**Metody sztucznej inteligencji**

Multi Layer Perceptron Classifier

(Wielo-warstwowy perceptron)

Rozpoznawanie emocji na podstawie głosu

Kamil Jamros 226821

Wtorek/TN 15:30

1. **Celem projektu**

Celem projektu jest przejście przez proces wyboru optymalnego rozwiązania do wykrywania emocji z ludzkiego głosu.

1. **Plan projektu**

* określenie założeń projektowych,
* pozyskanie danych wejściowych (zgromadzenie lub pobrania udostępnionych danych),
* określenie parametrów, użytych do nauki klasyfikatora,
* wybór odpowiedniego modelu oraz jego parametrów,
* określenie czy klasyfikator spełnia założenia projektowe,
* przetestowanie klasyfikatora na wybranym przykładzie.

1. **Założenia projektowe**

Trafność zaprojektowanego systemu, powinna być wyższa od prawdopodobieństwa wylosowania orła na rzuconej monecie (50%).

1. **Pozyskanie danych wejściowych**

Dane wykorzystane do procesu uczenia oraz walidacji zostały pobrane z Internetu. Wykorzystany zbiór plików to „RAVDESS” (The Ryerson Audio-Visual Database of Emotional Speech and Song), który zawiera nagrania 24 profesjonalnych aktorów (12 kobiet oraz 12 mężczyzn) z neutralnym akcentem dla Ameryki Północnej. W nazwach plików zostały zapisane cechy:

* Modalność (audio, video, audio + video),
* Kanał wokalny (mowa oraz śpiew),
* Emocje (neutralny, spokojny, szczęśliwy, smutny, zły, przerażony, wstręt, zaskoczony),
* Intensywność emocjonalna (normalna, silna),
* Wypowiadana kwestia (“Dog is sitting by the door”, “Kids are taking by the door”),
* Numer powtórzenia (pierwsze, drugie),
* Numer aktora oraz jego płeć (nieparzyści [mężczyzna], parzyste [kobieta]),

Wybrano tylko te pliki, które spełniały następujące założenia:

* Modalność = audio
* Kanał wokalny = mowa
* Emocje = (neutralny, spokojny, szczęśliwy, smutny, zły, przerażony, wstręt, zaskoczony)

Dzięki, czemu ograniczony został zestaw danych do 1440 nagrań audio.

Wszystkie pliki audio są jedno-kanałowe.

1. **Określenie parametrów , użytych do nauki klasyfikatora**

Z wyselekcjonowanych próbek mowy wyznaczono parametry sygnału szeroko stosowane w rozpoznawaniu mowy ludzkiej. Wśród nich znalazły się niżej opisane parametry.

* **Mel Frequency Cepstral Coefficient**, - Współczynniki cepstralne częstotliwości mel (MFCC) sygnału są niewielkim zestawem cech (zwykle około 10-20), które zwięźle opisują ogólny kształt obwiedni widmowej.
* **Przekształcenie Constant-Q** odpowiada analizie sygnału przez bank filtrów o stałej dobroci, czyli o paśmie zwiększającym się z częstotliwością.
* **Spektogram MEL**

1. **Wybór odpowiedniego modelu**

MLPClassifier to wielowarstwowy klasyfikator perceptronów; optymalizuje funkcję log-loss za pomocą „LBFGS”[[1]](#footnote-1) lub stochastycznego spadku gradientu. W przeciwieństwie do SVM lub Naive Bayes, MLPClassifier ma wewnętrzną sieć neuronową do celów klasyfikacji. MLPClassifier trenuje iteracyjnie, ponieważ na każdym etapie obliczane są częściowe pochodne funkcji straty w odniesieniu do parametrów modelu w celu aktualizacji parametrów. Parametry modelu:

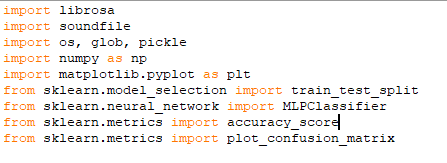
* hidden\_layer\_sizes= 300
* activation = ‘Relu’ [[2]](#footnote-2)
* solver = ‘adam’[[3]](#footnote-3)
* alpha = 0.01
* batch\_size[[4]](#footnote-4) = 256
* epsilon = 1e-08
* max\_iter = 500
* Pozostałe parametry = default

1. **Określenie czy klasyfikator spełnia założenia projektowe**

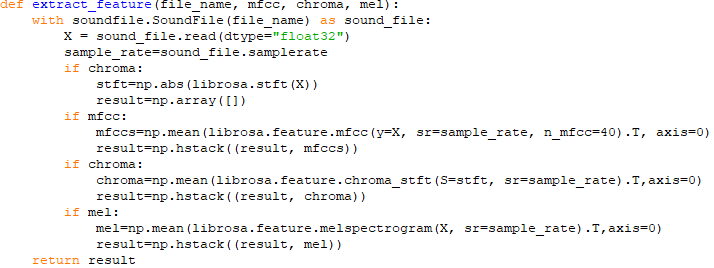
W celu określenia, czy klasyfikator spełnia założenia projektowe, napisano skrypt o nazwie SER.py w języku Python. Dzięki niemu, nauczono klasyfikator oraz przetestowano go na danych walidujących w celu określenia dokładności klasyfikacji.

Program rozpoczynam od zaimportowaniu wymaganych bibliotek:

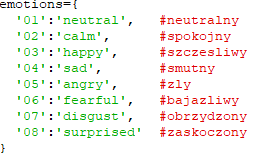
* Librosa – Biblioteka odpowiedzialna za przetwarzanie audio
* Soundfile – Biblioteka odpowiedzialna za odczyt/zapis audio
* OS – Biblioteka umożliwiająca określenie ścieżki do plików audio
* Glob – Biblioteka umożliwiająca przeszukanie określonej lokalizacji
* Pickle – Biblioteka implementująca binarne protokoły do ​​serializacji i deserirializacji struktury obiektu
* MatPlotLib - Biblioteka do rysowania 2D
* Slearn – Biblioteka odpowiedzialna za uczenie maszynowe



Następny krokiem, było zdefiniowanie funkcji „extract\_feature”, która zwracała parametry „mfcc”, „chroma” oraz „mel” z wybranego pliku audio w postaci 1 kolumny ze 180 wartościami



Kolejnym krokiem, było zadeklarowanie słownika, w którym przyporządkowane zostały emocje do określonych cyfr zapisanych w nazwie plików audio w bazie danych oraz listy emocji, które będą klasyfikowane.





Ostatnim punktem, było zdefiniowanie funkcji, która zwróci dwa zestawy danych, z czego jeden będzie zbiorem uczącym, zaś drugi walidującym. Funkcja określi parametry, tylko dla tych nagrań audio, na których zostały przedstawione obserwowane emocję.

**Główny algorytm programu**

Główny algorytm do uczenia klasyfikatora (w kwadratowych nawisach podano przyjęte wartości)

1. Określić przy pomocy stałych „ExtractFeaturesMfcc” [True], „ExtractFeaturesChroma” [True], „ExtractFeaturesMel” [True], jakie parametry mają zostać obliczone
2. Określić przy pomocy stałej „LoadDataTestSize” [0.25] jaka część danych wejściowych ma stanowić zbiór walidujący
3. Określić parametry zastosowanego modelu MLPC [parametry zostały określone w pkt. 5]
4. Nauczenie modelu MLCP, przy pomocy danych uczących
5. Zapisanie wyuczonego modelu
6. Określenie precyzji nauczonego modelu na podstawię danych walidujących
7. Wyznaczenie oraz wyrysowanie macierzy konfuzji.

**Wyniki**

W celu określenie średniej dokładność modelu, algorytm wykonano 10-krotnie obserwując 8 emocji: neutralny, spokojny, szczęśliwy, smutny, zły, przerażony, wstręt, zaskoczony.

Obraz zawierający zrzut ekranu

Opis wygenerowany automatycznie

Rys 1. Przykładowa macierz konfuzji

Przy pomocy funkcji „accuracy\_score”, wyznaczono średnią ze wszystkich prób, których wyniki zamieszczono w tabeli poniżej:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Numer próby | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | średnia |
| Dokładność [%] | 43,06 | 50,28 | 48,33 | 46,67 | 50,00 | 46,67 | 45,00 | 46,11 | 44,72 | 52,78 | 47,36 |

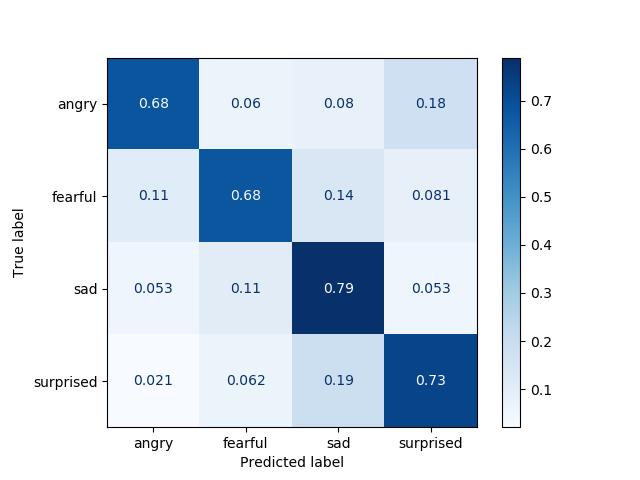
Tab 1. Wyznaczone dokładności 10-krotnego wyuczonego modelu dla 8 emocji

Z powodu zbyt niskiej średniej dokładności, postanowiono zmniejszyć ilość obserwowanych emocji w celu zwiększenie dokładności modelu.

Wykrywane emocje: złość, spokojny, zaskoczony, przerażenie

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Numer próby | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | średnia |
| Dokładność [%] | 68,23 | 56,77 | 66,67 | 66,67 | 72,92 | 68,23 | 69,27 | 72,40 | 69,27 | 72,40 | 68,28 |

Tab 2. Wyznaczone dokładnosci 10-krotnego wyuczonego modelu dla 4 emocj



Rys 2. Przykładowa macierz konfuzji dla 4 emocji (złość, spokojny, zaskoczony, przerażenie)

**Nauczony model w każdym z 10-ciu przypadków spełnia założenia projektowe. Cały skrypt jest zapisany w pliku SER.py**

1. **Przetestowanie klasyfikatora na wybranym przykładzie**

Do celów testowych napisano skrypt o nazwie „TestModel”, który klasyfikuje pliki audio, zgodne z założeniami [plik w formacie wav, jeden kanał, po znaku pierwszym znaku „\_” znajduję się emocja zakodowana w postaci cyfry] w folderze TestFiles. Do klasyfikacji wybrano model nr.5, który nauczono do wykrywania 4 emocji (jego dokładność wynosi 72,92% dla danych walidacyjnych).

Danych wejściowych, było 13 ścieżek dźwiękowych nagranych samodzielnie podczas wymawiania sformułowania “Kids are taking by the door” [z jednym wyjątkiem, gdzie nagrano „Ojejku”]

Na listingu poniżej, przedstawiono wynik klasyfikacji:

Dla ściężki audio: Actor\_05\_1.wav, która przedstawia emocje: 'angry' model sklasyfikował go jako: ['angry']

Dla ściężki audio: Kamil\_04\_1.wav, która przedstawia emocje: 'sad' model sklasyfikował go jako: ['angry']

Dla ściężki audio: Kamil\_05\_1.wav, która przedstawia emocje: 'angry' model sklasyfikował go jako: ['fearful']

Dla ściężki audio: Kamil\_06\_1.wav, która przedstawia emocje: 'fearful' model sklasyfikował go jako: ['fearful']

Dla ściężki audio: Kamil\_08\_1.wav, która przedstawia emocje: 'surprised' model sklasyfikował go jako: ['fearful']

Dla ściężki audio: Kamil\_08\_2.wav, która przedstawia emocje: 'surprised' model sklasyfikował go jako: ['fearful']

Dla ściężki audio: Kasia\_04\_1.wav, która przedstawia emocje: 'sad' model sklasyfikował go jako: ['angry']

Dla ściężki audio: Kasia\_05\_1.wav, która przedstawia emocje: 'angry' model sklasyfikował go jako: ['angry']

Dla ściężki audio: Kasia\_06\_1.wav, która przedstawia emocje: 'fearful' model sklasyfikował go jako: ['angry']

Dla ściężki audio: Kasia\_08\_1.wav, która przedstawia emocje: 'surprised' model sklasyfikował go jako: ['fearful']

1. **Wnioski**

„Wielo-warstwowy perceptron” wykorzystywany do wykrywanie emocji na podstawie współczynników ceptralnych częstotliwości mel, przekształcenia Constant-Q oraz Spektogramu MEL, umożliwia sklasyfikowanie ścieżek dźwiękowych do kategorii: złość, spokojny, zaskoczony oraz przerażenie na poziomie 68,28 %.

Testy wyuczonego modelu na ścieżkach dźwiękowych nagranych samodzielnie na dwóch osobach o przeciwnych płciach pochodzenia polskiego, wykazały, że klasyfikator tylko w 3 na 10 przypadkach, sklasyfikował poprawnie emocje.

W celu poprawienia dokładności, trzeba by było znaleźć większą bazę danych do mechanizmu uczenia, wykorzystać większą ilość parametrów sygnału mowy oraz dokonać ich optymalnego doboru do zadania

1. ‘Lbfgs - optymalizator w rodzinie metod quasi-Newton. [↑](#footnote-ref-1)
2. funkcja rektyfikowanej jednostki liniowej, zwraca f (x) = max (0, x) [↑](#footnote-ref-2)
3. odnosi się do stochastycznego optymalizatora opartego na gradiencie zaproponowanego przez Kingmę, Diederika i Jimmy'ego Ba [↑](#footnote-ref-3)
4. batch\_size – Ilość przechowywanych danych w metodzie forward [↑](#footnote-ref-4)