

Zastosowania algorytmów głębokiego uczenia maszynowego w multimedialnych

wykład z przedmiotu Technologie Multimedialne

dr inż. Adam Kurowski

mgr inż. Szymon Zaporowski



Wprowadzenie

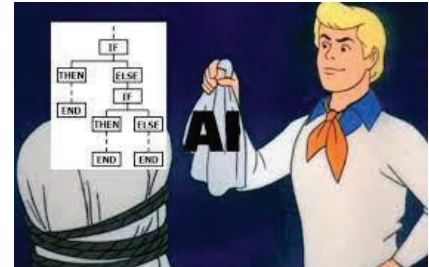
Sztuczna inteligencja umożliwiła wiele operacji na sygnałach składających się na przekaz multimedialny, które **do tej pory wydawały się niemożliwe**.

Efektom tego typu przemian jest powstanie algorytmów pozwalających między innymi na **generowanie obrazów, czy też muzyki, a nawet filmów**.

Możliwe także stało się **modyfikowanie treści multimedialnych w prosty sposób** przy jednoczesnym uzyskiwaniu rezultatów **bardzo wysokiej jakości**.

Klasyczne oblicze sztucznej inteligencji

- SVM
- Drzewa decyzyjne
- Lasy decyzyjne
- „Płytkie” sieci neuronowe



Rozwiązania te **nadal sprawdzają się w pewnych zastosowaniach**, jednak dla aplikacji przedstawionych w tym wykładzie **przetłomem** okazały się ***głębokie sieci neuronowe***.

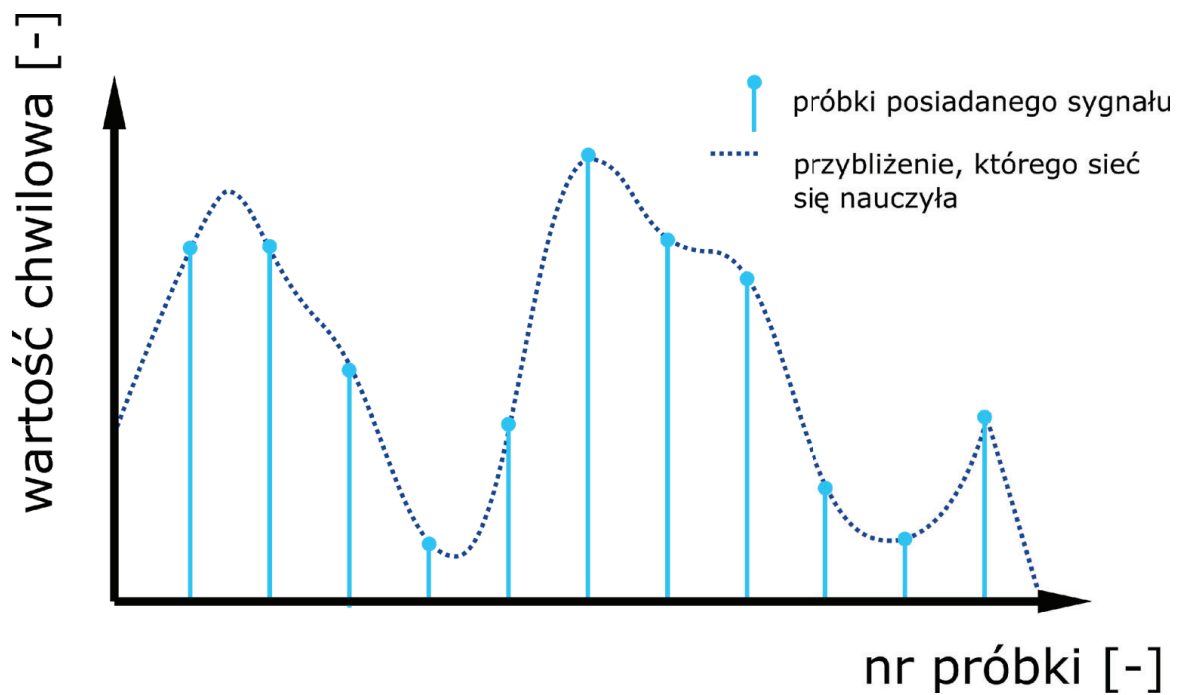
Podstawowe zadania AI

- **klasyfikacja** – czyli przypisywanie danych wejściowych (wektorów liczb, obrazków itp.) do danej klasy obiektów (pies, kot, dom, itp.),
- **regresja** – czyli przybliżanie funkcji, dla której znamy kilka jej wartości dla znanych argumentów,
- **klasteryzacja** – czyli łączenie zbiorów podobnych obiektów w grupy przez samą sieć,
- **interakcja ze środowiskiem** – czyli tzw. uczenie ze wzmocnieniem (analogia z teorią wzmocnień Pawłowa jest nieprzypadkowa), które polega na tym, że sieć uczy się wchodzić w korzystną dla niej interakcję z wybranym środowiskiem.

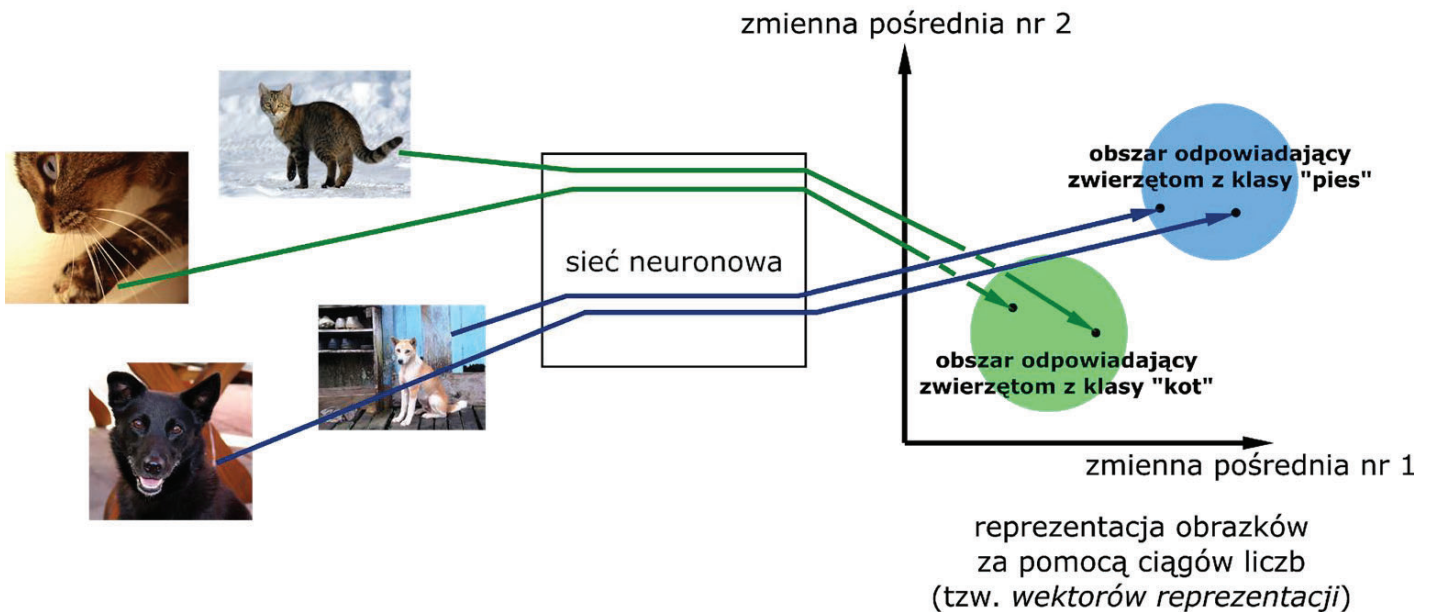
Przykład zastosowania: klasyfikacja



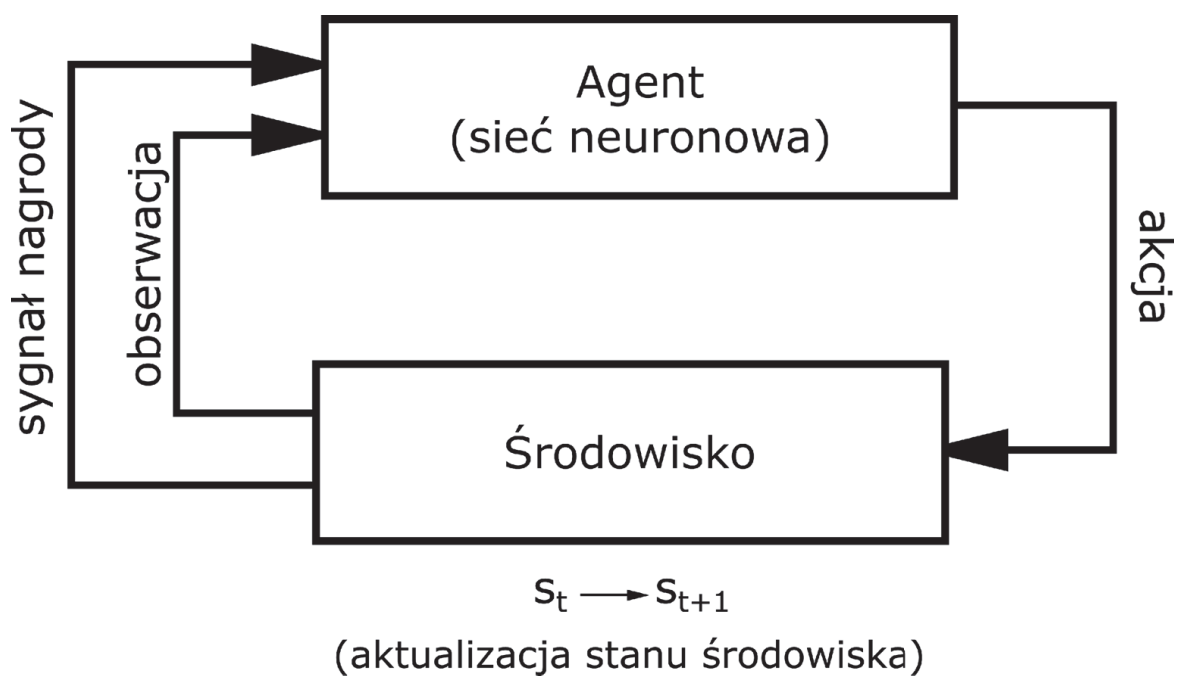
Przykład zastosowania: regresja



Przykład zastosowania: klasteryzacja

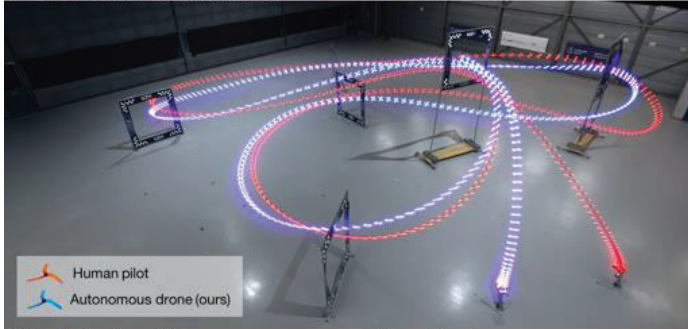


Przykład zastosowania: uczenie ze wzmocnieniem



Przykład zastosowania: uczenie ze wzmocnieniem

a Drone racing: human versus autonomous



b Head-to-head competition



c Human champions



W kwietniu **2025 roku** autonomiczny dron TU Delft wygrał prestiżowe zawody **A2RL Grand Challenge**, pokonując 13 innych dronów AI, a następnie w turnieju knockout pokonał **trzech byłych mistrzów świata DCL**, osiągając prędkości do **95,8 km/h** na wymagającym torze.

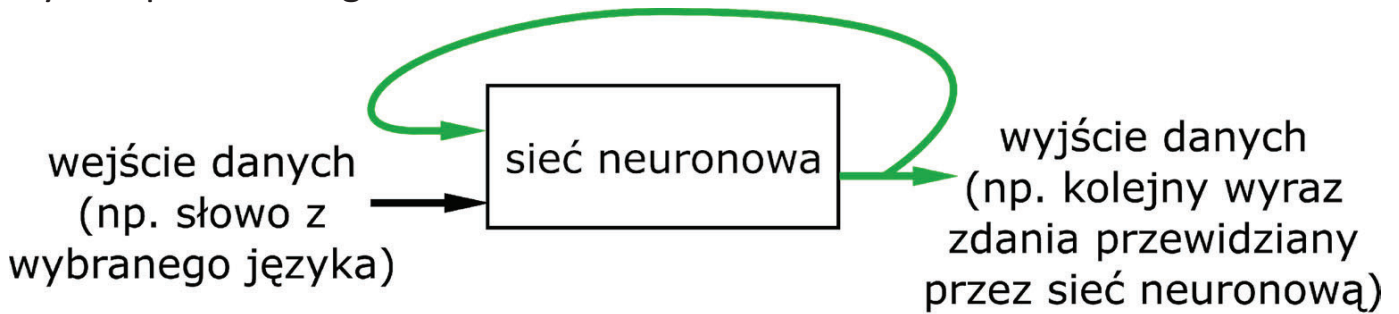
Historyczny przełom: po raz pierwszy autonomiczny robot fizyczny pobił człowieka w sporcie **zaprojektowanym dla i przez ludzi**

źródło: <https://techxplore.com/news/2025-04-autonomous-drone-defeats-human-champions.html>

Problemy trudne do rozwiązania podejściami klasycznymi

- **Generowanie tekstu** – zazwyczaj za pomocą sieci ze sprzężeniem zwrotnym, które na podstawie poprzednich wyrazów zdania przewidują „najlepiej pasujący” wyraz następny,
- **Analiza sentymentu** – czyli przydział (klasyfikacja) przez sieć np. komentarzy internetowych i określania który z nich był pisany z pozytywnym/neutralnym/negatywnym nastawieniem,
- **Generowanie obrazów** – np. obrazów twarzy, lub abstrakcyjnych wzorów,
- **Transfer stylu** – czyli takie przekształcanie już istniejącego obrazu, by np. wizualnie przypominał dzieło znanego artysty.

Przykład problemu: generowanie tekstu



Krok 1. *Ala ma* **ma** ...

Krok 2. *Ala ma kota* **kota** ...

Krok 3. *Ala ma kota* **<koniec_zdania>**

Istnieją także dużo bardziej skomplikowane rozwiązania, w tym przygotowane przez firmę **OpenAI** rozwiązania z rodziny **GPT**. Pierwsze modele z tej rodziny jak **GPT-2** oparte były na architekturze tzw. **transformatora** (*ang. transformer*). Obecnie stosowane są tzw. **LLM – Large Language Models**

Przykład problemu: generowanie tekstu

Ty
Opisz czym różni się Student ETI od studenta EIA na PG?

ChatGPT
Studenci ETI (Elektronika, Telekomunikacja i Informatyka) oraz EIA (Elektrotechnika i Automatyka) na Politechnice Gdańskiej mają różne profile kształcenia, które odpowiadają różnym obszarom specjalizacji i przyszłych ścieżek kariery.

- Specjalizacje i program nauczania:**
 - ETI:** Studenci tego kierunku koncentrują się na szeroko pojętej elektronice i informatyce. Program nauczania obejmuje takie dziedziny jak projektowanie systemów elektronicznych, telekomunikacja, sieci komputerowe, programowanie, przetwarzanie sygnałów i obrazów, oraz inteligencja sztuczna. Jest to kierunek bardziej skupiony na technologiach informatycznych i komunikacyjnych.
 - EIA:** Program nauczania na tym kierunku skupia się na elektrotechnice, automatyce i robotyce. Studenci uczą się o projektowaniu i analizie systemów elektrycznych, automatyce przemysłowej, napędach elektrycznych, energetyce, a także podstawach elektroniki. Jest to kierunek bardziej skoncentrowany na zastosowaniach elektrycznych i automatyki w przemyśle.
- Praktyczne zastosowanie wiedzy:**
 - ETI:** Studenci mają możliwość uczestniczenia w projektach związanych z nowoczesnymi technologiami informacyjnymi i telekomunikacyjnymi, co przygotowuje ich do pracy w sektorze IT, w firmach telekomunikacyjnych czy w dziedzinie rozwoju oprogramowania i sprzętu komputerowego.
 - EIA:** Absolwenci tego kierunku mogą znaleźć zatrudnienie w branżach takich jak energetyka, automatyka przemysłowa, projektowanie systemów sterowania, czy w sektorze produkcji i utrzymania systemów elektrycznych i automatyki.
- Typ projektów i prac dyplomowych:**
 - ETI:** Projekty często obejmują rozwój i testowanie aplikacji, systemów wbudowanych, sieci czy systemów telekomunikacyjnych.
 - EIA:** Projekty dyplomowe mogą koncentrować się na projektowaniu systemów automatyki, symulacjach systemów elektrycznych, projektowaniu układów napędowych, czy integracji systemów automatyki z urządzeniami elektrycznymi.

Podsumowując, główna różnica między studentami ETI a EIA na Politechnice Gdańskiej leży w specyfice kierunków: ETI bardziej skupia się na technologiach informatycznych i telekomunikacyjnych, natomiast EIA koncentruje się na elektrotechnice i automatyce przemysłowej. Wybór między tymi dwoma kierunkami zależy od zainteresowań i preferencji kariery zawodowej.

url:

<https://chat.openai.com/>

Jak mogę Ci dzisiaj pomóc?

- Utwórz plan ćwiczeń na trening siłowy
- Napiisz e-mail z prośbą o przedłużenie terminu mojego projektu
- Wyjaśnij nadprzewodnik jakoby miał pięć lat
- Zaplanuj podróż aby zobaczyć zorzę polarną w Norwegii

Wiadomość ChatGPT...

Chatbot ChatGPT może popełniać błędy. Rozważ sprawdzenie ważnych informacji.

Przykład problemu: generowanie tekstu, kodu etc.

Sz Explain quantum computing in simple terms

Quantum computing is a type of computing that uses quantum bits, or qubits, to process information in ways that traditional computers cannot.

In a classical computer, information is processed and stored in bits, which can only be in one of two states - 0 or 1. In contrast, a qubit can exist in multiple states simultaneously, a property known as superposition. This allows quantum computers to perform certain calculations exponentially faster than classical computers.

Another important aspect of quantum computing is entanglement, where two or more qubits can become correlated in a way that their states are dependent on each other. This property allows for quantum computers to solve certain problems that are practically impossible for classical computers to solve.

Overall, quantum computing has the potential to revolutionize fields such as cryptography, drug discovery, and optimization, but it is still an emerging technology that is challenging to build and operate.

źródło: <https://chat.openai.com/>

Sz Create 1D convolutional neural network code in python. Neural Network has to extract features from raw audio signal

Sure, here's an example 1D convolutional neural network in Python that extracts features from raw audio signals:

```
python
import tensorflow as tf
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Flatten, Dense

# Define the model
model = tf.keras.Sequential()
model.add(Conv1D(filters=32, kernel_size=3, activation='relu', input_shape=(8000, 1)))
model.add(MaxPooling1D(pool_size=2))
model.add(Conv1D(filters=64, kernel_size=3, activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(x_train, y_train, epochs=10, validation_data=(x_val, y_val))
```

Here's a brief explanation of what's happening in the code:

- * We first import the necessary modules from TensorFlow, including 'Conv1D', 'MaxPooling1D', 'Flatten', and 'Dense'.
- * We define our model as a 'Sequential' object, which allows us to stack layers on top of each other in a linear fashion.
- * We add a 1D convolutional layer with 32 filters, a kernel size of 3, and a ReLU activation function. The input shape is (8000, 1), which means our model expects input signals of length 8000 with one channel (i.e., mono audio signals).
- * We add a max pooling layer with a pool size of 2, which helps to reduce the dimensionality of the output from the convolutional layer.
- * We add another 1D convolutional layer with 64 filters, a kernel size of 3, and a ReLU activation function.
- * We add another max pooling layer with a pool size of 2.
- * We flatten the output from the convolutional layers so that it can be passed to a fully connected layer.
- * We add a fully connected layer with 128 neurons and a ReLU activation function.
- * We add another fully connected layer with a single neuron and a sigmoid activation function, which gives us a binary classification output (i.e., whether the input signal belongs to one class or the other).
- * We compile the model using the Adam optimizer, binary crossentropy loss function, and accuracy as the evaluation metric.
- * We train the model using the 'fit' method, providing our training data ('x_train' and 'y_train'), the number of epochs to train for, and validation data ('x_val' and 'y_val') to monitor the model's performance during training.

Przykład problemu: generowanie tekstu

url:
<https://chat.mistral.ai/>

Sz Why ChatGPT is better than Mistral?

ChatGPT and Mistral AI are both advanced language models, but they have different strengths and weaknesses. Here are a few reasons why some people might prefer ChatGPT:

1. Contextual Understanding: ChatGPT is built on the GPT-3 architecture, which is designed to understand and generate human-like text based on the context given. This allows it to handle a wider range of prompts and have more natural-sounding conversations.
2. Training Data: ChatGPT is trained on a diverse range of internet text, which can allow it to provide more detailed and accurate responses in some cases.
3. Versatility: ChatGPT can be used for a wide range of tasks, including drafting emails, writing code, answering questions, tutoring, translating languages, simulating characters for video games, and much more.
4. Customizability: With the GPT-3 API, developers can fine-tune ChatGPT on specific datasets to improve its performance in certain areas.

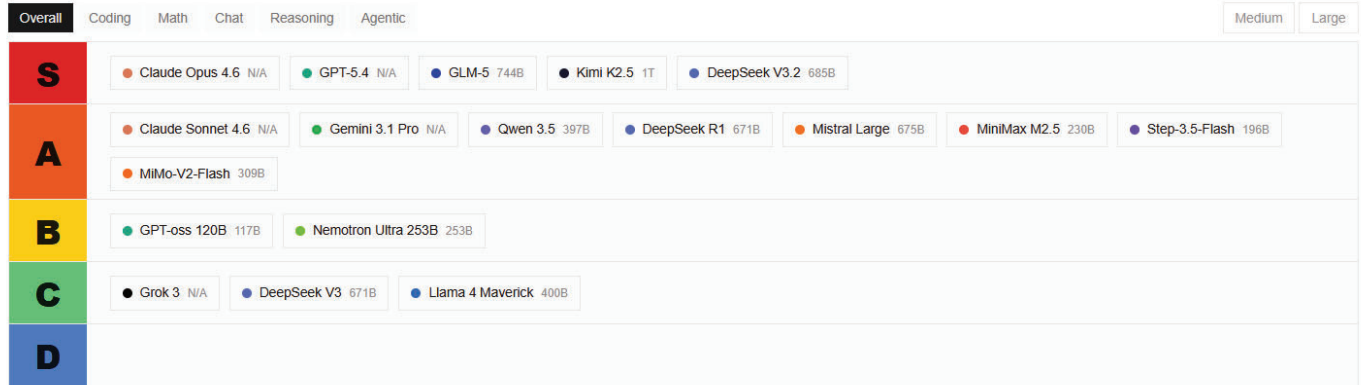
However, it's important to note that Mistral AI also has its own strengths, such as its multilingual capabilities and its focus on providing safe and respectful responses. The best model for you will depend on your specific needs and use cases.



Przykład problemu: generowanie tekstu, kodu etc.

Inne rozwiązania będące interesującymi LLM:

- Llama (META AI)
- Claude (ANTHROPIC - obecnie najlepszy LLM w wersji Claude 4.6 Opus)
- Gemini (GOOGLE)
- Gemma (GOOGLE)
- DeepSeek R1 (DEEPSEEK)
- Qwen (ALIBABA)



Przykład problemu: generowanie tekstu, kodu etc.

POLSKIE MODELE LLM

- Bielik (AGH)
- PLLUM (PWR)
- Qra (PG)

rank	T	Model	Lang	n_shot	#Params (B)	Average	Avg g	Avg mc	Avg RAG	polemo2-in_g
1	🗨️	Mistral-Large-Instruct-2411,max_length=4096	🇸🇰	5	123.0	69.84	73.07	65.64	74.96	89.86
2	🗨️	Meta-Llama-3.1-405B-Instruct-FP8,API	🇸🇰	5	485.0	69.44	72.51	65.44	74.34	89.86
3	🗨️	Mistral-Large-Instruct-2407,max_length=4096	🇸🇰	5	123.0	69.11	72.21	65.08	75.78	88.50
4	🗨️	Qwen2.5-72B-Instruct,max_length=4096	🇸🇰	5	72.0	67.92	69.79	65.50	71.32	86.57
5	🟢	Qwen2.5-72B,max_length=4096	🇸🇰	5	72.0	67.38	69.48	64.65	69.79	82.69
6	🗨️	QwQ-32B-Preview,max_length=4096	🇸🇰	5	32.0	67.01	68.35	65.26	73.80	86.57
7	🟢	Qwen2.5-32B,max_length=4096	🇸🇰	5	32.0	66.73	68.28	64.70	72.91	85.32
8	🗨️	gemma-3-27b-it,max_length=4096,add_bos_token=True	🇸🇰	5	27.0	66.48	69.13	63.03	73.92	87.12
9	🗨️	Llama-3.3-70B-Instruct,max_length=4096	🇸🇰	5	70.0	66.40	70.23	61.42	74.62	87.12
10	🟡	Qwen2-72B,max_length=4096	🇸🇰	5	72.0	66.02	68.30	63.06	75.48	85.46
12	🗨️	Bielik-11B-v3.0-Instruct	🇸🇰	5	11.0	65.93	68.92	62.05	72.65	86.29
13	?	Bielik-11B-v3.0-Instruct_Q4_K_M.gguf	?	5	0.0	65.91	68.97	61.93	71.20	86.57
14	🗨️	Bielik-11B-v2.4-Instruct-SL	🇸🇰	5	11.0	65.87	67.59	63.65	76.21	86.29
14	🟡	Qwen2-72B-Instruct,max_length=4096	🇸🇰	5	72.0	65.87	69.22	61.51	73.32	87.26
16	?	Bielik-11B-v3.0-Instruct_Q8_0.gguf	?	5	0.0	65.82	68.77	61.99	72.14	86.70
17	🗨️	MSH-v1-Bielik-v2.3-Instruct-MedIT-merge	🇸🇰	5	11.0	65.80	67.66	63.39	75.62	85.60
18	🗨️	Bielik-11B-v2.3-Instruct_Q8_0.gguf	🇸🇰	5	11.0	65.76	67.39	63.65	75.67	86.29

PYTANIE

Z których modeli / narzędzi (interfejsów) AI do generowania treści tekstowych korzystasz najczęściej?



Przykład problemu: analiza sentymentu

The screenshot shows a Kaggle dataset page for "Amazon Reviews for Sentiment Analysis". The page includes a header with the dataset name, a subtitle "A few million Amazon reviews in fastText format", and the creator's name "Adam Bittlingmayer" with an update date of "6 months ago (Version 7)". Below the header are navigation tabs for "Data", "Tasks", "Kernels (56)", "Discussion (2)", "Activity", and "Metadata". On the right side, there are buttons for "Download (493 MB)" and "New Notebook". A "Usability" score of 6.9 is displayed, along with tags: "business, computing, natural and physical sciences, reference, internet and 6 more". The "Description" section contains the following text: "This dataset consists of a few million Amazon customer reviews (input text) and star ratings (output labels) for learning how to train fastText for sentiment analysis. The idea here is a dataset is more than a toy - real business data on a reasonable scale - but can be trained in minutes on a modest laptop."

źródło:

<https://www.kaggle.com/bittlingmayer/amazonreviews>

Przykład problemu: analiza sentymentu

Automatically sort your data into sentiment categories: *Positive*, *Negative* and *Neutral*. [Sign up to MonkeyLearn](#) to try our full suite of classifiers – free to use and no code needed.

Paste your text in the model, below:

Test with your own text	Results				
<p>I really love this hardware, I can always trust your solutions!!!</p> <p>Classify Text</p>	<table border="1"><thead><tr><th>TAG</th><th>CONFIDENCE</th></tr></thead><tbody><tr><td>Positive</td><td>99.6%</td></tr></tbody></table>	TAG	CONFIDENCE	Positive	99.6%
TAG	CONFIDENCE				
Positive	99.6%				

źródło: <https://monkeylearn.com/sentiment-analysis-online/>

Przykład problemu: analiza sentymentu

Automatically sort your data into sentiment categories: *Positive*, *Negative* and *Neutral*. [Sign up to MonkeyLearn](#) to try our full suite of classifiers – free to use and no code needed.

Paste your text in the model, below:

Test with your own text	Results				
<p>This was a horrible night. Definitely do not go to that hotel.</p> <p>Classify Text</p>	<table border="1"><thead><tr><th>TAG</th><th>CONFIDENCE</th></tr></thead><tbody><tr><td>Negative</td><td>99.5%</td></tr></tbody></table>	TAG	CONFIDENCE	Negative	99.5%
TAG	CONFIDENCE				
Negative	99.5%				

źródło: <https://monkeylearn.com/sentiment-analysis-online/>

Przykład problemu: generowanie obrazów



źródło: <https://www.thispersondoesnotexist.com/>

Przykład problemu: generowanie obrazów



źródło: <https://www.thispersondoesnotexist.com/>

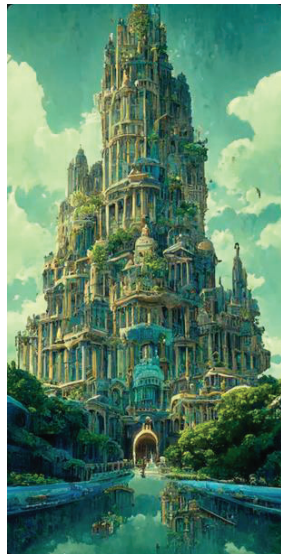
Przykład problemu: generowanie obrazów Stable Diffusion



źródło: <https://stability.ai/blog/stable-diffusion-public-release>

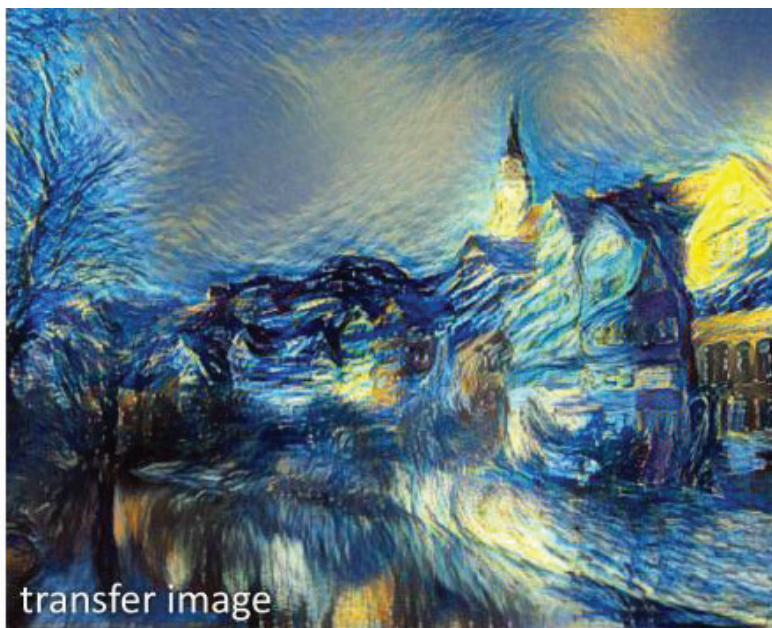
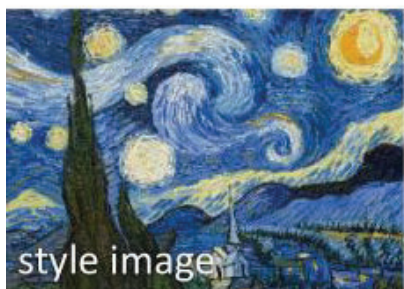
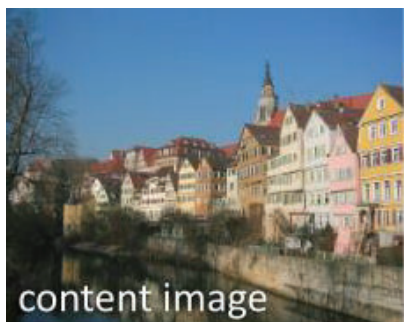


Przykład problemu: generowanie obrazów MidJourney

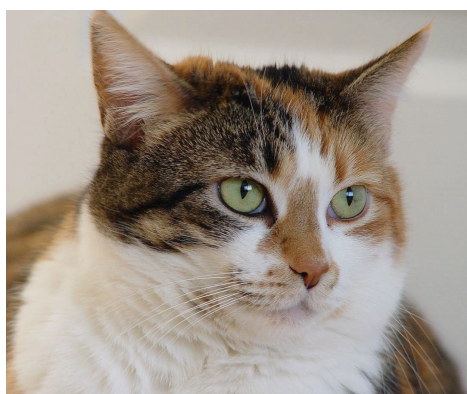


źródło: <https://www.midjourney.com/showcase/top/>

Przykład problemu: transfer stylu



źródło: Gatys et al. [2016] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423. IEEE, June 2016. doi: 10.1109/cvpr.2016.265.



Przykład problemu: transfer stylu (url:
<https://deepdreamgenerator.com>)

Generowanie video

W dwa lata dokonano gigantycznego skoku w jakości generowanych treści video

Obecnie stosowane modele do generowania video:

- **Google Veo 3.1**
- **OpenAI Sora 2**
- **Runway Gen-4.5**
- **Dreamina Seedance 2.0**
- **Wan2.2**



Przetwarzanie dźwięku, generowanie muzyki

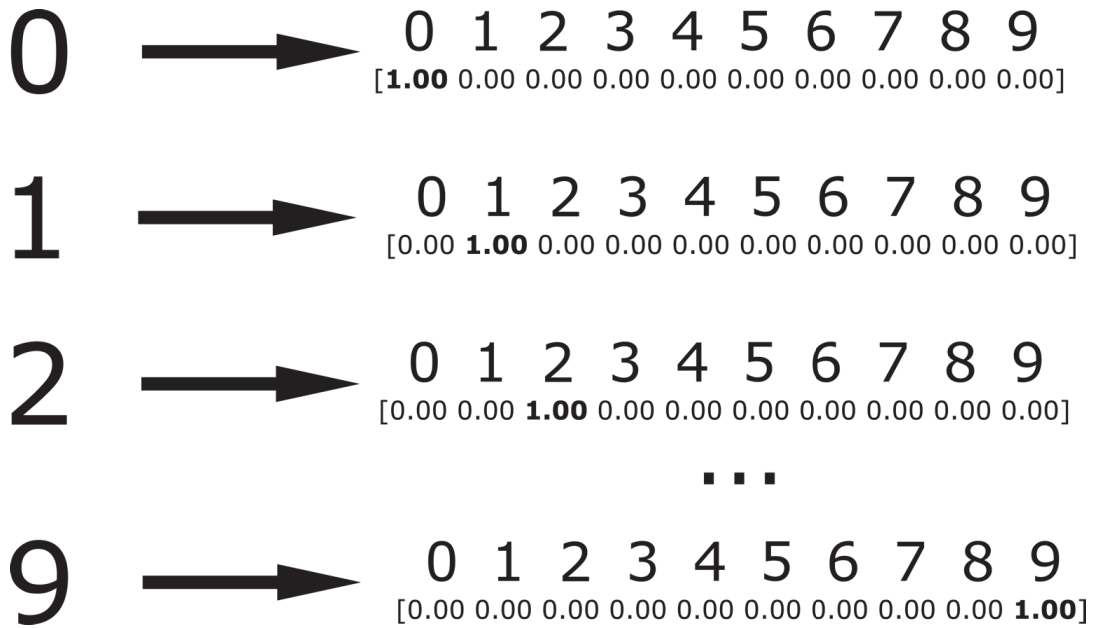
- Generatory takie jak SUNO AI tworzące od podstawy cały utwór w dowolnym, zdefiniowanym przez użytkownika stylu oraz z wykorzystaniem tekstu podanego przez użytkownika jako słów utworu
- Separacja ścieżek audio np. wokalu czy gitary z wykorzystaniem MOISES AI
- Klonowanie Głosu – ElevenLabs, Bark, VITS, Toucan – otwarte modele oraz usługi komercyjne

Uczenie głębokie – sieci splotowe (rozpoznawanie obrazów)

- Są to sieci neuronowe **bardzo rozpowszechnione** we wszelkich dziedzinach uczenia maszynowego, które wykorzystują *uczenie głębokie* (*ang. deep learning*),
- Podobnie jak zwykłe „płaskie” (*ang. feed-forward*) splotowe sieci neuronowe **składają się one z warstw, a te definiowane są przez ich wagi (współczynniki), które modyfikowane są w procesie treningu.**
- Warstwy przetwarzają **dane wejściowe, które są dwuwymiarowymi macierzami wartości** (tak jak np. obrazki w formacie RGB).
- **Każda warstwa aplikuje do obrazka splot** na zasadzie analogicznej jak np. filtry w takich programach jak GIMP, czy Photoshop. Współczynniki tego filtru są obliczane w procesie treningu sieci,

Uczenie głębokie – sieci splotowe (rozpoznawanie obrazów)

- Najprostsze zastosowanie – **klasyfikacja obrazów,**
- **Obrazy podzielone są na klasy,** na przykład powiązane z rodzajem obiektu/zwierzęcia lub imieniem osoby znajdującej się na zdjęciu,
- **Etykieta taka zapisywana jest w formacie one-hot,** który ułatwia interpretację odpowiedzi udzielonej przez sieć na podstawie przekazanego na jej wejście obrazka,
- **Sama sieć zwykle ma tyle wyjść ile elementów ma wektor one-hot,** ale nie zwraca ona wyjść dokładnie w takim formacie. **Zazwyczaj konieczne jest skonwertowanie jej odpowiedzi** na one-hot poprzez ustawienie elementu wektora o maksymalnej wartości wektora na 1 i wyzerowaniu pozostałych.



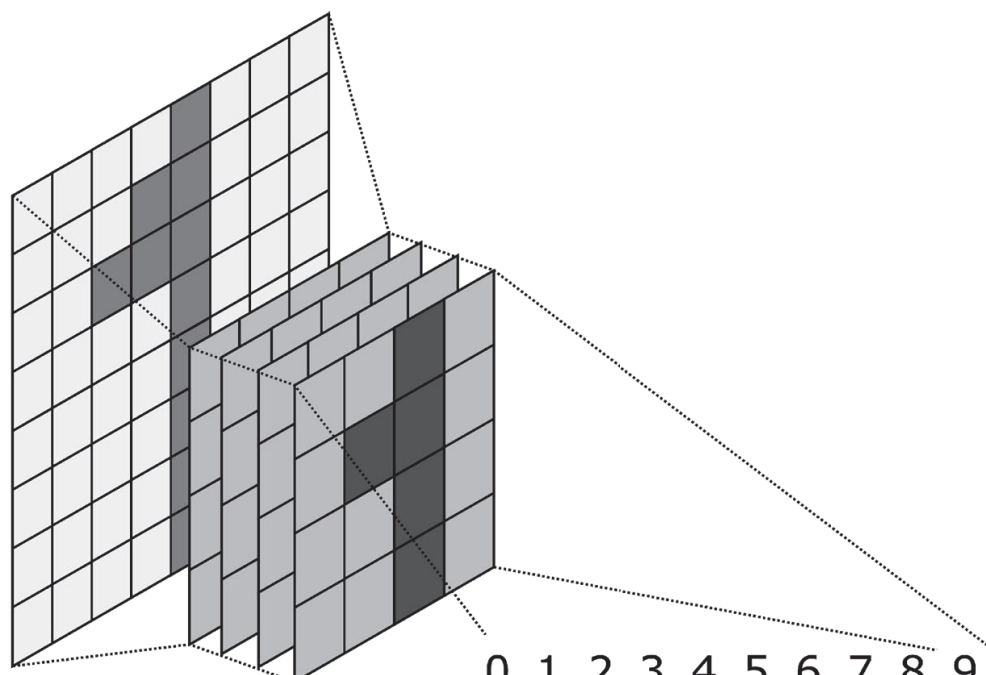
Przykład kodowania klas za pomocą wektora one-hot, którego charakterystyczną cechą jest to, że wektor kodujący daną klasę ma tyle elementów ile jest klas i kodowana klasa jest identyfikowana przez wartość 1 w przypisanym klasie elemencie wektora, a pozostałe elementy równe są zeru.



Przykład bazy pisma odręcznego (zawierającej ręczne zapisy cyfr) o nazwie MNIST, która często wykorzystywana jest do trenowania demonstracyjnych spłotowych sieci neuronowych.


<http://yann.lecun.com/exdb/mnist/>

wejście danych
(obrazek 2D)



0 1 2 3 4 5 6 7 8 9
[0.00 **1.00** 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]
wyjście (etykiątka w formacie one-hot)

Draw your number here



0123456789

Downsampled drawing: 8
First guess: 8
Second guess: 1

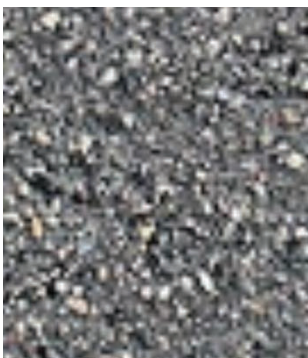
Layer visibility

- Input layer Show
- Convolution layer 1 Show
- Downsampling layer 1 Show
- Convolution layer 2 Show
- Downsampling layer 2 Show
- Fully-connected layer 1 Show
- Fully-connected layer 2 Show
- Output layer Show

Demonstracja wewnętrznych stanów i zasady działania siatki neuronowej.
demo dostępne jest on-line pod adresem: https://adamharley.com/nn_vis/

Uczenie głębokie – modele generatywne (automatycznie tworzenie treści, Deep Dream)

- Najprostszym podejściem do generowania nowej treści jest **odwrócenie procesu uczenia sieci neuronowej**, co zwykle jest realizowane za pomocą algorytmu gradientowego (*ang. gradient descent*).
- **Można ten proces odwrócić i zamiast modyfikować wagi** sieci neuronowej, **tak by ta maksymalizowała wartość zwracaną** w tym elemencie wektora one-hot, który odpowiada poprawnej odpowiedzi **modyfikowane są piksele obrazka** tak, aby obrazek ten coraz bardziej pobudzał wybrane wyjście (wytrenowanej już, umiejącej rozpoznawać daną klasę) sieci.
- Na takiej zasadzie działa znane podejście modyfikacji treści Deep Dream.



0 1 2 3 4 5 6 7 8 9
[0.11 0.09 0.07 0.09 0.13 0.11 0.07 0.13 0.05 0.15]



0 1 2 3 4 5 6 7 8 9
[0.09 0.26 0.06 0.07 0.11 0.07 0.06 0.11 0.04 0.13]



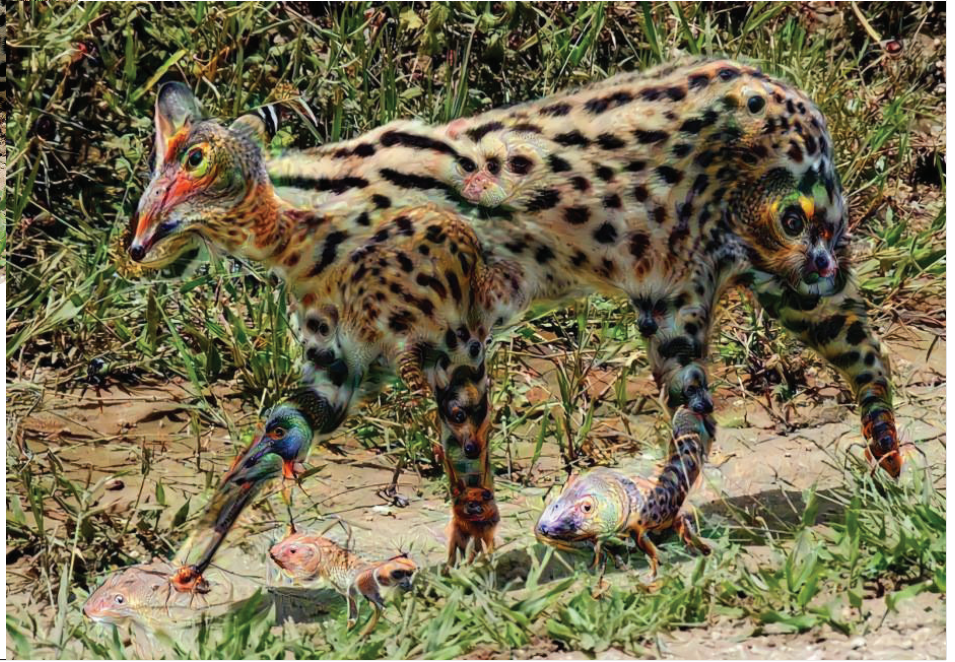
0 1 2 3 4 5 6 7 8 9
[0.05 0.49 0.06 0.05 0.08 0.05 0.04 0.07 0.01 0.10]



0 1 2 3 4 5 6 7 8 9
[0.01 0.75 0.03 0.03 0.04 0.03 0.03 0.02 0.01 0.05]



0 1 2 3 4 5 6 7 8 9
[0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00]



Uczenie głębokie – transfer stylu (kreatywne przetwarzanie treści)

Algorytm modyfikujący piksele można zmodyfikować w taki sposób, aby zachować tę część, która sprawia, że **piksele trenowanego obrazka są w bezpośredni sposób podobne do przykładu wymaganego od sieci.**

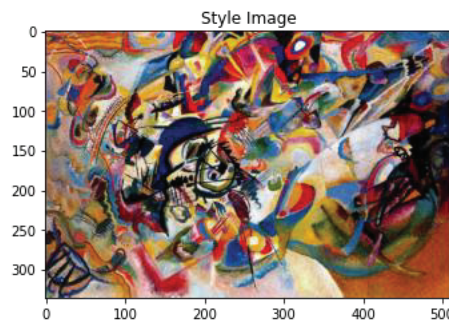
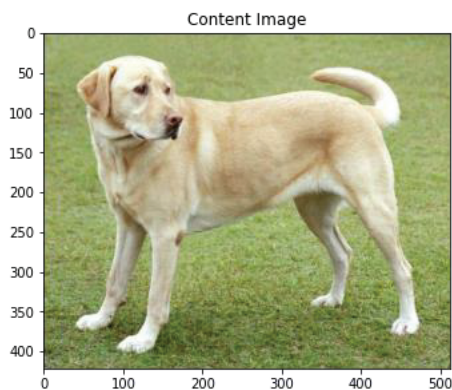
Wymusza się to na poziomie wag w środku samej sieci, która zwykle była już wytrenowana do jakiegoś zadania związanego z rozpoznawaniem obiektów (np. VGG16).

Można jednak dodać nowe wymaganie, które narzuca na sieć konieczność takiego realizowania tego podobieństwa, by **obrazek był także w pewien sposób podobny do trzeciego wejścia, stanowiącego tzw. wejście stylu.**

Uczenie głębokie – transfer stylu (kreatywne przetwarzanie treści)

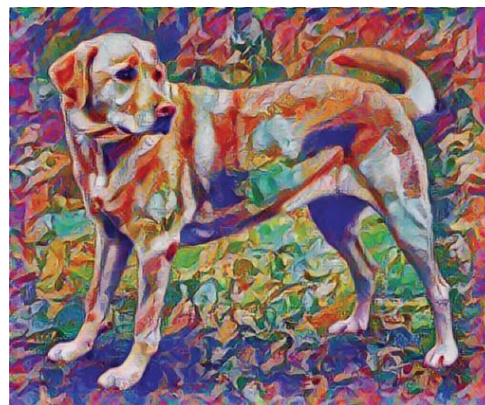
W tym celu **maksymalizuje się dwa podobieństwa:** bezpośrednie podobieństwo do obrazka definiującego treść i podobieństwo tzw. **macierzy Grama** wyliczanej na podstawie przetwarzanego obrazka i obrazka definiującego styl.

Dokładny opis działania tej metody jest dość złożony i wykracza poza ramy tematyczne i czasowe tego wykładu.



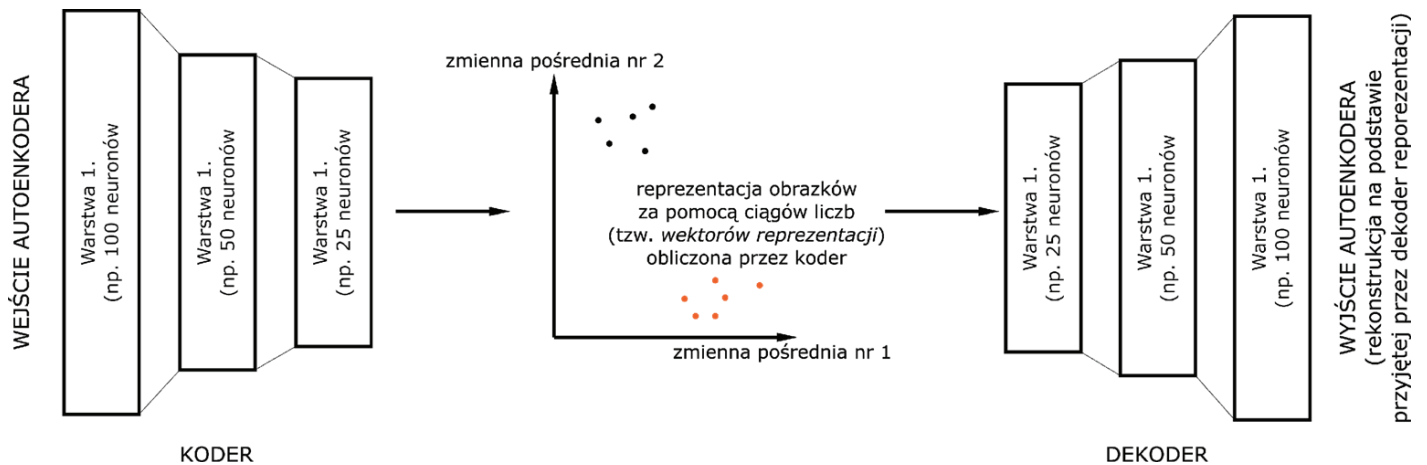
Źródło obrazków:

https://www.tensorflow.org/tutorials/generative/style_transfer



Uczenie głębokie – autoenkodery (kompresja danych, parametryzacja)

- Jest to specjalna modyfikacja struktury klasycznej sieci neuronowej realizującej zadanie klasyfikacji, która została zmodyfikowana tak, aby **zmusić sieć do kodowania danej treści (np. obrazka) na mniejszej ilości liczb, niż ilość liczb oryginalnie wykorzystanych do zapisu tej treści,**
- Może to być wykorzystane na przykład **do kompresji danych,**
- Autoenkoder można też wykorzystać **do kojarzenia danych w grupy i generowanie np. treści podobnych** wizualnie do podanego przykładu,
- **Do zadań grupowania (klasteryzacji) opracowano bardziej złożone i bardziej skuteczne procedury** (tzw. distance metric learning, np. metody softmax-loss i triplet-loss).



Zadaniem autoenkodera jest zrekonstruowanie danych wejściowych z możliwie niskim błędem (najczęściej średniokwadratowym).

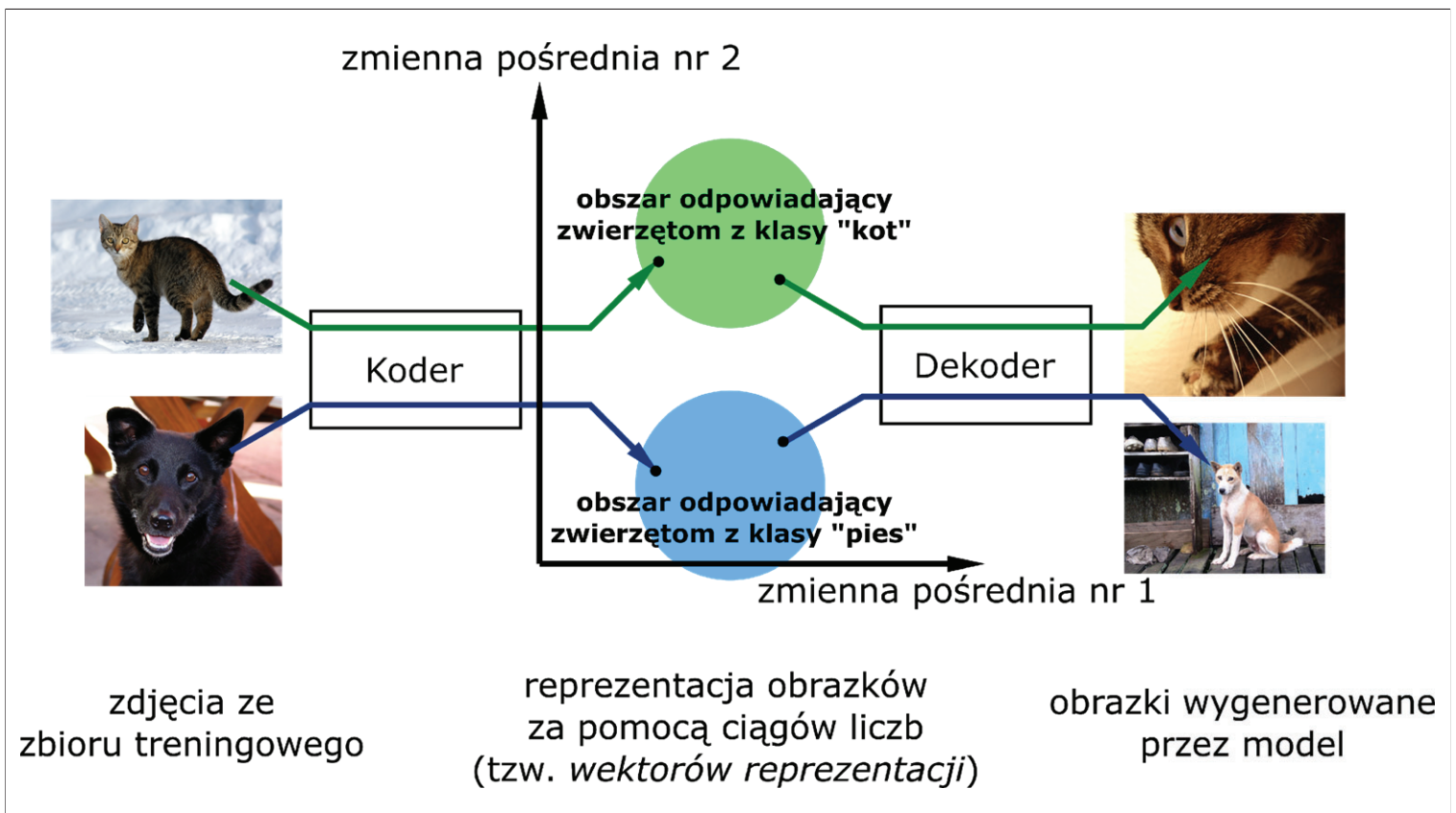
Wektor reprezentacji (w środku autoenkodera) pełni rolę „wąskiego gardła” przez które autoenkoder uczy się przekazywać największą możliwą ilość informacji.

Wariacyjny autoenkoder

- Autoenkoder nie zwraca uwagi na jeden fakt ważny w generowaniu nowej treści – punkty odpowiadające podobnym obrazkom wejściowym **nie zawsze są położone blisko siebie**.
- Możliwe jest dalsze zmodyfikowanie struktury autoenkodera tak, aby ten **nie uczył się dokładnie odwzorowywania 1:1 przykładów**.
- Uczony jest on w inny sposób, **każdy z przykładów wejściowych zamiast z prawdopodobieństwem wypadnięcia w dokładnie wybranym punkcie jest związany z całym obszarem i prawdopodobieństwem wypadnięcia w okolicy tego obszaru** (modelowane najczęściej rozkładem Gaussa)

Wariacyjny autoenkoder

- Dzięki temu **zmuszamy algorytm, by tak budował przestrzeń reprezentacji, by sąsiadujące punkty były podobne do siebie**. Dzięki temu jeżeli chcemy wygenerować obrazek podobny do innego obrazka, to wystarczy że zakodujemy go i zdekodujemy punkt, który jest położony blisko reprezentacji naszego wzorca uzyskanej z kodera.



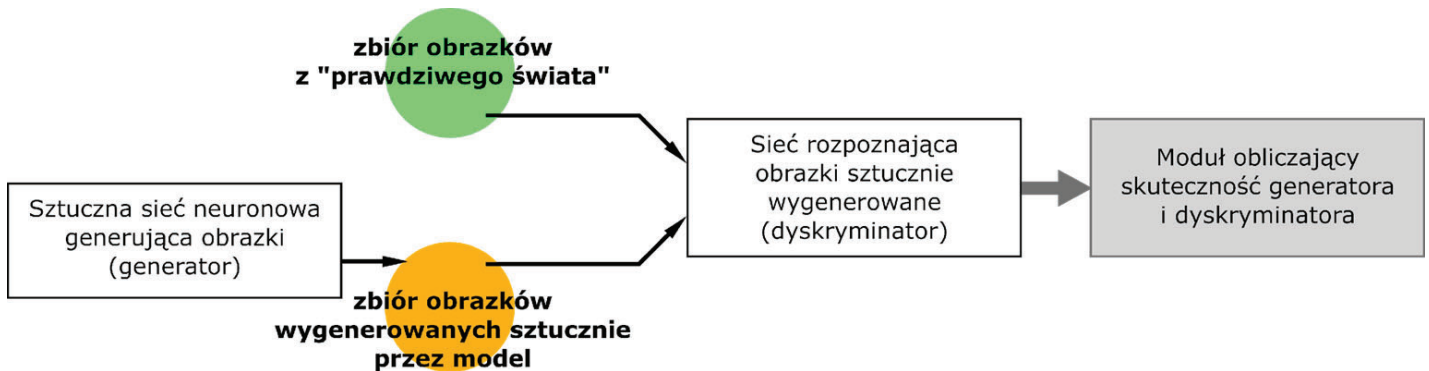
Sieci GAN

- Innym podejściem zmuszającym sieci neuronowe do imitowania kreatywności jest **wprowadzenie konkurencji**,
- W podejściu GAN (*ang. generative adversarial network*), co możemy przetłumaczyć jako model generatywny z dyskryminatorem, mamy dwie sieci,
- Pierwsza sieć, tzw. **generator ma za zadanie generować obrazki**, które są możliwie podobne do obrazków ze zbioru uczącego.
- Druga sieć, tzw. **adwersarz ma za zadanie pobierać dwójki obrazków** (obrazek wygenerowany, obrazek „prawdziwy”) i **uczyć się wykrywać, który obrazek jest prawdziwy, a który wygenerowany.**

Sieci GAN

- **Proces treningu to ciągła rywalizacja, generator uczy się generować coraz bardziej wiarygodne obrazki**, które przypominają obrazki ze zbioru uczącego.
- **Dyskryminator uczy się wykrywać które obrazki są wygenerowane sztucznie i „utrudnia zadanie” generatorowi**, zmuszając go jednocześnie do generowania coraz bardziej wiarygodnych obrazków.

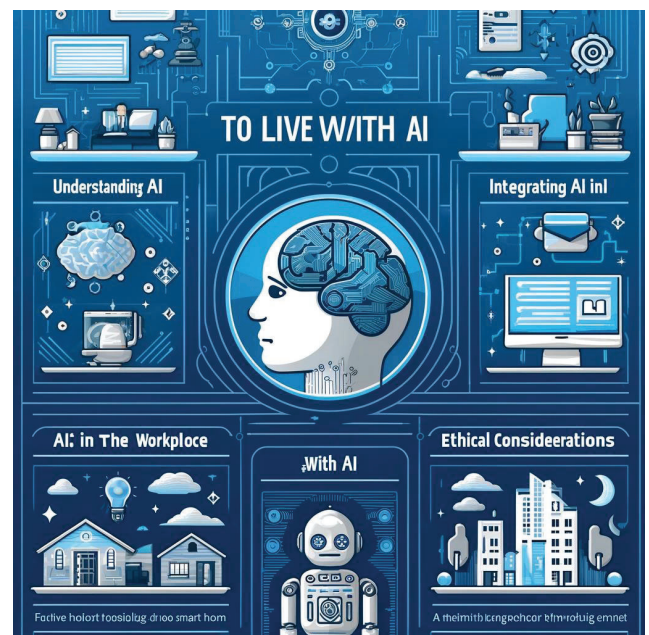
Sieci GAN



Sieć GAN służy między innymi do generowania zdjęć ze strony:
<https://www.thispersondoesnotexist.com/>

Jak żyć wraz rozwojem AI?

- AI to szansa na zwiększenie produktywności - nie na zastąpienie człowieka
- Powstają nowe miejsca pracy (Prompt Engineering) w miejsce redukowanych stanowisk (Testerzy)
- Tym bardziej konieczne jest weryfikowanie informacji



Gdzie bywać, co czytać, kogo znać?



Huggingface – zbiór modeli do szybkiego uruchomienia nawet na Colabie

Medium – artykuły i tutoriale, aby zacząć przygodę z ML

Na YT:

- Sebastian Raschka
- Yanic Kilcher
- Valerio Velardo
- 3blue1brown

Dziękuję za uwagę!