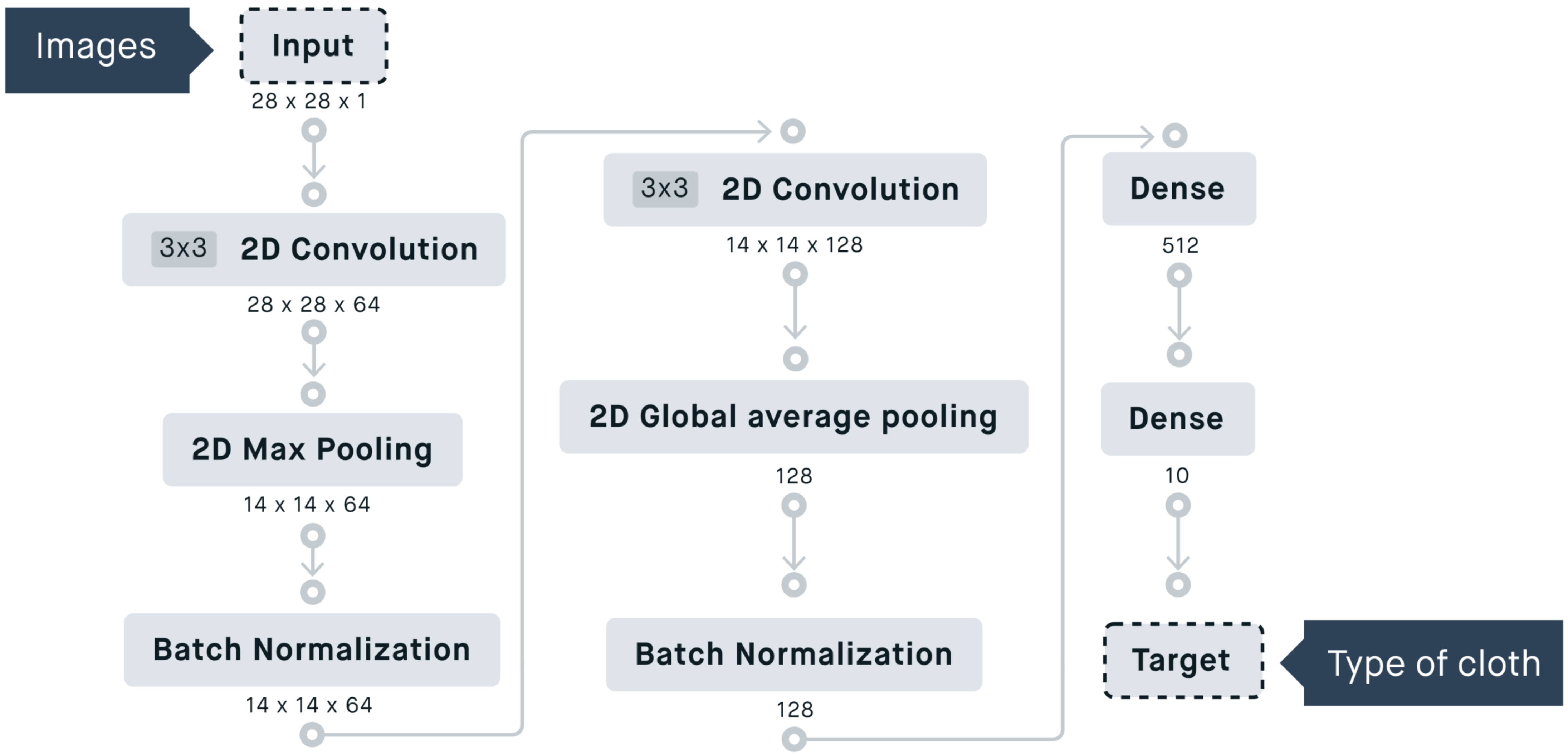
Architektura
głębokich sieci neuronowych

Weronika Hryniewska

Tensorflow playground

https://playground.tensorflow.org

- Jak wielkość sieci wpływa na szybkość trenowania?
- Czy można różnymi cechami rozwiązać to samo zadanie?
- Co robi learning rate, albo funkcja aktywacji?
- Co się stanie, gdy: zmienimy batch size, podział danych lub dodamy szum?
- Co jest bardziej korzystne – zwiększenie neuronów w warstwach czy liczby warstw?
- Czy lepiej wziąć za dużą sieć, czy za małą?



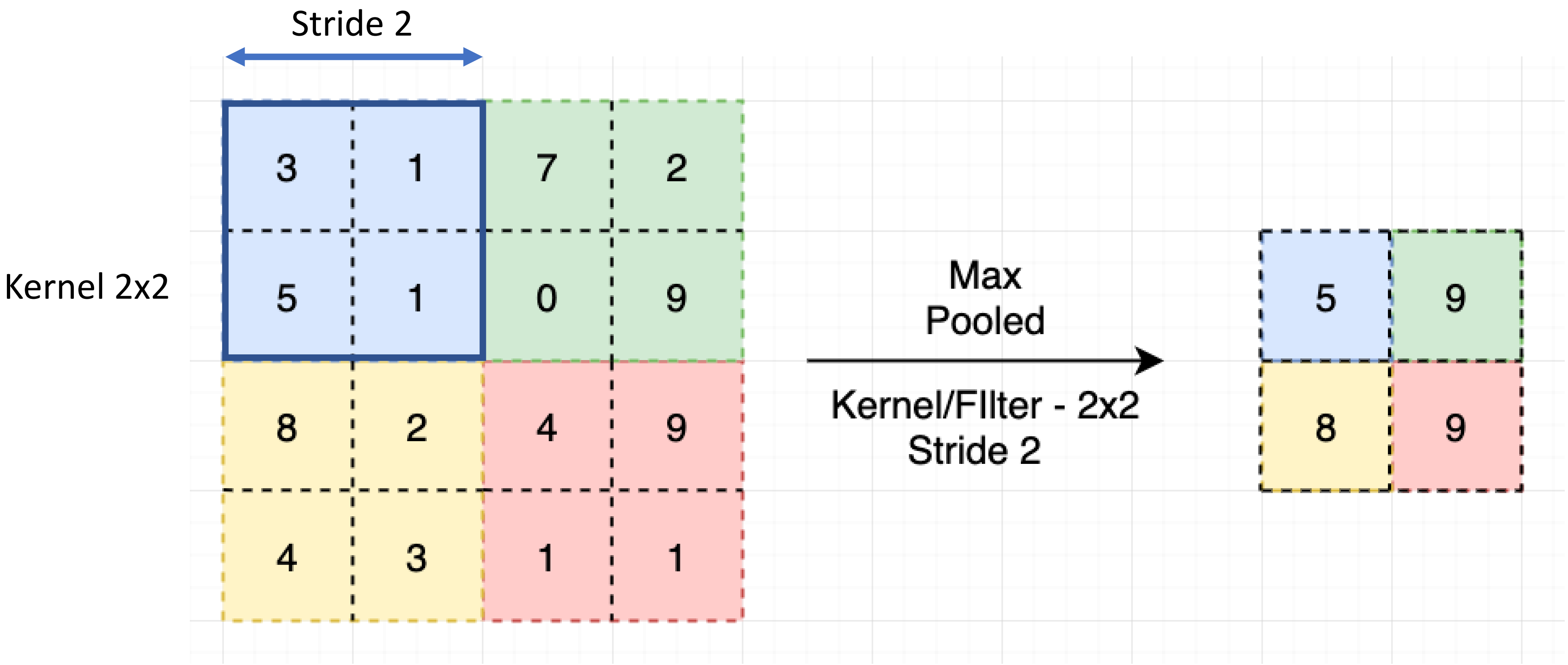
https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/blocks/

Max pooling

$$\text{Stride} = \frac{\text{Input Size}}{\text{Output Size}} = \frac{4}{2} = 2$$

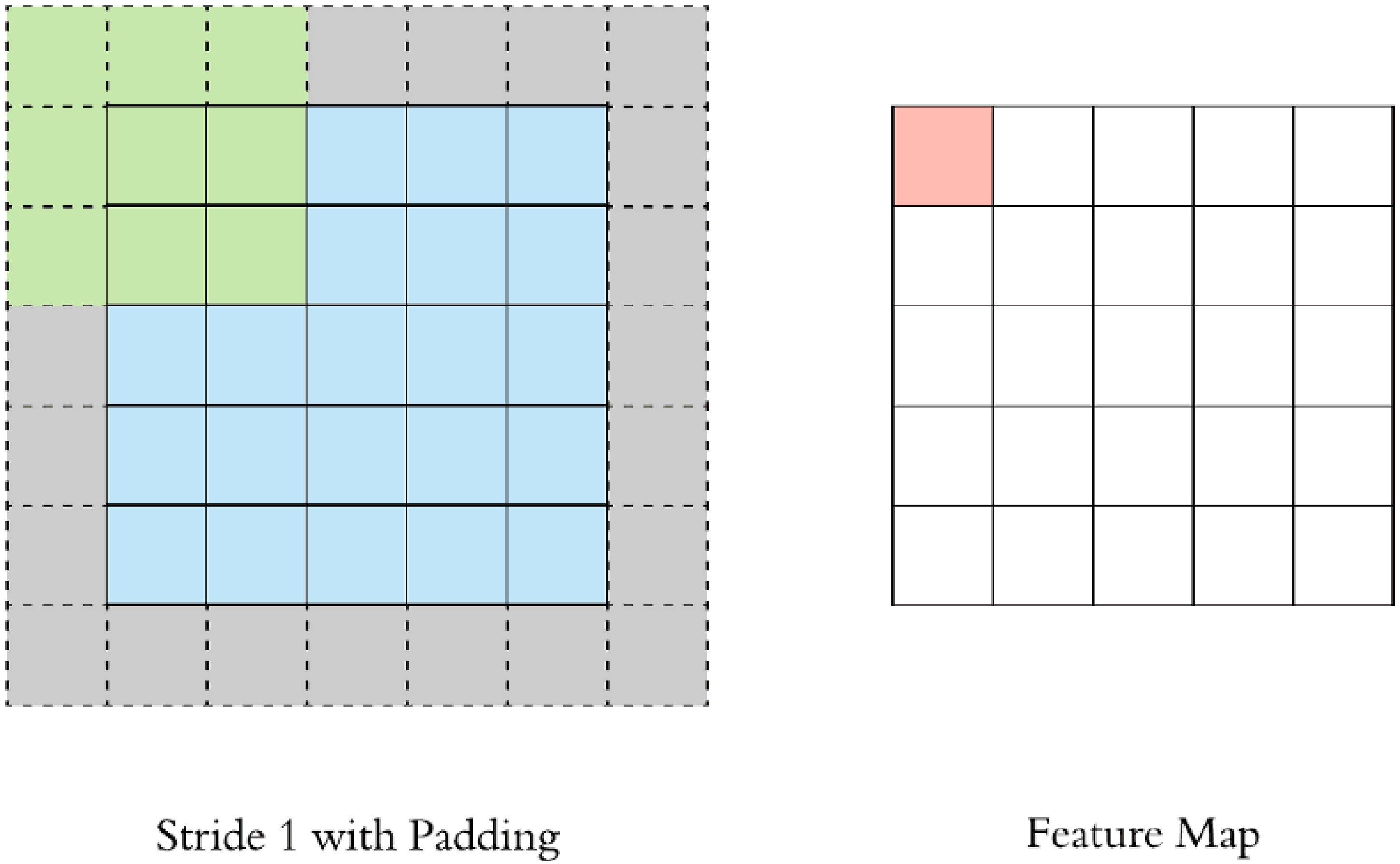
$$\text{Kernel Size} = \text{Input Size} - (\text{Output Size} - 1)*(\text{Stride}) = 4 - (2-1)*2 = 2$$

$$\text{Padding} = 0$$

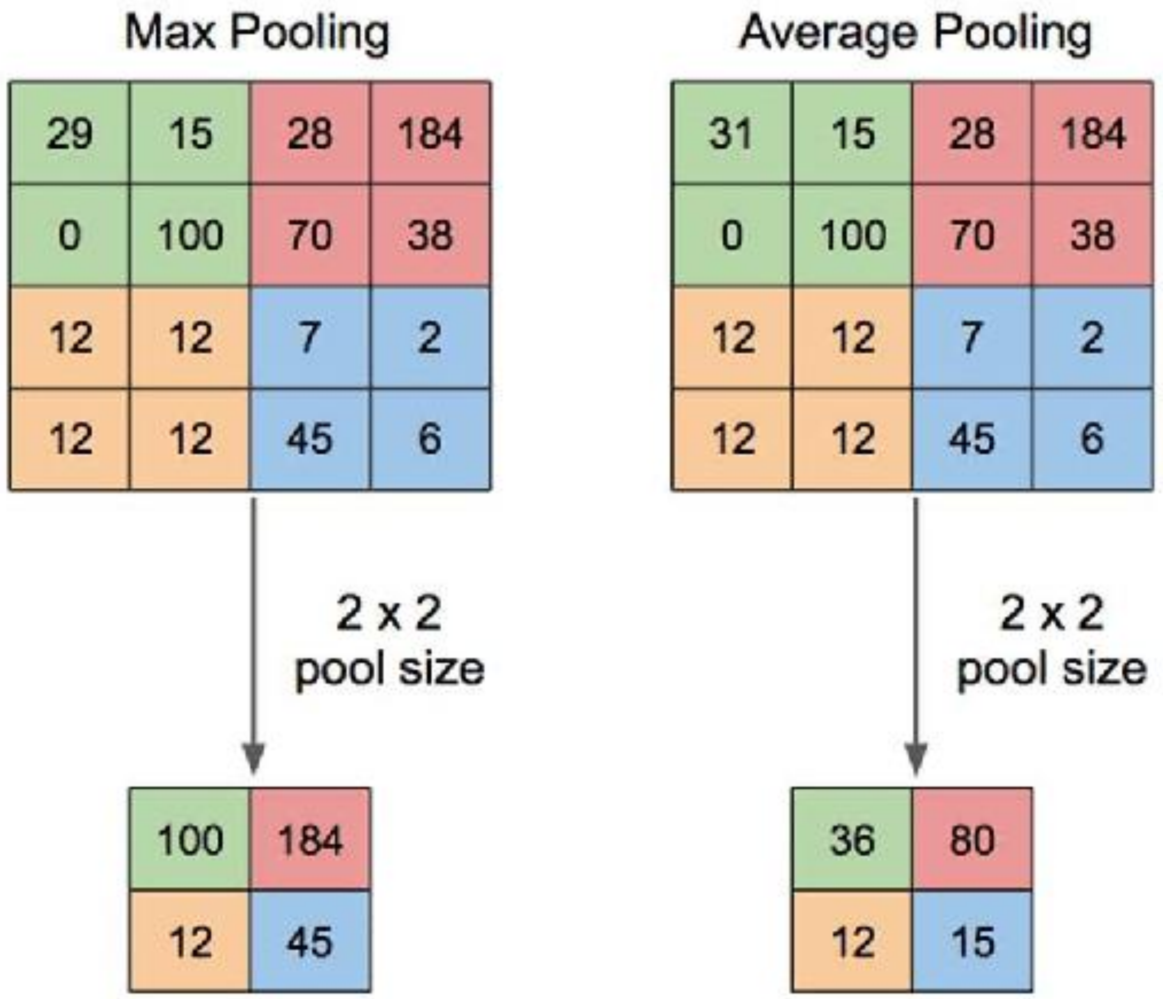


https://ai.plainenglish.io/pooling-layer-beginner-to-intermediate-fa0dbdce80eb

Padding i stride

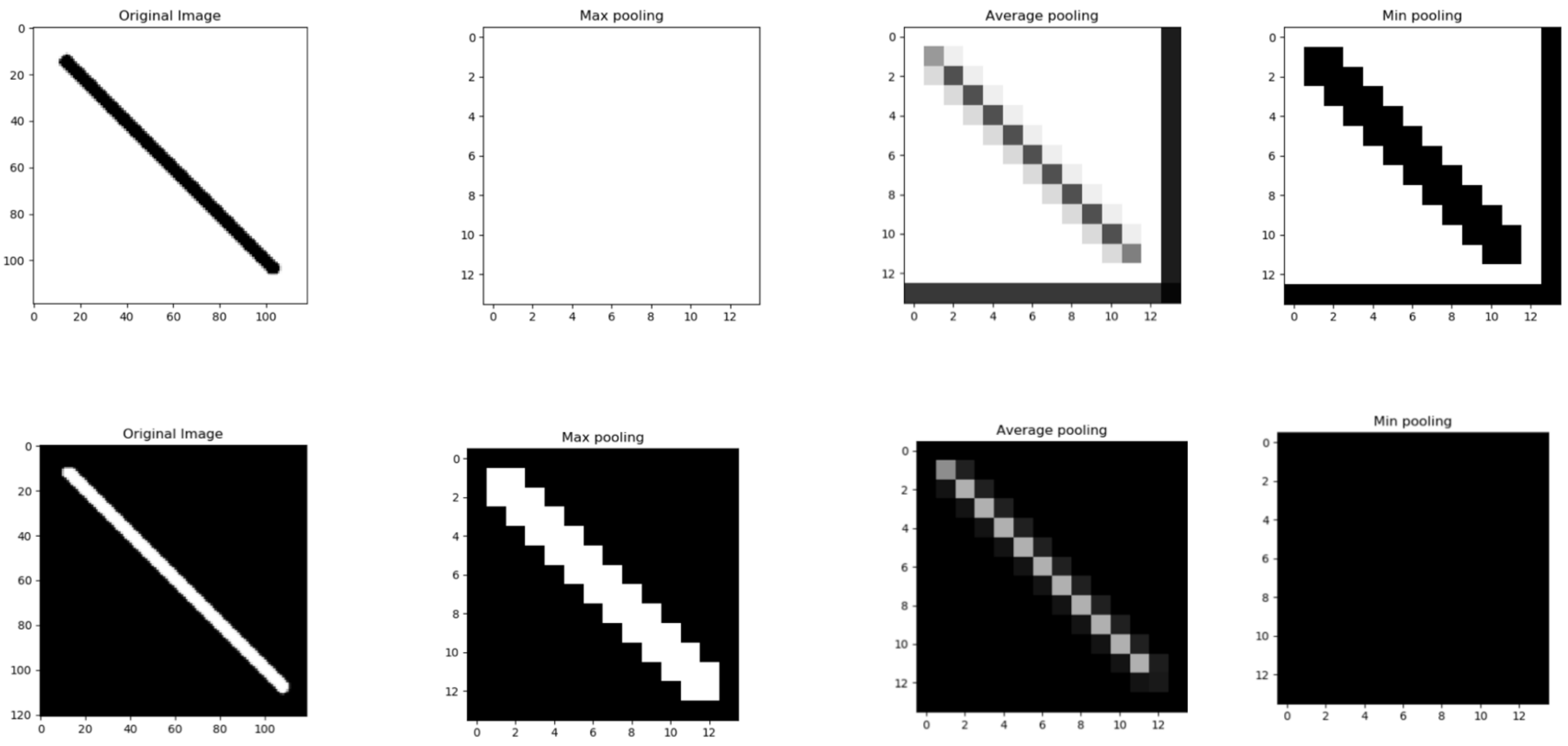


https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529



www.quora.com/What-is-max-pooling-in-convolutional-neural-networks

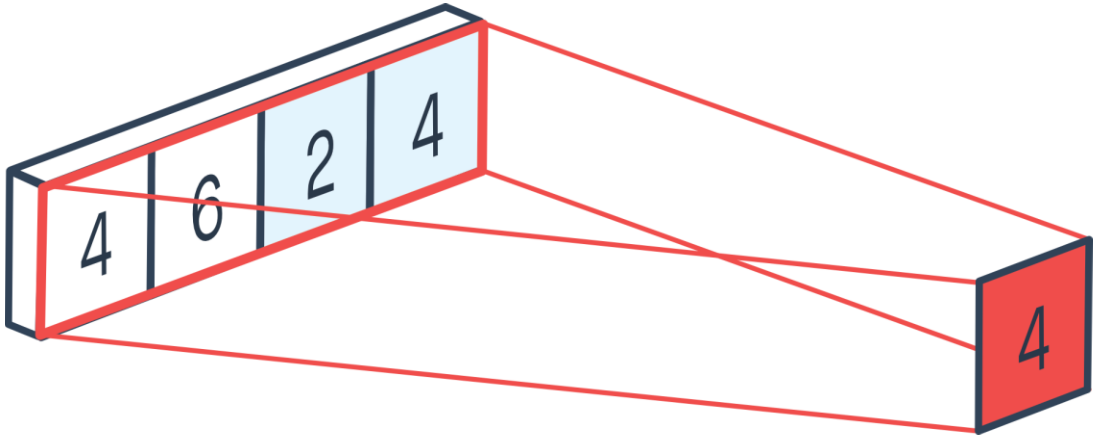
Porównanie operacji poolingu

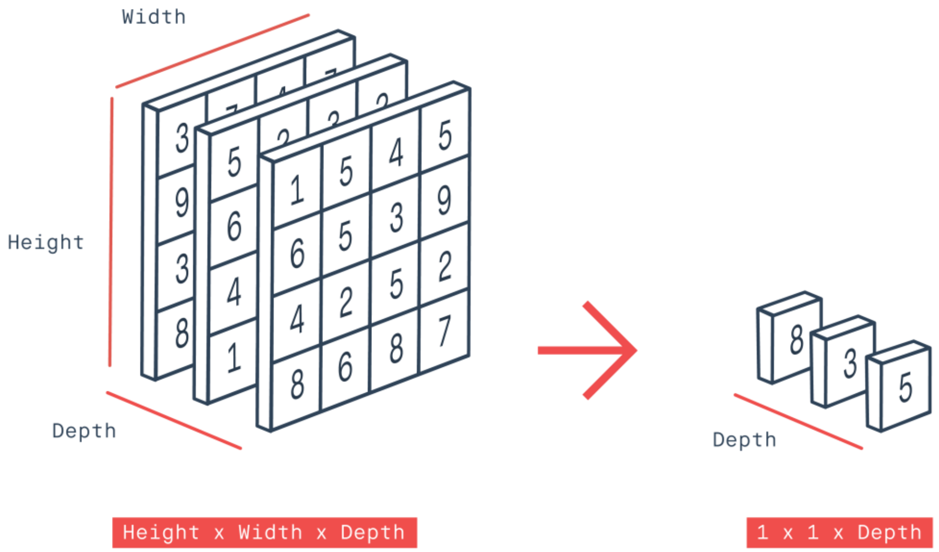


0 - czarny
255 - biały

https://iq.opengenus.org/pooling-layers

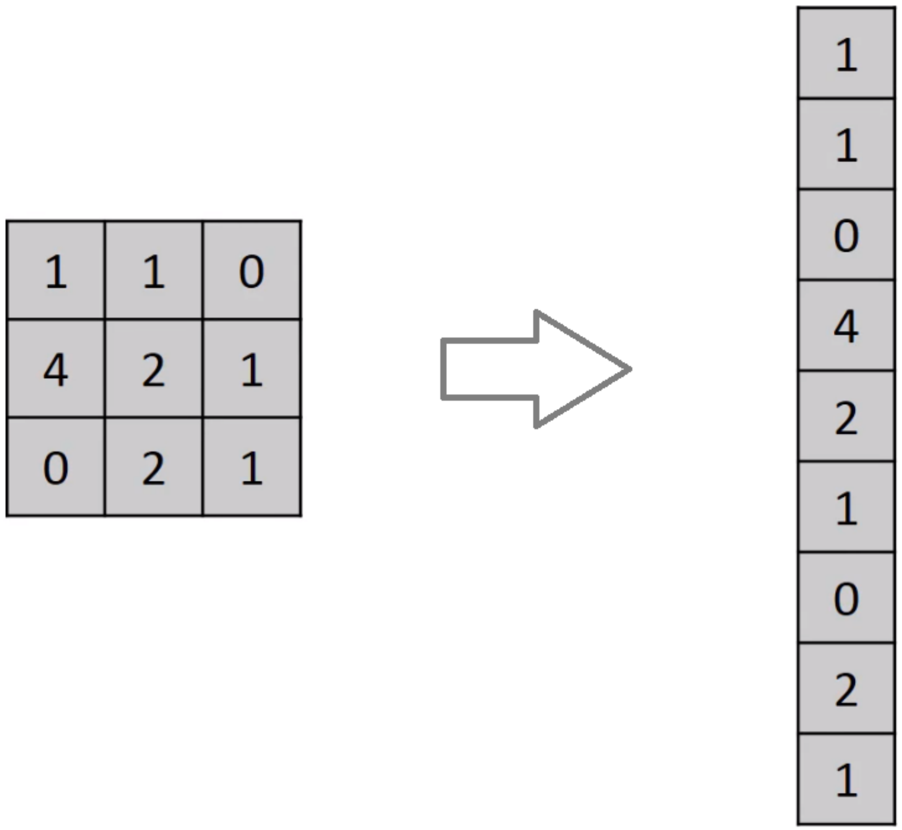
1D / 2D Global average pooling





https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/blocks/2d-global-average-pooling

Flatten



https://medium.com/@cdabakoglu/what-is-convolutional-neural-network-cnn-with-keras-cab447ad204c

Batch normalization

(batch - mean(batch)) / (var(batch) + epsilon) * gamma + beta

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

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

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

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

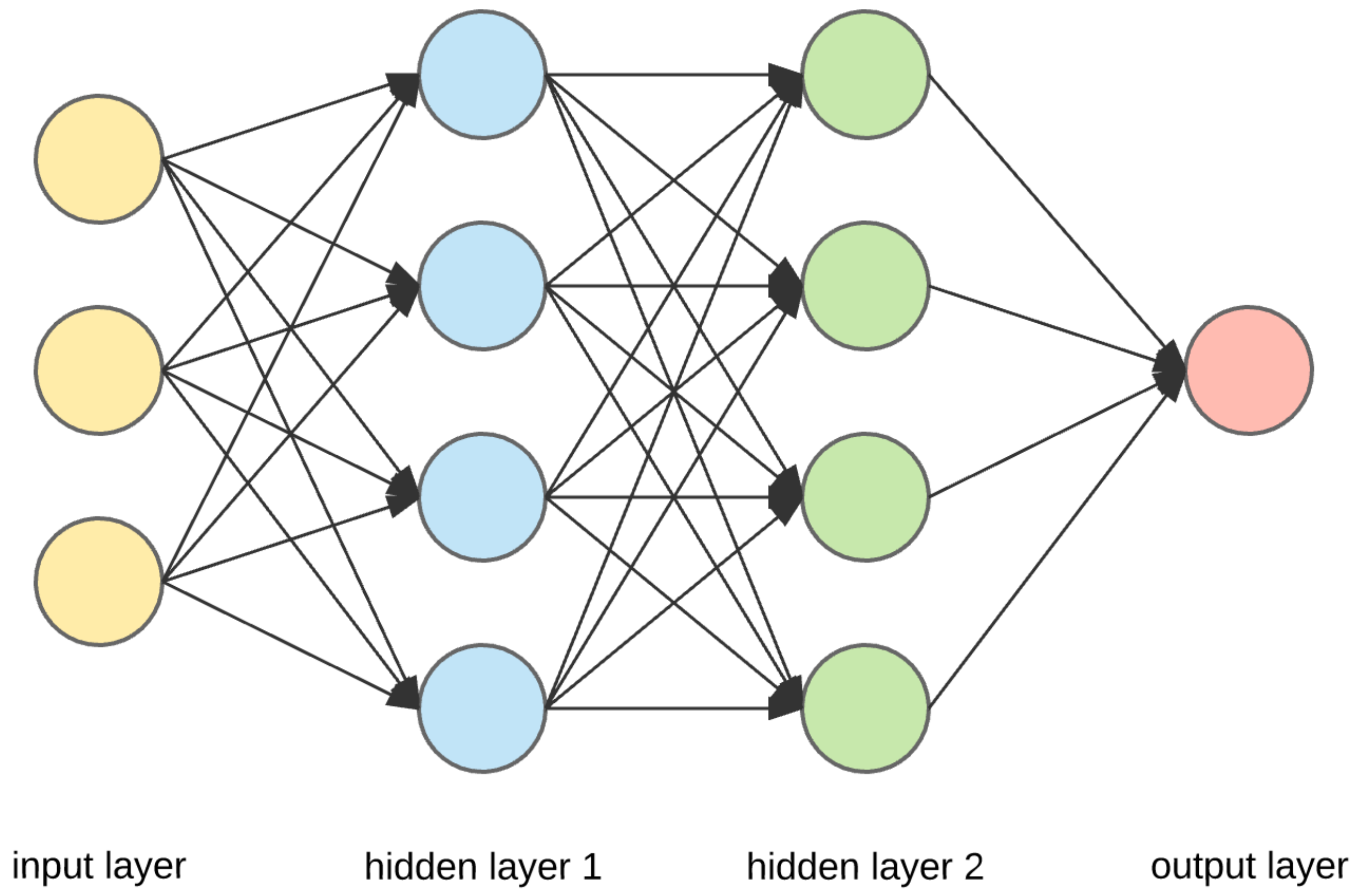
$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

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

- epsilon is small constant
- gamma is a learned scaling factor (initialized as 1)
- beta is a learned offset factor (initialized as 0)

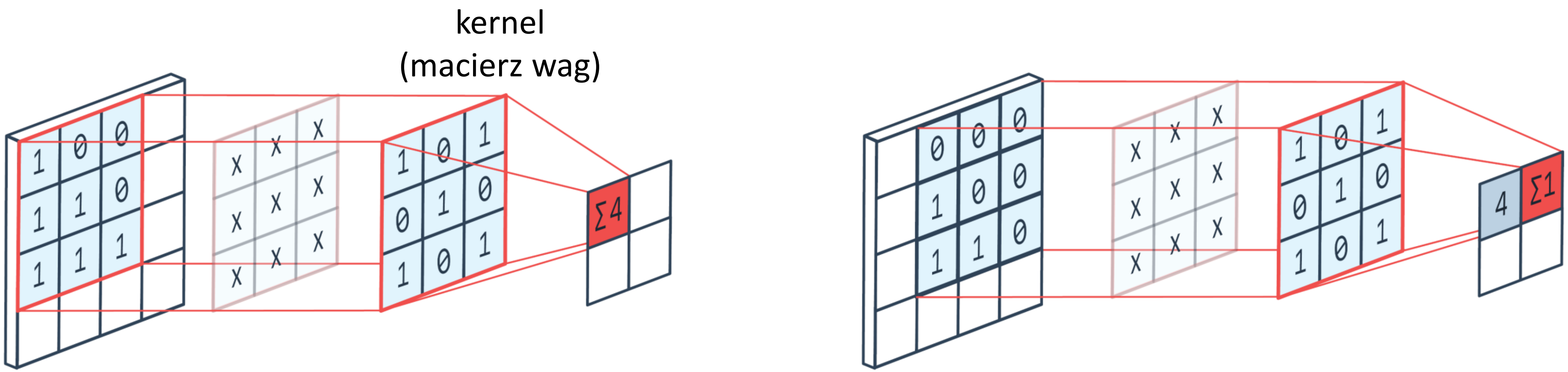
Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. PMLR. http://proceedings.mlr.press/v37/ioffe15.html

Dense (fully connected)



https://medium.com/@cdabakoglu/what-is-convolutional-neural-network-cnn-with-keras-cab447ad204c

2D Convolution block

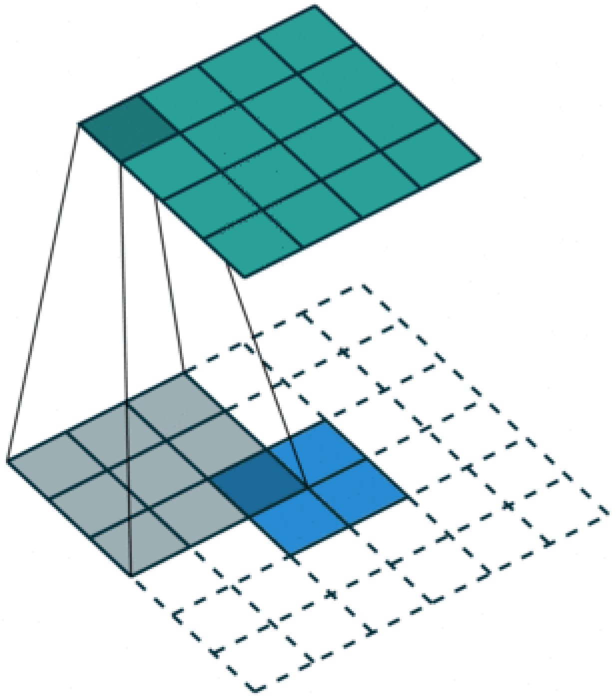


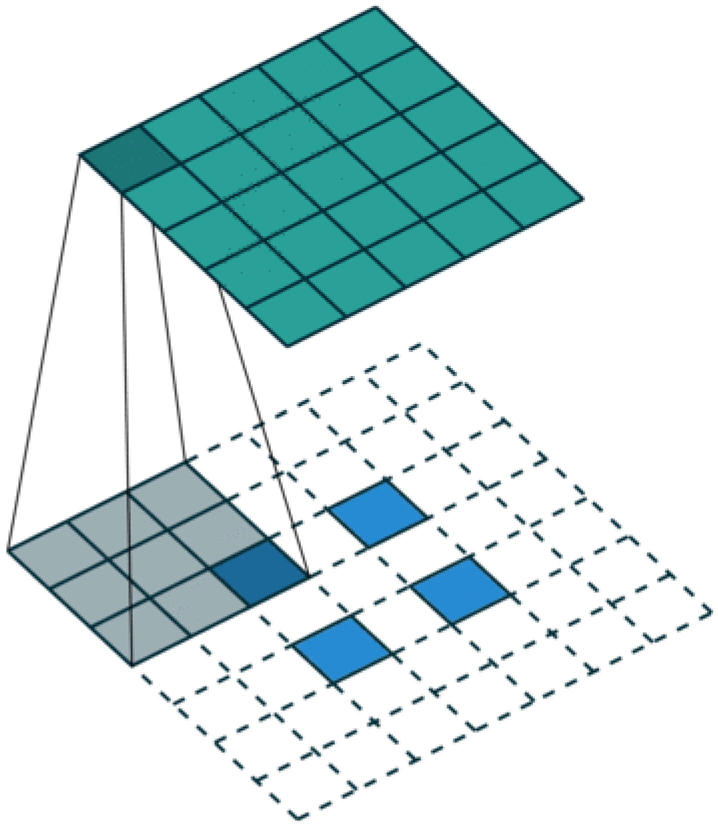
5×5=25 cech wejściowych i 3×3=9 cech wyjściowych

Gdyby to była warstwa w pełni połączona macierz wag miałaby: 25×9 = 225 parametrów

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

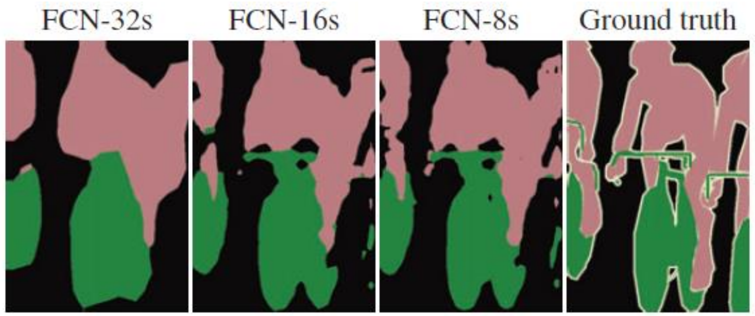
Deconvolution (transposed Convolution)

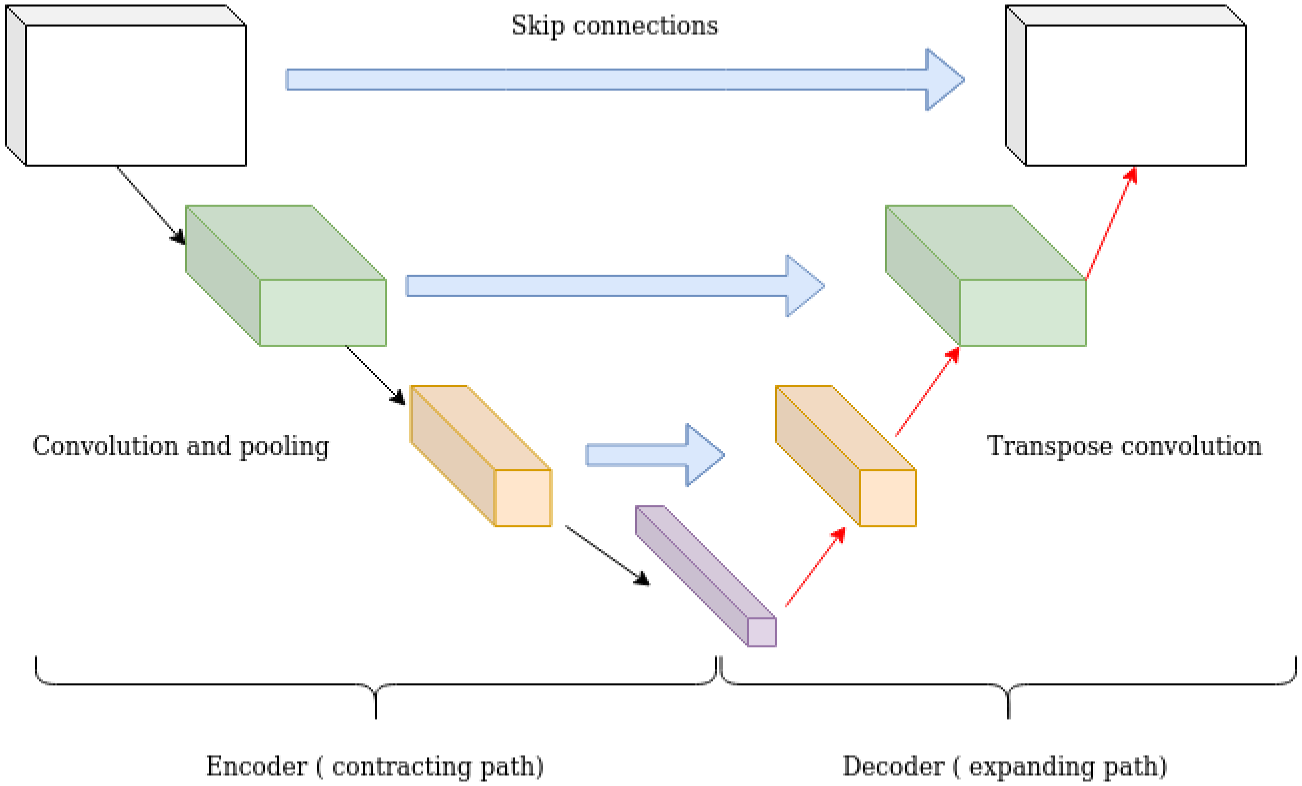




https://towardsdatascience.com/types-of-convolution-kernels-simplified-f040cb307c37

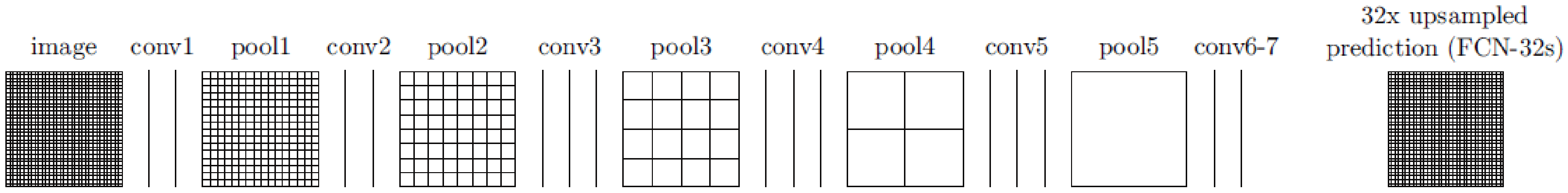
Skip connections

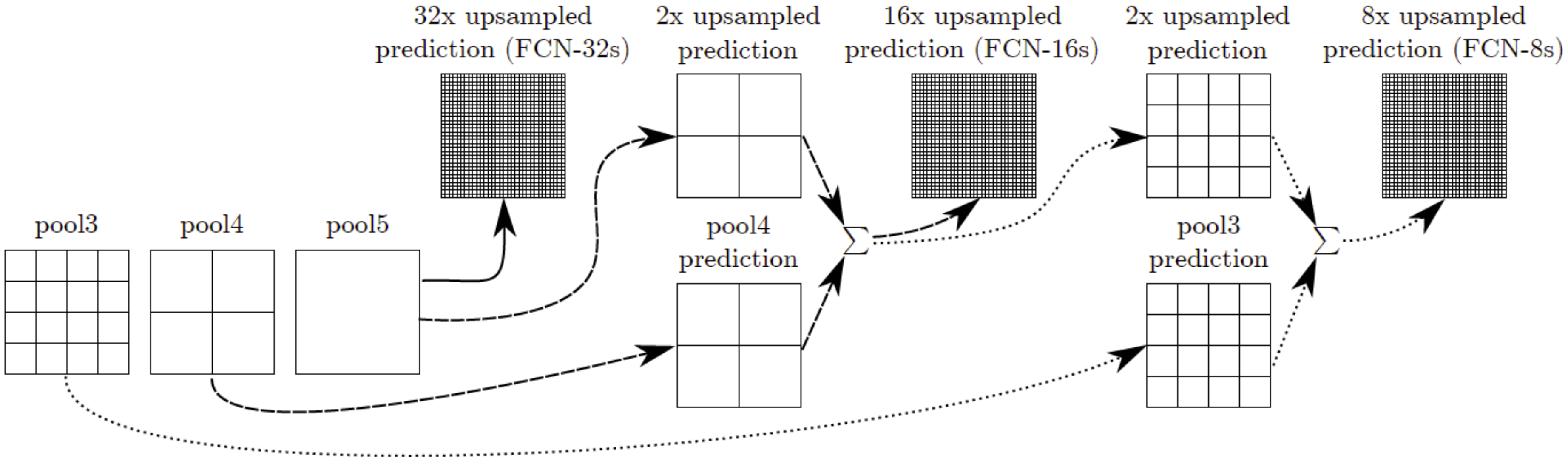




https://theaisummer.com/skip-connections/

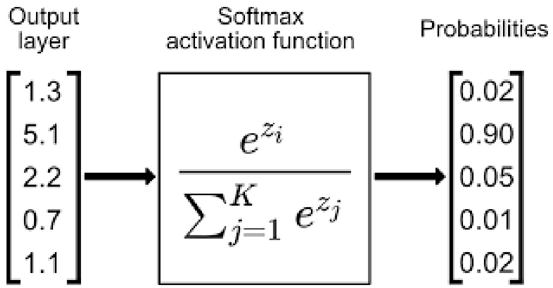
Skip connections



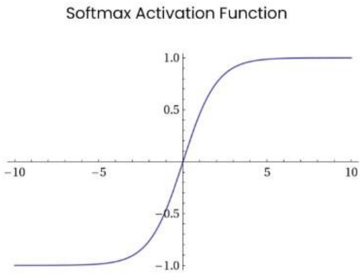


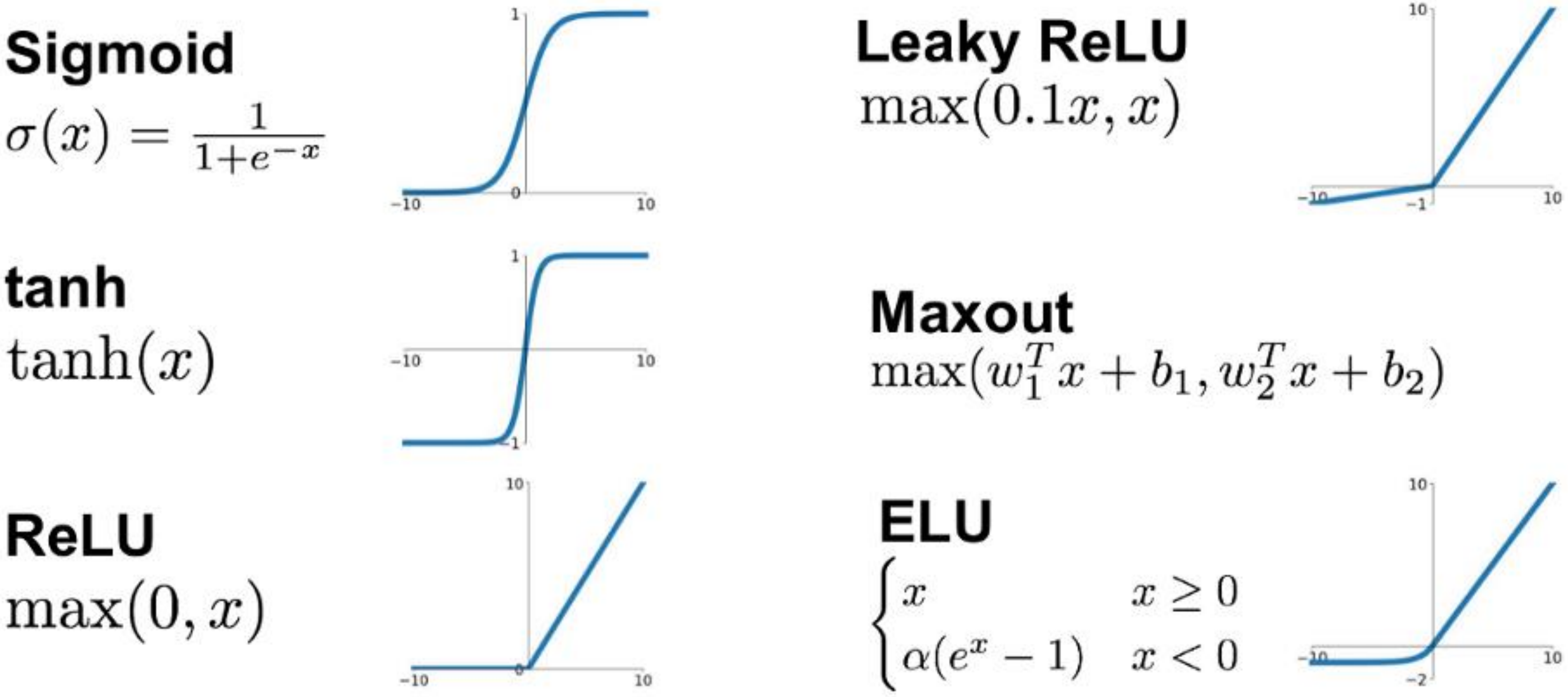
https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1

Funkcje aktywacji



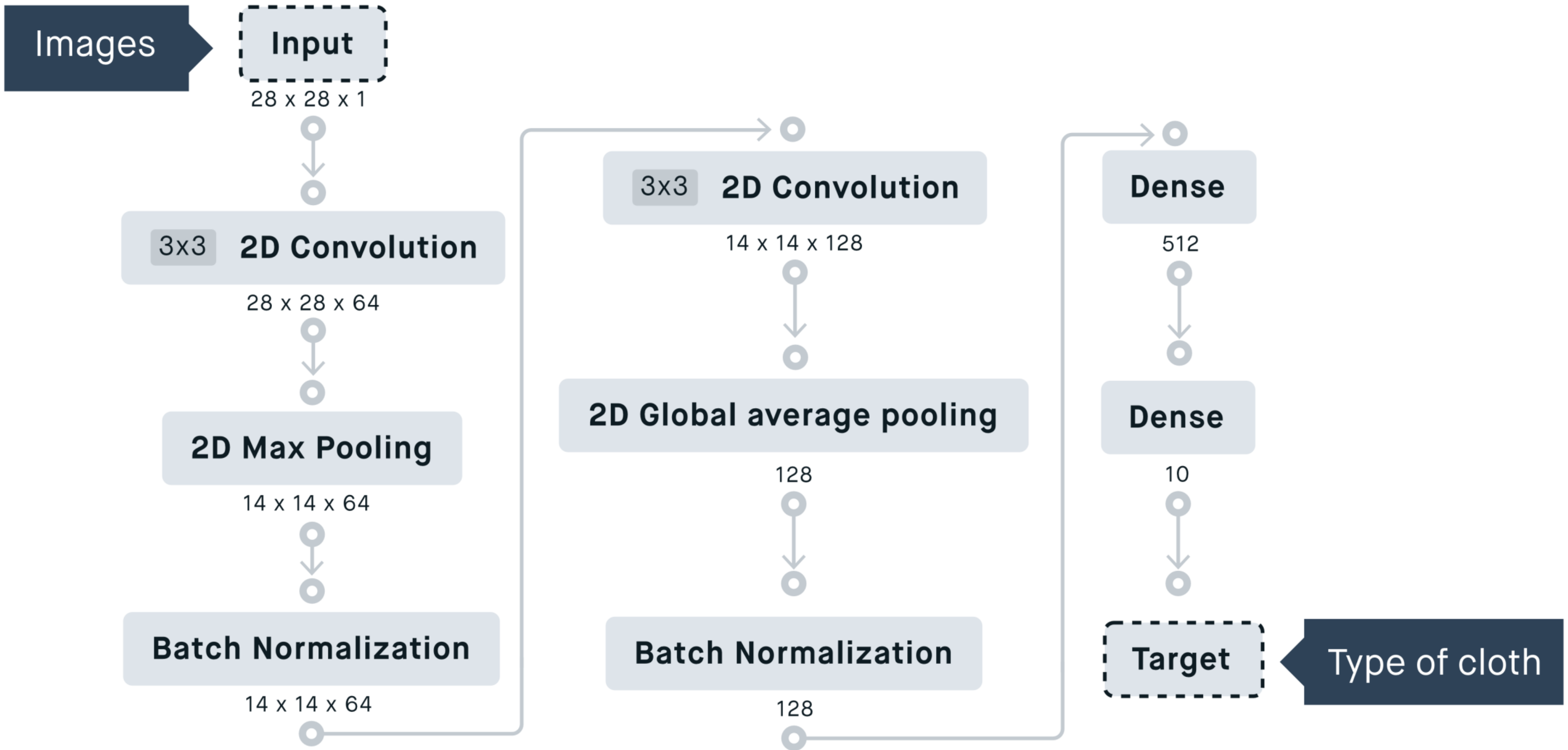
∑1.00





https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60
https://krisbolton.com/a-quick-introduction-to-artificial-neural-networks-part-2

https://laptrinhx.com/complete-guide-of-activation-functions-574622854

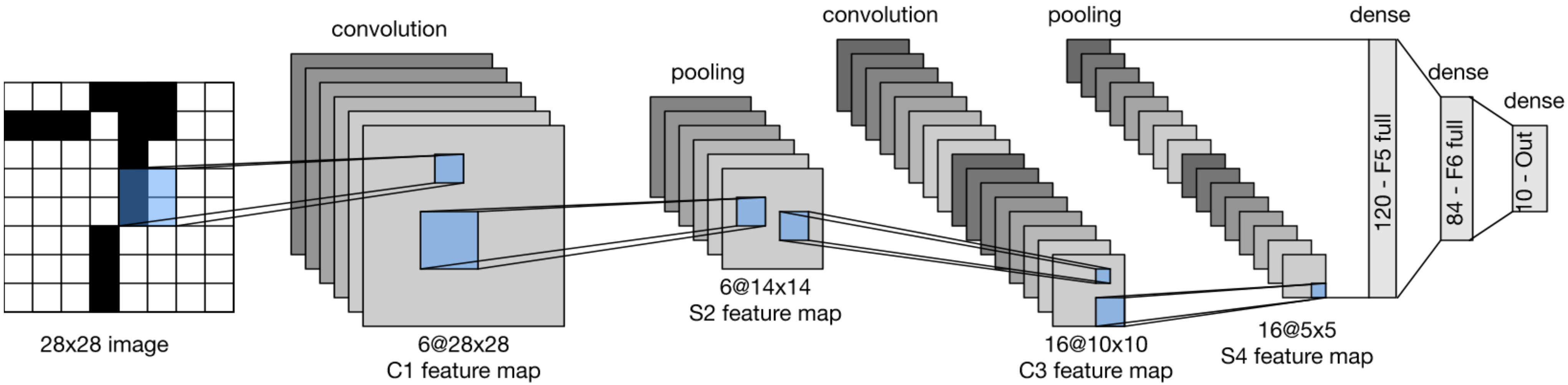



```
input_shape = (28, 28, 1)
NUM_CLASSES = 10
model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.BatchNormalization(),
        layers.Conv2D(128, kernel_size=(3, 3), activation="relu"),
        layers.GlobalAveragePooling(),
        layers.BatchNormalization(),
        layers.Dense(512, activation="relu"),
        layers.Dense(NUM_CLASSES, activation="softmax"),
    ]
)
```

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

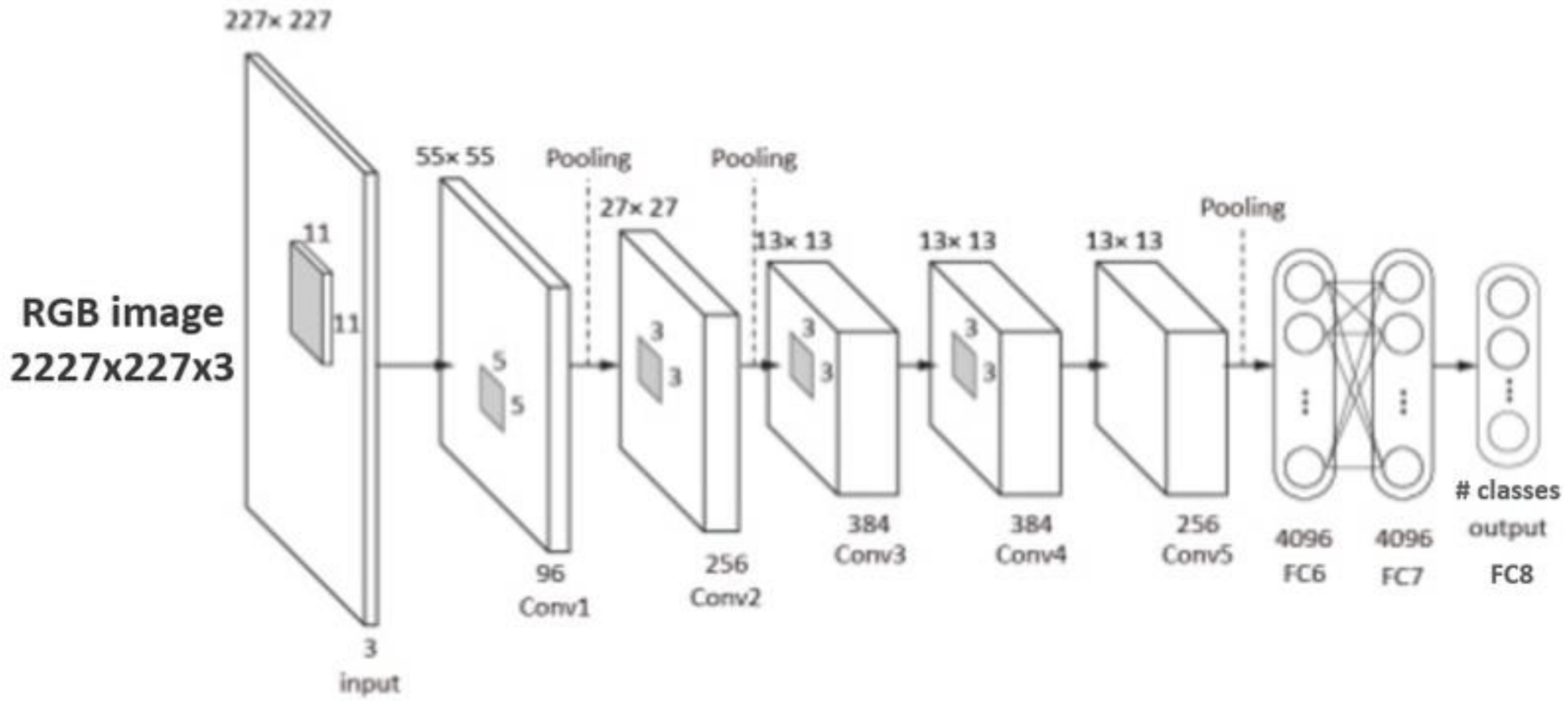
Przykłady architektur sieci neuronowych

LeNet



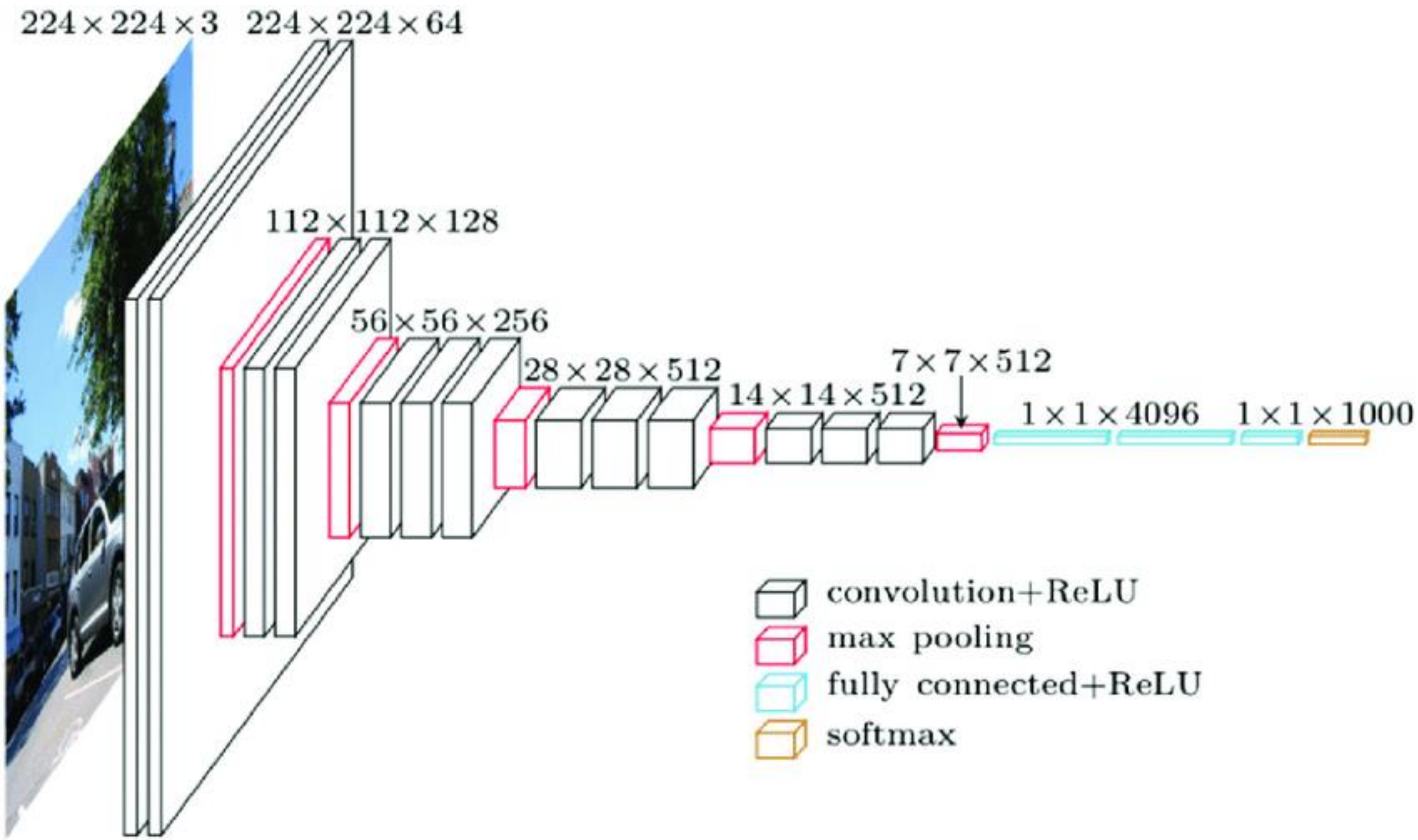
https://colab.research.google.com/github/d2l-ai/d2l-tensorflow-colab/blob/master/chapter_convolutional-neural-networks/lenet.ipynb

AlexNet



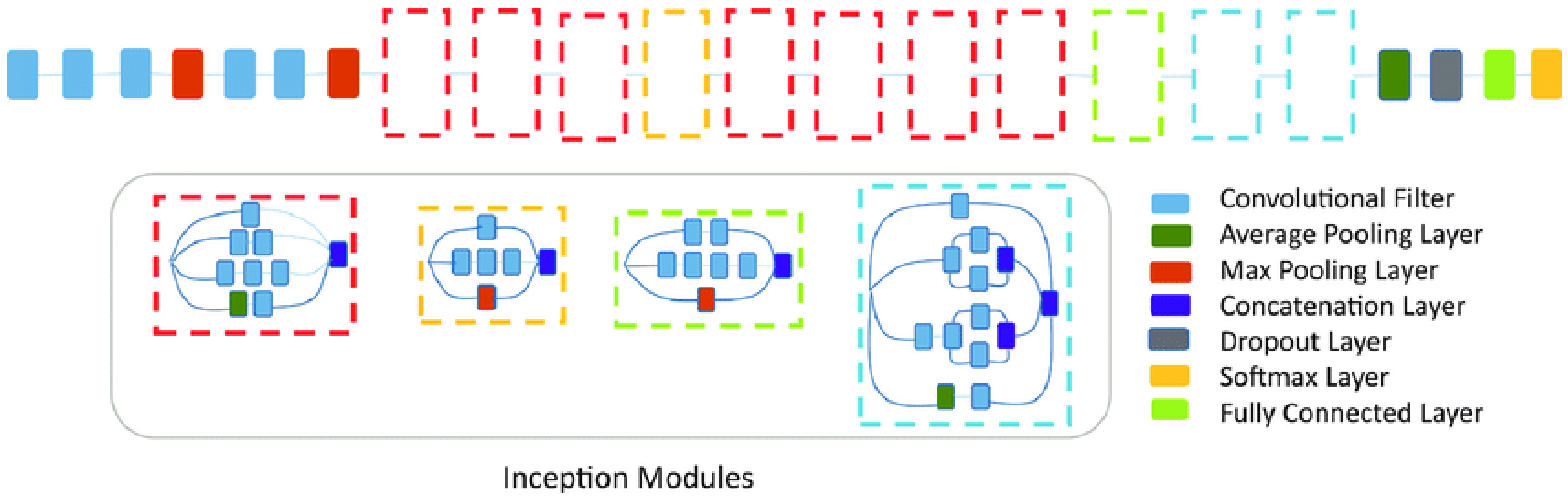
https://www.researchgate.net/figure/AlexNet-architecture-Includes-5-convolutional-layers-and-3-fullyconnected-layers_fig3_322592079

VGG



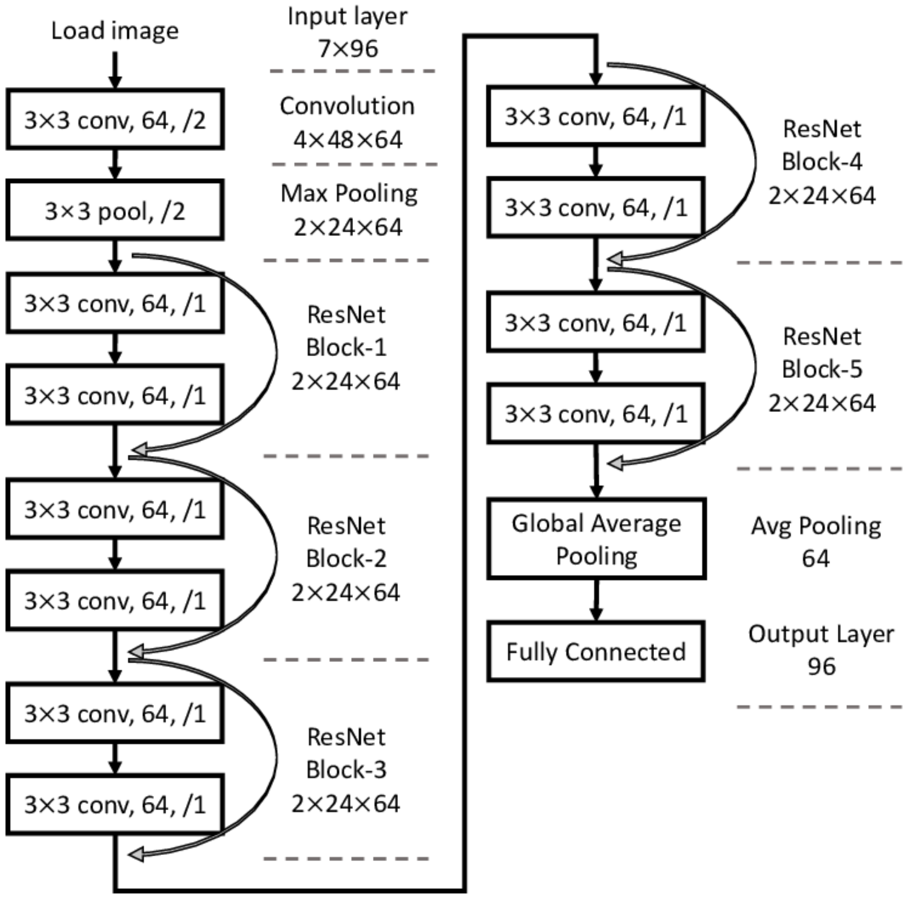
https://www.researchgate.net/figure/An-overview-of-the-VGG-16-model-architecture-this-model-uses-simple-convolutional-blocks_fig2_328966158

InceptionNet/GoogLeNet (inception modules)



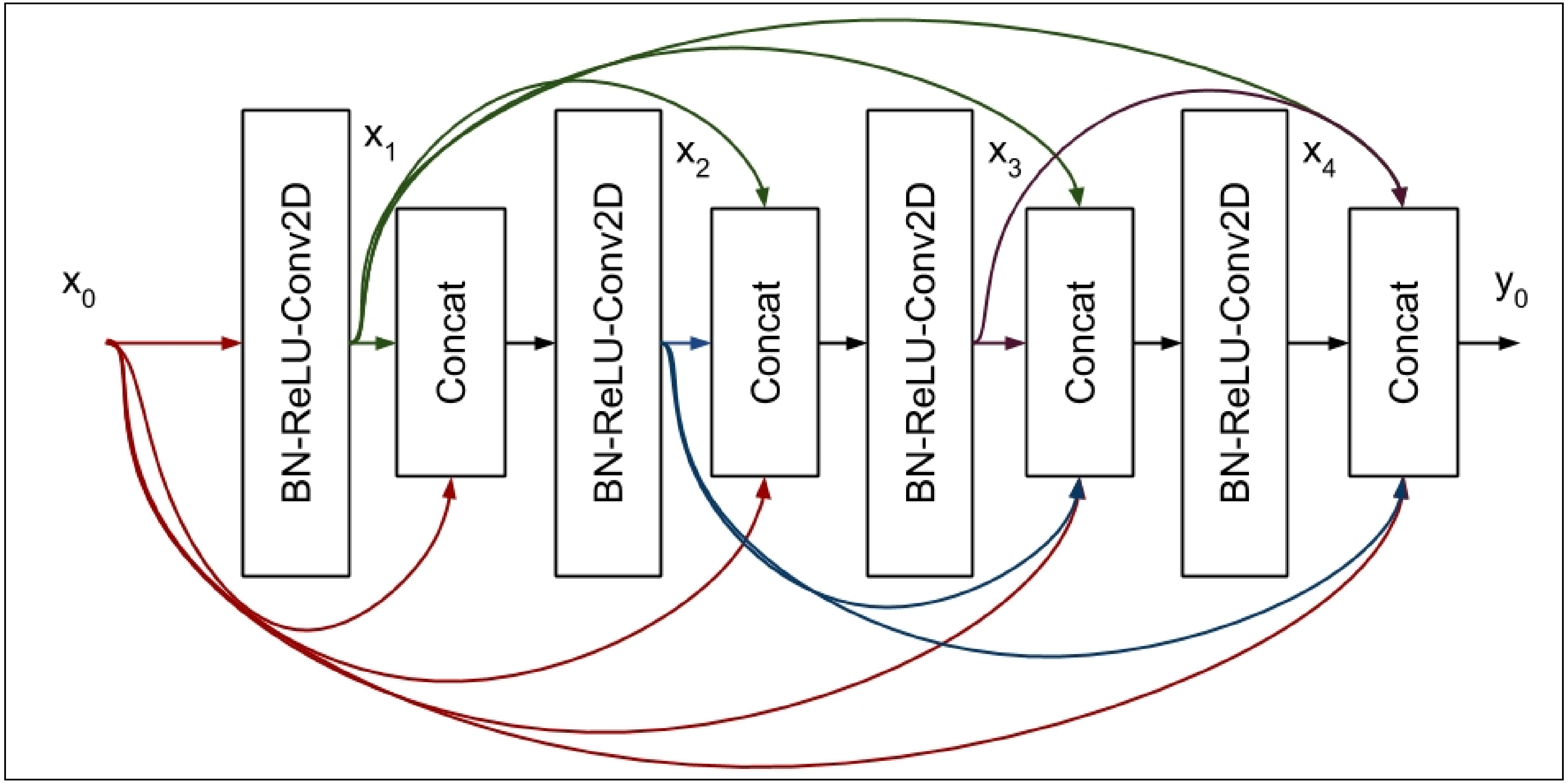
https://www.researchgate.net/figure/Convolutional-neural-network-architecture-Inception-v3-used-in-this-study-Inception-v3_fig3_328775405

ResNet



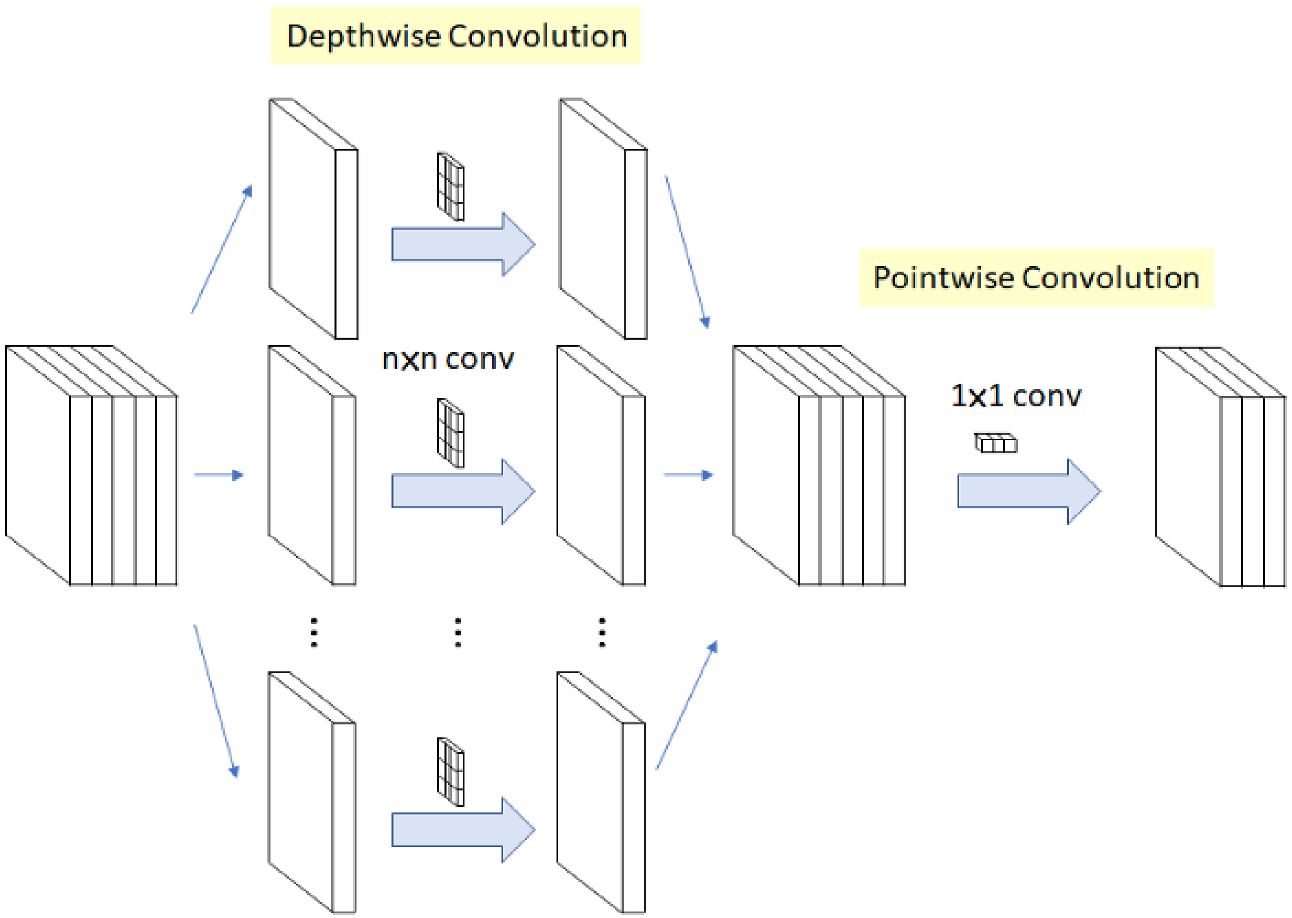
https://www.researchgate.net/figure/The-structure-of-ResNet-12_fig1_329954455

DenseNet



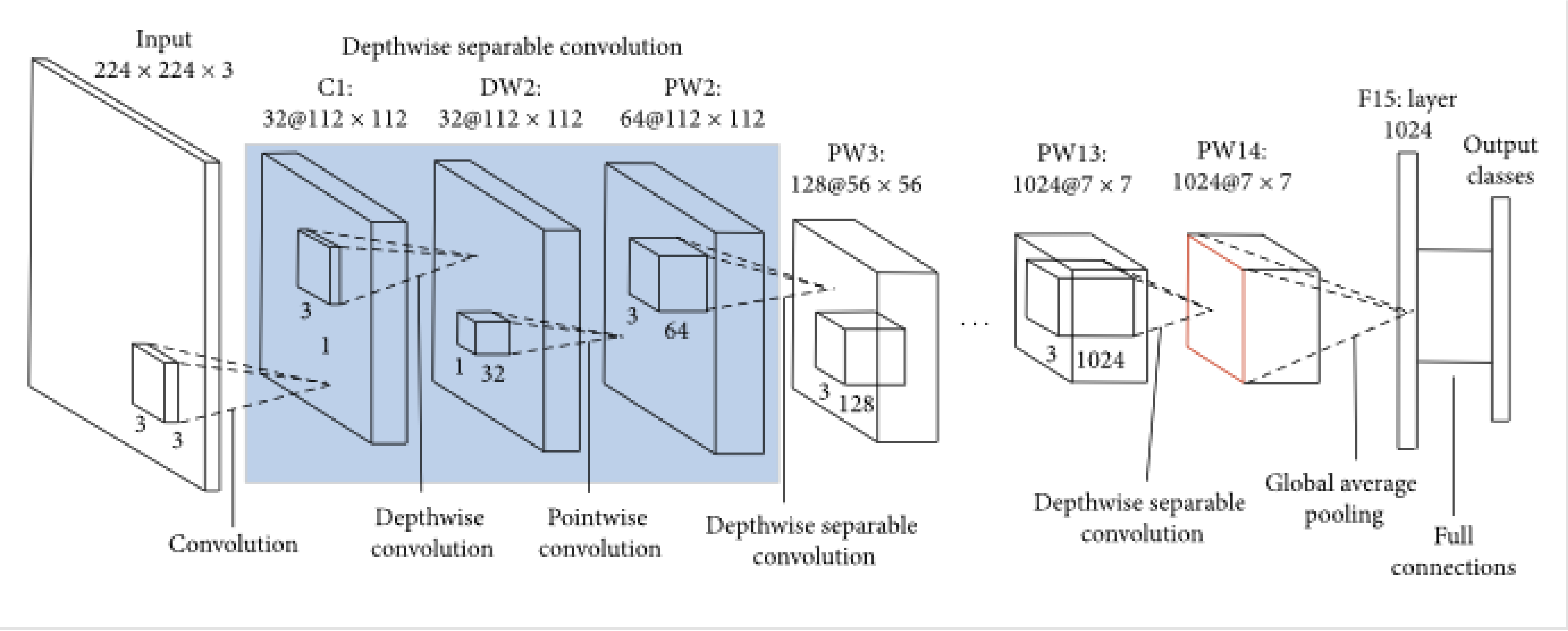
https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788629416/2/ch02lvl1sec14/densely-connected-convolutional-networks-densenet

Xception



https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568

MobileNet



https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568

Konwolucyjne sieci neuronowe

LeNet, AlexNet, VGG

ResNet, DenseNet

Inception

Xception, MobileNet

Fully Convolutional Network, DeconvNet (segmentacja)

Rekurencyjne sieci neuronowe

komórka LSTM

komórka GRU

bidirectional RNN

Uczenie nienadzorowane

self organizing maps, można na przykładzie sieci Kohonena

Restricted Boltzmann machine

autoenkoder wariacyjny VAE

Różności

Siamese Network

R-CNN

Mask R-CNN

Attention Module