

ZZSN - Dokumentacja wstępna

Uczenie klasyfikatorów dźwiękowych niewielką liczbą przykładów z wykorzystaniem sieci prototypowych

Sebastian Pietras, Arkadiusz Sadza

Opis zadania

Celem projektu jest stworzenie modelu sieci prototypowej do uczenia klasyfikatorów dźwiękowych niewielką liczbą przykładów (ang. *few-shot learning*). Wyniki tego podejścia powinny zostać porównane ze standardową metodą uczenia klasyfikatorów.

Zbiór danych

Wykorzystanym zbiorem danych będzie [FSD50K](#). Jest to zbiór krótkich (od 0.3 do 30 sekund) fragmentów audio w postaci plików .wav. Do każdego z nich przypisane są klasy, oznaczające czym jest dany dźwięk (np. gitara, alarm). W większości do próbek przypisana jest tylko jedna klasa. Jako że sieci prototypowe w oryginalnym zamyśle służą do klasyfikacji jednoetykietowej to próbki z przypisanymi wieloma klasami zostaną odrzucone.

Każdy fragment audio zostanie przekształcony do określonej długości, a następnie zamieniony na spektrogram. Jeśli dźwięk będzie dłuższy niż ustalony czas trwania to zostanie przycięty, a jeśli krótszy to powielony do wypełnienia całej długości.

Schemat rozwiązania

Jako klasyfikator zostanie wykorzystana spłotowa sieć neuronowa o architekturze zbliżonej do części Audio Subnetwork modelu [Look, Listen and Learn](#). Sieć na wejściu otrzyma spektrogram dźwięku i jej zadaniem będzie przypisanie mu jednej z możliwych klas.

Uczenie modelu zostanie przeprowadzone na kilka sposobów:

- poprzez standardową metodę uczenia klasyfikatorów (np. z wykorzystaniem warstwy Softmax i entropii krzyżowej) na całym zbiorze danych
- jak wyżej, lecz z ograniczoną liczbą próbek per klasa
- zgodnie z algorytmem opisanym w [artykule o sieciach prototypowych](#) (planowana implementacja albo w obrębie metody forward albo jako funkcja straty) dla różnej liczby próbek per klasa

Wyniki klasyfikacji na zbiorze testowym modeli wyuczonych powyższymi sposobami zostaną porównane ze sobą. Szczególnie interesującą kwestią może być umiejętność klasyfikacji przez sieci prototypowe danych ze zbioru testowego do klas, które nie występowały w zbiorze treningowym.

Szczegółowe decyzje dotyczące wyboru ostatecznej architektury sieci, parametrów, metody optymalizacji, sposobu generowania spektrogramu i organizacji kodu zostaną podjęte w toku implementacji.

Narzędzia

Rozwiązanie zostanie zaimplementowane korzystając z biblioteki [PyTorch Lightning](#). Do wygodnej pracy z danymi i kodem oraz łatwej prezentacji wyników planujemy użyć również notatniki [Jupyter](#).