Klasyfikacja gatunku muzycznego na podstawie właściwości utworu z użyciem sieci rekurencyjnej typu LSTM

Dominika Bocheńczyk, Dorota Meszka, Jolanta Śliwa

6 maja 2022

1 Cel obliczeń

Utworzenie prostej sieci neuronowej rozpoznającej gatunek muzyczny podanego utworu na podstawie jego cech przy użyciu neuronowej sieci rekurencyjnej.

2 Dane

Dane użyte w projekcie pochodzą ze zbioru GTZAN biblioteki Tensorflow. Zbiór ten składa się z 1000 ścieżek audio o długości 30 sekund. Każde 100 ścieżek należy do jednego z 10 gatunków muzycznych:

- blues
- muzyka klasyczna
- country
- disco
- hip hop
- jazz
- metal
- pop
- reggae
- rock

Każda ścieżka jest 16-bitowym plikiem audio 22050Hz Mono w formacie .wav. Dla każdego utworu wygenerowano jego reprezentację MFCC.

MFCC (Mel-frequency cepstral coefficients) reprezentują aspekty takie jak barwa, brzmienie oraz kompozycja w taki sposób, aby mogły później zostać odczytane przez komputer jako wektor wartości liczbowych. Reprezentacja przybliża sposób w jaki ludzki układ słuchowy odbiera dźwięki.

Aby nieco zwiększyć zbiór danych, zdecydowałyśmy się na podział każdej ścieżki muzycznej na krótsze fragmenty - początkowo 6-sekundowe, po modyfikacjach zdecydowałyśmy się na 3-sekundowe. W ostatecznym modelu użyłyśmy zatem 10 000 ścieżek muzycznych.

3 Oprogramowanie

Do wykonania zadania został użyty język Python z bibliotekami Tensorflow (w tym Keras), Scikit-learn, NumPy oraz pakiet Librosa.

4 Topologia sieci

Sieci rekurencyjne mają tendencję nie tylko do zanikania gradientu, ale również do jego gwałtownego wzrostu - eksplodowania. Dzięki zastosowaniu *tanh* jako funkcji aktywacyjnej utrzymujemy wartości pomiędzy -1 i 1, która zapobiega drugiemu zjawisku.

Ze względu na charakter przetwarzanych danych wybrałyśmy LSTM - Long Short Term Memory. W przypadku utworów muzycznych istotne jest uczenie się długoterminowych wzorów i powyższa sieć rekurencyjna nam to zapewnia - w przeciwieństwie do innych sieci nie powtarza modułu za każdym razem, gdy wejście otrzymuje nowe informacje tylko zapamiętuje problem na dłuższy czas.

Sieć składa się z warstwy wejściowej LSTM zawierającej 64 neurony, ukrytej również LSTM i również 64 neurony - obydwie z funkcją aktywacji *tanh* oraz funkcją aktywacji dla kroku rekurencyjnego Sigmoid, Dense z funkcją aktywacji ReLu zawierającej 64 neurony oraz wyjściowej składającej się z 10 neuronów - tylu ile gatunków muzycznych rozróżniamy, i z funkcją aktywacji Softmax. Dodano również jedno wywołanie metody optymalizacyjnej Dropout - polegającej na odrzuceniu losowo pewnej części neuronów w trakcie treningu dla wstępnej poprawy wyniku .

Dzięki zastosowaniu ReLu zmniejszamy prawdopodobieństwo pojawienia się problemu znikającego gradientu, a ponieważ Softmax zmienia uzyskany wektor liczb na wektor prawdopodobieństw (normalizuje wynik, w taki sposób, że całkowita suma kolejnych wartości wektora wynosi 1) jest odpowiednia do zadania klasyfikacji - każda pozycja obrazuje prawdopodobieństwo przynależności utworu do danego gatunku muzycznego. Dla warstw LSTM funkcje *tanh* oraz Sigmoid są najpopularniejsze - defaultowe dla LSTM z biblioteki Keras.

5 Symulacja

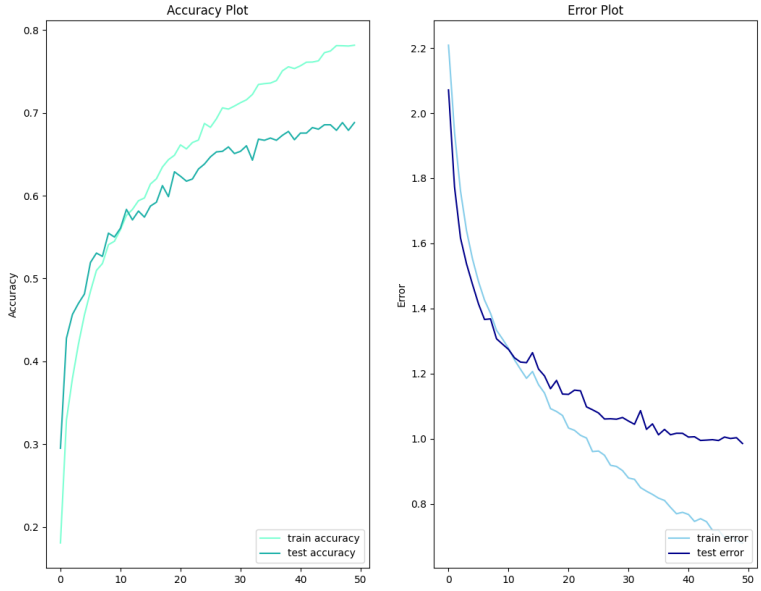
5.1 Parametry

Zbiór danych został podzielony na parametry i oczekiwane wyniki dla zbioru uczącego i zbioru testowego. Rozmiar zbioru testowego został ustawiony na 25%, zbioru walidacyjnego 20% z pozostałych 75%, a zbioru treningowego stanowi pozostałą część danych. Trenowanie sieci neuronowej trwało 50 epok. Podczas trenowania sieci wykorzystano optymalizator Adam, który charakteryzuje się tym, że nie przeprowadza optymalizacji funkcji dla wszystkich danych treningowych tylko dla kolejnych partii (batch) danych, oraz funkcji straty categorical crossentropy.

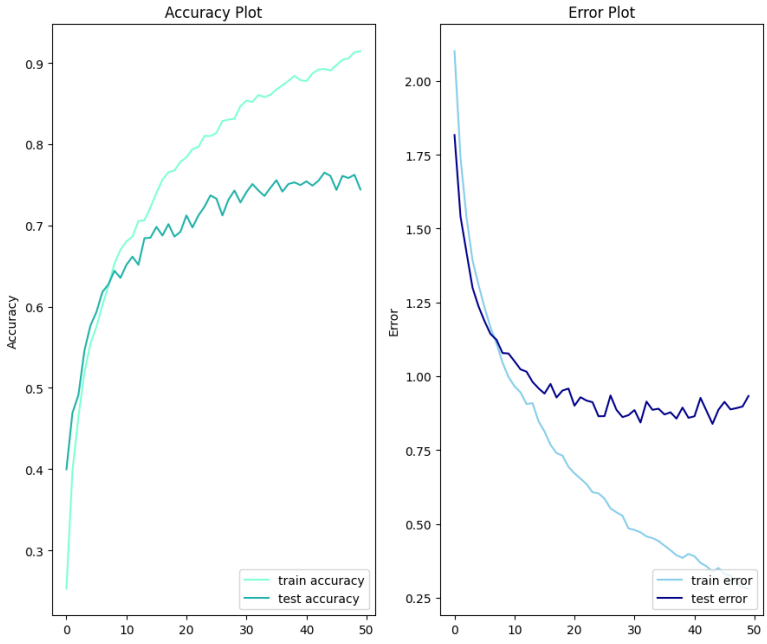
5.2 Wyniki

Przy pierwszej próbie poziom trafności dla zestawu treningowego wyniósł - 78%, a prawidłowość w rozpoznawaniu zestawu testującego jest mniejsza - 68%. Nie jest to jednak różnica tak duża, żeby wnioskować przeuczenie sieci. Mimo to dokonano kilku zmian aby spróbować poprawić pierwotny wynik.

Zmieniono liczbę neuronów w warstwach LSTM na 128 oraz dodano dodatkowy Dropout 30%. Po dokonaniu modyfikacji, wyniki wzrosły dla zbioru treningowego do 91%, a dla zbioru testowego udało się zwiększyć poziom trafności do około 77%.



Rysunek 1: Wykresy uczenia i straty dla 50 epok przy pierwszej próbie



Rysunek 2: Wykresy uczenia i straty dla 50 epok przy drugiej próbie

6 Wnioski

Model okazał się być wysoce skuteczny, ale i dość wolny - analiza tylko 50 epok trwała około 20 minut w pierwotnej wersji, a z dodatkowymi modyfikacjami godzinę. Jednak w porównaniu do tego samego problemu klasyfikacji dla tych samych danych sieć rekurencyjna LSTM już w początkowej wersji dawała lepsze wyniki niż klasyczna.