Autoreferat

1 Imię i nazwisko

Jan Kazimierz Chorowski

2 Posiadane dyplomy, stopnie naukowe/ artystyczne – z podaniem nazwy, miejsca i roku ich uzyskania oraz tytuł rozprawy doktorskiej

Grudzień 2012, Doktor nauk technicznych, University of Louisville, Louisville, Kentucky, USA pod przewodnictwem prof. Jacka Żurady. Tytuł rozprawy doktorskiej: „Learning Understandable Classifier Models"

Lipiec 2009, magister inżynier Elektroniki i Telekomunikacji, Wydział Elektroniki Mikrosystemów i Fotoniki, Politechnika Wrocławska

3 Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych/ artystycznych

Od października 2013 roku pracuję jako adiunkt na Wydziale Matematyki i Informatyki Uniwersytetu Wrocławskiego. Odbyłem również następujące staże i wizyty naukowe:

- W 2011 r. byłem na 3 miesięcznych praktykach w Microsoft Research Redmond, gdzie pracowałem nad zastosowaniem głębokich sieci neuronowych do rozpoznawania twarzy i gestów.
- W 2014 r. przebywałem przez 3 miesiące na Uniwersytecie Montrelskim w grupie prof. Bengio gdzie rozpocząłem badania nad głębokimi sieciami neuronowycmi wykorzystującymi mechanizm uwagi do rozpoznawania mowy.
- W latach 2016-2018 pracowałem przez jako "Visiting Scholar"w dziale badawczym Google Brain. Przez pierwsze 6 miesięcy byłem w Mountain View w Kalifornii, następnie kontynuowałem badania przebywając we Wrocławiu. W Google Brain pracowałem nad zastosowaniami głębokich sieci do nadzorowanego i nienadzorowanego rozpoznawania mowy i tłumaczenia mowy.

Ponadto od czerwca do sierpnia 2019 będę kierował badaniami dotyczącymi nienadzorowanego rozpoznawania mowy podczas warsztatów JSALT 2019 organizowanych przez Johns Hopkins University[1].

4 Wskazanie osiągnięcia naukowego

4.1 Tytuł osiągnięcia naukowego/artystycznego

Zastosowanie głębokich sieci neuronowych do rozpoznawania i przetwarzania mowy.

[1] Distant supervision for representation learning in speech and handwriting, `https://www.clsp.jhu.edu/workshops/19-workshop/`

4.2 (autor/autorzy, tytuł/tytuły publikacji, rok wydania, nazwa wydawnictwa, recenzenci wydawniczy)

[Hab1] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-Based Models for Speech Recognition. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 577–585. Curran Associates, Inc., 2015.

[Hab2] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4945–4949, March 2016.

[Hab3] Jan Chorowski and Navdeep Jaitly. Towards Better Decoding and Language Model Integration in Sequence to Sequence Models. pages 523–527. ISCA, August 2017.

[Hab4] Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. Sequence-to-Sequence Models Can Directly Translate Foreign Speech. pages 2625–2629. ISCA, August 2017.

[Hab5] J. Chorowski, R. J. Weiss, R. A. Saurous, and S. Bengio. On Using Backpropagation for Speech Texture Generation and Voice Conversion. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2256–2260, April 2018.

Mój indywidualny wkład w wymienione publikacje ze współautorami:

- [Hab1]: Mój udział polegał na pierwszym zastosowaniu sieci neuronowych wykorzystujących mechanizm uwagi do rozpoznawania mowy, zaproponowaniu i wykonaniu części eksperymentów oraz wprowadzeniu mechanizmu uwagi czułego na położenie oraz analizie błędów powstających podczas dekodowania długich nagrań.
- [Hab2]: Mój udział polegał na opracowaniu koncepcji integracji z modelami językowymi, przygotowaniu danych i modeli językowych, wspólnym ze współautorami wykonaniu eksperymentów i doborze parametrów.
- [Hab3]: Mój udział polegał na diagnozie problemów towarzyszących integracji modeli wykorzystujących mechanizm uwagi i zewnętrznych modeli językowych oraz zaprojektowaniu i wykonaniu eksperymentów ustalonych ze współautorami.
- [Hab4]: Mój udział polegał na zaimplementowaniu i wykonaniu eksperymentów dotyczących uczenia wielozadaniowego.
- [Hab5]: Mój udział polegał na opracowaniu zadania, przygotowaniu eksperymentów, analizie wyników i przygotowaniu tekstu publikacji.

4.3 Omówienie celu naukowego/artystycznego ww. pracy/prac i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania.

Rozwój sztucznych sieci neuronowych był stymulowany zadaniami percepcji, w szczególności zadaniami rozpoznawania obrazów i rozpoznawania mowy. Pierwsze sukcesy w zastosowaniach sztucznych sieci neuronowych przyniosło wprowadzenie w latach dziewięćdziesiątych sieci splotowych, zainspirowanych prostymi i złożonymi komórkami kory mózgowej [32], oraz zastosowanie ich do rozpoznawania pisma odręcznego [43] [42]. Osiągnięte w ostatniej dekadzie techniki uczenia głębokich sieci [30] [44] zaowocowały prawdziwym rozkwitem zastosowań przemysłowych. Techniki uczenia głębokiego doprowadziły do przełomowych zmian w rozpoznawaniu mowy [29], [P2], syntezie mowy [55] rozpoznawaniu obrazów [39], [14], generowaniu obrazów [20] [38] i przetwarzaniu języka naturalnego [69] [3] [P14].

Moje prace koncentrują się na stosowaniu głębokich sieci neuronowych do przetwarzania mowy. Główny nurt prac opisuje rozwój modeli rozpoznawania mowy, w których elementy klasyczne (ukryte łańcuchy Markowa, rozkłady mieszanin Gausowskich) zostały zastąpione ich neuronowymi odpowiednikami [P4] [Hab1] [Hab2] [Hab3]. Rozważane przeze mnie modele nie wymagają stosowania ręcznie tworzonych

leksykonów opisujących reguły wymawiania słów, umożliwiają uczenie na transkrypcjach całych zdań, które nie muszą być podzielone na słowa lub na fonemy i umożliwiają łączenie z dodatkowymi zasobami takimi jak modele językowe.

Ważną cechą modeli neuronowych jest ich uniwersalność. Często sieci danego typu mogą być stosowane dla wielu powiązanych zadań. W pracy [Hab4] wykorzystujemy podobieństwo sieci neuronowych służących do rozpoznawania mowy do sieci stosowanych w tłumaczeniu maszynowym proponując model umożliwiający bezpośrednie tłumaczenie mowy na tekst w innym języku. Natomiast w pracy [P14] stosujemy elementy sieci neuronowych rozpoznających mowę do zadań przetwarzania języka naturalnego: tagowania i parsowania.

Sieci neuronowe pozwalają również na przenoszenie technik między mniej powiązanymi zadaniami. W pracy [Hab5] dostosowuję metody syntezy tekstur i zmiany stylu obrazów do edycji głosu mówcy, oraz do określenia w których warstwach sieci następuje rozdzielenie treści wypowiedzi od cech mówcy. Natomiast w pracy [P11] pokazujemy, że stosowane przez mnie techniki regularyzacji sieci służących do rozpoznawania mowy dają dobre wyniki również w zadaniach przetwarzania języka naturalnego i rozpoznawania obrazów.

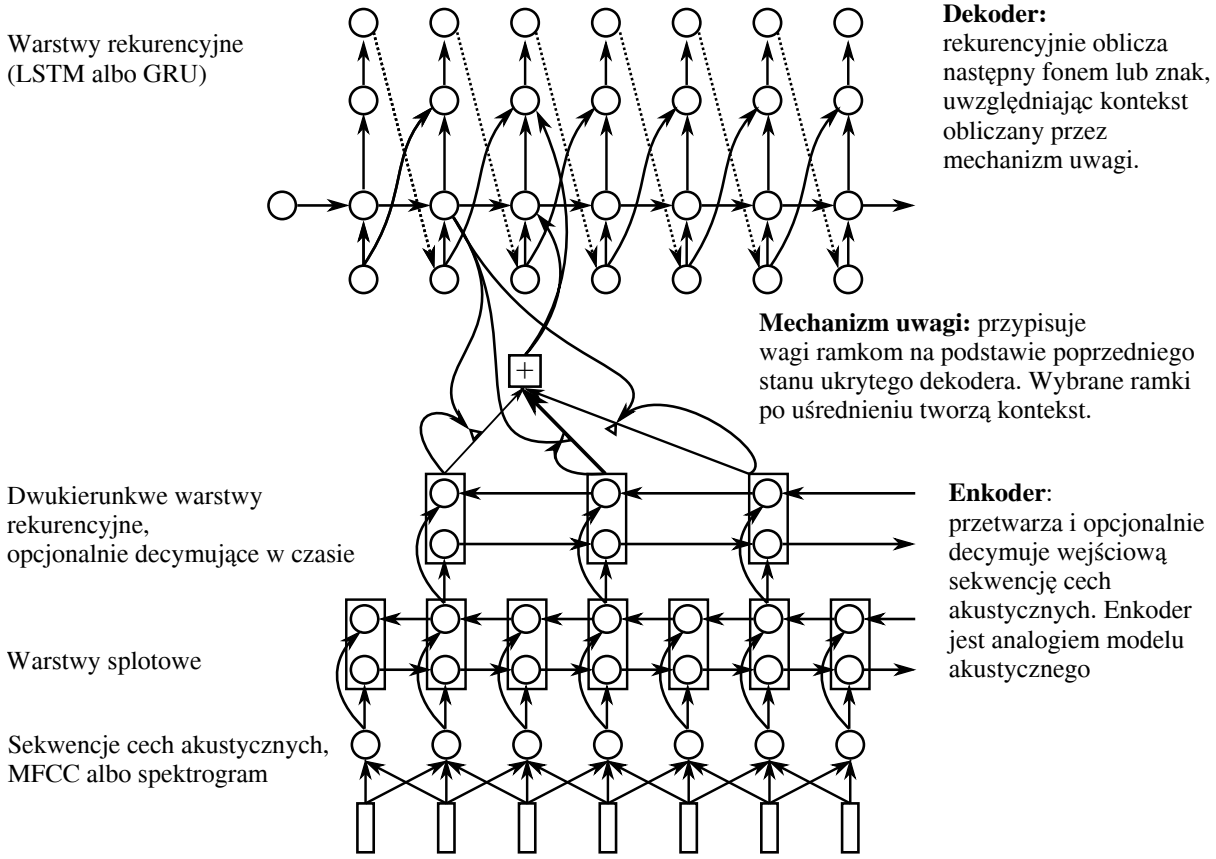
Obecnie pracuję nad metodami nienadzorowanego rozpoznawania mowy mającymi na celu odkrywanie reprezentacji dźwięków maksymalnie zachowującej treść nagrania, a odrzucającej cechy związane z mówcą i stylem wypowiedzi [P6]. Wstępne wyniki wykazują, że podejście to jest dokładniejsze od klasycznie stosowanych technik takich, jak np. ukryte rozkłady Markowa (HMMy) z rozkładem Dirichleta [13] [45].

4.3.1 Nowoczesne modele rozpoznawania mowy stosujące mechanizm uwagi

Zadanie rozpoznawania mowy polega na określeniu dla danego nagrania ciągu słów stanowiących jego transkrypcję. Główną trudnością jest znalezienie wyrównania (tj. dopasowania w czasie) między nagraniem a tekstem. Gdyby wyrównanie było znane, zagadnienie rozpoznawania mowy redukowałoby się do typowego uczenia nadzorowanego. Ponieważ jednak nie znamy wyrównania, modele rozpoznawania mowy muszą zawierać komponent umożliwiający dopasowywanie w czasie każdej transkrypcji do jej nagrania. W klasycznych modelach rolę tę pełnią ukryte łańcuchy Markowa (HMM) [60] [16]. Wprawdzie możliwe jest tworzenie systemów hybrydowych, łączących HMMy z sieciami neuronowymi [7] [6], [25], [68], jednak w mojej pracy skupiłem się na podejściu bardziej radykalnym, w którym HMM zastąpiony jest neuronowym mechanizmem uwagi.

Mechanizm uwagi został opracowany na potrzeby neuronowych systemów tłumaczących [4], w których znajduje on odpowiadające sobie fragmenty tekstu w dwóch językach. W moich badaniach dostosowałem sieci tłumaczące wykorzystujące mechanizm uwagi do potrzeb rozpoznawania mowy i przedstawiłem pierwsze wyniki osiągane za pomocą takich modeli na danych TIMIT [P4], [Hab1] i Wall Street Journal [Hab2]. W tym celu musiałem zmodyfikować mechanizm uwagi tak, aby poprawnie działał na znacznie dłuższych i bardziej zaszumionych sekwencjach cech akustycznych. Z tego powodu wprowadziłem *czuły na położenie mechanizm uwagi* umożliwiający sieci śledzenie transkrybowanego fragmentu nagrania [Hab1]. Eksperymentalnie stwierdziłem również, że sieci neuronowe wykorzystujące mechanizm uwagi mogą być bezpośrednio uczone na parach nagrań i transkrypcji za pomocą typowego algorytmu wstecznej propagacji błędów, bez konieczności inicjalizowania wyrównań modelem klasycznym.

Sieci z mechanizmem uwagi rozwiązują wiele problemów spotykanych w praktyce. Klasyczne modele rozpoznawania mowy zakładają monotoniczność tworzonych wyrównań, czyli wymuszają, aby dalsze fragmenty transkrypcji odpowiadały późniejszym fragmentom nagrania. W przeciwieństwie do nich, sieci wykorzystujące czuły na położenie mechanizm uwagi promują monotoniczność transkrypcji, ale dopuszczają drobne zmiany kolejności. Jest to istotne w zastosowaniach praktycznych, gdyż bardzo często konwencje wypowiadania dat, walut lub skrótowców nie odpowiadają ich zapisowi. Przykładowo po angielsku „$30" czytamy *„thirty dollars"* a datę „3/27/1986" przeczytamy *„twenty seventh of march nineteen eighty six"*. Podobnych konwencji jest bardzo wiele i są one niespójnie stosowane w mowie spontanicznej. Poprawne obsłużenie takich fraz w systemie klasycznym wymaga ręcznego stworzenia rozbudowanych słowników i reguł normalizujących wypowiedzi. Natomiast sieci z mechanizmem uwagi potrafią samoistnie nauczyć się konwencji wymowy z korpusu. Na przykład, sieć wyuczona na danych wyszukiwania głosowego Google Voice Search [12] samoczynnie odkryła że możliwymi wymowami skrótu „AAA" są *„american automobile association"*, „A A A" jak i z *„triple A"*. Z tego względu stosowanie sieci wykorzystujących mechanizm uwagi jest korzystne w budowanych w laboratoriach przemysłowych systemach wyszukiwania głosowego [P2], [63]. Innym obszarem zastosowań, w którym przydatne okazały



Rysunek 1: Typowa sieć rozpoznająca mowę z wykorzystaniem mechanizmu uwagi. W sieci wyróżniamy trzy bloki funkcjonalne: enkoder, dekoder oraz spajający ich pracę mechanizm uwagi.

się wyniki moich badań jest synteza mowy. Nowoczesne podejścia neuronowe [55] [64] często wykorzystują mechanizm uwagi do „tłumaczenia" między tekstem a informacjami potrzebnymi do syntezy dźwięków przez wokodery. Zastosowanie czułego na położenie mechanizmu uwagi znacznie przyspiesza ich uczenie [64].

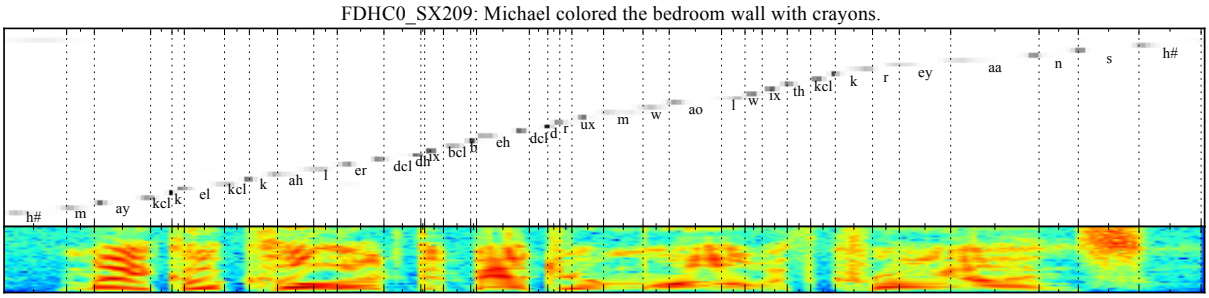
4.3.1.1 Budowa modeli wykorzystujących mechanizm uwagi

Na Rysunku 1 przedstawiono typową sieć rozpoznającą mowę. Sieć wczytuje nagranie przedstawione jako sekwencję wektorów cech akustycznych $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T]$, którymi najczęściej są spektrogramy wyrażone w skali Mela [49]. Wynikiem obliczeń jest określenie prawdopodobieństwa warunkowego $p(\boldsymbol{h}|\boldsymbol{x})$ przypisania sekwencji $\boldsymbol{x}$ transkrypcji $\boldsymbol{h} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_L]$ wyrażonej jako znaki, fonemy lub słowa. Sieć składa się z trzech bloków:

1. *Enkoder* pełni rolę modelu akustycznego. Jego zadaniem jest nieliniowe przekształcenie każdego elementu sekwencji wejściowej $\boldsymbol{x}$ w element reprezentacji ukrytej $\boldsymbol{h} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_{T'}]$. Typowo, enkoder wykorzystuje jednowymiarowe warstwy splotowe [43], warstwy scalające (ang. pooling) które umożliwiają skrócenie reprezentacji ukrytej i dwukierunkowe warstwy rekurencyjne [31] [62].

2. *Mechanizm Uwagi* dopasowuje nagranie i transkrypcję. W tym celu dla każdego emitowanego elementu transkrypcji określa on odpowiadający mu fragment nagrania, który przekazuje dekoderowi.

3. *Dekoder* jest warunkowym modelem językowym [24] obliczającym $p(\boldsymbol{h}_i|\boldsymbol{x}, \boldsymbol{h}_{<i})$. Dekoder najczęściej implementowany jest za pomocą sieci rekurencyjnych. W $i-$tym kroku dekoder łączy informacje o przetworzonym już fragmencie transkrypcji ($\boldsymbol{h}_{<i}$) i wybranym przez mechanizm uwagi fragmencie nagrania.



Rysunek 2: Działanie mechanizmu uwagi. Każdy wiersz w górnym panelu przedstawia ramki wybrane przez mechanizm uwagi służące do emisji kolejnego fonemu. Można zauważyć, że wybory dokonane przez model zlokalizowane są w pobliżu rzeczywistych emisji fonemów (oznaczonych pionowymi liniami). Model jest czuły na położenie i poprawnie rozróżnia między powtórzeniami frazy „kcl-l". Ponadto można zauważyć, że nie wszystkie ramki nagrania zostały użyte przez model. Za [Hab1].

Wszystkie parametry sieci Θ uczone są przez optymalizację funkcji wiarygodności $\mathcal{L}(\Theta) = \log p(\boldsymbol{h}|\boldsymbol{x};\Theta)$ za pomocą algorytmu wstecznej propagacji błędu [61].

Dekoder rekurencyjnie przypisuje prawdopodobieństwo $p(\boldsymbol{h}|\boldsymbol{x})$ możliwym transkrypcjom $\boldsymbol{h}$ nagrania $\boldsymbol{x}$, które rozpisujemy za pomocą reguły łańcuchowej $p(\boldsymbol{h}|\boldsymbol{x}) = \prod_{i=1}^{L} p(\boldsymbol{h}_i|\boldsymbol{x}, \boldsymbol{h}_{<i})$. W każdym kroku dekoder oblicza prawdopodobieństwo warunkowe $p(\boldsymbol{h}_i|\boldsymbol{x}, \boldsymbol{h}_{<i})$, w którym uwarunkowanie na wyemitowanym już fragmencie transkrypcji $\boldsymbol{h}_{<i}$ realizowane jest za pomocą obliczanych rekurencyjnie ukrytych stanów dekodera $\boldsymbol{s}_i$, zaś uwarunkowanie na nagraniu $\boldsymbol{x}$ uzyskiwane jest za pomocą mechanizmu uwagi. W i-tym kroku dekodera mechanizm uwagi porównuje poprzedni stan ukryty $\boldsymbol{s}_{i=1}$ z ramkami nagrania i przypisuje t-tej ramce nagrania jej wagę $\boldsymbol{\alpha}_{i,t}$. Wybrane ramki są uśredniane do wektora kontekstu $\boldsymbol{c}_i$, użytego do obliczenia kolejnego stanu ukrytego $\boldsymbol{s}_i$ oraz prawdopodobieństwa emisji i-tego elementu transkrypcji:

$$\boldsymbol{s}_i = f_R(\boldsymbol{h}_{i-1}, \boldsymbol{s}_{i-1}, \boldsymbol{c}_{i-1}) \tag{1}$$

$$\boldsymbol{\alpha}_i = f_A(\boldsymbol{H}, \boldsymbol{s}_i, \boldsymbol{\alpha}_{i-1}) \tag{2}$$

$$\boldsymbol{c}_i = \sum_{t=1}^{T} \boldsymbol{\alpha}_{i,t} \boldsymbol{h}_t \tag{3}$$

$$p(\boldsymbol{h}_i|\boldsymbol{H}, \boldsymbol{h}_{<i}) = f_E(\boldsymbol{s}_i, \boldsymbol{c}_i), \tag{4}$$

gdzie f_R jest krokiem sieci rekurencyjnej LSTM [31], f_E jest obliczane przez klikuwarstwową sieć neuronową, a f_A przedstawia obliczenia wykonywane przez mechanizm uwagi:

$$e_{i,t} = f_S(\boldsymbol{s}_i, \boldsymbol{\alpha}_{i-1}, \boldsymbol{h}_t) \tag{5}$$

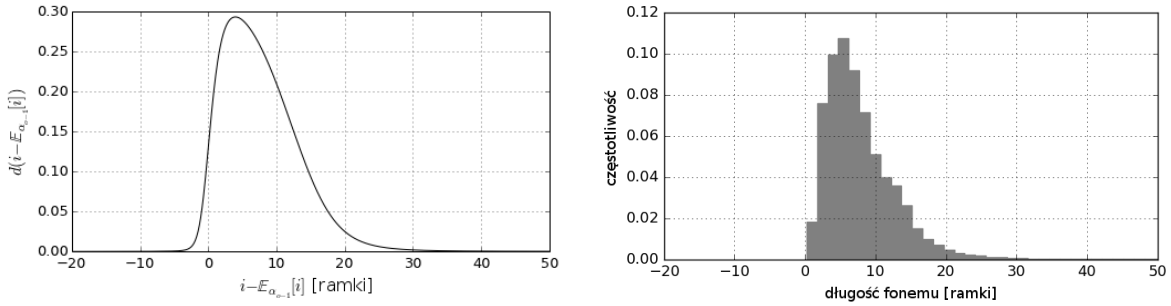
$$\hat{e}_{i,t} = \exp(e_{i,t}) \tag{6}$$

$$\boldsymbol{\alpha}_{i,t} = \frac{\hat{e}_{i,t}}{\sum_{t'} \hat{e}_{i,t'}} \tag{7}$$

$$f_A(\boldsymbol{H}, \boldsymbol{s}_i, \boldsymbol{\alpha}_{i-1}) = \boldsymbol{\alpha}_i, \tag{8}$$

gdzie f_S jest małą siecią neuronową dopasowującą poprzedni stan ukryty dekodera $\boldsymbol{s}_i$ do treści nagrania w ramce $\boldsymbol{h}_t$. Ponadto f_S uwzględnia wagi selekcji z poprzedniego kroku $\boldsymbol{\alpha}_{i-1}$. Rozszerzenie mechanizmu uwagi, stosowanego uprzednio do tłumaczenia maszynowego [4], o mechanizm czuły na położenie wybieranego fragmentu stanowiło kluczowy element dostosowania sieci do potrzeb rozpoznawania mowy [P4] [Hab1].

Działanie mechanizmu uwagi przedstawiono na Rysunku 2. Ukazuje on fragmenty nagrania wybierane przez sieć w kolejnych krokach: i-ty wiersz górnego panelu przedstawia jako obrazek w skali szarości zawartość wektora wag $\boldsymbol{\alpha}_i$. Możemy zaobserwować, że sieć wybiera ramki odpowiadające emitowanym fonemom. Ponadto sieć rozróżnia między powtórzonymi frazami, np. występującej wielokrotnie w nagraniu zbitce „kcl-k" co oznacza, że mechanizm uwagi nie wybiera jedynie ramek o odpowiedniej zawartości, ale również w odpowiednim położeniu.



a) b)

Rysunek 3: a) Funkcja okienkująca $d(\cdot)$ (por. równanie 9) samoistnie wyuczona przez sieć neuronową z czułym na położenie mechanizmem uwagi. b) Histogram czasów trwania fonemów. Za [P4].

W moich pracach przedstawiłem dwie implementacje czułego na położenie mechanizmu uwagi. W pracy [P4] interpretowałem wagi $\boldsymbol{\alpha}_i$ jako rozkład prawdopodobieństwa na ramkach nagrania. Umożliwiło mi to obliczenie oczekiwanego położenia emisji w i-tym kroku, a następnie modulację wyboru ramek w kroku $i+1$ przez funkcje okienkującą $d(\cdot)$ zależną od ich odległość od emisji w kroku i:

$$\hat{e}_{i,t} = d(t - \mathbb{E}_{\boldsymbol{\alpha}_{i-1}}[t]) \exp(e_{i,t}) = d\left(t - \sum_{k=1}^{T} k\boldsymbol{\alpha}_{i-1,k}\right) \exp(e_{i,t}) \tag{9}$$

$$\boldsymbol{\alpha}_{i,t} = \frac{\hat{e}_{i,t}}{\sum_{t'} \hat{e}_{i,t'}} \tag{10}$$

gdzie funkcja $d(\cdot)$ jest obliczana przez małą sieć neuronową. Znalezione przez sieć wartości funkcji $d(\cdot)$ przedstawiono na Rysunku 3a. Można zaobserwować, że model nauczył się monotoniczności wyrównania (funkcja $d(\cdot)$ zabrania wyboru ramek wcześniejszych niż te wybrane w poprzednich krokach), oraz że kształt funkcji $d(\cdot)$ odpowiada w przybliżeniu rozkładowi długości fonemów przedstawionemu na Rysunku 3b.

Uogólniony mechanizm uwagi czuły na położenie przedstawiono w pracy [Hab1]. Funkcję okienkującą $d(\cdot)$ zastąpiono jednowymiarowym splotem poprzednich wag $\boldsymbol{\alpha}_{i-1}$ z wyuczonymi filtrami F:

$$f_{i,t} = F * \boldsymbol{\alpha}_{i-1} \tag{11}$$

$$e_{i,t} = w^T \tanh(W\boldsymbol{s}_i + V\boldsymbol{h}_t + Uf_{i,t}) \tag{12}$$

$$\hat{e_{i,t}} = \exp(e_{i,t}) \tag{13}$$

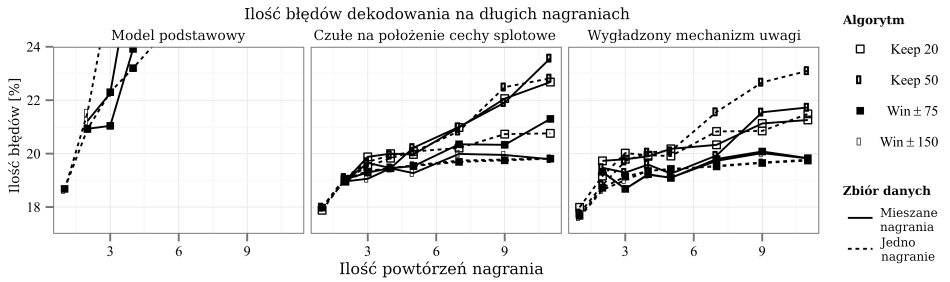
$$\boldsymbol{\alpha}_{i,t} = \frac{\hat{e_{i,t}}}{\sum_{t'} \hat{e_{i,t'}}} \tag{14}$$

Zaletami uogólnionego mechanizmu jest usunięcie założenia że selekcja w każdym kroku daje się dobrze aproksymować przez wybór jednej ramki, od której liczymy odległość, oraz łatwiejsze dostosowanie go do pracy z danymi wielowymiarowymi. Rozszerzenie to okazało się przydatne w dostosowaniu sieci do pracy z długimi nagraniami, opisanymi poniżej.

4.3.1.2 Dostosowanie sieci z mechanizmem uwagi do przetwarzania długich nagrań

Ważnym aspektem stosowania algorytmów uczenia maszynowego, w tym sztucznych sieci neuronowych jest generalizacja rozwiązania znalezionego na danych uczących na dane testowe. W przypadku rozpoznawania mowy oczekujemy ponadto, że model wyuczony na krótkich nagraniach (np. na pojedynczych zdaniach) będzie poprawnie dekodował dłuższe wypowiedzi. Stwierdziłem, że w przypadku sieci z mechanizmem uwagi sieci często nie potrafią transkrybować wypowiedzi znacznie dłuższych od tych w zbiorze uczącym [Hab1]. Zidentyfikowałem dwie przyczyny tego problemu:

1. Brak czułego na położenie mechanizmu uwagi.



Rysunek 4: Spadek jakości dekodowania dla długich nagrań dla modeli wykorzystujących mechanizm uwagi. W panelu po lewej widzimy, że model bez czułego na położenie mechanizmu uwagi nie jest w stanie dekodować sekwencji parokrotnie dłuższych niż te widziane w danych uczących. Dwie modyfikacje mechanizmu uwagi wprowadzone w [Hab1] (cechy splotowe, wygładzanie uwagi) oraz strategie odrzucania odległych ramek („keep", „win") powodują że model jest zdolny do generalizacji na nagrania nawet dziesięciokrotnie dłuższe niż te widziane podczas uczenia. Za [Hab1].

2. Szum wynikający z uwzględnianie wszystkich ramek. Mechanizm uwagi zakłada normalizację wag ramek (7) (10) (14) do czego konieczne jest obliczenie sumy nieujemnych ocen ramek $\hat{e}_{i,t}$. Typowo tylko kilka ramek jest ocenianych wysoko, a pozostałe dostają oceny bliskie zeru. Gdy nagranie jest długie, suma ocen prawie odrzuconych ramek może stać się na tyle duża, że uniemożliwia modelowi wskazanie istotnych ramek.

Rozwiązaniem było wprowadzenie omówionego już mechanizmu czułego na położenie i dodatkowe ograniczenie możliwych do wybrania ramek podczas dekodowania [Hab1]. Zmiany te umożliwiły poprawne dekodowanie wielokrotnie dłuższych nagrań niż dostępne w zbiorze uczącym, co przedstawia Rysunek 4. Widać na nim, że model podstawowy nie jest w stanie dekodować sekwencji jedynie 3 razy dłuższych niż te ze zbioru uczącego. Po dodaniu czułego na położenie mechanizmu uwagi oraz ograniczeniu ilości ramek użytych do normalizacji wag możliwe jest dekodowanie sekwencji nawet dziesięciokrotnie dłuższych niż te widziane podczas uczenia modelu, co nie jest osiągalne przy zastosowaniu zwykłego mechanizmu uwagi.

4.3.1.3 Użycie dodatkowych modeli językowych

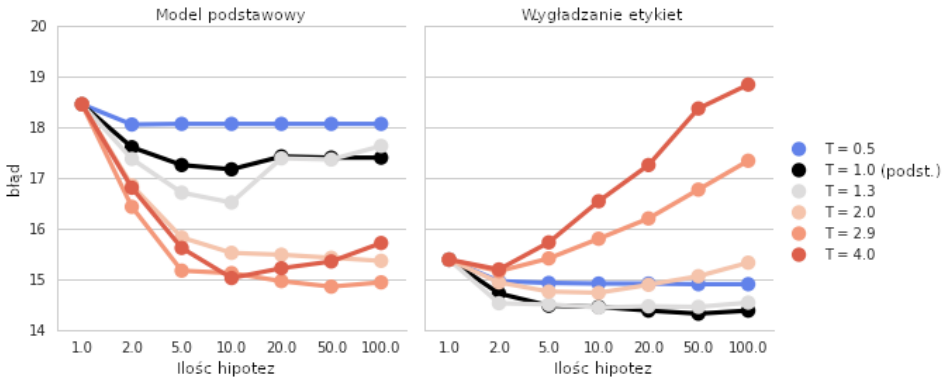
W rozpoznawaniu mowy często stosowane są zewnętrzne modele językowe. Przyczyny ich stosowania są dwojakie. Po pierwsze, często dysponujemy dużą ilością tekstu bez odpowiadających mu nagrań. Po drugie, używane słowa i konstrukcje językowe są dużo bardziej zróżnicowane niż reguły wymawiania słów. W praktyce oznacza to, że adaptacja systemów rozpoznawania mowy wykonywana jest przez dostosowanie modelu językowego, przy zachowaniu modelu akustycznego. Często model językowy jest dostosowywany dynamicznie: np. w wyszukiwaniu głosowym na urządzeniach mobilnych można zwiększyć prawdopodobieństwo rozpoznania nazw własnych miejsc znajdujących się w pobliżu użytkownika. Klasyczne systemy rozpoznawania mowy oddzielnie tworzą model akustyczny i językowy. W sieciach neuronowych wykorzystujących mechanizm uwagi funkcje modelu akustycznego pełni enkoder, a modelu językowego dekoder. Obydwa moduły są uczone łącznie, na tych samych danych. Wykorzystanie dodatkowych informacji językowych lub adaptacja wymaga dołączenia zewnętrznych modeli językowych.

Zagadnienie dekodowania z zewnętrznymi modelami językowymi rozważam w pracach [Hab2] i [Hab3]. Proces znajdowania transkrypcji dla nowego nagrania $\boldsymbol{x}$ (dekodowania) polega na wyznaczeniu najbardziej prawdopodobnej sekwencji

$$\hat{\boldsymbol{h}} = \arg\min_{\boldsymbol{h}} -\log p(\boldsymbol{h}|\boldsymbol{x}). \tag{15}$$

W modelach wykorzystujących mechanizm uwagi minimalizacja (15) realizowana jest najczęściej za pomocą heurystycznych algorytmów poszukiwania, takich jak poszukiwanie promieniste (ang. beam search). Dołączenie zewnętrznych modeli językowych realizowane jest przez rozszerzenie optymalizowanego kosztu o dodatkowe składniki [Hab2] [Hab3]:

$$\hat{\boldsymbol{h}} = \arg\min_{\boldsymbol{h}} -\log p(\boldsymbol{h}|\boldsymbol{x}) - \lambda \log p_{\mathrm{LM}}(y) - \gamma \mathrm{coverage}, \tag{16}$$



Rysunek 5: Wpływ ilości hipotez rozważanych przez poszukiwanie promieniste i temperatury warstwy SoftMax na dokładność dekodowania. W przypadku bazowym (bez wygładzania etykiet) zwiększanie temperatury SoftMaksu zmniejsza ilość błędów. Kiedy użyte jest wygładzanie etykiet poprawia się dokładność rozpoznawania, a dalsze zmiany temperatury SoftMaxu nie przynoszą dodatkowej poprawy. Za [Hab3].

gdzie $\log p(\boldsymbol{h}|\boldsymbol{x})$ jest obliczane przez sieć neuronową, $p_{\mathrm{LM}}(y)$ jest obliczane przez model językowy, a coverage jest opisanym poniżej składnikiem promującym długie transkrypcje.

Jako zewnętrzny model językowy można stosować modele neuronowe [51], albo modele n-gramowe [36]. Stosowanie tych ostatnich może być zrealizowane technikami typowymi dla rozpoznawania mowy: konwersję modelu n-gramowego na skończony transduktor [52], [1] który jest składany z leksykonem rozbijającym słowa na litery [50] [Hab2].

Zidentyfikowałem dwa problemy towarzyszące znajdowaniu transkrypcji z zewnętrznymi modelami językowymi [Hab3]:

1. Nieskalibrowanie rozkładów prawdopodobieństw zwracanych przez sieć neuronową z nieproporcjonalnie dużym prawdopodobieństwem przypisywanym jednej hipotezie;

2. Tendencja sieci do zwracania niepełnych transkrypcji.

Pierwszy problem spowodowany jest przeuczaniem się sieci, które jest typowe dla modeli trenowanych dyskryminatywnie [16], [68]. Aby zapobiec przeuczaniu wprowadzono wiele technik regularyzacji, z których najbardziej popularne to: przerywanie uczenia w momencie zidentyfikowania przeuczenia modelu [9], ograniczanie wielkości wag neuronów przez rozszerzenie funkcji kosztu o sumę kwadratów wag [9], [5], losowe wyłączanie podzbiórów neuronów w sieci (*dropout*) [65], ograniczanie precyzji wag przez dodawanie szumu [23]. Podczas moich badań stwierdziłem że spośród wymienionych technik najbardziej skuteczne jest ograniczanie precyzji wag przez dodawanie szumu [Hab1], [Hab2]. Dołączanie modeli językowych wymusiło jednak opracowanie nowych metod regularyzacji [Hab3].

Skuteczne stosowanie modeli językowych wymaga, aby odpowiedzi sieci umożliwiały wskazanie kilku prawdopodobnych wariantów transkrypcji, spośród których model językowy wybierze jedną. Okazuje się jednak, że bardzo często odpowiedź sieci skupiona jest na tylko jednej możliwości, nie dopuszczając alternatyw. Prawdopodobną przyczyną tego zjawiska jest fakt, że podczas uczenia sieć dosyć szybko osiąga bardzo wysoką dokładność przewidywania kolejnych symboli. Dalsza redukcja błędu $(-\log p(\boldsymbol{h}|\boldsymbol{x};\Theta)$ możliwa jest jedynie przez przypisywanie niemal stuprocentowego prawdopodobieństwa dla przewidywanych odpowiedzi. Powoduje to, że nawet jeśli sieć poprawnie szereguje swoje przewidywania, przypisane im prawdopodobieństwa nie odpowiadają rzeczywistej niepewności odpowiedzi. Zjawisko to można zobrazować analizując wyniki dekodowania modelu, w którym zmieniono jedynie temperaturę T wyjściowej warstwy Softmax [11]:

$$p(y_i) = \frac{\exp(l_i/T)}{\sum_j \exp(l_j/T)}. \tag{17}$$

Dla $T = 1$ otrzymujemy model podstawowy. Dla $T < 1$ rozkład prawdopodobieństwa wyostrza się, zaś dla $T > 1$ rozkład wygładza się, osiągając rozkład jednostajny przy $T \to \infty$. Na lewym panelu Rysunku 5 przedstawiono wpływ temperatury na wyniki dekodowania. Przypominamy, że sieć oblicza

Tablica 1: Wyniki rozpoznawania mowy na popularnych zbiorach danych osiągnięte za pomocą sieci z mechanizmem uwagi (seq2seq) porównane z wynikami CTC – konkurencyjnej techniki czysto neuronowej i systemów hybrydowych DNN + HMM.

TIMIT, 3.14h	PER %	Char. WSJ, 200h	WER %	VS, 12500h	WER %
LSTM + CTC [22]	17.7	LSTM + CTC [50]	7.5	Seq2seq [P2]	5.6
Seq2seq [Hab1]	17.6	Seq2seq [Hab3]	6.7	LSTM + HMM [P2]	6.7
HMM + DNN [67]	16.7	DNN + HMM [27]	3.7		

prawdopodobieństwo warunkowe $p(\boldsymbol{h}|\boldsymbol{x}) = \prod_i p(\boldsymbol{h}_i|\boldsymbol{h}_{<i}, \boldsymbol{x}$. Zmiana temperatury nie zmienia kolejności odpowiedzi sieci w danym kroku, jednak wpływa na całkowite prawdopodobieństwo przypisywane dla transkrypcji y: może się okazać, że wybór mniej prawdopodobnej litery w kroku i pozwoli nam lepiej wybrać w kroku j. Aby umożliwić poszukiwaniu promienistemu znalezienia rozwiązań lepszych niż przeszukiwanie zachłanne trzeba jednak, aby rozkład prawdopodobieństwa obliczany w kroku i dopuszczał kilka wariantów. Jeśli obliczane w i-tym kroku prawdopodobieństwo $\prod_i p(\boldsymbol{h}_i|\boldsymbol{h}_{<i}, \boldsymbol{x}$ jest zbyt wyostrzone, sieć i tak wybierze zachłannie najbardziej prawdopodobny symbol. Widać to na rysunku: dla niskich temperatur, gdy rozkłady zwracane przez sieć są bardzo wyostrzone, sieć nie pozwala poszukiwaniu wybrać symbolu innego niż maksymalnie prawdopodobny i zwiększanie promienia poszukiwania nie przynosi istotnie lepszych efektów. Dla wysokich temperatur, odpowiedzi sieci wygładzają się i sieć dopuszcza kilka wyborów w każdym kroku, co umożliwia znalezienie globalnie lepszych rozwiązań.

W pracy [Hab3] wprowadzam technikę regularyzacji dla danych sekwencyjnych polegającą na lokalnym wygładzaniu etykiet (ang. label smoothing), która przeciwdziała przypisywaniu przez sieć niemal 100% prawdopodobieństwa tylko jednej z możliwych odpowiedzi. Aby tego uniknąć, sieć uczona jest przewidywać z pewnym małym prawdopodobieństwem przewidywać sąsiadujące w sekwencji docelowej z poprawną odpowiedzią. Ma to dwojakie skutki: po pierwsze ogranicza prawdopodobieństwo przypisywane przez model najbardziej prawdopodobnej odpowiedzi. Po drugie, przez przewidywanie z małym prawdopodobieństwem sąsiadujących znaków, sieć dopuszcza wprowadzane przez model językowy zmian, które często znajdują się w małej odległości edycyjnej od jej przewidywań (np. korekcja literówek). Wyniki wygładzania etykiet widać na prawym panelu Rysunku 5. Efekt regularyzacji powoduje poprawę wyników dekodowania zachłannego. Ponadto, zmiany temperatury SoftMaxu pogarszają wyniki dekodowania, sugerując skalibrowanie odpowiedzi sieci.

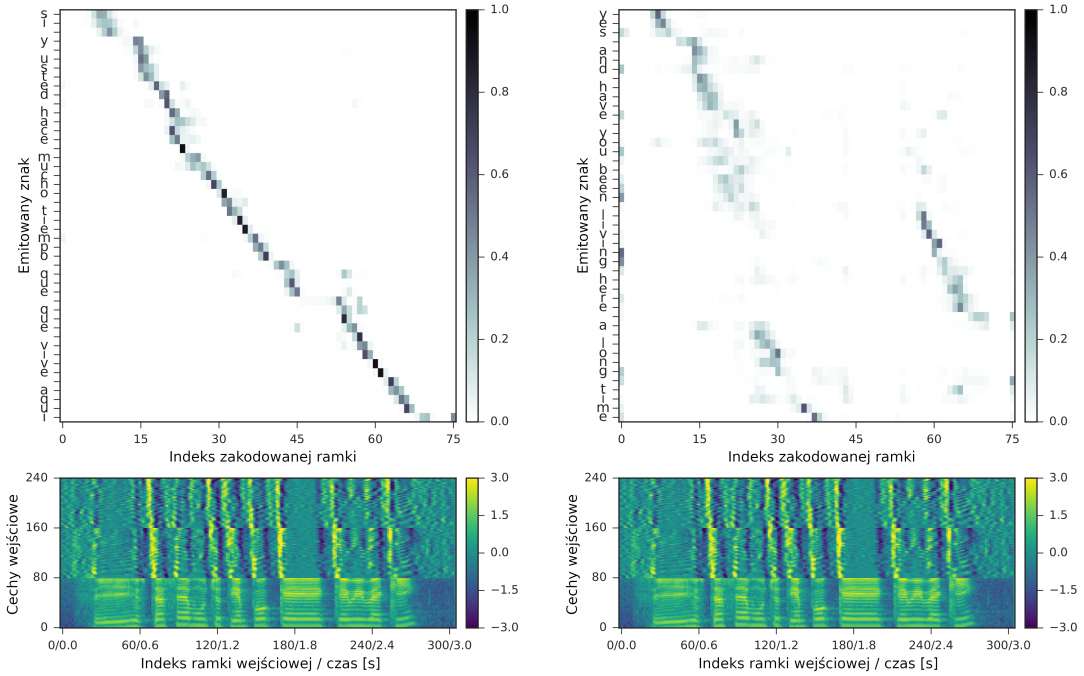
Wygładzanie etykiet może też być stosowane w zwykłym uczeniu nadzorowanym, np. w rozpoznawaniu obrazów [66]. Wraz ze współpracownikami potwierdziliśmy jego skuteczność również w zadaniach przetwarzania języka naturalnego [P11].

Drugi problem podczas dekodowania z zewnętrznymi modelami językowymi polega na gubieniu fragmentów transkrypcji. Należy zwrócić uwagę, że w odróżnieniu od modeli klasycznych mechanizm uwagi nie wymusza sklasyfikowania każdej z ramek nagrania (por. z Rysunkiem 2) i możliwe są pominięcia fragmentów nagrania. Z drugiej strony, każdy znak emitowany przez sieć powiększa koszt minimalizowany podczas dekodowania (15), co powoduje preferowanie przez sieć krótszych transkrypcji. W celu zapobieżenia temu zjawisku zaproponowałem, aby minimalizowany podczas dekodowania koszt rozszerzyć o liczbę ramek wybranych przez mechanizm uwagi, t.j. liczbę ramek dla których sumaryczna waga przypisana przez mechanizm uwagi przekroczyła próg τ : [Hab3]:

$$\text{coverage} = \sum_j \left[\left(\sum_i \boldsymbol{\alpha}_{i,j} \right) > \tau \right]. \tag{18}$$

Kryterium to promuje emisje symboli wyjściowych z jak największej liczby ramek, zapobiega jednak emisji nadmiarowych znaków poprzez wielokrotne wybieranie przez mechanizm uwagi tej samej ramki. Kryterium stanowi ulepszenie stosowanego poprzednio prostego promowania długich transkrypcji [28] [Hab2]. Połączenie techniki wygładzania etykiet i promowania pełnego pokrycia nagrania przez mechanizm uwagi zaowocowało znacznym zmniejszeniem błędów dekodowania i umożliwiło stosowanie silnych zewnętrznych modeli językowych.

W Tabeli 1 zebrano wyniki osiągane przez podejścia czysto neuronowe: CTC [25], sieci wykorzystujące mechanizm uwagi (seq2seq) oraz przez modele hybrydowe na zbiorach TIMIT [17], Wall Street Journal (WSJ) i Voice Search (VS). Dla mniejszych zbiorów danych sieci z mechanizmem uwagi dają wyniki



Rysunek 6: Działanie mechanizmu uwagi w systemie rozpoznającym mowę (lewy panel) i tłumaczącym mowę (prawy panel). W przypadku rozpoznawania mowy mechanizm wybiera kolejne ramki. W przypadku tłumaczenia system zmienia kolejność. W obydwu przypadkach emisje odpowiadających sobie fraz odwołują się do tych samych ramek. Za [Hab4].

zbliżone do CTC, ale gorsze od systemów hybrydowych. Na największym spośród analizowanych zbiorów danych, VoiceSearch, modele czysto neuronowe działają najlepiej. Model z pracy [P2] wykorzystuje wprowadzone przez mnie wygładzanie etykiet [Hab3].

4.3.1.4 Tłumaczenie mowy za pomocą sieci z mechanizmem uwagi

W zadaniu tłumaczenia mowy chcemy połączyć funkcjonalność systemu rozpoznawania mowy z systemem tłumaczącym. Naiwne rozwiązanie tego zadania stosuje dwa odrębne modele. Powoduje to jednak składanie się błędów: system tłumaczący nie ma dostępu do alternatywnych transkrypcji znajdowanych przez model rozpoznawania mowy. Klasyczne systemy tłumaczące umożliwiają częściowe rozwiązanie tego problemu przez przekazywanie wyników rozpoznawania mowy w formie kraty [53], [47], [59]. Nadal jednak obydwa modele uczone są oddzielnie i nie jest możliwa globalna minimalizacja błędu tłumaczenia.

Modele neuronowe umożliwiają bezpośrednie tłumaczenie mowy, realizowane przez jeden model łączący dwie funkcjonalności. Podejście to ma dwie zalety. Po pierwsze, możliwe jest uczenie systemu bez transkrypcji nagrań w ich języku. Znajduje to zastosowanie np. w dokumentowaniu wymierających języków, dla których nie ma nawet powszechnie stosowanych systemów zapisu i łatwiej jest nagrać próbki głosu i ich tłumaczenia [8], [2]. Ponadto model uczony jest łącznie, co powoduje że część sieci utożsamiana z warstwą akustyczną uczy się przygotowania reprezentacji ukrytej maksymalnie usprawniającej działanie części tłumaczącej.

Mimo, że model nie wymaga transkrypcji w języku oryginalnym, często są one dostępne i chcielibyśmy móc je wykorzystać podczas uczenia sieci. Neuronowy model monolityczny nie umożliwia tego bezpośrednio. Wykorzystanie wszystkich dostępnych danych jest jednak możliwe przez zastosowanie uczenia wielozadaniowego, w którym uczymy trzy modele: rozpoznawania mowy, tłumaczenia mowy i tłumaczenia tekstu. Modele te nie są jednak niezależne, ale współdzielą swoje części: rozpoznawanie i tłumaczenie mowy współdzieli utożsamiany z modelem akustycznym enkoder, zaś tłumaczenie mowy i tekstu współdzielą utożsamiany z modelem językowym dekoder. Rozwiązanie to sprawdziliśmy w pracy [Hab4], w której porównaliśmy dokładność kaskadowanego systemu rozpoznawania mowy i tłumaczenia,

Tablica 2: Ocena jakości BLEU systemów tłumaczących mowę. Monolityczny neuronowy system uczony łącznie na zadaniach tłumaczenia i rozpoznawania mowy osiąga najlepsze (najwyższe BLEU) wyniki. Za [Hab4].

	Fisher			Callhome	
Model	dev	dev2	test	devtest	evltest
End-to-end ST	46.5	47.3	47.3	16.4	16.6
Multi-task ST / ASR	48.3	49.1	48.7	16.8	17.4
ASR $\rightarrow$ NMT cascade	45.1	46.1	45.5	16.2	16.6
Post et al. [59]	–	35.4	–	–	11.7
Kumar et al. [41]	–	40.1	40.4	–	–

zintegrowanego modelu neuronowego uczonego tylko na zadaniu tłumaczenia mowy i wielozadaniowego modelu neuronowego.

Wyniki systemu przedstawiono jakościowo na Rysunku 6 i zebrano w Tabeli 2. Najlepszą jakość tłumaczenia w sensie metryki BLEU [57] osiągnięto dla wielokryterialnego systemu, w którym współdzielono enkoder modeli rozpoznających i tłumaczących mowę. Przewyższa ono nie tylko wyniki osiągane metodami klasycznymi, ale również kaskadą bardzo dokładnych głębokich modeli rozpoznawania mowy i tłumaczenia. Działanie systemu można zrozumieć analizując ramki wybierane przez mechanizm uwagi. W przypadku rozpoznawania mowy selekcje wykonywane przez mechanizm uwagi postępują monotonicznie i są skupione na pojedynczych ramkach. System tłumaczący mowę dokonuje zamiany kolejności emisji, jednak wybór ramek jest nieprzypadkowy: emisja angielskiej frazy „living here" używa ramek dźwięku odpowiadających hiszpańskiemu „vive aqui". Ponadto możemy zauważyć, że model tłumaczący mowę podczas emisji końców słów często wybiera pierwszą ramkę nagrania. Pokazuje to siłę wyrazu rekurencyjnych sieci neuronowych. Enkoder potrafi połączyć dźwięki w sąsiadujących ramkach w reprezentację słowa, dekoder zaś odwołuje się do nagrania jedynie aby wybrać następne emitowane słowo. Kiedy zostaje ono ustalone, emisja następuje „z pamięci", a mechanizm uwagi wybiera nieinformatywną pierwsza ramkę.

4.3.2 Przetwarzanie mowy za pomocą sieci neuronowych

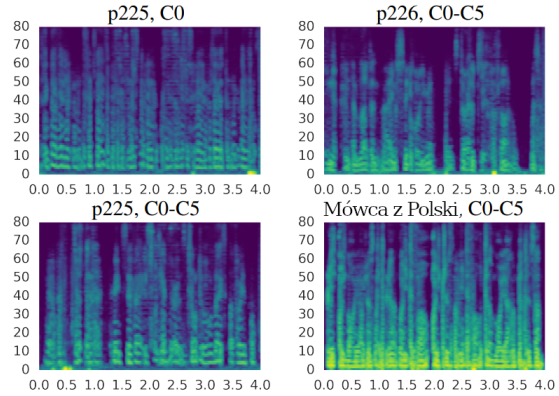
Powyżej opisałem jak za pomocą sieci neuronowych można zrealizować system rozpoznawania mowy. Czy jest możliwe odwrócenie działania systemu i odtworzenie nagrań na podstawie wartości obliczanych przez sieć? Na to pytanie odpowiadam w pracy [Hab5], w której zamieniam sieć rozpoznającą mowę w system syntezujący tekstury dźwiękowe oraz umożliwiający zamianę mówcy nagrania. Ponadto techniki odwracania sieci umożliwiły mi określenie, w których warstwach sieć uczy się niezmienności na cechy mówców.

Dokładne odwrócenie dyskryminatywnie uczonej sieci neuronowej jest niemożliwe. Można jednak, za pomocą optymalizacji gradientowej, szukać wejść (nagrań) aktywujących sieć w określony sposób. Podejście to było dotychczas stosowane dla sieci rozpoznających obrazy w celu zrozumienia działania sieci [46], znalezienia przykładów mylących sieć (adversarial examples) [21], syntezy tekstur [18] oraz przenoszenia stylu z jednego obrazu na drugi [19]. W pracy [Hab5] dostosowałem powyższe techniki do analizy sieci rozpoznających mowę.

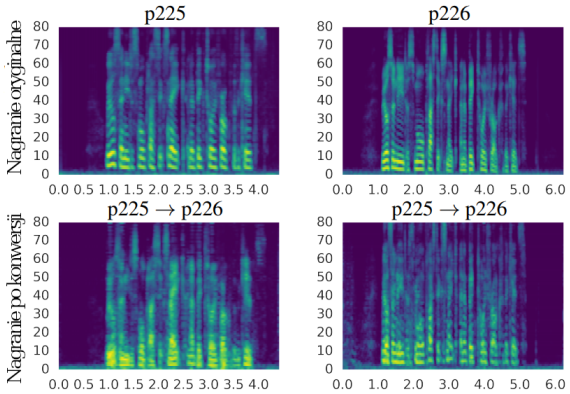
4.3.2.1 Synteza tekstur i przenoszenie stylu przez dopasowywanie statystyk

Według hipotezy Julesza [34], [35] rozróżnianie tekstur polega na analizie niskopoziomowych właściwości statystycznych obrazów, takich jak korelacje między sąsiednimi pikselami. Simoncelli ze współpracownikami [58] [48] wykazali, iż w podobny sposób można rozróżniać między teksturami dźwiękowymi. Ponadto wykazali, że hipotezę Julesza można wykorzystać do generowania tekstur. W tym celu poszukujemy nowych obrazów lub dźwięków które posiadają statystyki zbliżone do prawdziwych obrazów lub dźwięków. Podejście to umożliwiło syntezę prostych tekstur dźwiękowych, takich jak krople deszczu albo ogień. Nie udało się jednak uzyskać przekonywająco brzmiącej mowy ludzkiej.

Gatys et al. rozwinął podejście McDermotta [48] do syntezowania tekstur graficznych przez zastąpienie prostych modeli statystycznych korelacjami między aktywacjami neuronów w sieciach przeznaczonych dla rozpoznawania obrazów. Konkretnie, niech $C^{(n)} \in \mathbb{R}^{W \times H \times D}$ oznacza aktywacje neuronów w n-tej



Rysunek 7: Wyniki syntezy tekstur mowy dla macierzy Grama obliczonych na nagraniach z korpusu VCTK dla głosów kobiecych (lewa kolumna) i męskich (prawa kolumna), odpowiednio na pierwszej (C0) i pierwszych sześciu (C0-C6) warstwach splotowych. Użycie do syntezy głębszych warstw sieci skutkuje uzyskaniem ciekawszej struktury dźwięku. Intuicyjnie, słuchając nagrania „p255, C0" trudno usłyszeć granice słów, które są wyraźnie słyszalne w nagraniu „p255, C0-C5". Można również zauważyć występowanie wyższej częstotliwości bazowej dla głosu kobiecego. Za [Hab5].



Rysunek 8: Wyniki przeniesienia głosu mówców „p225" i „p226" miedzy nagraniami. Zsyntetyzowane nagrania zachowują treść nagrania oryginalnego, ale częstotliwość podstawową docelowego mówcy. Za [Hab5].

warstwie sieci splotowej, gdzie W jest szerokością warstwy, H jest jej wysokością, a D liczbą filtrów. Macierz Grama $G^{(n)} \in \mathbb{R}^{D\times D}$ definiujemy jako:

$$G^{(n)}_{i,j} = \frac{1}{WH}\sum_{w=1}^{W}\sum_{h=1}^{H} C^{(n)}_{whi} C^{(n)}_{whj}. \tag{19}$$

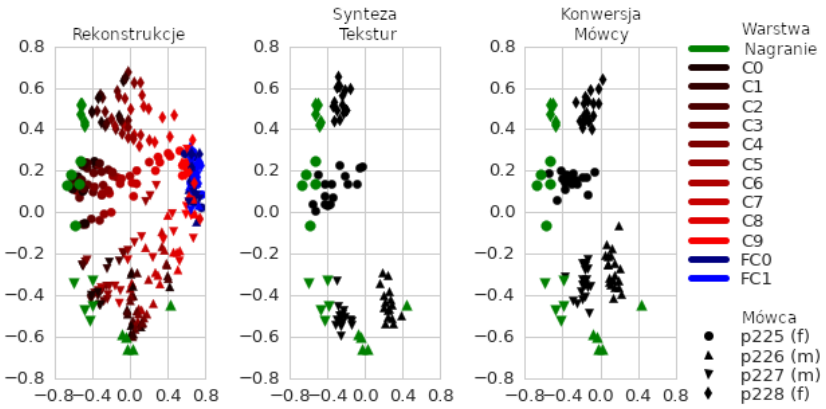
Realistycznie wyglądające tekstury można syntezować przez poszukiwanie za pomocą optymalizacji gradientowej nowych obrazów indukujących podobne macierze Grama. Innymi słowy, macierz Grama można utożsamiać ze statystyką wystarczającą do zdefiniowania tekstury.

Poszukiwanie obrazów o zadanej teksturze, t.j. indukujących daną macierz Grama, można połączyć z poszukiwaniem obrazów dających podobne aktywacje sieci neuronowej macierz. Połączenie tych dwóch kosztów skutkuje technikami przenoszenia stylu [19]. Szukamy w nich obrazu, dla którego aktywacje sieci odpowiadają aktywacjom na wybranym zdjęciu wejściowym, a indukowane macierze Grama odpowiadają macierzom osiąganym dla innego obrazu, którego styl chcemy zachować. Technika ta umożliwia np. stylizację zdjęć przez przypisanie im cech graficznych znanych malarzy takich jak van Gogh czy Monet.

Neuronowe techniki syntezy tekstur i przeniesienia stylu można stosować również dla dźwięku. Intuicyjnie interpretujemy wtedy spektrogram jako obraz, którego osiami są czas i częstotliwość. Dla przetworzonego przez sieć splotową spektrogramu możliwe jest obliczanie macierzy Grama w sposób analogiczny do (19), przy czym korelacje uśredniamy jedynie w czasie. Wynika to z faktu, że statystyki tekstur dźwiękowych są stacjonarne w czasie, ale nie w częstotliwości [Hab5]:

$$G^{(n)}_{ijlk} = \frac{1}{T}\sum_{t=1}^{T} C^{(n)}_{tik} C^{(n)}_{tjl}. \tag{20}$$

Zsyntezowanie tekstury dźwiękowej polega na znalezieniu nagrania indukującego podobną do wzorcowej macierz Grama. Można to uzyskać na dwa sposoby. W pierwszym podejściu ustalamy, że wejściem do sieci jest spektrogram. Dla zadanej macierzy Grama wyznaczamy spektrogram tekstury, który zamieniamy na dźwięk za pomocą procedury Griffin-Lim [26]. W drugim podejściu optymalizujemy bezpośrednio



Rysunek 9: Zanurzenie MDS cech mówcy dla: nagrań odtwarzanych z zaznaczonych warstw sieci (lewy panel), syntezowanych tekstur mowy (środkowy panel) i nagrań konwertowanych na innego mówcę za pomocą technik przeniesienia stylu (prawy panel). W wyższych warstwach sieci informacja o mówcy zanika. Ponadto syntezowane nagrania zachowują cechy docelowego mówcy. Za [Hab5].

próbki nagrania. W tym celu cały potok ekstrakcji cech mowy (okienkowanie, dithering, obliczanie transformaty Fouriera i cech MFCC) implementujemy w sposób umożliwiający obliczanie gradientu za pomocą algorytmu wstecznej propagacji błędów. Zastosowanie obydwu metod umożliwiło mi uzyskanie tekstur mowy [Hab5], rozszerzając wyniki McDermotta [48] i potwierdzając hipotezę Julesza dla tak skomplikowanego sygnału. Wyniki syntezy przedstawione są na Rysunku 7. Można zaobserwować, że sieci generują głosy męskie i żeńskie (częstotliwość podstawowa i jej harmoniczne są niższe dla głosów męskich). Oznacza to, że obliczane wzorem (20) macierze Grama są czułe na mówcę. Ponadto wykorzystanie dolnych warstw sieci skutkuje uzyskaniem dosyć jednorodnych w czasie spektrogramów (np. nagranie „p225, C0"), w których nie można rozróżnić słów. Wzięcie 5 warstw sieci pozwoliło na uzyskanie tekstur, w których słychać fragmenty słów (np. nagranie „p225, C0-C5").

Analogiem zmiany stylu obrazów jest konwersja nagrania na innego mówcę. Przedstawia to Rysunek 8. Można zaobserwować, że konwertowane nagrania zachowują strukturę czasową nagrań oryginalnych, jednak następuje zmiana częstotliwości bazowej na częstotliwość docelowego mówcy. Do konwersji potrzebujemy stosunkowo niewielkich długości nagrań docelowego mówcy, dla wyników na Rysunku 8 użyto jedynie 2 minut nagrań docelowego mówcy.

Odtwarzanie nagrania na podstawie aktywacji sieci pozwoliło mi ponadto na określenie zanikania informacji o mówcy w sieci neuronowej. W tym celu wykorzystałem system wyznaczający wektory charakterystyczne mówców na podstawie próbek dźwięku [10]. Znalezione wektory można umieścić w przestrzeni dwuwymiarowej używając techniki MDS [40] (Rysunek 9). Widać, że informacje o mówcy zanikają około siódmej warstwy splotowej (C7), zsyntezowane tekstury wykazują cechy zbliżone do oryginalnego mówcy, a wektory mówcy uzyskane z konwertowanych nagrań są bliżej wektorów reprezentujących docelowego mówcę.

4.3.2.2 Podsumowanie

W przedstawionym cyklu prac analizowałem stosowanie głębokich sieci neuronowych do przetwarzania nagrań mowy. Głównym wynikiem jest wprowadzenie i udoskonalenie nowej architektury sieci neuronowych rozpoznających mowę. Sieci te są obecnie badane przez wiele ośrodków, zarówno akademickich jak i przemysłowych. W moich pracach skupiłem się na dostosowaniu tych sieci do wymogów rozpoznawania mowy: możliwości pracy na długich nagraniach i dołączania zewnętrznych modeli językowych. W drugim wątku wprowadziłem techniki przetwarzanie dźwięku za pomocą sieci służących do rozpoznawania mowy. W moich pracach starałem się jednak aby opracowywane przeze mnie techniki były możliwie jak najbardziej ogólne i poza przetwarzaniem mowy można było je również stosować dla innych zagadnień, takich jak przetwarzania języka naturalnego.

Moje prace są wykorzystywane przez innych badaczy, łącznie mam ponad 2800 cytowań w bazie Google Scholar (indeks H 14) i 360 w bazie Web of Science (indeks H 6).

5 Omówienie pozostałych osiągnięć naukowo - badawczych (artystycznych).

5.1 Inne prace badawcze

5.1.1 Obecnie prowadzone prace badawcze

Obecnie pracuję nad metodami budującymi reprezentacje mowy w sposób nienadzorowany. Dysponując nieopisanymi nagraniami chcemy, aby system znalazł reprezentację, którą można podzielić na część odpowiadającą treści nagrania, prozodię i styl mowy oraz cechy mówcy. W tym celu stosuję głębokie modele uczenia reprezentacji, głównie auto-enkoder wariacyjny [38] i jego modyfikacje takie jak skwantyzowany autoenkoder wariacyjny [56]. Wstępne wyniki badań przedstawiłem w raporcie [P6]. W pracy tej stosujemy skwantyzowany auto-enkoder wariacyjny do zamiany nagrania na sekwencję dyskretnych, 12-bitowych identyfikatorów obliczanych co 20ms. Model można więc traktować jako kodek o bardzo wysokim stopniu kompresji (600bps). Znaleziona reprezentacja jest skorelowana z treścią nagrania (identyfikatory dobrze mapują się na fonemy), będąc nieczuła na mówcę. Nasz model okazał się najdokładniejszym rozwiązaniem dla zadań nienadzorowanego rozpoznawania mowy, takich jak ZeroSpeech 2017 [15]. Opracowane przeze mnie rozwiązanie stało się podstawą dla najlepszego pod względem jakości nienadzorowanego rozpoznawania mowy zgłoszenia w zadaniu ZeroSpeech 2019 zgłoszonym przez Cho et al. [33].

Badania dotyczące nienadzorowanego przetwarzania mowy stanowią podstawę realizowanego przeze mnie we współpracy z John Hopkins University warsztatami letnimi JSALT 2019[2]. Podczas warsztatów, przez 6 tygodni razem z badaczami z JHU, Uniwersytetu w Tulonie i w LeMans oraz studentami z JHU i MIT będziemy opracowywać metody przenoszenia naszej wiedzy o języku na reprezentacje wyłaniane bez nadzoru.

5.1.2 Sieci rekurencyjne z przełączanymi macierzami

Razem ze współpracownikami z Google Brain rozwinęliśmy wariant sieci rekurencyjnych dla danych dyskretnych, w których zamiast nielinowych funkcji aktywacji stosowane są przełączane macierze przejścia [P9]:

$$h_t = W_{x_t} h_{t-1}, \tag{21}$$

gdzie h_t to stan ukryty sieci, W_{x_t} to właściwa wejściu x_t macierz przejścia. Przykładowo, w modelu językowym pracującym na literach, model używa jednej macierzy przejścia dla każdej z liter.

Dzięki usunięciu nieliniowości działanie sieci można analizować za pomocą metod algebry liniowej. Wprowadziliśmy algorytm przedstawiający wartości obliczane przez sieć w czasie t jako kombinację liniową wejść z poprzednich kroków. Za pomocą zmiany bazy przestrzeni linowej aktywacji neuronów rozdzieliliśmy przestrzeń rozpinaną przez neurony sieci na część przetwarzającą wejście, wyjście, oraz pamięć tymczasową. Ponadto, uzyskaliśmy dokładną reprezentacje działania sieci jako automatu skończonego na zadaniach zliczania nawiasów. Oprócz możliwości łatwej analizy działania, sieci z przełączanymi macierzami wejściowymi dają dobre wyniki na rzeczywistych zadaniach modelowania języka.

5.1.3 Zastosowania sieci neuronowych do przetwarzania języka naturalnego

W pracy [P14] zastosowaliśmy sieci neuronowe z mechanizmem uwagi do tagowania i parsowania zdań. Model wczytywał zapis ortograficzny słów i korzystając z sieci splotowych przygotowywał ich niezależne zanurzenia [37]. Następnie zanurzenia słów były przetwarzane przez dwukierunkową sieć rekurencyjną [62] dzięki czemu reprezentacja obliczona dla każdego słowa zależała od całego zdania. Na wyjściu z sieci rekurencyjnej pracował moduł tagujący, przewidujący dla każdego słowa jego funkcję gramatyczną. Następnie, za pomocą mechanizmu uwagi zaimplementowaliśmy parser zależnościowy [36]: dla każdego słowa wskazywana była głowa zależności, oraz typ relacji. Parser uczyliśmy na danych Universal Dependencies [54]. Umożliwiło to wytworzenie modeli wspierających wiele języków. Stwierdziliśmy, że występują synergie między modelami dla podobnych języków. W szczególności nasz

[2]Strona mojego tematu dostępna jest pod adresem `https://www.clsp.jhu.edu/workshops/19-workshop/distant-supervision-for-representation-learning-in-speech-and-handwriting/`

Tablica 3: Uczona na danych polsko-rosyjskich sieć poprawnie grupuje polskie i rosyjskie słowa posiadające podobną formę gramatyczną. Za [P14]

Słowo polskie	Najbliższe słowa rosyjskie
przedwrześniowej	адренергической тренерской таврической непосредственной археологической философской *верхнюю*
większych	автомобильных *трёхдневные* технических практических официальных оригинальных
policyjnym	главным историческим глазным непосредственным *косыми* летним двухсимвольным

najdokładniejszy parser dla języka polskiego wykorzystywał również dane czeskie. Stwierdziliśmy również, że model uczony na połączonym korpusie polsko-rosyjskim nauczył się poprawnie mapować końcówki wyrazów pełniące podobne funkcje gramatyczne (Tabela 3).

W pracy [P10] stworzyliśmy system budowy reprezentacji zdań za pomocą głębokich sieci rekurencyjnych. Celem było przyporządkowanie każdemu zdaniu wektora liczb rzeczywistych tak, aby zdaniom powiązanym ze sobą przypisać wektory o małej odległości euklidesowej. Reprezentacje takie są przydatne w budowie np. systemów wyszukujących informacje, jak system dialogowy opisany w kolejnej części. W zaproponowanym przez nas modelu zanurzenie zdania obliczane jest przez głęboką sieć rekurencyjną. Stwierdziliśmy, że stosowanie zanurzeń neuronowych działa lepiej od uśredniania wektorów słów, przy niewielkim wzroście wymaganej liczby obliczeń.

5.2 Główne wyniki osiągnięte podczas doktoratu

Moja rozprawa doktorska dotyczyła metod umożliwiających zrozumienie działania sieci neuronowych. W tym celu rozważyłem dwa podejścia: analizę wyuczonych już sieci przez budowę reguł decyzyjnych odpowiadających działaniu sieci, oraz zmiany w architekturze sieci promujące zrozumienie już na etapie uczenia. W pierwszym podejściu opracowałem dwa algorytmy budujące diagramy decyzyjne przybliżające działanie sieci: poprzez scalanie diagramów opisujących pojedyncze przypadki uczące [P7] oraz poprzez inspirowany uczeniem drzew decyzyjnych algorytm typu dziel i zwyciężaj [P8]. W drugim podejściu przeanalizowałem sieci neuronowe z nieujemnymi wagami [P3]. Z powodu braku ujemnych wag, neurony w tych sieci uczą się prostych reguł i są łatwiejsze w interpretacji. Mimo tego, sieci uzyskiwały dobre wyniki na zadaniach klasyfikacji cyfr i dokumentów tesktowych.

5.3 Udział w zawodach

Wyniki moich badań wykorzystałem w zgłoszeniach konkursowych dotyczących przetwarzania języka naturalnego i rozpoznawania mowy. W 2017 roku stworzyliśmy system dialogowy umożliwiający prowadzenie konwersacji dotyczących artykułów prasowych [P5]. System łączył klasyczne techniki wyszukiwania informacji z technikami neuronowymi. Odpowiedzi na pytania o fakty były wyszukiwane za pomocą sieci neuronowej w paragrafach z Wikipedii. Ponadto wykorzystaliśmy korpus dialogowy używając wektorowych reprezentacji tekstu do znajdowania podobnych dialogów. System zajął ex-aequo pierwsze miejsce w konkursie NIPS 2017 Conversational AI Challenge[3].

W 2018 roku razem ze współpracownikami uczestniczyliśmy w sponsorowanym przez Airbus konkursie na system transkrybujący nagrania między pilotami a wieżami kontroli lotów[4]. Nasze rozwiązanie, łączące głębokie sieci neuronowe z n-gramowymi modelami językowymi, zajęło drugie miejsce. Ze względu na małe ilości dostępnych danych, w naszym zgłoszeniu kluczową rolę odegrały opracowane przeze mnie techniki regularyzacji sieci.

5.4 Rozwój oprogramowania naukowego

Uczestniczyłem w tworzeniu oprogramowania wykorzystywanego do badań. W szczególności podczas pobytu na Uniwersytecie Montrealskim uczestniczyłem w tworzeniu frameworków Theano [P1] i Blocks

[3] `http://convai.io/2017/`

[4] `https://www.irit.fr/recherches/SAMOVA/pagechallenge-airbus-atc-workshop.html`

[P13], zaś podczas podczas wizyty w Google rozwijałem bibliotekę do przetwarzania mowy Lingvo [P12].

5.5 Uczestnictwo w grantach i pracach badawczych

Jestem kierownikiem grantu NCN Sonata „Zastosowanie rekurencyjnych i głębokich sieci neuronowych do modelowania akustycznego sygnału mowy". Ponadto byłem wykonawcą w grantach NCBiR PBS „Audioscope - system do automatycznego wyszukiwania treści w nagraniach w języku polskim metodą hybrydową". NCBiR Innotech „Rozwinięcie technologii wykonywania wielkoformatowych folii GEM i kompatybilnych technologicznie systemów odczytu" realizowanym przez firmę Techtra Sp. z o.o. oraz Dolnośląskiego Bonu na Innowację „Stworzenie algorytmu wyznaczającego optymalną partię dostaw odnawiającą zapas magazynowy" dla firmy ProLogistica Sp. z o.o.

6 Omówienie osiągnięć dydaktycznych i organizacyjnych

Od 2013 roku prowadzę na Uniwersytecie Wrocławskim zajęcia dotyczące uczenia maszynowego:

- Wykład i ćwiczenia „Sieci Neuronowe i Deep Learning", który jest jednym z najpopularniejszych kursów nieobowiązkowych w ofercie Instytucie Informatyki UWr. Na wykładzie corocznie goszczę też studentów z innych wrocławskich uczelni.
- Kurs o narzędziach dla przetwarzania danych „Nowoczesne Języki Programowania - Matlab, R i Python".
- Seminaria dotyczące uczenia maszynowego, ćwiczenia ze sztucznej inteligencji.
- Kurs pojazdów samojezdnych podczas którego rozwijamy autonomiczną jazdę dla uniwersyteckiego łazika marsjańskiego.

Moje zajęcia stanowią podstawę nowo otwieranego na UWr kierunku studiów drugiego stopnia „Data Science"[5], na których mój nowy przedmiot „Uczenie Maszynowe" będzie jednym z 3 przedmiotów obowiązkowych.

Byłem wykładowcą na organizowanej w 2018 roku "Transylvanian Machine Learning Summer School"[6], w tym roku jestem członkiem jej rady programowej. Byłem również członkiem rady programowej konferencji PLinML [7].

Jestem promotorem prac dyplomowych dotyczących uczenia maszynowego, od 2013 wypromowałem 9 magistrantów, 3 inżynierów i 6 licencjatów. Od roku 2017 jestem promotorem pomocniczym doktoranta Michała Zapotocznego.

W latach 2014-2016 byłem opiekunem studenckiego koła naukowego „Continuum" zajmującego się budową łazika marsjańskiego. Razem ze studentami zajęliśmy 2. miejsce w międzynarodowych zawodach URC w 2016 roku.

Angażuję się też w propagowanie nauki. Uczestniczę w Dolnośląskich Festiwalach Nauki. W latach 2015 i 2017 koordynowałem ofertę Instytutu Informatyki oraz przygotowałem wykłady i pokazy dotyczące sztucznej inteligencji. W roku 2018 zorganizowałem warsztaty dla licealistów podczas których mogli zbudować autonomiczne autka sterowane komputerami Raspberry Pi.

Other Publications

[P1] Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermüller, Dzmitry Bahdanau, et al. Theano: A Python framework for fast computation of mathematical expressions. *CoRR*, abs/1605.02688, 2016.

[P2] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, et al. State-of-the-Art Speech Recognition with Sequence-to-Sequence Models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778, April 2018.

[5] `http://datascience.uni.wroc.pl/`
[6] `https://tmlss.ro/`
[7] `https://plinml.mimuw.edu.pl/`

[P3] J. Chorowski and J.M. Zurada. Learning Understandable Neural Networks With Nonnegative Weight Constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 26(1):62–69, January 2015.

[P4] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results. In *NIPS Deep Learning Workshop*, December 2014.

[P5] Jan Chorowski, Adrian Lancucki, Szymon Malik, Maciej Pawlikowski, Pawel Rychlikowski, and Pawel Zykowski. A Talker Ensemble: The University of Wroclaw's Entry to the NIPS 2017 Conversational Intelligence Challenge. In Sergio Escalera and Markus Weimer, editors, *The NIPS '17 Competition: Building Intelligent Systems*, The Springer Series on Challenges in Machine Learning, pages 59–77. Springer International Publishing, 2018.

[P6] Jan Chorowski, Ron J. Weiss, Samy Bengio, and Aäron van den Oord. Unsupervised speech representation learning using WaveNet autoencoders. *arXiv:1901.08810 [cs, eess, stat]*, January 2019.

[P7] Jan Chorowski and Jacek M. Zurada. Extracting Rules From Neural Networks as Decision Diagrams. *IEEE Transactions on Neural Networks*, 22(12):2435 –2446, December 2011.

[P8] Jan Chorowski and Jacek M. Zurada. Top-Down Induction of Reduced Ordered Decision Diagrams from Neural Networks. In Timo Honkela, Wlodzislaw Duch, Mark A. Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning - ICANN 2011 - 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part II*, volume 6792 of *Lecture Notes in Computer Science*, pages 309–316. Springer, 2011.

[P9] Jakob N. Foerster, Justin Gilmer, Jascha Sohl-Dickstein, Jan Chorowski, and David Sussillo. Input Switched Affine Networks: An RNN Architecture Designed for Interpretability. In *ICML*, pages 1136–1145, July 2017.

[P10] Szymon Malik, Adrian Lancucki, and Jan Chorowski. Efficient Purely Convolutional Text Encoding. In Rafal Rzepka, Jordi Vallverdú, and Andre Wlodarczyk, editors, *Proceedings of the Linguistic and Cognitive Approaches To Dialog Agents Workshop co-located with the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence (IJCAI-ECAI 2018), part of the Federated AI Meeting (FAIM 2018), Stockholm, Sweden, July 13, 2018*, volume 2202 of *CEUR Workshop Proceedings*, pages 14–23. CEUR-WS.org, 2018.

[P11] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing Neural Networks by Penalizing Confident Output Distributions. *ICLR Workshop Track*, abs/1701.06548, 2017.

[P12] Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, Mia X. Chen, et al. Lingvo: a Modular and Scalable Framework for Sequence-to-Sequence Modeling. *CoRR*, abs/1902.08295, 2019.

[P13] Bart van Merriënboer, Dzmitry Bahdanau, Vincent Dumoulin, Dmitriy Serdyuk, David Warde-Farley, Jan Chorowski, and Yoshua Bengio. Blocks and Fuel: Frameworks for deep learning. *arXiv:1506.00619 [cs, stat]*, June 2015.

[P14] Michał Zapotoczny, Paweł Rychlikowski, and Jan Chorowski. On Multilingual Training of Neural Dependency Parsers. In Kamil Ekštein and Václav Matoušek, editors, *Text, Speech, and Dialogue*, Lecture Notes in Computer Science, pages 326–334. Springer International Publishing, 2017.

Literatura

[1] Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In Jan Holub and Jan Žďárek, editors, *Implementation and Application of Automata*, number 4783 in Lecture Notes in Computer Science, pages 11–23. Springer Berlin Heidelberg, January 2007.

[2] Antonios Anastasopoulos, David Chiang, and Long Duong. An Unsupervised Probability Model for Speech-to-Translation Alignment of Low-Resource Languages. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1255–1263, Austin, Texas, November 2016. Association for Computational Linguistics.

[3] Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, et al. Globally Normalized Transition-Based Neural Networks. *arXiv:1603.06042 [cs]*, March 2016.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*, September 2014.

[5] P.L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

[6] Y. Bengio, Renato de Mori, G. Flammia, and R. Kompe. Global optimization of a neural network-hidden Markov model hybrid. *IEEE Transactions on Neural Networks*, 3(2):252–259, March 1992.

[7] Yoshua Bengio. Artificial neural networks and their application to sequence recognition. 1991.

[8] Steven Bird, Lauren Gawne, Katie Gelbart, and Isaac McAlister. Collecting Bilingual Audio in Remote Indigenous Communities. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1015–1024, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[9] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[10] H. Bredin. TristouNet: Triplet loss for speaker turn embedding. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5430–5434, March 2017.

[11] John S. Bridle. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In Francoise Fogelman Soulie and Jeanny Herault, editors, *Neurocomputing*, NATO ASI Series, pages 227–236. Springer Berlin Heidelberg, 1990.

[12] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, Attend and Spell. *arXiv:1508.01211 [cs, stat]*, August 2015.

[13] Hongjie Chen, Cheung-Chi Leung, Lei Xie, Bin Ma, and Haizhou Li. Parallel Inference of Dirichlet Process Gaussian Mixture Models for Unsupervised Acoustic Modeling: A Feasibility Study. page 5, 2015.

[14] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, June 2012.

[15] Ewan Dunbar, Xuan Nga Cao, Juan Benjumea, Julien Karadayi, Mathieu Bernard, et al. The Zero Resource Speech Challenge 2017. *arXiv:1712.04313 [cs]*, December 2017.

[16] Mark Gales and Steve Young. The Application of Hidden Markov Models in Speech Recognition. *Found. Trends Signal Process.*, 1(3):195–304, January 2007.

[17] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Technical Report N*, 93, February 1993.

[18] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture Synthesis Using Convolutional Neural Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 262–270. Curran Associates, Inc., 2015.

[19] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, Las Vegas, NV, USA, June 2016. IEEE.

[20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, et al. Generative Adversarial Networks. *arXiv preprint arXiv:1406.2661*, 2014.

[21] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv:1412.6572 [cs, stat]*, December 2014.

[22] A Graves, A-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649, May 2013.

[23] Alex Graves. Practical Variational Inference for Neural Networks. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2348–2356. Curran Associates, Inc., 2011.

[24] Alex Graves. Generating Sequences With Recurrent Neural Networks. *arXiv:1308.0850 [cs]*, August 2013.

[25] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *23rd ICML 2006*, pages 369–376, New York, USA, 2006.

[26] D. Griffin and J Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, April 1984.

[27] Hossein Hadian, Hossein Sameti, Daniel Povey, and Sanjeev Khudanpur. End-to-end Speech Recognition Using Lattice-free MMI. In *Interspeech 2018*, pages 12–16. ISCA, September 2018.

[28] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, et al. DeepSpeech: Scaling up end-to-end speech recognition. *arXiv:1412.5567 [cs]*, December 2014.

[29] G. Hinton, Li Deng, Dong Yu, G.E. Dahl, A Mohamed, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97, November 2012.

[30] G E Hinton and R R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–7, July 2006.

[31] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[32] D.H. Hubel and T.N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[33] Suhee Jo. VQVAE for Unsupervised Voice Conversion., March 2019.

[34] B. Julesz. Visual Pattern Discrimination. *IRE Transactions on Information Theory*, 8(2):84–92, February 1962.

[35] B Julesz, E N Gilbert, L A Shepp, and H L Frisch. Inability of Humans to Discriminate between Visual Textures That Agree in Second-Order Statistics—Revisited. *Perception*, 2(4):391–405, December 1973.

[36] Dan Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2009.

[37] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-Aware Neural Language Models. *arXiv:1508.06615 [cs, stat]*, August 2015.

[38] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, December 2013.

[39] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

[40] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, March 1964.

[41] G. Kumar, M. Post, D. Povey, and S. Khudanpur. Some insights from translating conversational telephone speech. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3231–3235, May 2014.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[43] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. In Michael A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 255–258. MIT Press, Cambridge, MA, USA, 1998.

[44] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[45] Chia-ying Lee and James Glass. A Nonparametric Bayesian Approach to Acoustic Model Discovery. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 40–49, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[46] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196, June 2015.

[47] E Matusov, S Kanthak, and Hermann Ney. On the Integration of Speech Recognition and Statistical Machine Translation. page 4, 2005.

[48] Josh H. McDermott and Eero P. Simoncelli. Sound Texture Perception via Statistics of the Auditory Periphery: Evidence from Sound Synthesis. *Neuron*, 71(5):926–940, September 2011.

[49] P. MERMELSTEIN. Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, pages 374–388, 1976.

[50] Yajie Miao, Mohammad Gowayyed, and Florian Metze. EESEN: End-to-End Speech Recognition using Deep RNN Models and WFST-based Decoding. *arXiv:1507.08240 [cs]*, July 2015.

[51] Tomás Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent neural network based language model. Makuhari, Chiba, Japan, September 2010.

[52] Mehryar Mohri, Fernando Pereira, and Michael Riley. Speech Recognition with Weighted Finite-State Transducers. In Prof Jacob Benesty Dr, Prof M. Mohan Sondhi, and Prof Yiteng (Arden) Huang Dr, editors, *Springer Handbook of Speech Processing*, pages 559–584. Springer Berlin Heidelberg, January 2008.

[53] H. Ney. Speech translation: coupling of recognition and translation. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 1, pages 517–520 vol.1, March 1999.

[54] Joakim Nivre, Željko Agić, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, et al. Universal Dependencies 1.2. *http://universaldependencies.github.io/docs/*, November 2015.

[55] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, et al. WaveNet: A Generative Model for Raw Audio. September 2016.

[56] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748 [cs, stat]*, July 2018.

[57] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[58] Javier Portilla and Eero P. Simoncelli. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision*, 40(1):49–70, October 2000.

[59] Matt Post, Gaurav Kumar, Adam Lopez, Damianos Karakos, Chris Callison-Burch, and Sanjeev Khudanpur. Improved speech-to-text translation with the Fisher and Callhome Spanish-English speech translation corpus. page 7, 2013.

[60] L. Rabiner and B.-H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, January 1986.

[61] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[62] M. Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, November 1997.

[63] C. Shan, J. Zhang, Y. Wang, and L. Xie. Attention-Based End-to-End Speech Recognition on Voice Search. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4764–4768, April 2018.

[64] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, et al. Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783, April 2018.

[65] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[66] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. *arXiv:1512.00567 [cs]*, December 2015.

[67] László Tóth. Convolutional Deep Maxout Networks for Phone Recognition. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[68] Karel Vesely, Arnab Ghoshal, Lukás Burget, and Daniel Povey. Sequence-discriminative training of deep neural networks. In *INTERSPEECH*, pages 2345–2349, 2013.

[69] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, abs/1609.08144, 2016.