

Praca domowa 07 – classification

Michał Szczygieł

Wstęp

Zadanie składa się z 2 plików:

- Main.java
- Classification.java

CEL: Odczytanie danych z pliku źródłowego, zawierającego pewną ilość wektorów. Zestaw wektorów składa się z danych do uczenia i do rozpoznania, po pierwszym wystąpieniu klasy „0” następuje proces uczenia (każdy vector posiada swoją określoną klasę).

Klasa Main

Zadaniem tej klasy jest wczytanie danych z pliku, oraz przekazanie tych wartości do klasy Classification. Również oprócz danych z pliku klasa ta winna przekazać parametr n-size (określający rozmiar wektora), który również jak i parametr file path nadawana jest jako argument wywoławczy programu.

Metody:

`public static void main(final String... args) throws FileNotFoundException`

- Metoda główna, przyjmuje dwa parametry filePath i size, przekazuje wczytane dane i rozmiar do klasy Classification, wykonując cel, oraz wyświetlenie klas dla rozpoznanych wektorów.

`private static LinkedList<Integer> readFile(File file) throws`

`FileNotFoundException` – Metoda ta wczytuje dane w postaci liczb rzeczywistych z podanej ścieżki oraz zwraca przefiltrowane dane.

Klasa Classification

Klasa ta, otrzymawszy dane od klasy wywołującej, tworzy odpowiednio 3 wektory:

```
/**
 * Vector Collection contains normalized distribution of vectors for the
 * relevant classes.
 */
private Map<Integer, ArrayList<Double>> vectorCollectionNormalized;

/**
 * Vector Collection contain data to "learn" application.
 */
private Map<Integer, ArrayList<ArrayList<Double>>> vectorCollectionToLearn;

/**
 * Vector Collection contain data to recognize.
 */
private Map<Integer, ArrayList<ArrayList<Double>>> vectorCollectionToRecognize;
```

Pozostałe metody zostały opisane w kodzie źródłowym programu.

Następnie rozpoczyna się proces uczenia, a w końcowym etapie rozpoznawanie wektorów i przypisywanie im odpowiednich klas.

Realizacja algorytmu Classification

W mojej realizacji algorytmu dla klasy Classification zastosowałem następujące kroki. Tworzę trzy wektory, jeden zawierający dane do uczenia, drugi będący znormalizowanym wektorem (zawiera średnie wartości wektorów dla danej klasy), oraz trzeci wektor zawierający dane do rozpoznania. W kolejnym etapie następuje wypełnienie tych wektorów danymi. W procesie uczenia pierwszy dwa są wykorzystywane. A w fazie klasyfikacji zostają wykorzystane dwa ostatnie (znormalizowany i do rozpoznania). Proces rozpoznania polega na wyznaczaniu najmniejszej różnicy odległości pomiędzy danymi klasami dla danej która ma zostać rozpoznana.

Złożoność obliczeniowa dla tego programu wynosi:

- Dla procesu uczenia – $O(2n^2)$
- Dla procesu rozpoznawania - $O(n^3)$

Dyskusja

Realizując powyższe rozwiązanie można byłoby dobrać inne metody uczenia i rozpoznawania. Inną metodą rozpoznawania danego wektora wydaje się być dobra metoda „Cosine similarity” wyznaczająca kąt pomiędzy dwoma wektorami. Natomiast dla procesu uczenia można było by stworzyć bardziej zaawansowane metody uczenia, jak tworzenie sieci neuronowych.