Sandra: parzyste

Jagoda: nieparzyste

Slajd 1

Slajd tytułowy 🡪 przedstawiamy siebie, tytuł pracy oraz cel pracy + zbiór, który wybrałyśmy i dlaczego

Dzień dobry, nazywamy się Lidia Opuchlik i Sandra Rawicz. Tematem naszej pracy jest analiza sentymentu w recenzjach filmowych. Celem pracy było stworzenie i zbadanie przydatności klasycznych modeli machine learningowych oraz modeli wykorzystujących sieci neuronowe umożliwiających analizę sentymentu recenzji filmowych — klasyfikację czy dana recenzja ma pozytywny czy negatywny wydźwięk.

Zbiór danych pochodzi z portalu IMDB i składa się z 50 000 recenzji w języku angielskim, z których każda oznakowana jest jako pozytywna lub negatywna. Zbiór jest zbalansowany. Podzieliłyśmy go na zbiór treningowy i testowy w proporcji 80 do 20. Zbiór ten jest powszechnie wykorzystywany w pracach naukowych dotyczących analizy sentymentu.

Slajd 2

Jakie algorytmy optymalizacja Las losowy + SVM

W celu użycia zbioru danych do trenowania modeli, zostały one najpierw przekształcone do formy zrozumiałej dla algorytmów przy pomocy Count Vectorizera, który z każdej recenzji stworzył wektor.

Pierwszym algorytmem jaki wykorzystałyśmy jest las losowy, który polega na trenowaniu dużej liczby drzew decyzyjnych i wykorzystujący agregację (bagging), polegającą na losowym wyborze przykładów uczących oraz cech, a następnie zwracanie predykcji jako rezultatu wygenerowanego przez większość drzew. Hiperparametry modelu – liczba drzew (estymatorów) i głębokość drzew – były optymalizowane za pomocą Randomized Search.

Kolejnym algorytmem była klasyczna liniowa wersja klasyfikatora binarnego maszyny wektorów nośnych. Algorytm ten polega na wyznaczeniu hiperpłaszczyzny rozdzielającej przestrzeń danych na dwie klasy. Najważniejszym hiperparametrem tego modelu jest szerokość marginesu (oznaczana jako C), którą optymalizowałyśmy początkowo z wykorzystaniem klasy Randomized Search (szukanie zgrubne), a następnie bardziej dokładne przeszukiwanie z użyciem Grid Search.

Slajd 3

Optymalizacja + sieci neuronowe (3 rodzaje)

Przed zastosowaniem algorytmów bazujących na sieciach neuronowych, dane zostały dostosowanie przy użyciu metody Word2Vec polegającej na przypisaniu każdemu słowu wektora zanurzenia (w sensie embedding) o wielkości 150. Do wytrenowania modelu użyłyśmy słownika zawierającego wszystkie recenzje filmowe ze zbioru treningowego.

Pierwsza skonstruowana sieć była siecią konwolucyjną (splotową) i składała się z warstwy wejściowej Embedding, następnie konwolucyjnej 1-wymiarowej o 128 filtrach, każdy o wymiarach 5x1, łączącej – MaxPooling1D – która spłaszcza dane wejściowe obniżając obciążenie obliczeniowe algorytmu. Kolejną warstwą była warstwa Dropout z prawdopodobieństwem 20%, którą stosuje się w celu zapobiegnięcia przetrenowania modelu. Na końcu były dwie warstwy w pełni połączone tzw. fully-connected o liczbie neuronów odpowiednio 10 oraz 1. Ostatnia, wyjściowa warstwa posiadała również sigmoidalną aktywację. Pozostałe warstwy używały funkcji aktywacji ReLU.

Następnie wytrenowałyśmy rekurencyjną sieć długoterminowej pamięci krótkoterminowej (LSTM). Składała się z warstwy Embedding, warstwy LSTM z wektorem wyjściowym o długości 128 oraz dropoutami – zwykłym i rekurencyjnym na poziomie 20% każdy oraz 1 warstwy całkowicie połączonej (fully-connected) z 1 neuronem i sigmoidalną funkcją aktywacji.

W przypadku obu sieci neuronowych optymalizacja dotyczyła hiperparametrów oraz architektury samych sieci. Odbywała się ręcznie poprzez przeprowadzanie wielu eksperymentów.

Slajd 4

Tabelka z porównaniem wyników.

Ostatnim testowanym przez nas modelem było połączenie wyników predykcji obu sieci neuronowych oraz wzięcie ich średniej jako ostatecznej predykcji. Jest to ostatnio często stosowana metoda, która umożliwia poprawienie wyników. Pomysł ten zaczerpnęłyśmy z publikacji Minaee i współautorzy.

W tabeli przedstawione są wyniki czasu trenowania i precyzji predykcji dla wszystkich stworzonych modeli oraz porównanie wyników dla CNN-LSTM otrzymanych przez nas oraz autorów publikacji.

Zgodnie z przewidywaniami, najlepszą precyzją charakteryzuje się model łączony CNN-LSTM. Jednakże czas potrzebny do jego wytrenowania jest najdłuższy ze wszystkich, gdyż trzeba wytrenować oddzielnie oba modele sieci neuronowych. Zaskakująco dobre wyniki daje prosty, liniowy model maszyny wektorów nośnych, który przy czasie trenowania ok. 1 sekundy daje precyzję na poziomie ….%

Slajd 5

Wnioski (bullet points)

- osiągnęłyśmy wyznaczony przez nas cel

- modele stworzone przez nas zostały również możliwie jak najlepiej zoptymalizowane i osiągnęły wyniki podobne jak w dostępnej literaturze

- całościowo, biorąc pod uwagę oba czynniki, zdecydowanym faworytem okazał się kla- syfikator SVM — model liniowy, bardzo prosty, o małej liczbie parametrów

- nie ma rozwiązania idealnego. Żeby osiągać optymalne wyniki, trzeba prawie zawsze iść na kompromis (czas trenowania vs. Precyzja (inna metryka))

- Ponadto, ważnym faktem, który zawsze trzeba wziąć pod uwagę, jest to, że niektóre algorytmy ulegają łatwemu prze- trenowaniu. Wysokie wyniki predykcji modelu, ale fałszywe wyniki nie wnoszą dużo wartości dodanej, a wręcz prowadzą do tego, że wyciągami nieprawidłowe wnioski