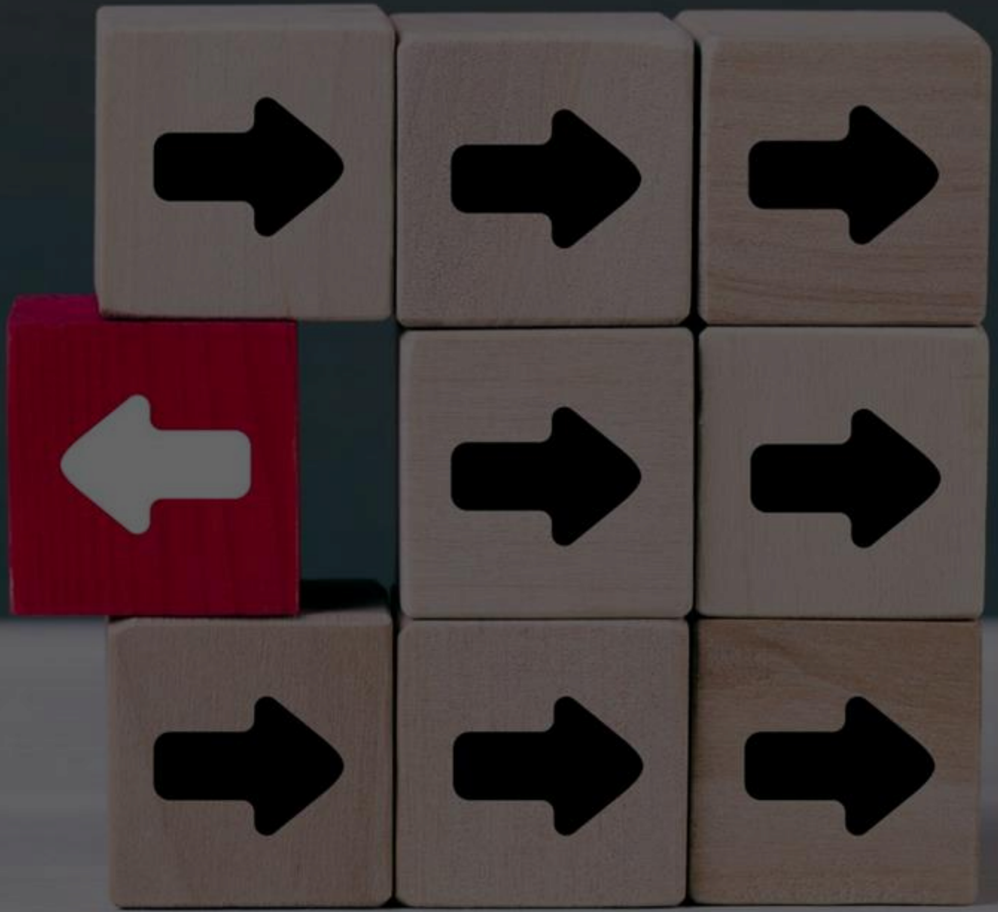
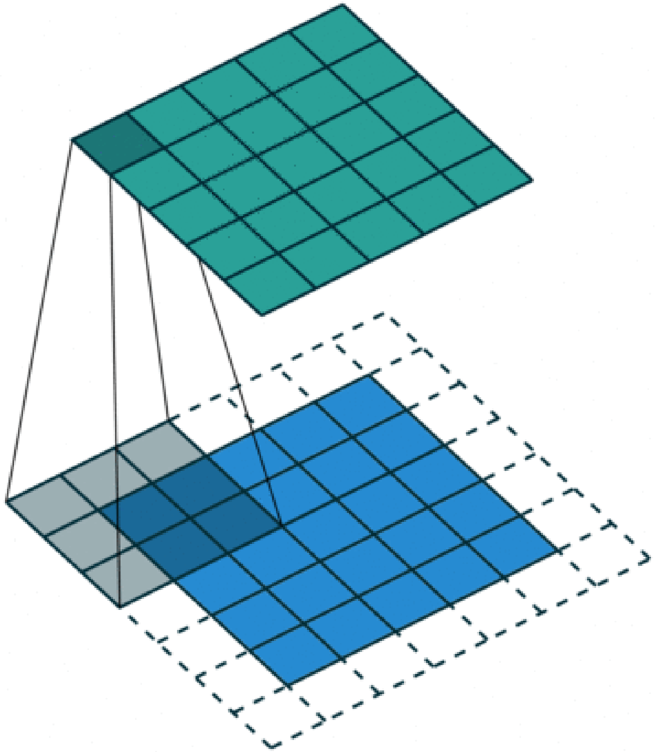
WARSZTATY BADAWCZE: „METODY PRZENOSZENIA WIEDZY W SIECIACH NEURONOWYCH”

-- SIECI CNN --

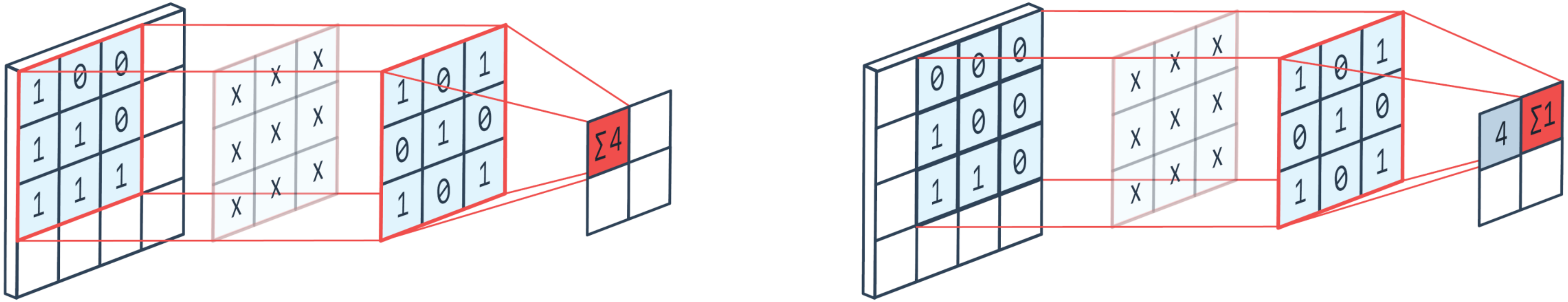


Paulina Tomaszewska

Convolutional layer – grayscale images

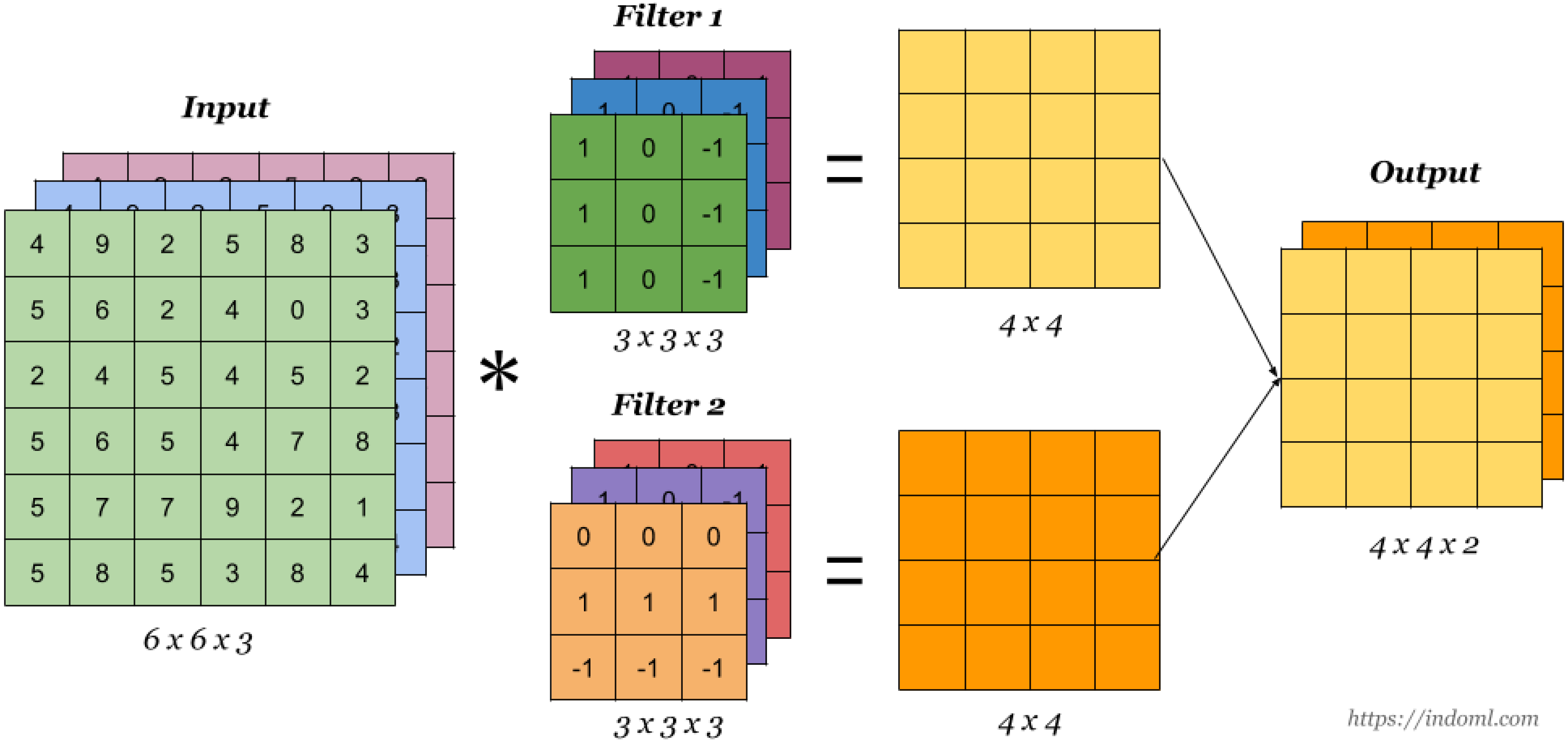


https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1



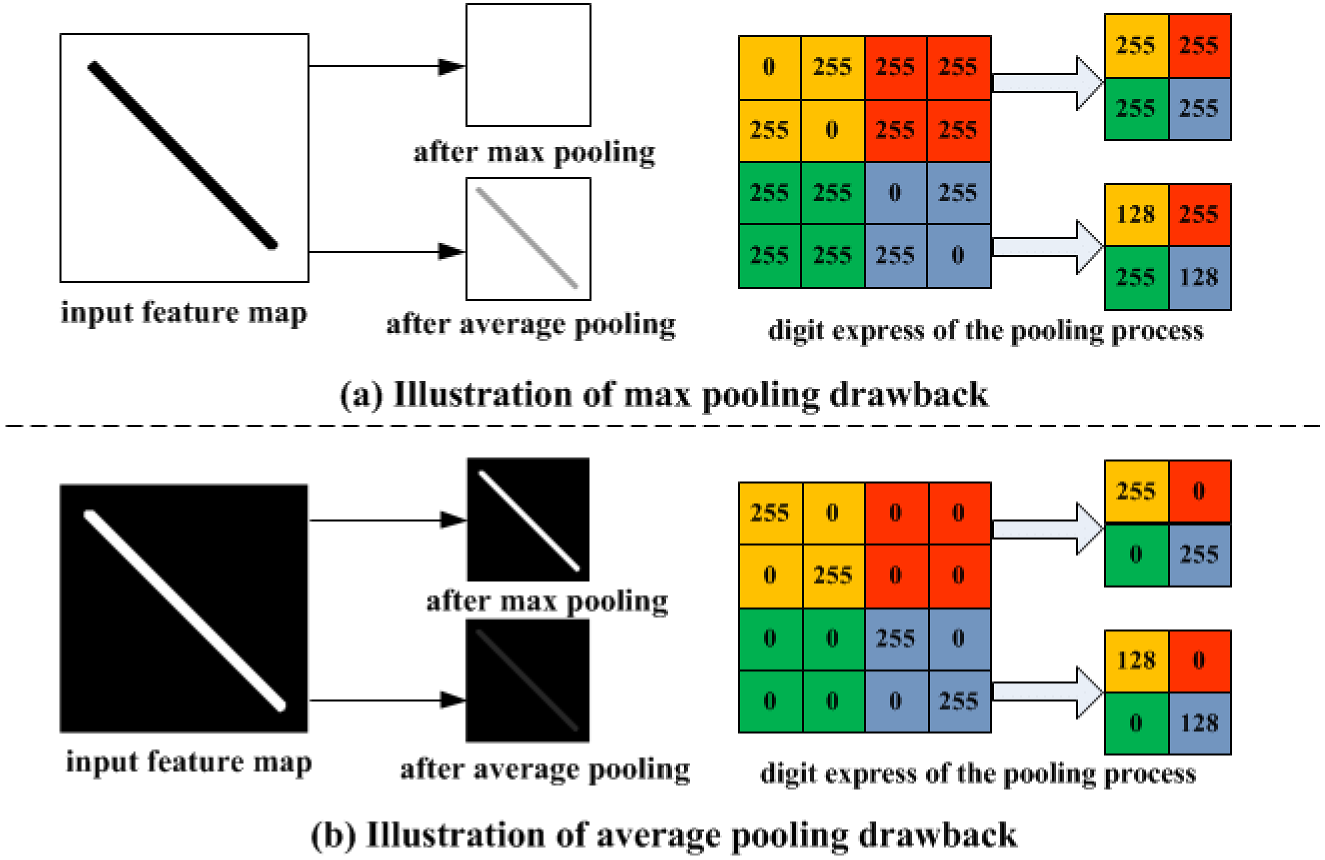
https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/blocks/2d-convolution-block

CNN for RGB



https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d

Pooling

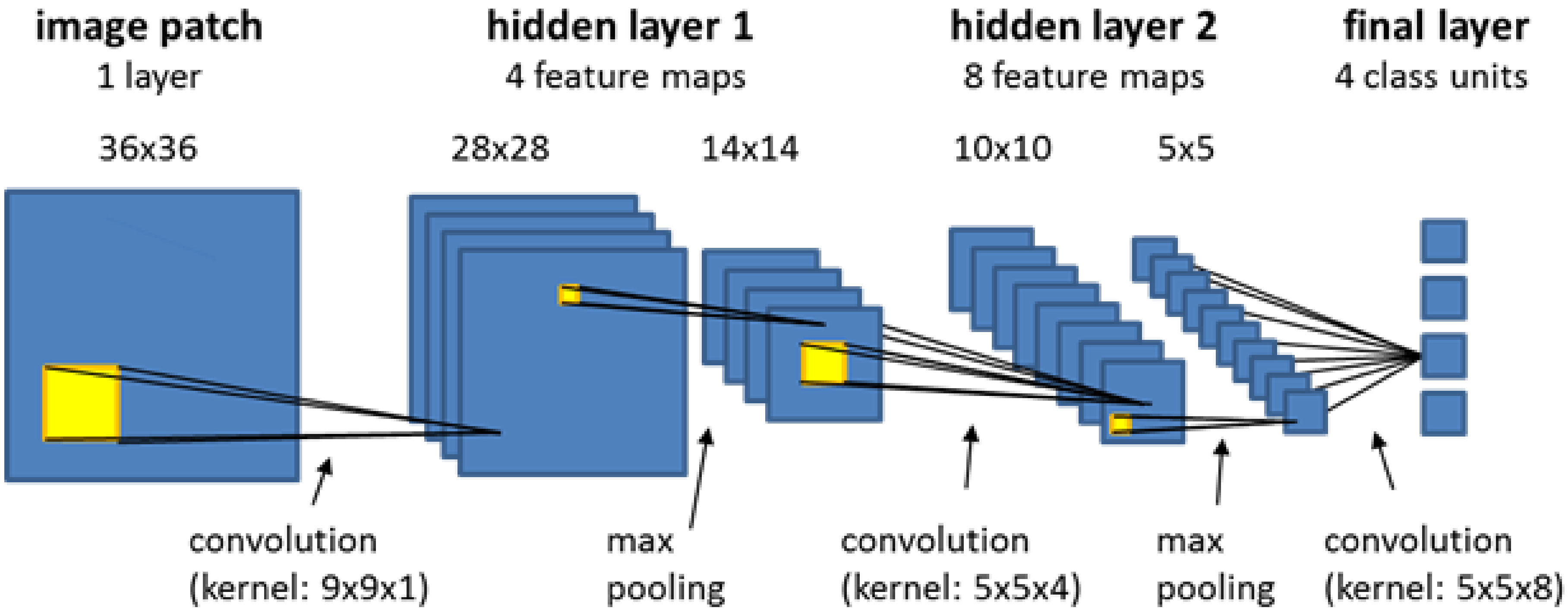


https://www.researchgate.net/figure/Toy-example-illustrating-the-drawbacks-of-max-pooling-and-average-pooling_fig2_300020038

Batch normalization

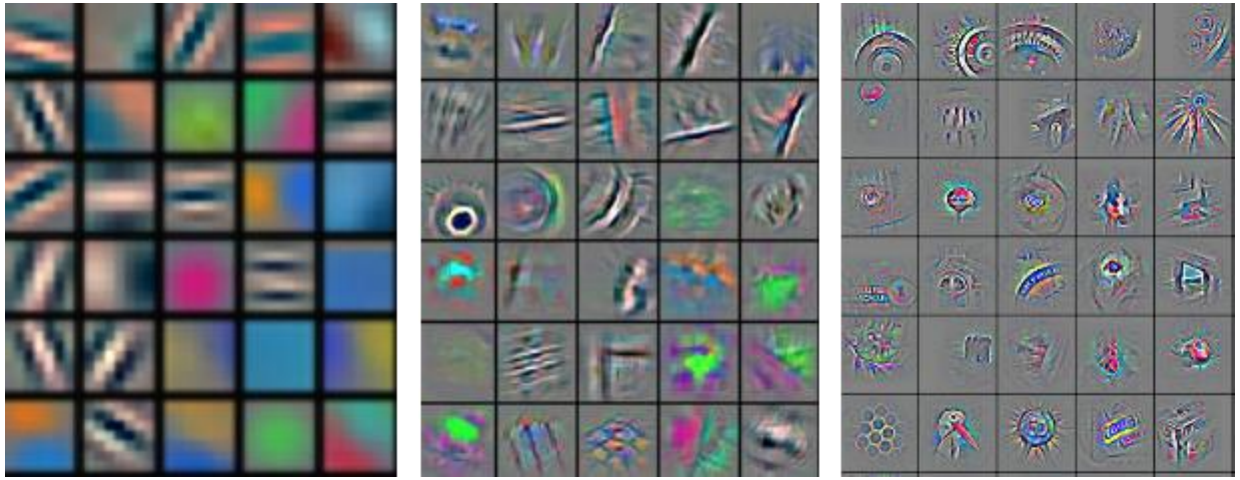
- https://learnopencv.com/batch-normalization-in-deep-networks/

Convolutional neural network (CNN)



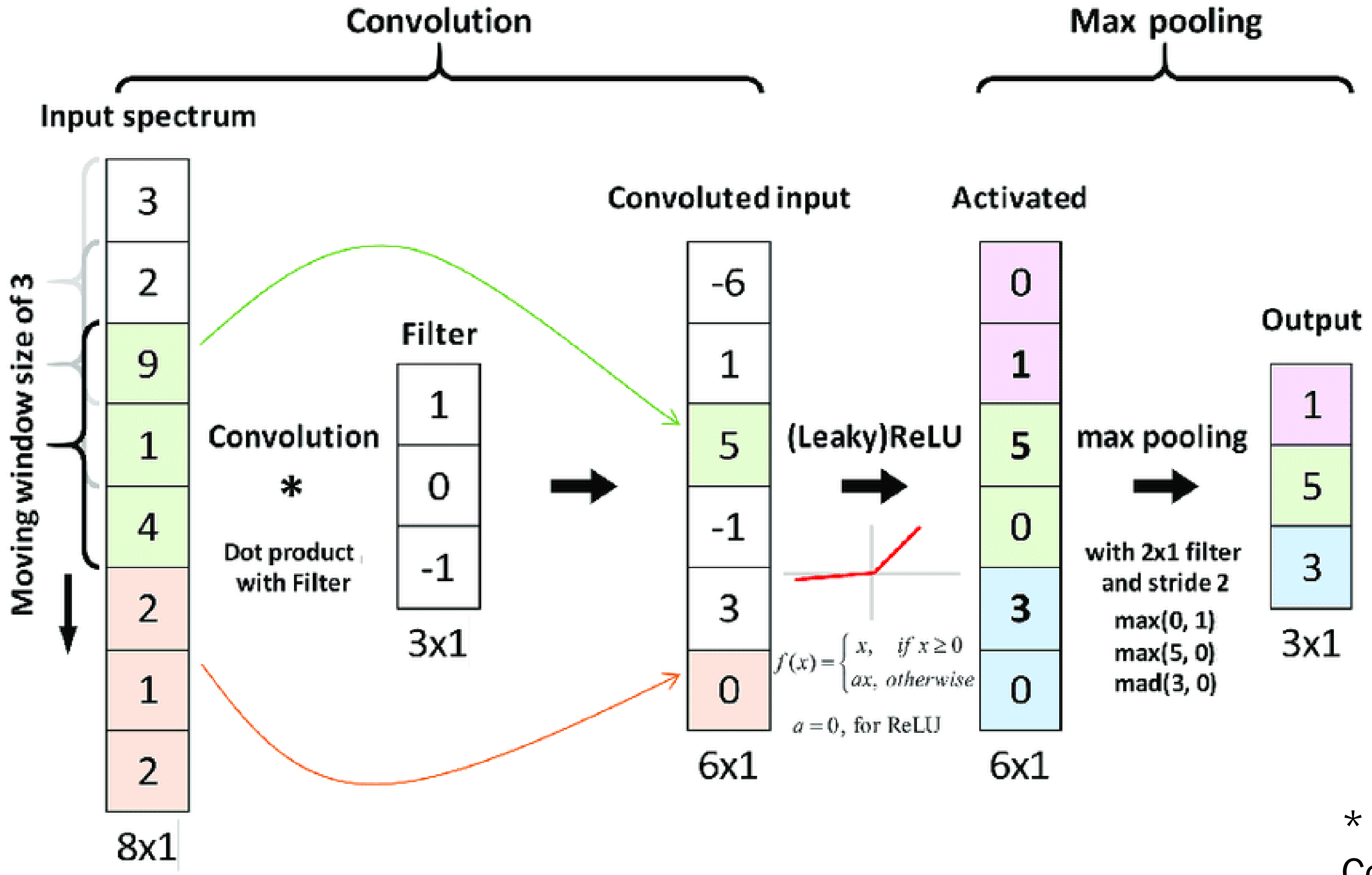
https://docs.ecognition.com/v9.5.0/eCognition_documentation/Reference%20Book/23%20Convolutional%20Neural%20Network%20Algorithms/Convolutional%20Neural%20Network%20Algorithms.htm

Gabor filters



https://stackoverflow.com/questions/41598858/how-can-filters-later-in-a-cnn-recognize-color

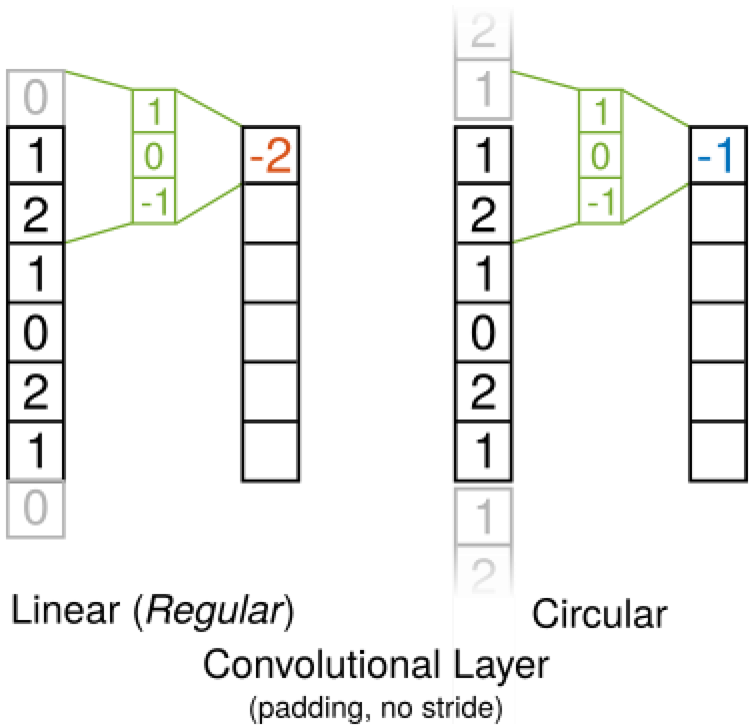
Convolution 1D*



* There is also Convolution 3D

https://www.researchgate.net/figure/A-schematic-diagram-of-convolution-and-max-pooling-layer-In-the-convolution-layer-a_fig2_337106643

Padding



https://www.tu-chemnitz.de/etit/proaut/en/research/ccnn.html

CNN - code

- https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/first_edition/5.1-introduction-to-convnets.ipynb

Data augmentation

- https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/first_edition/5.2-using-convnets-with-small-datasets.ipynb (data augmentation)

A Simple Framework for Contrastive Learning of Visual Representations

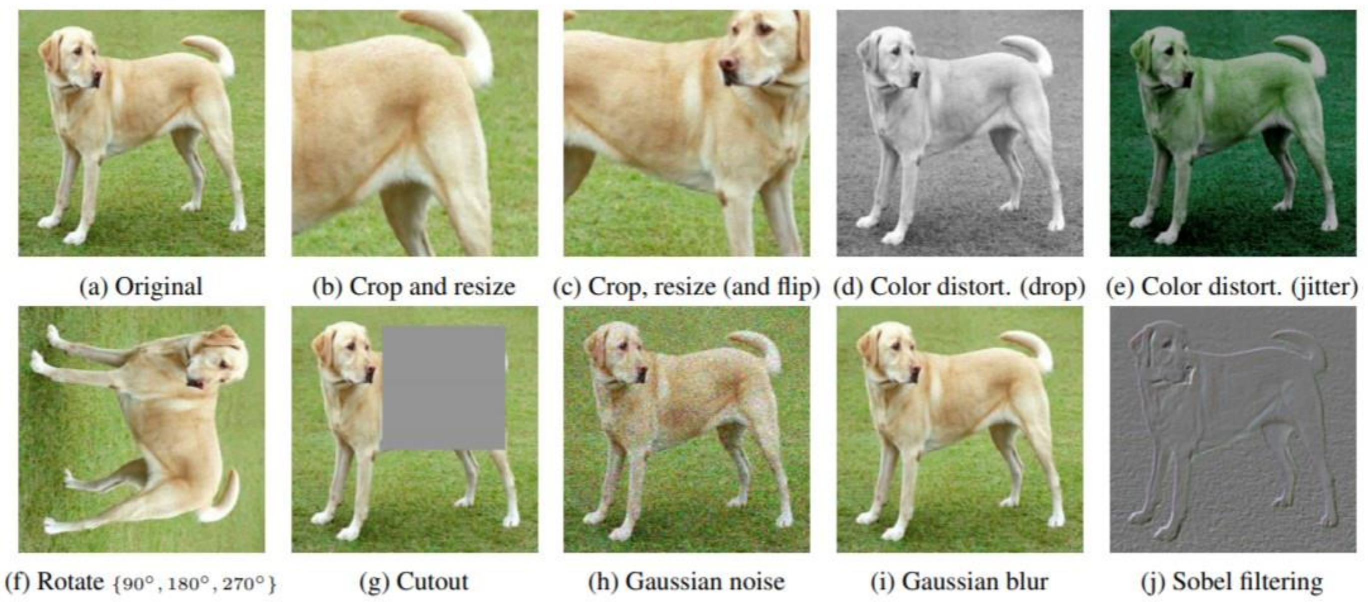
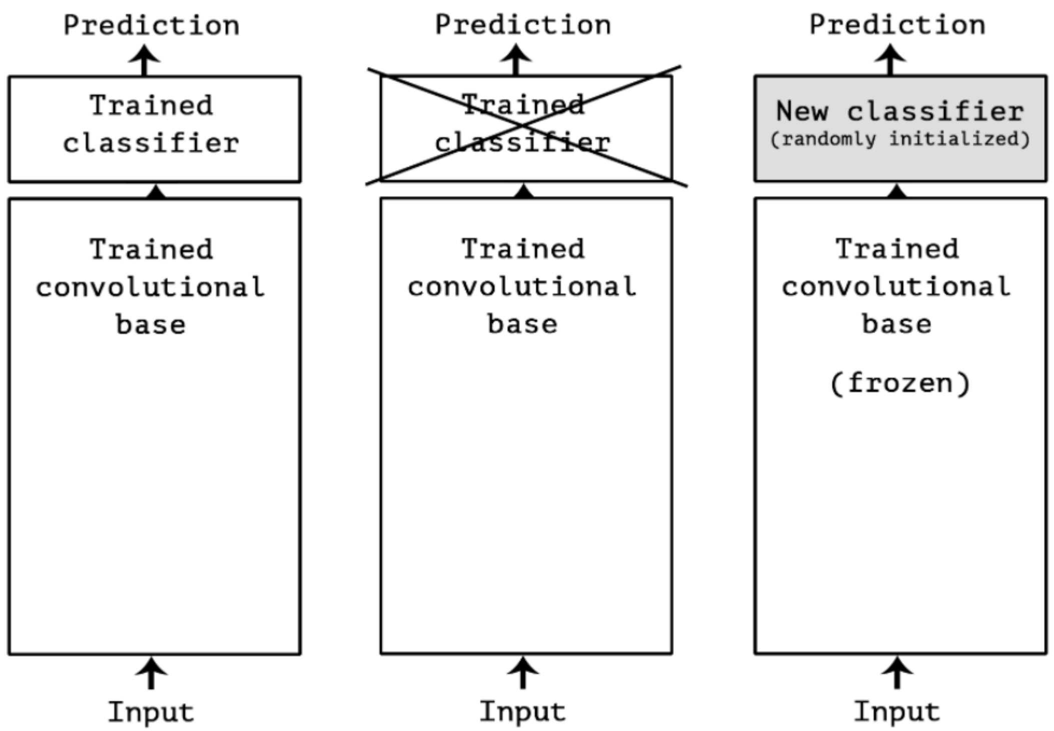


Figure 4. Illustrations of the studied data augmentation operators. Each augmentation can transform data stochastically with some internal parameters (e.g. rotation degree, noise level). Note that we *only* test these operators in ablation, the *augmentation policy used to train our models* only includes *random crop (with flip and resize)*, *color distortion*, and *Gaussian blur*. (Original image cc-by: Von.grzanka)

https://datacentricai.org/data-augmentation/

Transfer learning - code

- https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/first_edition/5.3-using-a-pretrained-convnet.ipynb



Common pretrained networks

- ~~AlexNet~~
- VGG
- ResNet
- DenseNet

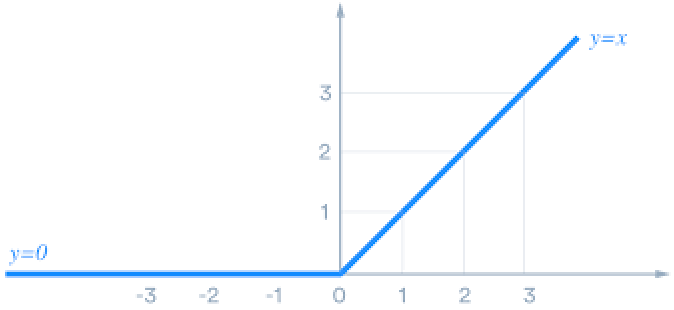
- Pretrained on ImageNet dataset (~14mln images, ~1000 classes, input size 224x224)

VGG - Very Deep Convolutional Network

- VGG-*X*: *X* refers to the number of layers (common 16, 19)
- e.g. VGG-16 = 13 convolutional layers + 3 fully connected (~138 million parameters)
- Architecture
  - *convolutional layers:*
    - kernel 3x3
    - stride 1 - to keep the spatial resolution preserved after convolution

$$\mathrm{size}_{\mathrm{out}} = \frac{size_{in} - kernel_{size} + 2 * padding}{stride} + 1$$

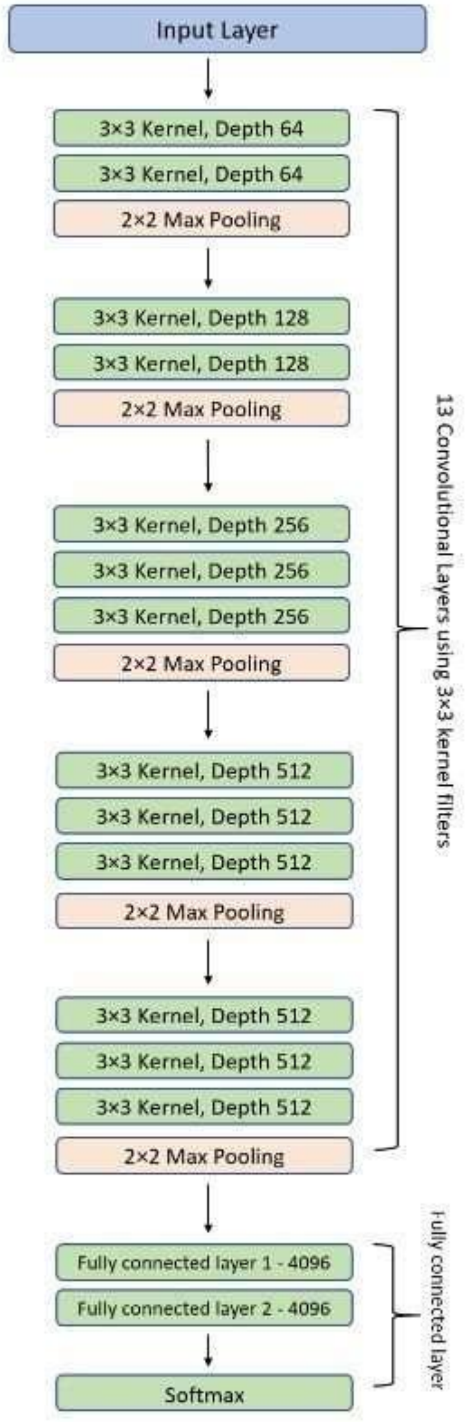
  - *ReLU activation functions (novelty comparing to AlexNet, that reduces training time)*



https://www.quora.com/What-is-the-difference-between-the-activation-functions-sigmoid-tanh-ReLU-and-ELU-in-a-neural-network

VGG-16

- Straightforward architecture
- Nowadays rather pretrained ResNet is used in practice



https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/

ResNet

- number of layers increases in CNN → the ability of the model to fit more complex functions also increases

BUT…

- vanishing gradient problem occurs (model deteriorates both on the training and testing data)

- Solution: residual neural networks (ResNet-34, ResNet-50, ResNet-101, ResNet-152)

https://viso.ai/deep-learning/resnet-residual-neural-network/

Architecture

- ResNets have fewer filters and lower complexity than VGG
- rules:
  - *the layers have the same number of filters for the same output feature map size*
  - *the number of filters doubled in case the feature map size was halved in order to preserve the time complexity per layer*
- skip connections:
  - *add the outputs from previous layers to the outputs of stacked layers (when input and output sizes are the same)*
  - *alleviate the issue of vanishing gradient by setting up an alternate shortcut for the gradient to pass through.*

- In ResNet-50, 3-layer blocks and in ResNet-34: 2-layer blocks

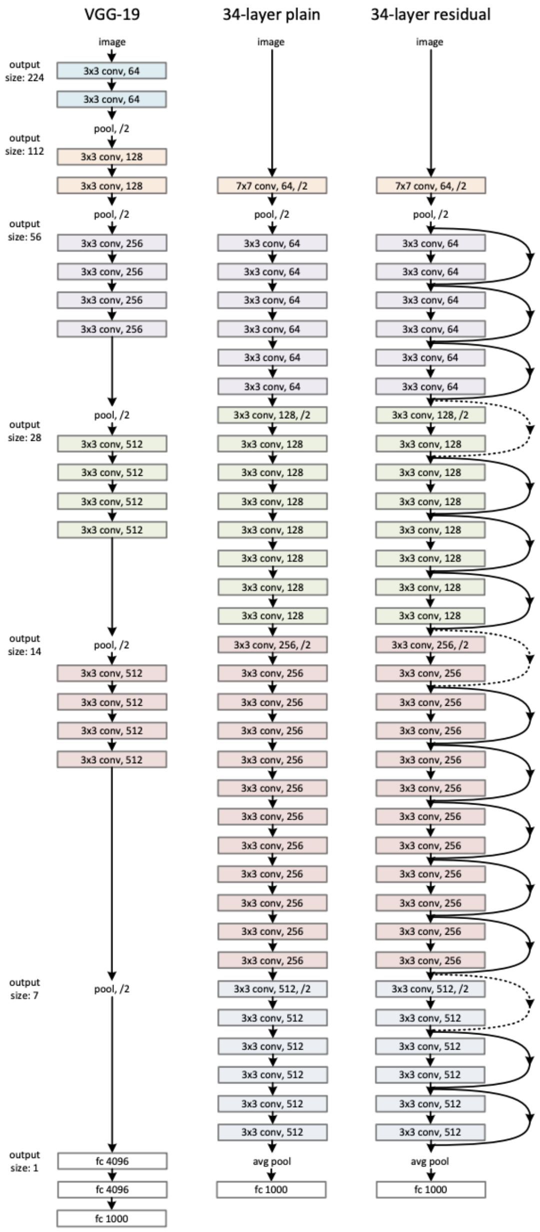
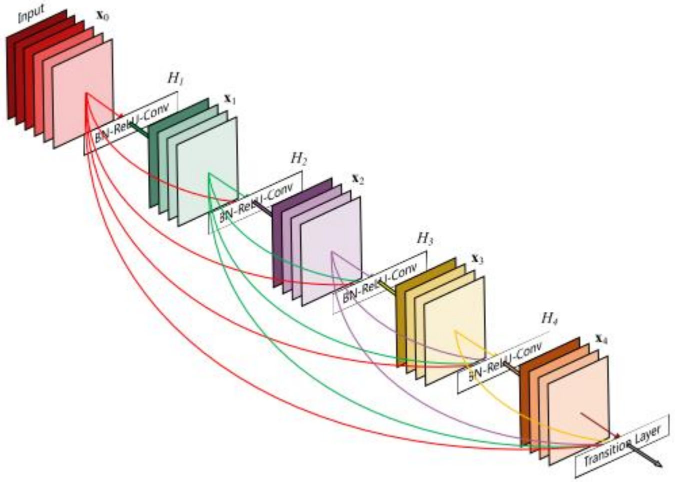


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

https://viso.ai/deep-learning/resnet-residual-neural-network/

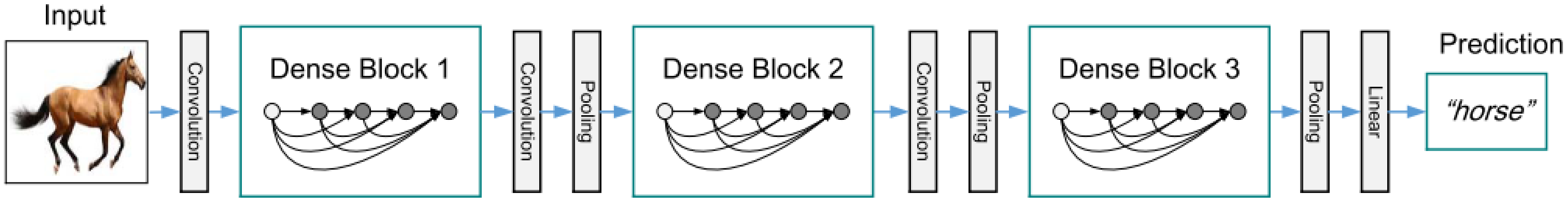
DenseNet



Can we concatenate feature maps?
- yes, if their shape is the same

BUT

an essential part of convolutional networks is down-sampling layers



From the paper:

> *DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.*

https://amaarora.github.io/2020/08/02/densenets.html

DenseNet vs. ResNet

$$\mathbf{x}_\ell = H_\ell(\mathbf{x}_{\ell-1}) + \mathbf{x}_{\ell-1}.$$

eq-1 ResNet architecture output

$$\mathbf{x}_\ell = H_\ell([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{\ell-1}]),$$

eq-2 DenseNet architecture output

In Dense block, each layer adds *K* features on top to the global state (*K* - growth rate of the network)