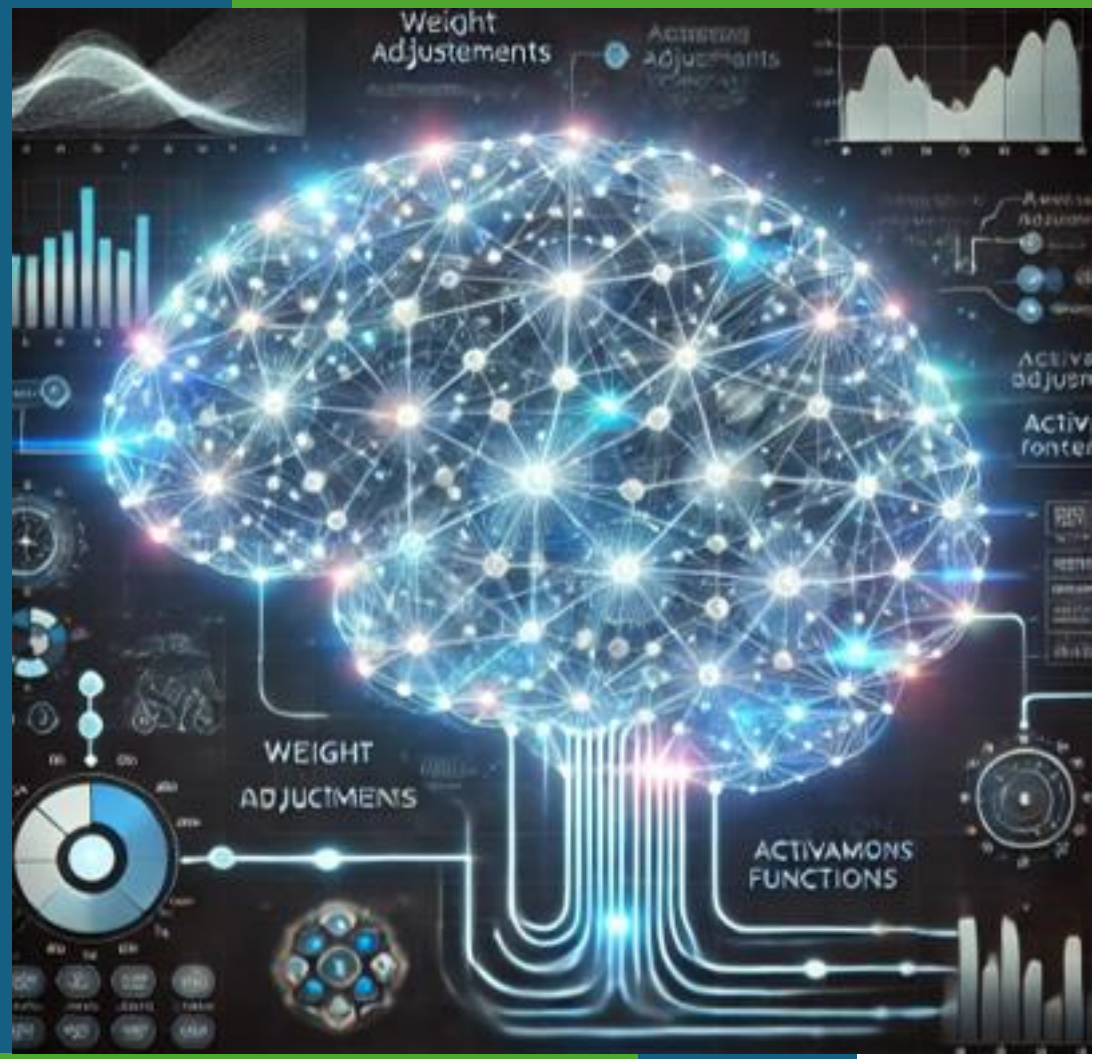
Temat 7 .

Mózg AI – Jak działają sztuczne sieci neuronowe?

dr inż. Michał Malinowski

https://www.drmalinowski.edu.pl

Wprowadzenie do Sztucznej Inteligencji



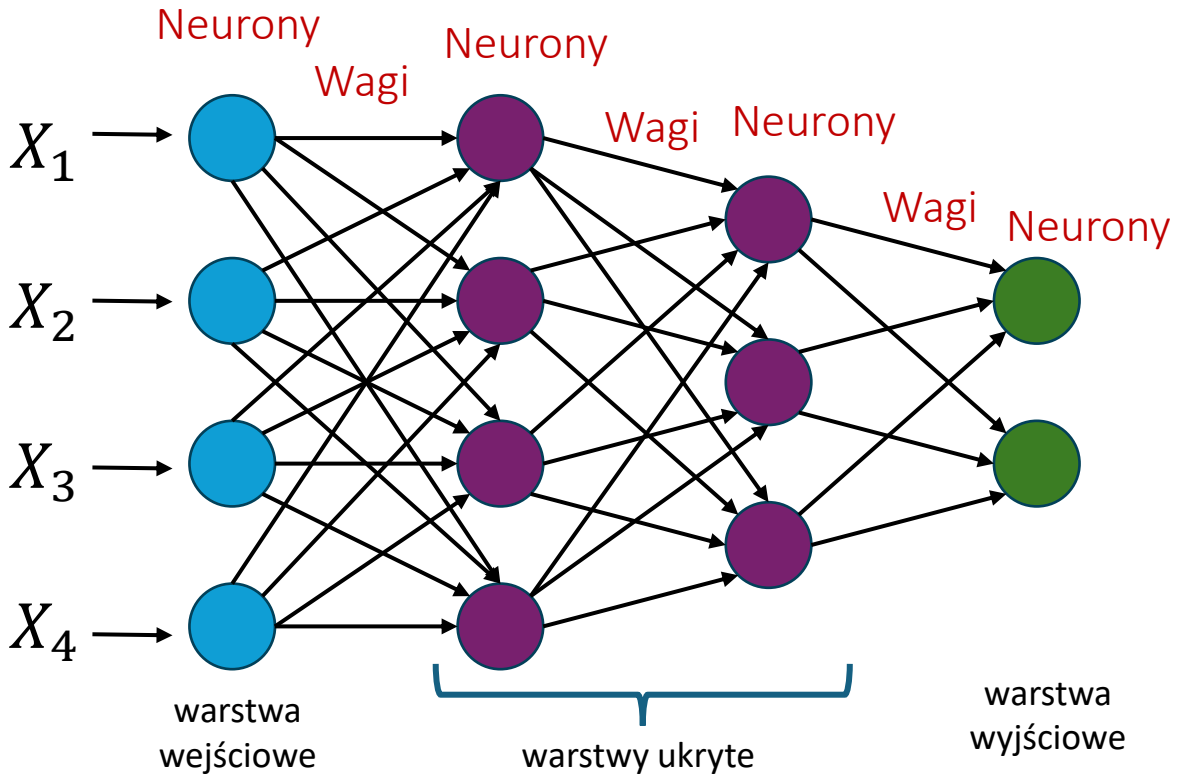
- Sztuczne sieci neuronowe (ANN)
- Głębokie Uczenie Maszynowe (DL)
- Przetwarzanie Języka Naturalnego (NLP)
- Tokenizacja, Lematyzacja, Embedding
- CPU vs GPU
- Bazy wektorowe, Miary podobieństw
- Architektura Transformers
- Duże Modele Językowe (LLMs)
- Generatywna sztuczna inteligencja
- Wizja komputerowa (CV)
- Mechatronika i robotyka, Internet rzeczy (IoT)
- Przetwarzanie Brzegowe

Sztuczne sieci neuronowe (ANN)

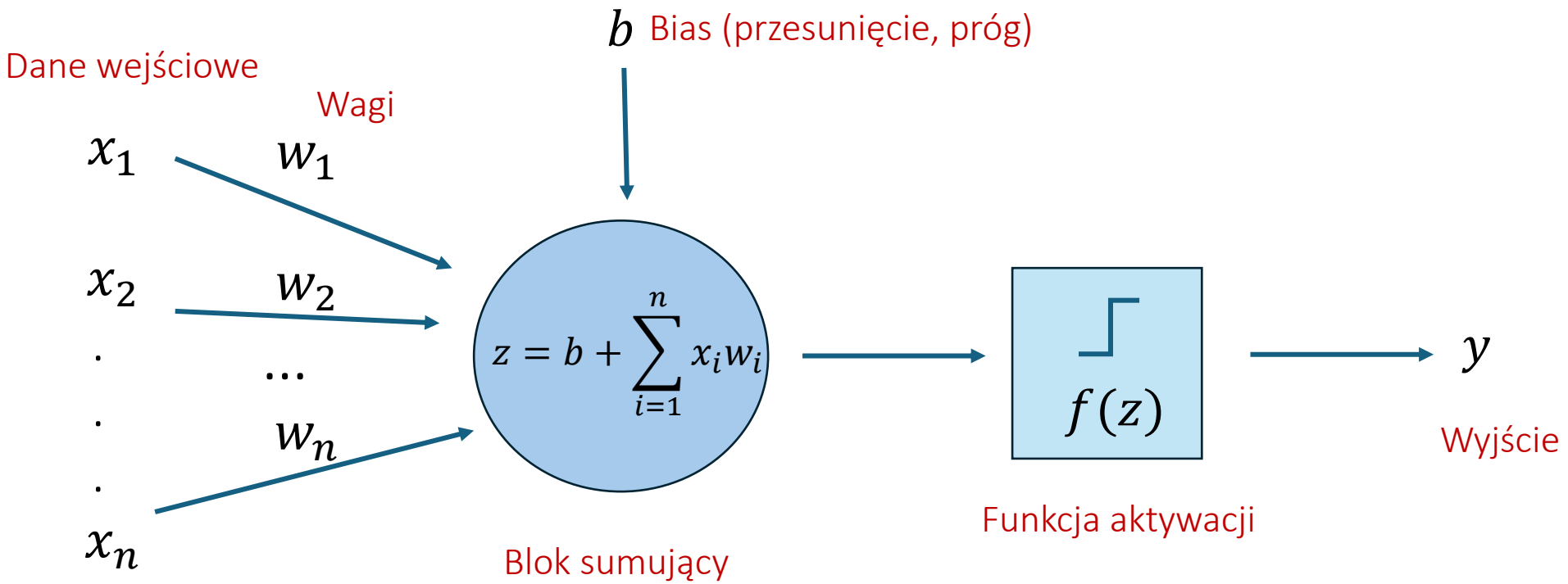
Sztuczne sieci neuronowe (Artificial Neural Networks, ANN)

to struktury obliczeniowe inspirowane biologicznymi sieciami neuronowymi, które naśladują sposób, w jaki ludzki mózg przetwarza informacje. Składają się z neuronów (węzłów) połączonych ze sobą krawędziami (wagi).

- **Warstwa wejściowa**
  odpowiada za wstępne przygotowanie danych wejściowych
- **Warstwa ukryta**
  każda dodatkowa warstwa pomiędzy warstwą wejściową i wyjściową, jej działanie nie może być bezpośrednio obserwowane
- **Warstwa wyjściowa**
  odpowiada za finalną prezentację końcowego wyniku, stanowi jedyne wyjście całej sieci,



Budowa neuronu w sieci neuronowej (perceptronu)



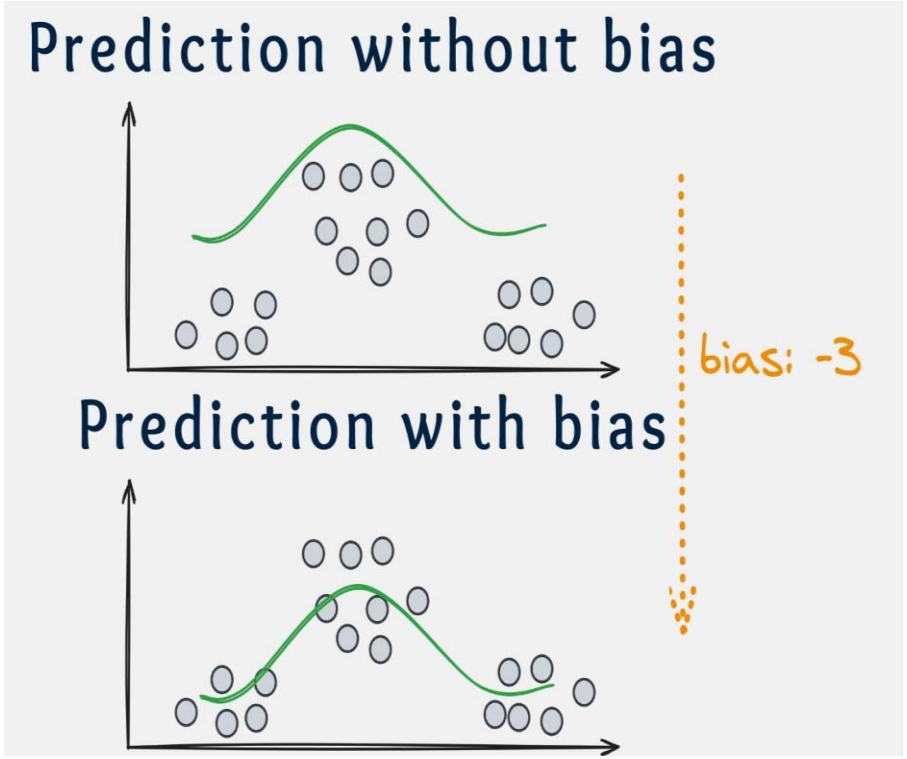
Sztuczne sieci neuronowe (ANN)

Budowa neuronu

Dane wejściowe
Wartości, które są wprowadzane do neuronu. Mogą pochodzić z danych wejściowych całej sieci lub z wyjść innych neuronów.
Symbol: $x_1, x_2, x_3, x_4, \ldots, x_n$

Bias (przesunięcie, próg)
Wartość dodawana do sumy ważonej danych wejściowych, która umożliwia przesunięcie funkcji aktywacji. Bias pozwala modelowi na dopasowanie do danych wejściowych. Dzięki temu neuron może efektywniej reprezentować różne wzorce i zależności w danych.
Symbol: b

Wagi
Parametry, które są modyfikowane podczas procesu uczenia, wpływające na znaczenie poszczególnych danych wejściowych. Każda dana wejściowa jest mnożona przez odpowiednią wagę.
Symbol: $w_1, w_2, w_3, w_4, \ldots, w_n$



Sztuczne sieci neuronowe (ANN)

Budowa neuronu

Blok sumujący
Element neuronu, który sumuje iloczyny danych wejściowych i ich odpowiednich wag. Wynik tej sumy jest przekazywany do funkcji aktywacji.
Równanie: $\mathrm{z} = \mathrm{b} + \sum_{i=1}^{n} x_i\, w_i$

Funkcja aktywacji
Funkcja matematyczna, która przekształca sumę ważoną danych wejściowych. Decyduje, czy neuron zostanie aktywowany. Istnieje wiele typów funkcji aktywacji.
Symbol: $f(z)$

Wyjście
Wynik przetworzenia danych wejściowych przez neuron, po przejściu przez funkcję aktywacji. Może być przekazywany jako dane wejściowe do kolejnych neuronów lub stanowić ostateczny wynik działania sieci neuronowej.
Symbol: y

Sztuczne sieci neuronowe (ANN)

Funkcja aktywacji

Aktywacja neuronu następuje, gdy **suma ważona** danych wejściowych, przetworzona przez **funkcję aktywacji**, przekroczy określony **próg.**
Wartość zwrócona przez funkcję aktywacji decyduje o sile sygnału wysyłanego dalej przez neuron, co wpływa na aktywność kolejnych neuronów w sieci.

Metody ustalenia progu aktywacyjnego

Trening Modelu

Dynamiczne dostosowywanie progu w trakcie treningu sieci neuronowej.

Wbudowany Próg

Statycznie ustawiony próg, np. w funkcji ReLU z progiem 0.

Projektowanie Architektury

Określanie wartości progowych na etapie budowy modelu, np. w warstwach sieci neuronowej.

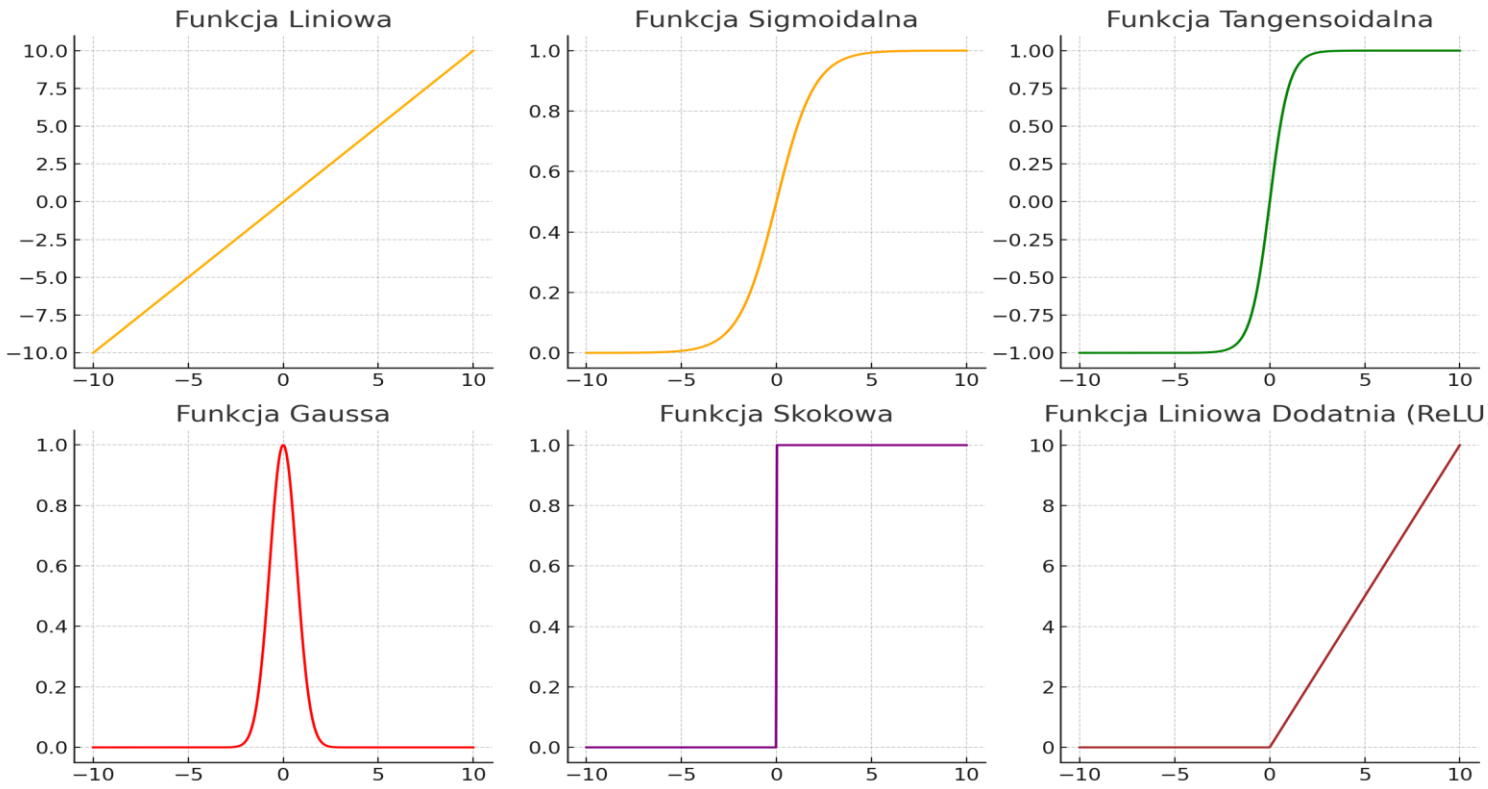
Algorytmy Heurystyczne

Dostosowywanie progu na podstawie doświadczenia i analizy danych.

Sztuczne sieci neuronowe (ANN)

Rodzaje funkcji aktywacji

Funkcja aktywacji	Neuron aktywny (y)	Neuron nieaktywny (y)
Liniowa	Aktywny $y > 0$	Brak nieaktywności $y \leq 0$
Sigmoidalna	y bliskie 1	y bliskie 0
Tangensoidalna (Tanh)	y bliskie 1 lub −1	y bliskie 0
Gaussa	y bliskie 1 (dla $z \approx 0$)	y bliskie 0 (dla $z \to \pm\infty$)
Skokowa (Step Function)	$y = 1$	$y = 0$
ReLU	$y > 0$	$y = 0$



Sztuczne sieci neuronowe (ANN)

Rodzaje funkcji aktywacji

Funkcja Aktywacji	Zastosowanie	Przykład	Położenie w Warstwach Sieci
Liniowa	Regresja liniowa, modele predykcyjne dla wartości ciągłych.	Prognozowanie cen mieszkań, analiza sprzedaży.	Ostateczna warstwa w sieciach neuronowych do regresji.
Sigmoidalna	Klasyfikacja binarna, modele podejmujące decyzje w skali 0-1.	Wykrywanie spamu w e-mailach, diagnoza medyczna (choroba/zdrowie).	Ostateczna warstwa w klasyfikatorach binarnych, ukryte warstwy.
Tangensoidalna (Tanh)	Klasyfikacja i regresja, normalizacja wartości w przedziale (-1, 1).	Analiza szeregów czasowych, rozpoznawanie emocji w tekście.	Ukryte warstwy w sieciach neuronowych.
Gaussowska (RBF)	Sieci radialne, modelowanie nieliniowych zależności w danych.	Aproksymacja funkcji, systemy sterowania robotami.	Ukryte warstwy, specjalistyczne sieci radialne (RBF).
Skokowa (Step Function)	Proste modele klasyfikacji binarnej, perceptrony jednopoziomowe.	Klasyfikacja „tak/nie”, proste rozpoznawanie wzorców.	Ukryte warstwy, ostateczna warstwa w perceptronach jednopoziomowych.
ReLU (Rectified Linear Unit)	Głębokie sieci neuronowe, rozwiązanie problemu zanikania gradientu.	Rozpoznawanie obrazów, sieci konwolucyjne (CNN).	Ukryte warstwy w głębokich sieciach neuronowych.

Sztuczne sieci neuronowe (ANN)

Przykład

Czy kandydat nadaje się na stanowiska kierownika projektu informatycznego? (1/0)

Wejście (cechy kandydata):

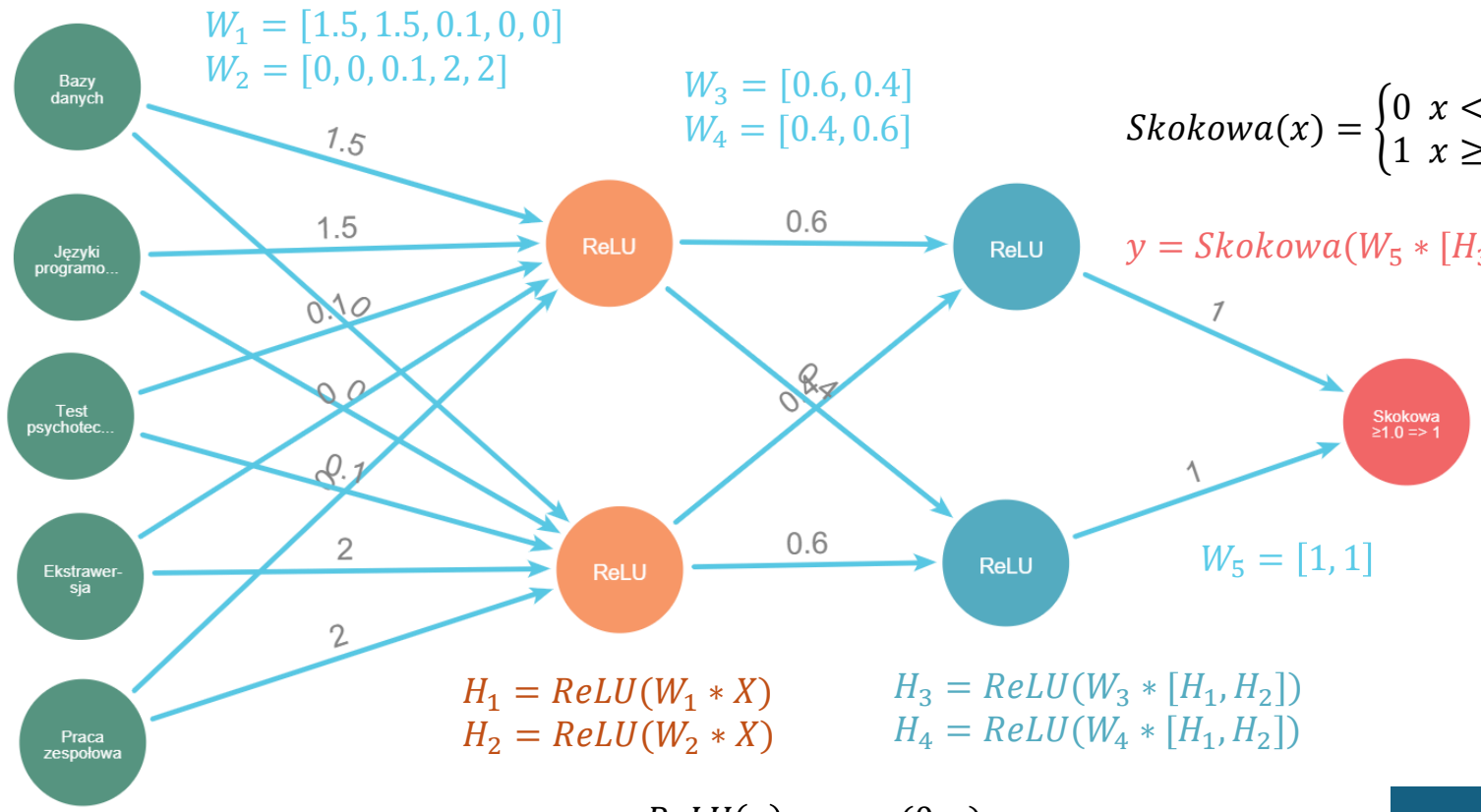
X_1 - Znajomość baz danych (0–10)

X_2 - Znajomość języków programowania (0–10)

X_3 - Wynik testu psychotechniczny (0–10)

X_4 - Umiejętność pracy zespołowej (0–10)

X_5 - Ekstrawersja (0–10)



Baza grafowa Neo4j

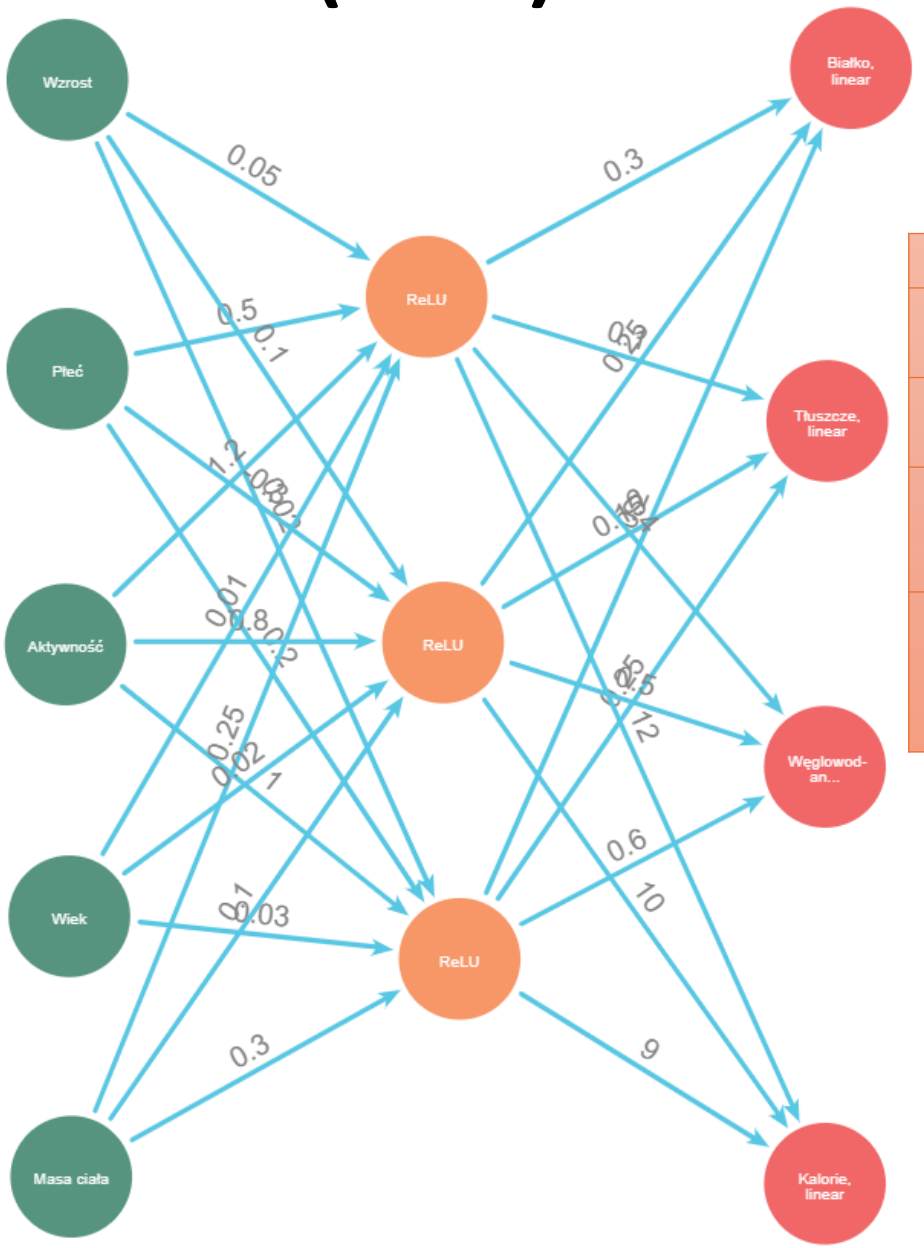
$ReLU(x) = \max(0, x)$

Sztuczne sieci neuronowe (ANN)

Przykład Dopasowanie dietę do osoby

Dane wejściowe

Nr	Cecha	Typ	Zakres typowy	Opis
1	Wiek	liczba	1–100 lat	Wiek osoby
2	Masa ciała	liczba	30–150 kg	Aktualna masa ciała
3	Wzrost	liczba	120–210 cm	Wzrost w centymetrach
4	Płeć	Binarny	0 = kobieta 1 = mężczyzna	Biologiczna płeć (upraszczająco w modelu binarnym)
5	Aktywność fizyczna	liczba	0.0–1.0	Subiektywna intensywność aktywności fizycznej (np. 0.3 = siedzący tryb życia, 1.0 = sportowiec)



Baza grafowa Neo4j

Dane wyjściowe

Nr	Składnik	Jednostka	Znaczenie
1	Białko	gramy	Zalecana dzienna ilość spożycia białka
2	Tłuszcze	gramy	Zalecana dzienna ilość spożycia tłuszczów
3	Węglowodany	gramy	Zalecana dzienna ilość spożycia węglowodanów
4	Kalorie	kcal	Szacowane dzienne zapotrzebowanie kaloryczne (na podstawie TDEE)

Sztuczne sieci neuronowe (ANN)

Zależności

Element	Od czego zależy	Uwagi
Liczba neuronów w warstwie wejściowej	**Od liczby cech danych wejściowych**. (Np. 28×28=784 piksele dla obrazka.)	Musi być równa liczbie zmiennych opisujących pojedynczy przykład.
Liczba warstw ukrytych (hidden layers)	**Od złożoności zadania** i **poziomu reprezentacji**, który chcemy uzyskać.	Więcej warstw → większa zdolność do reprezentowania złożonych funkcji (tzw. głębokie sieci).
Liczba neuronów w warstwach ukrytych	**Od potrzebnej "mocy obliczeniowej"**, czyli jak dużo wzorców musi model nauczyć się rozpoznawać.	Dobiera się często eksperymentalnie, np. zmniejszając liczbę neuronów w kolejnych warstwach.
Liczba neuronów w warstwie wyjściowej	**Od liczby klas lub wymiarów wyniku**, który chcemy uzyskać.	Przykład: 1 neuron dla regresji, N neuronów dla klasyfikacji N-klasowej.

Zadanie

Zwróć uwagę na fragment kodu budujący model AI



Treść

W Google Colab https://colab.research.google.com/ stwórz prosty system uczenia maszynowego bazujący na sztucznej sieci neuronowej.

1. Na podstawie danych treningowych zawartych w pliku mieszkania.csv wytrenuj model prostej sieci neuronowej, która będzie dokonywać predykcji Cena mieszkań na podstawie Liczba_pokoi i Powierzchnia.
2. Odczytaj z modelu i przedstaw warstwy i liczby neuronów w każdej warstwie tego wytrenowanego modelu.
3. Odczytaj z modelu i przedstaw wagi poszczególnych krawędzi tego wytrenowanego modelu.
4. Na podstawie wytrenowanego modelu dokonaj predykcji ceny dla mieszkań: liczba pokoi 4, powierzchnia 130, liczba pokoi 1 powierzchnia 32

https://colab.research.google.com/drive/1zz0ugBnnwRW5VgM48o2ATgt2yJpV0gdH?usp=sharing

Głębokie Uczenie Maszynowe (DL)

Perceptron Wielowarstwowy (Multilayer Perceptron, MLP)

to **podstawowy model sztucznej sieci neuronowej**, składający się z:
✓ **Warstwy wejściowej** – przyjmuje dane wejściowe.
✓ **Jednej lub więcej warstw ukrytych** – przekształca dane i uczy się wzorców.
✓ **Warstwy wyjściowej** – generuje finalną predykcję.

MLP stosuje **funkcje aktywacji** (np. ReLU, sigmoidę) i uczy się poprzez **propagację wsteczną (backpropagation)**, co pozwala modelowi optymalizować wagi i poprawiać dokładność.

Zastosowanie:
✅ Rozpoznawanie wzorców w danych (np. klasyfikacja obrazów).
✅ Prognozowanie wartości numerycznych (np. ceny akcji).
✅ Analiza sentymentu w tekstach.

Głębokie Uczenie Maszynowe (DL)

Głębokie Uczenie Maszynowe (Deep Learning, DL)

to **zaawansowana technika uczenia maszynowego**, która wykorzystuje **głębokie sieci neuronowe** składające się z **wielu warstw** do analizy i przetwarzania danych.

✓ **Automatyczne wykrywanie reprezentacji danych** – model samodzielnie identyfikuje wzorce w surowych danych.
✓ **Wysoka skuteczność w przetwarzaniu dużych zbiorów danych** – szczególnie w zadaniach wymagających rozpoznawania obrazów, dźwięku czy języka naturalnego.
✓ **Zastosowanie GPU i TPU** – umożliwia wydajne trenowanie dużych modeli.

Zastosowanie:

✅ Rozpoznawanie obrazów i twarzy.
✅ Przetwarzanie języka naturalnego (chatboty, tłumaczenie maszynowe).
✅ Autonomiczne pojazdy i systemy sterowania robotami.

Głębokie Uczenie Maszynowe (DL)

Różnica pomiędzy Uczeniem Maszynowym a Głębokim Uczeniem Maszynowym

Aspekt	Uczenie Maszynowe (Machine Learning)	Głębokie Uczenie Maszynowe (Deep Learning)
Struktura modeli	**Modele płytkie**, z jedną lub kilkoma warstwami przetwarzania.	**Głębokie sieci neuronowe** z wieloma warstwami przetwarzania (głębokie sieci).
Inżynieria cech (Feature Engineering)	**Wymaga ręcznego wyodrębniania i przekształcania cech** z danych (np. konwersja tekstu na liczby, normalizacja danych).	**Automatyczne wyodrębnianie cech** – model sam uczy się istotnych cech z surowych danych (np. wykrywanie wzorców w obrazach).
Skalowalność	**Mniej skuteczne** przy dużych zbiorach danych i złożonych zadaniach.	**Bardzo skuteczne** przy ogromnych zbiorach danych i skomplikowanych problemach.
Wymagania obliczeniowe	**Niższe** – można trenować na standardowych procesorach (CPU).	**Wysokie** – wymagają specjalistycznych jednostek obliczeniowych (GPU, TPU).
Przykłady zastosowań	**Klasyfikacja, regresja, analiza skupień** (np. przewidywanie cen mieszkań, wykrywanie oszustw bankowych).	**Rozpoznawanie obrazów, przetwarzanie języka naturalnego, autonomiczne pojazdy** (np. systemy rozpoznawania twarzy, chatboty AI).

Głębokie Uczenie Maszynowe (DL)

Wielkość wersji ChataGPT

Model AI	Liczba Parametrów (waga + bias)	Liczba Warstw	Długość Kontekstu (liczba tokenów)	Rozmiar Warstwy Ukrytej (liczba neuronów)
GPT-1	117 000 000	12	512	768
GPT-2	1 500 000 000	48	1 024	1 600
GPT-3	175 000 000 000	96	2 048	12 288
GPT-4	1 700 000 000 000	120	8k, 32k oraz 128k	16 384

Głębokie Uczenie Maszynowe (DL)

Parametry modelu

to wewnętrzne wartości liczbowe, które model uczenia maszynowego (np. GPT, Gemini, Copilot) **uczy się** podczas treningu, aby jak najlepiej dopasować swoje predykcje do danych.

Są one kluczowe dla działania sieci neuronowych i determinują ich zdolność do przetwarzania informacji.

Główne parametry w modelach AI:

🧠 Wagi (Weights)

- Określają **siłę połączeń** między neuronami w sieci.
- Są dostosowywane w trakcie treningu, aby poprawić dokładność modelu.
- Im więcej wag, tym większa zdolność modelu do rozpoznawania skomplikowanych wzorców.

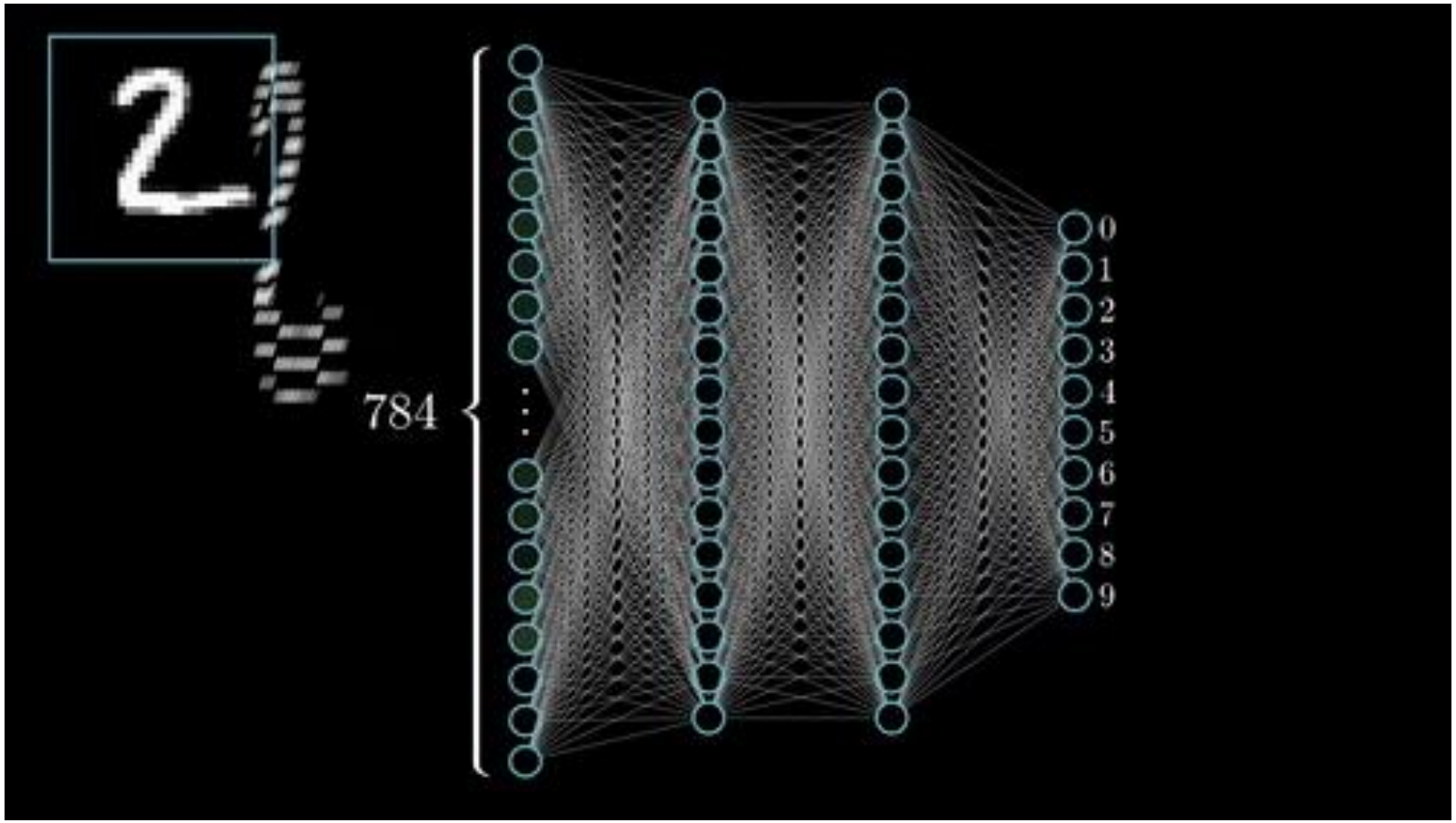
⚖️ Bias (Przesunięcie, odchylenie)

- To dodatkowa wartość, która pomaga neuronowi w podejmowaniu decyzji.
- Zapobiega sytuacji, w której wszystkie wyjścia sieci wynoszą 0 (czyli model jest "zablokowany").

Głębokie Uczenie Maszynowe (DL)

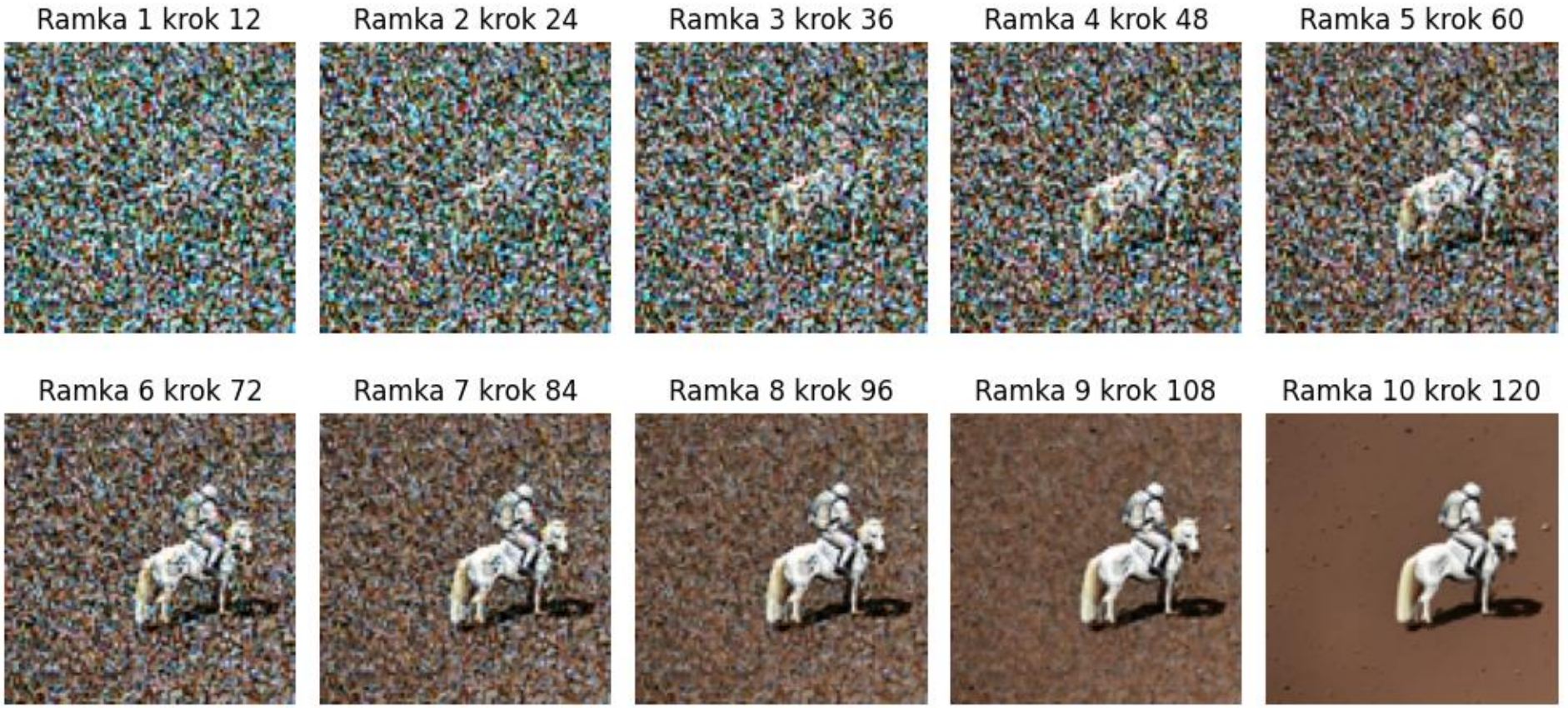
Rozpoznawanie cyfr

- **Sztuczna sieć neuronowa (ANN)** używana do **rozpoznawania cyfr ręcznie pisanych**.
- Każda cyfra jest reprezentowana jako zbiór **784 pikseli wejściowych** (28x28), które są przekształcane przez warstwy ukryte.
- Na końcu sieć klasyfikuje obraz jako jedną z cyfr **0-9**.



Generatywne AI

Generatywne AI a nie Kreatywne AI



Ludzka kreatywność



Efekt kobry (ang. Cobra Effect)

W ramach zwalczania szkodników w Delhi brytyjski rząd wyznaczył nagrodę pieniężną za martwe kobry. Ludzie zaczęli więc hodować kobry, żeby odebrać nagrodę. Wtedy Brytyjczycy zrezygnowali z wypłacania nagród. Na co ludzie wypuścili bezwartościowe kobry na wolność. Skutek: więcej kobr.



Dziś nie jeździ (hiszpański „Hoy No Circula”)

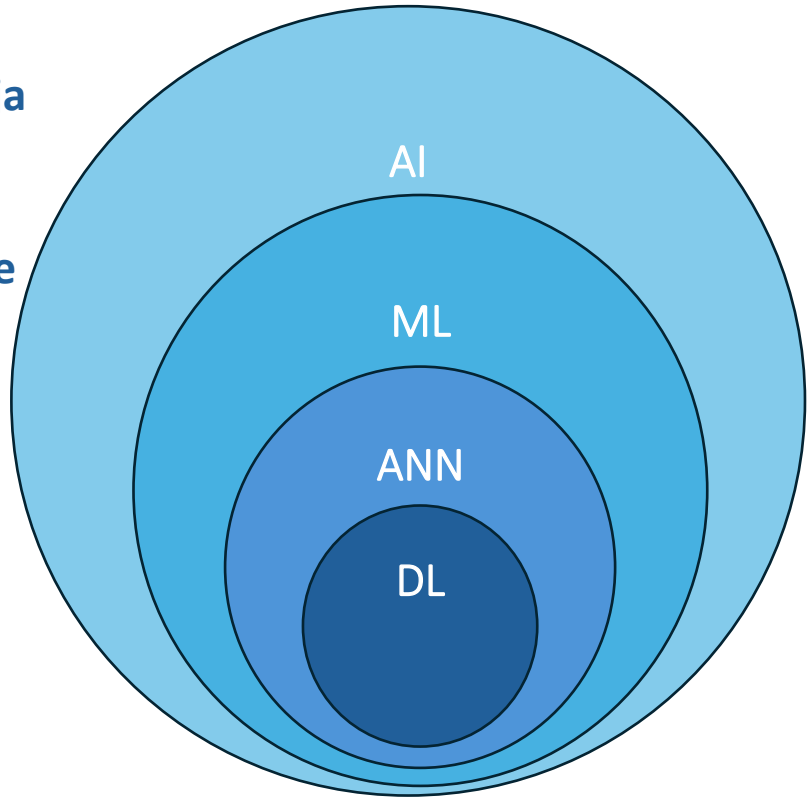
W roku 1989 miasto Meksyk spróbowało zredukować zanieczyszczenie powietrza, wydając zakaz jazdy konkretnym samochodom w określone dni. Niestety, zamiast jeździć autobusami, ludzie po prostu kupili więcej aut, żeby móc korzystać z tego, którego w danym dniu nie obejmował zakaz.

Głębokie Uczenie Maszynowe (DL)

Materiały

https://www.youtube.com/watch?v=jmmW0F0biz0

https://www.youtube.com/watch?v=aircAruvnKk

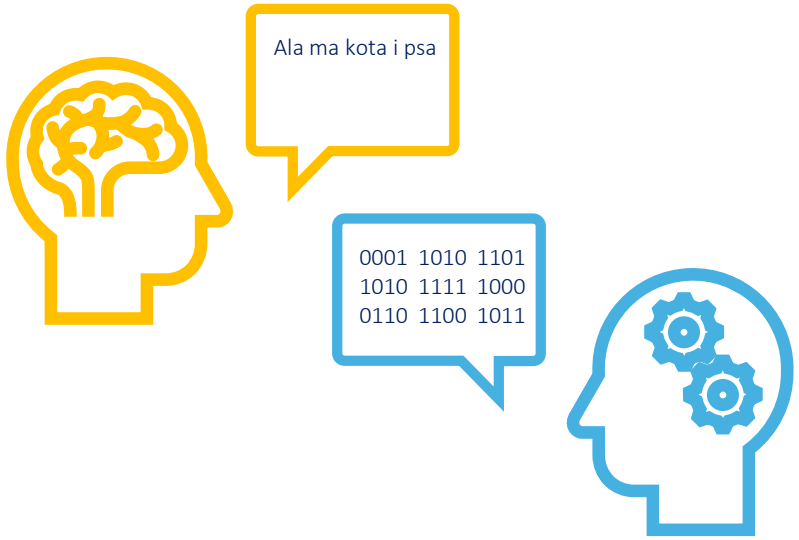


Przetwarzanie Języka Naturalnego (NLP)

Przetwarzanie Języka Naturalnego (Natural Language Processing, NLP)

to dziedzina **sztucznej inteligencji** i **informatyki**, zajmująca się analizą, interpretacją oraz generowaniem tekstu i mowy w językach naturalnych.
NLP pozwala komputerom **rozumieć, przekształcać i reagować** na język używany przez ludzi.

Kluczowe aspekty NLP:
✓ **Analiza składniowa i semantyczna** – rozpoznawanie struktury zdań i znaczenia słów.
✓ **Tłumaczenie maszynowe** – konwersja tekstu między różnymi językami (np. Google Translate).
✓ **Analiza sentymentu** – wykrywanie emocji w tekście (np. pozytywne/negatywne recenzje).
✓ **Rozpoznawanie mowy** – konwersja dźwięku na tekst (np. asystenci głosowi, Siri, Alexa).
✓ **Generowanie języka naturalnego (NLG)** – tworzenie treści przez AI (np. ChatGPT, Gemini).



Przetwarzanie Języka Naturalnego (NLP)

Etapy Przetwarzania Języka Naturalnego

Przygotowanie danych	Analiza składniowa	Analiza semantyczna	Analiza pragmatyczna	Generowanie języka	Ocena i optymalizacja
Tokenizacja	Rozbiór gramatyczny (Parsing)	Rozpoznawanie encji (nazw własnych, NER)	Identyfikacja intencji użytkownika	Tworzenie streszczeń tekstu	Metryki jakości (BLEU, ROUGE, perplexity)
Usuwanie znaków specjalnych	Przypisywanie części mowy do słów (POS-tagging)	Rozróżnianie znaczeń wyrazów (WSD)	Analiza sentymentu	Automatyczna odpowiedź na pytania	Uczenie transferowe
Normalizacja tekstu	Rozpoznawanie fraz	Analiza zależności semantycznych	Wykrywanie ironii i sarkazmu	Tłumaczenie maszynowe	Fine-tuning
Lematyzacja				Chatboty i systemy dialogowe	RAG
Usuwanie stop-words					
Embeding					

Przetwarzanie Języka Naturalnego (NLP)

Techniki i Algorytmy NLP

TF-IDF (Term Frequency-Inverse Document Frequency):	Technika statystyczna używana do oceny znaczenia słowa w dokumencie w kontekście całego korpusu.
	Przykład: Często występujące słowa w dokumencie mają wyższą wagę, ale ich znaczenie jest zmniejszane, jeśli występują często w całym korpusie.
Word Embeddings:	Reprezentacja słów w formie wektorów w przestrzeni wielowymiarowej, gdzie semantycznie podobne słowa są bliżej siebie.
	Przykłady: Word2Vec, GloVe.
Modele językowe:	Modele statystyczne lub neuronowe, które przewidują prawdopodobieństwo sekwencji słów.
	Przykłady: BERT, GPT-3, Transformer.
Sieci Neuronowe:	Zaawansowane modele uczenia maszynowego, które uczą się złożonych zależności między słowami i frazami.
	Przykłady: Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Transformers.

Przetwarzanie Języka Naturalnego (NLP)

Zastosowanie

Obszar	Opis	Przykłady
Tłumaczenie Automatyczne	Konwersja tekstu z jednego języka na inny z zachowaniem kontekstu i znaczenia.	Google Translate, DeepL
Chatboty i Asystenci Głosowi	Interakcja z użytkownikami poprzez rozumienie i generowanie języka naturalnego.	Siri, Alexa, Google Assistant
Rozpoznawanie Mowy	Konwersja mowy na tekst oraz analiza i klasyfikacja dźwięków mowy.	Systemy rozpoznawania mowy, asystenci głosowi
Ekstrakcja Informacji	Wydobywanie specyficznych informacji z dużych zbiorów tekstowych.	Ekstrakcja danych z dokumentów prawnych, raportów finansowych
Klasyfikacja Tekstu	Przypisywanie dokumentów lub fragmentów tekstu do określonych kategorii.	Filtracja spamu, klasyfikacja wiadomości
Streszczenie Tekstu	Tworzenie zwięzłych podsumowań długich dokumentów lub artykułów.	Automatyczne streszczanie artykułów, podsumowania raportów
Analiza Semantyczna	Analiza znaczenia tekstu, zrozumienie kontekstu i relacji między słowami.	Systemy wyszukiwania informacji, analizy semantyczne
Rozpoznawanie NER (Named Entity Recognition)	Identyfikacja nazw własnych w tekście, takich jak imiona, miejsca, organizacje.	Analiza dokumentów prawnych, raportów, artykułów
Analiza Sentimentów Finansowych	Analiza nastrojów w artykułach finansowych, mediach społecznościowych w kontekście rynku.	Przewidywanie ruchów giełdowych, analiza nastrojów inwestorów

Tokenizacja

Tokenizacja

to proces **dzielenia tekstu** na mniejsze jednostki zwane **tokenami**. Tokeny mogą być **słowami, znakami, frazami lub subword**

Tokenizacja to **podstawowy krok w przetwarzaniu języka naturalnego (NLP)**, pozwalający komputerom lepiej analizować tekst.

Zastosowanie

- ✅ Wyszukiwanie tekstowe i analiza sentymentu
- ✅ Tłumaczenie maszynowe
- ✅ Generowanie tekstu przez modele AI
- ✅ Systemy rozpoznawania mowy

Tokenizacja

Rodzaje tokenizacji

1 **Tokenizacja na poziomie słów** (*Word Tokenization*)
Podział tekstu na pojedyncze słowa. Przykład: *„Ala ma kota.”* → [„Ala”, „ma”, „kota”, „.”]

2 **Tokenizacja na poziomie znaków** (*Character Tokenization*)
Każdy znak traktowany jest jako oddzielny token. Przykład: *„Ala”* → [„A”, „l”, „a”]

3 **Tokenizacja na poziomie subwordów** (*Subword Tokenization*)
Podział na mniejsze jednostki, aby lepiej radzić sobie z nowymi słowami. Przykład: *„czytający”* → [„czyt”, „ający”]

4 **Tokenizacja na poziomie zdań** (*Sentence Tokenization*)
Podział na pełne zdania, często używane do analizy większych fragmentów tekstu.
Przykład: *„Ala ma kota. Kot jest biały.”* → [„Ala ma kota.”, „Kot jest biały.”]

Tokenizer OpenAI

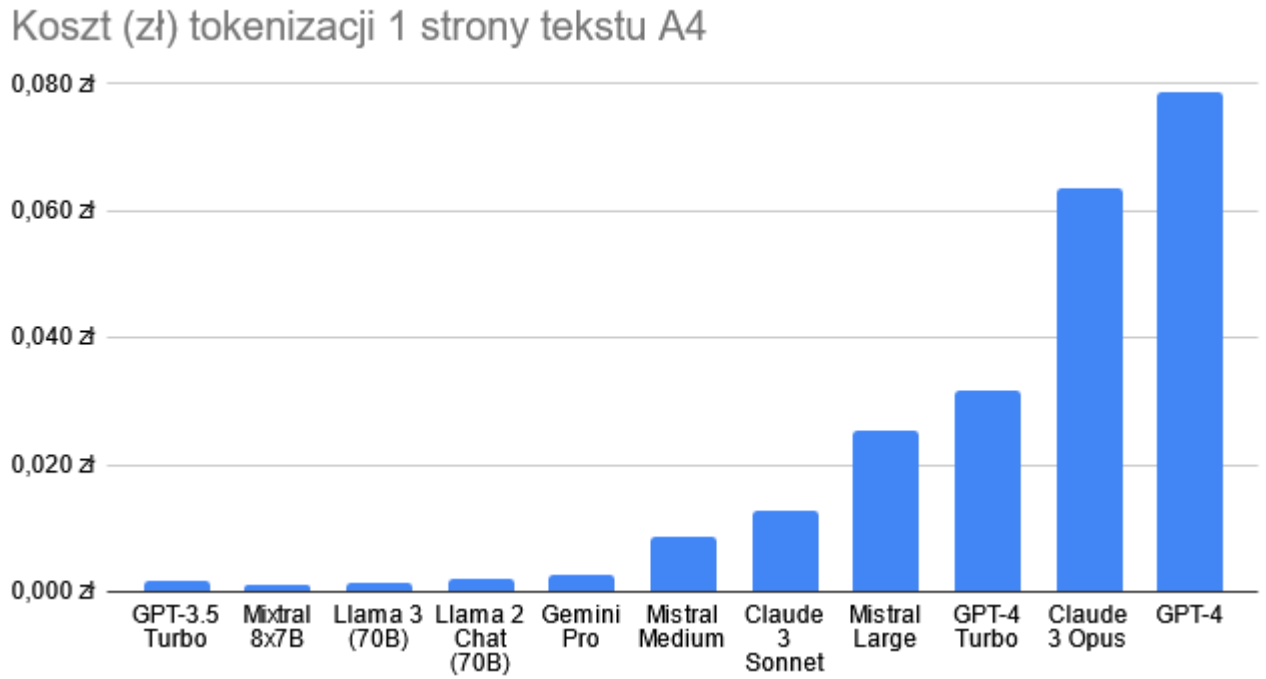
GPT-3.5 & GPT-4

Tokens	Characters		Tokens	Characters	
79	176	>	48	191	

Litwo, Ojczyzno moja! ty jesteś jak zdrowie;
Ile cię trzeba cenić, ten tylko się dowie,
Kto cię stracił. Dziś piękność twą w całej ozdobie
Widzę i opisuję, bo tęsknię po tobie.

Lithuania, my homeland! you are like health;
How much you should be valued, only he will know,
Who has lost you. Today your beauty in all its adornment
I see and describe, because I miss you.

Tokenizer
https://platform.openai.com/tokenizer



Lematyzacja

Lematyzacja

to proces przekształcania wyrazu do jego podstawowej, **słownikowej formy** (tzw. **lematu**). Jest kluczową techniką w **przetwarzaniu języka naturalnego (NLP)**, stosowaną w systemach sztucznej inteligencji do analizy tekstu.

✅ Przykłady lematyzacji (język polski):

Forma w tekście	Lemmatyzacja (forma słownikowa)
biegnący	biegać
byliśmy	być
samochody	samochód
mądrzejszy	mądry

✅ Przykłady lematyzacji (język angielski):

Forma w tekście	Lemmatyzacja (forma słownikowa)
running	run
better	good
children	child

Zadanie



Treść

W Google Colab https://colab.research.google.com/ dla poniższego tekstu wykonaj:

1. Zapamiętaj tekst w zmeinnej
   Rozwój sztucznej inteligencji i automatyzacji zmienia współczesny rynek pracy. Coraz więcej firm wdraża inteligentne algorytmy do analizowania danych, automatyzowania procesów i wspierania decyzji biznesowych. Niektórzy eksperci uważają, że AI zastąpi wiele zawodów, szczególnie tych związanych z powtarzalnymi zadaniami, jak księgowość czy obsługa klienta. Z drugiej strony, pojawiają się nowe możliwości zatrudnienia w sektorach wymagających umiejętności związanych z programowaniem, analizą danych i etyką AI. Czy sztuczna inteligencja stanie się zagrożeniem, czy może szansą na rozwój nowych kompetencji?

2. Zrób dla niego Tokenizację
3. Zrób dla tokenów Lematyzację po polsku

https://colab.research.google.com/drive/169CLNjy_yQhgCQE4YiYuWKlYLi8HnRwO?usp=sharing

Embedding

Embedding

to metoda reprezentacji słów, zdań lub dokumentów w postaci **wektorów liczbowych**.
Pozwala ona modelom NLP **rozumieć znaczenie słów** poprzez ich kontekst i relacje semantyczne.

🔷 Dlaczego embeddingi są potrzebne?

✓ **Słowa w języku naturalnym są niejednoznaczne** – modele AI muszą zrozumieć ich kontekst.
✓ **Reprezentacja liczbową ułatwia analizę języka** – modele przekształcają słowa na wektory, dzięki czemu mogą wykrywać zależności między nimi.
✓ **Poprawia skuteczność modeli NLP** – ułatwia przetwarzanie tekstu w zadaniach takich jak tłumaczenie czy analiza sentymentu.

🔷 Jak działają embeddingi?

1 Każde słowo zostaje zamienione na **wektor liczbowy** w wielowymiarowej przestrzeni.

2 Słowa o podobnym znaczeniu są umieszczane bliżej siebie.

3 Modele AI uczą się **relacji między słowami** (np. *„król” – „mężczyzna” + „kobieta” = „królowa”*).

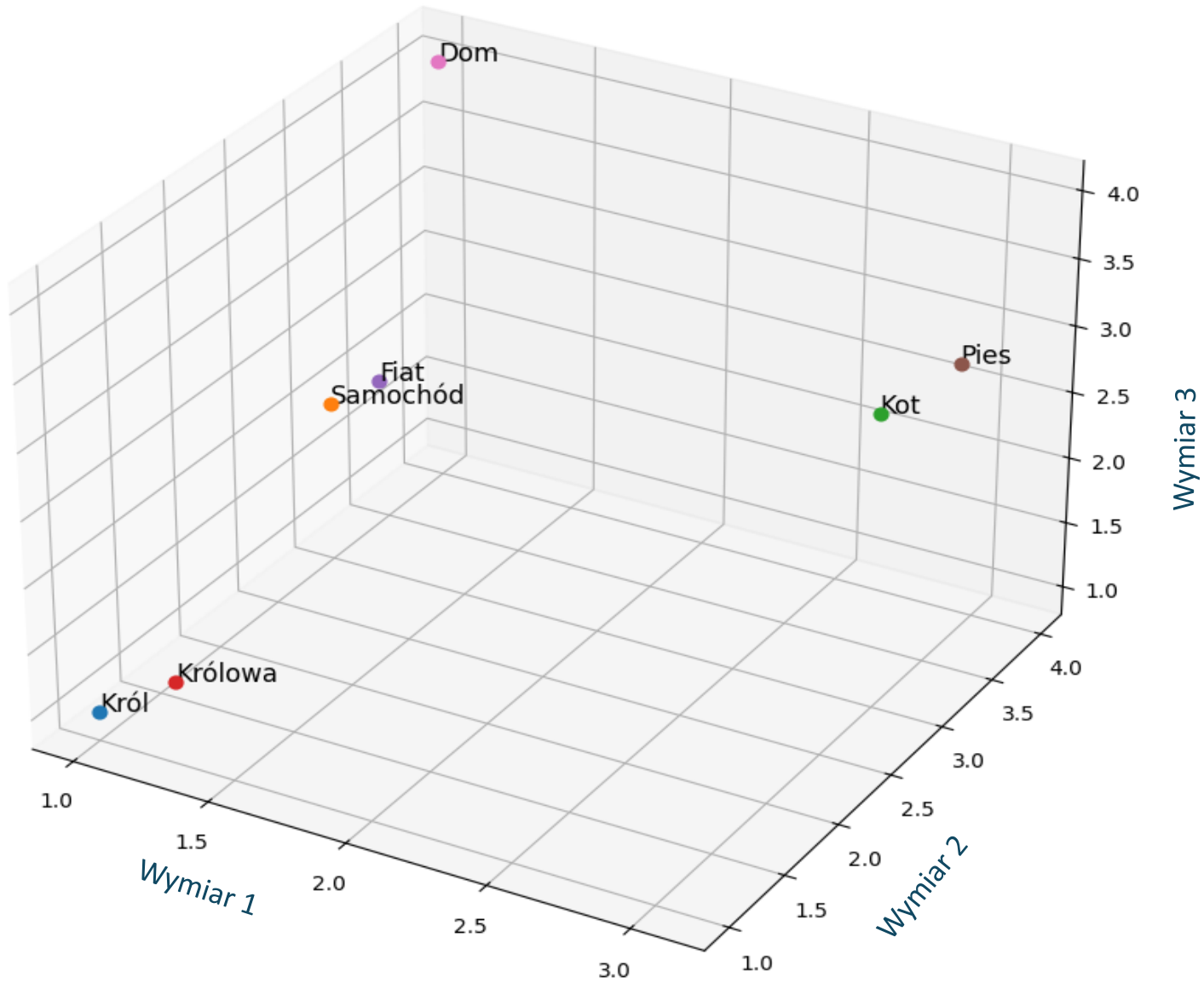
Embedding

Wektory

```
'Król'      ->([1.0, 1.1, 1.0])
'Samochód'->([1.0, 3.1, 2.1])
'Kot'       ->([3.1, 3.2, 3.2])
'Królowa'  ->([1.2, 1.2, 1.2])
'Fiat'      ->([1.1, 3.2, 2.1])
'Pies'      ->([3.1, 3.5, 3.1])
'Dom'       ->([1.1, 4.1, 4.1])
```

Macierz odległości (podobieństwa)

	Król	Samochód	Kot	Królowa	Fiat	Pies	Dom
Król	0,0000	2,2825	3,6959	0,3000	2,3728	3,8184	4,3151
Samochód	2,2825	0,0000	2,3728	2,1119	0,1414	2,3601	2,2383
Kot	3,6959	2,3728	0,0000	3,4073	2,2825	0,3162	2,3707
Królowa	0,3000	2,1119	3,4073	0,0000	2,1955	3,5369	4,1024
Fiat	2,3728	0,1414	2,2825	2,1955	0,0000	2,2561	2,1932
Pies	3,8184	2,3601	0,3162	3,5369	2,2561	0,0000	2,3152
Dom	4,3151	2,2383	2,3707	4,1024	2,1932	2,3152	0,0000

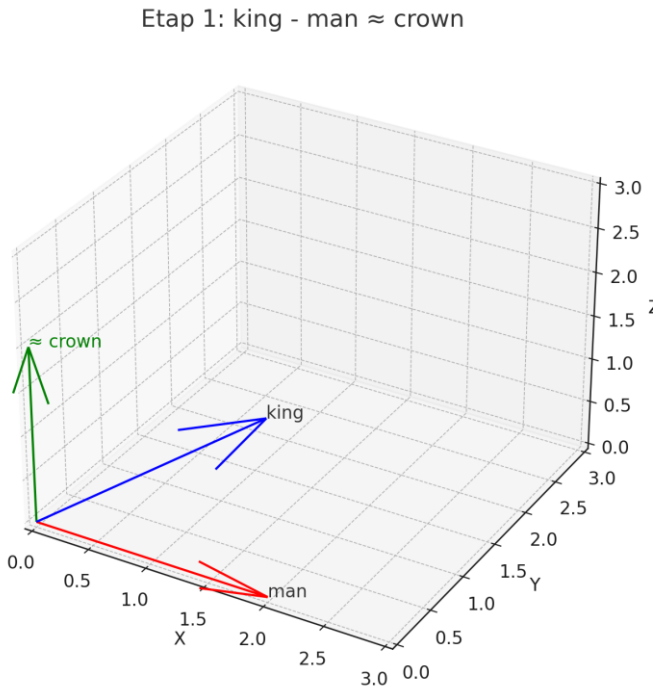


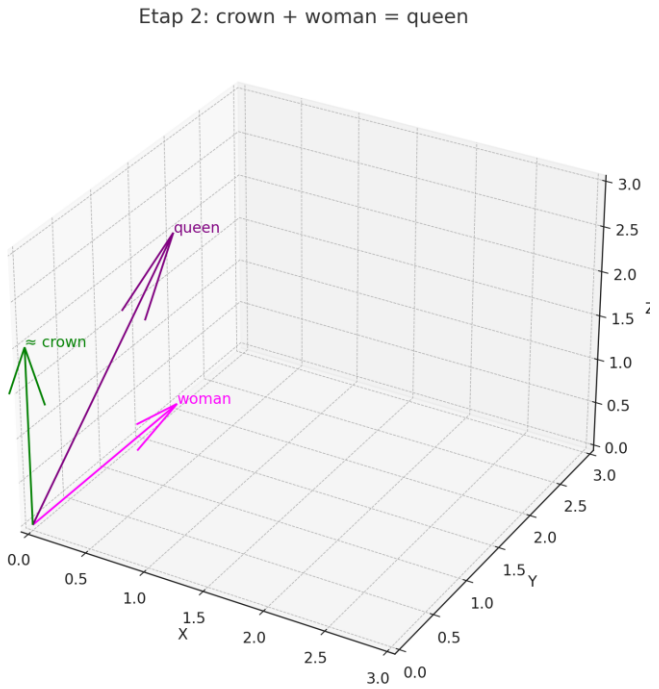
Algebra słów

Tokyo - Japan + Germany = Berlin
brother - man + woman = sister
Microsoft - Windows + Google = Android

january – winter + summer = ??
husband – man + woman = ??
actor – man + woman = ??

king - man + woman = queen
[2,0,2] - [2,0,0] + [0,2,0] = [0,2,2]
[0,0,2]
crown





Tensor

Tensor

to **wielowymiarowa struktura danych** używana w **algebrze liniowej** oraz **sztucznej inteligencji**. Można go traktować jako **rozszerzenie skalara, wektora i macierzy** na wyższe wymiary.

🔷 **Rodzaje tensorów:**

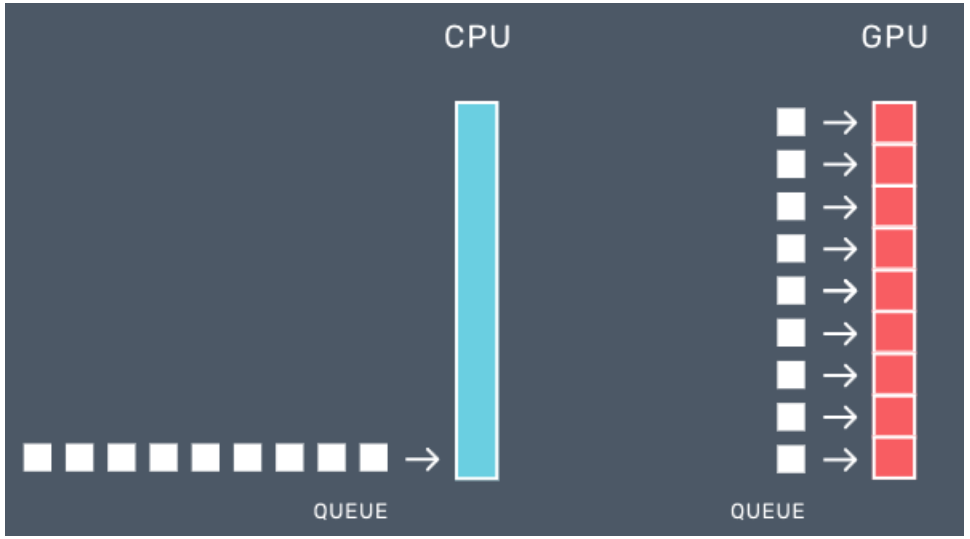
✓ **Tensor 0-wymiarowy** (*skalar*) – pojedyncza liczba (np. $x = 5$).
✓ **Tensor 1-wymiarowy** (*wektor*) – lista liczb (np. [1, 2, 3]).
✓ **Tensor 2-wymiarowy** (*macierz*) – tablica liczb (np. macierz 3×3).
✓ **Tensor n-wymiarowy** – struktura o więcej niż dwóch wymiarach, np. obraz RGB to tensor 3D (wysokość × szerokość × kanały kolorów).

🔷 **Zastosowanie tensorów w AI:**

✓ W **uczeniu maszynowym** i **sieciach neuronowych** tensory reprezentują **dane wejściowe, wagi modeli i wyniki obliczeń**.
✓ Są podstawową jednostką obliczeniową w bibliotekach takich jak **TensorFlow, PyTorch i JAX**.
✓ Umożliwiają szybkie przetwarzanie danych na **GPU i TPU**, które są zoptymalizowane pod kątem **operacji tensorowych**.

CPU vs GPU

GPU (Graphics Processing Unit)



🔷 Równoległe przetwarzanie danych
✓ **GPU składa się z tysięcy rdzeni**, które mogą wykonywać wiele obliczeń jednocześnie, podczas gdy CPU ma ograniczoną liczbę rdzeni zoptymalizowanych do operacji sekwencyjnych.
✓ Dzięki temu **GPU jest idealne do trenowania i inferencji modeli AI**, gdzie potrzeba ogromnej mocy obliczeniowej do operacji na macierzach.

🔷 Optymalizacja dla obliczeń tensorowych
✓ Nowoczesne GPU, takie jak **NVIDIA Tensor Cores**, są zaprojektowane specjalnie do przyspieszania **obliczeń tensorowych**, które są kluczowe w **sieciach neuronowych i głębokim uczeniu (Deep Learning)**.

🔷 Efektywność w operacjach na dużych zbiorach danych
✓ AI wymaga **przetwarzania ogromnych zbiorów danych**, a GPU może jednocześnie wykonywać operacje na wielu wartościach, co znacząco przyspiesza procesy związane z uczeniem maszynowym.

CPU vs GPU

Porównanie procesorów GPU

TFLOPS (Tera Floating Point Operations Per Second) – jednostka miary wydajności obliczeniowej procesorów (CPU, GPU, TPU) oznaczająca liczbę **bilionów (10^{12}) operacji zmiennoprzecinkowych na sekundę**.

Model GPU	Producent	VRAM	Wydajność (FP32 TFLOPS)	Cena (USD)	TDP (W)
NVIDIA A100	NVIDIA	80 GB HBM2	~19,5 TFLOPS	~$10 000–12 000	300 W (PCIe) / 400 W (SXM)
NVIDIA H100	NVIDIA	80 GB HBM3	~67 TFLOPS	~$30 000–40 000	350 W (PCIe) / 700 W (SXM)
AMD Instinct MI300X	AMD	192 GB HBM3	~81,7 TFLOPS	~$15 000	750 W
GeForce RTX 4090	NVIDIA	24 GB GDDR6X	~82,6 TFLOPS	~$1 599	450 W
GeForce RTX 4080	NVIDIA	16 GB GDDR6X	~48,7 TFLOPS	~$1 199	320 W
Radeon RX 7900 XTX	AMD	24 GB GDDR6	~61,4 TFLOPS	~$999	355 W

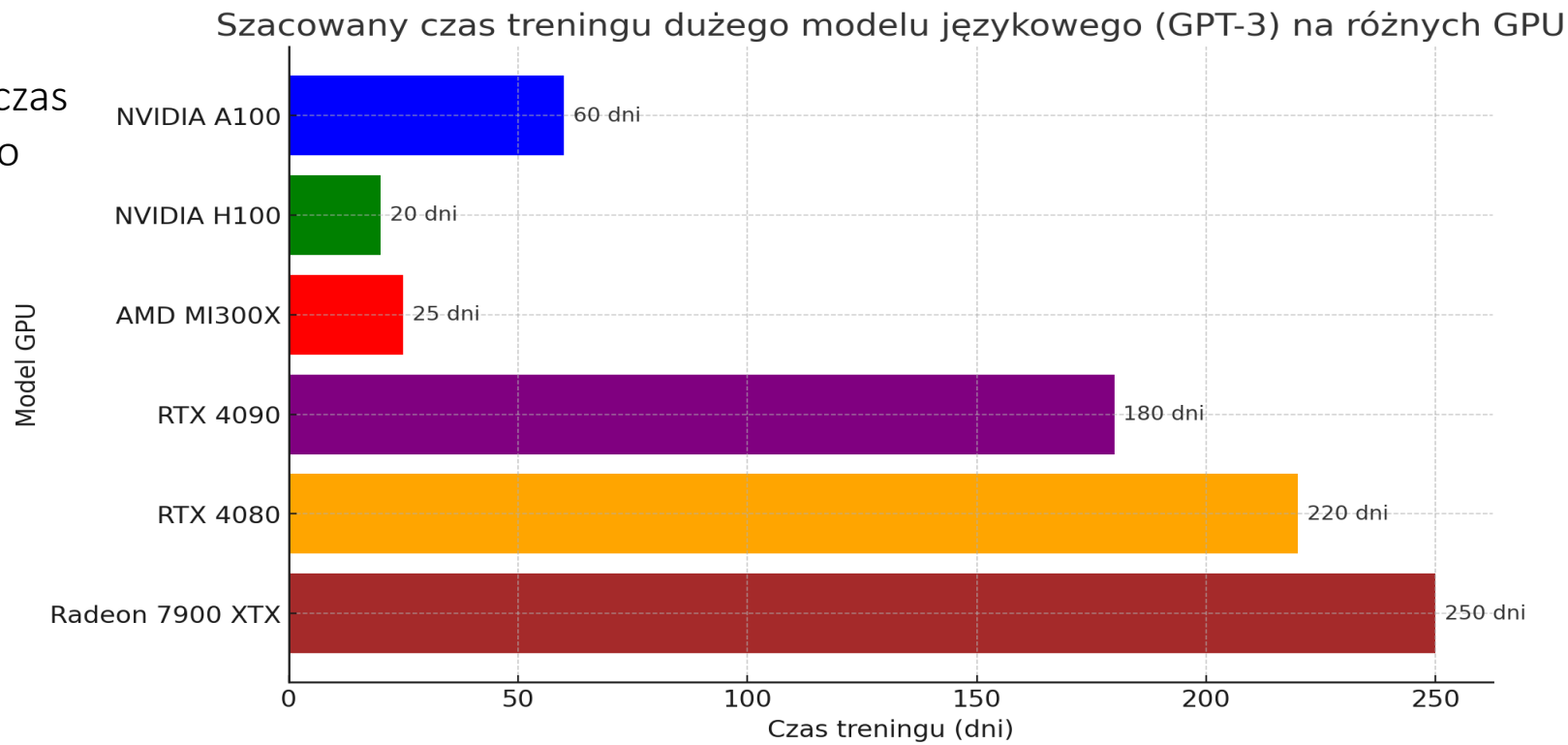
HBM (High Bandwidth Memory) – to rodzaj **wysokoprzepustowej pamięci DRAM**, zaprojektowanej do współpracy z procesorami graficznymi (GPU), akceleratorami AI oraz superkomputerami.

GDDR (Graphics Double Data Rate) – to rodzina **wysokowydajnych pamięci DRAM** zaprojektowanych specjalnie do zastosowań graficznych, takich jak **karty graficzne (GPU), konsole do gier, systemy HPC i AI**.

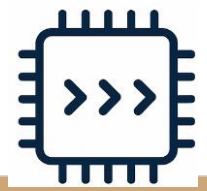
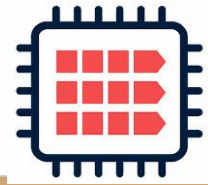
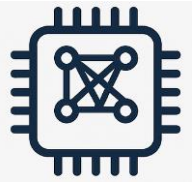
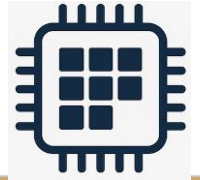
CPU vs GPU

Porównanie procesów GPU

wykres przedstawiający szacowany czas treningu dużego modelu językowego (GPT-3, ~175B parametrów)



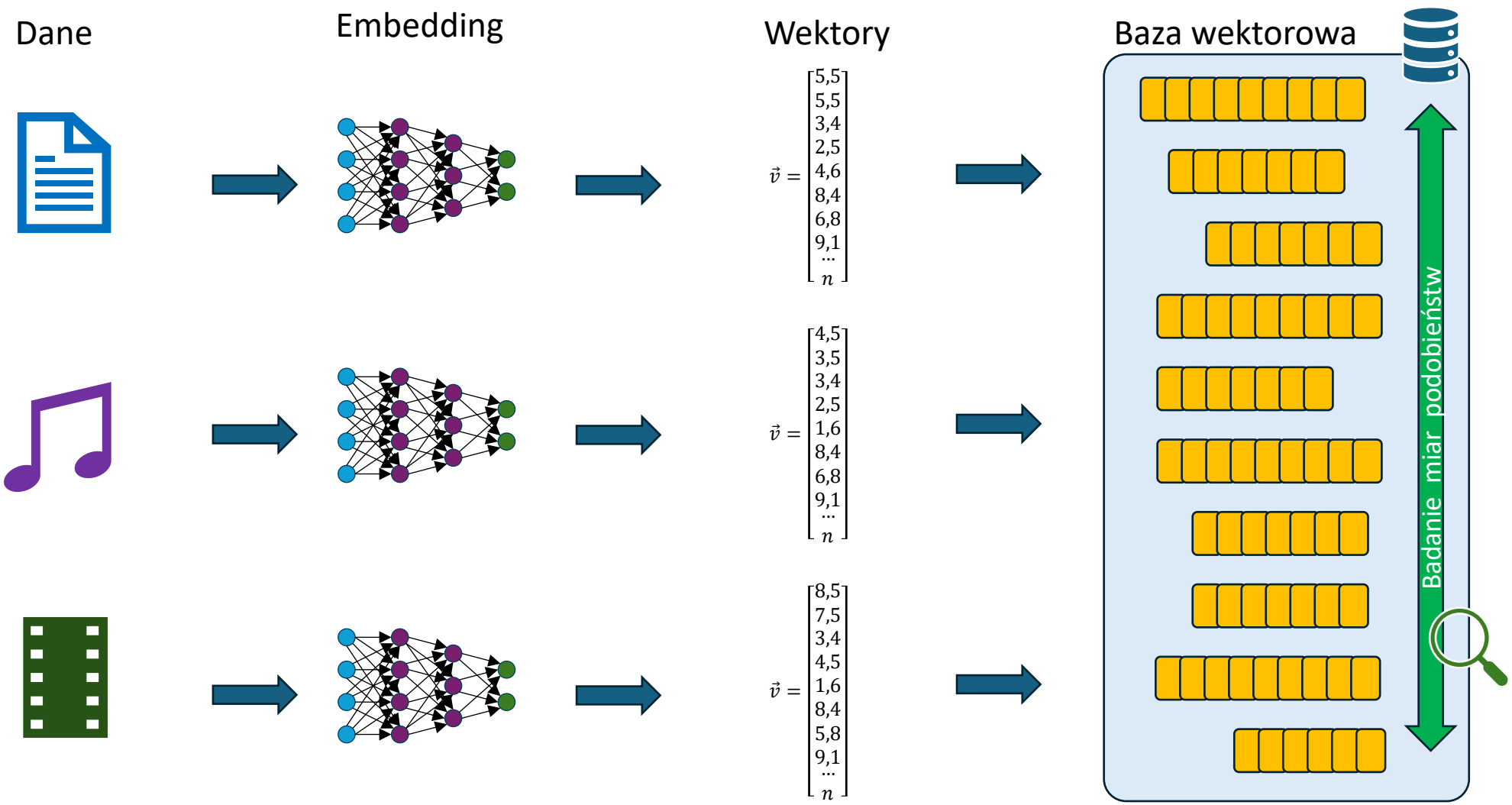
Procesory

Cecha / Aspekt	CPU (procesor ogólnego przeznaczenia)	GPU (procesor graficzny)	NPU (procesor neuronowy)	TPU (procesor tensorowy)
Główne zastosowanie	Obliczenia ogólne, logika systemowa	Przetwarzanie równoległe grafiki i trenowanie AI	Wydajne wnioskowanie AI w systemach lokalnych	Trenowanie i wdrażanie dużych modeli AI w środowisku chmurowym
Architektura	Kilka silnych rdzeni	Tysiące wyspecjalizowanych rdzeni	Układy zoptymalizowane pod zadania AI	Zoptymalizowany do operacji tensorowych (wektorowych)
Równoległość obliczeń	Niska do umiarkowanej	Bardzo wysoka	Średnia do wysokiej	Ekstremalnie wysoka
Najlepiej sprawdza się przy	Sterowaniu systemem, prostych zadaniach AI	Wizji komputerowej, grach, trenowaniu modeli	Wnioskowaniu AI w czasie rzeczywistym	Skalowalnym przetwarzaniu dużych modeli AI
Opóźnienia (latencja)	Umiarkowane	Umiarkowane do wysokich	Niskie (dla wnioskowania lokalnego)	Niskie (optymalizacja wsadowa)
Efektywność energetyczna	Średnia	Niska (wysokie zużycie energii)	Wysoka (zoptymalizowana dla obliczeń lokalnych)	Wysoka (optymalizacja chmurowa)
Typowe zastosowania	Komputery osobiste, serwery, systemy ogólne	Stacje robocze, centra danych, renderowanie AI	Systemy wbudowane, edge computing, IoT	Infrastruktura chmurowa do zadań AI na dużą skalę
Przykłady użycia	Aplikacje biurowe, OS, lekkie modele ML	Gry, analiza wideo, trenowanie modeli głębokich	Rozpoznawanie wzorców, analiza sensoryczna	Modele językowe, systemy rekomendacyjne
Elastyczność programowania	Bardzo wysoka	Wysoka	Średnia (specjalizowane interfejsy i SDK)	Ograniczona (głównie TensorFlow)
Producenci	Intel, AMD	NVIDIA, AMD	Apple, Huawei, Qualcomm	Google

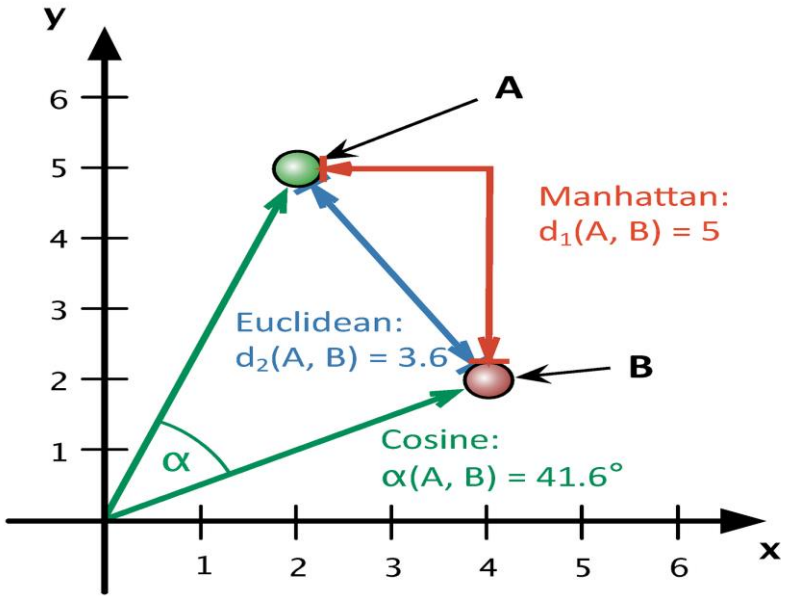
Bazy wektorowe



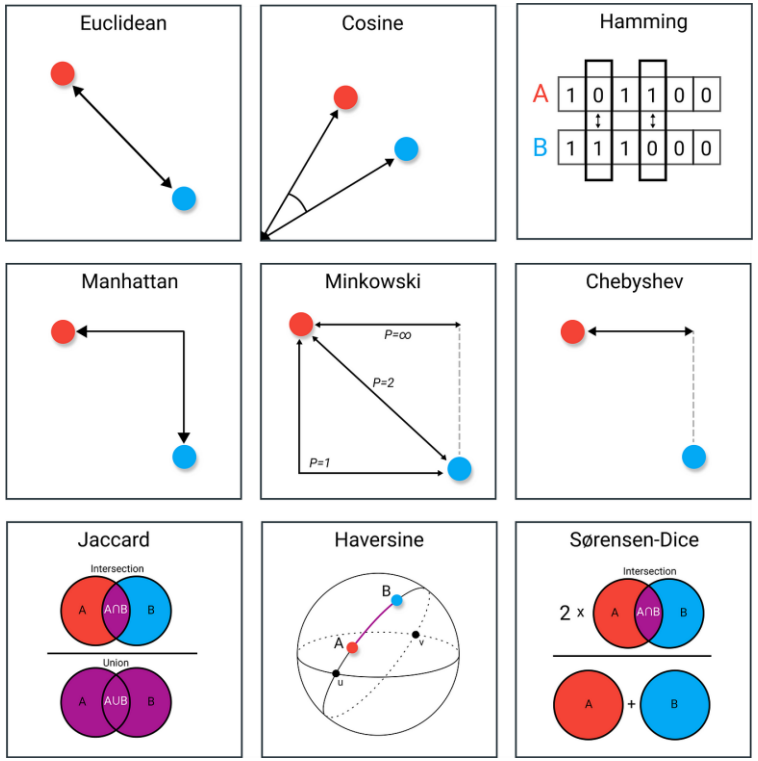


Miary podobieństw

Przez **miarę podobieństwa** (ang. *similarity measure*) rozumie się formalną funkcję lub procedurę służącą do ilościowej oceny stopnia podobieństwa pomiędzy dwiema jednostkami — mogą to być obiekty, zbiory danych, **wektory cech**, wzorce czy sygnały.
Innymi słowy, miara podobieństwa przypisuje parze obiektów **wartość liczbową**, która odzwierciedla, na **ile oba obiekty są do siebie zbliżone pod względem zdefiniowanych cech lub kryteriów**.



Miara	Opis	Zastosowanie w AI
Odległość euklidesowa	Mierzy rzeczywistą odległość między dwoma punktami w przestrzeni n-wymiarowej.	Algorytmy klastrowania (np. K-Means), analiza danych wielowymiarowych.
Odległość kosinusowa	Mierzy kąt między wektorami, nie uwzględniając ich długości.	NLP (porównywanie dokumentów, Word2Vec), analiza wektorów cech w AI.
Odległość Manhattan	Suma różnic wartości w każdej osi przestrzeni.	KNN, sieci neuronowe przy analizie obrazów i sygnałów.



Zadanie



Treść

W Google Colab https://colab.research.google.com/ zbadaj odległości pomiędzy słowami.

1. Dokonaj Embedding wyrazów: „król”, „królowa”, „dom”, „mieszkanie”, „samochód” i przedstaw wektory odpowiadające tym wyrazom.
2. Dokonaj pomiaru odległości i przedstaw w postaci tabeli odległości pomiędzy każdym z tych wyrazów.

https://colab.research.google.com/drive/1oaoFvuqRXkYCC7F-Got95Vtl6orOx7C-?usp=sharing

Architektura Transformers

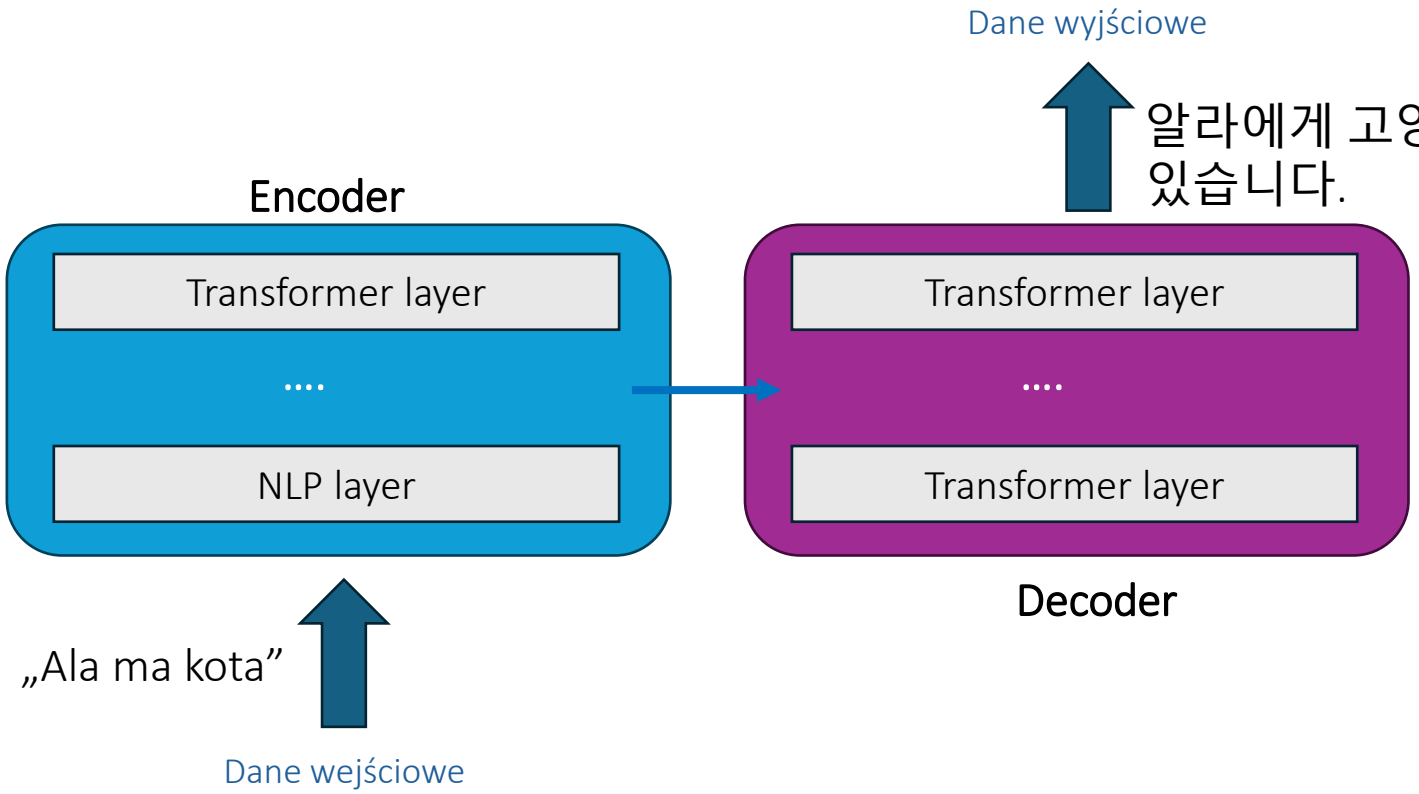
Architektura Transformers

to nowoczesna **architektura sieci neuronowych** zaprojektowana do przetwarzania **sekwencji danych**, takich jak tekst, dźwięk czy obrazy. Wykorzystuje **mechanizm uwagi (Attention)**, który pozwala modelowi analizować całe zdanie jednocześnie, zamiast przetwarzać je krok po kroku jak w starszych sieciach RNN (Recurrent Neural Networks).

Wprowadzona w 2017 roku przez Vaswani i in. w artykule "Attention is All You Need", architektura ta zrewolucjonizowała przetwarzanie języka naturalnego (NLP) dzięki zastosowaniu mechanizmu uwagi (attention) zamiast tradycyjnych rekurencyjnych sieci neuronowych (RNN).

Architektura Transformers

Struktura warstwowa Transformers



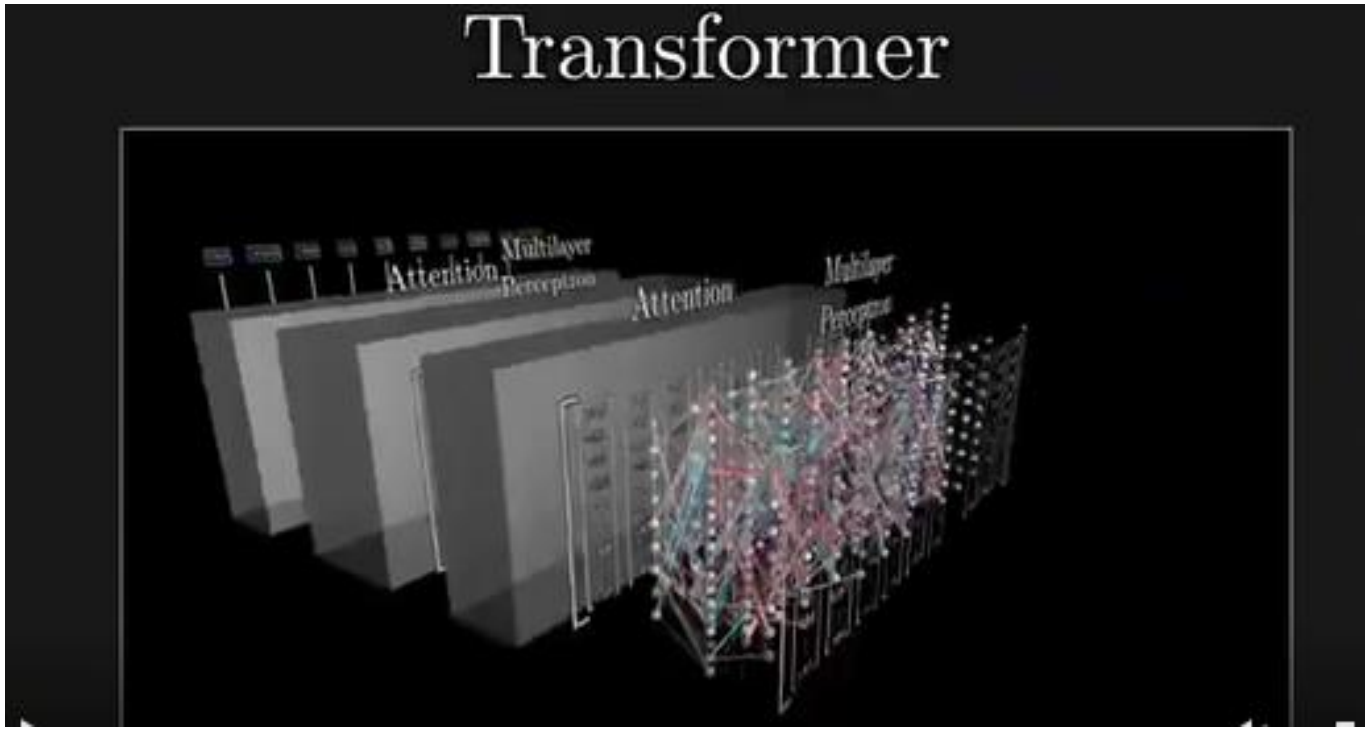
- **Enkoder**: Przekształca sekwencję wejściową w wewnętrzne reprezentacje zawierające kontekstowe informacje o wszystkich elementach sekwencji.
- **Dekoder**: Generuje sekwencję wyjściową na podstawie reprezentacji wewnętrznych z enkodera oraz wcześniej wygenerowanej części wyjściowej sekwencji, używając mechanizmu uwagi do uwzględnienia kontekstu.

Architektura Transformers

Główne Cechy

 Mechanizm Uwagi (Attention Mechanism)	Kluczowy element Transformerów, pozwalający modelowi skupić się na różnych częściach sekwencji wejściowej podczas generowania kolejnych elementów sekwencji wyjściowej. Umożliwia równoczesne przetwarzanie wszystkich elementów sekwencji, co znacznie przyspiesza trening i wnioskowanie (inference).
 Struktura Warstwowa	Składa się z enkodera (encoder) i dekodera (decoder). Enkoder przekształca sekwencję wejściową w wewnętrzną reprezentację, a dekoder generuje sekwencję wyjściową na podstawie tej reprezentacji.
 Self-Attention (Mechanizm Samoobserwacji)	W ramach każdej warstwy enkodera i dekodera, self-attention oblicza ważność każdego słowa w odniesieniu do wszystkich innych słów w sekwencji. Dzięki temu model może zrozumieć zależności między słowami niezależnie od ich odległości w tekście.
 Pozycjonalne Kodowanie (Positional Encoding)	Transformery nie mają wbudowanej sekwencyjności, jak sieci RNN, dlatego dodaje się pozycjonalne kodowanie. Informuje ono model o kolejności słów w zdaniu, co pozwala mu lepiej analizować kontekst.
 Multi-Head Attention (Wielogłowy Mechanizm Uwagi)	Zamiast pojedynczej warstwy uwagi, transformery używają wielu „głów” uwagi (multi-head attention). Pozwala to modelowi uchwycić różne rodzaje zależności między słowami w odmiennych podprzestrzeniach.

Architektura Transformers



https://www.youtube.com/watch?v=wjZofJX0v4M

Duże Modele Językowe (LLMs)

Duże Modele Językowe (Large Language Models, LLMs)

to **zaawansowane systemy AI** trenowane na miliardach danych tekstowych, które potrafią **rozumieć, analizować i generować język naturalny**.

Wykorzystują **głębokie sieci neuronowe**, w szczególności **architekturę Transformer**, do modelowania skomplikowanych zależności w języku.

Rozmiar Modelu:

- miliardy parametrów, co pozwala im na efektywne modelowanie złożonych wzorców językowych.

Architektura:

- transformers, która umożliwia równoczesne przetwarzanie sekwencji danych i uchwycenie długodystansowych zależności między słowami.

Zdolności Generacyjne:

- potrafią generować spójne i kontekstowo odpowiednie teksty, odpowiadać na pytani.

Trening:

- trenowane na ogromnych korpusach tekstowych, obejmujących różnorodne źródła…

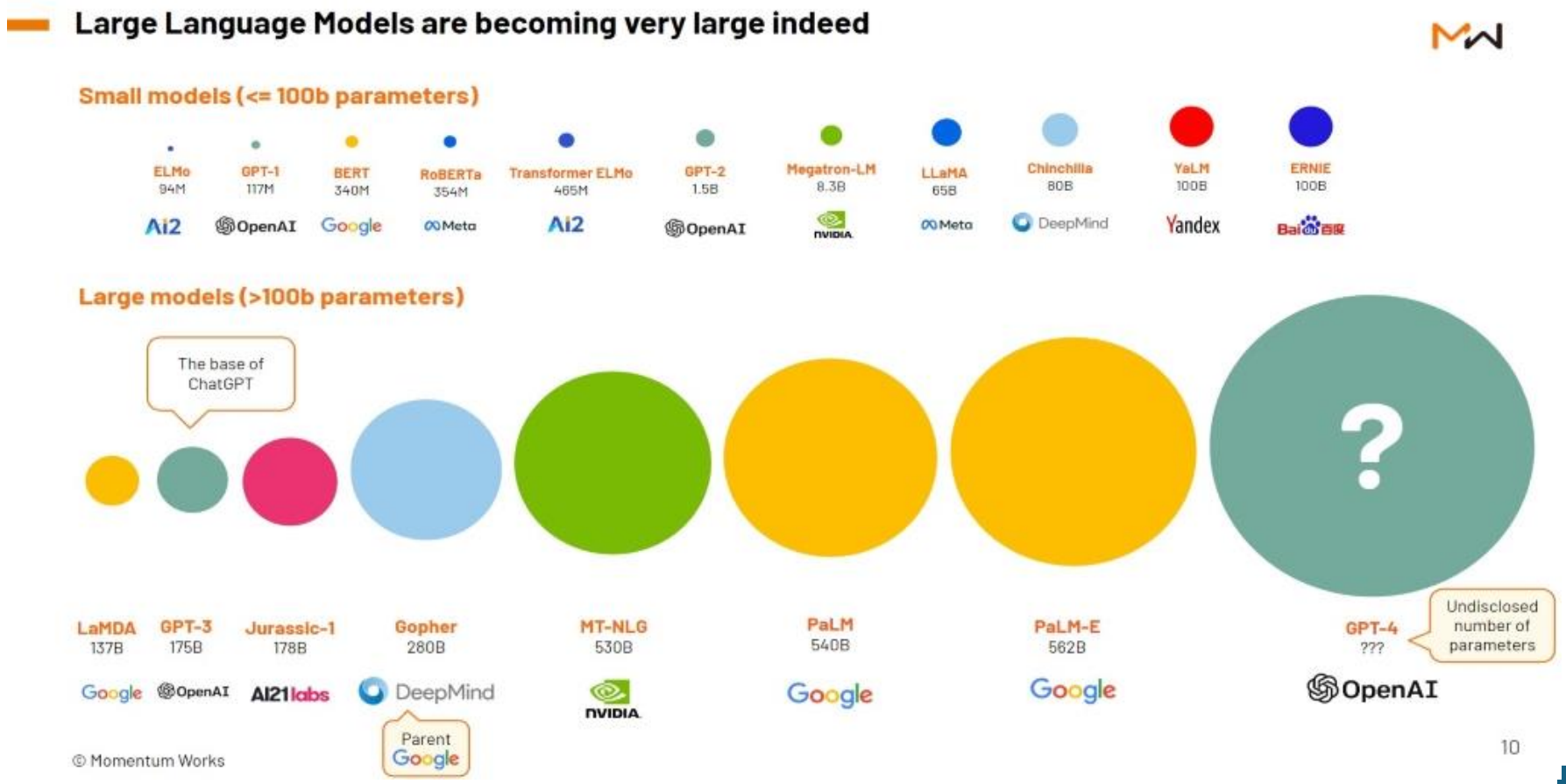
Duże Modele Językowe (LLMs)

Liczba parametrów

Liczba parametrów (mld)	Jakość odpowiedzi	Zalety	Wady	Przykładowe modele
<1	Niska	Szybkość działania, małe wymagania sprzętowe	Ograniczona wiedza, słaba generalizacja, częste błędy	GPT-2 (0.12B), BERT Small (0.11B)
1 - 10	Średnia	Lepsza jakość odpowiedzi, większa kontekstowość	Nadal ograniczona wiedza, możliwe błędy w trudniejszych zadaniach	GPT-2 Medium (1.5B), DistilBERT (1.3B), Llama 1 (7B)
10 - 50	Dobra	Dobre rozumienie kontekstu, mniej błędów, lepsza generalizacja	Większe wymagania sprzętowe, dłuższy czas inferencji	GPT-3 Small (13B), Llama 2 (13B), Mistral 7B (efektywnie działa jak 30B)
50 - 200	Bardzo dobra	Płynniejsze odpowiedzi, rzadziej halucynacje, wysoka kontekstowość	Wysokie koszty obliczeniowe, trudności w fine-tuningu	GPT-3 (175B), Llama 2 (70B), Claude 2 (100B)
200+	Ekspertowa	Najlepsza jakość, zdolność do tworzenia kompleksowych odpowiedzi	Bardzo wysokie wymagania sprzętowe, ogromne koszty trenowania i użytkowania	GPT-4 (>200B), Gemini Ultra (około 500B, szacowane), Claude 3 (>200B)

Duże Modele Językowe (LLMs)

Liczba parametrów

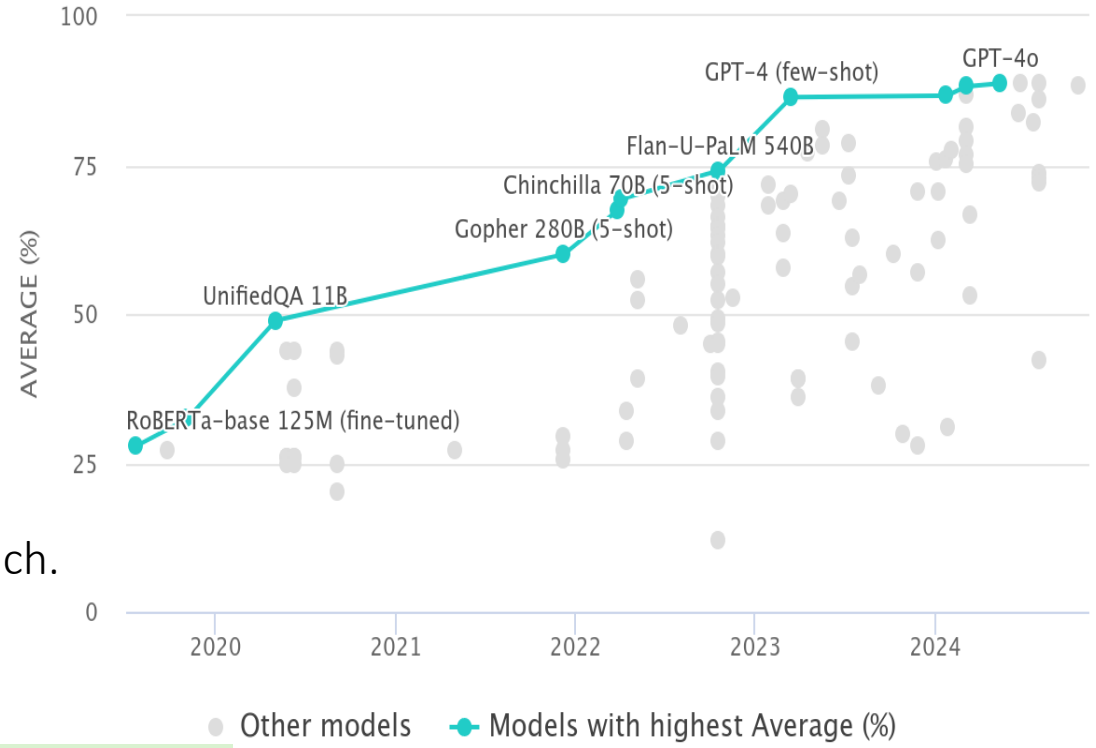


Duże Modele Językowe (LLMs)

MMLU (Massive Multitask Language Understanding)

to benchmark oceniający zdolność modeli sztucznej inteligencji do rozumienia języka i wiedzy ogólnej.

Testuje AI na szerokim zakresie tematów, od matematyki po nauki społeczne.



Dlaczego MMLU jest ważne?

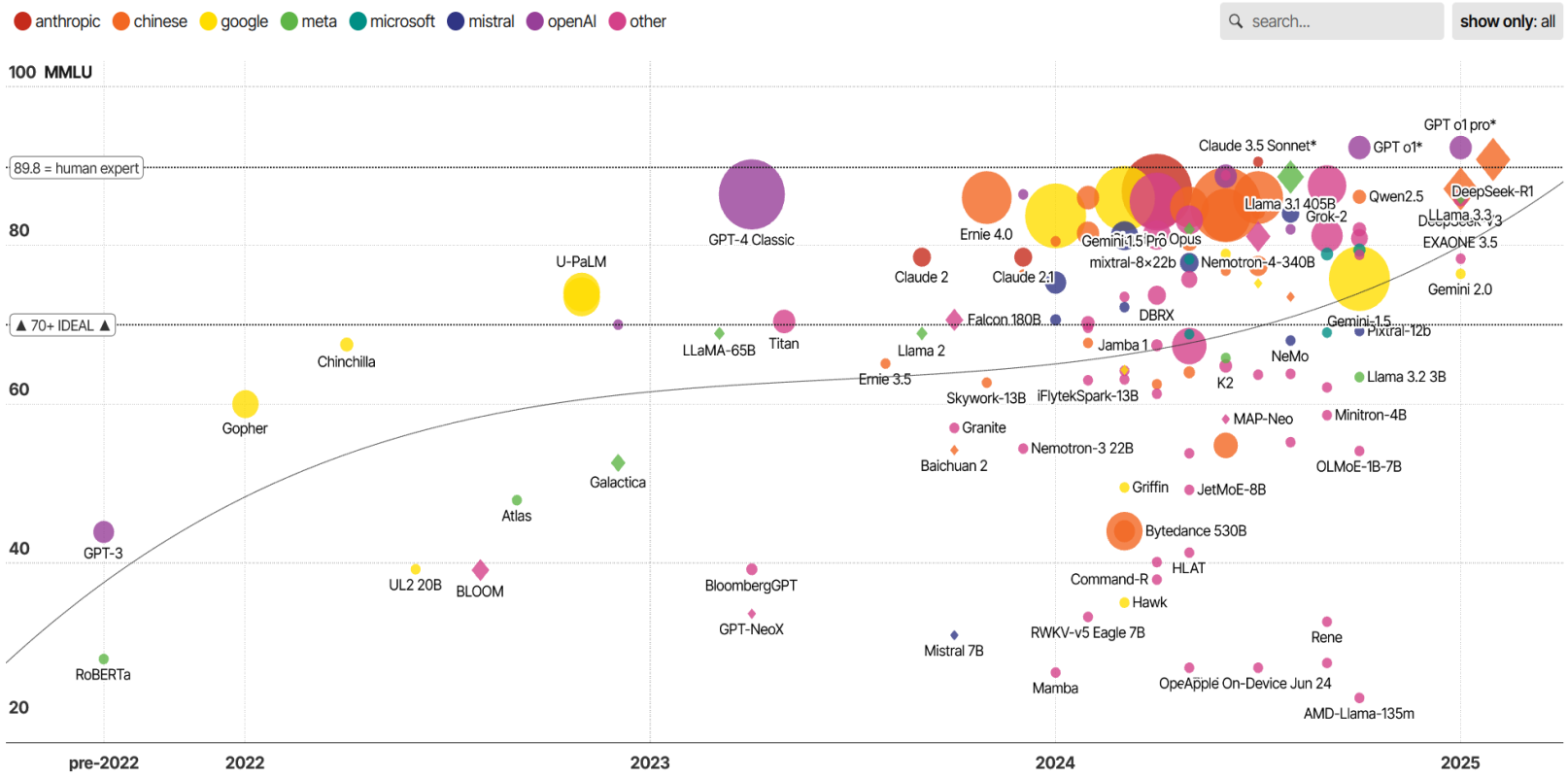
- Sprawdza **zdolność modelu do generalizacji** w różnych dziedzinach.
- Porównuje **wydajność AI z wynikami ludzkimi**.
- Wskazuje mocne i słabe strony modeli językowych.

Model **GPT-4** osiągnął **>80%** skuteczności, przewyższając wcześniejsze AI. Dla porównania, człowiek z wykształceniem akademickim osiąga ~90%.

https://paperswithcode.com/sota/multi-task-language-understanding-on-mmlu

Duże Modele Językowe (LLMs)

Mapa rozwoju generatywnej AI



The Rise of Generative AI & Large Language Models (LLMs)
https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/

Duże Modele Językowe (LLMs)

Perpleksja (Perplexity)

to miara statystyczna używana do oceny jakości modeli językowych. Oznacza **niepewność modelu** co do przewidywania kolejnego słowa w sekwencji.

📌 **Interpretacja**

Niższa perpleksja = model **lepiej przewiduje** dane wejściowe

Perpleksja = 1: model jest **pewny** – przewiduje jedno słowo z prawdopodobieństwem 100%

Perpleksja = |V| (wielkość słownika): model zgaduje losowo

🎯 **Przykład**

Model A: Perpleksja = 35

Model B: Perpleksja = 80

👉 Model A lepiej przewiduje tekst niż model B

Generatywna sztuczna inteligencja

Generatywna sztuczna inteligencja (Generative AI)

to technologia, która umożliwia tworzenie nowych treści, takich jak obrazy, teksty, dźwięki i wideo, na podstawie istniejących danych.

Generatywne Sieci Antagonistyczne (GAN Generative Adversarial Networks):

- Składają się z dwóch sieci neuronowych: generatora i dyskryminatora, które rywalizują ze sobą.

Wariacyjne Autoenkodery (VAE Variational Autoencoders)

- Wykorzystują probabilistyczne podejście do generowania nowych danych na podstawie kodowania i dekodowania.

Wizja komputerowa (CV)

Wizja komputerowa (Computer Vision, CV)

to dziedzina sztucznej inteligencji, która umożliwia **przetwarzanie, analizę i interpretację obrazów oraz wideo** w sposób zbliżony do ludzkiego wzroku.

Jej celem jest **rozpoznawanie obiektów, wykrywanie wzorców i interpretacja informacji wizualnych** przez komputery.

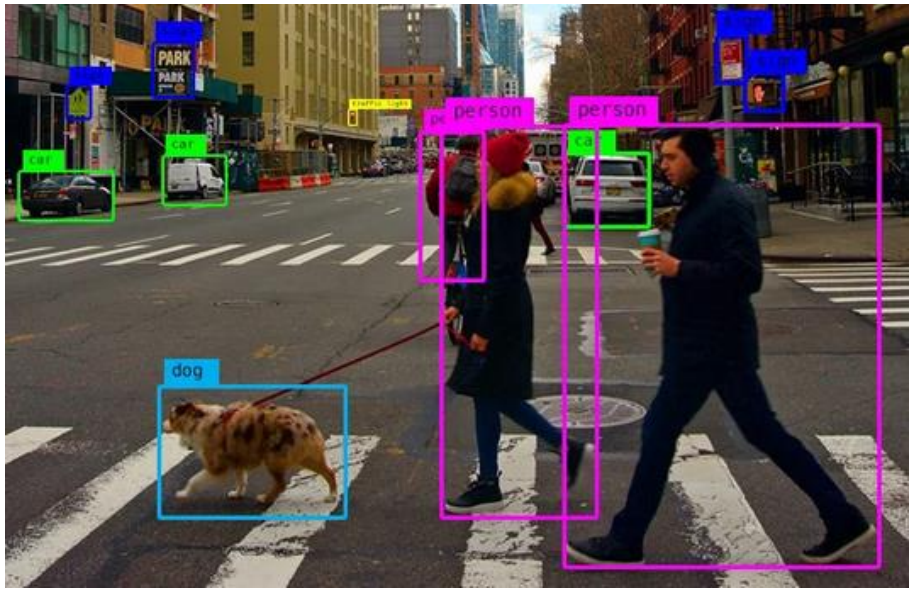
Przykłady zastosowań wizji komputerowej

✅ **Autonomiczne pojazdy** – rozpoznawanie znaków drogowych, pieszych i przeszkód.

✅ **Bezpieczeństwo i monitoring** – systemy nadzoru wizyjnego, wykrywanie twarzy.

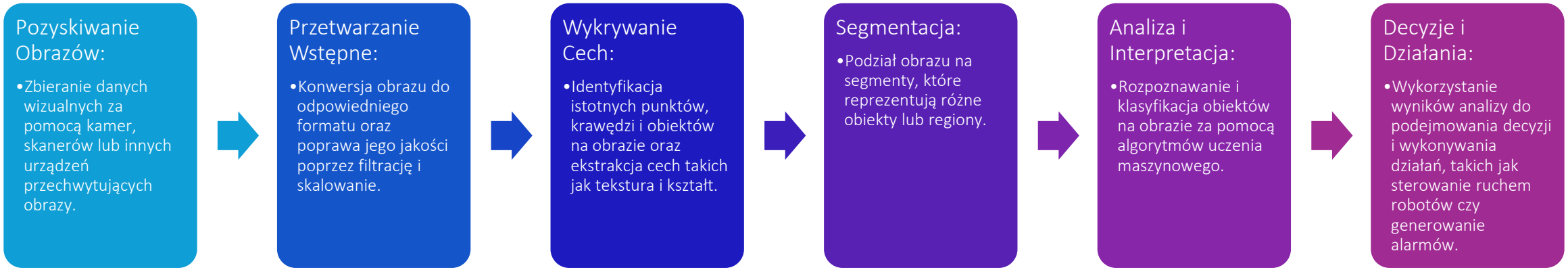
✅ **Diagnostyka medyczna** – analiza zdjęć rentgenowskich, wykrywanie nowotworów.

✅ **Przemysł i automatyzacja** – kontrola jakości produktów na liniach produkcyjnych.



Wizja komputerowa (CV)

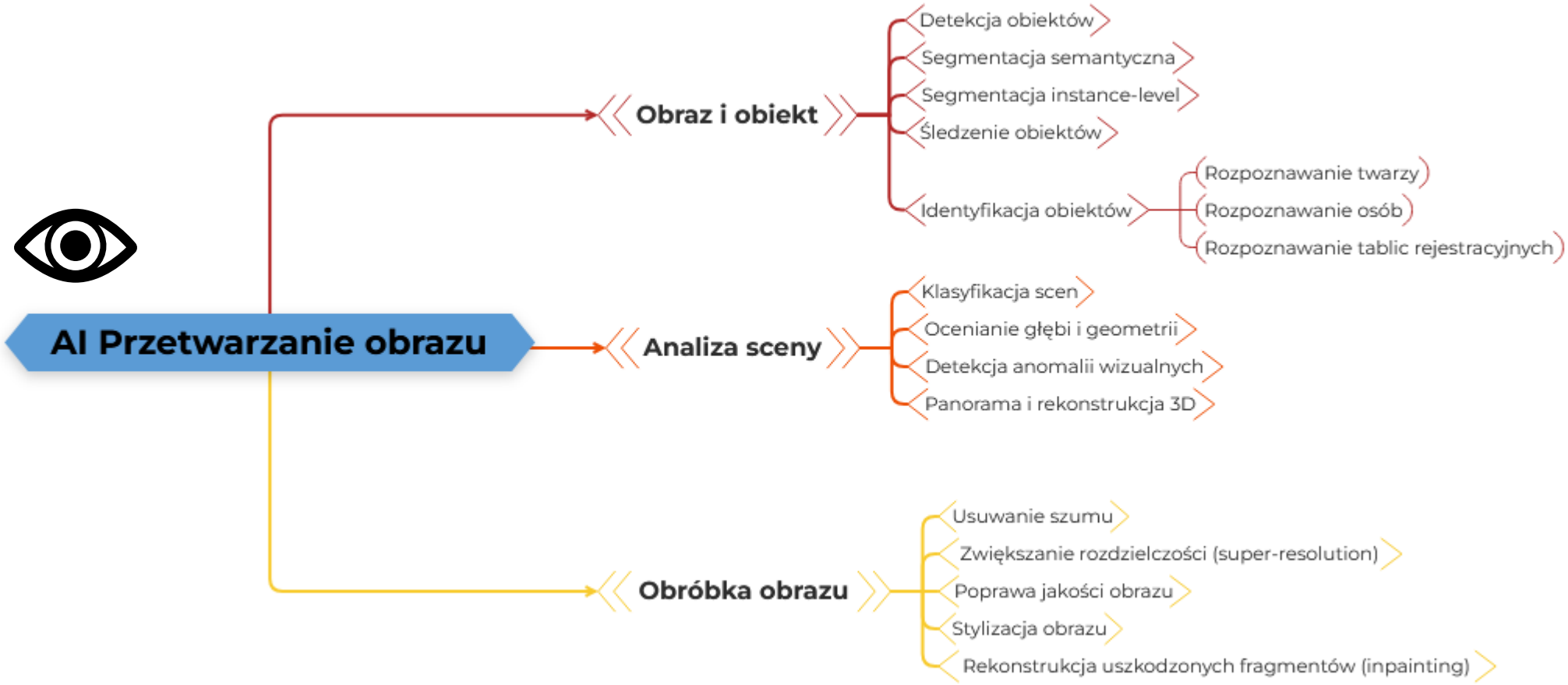
Proces przetwarzania i analizy obrazów w ramach CV



Wizja komputerowa (CV)

Kluczowe techniki	
Rozpoznawanie Obrazów:	Identyfikacja i klasyfikacja obiektów w obrazach na podstawie cech wizualnych.
	Przykłady: Systemy detekcji twarzy, systemy rozpoznawania obiektów.
Segmentacja Obrazów:	Podział obrazu na segmenty lub regiony w celu identyfikacji i analizy poszczególnych obiektów.
	Przykłady: Analiza medyczna, segmentacja komórek na obrazach mikroskopowych.
Wykrywanie Krawędzi:	Identyfikacja krawędzi obiektów w obrazach w celu ich wyodrębnienia.
	Przykłady: Systemy rozpoznawania pisma ręcznego, analiza zdjęć satelitarnych.
Rekonstrukcja 3D:	Tworzenie trójwymiarowych modeli obiektów na podstawie obrazów dwuwymiarowych.
	Przykłady: Modelowanie 3D w medycynie, skanowanie obiektów do gier komputerowych.
Śledzenie Obiektów:	Monitorowanie ruchu obiektów w sekwencjach wideo.
	Przykłady: Systemy monitoringu wizyjnego, analiza ruchu w sporcie

Przetwarzanie obrazu



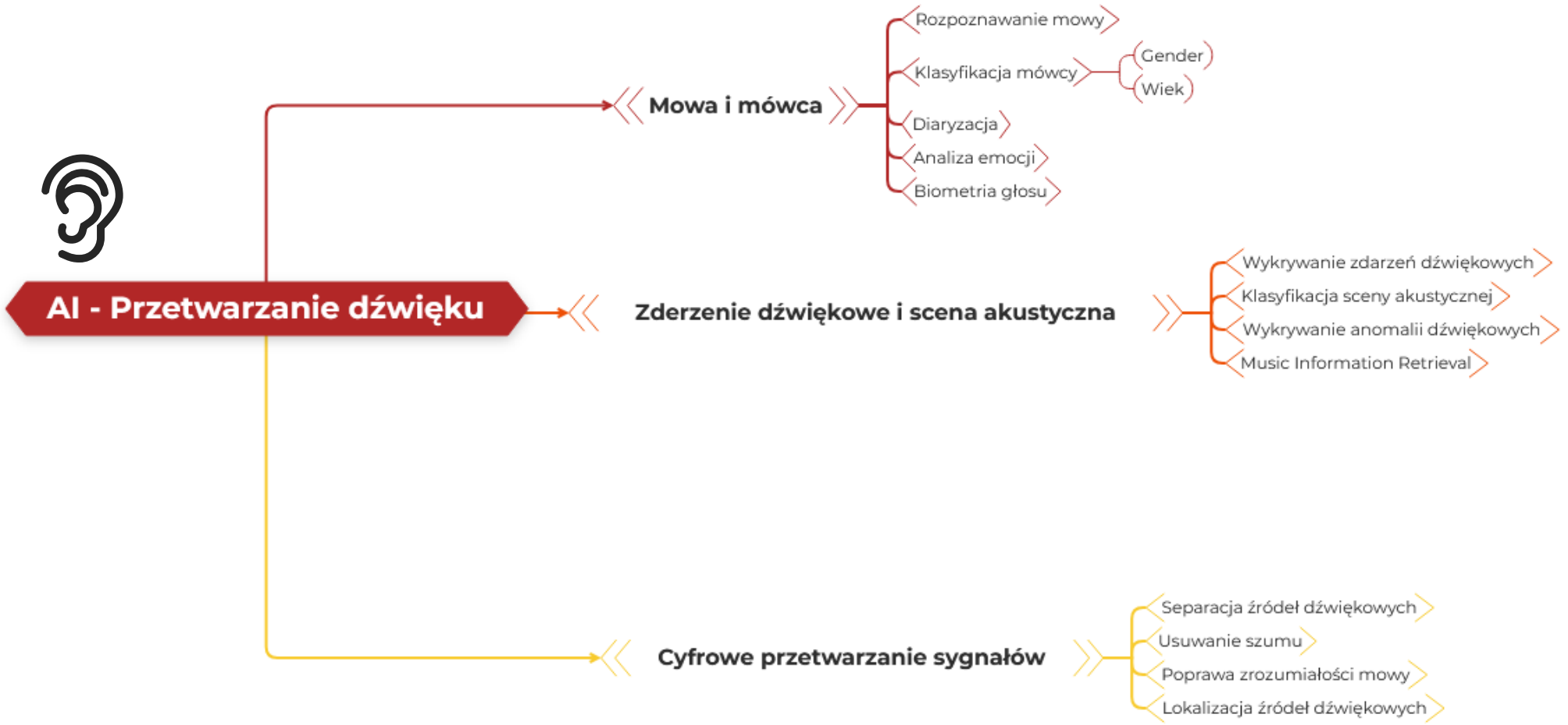
Przetwarzanie obrazu

Popularne modele



Model	Zastosowanie główne	Opis skrócony
Convolutional Neural Networks (CNN)	Klasyfikacja obrazów, detekcja obiektów	Podstawa większości klasycznych systemów widzenia komputerowego.
ResNet (Residual Network)	Klasyfikacja obrazów	Głębokie sieci neuronowe z "skip connections" – odporne na zanikający gradient.
YOLO (You Only Look Once)	Detekcja i lokalizacja obiektów w czasie rzeczywistym	Bardzo szybka i dokładna detekcja wielu obiektów na jednym obrazie.
Faster R-CNN	Detekcja obiektów, segmentacja	Precyzyjny, ale wolniejszy niż YOLO. Klasyk w analizie obrazów.
Mask R-CNN	Segmentacja instance-level	Rozszerzenie Faster R-CNN o maski pikselowe (segmentacja obiektów).
U-Net	Segmentacja medyczna i precyzyjna	Bardzo popularny w medycynie, biologii, mikroskopii.
Vision Transformer (ViT)	Klasyfikacja, detekcja, segmentacja	Transformer przeniesiony z NLP do wizji; lepszy przy dużych zbiorach danych.
CLIP (Contrastive Language–Image Pretraining)	Zrozumienie obrazów i tekstów razem	Łączy obraz i język – rozpoznaje, co znajduje się na obrazie, na podstawie opisu.
DINO / DINOv2	Uczenie reprezentacji bez nadzoru	Służy jako uniwersalny ekstraktor cech wizualnych (self-supervised learning).

Przetwarzanie dźwięku



Przetwarzanie dźwięku

Popularne modele



Model / Architektura	Zastosowania główne	Opis skrócony
Wav2Vec 2.0 (Facebook AI)	Rozpoznawanie mowy (ASR), diarizacja	Model samonadzorowany, uczy się reprezentacji bez etykiet. Bardzo skuteczny.
Whisper (OpenAI)	Rozpoznawanie mowy, transkrypcja, tłumaczenie	Uniwersalny, wielojęzyczny model ASR. Otwarty, szeroko stosowany.
HuBERT (Facebook AI)	Rozpoznawanie mowy, klasyfikacja mówcy	Samonadzorowany, podobny do Wav2Vec 2.0, bardzo skuteczny.
DeepSpeech (Mozilla)	Rozpoznawanie mowy	Klasyczny model oparty na RNN. Lekki i łatwy do wdrożenia.
YAMNet (Google)	Klasyfikacja zdarzeń dźwiękowych	Lekki model wykorzystujący MobileNet do klasyfikacji dźwięków.
OpenUnmix	Separacja źródeł dźwiękowych (np. wokal/instrument)	Skuteczny model do separacji muzyki na ścieżki.
Demucs (Facebook AI)	Separacja dźwięków, redukcja szumu	Najnowszy model do miksowania/oddzielania źródeł audio.
Tao-Net / SudoRM-RF	Odzyskiwanie jakości mowy, odszumianie	Nowoczesne modele typu speech enhancement (na SNR i jakość mowy).
AudioCLIP / CLAP	Połączenie dźwięku i tekstu/obrazu	Modele multimodalne: opisują dźwięki językiem naturalnym.

Zadanie



Treść

W Google Colab https://colab.research.google.com/ zbadaj zdjęcie.

1. Rozpoznaj obiekty w pliku ludzie.jpg z wykorzystaniem wizji komputerowej.
2. Wypisz w tabeli rozpoznane obiekty.
3. Wygeneruj oddzielne obrazki z rozpoznanymi obiektami.

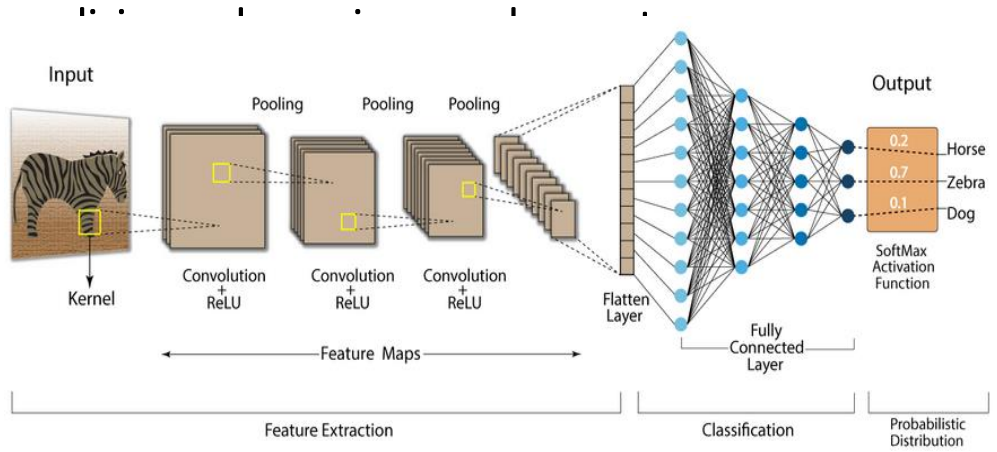
https://colab.research.google.com/drive/1H6BNvQ7nFHqmCgcpAck24U0qawWS9nj0?usp=sharing

Sieci Konwolucyjne (CNN)

Sieci Konwolucyjne (Convolutional Neural Networks, CNN)

to rodzaj sztucznych sieci neuronowych, **specjalizujących się w przetwarzaniu i analizie danych przestrzennych, takich jak obrazy i sygnały wizualne**.

CNN są szeroko stosowane w **rozpoznawaniu obrazów, przetwarzaniu wideo widzeniu (Computer Vision).**



🔷 Jak działają sieci konwolucyjne?

✅ **Warstwa konwolucyjna (Convolution Layer)**
✓ Wykorzystuje **filtry (kernels)** do wykrywania krawędzi, kształtów i wzorców w obrazie.

✅ **Warstwa aktywacji (np. ReLU)**
✓ Wprowadza **nieliniowość**, co pozwala modelowi lepiej wykrywać skomplikowane zależności.

✅ **Warstwa poolingowa (np. Max Pooling)**
✓ **Zmniejsza wymiarowość danych**, co zwiększa efektywność obliczeń.

✅ **Warstwa w pełni połączona (Fully Connected Layer, FC)**
✓ Ostateczne przekształcenie danych i klasyfikacja na podstawie wykrytych cech.

Przetwarzanie kognitywne (CC)

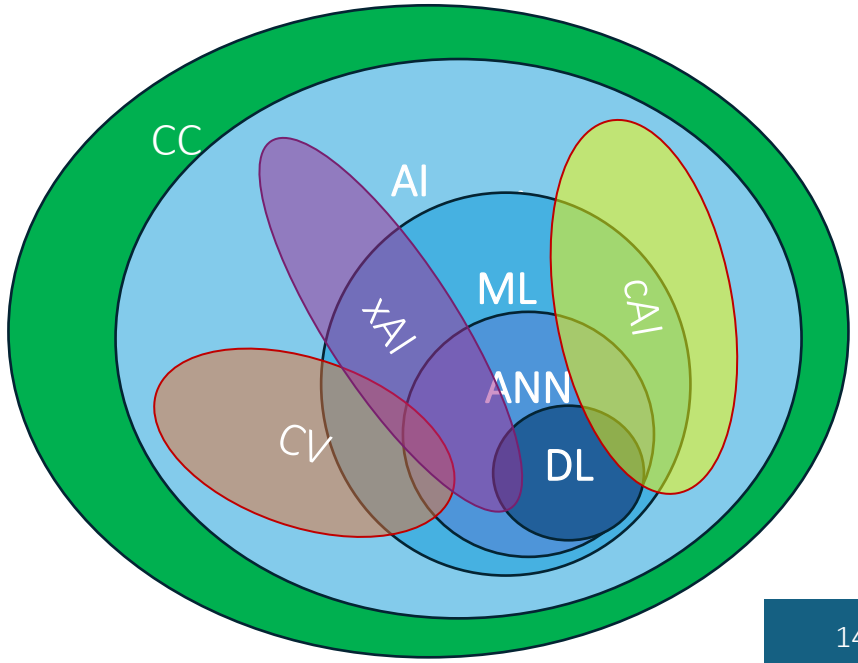
Przetwarzanie kognitywne (Cognitive Computing, CC)

to dziedzina informatyki, która naśladuje **procesy poznawcze ludzkiego mózgu**.

Wykorzystuje **dane, wzorce i język naturalny**, aby umożliwić komputerom **samodzielne rozwiązywanie złożonych problemów** bez ścisłego programowania reguł.

Zastosowania

✅ **Diagnostyka medyczna** – analiza wyników badań, wykrywanie chorób na podstawie obrazów.
✅ **Chatboty i wirtualni asystenci** – inteligentne systemy obsługi klienta, np. Watson AI.
✅ **Finanse i analiza ryzyka** – przewidywanie trendów rynkowych, wykrywanie oszustw.
✅ **Cyberbezpieczeństwo** – wykrywanie zagrożeń w sieciach IT, analiza wzorców ataków.
✅ **Systemy rekomendacyjne** – personalizacja treści w e-commerce, np. Netflix, Amazon.



Przetwarzanie kognitywne (CC)

Cztery atrybuty przetwarzania kognitywnego

Adaptacyjny (Adaptive):

Systemy muszą dostosowywać się do zmieniających się informacji i okoliczności w czasie rzeczywistym.

Interaktywny (Interactive):

Systemy muszą płynnie komunikować się z ludźmi i urządzeniami, umożliwiając użytkownikom jasne wyrażanie żądań.

Iteracyjny i stanowy (Iterative and Stateful):

Systemy przechowują wcześniejsze informacje i mogą zadawać dodatkowe pytania, aby uzyskać nowe dane do zrozumienia problemów.

Kontekstowy (Contextual):

Systemy muszą rozumieć kontekst problemu, korzystając z danych strukturalnych i niestrukturalnych, aby wykonywać złożone zadania.

Współpracująca Sztuczna Inteligencja (cAI)

Współpracująca Sztuczna Inteligencja (Collaborative AI, cAI)

to podejście w sztucznej inteligencji, które koncentruje się na **interakcji i współpracy AI z ludźmi oraz innymi systemami AI**.

Celem jest tworzenie inteligentnych systemów, które mogą **współdziałać, uczyć się i podejmować decyzje w sposób skoordynowany**.

🔷 **Kluczowe cechy cAI**
✓ **Interakcja z ludźmi** – AI jako wsparcie w podejmowaniu decyzji, automatyzacja rutynowych zadań.
✓ **Adaptacyjność** – uczenie się na podstawie współpracy i dostosowywanie do zmieniających się warunków.
✓ **Bezpieczeństwo i etyka** – AI wspierające ludzi, zamiast ich zastępować, zwiększając zaufanie i przejrzystość.



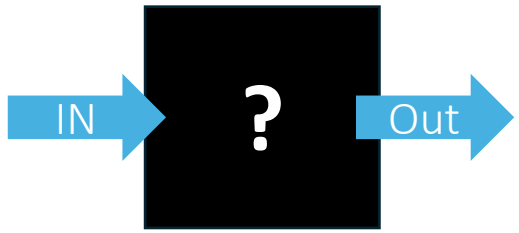
Systemy cAI są projektowane do współpracy, a nie zastępowania ludzi.

Wytłumaczalna Sztuczna Inteligencja (XAI)

Wytłumaczalna Sztuczna Inteligencja (Explainable AI, XAI)

to podejście w sztucznej inteligencji, które ma na celu **zwiększenie przejrzystości i interpretowalności działania modeli AI**.

Dzięki XAI użytkownicy mogą **zrozumieć, dlaczego model podjął określoną decyzję**, co zwiększa zaufanie do systemów AI.



🔷 Kluczowe cechy XAI

✓ **Przejrzystość** – AI wyjaśnia swoje decyzje, umożliwiając użytkownikom ich lepsze zrozumienie.
✓ **Sprawdzalność** – pozwala analizować i oceniać poprawność działania modelu.
✓ **Redukcja uprzedzeń** – ułatwia wykrywanie stronniczości i błędów w algorytmach.
✓ **Zwiększone zaufanie** – użytkownicy mają większą kontrolę nad AI i mogą podejmować lepsze decyzje.

Przeciwieństwo do XAI jest **Black-Box AI**

XAI = większa przejrzystość, kontrola i zaufanie do AI.

Wytłumaczalna Sztuczna Inteligencja (XAI)

Kluczowe powody problemów z wyjaśnialnością AI

✓ Nowoczesne **modele AI** składają się z **milionów, a nawet miliardów parametrów**, co sprawia, że ich działanie jest skomplikowane i trudne do przeanalizowania.

✓ **Modele takie jak GPT-4, Gemini czy sieci konwolucyjne (CNN)** działają w sposób trudny do śledzenia – nie wiadomo dokładnie, jakie cechy danych wpływają na decyzje.

✓ **Użytkownik widzi tylko wynik, ale nie zna procesu decyzyjnego.**

✓ **Modele AI uczą się z danych, które mogą zawierać ukryte uprzedzenia**, prowadząc do błędnych lub niesprawiedliwych decyzji.

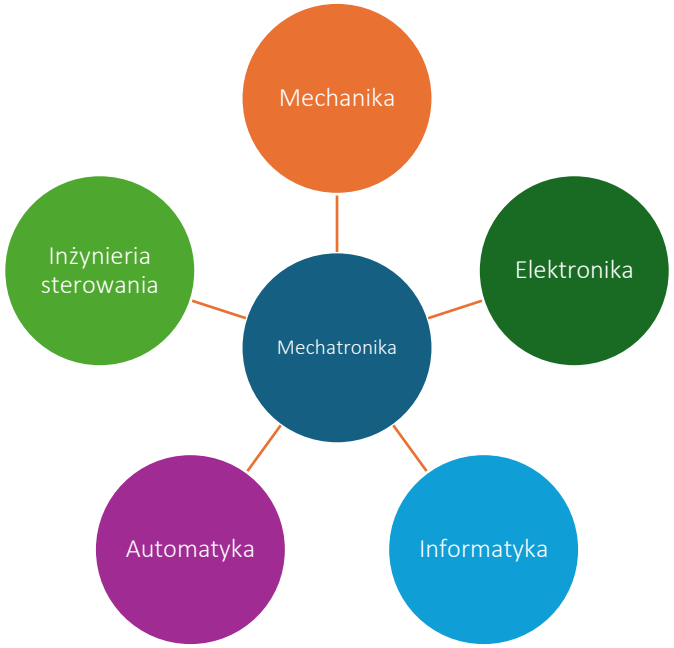
✓ Modele AI są trenowane, aby osiągać **najlepsze wyniki w przewidywaniach**, a nie, aby były zrozumiałe dla ludzi.

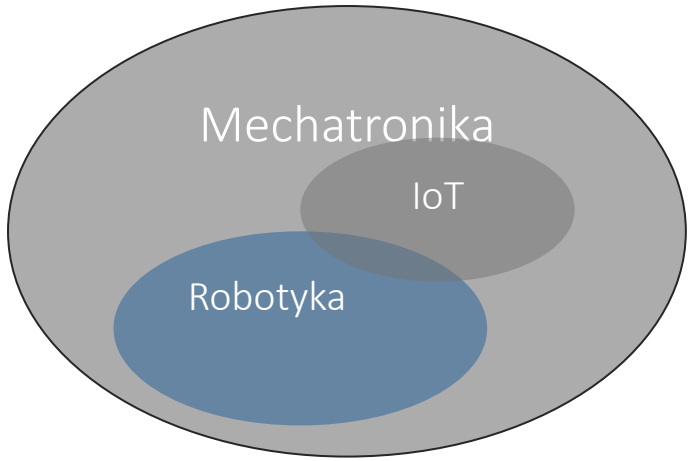
✓ W wielu branżach (np. medycyna, finanse) wymagana jest **przejrzystość decyzji**, ale brakuje uniwersalnych metod tłumaczenia wyników AI.

Mechatronika

Mechatronika

to dziedzina nauki i inżynierii zajmująca się projektowaniem i rozwijaniem **inteligentnych systemów** oraz **urządzeń**, które łączą funkcje sterowania, monitorowania i automatyzacji. Opiera się na integracji różnych technologii, umożliwiając tworzenie nowoczesnych rozwiązań stosowanych w przemyśle, medycynie, transporcie i codziennym życiu.





Celem mechatroniki jest projektowanie i tworzenie inteligentnych systemów oraz urządzeń.

Mechatronika

Zastosowanie

Obszar	Opis	Przykłady
Robotyka	Projektowanie i budowa robotów do różnych zastosowań.	Roboty przemysłowe, roboty medyczne, roboty serwisowe.
Przemysł motoryzacyjny	Systemy wspomagania kierowcy i elektroniczne układy sterowania silnikiem.	ABS, ESP, systemy kontroli trakcji, elektroniczne układy sterowania silnikiem (ECU).
Przemysł lotniczy i kosmiczny	Systemy sterowania lotem, autopiloty, systemy nawigacyjne i mechanizmy precyzyjnego sterowania.	Autopiloty, systemy nawigacyjne, mechanizmy precyzyjnego sterowania satelitami.
Przemysł medyczny	Urządzenia diagnostyczne, terapeutyczne, roboty chirurgiczne i systemy wspomagające rehabilitację.	MRI, ultrasonografy, roboty chirurgiczne, protezy.
Domowe urządzenia AGD	Inteligentne urządzenia domowe i systemy zarządzania energią.	Inteligentne pralki, lodówki, systemy automatyki domowej, systemy zarządzania energią.
Energetyka	Systemy sterowania i zarządzania energią w turbinach wiatrowych, elektrowniach solarnych oraz inteligentnych sieciach energetycznych.	Turbiny wiatrowe, elektrownie solarne, inteligentne sieci energetyczne (Smart Grids).

Robotyka

Robotyka

to interdyscyplinarna dziedzina nauki i inżynierii zajmująca się projektowaniem, budową, programowaniem oraz zastosowaniem **robotów** – maszyn zdolnych do wykonywania zadań w sposób autonomiczny lub sterowany.

Łączy elementy **mechaniki, elektroniki, informatyki i sztucznej inteligencji (AI)**, aby tworzyć systemy zdolne do interakcji z otoczeniem i adaptacji do zmieniających się warunków

Aspekt	Opis
Uczenie maszynowe	Roboty uczą się na podstawie danych i doświadczeń, np. poprzez deep learning
Przetwarzanie obrazu (CV)	Rozpoznawanie obiektów, wykrywanie ruchu, analiza otoczenia
Przetwarzanie języka (NLP)	Możliwość komunikacji z ludźmi (asystenci głosowi, chatboty)
Autonomia i sterowanie	Samodzielne podejmowanie decyzji i adaptacja do zmian
Interakcja człowiek-robot (HRI)	Współpraca robotów z ludźmi w przemyśle i codziennym życiu



Robotyka

RPA – Robotic Process Automation (Zrobotyzowana Automatyzacja Procesów)

to technologia automatyzująca **powtarzalne, regułowe procesy biznesowe** przy użyciu tzw. **robotów programowych** (software bots). Nie jest to AI per se, ale może być z nią zintegrowana.



🤝 **AI + RPA = Intelligent Automation (IA)**

Połączenie AI i RPA tworzy nową klasę rozwiązań:

- RPA wykonuje **mechaniczne czynności**
- AI dodaje **zdolność rozumienia, analizowania i uczenia się**

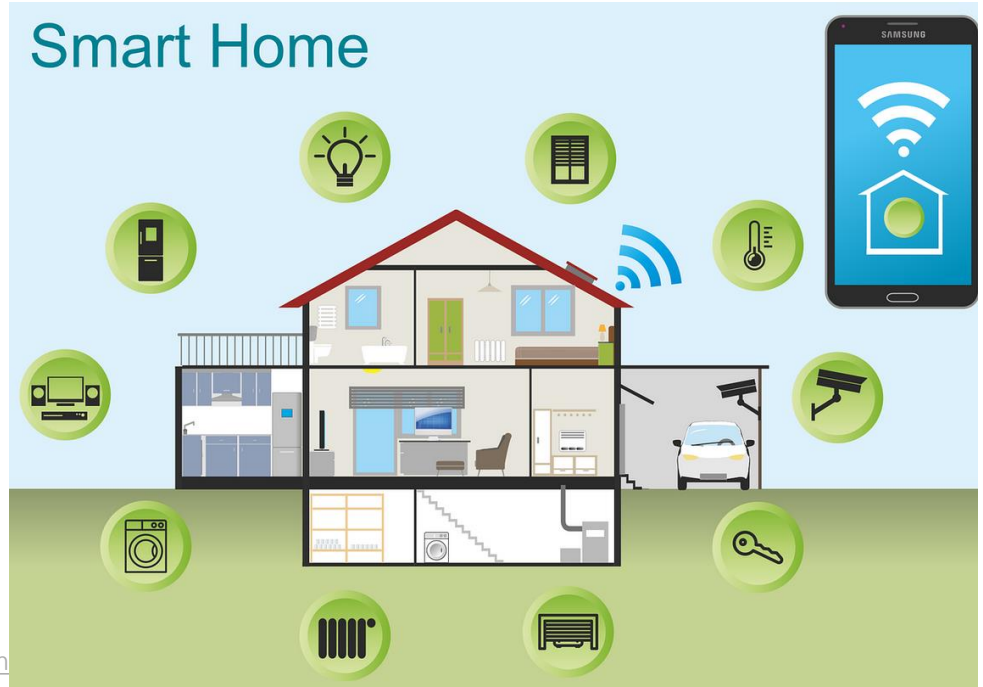
📌 Przykład:
OCR + NLP + RPA → automatyczne przetwarzanie maili i załączników PDF

Internet rzeczy (IoT)

Internet rzeczy (Internet of Things, IoT)

to sieć połączonych ze sobą urządzeń, które zbierają, przesyłają i analizują dane za pomocą Internetu.

Urządzenia te działają autonomicznie, komunikują się między sobą oraz reagują na zmiany w otoczeniu zamiast człowieka.





Przetwarzanie Brzegowe

Przetwarzanie Brzegowe (Edge Computing)

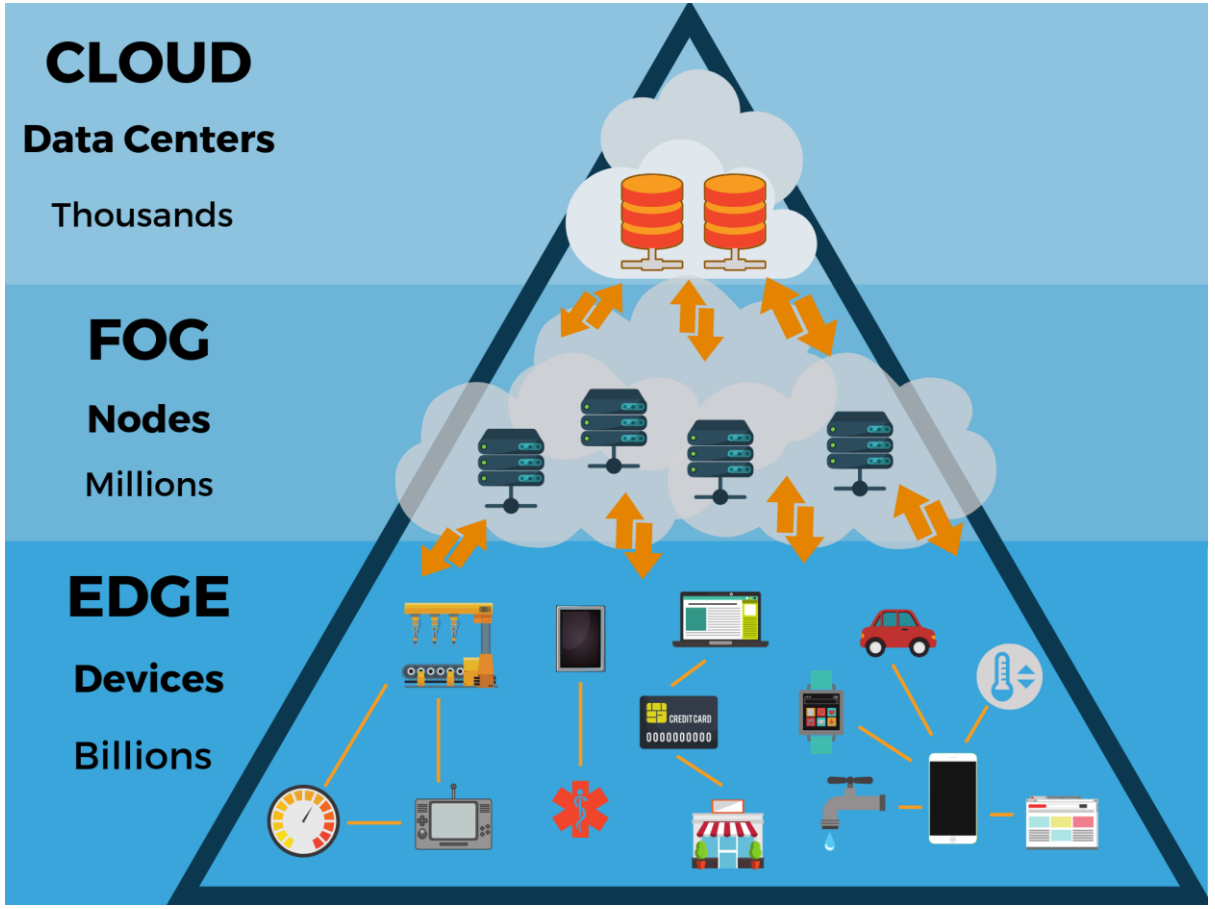
to model obliczeniowy, w którym przetwarzanie danych odbywa się **bliżej źródła ich powstawania** – na urządzeniach końcowych (np. czujnikach, kamerach, dronach) lub lokalnych serwerach brzegowych, zamiast w chmurze.

Zmniejsza to opóźnienia, oszczędza pasmo sieciowe i zwiększa bezpieczeństwo danych.

Fog Nodes (węzły mgły)

to pośrednie jednostki obliczeniowe w architekturze **Fog Computing**, które przetwarzają dane bliżej źródła ich generowania, ale dalej niż urządzenia brzegowe.

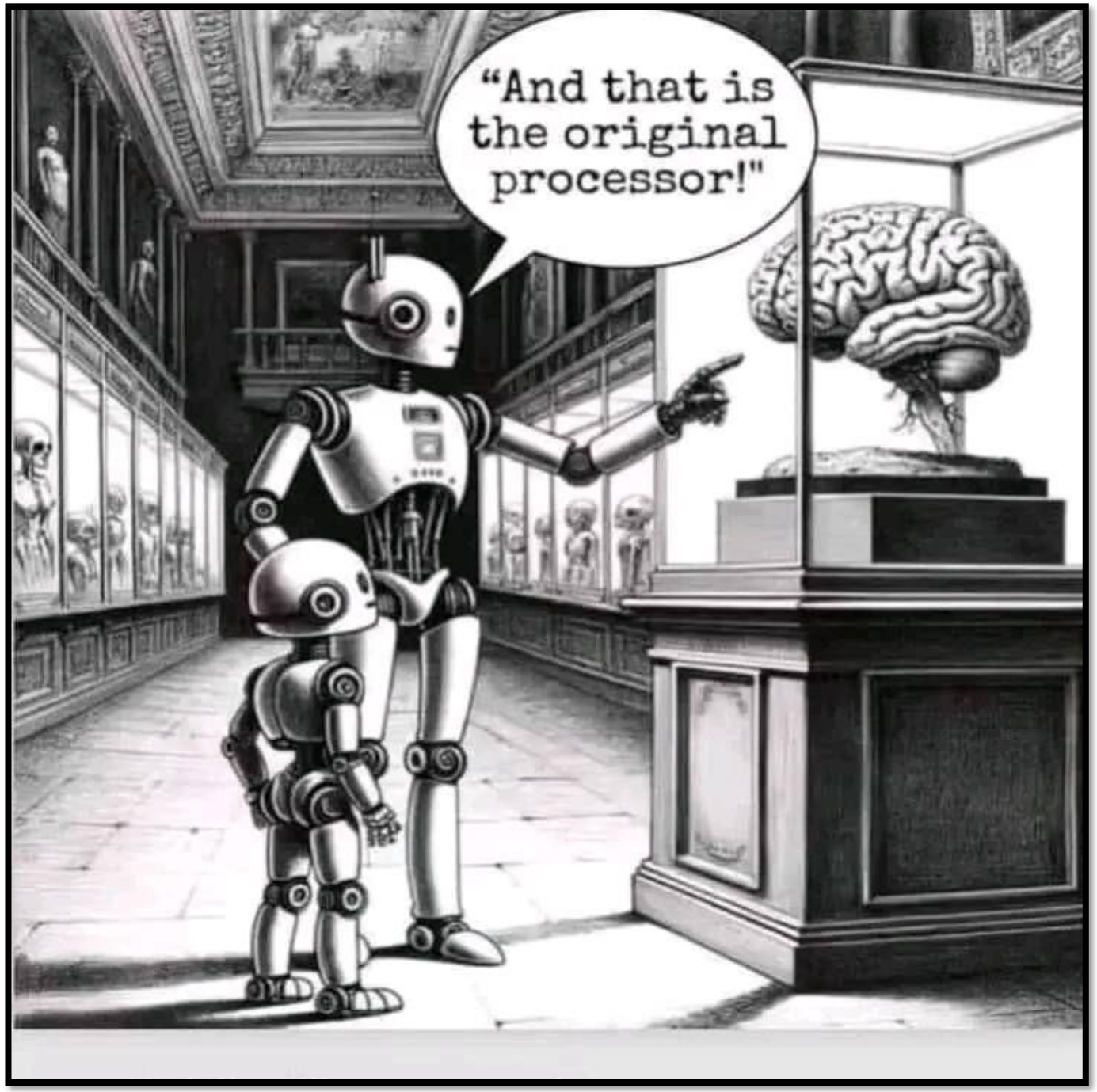
Działają jako pomost między urządzeniami IoT a chmurą, **redukując opóźnienia i obciążenie sieci**.



Przetwarzanie Brzegowe

Schemat blokowy przetwarzania brzegowego dla termostatu AI




"And that is the original processor!"