

Programowanie równoległe i rozproszone

Politechnika Krakowska

Laboratorium 1

Paweł Suchanicz,
Rafał Niemczyk

14 października 2019

Spis treści

| | | |
|----------|--|----------|
| 1 | Wstęp | 2 |
| 1.1 | Opis laboratorium | 2 |
| 1.2 | Specyfikacja sprzętowa | 2 |
| 1.3 | Zbiór danych | 2 |
| 2 | Wyniki | 3 |
| 2.1 | Normalizacja i standaryzacja | 3 |
| 2.1.1 | Implementacja w C++ | 3 |
| 2.1.2 | Implementacja w Python | 3 |
| 2.1.3 | Porównanie wyników | 3 |
| 2.2 | Klasyfikacja KNN | 4 |
| 2.2.1 | Implementacja w C++ | 4 |
| 2.2.2 | Implementacja w Python | 4 |
| 2.2.3 | Porównanie wyników | 4 |
| 3 | Podsumowanie | 5 |

1 Wstęp

1.1 Opis laboratorium

Celem laboratorium było wykorzystanie interfejsu OpenMP w celu zrównoleglenia kodu C++. Interfejs OpenMP składa się głównie z dyrektyw preprocesora a także z zmiennych środowiskowych i funkcji bibliotecznych. W laboratorium wykorzystywany będzie głównie do zrównoleglania pętli.

Algorytmy, które są implementowane a następnie zrównoleglane w ramach laboratorium to normalizacja min-max, standaryzacja rozkładem normalnym i klasyfikacja KNN (k-najbliższych sąsiadów). Zaimplementowany KNN uwzględnia jednego sąsiada i używa metryki euklidesowej.

Szybkość działania każdego algorytmu została zmierzona dla implementacji w C++, implementacji w C++ po zrównolegleniu dla różnej ilości wątków (1-4) oraz implementacji w Python (ze skorzystaniem z funkcji z pakietu scikit-learn).

1.2 Specyfikacja sprzętowa

Przy pomiarach szybkości wykonywania algorytmów wykorzystany był sprzęt w konfiguracji:

- Procesor: Intel Core i7-4712MQ 4 x 2.30GHz
- Ram: 8GB DDR3
- System: Linux (Fedora 22)

1.3 Zbiór danych

Wykorzystany został zbiór obrazów ręcznie pisanych cyfr MNIST. Wykorzystany zbiór ma format .csv i zawiera 60000 rekordów, gdzie każdy rekord odpowiada za jeden obrazek 28x28 pikseli w skali szarości. Pierwsza wartość w rekordzie jest cyfrą która widnieje na obrazku, a kolejne to wartości pikseli obrazka.

Dla zadań postawionych w laboratorium zbiór danych jest dość duży, więc został on obcięty do pierwszych 6000 rekordów, z czego 4500 przeznaczono do trenowania, a pozostałe 1500 do testowania.

2 Wyniki

2.1 Normalizacja i standaryzacja

Normalizacja:

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Standaryzacja:

$$x^* = \frac{x - \mu}{\sigma}$$

2.1.1 Implementacja w C++

2.1.2 Implementacja w Python

2.1.3 Porównanie wyników

2.2 Klasyfikacja KNN

2.2.1 Implementacja w C++

2.2.2 Implementacja w Python

2.2.3 Porównanie wyników

3 Podsumowanie