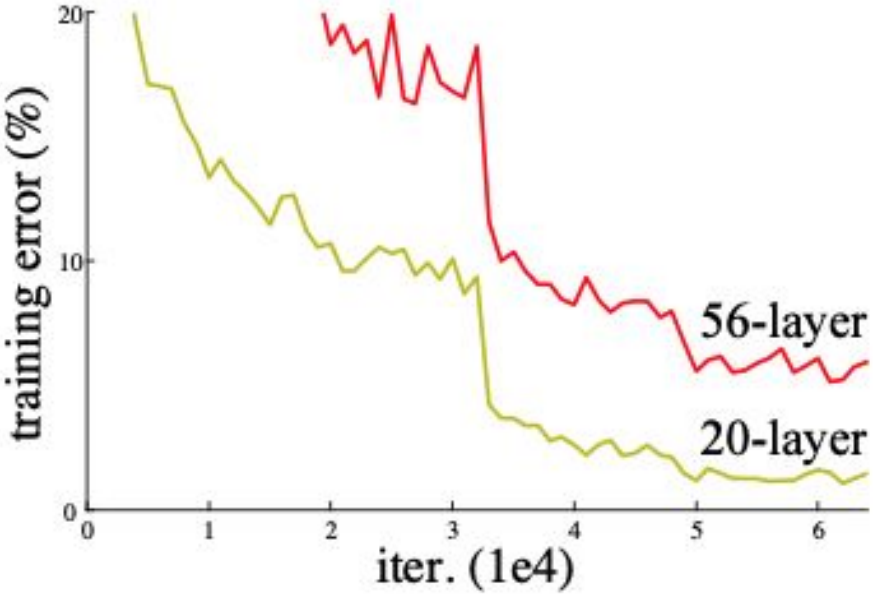
WE HAVE TO GO DEEPER!

09.01.2024 Computer Vision Seminar 23/24

Agenda

1. Inception - przypomnienie
2. Zanikający gradient
3. Połączenia “skrótowe”
4. ResNet
5. Tutorial transfer learning w PyTorch


THAT'S NOT ENOUGH
WE HAVE TO GO DEEPER



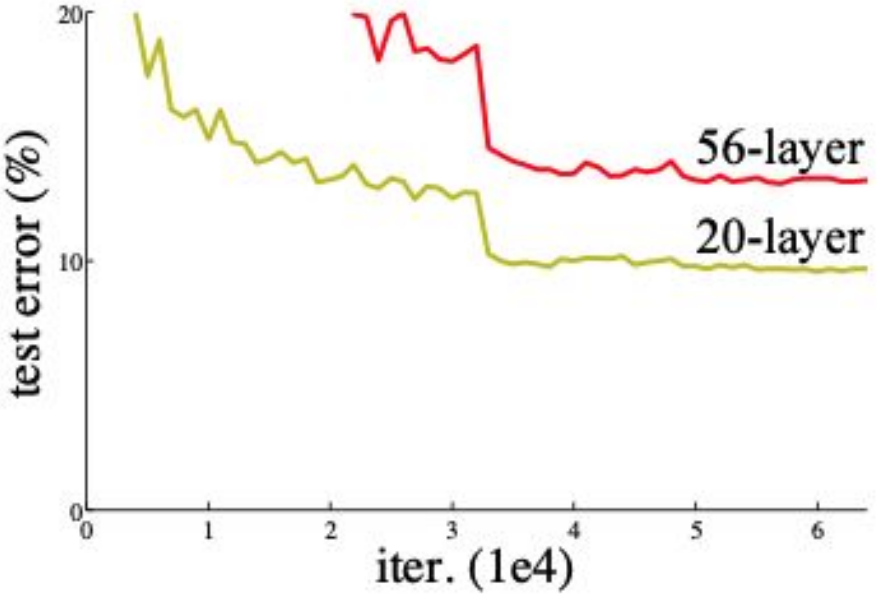
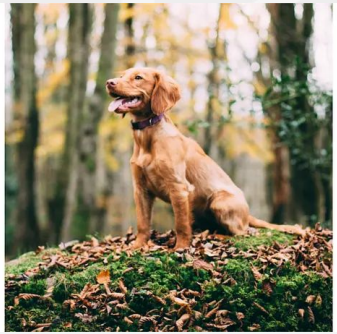
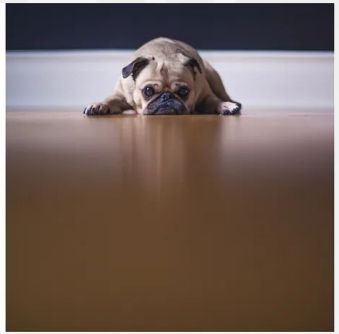


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

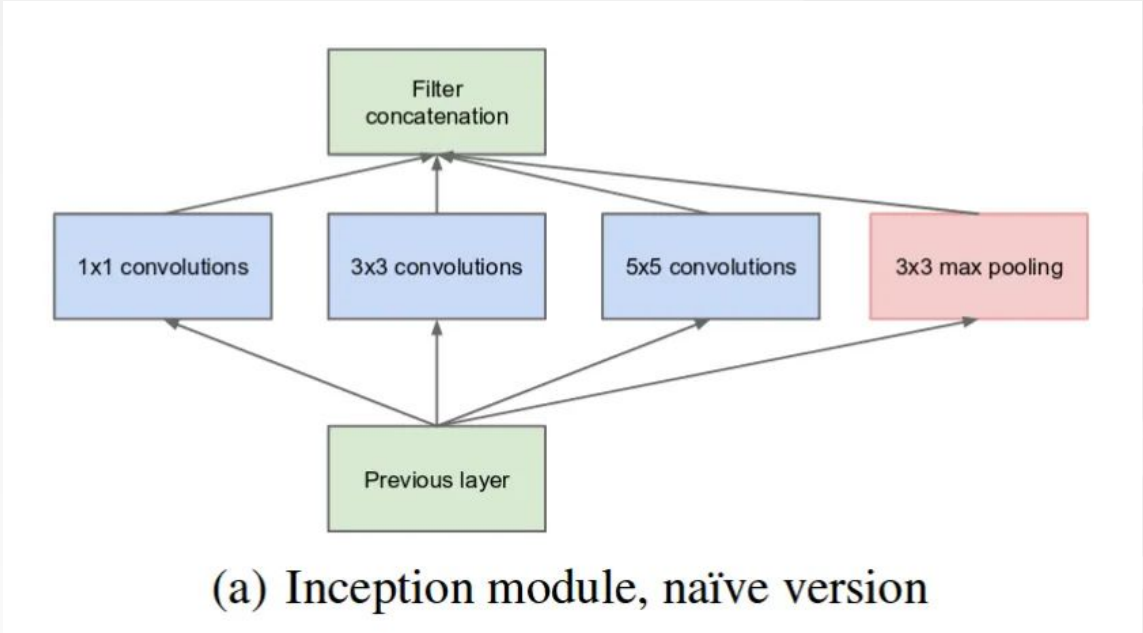
Inception - motywacja

- Jak dobrać rozmiar filtra? (dla cech różnego rozmiaru)
- Bardzo głębokim sieciom grozi przeuczenie i zanik gradientu
- Układanie wielu warstw konwolucyjnych jest kosztowne obliczeniowo

https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202

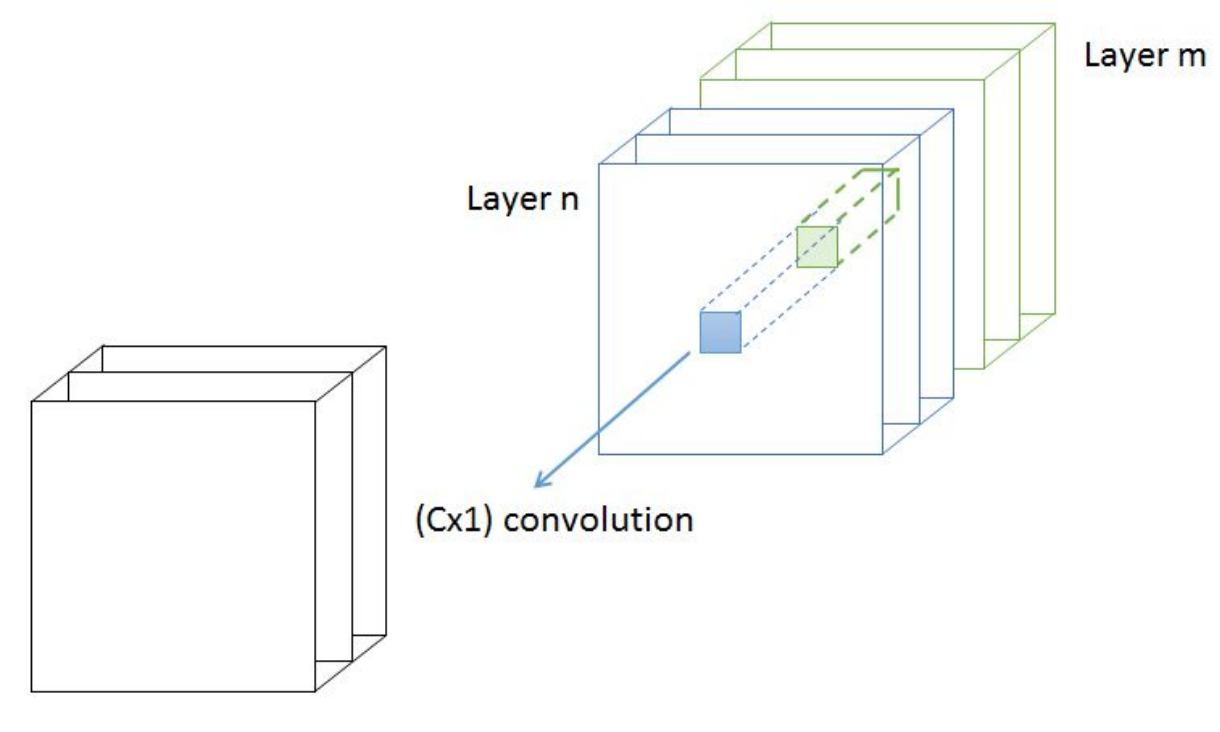




Rozwiązanie - różne wielkości filtrów na tym samym poziomie



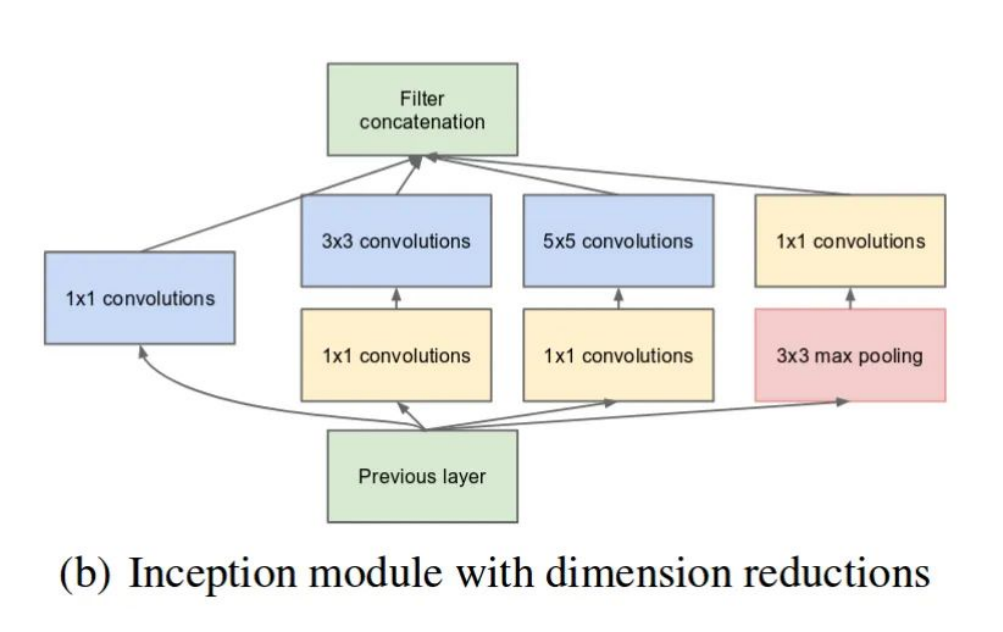
(a) Inception module, naïve version

Konwolucja 1x1



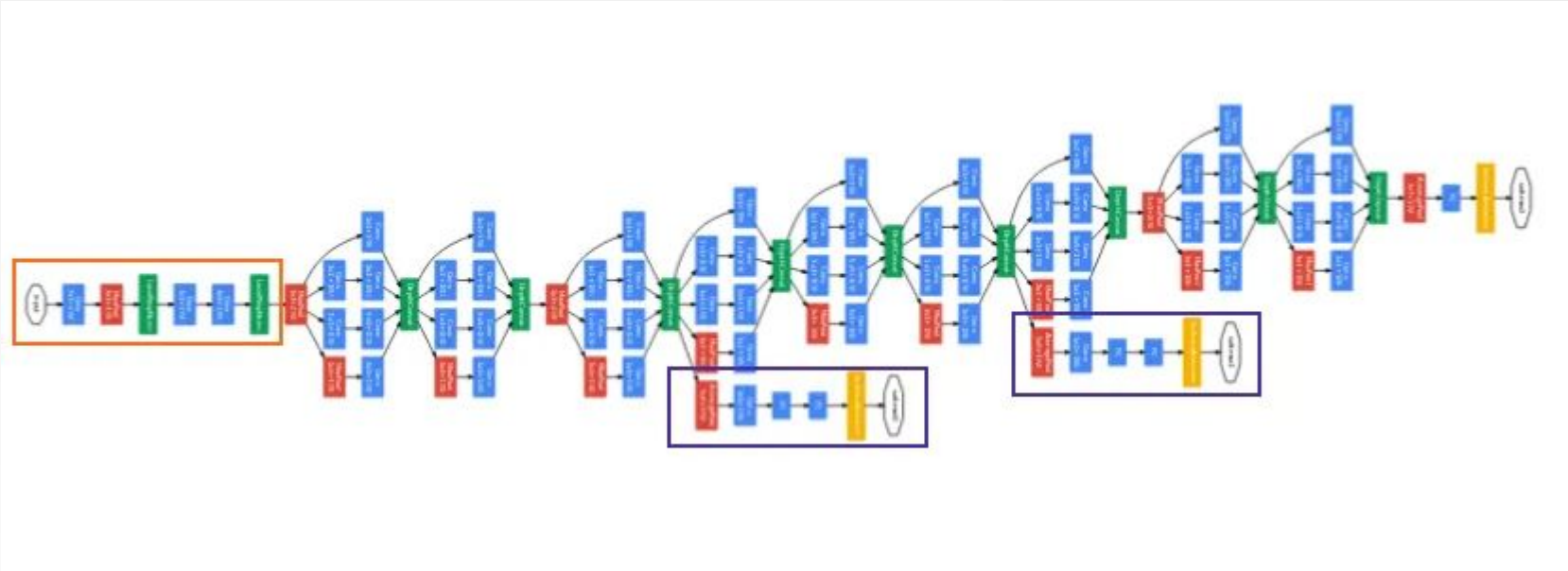
Redukcja wymiarów

- Aby obniżyć koszt obliczeniowy, autorzy dodali konwolucję 1x1 (redukcja wymiarów).
- Konwolucja 1x1 jest znacznie mniej złożona obliczeniowo niż konwolucja 3x3 czy 5x5



(b) Inception module with dimension reductions

GoogLeNet

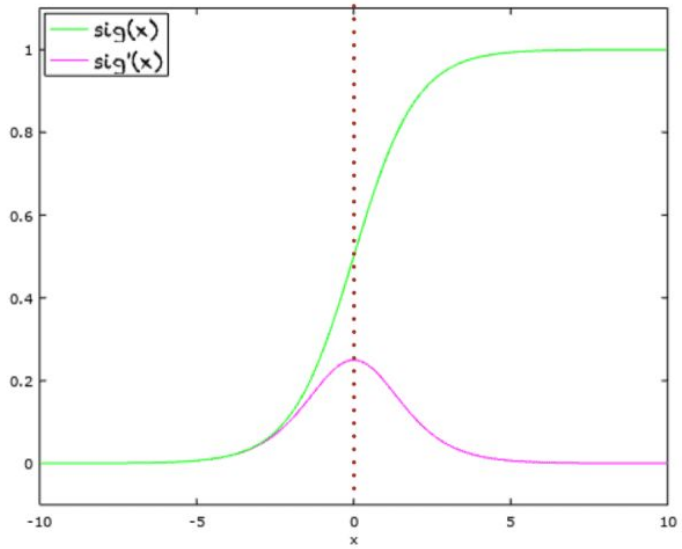


Zanik gradientu

- Gradient (wektor pochodnych) jest używany przy aktualizacji wag modelu
- Jeśli ułożymy kolejno kilka warstw o określonych funkcjach aktywacji, może dojść do zaniku
- Zbyt mały gradient uniemożliwia stabilne uczenie sieci

https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

Sigmoida i jej pochodna - przykład



Plot of σ(x) and its derivate σ'(x)

Domain: (−∞, +∞)
Range: (0, +1)
σ(0) = 0.5

Other properties

$$\sigma(x) = 1 - \sigma(-x)$$

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^{x}}{e^{x}+1}$$

$$\sigma'(x) = \sigma(x)(1-\sigma(x))$$

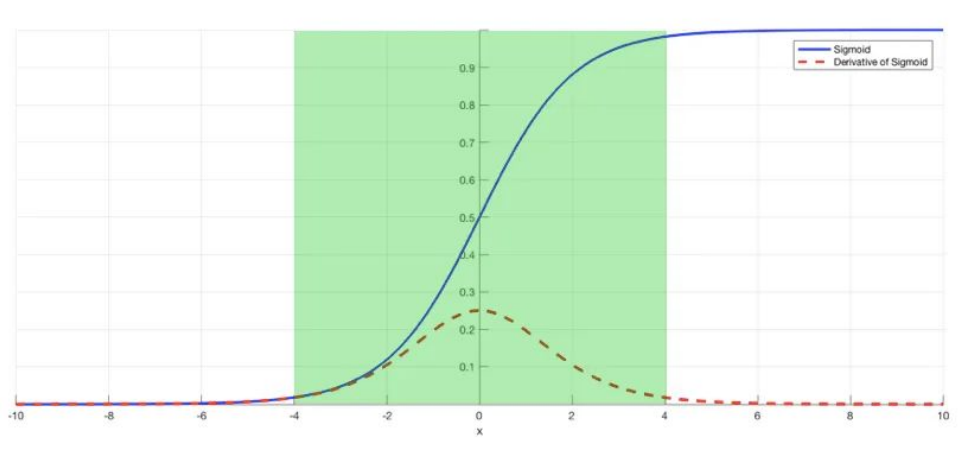
Sigmoida i jej pochodna - przykład

- Duża zmiana wartości - mała zmiana pochodnej
- W propagacji wstecznej wartości pochodnej są mnożone przez siebie (chain rule, różniczkowanie funkcji złożonej)
- Jeśli kilka kolejnych warstw używa aktywacji sigmoidą, mnożymy kilka bardzo małych wartości, co powoduje dążenie gradientu do 0

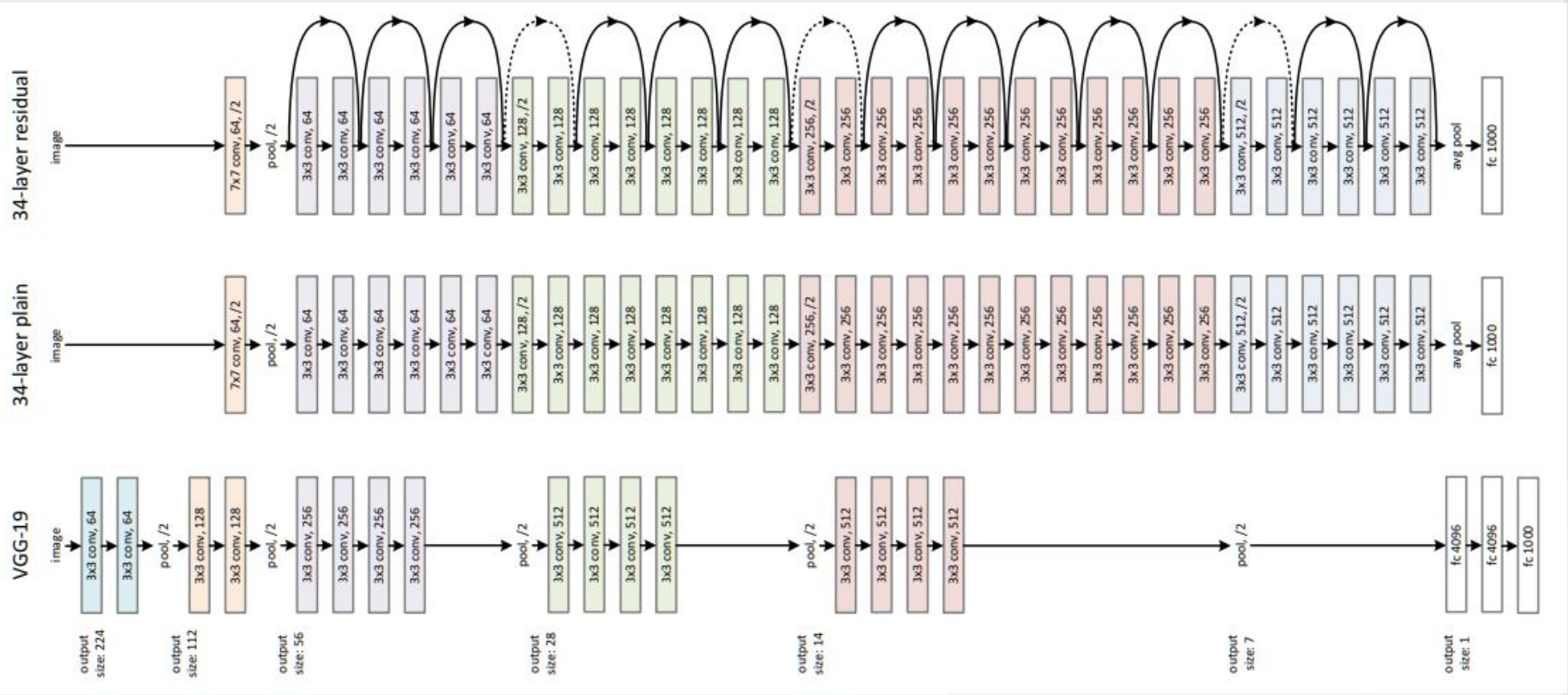
https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

Zanikający gradient - rozwiązania

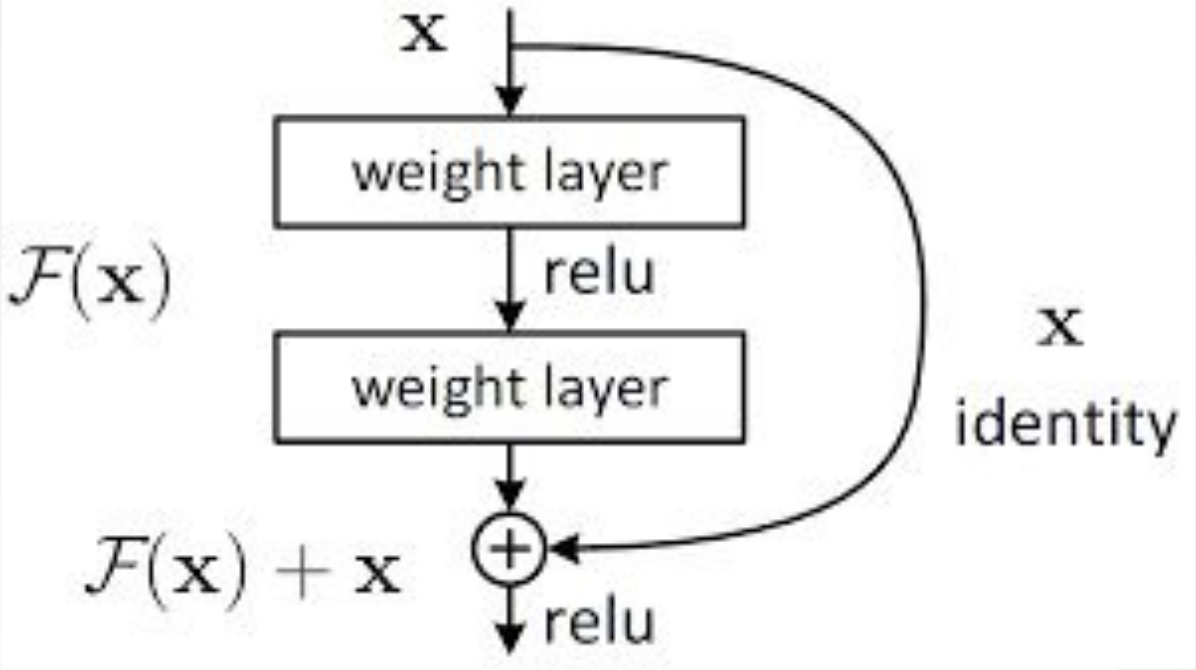
- Inne funkcje aktywacji, np. ReLU
- Batch normalization
- Połączenia rezydualne



https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

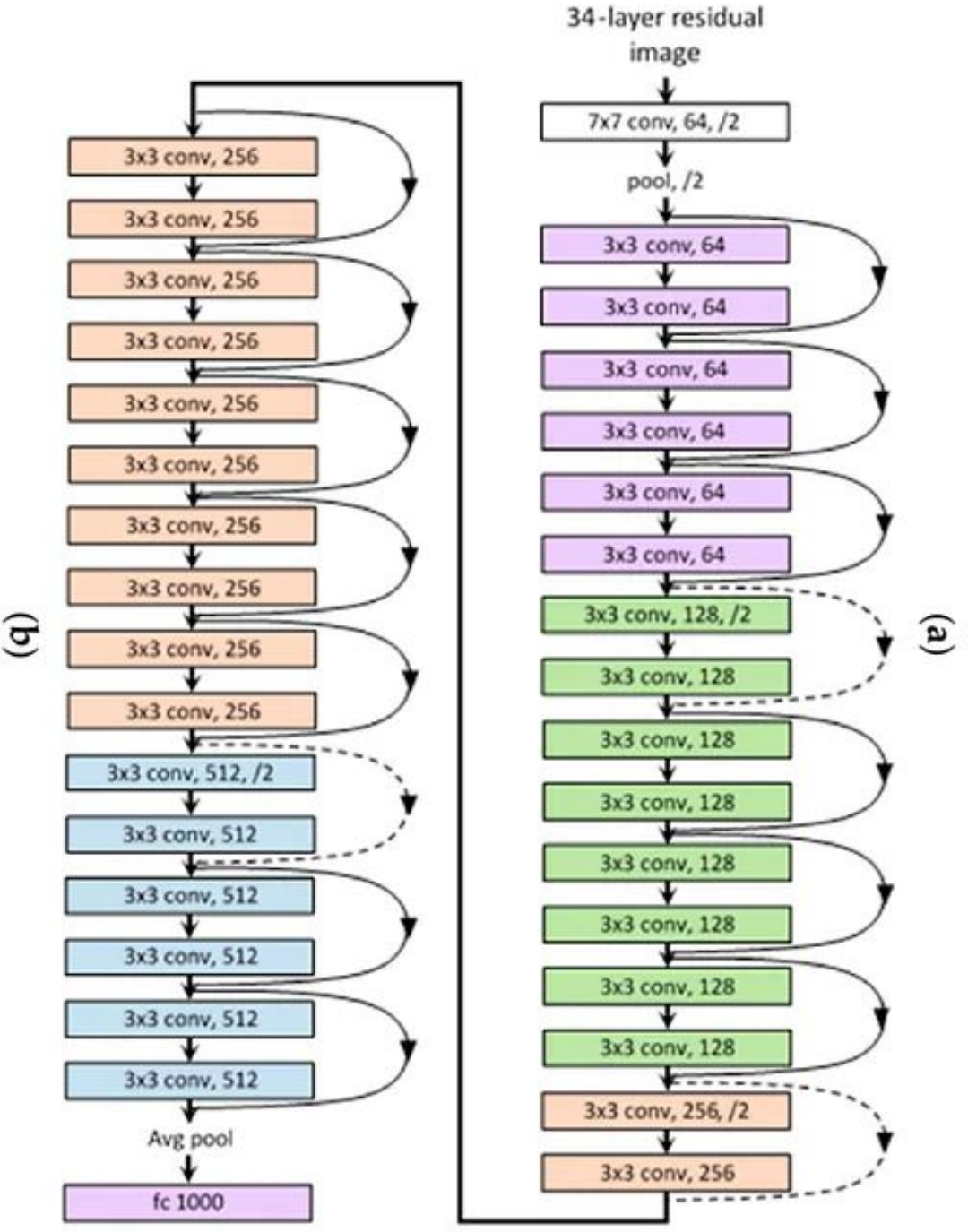

VGG-19
image
3x3 conv, 64
3x3 conv, 64
pool, /2
output size: 224
3x3 conv, 128
3x3 conv, 128
pool, /2
output size: 112
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
output size: 56
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 28
pool, /2
output size: 14
pool, /2
output size: 7
fc 4096
fc 4096
fc 1000
output size: 1
34-layer plain
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 256, /2
3x3 conv, 512, /2
avg pool
fc 1000
34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 256, /2
3x3 conv, 512, /2
avg pool
fc 1000

Blok ResNet - połączenia “skrótowe”

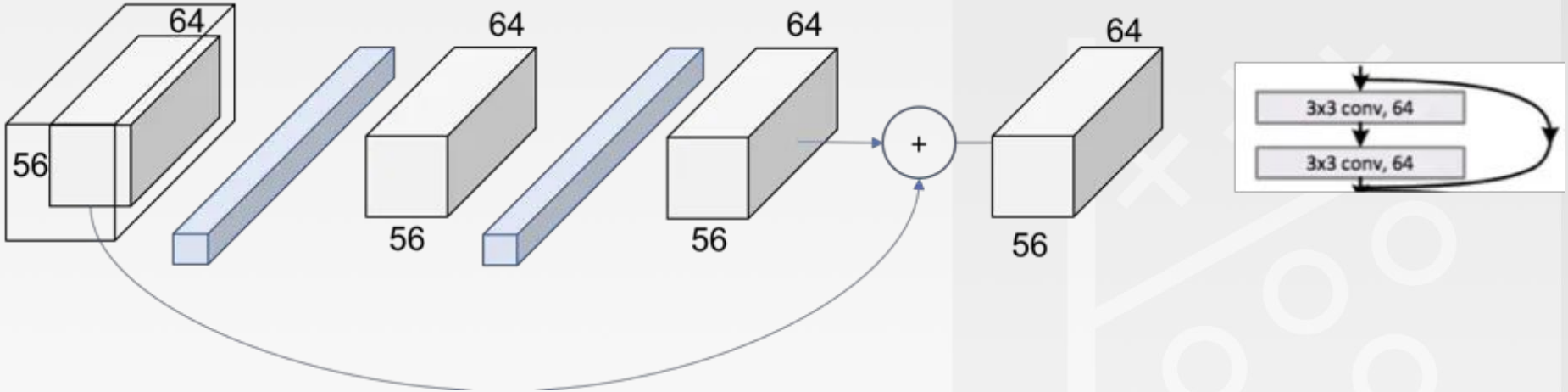


In this paper, we address the degradation problem by introducing a *deep residual learning* framework. Instead of hoping each few stacked layers directly fit a desired underlying mapping, we explicitly let these layers fit a residual mapping. Formally, denoting the desired underlying mapping as $\mathcal{H}(\mathbf{x})$, we let the stacked nonlinear layers fit another mapping of $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$. The original mapping is recast into $\mathcal{F}(\mathbf{x}) + \mathbf{x}$. We hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. To the extreme, if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers.

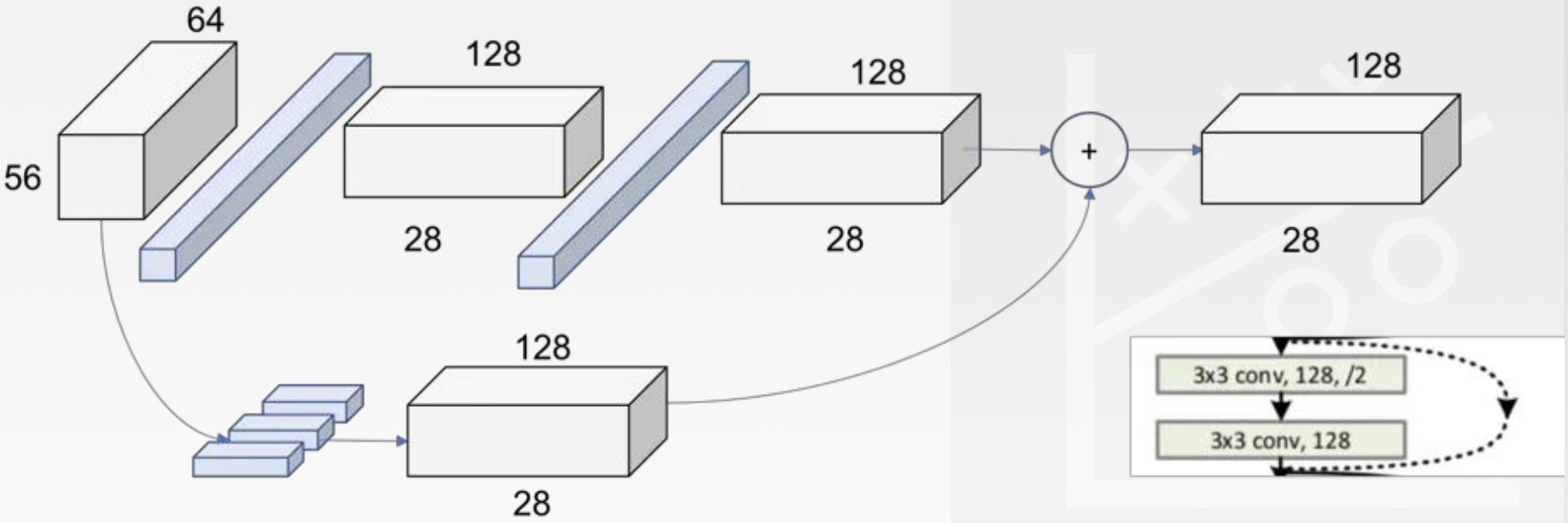
The formulation of $\mathcal{F}(\mathbf{x}) + \mathbf{x}$ can be realized by feedforward neural networks with “shortcut connections” (Fig. 2). Shortcut connections [2, 34, 49] are those skipping one or more layers. In our case, the shortcut connections simply perform *identity* mapping, and their outputs are added to the outputs of the stacked layers (Fig. 2). Identity shortcut connections add neither extra parameter nor computational complexity. The entire network can still be trained end-to-end by SGD with backpropagation, and can be easily implemented using common libraries (*e.g.*, Caffe [19]) without modifying the solvers.


34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
(a)
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
Avg pool
fc 1000
(b)

Blok ResNet



Blok ResNet - zwiększanie głębokości



model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Table 3. Error rates (%, **10-crop** testing) on ImageNet validation. VGG-16 is based on our test. ResNet-50/101/152 are of option B that only uses projections for increasing dimensions.

The Top-1 error is the proportion of the time the classifier does not provide the highest score to the correct class.
The Top-5 error rate is the percentage of times the classifier failed to include the proper class among its top five guesses.

Number of Layers	Number of Parameters
ResNet 18	11.174M
ResNet 34	21.282M
ResNet 50	23.521M
ResNet 101	42.513M
ResNet 152	58.157M

Table 1. ResNets architectures for ImageNet

ResNet - podsumowanie

- Sieci rezydualne rozwiązują problem zanikającego gradientu, ponieważ gradient może swobodnie przepływać przez połączenia skrótowe
- Dzięki zastosowaniu połączeń skrótowych, sieci mogą być znacznie głębsze, co z kolei przekłada się na lepsze wyniki

Materiały

- TRANSFER LEARNING FOR COMPUTER VISION TUTORIAL

https://colab.research.google.com/drive/1c9LNEwzSuSQrvFKfzDBJ4yLiNgAVyDsV?usp=sharing

https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html

- Understanding and visualizing ResNets

https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8