Specyfikacja Oprogramowania
IEE Std 830-1998

„Tensor Flow”

https://github.com/Gruschwick/tensorflow

Mateusz Paluchowski
Politechnika Gdańska 2018/2019

Spis treści

Streszczenie

Otwarto-źródłowa biblioteka programistyczna wykorzystywana w uczeniu maszynowym i głębokich sieciach neuronowych.

1 Wstęp

1.1 Cel

Celem projektu jest realizacja wybranego problemu związanego z funkcjonowaniem oprogramowania, który został zgłoszony przez innych użytkowników na portalu GitHub.

1.2 Zakres

- https://github.com/tensorflow/tensorflow/issues/24374

1.3 Definicje, akronimy, skróty

Projekt – zapoznanie i realizacja własnych modeli projektu TensorFlow oraz rozwiązanie zgłoszonego problemu związanego z tym oprogramowaniem

Oprogramowanie – ogół instrukcji zintegrowanych w dany program

Biblioteka – zespół skorelowanych funkcji przeznaczonych do wykorzystania w ramach danego języka programowania

Funkcja – cecha wykonująca odgórnie przyjęty zakres czynności

Otwarto-źródłowy – typ oprogramowania oparty na licencji pozwalającej uzyskać do niego dostęp niezależnym od siebie użytkownikom

Tensor – obiekt matematyczny będący uogólnieniem wektora

Uczenie maszynowe – dziedzina wchodząca w skład nauk zajmujących się problematyką sztucznej inteligencji

1.4 Referencje, odsyłacze do innych dokumentów

[1] IEEE Std 830-1998 outline
[2] tensorflow.org
[3] github.com/tensorflow/tensorflow

1.5 Krótki przegląd

Dokument przedstawia opis oprogramowania wraz z wymaganiami oraz dokładnym opisem funkcji programu. Przedstawiony został harmonogram pracy nad projektem.

2 Ogólny opis

2.1 Walory użytkowe i przydatność projektowanego programu

Zakres funkcjonalności funkcji, której rozwiązanie problemu jest przedmiotem projektu, stosowany jest wszędzie tam, gdzie realizowane są obliczenia numeryczne oraz abstrakcyjne operacje bazujące na algebrze liniowej. W praktyce jest to nieodzowny element uczenia maszynowego.

2.2 Funkcje i możliwości programu

Funkcja ‘tf.einsum’, której dotyczy problem jest uogólnioną kontrakcją pomiędzy tensorami o określonym wymiarze. Produktem funkcji jest tensor, którego elementy są zdefiniowane przez równianie napisane w skróconej formie inspirując się konwencją sumacyjną Einsteina. Jako przykład można przedstawić mnożenie dwóch macierzy A i B czego rezultatem jest macierz C. Według konwencji elementy macierzy C będą przedstawiały się w następujący sposób:

C [i, k] = sum_j A [i, j] * B [j, k]

Na tej podstawie można w przystępny sposób wyrazić wiele powszechnie stosowanych operacji, na przykład:
- iloczyn skalarny,
- iloczyn diadyczny,
- mnożenie macierzy,
- mnożenie macierzy partiami,
- operacja transpozycji.

2.3 Ogólne ograniczenia

Zachowanie spójności i poprawności modyfikowanych linii kodu zgodnie z przyjętą normą powstawania oprogramowania oraz techniką posługiwania się językiem Python zgodnie z wytycznymi firmy Google. Ograniczenie do stosowania rozwiązań uzupełnione przez rozwiązania innych uczestników projektu.

2.4 Charakterystyka użytkowników

Użytkownikami są programiści rozwijający projekt oraz wykorzystujący do pracy oprogramowanie.

2.5 Środowisko operacyjne

Zakres realizowanego problemu przeprowadzany jest w języku Python (wersja 3.6.6), który został zaimplementowany w środowisku Anaconda.

2.6 Założenia i zależności

Zakres realizowanej problematyki związanej z oprogramowaniem jest ściśle związana z całością jego funkcjonowania.

3. Specyficzne wymagania

3.1 Wymagania funkcjonalne

Code to reproduce the issue

```
import tensorflow as tf

matrix = tf.constant([[1, 2, 3],
                      [4, 5, 6],
                      [7, 8, 9]])

with tf.Session() as sess:
    print(tf.einsum('ii',matrix).eval())

# 45 (should be 15)
```

Obraz powyżej: elementarny przykład wyznaczenia śladu określonej macierzy, w którym można zidentyfikować realizowany problem. Według założenia funkcji ‘tf.einsum’ program na podstawie przyjętego podwójnego parametru ‘ii’ (osie – tutaj przekątna macierzy) powinien wyprodukować ślad macierzy, czyli sumę elementów ‘ii’ (przekątnej) macierzy. Mimo to, jako wynik została wyświetlona suma wszystkich poszczególnych elementów macierzy.

W elemencie kodu definiowanej funkcji ‘tf.einsum’ została zastosowana operacja wykorzystania innej funkcji ‘tf. reduce_sum’, którą określono w innym pliku biblioteki. Cechą funkcji ‘tf.reduce_sum’ jest obliczanie sumy elementów danego tensora na podstawie przyjętych

parametrów (współczynników). Jest to miejsce w kodzie, w którym identyfikuje się podstawę do wystąpienia wyżej ukazanego problemu.

Jako rozwiązanie proponuje się zastosowanie funkcji ‘tf.trace’ (została zdefiniowana w tym samym pliku co funkcja ‘tf.reduce_sum’), które to określone jako szczególny przypadek, w którym duplikuje się przyjmowane współczynniki dla funkcji ‘tf.einsum’. W przypadku, gdy liczba powtórzeń współczynnika będzie mniejsza od dwóch, funkcja ‘tf.einsum’ dokona obliczeń poprzez zastosowanie omawianej ‘tf.reduce_sum’. Dla przypadków wystąpienia powtórzenia współczynnika więcej niż dwa razy program wyświetli stosowny błąd, jako że tego typu wielokrotne powtórzenie, przy określonym wymiarze tensora, stanowi błąd logiczny.

3.2 Wymagania niefunkcjonalne

3.2.1 Wymagania dotyczące wymaganych zasobów :

Brak specjalnych wymagań dotyczących zasobów oraz możliwości technicznych sprzętu, na którym wykorzystywane jest oprogramowanie.

3.2.2 Wymagania dotyczące dokumentacji :

Wypunktowanie informacji dotyczącej miejsca w linii kodu, w którym rozpoznano i rozwiązano problem.

3.2.3 Wymagania dotyczące sposobów weryfikacji :

Możliwość weryfikacji poprzez opinię innych użytkowników rozwijających projekt uprzednio przygotowanego ‘pull request’, czyli zawiadomienia o próbie dokonania zmian w linii kodu.

3.2.4 Wymagania dotyczące bezpieczeństwa:

Zdefiniowana cecha nie może mieć istotnego, negatywnego wpływu na stabilność i bezpieczeństwo funkcjonowania całego pakietu oprogramowania.

3.2.5 Wymagania dotyczące jakości:

Rozwiązanie musi stosować się do zasad powstawania i rozwijania projektu oraz odgórnie przyjętej konwencji posługiwania się stosowanym językiem programowania.

4. Dodatki

4.1 Harmonogram prac nad projektem

3.12.2018r	Wybór issue dotyczącego oprogramowania w celu jego rozwiązania.
17.12.2018r	Zapoznanie się z założeniami projektu, nauka jego funkcjonalności poprzez projektowanie własnych modelów.
24.12.2018r	Realizacja bardziej złożonych modeli i zagadnień numerycznych.
7.01.2019r	Zapoznanie się z potrzebnymi bibliotekami.
14.01.2019r	Przygotowanie pull request, oddanie projektu.