

Głębokie Przetwarzanie Tekstu i Mowy

NLP z wykorzystaniem GPT-2 lub GPT-3

Adam Kurowski

Katedra Systemów Multimedialnych,
Wydział Elektroniki, Telekomunikacji i Informatyki PG



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

Wprowadzenie

Przetwarzanie języka naturalnego oraz innego rodzaju danych tekstowych jest jedną z tych sytuacji w których konieczne jest wykonanie **pogłębionego przetwarzania danych wejściowych**.

Problem ten możemy uogólnić i przedstawić jako tzw. **problem przetwarzania sekwencji**. Do rozwiązywania tego typu problemów istnieje specjalna klasa algorytmów takich jak np.:

- Sieci rekurencyjne
- Sieci LSTM, GRU i podobne,
- Sieci rekurencyjne z mechanizmem atencyjnym,
- Transformery.

Tokenizacja danych tekstowych

- Przypisanie do wyrazów predefiniowanych wektorów (kodowanie ich np. na zasadzie one-hot),
- **Continuous Bag of Words (CBOW)** – reprezentacje trenowane w taki sposób, by suma np. 2 sąsiadów z lewej i prawej strony w danym zdaniu w sumie dawały mniej więcej wektor reprezentacji słowa w środku,
- **Skip-gram** – kodowanie uzyskane poprzez wytrenowanie sieci neuronowej z 1 warstwą ukrytą do przewidywania sąsiadów tego słowa na podstawie jego reprezentacji (na podobnej zasadzie jak w algorytmie CBOW). Po czasie wyjście warstwy ukrytej może być stosowane jako reprezentacja danego słowa,

Tokenizacja danych tekstowych

- **Byte Pair Encoding (BPE)** - jest to technika oparta o tak samo nazwany sposób kompresji danych. Zasada działania tego algorytmu polega na podzieleniu mniej popularnych słów na podciągi liter. Podział ten premiuje podciągi częściej występujące. Najczęściej występujące litery i ciągi liter są łączone aby wygenerować słownik (rozmiar słownika to hiperparametr).

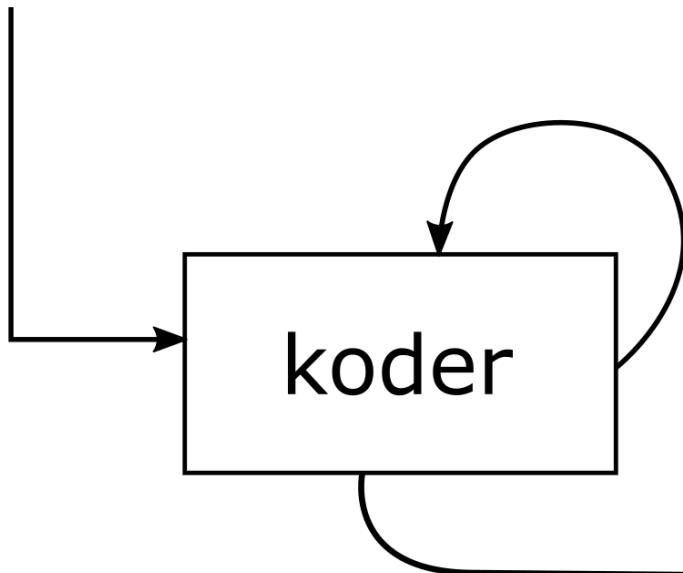
Przetwarzanie sekwencji w modelu koder-dekoder

Aby uczynić przetwarzanie bardziej uniwersalnym możliwe jest specjalne zaprojektowanie sieci tak, aby:

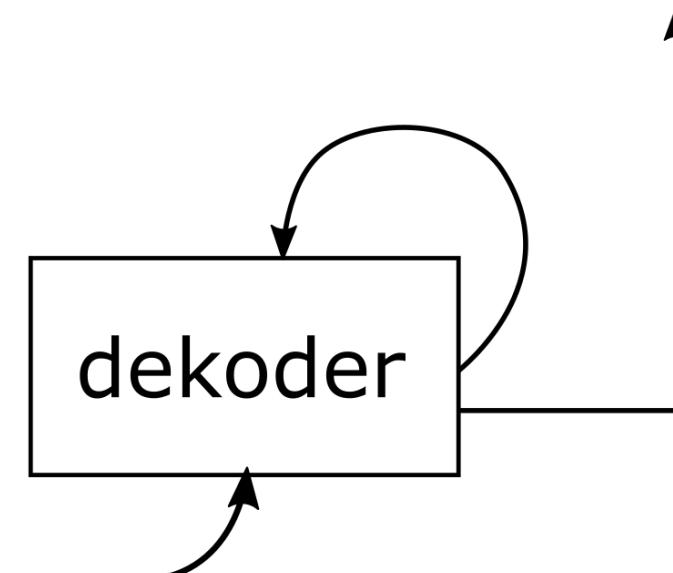
- **pierwsza jej część** przyjmowała jeden **specyficzny rodzaj wejścia** (np. serię obrazków, albo słowa w języku angielskim),
- **druga jej część** jest w stanie przyjąć specjalny, uniwersalny rodzaj wektora i na jego podstawie wygenerować **specyficzny rodzaj wyjścia** (np. opis tekstowy, albo tekst w języku francuskim),
- **pierwsza część generuje uniwersalne wyjście możliwe do zinterpretowania przez dowolny rodzaj drugiej części sieci**, w ten sposób możliwa jest realizacja sieci do generowania tekstu opisu filmów, albo tłumaczenia maszynowego.

Przetwarzanie sekwencji w modelu koder-dekoder

Ala ma kota



Alice has a cat



wektor pośredni

Sieci rekurencyjne i ich usprawnienia

Częstym wyborem do przetwarzania sekwencji są sieci posiadające jakiś rodzaj sprzężenia zwrotnego:

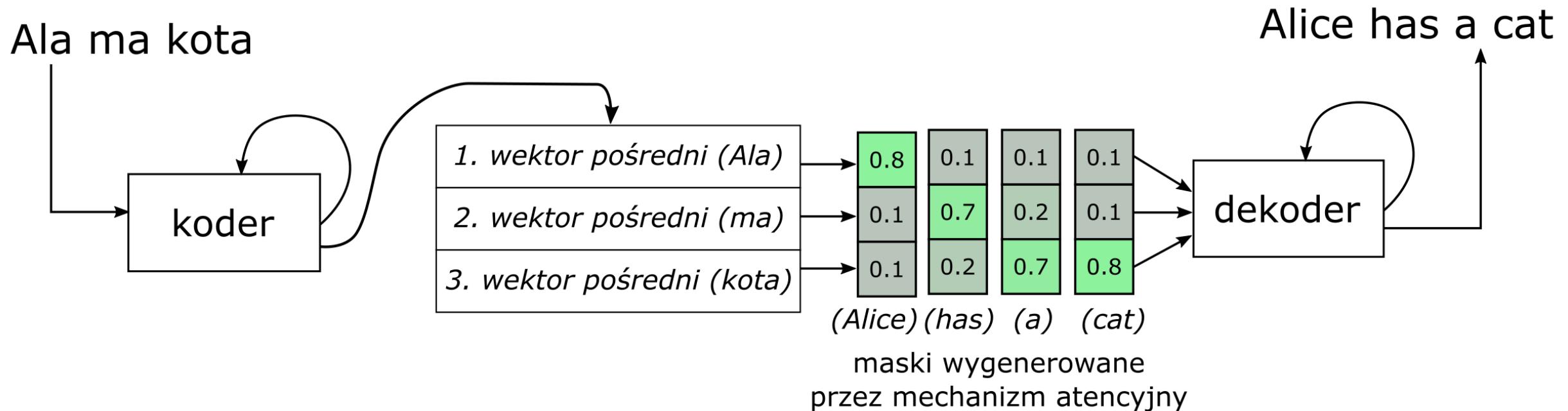
- **zwykłe sieci rekurencyjne** sprawdzają się do przetwarzania krótkich sekwencji, jednak niestety **nie są w stanie przetwarzać długich ciągów symboli** (mają problem z modelowaniem zależności pomiędzy odległymi znakami z sekwencji),
- pierwszą odpowiedzią na ten problem były sieci **LSTM** i nieco uproszczone względem LSTM sieci **GRU** – posiadają one specjalne **komórki pamięci, które nieco łagodzą problem** związany z modelowaniem długich sekwencji,

Sieci rekurencyjne i ich usprawnienia

Częstym wyborem do przetwarzania sekwencji są sieci posiadające jakiś rodzaj sprzężenia zwrotnego:

- dużym postępem w kwestii przetwarzania długich sekwencji znaków okazały się **sieci z mechanizmem atencyjnym** – radzą one sobie z tym problemem poprzez **przechowywanie wszystkich wyników pośrednich** kodera i sterowaniem dostępem do tych wyników za pomocą tzw. **funkcji atencji**,
- obecnie bardzo częstym i bardzo skutecznym sposobem przetwarzania sekwencji są tzw. **transformery**, o których będzie mowa w dalszej części prezentacji.

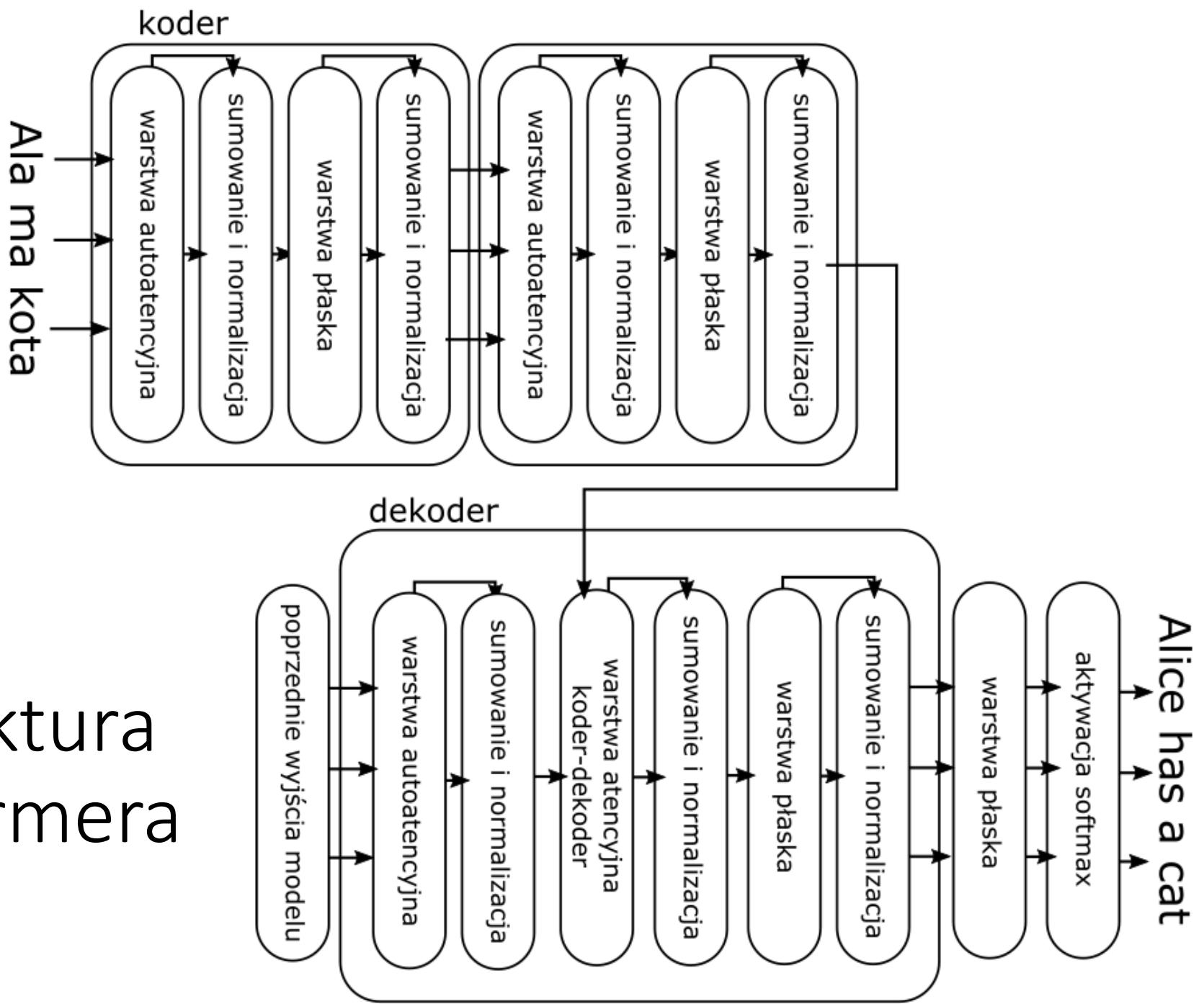
Sieci rekurencyjne z mechanizmem atencyjnym

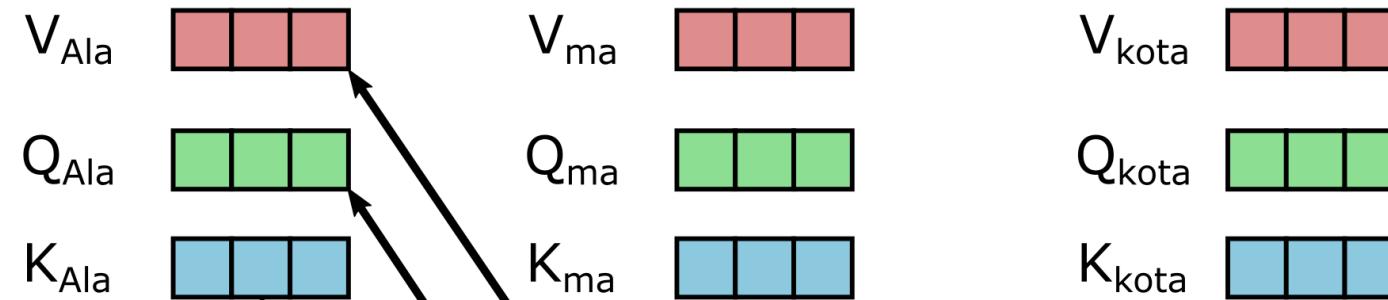
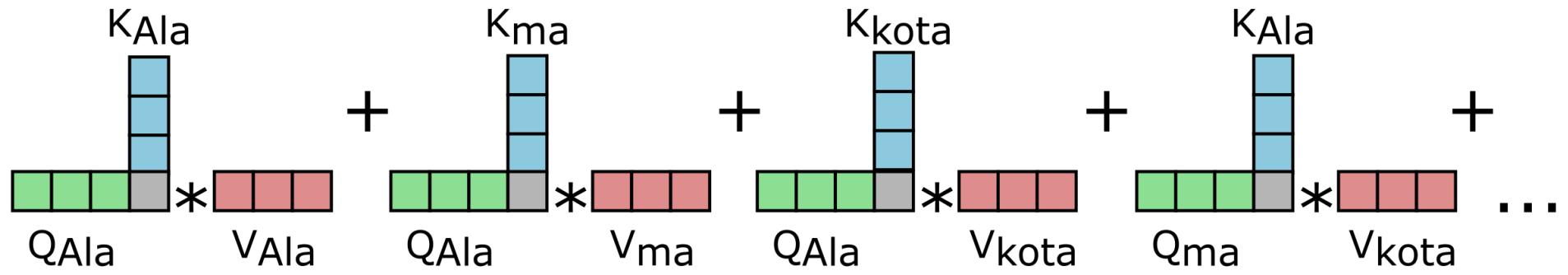


Transformery

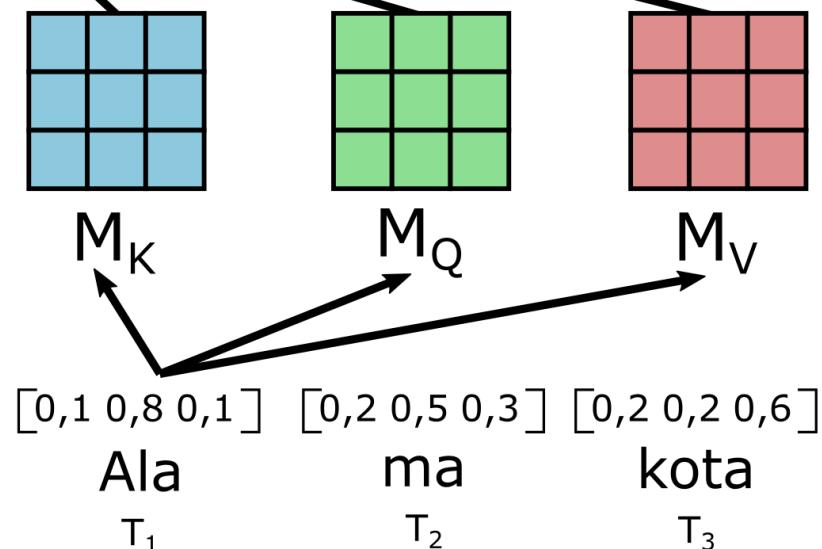
- Jest to architektura sieci neuronowej stanowiąca dalsze **rozszerzenie idei** sieci neuronowych z mechanizmem atencyjnym.
- Okazuje się, że możliwe jest stworzenie sieci neuronowej wykorzystującej prawie **w całości mechanizm atencyjny**.
- Okazało się, że jest to **wyjątkowo skuteczny** sposób na przetwarzanie sekwencji (w tym – języka naturalnego, mowy, muzyki, itp.).
- Jednym problemem w przypadku transformerów jest uwzględnienie kolejności tokenów. Rozwiązuje się to poprzez dodanie do reprezentacji tokenów specjalnych **kodów pozycyjnych**, które są unikalne dla każdego (1,2,3,... N-tego) tokenu.

Architektura transformera





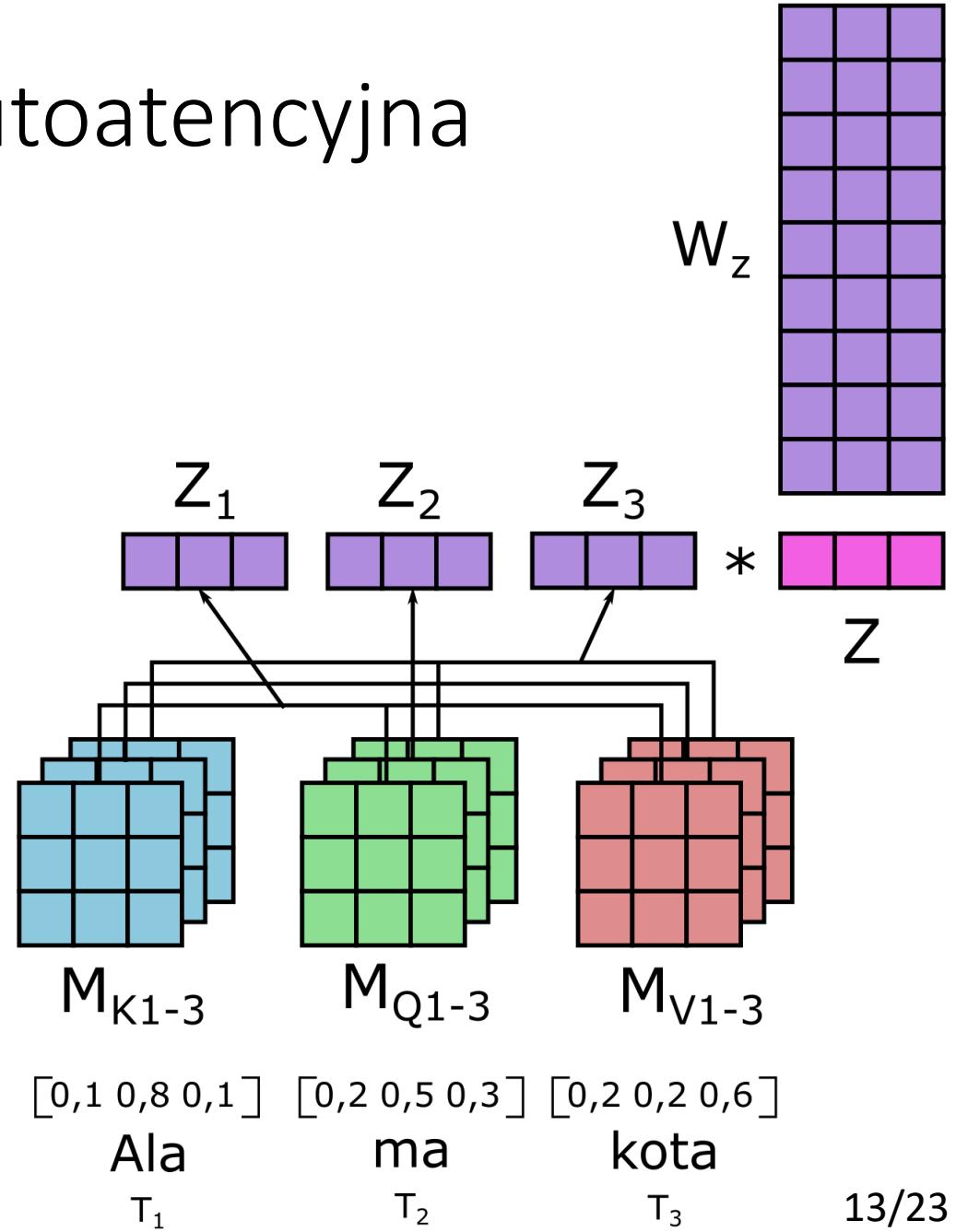
Warstwa autoatencyjna



Warstwa atencyjna koder-dekoder działa podobnie, ale klucze i wartości (K , V) są liczone na podstawie wyjścia kodera, a na podstawie wejścia z głębszych warstw dekodera liczone są tylko zapytania (Q)

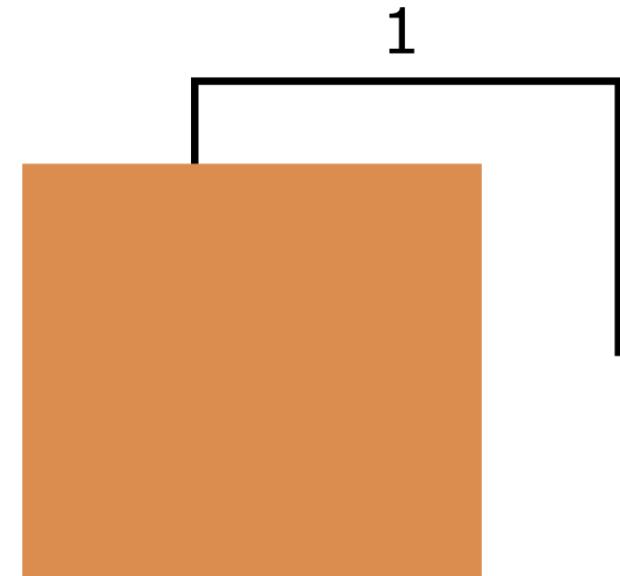
Wielowyjściowa warstwa autoatencyjna

- Warstwa autoatencyjna zwykle jest z wielokrotniona, co pozwala transformerowi uwzględniać różnego rodzaju konteksty.
- Efekty działania wszystkich warstw są następnie łączone poprzez przemnożenie przez macierz W_z .

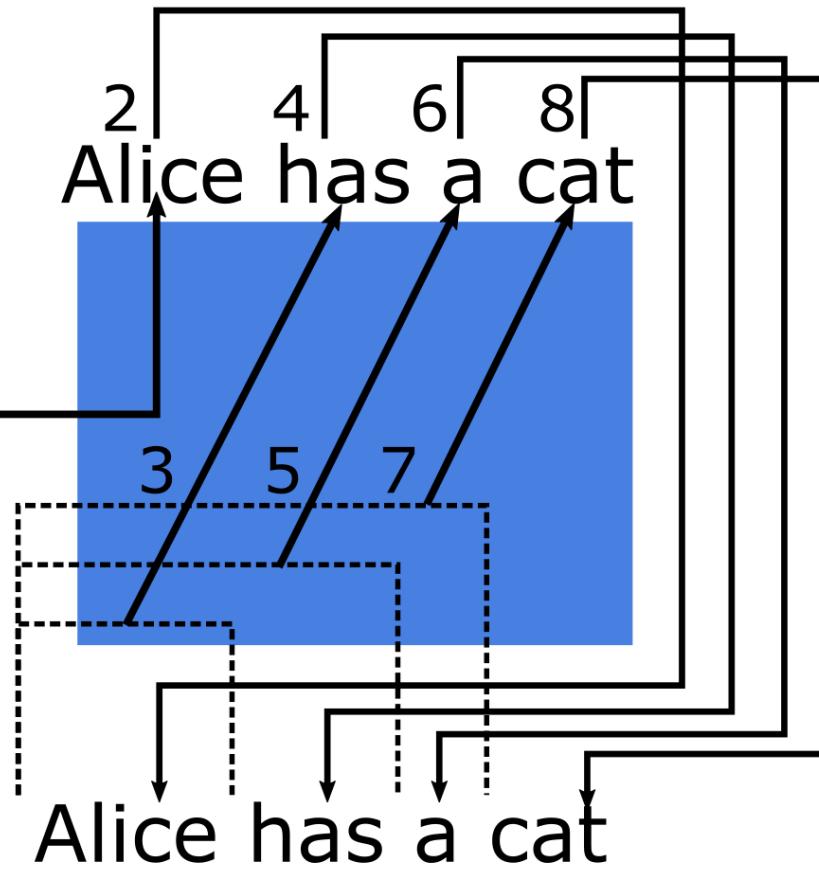


Przetwarzanie sekwencji przez transformer

koder



dekoder



Przykłady architektur przetwarzających język naturalny

Struktura transformera stała się podstawą dla zaprojektowania takich modeli sieci neuronowych jak:

- [Music Transformer](#) (projekt Magenta) – służący do przetwarzania muzyki,
- [Voice Transformer Network](#) – do zmiany barwy głosu ludzkiego,
- [BERT](#) – do przetwarzania języka naturalnego, opracowana przez Google,
- [GPT-1](#), [GPT-2](#), [GPT-3](#) – kolejne wersje sieci neuronowej do przetwarzania języka naturalnego opracowane przez OpenAI.

Architektura sieci GPT-2

W dalszej części wykładu skupimy się na przykładzie sieci GPT-2 (ang. *Generative Pre-trained Transformer, version 2*).

Jest to nietypowy transformer, gdyż **nie posiada on kodera**. Dekoder składa się z 12 warstw.

Główną cechą tej sieci jest fakt, że jest ona **wytrenowana na dużym korpusie tekstów anglojęzycznych** pochodzących z serwisu Reddit.

Wykorzystane zostały teksty posiadające przynajmniej trzy punkty karmy. Były one weryfikowane przez ludzkiego operatora.

Nienadzorowany pretrenining sieci GPT-2

Pierwszy etap treningu sieci GPT-2 miał charakter nienadzorowany.

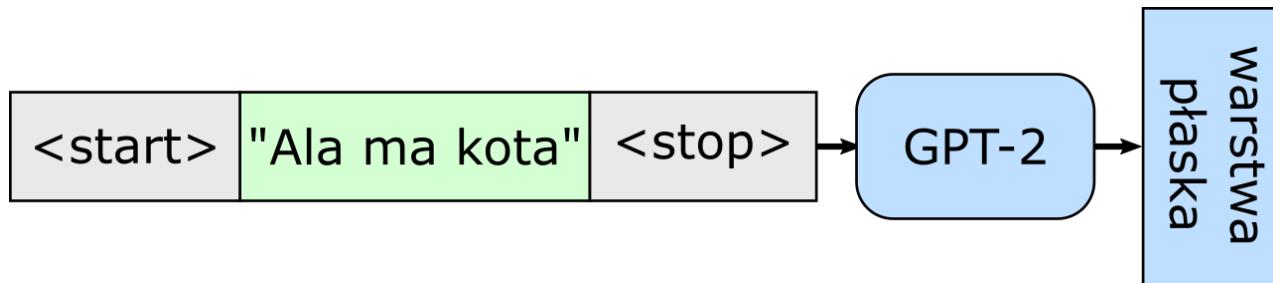
Jest to tzw. zadanie modelowania językowego, które polega na przewidywaniu kolejnego wyrazu w zdaniu przesłanym do sieci:

1. GPT-2(„Ala”) = „ma”
2. GPT-2(„Ala”, „ma”) = „kota”
3. GPT-2(„Ala”, „ma”, „kota”) = <koniec_zdania>

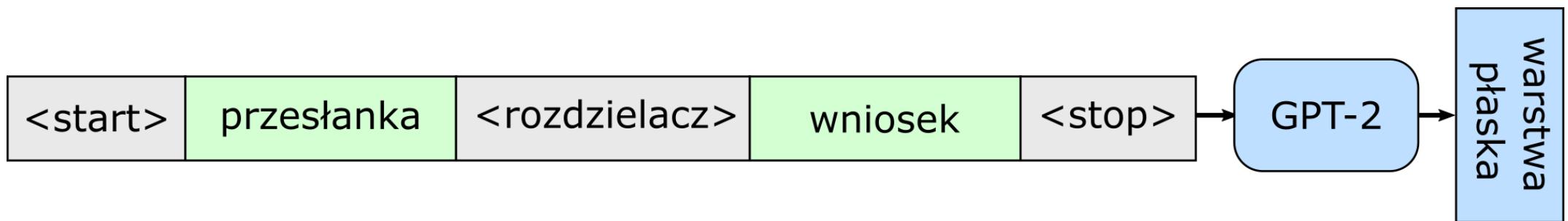
Sieć po tego typu treningu może stanowić bazę do realizacji innych zadań związanych z przetwarzaniem języka. Jest to drugi etap jej treningu, który jest zależny od konkretnego przeznaczenia sieci.

Zastosowania GPT-2: klasyfikacja, wnioskowanie

Dla klasyfikacji sieć GPT-2 ma prostą strukturę:



Dla wnioskowania należy uwzględnić dodatkowy token rozdzielający:

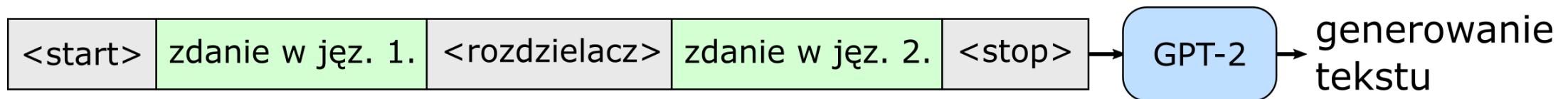


Zastosowania GPT-2: tłumaczenie maszynowe

Tłumaczenie maszynowe wymaga struktury podobnej, jak wnioskowanie, jednak na końcu sieci nie wykorzystujemy warstwy płaskiej, a pozwalamy modelowi generować tekst.

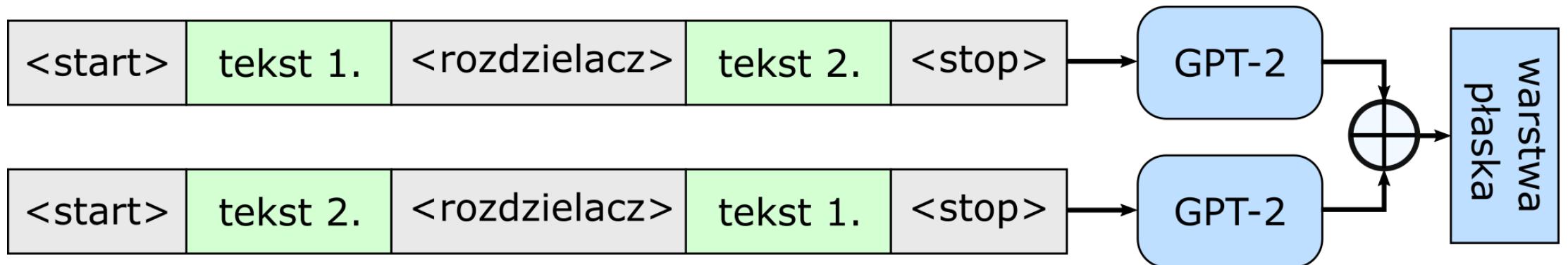
W treningu podajemy strukturę jak na obrazku

Wykorzystując sieć do tłumaczenia podajemy jedynie tokeny <start>, zdanie w języku 1. i <rozdzielacz>. Następnie pozwalamy sieci wygenerować resztę struktury, co skutkuje przetłumaczeniem zdania na język 2..



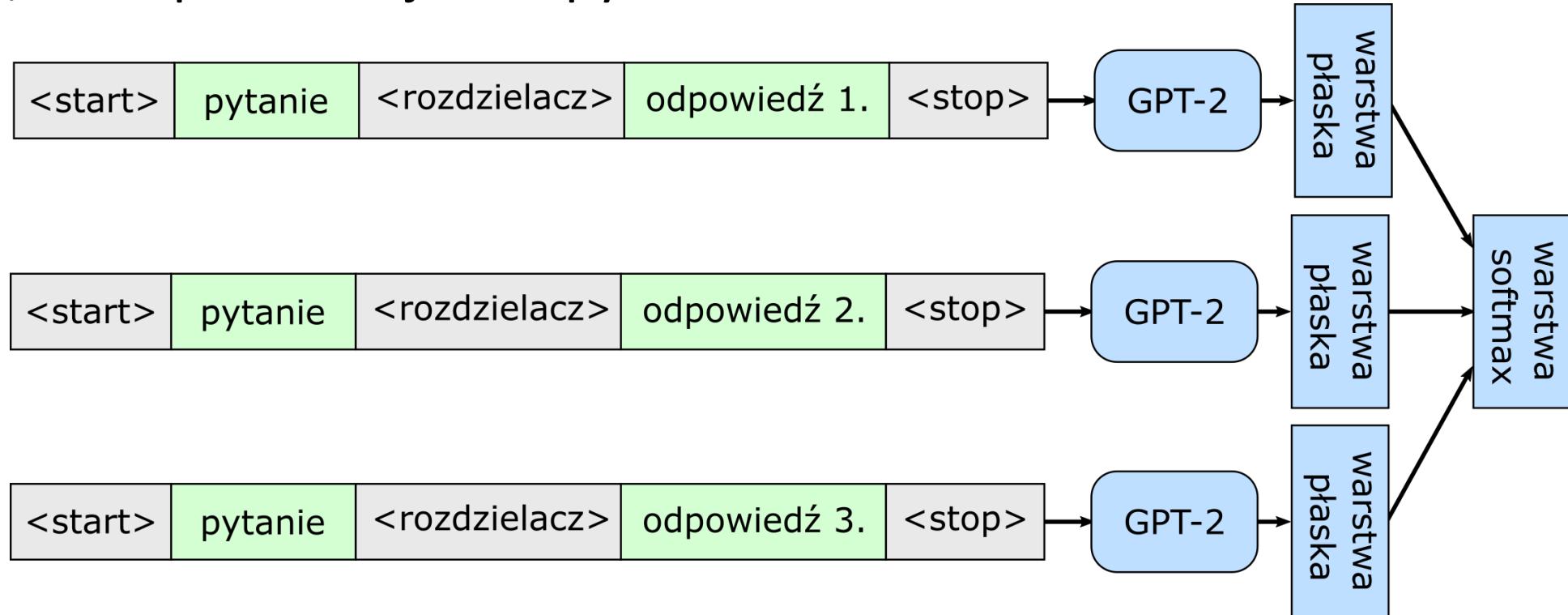
Zastosowania GPT-2: podobieństwo tekstów

Analiza podobieństwa tekstów wymaga podwójnego uruchomienia sieci w celu parametryzacji obu tekstów.



Zastosowania GPT-2: pytania wielokrotnego wyboru

Najbardziej skomplikowana jest struktura sieci odpowiadającej na pytanie wielokrotnego wyboru. Sieć GPT-2 musi być wykorzystana tyle razy, ile odpowiedzi jest w pytaniu:



Literatura

1. Goodfellow, I., Bengio, J., Courville, Aaron, Deep Learning, The MIT Press, 2016.
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Łukasz, Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in neural information processing systems, 2017, pp. 5998–6008.
3. Goodfellow, I., Bengio, Y., Courville, A. Deep Learning, MIT press, 2016,
4. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018.
5. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. Language Models Are Unsupervised Multitask Learners., 2019.
6. <https://jalammar.github.io/illustrated-transformer/> - ilustrowany przewodnik opisujący po kolei wszystkie bloki sieci typu transformer
7. <https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314> - opis algorytmów CBOW oraz Skip-gram
8. <https://towardsdatascience.com/byte-pair-encoding-the-dark-horse-of-modern-nlp-eb36c7df4f10> - opis algorytmu BPE

Dziękuję za uwagę



Fundusze
Europejskie
Polska Cyfrowa



Rzeczpospolita
Polska



Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

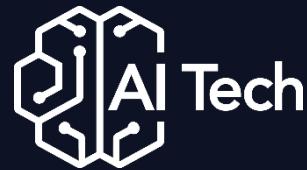
Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.



POLITECHNIKA
GDAŃSKA



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Głębokie Przetwarzanie Tekstu i Mowy

Użycie syntezatorów mowy opartych o głębokie uczenie typu Wavenet i Tacotron-2 do syntezy mowy

Szymon Zaporowski, Adam Kurowski



Rzeczpospolita
Polska



Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

Plan wykładu

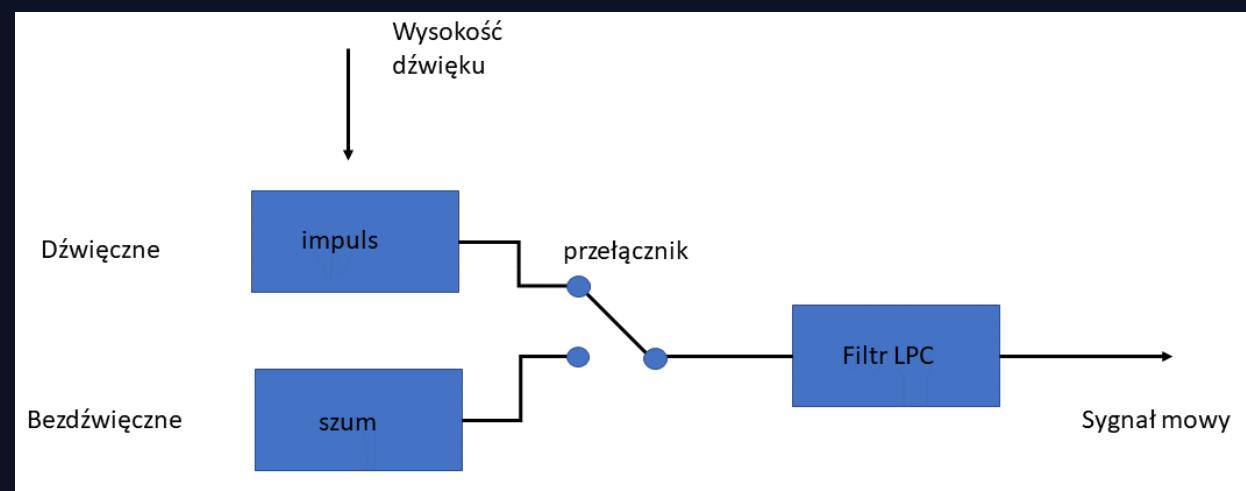
- Krótki wstęp do syntezy mowy
- Czym jest prozodia i dlaczego jest ważna?
- Czym jest WaveNet?
- Jak powstał WaveNet
- PixelRNN i PixelCNN a WaveNet
- U-law – do czego służy?
- Jak jest wygenerować próbke w WaveNet
- Jak zbudowany jest WaveNet?
- Ulepszenia sieci WaveNet

Synteza mowy – generowanie sygnału akustycznego, którego brzmienie naśladuje brzmienie ludzkiej mowy

Obecnie można wskazać 4 podstawowe podejścia do syntezy sygnału mowy:

- Odwzorowanie widma sygnału mowy – metoda formantowa, synteza LPC;
- Fizyczne odwzorowanie mechanizmów wytwarzania mowy – metoda artykulacyjna;
- Wykorzystanie nagranych próbek sygnału mowy – metoda konkatenacyjna
- Synteza z wykorzystaniem uczenia głębokiego – sieci typu WaveNet, Tacotron, Tacotron 2

- Synteza wykorzystuje liniowe kodowanie predykcyjne (ang. LPC – linear predictive coding) do odwzorowania charakterystyki przenoszenia traktu głosowego. Metoda LPC umożliwia podzielenie sygnału mowy na pobudzenie i transmitancję traktu głosowego, modelowaną przez filtr biegunkowy (all-pole filter).



- Synteza artykulacyjna – jest to próba fizycznego odwzorowania mechanizmu generowania mowy. Poprzez użycie modelowania matematycznego, uwzględniane są zjawiska, które mają miejsce podczas przenoszenia dźwięku przez trakt głosowy. Charakter generowanego sygnału jest zmienny w zależności od parametrów, m.in wymiarów i ustawienia poszczególnych organów mowy. Metoda jest w założeniu wierniejsza od formantowej jednak zdecydowanie bardziej skomplikowana
- Konieczne jest uzyskanie modelu geometrii traktu głosowego oraz pozyskanie parametrów na drodze analizy przekroju traktu głosowego lub rezonansu magentycznego

Krótki wstęp do syntezy mowy – synteza konkatenacyjna

- Synteza konkatenacyjna – polega na łączeniu (czyli konkatenacji) wypowiedzi z nagranych fragmentów głosu zawierających słowa, sylaby lub np.. złączenia głosek. Była to do czasu wprowadzenia syntezy z wykorzystaniem sieci neuronowych najczęściej wykorzystywana metoda syntezy mowy.
- Charakteryzuje się wysoką zrozumiałość i naturalność brzmienia.
- Dużą wadą jest konieczność zebranie bazy segmentów obejmujących cały system fonetyczny języka w przypadku poprawnego działania konkatenacyjnego systemu TTS

Krótki wstęp do syntezy mowy – synteza korpusowa

- Synteza korpusowa jest to specyficzny wariant syntezy konkatenacyjnej.
- W bazie przechowuje się segmenty o różnej długości (np. temat i końcówka słowa).
- Do łączenia wypowiedzi wybiera się możliwie najdłuższe segmenty. Dzięki temu możliwe jest uzyskanie bardzo wysokiej jakości dla często występujących w języku słów.

Czym jest prozodia i dlaczego jest ważna?

Prozodia to inaczej brzmieniowe właściwości mowy.

Składają się na nią:

- Akcent
- Intonacja (występuje zwłaszcza przy wyrażaniu emocji)
- Iloczas (różnicowanie czasu trwania sylab i głosek – ma to wpływ na znaczenie wyrazów w niektórych językach, w j. polskim może mieć wpływ na ekspresję wypowiedzi)

Prozodia jest istotna ze względu na naturalność mowy – jest to problem syntezatorów mowy, aby brzmieć naturalnie

Czym jest WaveNet?

- WaveNet jest architekturą generatywnej sieci neuronowej pozwalającej na generowanie audio w postaci tzw. „surowych” próbek – waveform-ów
- Sieć bazuje na probabilistyczce i autoregresji – każda kolejna wartość próbki bazuje na poprzednich

Zastosowanie sieci WaveNet:

- Generowanie mowy
- Generowanie muzyki
- Text-to-Speech
- Rozpoznawanie mowy

Jak powstał WaveNet

- DeepMind pracował nad problemami z sekwencjonowaniem dźwięku, szczególnie związanymi z uczeniem i emulacją ludzkiego języka. Uporanie się z tym problemem wydawało się być niezbędne, aby pozbyć się „głosu robotów”, a cyfrowi asystenci posiadali jak najbardziej zbliżone do ludzkich głosy.
- Jednym z obszarów, w których szukali inspiracji, było Przetwarzanie Obrazu i architektura Sieci Splotowych - idea filtrowania i splotów sprawdza się zarówno w przypadku obrazów i wideo, jak i aplikacji audio. Można założyć, że praca DeepMind umożliwiła korzystanie z zaawansowanych Asystentów głosowych np. Asystenta Google'a.

PixelRNN i PixelCNN a WaveNet

WaveNet bazuje na zasadzie przedstawionej w sieciach PixelRNN i PixelCNN – jednak przy zmienionej architekturze – wszystkie wymienione sieci są generatywnymi modelami autoregresyjnymi .

WaveNet nie jest siecią rekurencyjną, ale wykorzystuje **warstwy splotowe** skonstruowane w specyficzny sposób bazując na rozwiązaniach przedstawionych w PixelCNN

Zamiast stosować maski w splotach stosowane są tzw. swobodne sploty. Można je skonstruować poprzez maskowanie tensora danych i dokonanie mnożenia tej maski z jądrem splotowym przed jego zaaplikowaniem.

μ -Law i kwantyzacja

W potoku przetwarzania sieci WaveNet pierwszym krokiem jest kwantyzacja danych z użyciem μ -Law. Przykładowo „surowe” dane audio składają się z sekwencji 16 bitowych wartości stałoprzecinkowych (dla jednego kroku czasowego) – to oznacza 65536 wartości, które musiałby zostać przetworzone w jednym roku czasowym. Z tego względu stosuje się algorytm u-law będący algorytmem kompandorowym – zmieniającym dynamikę sygnału.

Wzór na kodowanie u-law:

$$f(x_t) = sign(x_t) \frac{\ln(1 + \mu|x_t|)}{\ln(1 + \mu)}$$

gdzie $-1 < x_t < 1$ oraz $\mu = 255$

Jak wygenerować próbę audio w Wavenet?

Łączne prawdopodobieństwo waveform-u $x = \{x_1, \dots, x_T\}$ można rozumieć jako produkt prawdopodobieństw warunkowych przedstawionych poniższym wzorem:

$$p(x) = \prod_{t=1}^T p(x_t | x_1, \dots, x_{t-1})$$

Równanie opisuje sposób generowanie nowych próbek poprzez przewidywanie prawdopodobieństwa następnych próbek bazując na prawdopodobieństwach próbek poprzednich i bieżących

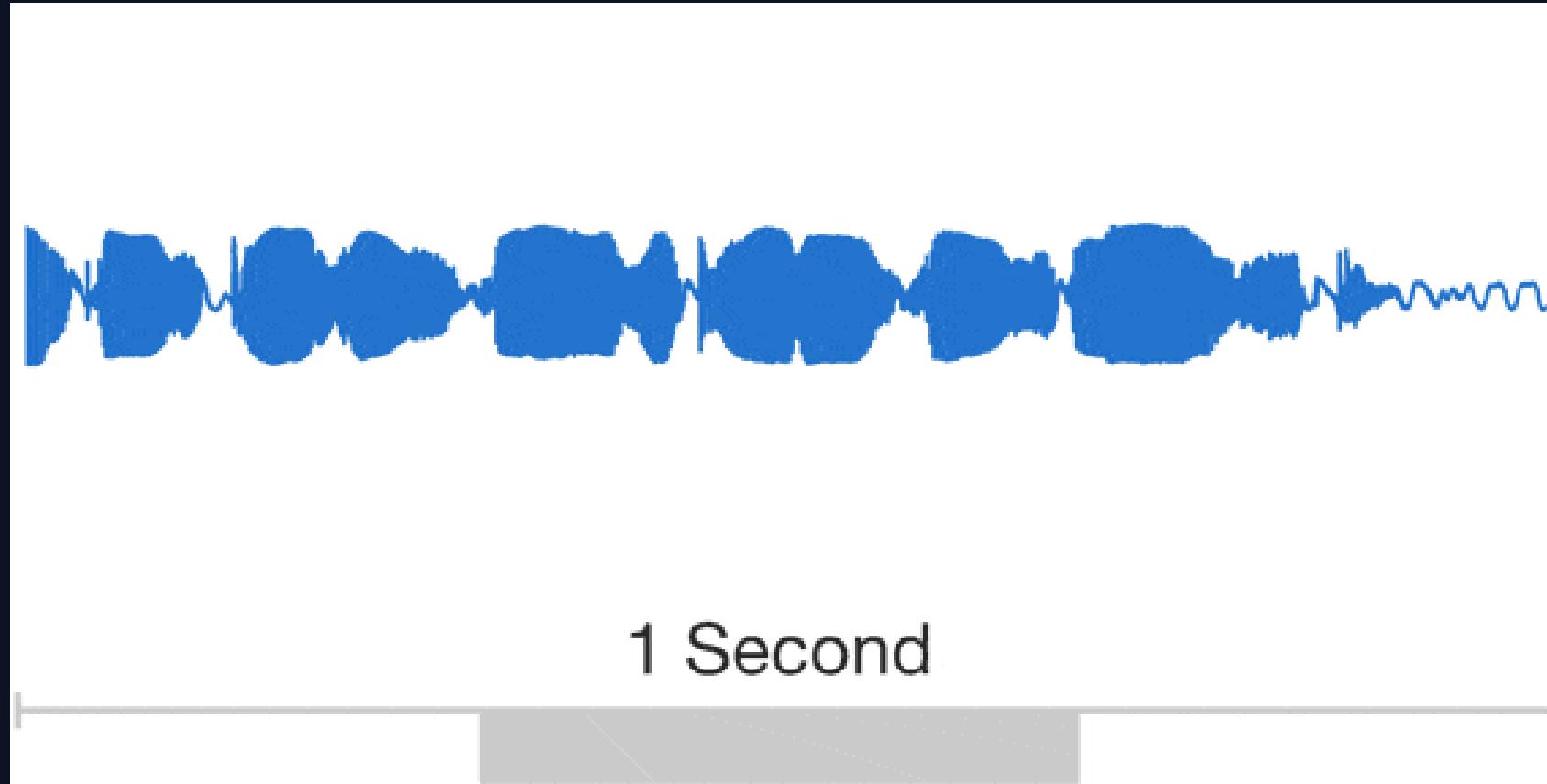
Każda próbka x_t jest uwarunkowana na próbkach we wszystkich poprzednich krokach czasowych

Jak wygenerować próbę audio w Wavenet?

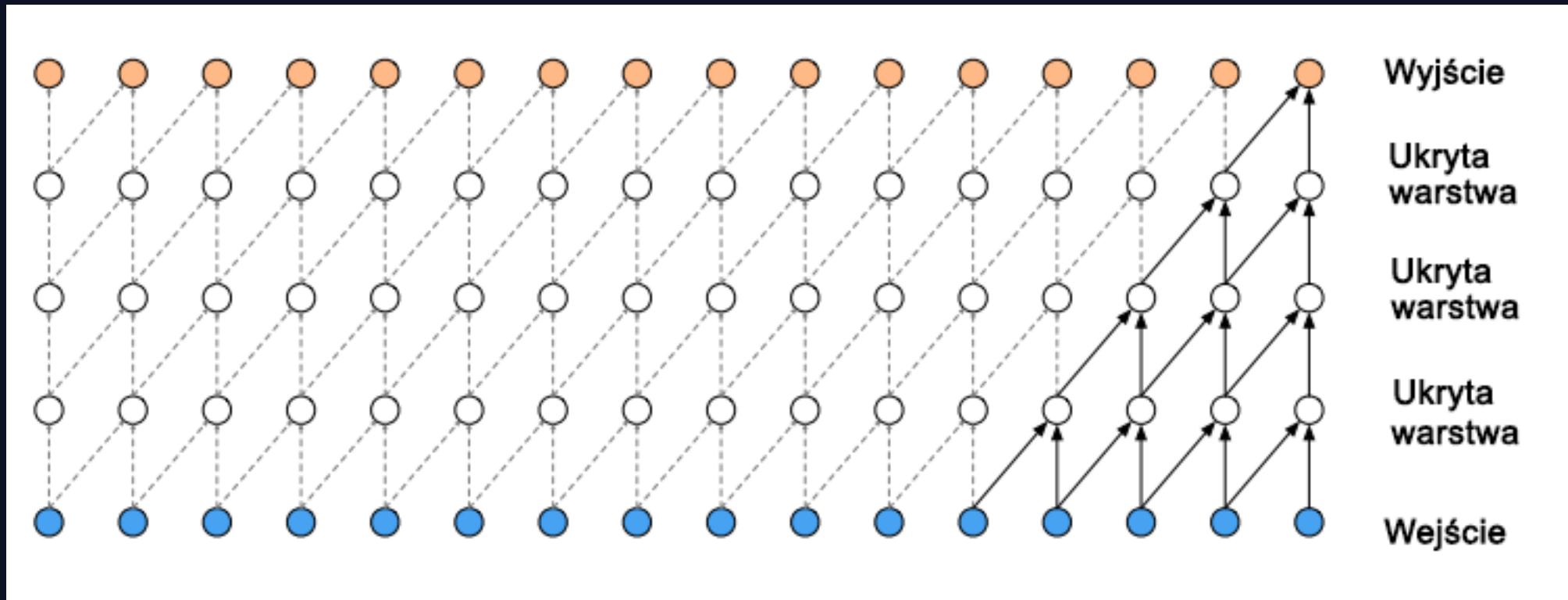
Dystrybucja prawdopodobieństwa warunkowego (czyli wzór pokazany na poprzednim slajdzie) jest modelowany z wykorzystaniem stosu warstw splotowych

Wykorzystując sploty swobodne zagwarantowane jest, aby model nie naruszał kolejności modelowania danych – predykcja $p(x_{t+1}|x_1, \dots, x_t)$ generowana przez model w korku czasowym t nie może zależeć od żadnego z następnych kroków czasowych $x_{t+1}, x_{t+2} \dots, x_T$

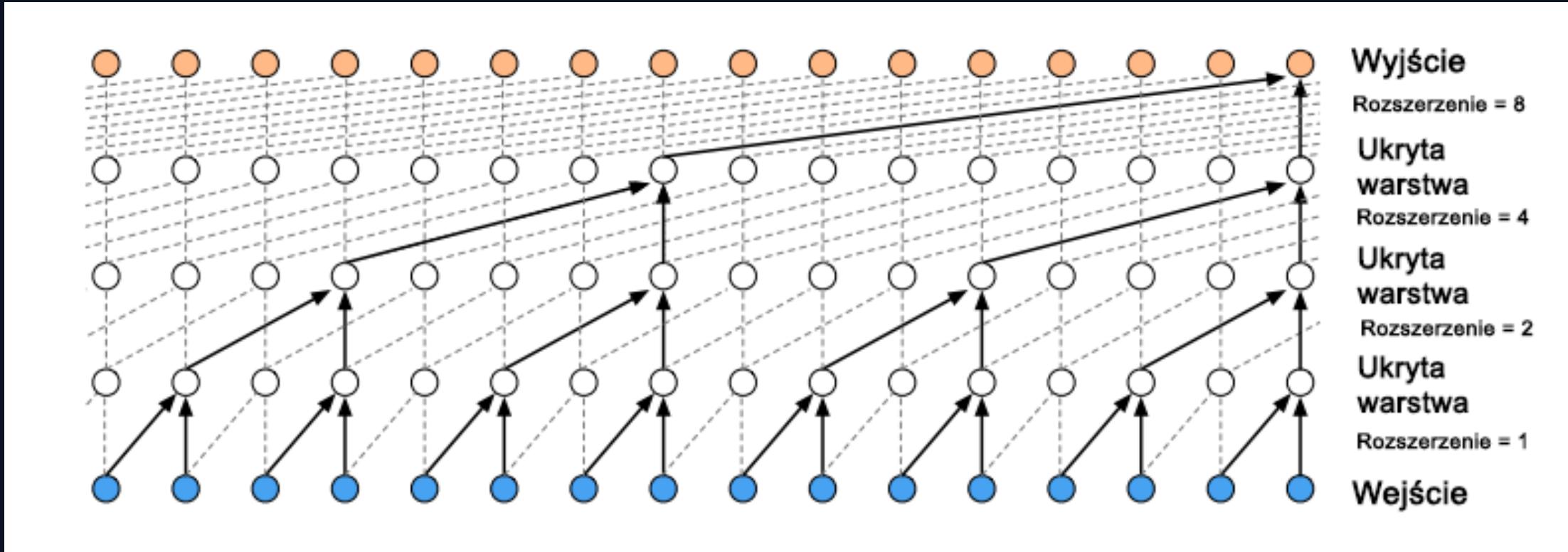
Wavenet – ile próbek musi przetworzyć sieć w 1 sekundzie



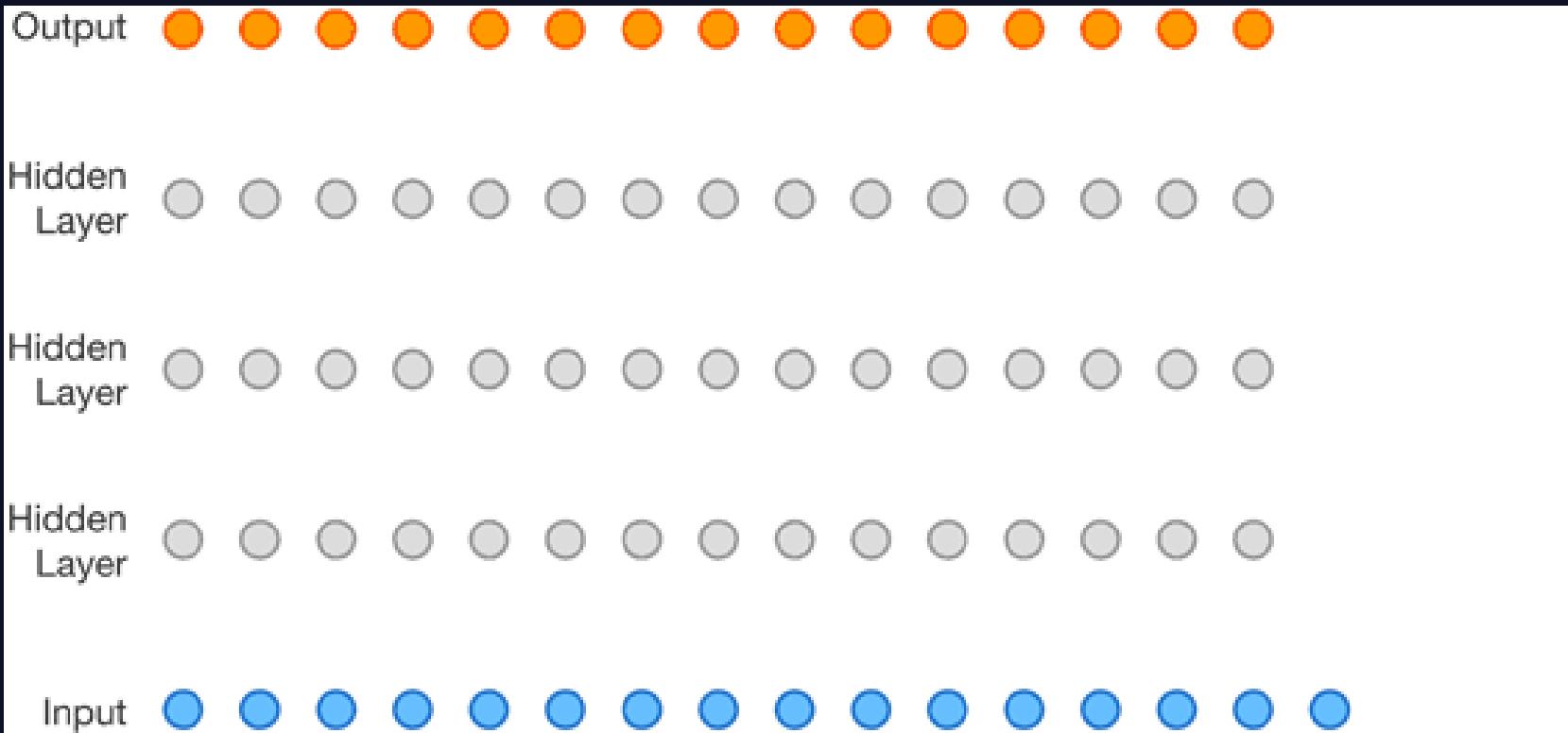
Warstwy splotowe – sploty swobodne



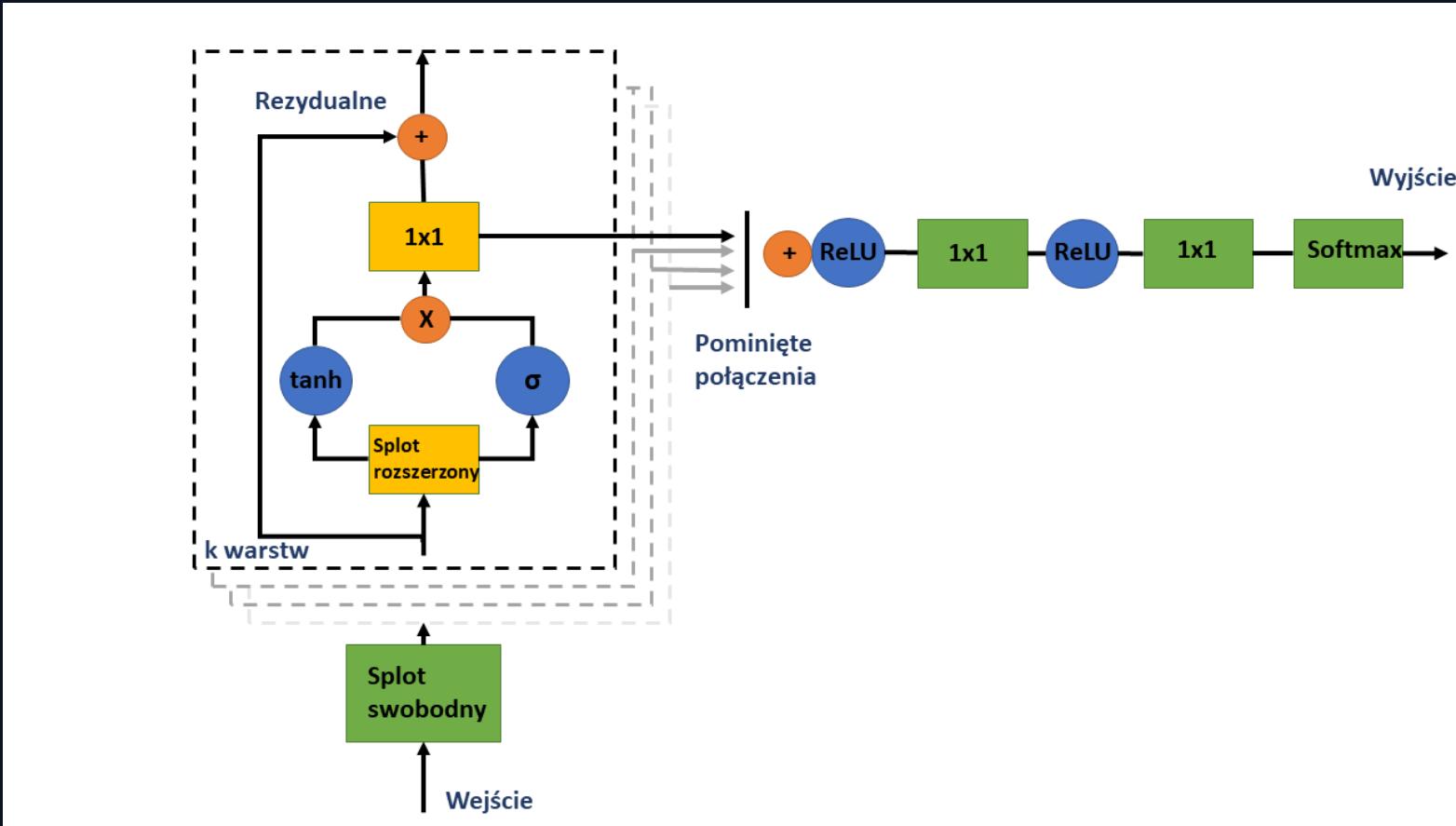
Warstwy splotowe – rozszerzone sploty swobodne



WaveNet- schemat generowania próbek



WaveNet – konstrukcja oryginalnej sieci - połączenia



Warunkowe generowanie w WaveNet

Dodając dodatkowe wejście h , WaveNet może modelować warunkową dystrybucję $p(x|h)$ wejścia audio.

Wtedy wzór na prawdopodobieństwo warunkowe przyjmuje postać:

$$p(x|h) = \prod_{t=1}^T p(x_t|x_1, \dots, x_{t-1}, h)$$

Wykorzystując warunkowe modelowanie można wybrać tożsamość mówcy w przypadku treningu na bazie wielu mówców podając tożsamość mówcy jako dodatkowe wejście

W przypadku TTS można podawać informacje o tekście jako dodatkowe wejście

Mögliwe jest warunkowanie lokalne i globalne.

Ulepszenia sieci WaveNet

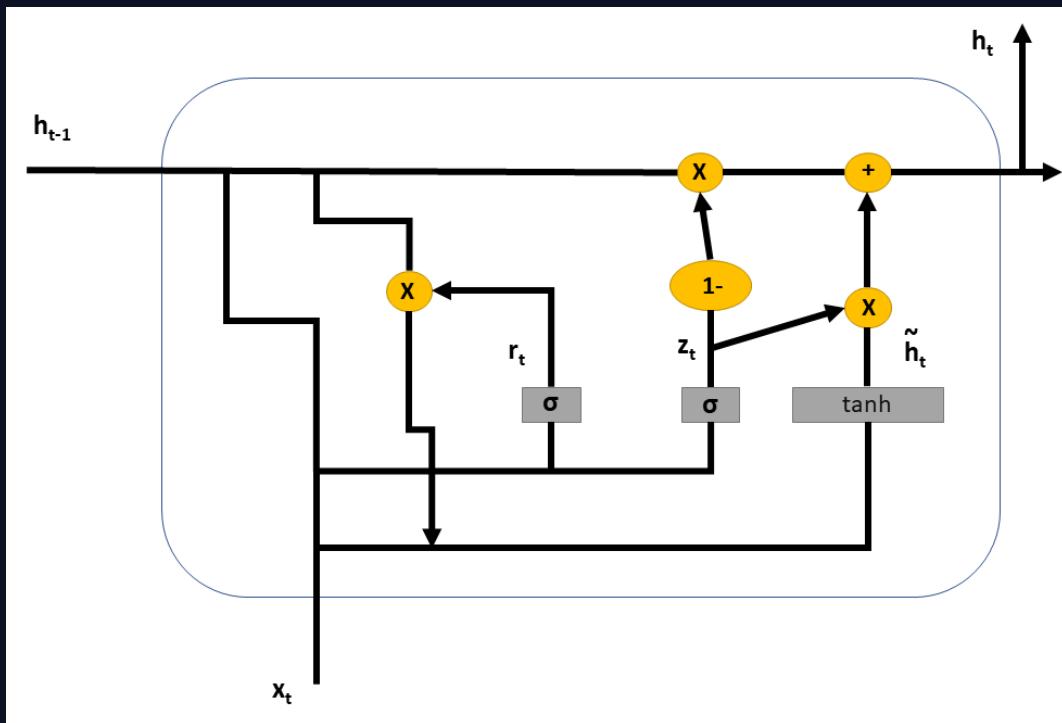
W 2017 roku aby wygenerować 1 sekundę audio z użyciem sieci WaveNet potrzeba było ok. 50 sekund – działanie w czasie rzeczywistym nie było możliwe

Dzięki wykorzystaniu destylacji wiedzy udało się stworzyć model typu feed-forward (oryginał był siecią autoregresyjną), na wejście którego podawany jest szum w celu wygenerowania dźwięku.

Taki model jest w stanie wygenerować 20 sekund audio w wyższej jakości niż oryginalna sieć w 1 sekundę

Ulepszenia sieci WaveNet

- W 2018 roku architekturę WaveNet związaną ze swobodnymi splotami zastąpiono komórkami GRU znymi z sieci rekurencyjnych
- Od roku 2018 DeepMind nie podało więcej informacji jakie usprawnienia zostały dodane do sieci WaveNet



Struktura komórki RNN GRU

Źródła

- <https://deepmind.com/blog/article/wavenet-generative-model-raw-audio>
- strona sieci WaveNet
- <https://www.youtube.com/watch?v=gIIZyYsB81M> – prezentacja modelu WaveNet w ramach festiwalu Electromagnetic Field 2018
- Zaporowski S.: Wykorzystanie sieci neuronowych do syntezy mowy wyrażającej emocje, Postępy badań w inżynierii dźwięku i obrazu Nowe trendy i zastosowania technologii multimedialnych : , 2019, s.71-98
- Zaporowski S., Blaszke M., Kurowski A.: Zastosowanie sieci neuronowych w cyfrowej syntezie dźwięku, Studium badawcze młodych akustyków 2017 : monografia Katedry Mechaniki i Wibroakustyki AGH, ed. Jarosław Rubacha Kraków: Katedra Mechaniki i Wibroakustyki AGH, 2018, s.203-220



POLITECHNIKA
GDAŃSKA



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Dziękuję

Szymon Zaporowski



Fundusze
Europejskie
Polska Cyfrowa



Rzeczpospolita
Polska



Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.



POLITECHNIKA
GDAŃSKA



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Głębokie Przetwarzanie Tekstu i Mowy Użycie syntezatorów mowy opartych o głębokie uczenie typu Wavenet i Tacotron-2 do syntezy mowy

Szymon Zaporowski



Rzeczpospolita
Polska



Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

Plan wykładu

- Jak oceniać jakość syntezy mowy?
- Przegląd wybranych systemów TTS i ich elementów bazujących na głębokich sieciach neuronowych
- Budowa sieci Tacotron
- Czym różni się Tacotron-2 od Tacotrona?
- Zastosowania sieci Tacotron-2 i pochodnych

Jak oceniać jakość syntezy mowy?

Miara MOS (ang. Mean Opinion Score) jest najpowszechniejszą miarą do oceny jakości wygenerowanej mowy. Zakres oceny MOS to 0-5, gdzie mowa „generowana” przez człowieka posiada oceną **4,5-4,8**

MOS wywodzi się z telekomunikacji i jest definiowana jako średnia arytmetyczna z pojedynczych ocen otrzymywanych przez ludzi dla zadanego bodźca w odsłuchowym teście subiektywnym.

Miarę MOS można w najbardziej naiwnym podejściu rozumieć jako średnią wartość oceny ludzi, którzy odsłuchali dane nagranie.

Obecnie miarę MOS oblicza się automatycznie z wykorzystaniem tzw. Objective Quality Models

Przegląd systemów TTS bazujących na głębokich sieciach neuronowych

Rodziny systemów TTS bazujących na uczeniu głębokim:

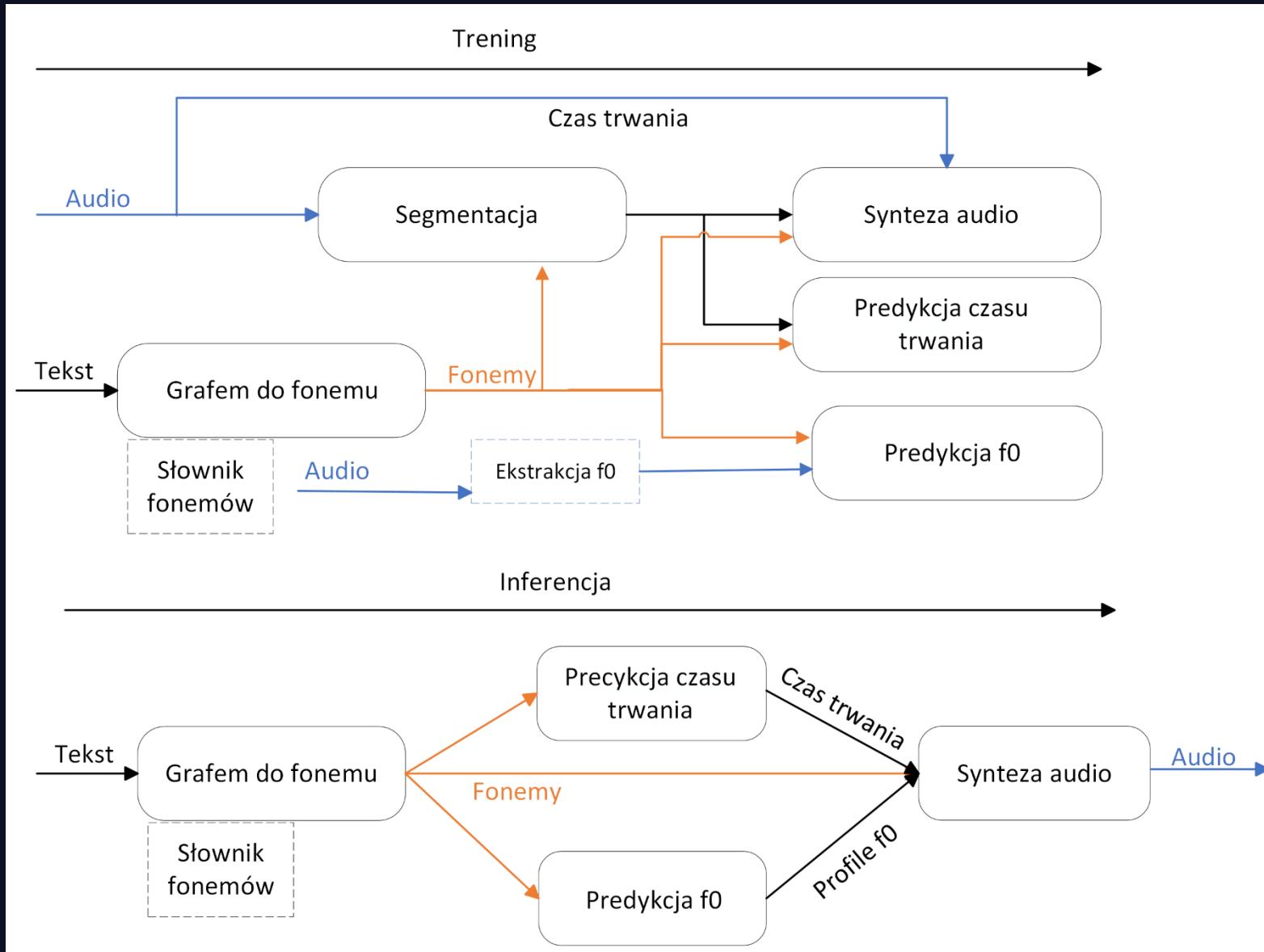
- WaveNet – WaveNet, FastWaveNet, ParallelWaveNet
- DeepVoice – DeepVoice, DeepVoice 2, Deep Voice 3
- Tacotron – Tacotron, Tacotron-2
- WaveGlow
- FastSpeech
- EATS

DeepVoice

Oryginalne rozwiązanie DeepVoice składa się z 4 różnych sieci neuronowych, które tworzą razem potok przetwarzania:

- Modelu segmentacji, którego zadaniem jest lokalizacja granic między fonemami – hybrydowa sieć typu CNN i RNN, która została wytrenowana do przewidywania powiązania między dźwiękiem głosu i docelowymi fonemami wykorzystując funkcję straty CTC
- Model, który konwertuje grafemy do fonemów. Zastosowano model wielowarstwowy enkoder-dekoder z komórkami GRU
- Model do przewidywania czasu trwania fonemów i częstotliwości podstawowej (F0- ton krtaniowy). Wykorzystane są dwie w pełni połączone warstwy, a następnie dwie jednokierunkowe warstwy GRU i całkowicie połączona warstwa
- Model do syntezy audio. Wykorzystano zmodyfikowany WaveNet
- Wynik MOS: 2.67

DeepVoice



CTC

- Obliczana jest strata między ciągłą serią czasową (niesegmentowaną) i docelową sekwencją. Wykonywane to jest poprzez sumowanie prawdopodobieństwa możliwych dopasować sekwencji wejściowej, wytwarzającą wartość straty, która różni się w odniesieniu do każdego węzła wejściowego.
- Zakłada się, że wyrównanie wejścia jest typu „wiele do jednego”, co ogranicza długość sekwencji docelowej w taki sposób, że musi być odpasowana do długości wejścia.

DeepVoice 2

Następca architektury Deep Voice

- Poprawa względem oryginału (wynik MOS: 3,53)
- Potok przetwarzania pozostał bez większych zmian
- Każdy model w potoku stworzony od podstaw – poprawa jakości
- Wsparcie dla podejścia typu multi-speaker

DeepVoice 2

Główne cechy poprawionej architektury:

- Rozdzielenie modeli odpowiedzialnych za czas trwania fonemów i częstotliwość podstawową
- Do każdego modelu wprowadzono embeddingi pochodzące od mówców, aby uzyskać możliwość działania typu multi-speaker
- Embeddingi mówców zawierają unikalne informacje dla każdego mówcy i są wykorzystywane do generowania stanów początkowy w sieciach RNN, nielionowych bias-ów i współczynników mnożnikowych.
- Oprócz tego dodano połączenia rezydualne i normalizacje batchy

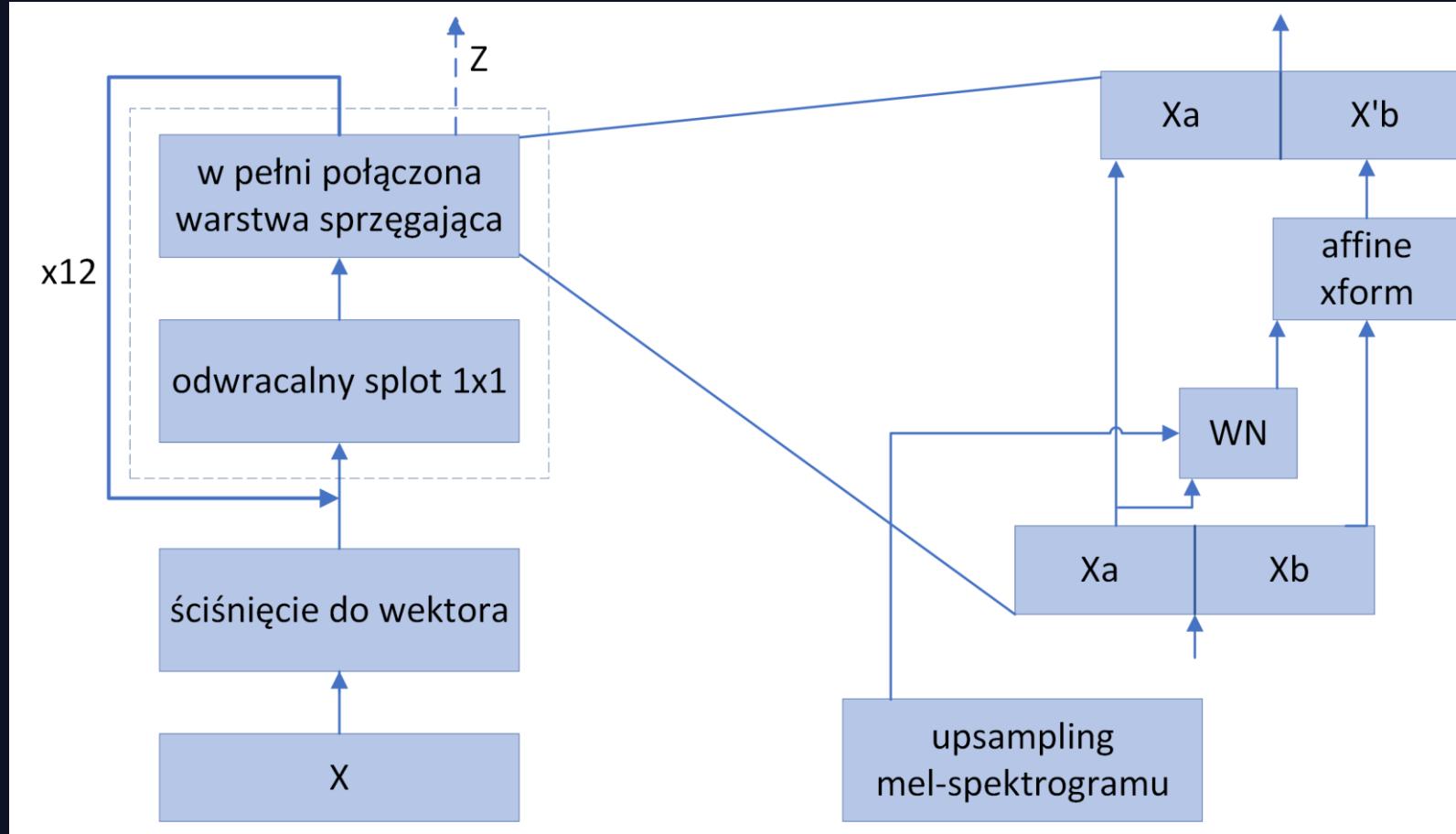
Deep Voice 3

- Zmiana architektury – zamiast 4 osobnych modeli w potoku przetwarzania zaproponowano jeden obszerny model
- Model splotowy znak – spektrogram (ang. character-to-spectrogram), idealny do obliczeń równoległych
- Zastosowano mechanizm atencyjny
- Jako przejście ze spektrogramu do audio wykorzystano WaveNet
- Wynik MOS: 3,78

WaveGlow

- Próba połączenia cech z architektury typu GLOW i WaveNet
- Cel to szybka i wydajna syntezata dźwięku bez korzystania z metod auto-regresyjnych
- WaveGlow jest wykorzystywany do generowania dźwięku ze spektrogramów zamiast sieci WaveNet – nie jest stricte systemem TTS end-to-end
- Model jest trenowany poprzez minimalizowanie negatywnej funkcji wiarygodności (ang. log-likelihood function) danych.
- Stosowane są sieci typu INN (ang. Invertible Neural Networks)
- Sieci INN są zazwyczaj tworzone z użyciem warstw sprzęgających (ang. coupling layers) – w WaveGlow użyto warstw affine
- Po wytrenowaniu modelu inferencja polega na losowym próbkowaniu wartości i przetwarzaniu ich przez sieć

WaveGlow



TTS z użyciem Transformerów

Wykorzystanie Transformerów – możliwość równoległego treningu

Budowa:

- Konwerter tekst-fonem
- Skalowalne enkodowanie pozycyjne – użycie sinusoidalnej formy do przechwytywania informacji o pozycji fonemów
- Enkoder typ Pre-Net – składa się z 3 warstw CNN – uczy się embeddingów fonemów
- Dekoder typu Pre-Net – przetwarza melspektrogram i osadza go w tej samej podprzestrzeni w której są embeddingi fonemów
- Właściwy enkoder – enkoder transformatorowy z atencją typu multi-head
- Właściwy dekoder - dekoder transformatorowy z atencją typu multi-head
- Rzut melowy i rzut liniowy – dwa różne rzuty (ang. linear projections) wykorzystywane do przewidywania mel spektrogramu i tokena <stop>

Fast Speech

FastSpeech uzyskało przyspieszenie architektury transferowej 38 razy

Zmiany względem poprzedniego podejścia:

- Równoległa generacja melspektrogramu
- Zmiana w podejściu do znajdowania dopasowań (ang. alignment) między fonemami i ich odwzorowaniem melspektrogramowym – twarde dopasowanie zamiast miękkiego
- Regulacja długości fonemów pozwalająca na dopasowywanie prędkości głosu poprzez zmianę długości generowanego spektrogramu
- Wynik MOS: 4.39

EATS - End-to-end Adversarial Text-to-Speech

Opracowany przez DeepMind – publikacja na konferencji ICLR 2021

- EATS wykorzystuje trening adwersarialny wykorzystywany głównie w sieciach typu GAN
- Sieć działa na surowym tekście lub surowych sekwencjach fonemów i generuje surowe waveformy
- EATS składa się z dwóch podstawowych podmodułów – alignera i dekodera
- Aligner otrzymuje surowe dane na wejście i tworzy nisko częstotliwościowe dopasowane cechy w abstrakcyjnej przestrzeni cech

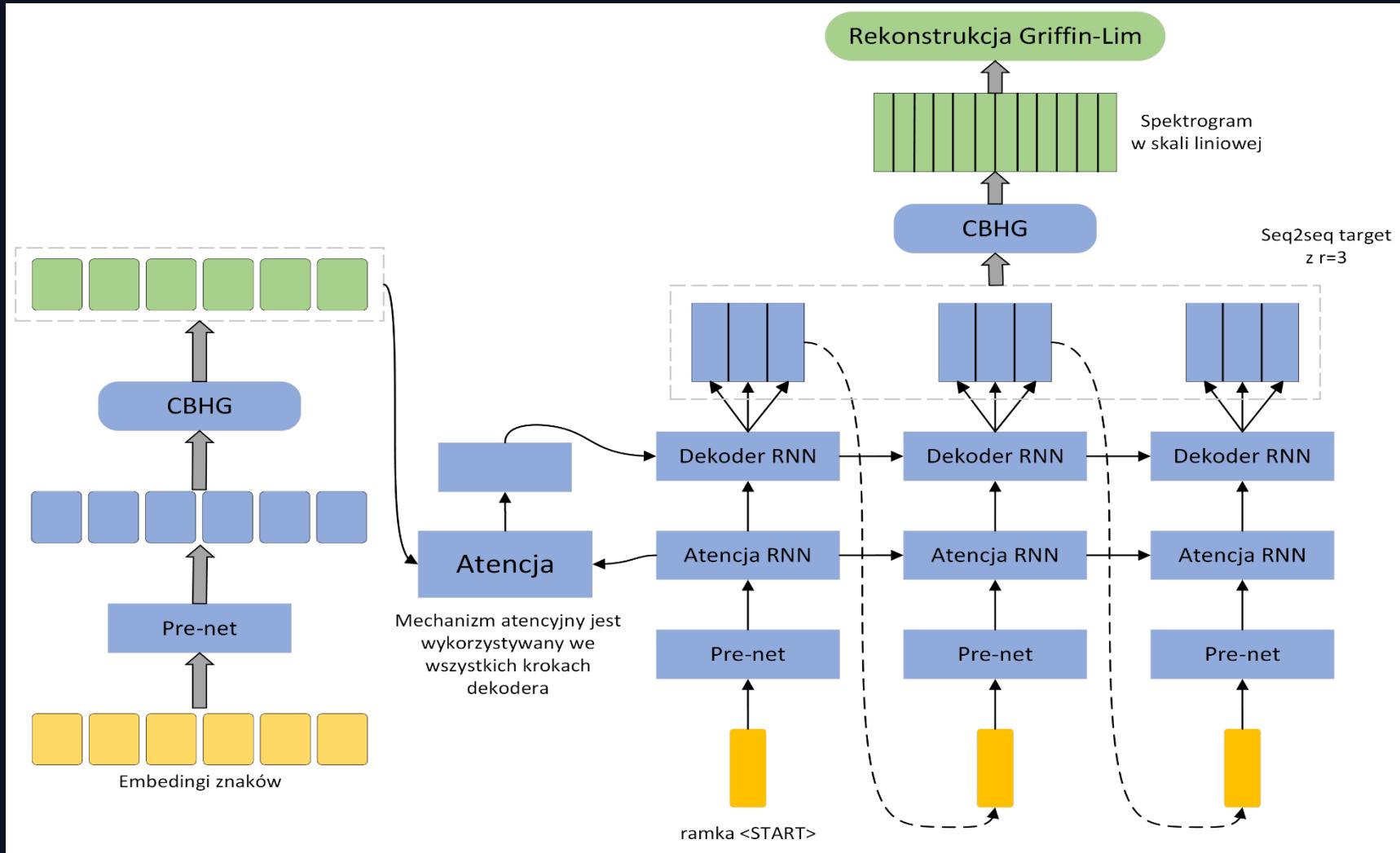
EATS - End-to-end Adversarial Text-to-Speech

- Zadaniem alignera jest mapowanie niedopasowanych sekwencji wejściowych do ich reprezentacji, które są już dopasowane do wyjścia
- Dekoder przejmuje te cechy i przeskalowuje je z wykorzystaniem splotów 1D do stworzenia waveformów
- Generator jest siecią typu feed-forward wykorzystującą różnicowe dopasowanie bazujące na predykcji długości tokenów
- Zastosowano algorytm Dynamic Time Warping, aby uzyskać w generowanym dźwięku tymczasowe zmienności sygnału
- Wynik MOS: 4.083

Tacotron

- Tacotron został przedstawiony przez Google w 2017 jako system typu end-to-end
- Wynik MOS: 3.82
- Jest to model typu seq2seq z wykorzystaniem architektury enkoder-dekoder oraz mechanizmu atencyjnego
- Model można podzielić na 3 segmenty – enkoder, dekoder bazujący na mechanizmie atencyjnym oraz sieć post-processową.
- Oryginalne rozwiązanie stosowało moduł CBHG – (1D Convolution Bank + Highway Network + bidirectional GRU)
- CBGH służyło do ekstakcji reprezentacji z sekwencji

Tacotron



Tacotron - CBGH

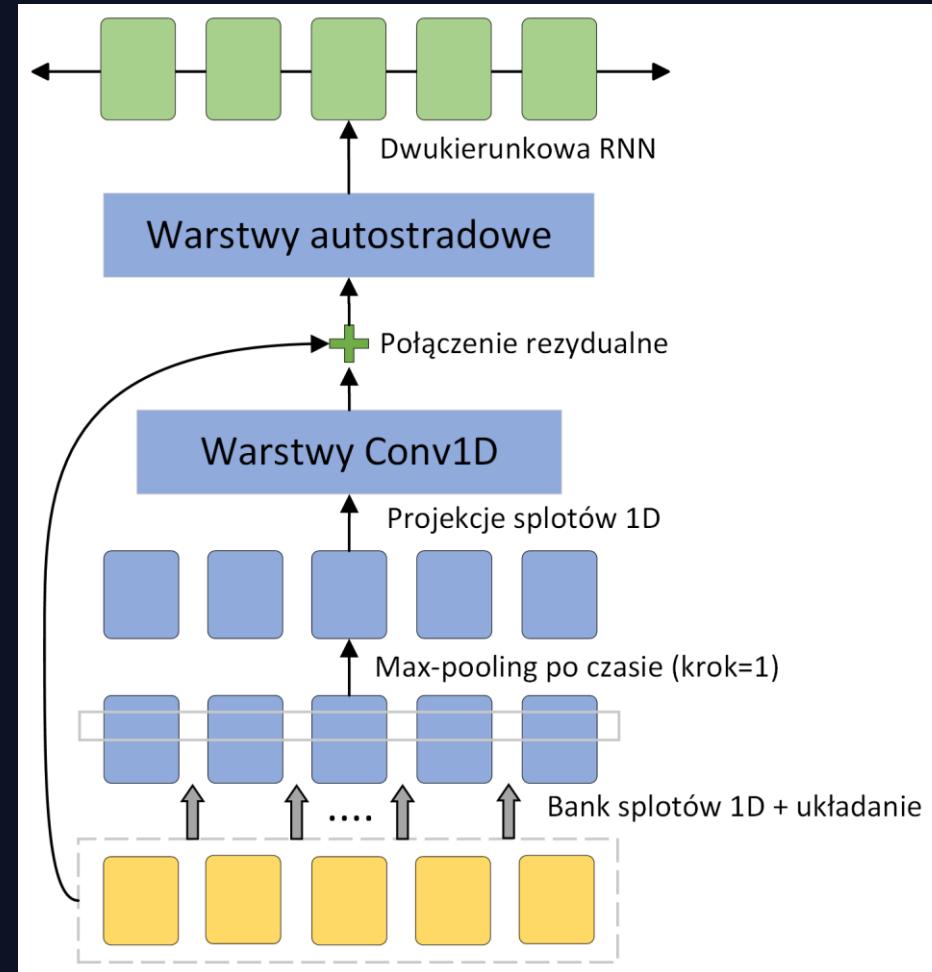
Mechanizm działania CBGH wyglądał następująco:

- Na sekwencji wejściowej dokonywany jest splot z zestawem K filtrów spłotowych 1D, gdzie k – ty zestaw zawiera C_k filtrów o szerokości k ($k = 1, 2 \dots K$)
- Filtry te wyraźnie modelują informacje lokalne i kontekstowe (podobnie do modelowania unigramów, bigramów aż do k-gramów)
- Wyjścia splotów są „stackowane” i poddawane max poolingowi aby zwiększyć lokalne stałe. Wykorzystuje się krok =1 by zachować rozdzielcość czasową
- Następnie przetworzone sekwencje są przenoszone na kilka splotów 1D o stałej szerokości, a ich wyjścia są sumowane z oryginalnym sekwencją wejściową poprzez połączenie rezydualne

Tacotron - CBGH

- Wyjście splotów są wprowadzane poprzez wielowarstwową sieć typu highway-network w celu ekstrakcji cech wysokopoziomowych.
- Wyjście sieci typu highway-network łączy się z dwukierunkowymi komórkami GRU w celu ekstrakcji cech sekwencyjnych pochodzących zarówno z kontekstu z przodu i z tyłu
- CBGH jest inspirowana architekturami tłumaczenia maszynowego

Tacotron - CBGH



Tacotron - enkoder

- Wejście do enkodera to sekwencja znaków, gdzie każdy znak jest reprezentowany jako wektor one-hot i osadzony w jednym, ciągłym wektorze
- Następnie w module pre-net następują transformacje nieliniowe
- Moduł CBHG przekształca wyjścia pre-net w finalne reprezentacje enkodera wykorzystywane przez mechanizm atencyjny

Tacotron - dekoder

- Warstwa pre-net na wejściu (podobnie jak w enkoderze)
- Wykorzystany jest dekoder atencyjny z funkcją tanh - warstwa rekurencyjna produkuje zapytanie atencyjne dla każdego kroku czasowego dekodera
- Wektor kontekstowy i wyjście warstwy atencyjnej jest łączone i tworzy wejście do dekodera będącego siecią RNN
- Wykorzystywany jest stos komórek GRU z pionowymi połączonymi rezydualnymi w dekoderze
- Następnie wykorzystywana jest w pełni połączona warstwa do predykcji „targetów” dekodera
- Stosowane jest jednoczesna predykcja wielu ramek – pozwala to na zmniejszenie rozmiaru modelu, czasu treningu i czasu inferencji

Tacotron – sieć post-processingowa

- Celem sieci jest przekształcenie wyniku przetwarzania seq2seq w postać, która może być syntezowana do postaci waveform
- W tym celu wykorzystany jest algorytm Griffin-Lim
- Sieć post processingowa przewiduje składowe harmoniczne – tworzenie spektrogramu i przetworzenie go na waveform
- Algorytm Griffin-Lim jest wykorzystany jako prosty algorytm i już w artykule opisującym sieć wskazano możliwość zastąpienia go wokoderem zbudowanym na sieci WaveNet

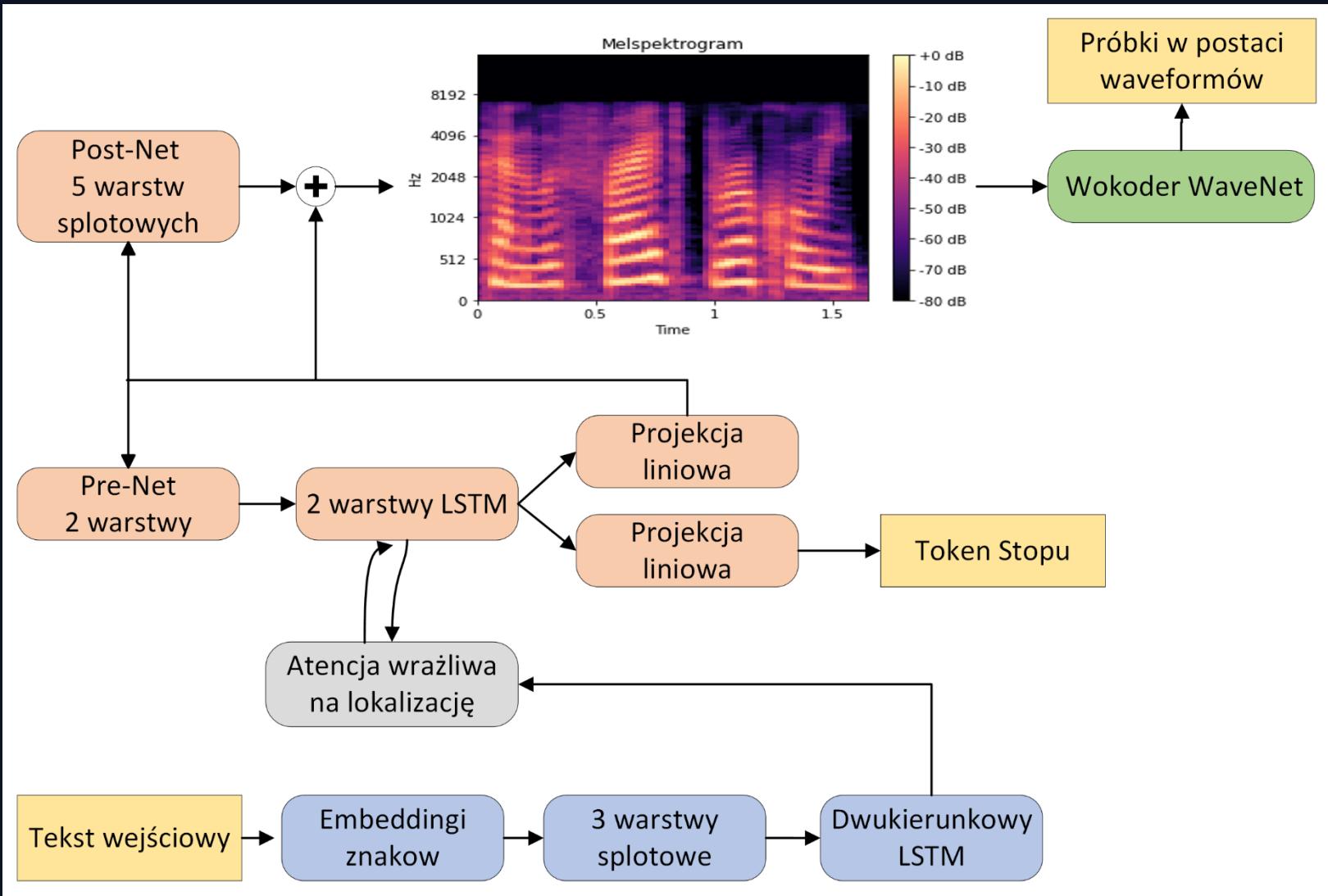
Tacotron-2

- Pokazany na koniec 2017 roku
- Ulepszona wersja Tacotron, brak większych zmian w samej architekturze sieci
- Główne zmiany to:
 - Enkoder składający się z 3 warstw splotowych i dwukierunkowych LSTM (pozbyto się modułów CBGH i pre-net)
 - Mechanizm atencyjny typu Location Sensitive
 - Dekoder składający się z autogregesywnej sieci RNN w skład której wchodzi sieć typu pre-net, 2 jednokierunkowe LSTM i 5 warstw splotowej sieci Post-Net

Tacotron-2

- Główne zmiany to:
 - Algorytm Griffin-Lim zastąpiono wokoderem opartym o sieć WaveNet (kolejną iterację WaveNet – 1000 razy szybszą od oryginalnego rozwiązania)
 - Zastosowano melspektrogramy zamiast opartych o skale liniową
- Efektem zmian był znaczący przyrost wyniku MOS
- Wynik MOS: **4.53**
- Najlepszy wynik do tej pory z sieci syntezujących głos – na równi z naturalną mową

Tacotron-2



Zastosowanie sieci Tacotron-2

- Automatyczne generowanie korpusów do uczenia maszynowego
- Synteza mowy w ramach TTS:
 - czytanie tekstu w aplikacjach czy w serwisach internetowych
 - lektorzy w filmach
 - boty konwersacyjne
 - wiele innych
- Sieć Tacotron umożliwiła powstanie dobrej jakości deep-fakeów głosowych, co za tym idzie zwróciła uwagę fałszerzy
- W oparciu o Tacotron powstał najpopularniejszy ogólnodostępny algorytm klonowania głosu – zagrożenie w postaci kradzieży tożsamości

Źródła

- **EATS explanation:**
<https://www.youtube.com/watch?v=WTB2p4bqtXU>
- Preprint opisujący architekturę Glow:
<https://arxiv.org/pdf/1807.03039.pdf>
- Preprint opisujących architekturę Tacotron:
<https://arxiv.org/pdf/1703.10135.pdf>
- Preprint opisujących architekturę Tacotron-2:
<https://arxiv.org/pdf/1712.05884.pdf>
- Preprint opisujących architekturę Deep Voice:
<https://arxiv.org/pdf/1702.07825.pdf>
- Preprint opisujących architekturę Deep Voice 2:
<https://proceedings.neurips.cc/paper/2017/file/c59b469d724f7919b7d35514184fdc0f-Paper.pdf>
- Preprint opisujących architekturę Deep Voice 3:
<https://arxiv.org/pdf/1710.07654.pdf>

Pytania

Pytania?

Dziękuję

Szymon Zaporowski



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

Głębokie przetwarzanie tekstu i mowy

Rozpoznawanie i uwierzytelnianie mówców z wykorzystaniem głębokiego uczenia

Szymon Zaporowski



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

Plan wykładu

- Biometria – czym jest?
- Budowa systemu biometrycznego
- Weryfikacja, a identyfikacja
- Cechy fizyczne i behawioralne
- Metryki w systemach biometrycznych
- Systemy uwierzytelniania biometrycznego
- Wektory cech w uwierzytelnianiu biometrycznym

Biometria – czym jest?

- Biometria to pomiar i analiza statystyczna unikalnych cech ludzkich – zarówno fizycznych jak i behawioralnych
- Biometria jako technologia jest głównie używana w celu identyfikacji i przyznawania dostępu lub do identyfikacji osób będących pod nadzorem
- Podstawowym założeniem uwierzytelniania biometrycznego jest to, że wewnętrzne cechy fizyczne lub behawioralne mogą dokładnie zidentyfikować każdą osobę

Biometria – czym jest?

- Uwierzytelnianie z użyciem weryfikacji biometrycznej staje się coraz częściej powszechnie w systemach bezpieczeństwa zarówno korporacyjnych jak i publicznych, tego typu zabezpieczenia można znaleźć również w elektronice konsumpcyjnej.
- Oprócz zabezpieczeń, siła napędowa weryfikacji biometrycznej była i jest wygodna - nie ma haseł do zapamiętania lub tokenów bezpieczeństwa do noszenia.
- Niektóre metody biometryczne, takie jak analiza chodu osoby czy też ruchów saakadowych oka, mogą działać bez bezpośredniego kontaktu z osobą uwierzytelnianą.

Biometria – jak zbudowany jest system biometryczny

System biometryczny składa się z następujących komponentów:

- czytnika lub urządzenia skanującego do rejestracji uwierzytelnianego czynnika biometrycznego;
- oprogramowania do konwersji zeskanowanych danych biometrycznych do znormalizowanego formatu cyfrowego oraz w celu porównania punktów wspólnych próbki weryfikacyjnej z zapisanymi danymi weryfikacyjnymi;
- baza danych do bezpiecznego przechowywania danych biometrycznych w celu dalszego porównywania wzorca z próbami weryfikacyjnymi.

Biometria – jak zbudowany jest system biometryczny

- Dane biometryczne mogą być przechowywane w scentralizowanej bazie danych
- Nowoczesne implementacje biometryczne często wykorzystują gromadzenie danych biometrycznych lokalnie
- Dodatkowo są one ukrywane kryptograficznie (szifrowanie), dzięki czemu **uwierzytelnianie** lub **identyfikacja** może zostać wykonana bez bezpośredniego dostępu do danych biometrycznych.

Weryfikacja kontra identyfikacja

Czym właściwie różni się weryfikacja od identyfikacji?

- Jeśli osoba twierdzi, że nazywa się np. Jan Kowalski, a głos ma posłużyć do weryfikacji tego faktu to taką operację nazywamy weryfikacją lub uwierzytelnieniem
- Jeśli mamy nieokreśloną tożsamość mówcy i mamy stwierdzić kim ta osoba jest na bazie próbki głosu to jest to identyfikacja
- Można to rozumieć jako dopasowanie wzorca 1:1 w przypadku weryfikacji i 1:N w przypadku identyfikacji

Biometria – cechy fizjologiczne i behawioralne

Dwa główne typy identyfikatorów biometrycznych są cechami fizjologicznymi lub cechami behawioralnymi. Identyfikatory fizjologiczne obejmują następujące elementy:

- Rozpoznawanie twarzy
- Odciski palców
- Geometria dloni
- Rozpoznawanie tęczówki
- Rozkład naczyń krwionośnych
- Skanowanie rogówki
- Dopasowanie DNA
- Podpis cyfrowy
- **Rozpoznawanie głosowe**

Biometria – cechy fizjologiczne i behawioralne

Identyfikatory behawioralne obejmują unikalne sposoby działania jednostki – człowieka, można przy tym rozróżnić następujące cechy :

- Rozpoznawanie wzorców pisania, ruchów myszką i palcami,
- Wzory zaangażowania w przypadku korzystania z mediów społecznościowych czy witryn internetowych,
- Ruchy gałek ocznych – saakadowe ruchy oka, śledzenie wzroku,
- inne gesty

Biometria – cechy fizjologiczne i behawioralne

Jak rozumieć rozróżnienie na cechę fizjologiczną i behawioralną w przypadku głosu?

Biometria głosu jest stricte biometrią opierającą się o cechy fizjologiczne – bazujące na trakcie głosowym, będącym praktycznie unikalnym dla każdego człowieka

Można jednak wyobrazić sobie sytuację, gdy osoba posługuje się specyficznym językiem, słownictwem i składnią pozwalającą na jednoznaczną identyfikację – może to być uważane za osobną, behawioralną cechę biometryczną - idiolekt

Zadanie

- Czy po zmianie głosu np. zamaskowaniu, tak jak w przypadku osób chcących zachować anonimowość możliwe jest rozpoznanie osób na podstawie sposobu mówienia?

Istotne metryki w systemach biometrycznych

- False Acceptance Rate (FAR) – definiowane jako prawdopodobieństwo, że system nieprawidłowo przypisał wzorzec z wejścia do niepasującego wzorca w bazie danych. Miara ta oblicza procent błędnych prób wejścia, które są akceptowane (wpuszczane do systemu)
- False Rejection Rate (FRR) – definiowane jako prawdopodobieństwo, że system nie będzie umiał znaleźć podobieństwa między wzorcem wejściowym i wzorcem w bazie danych. Miara ta oblicza procent poprawnych wejść, które są nieprawidłowo odrzucone przez system.

Systemy uwierzytelniania biometrycznego

- Celem weryfikacji mówcy znanej również jako uwierzytelnianie głosowe jest uwierzytelnianie żądanej tożsamości z pomiarów wykonywanych na sygnale głosowym.
- Zastosowania weryfikacji mówców obejmują kontrolę wejścia do ograniczonych pomieszczeń, dostęp do uprzywilejowanych informacji, transferu funduszy, autoryzację karty kredytowej, bankowości głosowej i podobnych transakcji.

Pytanie: Czy weryfikacja głosowa jest bezpieczna?

Systemy uwierzytelniania biometrycznego

Istnieją dwa rodzaje uwierzytelniania głosu:

- Jeden weryfikuje tożsamość mówcy w oparciu o właściwości głosowe specyficzne dla rozmówcy, odzwierciedlenie w słowach mówionych lub zdaniach
- Drugi na podstawie treści hasła mówionego lub frazy, takich jak osobisty numer identyfikacyjny (PIN), numer ubezpieczenia społecznego lub nazwisko panieńskie matki
- Są odpowiednio systemy typu Text-Independent i Text-Dependent

Systemy uwierzytelniania biometrycznego

- Istnieje kilka rodzajów taksonomii dla systemów uwierzytelniania głosowego - architektura, rodzaj parametryzacji, cel użycia, zależność lub niezależność od frazy itp.

Taxonomia oparta na architekturach:

- Na podstawie Mieszanych Modeli Gaussowskich z Uniwersalnym Modelem Tła – GMM UBM
- Na podstawie Ukrytych Modeli Markowa -HMM
- Na podstawie sztucznych sieci neuronowych

Systemy uwierzytelniania biometrycznego

Taksonomia związana z rodzajem parametryzacji sygnału wejściowego:

- Spektrogramy;
- Współczynniki Melcepstralne
- Współczynniki Gammatone
- Współczynniki LPC
- Transformacja falkowa

Wektory cech w uwierzytelnianiu głosowym

- Standardowo w modelach GMM UBM stosowano metodę i-vector
- i-vector to cecha reprezentująca idiosynkratyczną charakterystykę wzoru dystrybucyjnego na poziomie ramki – uwzględniamy nie tylko cechy związane z głosem, ale również z językiem
- Ekstrakcja i-vectora jest zasadniczo zmniejszeniem wymiarowości superwektora GMM
- Do jego ekstrakcji jest stosowana metoda JFA – Joint Factor Analysis

Wektory cech w uwierzytelnianiu głosowym

- D-wektor oraz x-wektor to dwa wektory cech, które pojawiły się wraz z rozwojem metod weryfikacji biometrycznej opartej o sztuczne sieci neuronowe, a konkretnie o uczenie głębokie
- Aby wyodrębnić D-Vector, model głębokiej sieci neuronowej bierze ustawione w stosy cechy banków filtrów np. filtry cepstralne lub melowe (proces podobny do modelu akustycznego stosowanego w ASR) i generuje etykietę mówcy typu one-hot (lub prawdopodobieństwo mówcy). D-Vector to uśredniona aktywacja z ostatniej ukrytej warstwy tej głębokiej sieci neuronowej – ścinana jest warstwa z wyjściem i ekstrahowane są cechy z przedostatniej warstwy

Wektory cech w uwierzytelnianiu głosowym

Czym różni się x-vector od d-wektora?

- Jedni powiedzą, że x-vector i d-wektor są tożsame
- Inni, że x- vector korzysta z okna przesuwnego z ramek sygnału jako wejścia, wykorzystuje sieć typu TDNN (Time Delay Neural Network) , do uzyskania kontekstu i przejścia do reprezentacji na poziomie ramki. Użyta jest też warstwa pooling, aby uzyskać średnią wartość osadzeń na poziomie ramki, a następnie przekazać tę informację do warstwy liniowej i uzyskać embedding na poziomie segmentu.
- Jeszcze inni, że x-vector korzysta z PLDA do obliczenia wyniku, a d-vector z podobieństwa kosinusowego

Pytania

Pytania?

Dziękuję

Szymon Zaporowski



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.



POLITECHNIKA
GDAŃSKA



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Głębokie przetwarzanie tekstu i mowy

Rozpoznawanie i uwierzytelnianie mówców z wykorzystaniem głębokiego uczenia – część 2

Szymon Zaporowski



Rzeczpospolita
Polska



Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

Plan wykładu

- Wektory cech - ogólnie
- i-wektor
- D-wektor
- X-wektor
- Embedding, a rodzaj treningu
- Rodzaje architektur, a wejście sieci
- Triplet loss
- Zbiory
- Diaryzacja

Wektory cech w uwierzytelnianiu głosowym

- Standardowo w modelach GMM UBM stosowano metodę i-vector
- i-vector to cecha reprezentująca idiosynkratyczną charakterystykę wzoru dystrybucyjnego na poziomie ramki – uwzględniamy nie tylko cechy związane z głosem, ale również z językiem
- Ekstrakcja i-vectora jest zasadniczo zmniejszeniem wymiarowości superwektora GMM
- Do jego ekstrakcji jest stosowana metoda JFA – Joint Factor Analysis

Wektory cech w uwierzytelnianiu głosowym

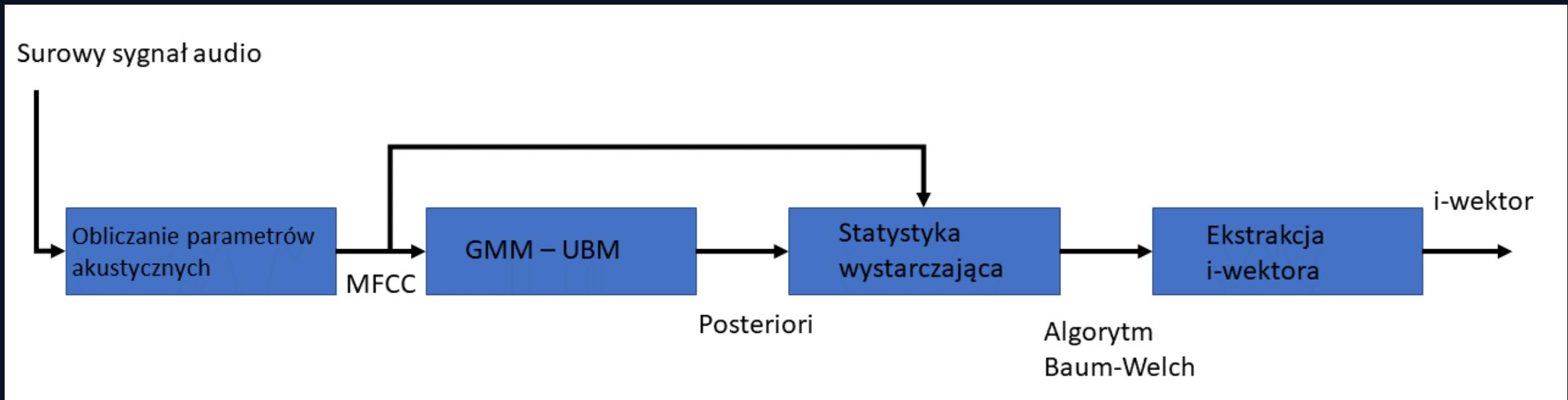
Czym różni się x-vector od d-wektora?

- W interenecie można znaleźć informacje, że że x-vector i d-wektor są tożsame
- Można znaleźć również informacje, że x- vector korzysta z okna przesuwnego z ramek sygnału jako wejścia, wykorzystuje sieć typu TDNN (Time Delay Neural Network) , do uzyskania kontekstu i przejścia do reprezentacji na poziomie ramki. Użyta jest też warstwa pooling, aby uzyskać średnią wartość osadzeń na poziomie ramki, a następnie przekazać tę informację do warstwy liniowej i uzyskać embedding na poziomie segmentu.
- Jeszcze inni, że x-vector korzysta z PLDA do obliczenia wyniku, a d-vector z podobieństwa kosinusowego, ale jest to praktycznie to samo
- **Co jest faktycznie prawdą?**

Wektory cech w uwierzytelnianiu głosowym

- d-wektor oraz x-wektor to dwa wektory cech, które pojawiły się wraz z rozwojem metod weryfikacji głosowej opartej o sztuczne sieci neuronowe, a konkretnie o uczenie głębokie
- Aby wyodrębnić d-vektor, model głębokiej sieci neuronowej bierze ustawione w stosy cechy banków filtrów np. filtry cepstralne lub melowe (proces podobny do modelu akustycznego stosowanego w ASR) i generuje etykietę mówcy typu one-hot (lub prawdopodobieństwo mówcy). d-vektor to uśredniona aktywacja z ostatniej ukrytej warstwy tej głębokiej sieci neuronowej – ścinana jest warstwa z wyjściem i ekstrahowane są cechy z przedostatniej warstwy

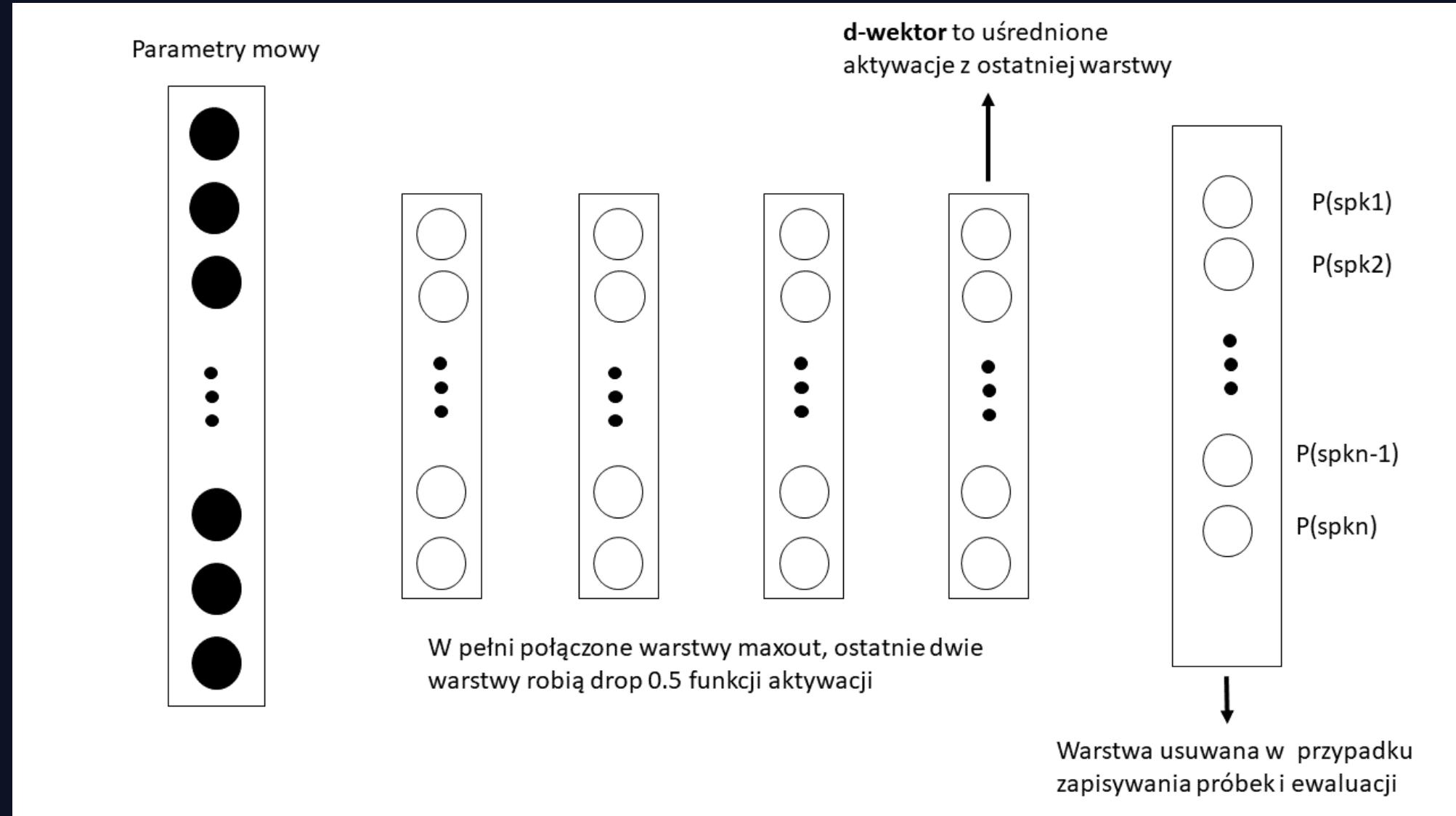
Jak policzyć i-wektor?



Jak policzyć d-wektor?

- Główną ideą stojącą za d-wektorem jest przypisanie tożsamości mówcy (ground truth) jako etykiety treningowej ramki należącej do wypowiedzi wykorzystywanej w fazie treningu, która zamienia trening modelu w klasyfikację
- d-wektor poszerza każdą ramkę treningową swoją zawartością i wykorzystuje głęboką sieć typu maxout do klasyfikacji ramek treningowej wypowiedzi do tożsamości osoby wypowiadającej dane zdanie. Sieć wykorzystuje softmax jako warstwę wyjściową, aby zminimalizować funkcję straty entropii krzyżowej pomiędzy etykietami ground-truth i wyjściem sieci

Jak policzyć d-wektor?



Czym jest maxout?

- Koncept zaproponowany przez Goodfellowa w 2013
- Warstwa maxout zawiera w sobie neurony typu maxout, które posiadają funkcję aktywacji będącą maksymalną wartością wejścia
- Dzięki takiemu rozwiązaniu możliwe jest odwzorowanie praktycznie dowolnej funkcji – MLP z dwiema jednostkami maxout jest w stanie zaimplementować Relu, leakyReLU, a w przypadku odpowiednio dużej liczby jednostek nawet funkcje wklęsłe i wypukłe (np. x^2 i $-x^2$)

Jak policzyć x-wektor?

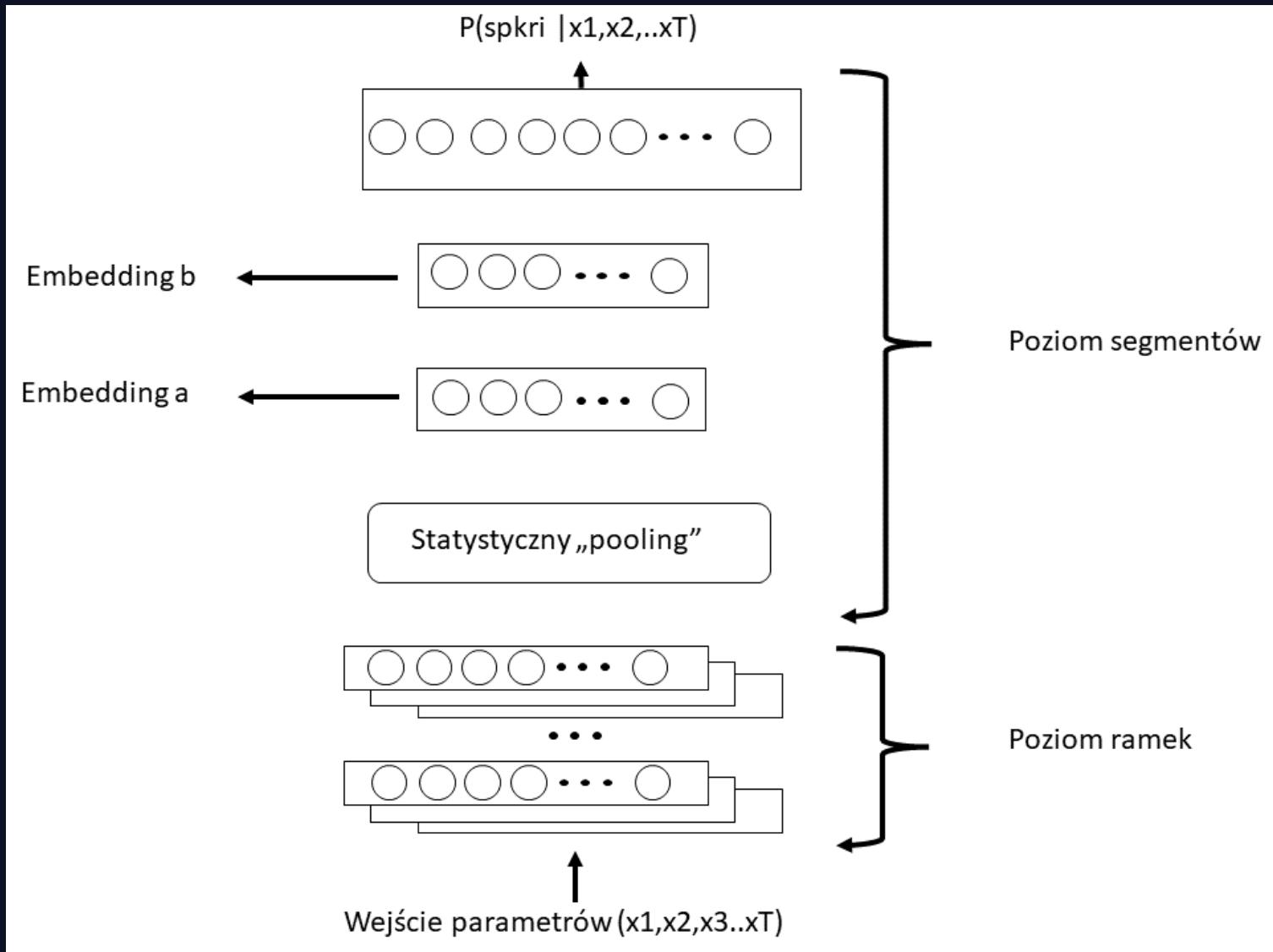
x-wektor jest ewolucją d-wektora – przechodzimy z poziomu ramek na poziom segmentów (całych wypowiedzi)

- Najpierw ekstrahowane są embeddingi z ramek mowy poprzez użycie sieci typu Time Delay,
- Następnie dokonywane jest połączenie wartości średniej i odchylenia standardowego (statystyczny pooling) na poziomie ramek w wektor na poziomie segmentów (poziom wypowiedzi) poprzez warstwę typu statistical pooling
- Na końcu używana jest sieć typu feed-forward do klasyfikacji cech na poziomie segmentów (przypisanie do mówcy)

Jak policzyć x-wektor?

- Warstwy time-delay, poolingu statystycznego i feed-forward są trenowane wspólnie
- X-wektor jest definowany jako embedding na poziomie segmentów i jest tworzony poprzez wyciągnięcie wartości z jednej z warstw sieci feed-forward, najczęściej od drugiej do ostatniej warstwy
- Augmentacja danych pozwala na znaczące poprawienie jakości działania x-wektorów

Jak policzyć x-wektor?



Charakterystyki różnych podejść do embeddingów

<i>Model</i>	<i>Typ</i>	<i>Strategia treningu</i>	<i>Rodzaj etykiet do treningu</i>	<i>Rodzaj backendu</i>	<i>Etykiety do treningu backendu</i>
i-wektor	Generatywny	Nienadzorowany	x	PLDA	Tożsamość mówcy
d-wektor	Dyskryminacyjny	Nadzorowany	Tożsamość mówcy	CS	-
x-wektor	Dyskryminacyjny	Nadzorowany	Tożsamość mówcy	PLDA	Tożsamość mówcy

Funkcje straty stosowane w rozpoznawaniu mówcy

- Jedną z najpopularniejszych funkcji straty jest triplet loss
- W tej funkcji wykorzystuje się 3 próbki – punkt zaczepienia (anchor), próbkę negatywną (negative) oraz próbkę pozytywną (positive)
- Celem tej funkcji jest możliwe „zblżenie” do siebie próbki pozytywnej i punktu zaczepienia oraz „oddalenie” próbki negatywnej
- Odległości między próbками liczone są z wykorzystaniem podobieństwa kosinusowego lub kwadratu dystansu euklidesowego
- Konieczne jest dokonanie normalizacji długości wektorów przed sprawdzeniem podobieństwa

Podsumowanie rodzajów sieci w zależności od rodzaju wejść

Wejście	CNN	LSTM	Struktry Hybrydowe
Waveform	CNN, SincNet	x	CNN-LSTM, CNN-GRU, Transformery
Spektrogram	Resnet,VGGNet, Inception	x	CNN-GRU
Bank filtrów	TDNN, ResNet, VGGNet, Inception	LSTM, GRU	BLSTM-ResNet, TDNN-LSTM
MFCC	TDNN,ResNet	x	TDNN-LSTM

Zbiory danych używane w SR i SI

Nazwa	Rok	Język	Zastosowanie	Liczba mówców	Ilość danych	Rodzaj anotacji
<i>NIST SRE</i>	<i>1996-2020</i>	<i>Multi</i>	<i>Rozpoznawanie mówców Text-ind</i>	<i>x</i>	<i>x</i>	<i>Ręczny</i>
<i>VoxCeleb 1</i>	<i>2017</i>	<i>Angielski</i>	<i>Weryfikacja mówców text-ind</i>	<i>1251</i>	<i>352 godziny</i>	<i>Automatyczny</i>
<i>VoxCeleb 2</i>	<i>2018</i>	<i>Multi</i>	<i>Weryfikacja mówców text-ind</i>	<i>6112</i>	<i>2442 godziny</i>	<i>Automatyczny</i>
<i>RSR2015</i>	<i>2015</i>	<i>Angielski</i>	<i>Werfykcja Text-dep</i>	<i>300</i>	<i>151 godzin</i>	<i>Ręczny</i>
<i>Librispeech</i>	<i>2015</i>	<i>Angieslki</i>	<i>Weryfikacja mówców text-ind</i>	<i>>9000</i>	<i>1000 godzin</i>	<i>Ręczny</i>
<i>Hi-MIA</i>	<i>2020</i>	<i>Angieslki, Chiński</i>	<i>Werfykcja Text-dep</i>	<i>340</i>	<i>1561 godzin</i>	<i>Ręczny</i>
<i>FFSVC 2020</i>	<i>2020</i>	<i>Mandaryński</i>	<i>Werfykcja Text-dep, rozpoznawanie text-ind</i>	<i>x</i>	<i>x</i>	<i>Ręczny</i>
<i>AMI</i>	<i>2005</i>	<i>Angielski</i>	<i>Diaryzacja</i>	<i>x</i>	<i>100 godzin</i>	<i>Ręczny</i>

Diaryzacja mówcy

- Oprócz weryfikacji i identyfikacji mówcy coraz większe znaczenie zyskuje diaryzacja mówcy – jest to proces w którym sygnał audio jest dzielony na jednorodne segmenty w zależności od tożsamości mówcy.
- Diaryzacja mówcy odpowiada na pytanie „kto mówi w danym czasie?”
- Diaryzacje jest kombinacją segmentacji mówców – zmiany mówcy w sygnale audio i ich klastrowania – grupowanie segmentów mowy bazując na charakterystyce mówcy

Pytania

Pytania?



POLITECHNIKA
GDAŃSKA



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Dziękuję

Szymon Zaporowski



Fundusze
Europejskie
Polska Cyfrowa



Rzeczpospolita
Polska



Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

Głębokie przetwarzanie tekstu i mowy

“Transfer Stylu”

Sebastian Cygert
Katedra Systemów Multimedialnych,
WETI



Fundusze
Europejskie
Polska Cyfrowa



Rzeczpospolita
Polska



Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.

Transfer stylu

1. Zastosowania
2. Transfer stylu w przetwarzaniu obrazów
3. Szybki transfer stylu
4. Transfer stylu w konwersji głosu

Zastosowania

- zastosowania rozrywkowe / artystyczne:
 - w 2018 sprzedano pierwszy obraz wygenerowany za pomocą transferu stylu za 425,000\$!¹
- kopiowanie głosu,
- rozszerzanie zbiory danych treningowych (ang. *data augmentation*),
- zagrożenia: “kradzież głosu / tożsamości”

1 - <https://medium.datadriveninvestor.com/ai-and-art-why-gans-sold-for-400k-6e27c06371c1>

Transfer stylu w obrazie

obraz wejściowy

A



obraz wyjściowy

B



źródło "stylu"

Źródło: Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016

Transfer stylu w obrazie

1. Jak opisać zawartość (*content*)?

2. Jak opisać styl?

Źródło: Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016

Transfer stylu w obrazie

1. Jak opisać zawartość (*content*)?
Możemy użyć wektora cech z sieci CNN!
2. Jak opisać styl?

Źródło: Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016

Wektor cech a klasteryzacja



- możliwe jest znalezienie **najbardziej podobnego obrazu** do zadanego znajdując najbliższego sąsiada dla obliczonego **wektora z ostatniej warstwy** (np. dystans Euklidesowy),
- wektor cech grupuje obrazy **semantycznie do siebie zbliżone**

Źródło: A. Krizhevsky, *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS 2012

Transfer stylu w obrazie

1. Jak opisać zawartość (*content*)?
Można użyć wektora cech z sieci CNN!

2. Jak opisać?
Można użyć macierzy Grama (*Gram matrix*)

Źródło: Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016

Macierz Grama

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l$$

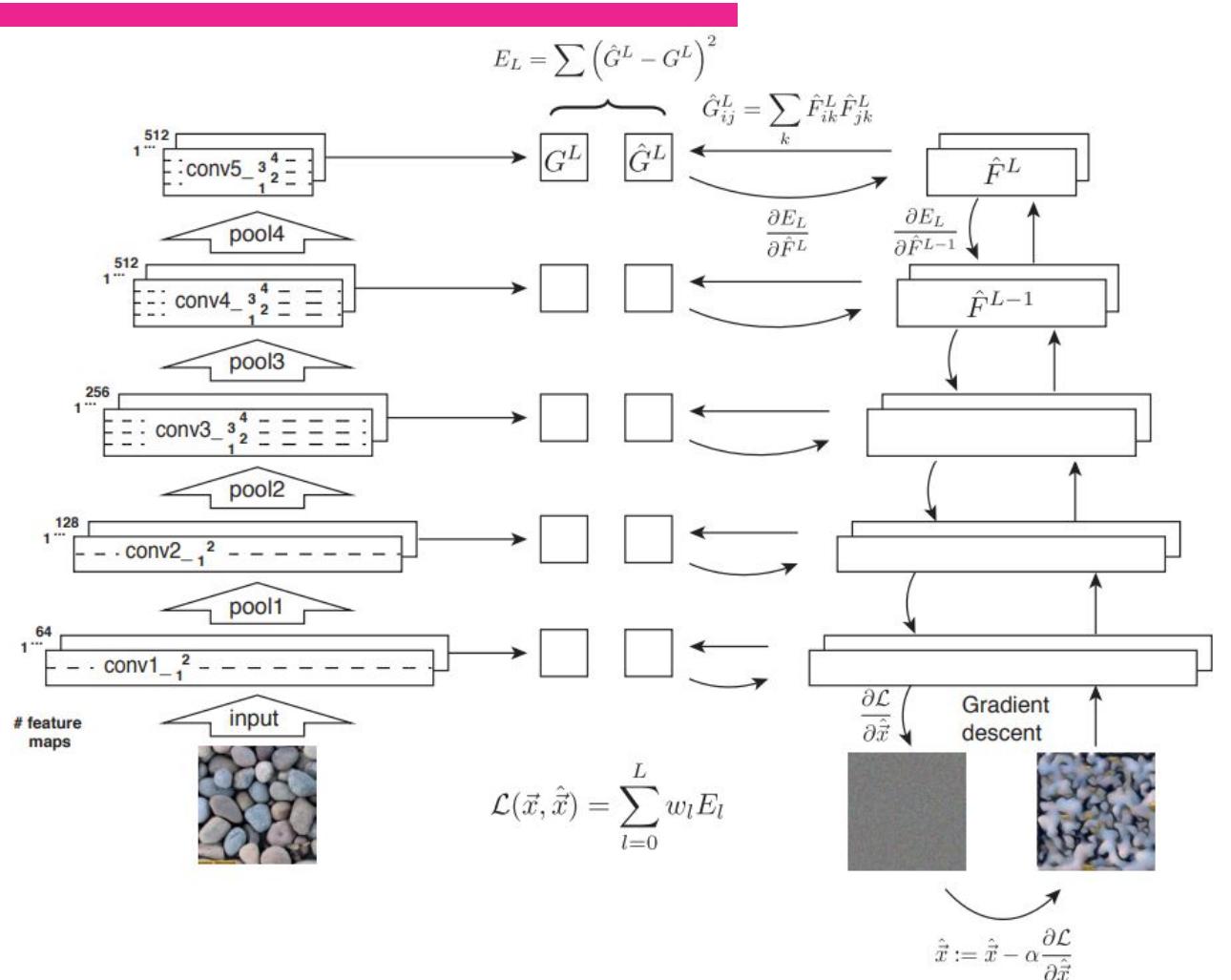
l - indeks warstwy,
 i, j - indeksy map wektorów cech,
 F - mapa wektora cech (*feature map*)
 k - liczba filtrów

- macierz Grama jest rozmiaru $k * k$,
- obliczany jest iloczyn skalarny (ang. *dot product*), który mówi o tym jak podobne do siebie są mapy wektorów cech o różnych indeksach,
- nie interesuje nas tutaj wartość (*content*), ale korelacja pomiędzy różnymi wektorami (**która utożsamiamy ze stylem**)

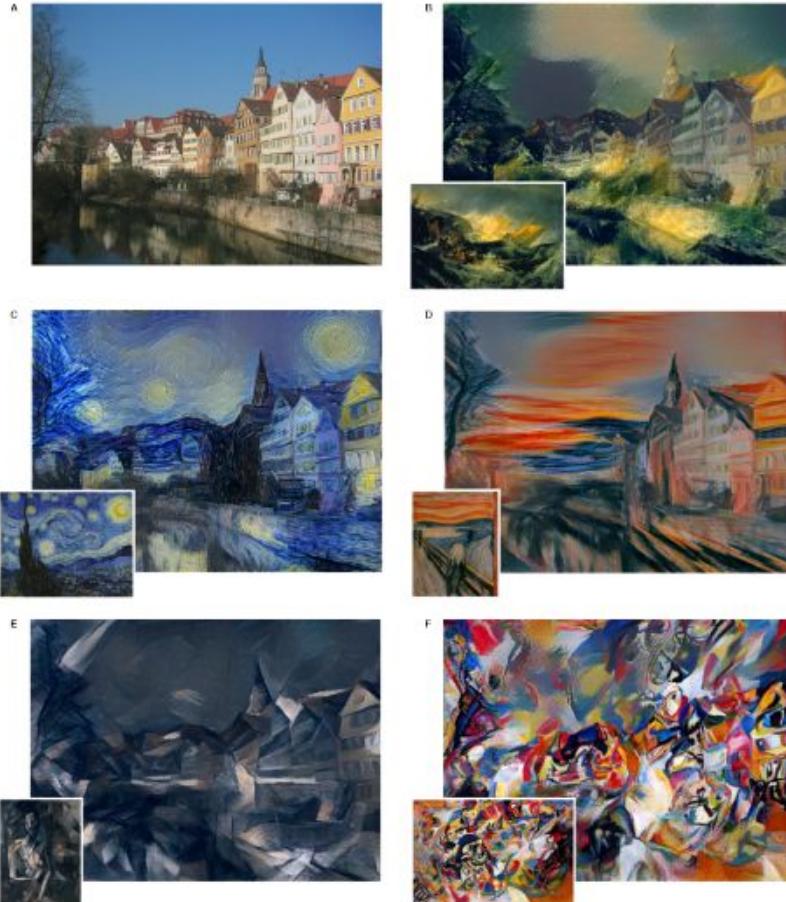
Jak wygenerować obraz?

Jak wygenerować obraz?

1. Uruchomić **wytrenowany (pre-trained)** model sieci CNN na wzorcowym obrazie
2. Zapisać aktywacje i obliczyć macierze Grama
3. Inicjalizacja generowanego obrazu (szum Gaussowski)
4. Przepuszczenie obrazu przez tę samą sieć, zapisanie macierzy Grama.
5. Obliczenie funkcji kosztu (różnica między macierzami wzorcowego i wygenerowanego obrazu)
6. Propagastacja wsteczna aby uzyskać gradient na obrazie
7. Aktualizacja obrazu zgodnie z gradientem.
8. Powrót do **kroku 4.**



Rezultaty



Źródło: Gatys, Ecker, and Bethge, "Image style transfer using convolutional neural networks", CVPR 2016

Szybki transfer stylu

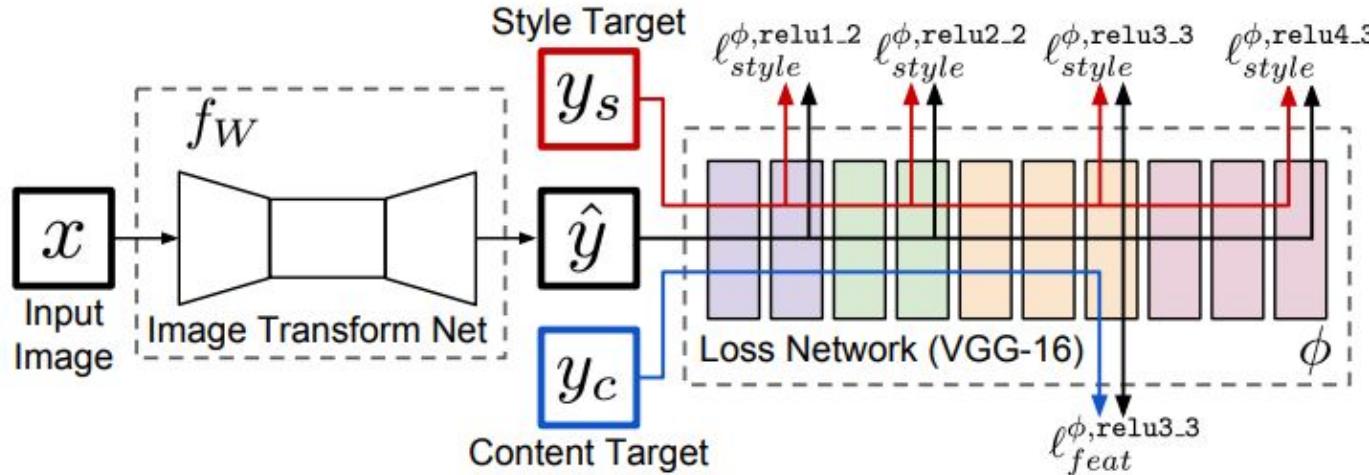
- opisana procedura jest bardzo czasochłonna. Wymaga kilkuset iteracji obliczania gradientu dla wybranej sieci,
- **jak szybko wygenerować obrazy?**

Szybki transfer stylu

- opisana procedura jest bardzo czasochłonna. Wymaga kilkuset iteracji obliczania gradientu dla wybranej sieci,
- **jak szybko wygenerować obrazy?**

Można wytrenować w tym celu dedykowaną sieć neuronową!

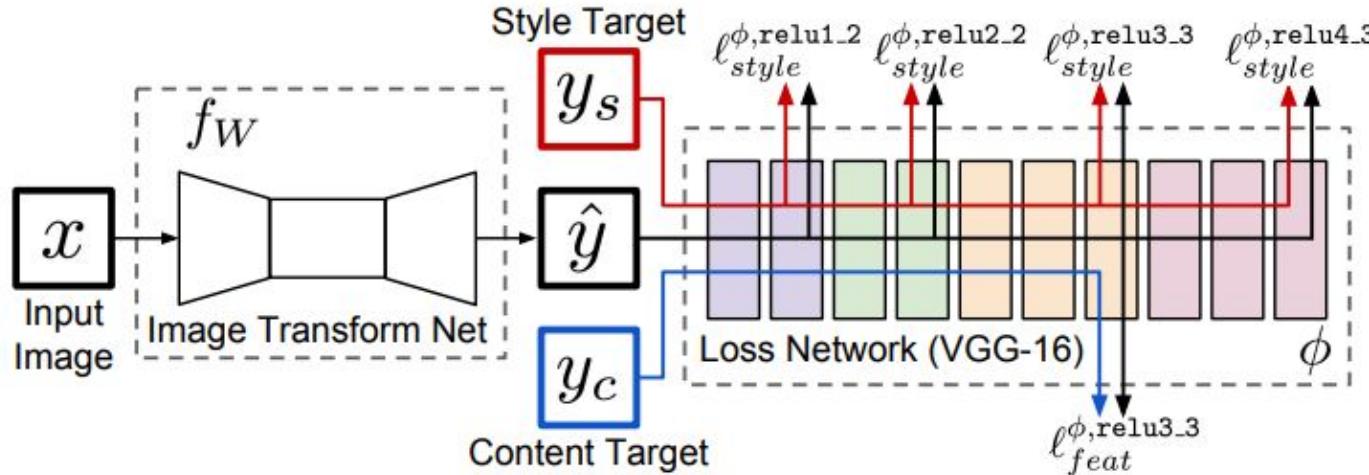
Szybki transfer stylu



1. Generujemy duży zbiór treningowy dla wybranego stylu korzystając z poprzednio opisanej metody.
2. Trenujemy sieć neuronową, której celem jest odwzorowanie stylu i zawartości.

Źródło: J. Johnson et al., "Perceptual losses for real-time style transfer and super-resolution." European conference on computer vision (ECCV) 2016.

Szybki transfer stylu

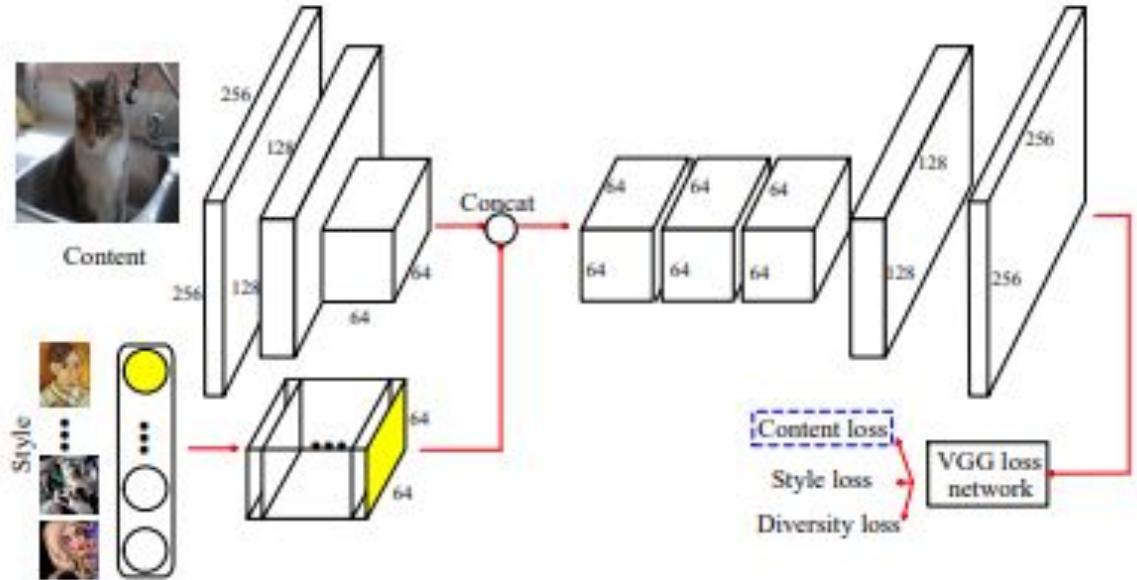


1. Generujemy duży zbiór treningowy dla wybranego stylu korzystając z poprzednio opisanej metody.
2. Trenujemy sieć neuronową, której celem jest odwzorowanie stylu i zawartości.

Działa szybko, ale wymaga **osobnych sieci** dla każdego stylu.

Źródło: J. Johnson et al., "Perceptual losses for real-time style transfer and super-resolution." European conference on computer vision (ECCV) 2016.

Uniwersalny szybki transfer stylu

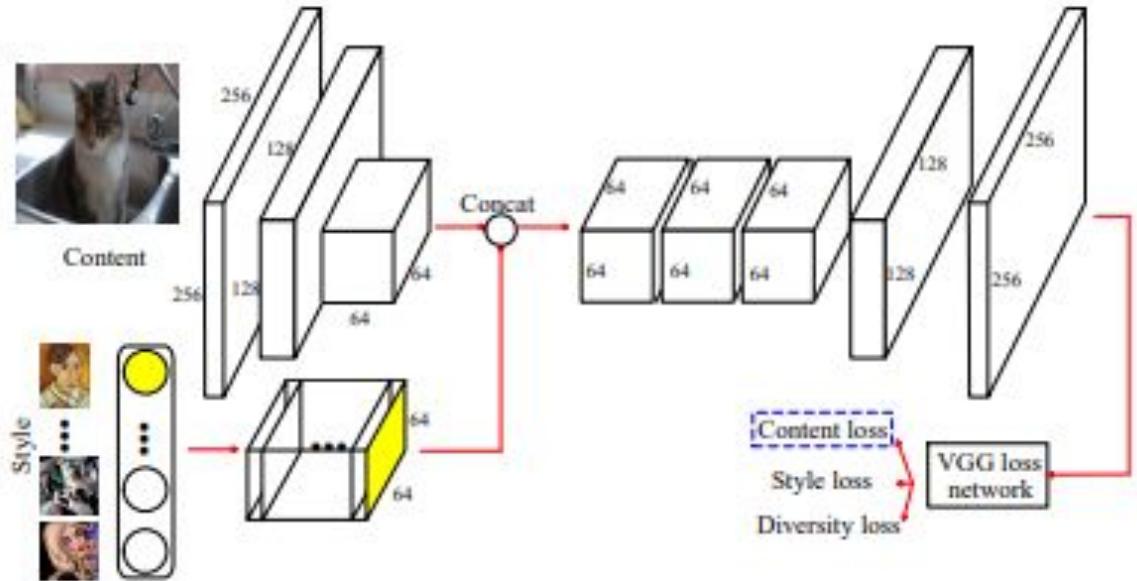


1. Oryginalny obraz i obraz stylu są przepuszczane przez sieć VGG aby uzyskać wektor cech.
2. Wektory cech są ze sobą łączone i przepuszczane przez dekoder, który generuje końcowy obraz.

Czy metoda może działać dla **dowolnego** stylu? (niewidzianego w czasie treningu)

Źródło: Li, Yijun, et al. "Diversified texture synthesis with feed-forward networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

Uniwersalny szybki transfer stylu



1. Oryginalny obraz i obraz stylu są przepuszczane przez sieć VGG aby uzyskać wektor cech.
2. Wektory cech są ze sobą łączone i przepuszczane przez dekoder, który generuje końcowy obraz.

Czy metoda może działać dla **dowolnego** stylu? (niewidzianego w czasie treningu)

W praktyce, tylko gdy w czasie treningu sieć zobaczy dużo różnych stylów i nowy styl nie będzie od nich znacząco inny

Źródło: Li, Yijun, et al. "Diversified texture synthesis with feed-forward networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.

Transfer stylu jako wzmacnienie danych



(a) Texture image
81.4% Indian elephant
10.3% indri
8.2% black swan



(b) Content image
71.1% tabby cat
17.3% grey fox
3.3% Siamese cat



(c) Texture-shape cue conflict
63.9% Indian elephant
26.4% indri
9.6% black swan

- Sieci rozpoznają obiekty, głównie patrząc na teksturę - nisko-poziomowe cechy obrazu,
- Można użyć transferu stylu do wygenerowania różnych wariantów danego obrazu - z różnym stylem (wzmocnienie danych - *data augmentation*),
- Zmusi to sieć do patrzenia nie tylko na teksturę obrazu.

Źródło: R. Geirhos et al., *ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness*. ICLR 2019

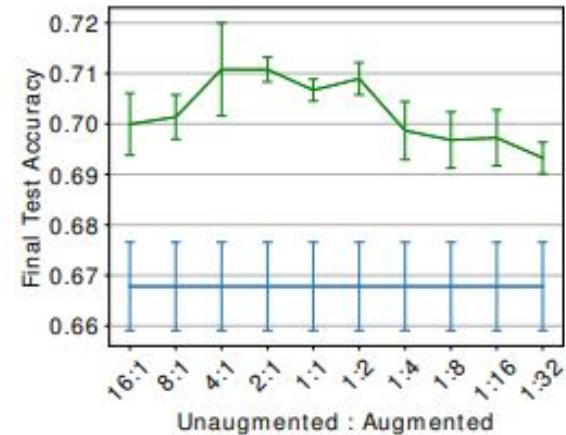
Transfer stylu jako wzmacnienie danych



- przykład syntetycznie wygenerowanych obrazów (za pomocą różnych stylów),
- tak wygenerowane obrazy można użyć do treningu sieci.

Źródło: R. Geirhos et al., *ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness*. ICLR 2019

Transfer stylu jako wzmacnienie danych



- użycie **samych stylizowanych** zdjęć może jednak pogorszyć dokładność na zwykłym zbiorze testowym,
- dlatego w czasie treningu mieszają stylizowane zdjęcia z oryginalnymi (najczęściej w proporcji 1:1),
- powyższa augmentacja danych zwiększa zdolności generalizacyjne modeli do nowych warunków testowych (zmienne warunki oświetleniowe, odmienne tło czy odporność na niewielki szum)

Źródło: P. T. G. Jackson et al., Style Augmentation: Data Augmentation via Style Randomization. CVPR Workshops 2019.

Wzmocnienie danych

Czy stosowanie transferu stylu jest konieczne?

Wzmocnienie danych

Czy stosowanie transferu stylu jest konieczne?

Okazuje się, że podobne rezultaty można osiągnąć stosując proste przekształcenia: zniekształcenia koloru, dodawanie szumu, zmiana kontrastu (jak na rysunku obok)



Źródło: S. Cygert et al., "Closer Look at the Uncertainty Estimation in Semantic Segmentation under Distributional Shift", IJCNN 2021

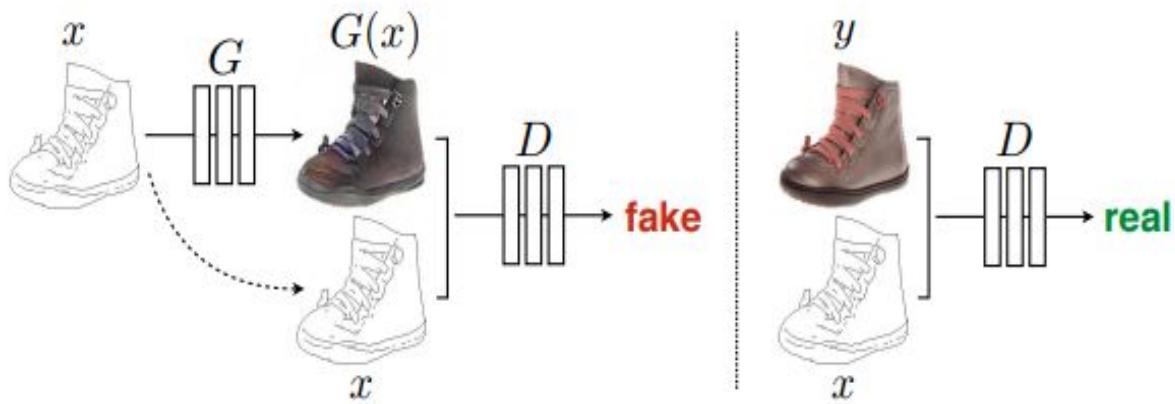
Transfer stylu

- opisany do tej pory transfer stylu korzystał z funkcji kosztu związanej z przetwarzaniem obrazu (macierz Grama),
- w rezultacie opisana metoda ma **ograniczone zastosowanie** i nie może być stosowana w innych zagadnieniach związanych z przetwarzaniem obrazu (np. jego kolorowanie) czy przetwarzaniem mowy,
- **zastosowanie jakiego rodzaju sieci pozwoliłoby na uzyskanie bardziej uniwersalnej architektury?**

Sieci generatywne

- można zastosować sieci typu GAN (ang. *Generative Adversarial Networks* - generatywne sieci współzawodniczące),
- **generator** generuje obraz wejściowy w danym stylu,
- dyskryminator otrzymuje na wejście obraz wzorcowy (ang. *ground-truth*) lub obraz stworzony przez generator. Zadaniem **dyskryminatora** jest ocena czy obraz na wejściu jest “prawdziwy” czy wygenerowany,
- wspólna optymalizacja obu sieci pozwala na generowanie coraz dokładniejszych obrazów w kolejnych iteracjach,
- taki rodzaj architektury może być stosowany dla wielu zadań (również w przetwarzaniu mowy),

Konwersja typu obraz-na-obraz



- G (generator), D (dyskryminator),
- Generator koloruje obraz na podstawie szkicu,
- Dyskryminator otrzymuje na wejście szkic oraz obraz wzorcowy lub wygenerowany przez Generator. Decyduje czy otrzymany obraz jest “prawdziwy”,
- Jako architektura generatora najczęściej stosowane są sieci typu U-Net (używane w segmentacji semantycznej)

Źródło: P. Isola, et al. "Image-to-image translation with conditional adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

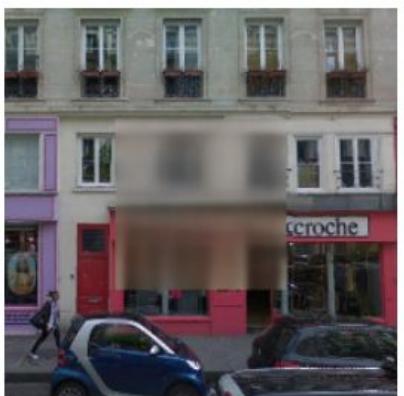
Rekonstrukcja obrazu



(a) Input context



(b) Human artist



(c) Context Encoder
(L2 loss)

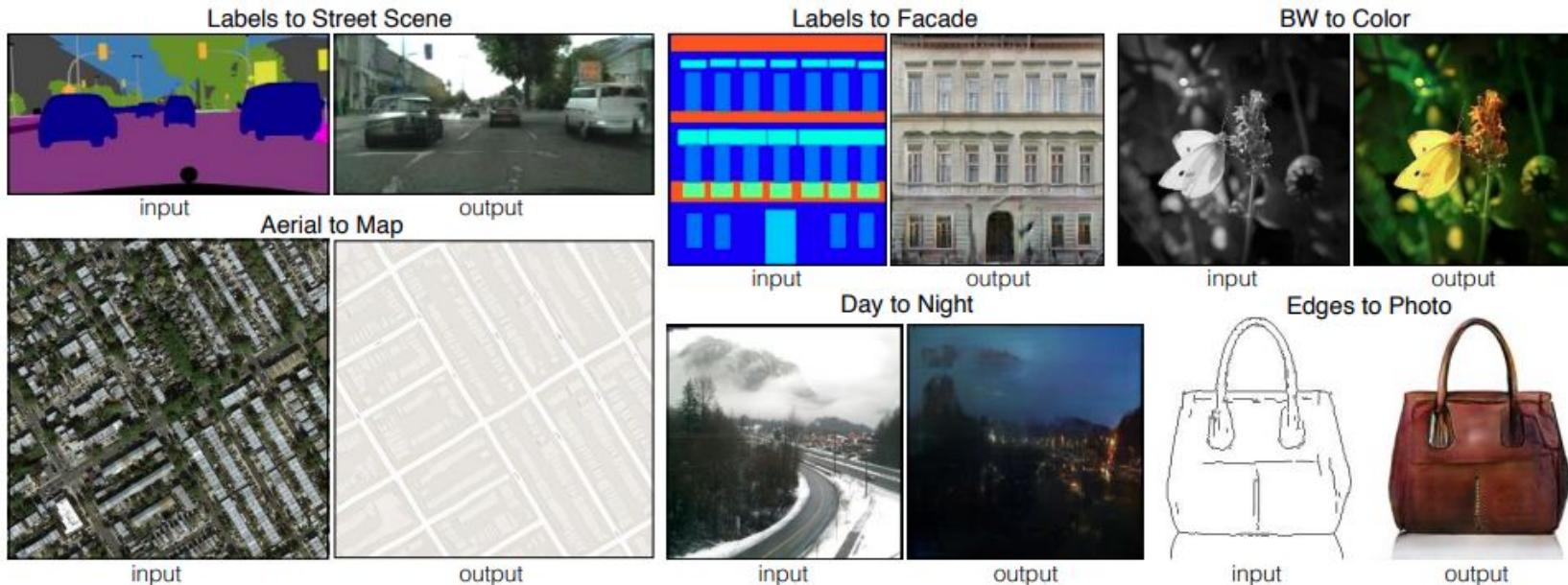


(d) Context Encoder
(L2 + Adversarial loss)

- zadanie: uzupełnienie fragmentu obrazu,
- zwykła sieć CNN trenowana funkcją kosztu **L_2 loss** (dystans euklidesowski) generuje rozmazany obraz,
- zastosowanie funkcji kosztu związanej z sieciami GAN (**adversarial loss**) pozwala na bardziej dokładną rekonstrukcję,
- **context encoder** - enkoder obrazu (wektor cech uzyskany z dowolnej sieci CNN)

Źródło: D. Pathak, et al. "Context encoders: Feature learning by inpainting." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Konwersja typu obraz-na-obraz (image-to-image translation)



- *BW to color* (zamiana czarno-białego obrazu na kolorowy),
- *Edges to photo* (kolorowanie szkicu obrazu),
- *Day to Night* (zmiana oświetlenia w obrazie z dziennego na nocne),
- *Aerial to Map* (zamiana zdjęcia satelitarnego na mapę miasta)

Źródło: P. Isola, et al. "Image-to-image translation with conditional adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

Konwersja typu obraz-na-obraz



- input (obraz wejściowy),
- ground truth (obraz wzorcowy),
- L1 - sieć trenowana za pomocą L1 Loss (metryka Manhattan) - **rozmazana** konwersja obrazu,
- cGAN - sieć trenowana za pomocą funkcji kosztu sieci GAN,
- L1 + cGAN - sieć trenowana przy wykorzystaniu obu funkcji kosztu - **najlepsze** rezultaty,

Źródło: P. Isola, et al. "Image-to-image translation with conditional adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

Konwersja typu obraz-na-obraz

- opisana architektura pozwala na różnorodne zastosowania,

Jakie posiada wady?

Źródło: P. Isola, et al. "Image-to-image translation with conditional adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

Konwersja typu obraz-na-obraz

- opisana architektura pozwala na różnorodne zastosowania,

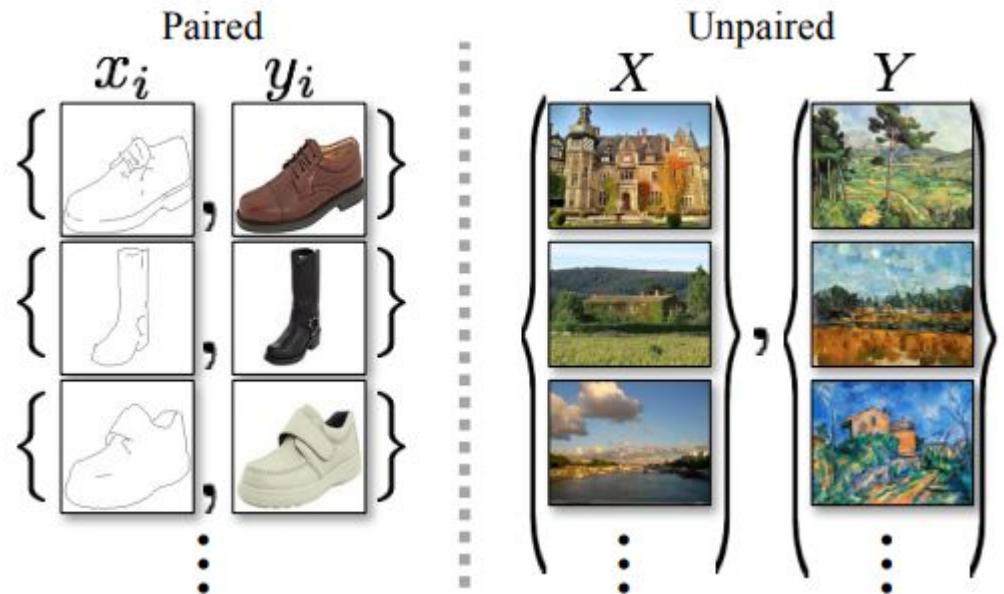
Jakie posiada wady?

- przy obecnej architekturze wymagane są **pary zdjęć** (czyli np. dla danego szkicu posiadamy również jego pokolorowaną wersję, posiadamy zdjęcie danego miejsca zarówno w świetle dziennym jak i nocnym). **Jest to duże ograniczenie w przypadku wielu zastosowań**,

Jak stworzyć model nieposiadający powyższego ograniczenia?

Źródło: P. Isola, et al. "Image-to-image translation with conditional adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

Niesparowana konwersja obrazu-na-obraz



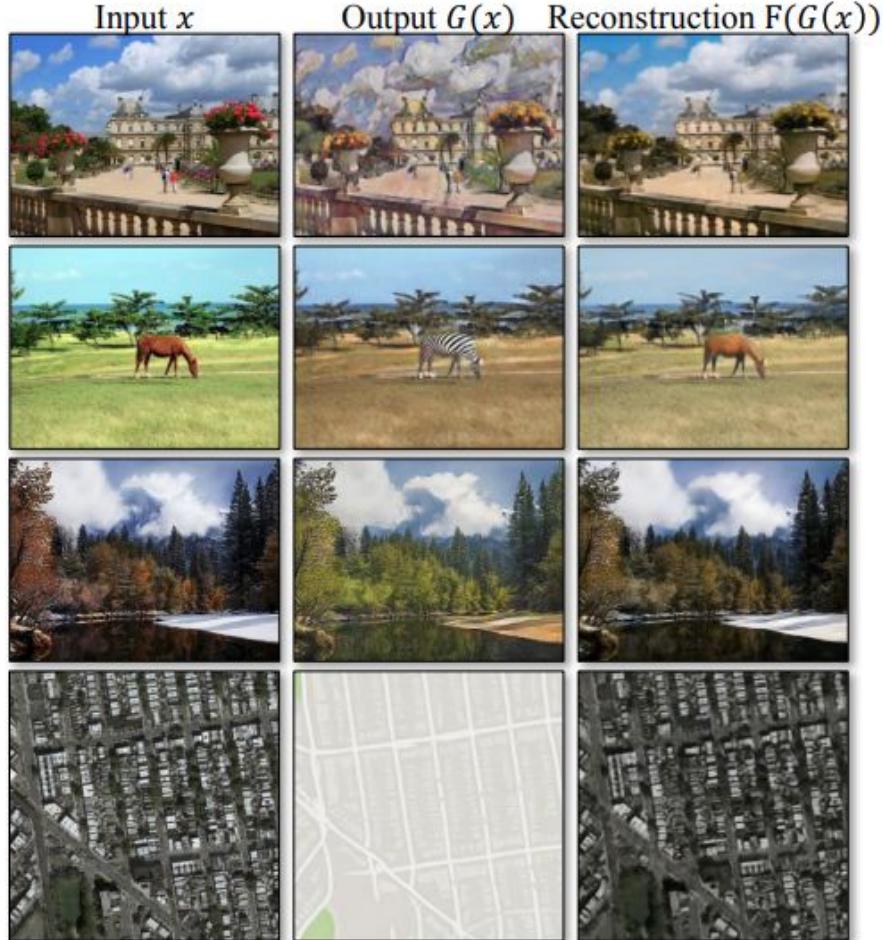
- poprzednia architektura zakładała sparowane dane (pierwsza kolumna),
- w rzeczywistości często dane przez nas posiadane będą niesparowane (druga kolumna)
- X - przykłady zdjęć krajobrazu,
- Y - obrazy w stylu malarstwa,

Źródło: J-Y. Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

Niesparowana konwersja obrazu-na-obraz

- pozwala na **dowolną konwersję obrazu** np. z obrazu w świetle dziennym na obraz w świetle nocnym, nawet gdy posiadamy zdjęcia przy świetle dziennym i nocnym **z różnych lokalizacji**,
- w tym celu zastosowany został tzw. ***cycle-consistency loss*** (funkcja kosztu związana ze spójnością cyklu),
- Generator przekształca obraz x z domeny X na Y (np. z nagrani w świetle dziennym do nagrania w świetle nocnym), a następnie wygenerowany obraz przekształca z powrotem do domeny X otrzymując x' ,
- Podobnie jak w poprzedniej architekturze zadaniem dyskryminatora jest ocena czy otrzymany obraz z domeny Y jest prawdziwy czy wygenerowany,
- Dodatkowo model jest regularyzowany poprzez dodanie funkcji kosztu związanej z rekonstrukcją zapewniając, że obraz x jest możliwie zbliżony do obrazu x' ,

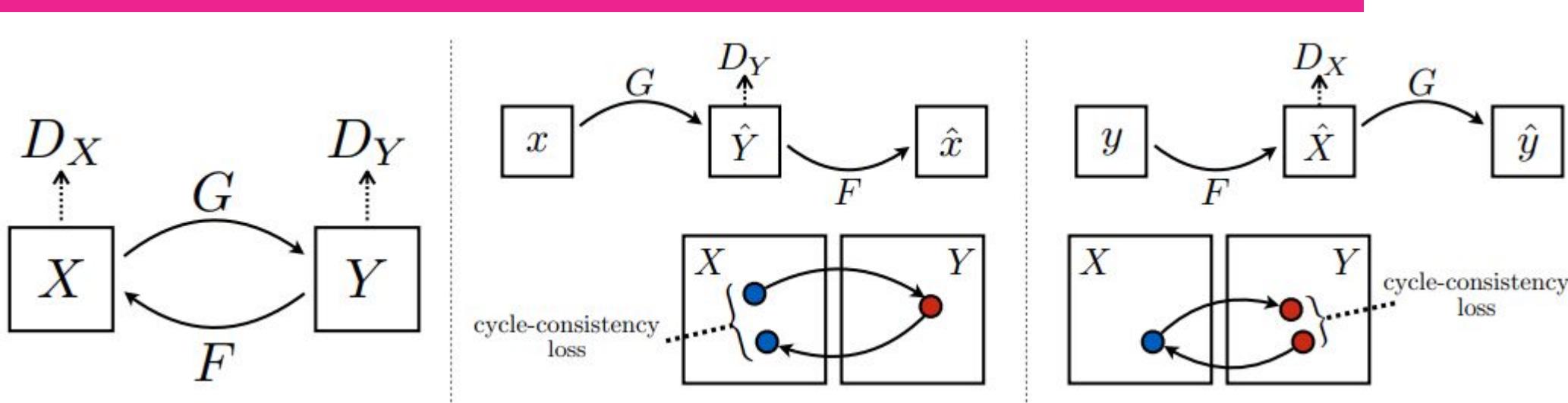
Niesparowana konwersja obrazu-na-obraz



- obraz wejściowy (x),
- obraz wygenerowany w nowej domenie (w tym przypadku styl malarSKI) przez generator G ,
- **rekonstrukcja wsteczna** przy pomocy generatora F (zamieniającego styl malarSKI za fotografię),
- funkcja kosztu związana ze spójnością (***cycle-consistency loss***) zapewnia, że x oraz $F(G(x))$ są do siebie zbliżone - np. dystans Euklidesowski,

Źródło: J-Y. Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

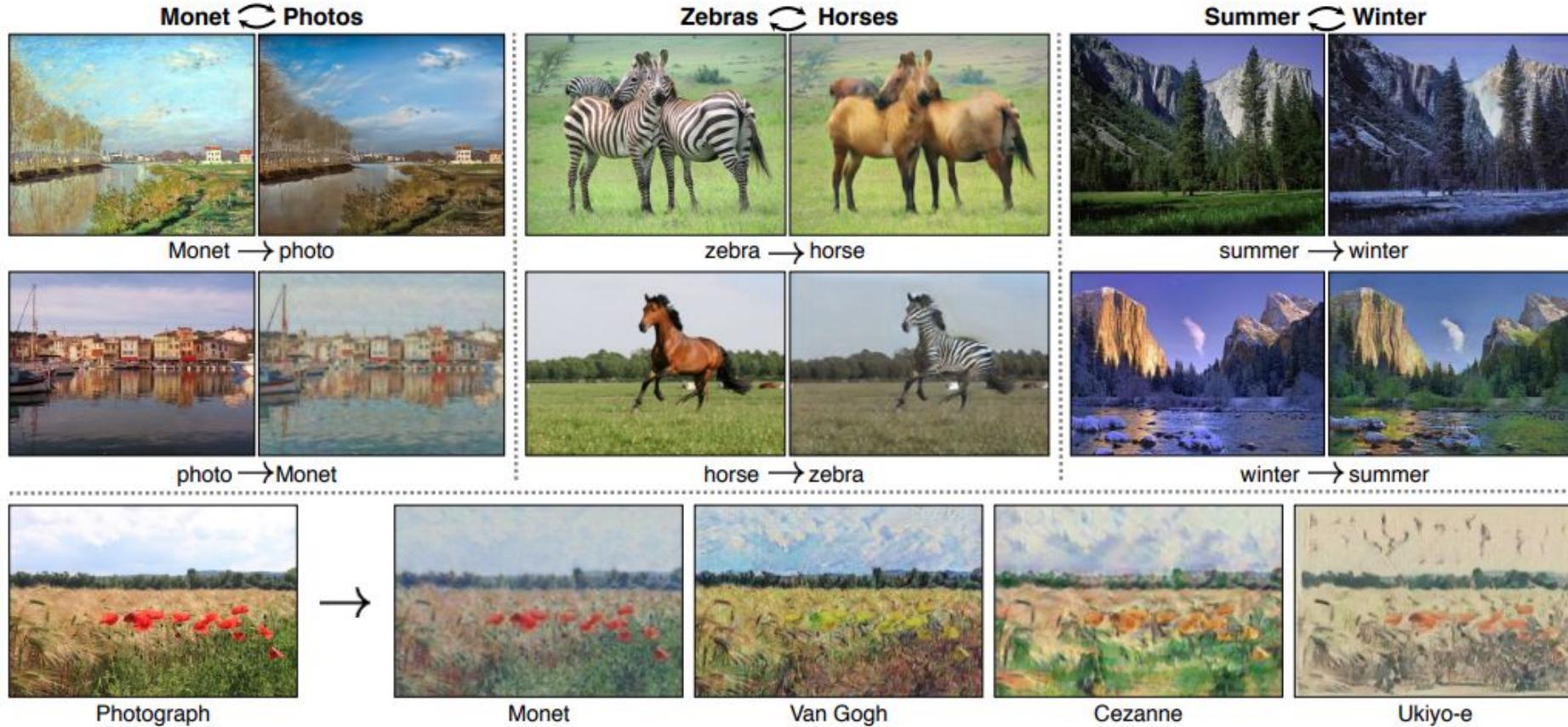
Niesparowana konwersja obrazu-na-obraz



- generator G przekształca obraz z domeny X do domeny Y, generator F w przeciwną stronę,
- dyskryminator D_X i D_Y decydują czy otrzymane obrazy są prawdziwe czy wygenerowane,
- dodatkowo wykorzystywany jest funkcja kosztu związana ze spójnością (*cycle-consistency loss*) zapewniająca, że obrazy przekształcone do oryginalnej domeny są zbliżone do obrazu początkowego

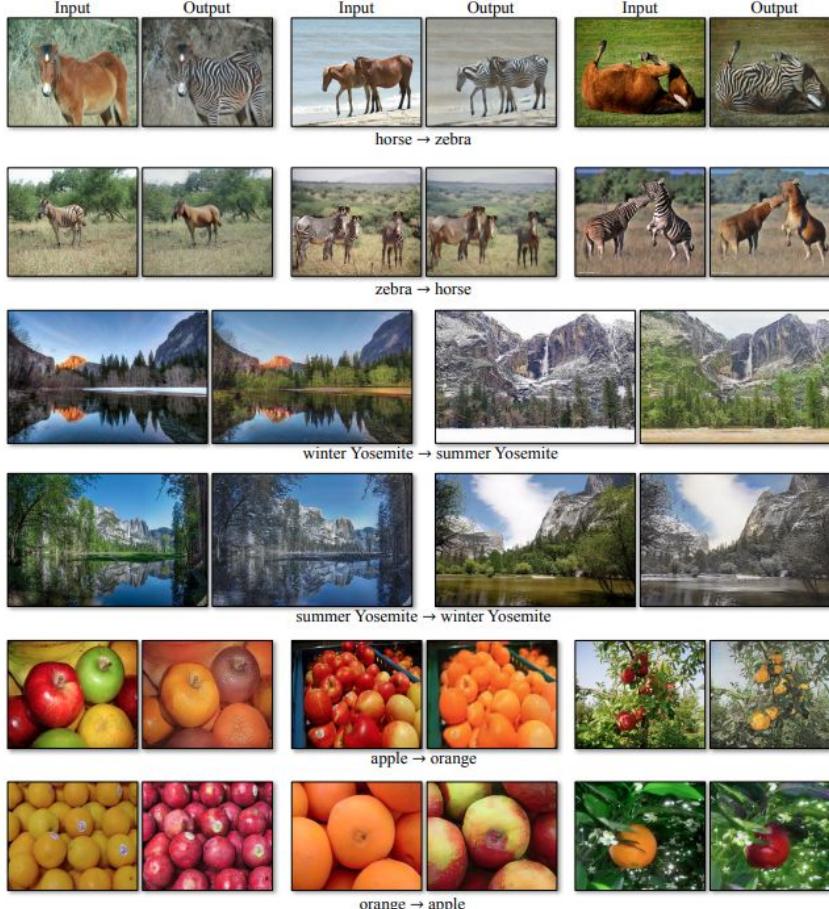
Źródło: J-Y. Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

Niesparowana konwersja obrazu-na-obraz



Źródło: J-Y. Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

Niesparowana konwersja obrazu-na-obraz

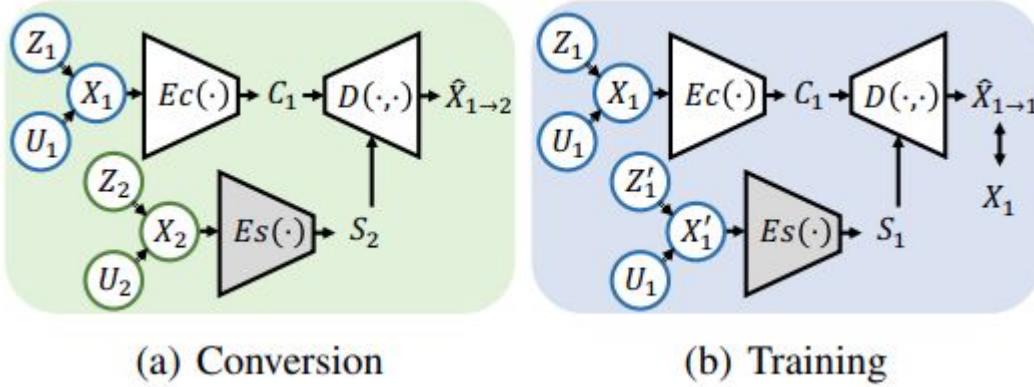


Źródło: J-Y. Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks.", Conference on computer vision and pattern recognition (CVPR), 2017.

Zastosowania w przetwarzaniu mowy

- konwersja głosu (*voice conversion*) - pozwala na generację tekstu za pomocą **dowolnego** (skopiowanego) głosu,
- konwersja stylu. Np. posiadamy próbkę audio gdzie tekst jest wymawiany w neutralny sposób i dokonujemy konwersji na styl bardziej emocjonalny,
- wyzwanie podobne jak przy transferze stylu dla obrazu: jak dla próbki mowy wyekstrachować wektor cech odpowiedzialny za treść (*content*) oraz za styl,

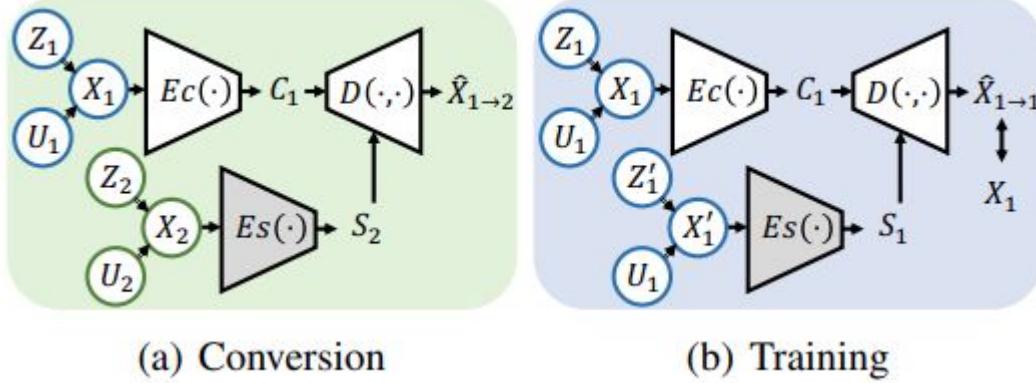
Konwersja głosu - architektura



- Z - zakodowana treść wypowiedzi,
 - U - wektor cech odpowiadający mówcy,
 - E_c (enkoder **treści** wypowiedzi), E_s (enkoder **stylu** mówcy),
 - D - **dekoder** generujący mowę

Źródło: Qian, Kaizhi, et al. "Autovc: Zero-shot voice style transfer with only autoencoder loss." International Conference on Machine Learning. ICML, 2019.

Konwersja głosu - architektura



- W czasie konwersji **wzorcowa wypowiedź** jest zakodowana korzystając z enkodera treści E_c . Równocześnie inna wypowiedź **docelowego mówcy** jest kodowana korzystając z enkoder stylu E_s . Otrzymane wektory są wejściem do dekodera D, który generuje mowę,
- W czasie treningu, ponieważ **nie posiadamy sparowanych danych** (czyli tego samego tekstu wymawianego przez różnych mówców) na wejście do enkodera wykorzystywane są różne wypowiedzi tego samego mówcy,
- **Jak skonstruować enkoder treści i stylu?**

Źródło: Qian, Kaizhi, et al. "Autovc: Zero-shot voice style transfer with only autoencoder loss." International Conference on Machine Learning. ICML, 2019.

Konwersja głosu - architektura

- do **zakodowania stylu (mówcy)** można wykorzystać wytrenowaną (na innym zbiorze danych) sieć do rozpoznawania mówcy,
- kluczem do efektywnej konwersji jest użycie możliwie małego wektora cech dla enkodera treści. **Gdy wektor cech będzie za duży** to możliwe, że **enkoder treści** będzie również kodować informacje na temat mówcy (czego chcemy uniknąć) ponieważ w wygenerowanej mowie usłyszmy dwa różne głosy. **Gdy wektor cech będzie zbyt mały** to niemożliwa będzie poprawna rekonstrukcja mowy,
- w praktyce rozmiar wektora cech dla enkodera treści jest **dobierany eksperymentalnie** tak aby zapewnić odpowiednią rekonstrukcję tekstu i transfer stylu

Konwersja głosu

- opisana architektura działa dobrze gdy mamy odpowiednio dużo danych i ostrożnie dobierzemy rozmiar wektora cech,
- w trudniejszych zastosowaniach praktycznych (gdy np. mamy do czynienia z mową nacechowaną emocjonalnie) powyższe rozwiązania może być niewystarczające,
- styl wypowiedzi charakteryzuje m. in. barwa i ton głosu czy tempo wypowiedzi,
- bardziej zaawansowane architektury wykorzystują wiedzę domenową aby odpowiednio zakodować styl wypowiedzi umożliwiając w ten sposób np. zmianę tempa wypowiedzi [3]

Podsumowanie

- transfer stylu znalazł pierwsze zastosowanie w **stylizacji obrazu**, gdzie styl obrazu został zakodowany za pomocą wiedzy eksperckiej (macierze Grama),
- **zastosowanie sieci generatywnych** pozwoliło na uniwersalną konwersję obrazu w praktycznie dowolnym zastosowaniu (również dla konwersji głosu),
- zastosowanie funkcji kosztu związanej ze spójnością (cycle-consistency loss) pozwoliło na zastosowanie konwersji gdy **nie posiadamy sparowanych danych**,
- sieci generatywne znalazły również zastosowanie w **konwersji głosu**,
- w zastosowaniach konwersji głosu często korzysta się również z **wiedzy eksperckiej** w celu poprawy jakości konwersji,

Dodatkowe materiały

- [1] dlaczego sieć VGG działa najlepiej dla transferu stylu:
<https://distill.pub/2019/advex-bugs-discussion/response-4/>,
- [2] augmentacja danych transferem stylu:
H. Lin et al., "What Can Style Transfer and Paintings Do for Model Robustness?", CVPR 2021,
- [3] demo konwersji głosu z artykułu K. Qian, et al. "Unsupervised speech decomposition via triple information bottleneck." International Conference on Machine Learning (ICML) 2020,
<https://auspicious3000.github.io/SpeechSplit-Demo/>,

Dziękuję

Sebastian Cygert



Fundusze
Europejskie
Polska Cyfrowa



Rzeczpospolita
Polska



Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego

Program Operacyjny Polska Cyfrowa na lata 2014-2020.

Oś priorytetowa nr 3 „Cyfrowe kompetencje społeczeństwa”, działanie nr 3.2 „Innowacyjne rozwiązania na rzecz aktywizacji cyfrowej”.

Tytuł projektu: „Akademia Innowacyjnych Zastosowań Technologii Cyfrowych (AI Tech)”.