

# Raport (I wersja) - Grammatical Facial Expression

Marek Parr, Michał Mitros

10.01.2020

## 1 Cel badania

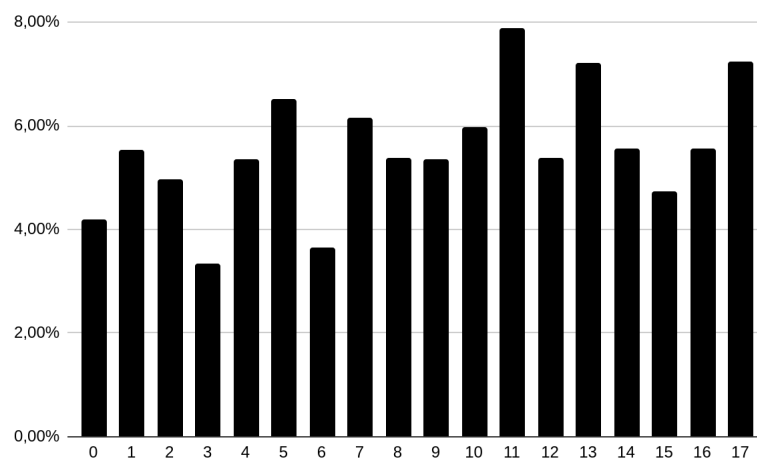
Celem badania jest wyznaczenie modelu klasyfikatora zdolnego do odróżniania gestów na twarzy człowieka na podstawie zapisów z kamer *Microsoft Kinect* (kamer z sensorami głębi umożliwiającymi wyznaczenie trójwymiarowej mapy twarzy). Model powinien powstać w wyniku uczenia na podstawie dostarczonego zbioru danych *Grammatical Facial Expression*.

## 2 Opis zbioru danych

Zbiór danych ma postać 18 par plików. Dane zostały stworzone w oparciu o nagrania dwóch osób wykonujących 9 gestów. Pierwszy plik danej pary zawiera w każdym wierszu czas zapisu (*timestamp*) oraz współrzędne (x, y, z) ponad 100 punktów na twarzy (oczy, nos, lewa brew, kontur twarzy, ...). W jednym pliku wiersze pochodzą z ok. 5 minut nagrania. Drugi plik pary zawiera etykiety wyznaczone ręcznie przez specjalistów (0 przy braku gestu, 1 podczas wykonywania gestu) w wierszach odpowiadających odpowiednim pozycjom w pierwszym pliku.

Współrzędne X i Y mierzone są w pikselach i odpowiadają pikselowi na nagraniu. Współrzędna Z mierzona jest jako odległość danego punktu od sensora (w milimetrach).

Razem we wszystkich plikach jest 27936 rekordów, wśród których jest 9877 rekordów z jednoznacznie oznaczonym gestem. Pozostałe rekordy interpretować można jako "niewykonywanie jednego z gestów", ale nie wiadomo, czy są one "brakiem gestu", który można klasyfikować jako oddzielną klasę, czy być może niektóre z nich powinny być zaklasyfikowane jako jeden z gestów, inny, niż ten, którego dotyczył plik źródłowy rekordu. Żadna z klas znacząco nie przeważa, ani nie przegrywa pod względem liczności, rozkład atrybutów decyzyjnych nie wymusza uwzględniania mniej licznych klas.



Rozkład klas decyzyjnych  
(0-17) - indeksy poszczególnych gestów

### 3 Metodologia

Problem klasyfikacji rozwiązany będzie za pomocą sieci neuronowej. Całość napisana zostanie w języku Python 3, a do utworzenia sieci posłuży biblioteka *numpy* dostarczająca potrzebne operacje na macierzach. Globalną funkcją aktywacji w sieci będzie sigmoid. Każdy neuron wejściowy będzie odpowiadał jednej współrzędnej składowej rekordu. Oznacza to, że sieć będzie miała ok. 300 neuronów wejściowych. Neuronów wyjściowych początkowo będzie 18 - każdy będzie interpretowany jako pojedyncza klasa.

### 4 Wstępne przetwarzanie danych

Pierwszym etapem będzie przygotowanie danych. Według źródła, z którego pochodzi zbiór, wśród atrybutów nie ma brakujących wartości, dlatego nie trzeba rozważać przypadku pustych kolumn. Pierwszym etapem pre-processingu jest konsolidacja plików z danymi, w wyniku której wszystkie rekordy znajdą się w jednym pliku. Z danych usunięta zostanie pierwsza kolumna (*timestamp*), ponieważ nie powinna ona być brana pod uwagę podczas klasyfikacji oraz dodana kolumna z atrybutem decyzyjnym (-1 - nieoznaczony gest, 0-17 - oznaczone gesty). Podczas pierwszych prób klasyfikacji atrybut decyzyjny -1 zostanie pominięty, dlatego dane należy jeszcze prze-filtrować. Wartości w danych są głównie trzycyfrowe, dlatego zastosowana jest też normalizacja danych - zastosowaliśmy normalizację min-max. Kolejnym krokiem jest też losowy podział danych na zbiór treningowy i testowy.

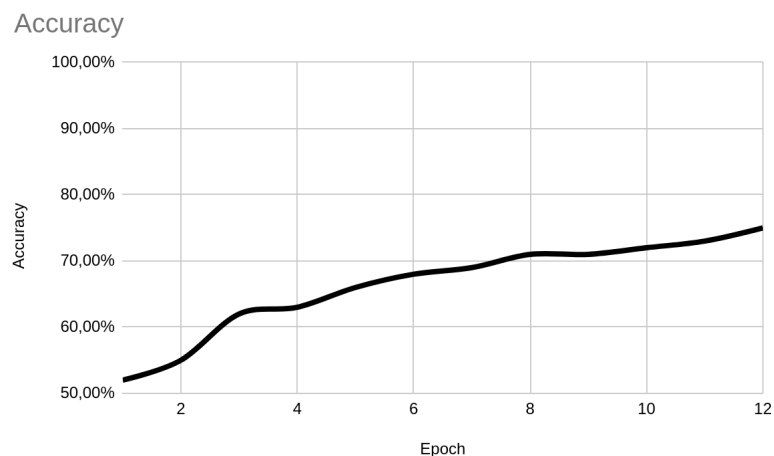
Przyjętą proporcją podziału jest 80/20%. Ze względu na źródło i charakter danych można przyjąć, że nie występuje zaszumienie.

## 5 Ocena jakości modelu

Model oceniany będzie na podstawie pomiarów *Accuracy* na zbiorze testowym. Na pierwszych etapach implementacji, model był tymczasowo oceniany na podstawie maksymalnej wartości *Mean-Squared-Error* ostatniej warstwy z jednej epoki uczenia. Metoda oceny została zmieniona na dokładność przy pierwszej możliwości.

## 6 Wyniki

W pierwszej próbie utworzenia klasyfikatora wyniki nie były satysfakcjonujące. Maksymalna osiągnięta dokładność wyniosła 75% po 12 epokach uczenia. Dalsze uczenie nie poprawiało znacząco wyników. Jakość wyuczonego modelu nie była wysoka, jednak wskazywała na fakt, że sieć neuronowa się uczy. Dodatkowo testy na prostszych przykładach udowodniły poprawność algorytmu uczenia.



Dokładność w kolejnych epokach uczenia

Problem musi więc tkwić we wstępnym przetwarzaniu danych. Pojawił się błąd w podziale danych na zbiory treningowy i testowy. Pierwotnie dane po wczytaniu z pliku były pogrupowane według klas, a proces podziału zbioru nie był losowy. Przez to niektóre klasy nie trafiały w ogóle do uczenia.

Problem można rozwiązać wprowadzając losowość do kolejności przed podziałem. Dodatkowo podczas nauki nie zmieniano kolejności danych uczących, co spowalnia naukę.

## 7 Komentarze i wnioski

W pierwszej wersji udało się utworzyć klasyfikator zdolny do uczenia, niestety drobne niedociągnięcia i przeoczenia spowodowały brak wysokich wyników. W finalnej wersji poprawiony być musi algorytm podziału danych na zbiór uczący i testowy oraz wprowadzona musi zostać losowa zmiana kolejności danych podczas uczenia. Największym problemem napotkanym podczas pierwszych etapów badań był jednak brak jednoznacznej interpretacji rekordów nie mających etykiet w plikach wejściowych. Ostatecznie zostały one pominięte w podstawowej klasyfikacji i trafiły na listę potencjalnych eksperymentów do wykonania po zrealizowaniu podstawy projektu. Płynie z tego wniosek, że podczas tworzenia zbiorów danych rozbitych na pojedyncze pliki należy pamiętać o właściwym oznaczaniu etykiet, aby nie doszło do niejednoznaczności.