

CNN

Convolutional Neural

Networks

CNN

- **Convolutional Neural Networks (CNN, ConvNet)**
wariant MLP inspirowany biologicznie, gdzie mnożenie macierzy wag i sygnału wejściowego zastąpione jest operacją splotu
- rzadka reprezentacja, współdzielone wagi, pooling
- zdolne do generalizacji sygnału posiadającego relacje przestrzenne, odporne na przestrzenne transformacje sygnału (skalowanie, obrót, przesunięcie):
 - 1D sygnały czasowe
 - 2D obrazy
 - 3D fMRI, video, obrazy RGB
 - sygnały wielokanałowe

Splot

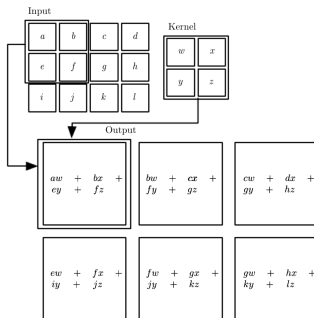
$$(x \star w)(t) = \int x(a)w(t - a)da$$

- wynikiem jest funkcja, np. uśredniona wartość $x(x)$ względem wszystkich pozycji $w(a)$ jeśli w spełnia wymagania gęstości prawdopodobieństwa
- W terminologii CNN:
 x - sygnał wejściowy
 w - kernel (filtr), wagi połączeń neuronu
wartości wyjściowe tworzą mapę cech (*feature map*)

$$s(i) = \sum_m x(m)w(i - m)$$

- w sieciach CNN w niezerowe tylko w organicznym obszarze (pole recepcyjne)

Splot 2D



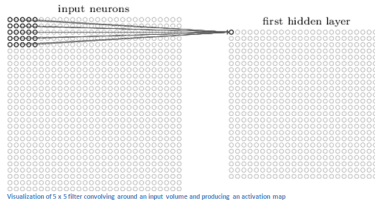
$$S_{ij} = \sum_l \sum_m W_{lm} X_{i+l, j+m}$$

Filtry graficzne 2D - przykłady

Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

Pola recepcyjne i mapy cech

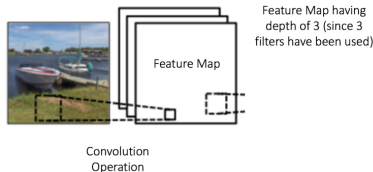


- **Pole recepcyjne** - obszar „widziany” przez neuron (rozmiar filtra)
- wyjście neuronu: splot sygnału i liniowego filtra, ewentualny wyraz wolny (bias) i nieliniowa funkcja wyjściowa (ReLU, tanh), generują **mapę cech** (*feature map*)

$$s_{ij} = \sigma \left(b + \sum_l \sum_m w_{lm} x_{i+l, j+m} \right)$$

Pola recepcyjne i mapy cech

- Warstwa zawierająca N neuronów (filtrów) tworzy N map (objętość, tensor n wymiarowy)

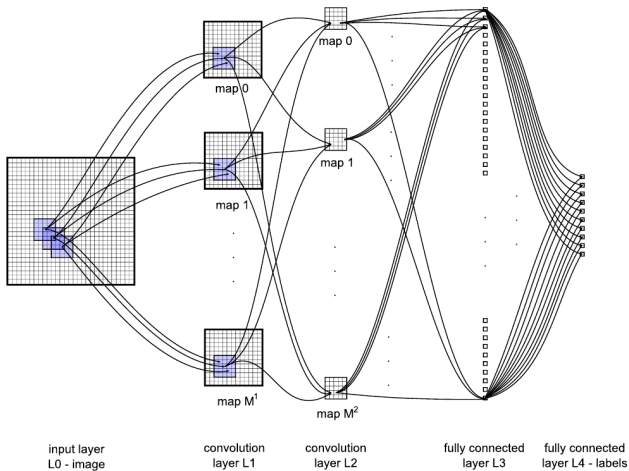


- Przy przetwarzaniu obrazów sygnał wejściowy to tensor 3D (kanał, szerokość, wysokość), w zastosowaniu rozszerzony do 4D o wymiar związany z rozmiarem mini-batcha

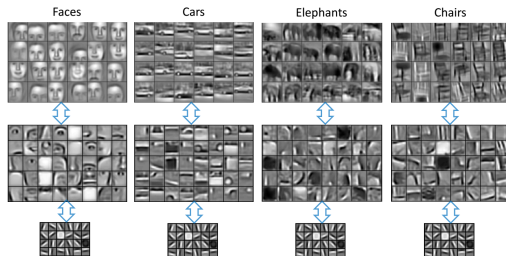
$$s(i, j, k) = \sum_{lmn} x(l, j + m, k + n)w(i, l, m, n)$$

- tensor wyjściowy warstwy zawierającej N filtrów (N, M_x, M_y) , gdzie M_x, M_y – rozmiary mapy wyjściowej

Typowa architektura CNN



Przykłady filtrów



Goofellow, 2016 [?]

Rozmiar mapy cech

- Warstwa zawierająca N neuronów (filtrów) tworzy N map (objętość, tensor n wymiarowy)
- Rozmiar mapy wyjściowej (M_x, M_y) zależy od:

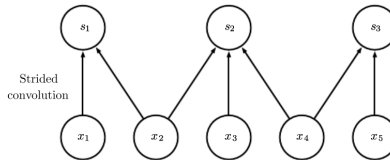
$$M_i^n = \frac{M_i^{n-1} - K_i^n + 2P_i^n}{S_i^n} + 1$$

- rozmiar filtra (K_x, k_y) (szerokość i długość)
- przesunięcie (*stride*) (S_x, S_y) w każdym z wymiarów, np. splot 2D

$$s(i, j, k) = \sum_{lmn} x(l, j \times S + m, k \times S + n) w(i, l, m, n)$$

- sposób uwzględnienia wartości brzegowych (*zero padding*), P_i
 - wielkość rozszerzenia brzegu w wymiarze i

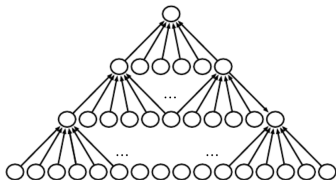
Stride



- redukcja wymiaru - zmniejszenie wymogów obliczeniowych i pamięciowych
- zmniejszenie rozdzielczości sygnału
- kosztem mniej dokładnej reprezentacji

Zero padding

valid zero padding - rozmiar kolejnych map maleje w kolejnych warstwach



- ogranicza to możliwość budowania głębokich sieci i wymusza stosowanie małych filtrów
- sygnał wejściowy na brzegach ma mniejszy wpływ na sygnał wyjściowy

Zero padding

same zero padding - brzegi wypełnione dodatkowymi wartościami (zerami) aby zapewniać odtworzenie wymiaru sygnału wejściowego



- pozwala budować bardzo głębokie sieci o dowolnych wielkościach filtrów $M_i^n = \lceil \frac{M_i^{n-1}}{S_i} \rceil$
- sygnał wejściowy na brzegach ma mniejszy wpływ na sygnał wyjściowy

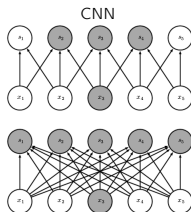
Właściwości CNN

- **rzadka reprezentacja** - rozmiar filtra jest dużo mniejszy od rozmiaru sygnału wejściowego, pojedyncze wejście oddziałuje tylko na grupę neuronów
- **współdzielenie parametrów** - warstwa splotowa może być widziana jako w pełni połączona warstwa ze współdzielonymi wagami (dużo mniejsza liczba parametrów w stosunku do MLP)
- **równoważność względem przesunięcia** sygnału - ta sama cecha znajdująca się w różnych miejscach obrazu będzie aktywowała ten sam filtr
- możliwość użycia sygnału wejściowego o zmiennym rozmiarze - większy obraz wejściowy wygeneruje większe mapy, w przypadku MLP zwiększenie rozmiaru wektora wejściowego wymaga rozbudowy architektury

Rzadka reprezentacja

pojedyncze wejście aktywuje tylko grupę neuronów w małym obszarze

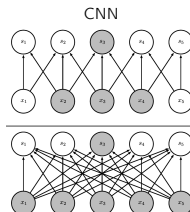
wyjście neuronu zależne od małego obszaru sygnału wejściowego



MLP

MLP: mnożenie macierzy $O(m \times n)$ parametrów

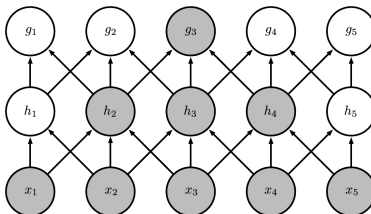
CNN: warstwa splotowa $O(k \times n)$, gdzie $k \ll m$



MLP

Efektywne pole recepcyjne

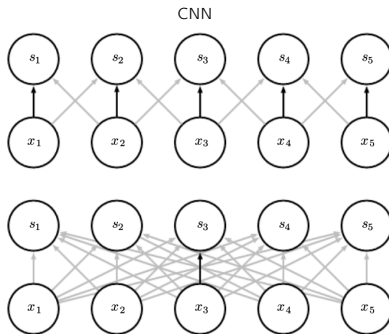
- **Efektywne pole recepcyjne** - obszar sygnału wejściowego pokryty przez neurony w wyższych warstwach, rośnie z głębokością



- stride, pooling i dilation (rozrzedzony splot) dodatkowo zwiększają efektywne pole recepcyjne
- pomimo rzadkich połączeń sieć jest w stanie w ten sposób modelować złożone zależności

Współdzielenie wag

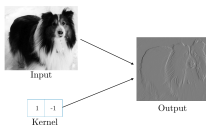
- ta sama waga jest używana przy przetwarzaniu każdego punktu wejściowego
- pojedynczy filtr pozwala wykryć tę samą cechę w różnych położeniach obrazu wejściowego (*equivariance to translation*)



Źródło grafiki: Goofellow, 2016 [?]

Rzadkie połączenia i współdzielenie wag

Przykład: wykrywanie krawędzi dla obrazu $n \times n$ w poziomie



- MLP: mnożenie pełnej macierzy $n^2 \times n^2 = n^4$
- CNN: splot filtrem 1×2 (2 wagi) wymaga $n^2 \times 3$ operacji (2 mnożenia i dodawanie)
- splot drastycznie bardziej wydajny w mapowaniu powtarzających się relacji w małych, lokalnych rejonach sygnału wejściowego

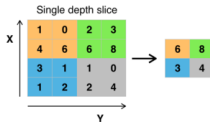
Źródło grafiki: Goofellow, 2016 [?]

Równoważność przy przesunięciu

- przesunięcie sygnału wejściowego powoduje identyczne przesunięcie sygnału wyjściowego na mapie cech
- własność pożądana w rozpoznawaniu obrazów, gdzie lokalne cechy (np. krawędzie) mogą wystąpić w każdym miejscu na obrazie
- współdzielenie wag dla całego wejścia nie zawsze jest pożądane (różne filtry w różnych obszarach obrazu), np. rozpoznawanie twarzy ze zdjęć paszportowych, posiadają charakterystyczne cechy występujące wyłącznie w określonych rejonach sygnału wejściowego
- splot nie jest niezmienniczy względem zmiany skali lub obrotu obrazu, osiąga się to poprzez rozszerzenie zbioru treningowego o przypadki zdeformowane (szum, skalowanie, obrót, zmiana kontrastu, etc..)

Pooling

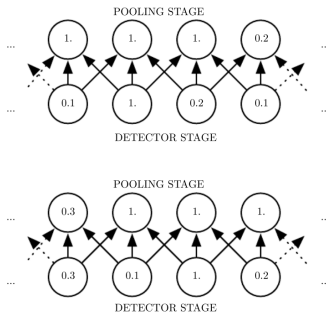
- typowa warstwa w sieciach CNN:
 - liniowa aktywacja (splot)
 - detekcja (nieliniowość np. ReLU)
 - funkcja redukcji (pooling) – „uogólnienie” wartości sąsiadujących wyjść
- **Max pooling** – maksimum z pewnego podobszaru (*winner takes all*)



- redukcja wymiarowości – przesunięcie filtra typowo równe jego wielkości (maksimum z rozłącznych obszarów)
- inne podejścia: *avg. pooling* (średnia z sąsiedztwa), *norm pooling* (norma z sąsiadujących wyjść), ważona średnia od centrum, niektóre architektury rezygnują z tej warstwy

Pooling względem obrazu wejściowego

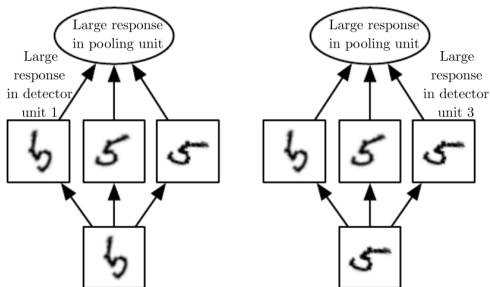
Pooling zapewnia niezmienniczość względem drobnego przesunięcia obrazu wejściowego



Źródło grafiki: Gooffellow, 2016 [?]

Pooling względem map cech

Redukcja względem wyjść różnych splotów umożliwia wprowadzenie do modelu niezmienniczości względem pewnych transformacji sygnału wejściowego (np. obrotu)



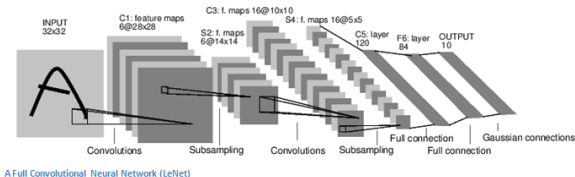
Źródło grafiki: Gooffellow, 2016 [?]

LeNet5 (LeCun 1998)

klasyfikacja cyfr pisanych ręcznie
(odczytywanie czeków)

👉 MNIST

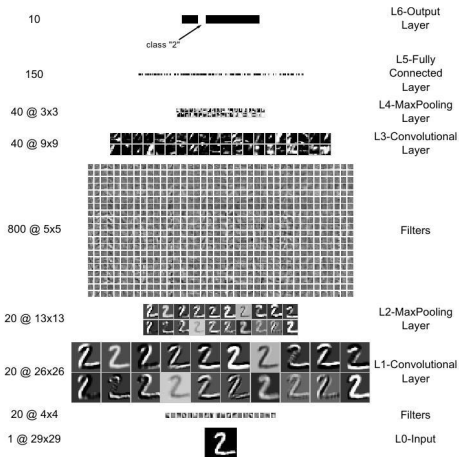
trening: 60k cyfr, 250 osób,
test: 10k cyfr



👉 LeNet-5, convolutional neural networks

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. *Gradient-based learning applied to document recognition. Proceedings of the IEEE, november 1998*

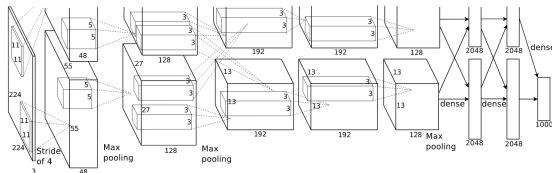
Przykład wizualizacji aktywacji sieci



👉 Wizualizacja sieci splotowej

AlexNet (Krizhevsky, 2012)

- Zwycięzca 🏆 ImageNet w 2012 z poprawnością 15.3% (poprawa z 26%)



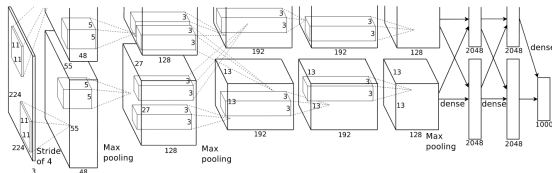
- głębsza od LeNet5 i więcej filtrów na warstwę (11x11, 5x5, 3x3)
- sekwencje warstw splotowych
- dropout
- 6 dni treningu (2x NVIDIA GTX 580 3GB GPUs).

AlexNet



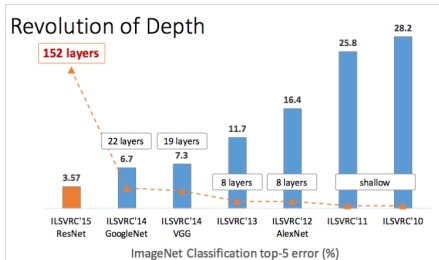
AlexNet (Krizhevsky, 2012)

- Zwycięzca 🏆 ImageNet w 2012 z poprawnością 15.3% (poprawa z 26%)



- głębsza od LeNet5 i więcej filtrów na warstwę (11x11, 5x5, 3x3)
- sekwencje warstw splotowych
- dropout
- 6 dni treningu (2x NVIDIA GTX 580 3GB GPUs).

ImageNet



- ZFNet(2013) - podobny do AlexNet
- GoogleNet (Inception) - małe sploty w celu zredukowania liczby parametrów
- 🖱️ CNNs Architectures used on ImageNet

VGGNet (Simonyan and Zisserman, 2014)

- dużo małych filtrów 3x3
- wielokrotne warstwy splotowe dodają nieliniowość oraz zwiększają efektywne pole recepcyjne
- stride=1, brak utraty informacji
- 5 warstw max pool
- od 11 do 19 warstw
- 138M parametrów
- trening 2-3 tygodnie (4x GPU).

conv-64

conv-64

maxpool

conv-128

conv-128

maxpool

conv-256

conv-256

maxpool

conv-512

conv-512

maxpool

conv-512

conv-512

maxpool

FC-4096

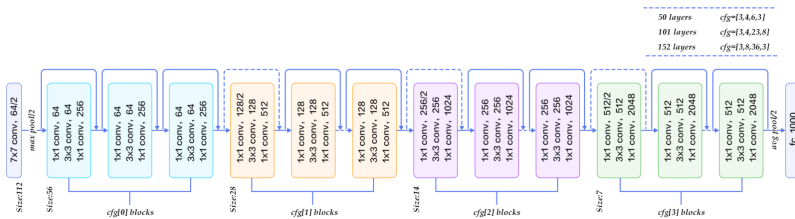
FC-4096

FC-1000

softmax

K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv technical report, 2014

ResNet (Residual Neural Network, Kaiming He et. al 2015)



- wiele bloków zawierających sploty 1x1, 3x3, 1x1
- przejścia „skrótowe” pomagają uczyć bardzo głębokie sieci (152 warstwy)
- batch normalization