Convolutional Neural Networks (CNNs)

Kamil Bogus

4th November 2022

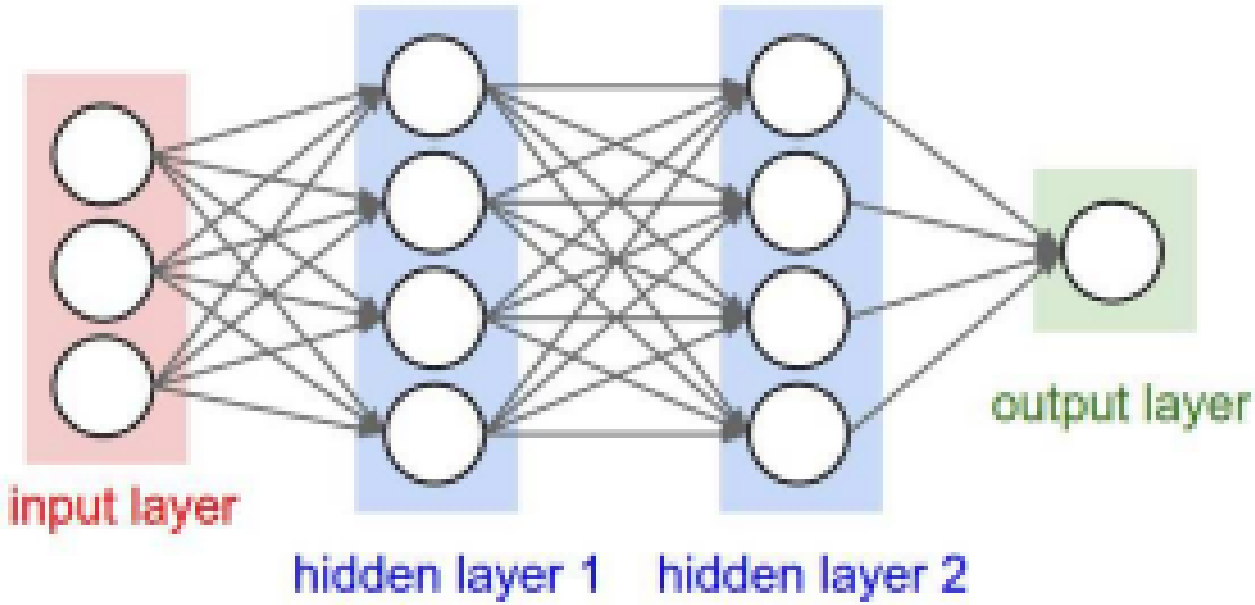
Motivations

We are interested in the processing of (potentially large) images, i.e.

- recognition (detection),
- assigning images to specific classes, and
- area segmentation.

Dense Neural Networks (DNN)

In Dense Neural Network all neurons between layers are fully connected:



Rysunek: https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/

Problems with DNNs

- Even simple networks with e.g. two layers can contain billions of parameters to train ...
- and this requires a lot of computing power, which affects time and costs!...
- The problem is also the potential loss of information resulting from the change of the shape of the photo.

Convolutional Neural Networks (CNNs)

In a convolutional network, neurons are connected only to a certain area in the preceding layer.

This area is determined by the size of the so-called filter, and the depth is derived from the number of filters used for the convolution operation.

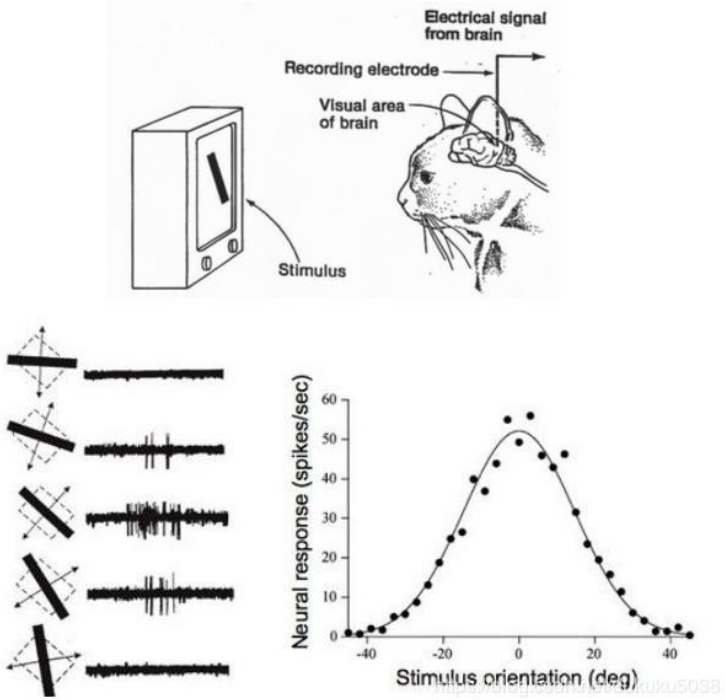
Applications of CNNs

Convolutional Neural Nets are mainly used in image processing, helping to:

- determine the likelihood of assigning an image to learned classes,,
- detect objects in the image and interpret the relationships between them,
- divide into areas.

History - the works by Hubel and Wiesel

In the late 1950s and early 1960s, **David Hubel and Torsten Wiesel** discovered two types of cells in the primary visual cortex called simple cells and complex cells.

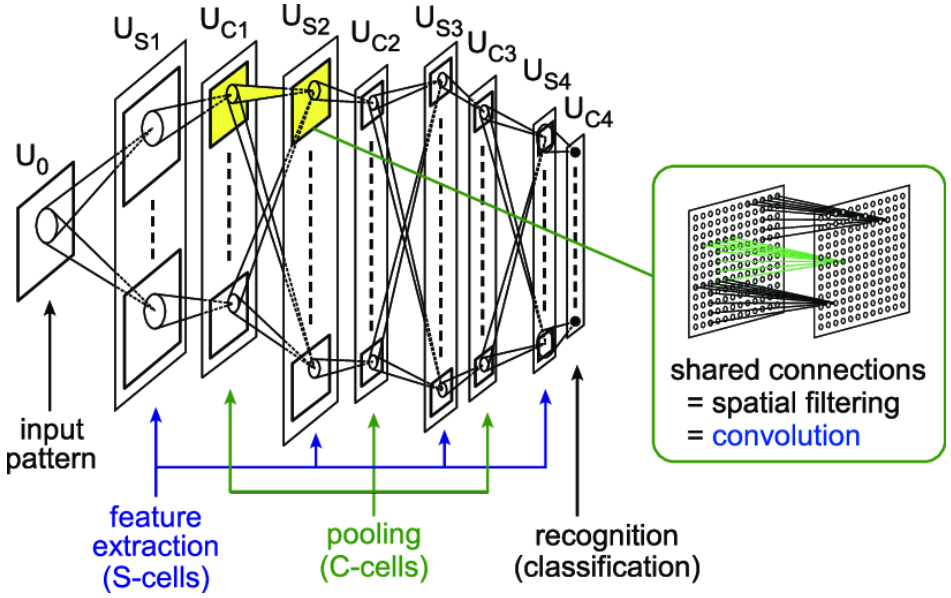


Rysunek: https://www.programmersought.com/article/3392117538/

They also proposed a cascade model of these two types of cells to be used in pattern recognition. For their research, they received the Nobel Prize in Medicine in 1981.

History - the works of Fukushima

In 1979 **Kunihiko Fukushima** develops the neocognitron.



Rysunek: Kunihiko Fukushima, Recent advances in the deep CNN neocognitron

This hierarchical, multi-layer artificial neural network was used for handwritten character recognition in Japanese and other recognition tasks.

History - LeCun's works

- **Yann LeCun** used backpropagation to modify the kernel coefficients for the handwritten picture digit (MNIST) image set.
- From then on, network learning became fully automatic, performed better than manually designing coefficients, and was suited to a wider range of image recognition problems and image types.
- In 1998, the LeNet-5 network (Y. LeCun, L. Bottou, Y. Bengio, P. Haffner) is created using previously known elements (fully interconnected layers, sigmoidal activation function, etc.) with new elements: **convolution layers and connecting layers**.

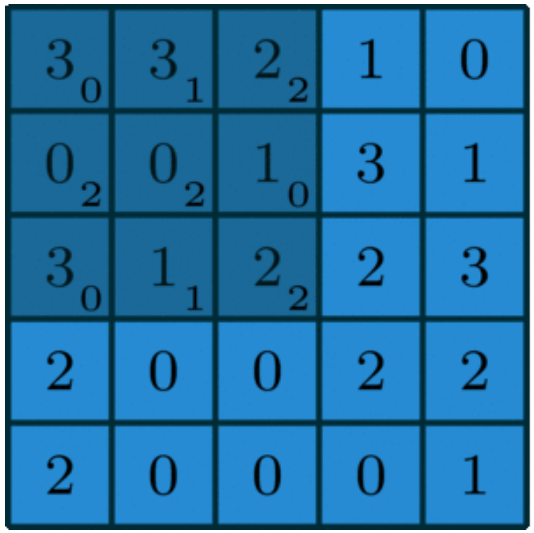
What is convolution?

- Convolution measures the integral of the point product of a function

$$(f \star g)(y) = \int_{-\infty}^{\infty} f(y-x)g(x)dx.$$

- This is a matrix transformation of image fragments to extract information about specific image characteristics.
- Such transformation (matrix) is called a filter or a kernel.
- One of the functions is a two-dimensional matrix containing the pixel values of an image (usually of large dimensions), and the second function is a filter (much smaller matrix), called the kernel convolution.
- As a result of the image convolution with the filter, we get a new image.

Convolution - example

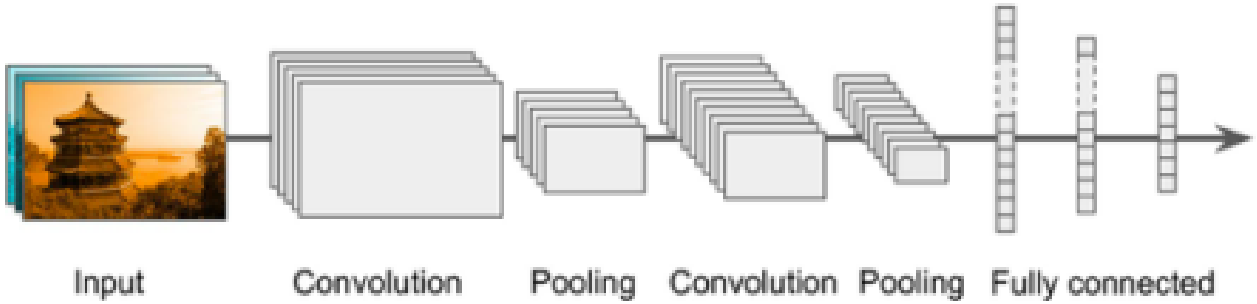


https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/

Architecture of CNNs

A typical weave network includes, among others begin itemize item input layer, item weave layers item linking layers. A typical weave network includes

- input layer,,
- convolution layers
- connecting layers.



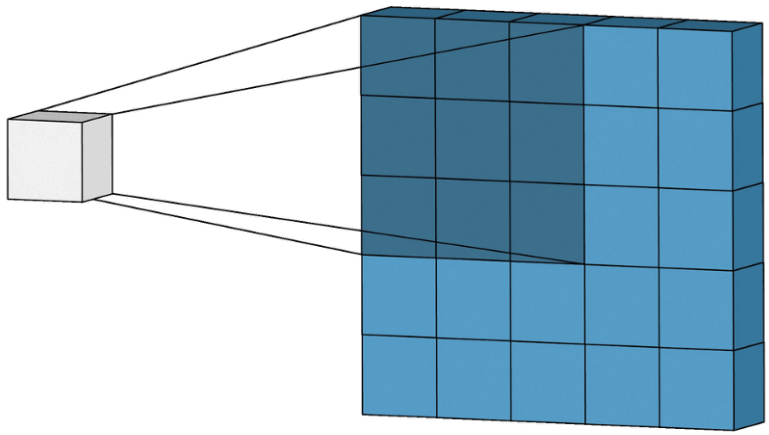
Rysunek: Aurélien Géron. Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow. Helion, 2018

Input layer

- It represents the input image to the network.
- For color photos we will be using three RGB input channels corresponding to the red, green and blue channels, respectively.
- For MNIST we have grayscale and therefore only one channel.

Convolutional layer

- They contain learned filters (kernels) that isolate the characteristics that distinguish different images from each other.
- The values in the filters are selected and optimized during network training.
- Filters are shared on the whole photo, i.e. weights are matched to the filter which is then moved over the whole photo.
- The size of the image itself does not affect the number of weights in the convolution layer.



https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/

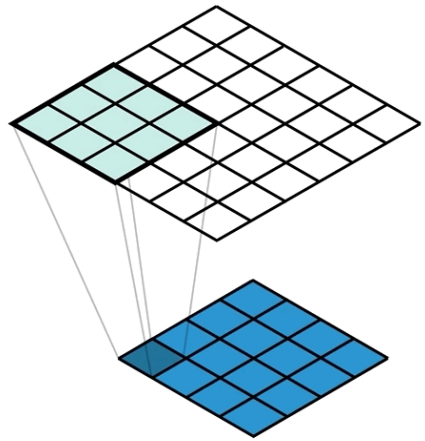
Hyperparameters for the convolutional layer

To fully define the convolutional layer, we have to provide:

- Kernel Size (filter size)
- Padding
- Strides
- Activation functions

Kernel Size

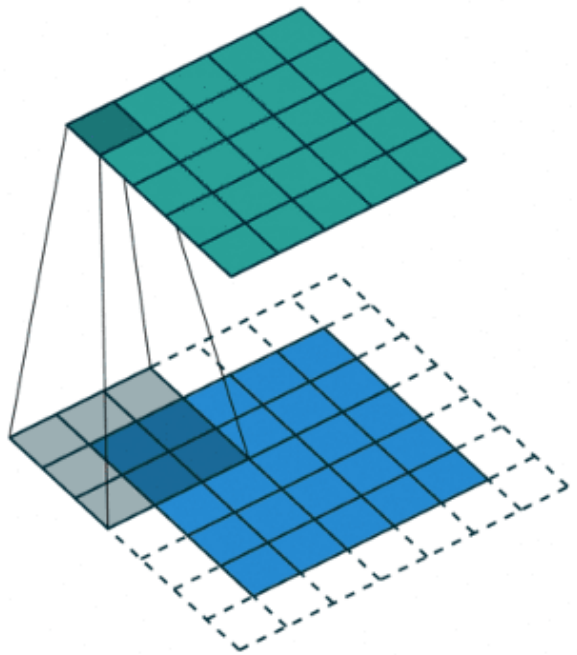
- Refers to the dimensions of the sliding window above the entrance
- Small kernels are able to extract much more information from the input containing highly local functions
- Smaller kernel size also leads to a smaller reduction in layer dimensions, allowing for deeper architecture
- A large kernel size extracts less information, leading to a faster reduction in layer dimensions, often resulting in poorer performance
- Large kernels are better suited to extracting larger items.



https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/

Padding

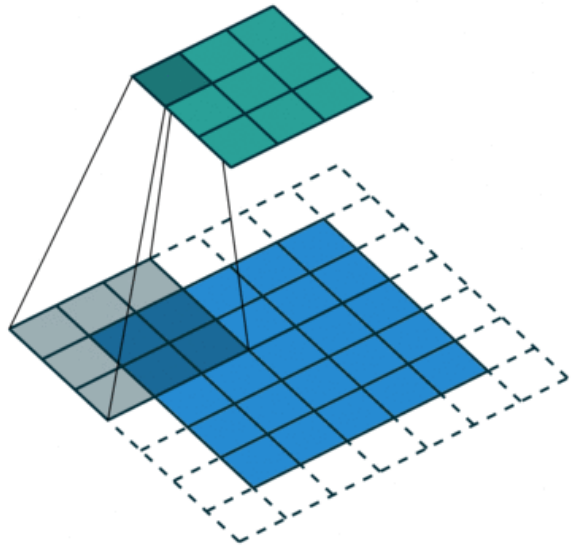
- Padding allows you to get the output size the same as the input size (assuming strides is an offset of one)
- This is achieved by adding additional (artificial) weights to the edges (usually with a value of zero).
- Thanks to the added frame, the output image (green square) is 5×5, which is the same size as the input image (blue square).
- If there was no added frame then the output size would be 3×3.



https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/

Strides

- The strides parameter is the step for moving the filter window.
- Usually it is the odd number.
- The dot product is executed on the input window e.g. 3×3 to get the output value, and then it is shifted by one pixel for each subsequent operation.
- The below-given picture shows a 3×3 kernel (gray square) around the 5×5 pixel image with strides=2 and padding=1.



https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/

Activation functions

- If there were no non-linear activation function, the deep CNN networks would transform into a single, equivalent convolution layer that would not perform as well.
- An example of a nonlinear activation function is e.g. ReLU (Rectified Linear Activation) defined by the formula:

$$ReLU(x) = \max\{x, 0\}$$

- Another commonly used function is Softmax:

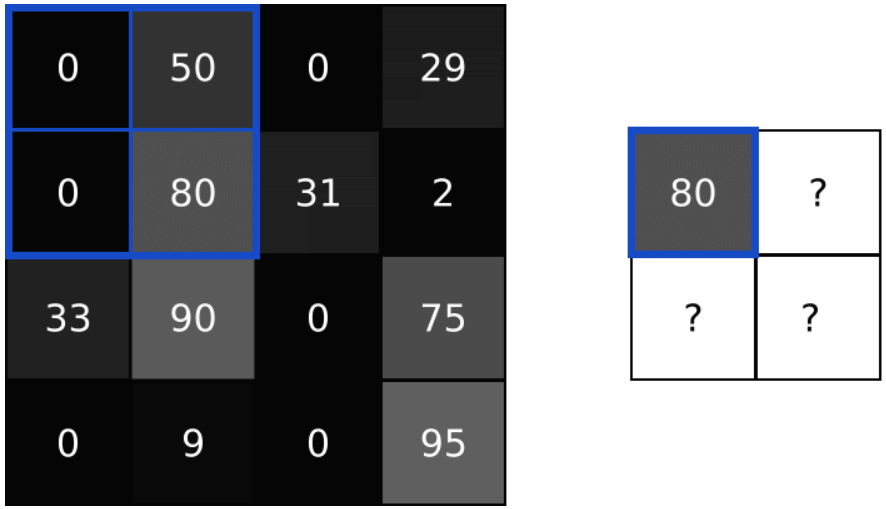
$$Softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Pooling layer

- It helps to shrink the input image and consequently to reduce overhead computational, memory usage and number of parameters.
- Every neuron that is part of the connecting layer connects to outputs of a certain number of neurons of the previous layer, located in the area of a small reception area (as in the case of convolution layers).
- We define the field size, step value, type of filling zeros, etc. (similar to convolution layers).
- Pooling layer does not contain weights.

Pooling - example

- In Pooling we map a few pixels, e.g. the size of 2×2 to 1 pixel.
- One type of pooling is max pooling, where we take the maximum value from the pixel data.
- Another example is avg pooling, where we take the average value of the pixel data.
- Below we can see the max pooling with the (size) parameter is 2, which means traversing the image using the size 2×2 and taking the maximum value as the output pixel.



https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/

Examples of CNNs

- LeNet (1998),
- AlexNet (2012),
- GoogLeNet (2014),
- VGGNet (2014).
- ResNet (2015),

Popular datasets for CNNs

- MNIST,
- Fashion-MNIST,
- CIFAR-10,
- CIFAR-100,
- STL-10,
- SVHN (The Street View House Numbers).

Bibliography

- Michel Nielsen "Neural Networks and Deep Learning", http://neuralnetworksanddeeplearning.com
- Tianyi Liu, Shuangsang Fang, Yuehui Zhao, Peng Wang, Jun Zhang "Implementation of Training Convolutional Neural Networks", https://arxiv.org/abs/1506.01195
- https://poloclub.github.io/cnn-explainer/
- https://miroslawmamczur.pl/jak-dzialaja-konwolucyjne-sieci-neuronowe-cnn/
- https://home.agh.edu.pl/ horzyk/lectures/ai/SztucznaInteligencja-UczenieGłębokichSieciNeuronowych.pdf
- http://www.cs.put.poznan.pl/alawrynowicz/SI_ML_CNN_2020_lawrynowicz.pdf
- Zeiler, M.D., Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer, Cham. https://doi.org/10.1007/978-3-319-10590-1_53