Deep Learning

- Convolutional Neural Network -

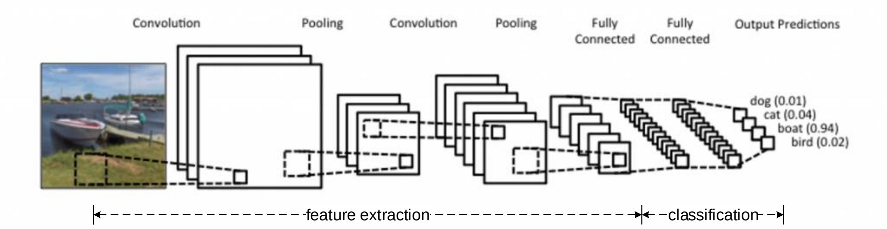
2019 - 2020

Ando Ki, Ph.D.
adki@future-ds.com

Table of contents

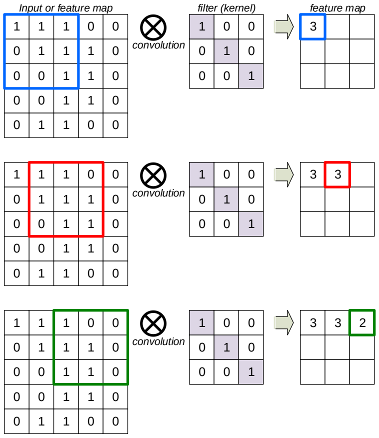
CNN: Convolutional Neural Network

- CNN is a neural network that uses convolution in place of general matrix multiplication in at least one of their layers.
- General form of CNN (Convolutional Neural Network) for image classification
  - Feature extraction
    - Convolution
    - Pooling (sub-sampling)
  - Classification
    - Regression



CNN: convolution

- Convolution is a matrix multiplication



Input or feature map ⊗ *convolution* *filter (kernel)* → *feature map*

1	1	1	0	0
0	1	1	1	0
0	0	1	1	0
0	0	1	1	0
0	1	1	0	0

1	0	0
0	1	0
0	0	1

3		

convolution

1	1	1	0	0
0	1	1	1	0
0	0	1	1	0
0	0	1	1	0
0	1	1	0	0

1	0	0
0	1	0
0	0	1

3	3	

convolution

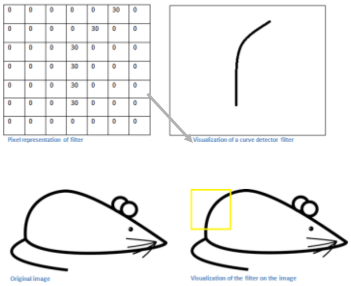
1	1	1	0	0
0	1	1	1	0
0	0	1	1	0
0	0	1	1	0
0	1	1	0	0

1	0	0
0	1	0
0	0	1

3	3	2

When the value is large after convolution, it means there is a feature about it.

- It can be seen as a feature extractor



0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Visualization of a curve detector filter

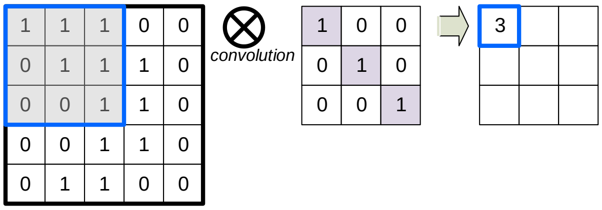
Original image

Visualization of the filter on the image

https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/

CNN: convolution padding

- No padding
  - **Valid padding**

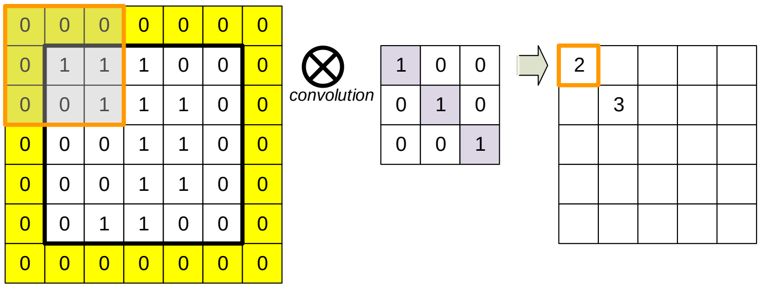


1	1	1	0	0
0	1	1	1	0
0	0	1	1	0
0	0	1	1	0
0	1	1	0	0

1	0	0
0	1	0
0	0	1

3		

- Zero padding
  - **Same padding** due to input and out have the same dimensions.



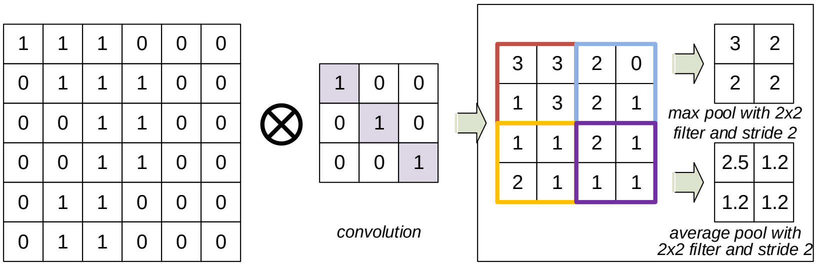
0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	0	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	0	0	0	0	0	0

1	0	0
0	1	0
0	0	1

2				
	3			

CNN: Pooling

- Pooling, i.e., sub-sampling
  - Max pooling
  - Average pooling



1	1	1	0	0	0
0	1	1	1	0	0
0	0	1	1	0	0
0	0	1	1	0	0
0	1	1	0	0	0
0	1	1	0	0	0

1	0	0
0	1	0
0	0	1

3	3	2	0
1	3	2	1
1	1	2	1
2	1	1	1

3	2
2	2

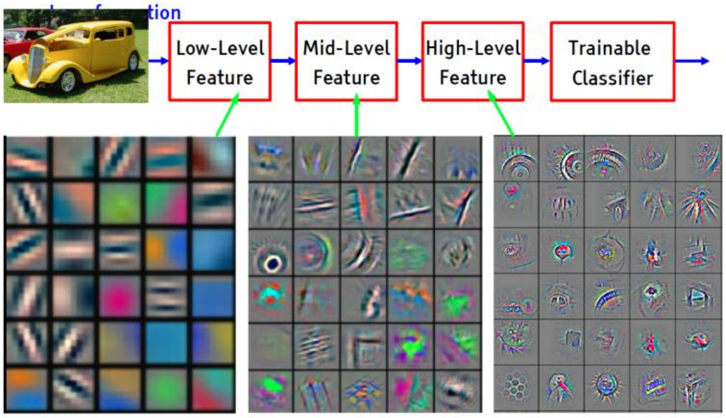
2.5	1.2
1.2	1.2

How to choose filters

- With CNN/ConvNet the goal is to learn the filters; you don't actually design these filters (or kernels). They will be learned during training as long as the training converges.
- Initializing the these filter parameters with good defaults before starting the training is key to convergence especially in very deep networks.

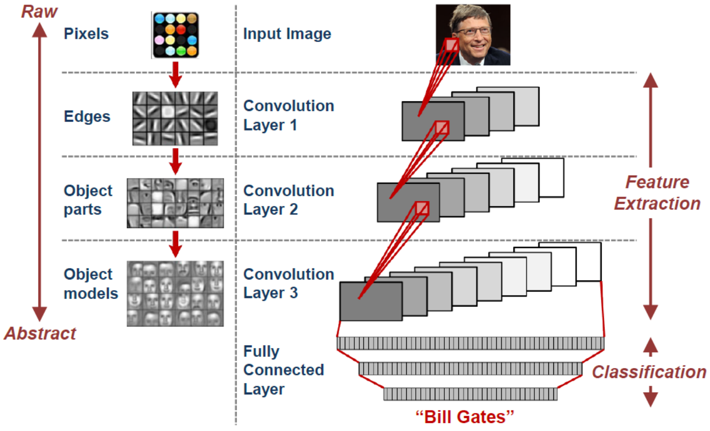
- Convolution filters can be initialized in one of the following ways.
  - 1. Randomly assigning weights for the different filters.
  - 2. Handcrafting the weights of the different filters to detect specific features during convolution.
  - 3. Learning filter weights using unsupervised training schemes.

Deep learning: Learning Hierarchical Representations



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

CNN abstraction



Raw

Abstract

Pixels

Edges

Object parts

Object models

Input Image

Convolution Layer 1

Convolution Layer 2

Convolution Layer 3

Fully Connected Layer

Feature Extraction

Classification

“Bill Gates”

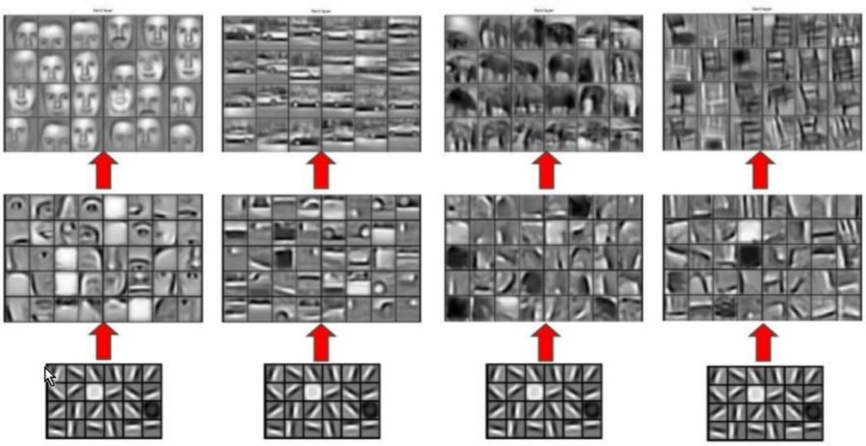
CNN abstraction

Faces

Cars

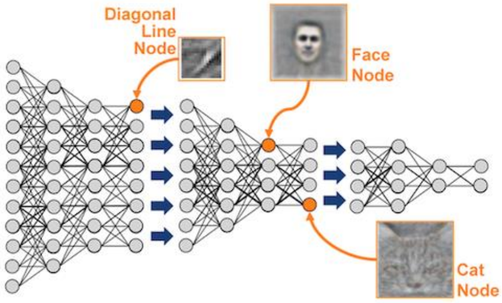
Elephants

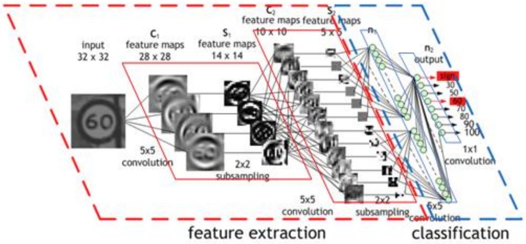
Chairs



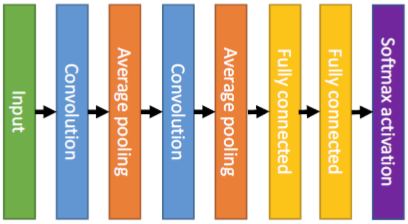
Convolutional deep belief networks for scalable unsupervised learning of hierarchical representation”, Lee et al., 2012

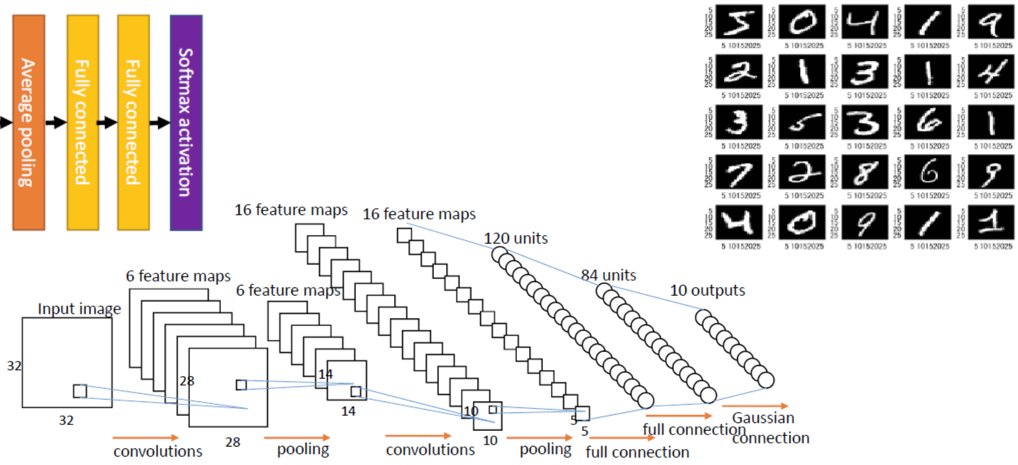
CNN examples





CNN examples: LeNet-5 (1989)



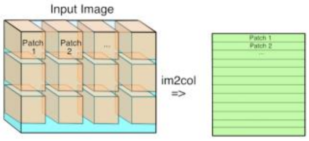


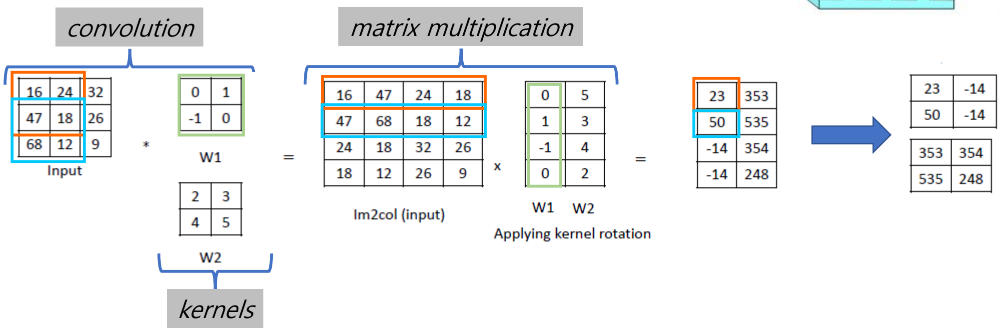
Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. NIPS 1989.

Table of contents

Convolution by matrix multiplication

- 1-channel 2D convolution example: 3x3 input with 2x2 two kernel case
  - ▶ Refer to ‘im2col’





Input

16	24	32
47	18	26
68	12	9

W1

0	1
-1	0

W2

2	3
4	5

Im2col (input)

16	47	24	18
47	68	18	12
24	18	32	26
18	12	26	9

W1 W2

0	5
1	3
-1	4
0	2

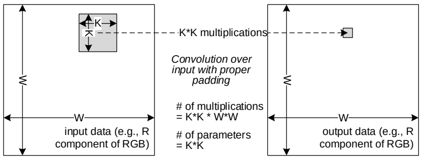
23	353
50	535
-14	354
-14	248

23	-14
50	-14

353	354
535	248

Standard convolution

- Single channel with single kernel

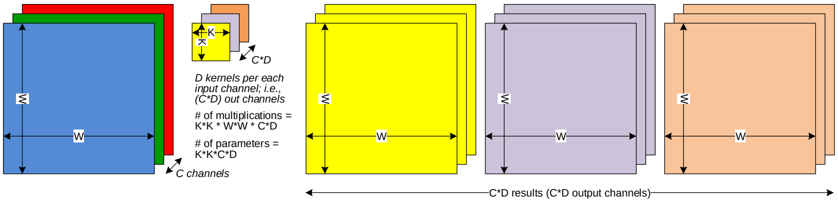


Convolution over input with proper padding

# of multiplications = K*K * W*W

# of parameters = K*K

- Multiple channels and kernels



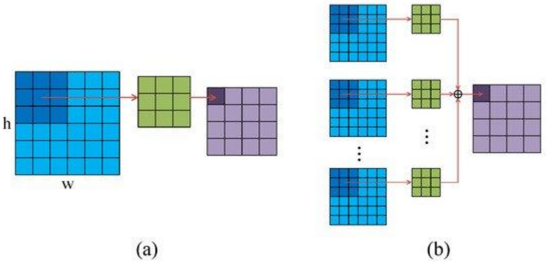
*D kernels per each input channel; i.e., (C*D) out channels*

# of multiplications = K*K * W*W * C*D

# of parameters = K*K*C*D

Convolution for multi-channel

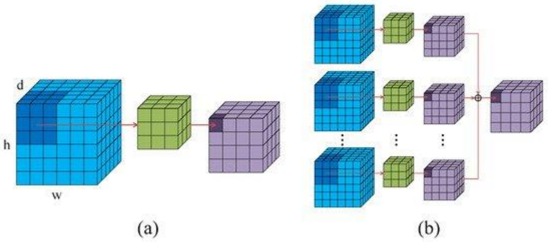
- 2D convolution
  - single and multi-channel



(a)

(b)

- 3D convolution
  - single and multi-channel

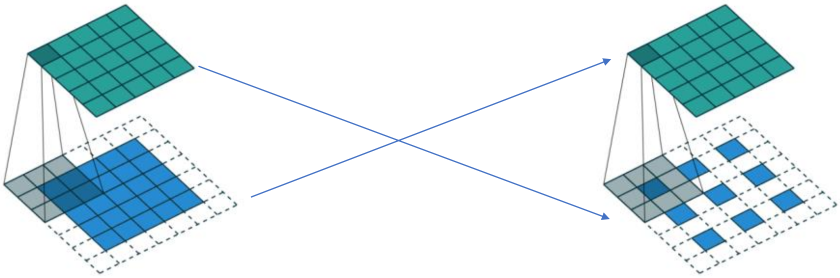


(a)

(b)

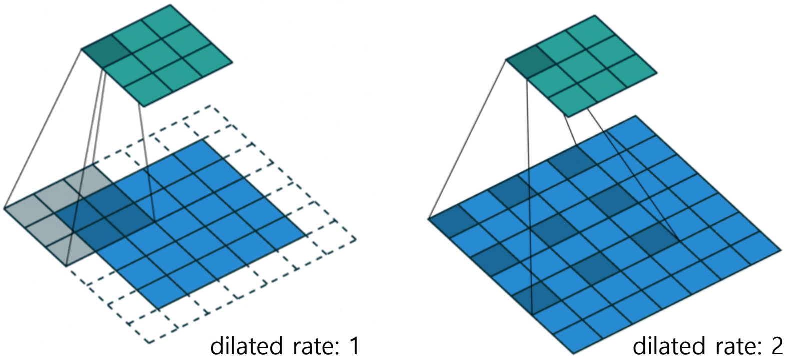
Convolution and deconvolution

- Standard convolution (discrete convolution)
  - to extract feature map
- Standard deconvolution
  - known as **transposed convolution**
  - to reconstruct original image
  - a reverse operation of convolution



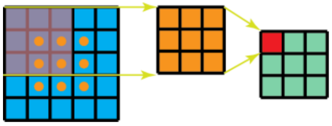
Dilated convolution (atrous convolutions)

- Similar with deconvolution used in real-time segmentation
- smaller kernel for wider view
- not reverse operation (i.e, not reconstruction of original image)



Separable convolution: spatially separable

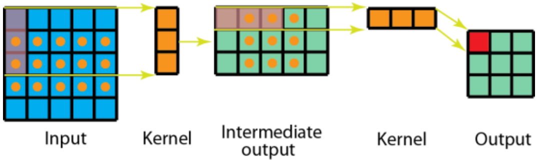
- standard convolution
  - multiplications
    - ➲ K*K * W*W



- kernel divided

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

- spatially separable convolution
  - multiplications
    - ➲ K*W*W + K*W*W
      - 2/K ratio comparing to standard convolution

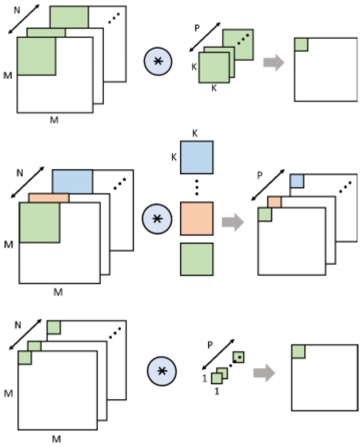


$$\begin{bmatrix}3&6&9\\4&8&12\\5&10&15\end{bmatrix} = \begin{bmatrix}3\\4\\5\end{bmatrix} \times \begin{bmatrix}1&2&3\end{bmatrix}$$

Although spatially separable convolutions save cost, it is rarely used in deep learning. One of the main reason is that *not all kernels can be divided into two, smaller kernels*..

Separable convolution: depthwise separable

- Standard convolution and depthwise separable (no channel-wise conv)

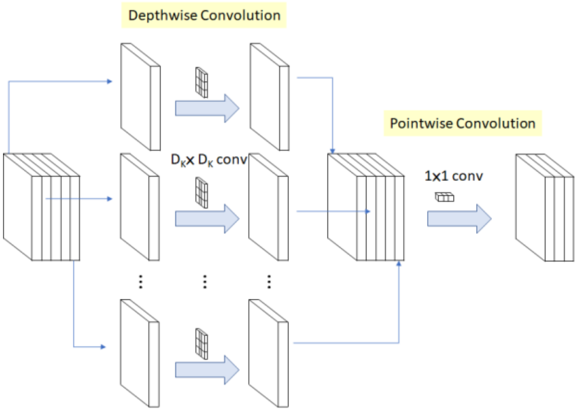


Standard conv.

Depth-wise conv.

Point-wize conv.

- MobileNet case



Separable convolution: depthwise separable

- Standard convolution
  - uses kernels of a number of output channels
- Depth wise separable
  - Depthwise convolution: filtering stage
    - uses kernels of a number of input channels
  - Pointwise convolution: combining state
    - uses kernels of a number of output channels