Neural Networks

Shikui Tu

Department of Computer Science and Engineering, Shanghai Jiao Tong University

2021-05-18

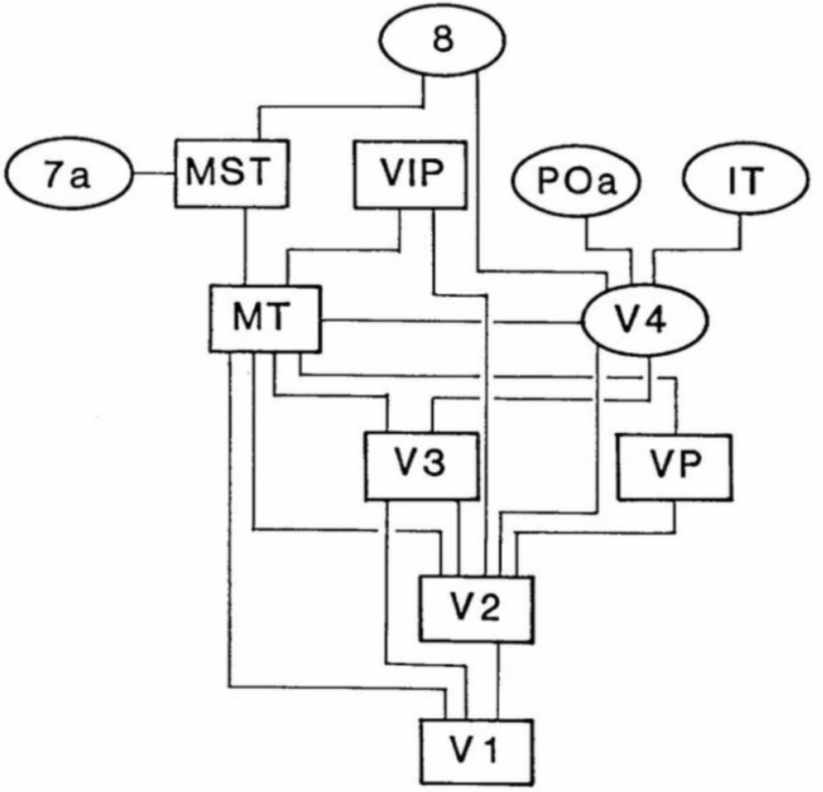
Outline

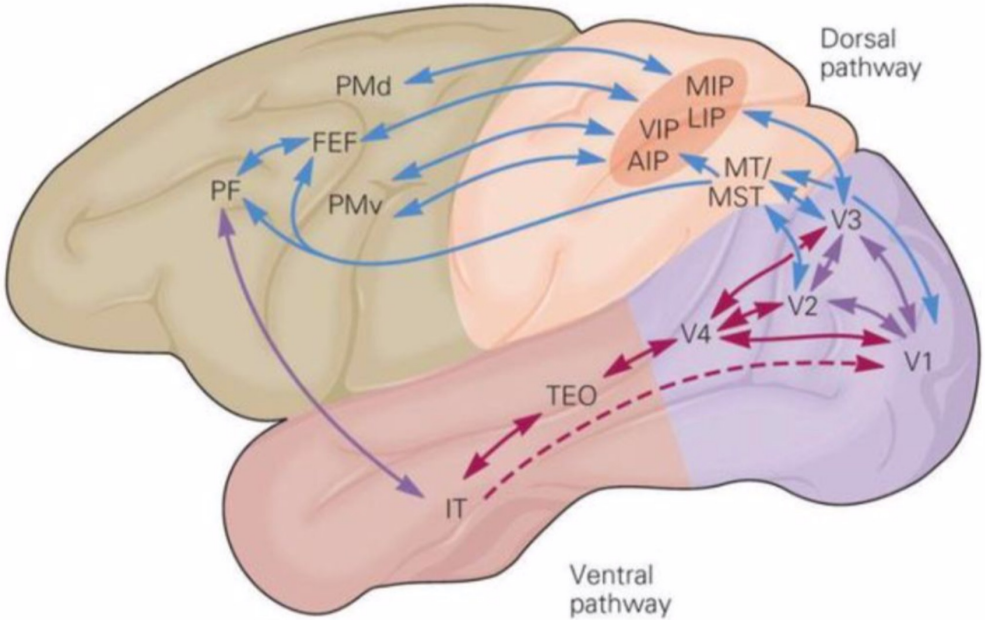
- Training the neural networks
  - Backpropagation (BP) algorithm

- Convolutional neural networks (CNN)
  - Overview
  - Network details: convolution, pooling, activation, loss functions

- ResNet
- DenseNet
- RNN

Visual cortex



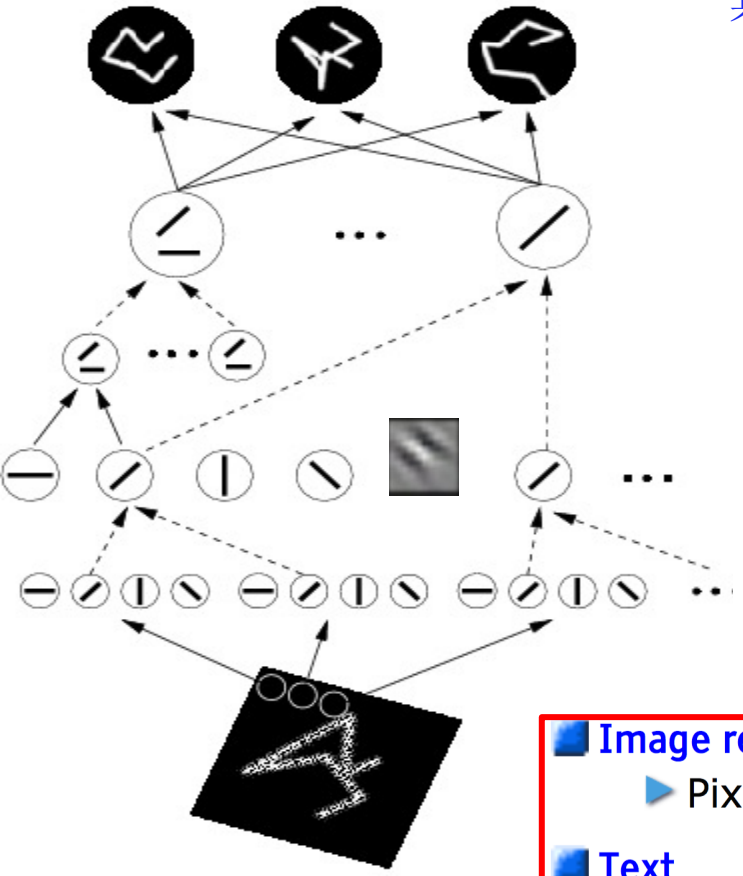


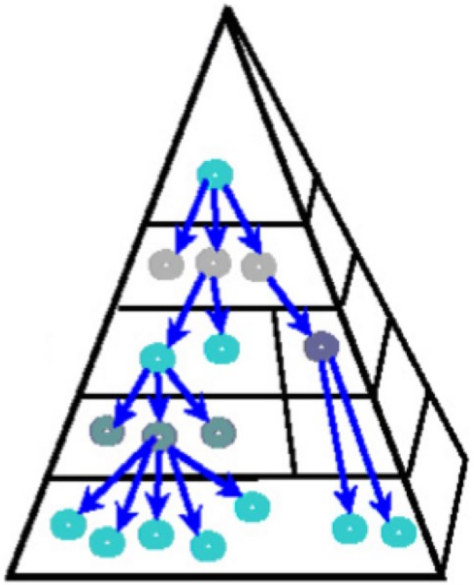
Starting from V1 primary visual cortex, visual signal is transmitted upwards, becoming more complicated and abstract.

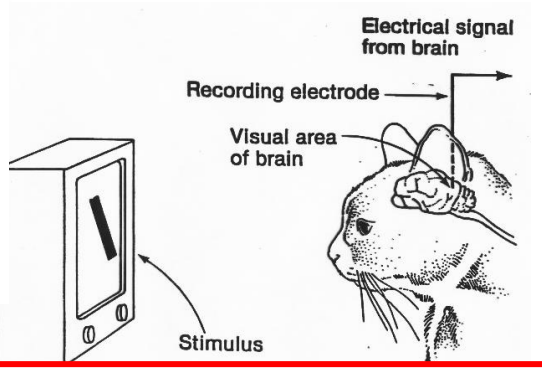
Feature Detection Theory 特征检测理论（1960）

Hubel Wiesel

具备广泛的普遍性！



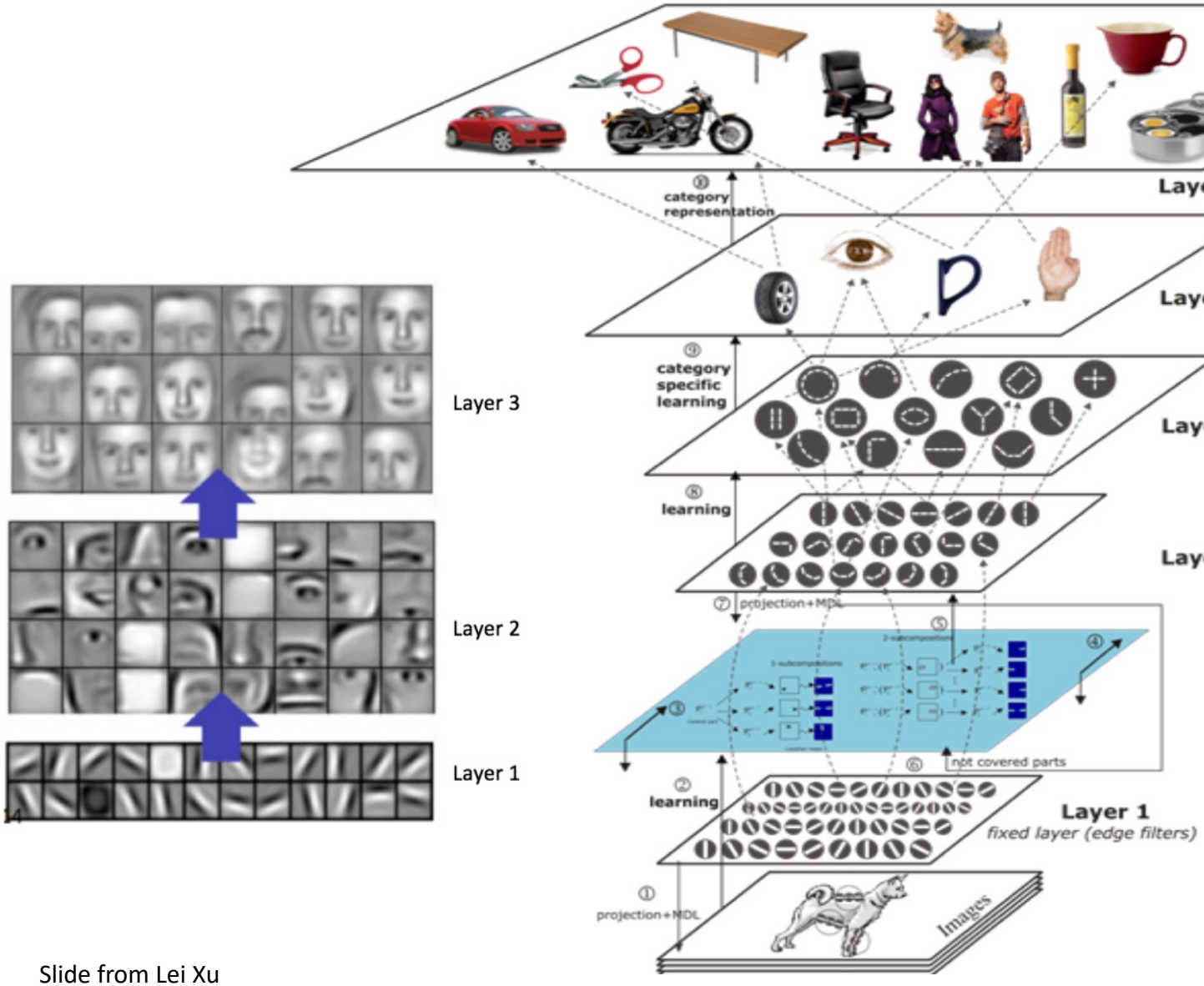




- **Image recognition**
  - Pixel → edge → texton → motif → part → object
- **Text**
  - Character → word → word group → clause → sentence → story
- **Speech**
  - Sample → spectral band → sound → … → phone → phoneme → word →

Slide from Lei Xu

Hierarchical representations



Slide from Lei Xu

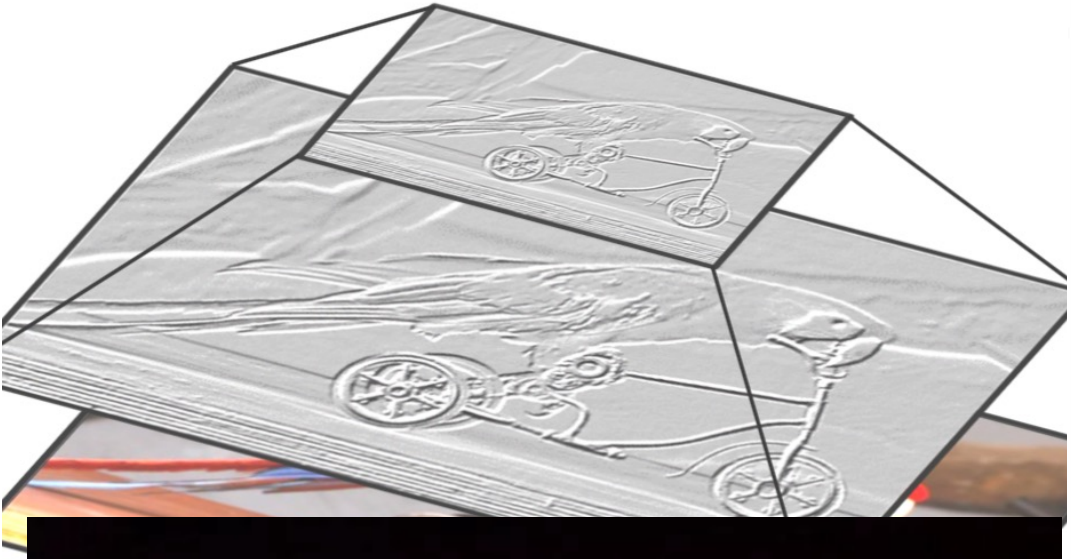
Outline

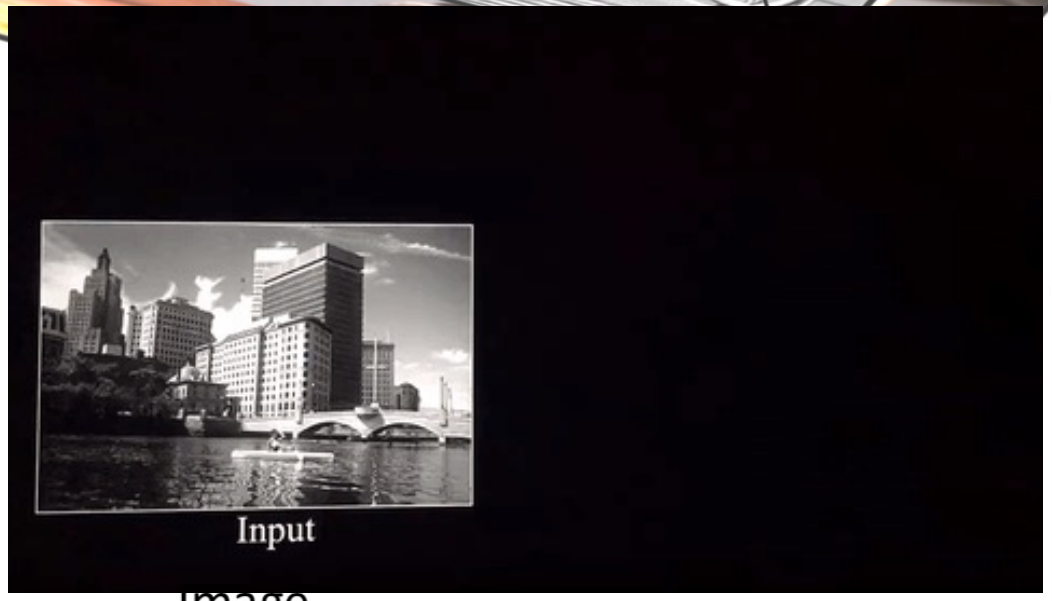
- Background introduction

- **Convolutional neural networks (CNN)**
  - Overview
  - Network details: convolution, pooling, activation, loss functions

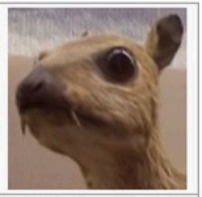
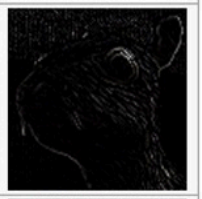
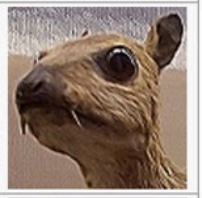
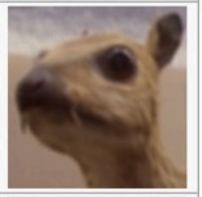
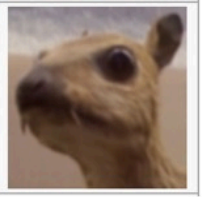
- ResNet
- DenseNet
- RNN

A Basic Module of the **Convolutional Neural Nets**





Slide from Lei Xu

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Some influential CNN architectures

- **LeNet (1990s):** LeNet-5: a pioneering 7-level convolutional network by LeCun et al. in 1998
- **1990s to 2012:** In the years from late 1990s to early 2010s convolutional neural network were in incubation. As more and more data and computing power became available, tasks that convolutional neural networks could tackle became more and more interesting.
- **AlexNet (2012) –** In 2012, Alex Krizhevsky (and others) released AlexNet which was a deeper and much wider version of the LeNet and won by a large margin the difficult ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. It was a significant breakthrough with respect to the previous approaches and the current widespread application of CNNs can be attributed to this work.
- **ZF Net (2013) –** The ILSVRC 2013 winner was a Convolutional Network from Matthew Zeiler and Rob Fergus. It became known as the ZFNet (short for Zeiler & Fergus Net). It was an improvement on AlexNet by tweaking the architecture hyperparameters.
- **GoogLeNet (2014) –** The ILSVRC 2014 winner was a Convolutional Network from Szegedy et al. from Google. Its main contribution was the development of an *Inception Module* that dramatically reduced the number of parameters in the network (4M, compared to AlexNet with 60M).
- **VGGNet (2014) –** The runner-up in ILSVRC 2014 was the network that became known as the VGGNet. Its main contribution was in showing that the depth of the network (number of layers) is a critical component for good performance.
- **ResNets (2015) –** Residual Network developed by Kaiming He (and others) was the winner of ILSVRC 2015. ResNets are currently by far state of the art Convolutional Neural Network models and are the default choice for using ConvNets in practice (as of May 2016).
- **DenseNet (August 2016) –** Recently published by Gao Huang (and others), the Densely Connected Convolutional Network has each layer directly connected to every other layer in a feed-forward fashion. The DenseNet has been shown to obtain significant improvements over previous state-of-the-art architectures on five highly competitive object recognition benchmark tasks.

Early CNN: LeNet-5

LeNet-5, a pioneering 7-level convolutional network by LeCun et al. in 1998, that classifies digits, was applied by several banks to recognize hand-written numbers on checks digitized in 32x32 pixel images.

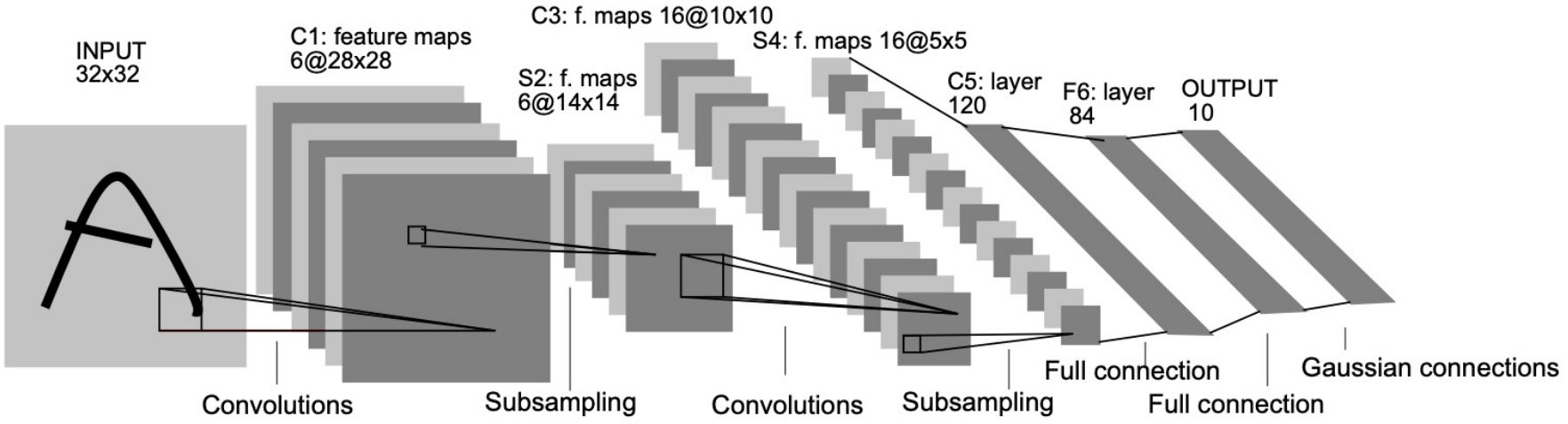


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann; Léon Bottou; Yoshua Bengio; Patrick Haffner (1998). "Gradient-based learning applied to document recognition" (PDF). Proceedings of the IEEE. 86 (11): 2278–2324.

Philip Marlowe
6381 Hollywood Blvd # 615
Los Angeles, CA 90028

PORTLAND OR 970
16 JAN 2014 PM 1 L

Dave Fennie
c/o Letter, inc
509 Cascade Ave, Suite H
Hood River, OR 97031

9703120608O

CARROLL O'CONNOR
BUSINESS ACCOUNT
% NANAS, STERN, BIERS AND CO.
9454 WILSHIRE BLVD., STE. 405 273-2501
BEVERLY HILLS, CALIF. 90212

11725

715

March 10 1980

16-24/63
1220

PAY TO THE ORDER OF Pollard-Wittman-Robb Chevrolet $5000.00

Five thousand xx/00 DOLLARS

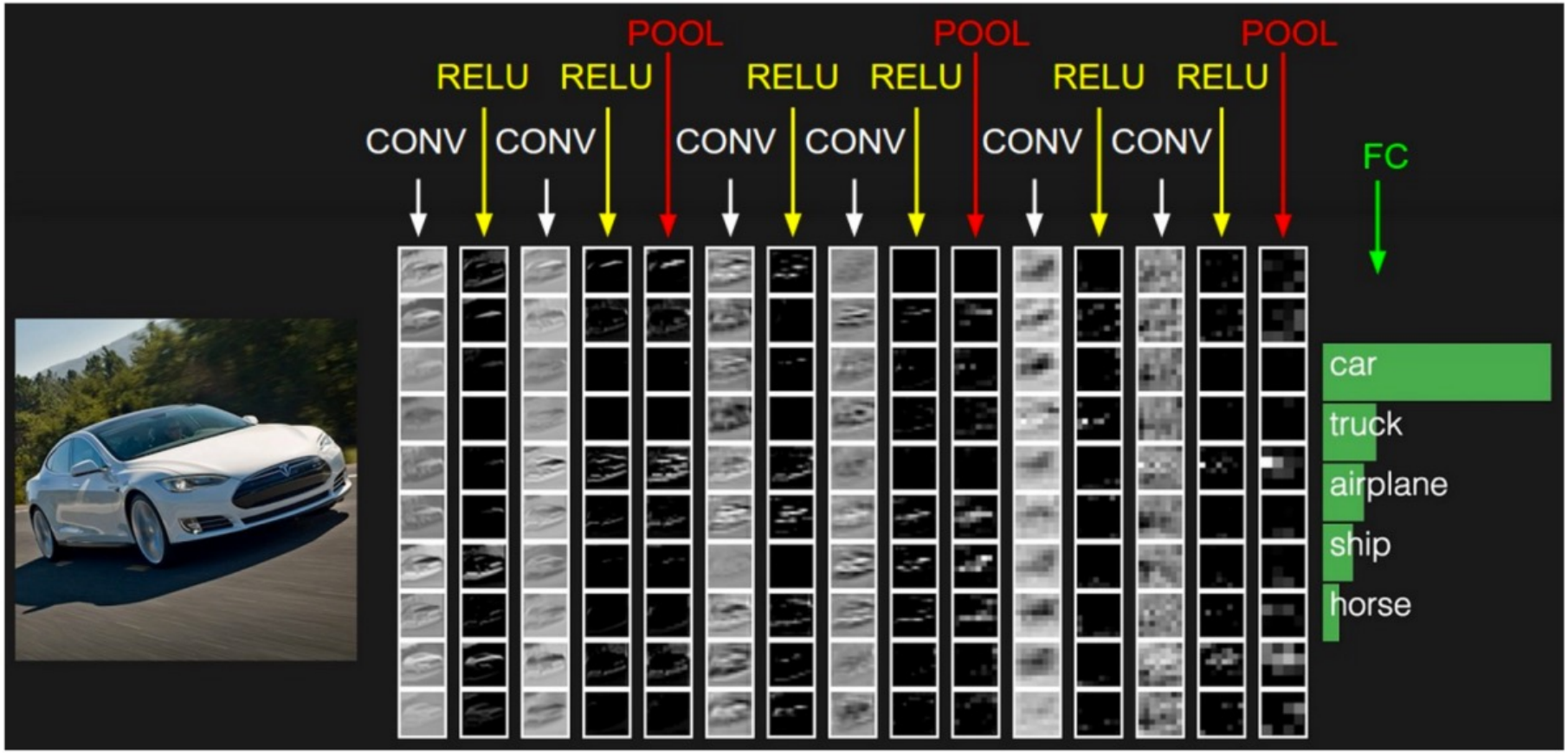
WILSHIRE-DOHENY OFFICE
WELLS FARGO BANK
NATIONAL ASSOCIATION
9101 WILSHIRE BOULEVARD
BEVERLY HILLS, CALIFORNIA 90211
SINCE 1852

MEMO deposit 1980 Chev. pickup for HCO.

Carroll O'Connor

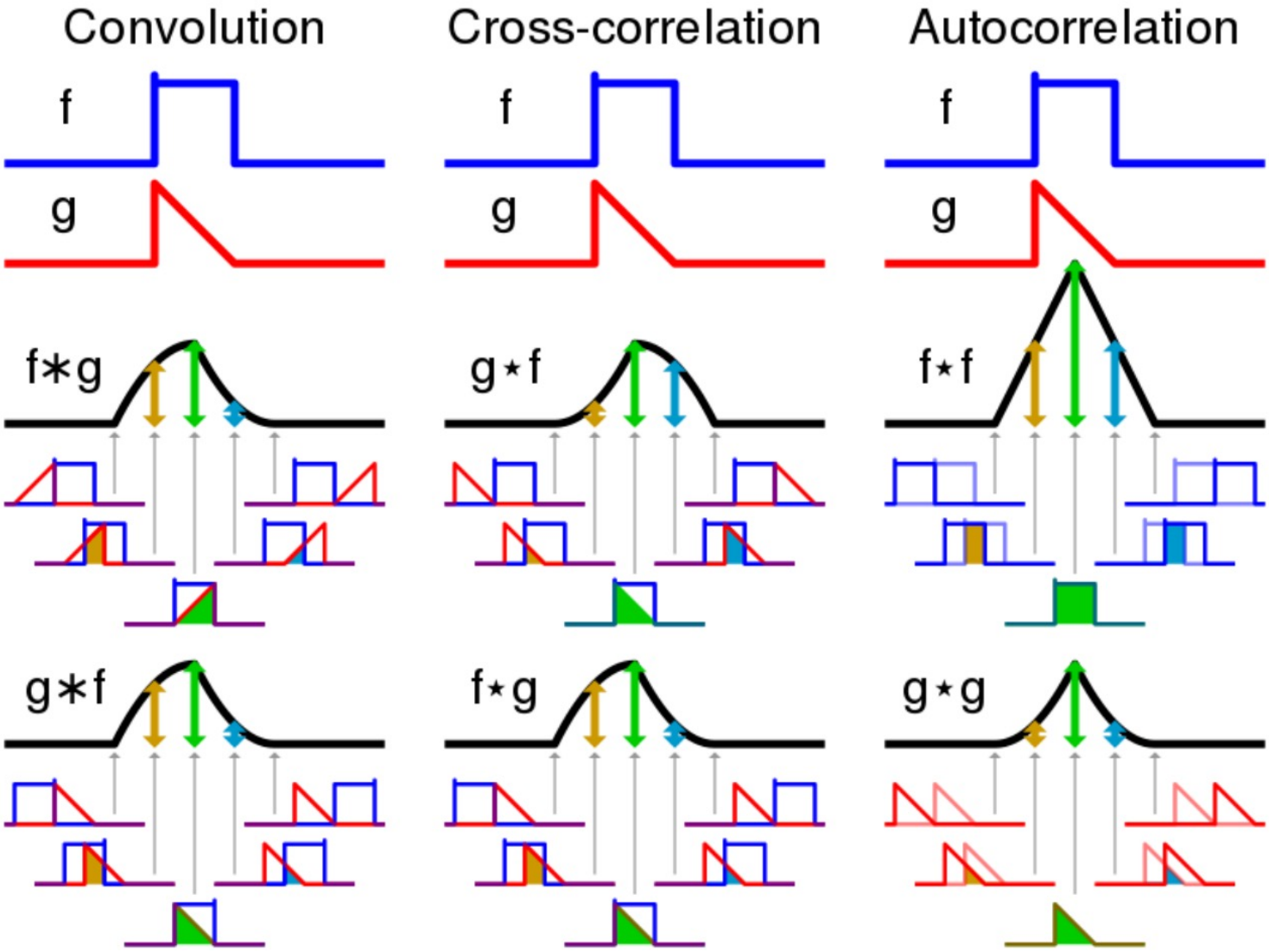
⑆122000024⑆715 0635 11875⑈ ⑈0000500000⑈

A typical modern CNN



In fact, some of the best performing ConvNets today have tens of Convolution and Pooling layers! Also, it is not necessary to have a Pooling layer after every Convolutional Layer.

Convolution

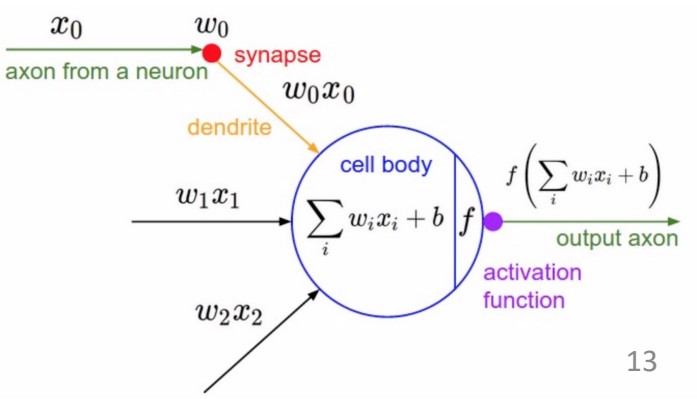


$$(f * g)(t) \triangleq \int_{-\infty}^{\infty} f(\tau) g(t - \tau)\, d\tau.$$

Affine transformations

- A vector is received as input and is multiplied with a matrix to produce an output (to which a bias vector is usually added before passing the result through a non-linearity).

- Applicable to any type of input
  - flattened into a vector
  - Image, sound clip, etc.



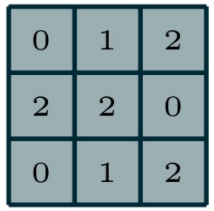
V. Dumoulin and F. Visin, arXiv:1603.07285, 2016

Affine transformations are not good for image-like structured data

- Images, sound clips and many other similar kinds of data have an intrinsic structure.
- They are stored as multi-dimensional arrays.
- They feature one or more axes for which ordering matters (e.g., width and height axes for an image, time axis for a sound clip).
- One axis, called the channel axis, is used to access different views of the data (e.g., the red, green and blue channels of a color image, or the left and right channels of a stereo audio track).

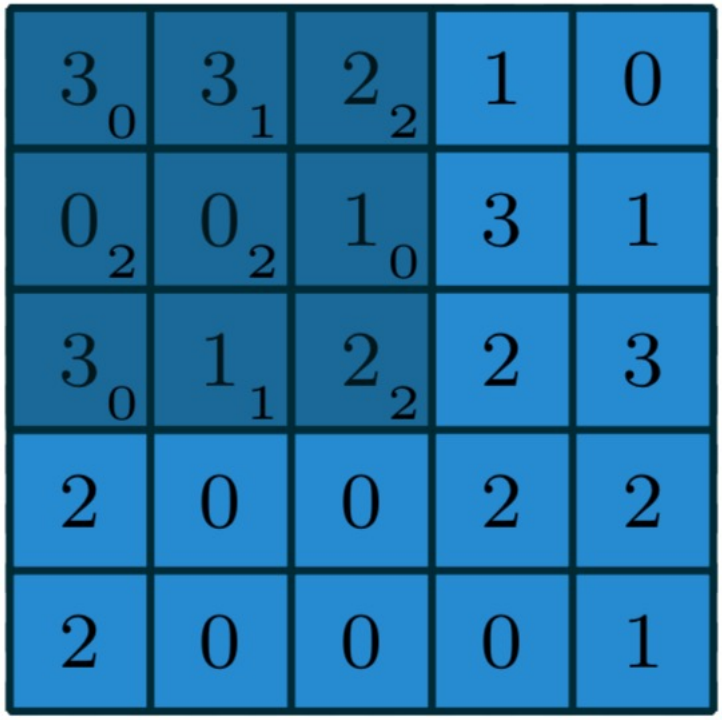
V. Dumoulin and F. Visin, arXiv:1603.07285, 2016

Discrete convolution

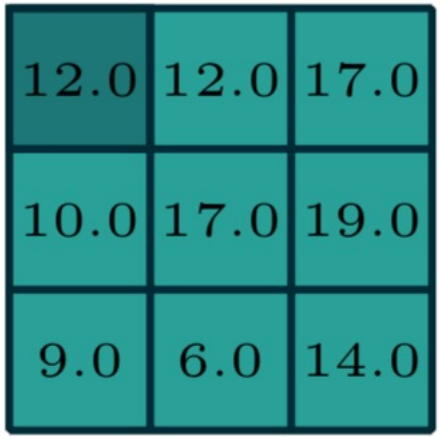


Kernel (3x3)

Input feature map



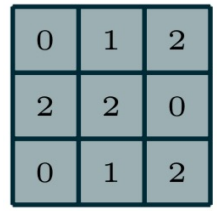
Output feature map



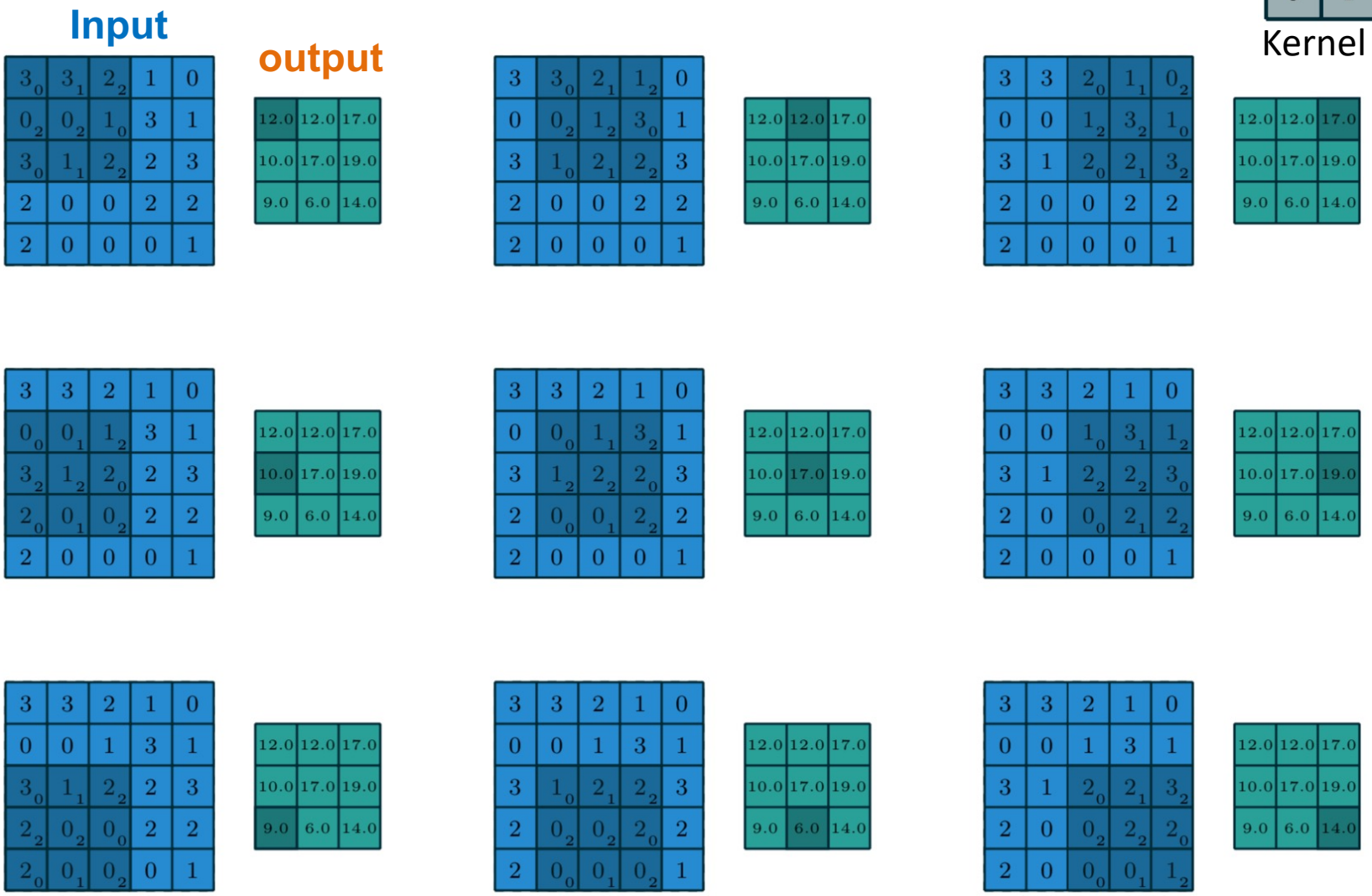
0 x 3 + 1 x 3 + 2 x 2
+ 2 x 0 + 2 x 0 + 0 x 1
+ 0 x 3 + 1 x 1 + 2 x 2
= 12.0

V. Dumoulin and F. Visin, arXiv:1603.07285, 2016

Discrete convolution



Kernel (3x3)



V. Dumoulin and F. Visin, arXiv:1603.07285, 2016

N-D convolution (N=2,3,…)

axis $j = 1, 2, \ldots, N$.

$n \equiv$ number of output feature maps,

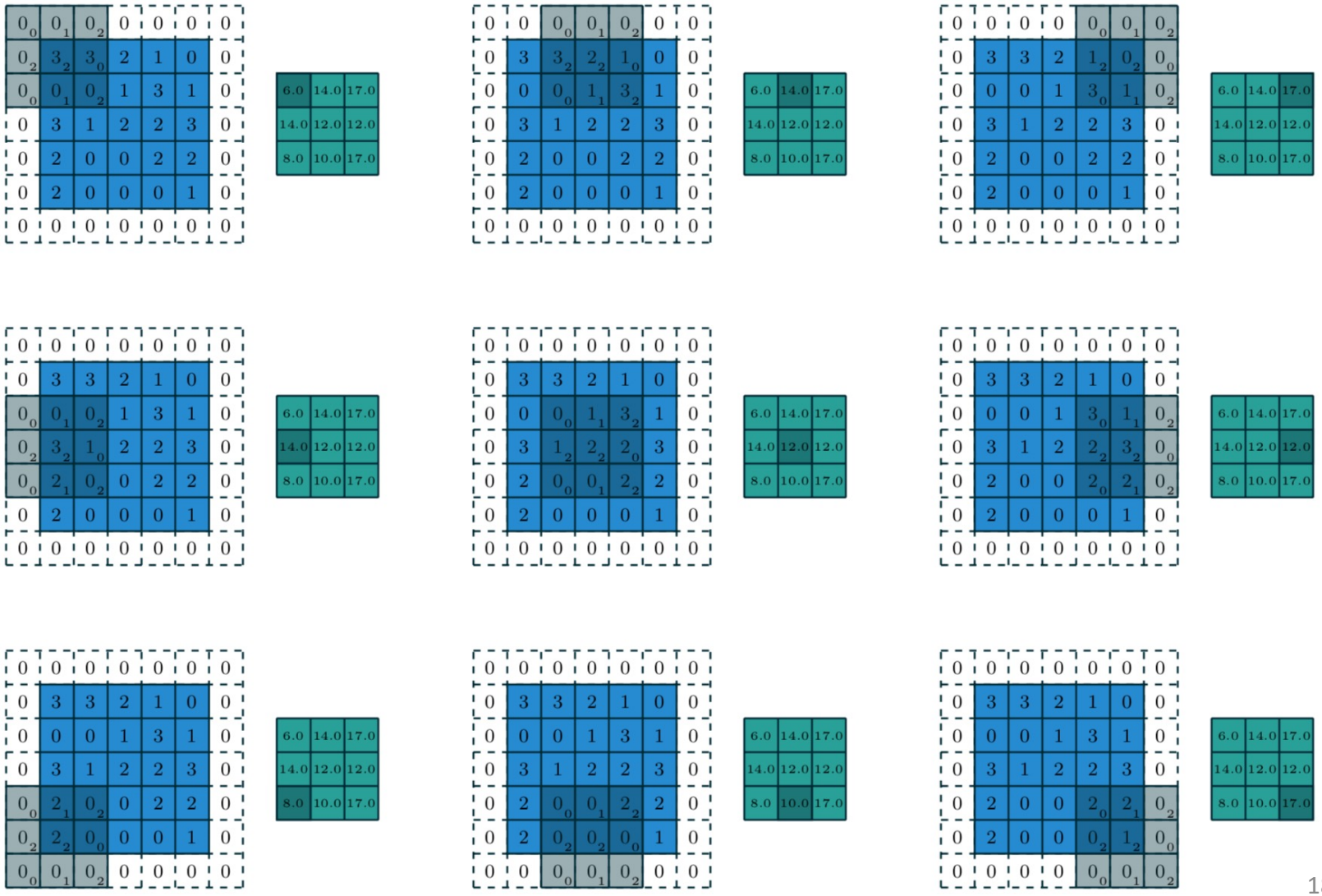
$m \equiv$ number of input feature maps,

$k_j \equiv$ kernel size along axis j.

- i_j: input size along axis j,
- k_j: kernel size along axis j,
- s_j: stride (distance between two consecutive positions of the kernel) along axis j,
- p_j: zero padding (number of zeros concatenated at the beginning and at the end of an axis) along axis j.

Example with zero paddings

for $N = 2$, $i_1 = i_2 = 5$, $k_1 = k_2 = 3$, $s_1 = s_2 = 2$, and $p_1 = p_2 = 1$.

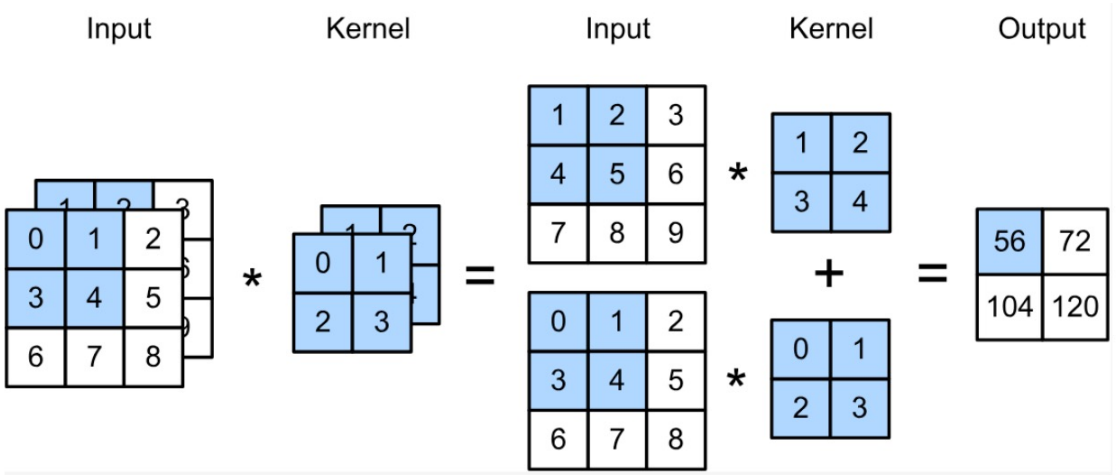


Multiple Input Channels

Red Green Blue

Multiple Input Channels

- One kernel for each channel, and then sum results over channels

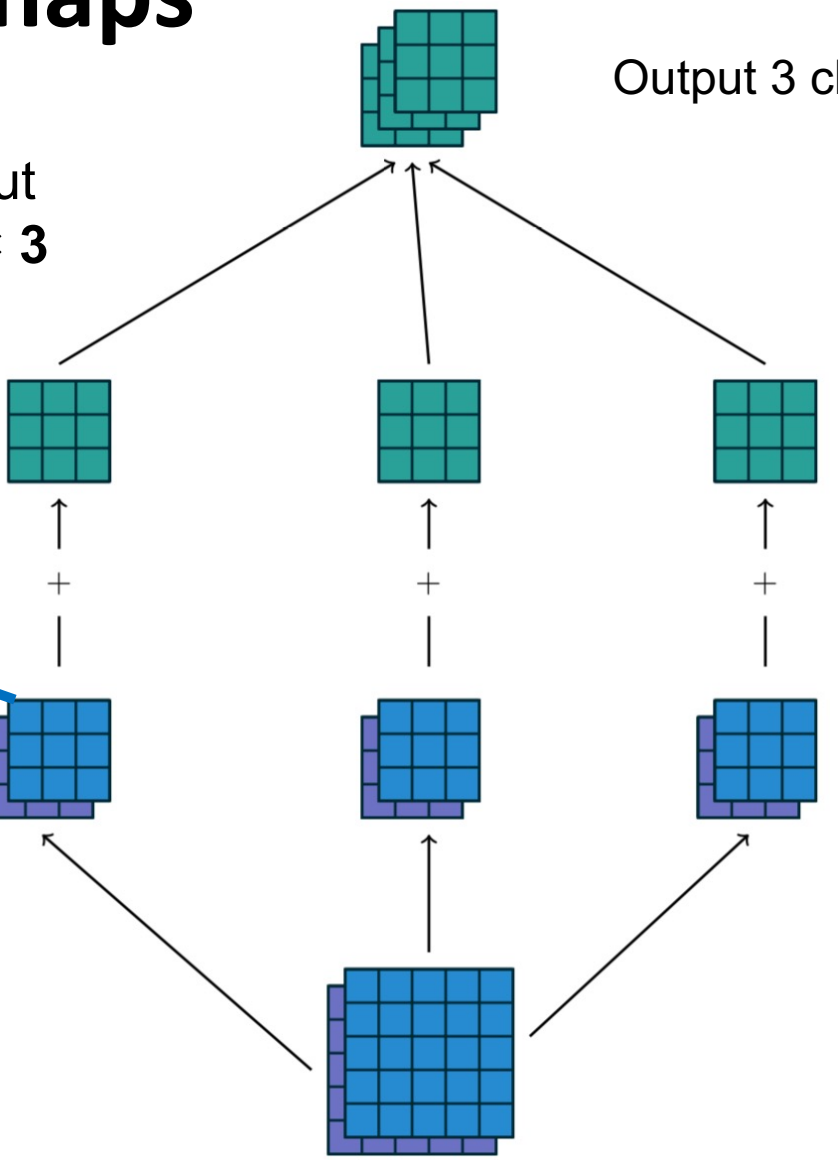


$$(1\times 1+2\times 2+4\times 3+5\times 4)$$
$$+(0\times 0+1\times 1+3\times 2+4\times 3)$$
$$=56$$

Example with three output feature maps

A convolution mapping from two input feature maps to three output feature maps using a 3 × 2 × **3 × 3** collection of kernels **w**.

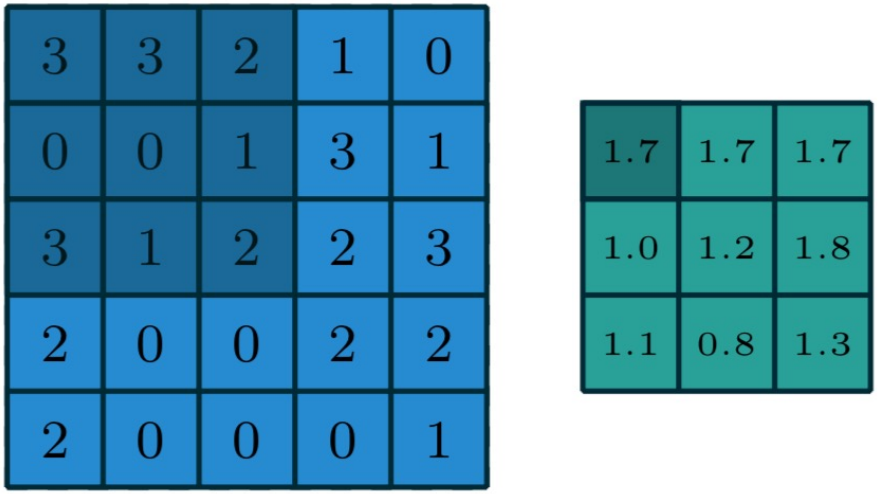
- input feature map 1 is convolved with kernel $\mathbf{w}_{1,1}$.
- input feature map 2 is convolved with kernel $\mathbf{w}_{1,2}$.



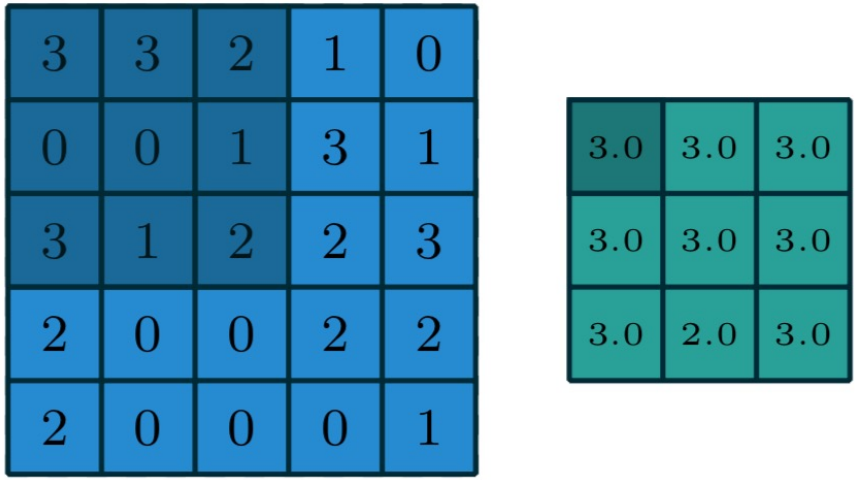
Pooling

- Pooling operations reduce the size of feature maps by using some function to summarize subregions, such as taking the **average** or the **maximum** value.

Average pooling

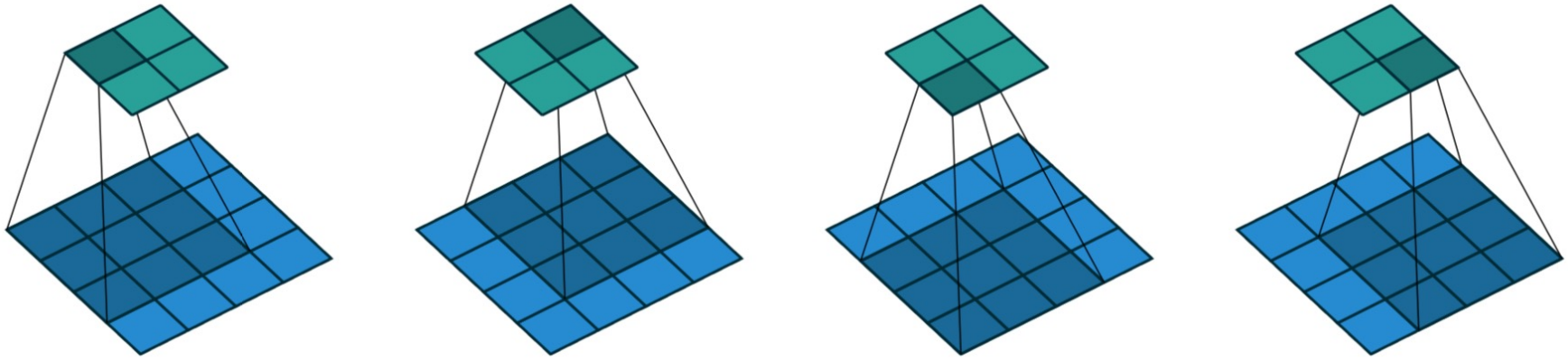


Maximum pooling



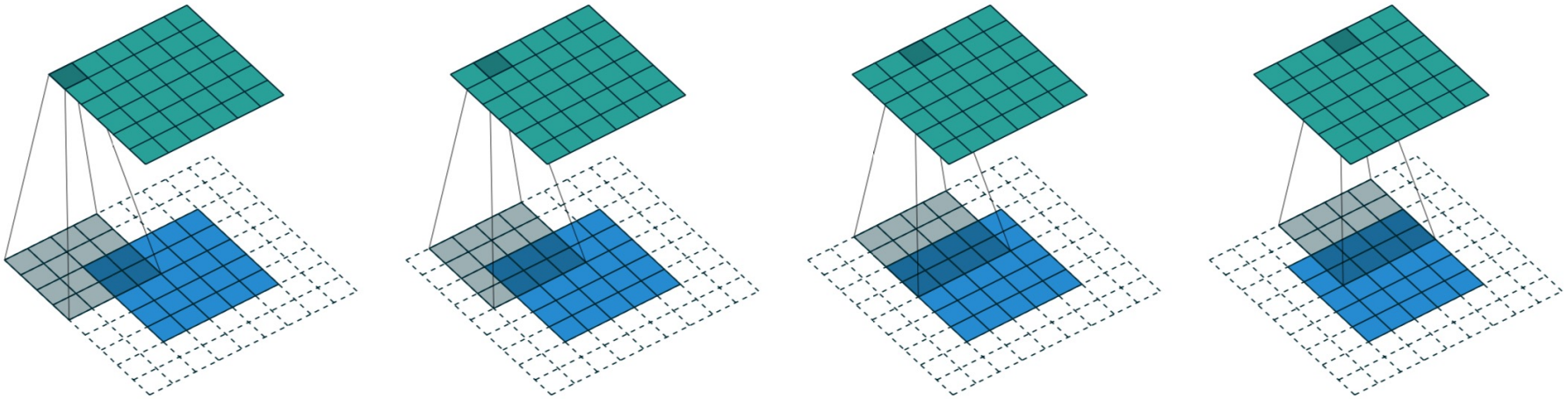
Example -- No padding, no strides

Convolving a 3 × 3 kernel over a 4 × 4 input using unit strides (i.e., i = 4, k = 3, s = 1 and p = 0).

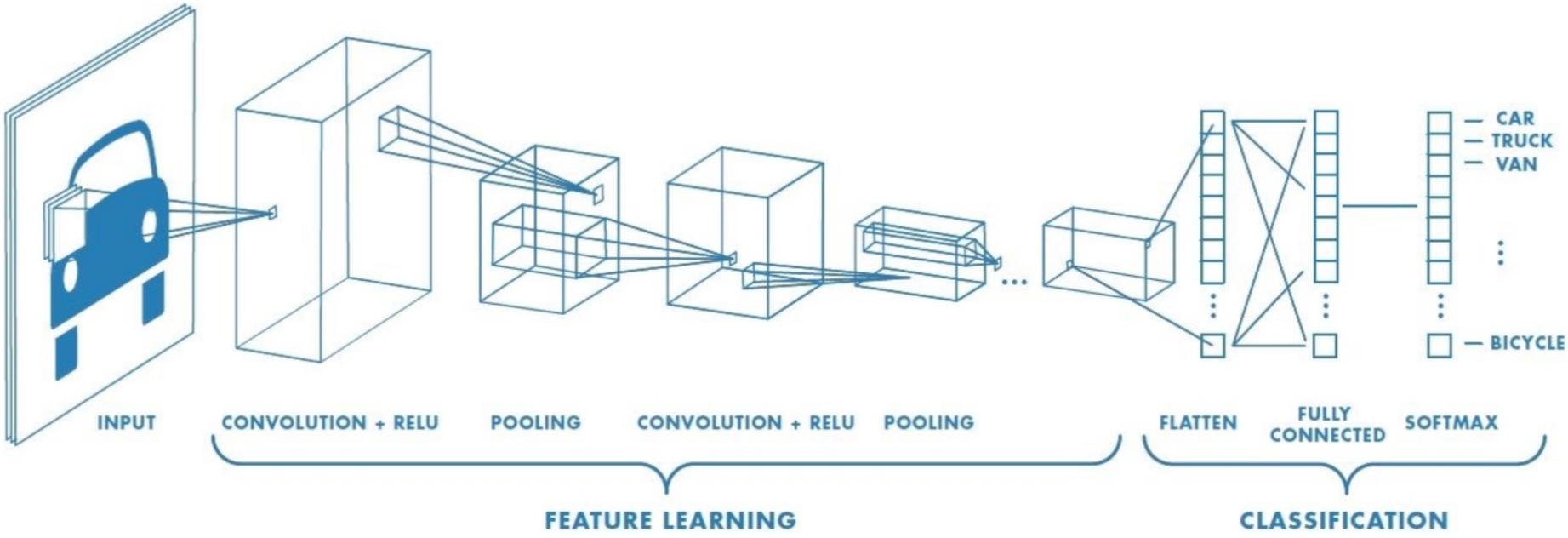


Example -- Arbitrary padding, no strides

Convolving a 4 × 4 kernel over a 5 × 5 input padded with a 2 × 2 border of zeros using unit strides (i.e., i = 5, k = 4, s = 1 and p = 2).

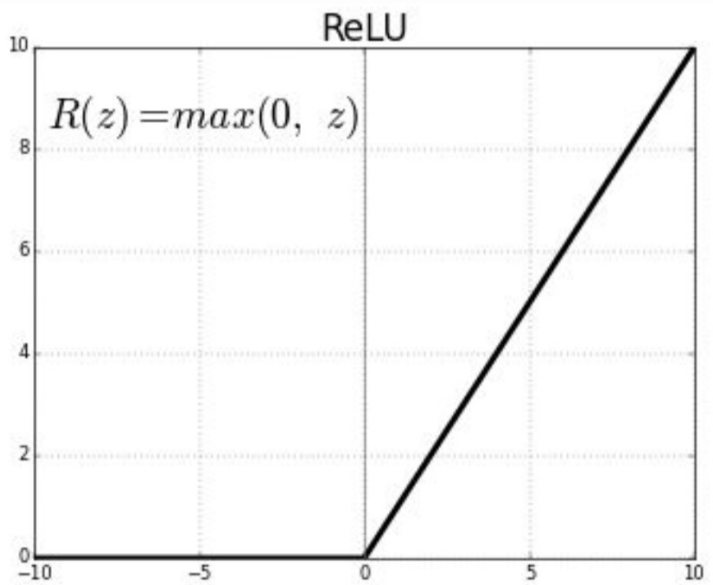


Sample architecture of CNN



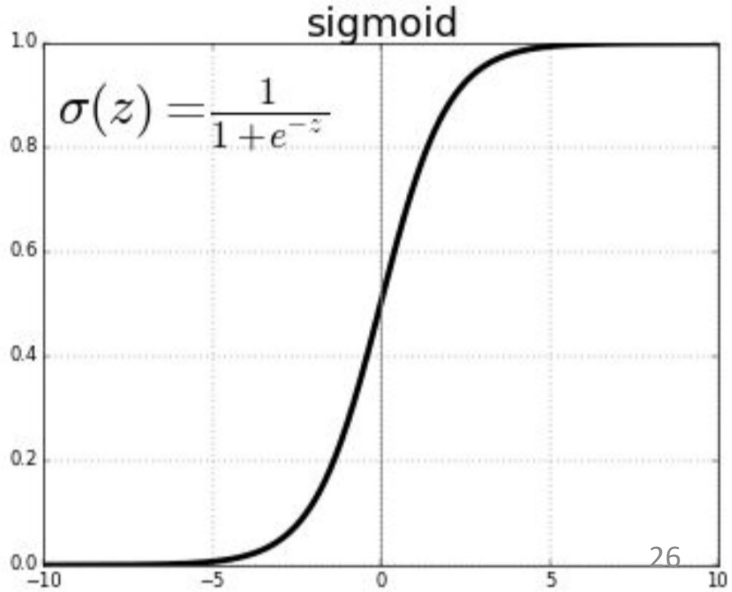
Activation layer

- Used to increase non-linearity of the network without affecting receptive fields of conv layers
- Prefer **ReLU**, results in faster training
- **LeakyReLU** addresses the vanishing gradient problem



Other types:

- Leaky ReLU,
- Randomized Leaky ReLU,
- Parameterized ReLU
- Exponential Linear Units (ELU),
- Scaled Exponential Linear Units Tanh,
- hardtanh, softtanh, softsign, softmax, softplus...



Softmax

- A special kind of activation layer, usually at the end of FC layer outputs
- Can be viewed as a fancy **normalizer** (a.k.a. Normalized exponential function)
- Produce a **discrete probability distribution** vector
- Very convenient when combined with cross-entropy loss

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$$

- Sample vector **x**
- Weight vector **w**

Loss layer

- **L1, L2** loss
- **Cross-Entropy** loss (works well for classification, e.g., image classification)
- **Hinge** Loss
- **Huber** Loss, more resilient to outliers with smooth gradient
- **Minimum Squared Error** (works well for regression task, e.g., Behavioral Cloning)

$$J = -\sum_{i=1}^{N} y_i \log(h_\theta(x_i)) + (1 - y_i) \log(1 - h_\theta(x_i))$$

$$p(y_i|x_i) = [h_\theta(x_i)]^{(y_i)}[1 - h_\theta(x_i)]^{(1-y_i)}$$

$$p(y_i = 1|x_i) = h_\theta(x_i) \qquad p(y_i = 0|x_i) = 1 - h_\theta(x_i)$$

$$\sum_i max(0, 1 - y_i * h_\theta(x_i))$$

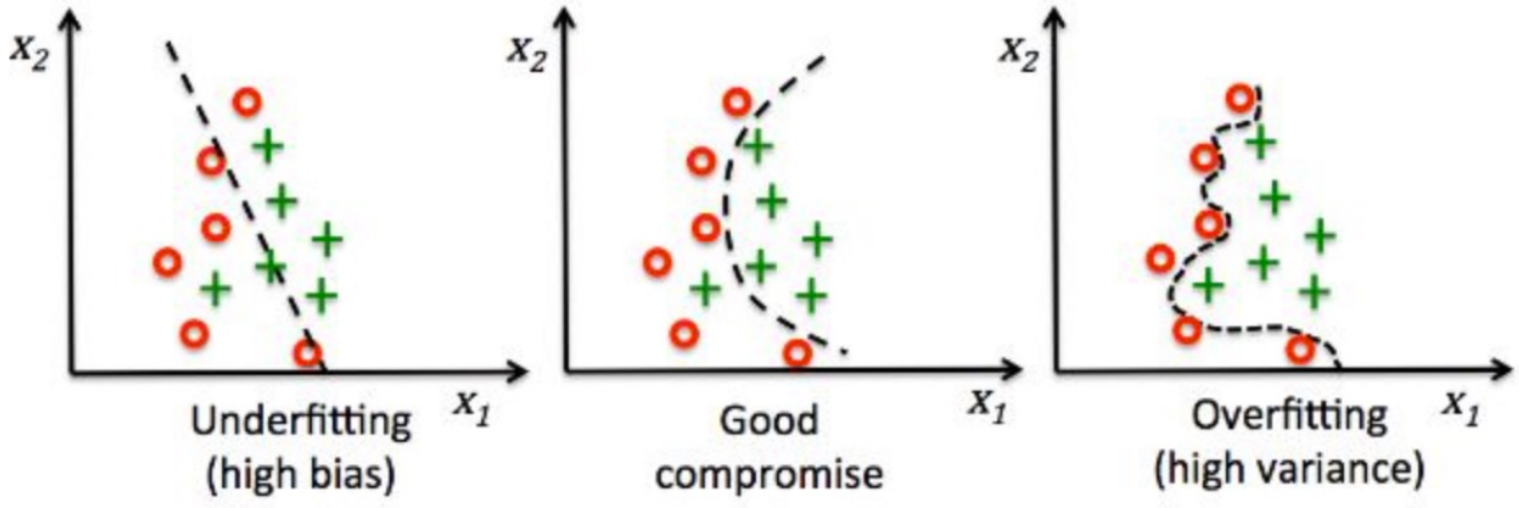
$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

$$J = \frac{1}{N} \sum_{i=1}^{N} (y_i - h_\theta(x_i))^2$$

https://rohanvarma.me/Loss-Functions/

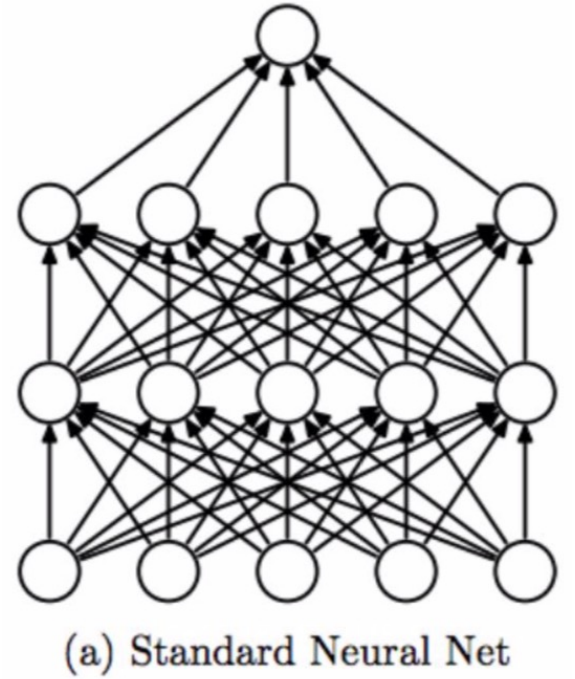
Regularization

- L1 / L2
- Dropout
- Batch norm
- Gradient clipping
- Max norm constraint

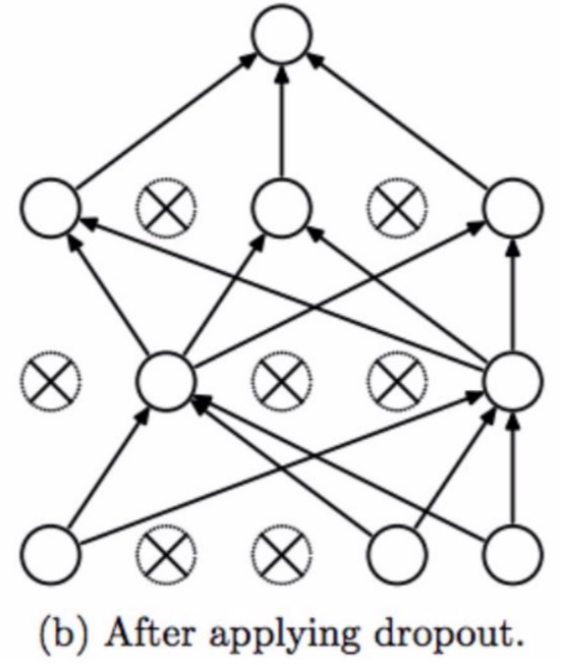


Dropout

- During training, randomly ignore activations by probability ***p***
- During testing, use all activations but scale them by ***p***
- Effectively prevent overfitting by reducing correlation between neurons



(a) Standard Neural Net



(b) After applying dropout.

Batch Normalization

- Makes networks robust to bad initialization of weights
- Usually inserted right before activation layers
- Reduce covariance shift by normalizing and scaling inputs
- The scale and shift parameters are trainable to avoid losing stability of the network

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

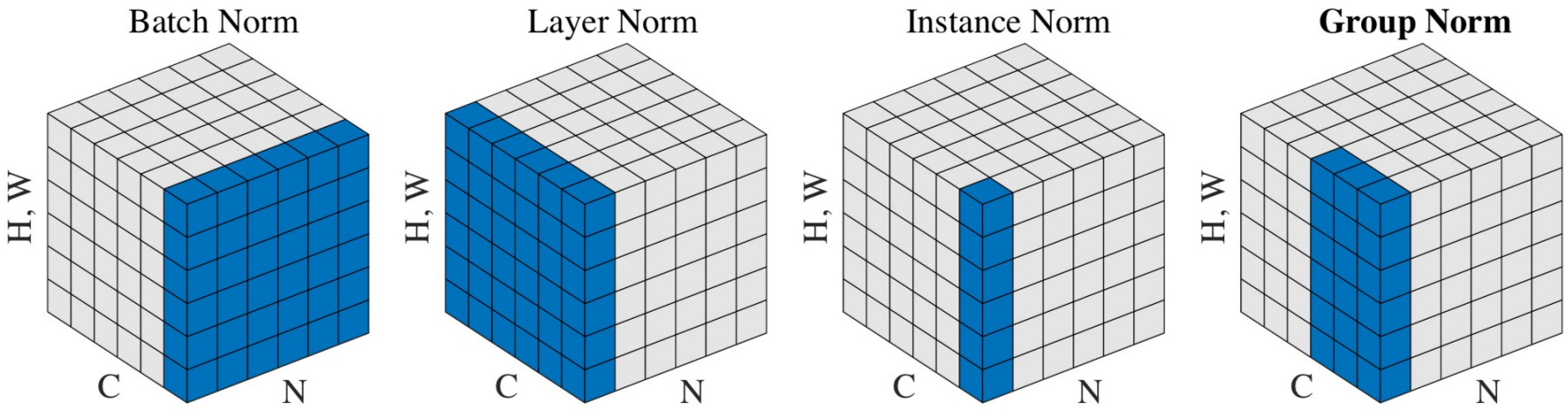
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.



Group Normalization

Yuxin Wu Kaiming He

Facebook AI Research (FAIR)

{yuxinwu,kaiminghe}@fb.com

arXiv:1803.08494v3

https://mlexplained.com/2018/11/30/an-overview-of-normalization-methods-in-deep-learning/

Summary

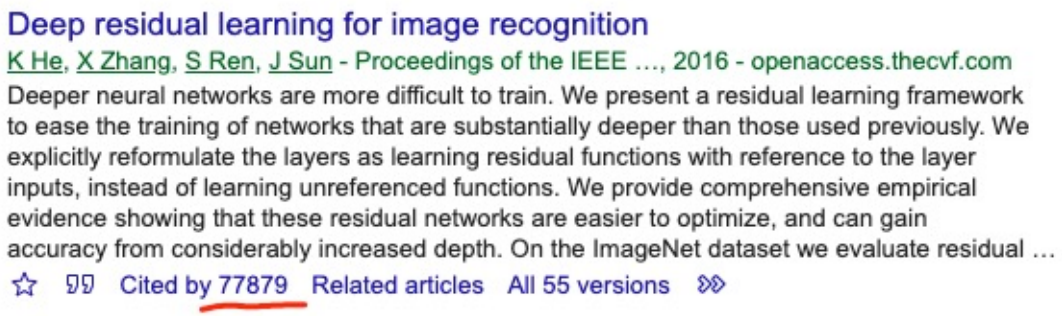
- **Convolutional layer**
  - Reduced model capacity compared to dense layer
  - Efficient at detecting spatial patterns
  - High computation complexity
  - Control output shape via padding, strides and channels
- **Max/Average Pooling layer**
  - Provides some degree of invariance to translation

Outline

- Background introduction
- Convolutional neural networks (CNN)
  - Overview
  - Network details: convolution, pooling, activation, loss functions
- ResNet
- DenseNet
- RNN

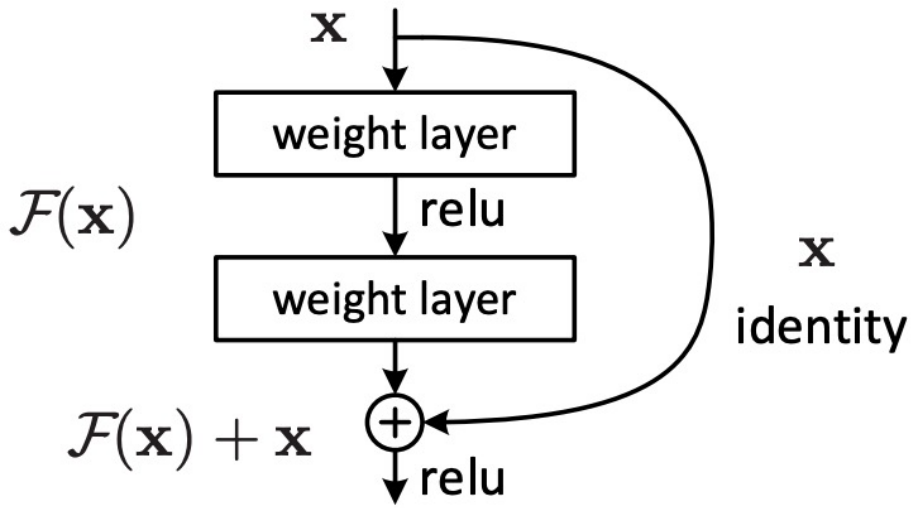
ResNet

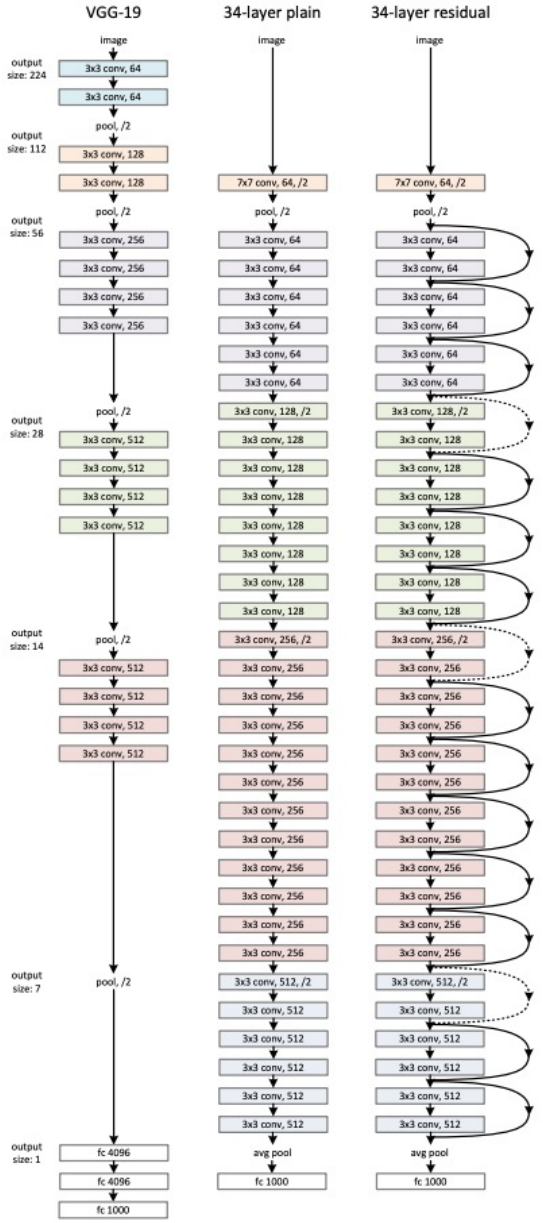
- Residual Network, by Kaiming He (2015)
- Heavy usage of "**skip connections**" which are similar to RNN Gated Recurrent Units (GRU)
- Commonly used as visual feature extractor in all kinds of learning tasks, ResNet50, ResNet101, ResNet152
- 3.57% Top-5 accuracy, **beats human**



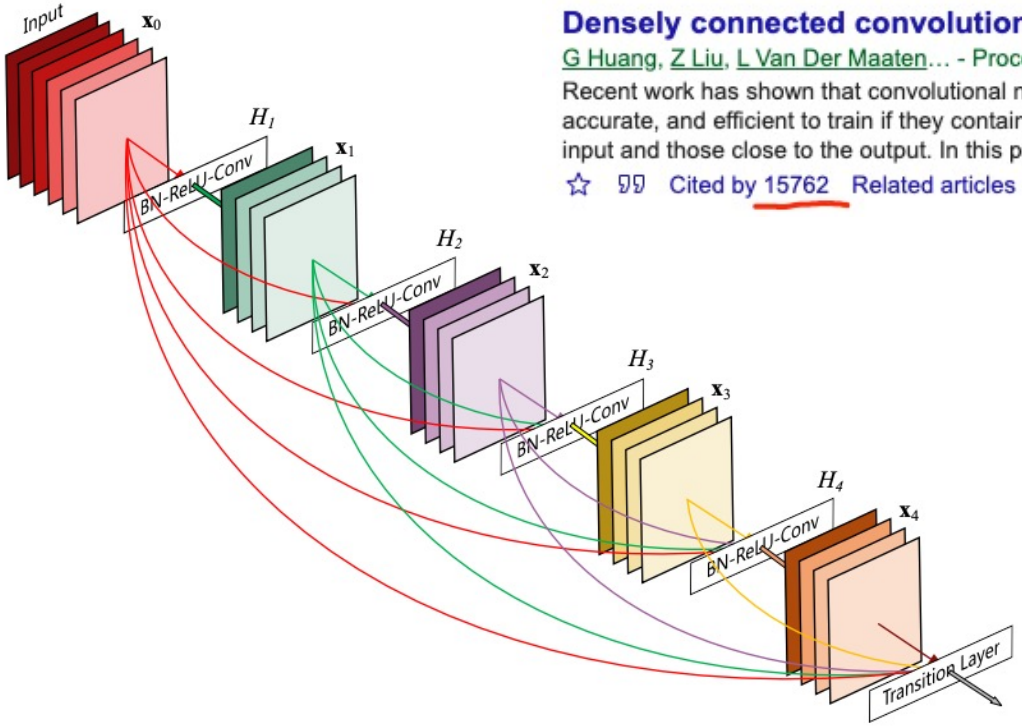
ResNet architecture

Residual learning: a building block





DenseNet



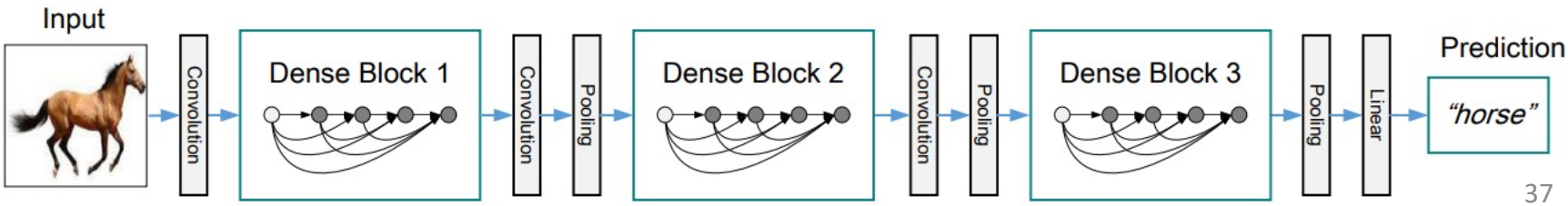
Densely connected convolutional networks

G Huang, Z Liu, L Van Der Maaten... - Proceedings of the ..., 2017 - openaccess.thecvf.com

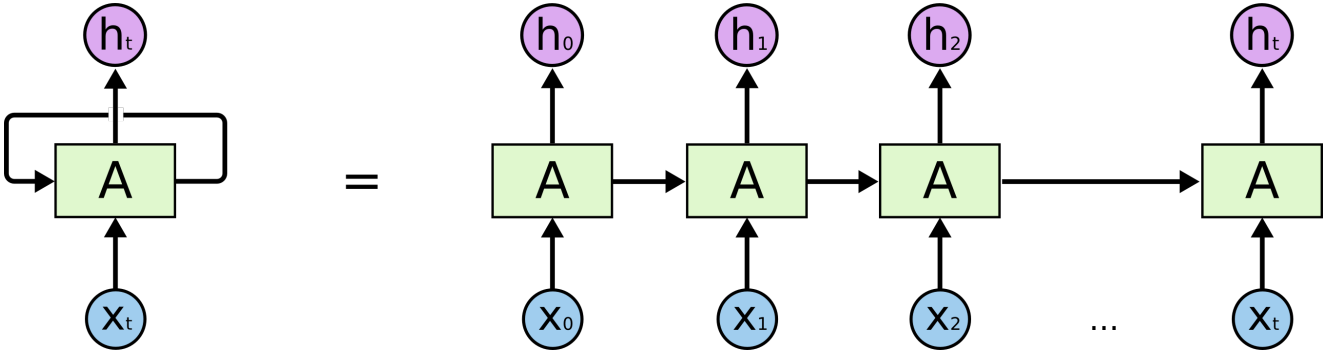
Recent work has shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. In this paper, we embrace this observation and introduce ...

☆ 99 Cited by 15762 Related articles All 30 versions »

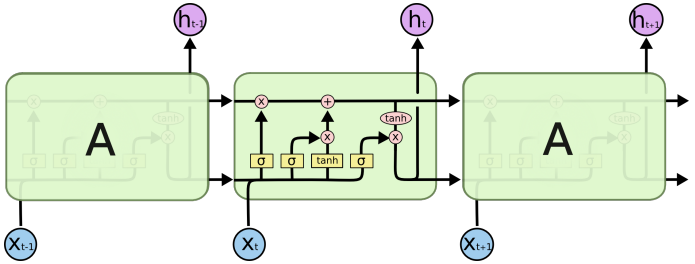
Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

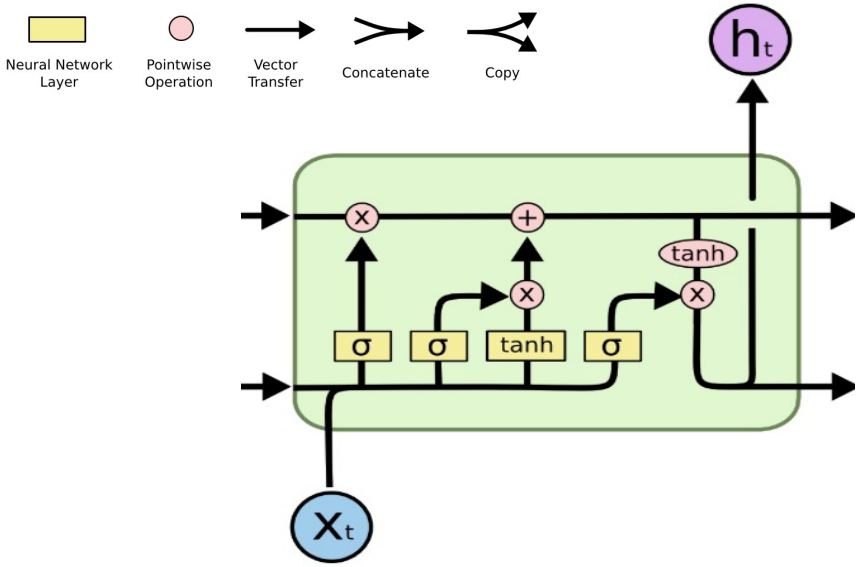


Recurrent neural networks



Long Short Term Memory networks (LSTM)





$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

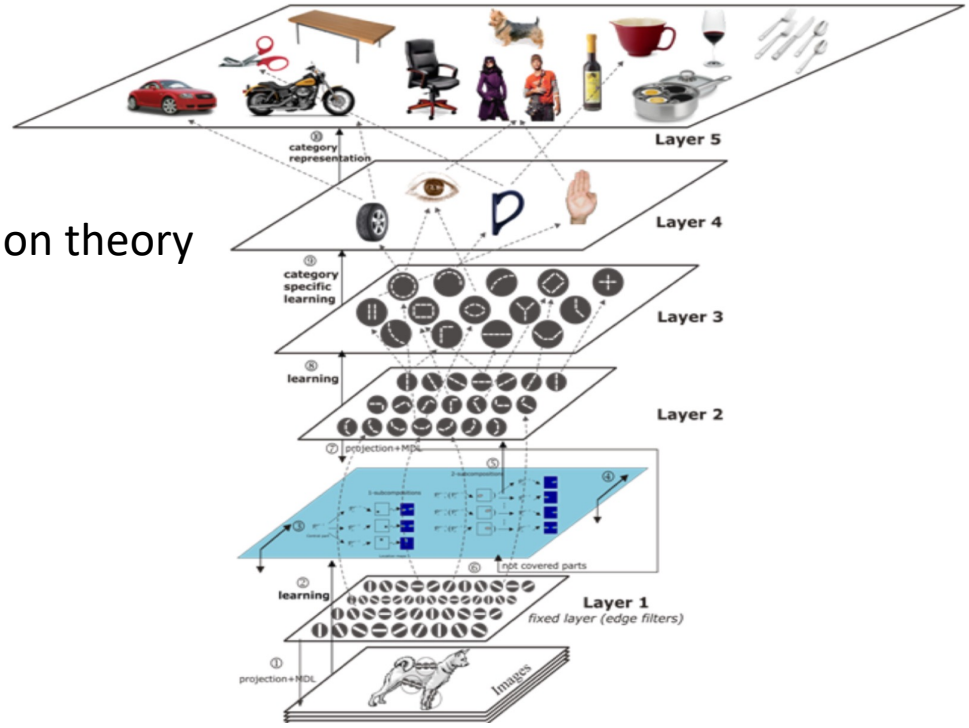
$$h_t = o_t * \tanh\left(C_t\right)$$

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Discussion

- Discuss the advantages and limitations of the deep neural network representation learning under the local-to-global assumption.

Hubel-Wiesel's feature detection theory



Topological Structure in Visual Perception

Abstract. *Three experiments on tachistoscopic perception of visual stimuli demonstrate that the visual system is sensitive to global topological properties. The results indicate that extraction of global topological properties is a basic factor in perceptual organization.*

A primitive and general function of the visual system may be the perception of global topological properties (*1*). Consider the relationship of figure to background. When a perceptual stimulus is a tachistoscope controlled the duration of visual information processing and this topological structure was revealed.

From our intuitive experience, circles, triangles, and squares are seen to be far

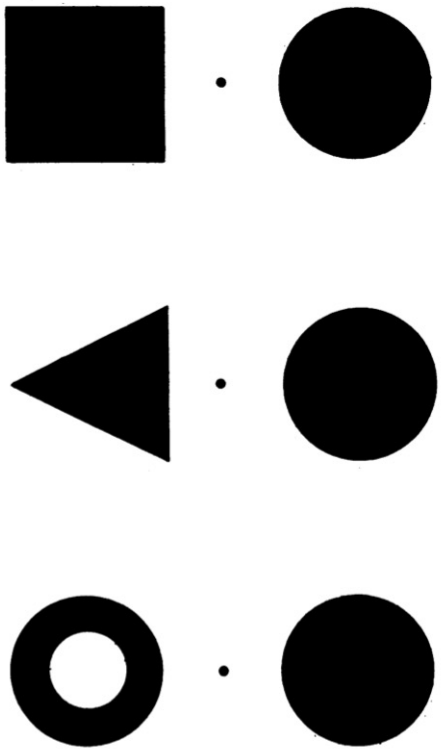


Fig. 1. The three stimulus displays used in experiment 1.

experiment has not been tried yet. Now, under conditions of tachistoscopic presentation, his hypothesis about the topological structure in visual perception has been supported.

Some of the configural superiority effects reported (*6*) may be due to topological properties, such as connectedness or closedness. Indeed, connectedness and closedness have been considered important structural components of perceptual representation (*7*). A second experiment

SCIENCE, VOL. 218, 12 NOVEMBER 1982

L. CHEN

China University of Science and Technology, Hofei, Anhuei Province, People's Republic of China, and *Center for Human Information Processing, University of California at San Diego, La Jolla 92093*

VISUAL COGNITION, 2005, 12 (4), 553–637

Psychology Press
Taylor & Francis Group

The topological approach to perceptual organization

Lin Chen

Key Laboratory of Cognitive Science, Graduate School and Institute of Biophysics, Chinese Academy of Sciences, Beijing, China

To address the fundamental question of ``what are the primitives of visual perception'', a theory of topological structure and functional hierarchy in visual perception has been proposed. This holds that the global nature of perceptual organization can be described in terms of topological invariants, global topological perception is prior to the perception of other featural properties, and the primitives of visual form perception are geometric invariants at different levels of structural stability. In Part I of this paper, I will illustrate why and how the topological approach to perceptual organization has been advanced. In Part II, I will provide empirical evidence supporting the early topological perception, while answering some commonly considered counter accounts. In Part III, to complete the theory, I will apply the mathematics of tolerance spaces to describe global properties in discrete sets. In Part IV, I will further present experimental data to demonstrate the global-to-local functional hierarchy in form perception, which is stratified with respect to structural stability defined by Klein's Erlangen Program. Finally, in Part V, I will discuss relations of the global-to-local topological model toother theories: The topological approach reformulates both classical Gestalt holism and Gibson's direct perception of invariance, while providing a challenge to computational approaches to vision based on the local-to-global assumption.

RESEARCH ARTICLE

PLOS COMPUTATIONAL BIOLOGY

Deep convolutional networks do not classify based on global object shape

December 7, 2018

Nicholas Baker[1]*, **Hongjing Lu**[1], **Gennady Erlikhman**[2¤], **Philip J. Kellman**[1]

1 Department of Psychology, University of California, Los Angeles, Los Angeles, California, United States of America, **2** University of Nevada, Reno, Nevada, United States of America

¤ Current address: Department of Psychology, University of California, Los Angeles, Los Angeles, California, United States of America
* nbaker9@ucla.edu

(a) (b)

Fig 1. Demonstration of the importance of global shape in object recognition. (a) Silhouette of a bear; (b) Scrambled natural image of a bear (See text). Image URLs are in S2 File.

Display Image	Shape-Object	Texture-Object	1st Choice	2nd Choice	3rd Choice	4th Choice	5th Choice
	Otter (0%)	Odometer (0.03%)	Can Opener (12.12%)	Electric Guitar (7.66%)	Hook (3.64%)	Remote Control (3.54%)	Corkscrew (3.53%)
	Ram (31.8%)	Bison (0.06%)	Ibex (46.51%)	Ram (31.8%)	Bighorn (18.1%)	Chesapeake Bay Retriever (1.09%)	Hyena (0.48%)
	Sturgeon (0.02%)	Honeycomb (0.48%)	Starfish (15.36%)	Banded Gecko (5.6%)	Electric Ray (5.57%)	Snail (5.56%)	Kite (4.52%)
	Bee (0.04%)	Velvet (6.97%)	Stole (29.94%)	Wool (10.24%)	Velvet (6.97%)	Bonnet (4.53%)	Poncho (4.2%)
	Bison (0.01%)	Stone Wall (8.72%)	Stone Wall (8.72%)	Parachute (7.9%)	Tile Roof (5.19%)	Stole (3.6%)	Kite (3.26%)
	Dugong (0.03%)	Gorilla (0%)	Bluetick (8.53%)	German Short-Haired Pointer (7.73%)	Egyptian Cat (6.19%)	Tabby (5.79%)	Kerry Blue Terrier (5.6%)

Thank you!