Basics of Convolutional Neural Network (CNN)

EE 5179: Deep learning for Imaging

Instructor: Kaushik Mitra

2. Convolutional Neural Networks (CNNs)

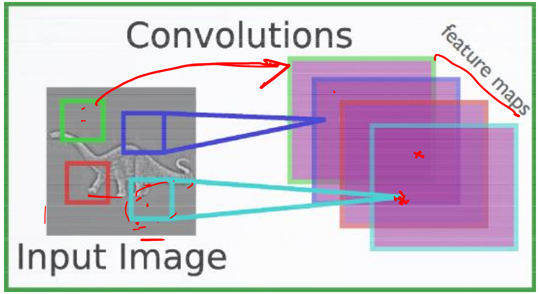
CNNs vs MLPs

- Naively using MLP to classify 224x224x3 (~ 3 x 40,000) typical ImageNet image -> parameter explosion
  - ❑ Doesn't exploit local spatial information
- Can we build special neural nets for images exploiting
  - ❑ 2D topology of pixels
  - ❑ Achieve invariance to translation?

Convolutional networks leverage these ideas,

- ❑ Local connectivity
- ❑ Parameter sharing
- ❑ Pooling/ Subsampling
- ❑ ReLu (rectifier) nonlinearity

Category: tiger
ImageNet



*slide courtesy, Hugo Larochelle course on Neural networks

Topics

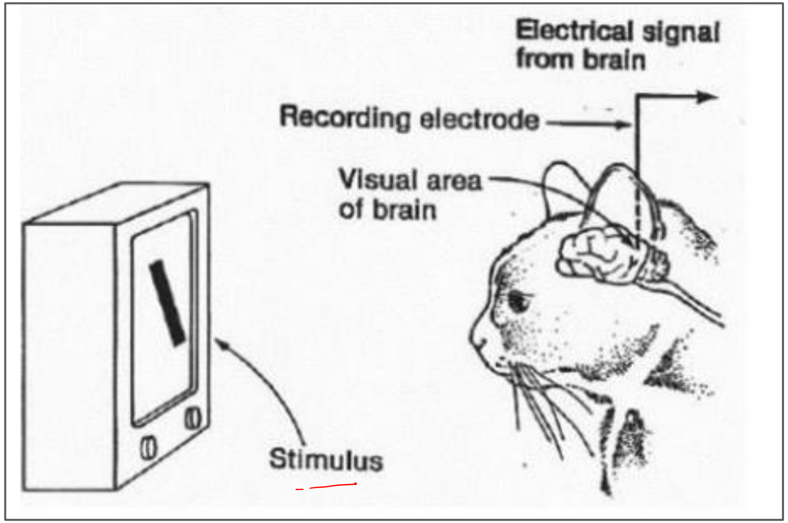
General and biological motivation.

CNNs over fully connected networks.

Different layers in architecture (pooling, relu, etc.)

Biological motivation - Mammalian vision system.

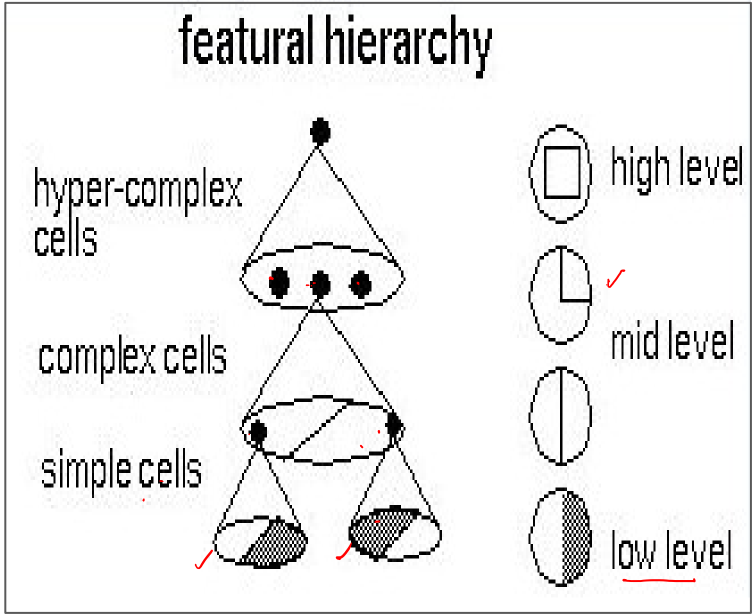
Hubel and Wiesel (1959)

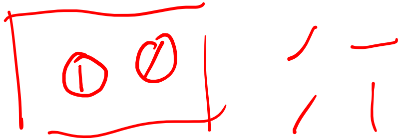


Experimental setup

Suggested a 'hierarchy' of feature detectors in the mammalian visual cortex.

Biological motivation - Mammalian vision system.





Simple cells:

1. Activity characterized by a linear function of the image.
2. Operates in a spatially localized (SL) receptive field.
3. Each set responds to edges of different orientation.

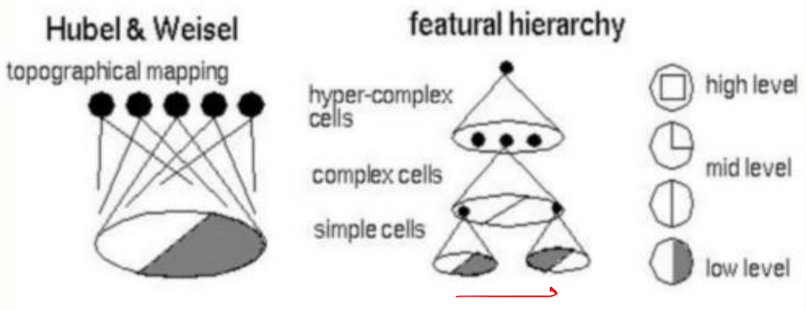
Complex cells:

1. Operates in large SL receptive field
2. Receive input from lower level simple cells.
3. Acts as motion detectors

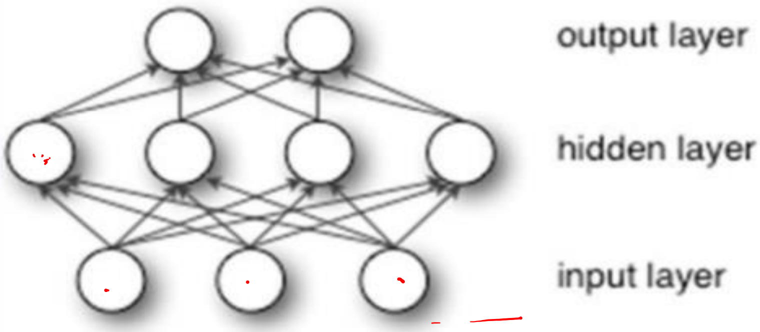
Hyper-complex cells:

1. Larger receptive field
2. Receive input from lower level complex cells.
3. Acts as angle detectors

Biological motivation - Mammalian vision system.



Hubel and Weisel's architecture



Multi-layer Neural Network
- A *non-linear* classifier

Biological motivation - Grandmother cell

The grandmother cell is a hypothetical neuron that represents a complex but specific concept or object proposed by cognitive scientist Jerry Letvin in 1969.

But this hypothesis is currently being doubted since the number of objects/concepts is larger than number of neurons.

Biological motivation - Biological NN to Artificial NN.

Neocognitron [Fukushima, Biological Cybernetics 1980]

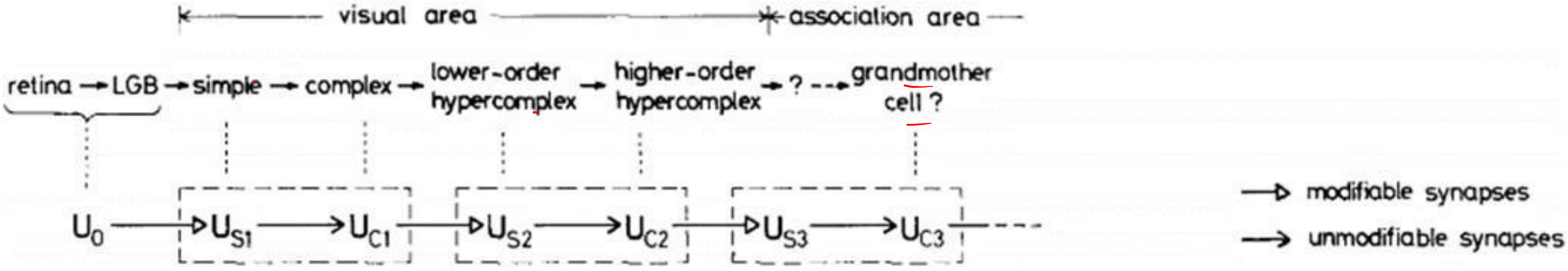
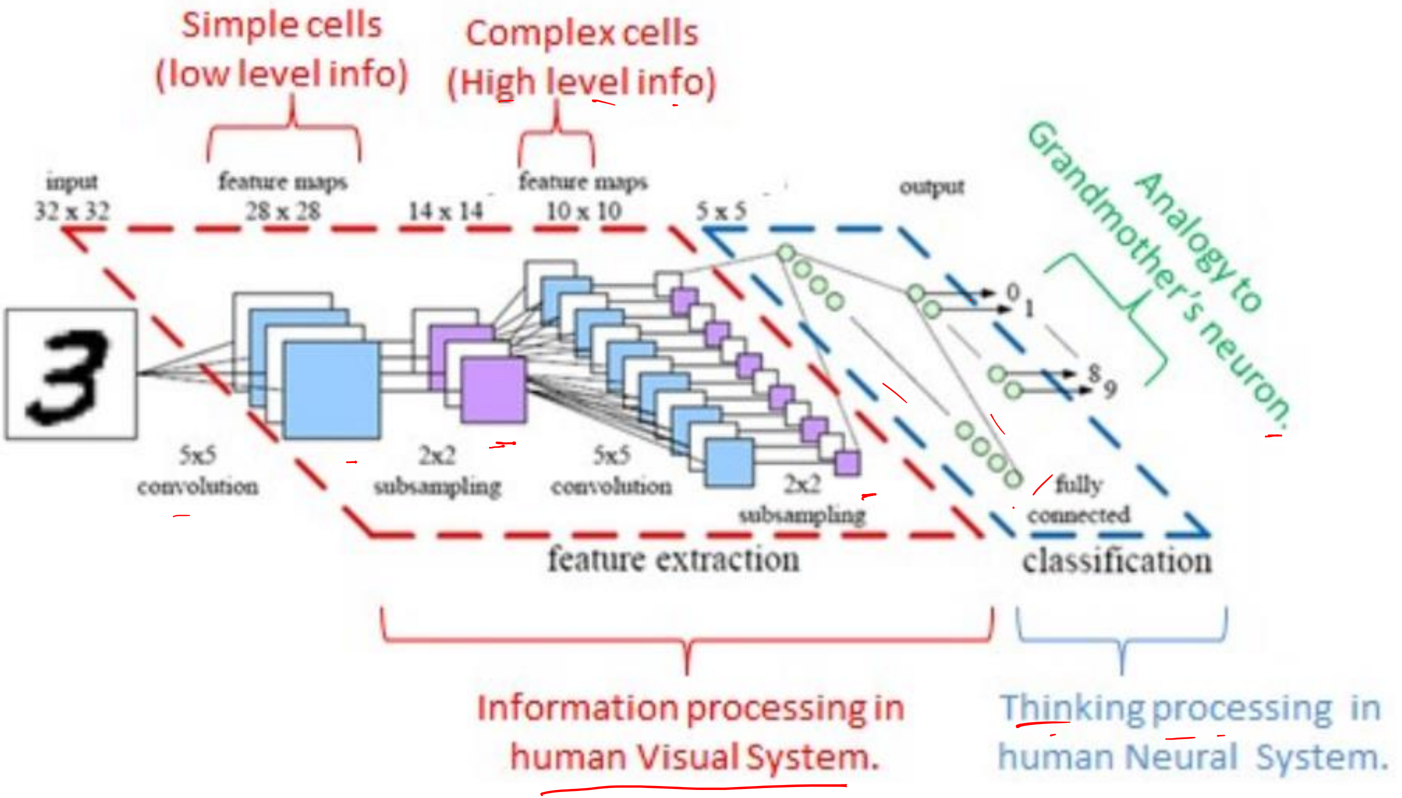


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

1. But neuroscience has told us relatively less about how to train networks.
2. Neocognitron used layer-wise unsupervised pretraining algorithm.

Biological motivation - CNN.

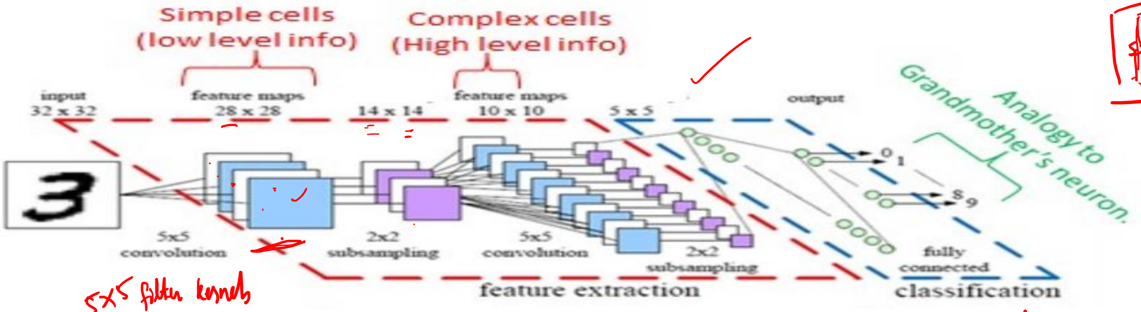
Back-propagation [Lang and Hinton, 1988], and modern CNN [LeCun *et al*., 1989]

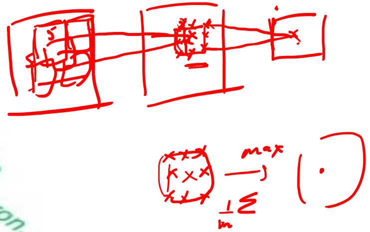


CNN proposed by LeCun *et al.* for document recognition.

Simple cells and low-level filters in a CNN

Marčelja, S. [1980] suggests that simple cells in visual cortex can be modeled as **Gabor filters**

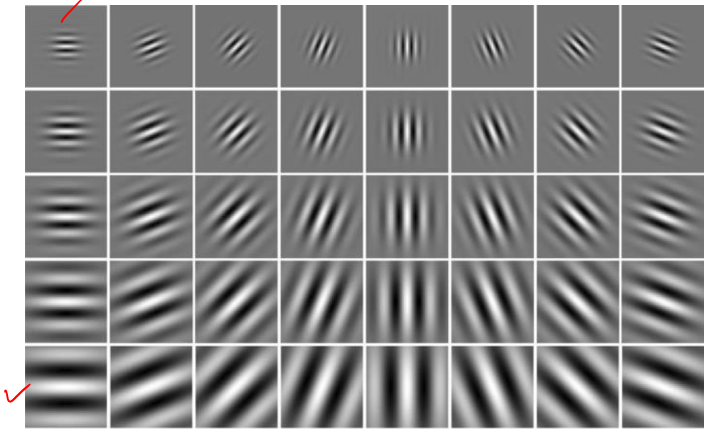




5x5 filter kernels

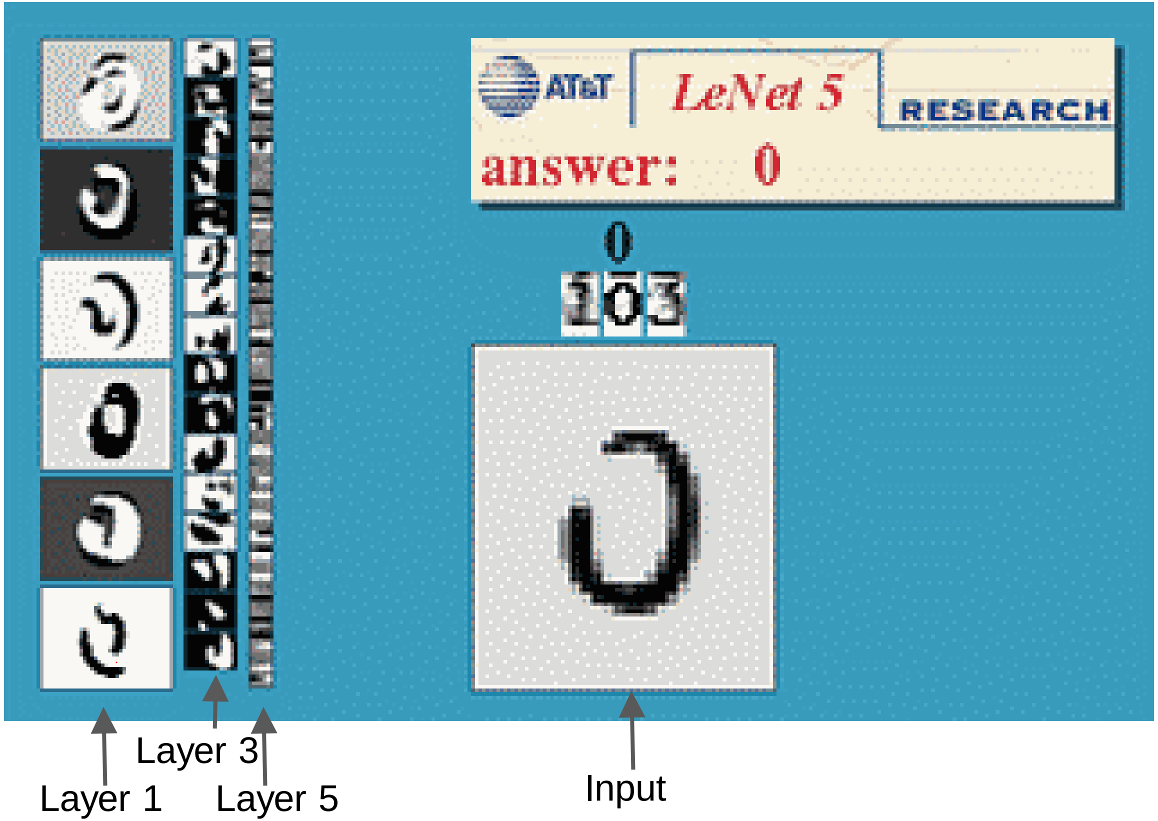
sinusoid multiplied by Gaussian function

Low-level learnt filters of CNN (from Alexnet, 2012)



Gabor filters

CNN for document recognition [LeCun *et al*., 1989].



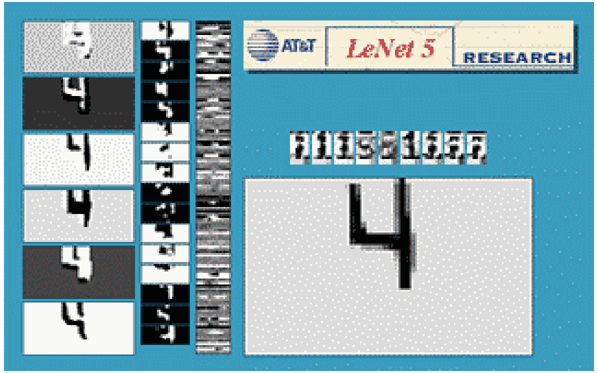
All images are 28x28 grayscale.

60k training examples.

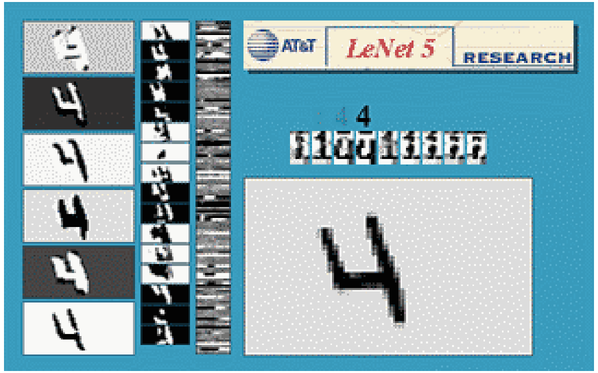
10k test examples

Output value is integer from 0-9

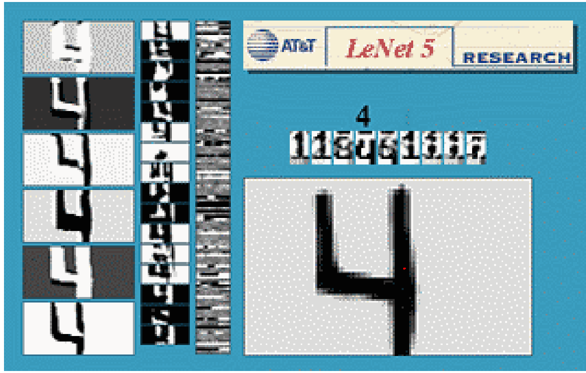
CNN for document recognition [LeCun *et al*., 1989]. LeNet



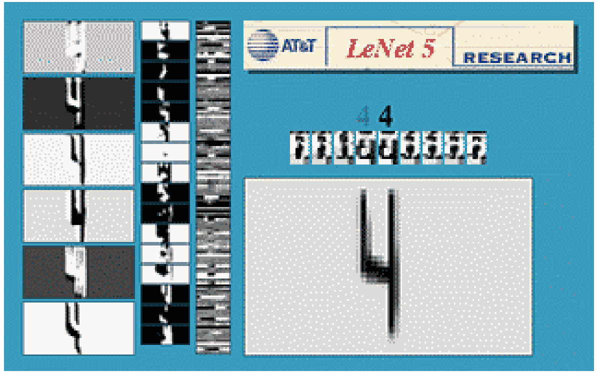
Translation invariance



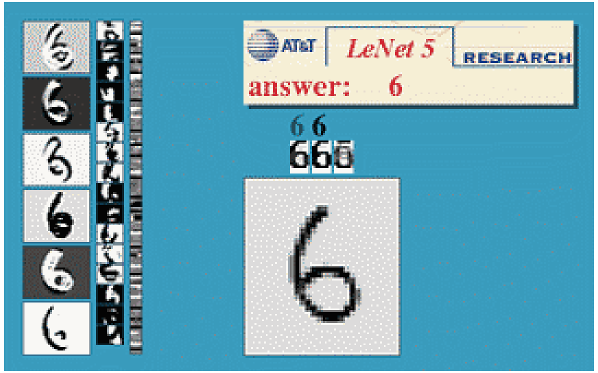
Rotation invariance



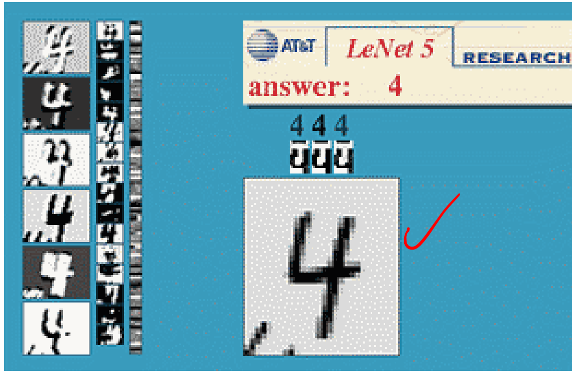
Scale invariance



Squeeze invariance

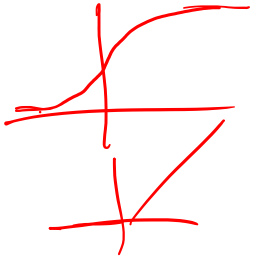


Stroke-width invariance



Noise invariance

Then why DL didn't take-off in 90's?



1. ***Limited*** *big data availability*
2. ***Limited*** *computational power to crunch data*

Why DL is trending now?

Big data availability

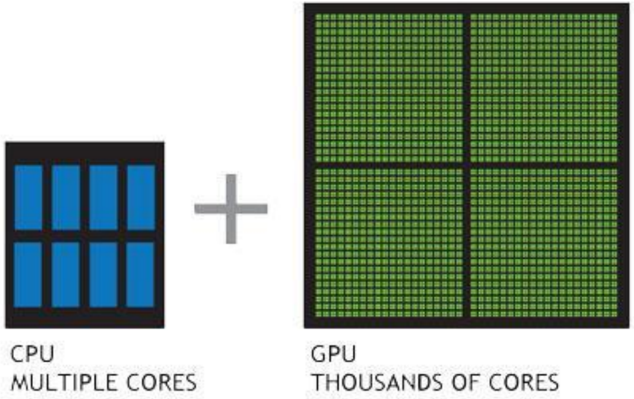
One trillion images.

350 million images uploaded **per day**.

100 hrs of video uploaded **per minute**.

2.5 Petabytes data **every minute**.

Computational power to crunch data



Parallel processing units - GPUs

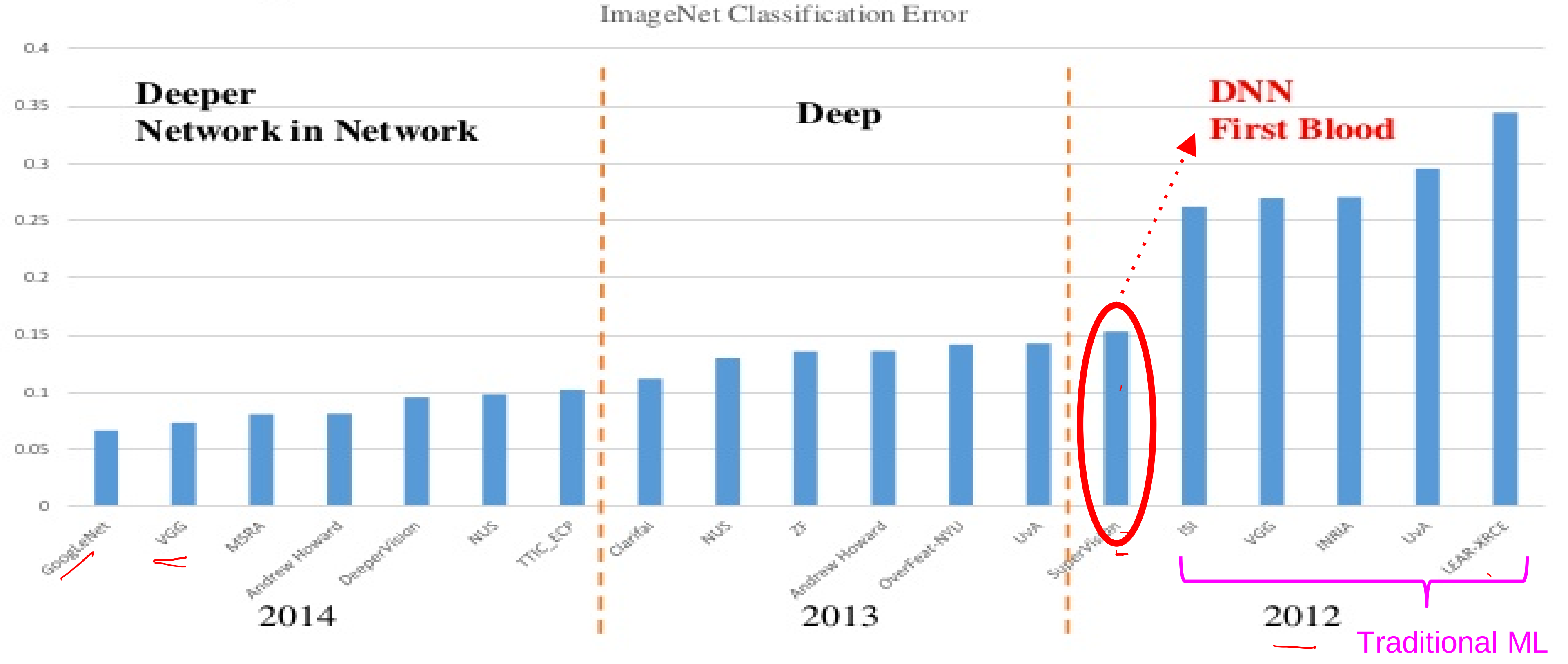
When/how was deep-learning reclaimed?

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

ImageNet Classification

- **1000** categories and **1.2** million training images



Li Fei-Fei: ImageNet Large Scale Visual Recognition Challenge, 2014 http://image-net.org/

Topics

General and biological motivation.

CNNs over fully connected networks.

Different layers in architecture (pooling, relu, etc.)

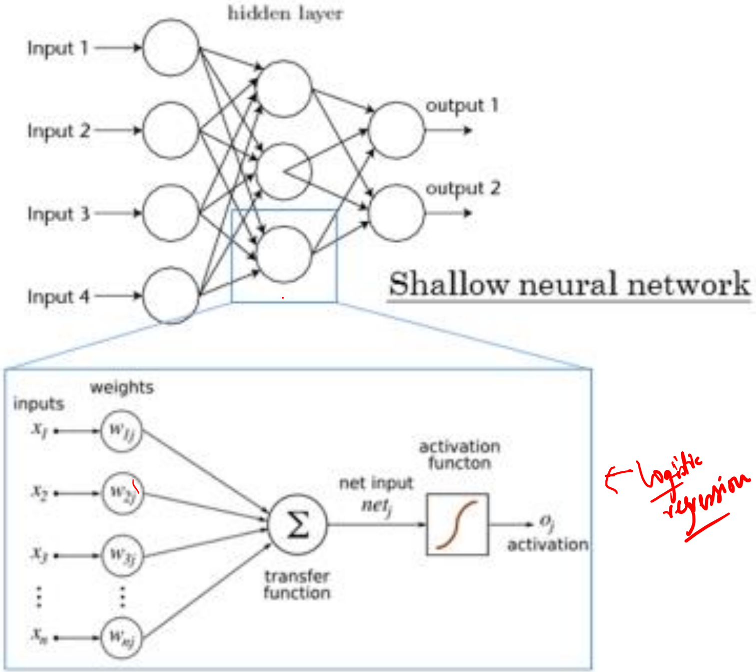
Topics

General and biological motivation.

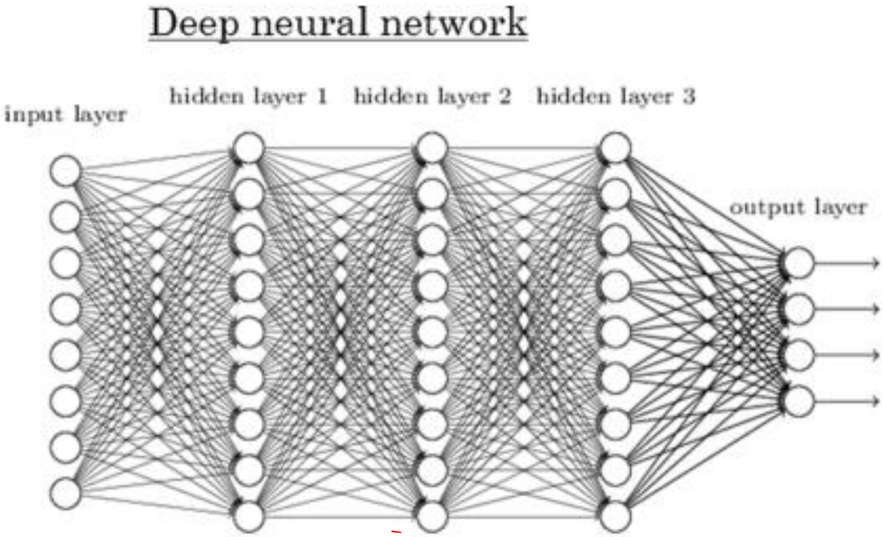
CNNs over multi-layer neural networks.

Different layers in architecture (pooling, relu, etc.)

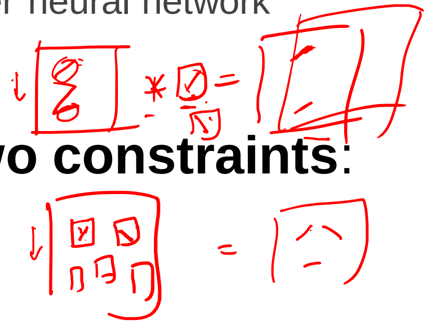
CNNs over Multi-layer neural networks (MLNN)



← logistic regression



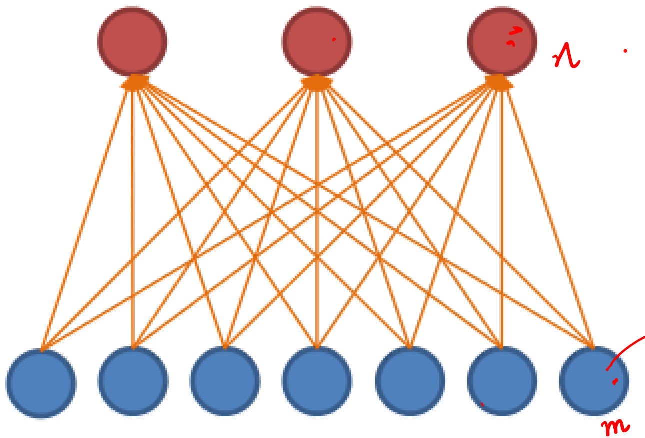
Multi-layer neural network



CNNs are **multi-layer neural network with two constraints**:

1. Local connectivity ✓
2. Parameter sharing ✓ ⇒ spatial invariance

CNN: Local connectivity (LC)

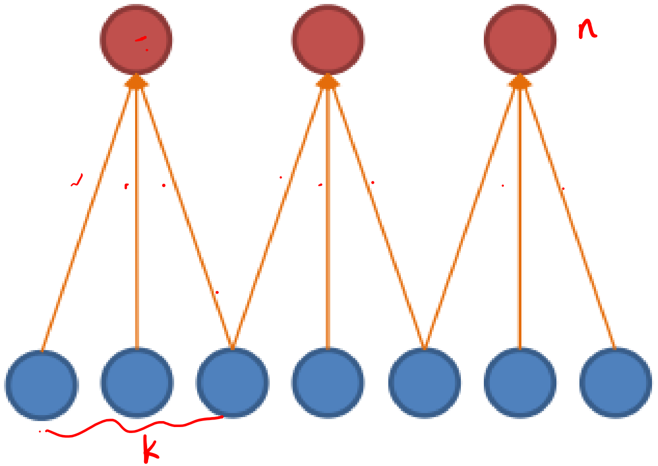


Hidden layer (3 nodes)

$h_{3\times1} = \underset{3\times7}{W}\, x_{7\times1}$

$x_{7\times1}$

Input layer (7 nodes)



MLNN (7 X 3 = 21 parameters)

MLNN-LC (3 X 3 = 9 parameters)
2.3X runtime and storage efficient.

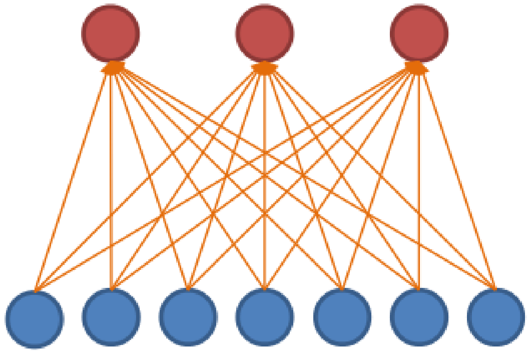
In general for a level with *m* input and *n* output nodes and CNN-local connectivity of *k* nodes *(k<m)*:

MLNN have
1. *m x n* parameters to store.
2. O(*m x n*) runtime

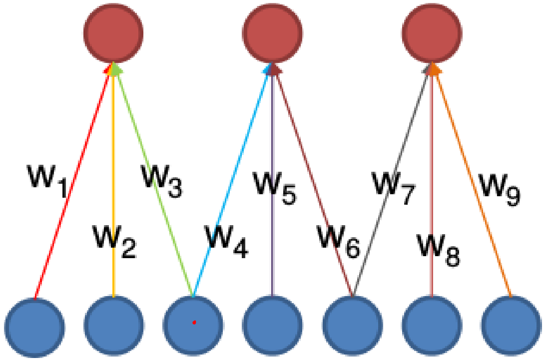
Gain $= \frac{m}{k}$

MLNN-LC have:
1. *k x n* parameters to store.
2. O(*k x n*) runtime

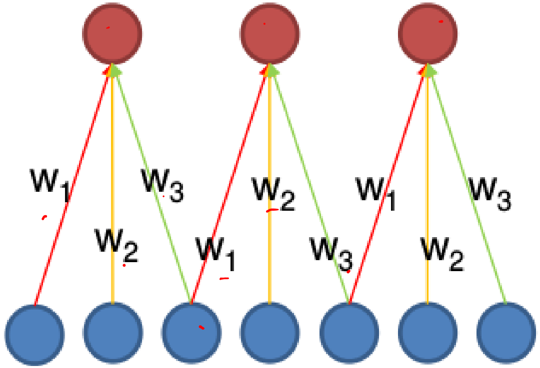
CNN: Parameter sharing (PS)



MLNN (21 parameters)



MLNN-LC (3 X 3 = 9 parameters)
2.3X runtime and storage efficient.



MLNN-LC-PS (3 parameters)
2.3X faster,
& 7X storage efficient.

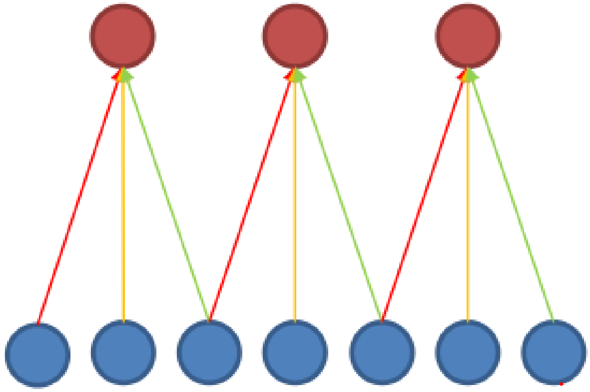
In general for a level with m input and n output nodes and CNN-local connectivity of k nodes $(k<m)$:

MLNN have
1. $m \times n$ parameters to store.
2. $O(m \times n)$ runtime

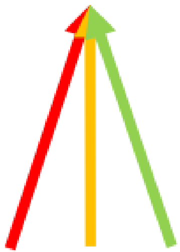
MLNN-LC have:
1. $k \times n$ parameters to store.
2. $O(k \times n)$ runtime

MLNN-LC-PS have:
1. k parameters to store.
2. $O(k \times n)$ runtime

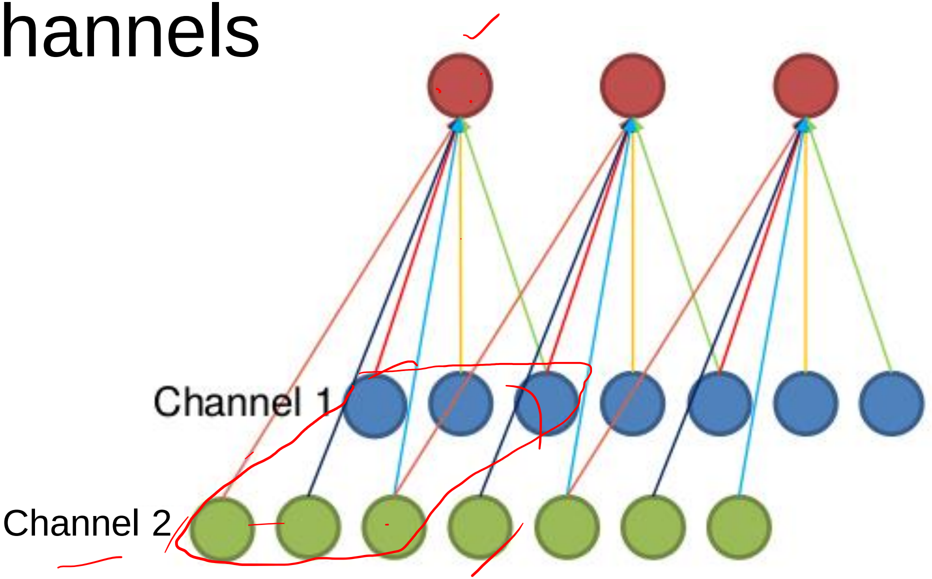
CNN with multiple input channels



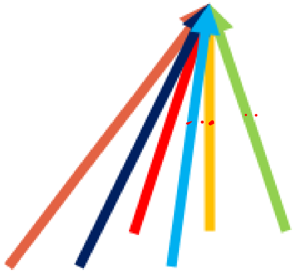
Single input channel



Filter weights

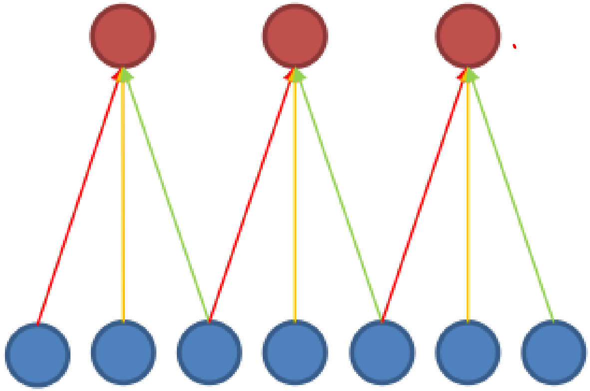


Two input channels

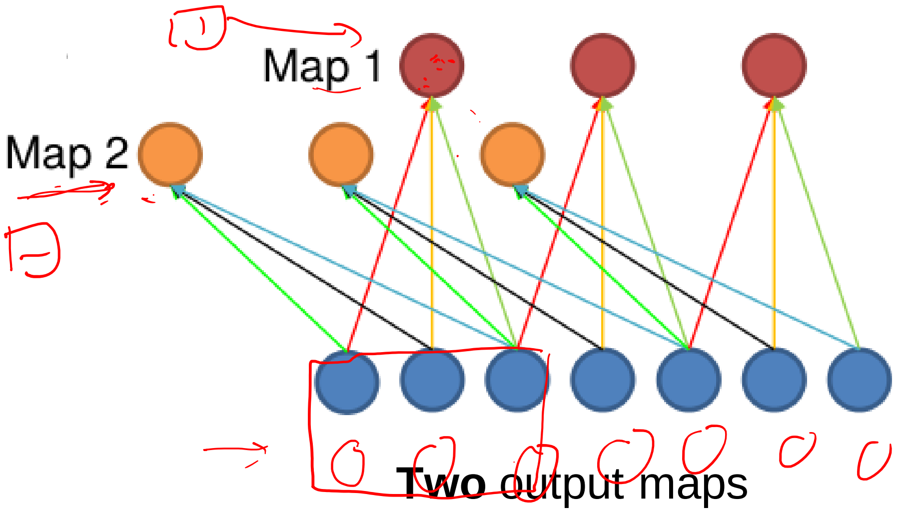


Filter weights

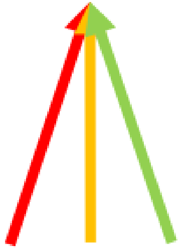
CNN with multiple output maps



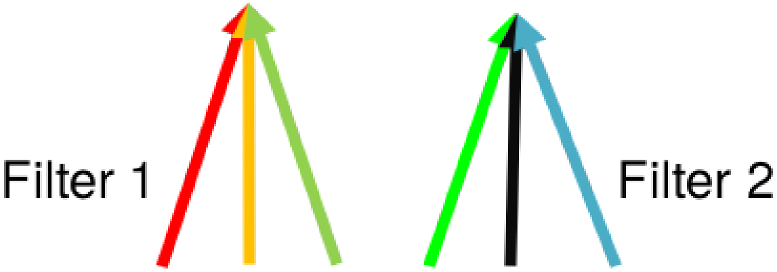
Single output map



Two output maps

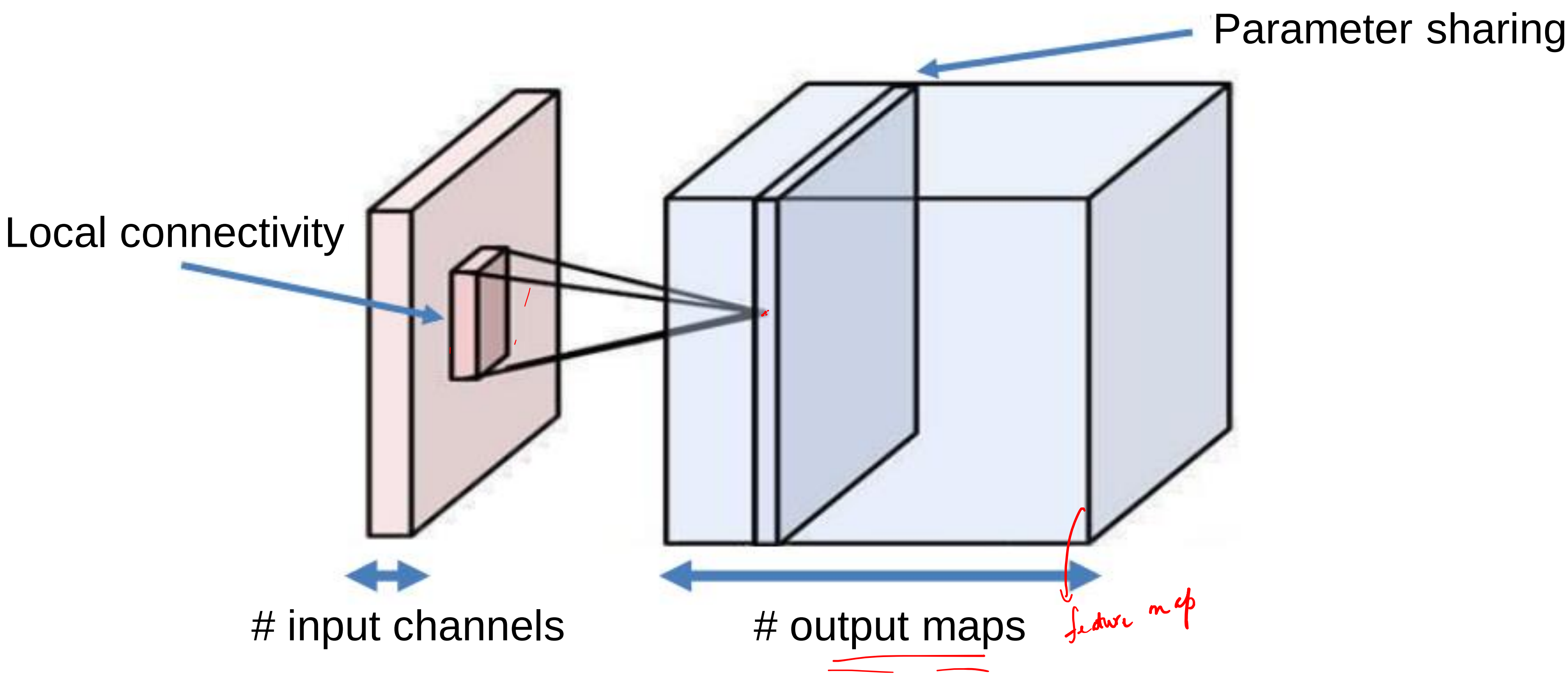


Filter weights



Filter weights

A generic level of CNN



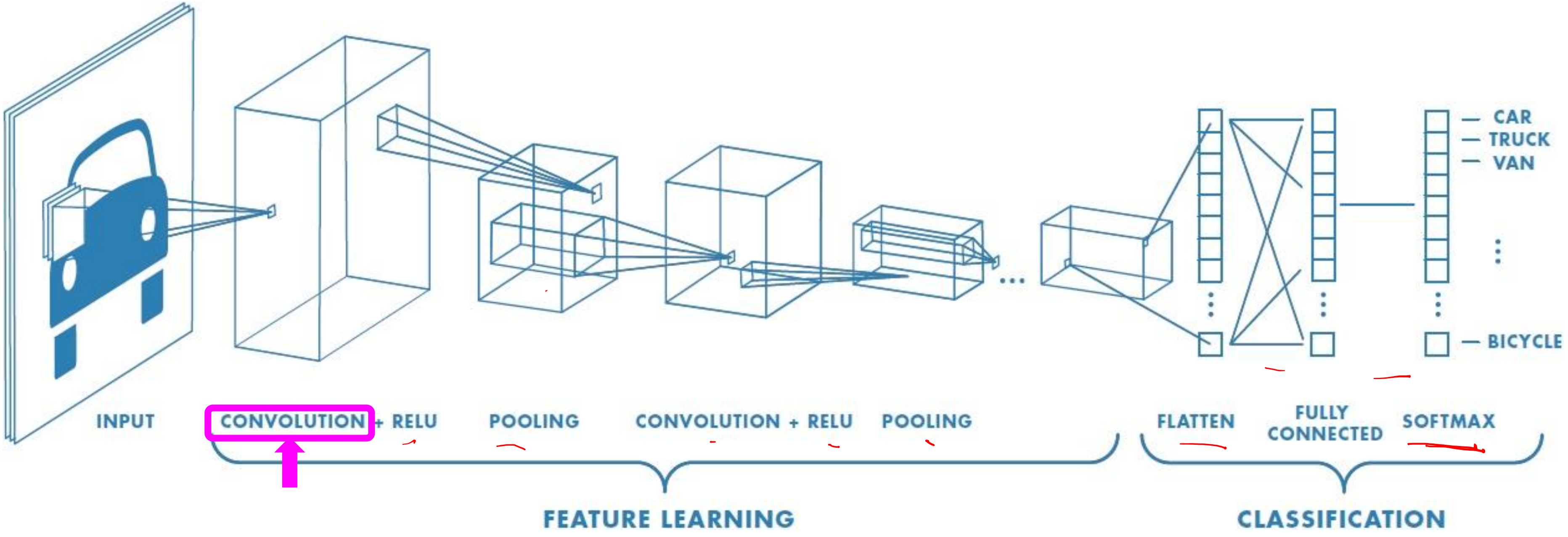
Topics

General and biological motivation.

CNNs over multi-layer neural networks.

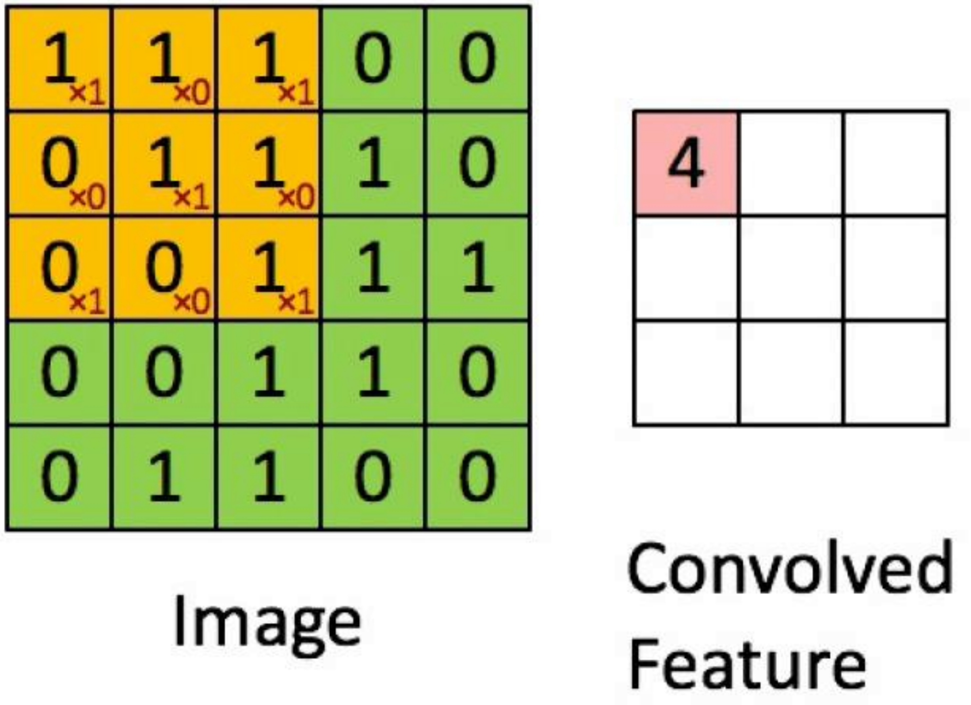
Different layers in CNN architecture (pooling, relu, etc.)

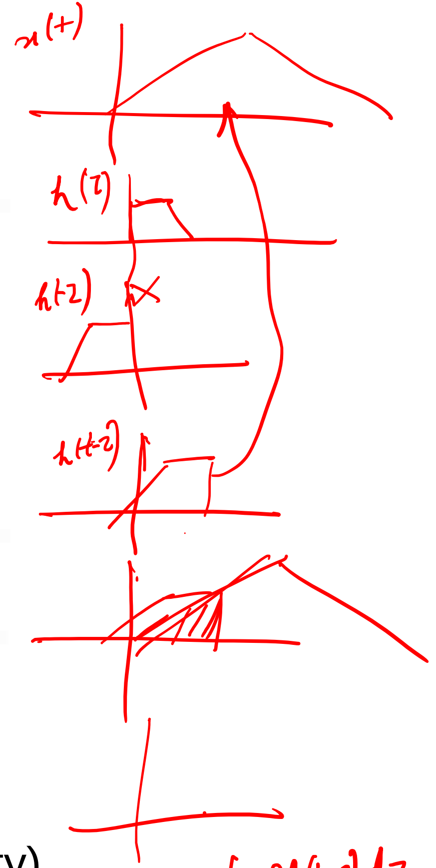
Different layers of CNN architecture



CNN: Convolutional layer

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



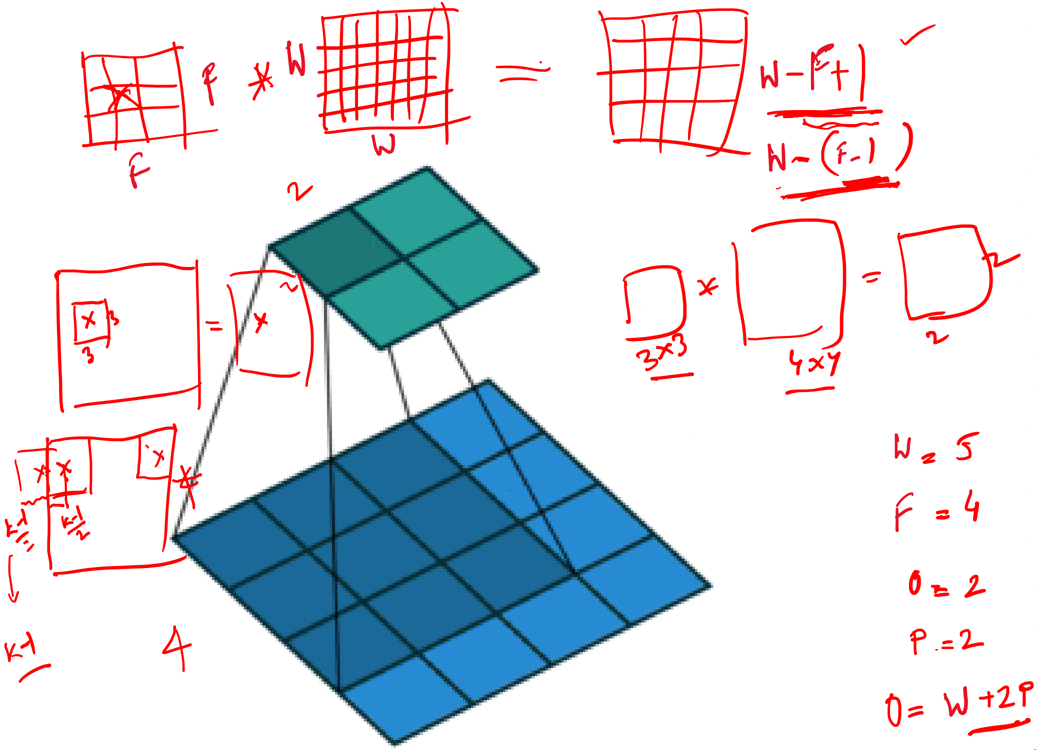


$$y(t) = \int x(\tau)\,h(t-\tau)\,d\tau$$

1. To reduce the number of weights (through local connectivity).
2. To provide spatial invariance (through parameter sharing).

Hyper parameters for convolutional layer.

1. Zero padding (to control input size spatially.)



Without padding (i.e., [0,0])

$W = 5$

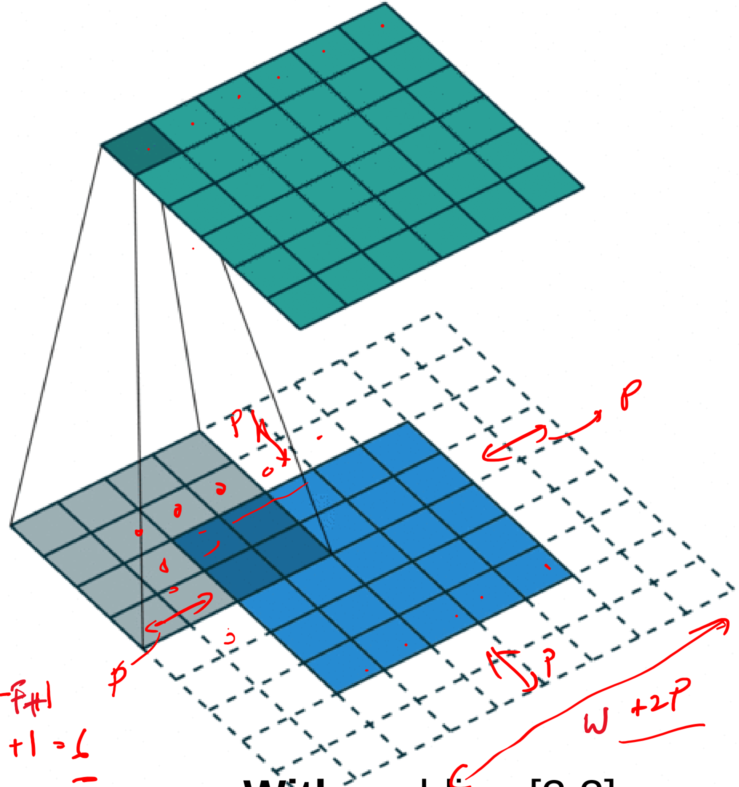
$F = 4$

$O = 2$

$P = 2$

$O = W + 2P - F + 1$

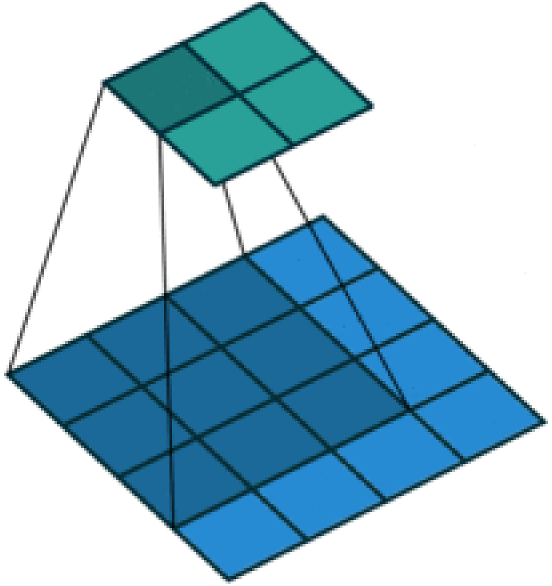
$= 9 - 4 + 1 = 6$



With padding [2,2]

Hyper parameters for convolutional layer.

2. Stride (to produce smaller output volumes spatially.)



Without stride (i.e., [1,1])

$W = 5$

$F = 3$

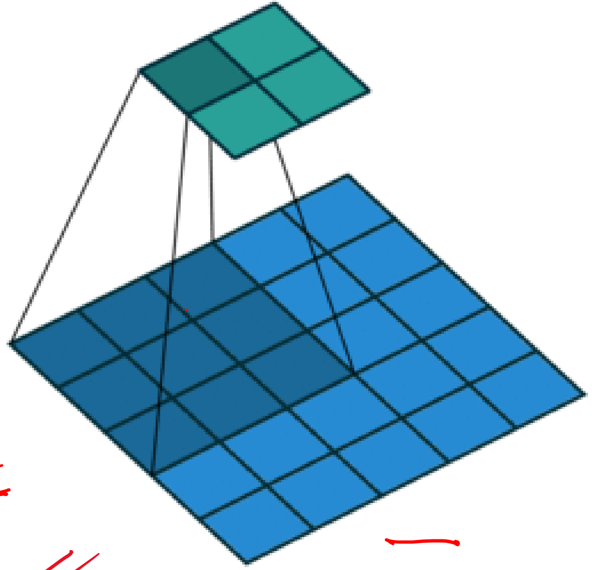
$S = 2$

$O = 2$

$O = \frac{W - F + 1}{S}$

$\frac{5 - 3 + 1}{2} = 1.5$

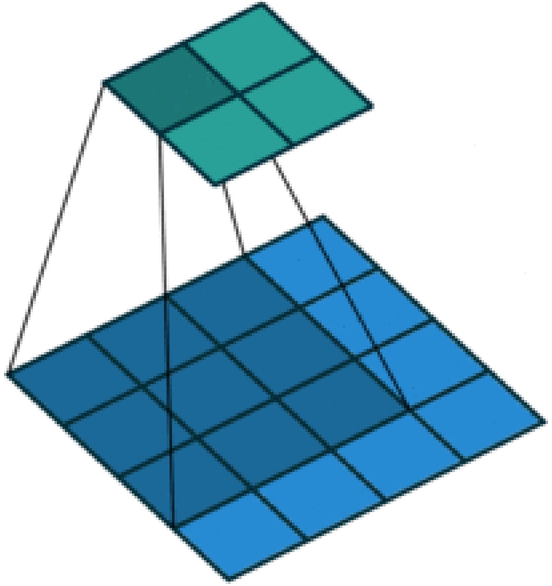
$O = \left(\frac{W - F}{S}\right) + 1$



With stride [2,2]

Hyper parameters for convolutional layer.

Both padding and stride



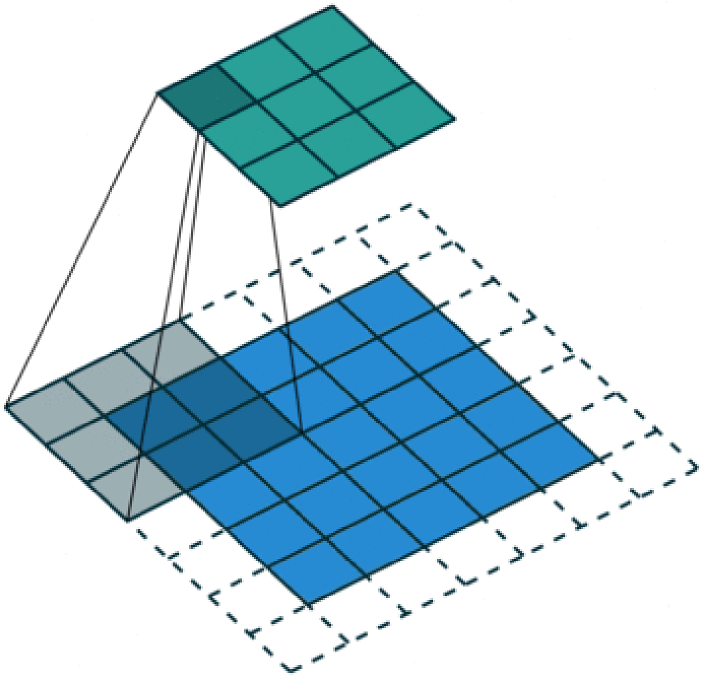
$W \rightarrow$ i/p width

$F \rightarrow$ kernel size

$P \rightarrow$ Paddi

$S \rightarrow$ stride

$$O = \frac{W + 2P - F}{S} + 1$$



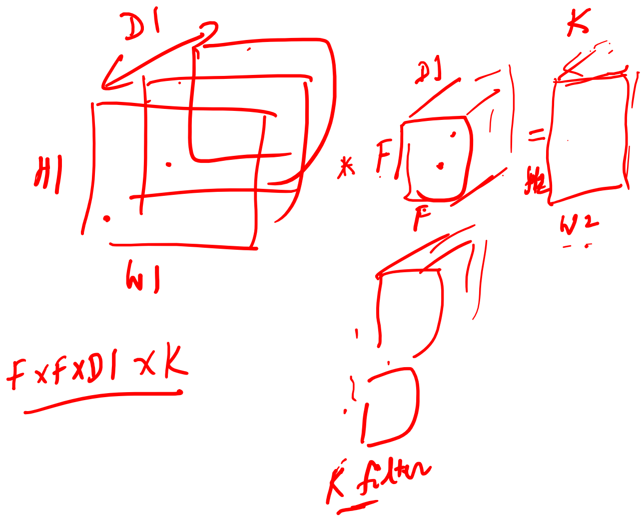
Without padding and stride

With padding [1,1] & stride [2,2]

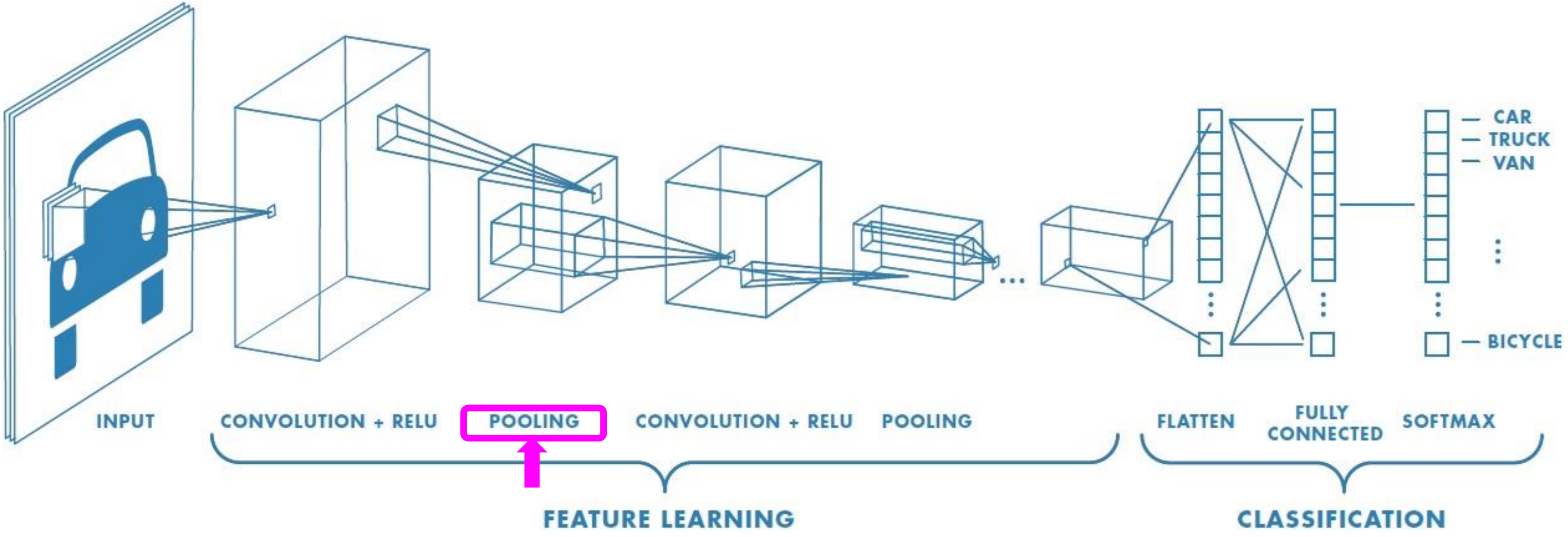
CONVOLUTIONAL LAYER

1. Accepts a volume of size ***W1* X *H1* X *D1*.**
2. Requires four hyperparameters:
   a. Number of filters ***K***
   b. their spatial extent ***F***
   c. their stride ***S***
   d. the amount of zero padding ***P***
3. Produces an output volume of size ***W2* X *H2* X *D2*** where:
   W2=(W1−F+2P)/S+1, H2=(H1−F+2P)/S+1, D2=K

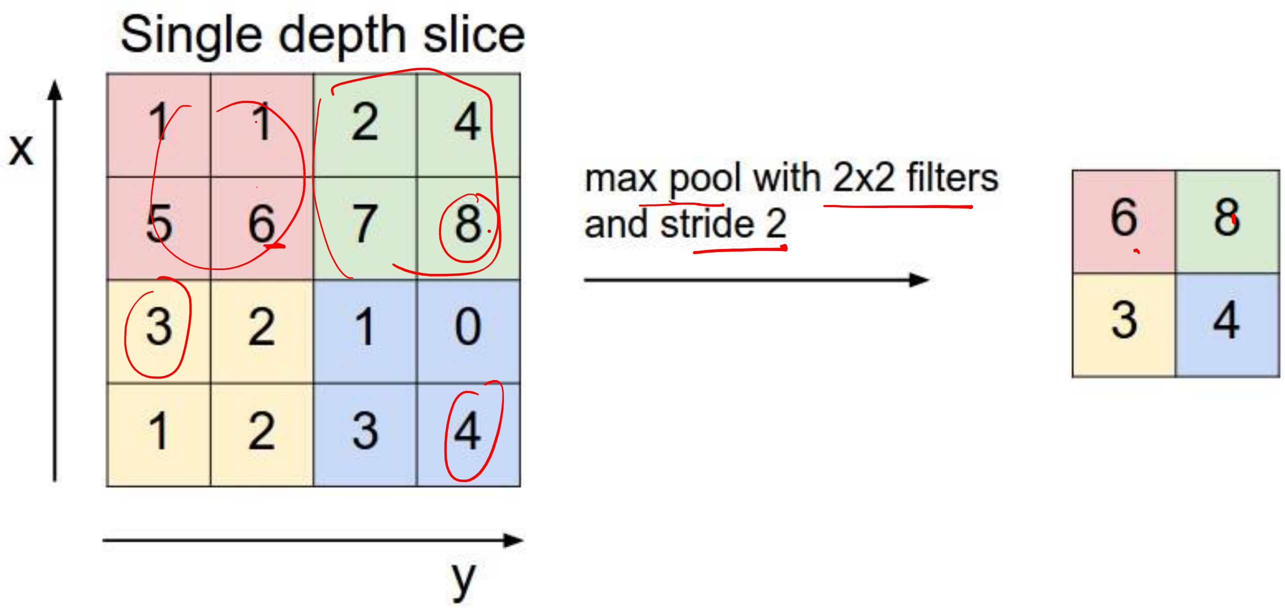
1. With parameter sharing, it introduces ***F·F·D1*** weights per filter, for a total of ***(F·F·D1)·K*** weights and ***K*** biases.
2. In the output volume, the ***d***-th depth slice (of size ***W2* X *H2***) is the result of performing a valid convolution of the ***d***-th filter over the input volume with a stride of ***S***, and then offset by ***d***-th bias.



Different layers of CNN architecture



CNN: Pooling layer



1. To reduce the spatial size of the representation to reduce the amount of parameters and computation in the network.
2. Average pooling or L2 pooling can also be used, but not popular like max pooling.

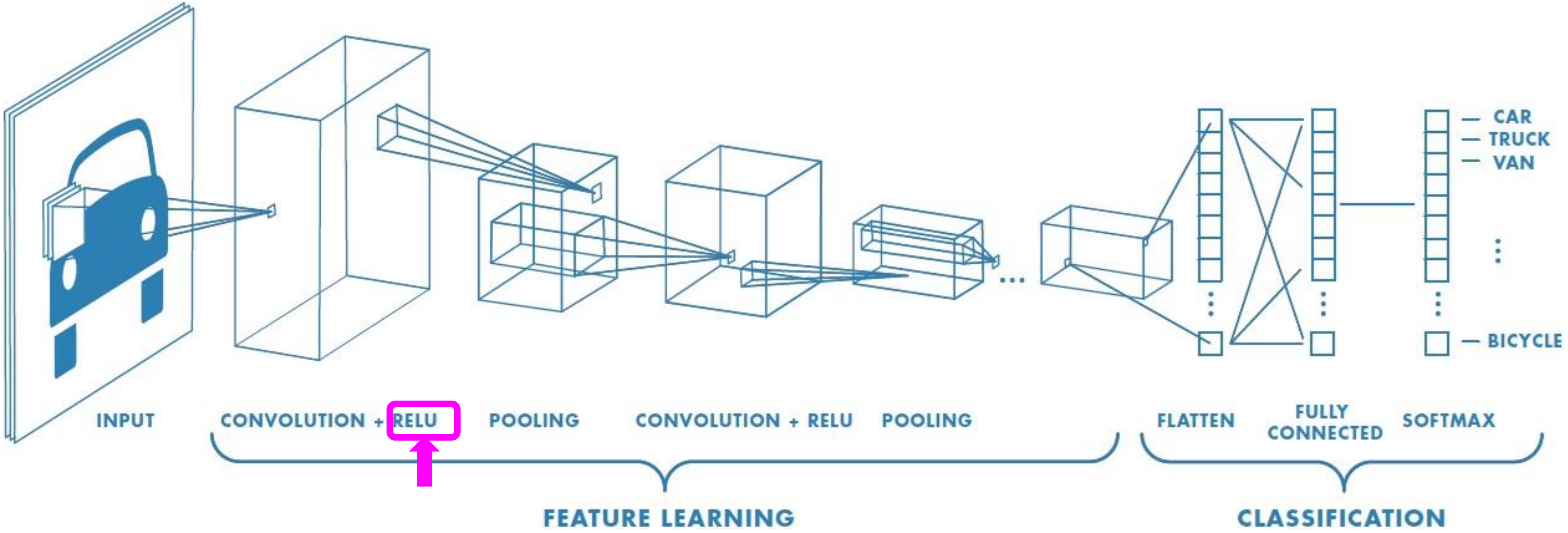
POOLING LAYER

1. Accepts a volume of size ***W1* X *H1* X *D1***.
2. Requires two hyperparameters:
    a. their spatial extent ***F***
    b. their stride ***S***
    c. the amount of zero padding ***P*** (commonly ***P*** = 0).
3. Produces an output volume of size ***W2* X *H2* X *D2*** where:

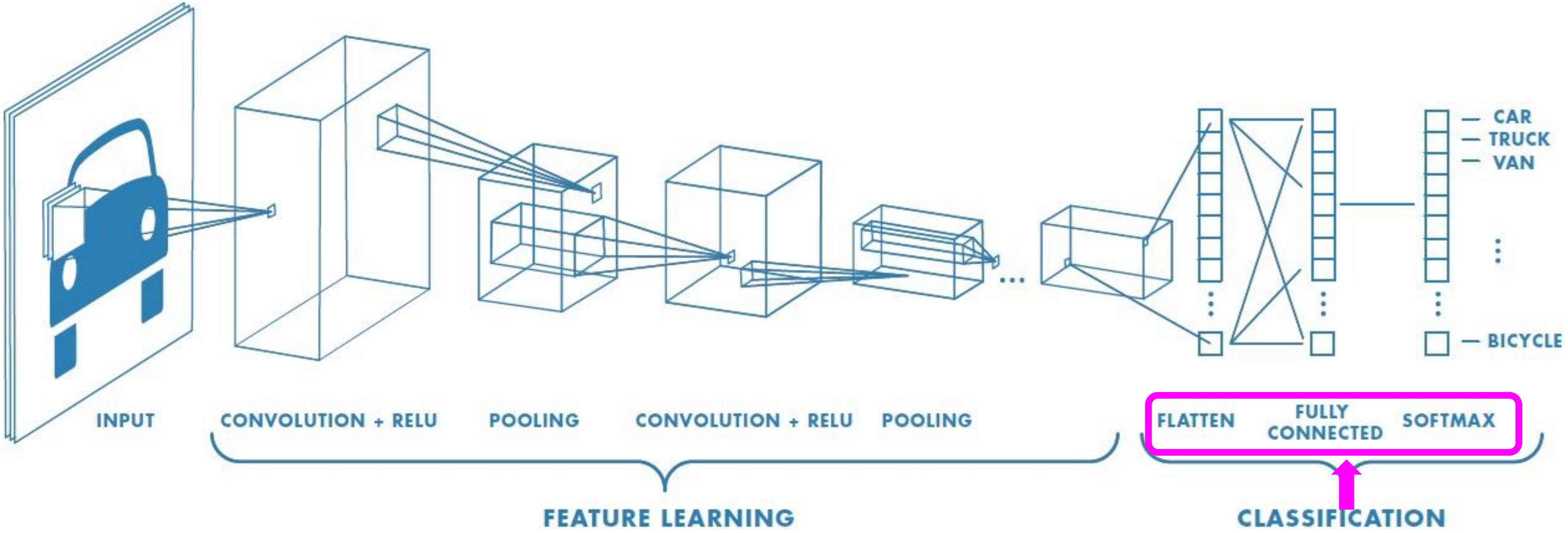
    W2=(W1−F+2P)/S+1, H2=(H1−F+2P)/S+1, D2=D1

1. Introduces **zero** parameters since it computes a fixed function of the input.

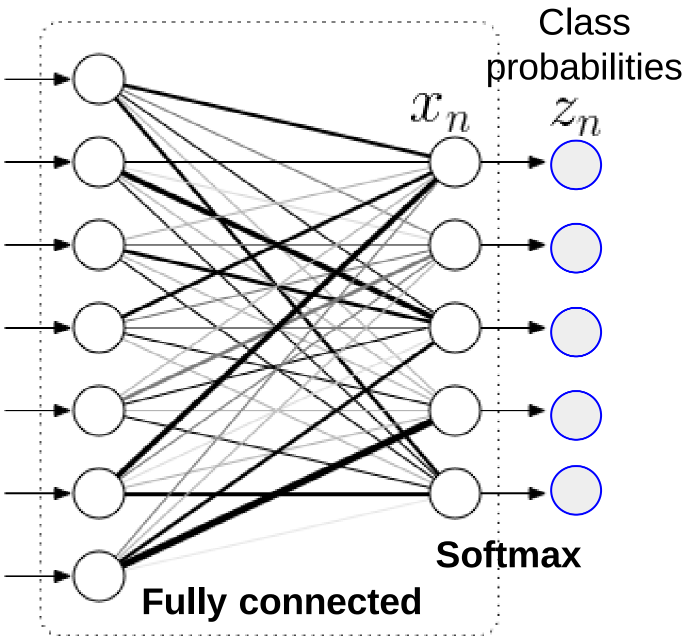
Different layers of CNN architecture



Different layers of CNN architecture



Flattening, fully connected (FC) layer and softmax



Flattening

1. Vectorization (converting ***M X N X D*** tensor to a ***MND X 1*** vector).

FC layer

1. Multilayer perceptron.
2. Generally used in final layers to classify the object.
3. Role of a classifier.

Softmax layer

1. Normalize output as discrete class probabilities.

$$z_n = \frac{e^{x_n}}{\sum_{i=1}^{K} e^{x_i}}$$