Inteligencja obliczeniowa - Deep Learning: Konwolucyjna sieć neuronowa

Grzegorz Madejski

Definicja

Deep learning (uczenie głębokie)

Dziedzina uczenia maszynowego wykorzystująca głębokie (wielowarstwowe) modele, do wykonywania skomplikowanych zadań (np. rozpoznawania obiektów na obrazach, rozpoznawania mowy, tłumaczenia, projektowania leków). Charakterystyczna dla głębokiego uczenia jest umiejętność rozpoznawania cech obiektu (ang. features). Im wyższa warstwa, tym bardziej wysokopoziomowe (skomplikowane) cechy.

Popularne głębokie modele

Kilka przykładów:

- *Konwolucyjne (splątaniowe) sieci neuronowe* (ang. convolutional neural network, CNN) - przetwarzanie obrazów
- *Rekurencyjne sieci neuronowe* (ang. recurrent neural network, RNN) - przetwarzanie języka (wśród nich popularne modele to LSTM, GRU)
- *Głębokie uczenie przez wzmacnianie* (ang. deep reinforcement learning) - robotyka, gry, język, obrazy
- *Generatywne sieci przeciwstawne* (ang. generative adversarial network, GAN) - generowanie obrazów

Popularne głębokie modele

Głębokie uczenie najlepiej przeprowadzać z użyciem specjalistycznych bibliotek:

- *keras* - prosta, przyjazna do nauki biblioteka; od 2017 roku stała się częścią tensorflow
- *tensorflow* - potężna, dobrze udokumentowana biblioteka, z dużą społecznością, wieloma opcjami np. wizualizacją danych
- *pytorch* - popularna biblioteka, dobrze zoptymalizowana, elastyczna, wydajna

Definicja

Konwolucyjna sieć neuronowa

Konwolucyjna (splotowa, splątana) sieć neuronowa (ang. convolutional neural network, CNN, ConvNet) - głęboka sieć neuronowa do rozpoznawania obrazów, wykorzystująca filtry do wykrywania cech obrazu.

Sieci konwolucyjne inspirowane sa biologicznie. Naśladują działanie naszego mózgu i oczu, do przetwarzania bodźców wzrokowych. Różne części naszego mózgu (grupy neuronów) wykrywają różne cechy obrazu (linie, kolory, krawędzie). Pierwsze neurony zbierają proste bodźce, dalsze neurony zbierają je w skomplikowane kształty.

Typy warstw

Charakterystyczne dla CNN są trzy typy warstw:

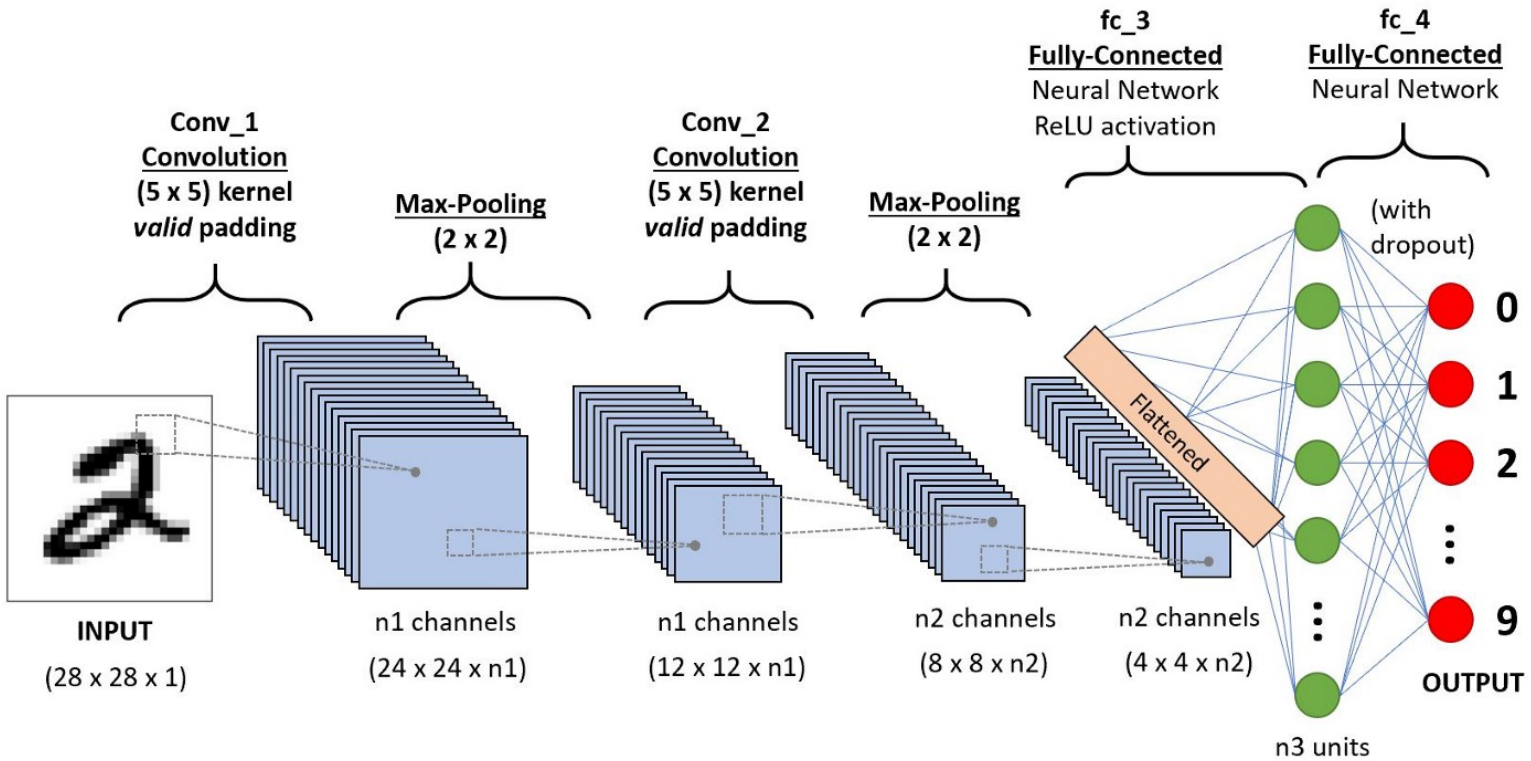
- *warstwy w pełni połączone* (ang. fully connected layer, dense layer) - znamy ze zwykłych sieci neuronowych, warstwa o m neurnonach łączona jest z warstwą o n neuronach $m \cdot n$ połączeniami (każde ma swoją wagę)
- *warstwa splotowa* (ang. convolutional layer) - wykorzystuje specjalny filtr (zwany *jądrem łaczącym*, kernelem) do zamiany jednej warstwy neurnonów na drugą
- *warstwa łącząca* (ang. pooling layer) - warstwa kompresująca dane do mniejszego rozmiaru

Struktura CNN

Ogólne zasady tworzenia CNN:

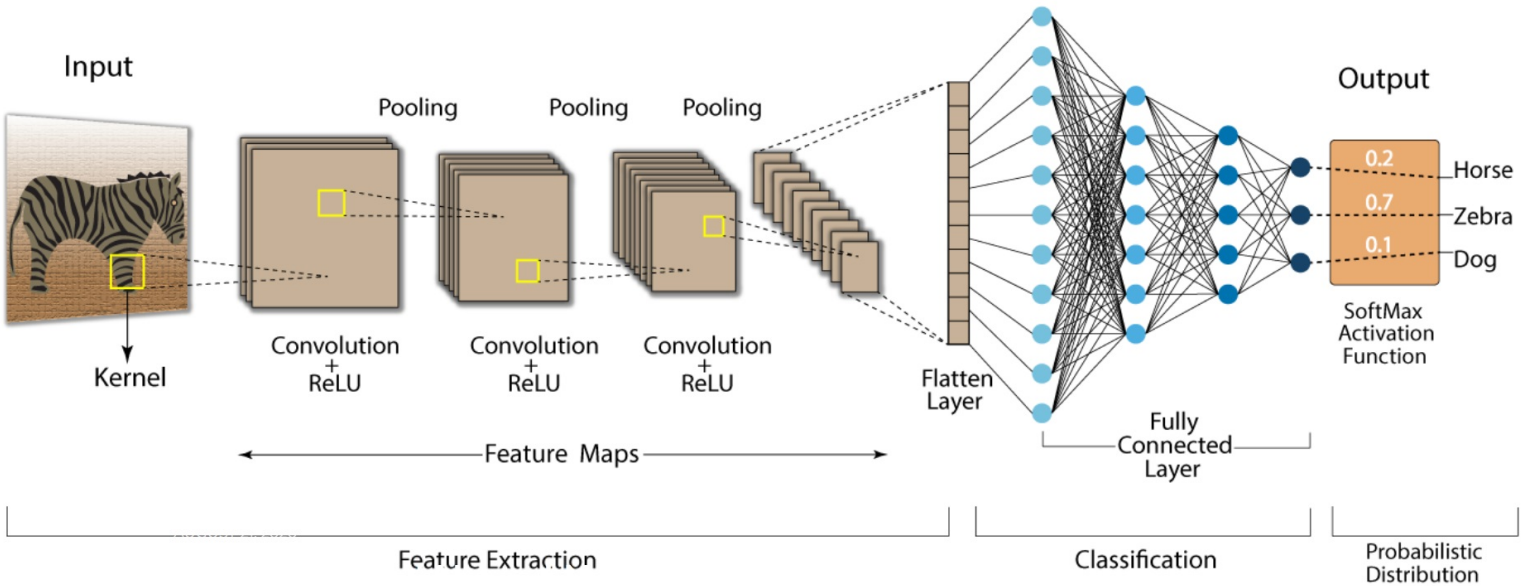
- Dostajemy obrazek (jako zestaw pikseli). Jeśli jest kolorowy, to dostajemy 3 kanały (R, G, B).
- Obrazek przetwarzamy za pomocą zestawu filtrów zwanych jądrami (kernel). Każdy filtr wytwarza nowy obraz (kanał, mapę).
- Kanały pomniejszamy warstwami typu pooling (łączącymi), zamieniającymi grupę pikseli na jeden piksel.
- Warstwy splotowe i łączące na ogół są zamiennie stosowane, aż kanały zmniejszą się do małych kilkupikselowych (jednopikselowych) obrazów.
- Wówczas traktujemy je jako input dla normalnej sieci neuronowej i przetwarzamy warstwami w pełni połączonymi.

Przykład sieci konwolucyjnej



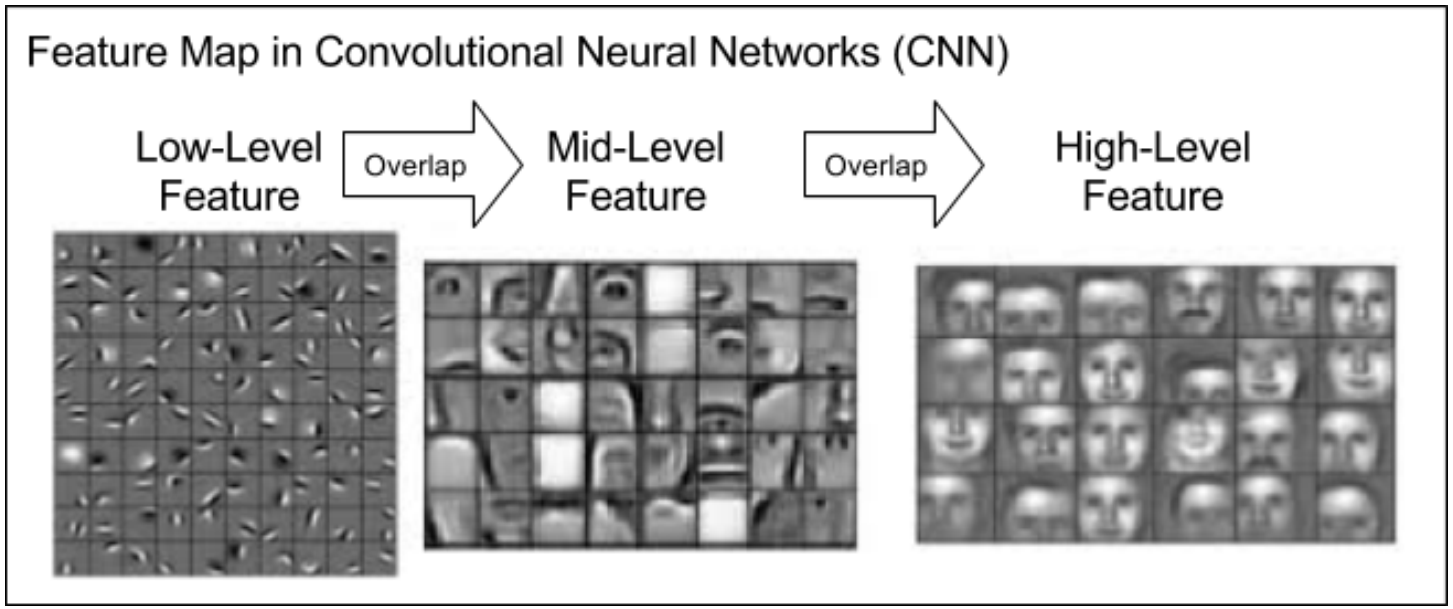
Źródło: `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3b`

Przykład sieci konwolucyjnej



Źródło: `https://developersbreach.com/convolution-neural-network-deep-learning/`

Wykrywanie cech



Uczenie CNN

- Podobnie jak zwykłe sieci neuronowe, uczymy algorytmem wstecznej propagacji (metoda spadku gradientu).
- Poprawiane są wagi na połączeniach warstw w pełni połączonych oraz struktury filtrów w warstwach splotowych. Sieć tak poprawia kernele, żeby wydobyć z obrazu najważniejsze cechy.

Zalety CNN

- Jeśli obrazek ma 1000 x 1000 pikseli, to mamy 1 mln pikseli.
- Jeśli stworzymy warstę ukrytą o 500 tys. neuronów, i połączymy dwie warstwy, to mamy do wytrenowania $1000000 \cdot 500000 = 500000000000$ (500 miliardów) wag.
- Jeśli do przetworzenia obrazka wykorzystamy 10 filtrów 5x5 to mamy do wytrenowania $10 \cdot 5 \cdot 5 = 250$ liczb. Różnica jest kolosalna.
- Uczenie CNN jest szybkie (czas i pamieć). Nie ma tendencji do przeuczania się.

Optymalizacja uczenia: funkcje aktywacji

- W warstwach ukrytych CNN bardzo często stouje się funkcję aktywacji zwaną *rektyfikatorem liniowym* (ang. rectified linear unit, ReLU) daną wzorem:

$$ReLU(x) = \begin{cases} 0, & \text{dla } x < 0 \\ x, & \text{dla } x \geqslant 0 \end{cases}$$

- Funkcja ta jest prosta i zapobiega długim czasom uczenia się.
- Funkcja ta wyklucza wartości negatywne, co można rozumieć tak: jeśli neuron nie wnosi nic w zrozumienie obrazka, to wykluczmy jego wpływ na decyzję (zamiast uwzględniać negatywny wpływ).
- Funkcja liniowa ma stałą pochodną (więc stały gradient). Nie mamy sytuacji, gdy gradient zanika (vanishing gradient).

Optymalizacja uczenia: funkcje aktywacji

- Warstwa wyjściowa o k outputach x_1 do x_k często ma funkcję aktywacji *softmax* daną wzorem

$$\sigma(x_1, x_2, ..., x_k) = \left(\frac{e^{x_1}}{e^{x_1} + ... + e^{x_k}}, ..., \frac{e^{x_k}}{e^{x_1} + ... + e^{x_k}} \right)$$

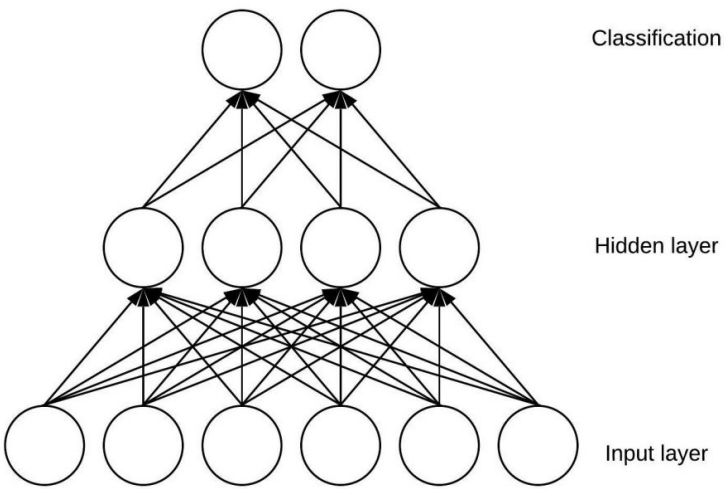
- Funkcja ta przedstawia wartości symulujące prawdopodobieństwa danych odpowiedzi. Ma zakres od 0 do 1. Wartości sumują się do jedynki. Przykład:

$$\sigma(0.5, 1.1, -0.2) = (0.3, 0.55, 0.15)$$

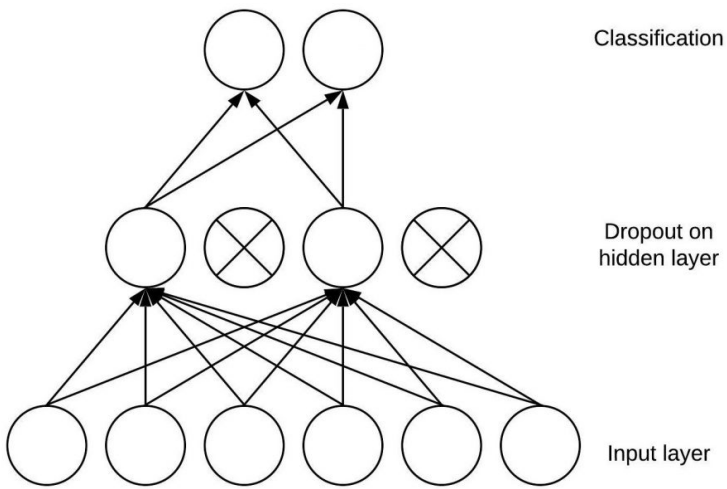
- Funkcja wykładnicza e^x jest stosowana po to, żeby pozbyć się negatywnych wartości.

Optymalizacja uczenia: dropout

Dropout to technika, która losowo wyklucza niektóre neurony w procesie uczenia sie. Działa jak dodatkowa warstwa (lub maska), którą nakładamy na wybrane warstwy (wejściową lub ukryte). Dropout w dużych sieciach przeciwdziała przeuczeniu się sieci.



Without Dropout



With Dropout

Źródło: `https://www.baeldung.com/cs/ml-relu-dropout-layers`

Optymalizacja uczenia: normalizacja batchowa

Zwykle normalizujemy dane wejściowe. Techniką przyspieszającą uczenie się sieci, jest też normalizacja danych wchodzących do warstw ukrytych. Proces ten nazywamy *normalizacją batchową* (ang. batch normalization), bo wykorzystuje ona dane wyliczane z mini-batchy. Przybliżony schemat działania:

- Policz średnią μ_B i odchylenie standardowe σ_B dla batcha B.
- Dla wybranej warstwy w sieci znormalizuj dane wejściowe wg wzoru $x_{norm} = (x - \mu_B)/\sigma_B$. Dane są dosyć "zgniecione", więc dodajemy krok:
- Przeskaluj i przesuń wyniki wg wzoru $y = \gamma x_{norm} + \beta$, gdzie γ i β to współczynniki poprawiane w procesie uczenia się.

Optymalizacja uczenia: regularyzacja

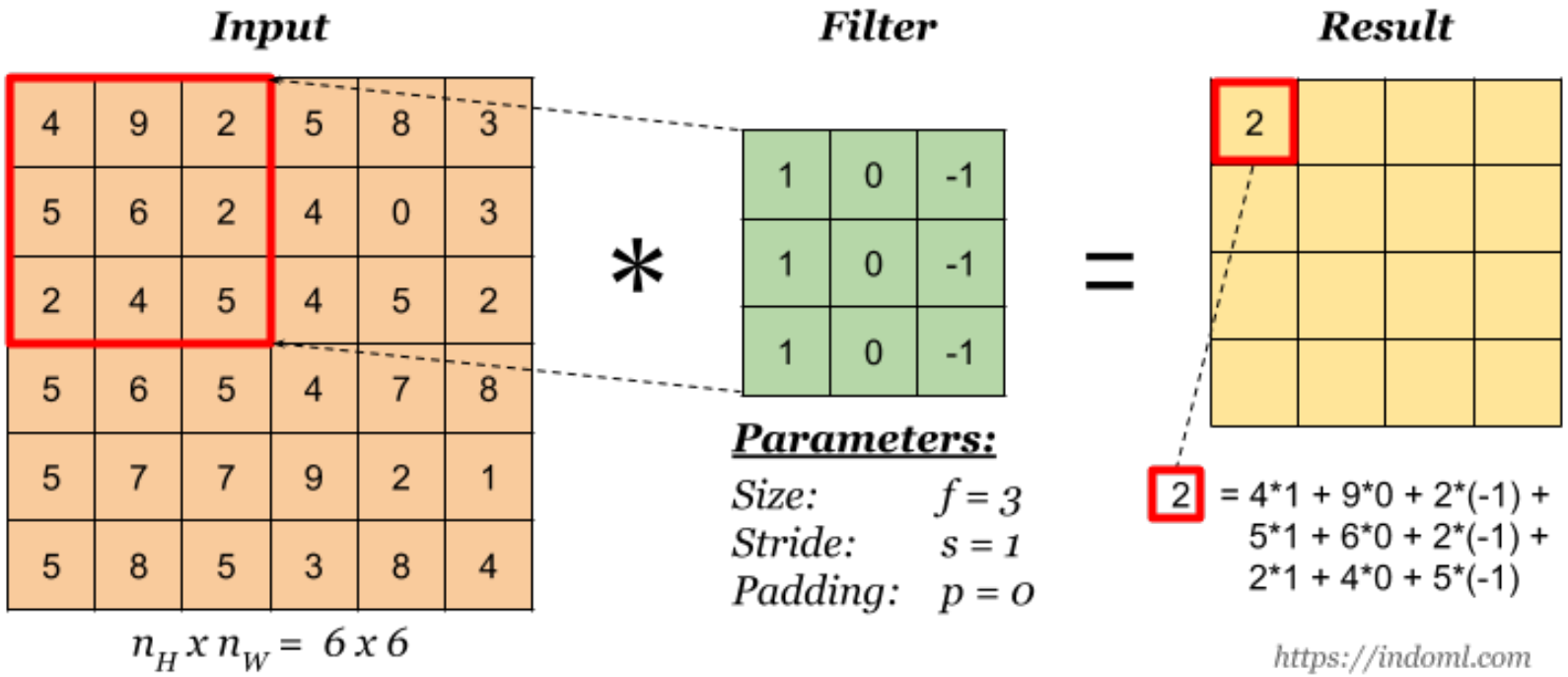
Zarówno *dropout*, jak i *normalizacja batchowa* to techniki, które zapobiegają przeuczeniu sieci poprzez zmniejszenie ich mocy obliczeniowej. Proces ten nazywamy *regularyzacją* (ang. regularization).

Warstwa splotowa - działanie

- Obraz otrzymujemy na wejściu. Obrazem może mieć trzy kanały (R, G, B).
- CNN przetwarza obraz produkując nowe obrazy. Nazywamy je *kanałami* lub *mapami cech*.
- Jeśli do dyspozycji mamy n filtrów, to kanał zostanie zamieniony na n nowych kanałów, każdy powstaje z osobnego filtra.
- Filtry nazywamy *kernelami* lub *jądrami* łączącymi
- Jądro ma z reguły postać małej kwadratowej macierzy (np. 3x3, 5x5) z liczbami.
- Jądro przesuwa się od lewej do prawej, od góry do dołu przez obraz, mnoży swoje parametry przez piksele i je dodaje.

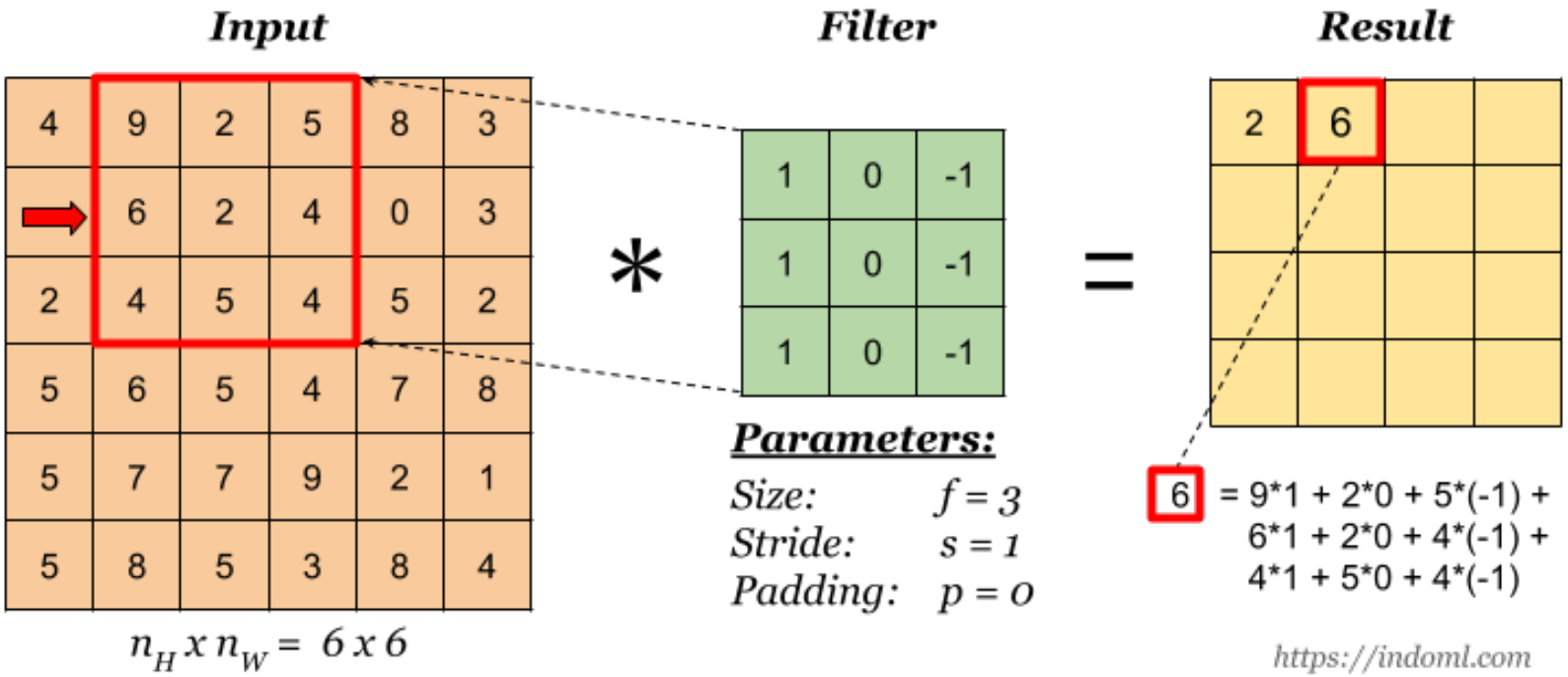
Warstwa splotowa - działanie

Z poradnika: https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/



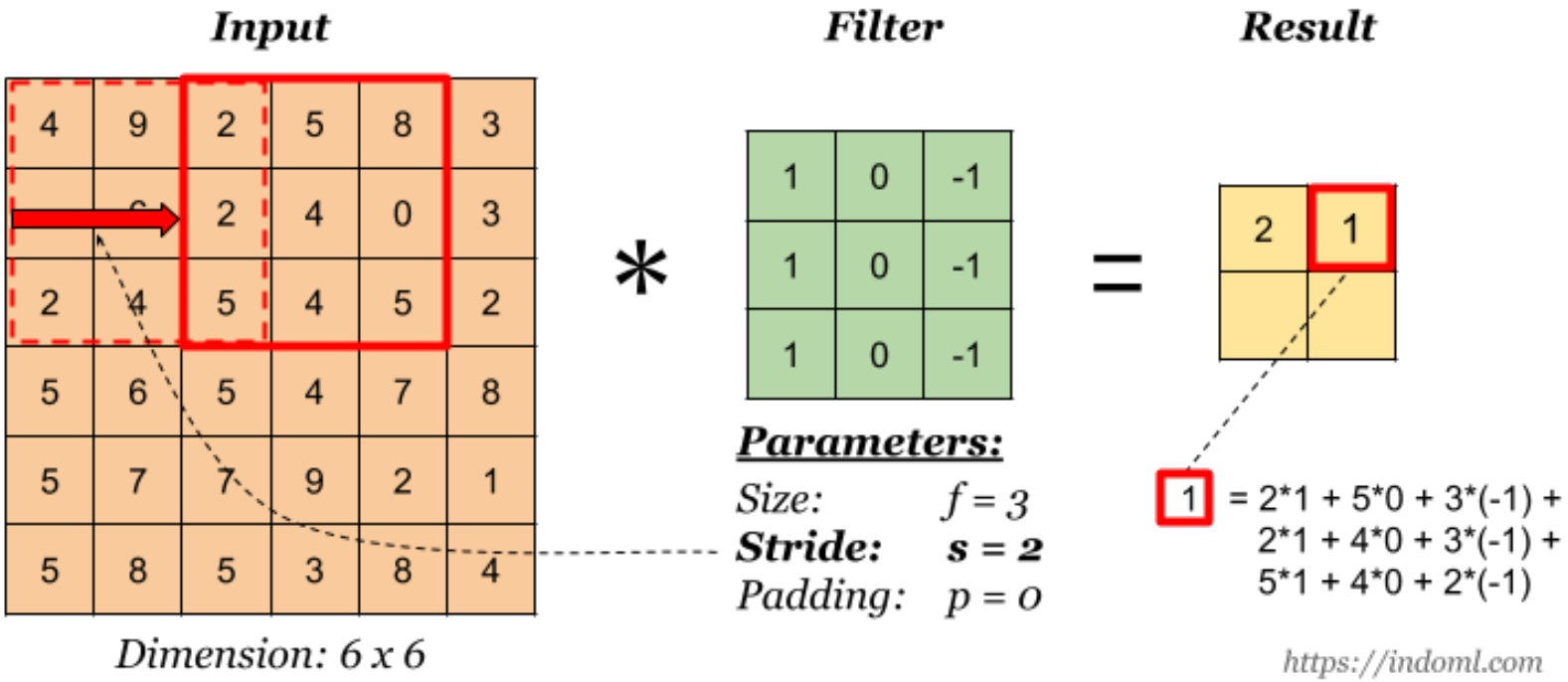
Warstwa splotowa - działanie

Z poradnika: https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/



Warstwa splotowa - działanie

Z poradnika: https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/

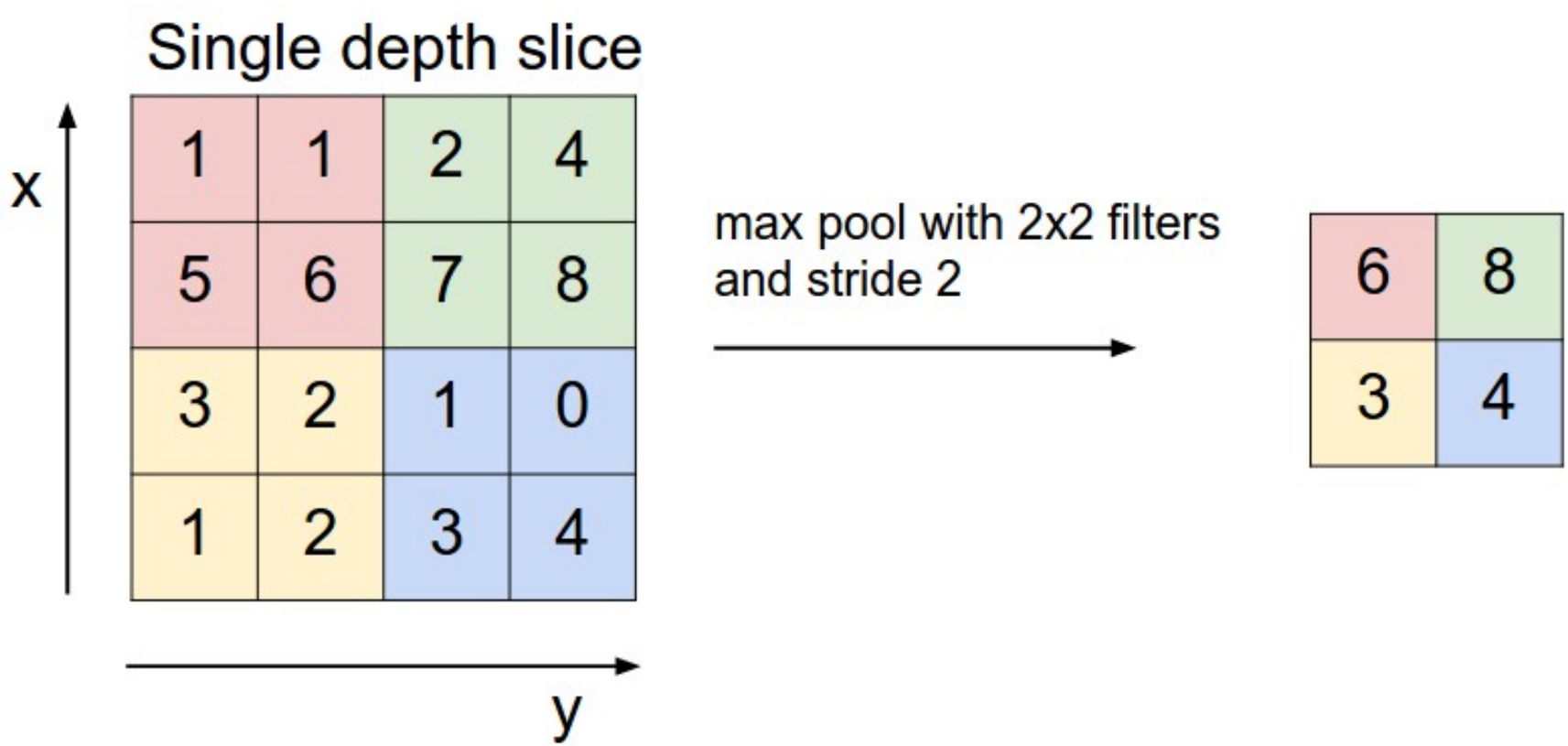


Warstwa łącząca - działanie

- Warstwa łącząca ma specjalne okienko (2x2, 3x3), które przesuwa się od lewej do prawej, od góry do dołu i zamienia grupy pikseli (4, 9) na jeden piksel.
- Są różne techniki zamianany kilku pikseli na jeden:
  - max-pooling - wybieramy piksel o największej wartości
  - avg-pooling - wyliczamy srednią z pikseli w okienku
  - min-pooling - wybieramy piksel o najmniejszej wartości
- Warstwa łącząca nie jest wyuczana, służy jedynie do zmniejszenia wymiarowości warstw.

Warstwa łącząca - działanie

Przykład max-poolingu ze stride (przesunięciem) równym 2. Gdyby stride było równe 1, to kwadraty by na siebie nachodziły.

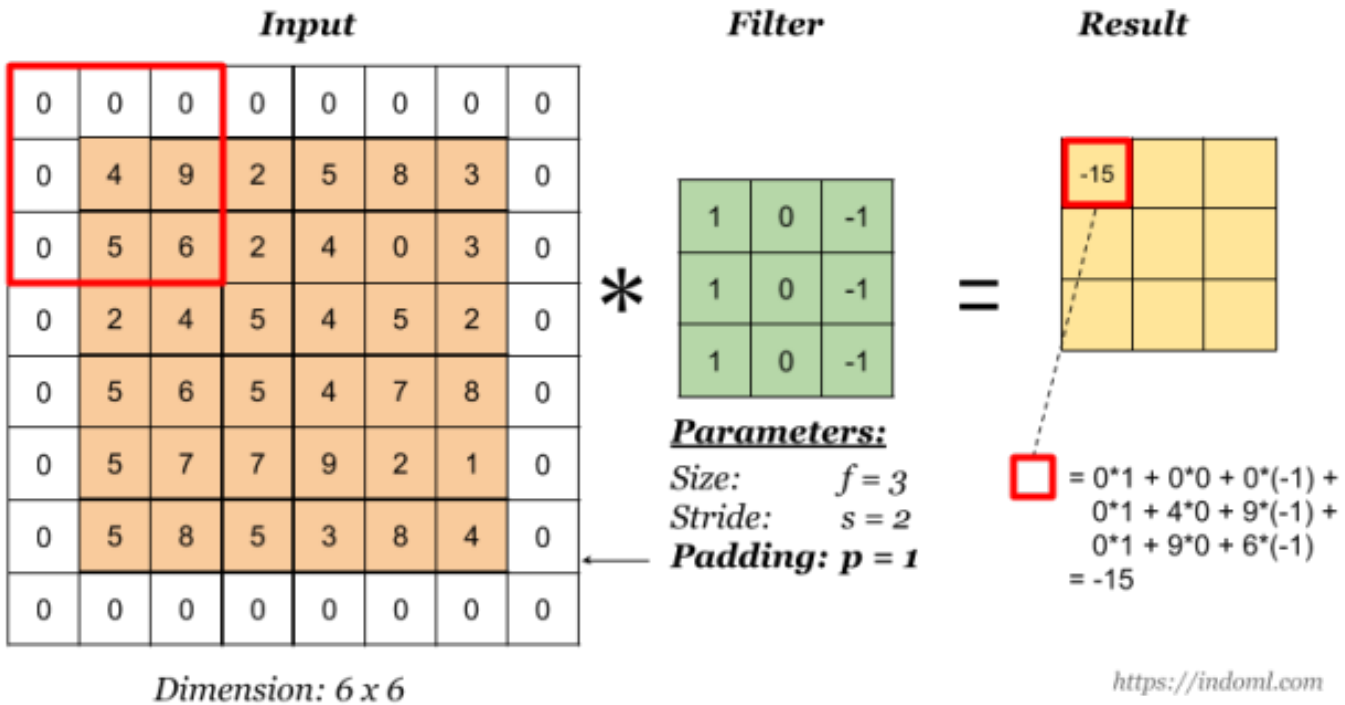


Padding

- Przesuwanie okienka z jądrem lub z poolingiem może nie pokryć dobrze obrazka.
- Stosujemy dwie taktyki:
  - Dodajemy padding na brzegu obrazka. Dzięki temu wynikowy kanał nam się nie zmniejszy.
  - Nie dodajemy paddingu, wówczas wynikowy kanał bedzie mniejszy o parę pikseli.

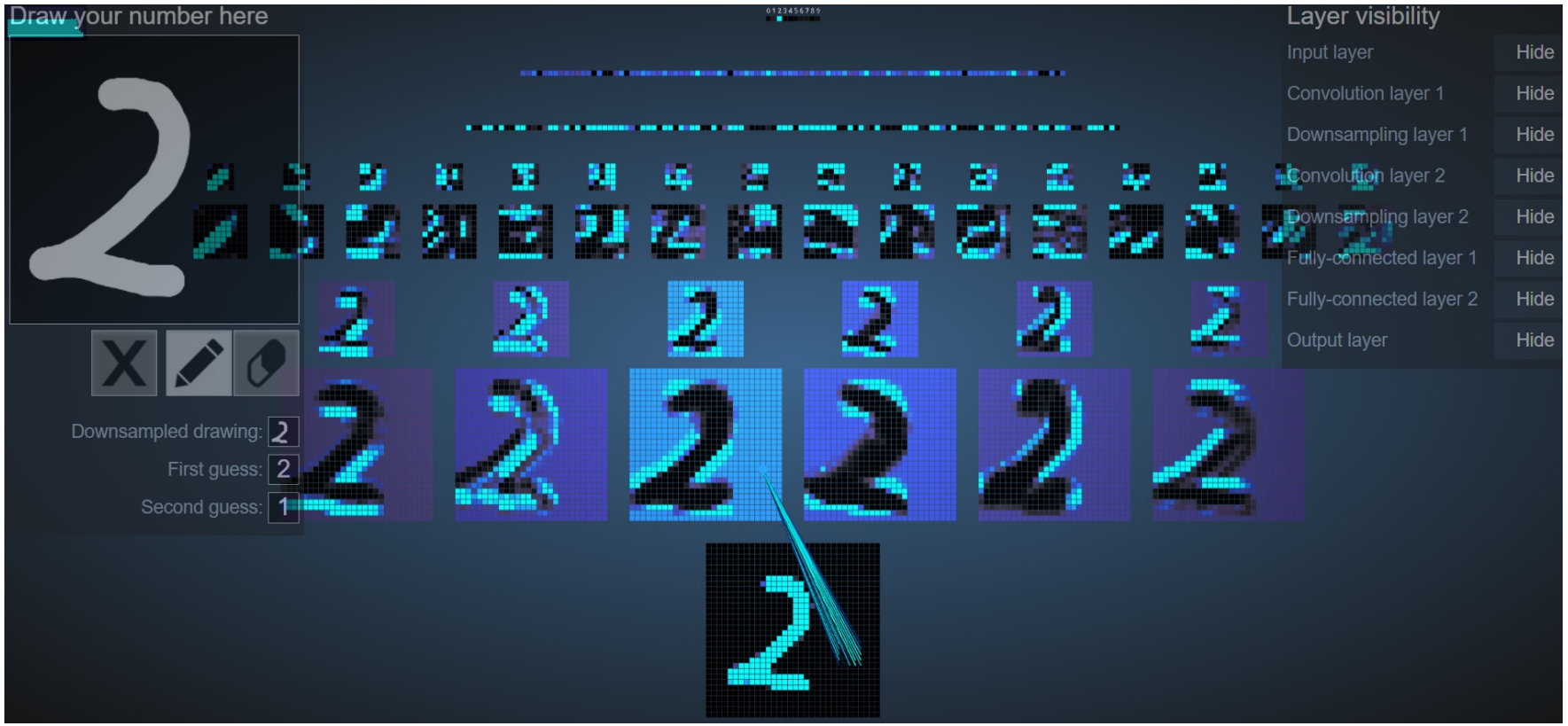
Padding

Padding ustawiony na 1.



CNN dla cyfr - wizualizacja

Na stronie: `https://adamharley.com/nn_vis/` na wejść w 2D- lub 3D-covn net i zapoznać się z siecią do rozpoznawania cyfr.



CNN dla cyfr - wizualizacja

Przyjrzyj się sieci konwolucyjnej ze strony i odpowiedz na pytania:

- W pierwszej warstwie (konwolucyjnej) ile stosujemy filtrów/kerneli?
- W pierwszej warstwie (konwolucyjnej) jaki wymiar mają kernele?
- Jaki pooling stosowany jest w sieci?

Inne CNNy - wizualizacja

Przykłady działania różnych CNN z możliwością konfiguracji uczenia.
`https://cs.stanford.edu/people/karpathy/convnetjs/`

Kaggle

Gdzie szukać baz danych do klasyfikacji / uczenia?

`https://www.kaggle.com/datasets`