Projekt 12 Mikołaj Pokrywka

06-DUMAUI0 2022/SL

Cel projektu

Celem projektu było stworzenie modelu, który przewiduje, czy firma wystawia fałszywe ogłoszenia o prace. Projekt głównie skupia się na użyciu funkcjonalności autocast() znajdującej się w bibliotece pytorch, która pozwala na używanie tak zwanego mixed precision, czyli zmiany reprezentacji wartości z float32 na float16. Dzięki czemu możemy szybciej trenować modele, jednocześnie zachowując podobną jakość. (Mixed precision: https://pytorch.org/docs/stable/notes/amp_examples.html)

Dane

Dane pochodzą z wyzwania Real / Fake Job Posting Prediction

z platformy Kaggle (link: https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction).

Jako danych użyto tabeli *company_profile,* dane te zostały z wektoryzowane za pomocą TF-IDF

Po z procesowaniu danych podzielono zbiór na dane:

trenujące: 12571 linijek

ewaluacyjne: 1000 linijek

testowe: 1000 linijek

Modele

W projekcie porównano działanie 3 modeli:

- Regresja logistyczna binarna.
- Sieć neuronowa typu forward z
  - 8 warstwami Linear layer + Relu layer.
  - 1 warstwa Linear layer + LogSofmtax layer

  Trenowano przez 150 epok, pomiędzy warstwami użyto wielkości wektora 512
- Powyższa sieć neuronowa z funkcjonalnością **autocast()**

Ewaluacja

Wykorzystano metryki *accuracy, F1-score*. Wyniki ewaluacji przedstawia poniższa tabelka (Poniższe wyniki odpowiadają ewaluacji ze zbioru testowego):

Model	Accuracy	F1-score	Time (in seconds)
Regresja logistyczna	0.993	0.818	0.295
Sieć neuronowa	0.999	0.987	21.496

Sieć neuronowa z Mixed precision	0.999	0.987	7.143

Zmiana wielkości wektora wejściowego i wyjściowego

Zmiana wielkości wektora wejściowego i wyjściowego w poszczególnych warstwach ma znaczenie na długość trenowania sieci, oraz na jej wyniki.

Poniższa tabelka przedstawia wyniki przy różnych wielkościach wektora wejściowego i wyjściowego w poszczególnych warstwach

Model,	Accuracy	F1-score	Time (in seconds)
Domyślna **64**	0.998	0.971	2.922
Mixed precision **64**	0.998	0.971	2.518
Domyślna **512**	0.999	0.987	18.496
Mixed precision **512**	0.999	0.987	21.496
Domyślna **1024**	0.999	0.987	7.143
Mixed precision **1024**	0.996	0.946	13.051

Uwagi:

Eksperyment został powtórzony wiele razy i w każdym eksperymencie metryki i czas trenowania były podobne.

Zdaję sobie sprawę, że wielkość danych nie jest zadowalająca, jednakże ogranicznikiem eksperymentu był sprzęt. W tym projekcie używano platformy *colab (https://colab.research.google.com/),* gdzie jest ograniczony czas używania karty graficznej.

Wnioski

Jak wynika z powyższych eksperymentów używanie funkcjonalności autocast pozwala znacząco zredukować czas trenowania, jednocześnie osiągając podobne wyniki.