

Zastosowania algorytmów głębokiego uczenia maszynowego w multimedialnych

wykład z przedmiotu Technologie Multimedialne

dr inż. Adam Kurowski

mgr inż. Szymon Zaporowski



1

Wprowadzenie

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

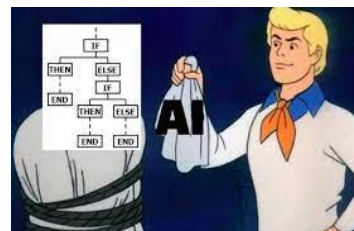
Efektem tego typu przemian jest powstanie algorytmów pozwalających między innymi na **generowanie obrazów, czy też muzyki**.

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

2

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 **przełomem** okazały się ***głębokie sieci neuronowe***.

3

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.

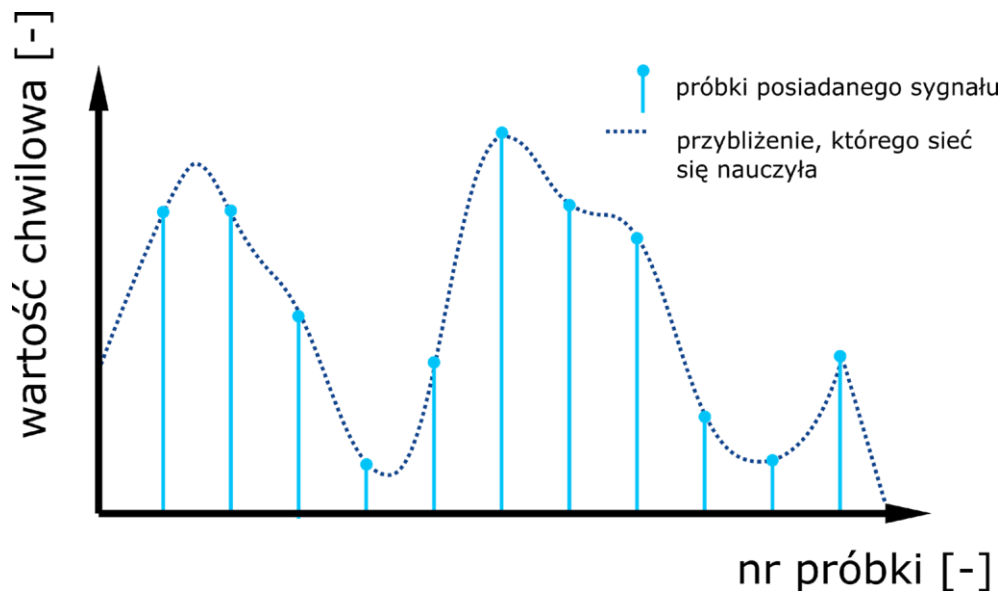
4

Przykład zastosowania: klasyfikacja



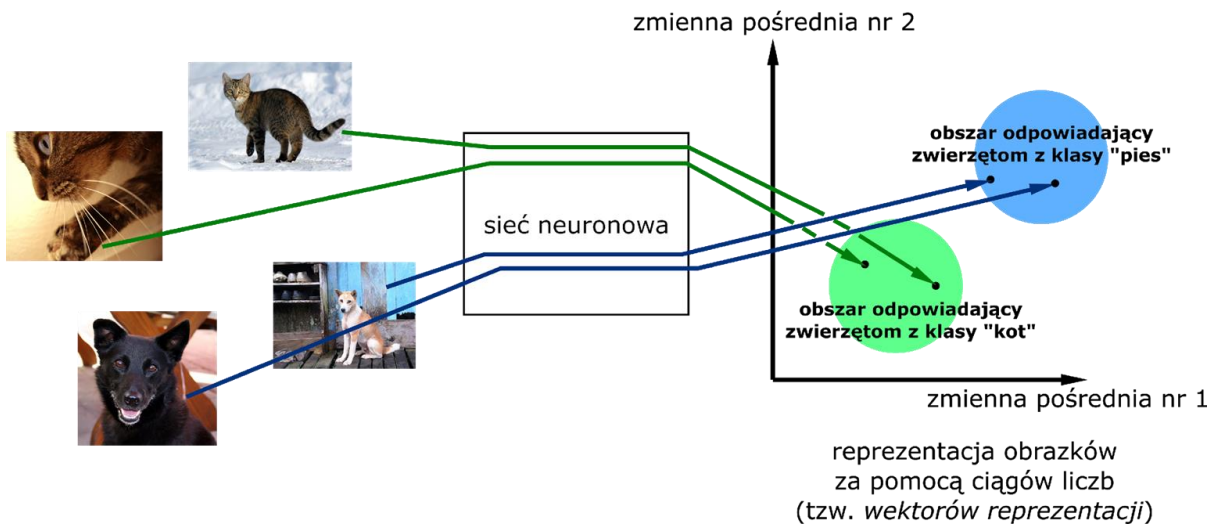
5

Przykład zastosowania: regresja



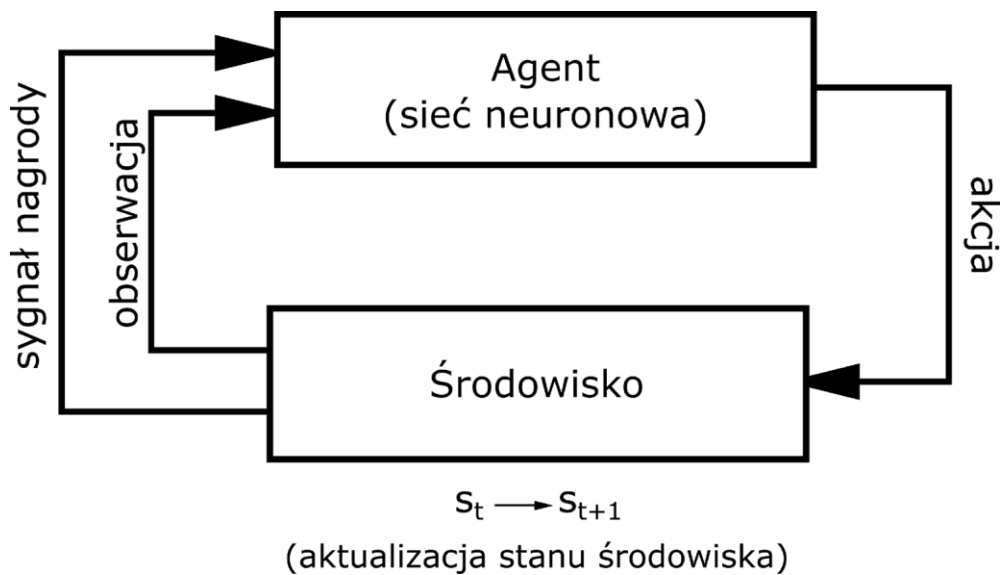
6

Przykład zastosowania: klasteryzacja



7

Przykład zastosowania: uczenie ze wzmocnieniem



8

Przykład zastosowania: uczenie ze wzmocnieniem

GAMING \ TECH \ ARTIFICIAL INTELLIGENCE

DeepMind's StarCraft 2 AI is now better than 99.8 percent of all human players

16

AlphaStar is now grandmaster level in the real-time strategy game

By Nick Statt | @nickstatt | Oct 30, 2019, 2:00pm EDT

f t ↗ SHARE

DeepMind today announced a new milestone for its artificial intelligence agents trained to play the Blizzard Entertainment game *StarCraft II*. The Google-owned AI lab's more sophisticated software, still called AlphaStar, is now grandmaster level in the real-time strategy game, capable of besting 99.8 percent of all human players in competition. The findings are to be published in a research paper in the scientific journal *Nature*.

źródło: <https://www.theverge.com/2019/10/30/20939147/deepmind-google-alphastar-starcraft-2-research-grandmaster-level>

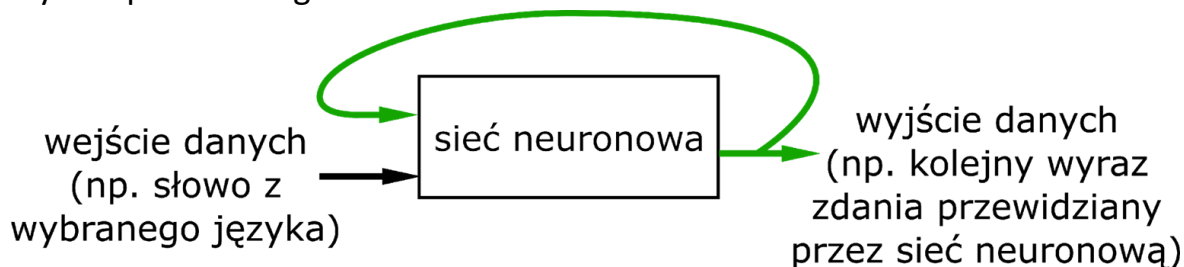
9

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.

10

Przykład problemu: generowanie tekstu



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

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

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

Istnieją także dużo bardziej skomplikowane rozwiązania, w tym przygotowane przez firmę **OpenAI** rozwiązanie **GPT-2**, które swego czasu wzbudziło wiele kontrowersji, bo obawiano się jego wykorzystania do generowania spamu. Sieć ta oparta była na architekturze tzw. **transformatora** (ang. **transformer**).

11

Przykład problemu: generowanie tekstu

url:
https://talktotransformer.com/

Talk to Transformer

See how a modern neural network completes your text. Type a custom snippet or try one of the examples. [Learn more](#) below.

Custom prompt

Hey, I don't mean to be rude, but i have to go to work...|

COMPLETE TEXT

About

Built by [Adam King](#) (@AdamDanielKing) as an easier way to play with OpenAI's new machine learning model. In February, OpenAI unveiled a [language model called GPT-2](#) that generates coherent paragraphs of text one word at a time.

12

Przykład problemu: generowanie tekstu

url:
<https://talktotransformer.com/>

GENERATE ANOTHER

Completion

Hey, I don't mean to be rude, but i have to go to work...

You do? Great. Don't go wandering around the Net then.

I mean, sometimes I'm just too lazy to go anywhere.

Ah, well, okay then. Let's see, now that we're doing this for you,

I'll make a list of chores we can do.

Yeah, I like that idea. I'm willing to assist you in whatever way I can.

You can ask me what I want to do.

I'll just write down the name of what I want to do...

And what the necessities are.

You can tell me about your tasks or just tell me

13

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(10, activation='softmax'))
model.add(Dense(1, activation='sigmoid'))

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

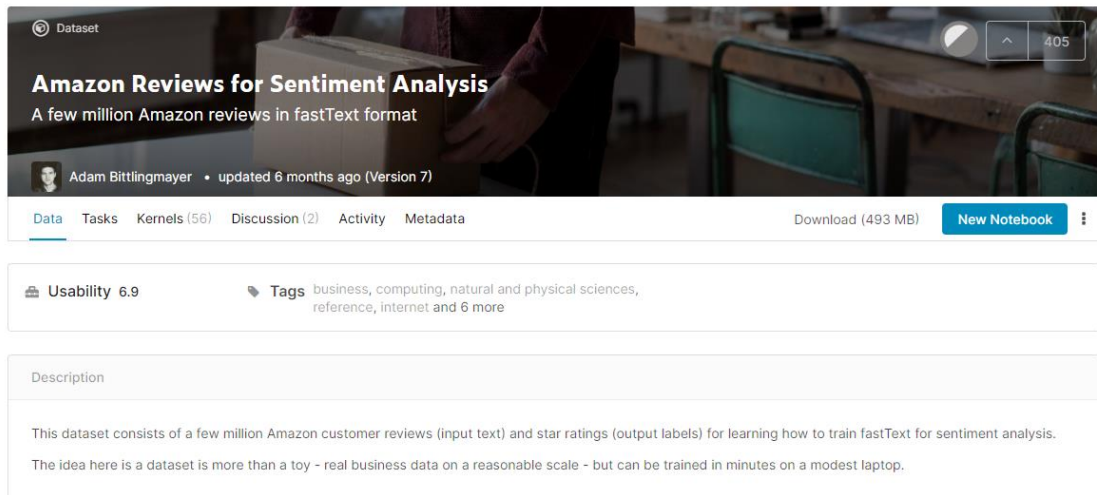
# Train the model
model.fit(x_train, y_train, epochs=10, validation_data=(x_val, y_val))
```

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 108 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.

14

Przykład problemu: analiza sentymentu



The screenshot shows a Kaggle dataset page titled "Amazon Reviews for Sentiment Analysis". The subtitle is "A few million Amazon reviews in fastText format". The dataset is by Adam Bittlingmayer, updated 6 months ago (Version 7). It has a usability score of 6.9 and tags including "business, computing, natural and physical sciences, reference, internet and 6 more". The description states: "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>

15

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

I really love this hardware, I can always trust your solutions!!!

Results

| TAG | CONFIDENCE |
|----------|------------|
| Positive | 99.6% |

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

16

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

This was a horrible night.
Definitely do not go to that hotel.

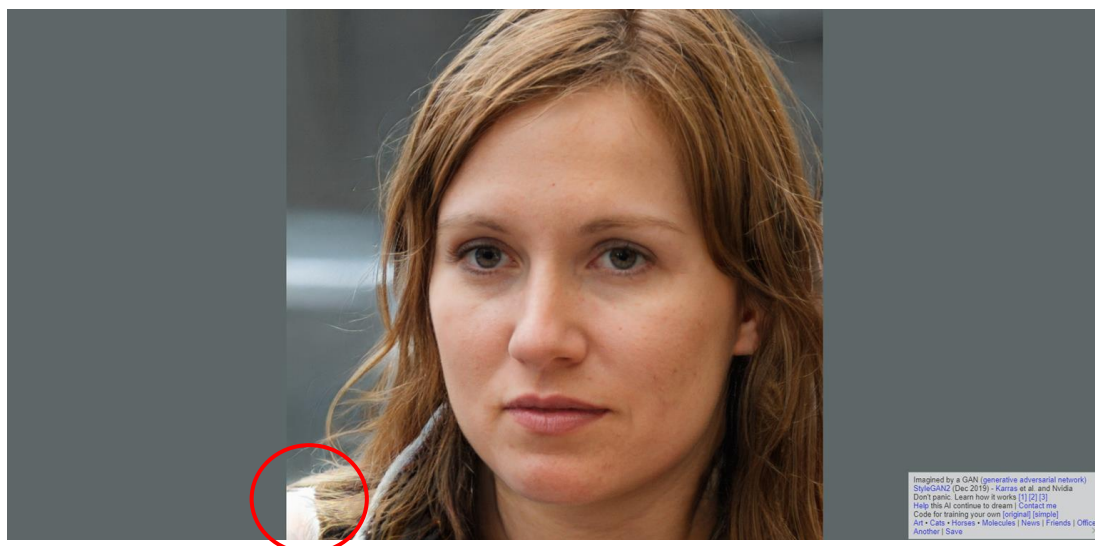
Results

| TAG | CONFIDENCE |
|----------|------------|
| Negative | 99.5% |

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

17

Przykład problemu: generowanie obrazów



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

18

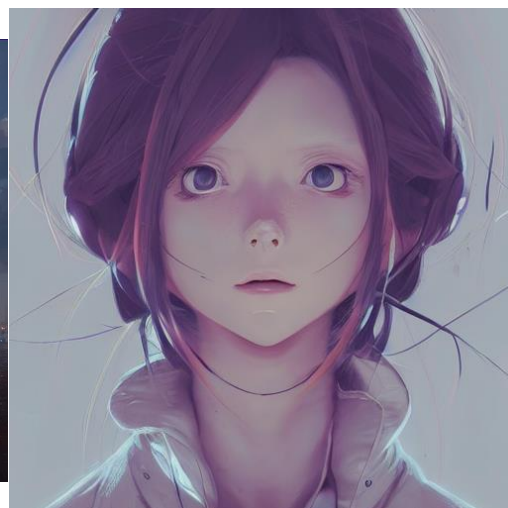
Przykład problemu: generowanie obrazów



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

19

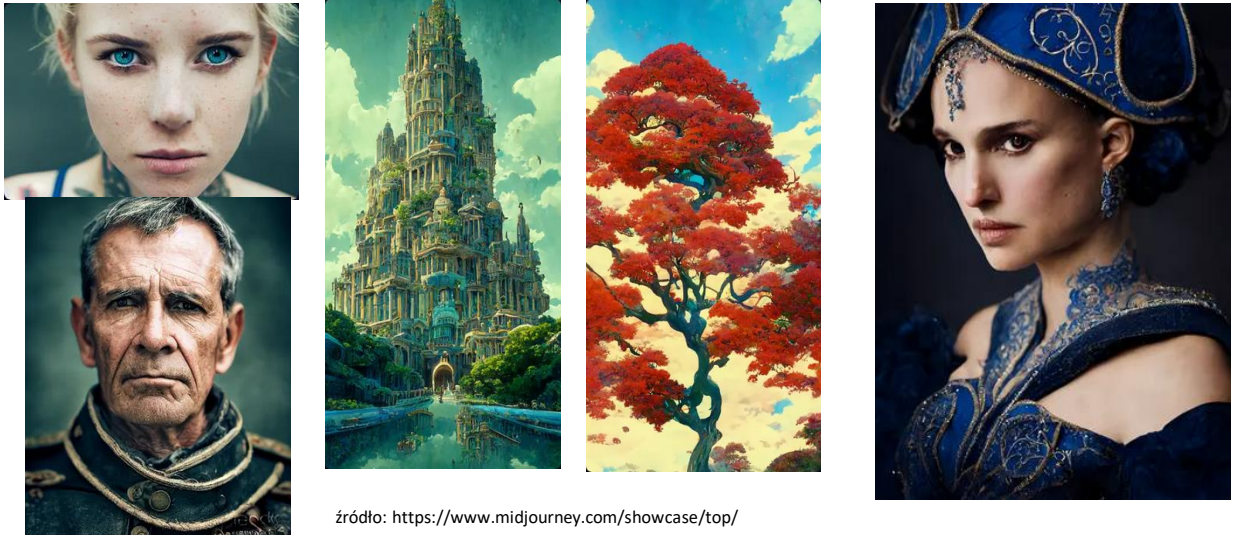
Przykład problemu: generowanie obrazów Stable Diffusion



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

20

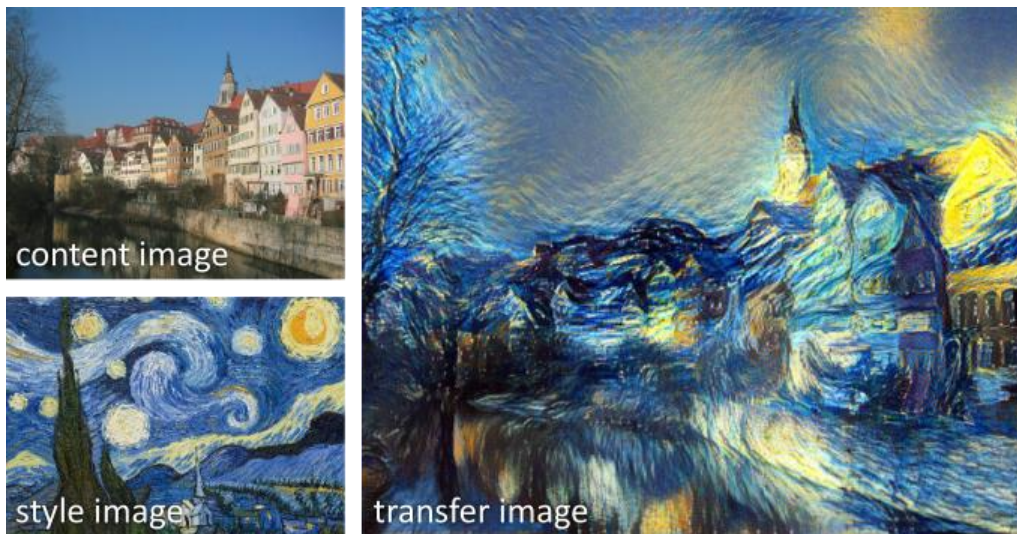
Przykład problemu: generowanie obrazów MidJourney



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

21

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.

22



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

23

Uczenie głębokie – sieci spłotowe (rozpoznawanie obrazów)

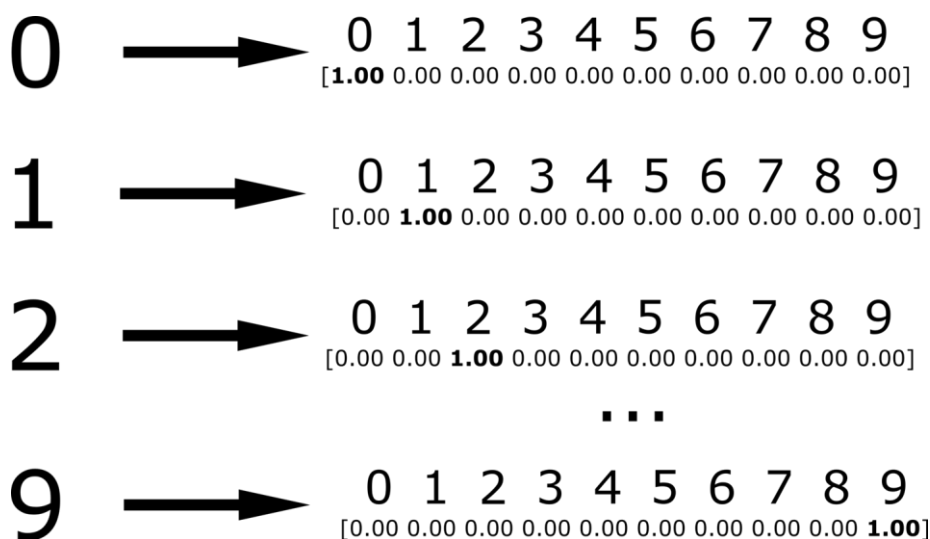
- Są to sieci neuronowe **bardzo rozpowszechnione** we wszelkich dziedzinach uczenia maszynowego, które wykorzystują *uczenie głębokie* (*ang. deep learning*),
- Podobnie jak zwykłe „płaskie” (*ang. feed-forward*) spłotowe sieci neuronowe **składają się one z warstw, a te definiowane są przez ich wagi (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 spłot** na zasadzie analogicznej jak np. filtry w takich programach jak GIMP, czy Photoshop. Współczynniki tego filtru są obliczane w procesie treningu sieci,

24

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.

25



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.

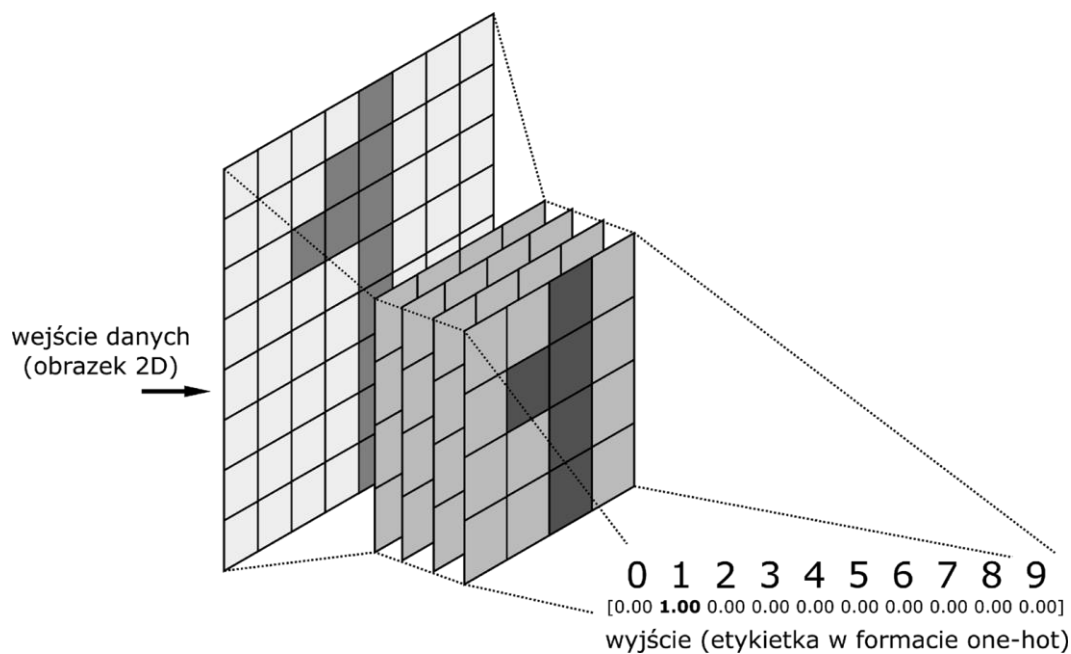
26

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

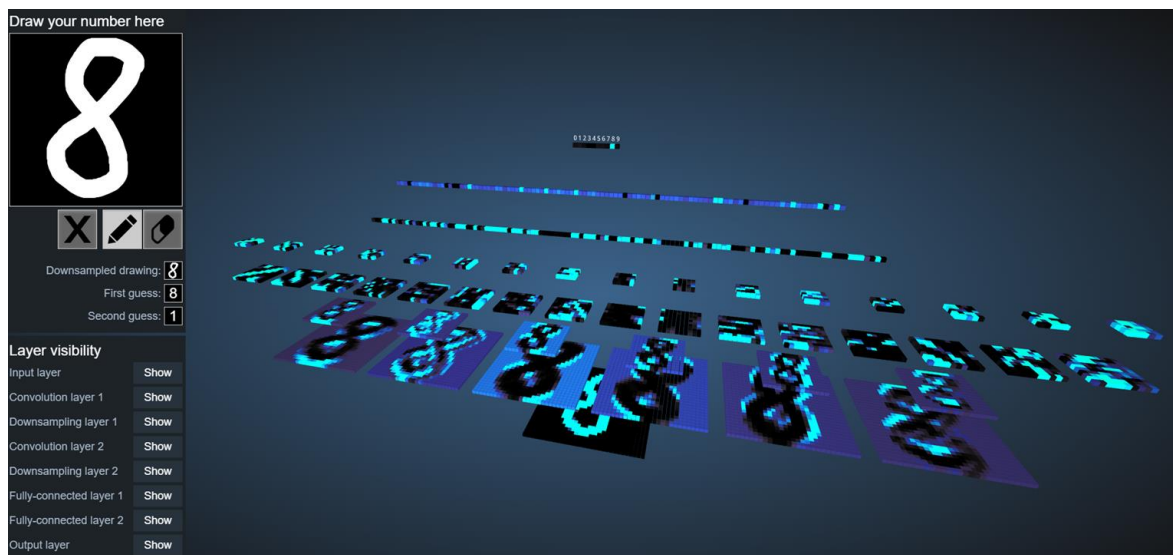
Przykład bazy pisma odręcznego (zwierającej ręczne zapisy cyfr) o nazwie MNIST, która często wykorzystywana jest do trenowania demonstracyjnych splotowych sieci neuronowych.

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

27



28



Demonstracja wewnętrznych stanów i zasady działania splotowej sieci neuronowej.

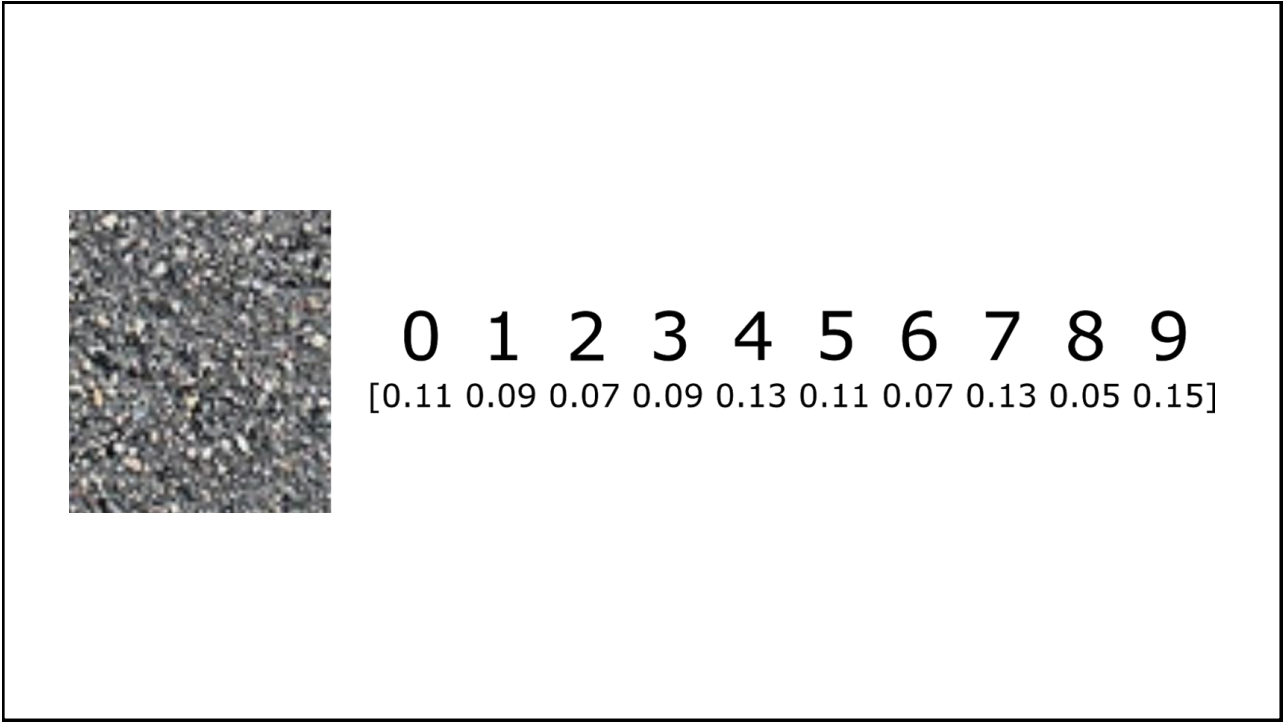
demo dostępne jest on-line pod adresem: <https://www.cs.ryerson.ca/~aharley/vis/conv/>

29

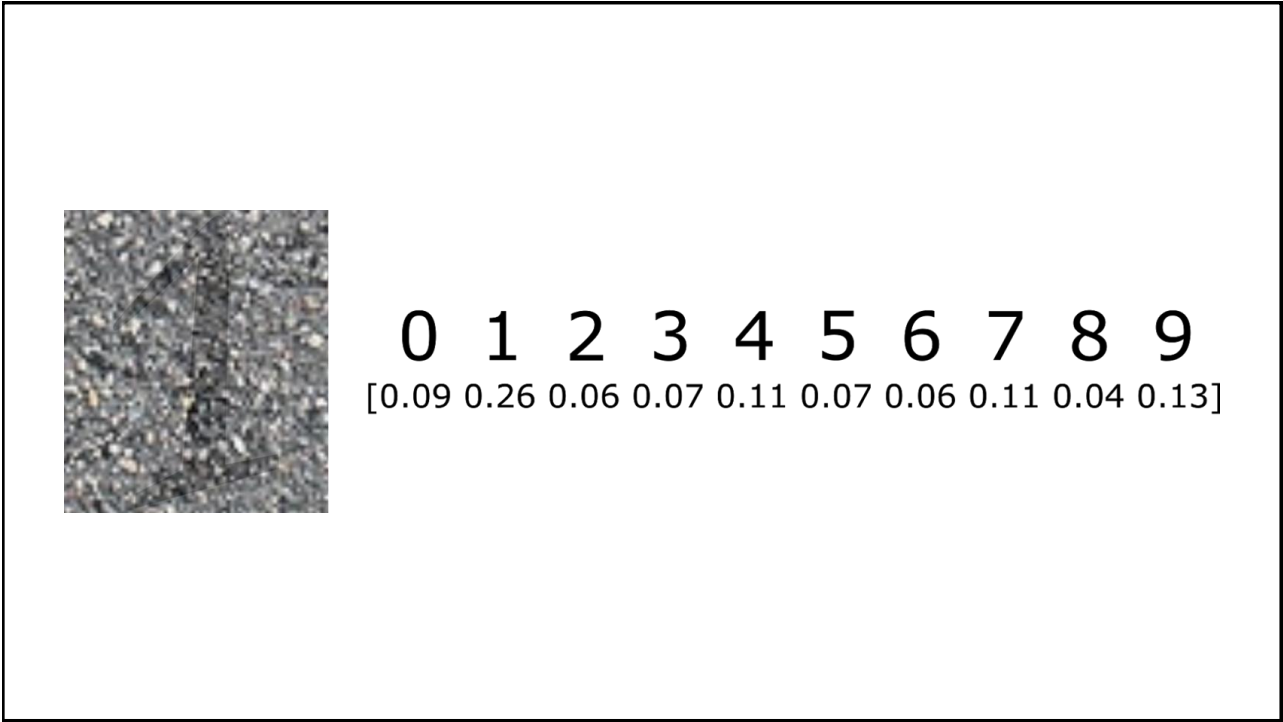
Uczenie głębokie – modele generatywne (automatycznie tworzenie treści, Deep Dream)

- 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.


30



31




32



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]

33



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]

34

1

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]

35



36



37

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 przykładowego wymagane 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.**

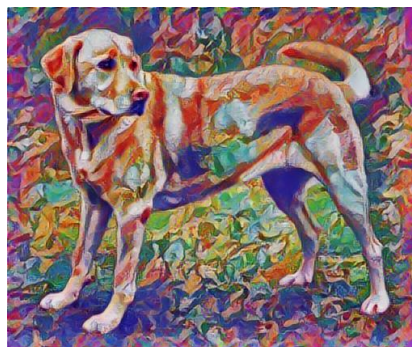
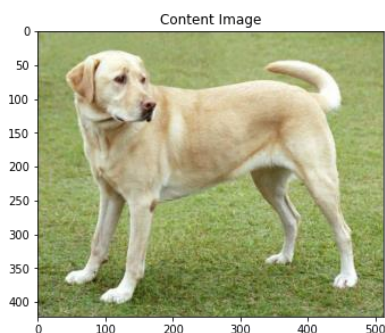
38

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.

39



Źródło obrazków:

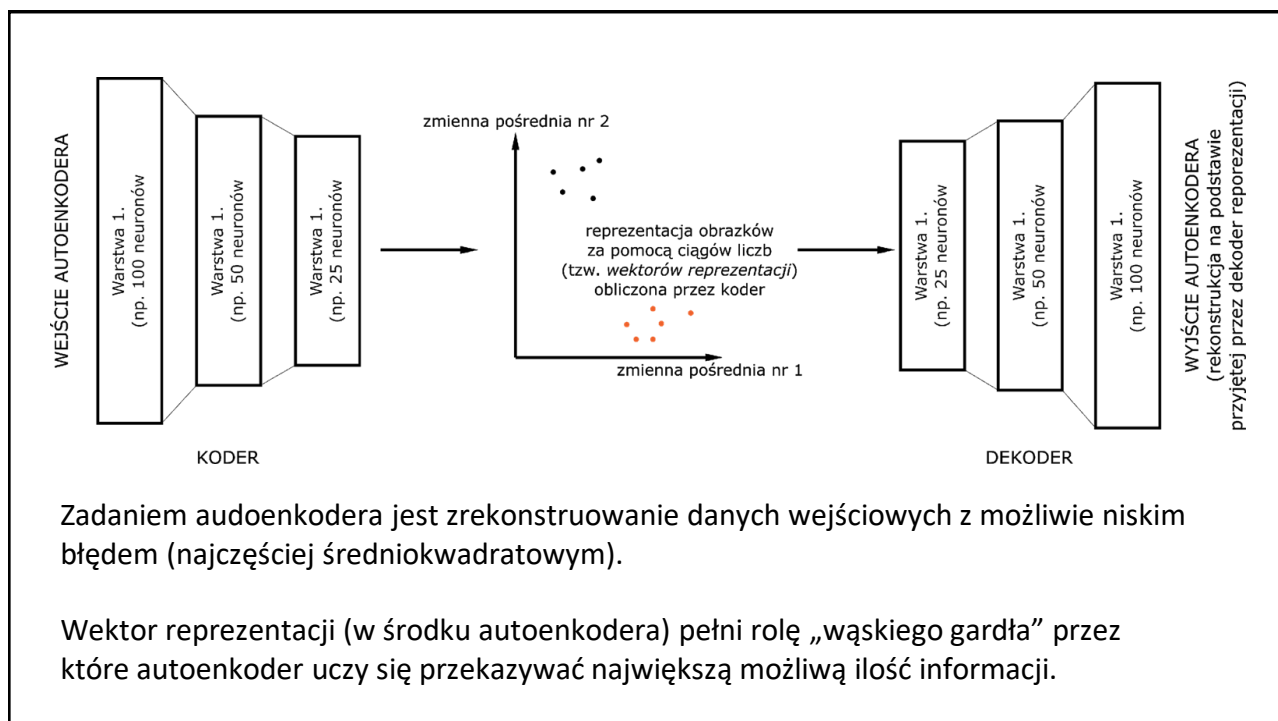
https://www.tensorflow.org/tutorials/generative/style_transfer

40

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).

41



42

Wariacyjny autoenkoder

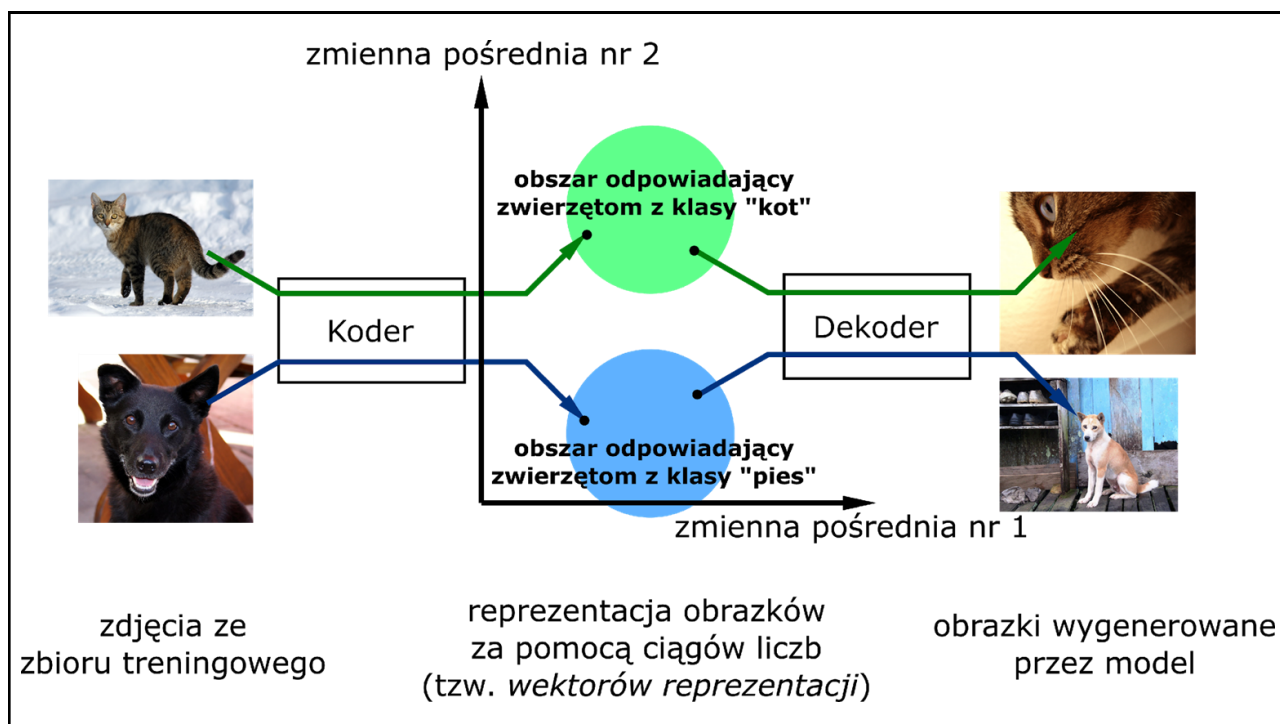
- Autoenkoder nie zwraca uwagi na jeden fakt ważny w generowaniu nowej treści – punkty odpowiadające podobnym obrazkom wejściowym **nie zawsze są położone blisko siebie**.
- Możliwe jest dalsze zmodyfikowanie struktury autoenkodera tak, aby ten **nie uczył się dokładnie odwzorowywania 1:1 przykładów**.
- 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)

43

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.

44



45

Sieci GAN

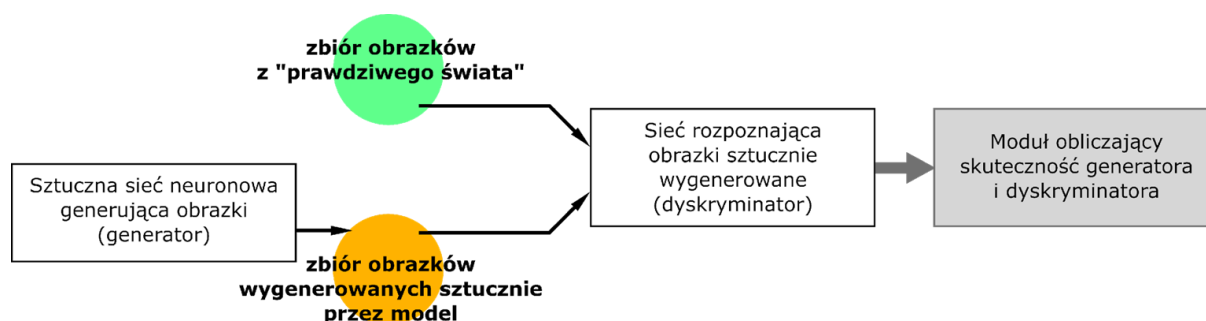
- Innym podejściem zmuszającym sieci neuronowe do imitowania kreatywności jest **wprowadzenie konkurencji**,
- W podejściu GAN (*ang. generative adversarial network*), co możemy przetłumaczyć jako model generatywny z dyskriminatorem, mamy dwie sieci,
- Pierwsza sieć, tzw. **generator** ma za zadanie **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**.

46

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.**

47



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

48

Dziękuję za uwagę!