Problemy klasycznych sieci neuronowych

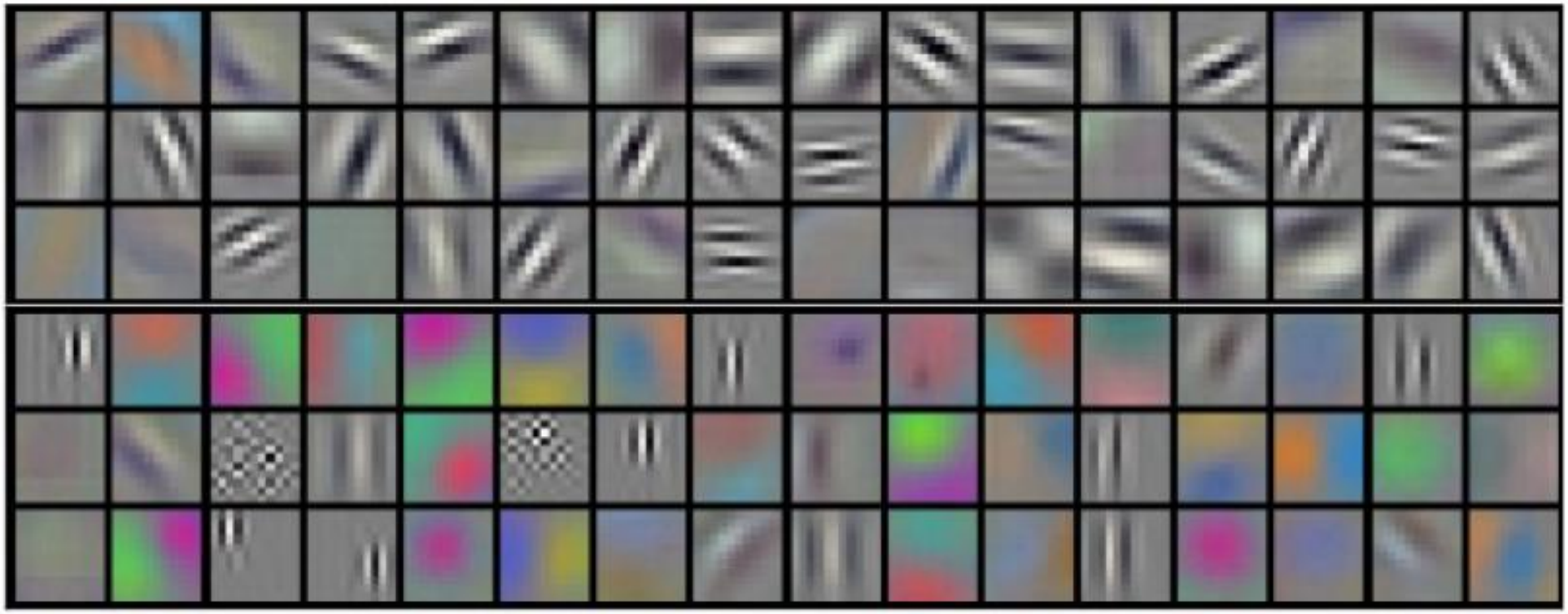
Dość bolesną sprawą w uczeniu klasycznych sieci neuronowych jest zanikanie gradientu we wstecznej propagacji błędu.

Można też zastanawiać się nad sensownością brania pod uwagę całego obrazu (być może dużego) do wyznaczenia wartości każdego z neuronów w każdej warstwie. Jeśli interpretować pierwsze warstwy, jako warstwy wydobywające cechy, to naturalne wydaje się skorzystanie z lokalności charakterystyki obrazu.

Takie podejście umożliwi wydobycie na pierwszym poziomie prostych cech obrazu (krawędzi, plam koloru), z których w następnych warstwach mogłyby być składane bardziej złożone cechy opisujące obraz.

Sporo tego typu cech można skonstruować ręcznie, ale o ileż przyjemniej byłoby uruchomić uczenie, które wydobędzie je automatycznie i dopasuje do konkretnego problemu.

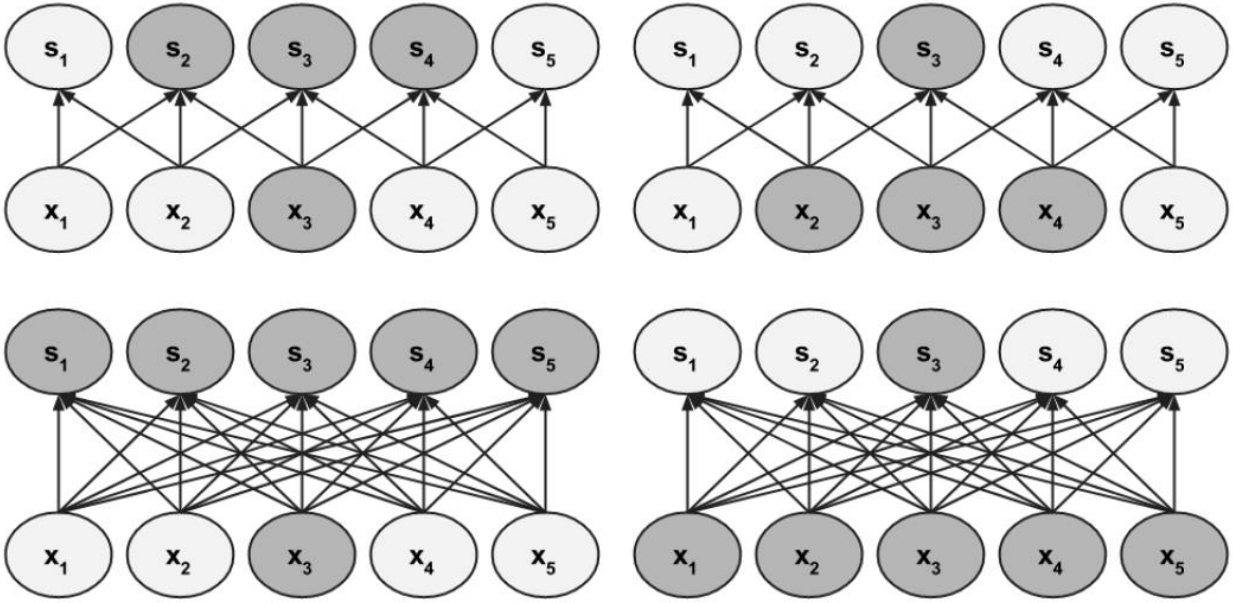
Na przykład…



96 filtrów 11x11x3 z pierwszej warstwy sieci AlexNet uczonej na zbiorze ImageNet.

Lokalność cech i analizy

W sieciach splotowych filtr ma mniejszy rozmiar, niż wejście:



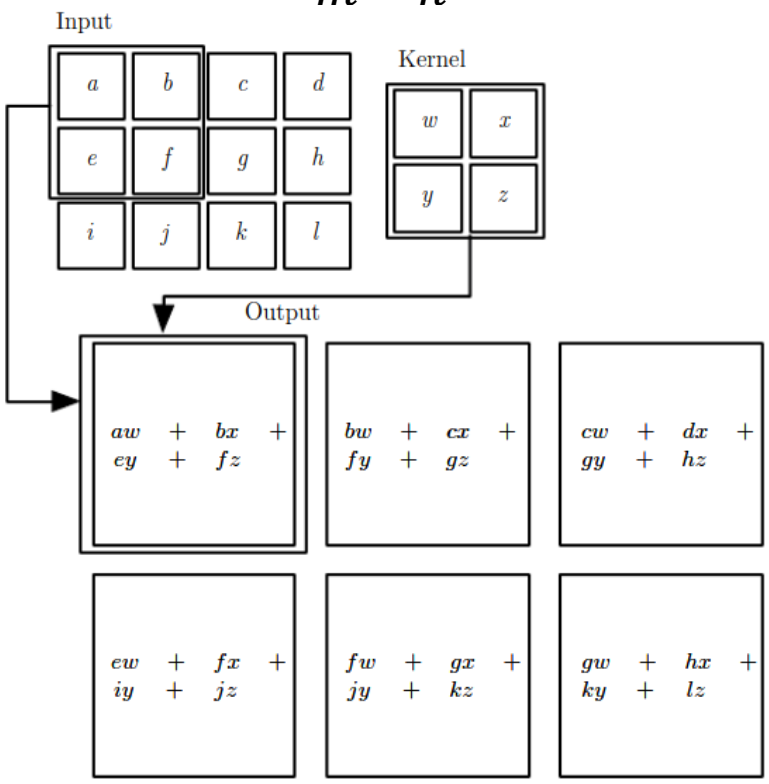
Względem tradycyjnych sieci neuronowych w sieciach splotowych mamy trzy istotne cechy:

- Lokalność połączeń
- Współdzielenie parametrów
- Przenoszenie przesunięcia (*translation equivariance*)

Splot

Splot jest operacją ***liniową***, która na podstawie obrazu i filtru (jądra) liczy wartości cech (mapę cech).

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n)$$



Parametry operacji splotu

Na sposób realizacji splotu, szczególnie na rozmiar mapy wyjściowej, mają wpływ trzy parametry:

- f : Rozmiar filtra (jądra) – *filter (kernel) size*
  Oczywiste, ale warto pamiętać, że filtr bierze całą głębokość obrazu (np. 11x11x3 – rozmiar filtru pierwszej warstwy splotowej w AlexNet)
- s : Wielkość kroku – *stride*
- p : Uzupełnianie przy krawędziach – *padding* (zwykle $p = \lfloor f/2 \rfloor$)
  Bez uzupełniania (*valid padding*) – oznacza, że wyjście splotu będzie mieć mniejszy rozmiar niż oryginalny obraz ($p = 0$).
  Z uzupełnianiem (*same padding*) – filtr pracuje także przy krawędziach obrazu i trzeba zdecydować, jakie wartości nadać „brakującym” pikselom ($p \geq 1$).

$$output_{size} = \left\lfloor \frac{n - f \ + 2p}{s} + 1 \right\rfloor$$

Uzupełnianie przy krawędziach (padding)

Uzupełnianie zerami (zero padding)	Uzupełnianie sąsiadem (replicated padding)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	4	2	3	0	0
0	0	6	1	1	7	0	0
0	0	1	2	1	4	0	0
0	0	1	5	1	2	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

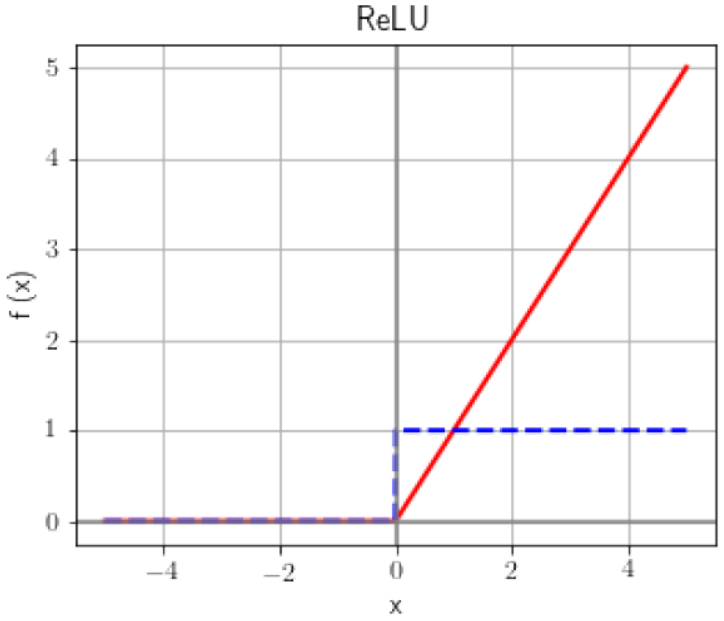
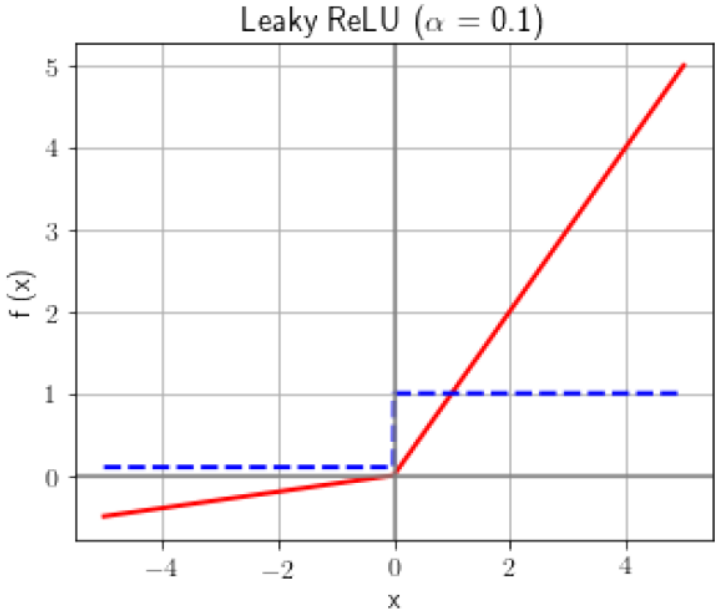
1	1	1	4	2	3	3	3
1	1	1	4	2	3	3	3
1	1	1	4	2	3	3	3
6	6	6	1	1	7	7	7
1	1	1	2	1	4	4	4
1	1	1	5	1	2	2	2
1	1	1	5	1	2	2	2
1	1	1	5	1	2	2	2

Uzupełnianie przy krawędziach (padding)

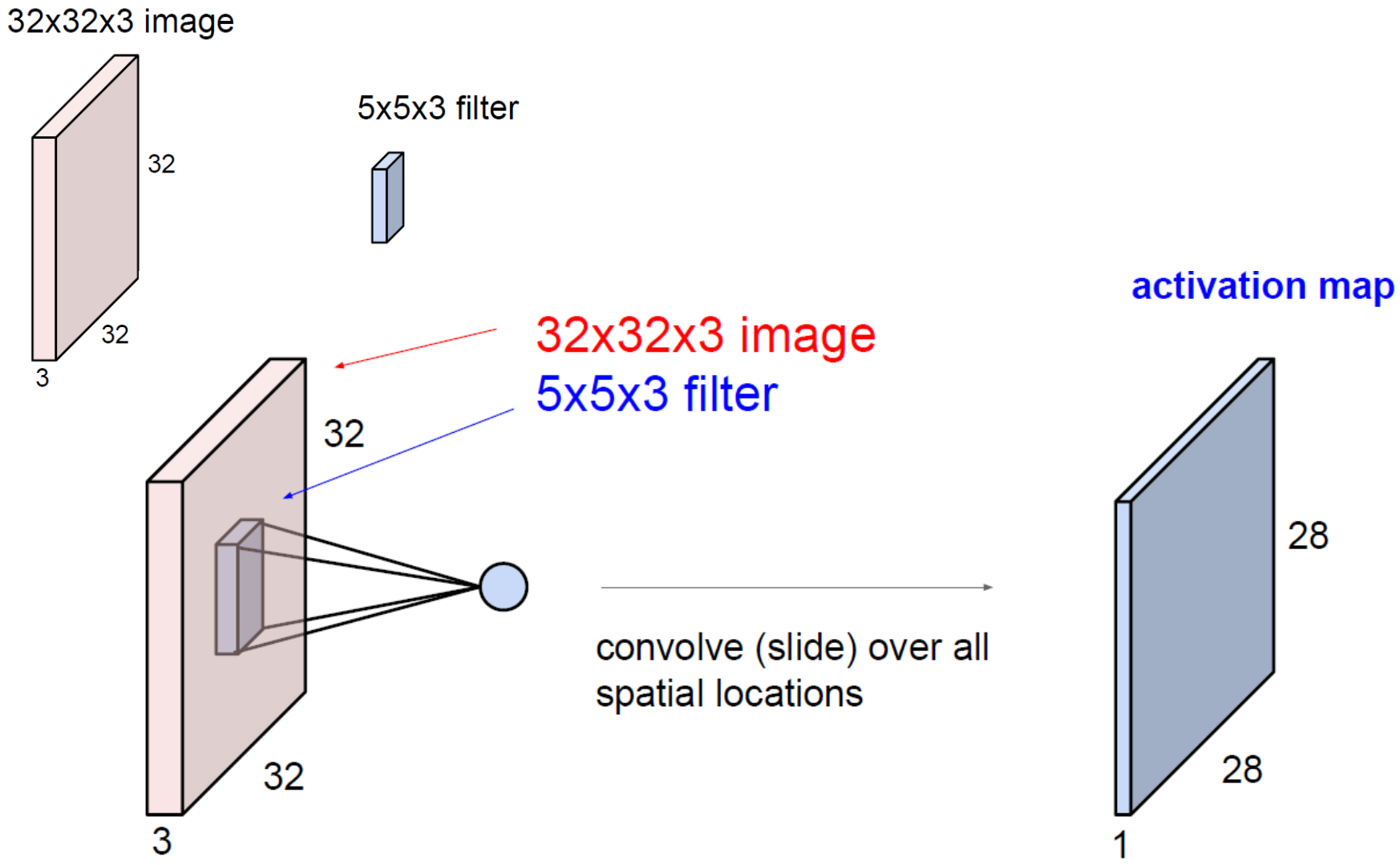
Uzupełnianie z odbiciem (symmetric padding)	Uzupełnianie z zawinięciem (circulated padding)
	

Dodanie nieliniowości

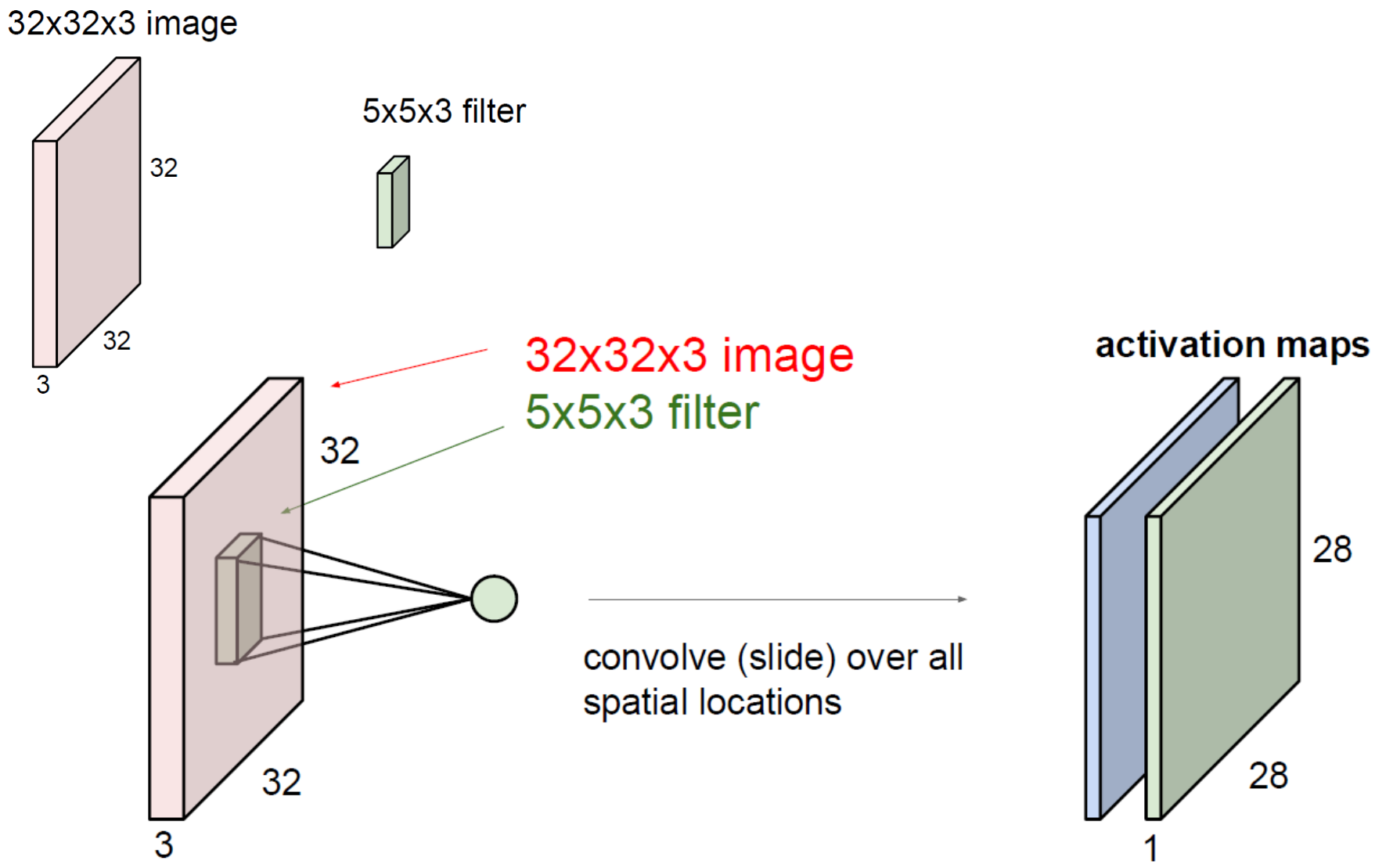
Splot jest operacją liniową na danych wejściowych. Wprowadzenie nieliniowości jest niezbędne, żeby złożenie warstw nie dało w efekcie innej operacji liniowej względem wejścia. Typowo używane funkcje aktywacji, to:

ReLU Rectified Linear Unit $f(x) = \max(0, x)$	Leaky ReLU Leaky Rectified Linear Unit $f(x) = \max(\alpha x, x)$
	

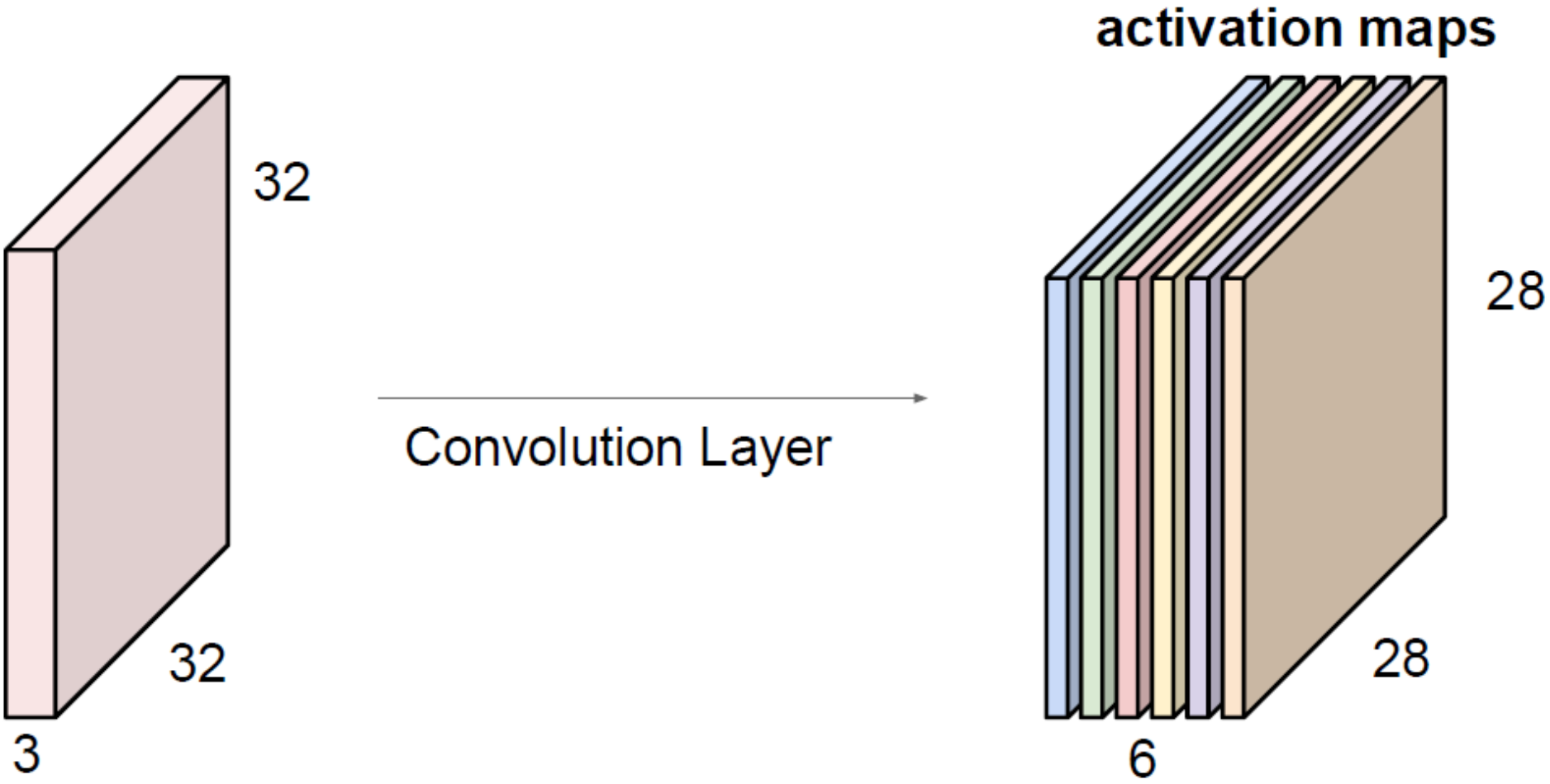
Warstwa splotowa (wejście RGB)



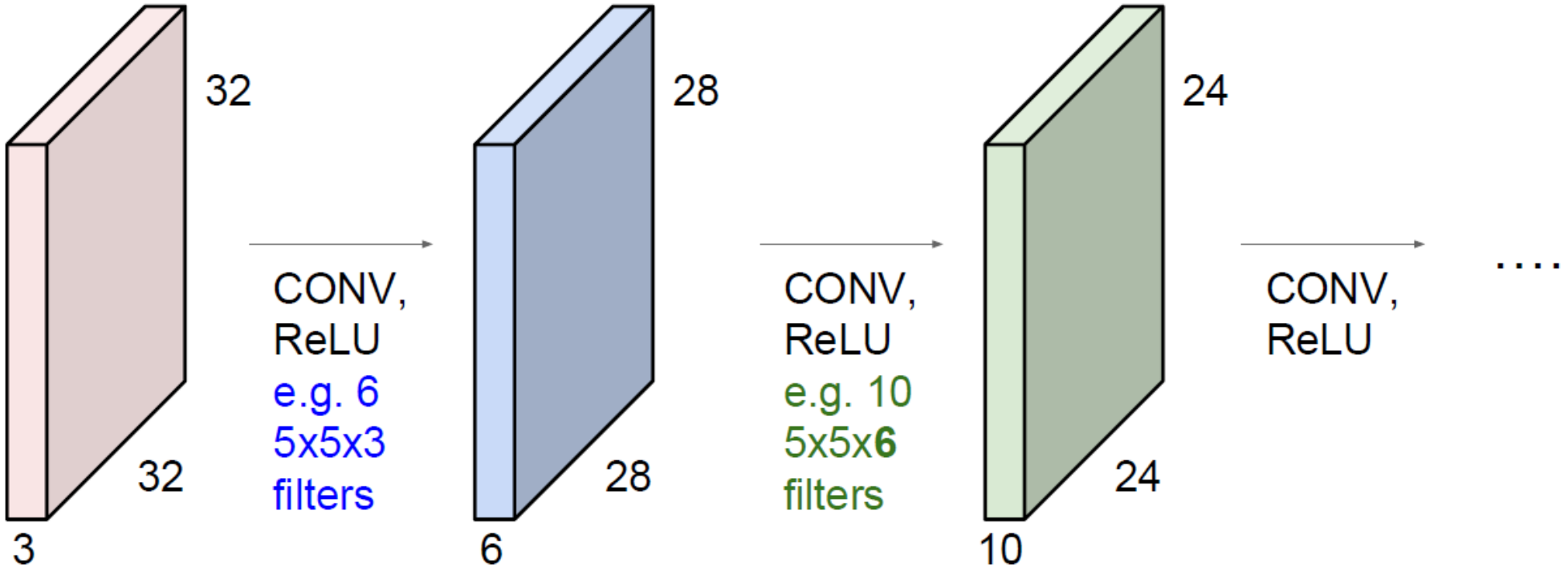
Warstwa splotowa (drugi filtr)



Warstwa splotowa (sześć filtrów)

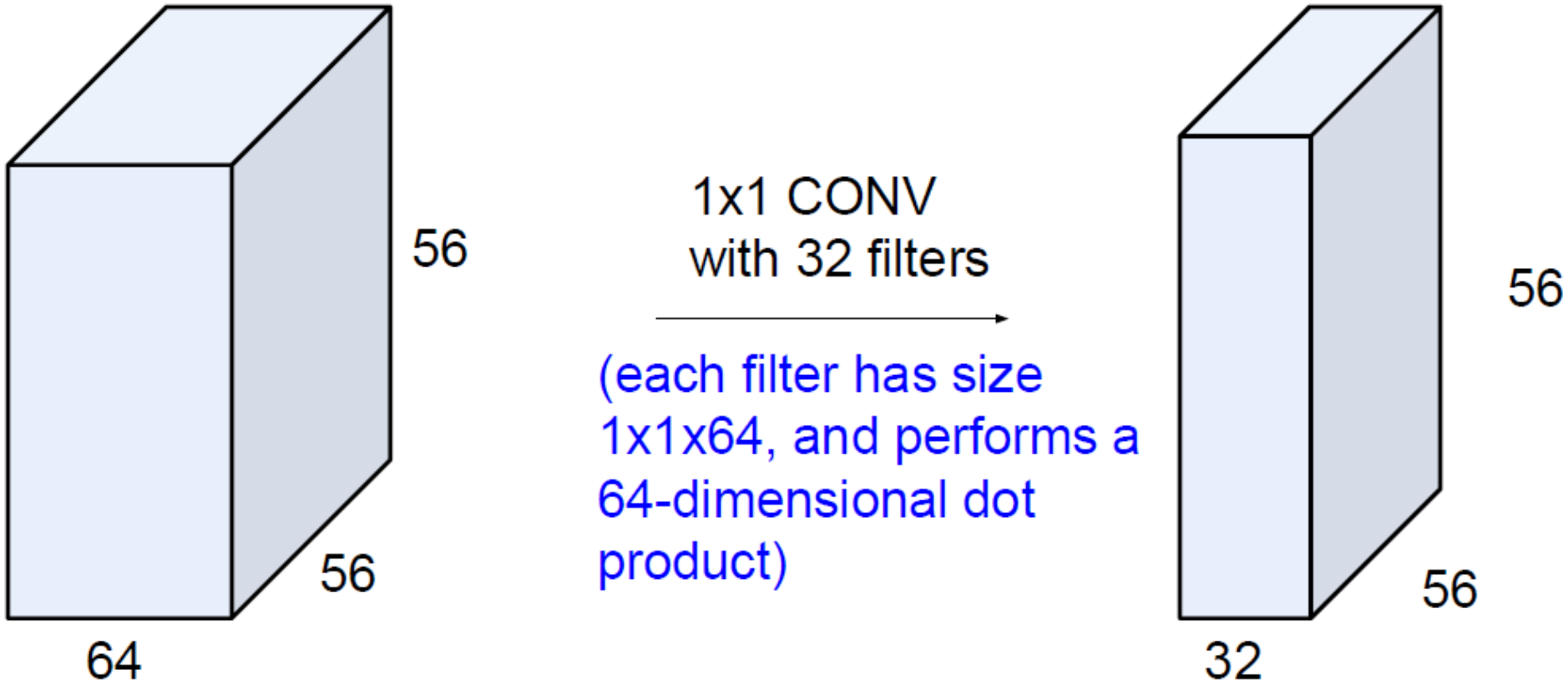


Kilka warstw splotowych



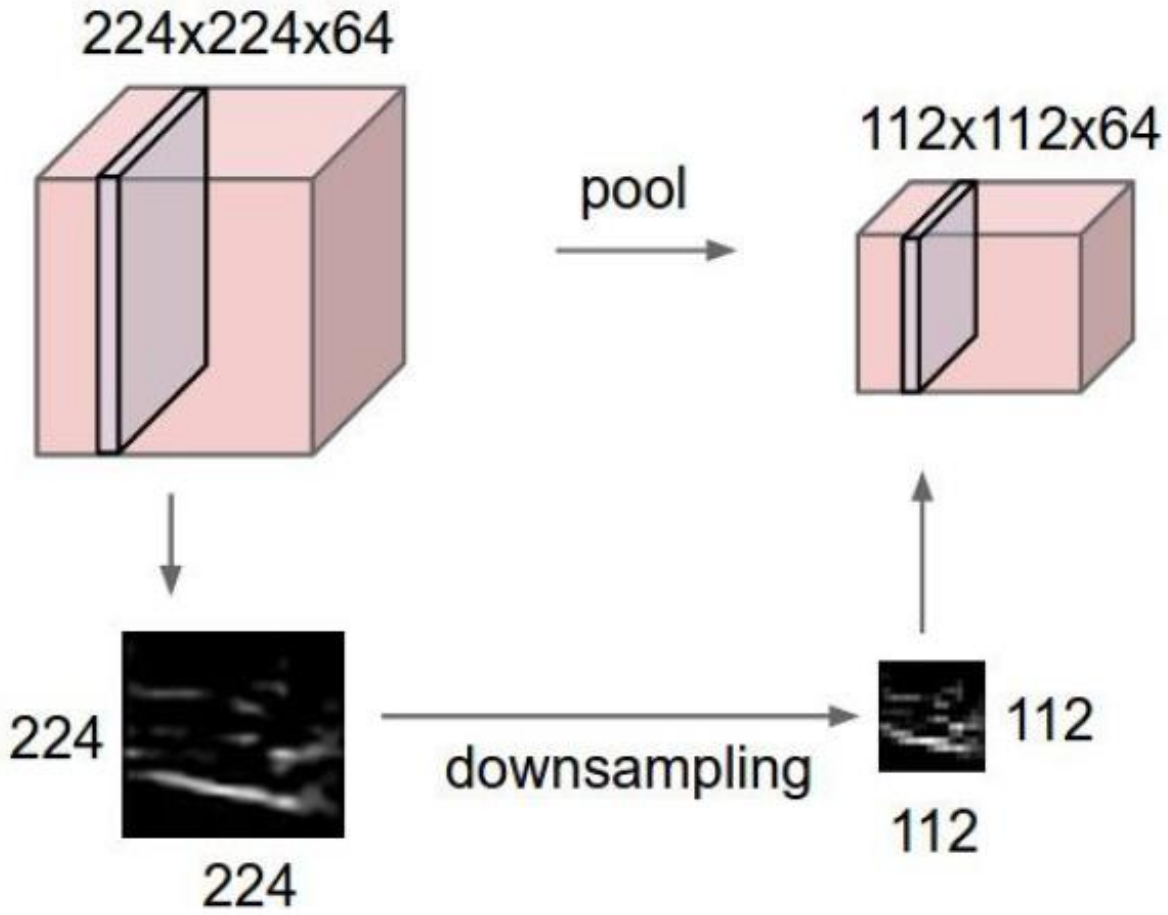
Przy braku uzupełniania zmniejsza się szerokość i wysokość mapy cech; liczba filtrów powoduje zwiększenie głębokości mapy, co pociąga za sobą zwiększenie rozmiaru filtrów.

Redukcje mapy



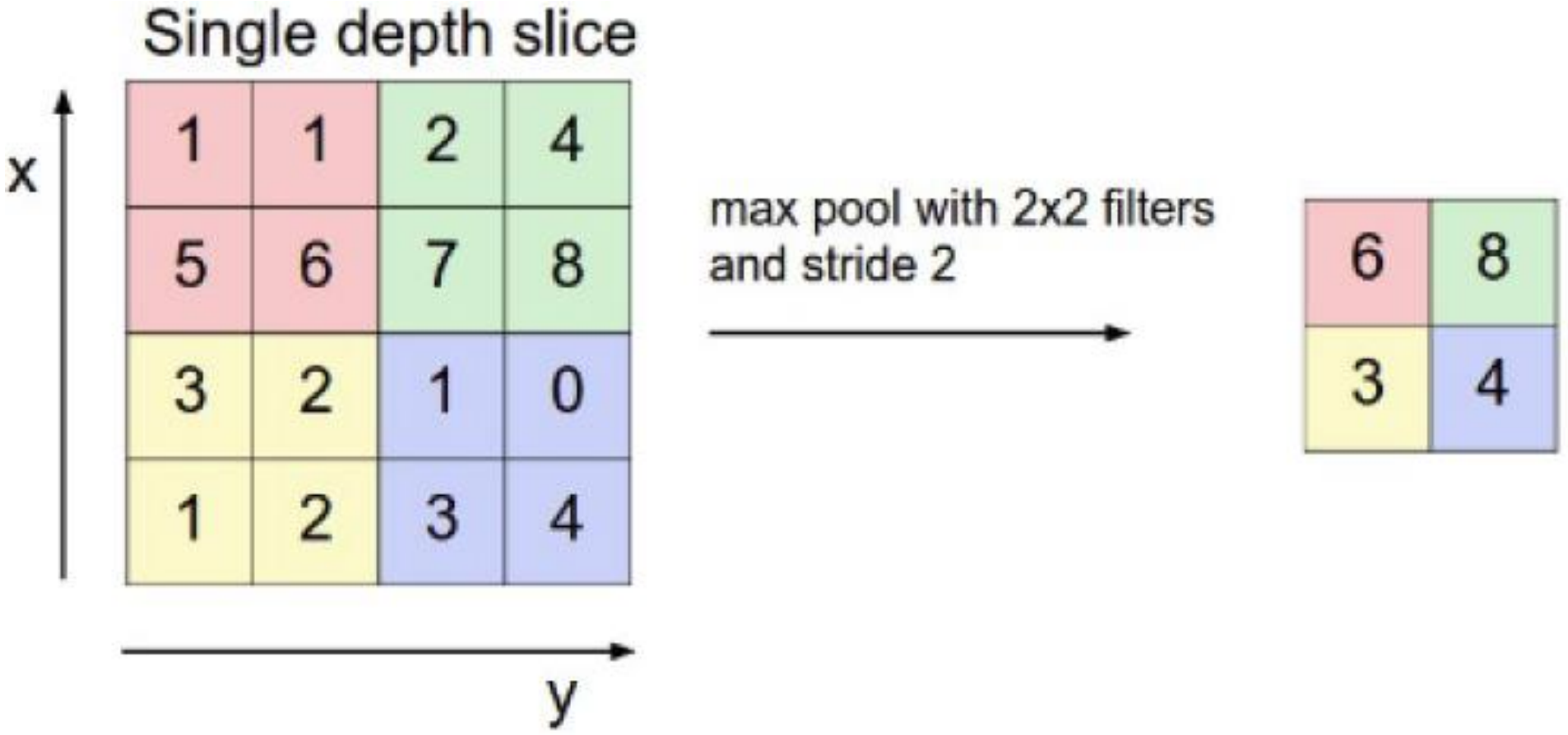
Filtr 1x1 ma spory sens – pozwala na redukcję głębokości mapy.

Redukcje szerokości i wysokości (*pooling*)



Redukcje działają na każdej warstwie niezależnie. Najczęściej stosuje się branie wartości średniej lub maksymalnej w oknie.

Redukcja – wartość maksymalna



Normalizacja

Stosowane są dwie podstawowe podejścia do normalizacji:

Lokalna normalizacja odpowiedzi – zasadniczo chodzi o blokowanie sąsiadów neuronu pobudzonego (rzadko obecnie stosowana, bo wpływ na jakość uczenia ma minimalny);

Normalizacja wsadu – przeprowadza wybielanie (*whitening*) wejść próbek tworzących wsad.

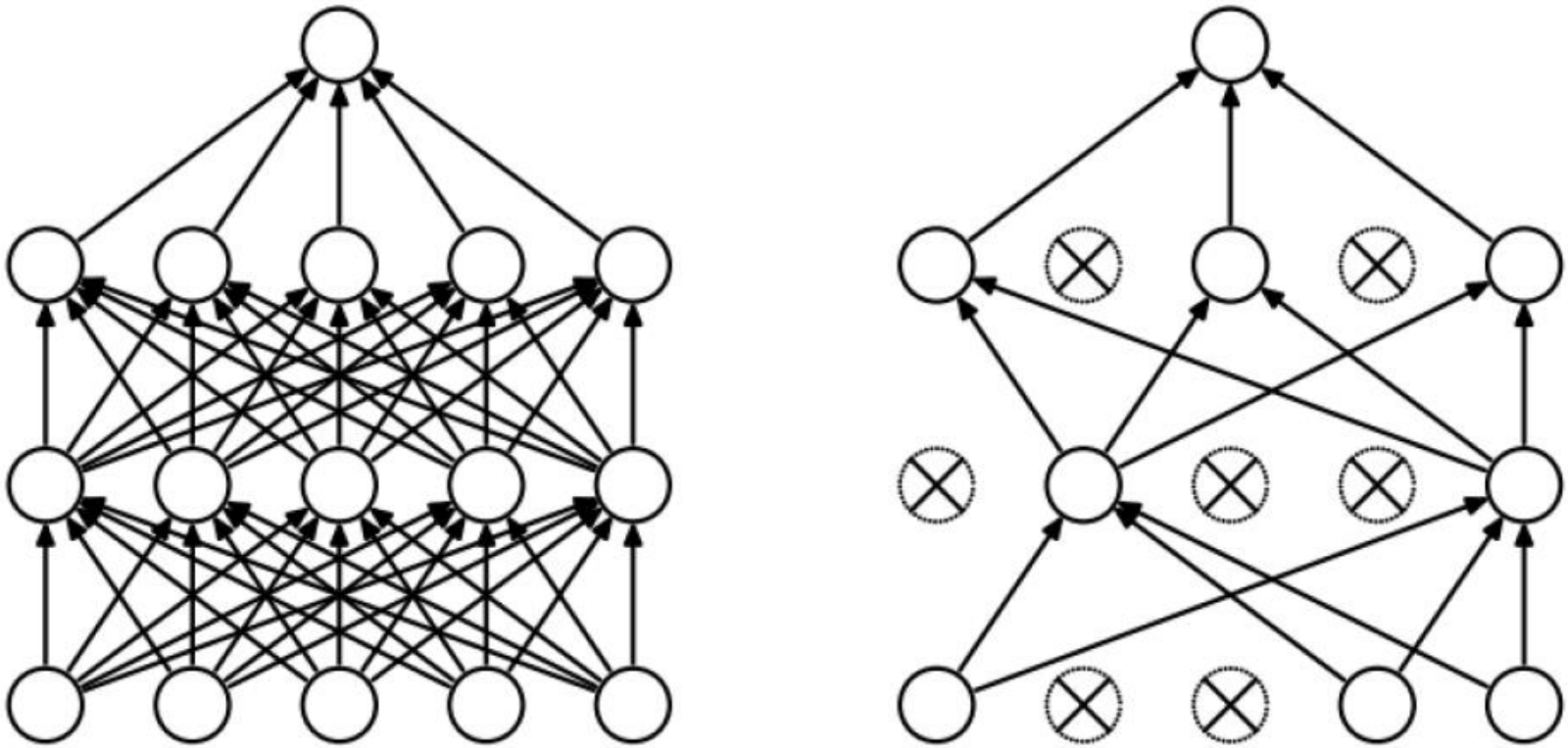
Regularyzacja

L1 regularization defines the regularization penalty as $\lambda|w|$. It tends to drive the weights sparse: some weights are very close to zero and some have large values. In other words, L1 regularization ends up using a sparse subset of the inputs and become invariant to noisy inputs.

L2 regularization defines the regularization penalty as λw^2. It is common to see a constant 1/2 making the penalty equal to $\lambda/2w^2$ to make the gradient simply λw instead of $2\lambda w$. Since L2 regularization gives high penalty for large weight values, the weights tend to be diffused with smaller values. In practice, if explicit feature selection is not concerned, L2 regularization is superior to L1.

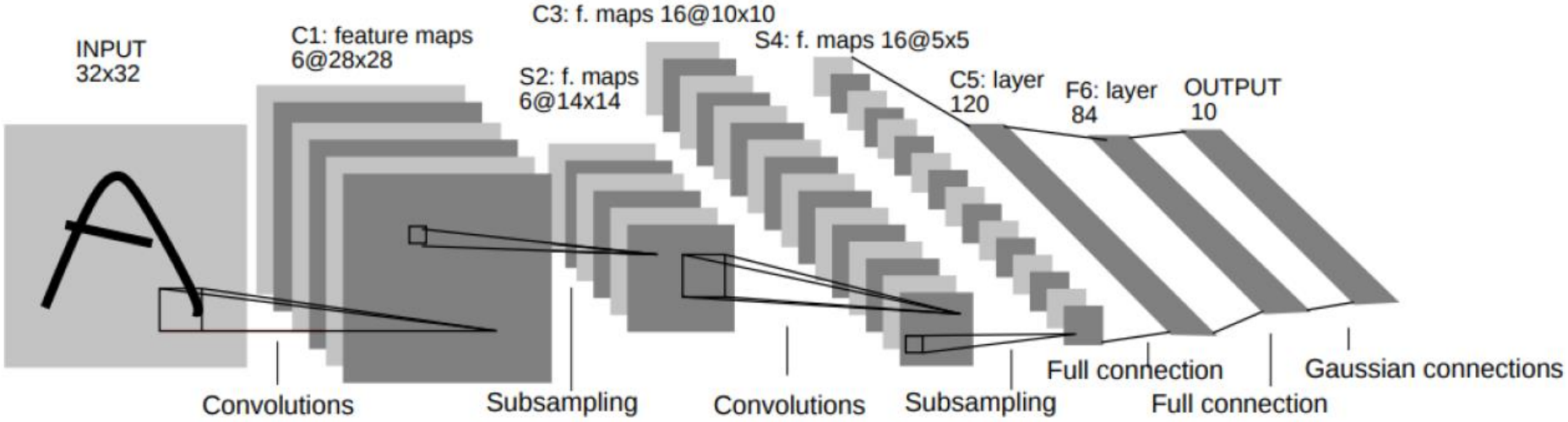
Elastic net regularization is a combination between L1 and L2 regularization with penalty equals to $\lambda|w|+ \lambda w^2$

Drop-out



LeNet-5

Y. Lecun, L. Bottou, Y. Bengio, and P. Haner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, vol. 86, pp. 2278-2324, Nov 1998

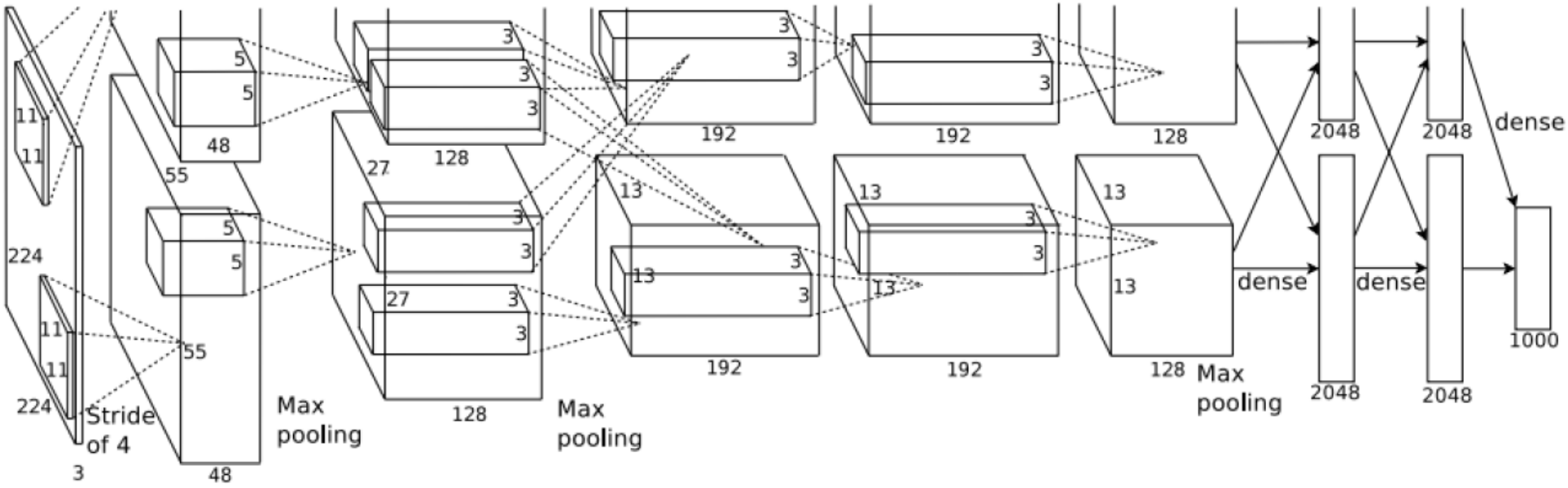


Kolumny pokazują, które warstwy aktywacji z S2 w mapach warstwy C3

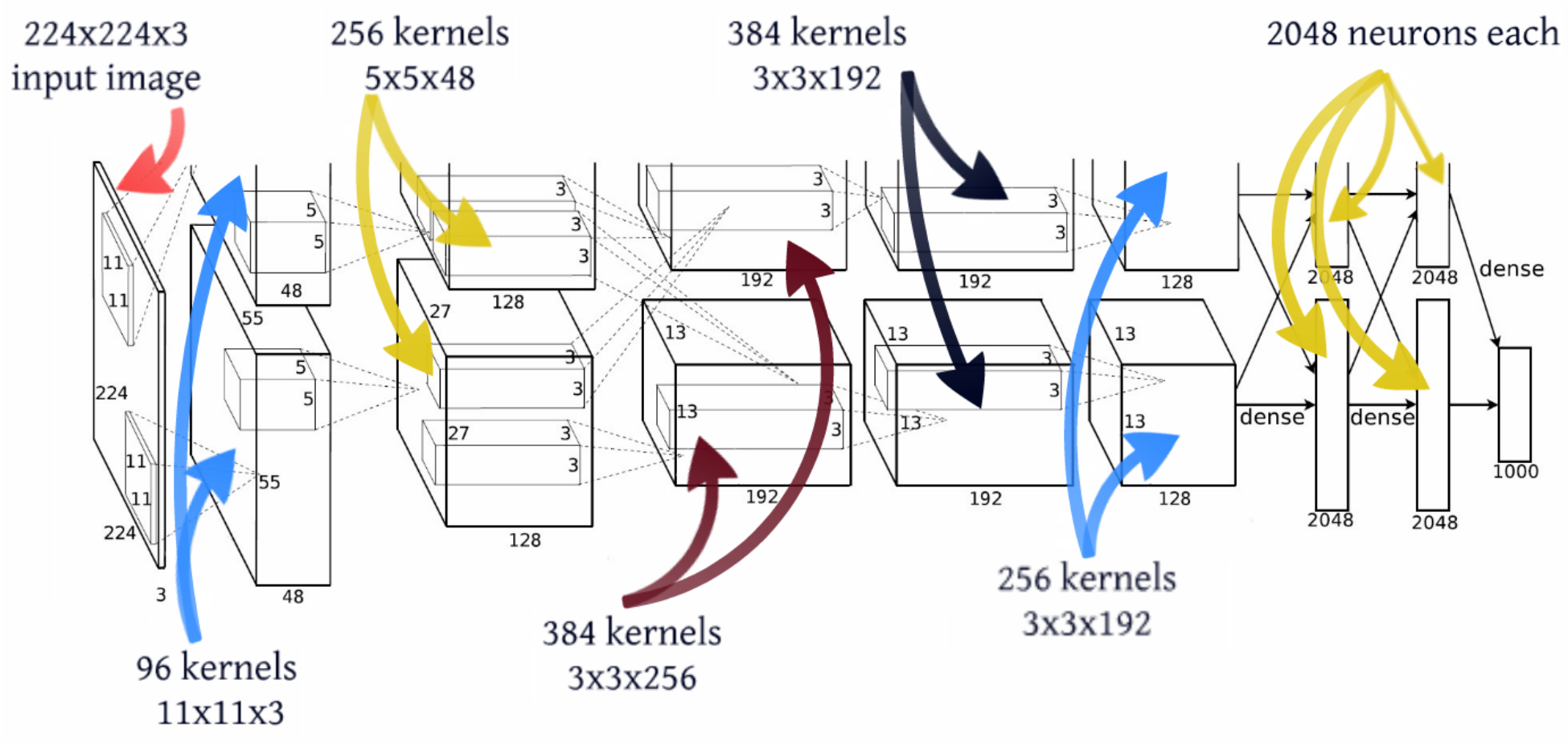
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

AlexNet

A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, (USA), pp. 1097-1105, Curran Associates Inc., 2012



AlexNet – nieco więcej szczegółów

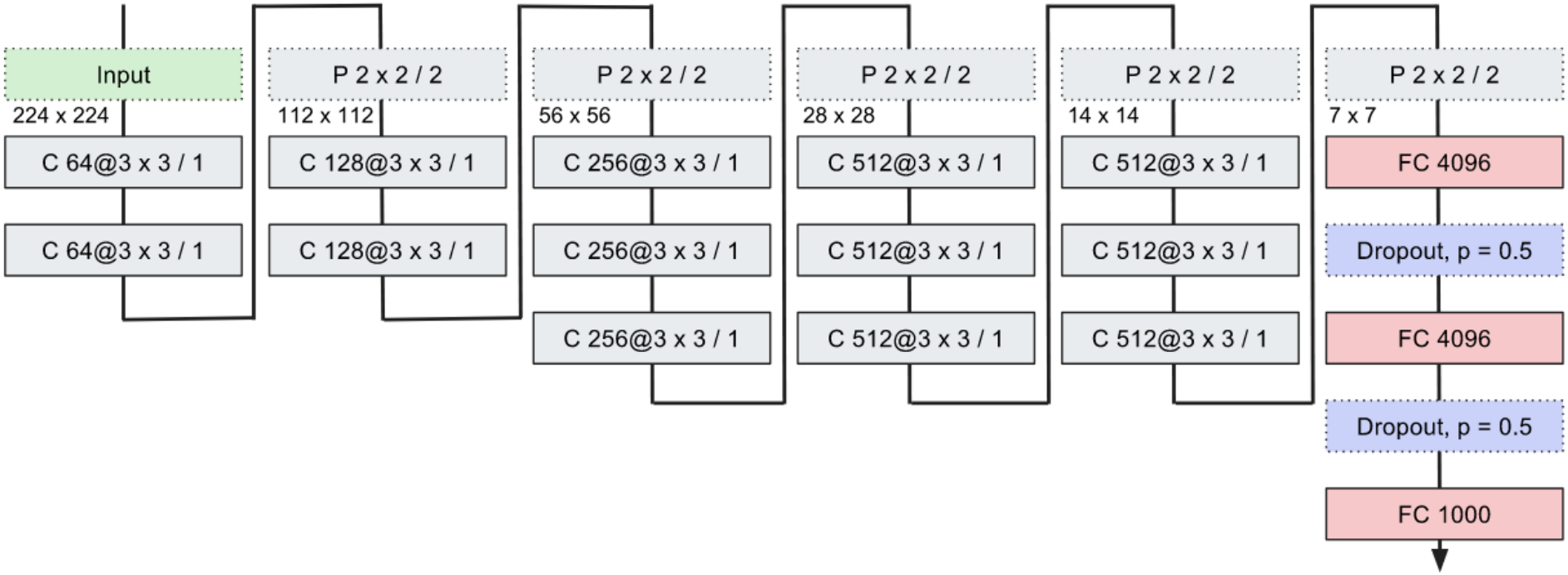


AlexNet – klasyfikacja

AlexNet – wyszukiwanie

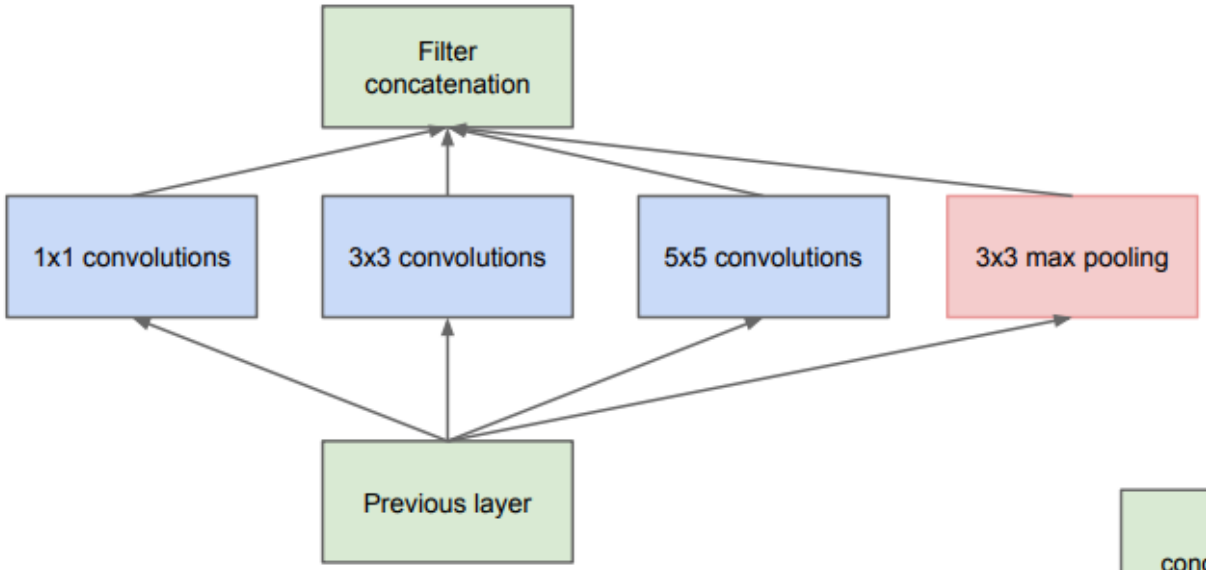
VGG16-D

K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, CoRR, vol. abs/1409.1556, 2014

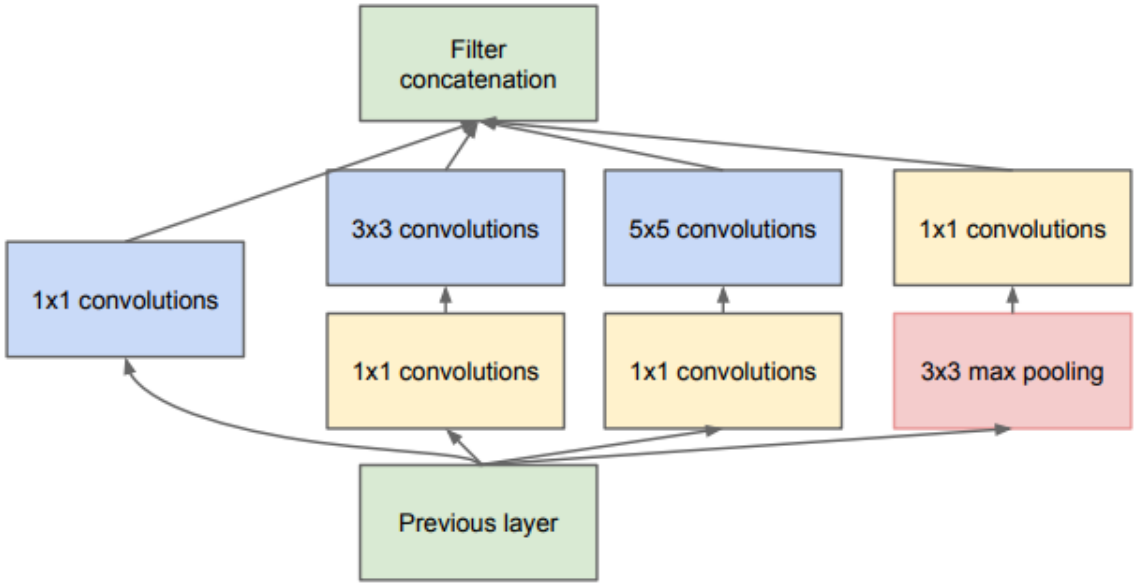


GoogLeNet / Inception

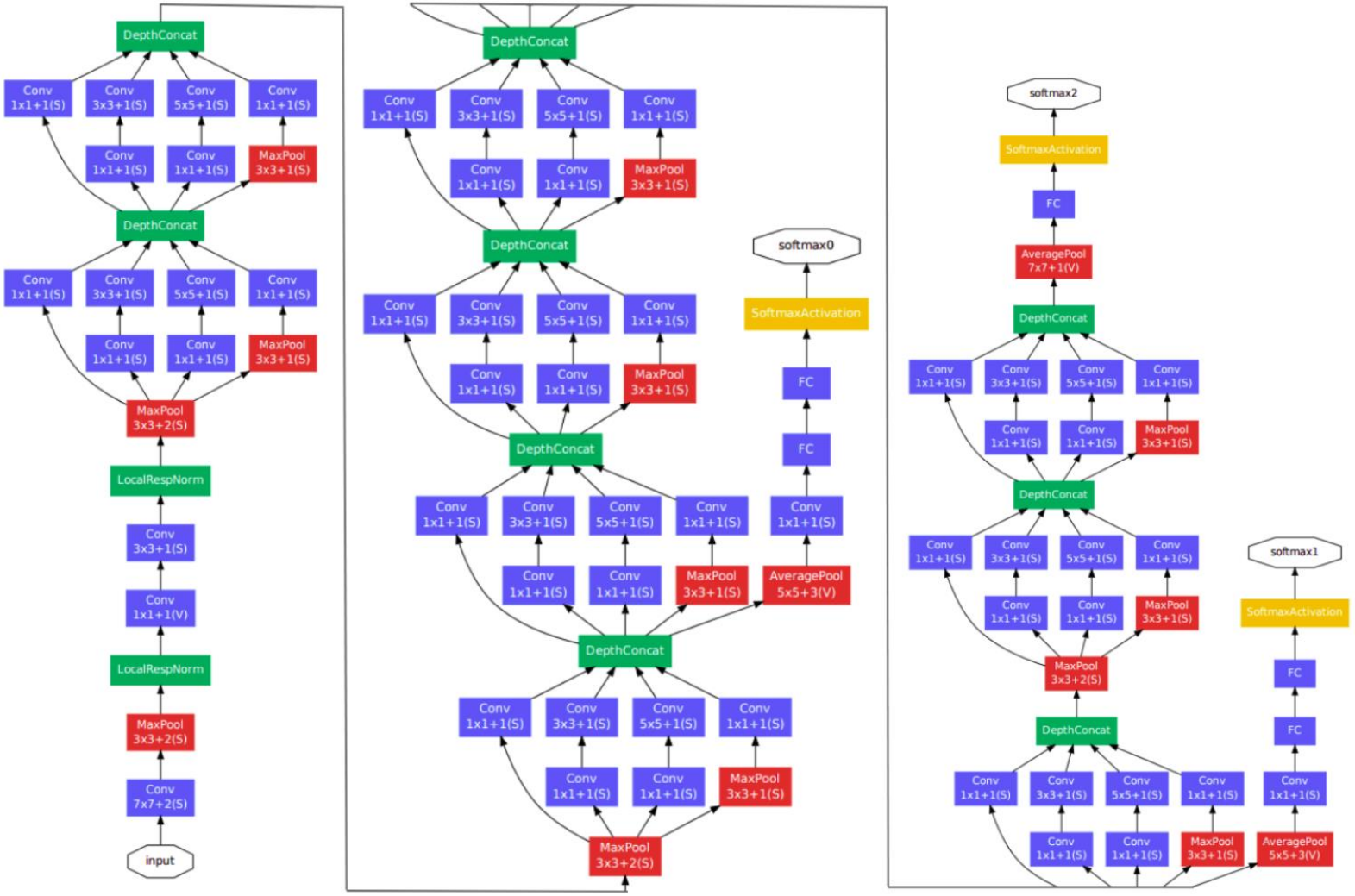
wersja naiwna



Wersja z redukcjami wymiarów

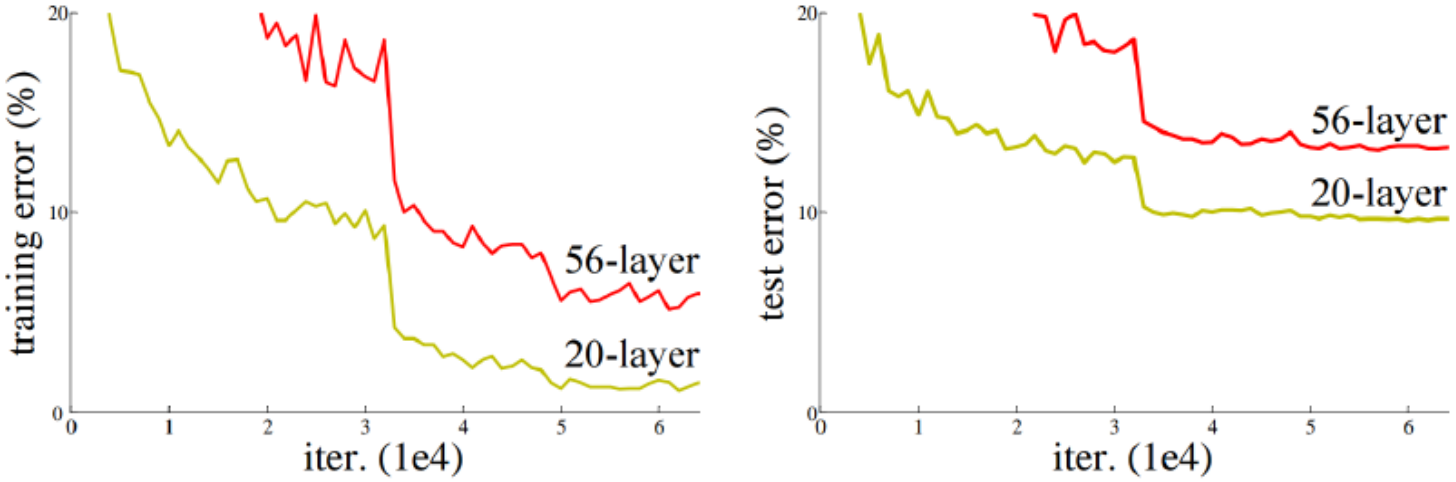


GoogleNet / Inception



Problemy głębokich sieci

Okazuje się, że nie wystarczy po prostu zwiększać liczbę warstw sieci, żeby uzyskać lepszą klasyfikację. Na przeszkodzie staje kilka czynników:
Zjawisko zanikających (lub eksplodujących) gradientów znakomicie utrudnia, o ile nie dewastuje w ogóle, zbieżność procesu uczenia;
Przy głębszych sieciach pojawia się problem degradacji: jakość sieci nasyca się i degraduje szybko przy zwiększaniu liczby warstw.



ResNet

K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, June 2016

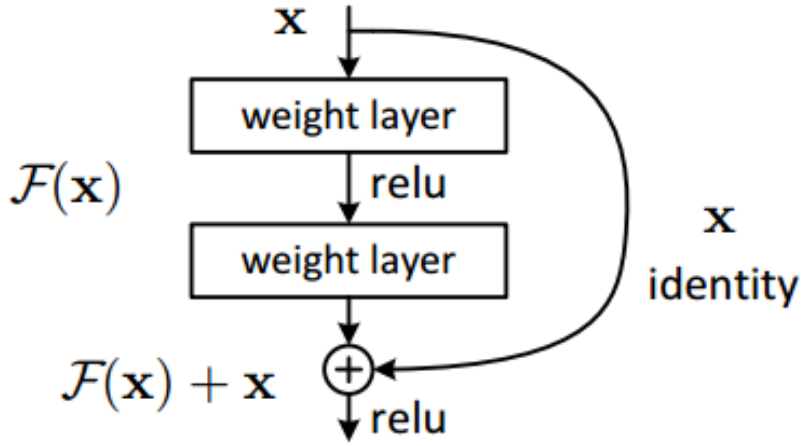
Dajmy warstwie szansę nie robić nic – działać jako zestaw przekazujący wejście na wyjście. Taka warstwa tożsamościowa nie pogorszy wyniku klasyfikacji, nawet jeśli zwiększamy liczbę warstw. Ten sposób okazał się dobrym remedium na problem degradacji sieci.

Formalnie blok rezydualny można zapisać jako:

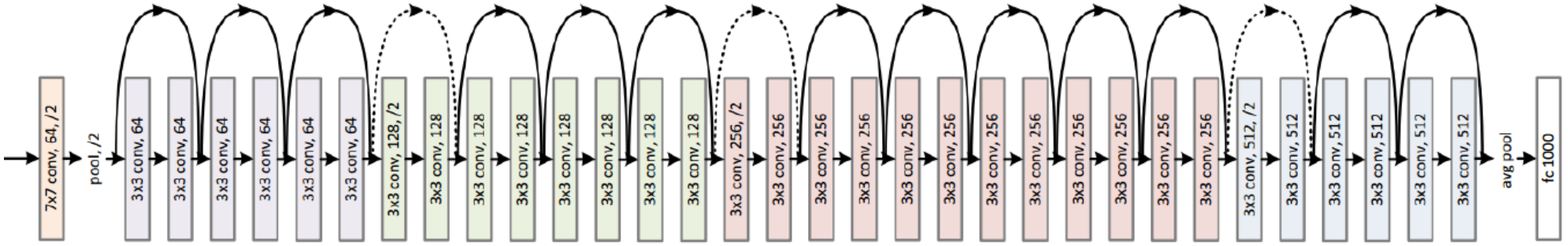
$$y = \mathcal{F}(x, \{W_i\}) + x$$

a w przypadku różnicy wymiarów:

$$y = \mathcal{F}(x, \{W_i\}) + W_s x$$



ResNet-34

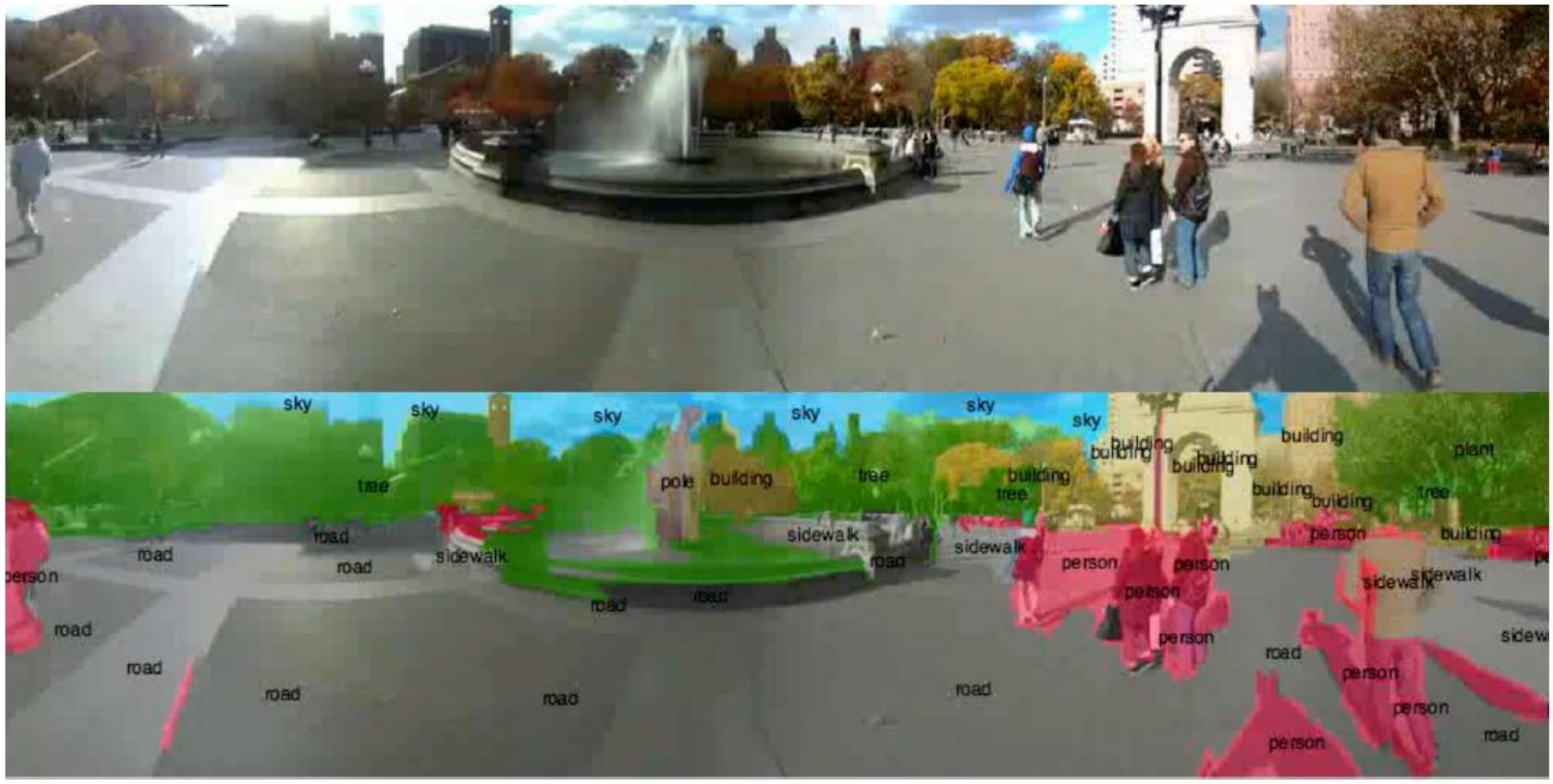

7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
avg pool
fc 1000

Transfer wiedzy

Transfer learning is a technique in machine learning where a model trained on a particular task (source task) is repurposed for another task (target task) in such a way that the representation learned in source task is exploited to improve generalization in target task.
In general, there are three main questions in transfer learning: what to transfer, how to transfer, and when to transfer.
What to transfer asks which part of knowledge can be transferred across tasks or domains. Some knowledge is specific for individual domains or tasks, and some knowledge may be common between different domains. After discovering which knowledge that can be transferred, learning algorithms need to be developed which corresponds to the **how** to transfer issue. **When** to transfer asks in which situations transferring should be done. In some situations, when the source domain and target domain are not related to each other, transfer learning may not be successful.

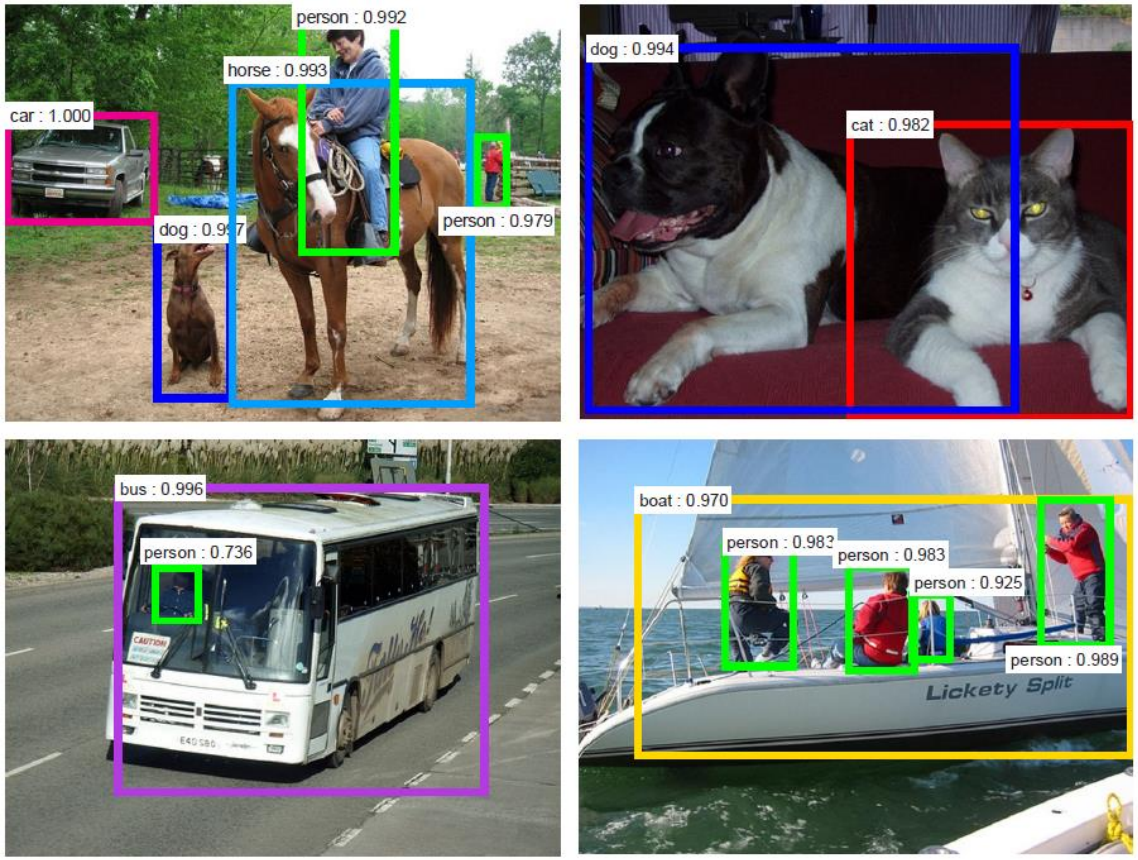
Zastosowania – segmentacja

C. Farabet, C. Couprie, L. Najman and Y. LeCun, *Learning Hierarchical Features for Scene Labeling*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1915-1929, Aug. 2013



Zastosowania – detekcja

S. Ren, K. He, R. Girshick and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, June 2017



Zastosowania – opisywanie obrazów

No errors

A white teddy bear sitting in the grass

A man riding a wave on top of a surfboard

Minor errors

A man in a baseball uniform throwing a ball

A cat sitting on a suitcase on the floor

Somewhat related

A woman is holding a cat in her hand

A woman standing on a beach holding a surfboard

Zastosowania – stylizacja