

Jak uczenie maszynowe jest wykorzystywane w geoinformatyce ?

Warsztat poprowadzą:
Dominik Mielczarek
Piotr Czubek



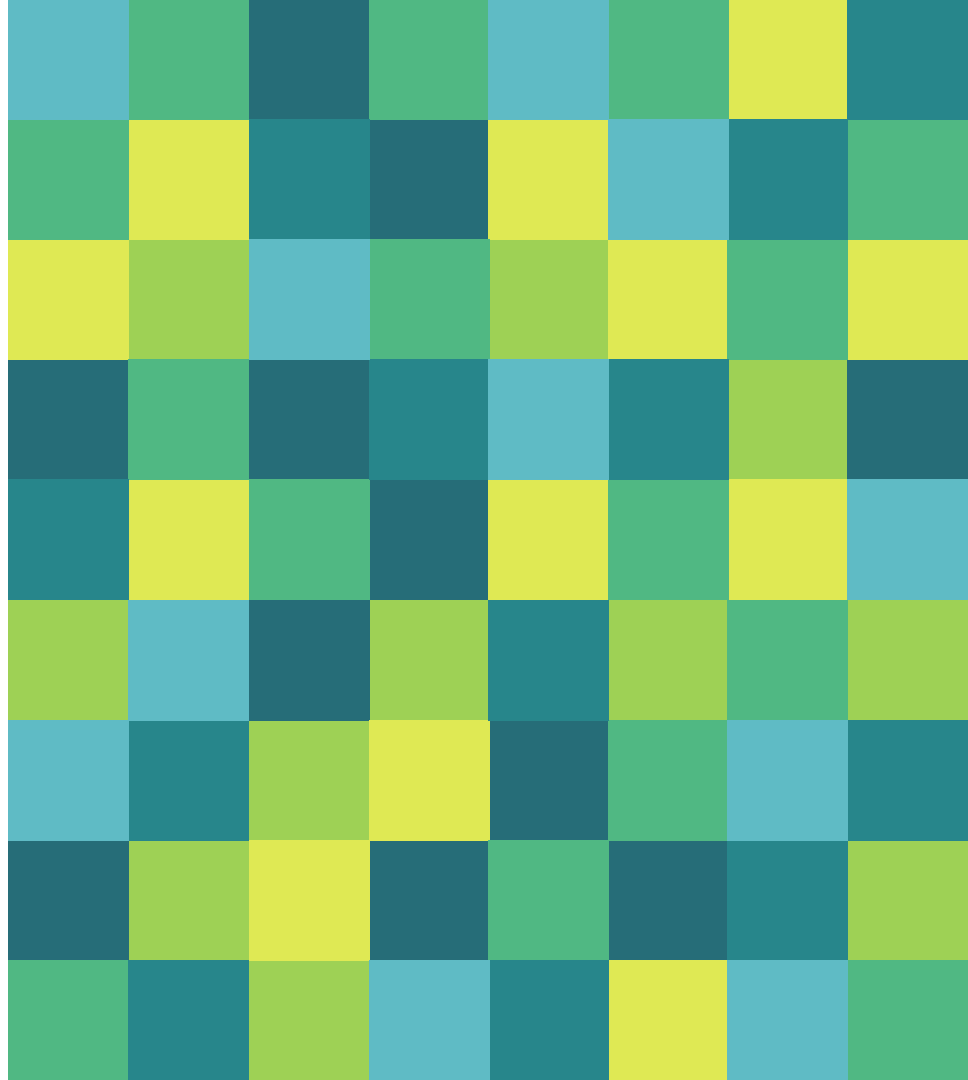
Plan prezentacji

1. Wstęp
2. Dane przestrzenne
3. Uczenie maszynowe
 - a. Ćwiczenie - klasyfikacja chmury punktów na podstawie cech geometrycznych
 - b. Ćwiczenie - klasyfikacja zdjęć lotniczych za pomocą modelu lasów losowych
4. Uczenie głębokie
 - a. Ćwiczenie - klasyfikacja chmury punktów za pomocą VNN
 - b. Ćwiczenie - klasyfikacja zdjęć lotniczych za pomocą CNN

1.

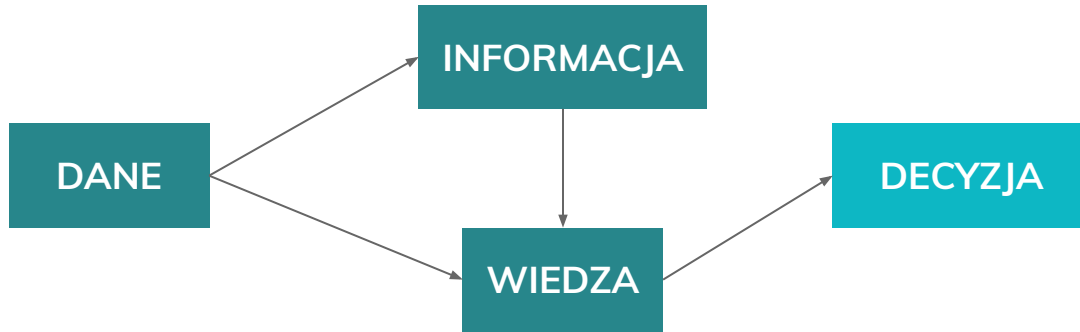
Wstęp

Czym zajmuje się
geoinformatyka ?



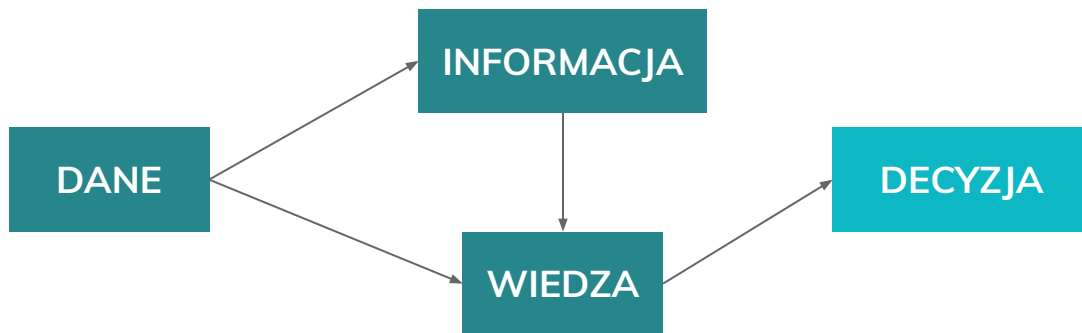
System Informacji Geograficznej

- Systemy geoinformacyjne służą przede wszystkim do **analizy rzeczywistych danych przestrzennych** w celu uzyskania cennej informacji i ułatwieniu podjęcia decyzji.



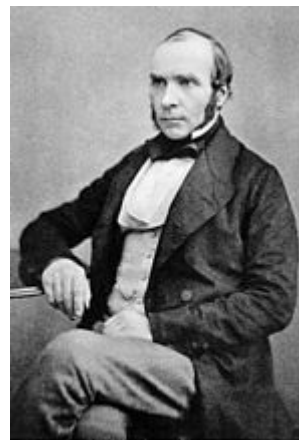
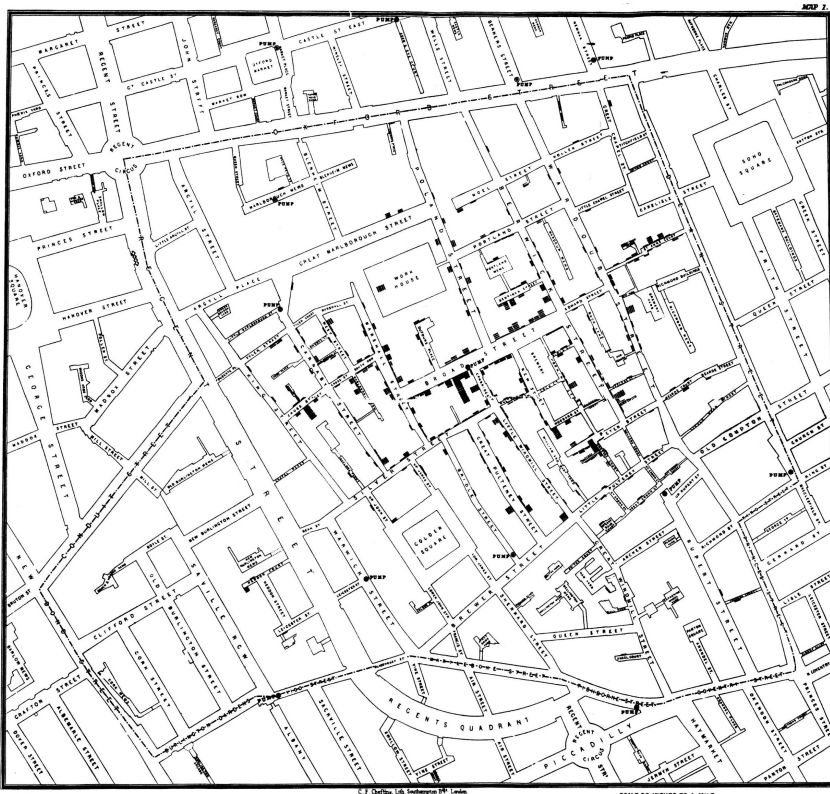
System Informacji Geograficznej

- Systemy geoinformacyjne służą przede wszystkim do **analizy rzeczywistych danych przestrzennych** w celu uzyskania cennej informacji i ułatwieniu podjęcia decyzji.



- Jak dojadę do ... ? / Ile kilometrów ... ?
- Gdzie znajde najbliższy sklep ... ? / Gdzie swoje placówki ma ... ?
- Czy w miejscu docelowym będę miał dostęp do ... ?
- Jaka jest temperatura w ... ?
- W którym budynku ... ?
- Który wariant budowy drogi jest ? / Który hotel wybrać ... ?

System Informacji Geograficznej - początek



John Snow



Systemy Informacji Geograficznej - dziś

The screenshot displays the Booking.com interface. On the left, there are filter options for star ratings (1 to 5 stars, and 'Bez kategorii') and popular filters like 'Centrum', 'Śródmieście', and 'Ulubiona okolica Gości'. The main area shows a map of Warsaw with numerous blue location pins. A callout box highlights several hotels with their ratings and review counts:

- Novotel Warszawa Centrum** (4 stars): 8,5 rating, 10 360 opinions, 'Bardzo dobry'.
- SCSK Żurawia**: 7,5 rating, 1 053 opinions, 'Dobry'.
- Noctis Apartament Grzybowska**: 8,7 rating, 42 opinions, 'Fantastyczny'.
- Radisson Blu Centrum Hotel** (5 stars): 8,3 rating, 4 665 opinions, 'Bardzo dobry'.
- InterContinental Warszawa**: 9,2 rating, 5 217 opinions, 'Znakomity'.

Turyści,
podróżnicy

Systemy Informacji Geograficznej - dziś

The image is a composite of three main elements:

- Top Left:** A screenshot of the Booking.com website. It shows a search filter for 'Liczba gwiazdek' (Number of stars) with options from 1 to 5 stars, and 'Popularne filtry' (Popular filters) including 'Centrum' (Center) and 'Ulubiona okolica Gości' (Guest favorite area). Two hotel listings are visible: 'Novotel Warszawa Centrum' with a rating of 8.5 and 'SCSK Żurawia' with a rating of 7.5.
- Top Right:** A small map of Warsaw showing various locations marked with blue pins. A white box with a black border contains the text 'Turyści, podróżnicy' (Tourists, travelers).
- Bottom:** A detailed land use map of a specific area in Warsaw. A yellow-outlined polygon highlights a plot of land. A popup window titled 'Identyfikacja' (Identification) provides the following data:
 - Opis wydziału: PGL LP
 - Rodzaj powierzchni: D-STAN
 - Typ siedliskowy lasu: L5W
 - Gospodarstwo: O
 - Funkcja lasu: OCHR
 - Budowa planowa: KO
 - Powierzchnia wydziału [ha]: 1,92
 - Kategoria ochronności: OCH MIAST
 - Udział: 7
 - Wiek gatunku: 151
 - Wiek dojrz. rębnej: 110

Systemy Informacji Geograficznej - dziś

Booking.com search results for hotels in Warsaw. The interface shows filters for star ratings (1 to 5) and popular filters like 'Centrum' and 'Ulubiona okolica Gości'. Two hotel listings are visible: 'Novotel Warszawa Centrum' with a rating of 8.5 and 'SCSK Żurawia' with a rating of 7.5. A map in the background shows the city center with various location pins.

Turyści,
podróżnicy

A GIS map showing land use and forest statistics. The map is color-coded by land use type. A pop-up window titled 'Identyfikator' provides details for a specific plot: 'Opisy wydziałów PGL LP', 'Rodzaj powierzchni: O-STAN', 'Typ siedliskowy lasu: LSW', 'Gospodarstwo: O', and 'Funkcja lasu: OCHR'. The interface includes a legend and various map controls.

Leśnicy

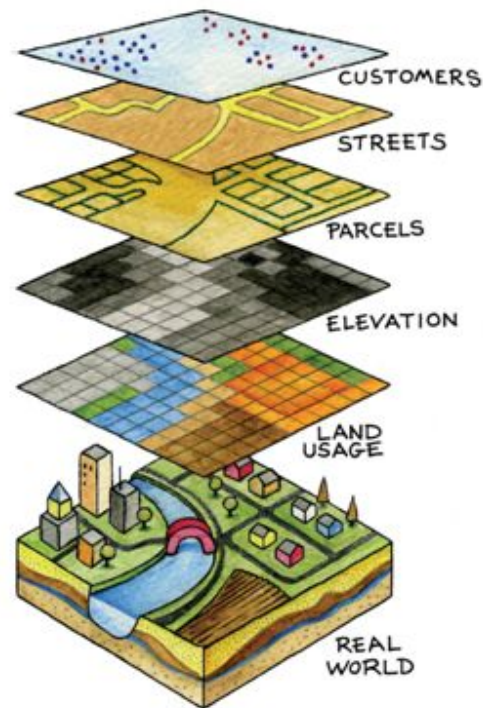
The Ełbiński System Informacji Przeszennej (Ełbiński Historical Information System) GIS interface. It displays a map of Ełbiąg with various historical data layers. A legend window is open, showing layers such as 'Ortofotomapa miasta Ełbiąg 1995', 'Ortofotomapa miasta Ełbiąg 2005', 'Ortofotomapa miasta Ełbiąg 2012', 'Ortofotomapa miasta Ełbiąg 2013', 'Ortofotomapa miasta Ełbiąg 2014', 'Ortofotomapa termalna', 'Cieniony Zmierzające', 'OpenStreetMap', 'Numeryczny Model Terenu ISOK - Cieniony', 'Numeryczny Model Terenu ISOK - Cieniony', 'Przewodność RDOT', and 'VMAPL2'. The interface also includes a search bar and various map controls.

Organizacje
rządowe,
obywatele

Proces tworzenia opracowania kartograficznego

Na proces tworzenia opracowania kartograficznego składają się:

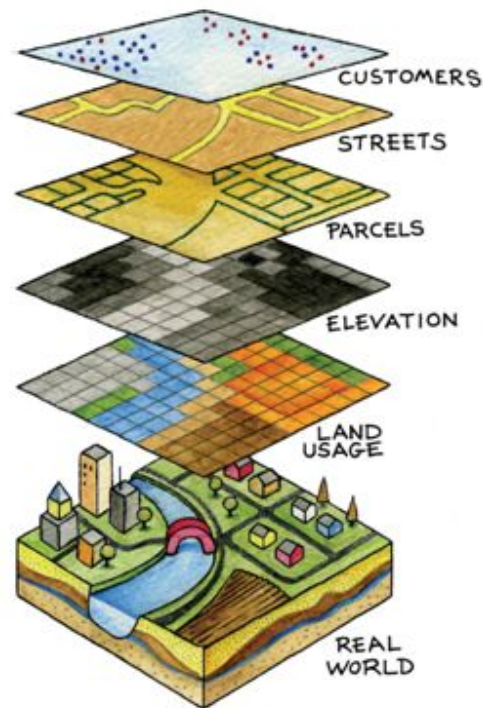
- określenie **celu** jakiemu ma służyć mapa,
- zdefiniować **skalę** w jakiej ma zostać opracowana,



Proces tworzenia opracowania kartograficznego

Na proces tworzenia opracowania kartograficznego składają się:

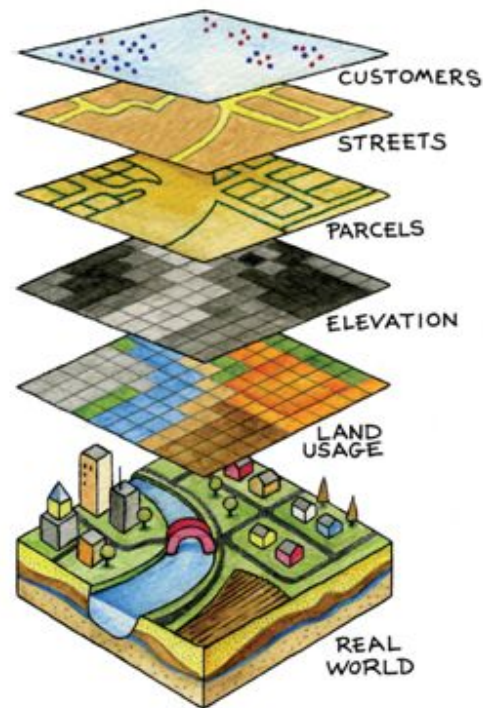
- określenie **celu** jakiemu ma służyć mapa,
- zdefiniować **skalę** w jakiej ma zostać opracowana,
- wybranie **obiektów** świata rzeczywistego jakie mają zostać przedstawione na mapie,
- określenie **modelu danych** w jakim zostaną przedstawione obiekty (rastrowy, wektorowy),
- przeprowadzenie procesu **generalizacji** obiektów ,



Proces tworzenia opracowania kartograficznego

Na proces tworzenia opracowania kartograficznego składają się:

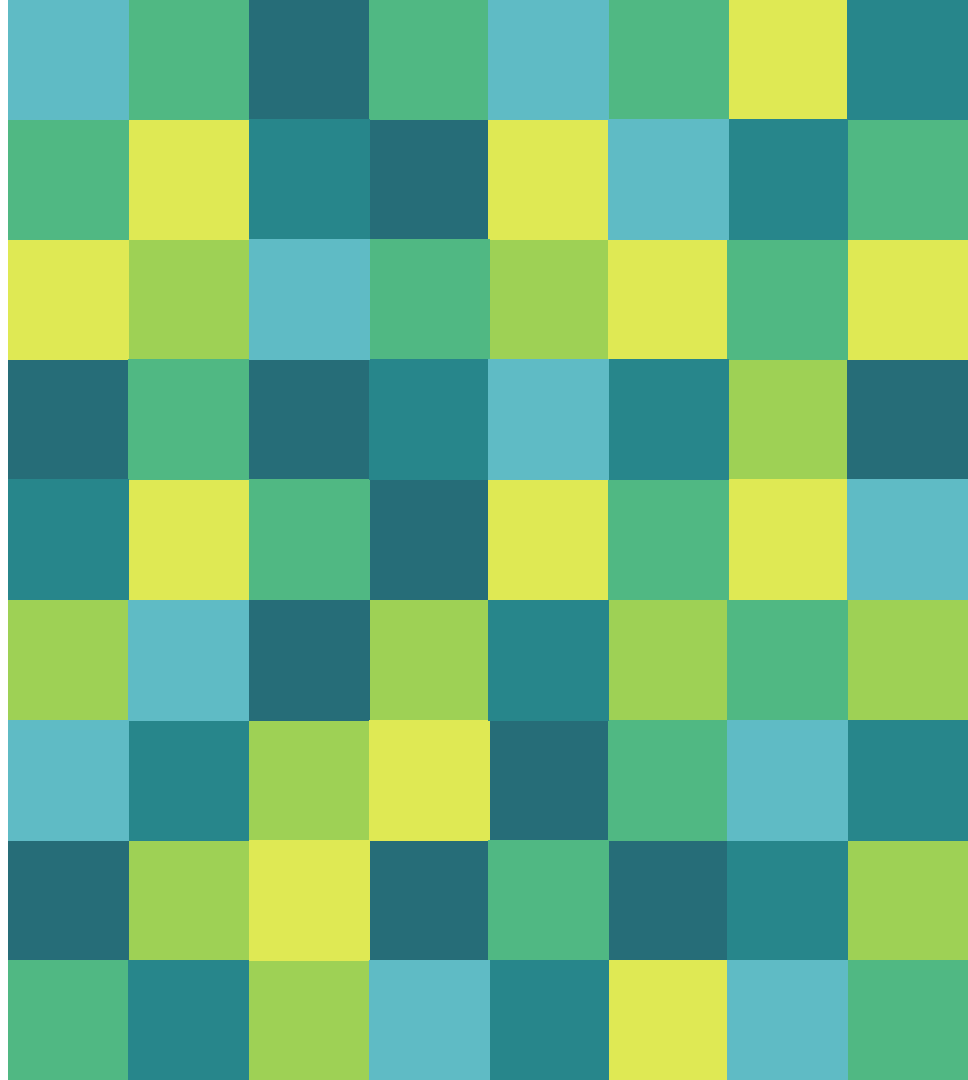
- określenie **celu** jakiemu ma służyć mapa,
- zdefiniować **skalę** w jakiej ma zostać opracowana,
- wybranie **obiektów** świata rzeczywistego jakie mają zostać przedstawione na mapie,
- określenie **modelu danych** w jakim zostaną przedstawione obiekty (rastrowy, wektorowy),
- przeprowadzenie procesu **generalizacji** obiektów ,
- określenie **odwzorowania** w jakim zostaną przedstawione obiekty na płaszczyźnie,
- określenie **układu współrzędnych** do zlokalizowania obiektów w przestrzeni,
- opracowanie **opisu pozaramkowego**.



2.

Dane przestrzenne

Czym są i w jakiej formie
przetwarza się dane
przestrzennych ?



Systemy określania położenia

- Podstawową właściwością każdego opracowania jest skala i poziom szczegółowości danych przestrzennych.

SYSTEM	POZIOM UNIKALNOŚCI	METRYCZNOŚĆ	PRZYKŁAD	ROZDZIELCZOŚĆ PRZESTRZENNA
Nazwa geograficzna	zróznicowany	nie	Warszawa, województwo mazowieckie	zróznicowana
Adres pocztowy	globalny	nie	Aleje Jerozolimskie 4, ulica Marszałkowska 50	wielkość skrzynki na listy
Kod pocztowy	państwowy	nie	00-001, 82-300	obszar odpowiadający numerowi
Prefiks	państwowy	nie	+48 , +49, +39	obszar odpowiadający numerowi
Długość/ Szerokość geograficzna	globalny	tak	52°14'N 21°1'E	nieskończenie dokładna
Układ współrzędnych	zróznicowany	tak	637901.838368, 489597.35946	nieskończenie dokładna

Metody zasilenia baz danych przestrzennych

UŻYTKOWNICY

POMIAR

MIGRACJA
DANYCH

DIGITALIZACJA

KLASYCZNY

Pomiar wykonywany w terenie z wykorzystaniem przyrządów mierniczych (np. tachimetr, odbiornik GPS). Osoba pomiarowa ma bezpośredni kontakt z mierzonym obiektem.

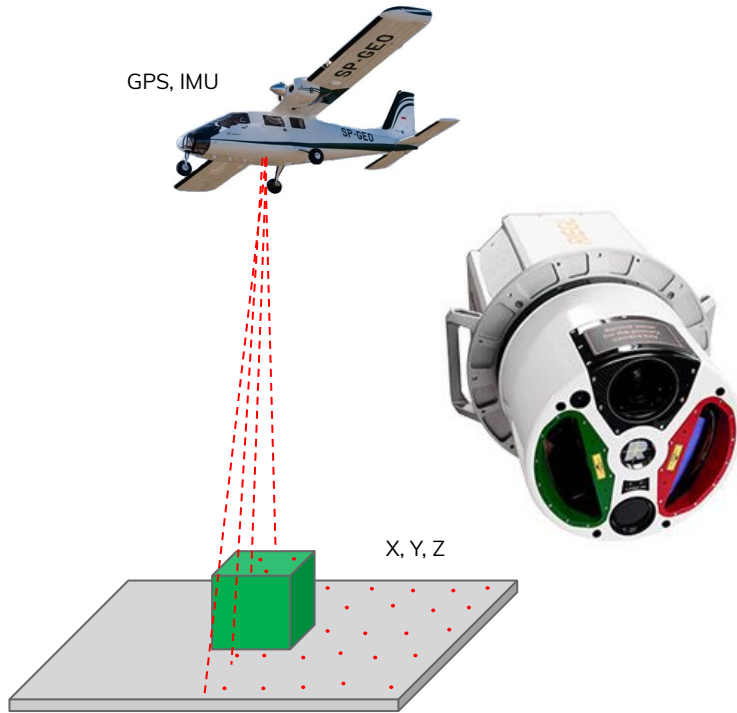


ZDALNY

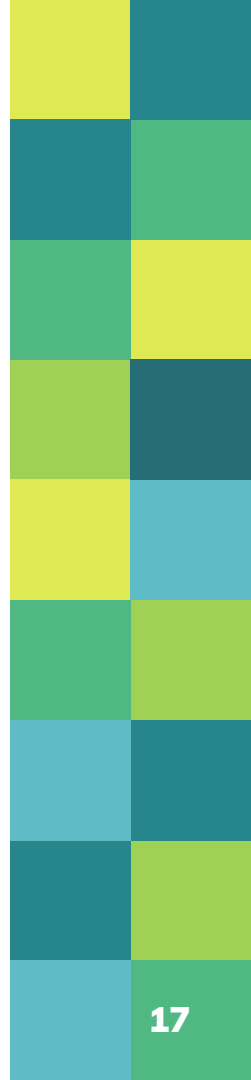
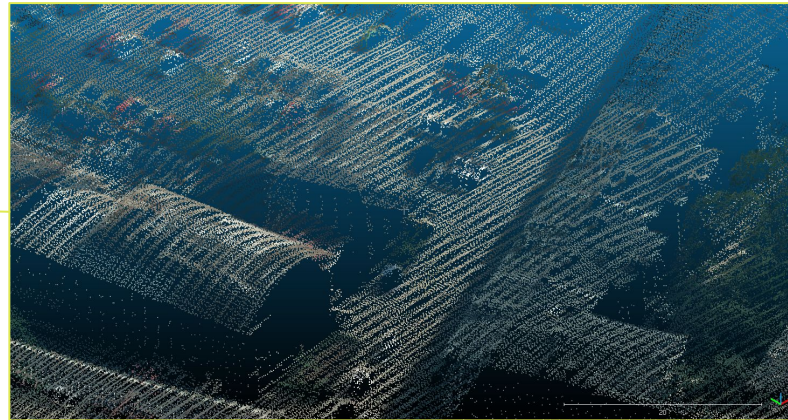
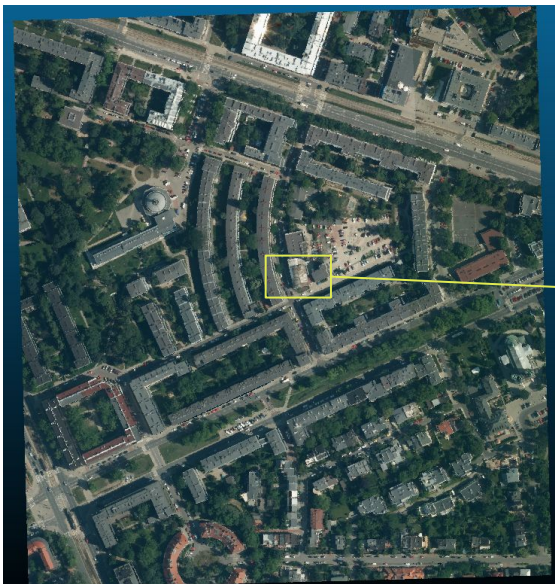
Pomiar wykonywany z wykorzystaniem sensorów zainstalowanych na pokładzie jednostki pomiarowej. Możemy wyróżnić sensory pracujące aktywnie (np. LIDAR) oraz pasywnie (np. kamera lotnicza).

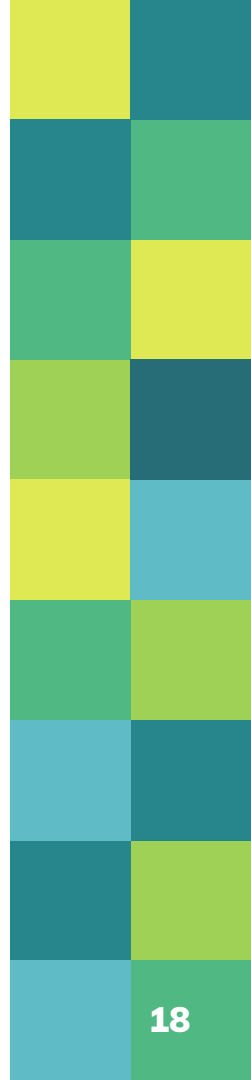
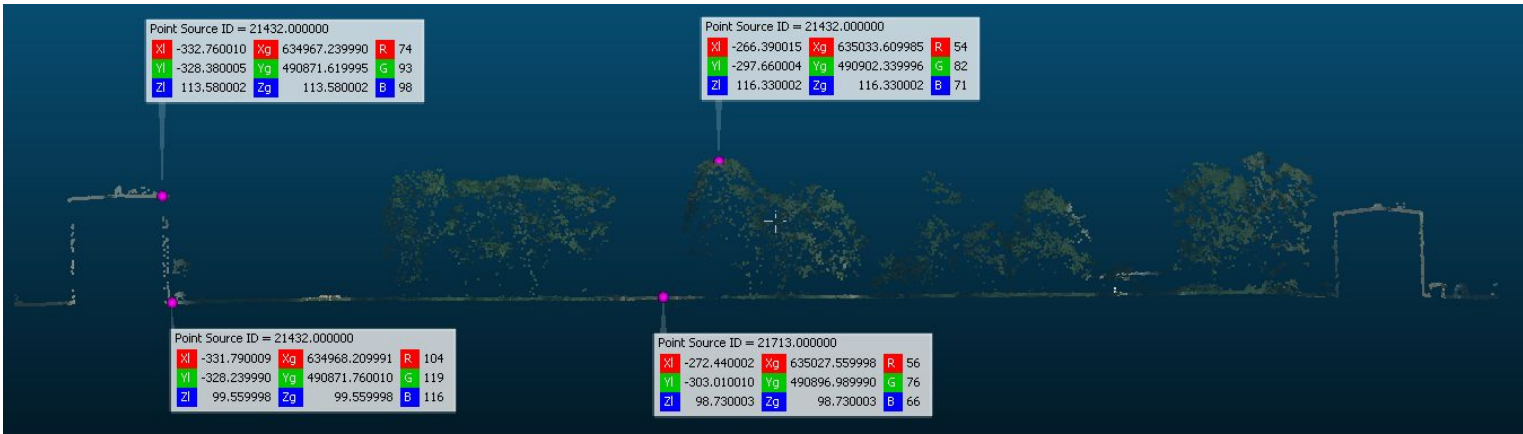
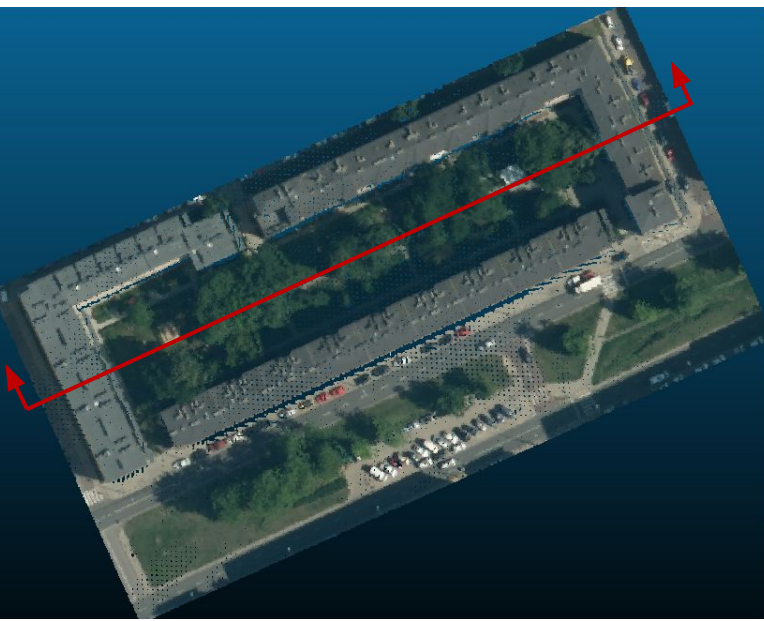


LIDAR - Light Detection and Ranging

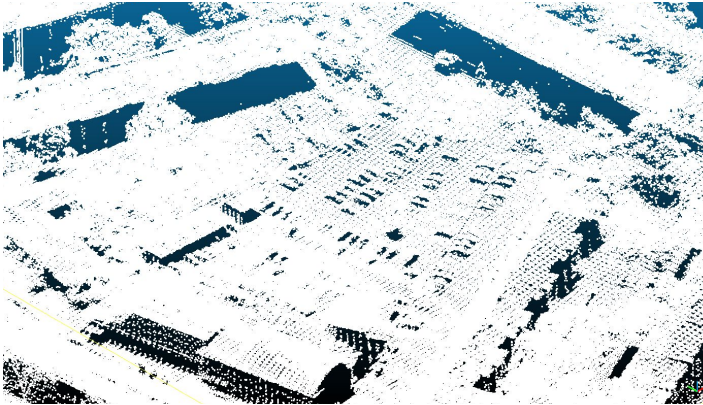


LIDAR to technologia wykorzystująca odbicie światła lasera do pomiaru odległości. Lotniczy skaner laserowy pozyskuje bardzo duże ilości danych, które tworzą **chmurę punktów**. Stanowi ona dokładną reprezentację powierzchni terenu oraz dobrze przenika przez roślinność, aby dotrzeć do gruntu. Chmura punktów używana jest przy tworzeniu NMT i NMPT oraz wysokościowych profili terenów. Chmura punktów zyskuje jeszcze większą wartość po **klasyfikacji obiektowej** (budynki, woda, poziomy wegetacji). Pozwala to na tworzenie obiektów wektorowych np. modeli miast 3D.



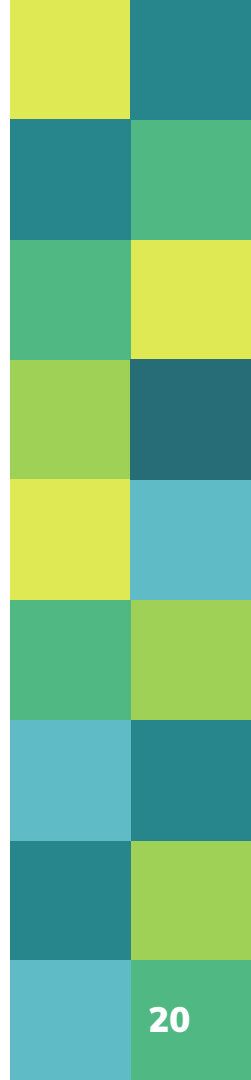
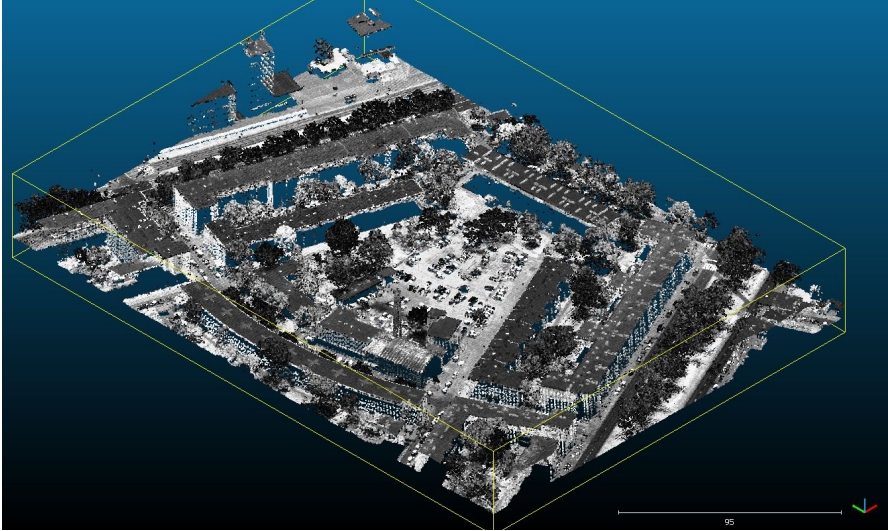


LIDAR - atrybuty

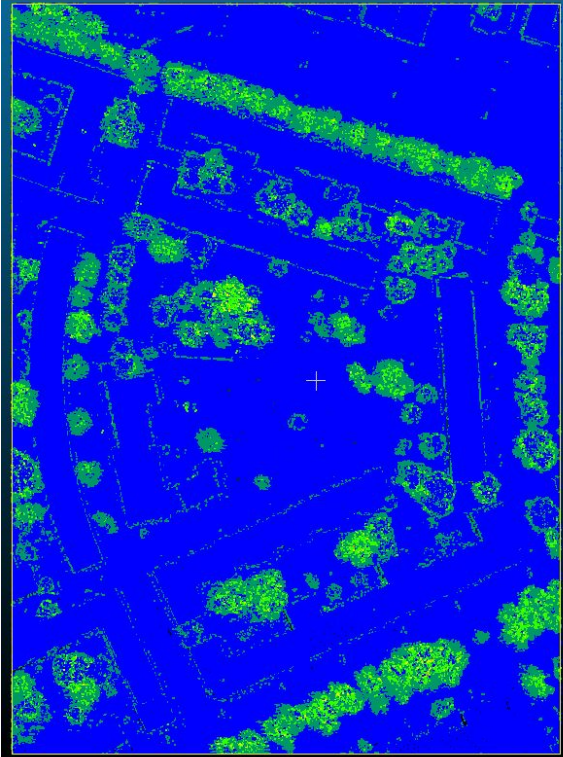
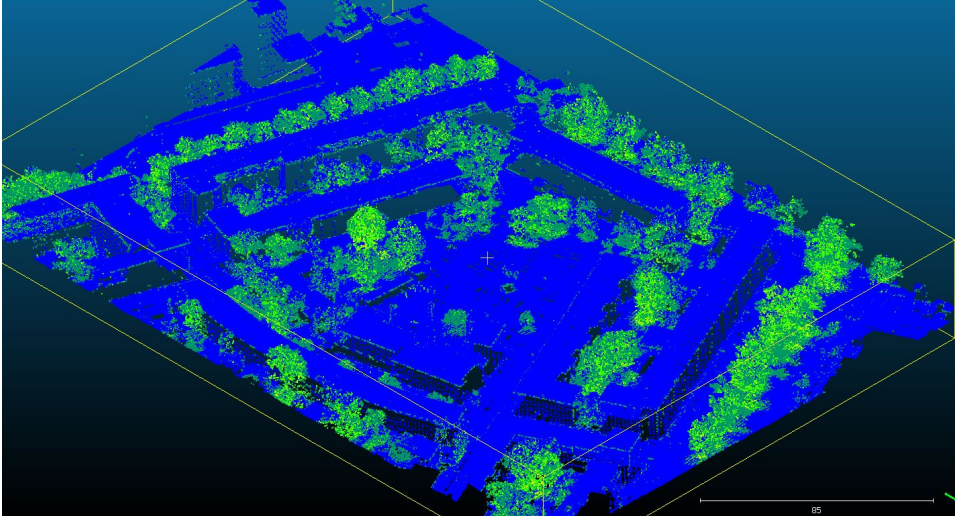


Lp.	Nazwa atrybutu
1	X
2	Y
3	Z
4	Intensity
5	Return Number
6	Number of Returns
7	Scan Direction
8	Edge of Flight Line
9	Classification
10	Scan Angle Rank
11	User Data
12	Point Source ID
13	GPS Time

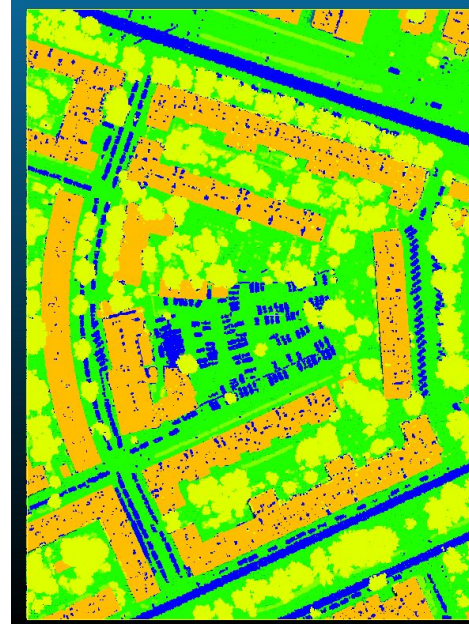
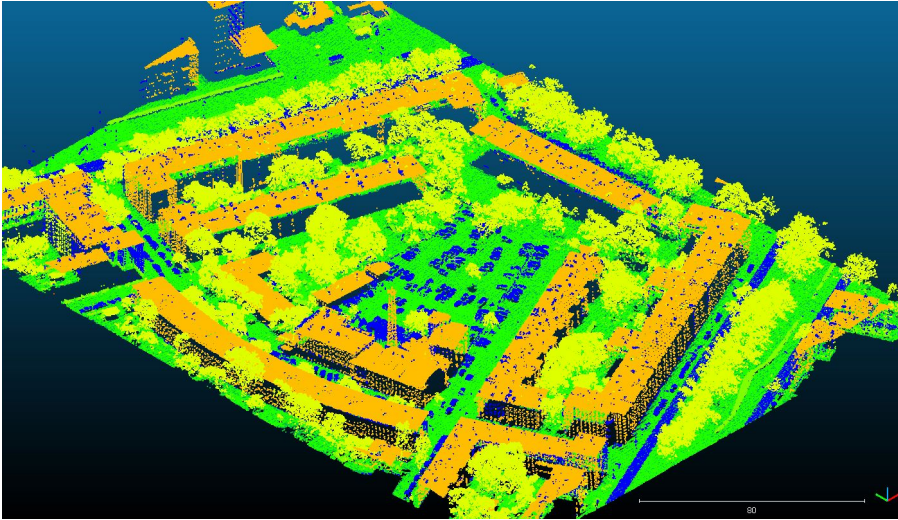
LIDAR - Intensywność (ang. *Intensity*)



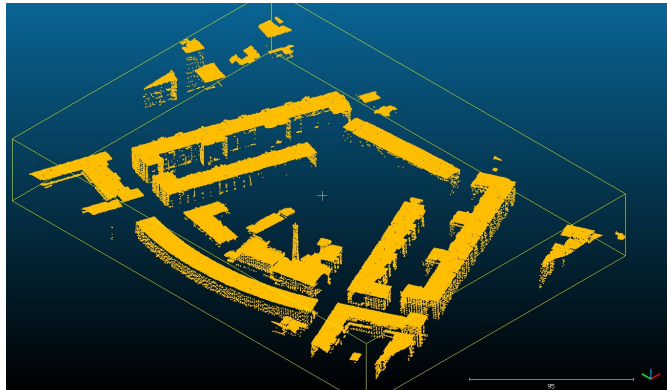
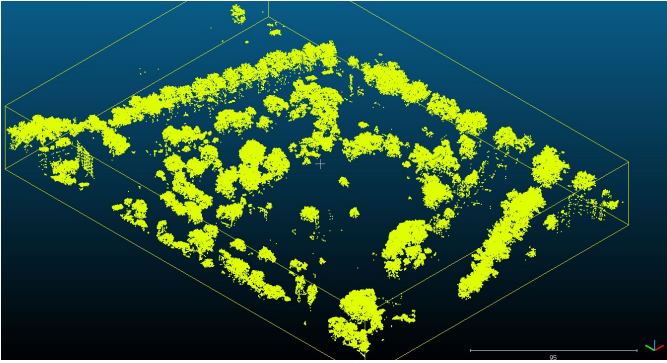
LIDAR - Liczba odbić (ang. *Number of Returns*)



LIDAR - klasyfikacja (ang. *Classification*)

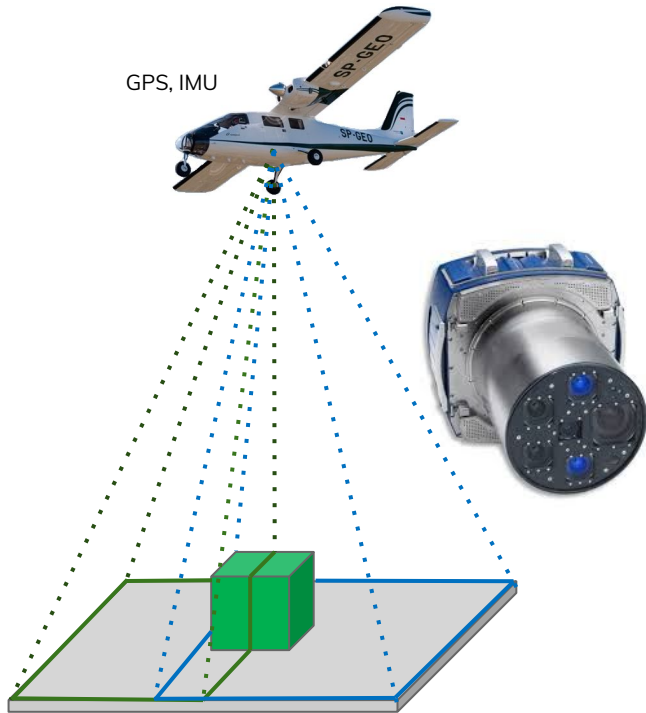


LIDAR - klasyfikacja (ang. *Classification*)



Lp.	Klasa	Nazwa klasy
1	0	Created, never classified
2	1	Unclassified
3	2	Ground
4	3	Low Vegetation
5	4	Medium Vegetation
6	5	High Vegetation
7	6	Building
8	7	Low Point (noise)
9	8	Model Key-point (mass point)
10	9	Water
11	10-11, 13-31	Reserved for ASPRS Definition
12	12	Overlap Points

Zdjęcia lotnicze



Zdjęcie lotnicze to obraz powierzchni Ziemi uzyskany z pokładu samolotu za pomocą kamery lotniczej skierowanej prostopadle do powierzchni Ziemi. Poprzez dopasowanie kamery można uzyskać obraz:

- **w barwach rzeczywistych RGB** - rejestracja obrazu w barwach rzeczywistych, tzn. w zakresie światła widzialnego
- **w bliskiej podczerwieni CIR** - stanowią kompozycję barwną opartą na bliskiej podczerwieni (ang. NIR),
- **termalny** - obrazowanie odbywa się w paśmie podczerwieni termalnej.

Rozdzielczość przestrzenna

Rozdzielczość przestrzenna jest zdefiniowana jako rozmiar piksela na obrazie reprezentujący obszar powierzchnia pomierzony na gruncie.



Rozdzielczość spektralna

Rozdzielczość spektralna definiowana jest przez liczbę kanałów spektralnych i ich szerokości.

(R, G, B)

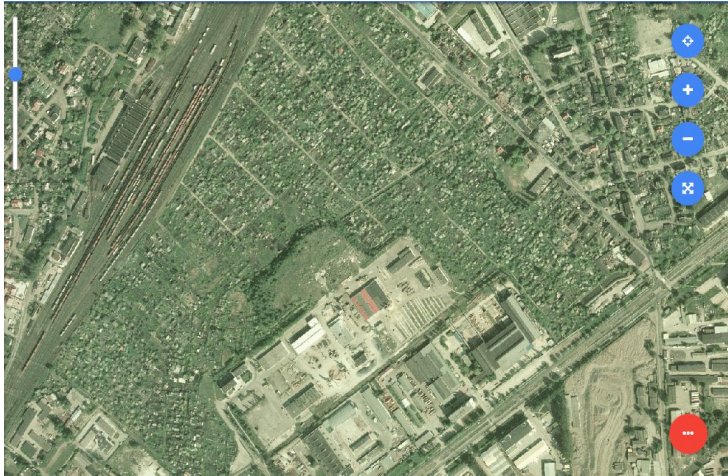


(R, G, B,
CIR)

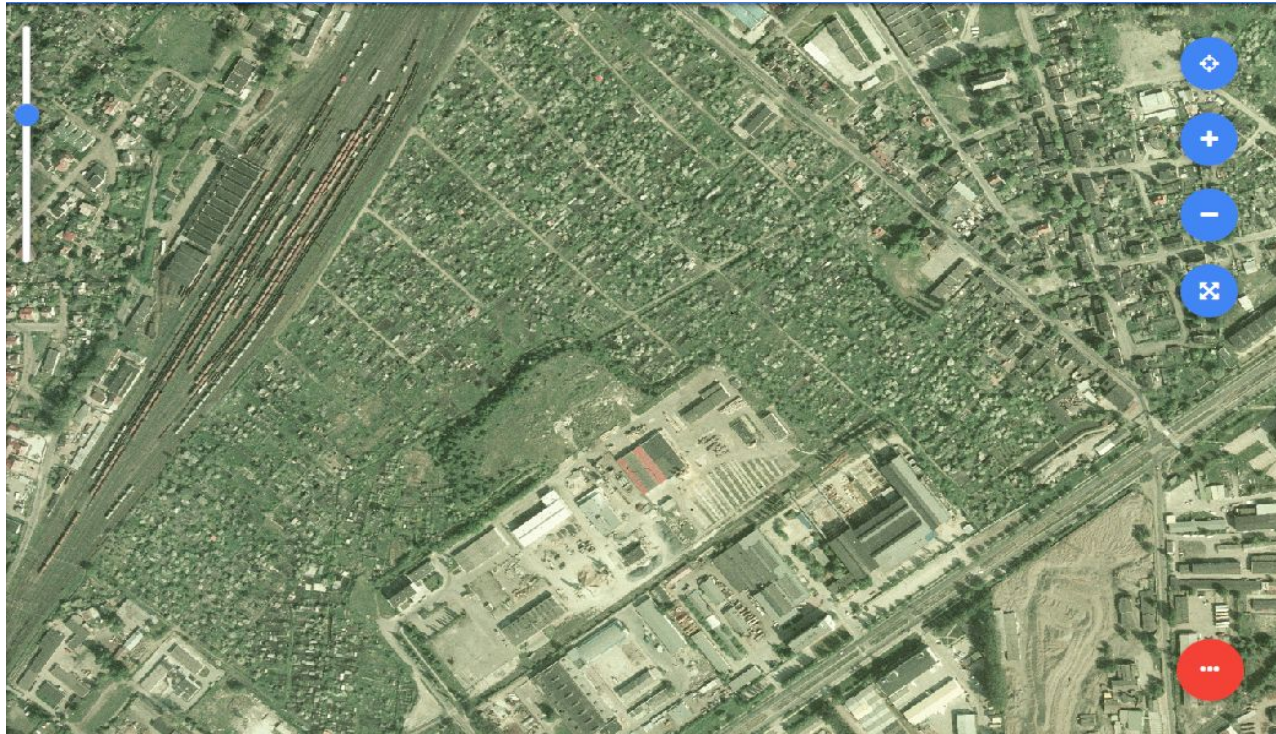


Rozdzielczość czasowa

Rozdzielczość czasowa to okres jaki upływa pomiędzy dwoma identycznymi nalotami nad tym samym punktem. Zwana jest także czasem rewizyty.



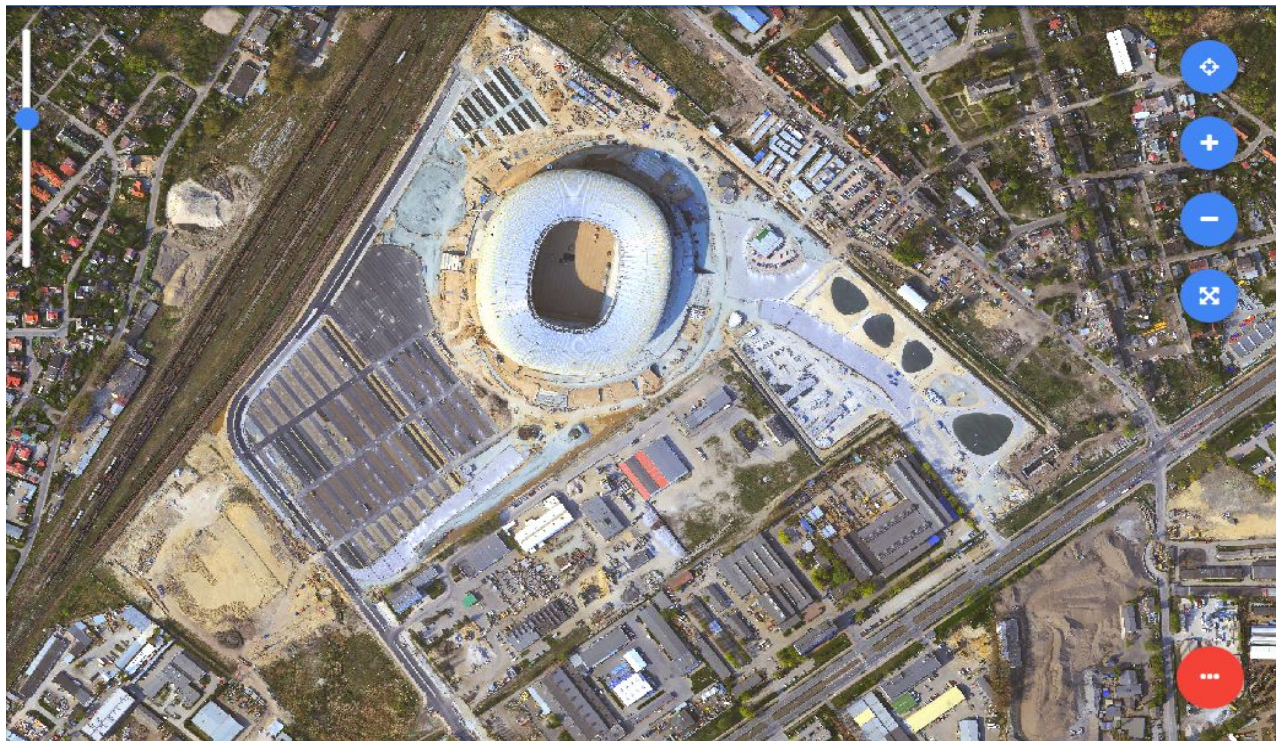
Rozdzielczość czasowa - rok 2000



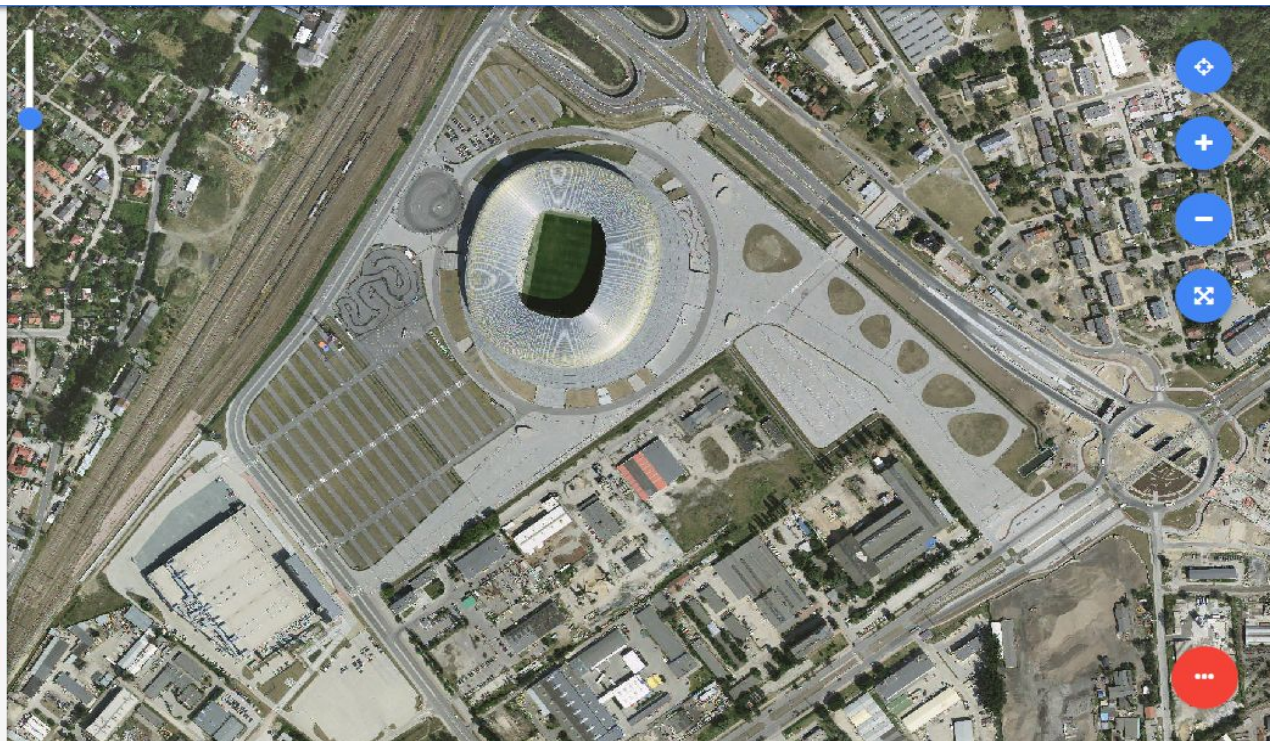
Rozdzielczość czasowa - rok 2008



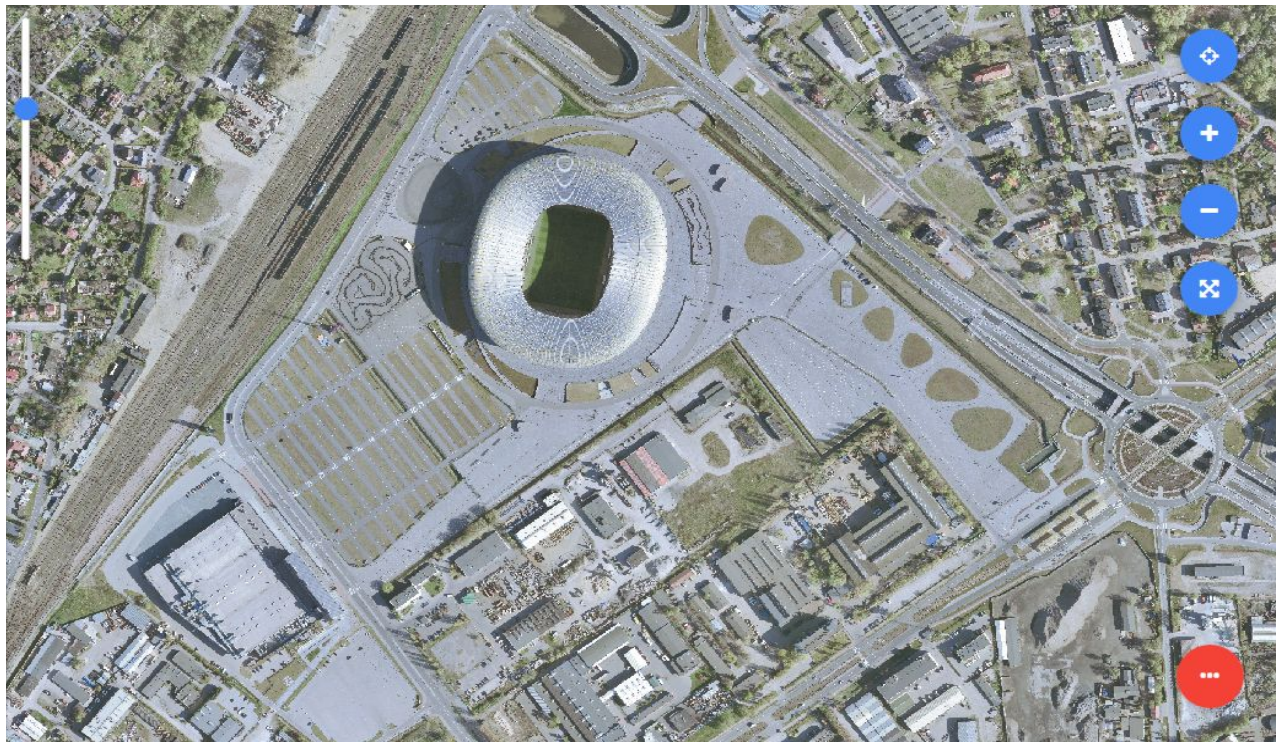
Rozdzielczość czasowa - rok 2011



Rozdzielczość czasowa - rok 2014



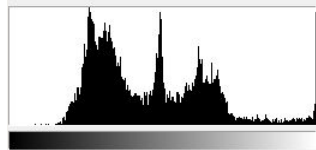
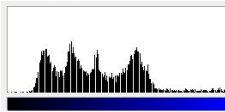
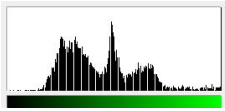
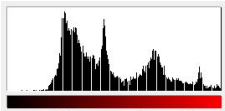
Rozdzielczość czasowa - rok 2016



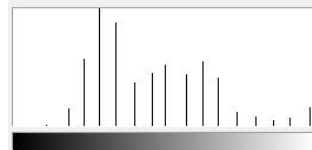
Rozdzielczość radiometryczna

Rozdzielczość radiometryczna mówi o tym, jak szczegółowo mogą być rozróżniane na obrazie różnice w jasnościach. Jest ona mierzona za pomocą liczby odcieni szarości.

$$2^8 = 256$$



$$2^4 = 128$$



$$2^1 = 2$$



4 121

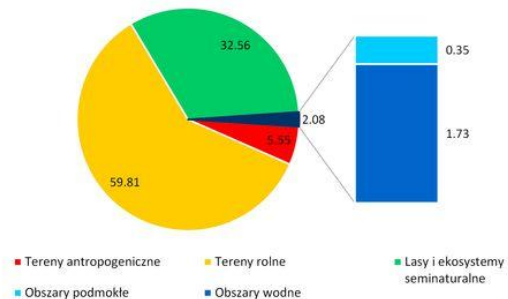


7 804

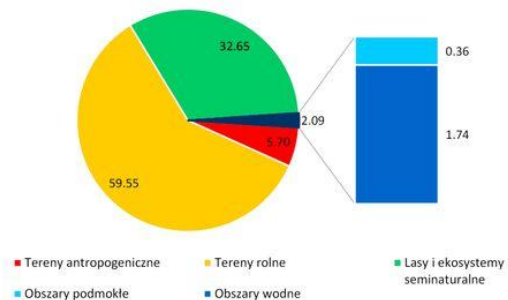
Klasyfikacja zdjęć satelitarnych



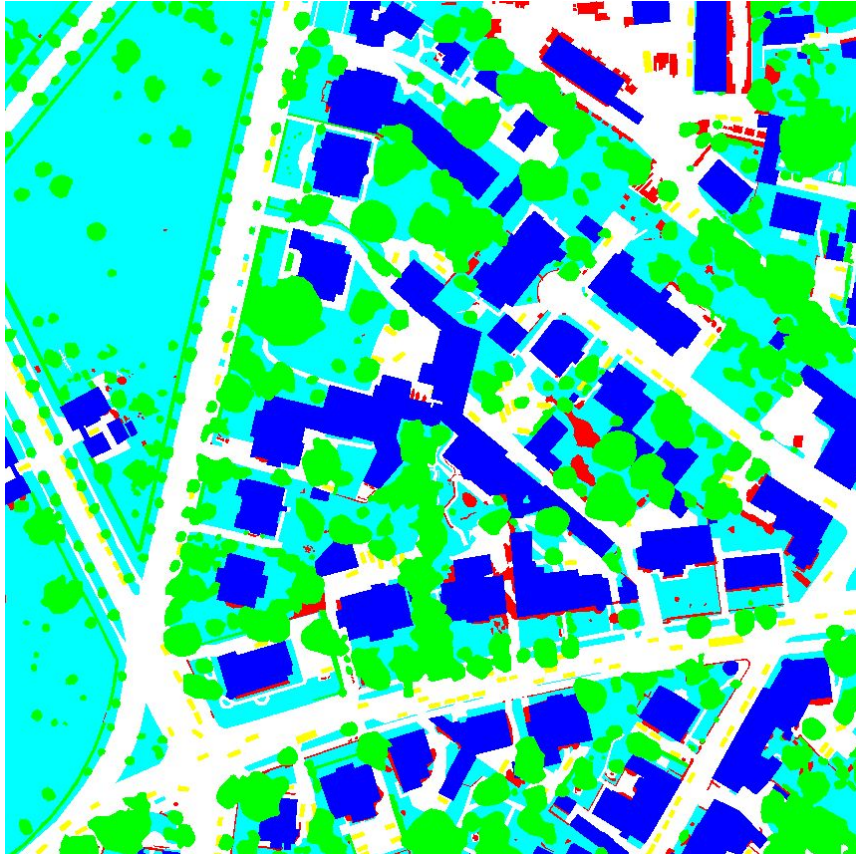
Powierzchnia klas [%] CLC 2006 poziom I - Polska



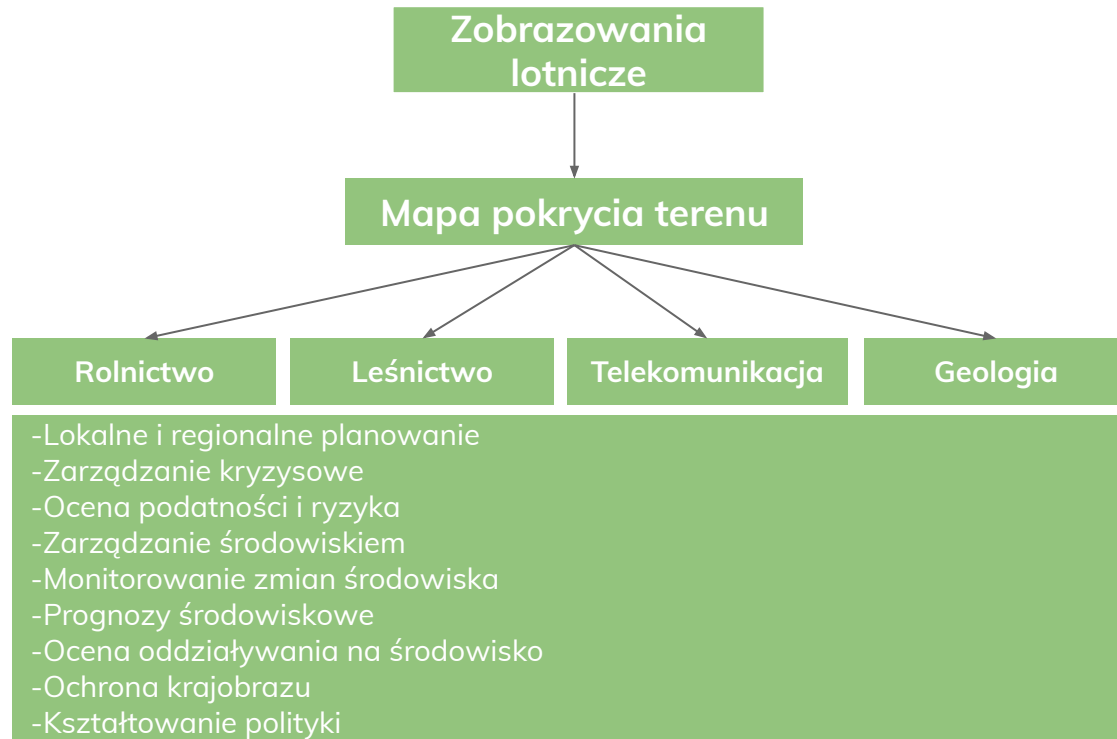
Powierzchnia klas [%] CLC 2012 poziom I - Polska



Klasyfikacja zdjęć lotniczych



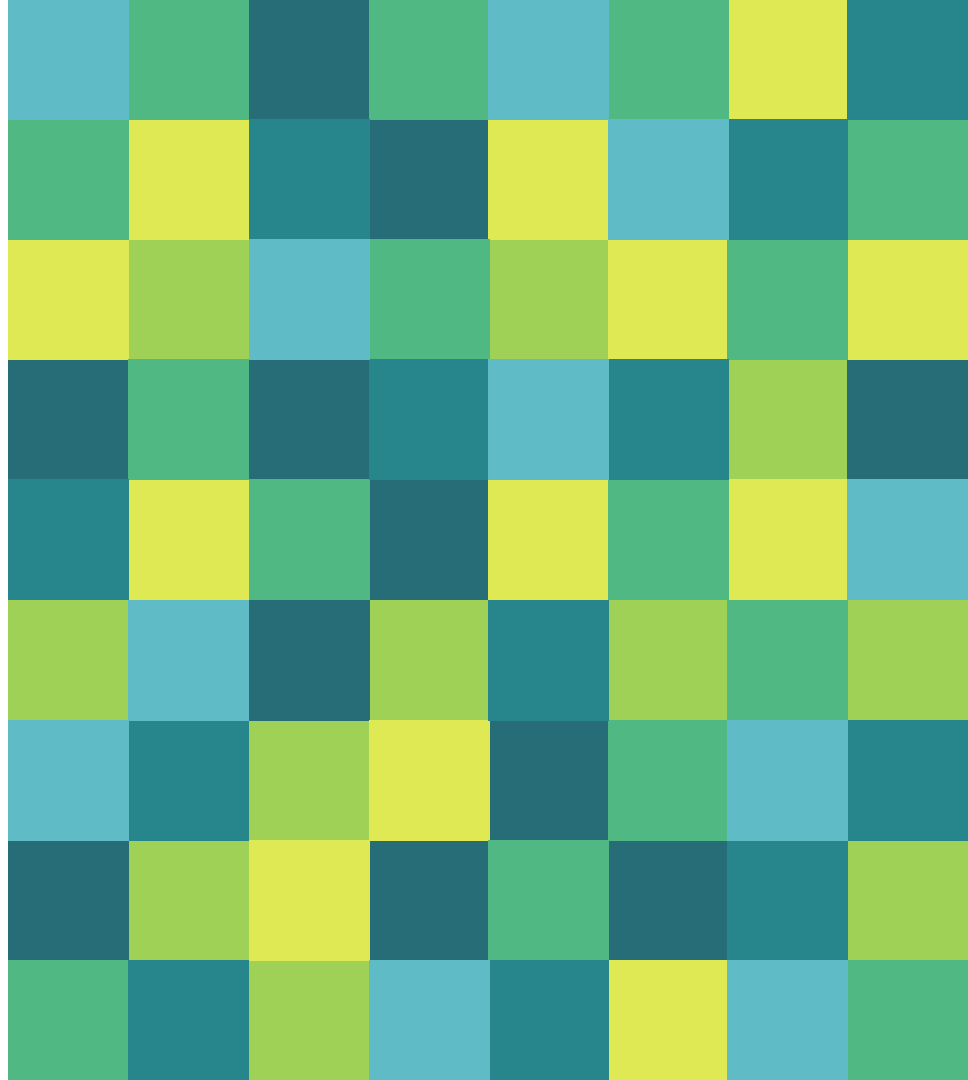
Wykorzystywanie map pokrycia terenu



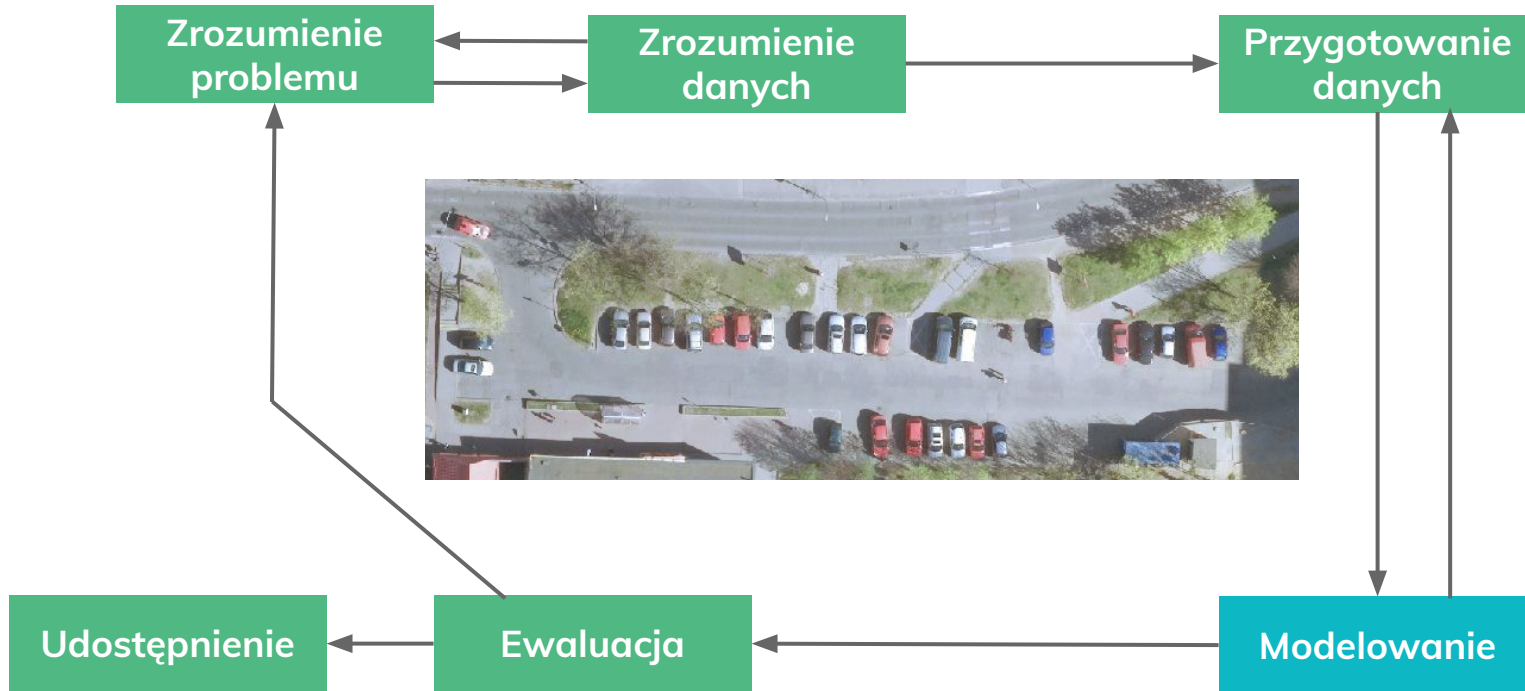
3.

Uczenie maszynowe

Czym jest uczenie maszynowe ?

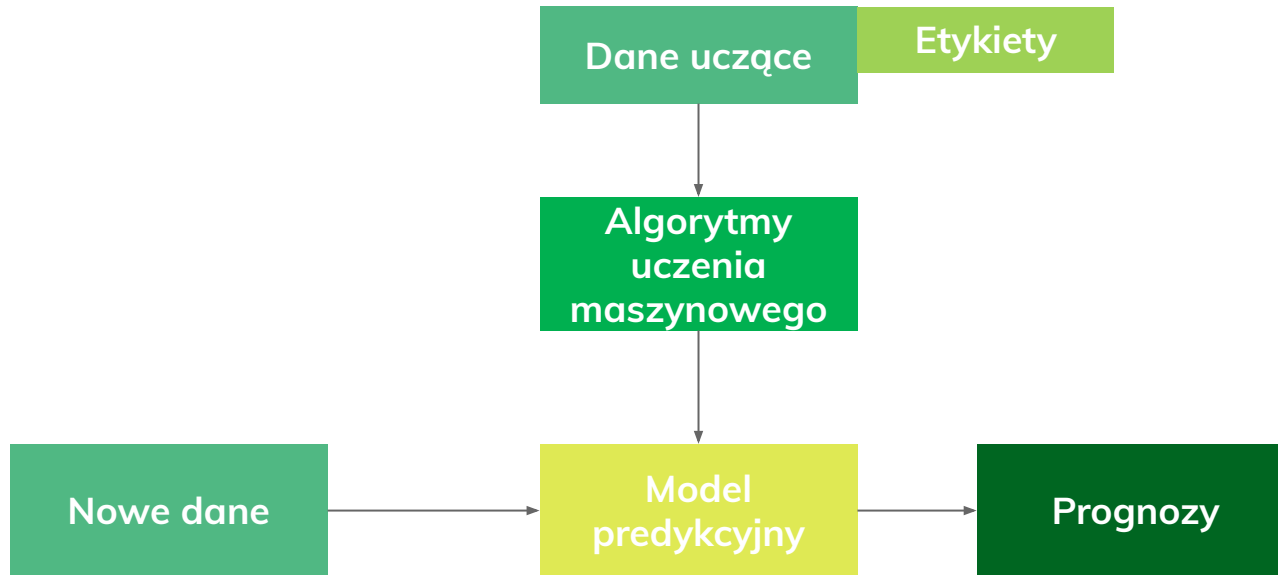


Metodyka CRISP-DM



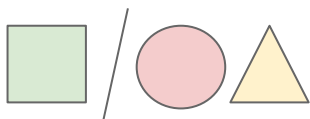
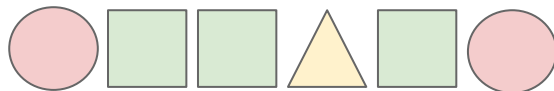
Uczenie maszynowe

- Uczenie nienadzorowane
- Uczenie nadzorowane
- Uczenie przez wzmacnianie

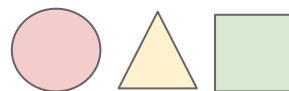


Zagadnienie klasyfikacji

- Klasyfikacja stanowi podkategorię uczenia nadzorowanego służącą do przewidywania etykiet klas w nowych wystąpieniach na podstawie **dotychczasowych obserwacji**.



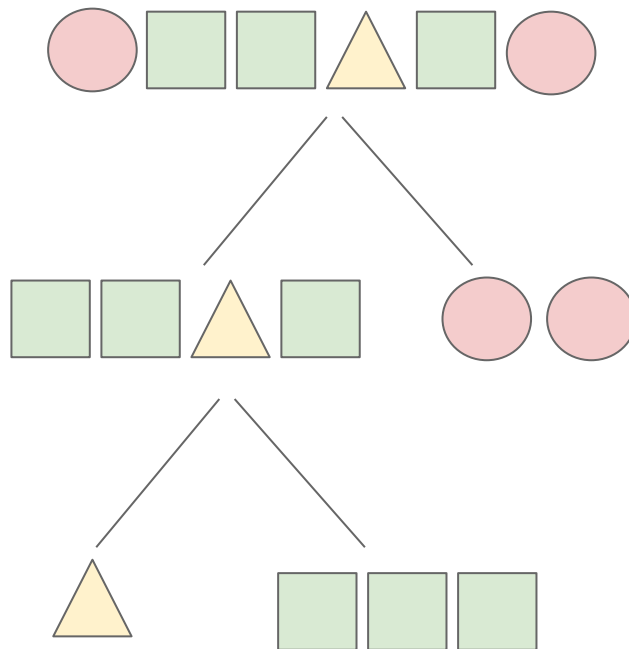
Binarna



Wieloklasowa

Uczenie drzew decyzyjnych

Zasada wykorzystania modeli **drzew decyzyjnych** (ang. decision tree) opiera się na klasyfikacji danych poprzez podejmowanie decyzji na podstawie szeregu odpowiedzi. Na podstawie cech zestawu uczącego model drzewa decyzyjnego wykorzystuje szereg pytań do określania etykiet próbek. Za pomocą algorytmu decyzyjnego tworzymy korzeń drzewa i **rozdzielamy dane wobec cechy mającej największy przyrost informacji**. Poprzez wielokrotne iteracje możemy powtarzać procedurę rozdzielania danych w każdym potomnym **węźle**, aż uzyskamy same **liście**, czyli wszystkie próbki w danym węźle będą przynależą do tej samej klasy. Kluczowym parametrem jest liczba rozgałęzień.



Uczenie drzew decyzyjnych - podział

Podział dokonywany jest poprzez obliczenie funkcji celu **maksymalizującej przyrost informacji** w każdym rozgałęzieniu (ang. information gain):

$$IG(D_p, f) = I(D_p) - \frac{N_{lewy}}{N_p} I(D_{lewy}) - \frac{N_{prawy}}{N_p} I(D_{prawy})$$

D – zestaw danych,

N – liczba danych,

f – cecha podziału,

I – miara zanieczyszczenia

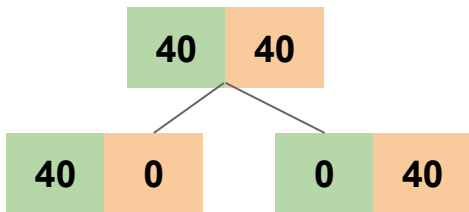
Najpopularniejsze **miary zanieczyszczenia**:

- wskaźnik Giniego $I_G(t) = 1 - \sum_{i=1}^c p(i | t)^2$
- entropia $I_H(t) = - \sum_{i=1}^c p(i | t) \log_2 p(i | t)$
- błąd klasyfikacji $I_E(t) = 1 - \max\{p(i | t)\}$

c – liczba klas,

$p(i | t)$ – stosunek liczby przypadków i – tej klasy do wszystkich przypadków w węźle t ,

Uczenie drzew decyzyjnych - przykład



- Dla zdefiniowanej funkcji celu:

$$IG(D_p, f) = I(D_p) - \frac{N_{lewy}}{N_p} I(D_{lewy}) - \frac{N_{prawy}}{N_p} I(D_{prawy})$$

- oraz przyjętej miary zanieczyszczenia:

$$I_G(t) = 1 - \sum_{i=1}^c p(i | t)^2$$

$$I_G(D_p) = 1 - \left(\left(\frac{40}{80} \right)^2 + \left(\frac{40}{80} \right)^2 \right)$$

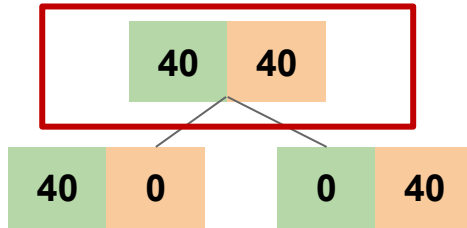
$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$I_G(D_{lewy}) = 1 - \left(\left(\frac{40}{40} \right)^2 + \left(\frac{0}{40} \right)^2 \right) = 0$$

$$I_G(D_{prawy}) = 1 - \left(\left(\frac{0}{40} \right)^2 + \left(\frac{40}{40} \right)^2 \right) = 0$$

$$IG_G = 0,5 - \frac{40}{80} \cdot 0 - \frac{40}{80} \cdot 0 = 0,5$$

Uczenie drzew decyzyjnych - przykład



- Dla zdefiniowanej funkcji celu:

$$IG(D_p, f) = I(D_p) - \frac{N_{lewy}}{N_p} I(D_{lewy}) - \frac{N_{prawy}}{N_p} I(D_{prawy})$$

- oraz przyjętej miary zanieczyszczenia:

$$I_G(t) = 1 - \sum_{i=1}^c p(i | t)^2$$

$$I_G(D_p) = 1 - \left(\left(\frac{40}{80} \right)^2 + \left(\frac{40}{80} \right)^2 \right)$$

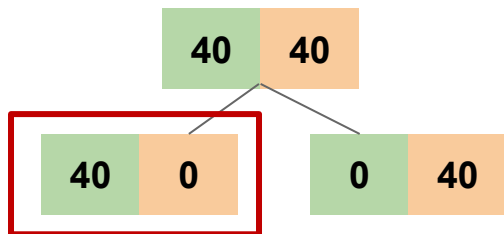
$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$I_G(D_{lewy}) = 1 - \left(\left(\frac{40}{40} \right)^2 + \left(\frac{0}{40} \right)^2 \right) = 0$$

$$I_G(D_{prawy}) = 1 - \left(\left(\frac{0}{40} \right)^2 + \left(\frac{40}{40} \right)^2 \right) = 0$$

$$IG_G = 0,5 - \frac{40}{80} \cdot 0 - \frac{40}{80} \cdot 0 = 0,5$$

Uczenie drzew decyzyjnych - przykład



- Dla zdefiniowanej funkcji celu:

$$IG(D_p, f) = I(D_p) - \frac{N_{lewy}}{N_p} I(D_{lewy}) - \frac{N_{prawy}}{N_p} I(D_{prawy})$$

- oraz przyjętej miary zanieczyszczenia:

$$I_G(t) = 1 - \sum_{i=1}^c p(i | t)^2$$

$$I_G(D_p) = 1 - \left(\left(\frac{40}{80} \right)^2 + \left(\frac{40}{80} \right)^2 \right)$$

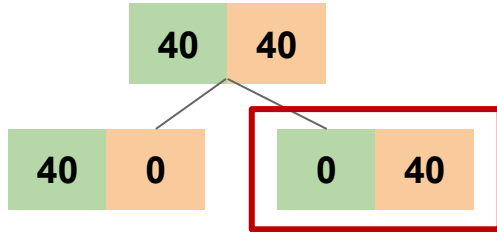
$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$I_G(D_{lewy}) = 1 - \left(\left(\frac{40}{40} \right)^2 + \left(\frac{0}{40} \right)^2 \right) = 0$$

$$I_G(D_{prawy}) = 1 - \left(\left(\frac{0}{40} \right)^2 + \left(\frac{40}{40} \right)^2 \right) = 0$$

$$IG_G = 0,5 - \frac{40}{80} \cdot 0 - \frac{40}{80} \cdot 0 = 0,5$$

Uczenie drzew decyzyjnych - przykład



- Dla zdefiniowanej funkcji celu:

$$IG(D_p, f) = I(D_p) - \frac{N_{lewy}}{N_p} I(D_{lewy}) - \frac{N_{prawy}}{N_p} I(D_{prawy})$$

- oraz przyjętej miary zanieczyszczenia:

$$I_G(t) = 1 - \sum_{i=1}^c p(i | t)^2$$

$$I_G(D_p) = 1 - \left(\left(\frac{40}{80} \right)^2 + \left(\frac{40}{80} \right)^2 \right)$$

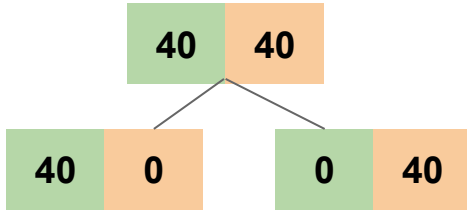
$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$I_G(D_{lewy}) = 1 - \left(\left(\frac{40}{40} \right)^2 + \left(\frac{0}{40} \right)^2 \right) = 0$$

$$I_G(D_{prawy}) = 1 - \left(\left(\frac{0}{40} \right)^2 + \left(\frac{40}{40} \right)^2 \right) = 0$$

$$IG_G = 0,5 - \frac{40}{80} \cdot 0 - \frac{40}{80} \cdot 0 = 0,5$$

Uczenie drzew decyzyjnych - przykład



- Dla zdefiniowanej funkcji celu:

$$IG(D_p, f) = I(D_p) - \frac{N_{lewy}}{N_p} I(D_{lewy}) - \frac{N_{prawy}}{N_p} I(D_{prawy})$$

- oraz przyjętej miary zanieczyszczenia:

$$I_G(t) = 1 - \sum_{i=1}^c p(i | t)^2$$

$$I_G(D_p) = 1 - \left(\left(\frac{40}{80} \right)^2 + \left(\frac{40}{80} \right)^2 \right)$$

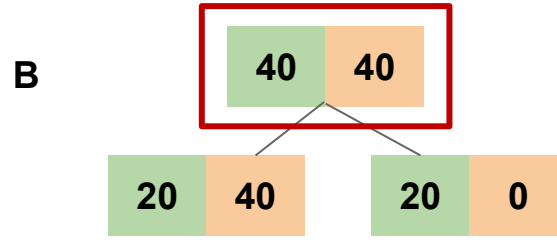
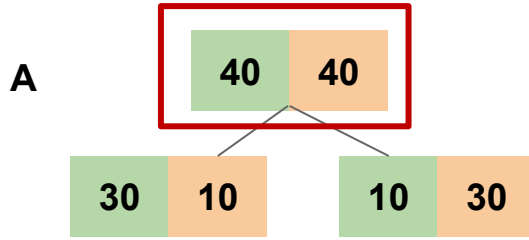
$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$I_G(D_{lewy}) = 1 - \left(\left(\frac{40}{40} \right)^2 + \left(\frac{0}{40} \right)^2 \right) = 0$$

$$I_G(D_{prawy}) = 1 - \left(\left(\frac{0}{40} \right)^2 + \left(\frac{40}{40} \right)^2 \right) = 0$$

$$IG_G = 0,5 - \frac{40}{80} \cdot 0 - \frac{40}{80} \cdot 0 = 0,5$$

Uczenie drzew decyzyjnych - przykład



$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$A : I_G(D_{lewy}) = 1 - ((\frac{3}{4})^2 + (\frac{1}{4})^2) = \frac{3}{8} = 0,375$$

$$B : I_G(D_{lewy}) = 1 - ((\frac{2}{6})^2 + (\frac{4}{6})^2) = \frac{4}{9} = 0,4$$

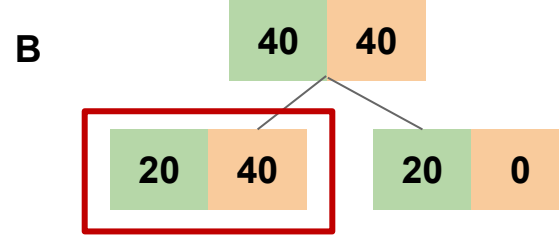
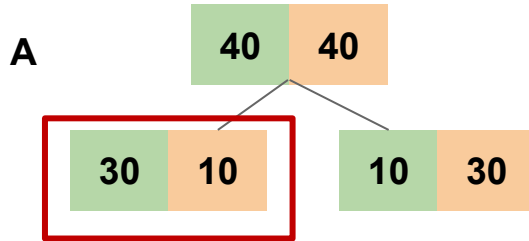
$$A : I_G(D_{prawy}) = 1 - ((\frac{1}{4})^2 + (\frac{3}{4})^2) = \frac{3}{8} = 0,375$$

$$B : I_G(D_{prawy}) = 1 - (1^2 + 0^2) = 0$$

$$A : IG_G = 0,5 - \frac{4}{8}0,375 - \frac{4}{8}0,375 = 0,125$$

$$B : IG_G = 0,5 - \frac{6}{8}0,4 - 0 = 0,16$$

Uczenie drzew decyzyjnych - przykład



$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$A: I_G(D_{lewy}) = 1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = \frac{3}{8} = 0,375$$

$$A: I_G(D_{prawy}) = 1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right) = \frac{3}{8} = 0,375$$

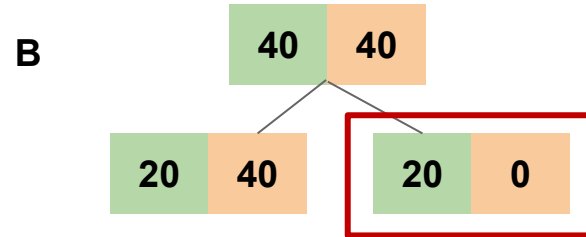
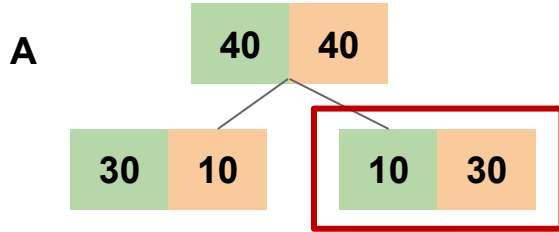
$$A: IG_G = 0,5 - \frac{4}{8}0,375 - \frac{4}{8}0,375 = 0,125$$

$$B: I_G(D_{lewy}) = 1 - \left(\left(\frac{2}{6}\right)^2 + \left(\frac{4}{6}\right)^2\right) = \frac{4}{9} = 0,4$$

$$B: I_G(D_{prawy}) = 1 - (1^2 + 0^2) = 0$$

$$B: IG_G = 0,5 - \frac{6}{8}0,4 - 0 = 0,16$$

Uczenie drzew decyzyjnych - przykład



$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$A : I_G(D_{lewy}) = 1 - \left(\left(\frac{3}{4} \right)^2 + \left(\frac{1}{4} \right)^2 \right) = \frac{3}{8} = 0,375$$

$$A : I_G(D_{prawy}) = 1 - \left(\left(\frac{1}{4} \right)^2 + \left(\frac{3}{4} \right)^2 \right) = \frac{3}{8} = 0,375$$

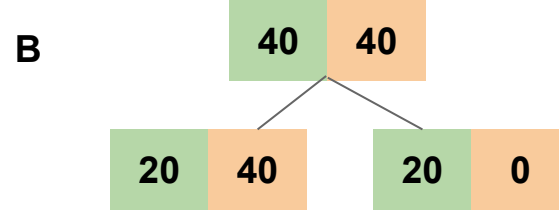
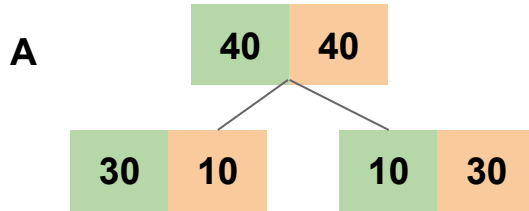
$$A : IG_G = 0,5 - \frac{4}{8}0,375 - \frac{4}{8}0,375 = 0,125$$

$$B : I_G(D_{lewy}) = 1 - \left(\left(\frac{2}{6} \right)^2 + \left(\frac{4}{6} \right)^2 \right) = \frac{4}{9} = 0,4$$

$$B : I_G(D_{prawy}) = 1 - (1^2 + 0^2) = 0$$

$$B : IG_G = 0,5 - \frac{6}{8}0,4 - 0 = 0,16$$

Uczenie drzew decyzyjnych - przykład



$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$A : I_G(D_{lewy}) = 1 - \left(\left(\frac{3}{4} \right)^2 + \left(\frac{1}{4} \right)^2 \right) = \frac{3}{8} = 0,375$$

$$B : I_G(D_{lewy}) = 1 - \left(\left(\frac{2}{6} \right)^2 + \left(\frac{4}{6} \right)^2 \right) = \frac{4}{9} = 0,4$$

$$A : I_G(D_{prawy}) = 1 - \left(\left(\frac{1}{4} \right)^2 + \left(\frac{3}{4} \right)^2 \right) = \frac{3}{8} = 0,375$$

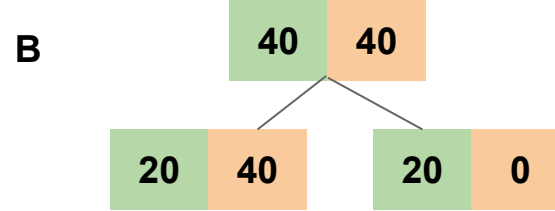
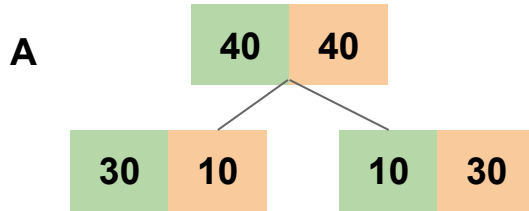
$$B : I_G(D_{prawy}) = 1 - (1^2 + 0^2) = 0$$

$$A : IG_G = 0,5 - \frac{4}{8}0,375 - \frac{4}{8}0,375 = 0,125$$

$$B : IG_G = 0,5 - \frac{6}{8}0,4 - 0 = 0,16$$

$$IG(D_p, f) = I(D_p) - \frac{N_{lewy}}{N_p} I(D_{lewy}) - \frac{N_{prawy}}{N_p} I(D_{prawy})$$

Uczenie drzew decyzyjnych - przykład



$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$A: I_G(D_{lewy}) = 1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = \frac{3}{8} = 0,375$$

$$B: I_G(D_{lewy}) = 1 - \left(\left(\frac{2}{6}\right)^2 + \left(\frac{4}{6}\right)^2\right) = \frac{4}{9} = 0,4$$

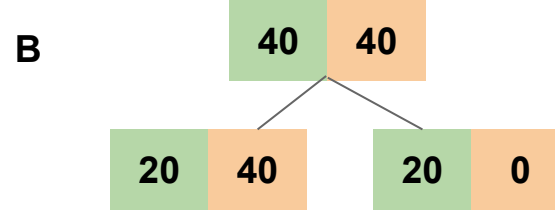
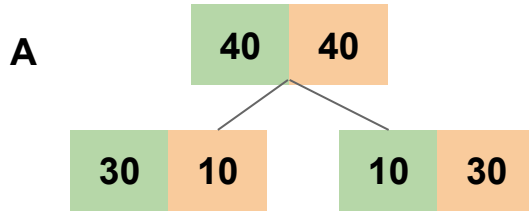
$$A: I_G(D_{prawy}) = 1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right) = \frac{3}{8} = 0,375$$

$$B: I_G(D_{prawy}) = 1 - (1^2 + 0^2) = 0$$

$$A: IG_G = 0,5 - \frac{4}{8}0,375 - \frac{4}{8}0,375 = 0,125$$

$$B: IG_G = 0,5 - \frac{6}{8}0,4 - 0 = 0,16$$

Uczenie drzew decyzyjnych - przykład



$$I_G(D_p) = 1 - (0,5^2 + 0,5^2) = 0,5$$

$$A: I_G(D_{lewy}) = 1 - \left(\left(\frac{3}{4}\right)^2 + \left(\frac{1}{4}\right)^2\right) = \frac{3}{8} = 0,375$$

$$B: I_G(D_{lewy}) = 1 - \left(\left(\frac{2}{6}\right)^2 + \left(\frac{4}{6}\right)^2\right) = \frac{4}{9} = 0,4$$

$$A: I_G(D_{prawy}) = 1 - \left(\left(\frac{1}{4}\right)^2 + \left(\frac{3}{4}\right)^2\right) = \frac{3}{8} = 0,375$$

$$B: I_G(D_{prawy}) = 1 - (1^2 + 0^2) = 0$$

$$A: IG_G = 0,5 - \frac{4}{8}0,375 - \frac{4}{8}0,375 = 0,125$$

$$B: IG_G = 0,5 - \frac{6}{8}0,4 - 0 = 0,16$$

“obliczenie funkcji celu **maksymalizującej przyrost informacji** w każdym rozgałęzieniu ”

Algorytm Lasów Losowych

Metoda **lasów losowych** (ang. *random forest*) intuicyjnie możemy rozumieć jako **zespół drzew decyzyjnych**. Koncepcja kryjąca się za uczeniem zespołów polega na łączeniu słabych klasyfikatorów (ang. *Weak learners*) w jeden skuteczny model - silny klasyfikator (ang. *Strong learner*), mający mniejszy błąd uogólniania oraz wykazujący niższą wrażliwość na przetrenowanie.

Algorytm Lasów Losowych

Metoda **lasów losowych** (ang. *random forest*) intuicyjnie możemy rozumieć jako **zespół drzew decyzyjnych**. Koncepcja kryjąca się za uczeniem zespołów polega na łączeniu słabych klasyfikatorów (ang. *Weak learners*) w jeden skuteczny model - silny klasyfikator (ang. *Strong learner*), mający mniejszy błąd uogólniania oraz wykazujący niższą wrażliwość na przetrenowanie.

Algorytm losowego lasu można rozpisać na cztery proste etapy:

1. Wprowadź losowanie n próbek początkowych (ang. *bootstrap*; losowo dobierz n próbek z zestawu uczącego i wstaw za nie próbki zastępcze).

Algorytm Lasów Losowych

Metoda **lasów losowych** (ang. *random forest*) intuicyjnie możemy rozumieć jako **zespół drzew decyzyjnych**. Koncepcja kryjąca się za uczeniem zespołów polega na łączeniu słabych klasyfikatorów (ang. *Weak learners*) w jeden skuteczny model - silny klasyfikator (ang. *Strong learner*), mający mniejszy błąd uogólniania oraz wykazujący niższą wrażliwość na przetrenowanie.

Algorytm losowego lasu można rozpisać na cztery proste etapy:

1. Wprowadź losowanie n próbek początkowych (ang. *bootstrap*; losowo dobierz n próbek z zestawu uczącego i wstaw za nie próbki zastępcze).
2. Wygeneruj drzewo decyzyjne na podstawie próbek początkowych. W każdym węźle:
 - a. Dobierz losowo d cech i nie zastępuj ich innymi.
 - b. Rozdziel węzeł za pomocą cechy gwarantującej najlepsze rozgałęzienie pod kątem funkcji celu (np. Maksymalizując przyrost informacji).

Algorytm Lasów Losowych

Metoda **lasów losowych** (ang. *random forest*) intuicyjnie możemy rozumieć jako **zespół drzew decyzyjnych**. Koncepcja kryjąca się za uczeniem zespołów polega na łączeniu słabych klasyfikatorów (ang. *Weak learners*) w jeden skuteczny model - silny klasyfikator (ang. *Strong learner*), mający mniejszy błąd uogólniania oraz wykazujący niższą wrażliwość na przetrenowanie.

Algorytm losowego lasu można rozpisać na cztery proste etapy:

1. Wprowadź losowanie n próbek początkowych (ang. *bootstrap*; losowo dobierz n próbek z zestawu uczącego i wstaw za nie próbki zastępcze).
2. Wygeneruj drzewo decyzyjne na podstawie próbek początkowych. W każdym węźle:
 - a. Dobierz losowo d cech i nie zastępuj ich innymi.
 - b. Rozdziel węzeł za pomocą cechy gwarantującej najlepsze rozgałęzienie pod kątem funkcji celu (np. Maksymalizując przyrost informacji).
3. Powtórz kroki 2. i 3. k -krotnie

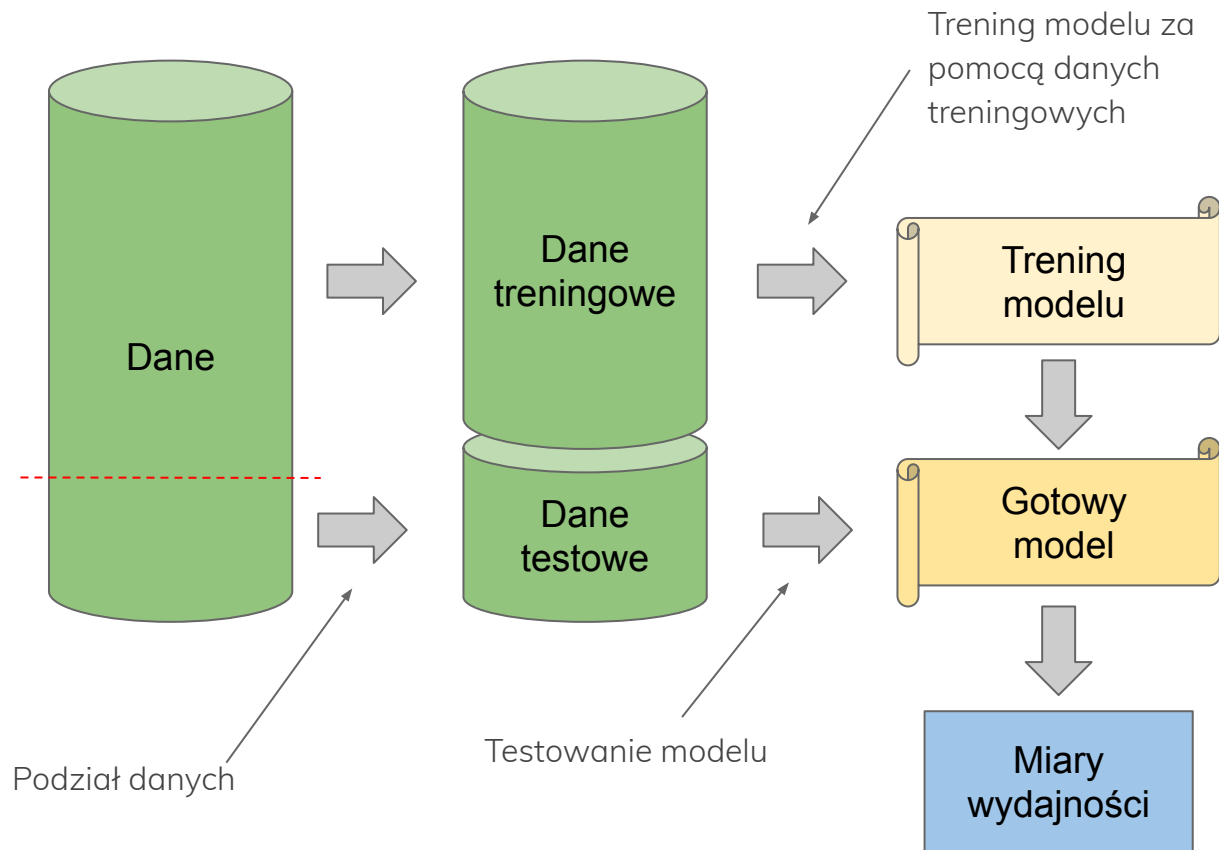
Algorytm Lasów Losowych

Metoda **lasów losowych** (ang. *random forest*) intuicyjnie możemy rozumieć jako **zespół drzew decyzyjnych**. Koncepcja kryjąca się za uczeniem zespołów polega na łączeniu słabych klasyfikatorów (ang. *Weak learners*) w jeden skuteczny model - silny klasyfikator (ang. *Strong learner*), mający mniejszy błąd uogólniania oraz wykazujący niższą wrażliwość na przetrenowanie.

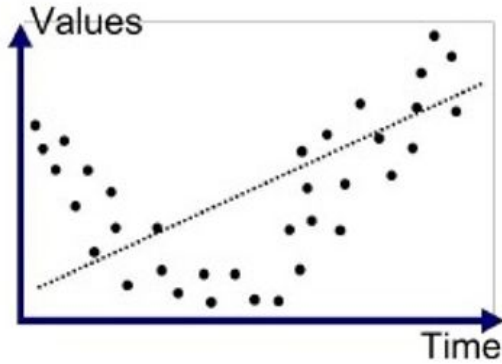
Algorytm losowego lasu można rozpisać na cztery proste etapy:

1. Wprowadź losowanie n próbek początkowych (ang. *bootstrap*; losowo dobierz n próbek z zestawu uczącego i wstaw za nie próbki zastępcze).
2. Wygeneruj drzewo decyzyjne na podstawie próbek początkowych. W każdym węźle:
 - a. Dobierz losowo d cech i nie zastępuj ich innymi.
 - b. Rozdziel węzeł za pomocą cechy gwarantującej najlepsze rozgałęzienie pod kątem funkcji celu (np. Maksymalizując przyrost informacji).
3. Powtórz kroki 2. i 3. k -krotnie
4. Zbierz prognozy otrzymane z każdego drzewa i przydzielaj próbkom etykiety klas poprzez **większościowe głosowanie**.

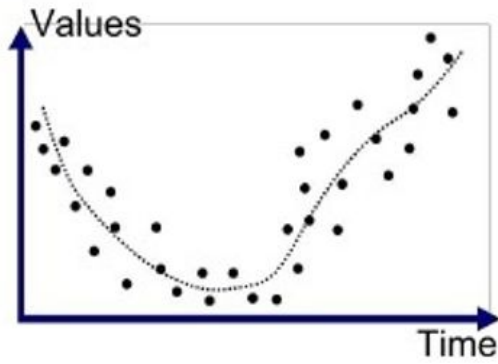
Podział zbioru danych



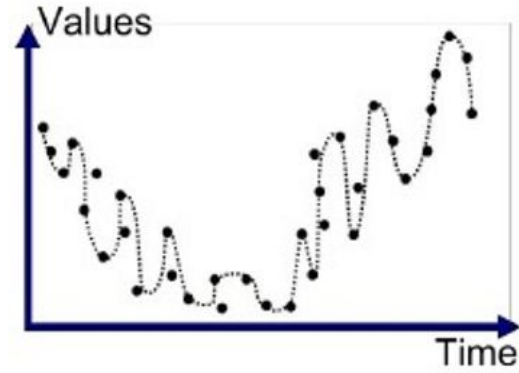
Przeuczenie i niedouczenie modelu



Underfitted



Good Fit/Robust



Overfitted

Rodzaje błędów

PP - prawdziwie pozytywne - próbki zaklasyfikowane poprawnie do docelowej klasy

PN - prawdziwie negatywne - próbki zaklasyfikowanie poprawnie do innej klasy niż docelowa

FP - fałszywie pozytywne - próbki zaklasyfikowane błędnie zaklasyfikowane do docelowej klasy (błąd pierwszego rodzaju)

FN - fałszywie negatywne - próbki błędnie zaklasyfikowane, które powinny zostać zaklasyfikowane w docelowej klasie (błąd drugiego rodzaju)

Rzeczywiście

		Warunki pozytywne	Warunki negatywne
Predykcja	Warunki pozytywne	PP	FP ← błąd pierwszego rodzaju
	Warunki negatywne	FN ← błąd drugiego rodzaju	PN

Miary wydajności

Macierz pomyłek (ang. Confusion matrix).

Polega na ilościowym porównaniu wyników predykcji z docelowymi klasami.

Każdy rząd macierzy reprezentuje predykcje klas, natomiast kolumny reprezentują stan faktyczny.

Macierz pomyłek pozwala na weryfikację jaki odsetek próbek został zaklasyfikowany do innej klasy niż docelowa w kontekście wieloklasowym.

		Rzeczywiście			
		Water	Forest	Urban	Buildings
Predykcja	Water	0.9	0.05	0.025	0.025
	Forest	0.2	0.7	0.1	0.0
	Urban	0.2	0.18	0.75	0.05
	Buildings	0.2	0.07	0.05	0.67

Miary wydajności

Precyzja (Precision)

to zdolność klasyfikatora do nie oznaczania instancji jako pozytywnej, która jest w rzeczywistości negatywna. Dla wszystkich przypadków sklasyfikowanych jako pozytywne, jaki procent był prawdziwy.

Jaki odsetek pozytywnych przypadków był prawdziwy?

$$\text{Precyzja} = \frac{PP}{PP+FP}$$

Pełność (Recall)

to zdolność do oznaczania pozytywnych instancji. Dla każdej klasy zdefiniowany jako stosunek wyników prawdziwych do sumy wyników prawdziwych i fałszywych.

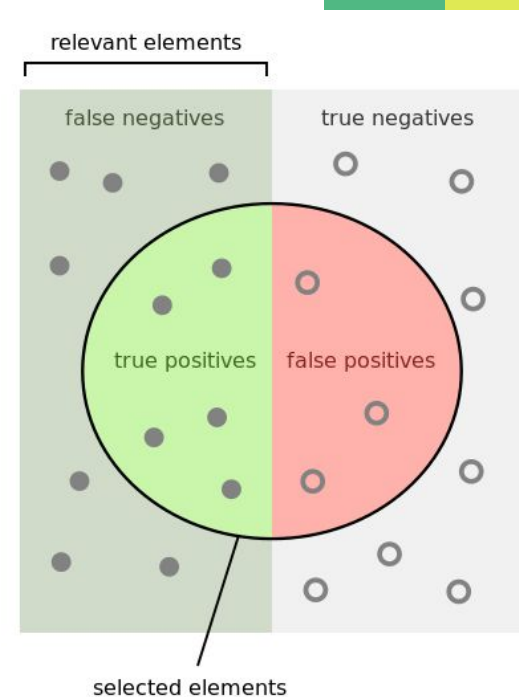
Jaki odsetek pozytywnych przypadków był prawdziwy spośród tych które były oznaczone jako prawdziwo pozytywne oraz tych które powinny należeć do klasy docelowej (PP)?

$$\text{Pełność} = \frac{PP}{PP+FN}$$

Wynik F1 (F1-score)

Wynik F1 jest ważoną harmoniczną średnią precyzji i pełności, najlepszy wynik to 1.0, najgorszy to 0.0. Zgodnie z zasadą średnia ważona F1 powinna być używana do porównywania modeli klasyfikatorów a nie do globalnej dokładności.

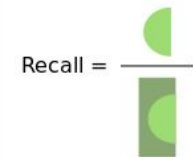
$$F1 = \frac{PP}{PP + \frac{FN+FP}{2}}$$



How many selected items are relevant?



How many relevant items are selected?



Ćwiczenie 3. A

Klasyfikacja chmury punktów na podstawie cech geometrycznych.

Proszę wejść do folderu:

`./cwiczenia_LIDAR/LasyLosowe`

Ćwiczenie 3. A

Klasyfikacja zdjęć lotniczych za pomocą modelu lasów losowych.

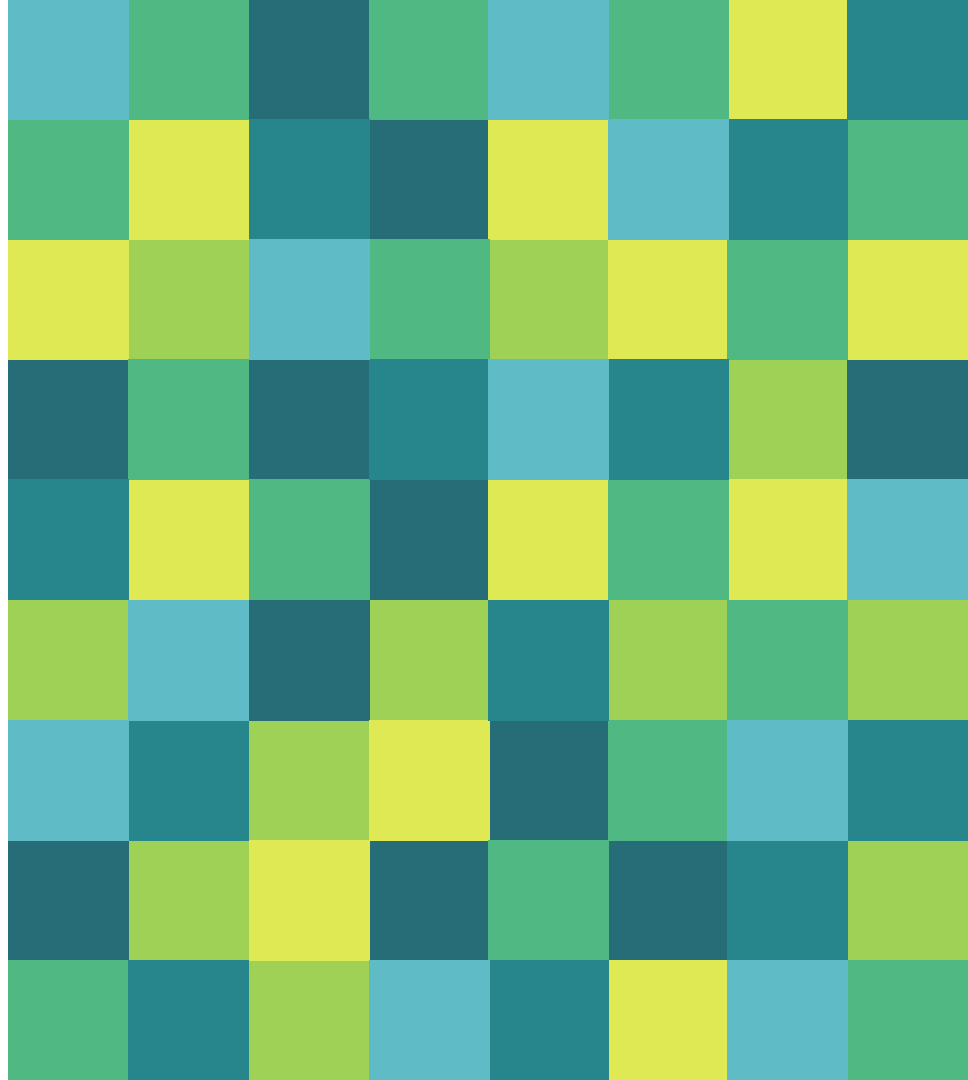
Proszę wejść do folderu:

`./cwiczenia_Imagery/LasyLosowe`

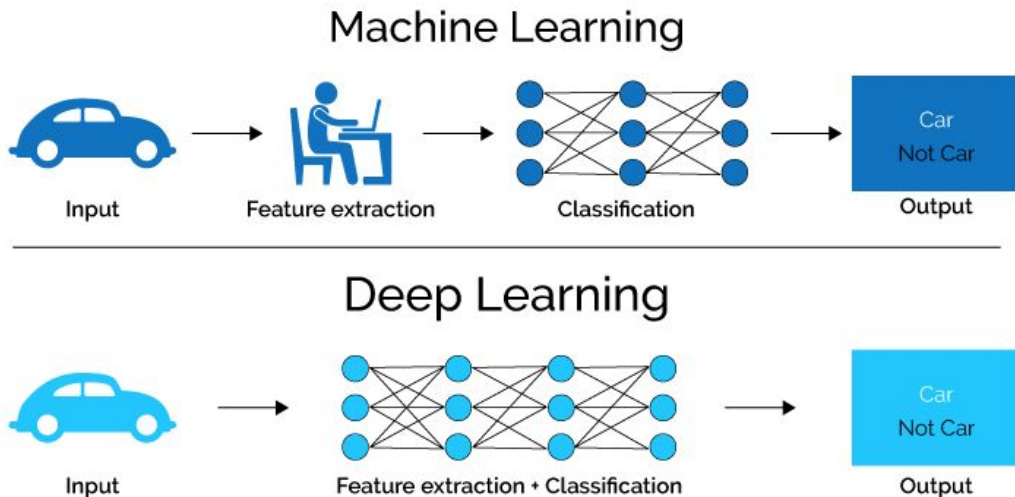
4.

Uczenie głębokie

Czym jest uczenie głębokie ?

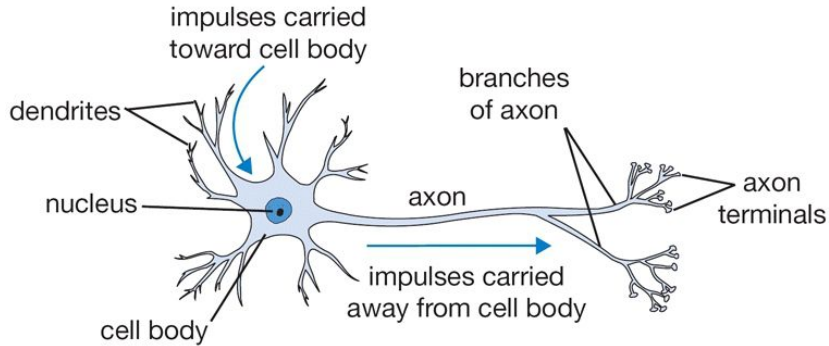


ML vs DL

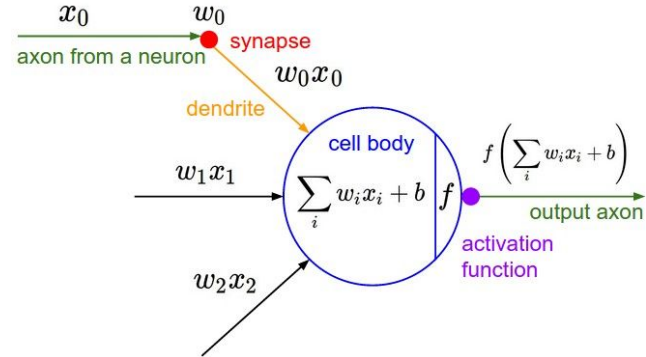


W **uczeniu głębokim** (ang. *Deep Learning*) w odróżnieniu do **uczenia maszynowego** (ang. *Machine Learning*) nie jest stosowana jawna inżynieria cech podczas której reprezentacja danych jest przygotowywana przez ekspertów. Model uczy się cech w sposób nienadzorowany, bez interakcji z użytkownikiem. Różnicą pomiędzy podejściami jest również wymagana liczba danych wejściowych do opracowania modelu.

Sztuczna sieć neuronowa - perceptron



Schemat biologiczny ludzkiego neuronu



Sztuczny neuron

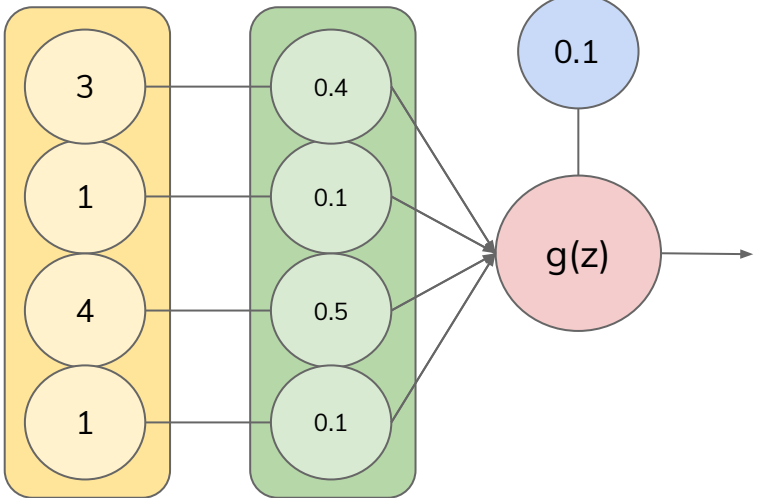
Inspirowana biologią podstawowa jednostka SSN jest zbudowany z użyciem **sumowania i wartości progowej**. Logiczne dane wejściowe są otrzymywane za pomocą pośrednictwem odpowiedników dendrytów po czym obliczana jest suma z uwzględnieniem **wag**. Jeśli dane wyjściowe przekroczą określony **próg** (ang. *threshold*) neuron generują wartość **pozytywną**. Dane wyjściowe mogą być przekazywane jako dane wejściowe dla następnych neuronów za pośrednictwem odpowiedników dendrytów.

Funkcja aktywacji

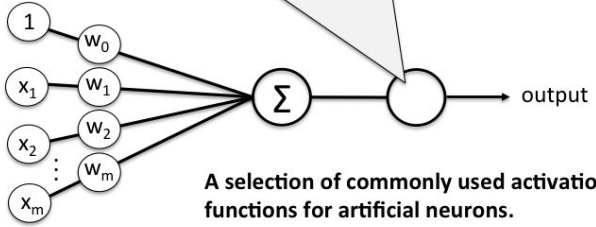
Wektor cech - x

Wektor wag - w

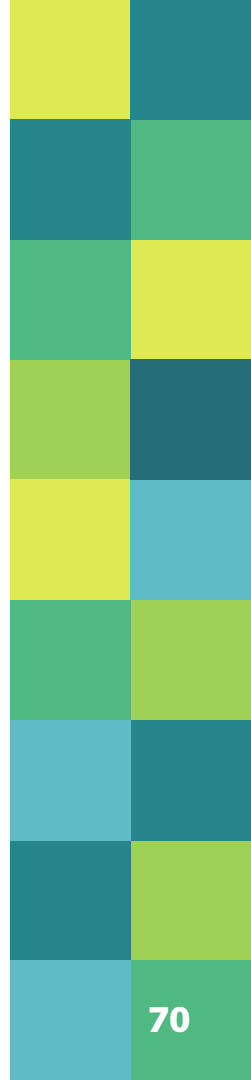
Wyraz wolny - b



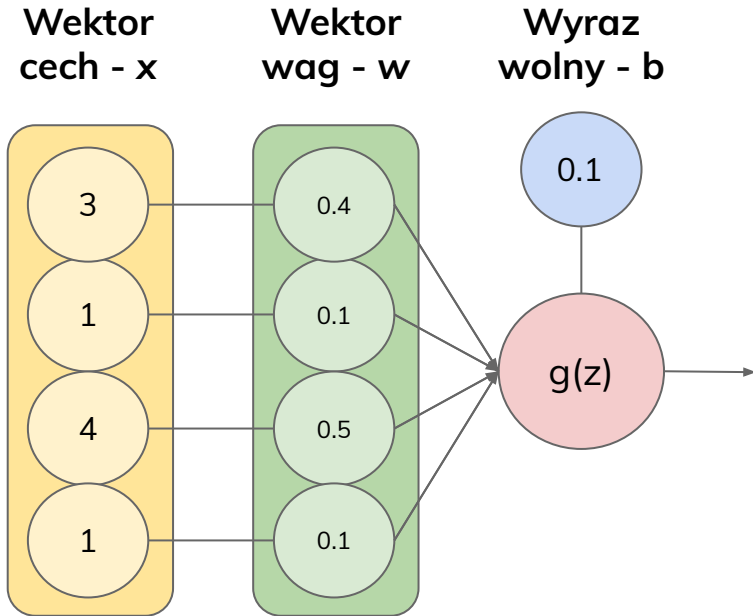
	Unit step	$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise.} \end{cases}$
		$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise.} \end{cases}$
	Linear	$g(z) = z$
	Logistic (sigmoid)	$g(z) = 1 / (1 + \exp(-z))$
	Hyperbolic tangent (sigmoid)	$g(z) = \frac{\exp(2z) - 1}{\exp(2z) + 1}$
...		



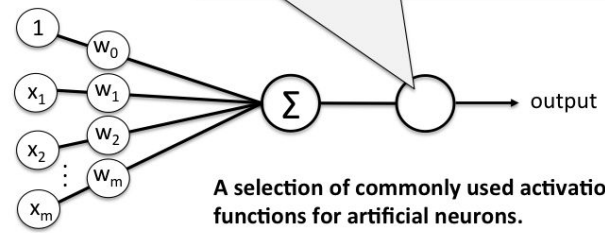
A selection of commonly used activation functions for artificial neurons.



Funkcja aktywacji



	Unit step	$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise.} \end{cases}$
		$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise.} \end{cases}$
	Linear	$g(z) = z$
	Logistic (sigmoid)	$g(z) = 1 / (1 + \exp(-z))$
	Hyperbolic tangent (sigmoid)	$g(z) = \frac{\exp(2z) - 1}{\exp(2z) + 1}$
...		



A selection of commonly used activation functions for artificial neurons.

$$z = \sum_{i=0} w_i x_i + b$$

$$z = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b$$

$$z = 1.2 + 0.1 + 2 + 0.1 + 0.1 = 3.5$$

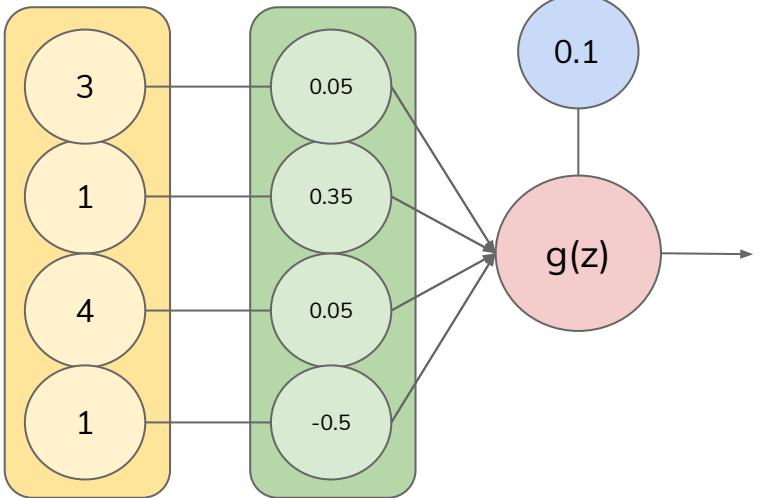
$$g(z) = \frac{1}{1 + \exp(-z)} = \frac{1}{1 + \exp(-3.5)} \approx 0.97$$

Funkcja aktywacji

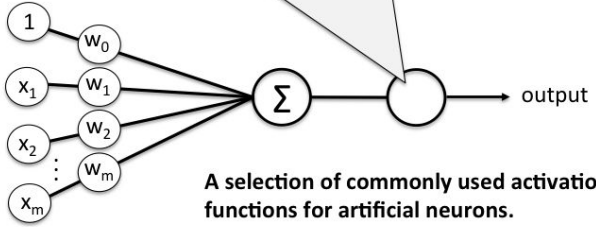
Wektor cech - x

Wektor wag - w

Wyraz wolny - b

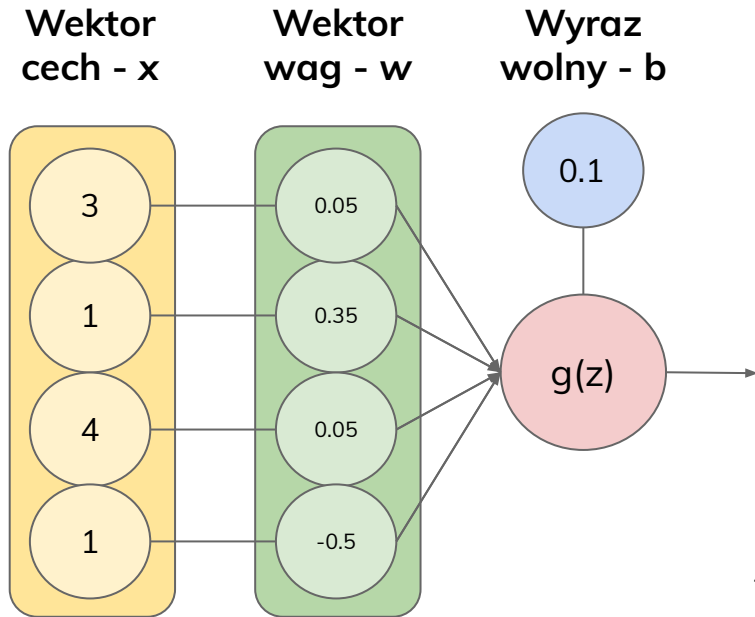


	Unit step	$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise.} \end{cases}$
		$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise.} \end{cases}$
	Linear	$g(z) = z$
	Logistic (sigmoid)	$g(z) = 1 / (1 + \exp(-z))$
	Hyperbolic tangent (sigmoid)	$g(z) = \frac{\exp(2z) - 1}{\exp(2z) + 1}$
...		

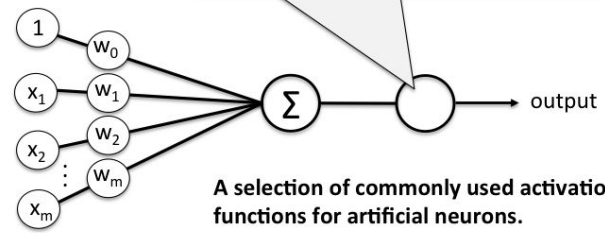


A selection of commonly used activation functions for artificial neurons.

Funkcja aktywacji



	Unit step	$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{otherwise.} \end{cases}$
		$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise.} \end{cases}$
	Linear	$g(z) = z$
	Logistic (sigmoid)	$g(z) = 1 / (1 + \exp(-z))$
	Hyperbolic tangent (sigmoid)	$g(z) = \frac{\exp(2z) - 1}{\exp(2z) + 1}$
...		



A selection of commonly used activation functions for artificial neurons.

$$z = \sum_{i=0} w_i x_i + b$$

$$z = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + b$$

$$z = 0.15 + 0.35 + 0.2 - 0.5 + 0.1 = 0.3$$

$$g(z) = \frac{1}{1 + \exp(-z)} = \frac{1}{1 + \exp(-0.3)} \approx 0.57$$

Sztuczna sieć neuronowa - MLP

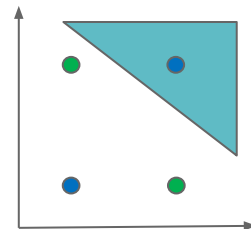
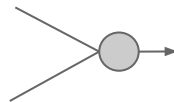
Tablica prawdy dla alternatywy rozłącznej

p	q	$p \oplus q$
0	0	0
0	1	1
1	0	1
1	1	0

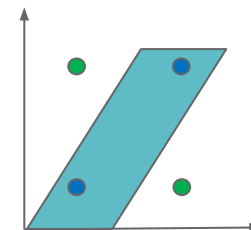
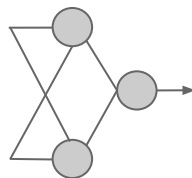
1 - zdanie prawdziwe

0 - zdanie fałszywe

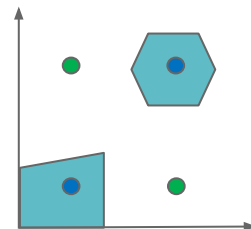
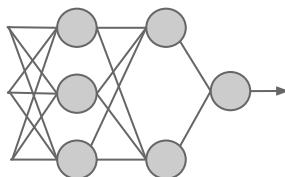
1 warstwa



2 warstwy



3 warstwy



Propagacja wsteczna

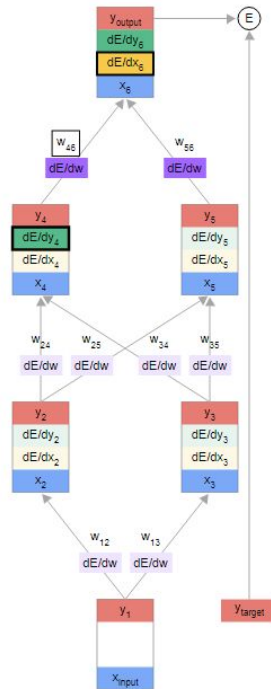
Wartość **wag** ustala się w **wyniku uczenia sieci**. Proces ten realizowany następująco:

1. Inicjacja wszystkich wag **losowymi wartościami**,

Back propagation

And using the chain rule, we can also get $\frac{dE}{dy}$ from the previous layer. We have made a full circle.

$$\frac{\partial E}{\partial y_i} = \sum_{j \in \text{out}(i)} \frac{\partial x_j}{\partial y_i} \frac{\partial E}{\partial x_j} = \sum_{j \in \text{out}(i)} w_{ij} \frac{\partial E}{\partial x_j}$$



Propagacja wsteczna

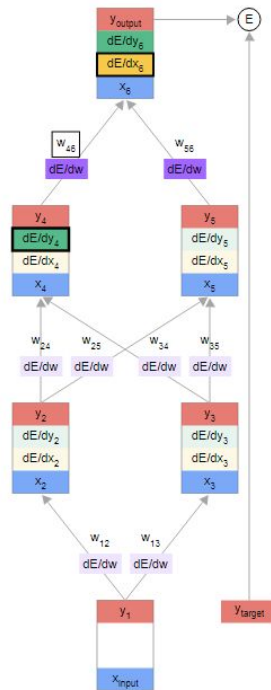
Wartość **wag** ustala się w **wyniku uczenia sieci**. Proces ten realizowany następująco:

1. Inicjacja wszystkich wag **losowymi wartościami**,
2. **Przepuszczenie** jednego elementu z danych treningowych przez sieć,

Back propagation

And using the chain rule, we can also get $\frac{\partial E}{\partial y}$ from the previous layer. We have made a full circle.

$$\frac{\partial E}{\partial y_i} = \sum_{j \in \text{out}(i)} \frac{\partial x_j}{\partial y_i} \frac{\partial E}{\partial x_j} = \sum_{j \in \text{out}(i)} w_{ij} \frac{\partial E}{\partial x_j}$$



Propagacja wsteczna

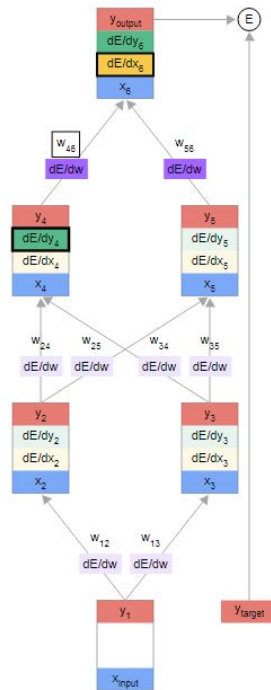
Wartość **wag** ustala się w **wyniku uczenia sieci**. Proces ten realizowany następująco:

1. Inicjacja wszystkich wag **losowymi wartościami**,
2. **Przepuszczenie** jednego elementu z danych treningowych przez sieć,
3. **Obliczany jest błąd** na podstawie otrzymanych wartości wyjściowych i stosowana jest **propagacja wsteczna** tego błędu przez sieć,

Back propagation

And using the chain rule, we can also get $\frac{\partial E}{\partial y}$ from the previous layer. We have made a full circle.

$$\frac{\partial E}{\partial y_i} = \sum_{j \in \text{out}(i)} \frac{\partial x_j}{\partial y_i} \frac{\partial E}{\partial x_j} = \sum_{j \in \text{out}(i)} w_{ij} \frac{\partial E}{\partial x_j}$$



Propagacja wsteczna

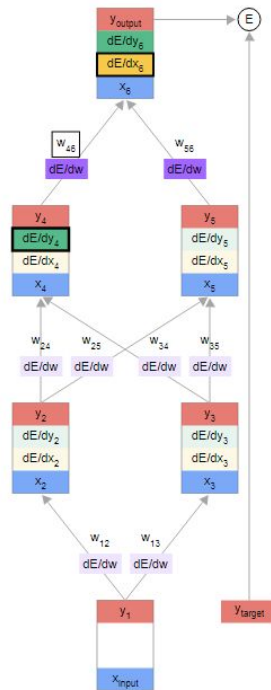
Wartość **wag** ustala się w **wyniku uczenia sieci**. Proces ten realizowany następująco:

1. Inicjacja wszystkich wag **losowymi wartościami**,
2. **Przepuszczenie** jednego elementu z danych treningowych przez sieć,
3. **Obliczany jest błąd** na podstawie otrzymanych wartości wyjściowych i stosowana jest **propagacja wsteczna** tego błędu przez sieć,
4. Każda waga w sieci jest zmieniana w kierunku, który pozwala **zminimalizować błąd sieci**.
5. Proces ten jest powtarzany do momentu spełnienia **warunku końcowego** (np. liczba iteracji).

Back propagation

And using the chain rule, we can also get $\frac{dE}{dy}$ from the previous layer. We have made a full circle.

$$\frac{\partial E}{\partial y_i} = \sum_{j \in \text{out}(i)} \frac{\partial x_j}{\partial y_i} \frac{\partial E}{\partial x_j} = \sum_{j \in \text{out}(i)} w_{ij} \frac{\partial E}{\partial x_j}$$



Sztuczna sieć neuronowa - przykład

Tinker With a **Neural Network** Right Here in Your Browser.
Don't Worry, You Can't Break It. We Promise.

Epoch: 000,000 Learning rate: 0.03 Activation: Tanh Regularization: None Regularization rate: 0 Problem type: Classification

DATA
Which dataset do you want to use?
Ratio of training to test data: 50%
Noise: 0
Batch size: 10
REGENERATE

FEATURES
Which properties do you want to feed in?
 X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$

2 HIDDEN LAYERS
4 neurons 2 neurons

This is the output from one neuron. Hover to see it larger.

The outputs are mixed with varying weights, shown by the thickness of the lines.

OUTPUT
Test loss 0.539
Training loss 0.511

<https://playground.tensorflow.org/>

Minimalizacja funkcji kosztu

Funkcja kosztu zdefiniowana jako suma błędów kwadratowych SSE.

$$\mathbf{a)} J(w) = \frac{1}{2} \sum (y^{(i)} - \phi(z^{(i)}))^2$$

$$\mathbf{b)} J(w) = (y^{(i)} - \phi(z^{(i)}))^2$$

Gdzie:

$\phi(w^T x)$ - całkowite pobudzenie neuronu

$y^{(i)}$ - wartości oczekiwane

Jeśli korzystamy z **gradientu prostego** (a) wtedy aktualizowane są jednocześnie wszystkie wagi.

Natomiast jeżeli korzystamy z **stochastycznego spadku wzdłuż gradientu** (b), wtedy wagi aktualizowane są przyrostowo dla każdej próbki.

Aby wyliczyć gradient funkcji kosztu obliczamy pochodną cząstkową tej funkcji.

$$\mathbf{a)} \Delta w_{(j)} = -\eta \frac{\partial J}{\partial w_j} = \eta \sum (y^{(i)} - \phi(z^{(i)})) x_j^{(i)} \quad \mathbf{b)} \Delta w_{(j)} = -\eta \frac{\partial J}{\partial w_j} = \eta (y^{(i)} - \phi(z^{(i)})) x_j^{(i)}$$

Zatem zmiana wag ma postać:

$$\mathbf{w} := \mathbf{w} + \Delta \mathbf{w},$$

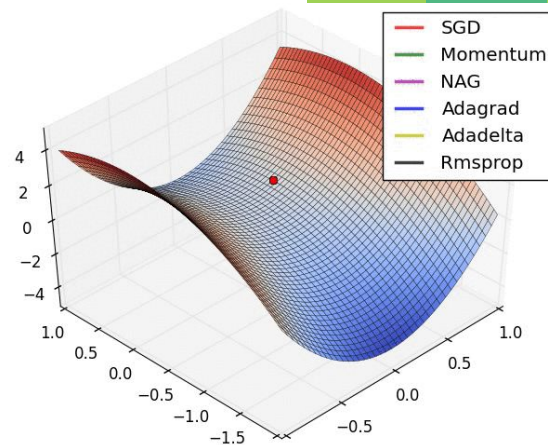
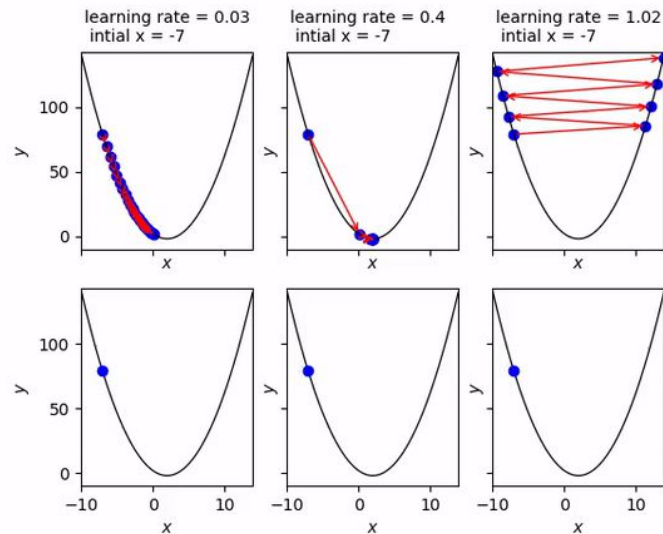
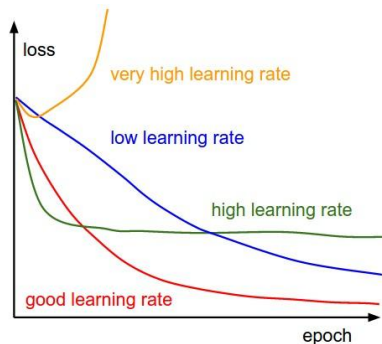
Gradient prosty

Zasadę działania algorytmu możemy wyjaśnić za pomocą metafory:

“Załóżmy, że zabłądziliśmy w górach z powodu gęstej mgły; wyczuwamy jedynie nachylenie gruntu pod stopami, logicznym rozwiązaniem zejścia na dno doliny jest podążanie w dół po jak największej pochyłości”

Pierwszym krokiem jest wypełnienie wektora wag losowymi wartościami - inicjacja losowa, które następnie są aktualizowane aż do uzyskania zbieżności algorytmu z wartością minimalną funkcji kosztu.

Istotnym parametrem algorytmu jest wielkość współczynnika uczenia. Jeżeli jego wartość będzie zbyt mała proces uczenia będzie trwał bardzo długo. Jeżeli współczynnik uczenia będzie zbyt duży, algorytm może nigdy nie trafić w punkt globalnego minimum.



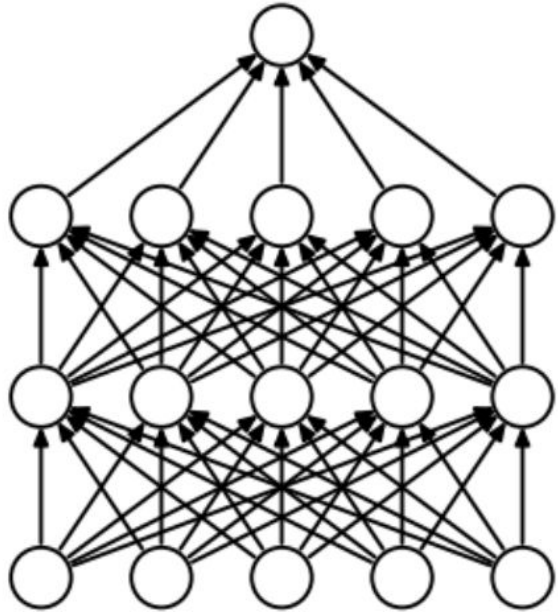
Regularyzacja

Regularyzacja to zestaw technik mających na celu uniknięcie wystąpienia zjawiska przeuczenia modelu.

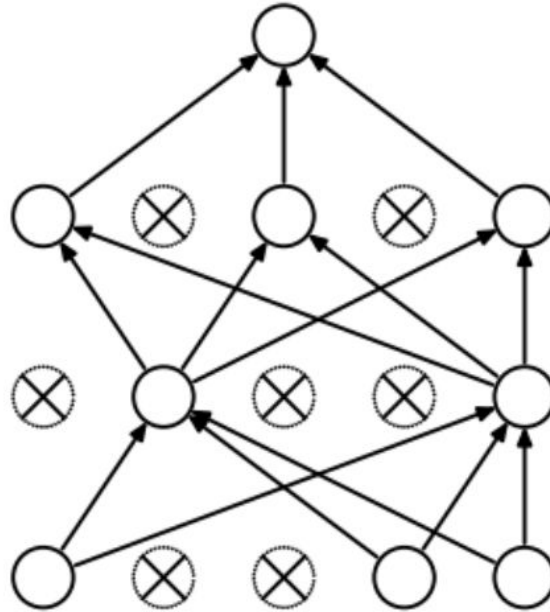
Najpopularniejszymi technikami regularyzacji są:

- technika wczesnego zatrzymania
- dogenerowanie danych
- regularyzacja L1 i L2
- normalizacja wsadowa

Regularyzacja - porzucanie

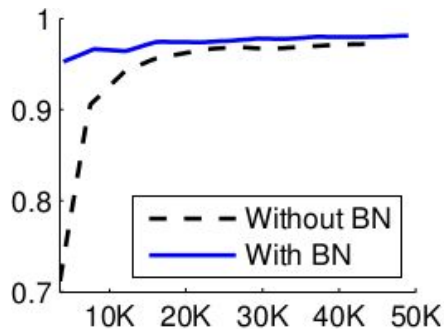


(a) Standard Neural Net

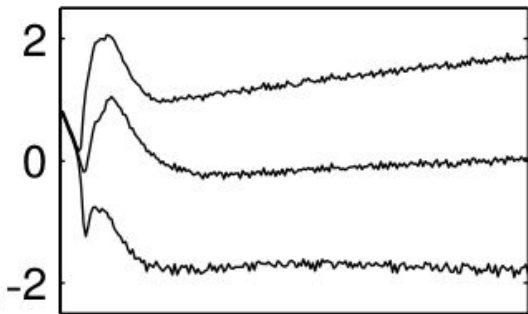


(b) After applying dropout.

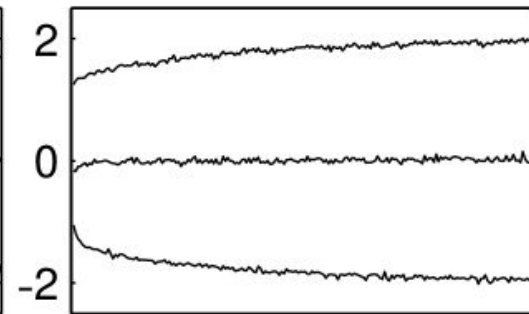
Regularyzacja - normalizacja wsadowa



(a)



(b) Without BN



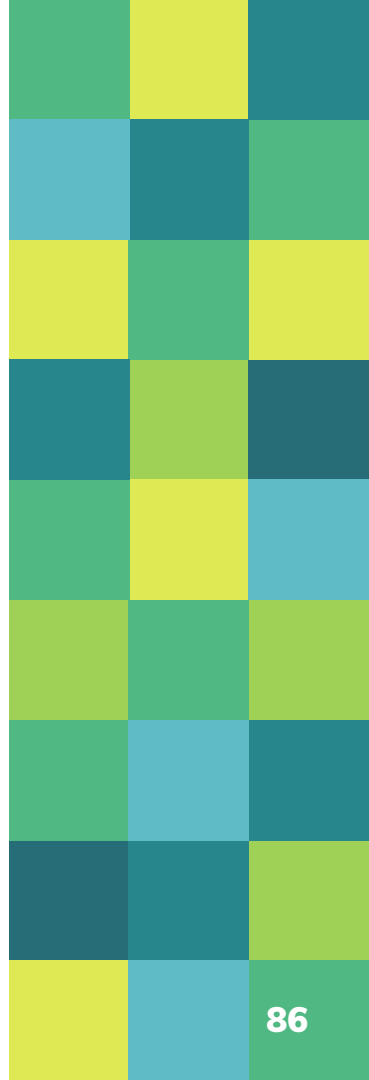
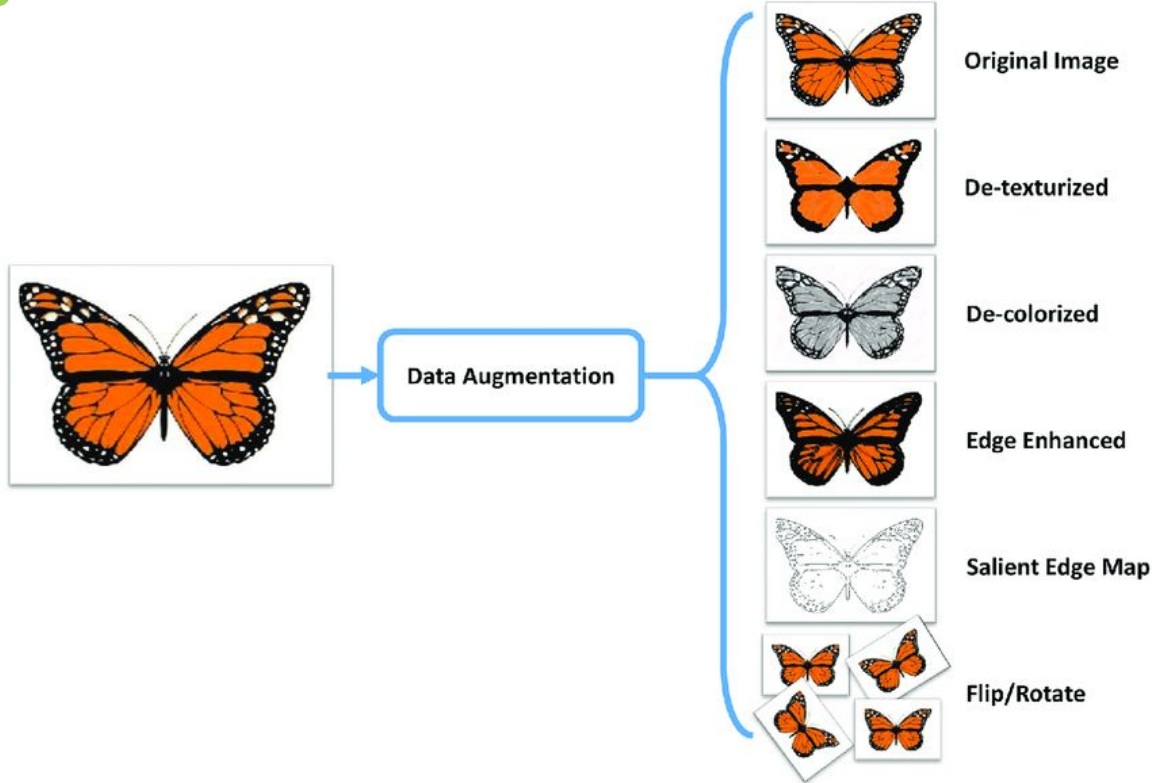
(c) With BN

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1..m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

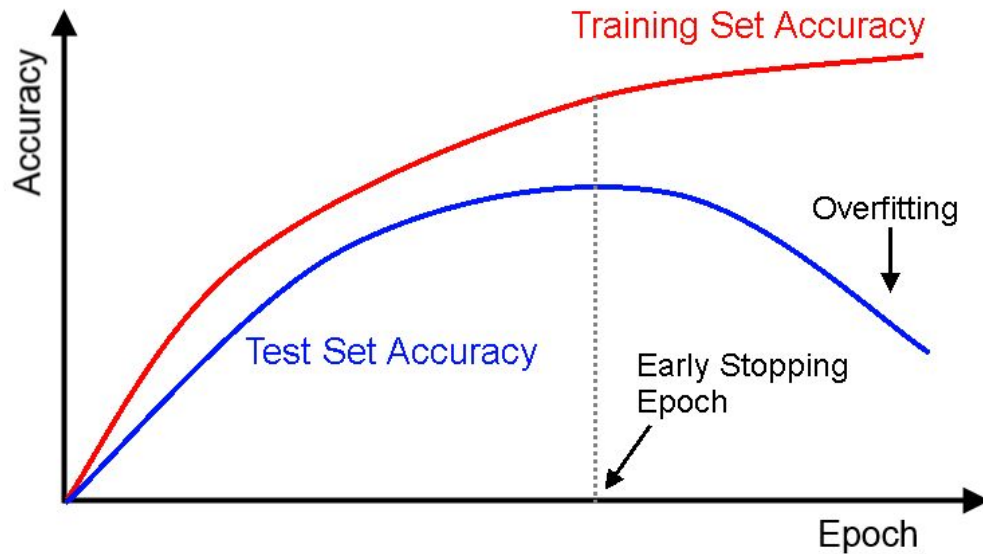
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$
$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$
$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Regularyzacja - dogenerowanie danych



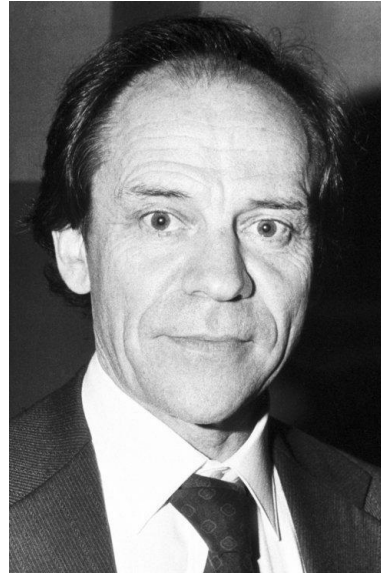
Regularyzacja - wczesne zatrzymanie



Na czym polega proces uczenia sieci konwulucyjnej

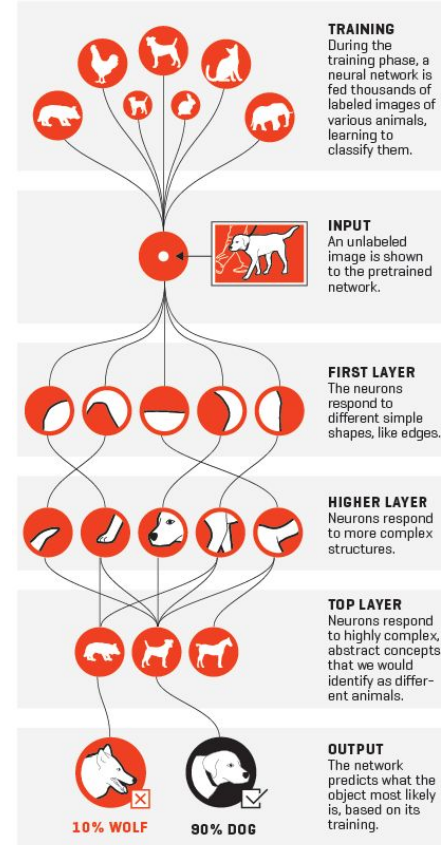


David H. Hubel



Torsten Wiesel

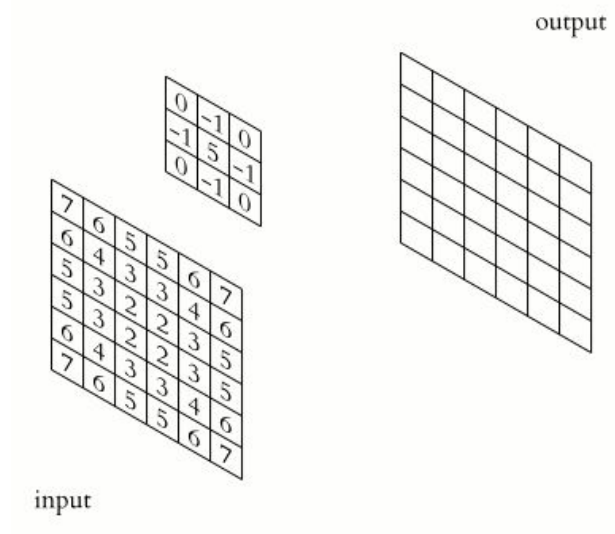
HOW NEURAL NETWORKS RECOGNIZE A DOG IN A PHOTO



Elementy sieci - warstwa konwolucyjna

Najistotniejszym elementem sieci CNN jest warstwa splotowa. Neurony w warstwie konwolucyjnej nie są połączone z każdym pikselem obrazu wejściowego lecz wyłącznie ze znajdującymi się w ich polu recepcyjnym.

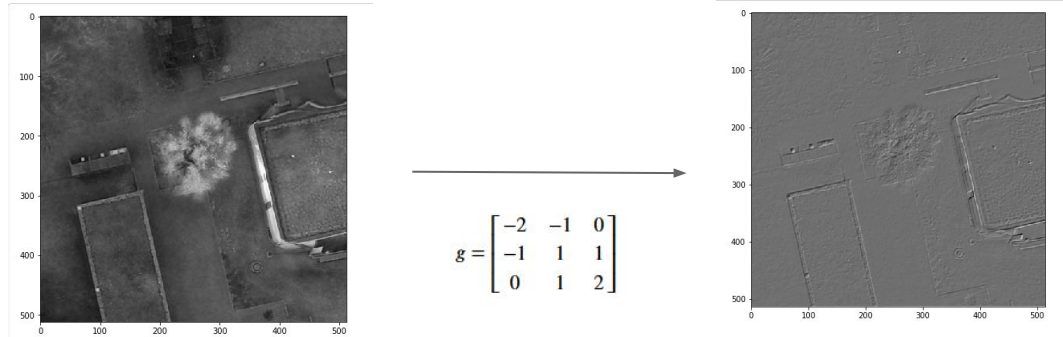
W drugiej warstwie splotowej każdy neuron łączy się z neuronami pierwszej warstwy znajdującymi się w polu recepcyjnym.



Konwolucja - operacja splotu

Splot funkcji czyli konwolucja to działanie na funkcjach, prowadząca do powstania kolejnej.

Pozwala na przetwarzanie obrazu źródłowego za pomocą filtrów, które następnie umożliwiają identyfikację obiektów znajdujących się na obrazach.



W praktyce idea konwolucji opiera się na przesuwaniu okna z wartościami z g (filtr) wzdłuż sygnału f , przemnażaniu odpowiadających wartości oraz dodawaniu tych iloczynów do siebie.

$$h[m, n] = (f * g)[m, n] = \sum_j \sum_k f[j, k]g[m - j, n - k]$$

Gdzie:

- f - Jest dwuwymiarową macierzą zawierającą wartości komórek rastra
- g - Jest dwuwymiarową macierzą filtra
- h - Macierz wynikowa

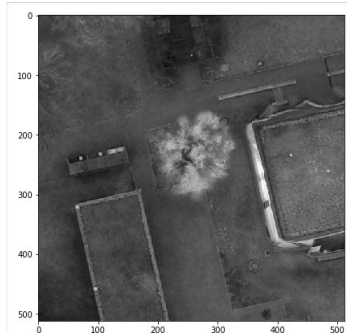
Konwolucja - operacja splotu

gdzie `begin{cases}`

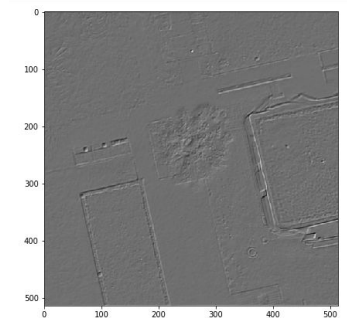
`i' = i \times s_h + u`
Splot funkcji czyli `end{cases}` to jest operacja na funkcjach, prowadząca do powstania kolejnej.

`j' = j \times s_w + v`

Pozwala na przetwarzanie obrazu źródłowego za pomocą filtrów, które następnie umożliwiają identyfikację obiektów znajdujących się na obrazach.



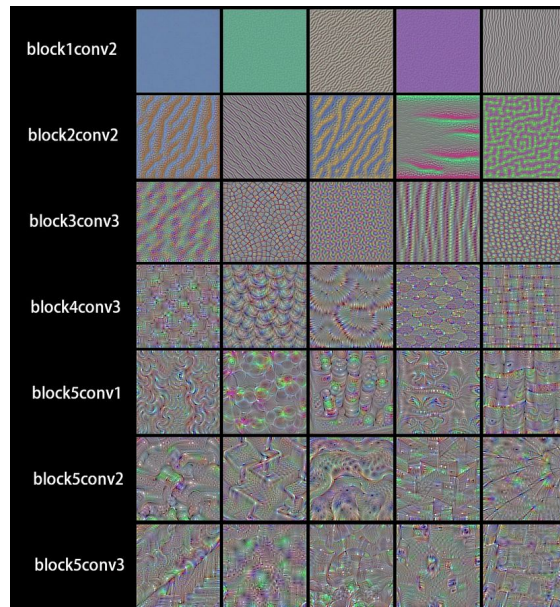
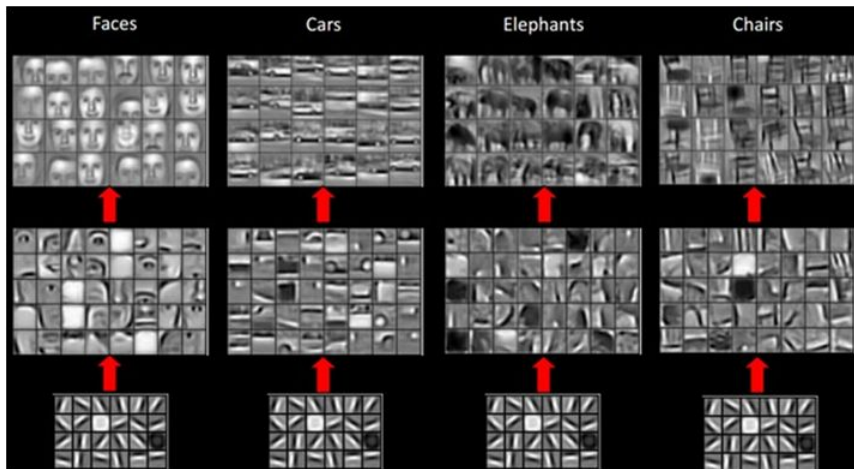
$$g = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$



$$\text{gdzie} \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases}$$

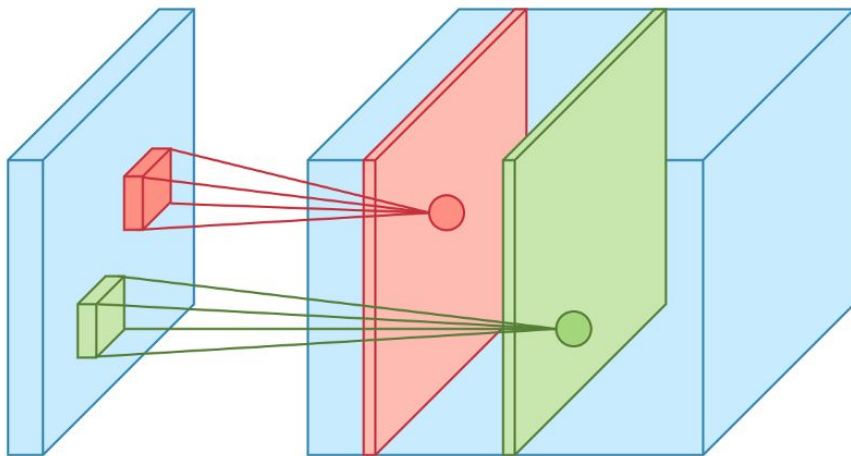
Elementy sieci - filtry

W trakcie procesu trenowania modelu, sieć CNN wyszukuje najbardziej przydatne do danego zadania filtry (zbiory wag) i dzięki nim tworzy mapy cech. Przykładowy filtr może być macierzą rozmiarach 3x3 która 1 i 3 kolumnę będzie wypełniona zerami natomiast 2 będzie wypełniona jedynekami, taki filtr będzie identyfikował pionowe lub posiadające pionowe krawędzie obiekty ponieważ obiekty znajdujące się poza tą linią będą przemnażane przez 0.



Stosy map cech

W zakresie jednej mapy cech wszystkie neurony współdzielą te same parametry (wagi i człon obciążenia), natomiast inne mapy cech mogą mieć inne wartości parametrów. Warstwa spłotowa stosuje równocześnie różne filtry na wejściach, dzięki czemu wykrywa jednocześnie wiele cech w dowolnym obszarze obrazu.



Elementy sieci - warstwa łącząca

Warstwy łączące (pooling layers) mają na celu podpróbkowanie obrazu wejściowego oraz zredukowanie obciążenie obliczeniowe związanego z wykorzystaniem pamięci i liczby parametrów (czyli ryzyka przetrenowania).

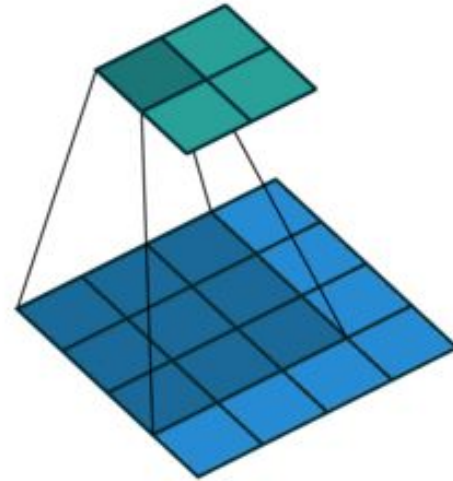
Każdy neuron warstwy stanowiący część warstwy łączącej łączy wyjścia z wyjściami określonej liczby neuronów warstwy poprzedniej.

Warstwa łącząca nie zawiera żadnych wag, jej głównym celem jest zadaniem jest gromadzeniem danych wejściowych za pomocą funkcji agregujących:

- maksymalizującą warstwę łączącą (**max pooling layer**) - przekazującą maksymalną wartość z obszaru jądra
- uśredniającą warstwę łączącą (**mean pooling layer**)

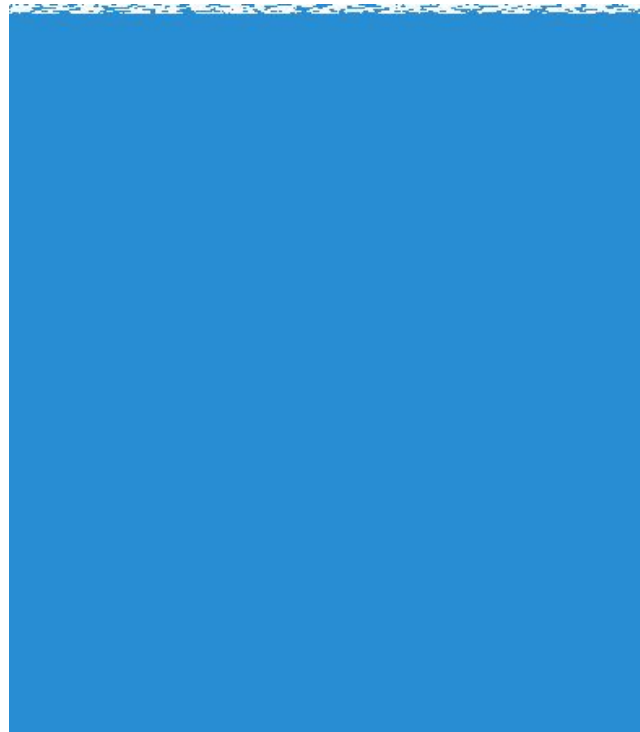
Przekazującą średnią wartość z obszaru jądra

Czynnikiem mającym wpływ na rezultaty jest nie tylko rodzaj agregacji ale również rozmiar jądra łączącego, w którym definiujemy rozmiar kernela (ksize), wielkość kroku (stride) oraz sposób uzupełnienia braku (padding)



Elementy sieci - krok

Krok (Strides) to czteroelementowy, jednowymiarowy tensor, w którym dwa środkowe elementy określają rozmiar kroku w kierunku pionowym i poziomym. Służą do określania wielkości kroku wobec minigrupy. Użyteczne są także jeśli chcemy jakiś obszar obrazu lub kanał pominąć.

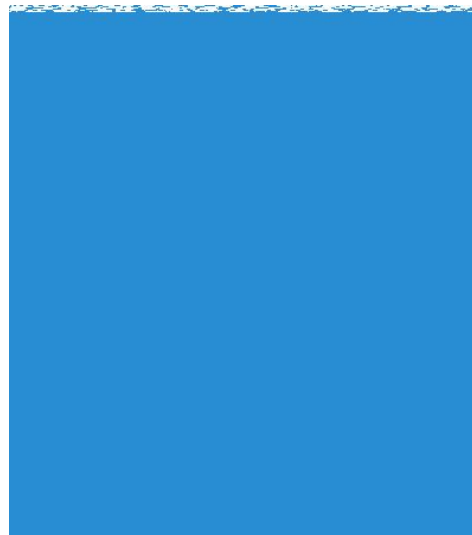
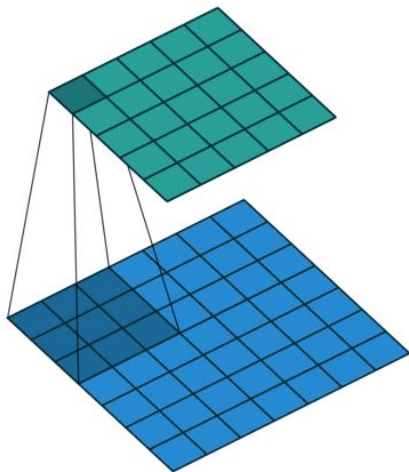


Elementy sieci - dopełnianie

Dopełnianie (Padding) jest parametrem, którego zadaniem jest uzupełnianie z zerami w przypadku gdy krok filtra wykracza poza obszar obrazu. Wykorzystuje się dwa rodzaje parametru padding:

-VALID - warstwa spłotowa nie używa uzupełniania zerami oraz może ignorować dolne rzędy i kolumny, w zależności od rozmiaru kroku.

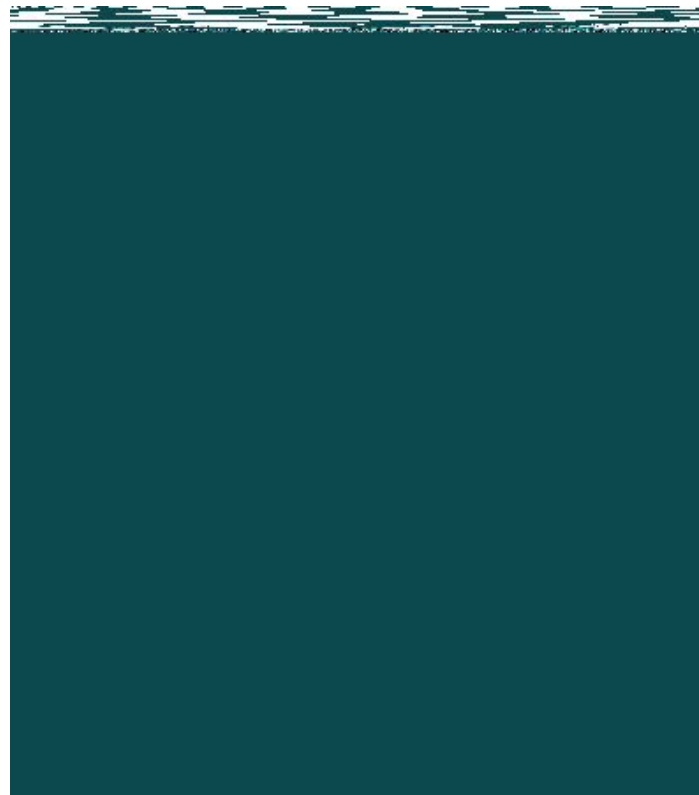
-SAME - uzupełnianie zerami będzie wykorzystywane w zależności od potrzeby. W takim przypadku liczba neuronów jest równa zaokrąglonemu ilorazowi iloczynowi neuronów wejściowych i rozmiaru kroku. Następnie dodawane są zera wokół danych wejściowych w taki sposób aby wyszło jak najrówniej.



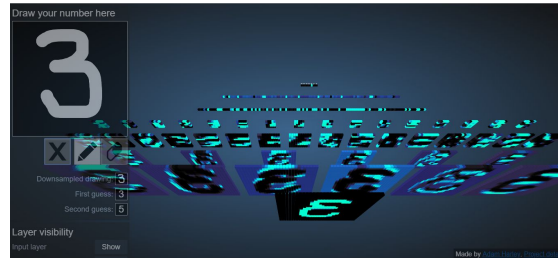
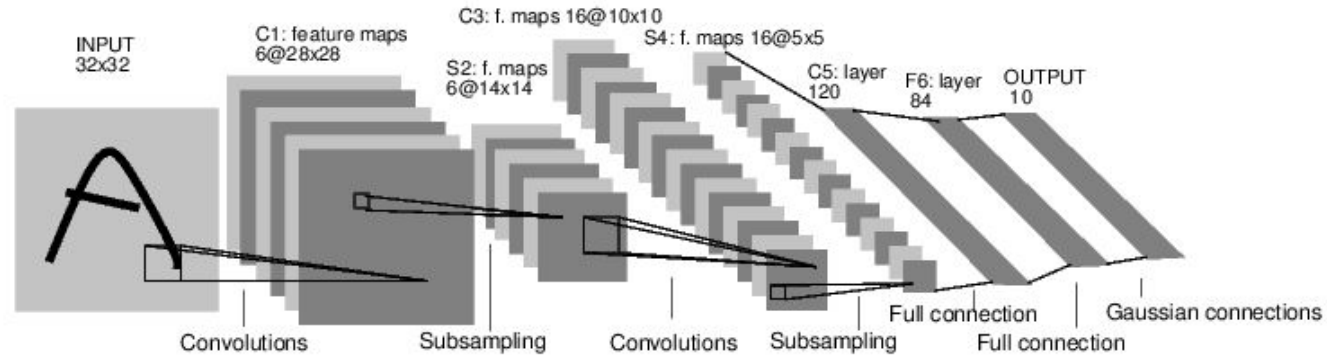
Elementy sieci - transponowana konwolucja

Transponowana konwolucja (Deconvolution, fractionally strided convolutions) dokonuje regularnego splotu, ale odwraca jego przestrzenną transformację.

Odtwarza przestrzenną rozdzielczość z poprzednich warstw i wykonuje splot. Umożliwia to połączenie skalowania obrazu ze splotem, bez potrzeby wykonywania dodatkowych operacji.



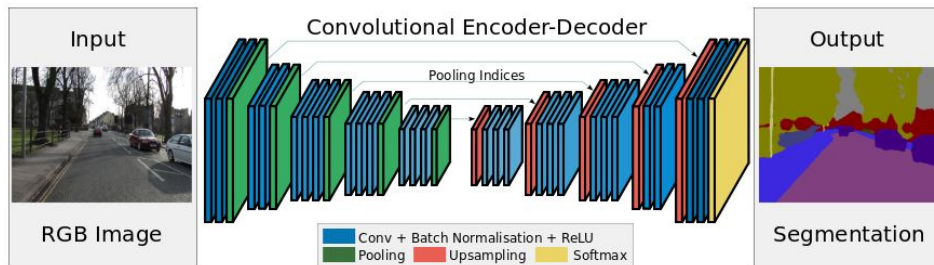
Działanie sieci



<http://scs.ryerson.ca/~aharley/vis/conv/>

Przykładowa architektura sieci konwolucyjnej - UNET

U-Net łączy warstwy enkodera z mapami obiektów dekodera tworząc strukturę przypominającą drabinę. Cechą szczególną sieci U-Net jest to, że na każdym etapie umożliwia na zapoznanie się z cechami, które zostały utracone w wyniku działania warstwy łączącej



2

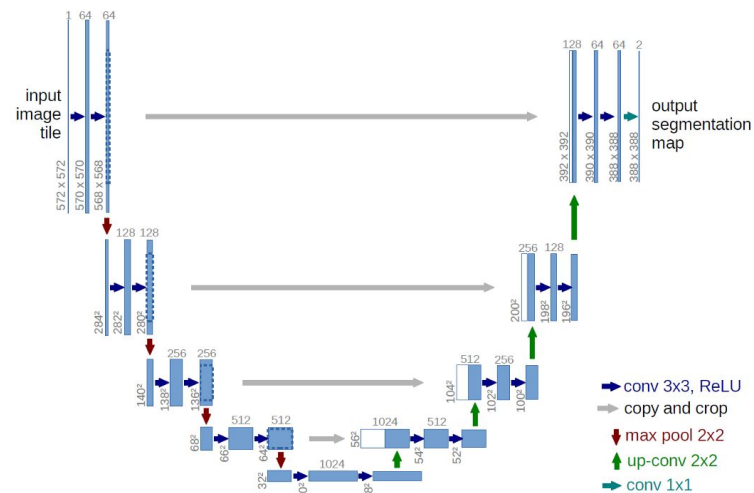
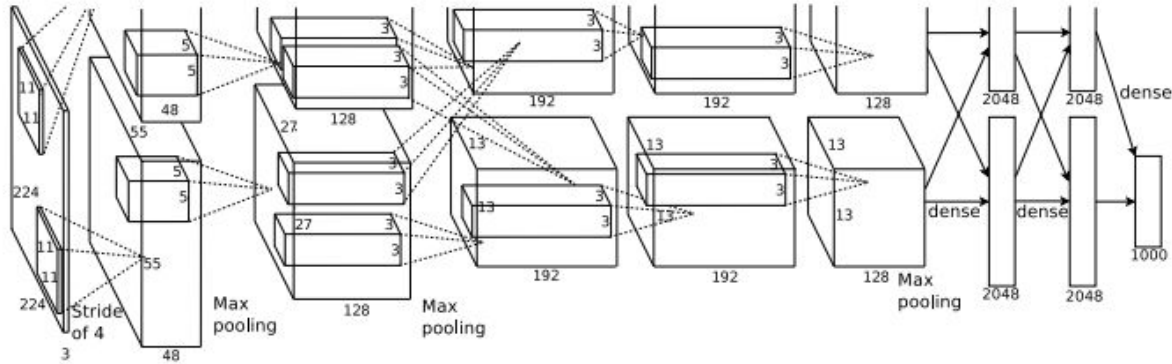


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

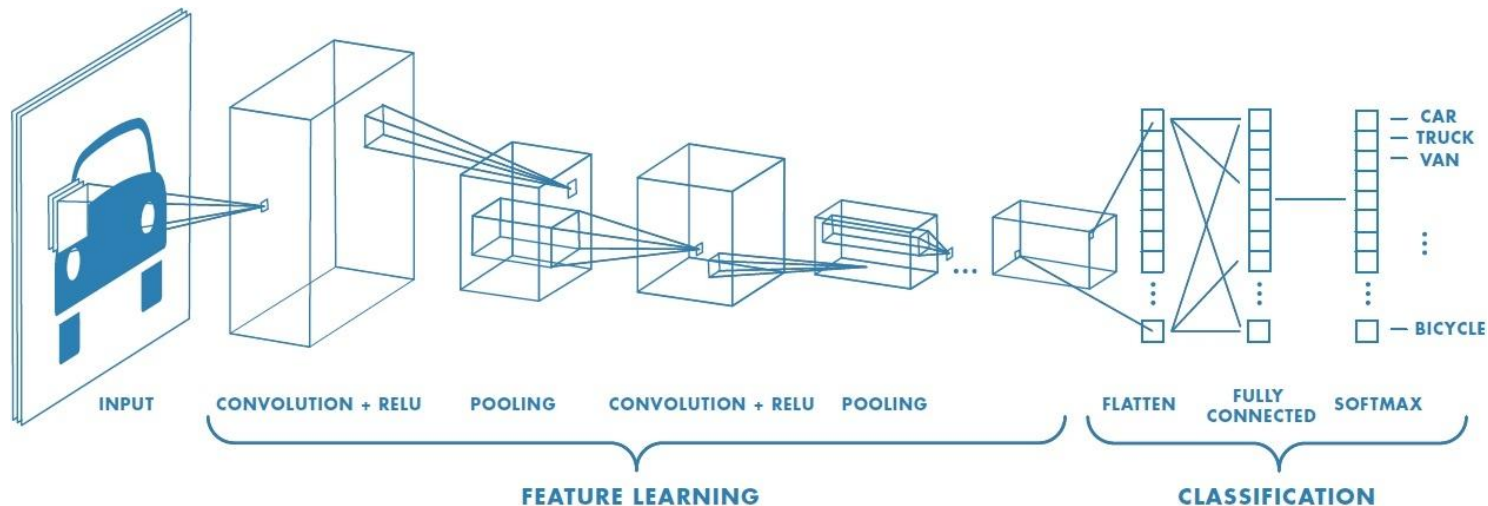
Przykładowa architektura sieci konwolucyjnej - AlexNet



W 2012 roku firma AlexNet znacznie wyprzedziła wszystkich dotychczasowych konkurentów i wygrała wyzwanie, zmniejszając błąd w pierwszej piątce z 26% do 15,3%. Drugie miejsce, w którym poziom błędu top-5, który nie był zmianą CNN, wynosił około 26,2%.

Wynik uzyskano dzięki zastosowaniu dwóch różnych technik regularyzacji oraz normalizacji odpowiedzi lokalnej.

Przykładowa architektura sieci konwolucyjnej - VGG16



Drugie miejsce w konkursie ILSVRC 2014 zostało nazwane przez społeczność VGGNet i zostało opracowane przez Simonyana i Zissermana. VGGNet składa się z 16 warstw splotowych i jest bardzo atrakcyjny ze względu na bardzo jednolitą architekturę. Jest to obecnie najbardziej preferowany w społeczności do wydobywania cech z obrazów.

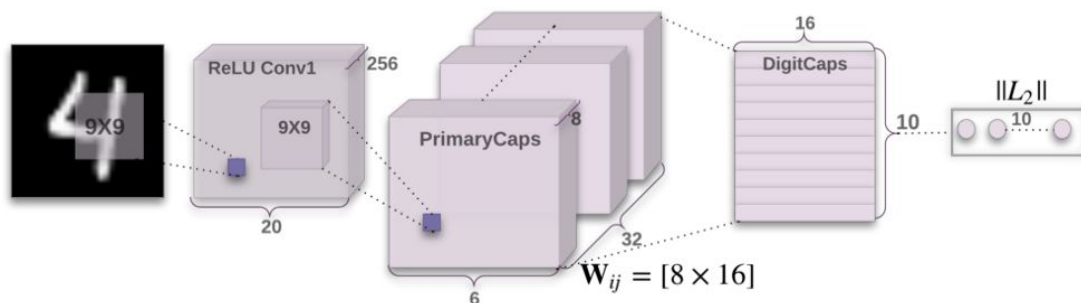
Sieć Kapsułowa

Grafika komputerowa zajmuje wewnętrzną reprezentację obiektów i tworzy obraz. Sieci kapsuł działają odwrotnie. Hinton nazywa je odwrotną grafiką. Od wizualnych informacji, obrazów dekonstruują hierarchiczną reprezentację i próbują dopasować je do poznanych już wzorców. Kluczową ideą jest to, że reprezentacja obiektów nie jest zależna od kąta widzenia.

Sieci kapsułowe wymagają zdecydowanie mniej danych do uczenia się niż sieci konwolucyjne.

Hinton i jego zespół zaproponowali sposób na wyszkolenie takiej sieci składającej się z kapsułek i pomyślnie wyszkolili ją na prostym zestawie danych, osiągając najnowocześniejsze wyniki.

Obecne wdrożenia są znacznie wolniejsze niż inne nowoczesne modele głębokiego uczenia się. Czas pokaże, czy sieci kapsuł można szybko i skutecznie szkolić. Ponadto musimy sprawdzić, czy działają dobrze w trudniejszych zestawach danych i w różnych domenach.



Ćwiczenie 4. A

Klasyfikacja chmury punktów za pomocą VNN.

Proszę wejść do folderu

`./cwiczenia_LIDAR/SiecWolumetryczna`

Ćwiczenie 4. A

Klasyfikacja zdjęć lotniczych za pomocą CNN.

Proszę wejść do folderu

`./cwiczenia_Imagery/Sieckonw`

Podsumowanie



Bibliografia

1. Python. *Uczenie maszynowe*, S. Raschka, 2017
2. *Inteligentna sieć. Algorytmy przyszłości.*, D. McIlwraith, H. Marmanis, D. Babenko, 2017
3. *GIS teoria i praktyka*, Longley Paul A., Goodchild Michael F., Rhind David W., 2008
4. *Fotogrametria*, Z. Kurczyński, 2014
5. *Uczenie maszynowe z użyciem Scikit-Learn i Tensorflow*, Aurelien Geron, 2018
6. *Zaawansowane uczenie maszynowe z językiem Python*, John Hearty, 2016
7. <https://ksopyla.com/>

Motyw graficzny prezentacji:

<https://www.slidescarnival.com/nym-free-presentation-template/2056>

Dziękujemy za uwagę



Dominik Mielczarek
Piotr Czubek

dominik.mielczarek@opegieka.pl
piotr.czubek@opegieka.pl