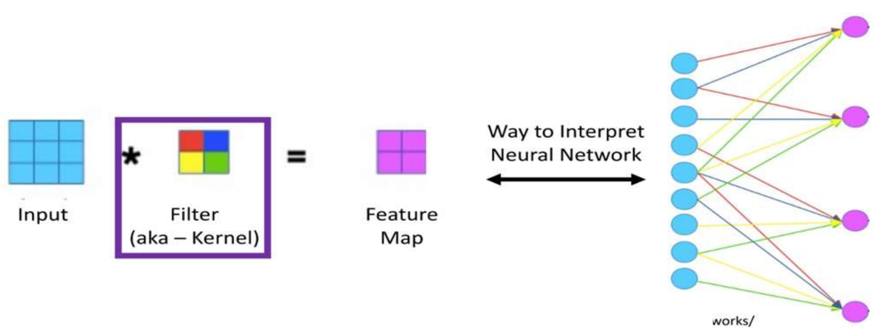


- Assume 2 layer model with 100 nodes per layer
  - e.g., how many weights are in a 640x480 grayscale image?
    - 640x480x100 + 100x100 + 100x1 = 30,730,100
  - e.g., how many weights are in a 3.1 Megapixel grayscale image (2048X1536)?
    - 2048x1536x100 + 100x100 + 100x1 = 314,582,900

Convolutional Neural Networks: Key Idea 2

10

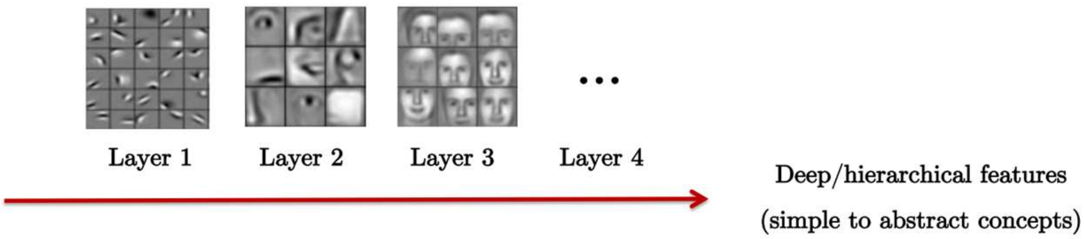
- **Share matrix of parameters across units**
  - Constrain units within a depth slice (at all positions) to have same weights.
  - Feature map can be computed via discrete convolution with a kernel matrix.
- **Sparse connectivity**
  - Convolutional layers dramatically reduce number of model parameters!



Hypothesis of hierarchical representations

11

- Image data are compositional :
  - formed from hierarchical local stationary patterns.
  - Local feature patterns can be composed to form abstract complex patterns :

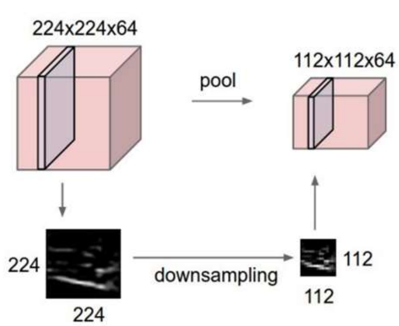


- Hard to be represented by non-hierarchical models (exponential number of parameters)
- Easily represented by hierarchical models (polynomial number of parameters)

Convolutional Neural Networks: Key Idea 3

12

- **Pooling/subsampling of hidden units in same neighborhood.**



3	3	0	1	1	1
7	2	1	1	5	2
7	9	1	0	5	6
5	6	1	0	3	2
9	7	0	0	0	0
4	7	0	0	0	1

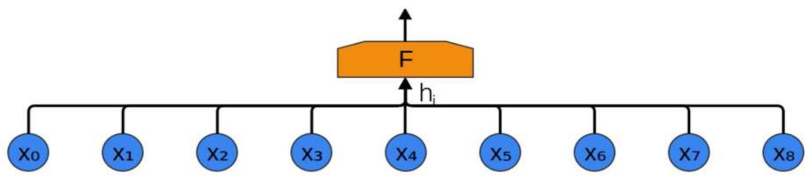
Down-sampling + max pooling $\Rightarrow$

7	1	5
9	1	6
9	0	1

Feed-forward NNs to CNNs

13

- Consider **1-D inputs** for simplicity, then generalize to 2-D.
- In a standard FF-NN, we feed all the input features to the hidden layer:



$$h_i = \phi(\mathbf{w}^\top \mathbf{x} + b), \mathbf{x} = [x_0, ..., x_8]$$

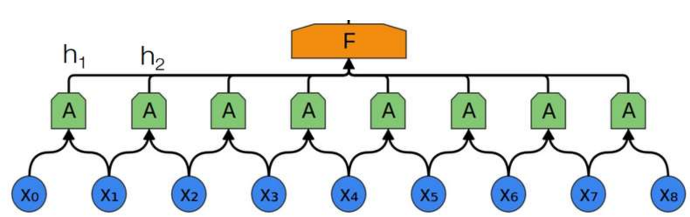
Feed-forward NNs to CNNs

14

- In a convolutional layer, we group the input units together, and apply the same function to these different groups.
- This is the idea of parameter sharing and local receptive fields.
- Mathematically equivalent to a discrete convolution operation.

$$h_1 = \phi(\mathbf{w}^\top [x_0, x_1] + b)$$
$$h_2 = \phi(\mathbf{w}^\top [x_1, x_2] + b)$$

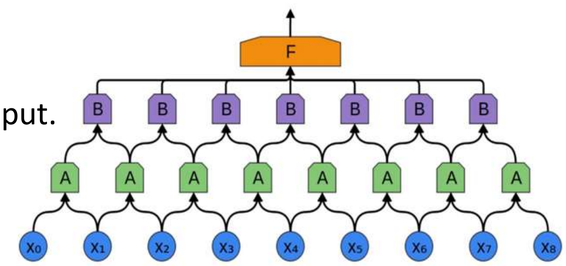
The parameters for the hidden units are shared… but the inputs are different and local!

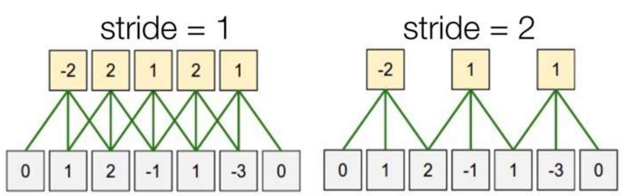


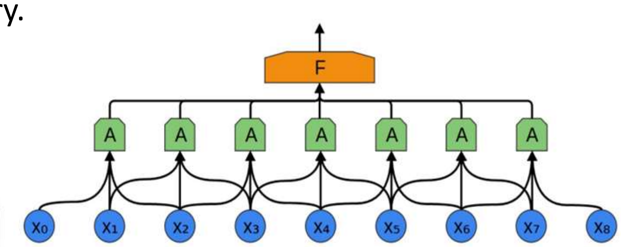
Feed-forward NNs to CNNs

15

- Can stack multiple convolutional layers.
- Can vary the width/size of the receptive field.
- Can apply multiple convolutions to the same input.
- Can vary the "stride" (i.e., spacing between receptive fields).
- It is common to add zero-padding to allow the application of the kernel near the boundary.





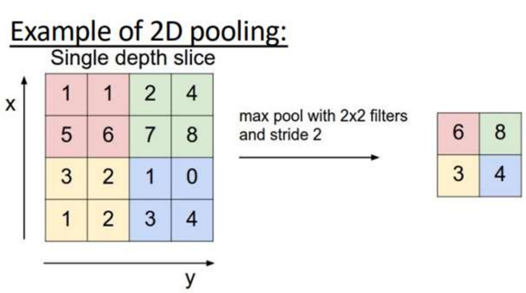


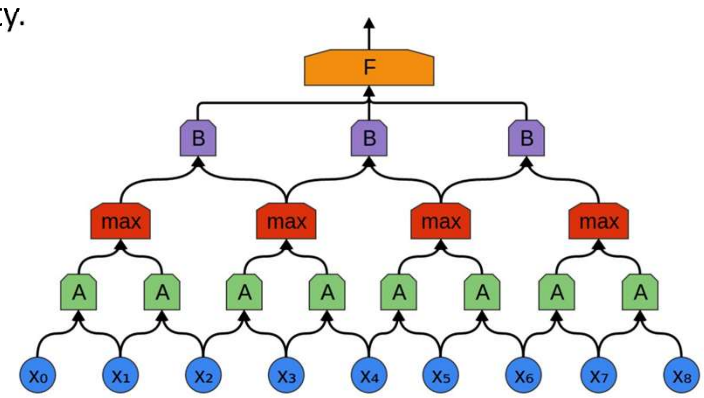
Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Feed-forward NNs to CNNs

16

- Pooling layers are often inserted between convolutional layers.
- Simply take the max of inputs in a receptive field.
- Further reduces the dimensionality.
- (Less popular in last couple years)



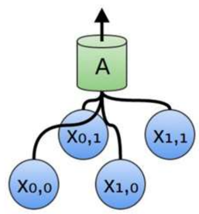


Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

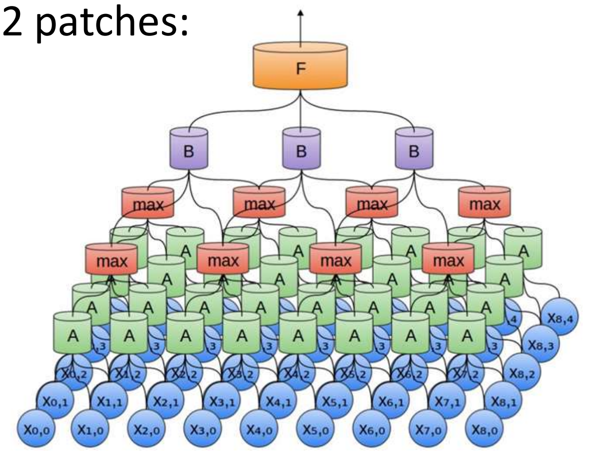
Feed-forward NNs to CNNs

17

- **In 2D** we group patches together in receptive fields.
- i.e., a 2x2 convolution groups 2x2 patches:



$$h_1 = \phi(\mathbf{W}[x_{0,0}, x_{0,1}, x_{1,1}, x_{1,1}]^\top + b)$$

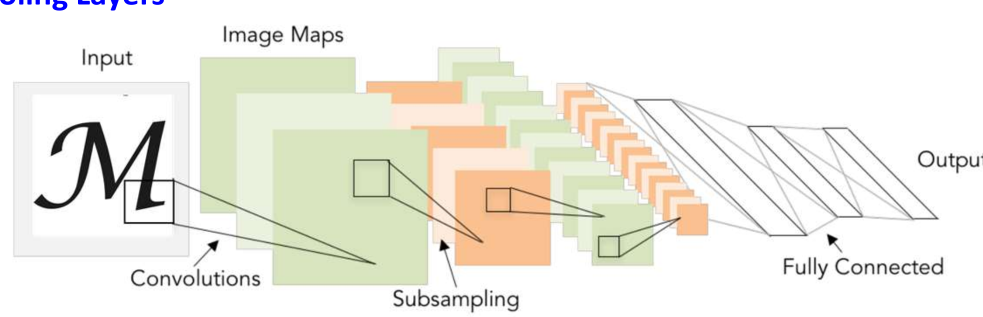


Putting it all together

18

CNN
A **ConvNet** is a sequence of convolutional layers, interspersed with activation functions (and possibly other layer types)

- Major Components of a Convolutional Network
  - **Convolution Layers**
  - **Pooling Layers**



19

"CNN" vs "ConvNet"

- There are many papers that use either phrase, but "ConvNet" is the preferred term, since "CNN" clashes with other things called CNN

--- Yann André LeCun

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Introduction

20

- Convolutional Neural Networks (ConvNets) are a specialized kind of neural networks for processing data that has a **known grid like topology**.
- Example of such data can be 1-D time series data sampled at regular intervals, or 2-D images.
- As the name suggests, these networks employ the mathematical **convolution** operator.
- **Convolutions** are a special kind of **linear** operators that an be used instead of general matrix multiplication.

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

The Convolution Operation: 1D Case

21

- **Mathematical Definition**
  - The convolution operator is mathematically defined as:

$$s(t) = (x * w)(t) = \int x(a)w(t-a)da$$
$$= \sum_{a=-\infty}^{\infty} x(a)w(t-a)$$

  - Note that the infinite summation can be implemented as a finite one as it is assumed that these functions are zero everywhere except at t where a measurement is provided.

Mathematical Definition : 2D Case

22

- The convolution operator is **mathematically defined as**:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$
$$= \sum_m \sum_n I(i-m, j-n)K(m,n)$$

- Usually in convolutions, we flip the kernel, which gives rise to the above commutative property.

Terminology

23

- I is usually a multidimensional array of data termed the **input**.
- K is usually a multidimensional array of parameters termed the **kernel** or the **filter**.
- S is the **output** or **feature map**.

Mathematical Definition : 2D Case

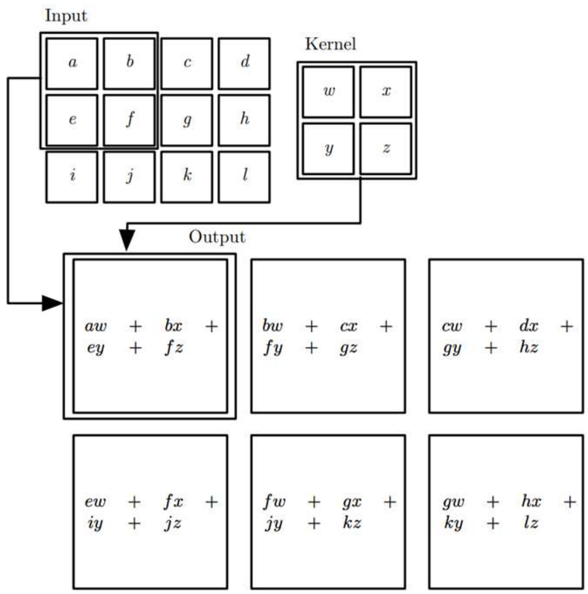
24

- Most machine learning libraries implement cross-correlation while calling it convolutions.
- Cross correlation is defined as:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i+m, j+n)$$
$$= \sum_m \sum_n I(i+m, j+n)K(m,n)$$

- The logic behind the above is that usually, we learn the kernel and thus it does not matter if it is flipped or not.

Convolution Example



Convolution Example

- Exact formula for the convolution layer:
  - Each filter generates an activation map.

$$\begin{bmatrix} 0 & 3 & 0 & 1 \\ 2 & 0 & 1 & 2 \\ 3 & 0 & 0 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix} \star \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 7 & -2 & 5 \\ 4 & 1 & 3 \\ -2 & 4 & 6 \end{bmatrix}$$

Input image

One filter

One activation map

Convolution

27

- Exact formula for the convolution layer:
  - Each filter generates an activation map.

$(0)(-1) + (3)(1) + (2)(2) + (0)(1) = 7$

$$\begin{bmatrix} 0 & 3 & 0 & 1 \\ 2 & 0 & 1 & 2 \\ 3 & 0 & 0 & 2 \\ 0 & 1 & 2 & 0 \end{bmatrix} \star \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} \boxed{7} & -2 & 5 \\ 4 & 1 & 3 \\ -2 & 4 & 6 \end{bmatrix}$$

Input image

One filter

One activation map

28

Questions?

Acknowledgement: Prof. W. L. Hamilton, Fei-Fei Li, Ranjay Krishna, Danfei Xu, Justin Johnson, Xavier Bresson