

Convolutional Neural Networks

Introduction to Machine Learning – GIF-7015

Professor: Christian Gagné

Week 10



10.1 Convolution and image processing

Convolution

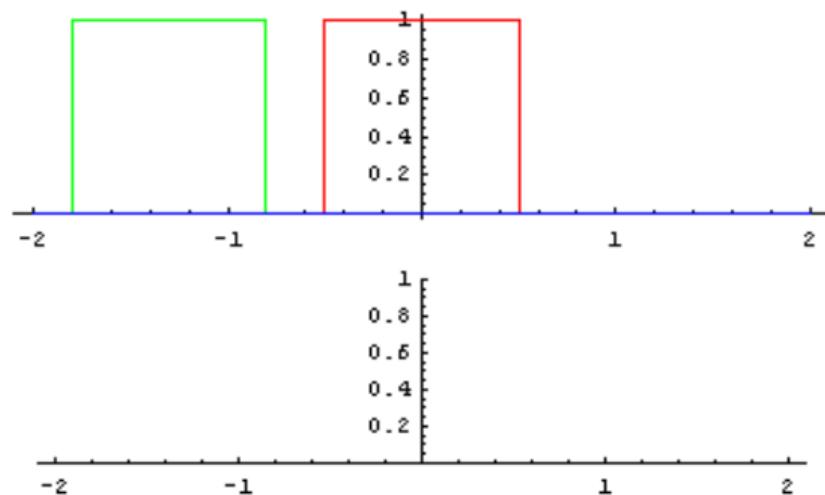
- Convolution: product of two functions on the same domain

$$f(x) * g(x) \equiv \int_{t=-\infty}^{\infty} f(x-t) g(t) dt$$

- Discrete formulation

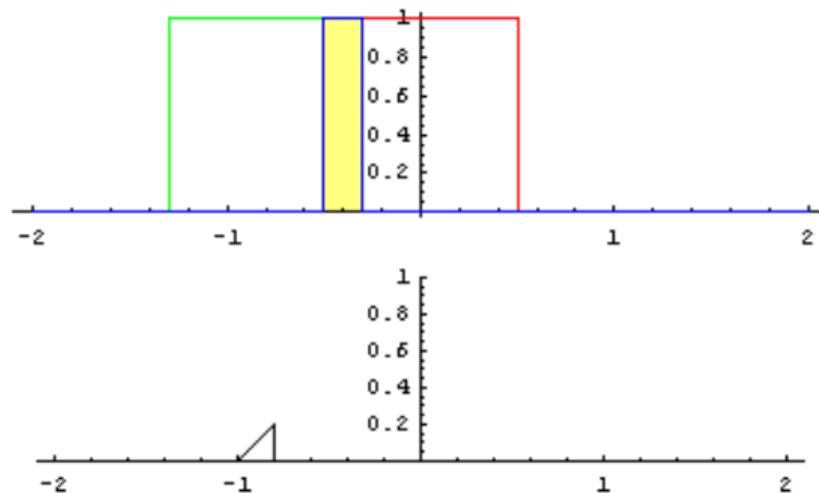
$$f(x) * g(x) \equiv \sum_{t=-\infty}^{\infty} f(x-t) g(t)$$

Convolution example



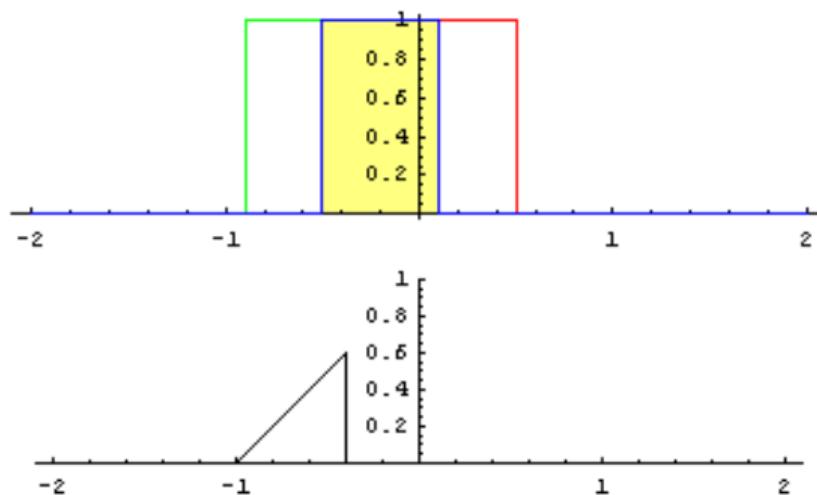
By Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Convolution example



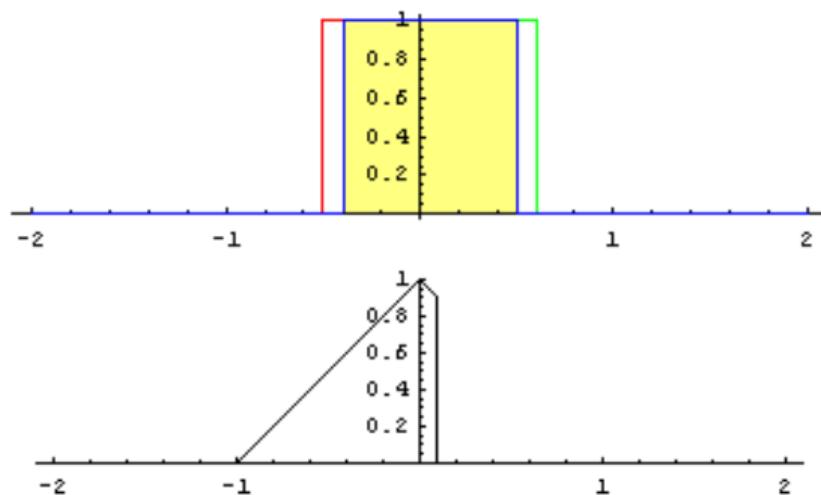
By Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Convolution example



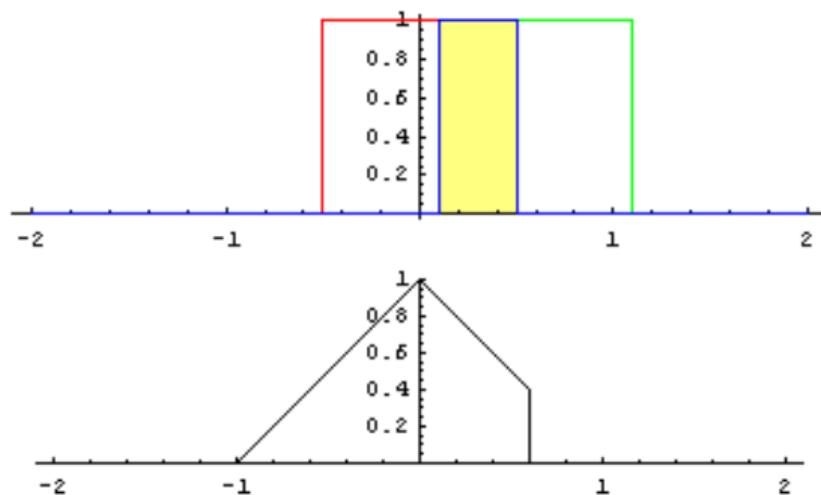
By Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Convolution example



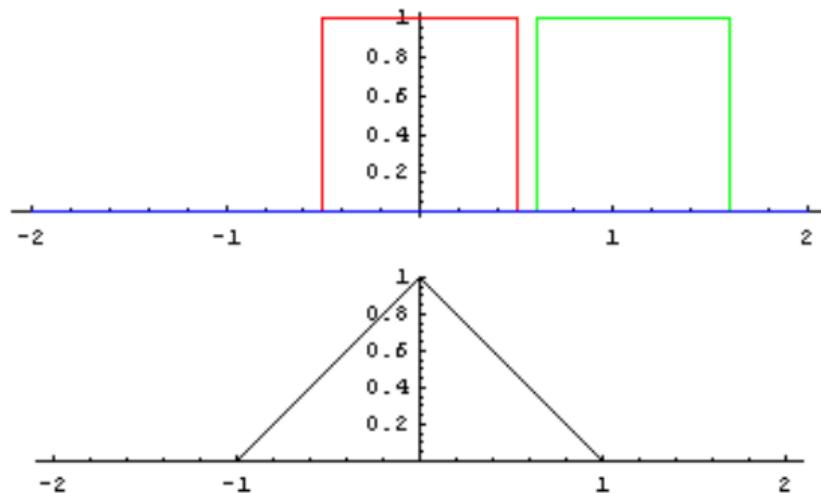
By Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Convolution example



By Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Convolution example



By Lautaro Carmona, CC-SA 4.0, https://commons.wikimedia.org/wiki/File:Convolucion_Funcion_Pi.gif.

Convolution and density estimation

- Off-center Dirac distribution

$$\delta(x - t) = \begin{cases} \infty & \text{if } x = t \\ 0 & \text{otherwise} \end{cases}, \quad \int_{x=-\infty}^{\infty} \delta(x - t) dx = 1.$$

- Convolution on off-center Diracs

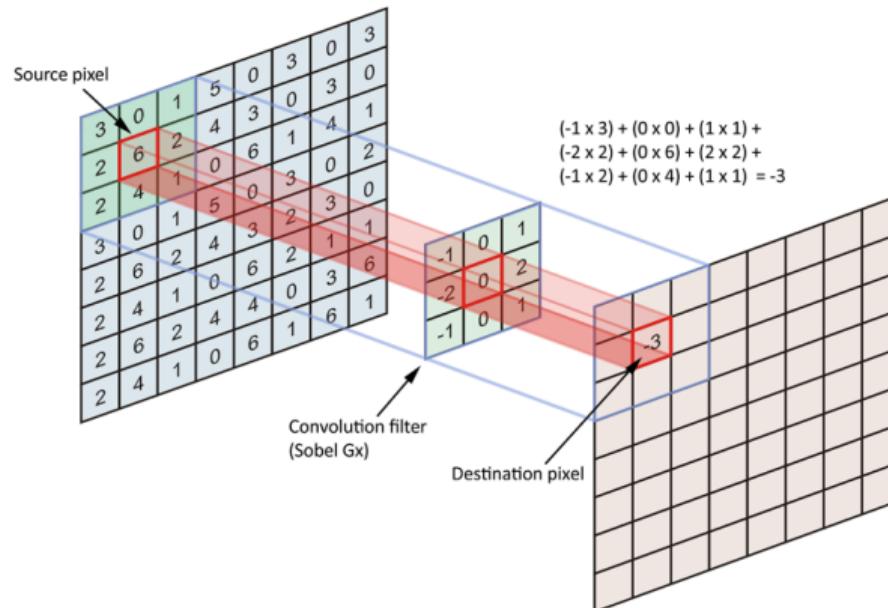
$$f(x) * \delta(x - u) = f(x - u)$$

- Kernel density estimation: kernel convolution with several Diracs centred on the data

$$\hat{p}(x) = \frac{1}{Nh} \sum_{t=1}^N K\left(\frac{x - x^t}{h}\right) = \frac{1}{Nh} \sum_{t=1}^N K\left(\frac{x}{h}\right) * \delta(x - x^t)$$

Image processing

- 2D convolution is a building blocks for image processing



Source: <https://thigiacmaytinh.com/wp-content/uploads/2018/05/kernel.png>, accessed November 13, 2018.

Examples of filters

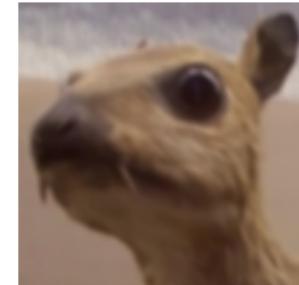
Identity (3×3):

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



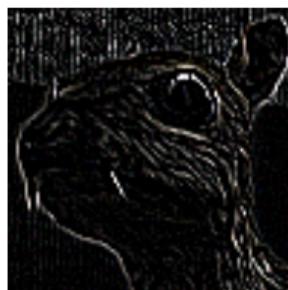
Gaussian blur:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$



Edge detection:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



Sharpen:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



By Michael Plotke, CC-BY-SA 3.0, <https://commons.wikimedia.org/wiki/File:Vd-Orig.png>,

<https://commons.wikimedia.org/wiki/File:Vd-Blur1.png>,

<https://commons.wikimedia.org/wiki/File:Vd-Edge3.png>, <https://commons.wikimedia.org/wiki/File:Vd-Sharp.png>.

Sobel operator

- Classic filter for edge detection
 - Compute local gradients of image intensity
 - Uses two convolutions to obtain the vertical gradient \mathbf{G}_x and the horizontal gradient \mathbf{G}_y of an image \mathbf{A} , the result is an image $\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$

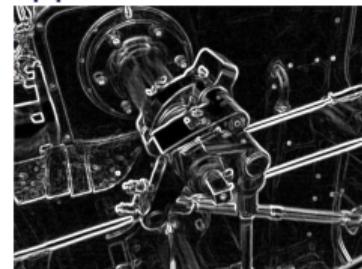
$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}, \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Original image:



By Simpsons contributor, CC-BY-SA 3.0, [https://commons.wikimedia.org/wiki/File:Valve_original_\(1\).PNG](https://commons.wikimedia.org/wiki/File:Valve_original_(1).PNG)

Application of Sobel:



By Simpsons contributor, CC-BY-SA 3.0, [https://commons.wikimedia.org/wiki/File:Valve_sobel_\(3\).PNG](https://commons.wikimedia.org/wiki/File:Valve_sobel_(3).PNG)

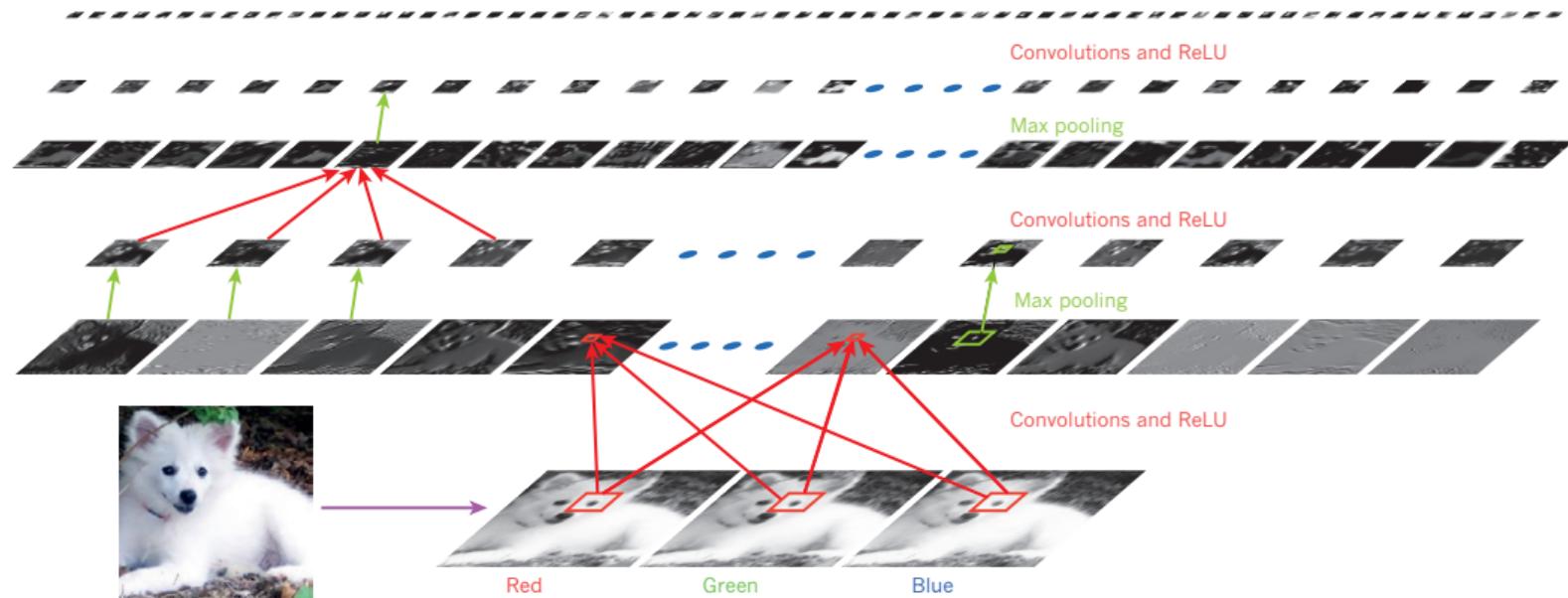
10.2 Convolutional neural networks

Convolutional neural networks

- Idea: Define neural networks with convolution operations
 - Learning the numerical values of convoluted filters
 - Define a network exploiting elements of the data structure
 - Sound or speech: temporal data (1D convolutions)
 - Image: spatial data (2D convolutions)
 - Video: spatio-temporal data (3D convolutions)
 - Sequence of convolution stages, filtering output of previous layer
 - Allows for more compact modelling than fully connected networks and translation invariant
- Some components of a convolution network
 - Layer of convoluted filters on the different channels
 - Pooling: maximum (max pool) or average (avg pool) value in a certain convoluted window
 - Transfer functions: ReLU, etc.
 - Near output, fully connected layers (like with multi-layer perceptron)

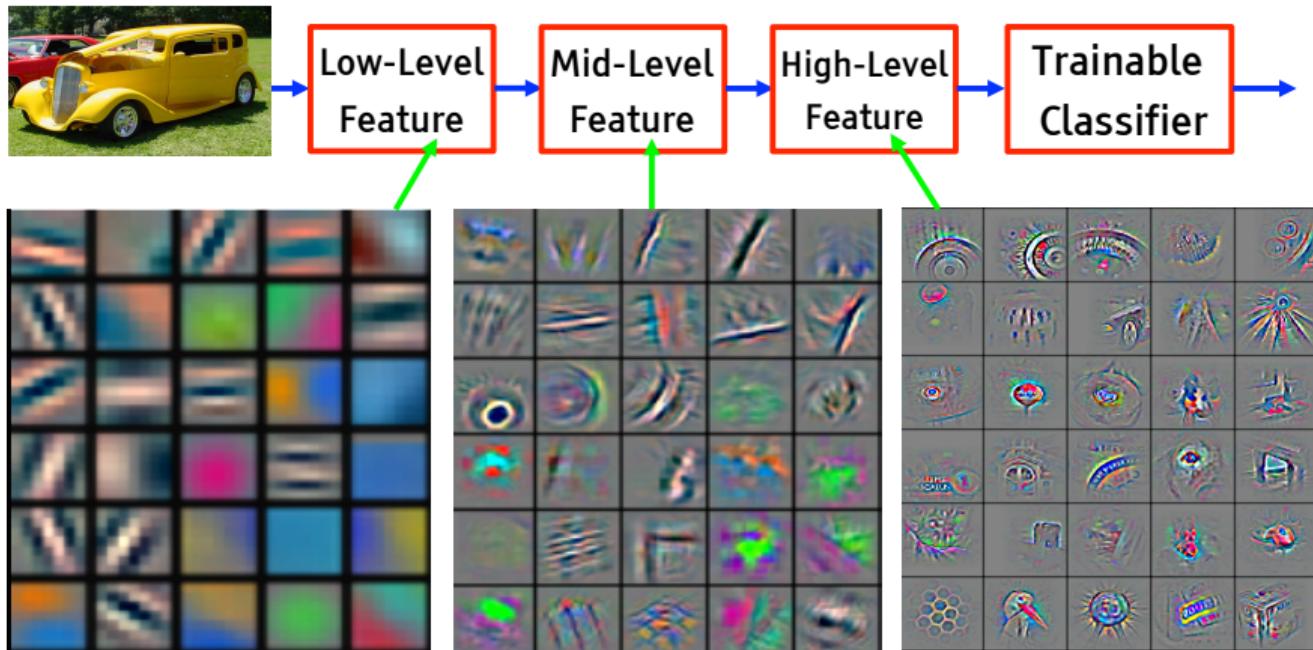
Convolution network

Samoyed (16); Papillon (5.7); Pomeranian (2.7); Arctic fox (1.0); Eskimo dog (0.6); white wolf (0.4); Siberian husky (0.4)



From Y. LeCun, Y. Bengio and G. Hinton, Deep Learning, Nature, vol. 521, 28 mai 2015. Accessed online November 6, 2020 at <https://www.nature.com/articles/nature14539>.

Filters composition

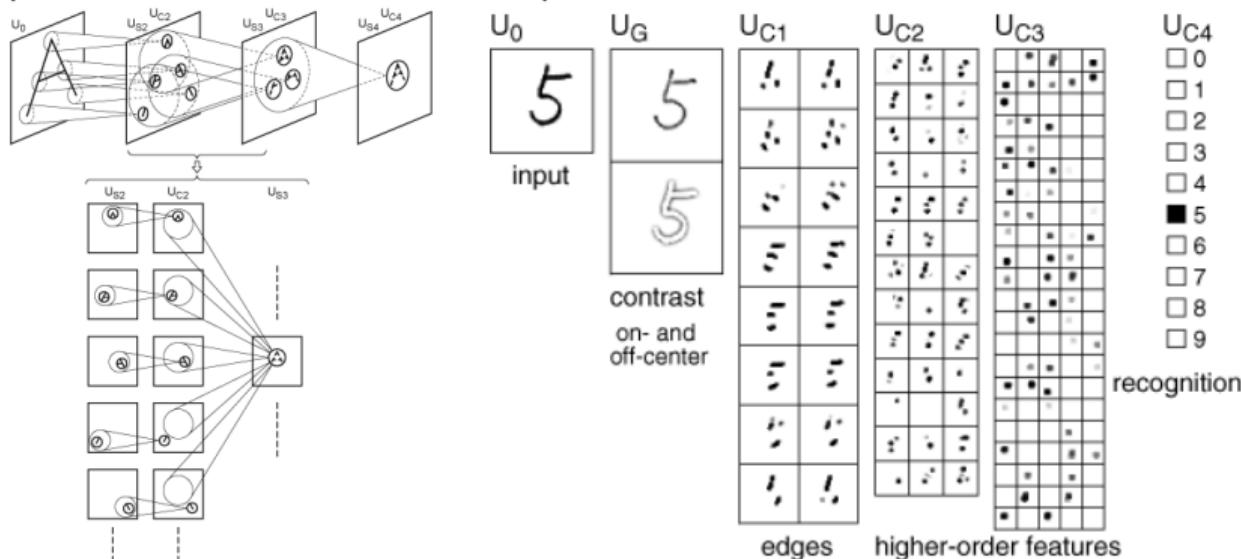


From G. Hinton, Y. Bengio and Y. LeCun, Deep Learning NIPS'15 Tutorial, 2015. Accessed online on November 6, 2020 at <https://nips.cc/Conferences/2015/Schedule?showEvent=4891>.

10.3 Examples of convolution networks

Neocognitron

- Neocognitron: the ancestor of convolutional networks
 - Proposed by Kunihiko Fukushima in the 1980s
 - Inspiration for LeCun in the development of convolutional networks

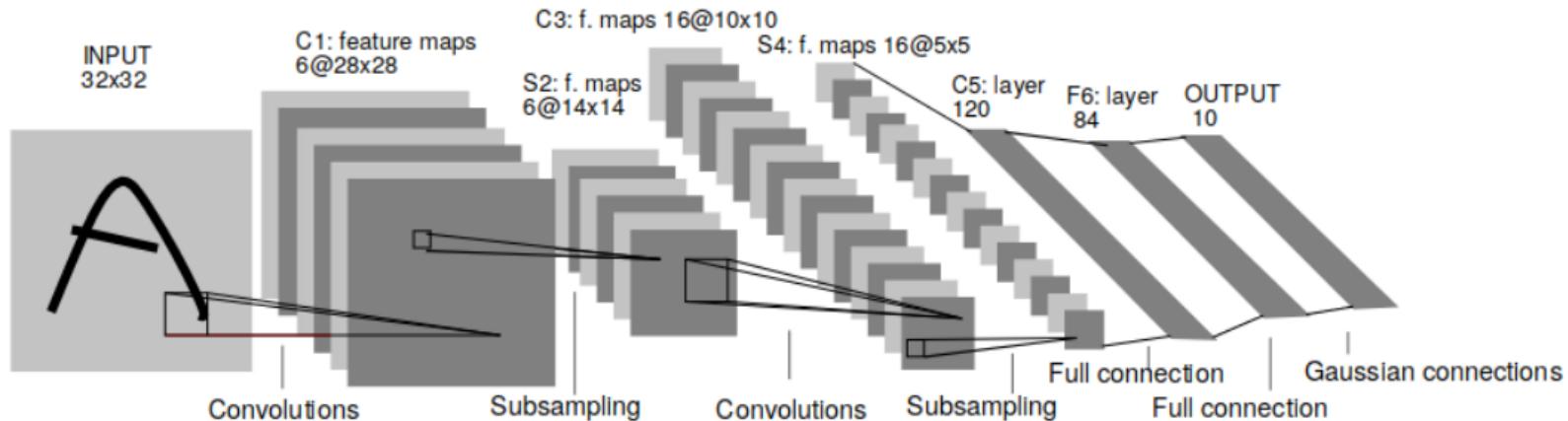


Source: <http://www.scholarpedia.org/article/File:ScholarFig2.gif>,
accessed November 13, 2018.

Source:
<http://www.scholarpedia.org/article/File:ScholarFig4.gif>,
accessed November 13, 2018.

LeNet5

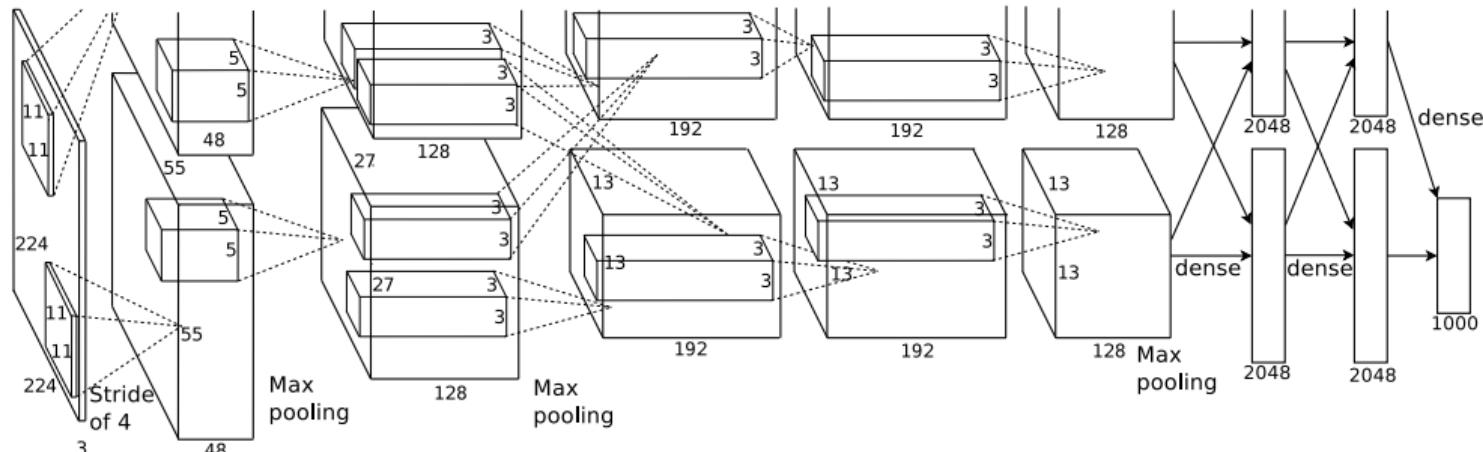
- LeNet5: classical convolutional network, proposed in the 1990s
 - 3 convolution layers, 2 average pooling layers, 2 fully connected layers
 - 60k parameters (from 10M to 100M with modern networks)



From Y. LeCun, L. Bottou, Y. Bengio et P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, 86(11), 1998. Accessed online on November 6, 2020 at <http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf>.

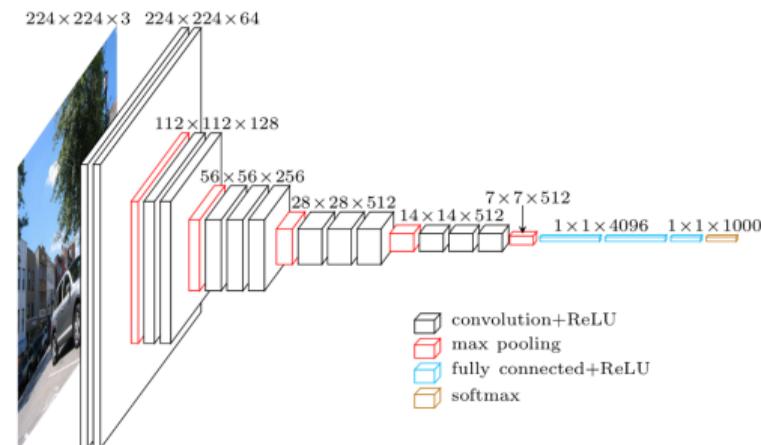
AlexNet

- AlexNet: network for object recognition
 - Winner of the ImageNet 2012 contest
 - Implemented for GPU Computing
 - Often used as a basic model for representation transfer
 - 8 convolution layers, some max pooling layers, 3 fully connected layers



From A. Krizhevsky, I. Sutskever, et G. Hinton, *Imagenet classification with deep convolutional neural networks*. NIPS, 2012. Accessed online November 6, 2020 at <https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.

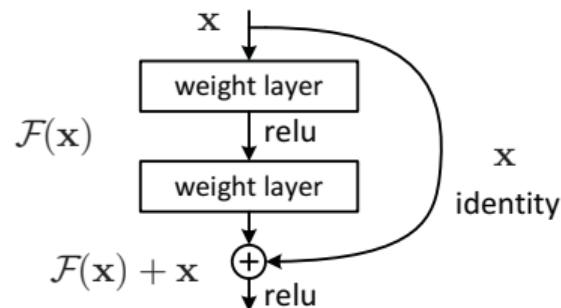
- VGGNet: greater depth with simplified topology
 - Winner of the ImageNet 2013 contest
 - Depth is critical for good performance
 - Similar to AlexNet, but with only 3×3 convolutions, 2×2 max pooling, 3 layers fully connected and 16 layers in total (VGG-16)



Source: <https://heuritech.files.wordpress.com/2016/02/vgg16.png>, accessed November 13, 2018.

ResNet

- Residual networks: allowing direct connections between non-adjacent layers (*skip links*)



From K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition. CVPR, 2016. Accessed online November 6, 2020 at <https://arxiv.org/abs/1512.03385>.

- Allows for much deeper and more efficient networks
 - Winner of ImageNet 2015 competition (3.57 % top 5 error)
 - Facilitates signal optimization and propagation across the network
 - Residual block must do better than a treatment directly on the previous block

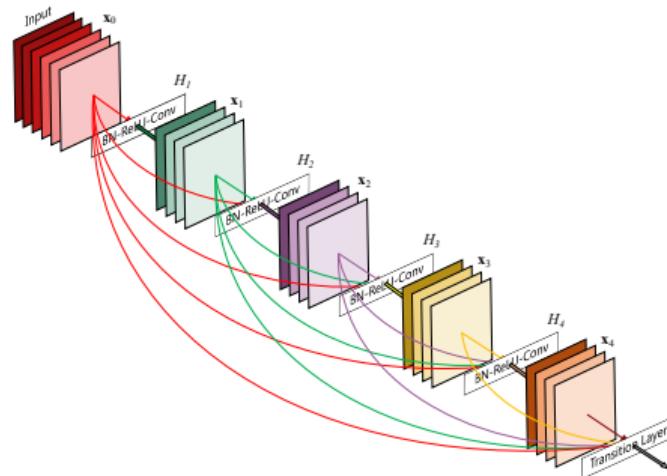
ResNet



From K. He, X. Zhang, S. Ren, et J. Sun, Deep residual learning for image recognition. CVPR, 2016. Accessed online November 6, 2020 at <https://arxiv.org/abs/1512.03385>.

DenseNet

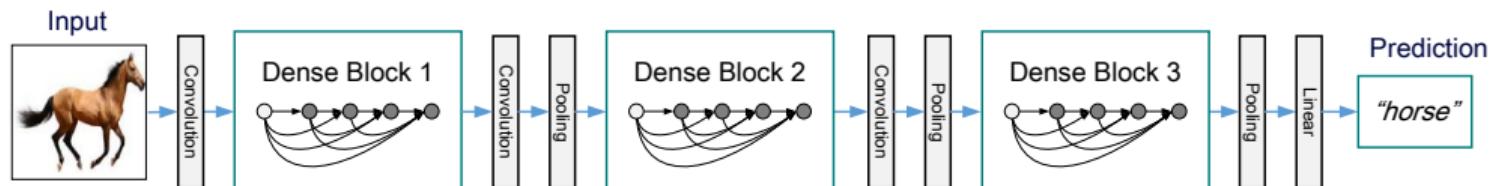
- Observation: convolution networks can be deeper and get better performance with close connections throughout the network at its input.
- DenseNet: connect each layer to all of the above layers
 - Network with L layers will have $L(L + 1)/2$ direct connections between layers



From G. Huang, Z. Liu, L. Van Der Maaten et K.Q. Weinberger, *Densely Connected Convolutional Networks*. CVPR, 2017. Accessed online on November 6, 2020 at <https://arxiv.org/abs/1608.06993>.

DenseNet

- In practice, we create dense blocks separated by convolution and pooling layers

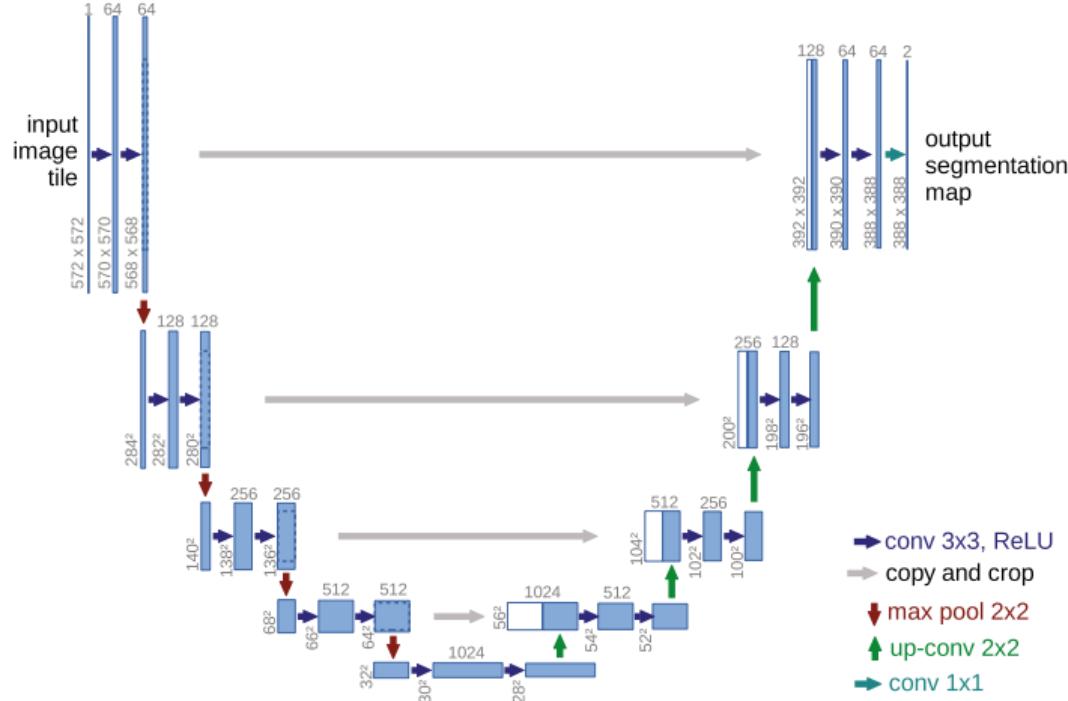


From G. Huang, Z. Liu, L. Van Der Maaten et K.Q. Weinberger, *Densely Connected Convolutional Networks*. CVPR, 2017. Accessed online on November 6, 2020 at <https://arxiv.org/abs/1608.06993>.

- Each layer in a dense block can be relatively narrow, i.e. can contain few neurons.

- Networks presented so far first proposed and tested for object recognition (classification)
 - Other possible tasks in vision: detection, tracking, etc.
- Segmentation: identify coherent regions of the image
 - Separate the different regions
 - Give a label to each region
- U-Net: network proposed for biomedical imaging
 - Fully convolutional network, gives an output image
 - Compression of information in a network environment, similar to an auto-encoder
 - Skip links allow to preserve spatial structure

U-Net



From O. Ronneberger, P. Fischer, et T. Brox, U-net: Convolutional networks for biomedical image segmentation. MICCAI, 2015. Accessed online on November 6, 2020 at <https://arxiv.org/abs/1505.04597>.