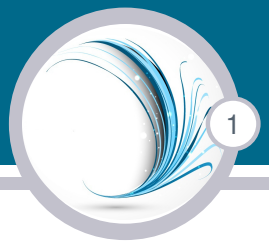
Katedra Sztucznej Inteligencji i Matematyki Stosowanej
WI ZUT Szczecin

Sztuczna inteligencja i maszynowe uczenie w systemach interaktywnych

Joanna Kolodziejczyk
jkolodziejczyk@zut.edu.pl

December 14, 2020

Streszczenie



Sieci konwolucyjne lub splotowe sieci neuronowe
- Inspiracje
- Sieci splotowe podstawy
- Splot
- Warstwa łącząca
  - Przegląd architektur

Sieci konwolucyjne lub splotowe sieci neuronowe

Rozwój

Konwolucyjne sieci neuronowe (CNN) wyłoniły się z badań nad wzrokiem i korą mózgowa, a w rozpoznawaniu obrazu są one stosowane od lat 80.
W kilku ostatnich latach, dzięki wzrostowi mocy obliczeniowej, ilości dostępnych danych oraz poprawy uczenia w sieciach głębokich zaczęły dawać bardzo dobre wyniki.

Przykładowe zastosowania

- klasyfikacja obrazów - rozpoznawanie w czasie rzeczywistym
- rozpoznawanie głosu
- przetwarzanie języka naturalnego (NLP)

Architektura kory wzrokowa

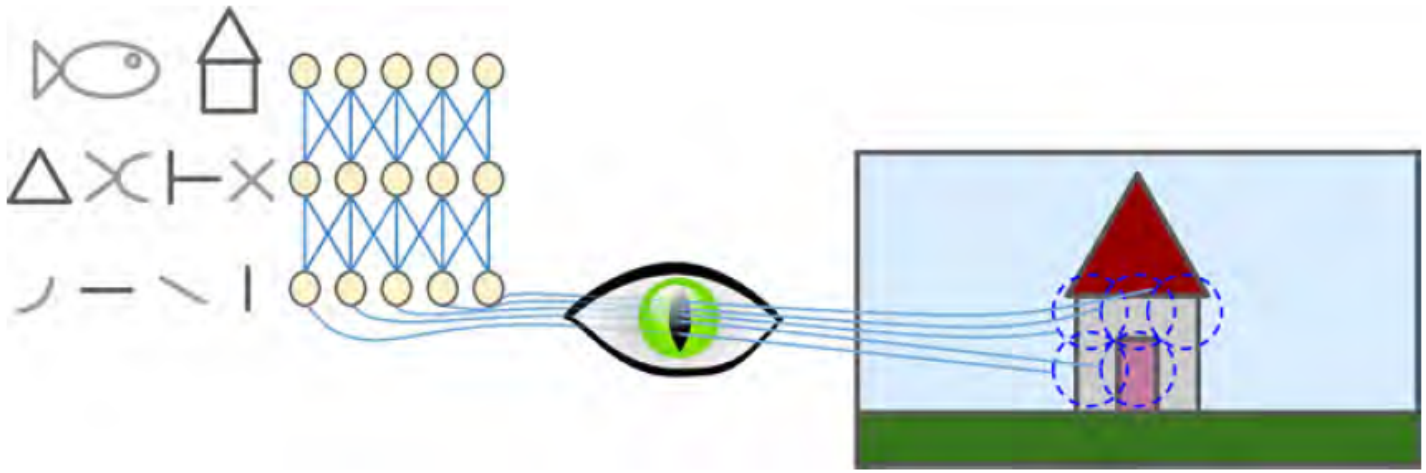
Nobel

David H. Hubel i Torsten Wiesel przeprowadzili serię eksperymentów na kotach w latach 1958 i 1959 (a kilka lat później na małpach), dając istotny wgląd w strukturę kory wzrokowej (autorzy otrzymali za swoją pracę Nagrodę Nobla w dziedzinie fizjologii lub medycyny w 1981 roku).

Wyniki badań

Wiele neuronów w korze wzrokowej ma małe lokalne pole recepcyjne, co oznacza, że reagują one tylko na wizualne bodźce znajdujące się w ograniczonym obszarze pola widzenia. Pola recepcyjne różnych neuronów mogą się na siebie nakładać i razem pokrywają całe pole widzenia.

Architektura kory wzrokowa



- lokalne pola recepcyjne pięciu neuronów są reprezentowane przez przerywane okręgi
- każdy neuron jest połączony tylko z kilkoma neuronami z poprzedniej warstwy

Architektura kory wzrokowa

Wyniki badań

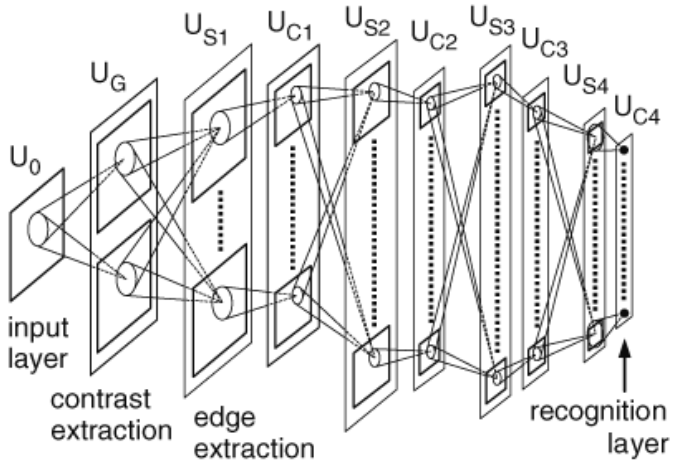
- Niektóre neurony reagują tylko na obrazy linii poziomych, podczas gdy inne tylko na linie o różnych orientacjach (dwa neurony mogą mieć to samo pole widzenia, ale reagują na różne orientacje linii).
- Niektóre neurony mają większe pole recepcyjne i reagują na bardziej złożone wzory, które są kombinacją wzorów niższego rzędu.

Pomysł

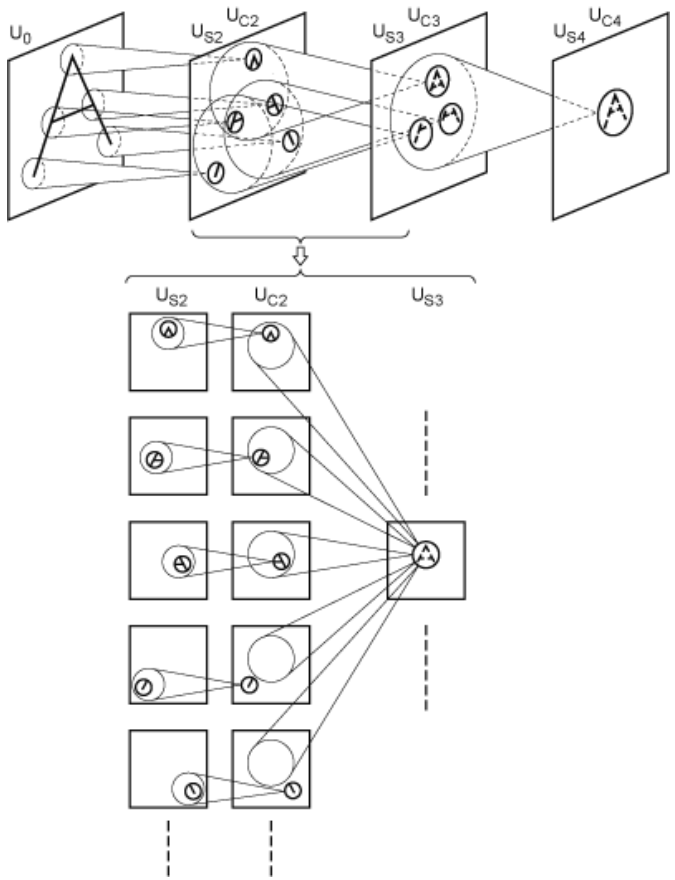
Neurony wyższego rzędu opierają się na wyjściach sąsiednich neuronów niższego rzędu. Ta potężna architektura jest w stanie wykryć wszelkiego rodzaju złożone wzory w dowolnym obszarze pola widzenia.

Neocognitron - 1980 by Fukushima

Badania kory wzrokowej zainspirowały wprowadzony w 1980 r. Neocognitron.



Neocognitron - 1980 by Fukushima



LeNet-5 1998

W 1998 r Yann LeCuna, Léon Bottou, Yoshua Bengio i Patrick Haffner opisali architekturę LeNet-5 do rozpoznawania odręcznie pisanych cyfr.

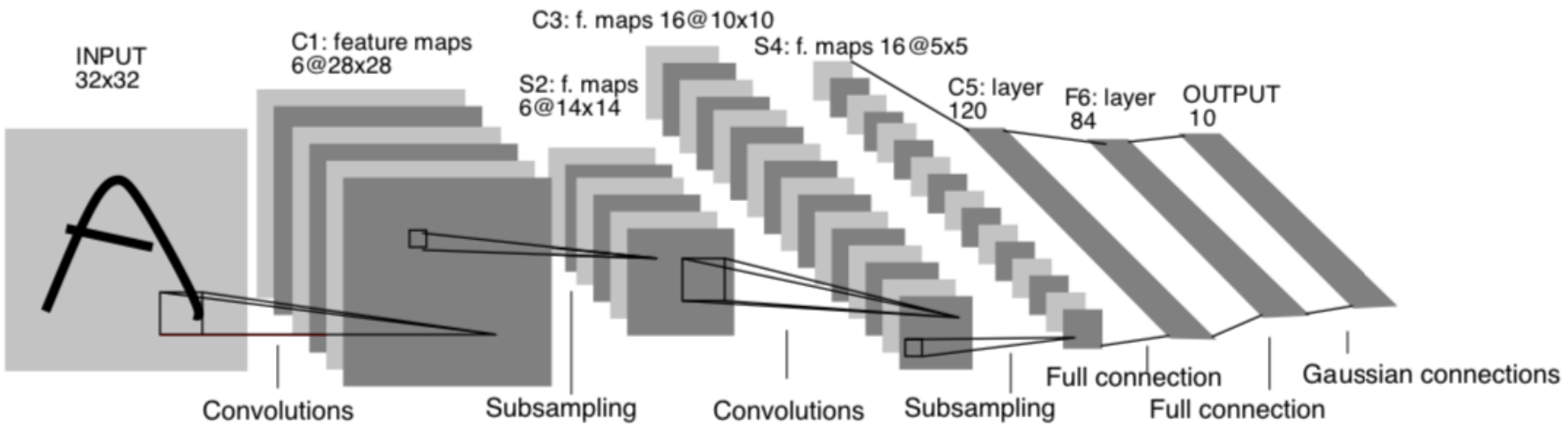


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Motywacja do powstania CNN

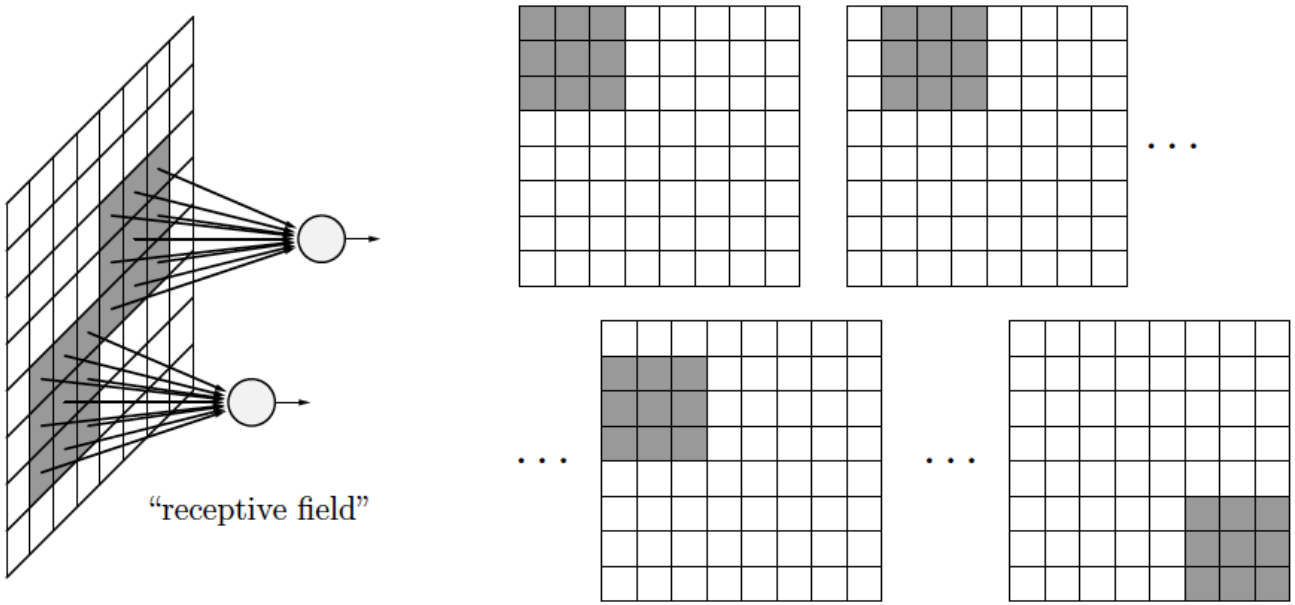
Dlaczego nie używać DNN?

- Działa dobrze w przypadku małych obrazów (np. MNIST).
- Z powodu ogromnej liczby wymaganych parametrów psuje się w przypadku większych obrazów.
- Na przykład obraz 100 x 100 ma 10 000 pikseli, w pierwszej warstwie znajduje się tylko 1000 neuronów (co już poważnie ogranicza ilość informacji przesyłanych do następnej warstwy), to w sumie w sieci jest 10 milionów połączeń (a to tylko pierwsza warstwa).
- Sieci CNN rozwiązują ten problem używając częściowo połączonych warstw.

Założenia CNN

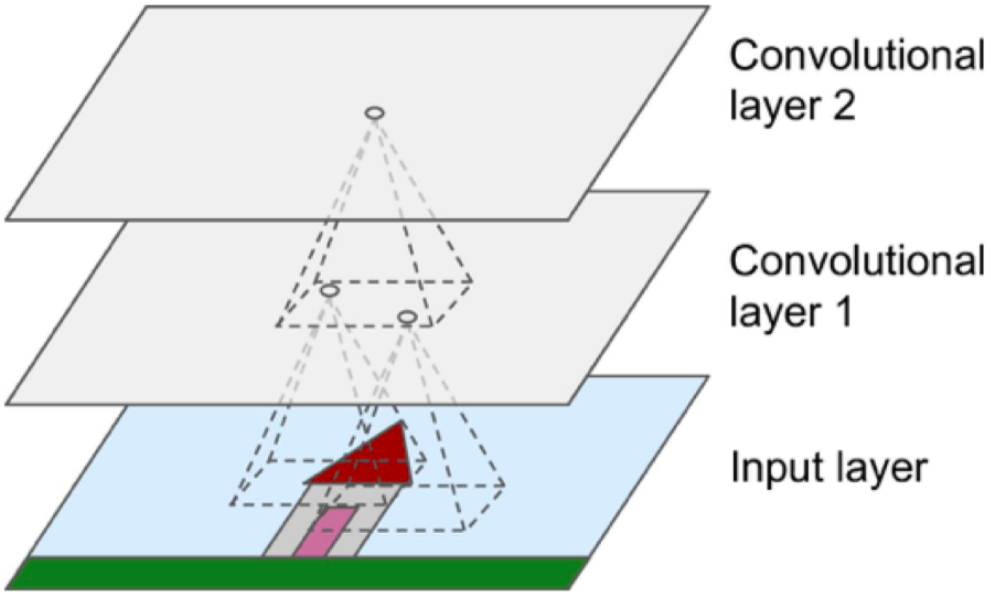
- Różnica: warstwy gęste uczą się parametrów globalnych w przestrzeniach wejściowych, a warstwy splotowe uczą się lokalnych wzorców.
- Wzorce rozpoznawane przez CNN są niezależne od przesunięcia.
- CNN mogą uczyć się przestrzennej hierarchii wzorców.

Pole recepcyjne (odbiorcze), splot



- Każdy neuron (pierwszej) warstwy ukrytej jest połączony z niewielką liczbą neuronów wejściowych, które odnoszą się do sąsiadującego obszaru obrazu wejściowego (po lewej).
- Wagi połączeń są współdzielone, ta sama sieć jest oceniana w różnych miejscach. Pole wejściowe jest "przesuwane" krok po kroku nad całym obrazem (po prawej).
- Splot o małym jądrze.

Pole recepcyjne (odbiorcze), splot



- Pierwsza warstwa nie jest połączona z każdym pojedynczym pikselem w obrazie wejściowym, tylko z fragmentem - polem recepcyjnym
- Z kolei neurony w drugiej warstwie konwolucyjnej są połączone tylko do neuronów umieszczonych w obrębie małego prostokąta w pierwszej warstwie.
- Niższe warstwy rozpoznają prostsze kształty, a wyższe warstwy łączą je w bardziej złożone kształty.

Splot

Splot

W kontekście CNN, splot jest operacją liniową, która polega na mnożeniu zestawu wag z filtra. Mnożenie jest odbywa się podobnie jak w przypadku tradycyjnej sieci neuronowej.

Biorąc pod uwagę, że technika ta została zaprojektowana dla dwuwymiarowego wejścia, mnożenie jest wykonywane pomiędzy macierzą danych wejściowych a dwuwymiarową tablicą wag, zwaną filtrem lub jądrem.

Filtr

Filtr

Filtr jest mniejszy od danych wejściowych. Stosuje się iloczyn skalarny dwóch macierzy (fragment wejścia i filtr – wagi).

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

Wynik jest zawsze wartością skalarną (stąd nazwa iloczynu).

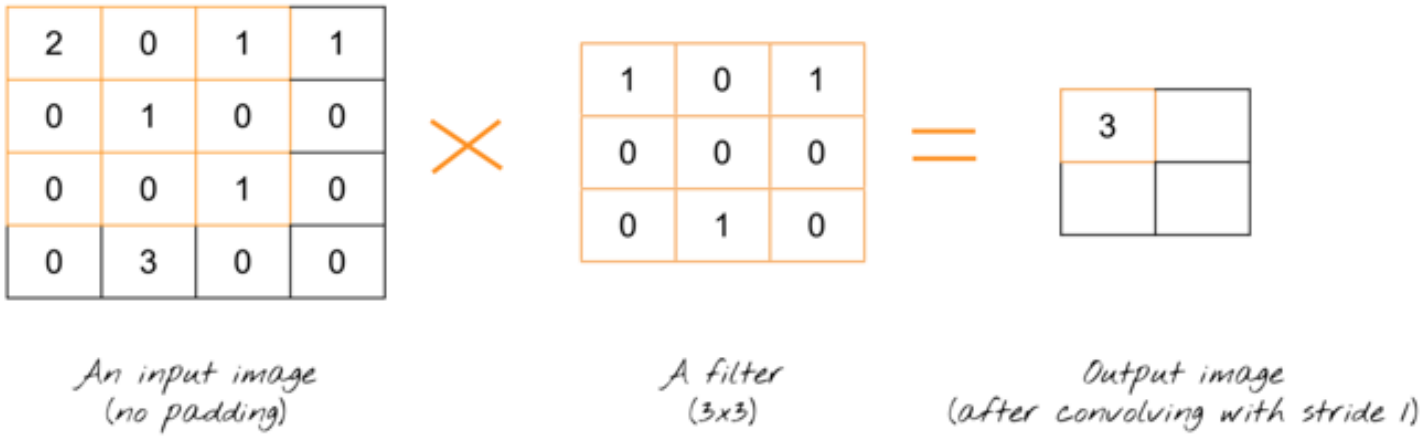
Użycie filtra mniejszego niż wejście jest celowe, ponieważ pozwala na wielokrotne pomnożenie tego samego filtra (zestawu wag) przez wejścia w różnych punktach. Filtr jest stosowany systematycznie do każdej nakładającej się części lub fragmentu danych wejściowych o wielkości filtra, od lewej do prawej, od góry do dołu.

Przykład filtra

-1	-1	-1
-1	8	-1
-1	-1	-1

https://www.cs.columbia.edu/education/courses/course/COMSW4995-7/26050/

Splot 2D



Mathematically, it's (2 * 1) + (0 * 0) + (1 * 1) + (0 * 0) + (1 * 0) + (0 * 0) + (0 * 0) + (0 * 1) + (1 * 0) = 3

Splot 2D - przykład

2	0	1	1
0	1	0	0
0	0	1	0
0	3	0	0

An input image
(no padding)

1	0	1
0	0	0
0	1	0

A filter
(3x3)

3	2

Output image
(after convolving with stride 1)

Splot 2D - przykład

2	0	1	1
0	1	0	0
0	0	1	0
0	3	0	0

An input image
(no padding)

1	0	1
0	0	0
0	1	0

A filter
(3x3)

3	2
3	

Output image
(after convolving with stride 1)

Splot 2D - przykład

2	0	1	1
0	1	0	0
0	0	1	0
0	3	0	0

An input image
(no padding)

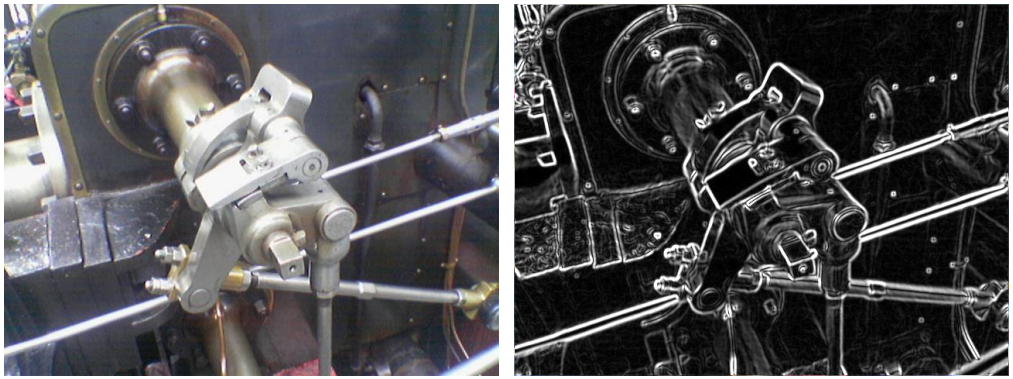
1	0	1
0	0	0
0	1	0

A filter
(3x3)

3	2
3	1

Output image
(after convolving with stride 1)

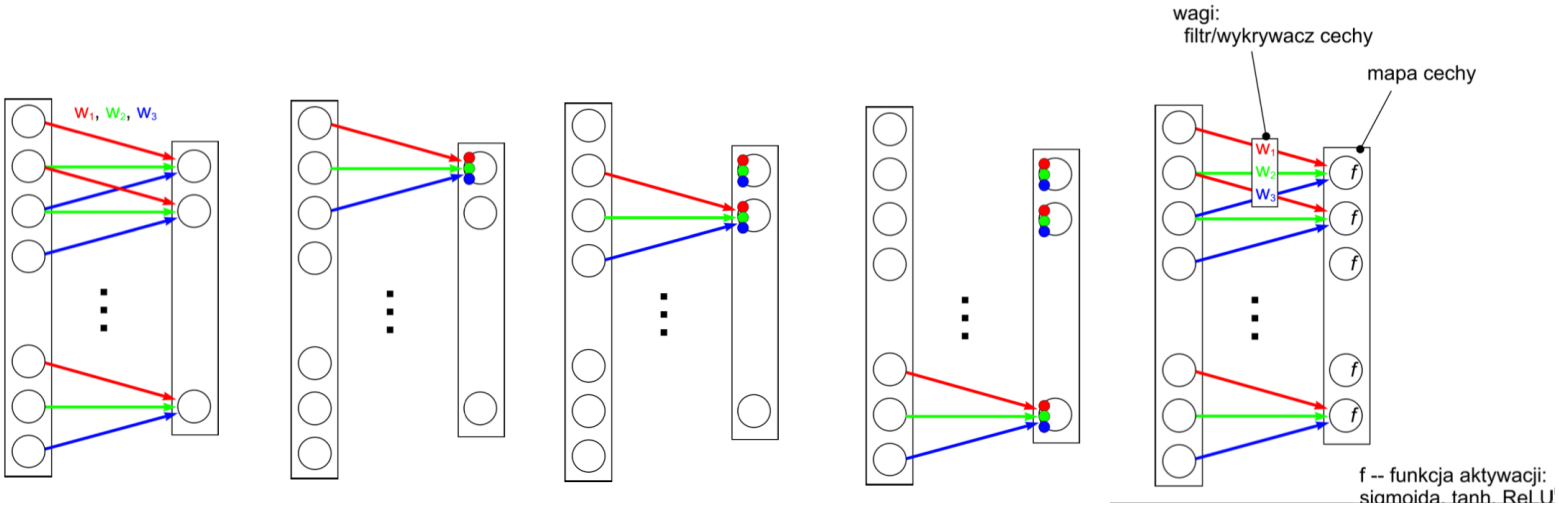
Filtr Sobela - wykrywanie krawędzi



https://en.wikipedia.org/wiki/Sobel_operator

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Współdzielenie wag

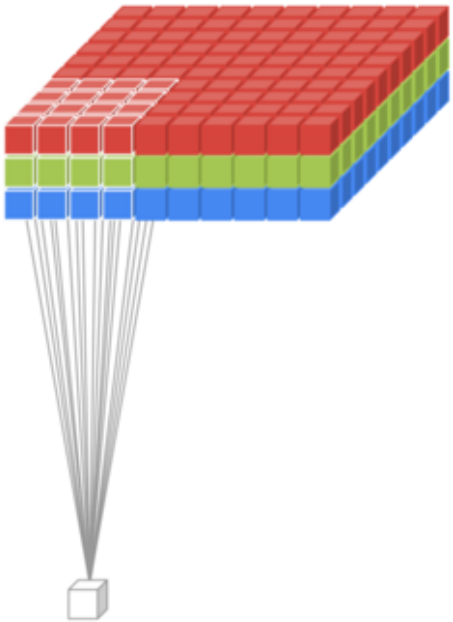


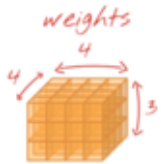
http://www-users.mat.umk.pl/ rudy/wsn/wyk/wsn-wyklad-17-convnets1.pdf

Współdzielenie wag

Fakt, że wszystkie neurony w mapie cech mają te same parametry, drastycznie zmniejsza liczbę parametrów modelu, ale co najważniejsze, oznacza, że gdy CNN nauczy się rozpoznania wzorca w jednym miejscu, może to zrobić w każdym innym polu. Natomiast, gdy zwykły DNN nauczył się rozpoznawać wzór w jednej lokalizacji, może go rozpoznać tylko w tej konkretnej lokalizacji.

Splot 3D (RGB) - przykład



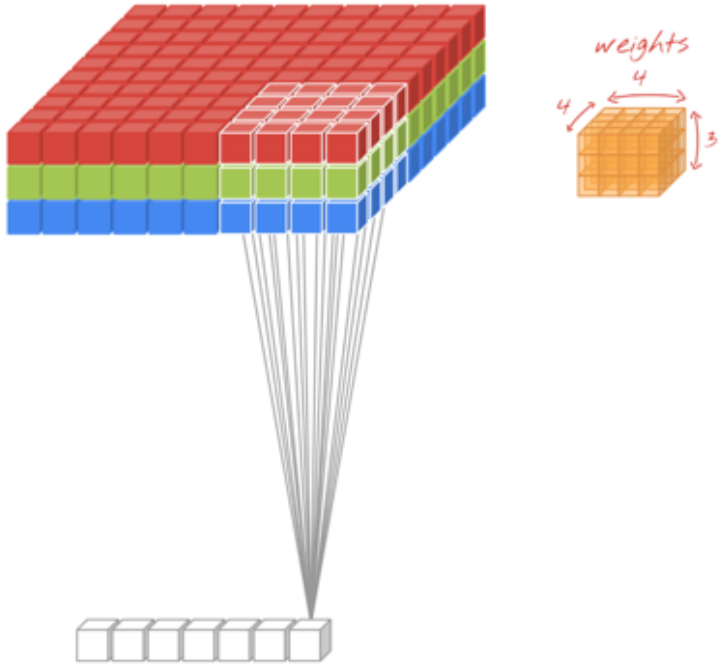


Applied in the same way as 2d (sum of weight * pixel value as they slide across the image).

https://twitter.com/martin_gorner

Splot 3D (RGB) - przykład

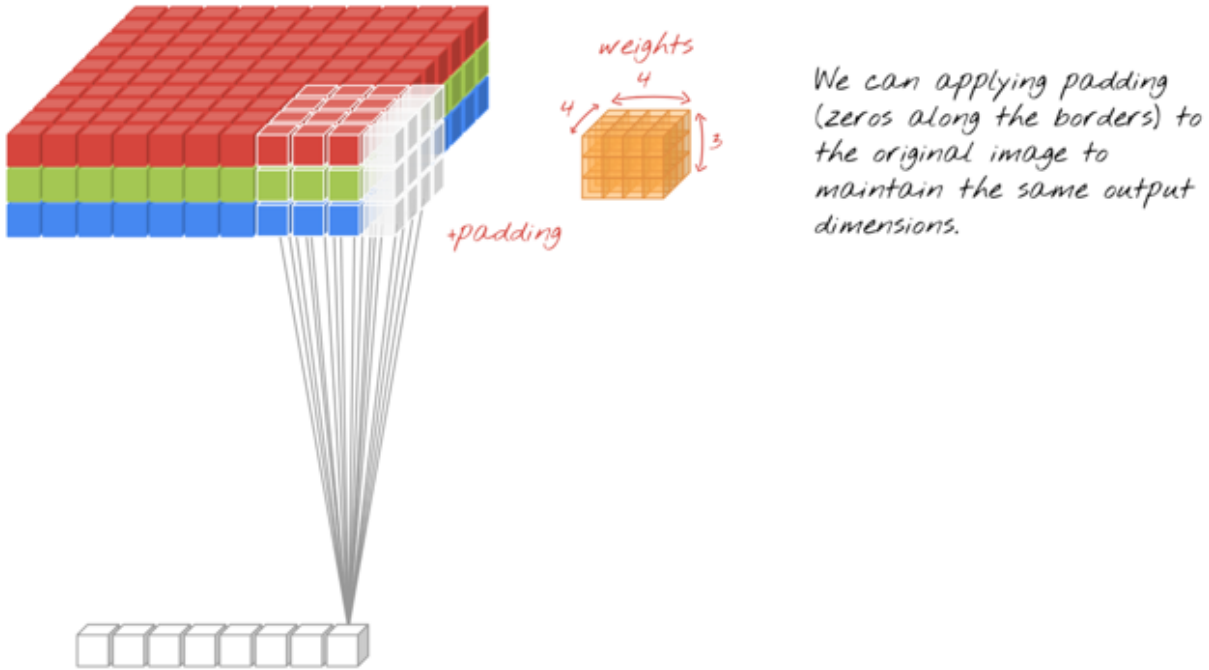
Normalnie, szerokość wyjścia staje się mniejsza, podobnie jak wielkość wyjścia w przypadku 2D.



https://twitter.com/martin_gorner

Splot 3D (RGB) - przykład

Aby utrzymać obraz wyjściowy tej samej szerokości i wysokości bez zmniejszania rozmiaru filtra, można dodać do oryginalnego obrazu ramkę z zerami i wykonać splot.



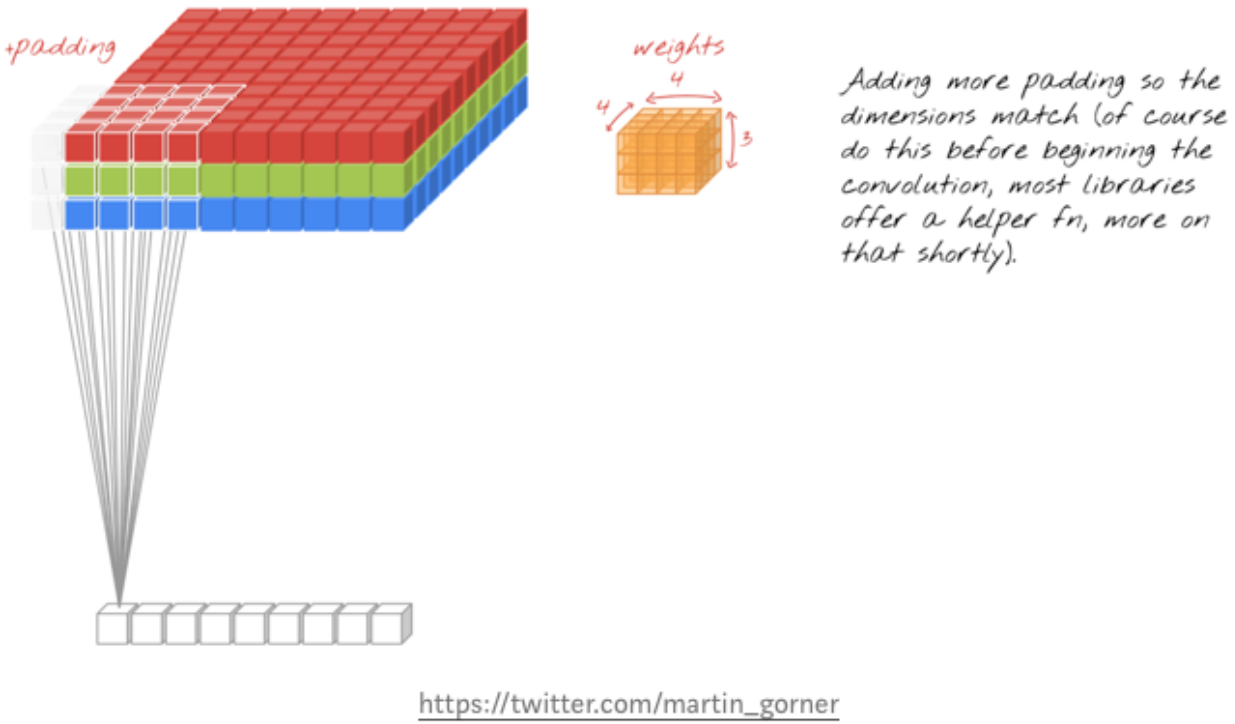
https://twitter.com/martin_gorner

Splot 3D (RGB) - przykład

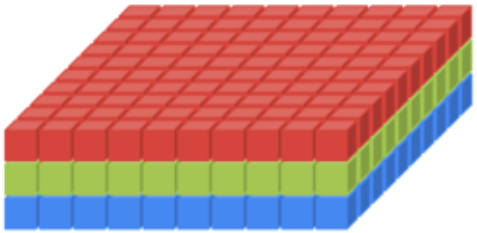
Proste równanie może pomóc w ustaleniu kształtu wyjścia:

$$n + 2p - f + 1$$

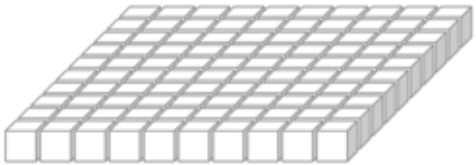
Gdzie n jest wielkością wejścia, p jest wielkością dopełnienia, a f jest wielkością filtra.



Splot 3D (RGB) - przykład



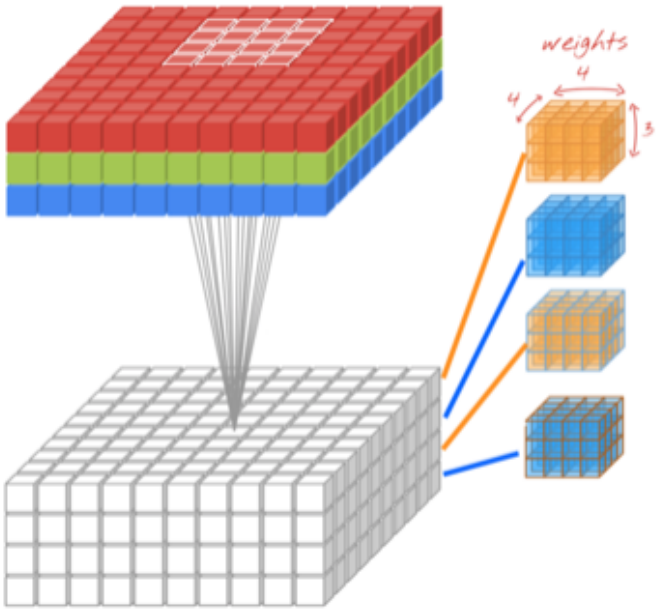
Applying the convolution over the rest of the input image.



https://twitter.com/martin_gorner

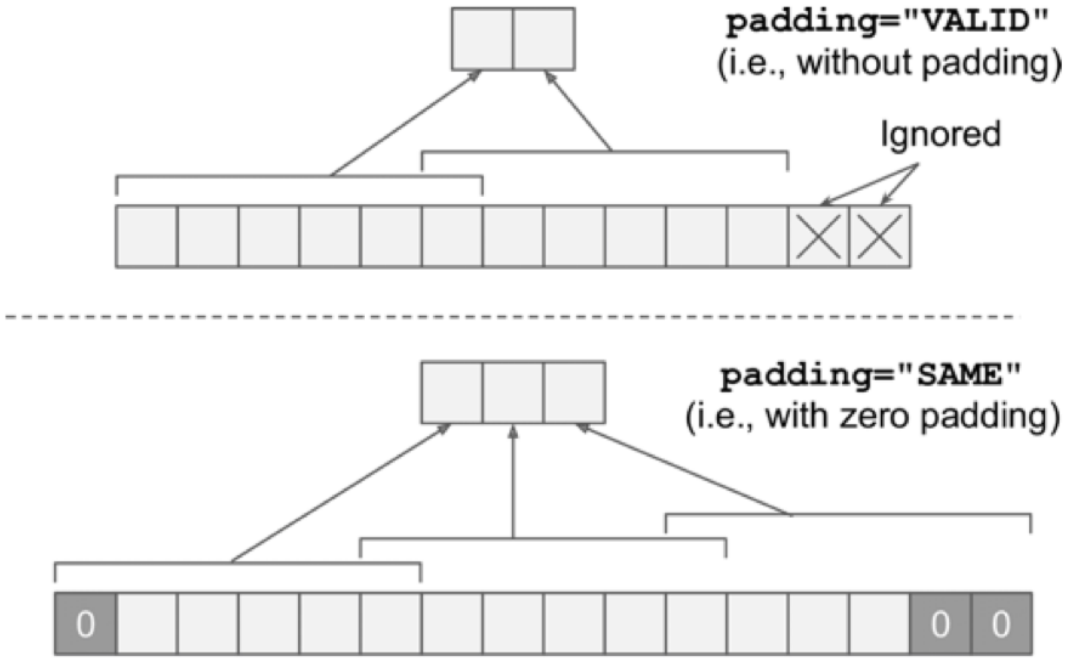
Splot 3D (RGB) - przykład

W miarę dodawania kolejnych filtrów zwiększa się głębię obrazu wyjściowego. Jeśli głębokość dla obrazu wyjściowego jest 4, to zostały użyte 4 filtry. Każda warstwa odpowiada jednemu filtrowi i uczy się jednego zestawu wag. Filtr nie zmienia się gdy przesuwa się po obrazie.



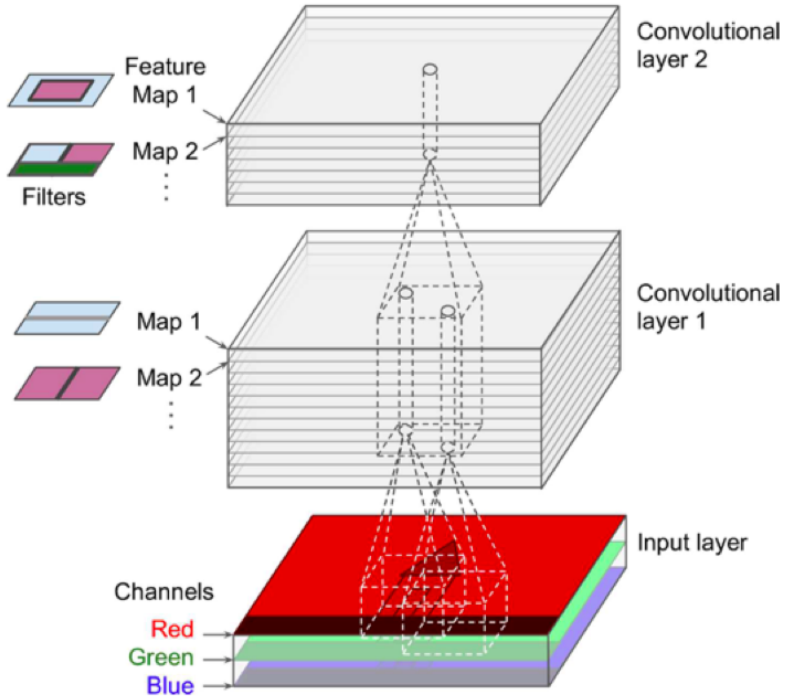
https://twitter.com/martin_gorner

Dodawanie marginesu (padding)



- „Valid” prezentuje brak uzupełniania i pomijanie niektórych wejść.
- „Same” uzupełnienie - dodanie marginesu (wielkość dopasowana do kroku)

Układanie map cech w stosy



- Na jednej mapie cech wszystkie neurony mają te same parametry (wagi i biasy), ale różne mapy cech mają różne parametry.
- Pole recepcyjne neuronu rozciąga się na wszystkie mapy cech warstw poprzednich.
- Warstwa splotowa nakłada na wejścia wiele filtrów jednocześnie, dzięki temu jest w stanie wykryć wiele cech w każdym „obszarze” wejścia.

Uczenie warstwy konwolucyjnej

- Wsteczna propagacja błędu: strategie minimalizacji takie jak przy innych sieciach
- W sieciach splotowych podczas kroku wstecznej propagacji błąd jest również propagowany za pomocą operacji splotu
- `https://becominghuman.ai/back-propagation-in-convolutional-neural-networks-intuition-`

Warstwa łącząca

Pooling layer

Ich celem jest próbkowanie (tzn. zmniejszanie) obrazu wejściowego, tak by zredukować liczbę obliczeń, zużycia pamięci i liczby parametrów i tym samym ograniczyć ryzyko przetrenowania. Powoduje, że sieć neuronowa toleruje nieco przesunięcie obrazu.

- Każdy neuron w warstwie łączącej jest połączony z wyjściem ograniczonej liczby neuronów w warstwie poprzedniej znajdujących się w obrębie małego prostokątnego pola recepcyjnego.
- Trzeba zdefiniować jego rozmiar, skok i rodzaj marginesu.
- Neuron łączący nie ma biasów, łączy wejścia wykorzystując funkcję agregacji taką jak maksimum lub średnia.

Max-pooling

2	0	1	1
0	1	0	0
0	0	1	0
0	3	0	0

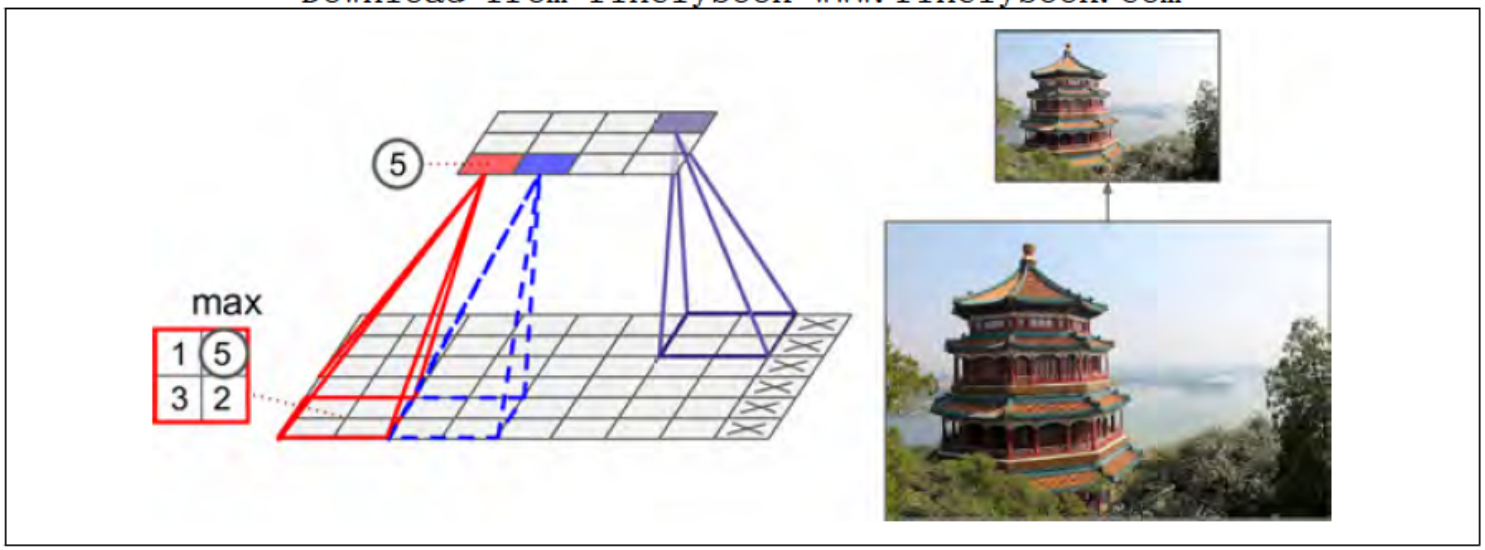
Max pooling with a 2x2 window and stride 2

2	1
3	1

http://cs231n.github.io/convolutional-networks/

Max-pooling

Jądro łączące ma rozmiar 2 x 2, krok 2.



Działanie tej warstwy jest nazywane downsamplingiem.

Average-pooling

http://cs231n.github.io/convolutional-networks/

Uczenie warstwy łączącej

Warstwa typu Max

- W warstwie łączącej, propagacja w przód powoduje zmniejszenie bloku do jednej wartości tzw, „jednostki zwycięskiej” – operacja max.
- Propagacja wstecz warstwy łączącej typu max: błąd propagowany tylko do zwycięzcy.

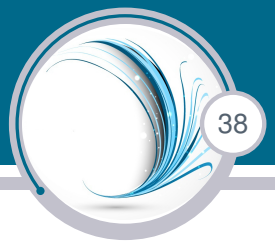
Warstwa typu Avg

- W warstwie łączącej, propagacja w przód powoduje zmniejszenie bloku do średniej wartości wszystkich neuronów.
- Propagacja wsteczna warstwy łączącej typu mean: błąd propagowany po równo do wszystkich w poprzedniej warstwie

Ostatni typ — warstwa w pełni połączona

Neurony w tej warstwie mają połączenia do wszystkich neuronów z poprzedniej warstwy, tak jak w jednokierunkowych sieciach neuronowych.
Aktywacje neuronów w tej warstwie są obliczane za pomocą iloczynu macierzowego, po którym następuje przesunięcie przez bias.

Przykładowa architektura



Do budowy CNN wykorzystujemy trzy główne rodzaje warstw:

1. warstwa konwolucyjna (Convolutional Layer),
2. warstwa łącząca (Pooling Layer)
3. warstwa w pełni połączona (Fully-Connected Layer).

Warstwy układa się w stosy i tak powstaje pełna architektura CNN.

Przykładowa architektura

Przykład architektury CNN dla zbioru CIFAR-10:[1]

INPUT –> CONV –> RELU –> POOL –> FC

INPUT [32x32x3] wprowadza wartości pikseli obrazu, w tym przypadku obraz o szerokości 32, wysokości 32, oraz z trzema kanałami kolorów R,G,B.

[1] `https://www.cs.toronto.edu/~kriz/cifar.html`

Przykładowa architektura

Przykład architektury CNN dla zbioru CIFAR-10:[2]

INPUT –> **CONV** –> RELU –> POOL –> FC

Warstwa CONV obliczy wyjście neuronów, które są podłączone do lokalnych obszarów na wejściu. Każdy neuron obliczy iloczyn skalarny pomiędzy jego wagą a małym obszarem, do którego jest podłączony w warstwie wejściowej. W efekcie może powstać warstwa o wielkości [32x32x12], jeśli użyte zostanie 12 filtrów.

[2] https://www.cs.toronto.edu/~kriz/cifar.html

Przykładowa architektura

Przykład architektury CNN dla zbioru CIFAR-10:[3]

INPUT –> CONV –> **RELU** –> POOL –> FC

Warstwa RELU zastosuje funkcję aktywacji np. ReLU, cyli max(0,x).
Wielkość macierzy nie zmienia się: [32x32x12].

[3] https://www.cs.toronto.edu/~kriz/cifar.html

Przykładowa architektura

Przykład architektury CNN dla zbioru CIFAR-10:[4]

INPUT –> CONV –> RELU –> **POOL** –> FC

Warstwa POOL przeprowadzi operację próbkowania (szerokość, wysokość), w wyniku czego powstanie warstwa o rozmiarach [16x16x12].

[4] https://www.cs.toronto.edu/~kriz/cifar.html

Przykładowa architektura

Przykład architektury CNN dla zbioru CIFAR-10:[5]

INPUT –> CONV –> RELU –> POOL–> **FC**

Warstwa FC (tzn. w pełni połączona) obliczy wynik, czyli klasę. Wynikiem będzie wektor [1x1x10], gdzie każda z 10 liczb odpowiada klasie jednej z 10 kategorii CIFAR-10.

[5] `https://www.cs.toronto.edu/~kriz/cifar.html`

Przykładowa architektura – podsumowanie

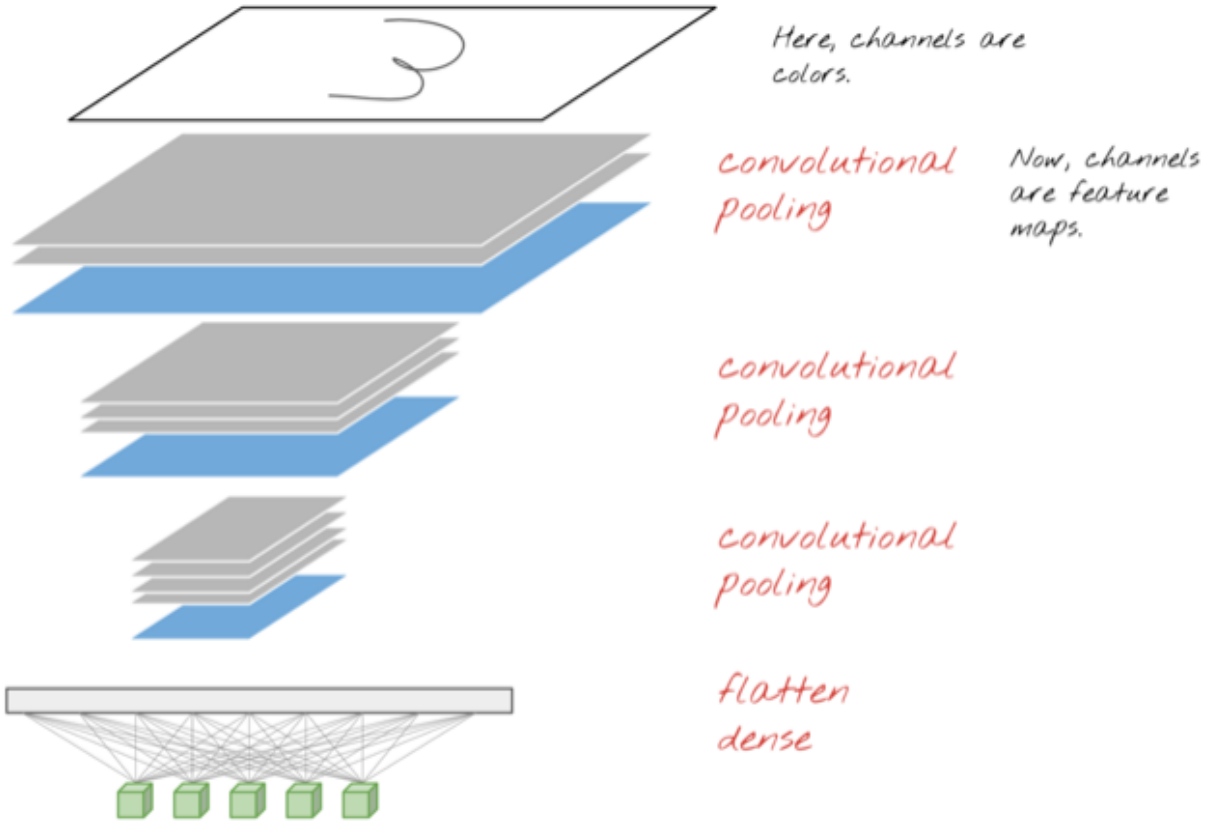
Przykład architektury CNN dla zbioru CIFAR-10:[6]

INPUT –> CONV –> RELU –> POOL–> FC

- CNN przekształca oryginalny obraz warstwa po warstwie z oryginalnych wartości - pikseli do końcowej klasy.
- Warstwy CONV i FC dokonują transformacji stosując parametry sieci (wag i biasów).
- Warstwy RELU i POOL będą implementować stałą funkcję.
- Parametry w warstwach CONV i FC zostaną wytrenowane metodą gradientową, tak aby wyniki klasy, które obliczy CNN, były zgodne z etykietami w zestawie treningowym dla każdego obrazu.

[6] https://www.cs.toronto.edu/~kriz/cifar.html

Typowa architektura



LeNet 1990

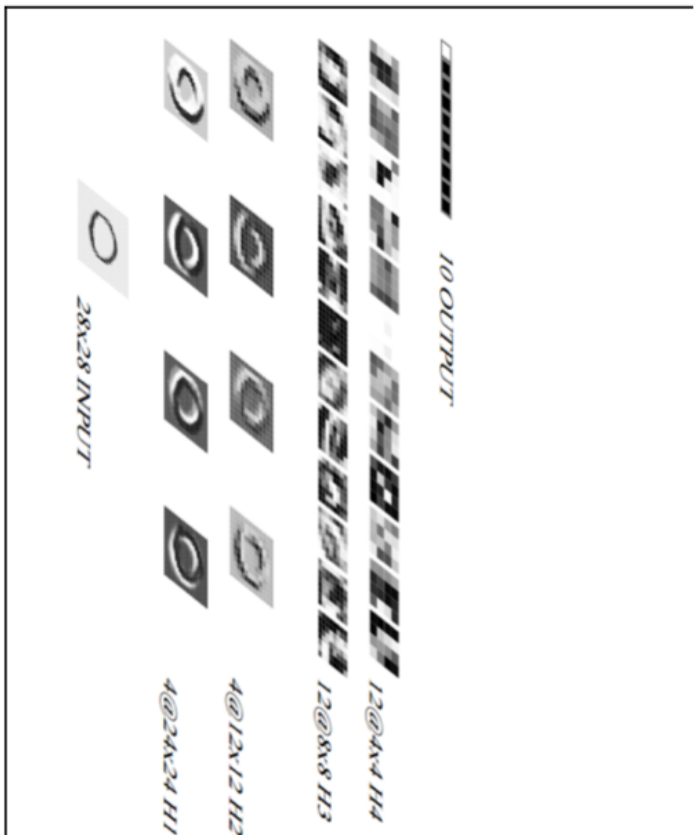
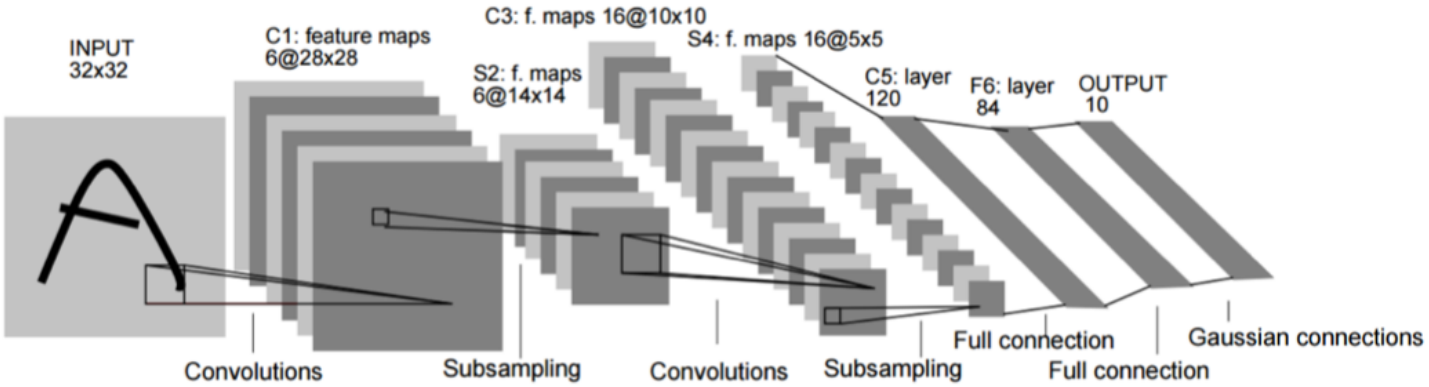


Figure 4: Network Architecture with 5 layers of fully-adaptive connections.

	1	2	3	4	5	6	7	8	9	10	11	12
1	X	X	X		X	X						
2		X	X	X	X	X						
3							X	X	X		X	X
4								X	X	X	X	X

Table 1: Connections between H2 and H3.

LeNet5 1998



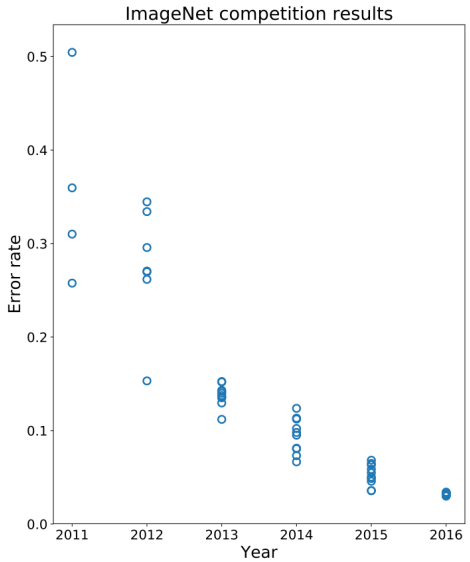
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

TABLE I

EACH COLUMN INDICATES WHICH FEATURE MAP IN S2 ARE COMBINED BY THE UNITS IN A PARTICULAR FEATURE MAP OF C3.

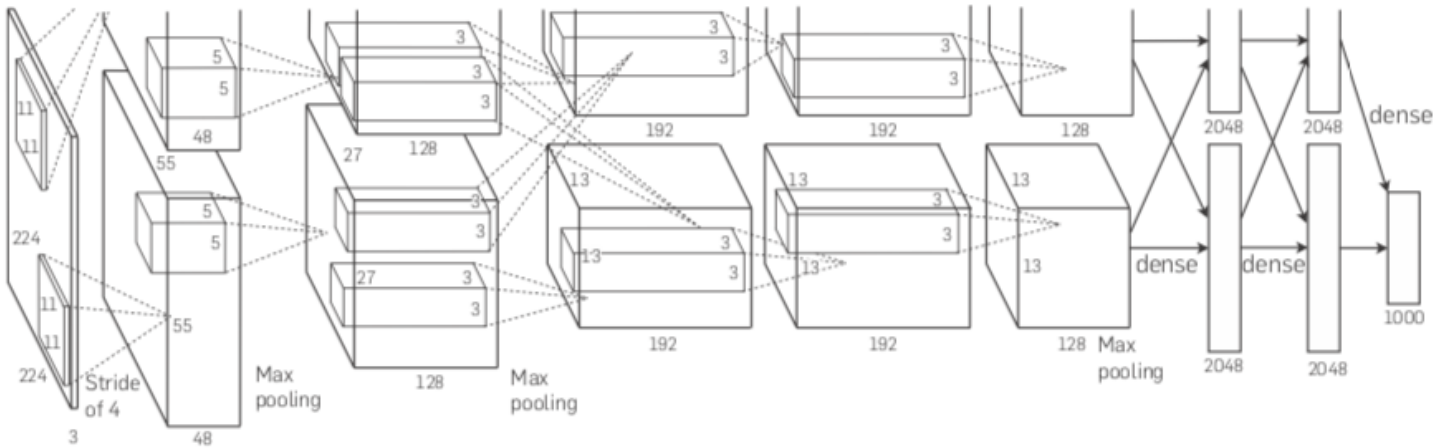
ILSVRC

- Large Scale Visual Recognition Challenge
- Przykładowe zadania: 150 tyś obrazów ręcznie etykietowanych do 1000 kategorii.
- Cel: wykrycie 5 klas obiektów prezentowanych na obrazie
- `http://www.image-net.org/challenges/LSVRC/`



AlexNet 2012

Figure 2. An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 290,400–186,624–64,896–64,896–43,264–4096–4096–1000.



Źródła

1. http://www.scholarpedia.org/article/Neocognitron
2. Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.
3. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 1 edition, 2017, O'Reilly Media,
4. https://towardsdatascience.com/a-beginners-guide-to-convolutional-neural-networks-cnns-14649dbddce8
5. https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/
6. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (June 2017), 84–90. DOI:https://doi.org/10.1145/3065386
7. https://medium.com/@ngocson2vn/a-gentle-explanation-of-backpropagation-in-convolutional-neural-network-cnn-1a70abff508b