Raport z projektu analizy sentymentu recenzji

Celem projektu było wytrenowanie modelu do analizy sentymentu recenzji. Wykorzystując dane recenzji z platformy Amazon wytrenowany został model, który na podstawie tekstu recenzji przewiduje negatywny lub pozytywny sentyment.

Dane

Dane pobrane zostały z platformy Kaggle: Amazon Reviews. Ten zestaw danych zawiera recenzje produktów wraz z ich ocenami, 1-2 gwiazdki w tym zestawie opisane są jako *__label__1*, a oceny 4-5-gwiazdkowe są opisane jako *__label__2*. Oceny neutralne 3-gwiazdkowe zostały usunięte z zestawu. Dane zostały już podzielone na zestaw testowy i treningowy. Zestaw testowy zawiera 400 000 recenzji a treningowy 3 600 000 jednak ze względu na ograniczenia sprzętowe do treningu używałem tylko 1 000 000 danych. Liczba ocen pozytywnych i negatywnych jest w każdym z tych zestawów zbliżona. Do walidacji wykorzystane zostało 10% zbioru danych z zachowaniem proporcji klas.

Dane nie zawierały wartości *None* lub *null*. Do ich wczytywania z pliku i przekształcenia wykorzystywana jest klasa *Dataloader*. W klasie tej dane są wczytywane z pliku dzielone na kolumnę recenzji i oceny a następnie zamieniane na format liczbowy. W przypadku ocen *__label__1* zamieniane jest na 0 a *__label__2* na 1. Do zamiany recenzji na ciągi liczb wykorzystany został Tokenizer dostarczony przez bibliotekę *keras*. Na podstawie wszystkich słów w zestawie danych tworzy on słownik i przypisuje słowom liczby. W przypadku gdy w zestawie testowym znajduje się słowo, którego nie ma w słowniku utworzonym podczas treningu, w to miejsce wstawiany jest tzw. *out_of_vocabulary_token.* W te sposób powstają ciągi liczb reprezentujące poszczególne recenzje. Są one następnie wyrównywane przez przycięcie lub wypełnienie do długości 100.

Model

Wykorzystanym modelem jest jednokierunkowa sieć neuronowa. Pierwszą jej warstwą jest warstwa Embedding, pozwala ona na powiązanie ze sobą słów o podobnych znaczeniach poprzez zamianę ich na wektory, wektory te mają długość 128. Następnie z tych wektorów liczona jest średnia w warstwie GlobalAveragePooling1D i przekazywana do warstw Dense. Po porównanie kilku możliwych architektur najlepsze okazały się dwie ukryte warstwy po 512 neuronów. Funkcja aktywacji to ReLU. W ostatniej warstwie klasyfikującej jest to sigmoid. Jedynym optymalizatorem dla którego sieć uczyła się w zadowalającym stopniu jest Adam.

Rezultaty

Model trenowany był przez dwie epoki, funkcją straty wykorzystaną była entropia krzyżowa. Wykorzystywany rozmiar próbki to 32. Sieć nauczyła się w zadowalającym stopniu. Na zestawach walidacyjnych i testowych dokładność wyniosła 0,907. Na zestawie treningowym było to 0,915. Trenowanie danych przez drugą epokę nie przynosiło znaczącej poprawy już po pierwszej dokładność na zestawie walidacyjnym wynosiła ponad 0,9. Podobnie już po wytrenowaniu modelu na 10000 danych wynosiła ona ponad 0,85. Sugeruje to, że trenowanie modelu dłużej na tych samych danych nie poprawi znacząco rezultatów, również zwiększenie rozmiaru danych treningowych może nie być skuteczne. W celu poprawy rezultatów można sprawdzić działanie warstw lepiej przystosowanych do przetwarzania tekstu.