

WYKŁAD 1

Inteligentne systemy internetowe

Od uczenia maszynowego do mechanizmów wyszukiwania

Prowadzący: Jarosław Szkoła

Plan wykładu

- 1 Wprowadzenie do AI
- 2 Metody uczenia maszynowego
- 3 Metody predykcji
- 4 Klastrowanie
- 5 Sieci neuronowe i uczenie głębokie
- 6 Analiza obrazu i sieci konwolucyjne
- 7 Systemy rekomendacyjne
- 8 Mechanizmy wyszukiwania

01

Wprowadzenie do AI

Czym jest sztuczna inteligencja?

- Systemy wykonujące zadania wymagające inteligencji: percepcja, wnioskowanie, uczenie, język
- Podejście symboliczne (reguły, logika) vs. statystyczne (uczenie z danych)
- AI wąska (jedno zadanie) vs. ogólna AGI (hipotetyczna)
- Dziedziny: ML, widzenie komputerowe, NLP, robotyka, systemy ekspertowe

Historia AI

- 1950: test Turinga; 1956: konferencja w Dartmouth i narodziny nazwy „AI”
- Lata 70. i 80.: „zimy AI”, systemy ekspertowe
- 2012: AlexNet wygrywa ImageNet – początek ery głębokiego uczenia
- 2017: architektura transformer; po 2020: duże modele językowe i generatywne

Zastosowania, wyzwania i etyka

- Zastosowania: diagnostyka medyczna, wykrywanie oszustw, pojazdy autonomiczne, wyszukiwanie
- Wyzwania: jakość i ilość danych, interpretowalność, zasoby obliczeniowe
- Ryzyka: uprzedzenia (bias), prywatność, dezinformacja, odpowiedzialność
- Regulacje: unijny AI Act, podejście oparte na poziomach ryzyka

AI, uczenie maszynowe i głębokie uczenie



AI

systemy realizujące zadania wymagające inteligencji

ML

algorytmy uczące się wzorców z danych

DL

wielowarstwowe sieci neuronowe

02

Metody uczenia maszynowego

Uczenie nadzorowane i nienadzorowane

- Nadzorowane: dane z etykietami, model uczy się odwzorowania $X \rightarrow y$
- Nienadzorowane: odkrywanie struktury w danych bez etykiet (grupowanie, redukcja wymiarów)
- Półnadzorowane: mało etykiet i dużo danych nieoznaczonych
- Samonadzorowane: etykiety tworzone z samych danych (podstawa dużych modeli)

Uczenie ze wzmocnieniem

- Agent wykonuje akcje w środowisku i otrzymuje nagrody
- Cel: polityka maksymalizująca skumulowaną nagrodę
- Algorytmy: Q-learning, Deep Q-Network, metody policy gradient
- Przykłady: gry (AlphaGo), robotyka, sterowanie procesami

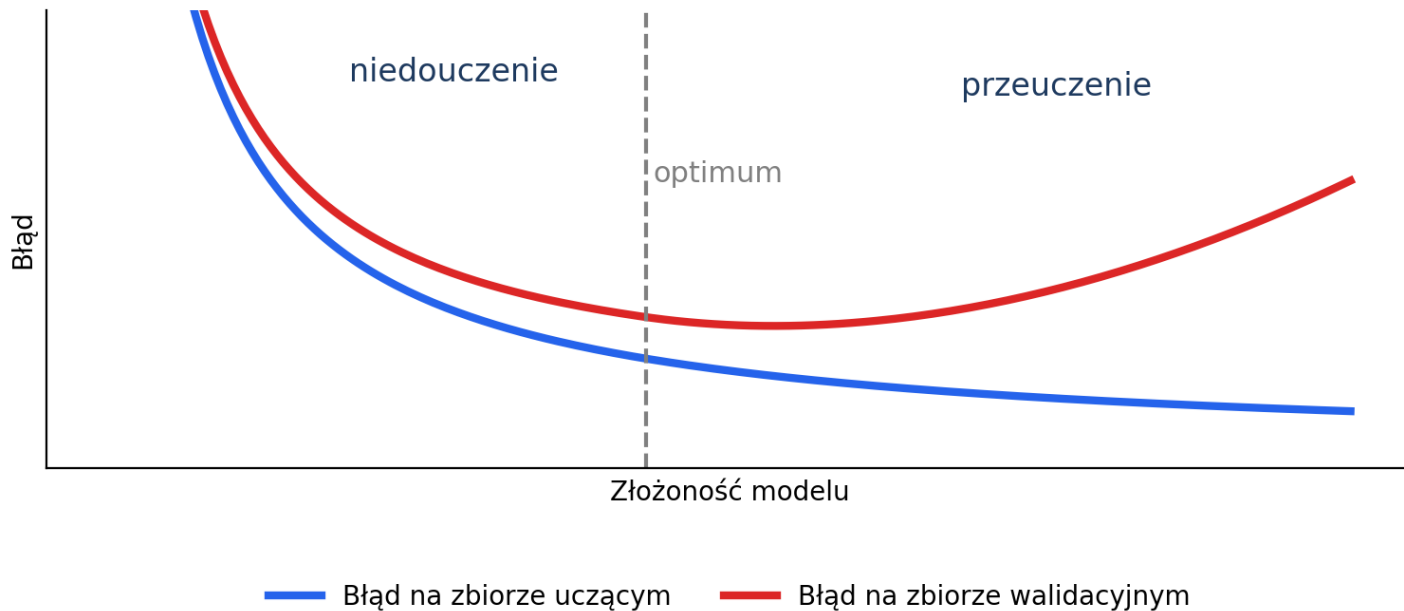
Proces budowy modelu

- Dane → czyszczenie → inżynieria cech → trening → walidacja → test
- Przeuczenie (overfitting) vs. niedouczenie (underfitting); kompromis obciążenie–wariancja
- Regularyzacja (L1, L2), wczesne zatrzymanie, więcej danych
- Zbiory: uczący, walidacyjny, testowy; walidacja krzyżowa

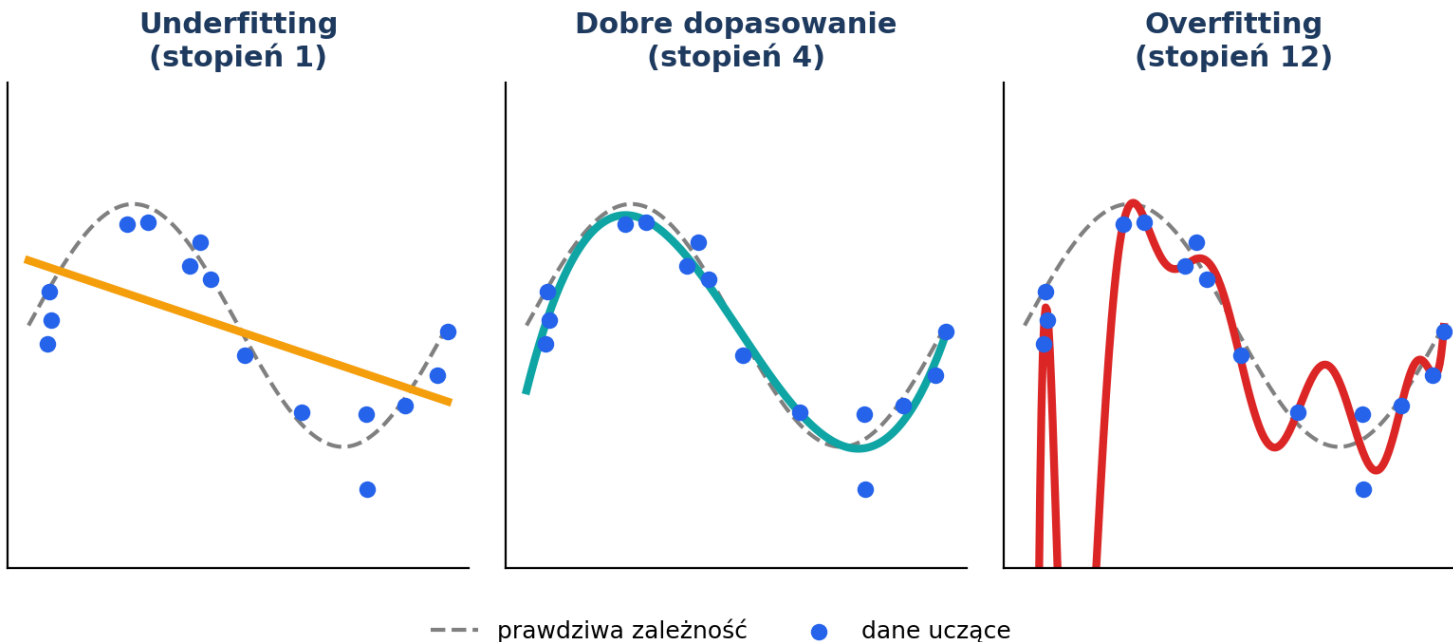
Overfitting i underfitting – objawy i przyczyny

- Underfitting: model zbyt prosty – duży błąd zarówno na zbiorze uczącym, jak i testowym
- Overfitting: model „zapamiętuje” dane uczące, w tym szum – mały błąd uczący, duży testowy
- Przyczyny overfittingu: zbyt duża złożoność, mało danych, zbyt długi trening, zbędne cechy
- Diagnoza: krzywe uczenia i porównanie błędu uczącego z walidacyjnym

Niedouczenie i przeuczenie modelu



Dopasowanie wielomianów różnego stopnia



Jak zwalczać overfitting i underfitting

- Overfitting: więcej danych, augmentacja, regularyzacja L1/L2, dropout, wczesne zatrzymanie
- Overfitting: uproszczenie modelu, selekcja cech, przycinanie drzew, zespoły modeli
- Underfitting: bardziej złożony model, nowe cechy, mniejsza regularyzacja, dłuższy trening
- Kompromis obciążenie–wariancja: $\text{błąd} = \text{obciążenie}^2 + \text{wariancja} + \text{szum}$

K-fold cross validation – idea

- Dane dzielimy na k równych części (foldów), np. $k = 5$ lub 10
- W każdej iteracji jeden fold jest testowy, a pozostałe $k-1$ uczące
- Po k iteracjach każda obserwacja została użyta do testu dokładnie raz
- Wynik końcowy: średnia (i odchylenie standardowe) z k ocen

Schemat 5-krotnej walidacji krzyżowej

Zbiór danych podzielony na 5 foldów

Iteracja 1	test	uczenie	uczenie	uczenie	uczenie	dokładność = 0,86
Iteracja 2	uczenie	test	uczenie	uczenie	uczenie	dokładność = 0,84
Iteracja 3	uczenie	uczenie	test	uczenie	uczenie	dokładność = 0,88
Iteracja 4	uczenie	uczenie	uczenie	test	uczenie	dokładność = 0,85
Iteracja 5	uczenie	uczenie	uczenie	uczenie	test	dokładność = 0,87
folds uczące fold testowy średnia = 0,86 ($\pm$ 0,01)						

K-fold: wybór k i warianty

- Mały k (2–3): szybko, ale większe obciążenie oceny; duży k: dokładniej, ale drożej
- Stratified k-fold: zachowuje proporcje klas w każdym foldzie (dane niezbalansowane)
- Leave-one-out ($k = n$); TimeSeriesSplit dla szeregów czasowych (bez „zaglądania w przyszłość”)
- Nested CV: strojenie hiperparametrów bez wycieku informacji do oceny końcowej

03

Metody predykcji

Klasyfikacja – podstawy

- Cel: przypisanie obiektu do jednej z klas (np. spam / nie spam)
- Binarna, wieloklasowa i wieloetykietowa
- Granica decyzyjna oddziela klasy w przestrzeni cech
- Dane niezbalansowane wymagają specjalnego traktowania (próbkiowanie, wagi klas)

Algorytmy klasyfikacji

- k-NN: klasa większości k najbliższych sąsiadów
- Drzewa decyzyjne: kolejne podziały wg cechy (entropia, indeks Giniego)
- Lasy losowe i boosting (XGBoost): zespoły drzew
- SVM i regresja logistyczna: separacja liniowa lub z użyciem jąder

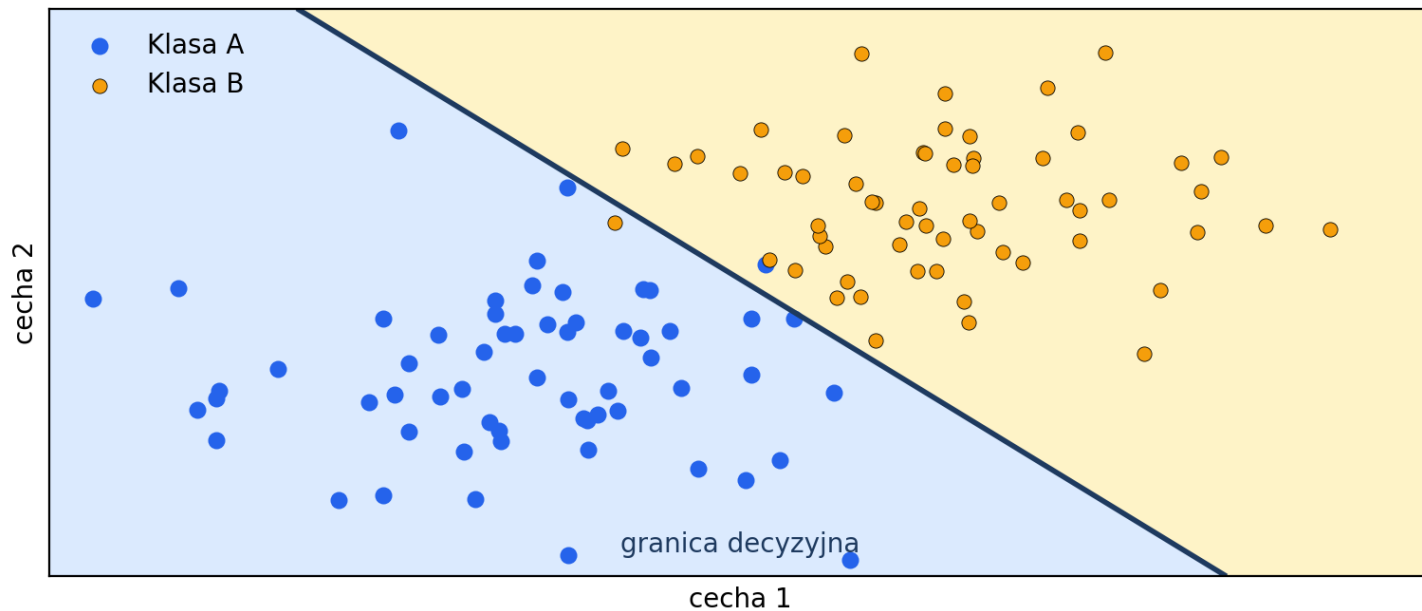
Regresja

- Cel: przewidywanie wartości ciągłej (cena, temperatura, popyt)
- Regresja liniowa: $y = w \cdot x + b$, dopasowanie metodą najmniejszych kwadratów
- Regresja wielomianowa, Ridge, Lasso; drzewa i lasy regresyjne
- Założenia i pułapki: współliniowość, wartości odstające, ekstrapolacja

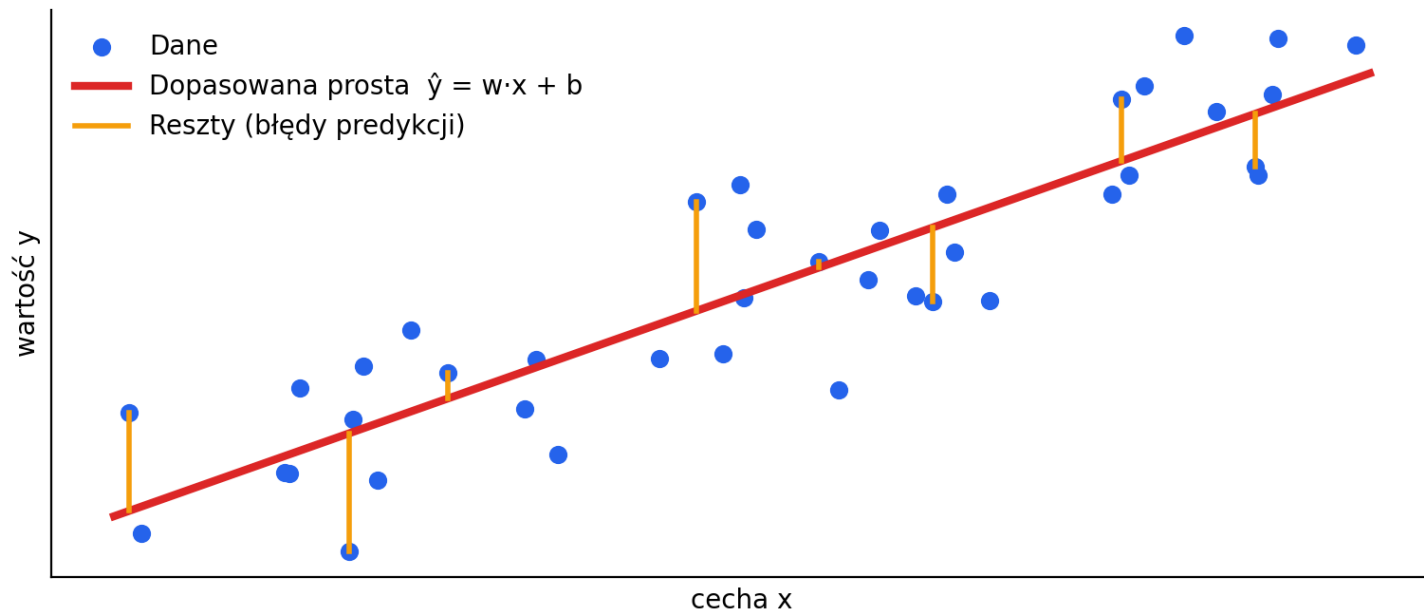
Ocena jakości modeli

- Klasyfikacja: macierz pomyłek, accuracy, precision, recall, F1, ROC-AUC
- Regresja: MAE, MSE, RMSE, współczynnik determinacji R^2
- Precision vs. recall: koszt błędu zależy od zastosowania (np. diagnostyka)
- Walidacja krzyżowa k-krotna daje stabilniejsze oszacowanie

Klasyfikacja i granica decyzyjna



Regresja liniowa i reszty



Macierz pomyłek: TP, TN, FP, FN

- TP (true positive): przypadek pozytywny poprawnie rozpoznany jako pozytywny
- TN (true negative): przypadek negatywny poprawnie rozpoznany jako negatywny
- FP (false positive): fałszywy alarm – negatyw uznany za pozytyw (błąd I rodzaju)
- FN (false negative): przeoczenie – pozytyw uznany za negatyw (błąd II rodzaju)

Macierz pomyłek i miary jakości

	przewidziana: pozytywna	przewidziana: negatywna
rzeczywista: pozytywna	TP = 40 prawdziwie pozytywne	FN = 20 fałszywie negatywne
rzeczywista: negatywna	FP = 10 fałszywie pozytywne	TN = 130 prawdziwie negatywne

Miary jakości

Accuracy	0,85
Precision	0,80
Recall	0,67
Specyficzność	0,93
F1	0,73

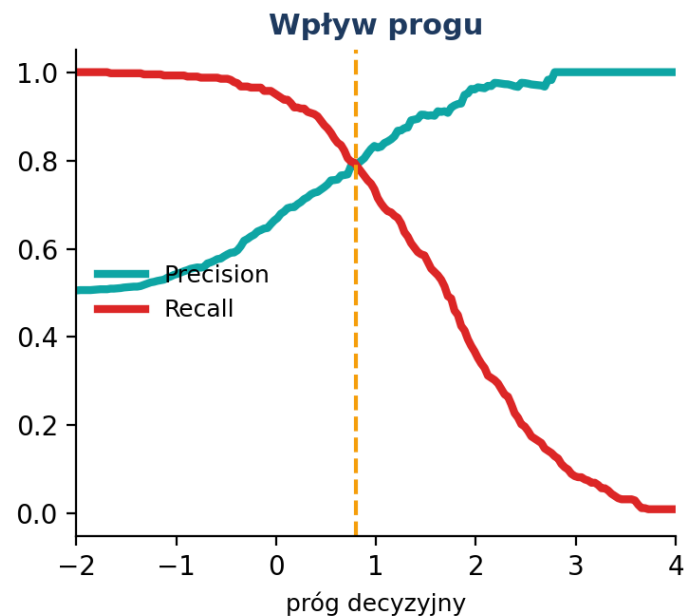
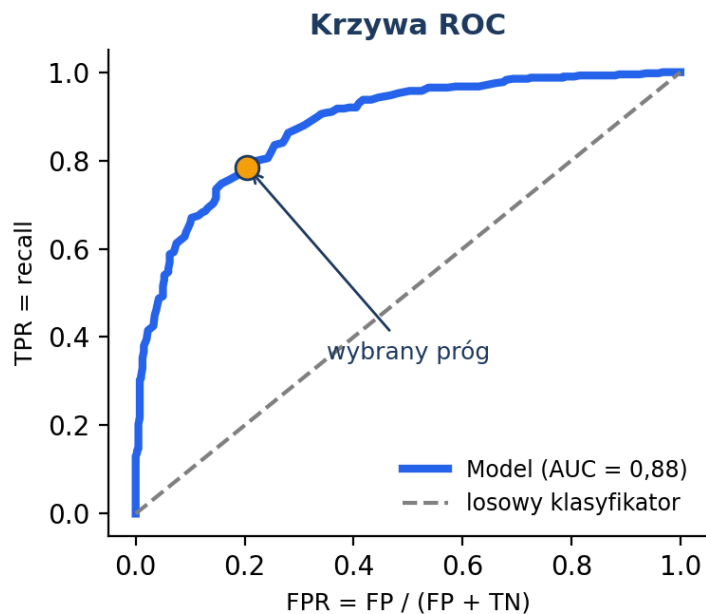
Miary jakości klasyfikatora

- $\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$ – odsetek poprawnych odpowiedzi
- $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$ – ile wskazanych pozytywów jest naprawdę pozytywnych
- $\text{Recall (czułość)} = \text{TP} / (\text{TP} + \text{FN})$; $\text{specyficzność} = \text{TN} / (\text{TN} + \text{FP})$
- $\text{F1} = 2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall})$ – średnia harmoniczna

Przykład liczbowy

- Test na 200 przypadkach: TP = 40, FN = 20, FP = 10, TN = 130
- Accuracy = $170/200 = 0,85$; Precision = $40/50 = 0,80$
- Recall = $40/60 \approx 0,67$; specyficzność = $130/140 \approx 0,93$
- F1 $\approx 0,73$ – wysoka accuracy nie oznacza wykrycia wszystkich pozytywów

Krzywa ROC oraz wpływ progu decyzyjnego



Dobór miary do problemu

- Dane niezbalansowane: accuracy myli (klasyfikator „zawsze negatywny” ma 99%)
- Diagnostyka medyczna, wykrywanie oszustw: priorytet to recall (unikać FN)
- Filtr spamu, rekomendacje: priorytet to precision (unikać FP)
- ROC-AUC i PR-AUC oceniają model niezależnie od progu decyzyjnego

04

Klastrowanie

Idea i zastosowania grupowania

- Cel: grupy obiektów podobnych wewnątrz i różnych między sobą
- Zastosowania: segmentacja klientów, grupowanie dokumentów, wykrywanie anomalii
- Kluczowe pojęcia: miara odległości, reprezentacja obiektów, liczba grup
- Standaryzacja cech jest zwykle niezbędna

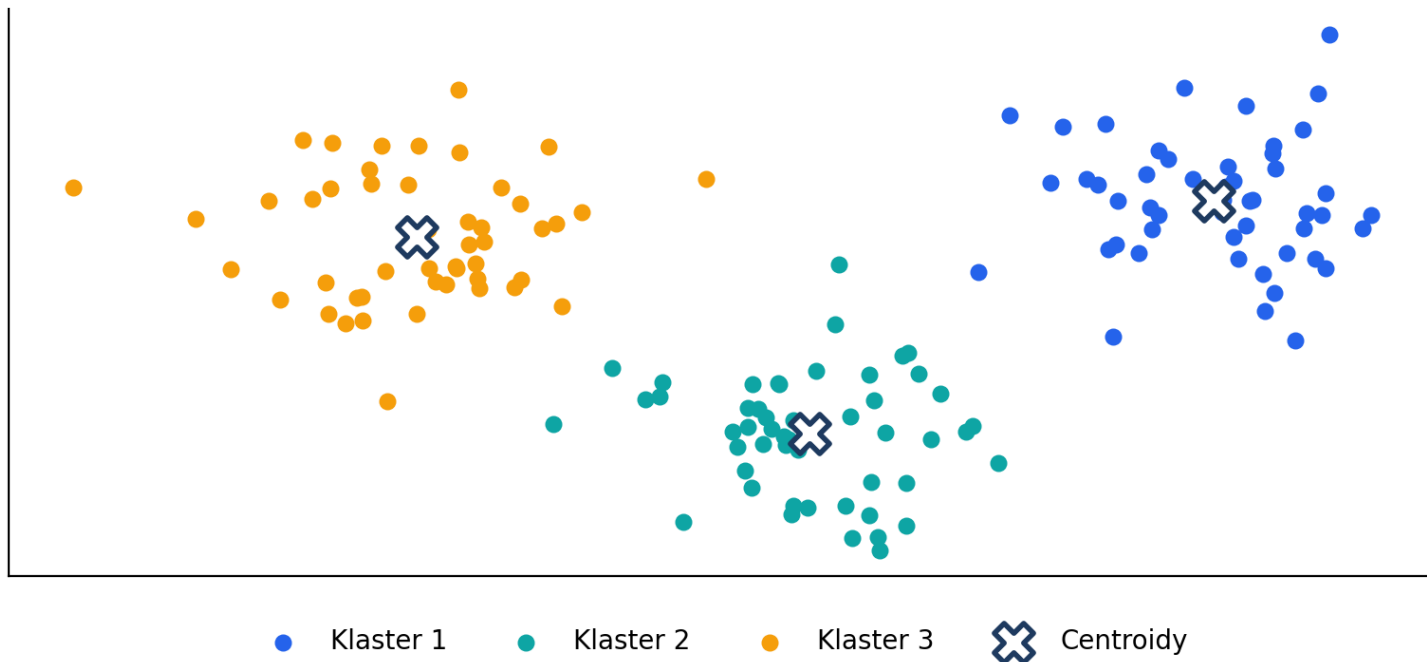
Algorytmy grupowania

- k-średnich: przypisanie do najbliższego centroidu, aktualizacja centroidów, powtarzanie
- Hierarchiczne: dendrogram, łączenie pojedyncze, pełne, Warda
- DBSCAN: grupy jako obszary o dużej gęstości, punkty szumu
- Modele mieszanin Gaussa: miękkie przypisanie, algorytm EM

Ocena i dobór liczby klastrów

- Metoda łokcia: zależność sumy kwadratów odległości od k
- Współczynnik sylwetki (silhouette) – spójność vs. separacja
- Indeks Daviesa-Bouldina, indeks Calinskiego-Harabasza
- Ocena zewnętrzna (gdy znane etykiety): ARI, czystość

Grupowanie metodą k-średnich



05

Sieci neuronowe i uczenie głębokie

Neuron i sieć wielowarstwowa

- Neuron: suma ważona wejść plus obciążenie, przepuszczona przez funkcję aktywacji
- Aktywacje: sigmoid, tanh, ReLU, softmax (wyjście wieloklasowe)
- Warstwy: wejściowa, ukryte, wyjściowa; perceptron wielowarstwowy (MLP)
- Twierdzenie o uniwersalnej aproksymacji

Uczenie sieci

- Funkcja straty: entropia krzyżowa (klasyfikacja), MSE (regresja)
- Propagacja wsteczna: obliczanie gradientów regułą łańcuchową
- Optymalizatory: SGD, momentum, Adam; dobór współczynnika uczenia
- Mini-batche i epoki; problem zanikającego gradientu

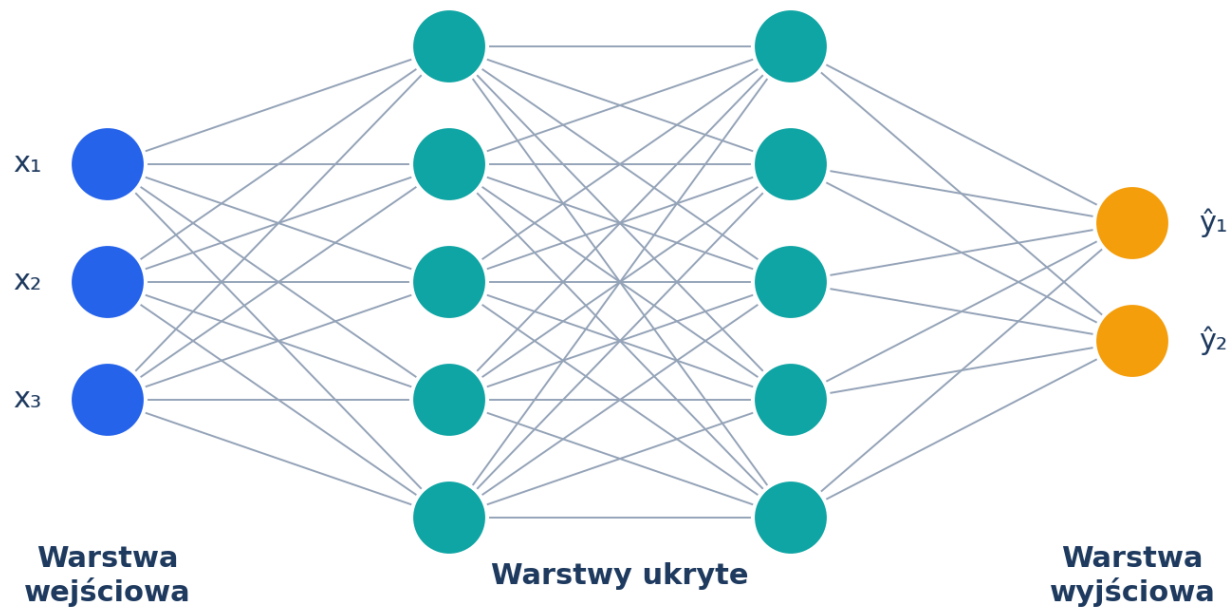
Regularyzacja w głębokich sieciach

- Dropout: losowe wyłączanie neuronów w trakcie treningu
- Normalizacja wsadowa (batch normalization) stabilizuje uczenie
- Augmentacja danych, wczesne zatrzymanie, regularyzacja wag
- Transfer learning: ponowne użycie wstępnie wytrenowanych modeli

Architektury głębokie

- CNN: dane przestrzenne (obrazy)
- RNN, LSTM, GRU: sekwencje i szeregi czasowe
- Transformery: mechanizm uwagi (attention), podstawa dużych modeli językowych
- Autoenkodery i sieci generatywne (GAN, modele dyfuzyjne)

Schemat sieci neuronowej



06

Analiza obrazu i sieci konwolucyjne

Reprezentacja obrazu

- Obraz jako tensor: wysokość $\times$ szerokość $\times$ kanały (np. RGB)
- Klasyczne cechy: krawędzie (Sobel), histogramy, SIFT, HOG
- Problem: w pełni połączone sieci nie skalują się dla obrazów
- CNN wykorzystują lokalność i współdzielenie wag

Warstwa konwolucyjna i pooling

- Filtr (jądro) przesuwa się po obrazie, tworząc mapę cech
- Parametry: rozmiar filtra, krok (stride), dopełnienie (padding)
- Pooling (max, średni) zmniejsza rozdzielczość i zwiększa odporność na przesunięcia
- Głębsze warstwy wykrywają coraz bardziej złożone wzorce: krawędzie → tekstury → obiekty

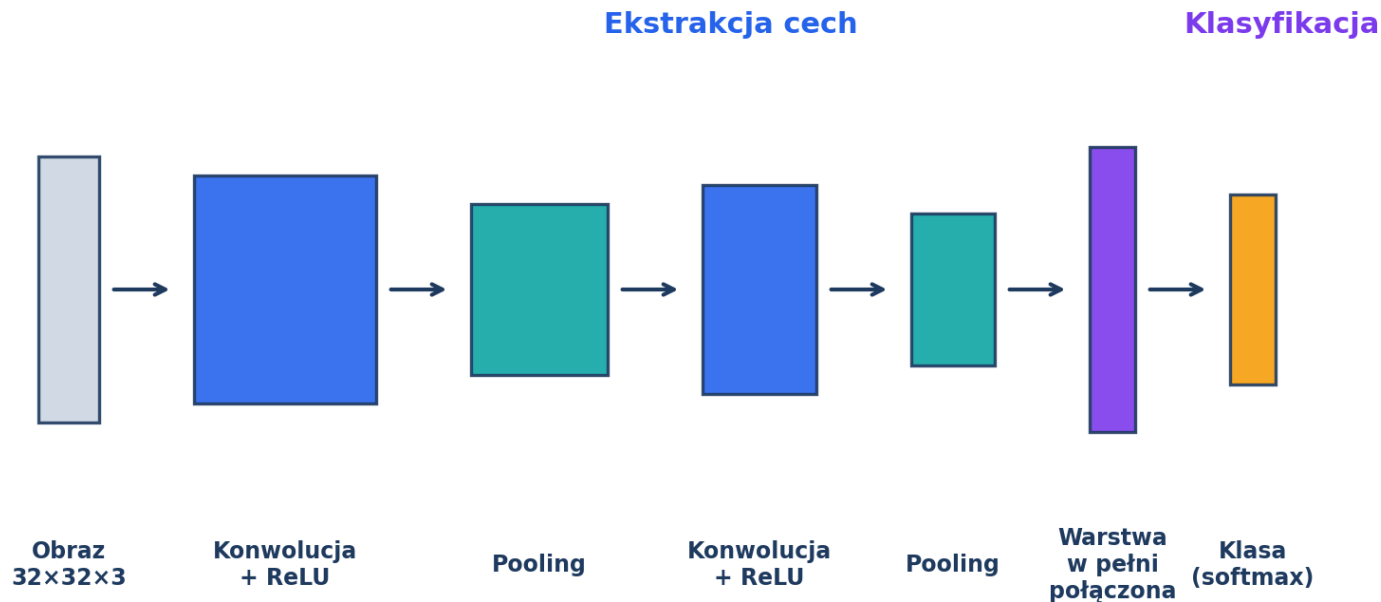
Architektury CNN

- LeNet-5 (1998): rozpoznawanie cyfr
- AlexNet (2012), VGG: coraz głębsze sieci
- ResNet: połączenia rezydualne umożliwiają trening ponad 100 warstw
- EfficientNet, MobileNet: kompromis między dokładnością a kosztem

Zadania widzenia komputerowego

- Klasyfikacja obrazów (ImageNet)
- Detekcja obiektów: YOLO, Faster R-CNN
- Segmentacja semantyczna i instancyjna: U-Net, Mask R-CNN
- Zastosowania: obrazowanie medyczne, kontrola jakości, pojazdy autonomiczne

Architektura sieci konwolucyjnej



Konwolucja krok po kroku

- Filtr (np. 3×3) przesuwa się po obrazie z zadany krokiem (stride)
- W każdej pozycji: mnożenie elementów i sumowanie → jedna wartość mapy cech
- Te same wagi w całym obrazie: mało parametrów, odporność na przesunięcia
- Wartości filtrów są uczone przez propagację wsteczną, a nie projektowane ręcznie

Konwolucja 3×3 na obrazie 5×5

Obraz 5×5

1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	0	0

*

Filtr 3×3

1	0	-1
1	0	-1
1	0	-1

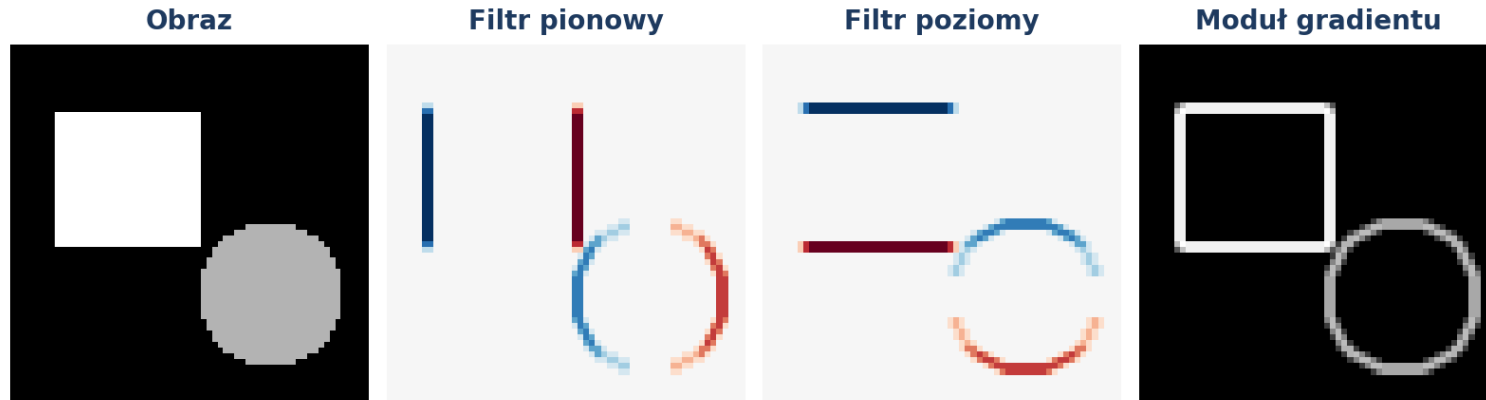
=

Mapa cech 3×3

0	3	3
0	3	3
0	3	3

Wartość w ramce = $3 \cdot (1 \cdot 1 + 1 \cdot 0 + 1 \cdot (-1)) = 0$; duże wartości w mapie cech wskazują pionową krawędź

Filtry krawędziowe na przykładowym obrazie



Filtry krawędziowe Sobela – sieć CNN uczy się podobnych filtrów samodzielnie

Pooling i funkcje aktywacji

- ReLU: $\max(0, x)$ – wprowadza nieliniowość i przyspiesza uczenie
- Max pooling: wartość największa z okna; average pooling: wartość średnia
- Pooling zmniejsza rozmiar map cech i liczbę obliczeń
- Zwiększa odporność na małe przesunięcia i zniekształcenia obrazu

Max i average pooling (okno 2×2)

Mapa cech 4×4

1	3	2	1
4	6	5	2
7	2	9	3
1	8	4	6

Max pooling 2×2

6	5
8	9

Average pooling 2×2

3,5	2,5
4,5	5,5

Rozmiar mapy zmniejsza się czterokrotnie ($4 \times 4 \rightarrow 2 \times 2$), zachowane są najsilniejsze odpowiedzi

Wymiary map cech i liczba parametrów

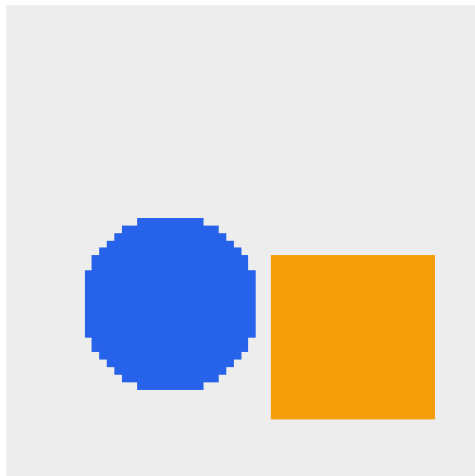
- Rozmiar wyjścia: $(W - F + 2P) / S + 1$ (F – filtr, P – padding, S – stride)
- Parametry warstwy: $(F \cdot F \cdot C_{we} + 1) \cdot C_{wy}$
- Przykład: wejście $32 \times 32 \times 3$, 16 filtrów $3 \times 3 \rightarrow 16 \cdot (27 + 1) = 448$ parametrów
- Warstwa w pełni połączona ma zwykle znacznie więcej parametrów niż konwolucyjna

Uczenie CNN w praktyce

- Augmentacja: obroty, odbicia, przycięcia, zmiany jasności – więcej różnorodnych danych
- Transfer learning: sieć wstępnie wytrenowana na ImageNet (ResNet, EfficientNet)
- Fine-tuning: douczanie ostatnich warstw na własnym, mniejszym zbiorze
- Kontrola overfittingu: krzywe uczenia, dropout, wczesne zatrzymanie

Klasyfikacja, detekcja i segmentacja

Klasyfikacja



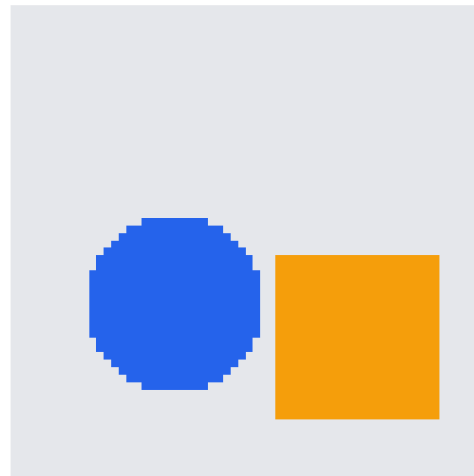
Etykieta: „kształty”

Detekcja obiektów



Ramki + klasy

Segmentacja



Klasa każdego piksela

07

Systemy rekomendacyjne

Idea i rodzaje systemów rekomendacyjnych

- Cel: przewidzieć, które obiekty zainteresują użytkownika (filmy, produkty, muzyka)
- Filtrowanie kolaboratywne: podobni użytkownicy lubią podobne rzeczy
- Oparte na treści: rekomendacja obiektów podobnych do ocenianych wcześniej
- Systemy hybrydowe łączą oba podejścia

Miary podobieństwa

- Podobieństwo cosinusowe: kąt między wektorami, niezależne od długości
- Korelacja Pearsona: uwzględnia różne skale ocen użytkowników
- Indeks Jaccarda: podobieństwo zbiorów (np. kupione produkty)
- Odległość euklidesowa i Manhattan; dobór miary zależy od danych

Filtrowanie kolaboratywne i faktoryzacja

- Macierz użytkownik $\times$ obiekt, w większości pusta (rzadkość danych)
- Metody sąsiedztwa: user-based i item-based
- Faktoryzacja macierzy (SVD, ALS): ukryte czynniki użytkowników i obiektów
- Predykcja oceny: iloczyn skalarny wektorów ukrytych

Dopasowanie obiektów i ocena

- Osadzenia (embeddings) użytkowników i obiektów; szybkie wyszukiwanie najbliższych sąsiadów
- Problem zimnego startu, różnorodność i „bańka filtrująca”
- Metryki: RMSE, precision@k, recall@k, MAP, NDCG
- Testy A/B w środowisku produkcyjnym

Macierz ocen i podobieństwo cosinusowe

Macierz ocen użytkownik × obiekt

Użytkownik 1	5	4	?	1	?
Użytkownik 2	4	5	3	?	1
Użytkownik 3	1	?	4	5	4
Użytkownik 4	?	2	5	4	5
	Film 1	Film 2	Film 3	Film 4	Film 5

Podobieństwo cosinusowe

$$\text{sim}(u, v) = \frac{u \cdot v}{\|u\| \|v\|}$$

Użytkownik 1 i 2 (wspólne filmy):
 $\text{sim} \approx 0.98$

08

Mechanizmy wyszukiwania

Architektura wyszukiwarki

- Crawler zbiera strony, parser wyodrębnia tekst i linki
- Indeksator buduje struktury przyspieszające wyszukiwanie
- Moduł rankingu porządkuje wyniki według trafności
- Interfejs: analiza zapytania, korekta literówek, podpowiedzi

Indeksowanie

- Przetwarzanie tekstu: tokenizacja, normalizacja, usuwanie słów stop, stemming
- Indeks odwrócony: słowo → lista dokumentów (z pozycjami i częstościami)
- Zapytania boolowskie i frazowe realizowane przez operacje na listach
- Kompresja indeksów i dystrybucja na wiele serwerów

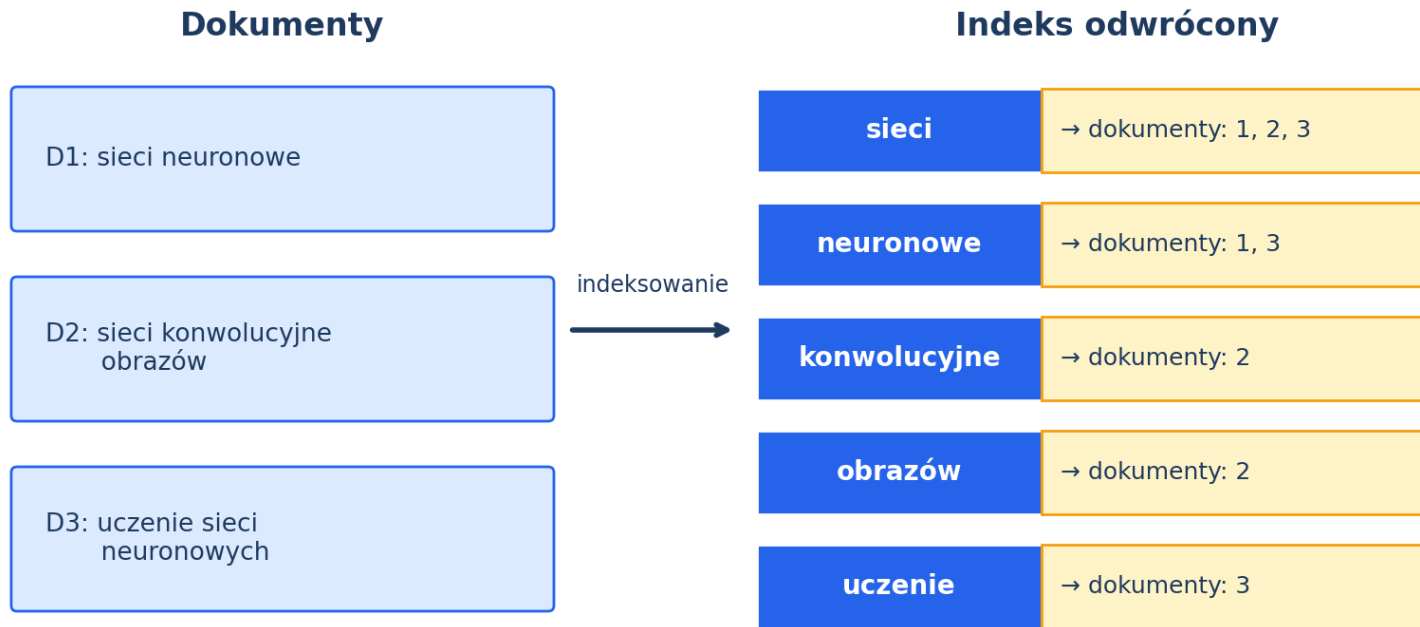
Ranking oparty na treści

- Model wektorowy: dokument i zapytanie jako wektory, podobieństwo cosinusowe
- TF-IDF: częstość słowa w dokumencie ważona rzadkością w kolekcji
- BM25: ulepszona funkcja rankingu z normalizacją długości dokumentu
- Miary jakości: precision, recall, MAP, NDCG

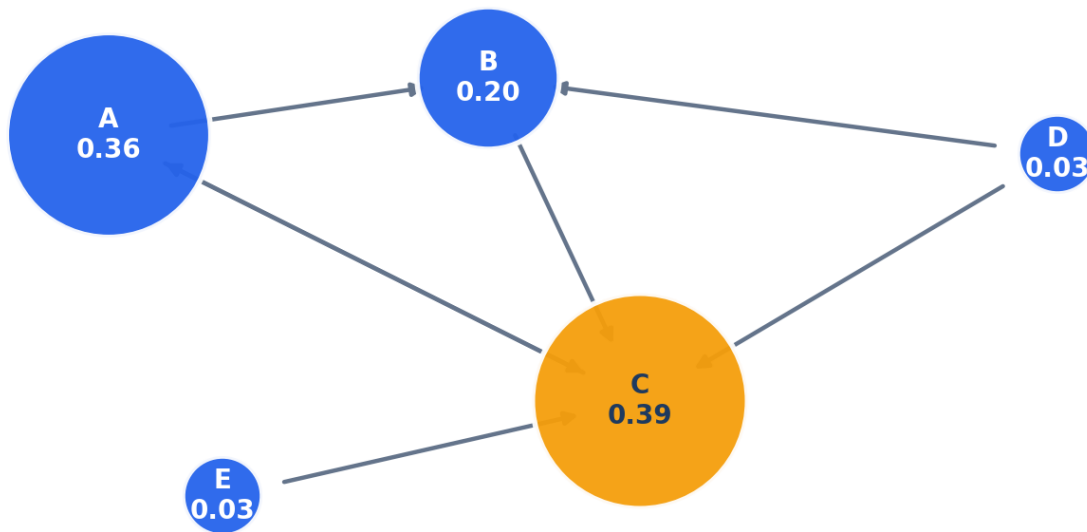
Linki przychodzące i PageRank

- Linki przychodzące to „głosy” innych stron; ważne są ich liczba i jakość
- PageRank: strona jest ważna, gdy linkują do niej ważne strony
- Model losowego surfera: z prawdopodobieństwem $d \approx 0,85$ idzie za linkiem, inaczej skacze losowo
- Obliczenie metodą iteracji potęgowej; ryzyko spamu linkowego (link farms)

Indeks odwrócony (słowo wskazuje listę dokumentów)



Rysunek: przykład PageRank



Rozmiar węzła
= ważność strony
(PageRank)

$d = 0,85$

Podsumowanie i literatura

- AI to zbiór metod: predykcja, grupowanie, sieci neuronowe, rekomendacje i wyszukiwanie
- Jakość danych i dobra walidacja są ważniejsze niż wybór konkretnego algorytmu
- Russell, Norvig – Artificial Intelligence: A Modern Approach
- Goodfellow, Bengio, Courville – Deep Learning
- Manning, Raghavan, Schütze – Introduction to Information Retrieval
- Narzędzia: scikit-learn, PyTorch, TensorFlow, Elasticsearch