SKRYPT DO LABORATORIUM

**GŁĘBOKIE PRZETWARZANIE TEKSTU I MOWY**

LABORATORIUM 1:

**Sygnały cyfrowe, reprezentacje, parametryzacje**

**Szymon Zaporowski**

**1. Opis ćwiczenia**

**Wymagania wstępne:**

**Cele ćwiczenia:**

**Spodziewane efekty kształcenia ‐ umiejętności i kompetencje:**

**Metody dydaktyczne:**

**Zasady oceniania/warunek zaliczenia ćwiczenia**

**Wykaz literatury podstawowej do ćwiczenia:**

**2. Przebieg ćwiczenia**

**3. Wprowadzenie do ćwiczenia**

**4. Forma i zawartość sprawozdania**

**Dodatki**

1. **Zagadnienia wstępne**

**1.1.Wymagania w odniesieniu do studenta**

Osiągnięcie celów ćwiczenia będzie możliwe poprzez właściwe przygotowanie się studenta do zajęć. Będzie to możliwe poprzez:

- powtórzenie wiedzy nabytej w czasie wykładów,

- zapoznanie się (przed ćwiczeniem) z instrukcją do ćwiczenia ilustrującą podstawowe zagadnienia z zakresu tematyki ćwiczenia.

Ponadto wymagana jest od studenta:

* podstawowa umiejętność programowania,
* podstawowa znajomość języka Python,
* podstawowa znajomość algorytmów uczenia maszynowego przedstawionych w trakcie trwania wykładów
* umiejętność twórczego myślenia.

Wyżej wymienione wymagania bazują na wiedzy i umiejętności jakie studenci powinni zdobyć w czasie edukacji na I poziomie studiów oraz w trakcie trwania 1 i 2 semestru studiów II stopnia. W przeciwnym przypadku studenci powinni nadrobić samodzielnie stosowny materiał.

**1.2.Wymagania w odniesieniu do stanowiska laboratoryjnego**

Stanowisko laboratoryjne powinno być wyposażone w komputer z dostępem do sieci komputerowej oraz z następującymi zasobami:

* przykładowe programy (kody programów) podane w załączniku
* zainstalowany język Python w wersji przynajmniej 3.6.8
* Jupyter Notebook
* Biblioteki niezbędne do realizacji ćwiczenia sckit-learn, numpy, matplotlib, seaborn, pandas, librosa, spafe, tensorflow
* inne: przeglądarka WWW.

Ponadto istnieje możliwość realizacji ćwiczenia w zewnętrznym środowisku Colaboratory na platformie Google, gdzie wymagane jest od użytkownika posiadanie konta Google, natomiast nie ma potrzeby konfiguracji środowiska.

Na komputerze dostępne powinna być zainstalowana instrukcja do ćwiczenia w wersji elektronicznej oraz pliki testowe. Przed zajęciami prowadzący powinien sprawdzić czy dostępne są na stanowisku wymagane zasoby i programy. Szczególnie w zakresie potrzebnych bibliotek języka Python.

**1.3.Cele ćwiczenia**

Celem ćwiczenia jest praktyczna demonstracja wiedzy teoretycznej, przedstawionej w ramach wykładów. Realizując ćwiczenie laboratoryjne studenci zdobędą szereg nowych umiejętności w zakresie implementacji wybranych sposobów parametryzacji sygnałów fonicznych oraz zastosowania ich z wybranymi modelami uczenia maszynowego. Studenci w ramach zajęć dokonają pełnego procesu implementacji algorytmów, w tym:

* analizy danych wejściowych oraz ich dostosowania,
* implementacji wybranych sposobów parametryzacji danych,
* właściwej implementacji wybranych algorytmów uczenia maszynowego,
* analizy wpływu sposobu parametryzacji na wynik końcowy klasyfikacji
* redukcji wymiarowości zbiorów danych,
* prezentacji otrzymanych wyników z wykorzystaniem bibliotek seaborn i matplotlib.

Oprócz tego studenci zapoznają się z wybranymi, najpopularniejszymi formatami zapisu dźwięku i poznają ich zastosowanie oraz wady i zalety.

* 1. **Spodziewane efekty kształcenia ‐ umiejętności i kompetencje:**

Student po zajęciach laboratoryjnych będzie posiadał umiejętność przygotowania danych, doboru odpowiedniej metody reprezentacji i parametryzacji sygnału fonicznego w zależności od rozwiązywanego problemu oraz zastosowanie do nich metody klasyfikacji z wykorzystaniem algorytmów uczenia maszynowego poznanych w ramach zajęć prowadzonych na wcześniejszych semestrach. Dodatkowo student będzie w stanie zinterpretować uzyskane wyniki w standardowych problemach uczenia maszynowego.

**1.5.Metody dydaktyczne**

Student w pierwszej kolejności realizuje zadania przykładowe podczas których przygotowuje odpowiednie środowisko programistyczne, poprzez które testuje poszczególne elementy pracy z danymi do których można wykorzystać metodę omawiane wcześniej algorytmy nadzorowanego uczenia maszynowego. Następnie, na podstawie posiadanej wiedzy i zdobytej umiejętności z zadań demonstracyjnych, realizuje przydzielone zadania, których poziom trudności wzrasta wraz z kolejnymi przykładami. Dodatkowo pośród poleceń pojawiają się pytania dotyczące zagadnień teoretycznych, sprawdzające przygotowanie studentów do zajęć. Wszystkie kody źródłowe programów wpisuje w dokumencie elektronicznym (lub pokazuje prowadzącemu), który jednocześnie staje się sprawozdaniem z ćwiczenia laboratoryjnego.

Materiały wprowadzające i pomocnicze:

* Python – dokumentacja języka <https://docs.python.org/pl/3/>
* SciKit-Learn – dokumentacja do podstawowej biblioteki uczenia maszynowego stosowanej w języku Python: <https://scikit-learn.org/stable/tutorial/basic/tutorial.html>
* Jupiter notebook – narzędzie ułatwiające implementację kodu, uruchamianie oraz testowanie: <https://realpython.com/jupyter-notebook-introduction/>
* Google Colab – narzędzie umożliwiające pracę z kodem w notebooku Jupiter oraz jego uruchamianie w chmurze (konieczność posiadania konta Google): <https://colab.research.google.com/github/cs231n/cs231n.github.io/blob/master/python-colab.ipynb>
* Librosa – dokumentacja biblioteki służącej do obliczania parametryzacji sygnałów fonicznych w języku Python:

<https://librosa.org/doc/latest/index.html>

* SPAFE – biblioteka umożliwiająca obliczanie parametrów sygnałów fonicznych w języku Python:

<https://spafe.readthedocs.io/en/latest/index.html>

**1.5.Zasady oceniania**

Ocenie podlegać będzie realizacja poszczególnych zadań w ramach danego ćwiczenia laboratoryjnego. Dodatkowo w czasie realizacji ćwiczenia mogą zostać przydzielone studentowi dodatkowe zadania. Poszczególnym zadaniom przydzielono liczbę punktów jakie można zdobyć za ich wykonania. Maksymalna liczba punktów wynosi 10.

**Wykaz literatury podstawowej do ćwiczenia:**

**Wykaz literatury podstawowej:**

|  |  |
| --- | --- |
| 1. | Skrypt z materiałami do przedmiotu „Głębokie przetwarzanie tekstu i mowy” |
| 2. | Materiały do przedmiotu opracowane w formie edukacji na odległość, dostęp: https://enauczanie.pg.edu.pl/moodle/course/view.php?id=12525 |
| 3. | Aurélien Géron, Uczenie maszynowe z użyciem Scikit-Learn i TensorFlow. Wydanie II, Helion 2020 |
| 4. | Sebastian Raschka, Python. Uczenie maszynowe, Helion 2017 |
| 5. | Alberto Boschetti, Luca Massaron, Python. Podstawy nauki o danych. Wydanie II, Helion 2017 |
| 6. | T. P. Zieliński, Cyfrowe przetwarzanie sygnałów: od teorii do zastosowań. Warszawa: Wydawnictwa Komunikacji i Łączności WKŁ, 2007 |
| 7. | E. Ozimek, Dźwięk i jego percepcja: aspekty fizyczne i psychoakustyczne. Wydawnictwo Naukowe PWN SA, 2020 |

**Wykaz literatury uzupełniającej:**

|  |  |
| --- | --- |
| 1. | Dokumentacja biblioteki scikit-learn: https://scikit-learn.org/stable/ |
| 2. | Dokumentacja platformy Colaboratory: https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=-gE-Ez1qtyIA |
| 3. | Francois Chollet, Deep Learning. Praca z językiem Python i biblioteką Keras, Helion 2019 |

**2. Przebieg ćwiczenia**

|  |  |
| --- | --- |
| L.p. | Zadanie |
| 1. | Zapoznanie się z instrukcją laboratoryjną (przed ćwiczeniem) |
| 2. | Wprowadzenie kierownika ćwiczenia. Zapoznanie się uczestników ze stosowanymi programami, przegląd dokumentacji (5 min.) |
| 3. | Rodzaje formatów zapisu danych fonicznych (20 min.); |
| 4. | Rodzaje parametryzacji sygnałów fonicznych (20 min.); |
| 5. | Rodzaje sposobów reprezentacji sygnałów fonicznych (20 min.); |
| 6. | Przykład wykorzystania reprezentacji/parametryzacji sygnałów fonicznych w uczeniu maszynowym (20 min.); |
| 7. | Przesłanie/pokazanie zbioru dokumentów/programów do oceny (5 min.). |

**UWAGI!**

PRZED przystąpieniem do ćwiczenia należy utworzyć własny, unikalny katalog w lokalizacji wskazanej przez prowadzącego, a następnie należy tam umieścić pliki testowe oraz własne implementacje kodu programów dokonujących rozwiązanie zadanie. Należy pracować wyłącznie na wykonanej kopii plików.

**3. Wprowadzenie do ćwiczenia**

Rodzaje formatów zapisu danych audio można podzielić na formaty stratne i bezstratne. W przypadku formatów bezstratnych najczęściej stosowanymi formatami są pliki o rozszerzeniach wav oraz flac. Pliki wav są kontenerami, oznacza to, że w środku pliku z takim rozszerzeniem może znajdować się praktycznie dowolny sposób kodowania sygnału audio. Najczęściej jednak jest to oryginalny zapis z wykorzystaniem kodowania PCM (ang. Pusle Code Modulation), nie posiadający żadnej kompresji, natomiast format flac posiada kompresję ok 40-60%. Kompresja FLAC wykorzystuje metodę predykcji liniowej. Oznacza to, iż wartość pojedynczej próbki dźwięku jest przewidywana z wykorzystaniem wartości kilku wcześniejszych próbek, natomiast zapisywana jest tylko różnica pomiędzy wartością rzeczywistą a obliczoną. Należy zaznaczy, że różnice mają zwykle niewielkie wartości.

Formaty stratne zapisu audio, które najczęściej można spotkać w ramach korpusów danych audio to głównie format MP3. Czasami pojawiają się pliki z kodekami OPUS w kontenerach matroska (.mka). Zgodnie z nazwą, formaty stratne cechuje utrata części informacji zawartej w sygnale na rzecz zmniejszenia rozmiaru pliku zawierającego dane. W przypadku formatu MP3 stosuje się tzw. kodowanie perceptualne. Dokładny sposób działania tego kodowania został wyjaśniony w trakcie wykładu. W ramach skrótowego przypomnienia, kodowanie percepturalne polega na usunięciu z sygnału audio informacji, które ludzki narząd zmysłu słuchu postrzega jako nadmiarowe lub nie przenoszące zbyt wiele informacji lub też jako częstotliwości, których ludzki słuch nie będzie w stanie przetworzyć z różnych względów (sposoby maskowania dźwięku lub zakres słyszalnego pasma akustycznego przez człowieka).

Istnieją również kodeki, które wewnątrz wcześniej wspomnianego konterera np. wav pozwalają na zapis pliku w innym formacie. Oznacza to, że przykładowo w pliku o rozszerzeniu .wav może pojawić się plik mp3. Najbardziej popularnym przykładem takiego kodeka jest rodzina kodeków telekomunikacyjnych w skład, której wchodzą kodeki: G.711, G.726, G.729. Są to kodeki przeznaczone do przechowywania informacji w paśmie telekomunikacyjnym, charakteryzują się przez to niewielkim rozmiarem oraz znacznie gorszą jakością niż wcześniej podany kodek MP3.

Parametrami, które cechują jakość nagrania audio są:

* Częstotliwość próbkowania
* Rozdzielczość biotwa
* Przepływność

Najczęściej spotykanymi częstotliwościami próbkowania są częstotliwości 44100 Hz oraz 48000 Hz. Z częstotliwością próbkowania powiązane jest pasmo sygnału. Zgodnie z twierdzeniem o próbkowaniu, znanym też jako twierdzenie Shannona, na podstawie sygnału próbkowanego z pewną częstotliwością Fs jesteśmy w stanie odtworzyć bez zniekształceń maksymalną częstotliwość będącą wartością ½ z Fs.

Oznacza to, że maksymalna częstotliwość jaka można odtworzyć z sygnału próbkowanego z Fs=48000 Hz to 24 kHz. Jak wiadomo, pasmo słyszalne dla człowieka to ok. 20 Hz – 20 kHz. Oznacza to, że sygnał próbkowany z taką częstotliwością może, zgodnie z tw. Shannona, zawierać częstotliwości w pełnym paśmie słyszalnym dla człowieka.

Rozdzielczość bitowa oznacza w sygnale cyfrowym pozwala na stwierdzenie jak dokładnie będzie odwzorowany sygnał analogowy na sygnał cyfrowy. Standardowe wartości rozdzielczości bitowej to 8,16,24 oraz 32 bity. Oznacza to, że w przypadku sygnału z rozdzielczością bitową 8 bit, sygnał jest odwzorowywany poprzez 28 stanów. Im większa rozdzielczość bitowa, tym wierniejsze odwzorowanie sygnału analogowego na sygnał cyfrowy. Obecnie stosowany standardem są sygnały 24 bitowe, jeszcze kilka lat temu standardem było 16 bitów. Pojawia się również coraz więcej plików zapisywanych w rozdzielczości 32 bitowej.

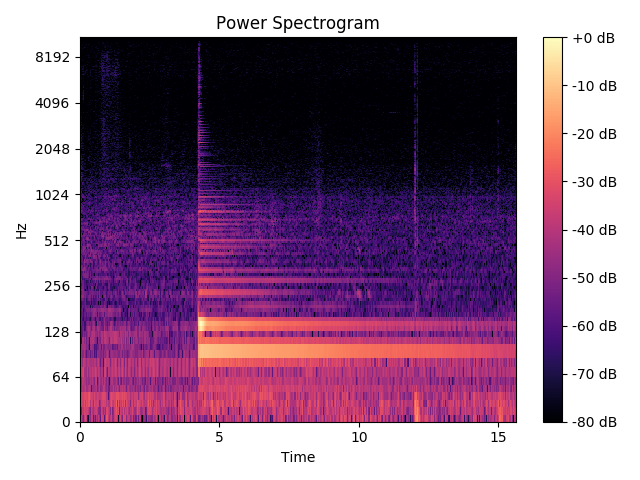
Istotnym parametrem jest również przepływność sygnału. Parametr ten oznacza ilość informacji (bitów) transmitowanych w jednostce czasu poprzez kanał komunikacyjny. Im więcej bitów jest transmitowanych, tym teoretycznie lepsza jakość sygnału. Należy jednak pamiętać, że w przypadku niektórych formatów audi np. MP3, można stosować zmienną lub stałą przepływność sygnału w zależności od tego, czy dany fragment zawiera więcej informacji.

Modele uczenia maszynowego najczęściej nie pozwalają na wykorzystanie „surowych” danych w postaci plików audio. Sygnał zawarty w plikach musi zostać poddany pewnym operacjom, aby stał się zrozumiały dla modeli. Operacje taki nazywane są parametryzacją, gdyż na podstawie cyfrowego sygnału fonicznego obliczane są pewne paramtetry. Rozróżniamy dwie główne grupy parametrów- parametry czasowe oraz widmowe. W skład parametrów czasowych wchodzą parametry takie jak energia sygnału, środek ciężkości sygnału, gęstość przejść przez zero czy też obwiednia sygnału. Parametry widmowe to przede wszystkim widmo mocy, momenty widmowe, płaskość widmowa itp.

Jako parametry związane z widmem można również klasyfikować jedne z najczęściej wykorzystywanych w uczeniu maszynowym parametrów jakimi są współczynniki melcepstralne (ang. Mel Frequency Cepstral Coefficent - MFCC). Sposób obliczania współczynników został w wyczerpujący sposób przedstawiony w ramach wykładu. Są one jednak zaliczane również do grupy parametrów formantowych, podobnie jak inne parametry takie jak kolejne częstotliwości formantowe F0,F1,F2, F3.

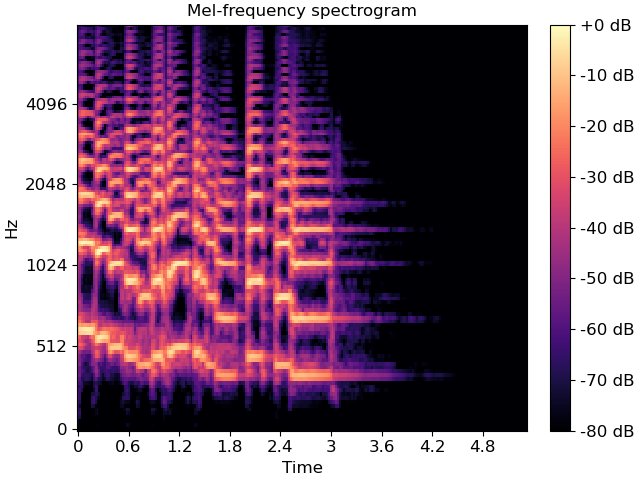
Inną grupę parametrów stanowią parametry powiązane z Liniowym Kodowanie Predykcyjnym (ang. Linear Predictive Coding - LPC). Jeszcze innymi parametrami, które są powiązane z MFCC są współczynniki Gammatone znane jako GFCC. Ten rodzaj parametrów również został przedstawiony w ramach wykładu.

Oprócz parametrów wyróżniamy również sposoby reprezentacji sygnałów fonicznych. Klasycznym sposobem reprezentacji jest tzw. waveform czyli przedstawienie amplitudy sygnału w zależności od czasu trwania. W przypadku przetwarzania audio w uczeniu maszynowym taki sposób reprezentacji nie znajduje zbyt wielu zastosowań. Znacznie częściej wsytępuje reprezentacja sygnału w postaci spektrogramu czyli przedstawienie sygnału audio w postaci widma sygnału w funkcji czasu i wybarwienie amplitudy poszczególnych składowych kolorami. Przykład spektrogramu przedstawiono na rysunku 1.



Rys 1. Spektrogram sygnału audio wykonany z wykorzystaniem biblioteki librosa w języku Python.

Oprócz klasycznego przedstawienia sygnału w postaci spektrogramu istnieją inne sposoby, które charakteryzują się głownie zmianą skali częstotliwości np. melspektrogram, który stosuje skalę melową lub chromagram, który stosuje skalę chromatyczną. Melspektrogram przedstawiono na rysunku 2.



Rys 2. Melspektrogram sygnału audio wykonany z wykorzystaniem biblioteki librosa w języku Python

W tym laboratorium skupimy się na przedstawionych we wprowadzeniu zagadnieniach związanych z sposobami zapisu sygnału audio, jego parametrami oraz przygotowywaniu sygnału fonicznego do użycia w modelach uczenia maszynowego . W ramach tego laboratorium będą postawione następujące zadania i problemy do rozwiązania:

* Zapoznanie się ze sposobami zapisu sygnału fonicznego, wpływem parametrów na jego jakość
* Ocena jakości sygnału audio w zależności od podstawowych parametrów plików audio
* Implementacja skryptów w języku Python do obliczania parametrów typu MFCC, GFCC, LPCC z wykorzystaniem różnych dostępnych bibliotek
* Tworzenie reprezentacji sygnałów fonicznych takich jak spektrogramy czy chromagramy
* Ocena możliwości zastosowania poszczególnych rodzajów plików i sposobów parametryzacji w uczeniu mazynowym

Szczegółowe opisy oraz poszczególne moduły kodów zostały umieszczone w skryptach jupyter notebook, które są nierozłącznym elementem instrukcji do tego ćwiczenia laboratoryjnego (lista plików w dodatku).

**4. Forma i zawartość sprawozdania**

Ocenie podlegać będzie realizacja poszczególnych zadań w ramach danego ćwiczenia laboratoryjnego. Dodatkowo w czasie realizacji ćwiczenia mogą zostać przydzielone studentowi dodatkowe zadania. Poszczególnym zadaniom przydzielono liczbę punktów jakie można zdobyć za ich wykonania. Maksymalna liczba punktów wynosi 10.

**Dodatki:**Pliki instruktażowe oraz pliki z zdanymi niezbędne do wykonania ćwiczenia laboratoryjnego:

1. *Lab1\_ex1.ipynb*
2. *Lab1\_ex2.ipynb*
3. *Folder z plikami m\_2\_row\_straight*
4. *Folder z plikami Speaker\_8*
5. *Folder z plikami punch-deck-bittersweet*
6. *Folder Dane z plikami niezbędnymi do zadań w notatnikach Jupyter*
7. *Plik Zad1.xls*
8. *Formatka sprawozdania GPTIM\_LAB1*

*Wszystkie wykorzystane pliki audio pochodzą ze zbiorów otwartoźródłowych na licencjach pozwalających na bezpłatne wykorzystanie komercyjne oraz w celach edukacyjnych.*