**Strojové učenie a neurónové siete**

**Zadanie číslo 2**

Andrej Dubeň

74657

Obsah

[Zadanie 3](#_Toc496816366)

[Riešenie 3](#_Toc496816367)

[Úloha 1. 3](#_Toc496816368)

[Úloha 2. 4](#_Toc496816369)

[Kmeans 4](#_Toc496816370)

[DBScan 5](#_Toc496816371)

[SOM 6](#_Toc496816372)

# Zadanie

Zadanie č. 2

Pripravené dáta z úlohy č.1 môžu obsahovať veľa rovnakých vzoriek. Cieľom tohoto zadania bude vyhľadať rovnaké alebo podobné vzorky, ktoré sa nachádzajú medzi množinami. Tieto vzorky následne odstránime a tak sa vyheme ich duplicite (Odstránte ich z trénovacej množiny).

Úloha č.1.

Zistite prienik medzi trénovacími, validačnými a testovacími dátami. Myslite na to, že obrazy nemusia byť úplne rovnaké ale aj veľmi podobné. (P.S.: Máte veľké množstvo dát. Dajte si pozor aj na rýchlosť spracovania. Pokiaľ nenájdete úplné riešenie pracujte s menšou mnoužinou dát.)

Úloha č.2.

Nájdite podobné skupiny znakov pomocou zhlukovacích algoritmov K-means, DBSCAN a SOM. Po nájdení skupiny podobných znakov (zhlukov), vykreslite priemerný obraz a obraz najbližší priemernému obrazu. Pre každý nájdený zhluk vypíšte počet prvkov v zhluku a štatistickú odchýlku celého zhluku. DBSCAN musí vyhľadať aspoň 10 zhlukov pričom každý zhluk musí obsahovať apsoň 30 prvkov. Tieto algoritmy spustite aspoň na vašich validačných a testovacích dátach, ktoré budú mať aspoň 10 000 prvkov.

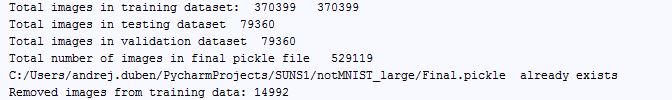
# Riešenie

## Úloha 1.

Vymazal som rovnaké a podobné obrázky medzi datasetmi z trénovacieho datasetu. Postup je nasledovný:

* Vytvoril som set() lebo nebude obsahovať duplicitné haše
* Zahašoval validačné dáta a pridal ich do setu
* Zahašoval testovacie dáta a pridal ich do setu
* V cykle hašhoval trénovacie dáta a hneď aj porovnával so setom, ak sú zhodné tak som si do arrayu pridal ich index
* Vymazal som dáta podla indexov zhodných dát

Z 530k obrázkov môj algoritmus vymazal 357



## Úloha 2.

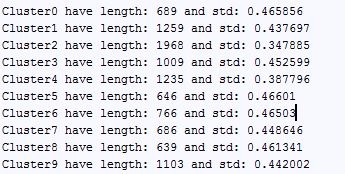
V tejto úlohe som naimplementoval zhlukovacie algoritmi Kmeasns, DBScan a SOM. Každý algoritmus vypíše počet dát v zhluku a ich štaticstickú odchíľku, vykreslí priemerný obraz a najbližší obraz k priemernému obrazu.

### Kmeans

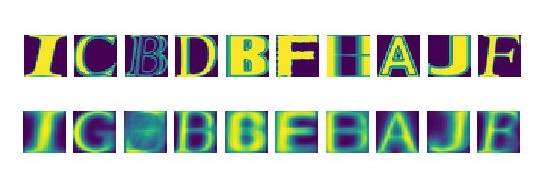
Na vytvorenie zhlukov som použil knižnicu: sklear.cluter.Kmeans a jej metódy:

* KMeans(n\_clusters) – počet zhlukov nastavený na 10
* Kmeans.fit(dataset) – dataset bolo potrebné zmeniť na 2D (index, plocha)

Výsledné zhluky:



Zobrazenie priemerných obrazov(dole) a najbližších k nim (hore):

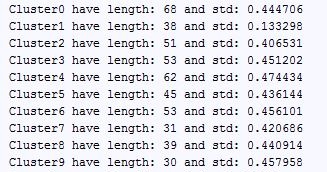


### DBScan

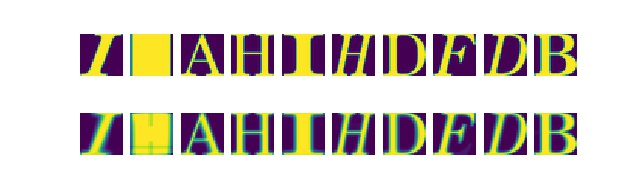
Na vytvorenie zhlukov som použil knižnicu: sklear.cluter.DBScan a jej metódy:

* DBScan(eps, min\_samples) – eps je vzdiaľenosť medzi dátami, min\_samples – minimálny počet dát v zhluku
* DBSCAn.fit(dataset) – dataset bolo potrebné zmeniť na 2D (index, plocha)

Výsledné zhluky:



Zobrazenie priemerných obrazov(dole) a najbližších k nim(hore):

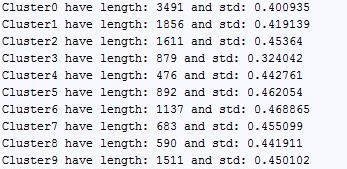


### SOM

Na vytvorenie zhlukov som použil knižnicu: MiniSom a jej metódy:

* MiniSom(počet neurónov, počet neurónových vrstiev, plocha, počiatočné rozloženie, počiatočný training\_rate)
* Train\_random(dataset, iterations) – dáta náhodne preusporiadané
* Train\_batch(dataset, iterations) – dáta usporiadané podla usporiadania datasetu
* Winner(data) – vráti výherný neurón pre daný obrázok

Výsledné zhluky:



Zobrazenie priemerných obrazov(dole) a najbližších k nim(hore):

