Κ23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα
Χειμερινό εξάμηνο 2020-21

3η Προγραμματιστική Εργασία

Διανυσματική αναπαράσταση εικόνας σε χώρο χαμηλότερης διάστασης με χρήση νευρωνικού δικτύου αυτοκωδικοποίησης. Αναζήτηση και συσταδοποίηση των εικόνων στον νέο χώρο και σύγκριση με προσεγγιστική και εξαντλητική αναζήτηση και συσταδοποίηση στον αρχικό χώρο (διάστασης 784). Για την υλοποίηση του νευρωνικού δικτύου θα χρησιμοποιηθεί η γλώσσα Python (3.8) και η προγραμματιστική διεπαφή Keras επί της πλατφόρμας νευρωνικών δικτύων TensorFlow.

Η εργασία πρέπει να υλοποιηθεί σε σύστημα Linux και να υποβληθεί στις Εργασίες του e-class το αργότερο την Κυριακή 10/1/2021 στις 23.59.

Περιγραφή της εργασίας

Α) Κατασκευάστε νευρωνικό δίκτυο αυτοκωδικοποίησης εικόνων το οποίο θα περιλαμβάνει στρώματα συμπίεσης και αποσυμπίεσης ("bottleneck"). Θα πρέπει να πραγματοποιήσετε πειράματα εκπαίδευσης του δικτύου με διαφορετικές τιμές υπερπαραμέτρων [αριθμού συνελικτικών στρωμάτων, μεγέθους συνελικτικών φίλτρων, αριθμού συνελικτικών φίλτρων ανά στρώμα, αριθμού εποχών εκπαίδευσης (epochs), μεγέθους δέσμης (batch size), διάστασης συμπίεσης (latent dimension, default=10)] ώστε να ελαχιστοποιήσετε το σφάλμα (loss) αποφεύγοντας την υπερπροσαρμογή (overfitting). Τα δεδομένα του συνόλου εισόδου πρέπει να χωριστούν κατάλληλα σε σύνολο εκπαίδευσης (training set) και σε σύνολο επικύρωσης (validation set). Βάσει των πειραμάτων, επιλέγετε τη βέλτιστη δομή για το νευρωνικό δίκτυο, και το διάνυσμα συμπίεσης (latent vector) χρησιμοποιείται για την αναπαράσταση των εικόνων στον νέο διανυσματικό χώρο.

Β) Επεκτείνετε και χρησιμοποιείστε το παραδοτέο της πρώτης εργασίας για την εύρεση του πλησιέστερου γείτονα των εικόνων του συνόλου αναζήτησης στον **νέο** διανυσματικό χώρο (εξαντλητική αναζήτηση) καθώς και του πραγματικού (true) και του προσεγγιστικού (LSH) πλησιέστερου γείτονα στον αρχικό διανυσματικό χώρο: όλες οι αναζητήσεις γίνονται με τη μετρική Manhattan. Τα αποτελέσματα συγκρίνονται ως προς τον χρόνο αναζήτησης και το κλάσμα προσέγγισης στον **αρχικό** χώρο, δηλ. τη μέση απόσταση Manhattan προσεγγιστικού (NeuralNet ή LSH) / πραγματικού πλησιέστερου γείτονα από το διάνυσμα επερώτησης στον **αρχικό** χώρο.

Γ) Υλοποιήστε τη μετρική Earth Mover's Distance (EMD) που ανάγεται σε επίλυση προβλήματος Γραμμικού Προγραμματισμού (Linear Programming). Βρείτε εξαντλητικά τους 10 πλησιέστερους γείτονες και συγκρίνετε τον χρόνο εκτέλεσης και την «ορθότητα» έναντι της εξαντλητικής αναζήτησης του ερωτήματος Β. Για την «ορθότητα» χρησιμοποιείται η πληροφορία που δίνουν τα labels των εικόνων. Ως μέτρο ορθότητας ορίζεται το ποσοστό των πλησιέστερων γειτόνων που έχουν το ίδιο label με την εικόνα επερώτησης. Εκτελέστε πειράματα για διαφορετικό μέγεθος clusters κατά τον υπολογισμό της απόστασης EMD και σχολιάστε τα αποτελέσματα ως προς τον χρόνο και την «ορθότητα».

Δ) Πραγματοποιήστε συσταδοποίηση **Σ1** k-medians των εικόνων του συνόλου εισόδου στον νέο χώρο και

έστω **Σ2** στον αρχικό χώρο. Χρήση του παραδοτέου της 2ης εργασίας για κατηγοριοποίηση των εικόνων του συνόλου εισόδου και συσταδοποίηση **Σ3** βάσει αυτής. Επέκταση και χρήση του παραδοτέου της 1ης εργασίας για σύγκριση των τριών συσταδοποιήσεων ως προς το silhouette και την αποτίμηση της συνάρτησης-στόχου **στον αρχικό χώρο** (k ~ 10) με μετρική Manhattan.

Τα πειράματα και τα αποτελέσματα των ερωτημάτων Α έως Δ περιγράφονται και σχολιάζονται αναλυτικά στην αναφορά που παραδίδεται.

ΕΚΤΕΛΕΣΗ

Α) Το αρχείο που δίνεται στην είσοδο έχει την ακόλουθη μορφή:

```
[offset] [type]          [value]          [description]
0000     32 bit integer  <ΟΤΙΔΗΠΟΤΕ>      magic number
0004     32 bit integer  <ΑΡΙΘΜΟΣ ΕΙΚΟΝΩΝ>number of images
0008     32 bit integer  28               number of rows
0012     32 bit integer  28               number of columns
0016     unsigned byte   ??               pixel
0017     unsigned byte   ??               pixel
........
xxxx     unsigned byte   ??               pixel
```

Το αρχείο δίνεται μέσω παραμέτρου στη γραμμή εντολών. Η εκτέλεση θα γίνεται μέσω της εντολής:

```
$python reduce.py –d <dataset> -q <queryset> -o <output_file>
```

Το πρόγραμμα `reduce.py` χρησιμοποιεί το μοντέλο που έχει επιλεχθεί και εκπαιδευτεί βάσει των πειραμάτων για την επιλογή των βέλτιστων υπερπαραμέτρων.

Το αρχείο εξόδου έχει παρόμοια μορφή με το αρχείο εισόδου με αριθμό γραμμών **1** και αριθμό στηλών ίσο με το πλήθος των διαστάσεων. Οι συντεταγμένες των διανυσμάτων κανονικοποιούνται σε ακέραιες τιμές στο σύνολο {0,1, …, 25500} και φυλάσσονται σε 2 διαδοχικά bytes.

Β. Η είσοδος τροποποιεί τις προδιαγραφές του ερωτήματος Α. της 1ης εργασίας ως εξής:

```
$./search –d <input file original space> -i <input file new space> –q <query file original space> -s <query file new space> –k <int> -L <int> -o <output file>
```

(ο αριθμός πλησιέστερων γειτόνων που αναζητούνται Ν, λαμβάνεται σταθερός και ίσος με 1 και η αναζήτησης γειτόνων εντός ακτίνας R απενεργοποιείται).
Η έξοδος τροποποιεί τις προδιαγραφές του ερωτήματος Α. της πρώτης εργασίας ως εξής:

```
Query: image_number_in_query_set
Nearest neighbor Reduced: image_number_in_data_set
Nearest neighbor LSH: image_number_in_data_set
Nearest neighbor True: image_number_in_data_set
distanceReduced: <double>
distanceLSH: <double>
distanceTrue: <double>
```

```
...
Query: image_number_in_query_set
Nearest neighbor Reduced: image_number_in_data_set
Nearest neighbor LSH: image_number_in_data_set
Nearest neighbor True: image_number_in_data_set
distanceReduced: <double>
distanceLSH: <double>
distanceTrue: <double>

tReduced: <double ms>
tLSH: <double ms>
tTrue: <double ms>
Approximation Factor: <double>
```

Γ. Η εκτέλεση γίνεται με τις εξής παραμέτρους.

```
./search –d <input file original space> –q <query file original space> -l1
<labels of input dataset> -l2 <labels of query dataset> -o <output file> -EMD
```

Το αρχείο που δίνεται με την παράμετρο –l έχει την ακόλουθη δομή:

```
[offset] [type]          [value]          [description]
0000     32 bit integer  <ΟΤΙΔΗΠΟΤΕ>      magic number (MSB first)
0004     32 bit integer  60000            number of items
0008     unsigned byte   ??               label
0009     unsigned byte   ??               label
........
xxxx     unsigned byte   ??               label
```

Η έξοδος έχει την ακόλουθη μορφή

```
Average Correct Search Results EMD: <double>
Average Correct Search Results MANHATTAN: <double>
```

Δ. Η είσοδος τροποποιεί τις προδιαγραφές του ερωτήματος Β. της 1ης εργασίας ως εξής:

```
$./cluster –d <input file original space> -i <input file new space>
-n <classes from NN as clusters file> –c <configuration file> -o <output file>
```

Παραλείπεται η παράμετρος της μεθόδου ανάθεσης καθώς χρησιμοποιείται η επιλογή Classic, παραλείπεται και απενεργοποιείται η παράμετρος complete.

Το αρχείο configuration έχει την εξής μορφή:

```
number_of_clusters: <int>                 // K of K-medians
number_of_vector_hash_tables: <int>       // default: 3
number_of_vector_hash_functions: <int>    // default: 4
```

Το αρχείο στο οποίο δίνεται η κατηγοριοποίηση των εικόνων που προέκυψε από τη χρήση του παραδοτέου της 2ης εργασίας δίνεται με την παράμετρο –n και έχει την ακόλουθη μορφή:

```
CLUSTER-1 { size: <int>, image_numberA, ..., image_numberX}
.     .     .     .     .     .     .
CLUSTER-K { size: <int>, image_numberR, ..., image_numberZ}
```

Η έξοδος τροποποιεί τις προδιαγραφές του του ερωτήματος Β. της 1ης εργασίας ως εξής:

```
NEW SPACE
CLUSTER-1 {size: <int>, centroid: πίνακας με τις συντεταγμένες του centroid}
.     .     .     .     .     .     .
CLUSTER-K {size: <int>, centroid: πίνακας με τις συντεταγμένες του centroid}
clustering_time: <double> //in seconds
Silhouette: [s1,...,si,...,sK, stotal]
Value of Objective Function: <double>

ORIGINAL SPACE
CLUSTER-1 {size: <int>, centroid: πίνακας με τις συντεταγμένες του centroid}
.     .     .     .     .     .     .
CLUSTER-K {size: <int>, centroid: πίνακας με τις συντεταγμένες του centroid}
clustering_time: <double> //in seconds
Silhouette: [s1,...,si,...,sK, stotal]
Value of Objective Function: <double>

CLASSES AS CLUSTERS
Silhouette: [s1,...,si,...,sK, stotal]
Value of Objective Function: <double>
```

Επιπρόσθετες απαιτήσεις

1. Αρχείο (ή ενότητα στο Readme) που να σχολιάζει τα αποτελέσματα.
2. Το παραδοτέο πρέπει να είναι επαρκώς τεκμηριωμένο με πλήρη σχολιασμό του κώδικα και την ύπαρξη αρχείου Readme το οποίο περιλαμβάνει κατ' ελάχιστο: α) τίτλο και περιγραφή του προγράμματος, β) κατάλογο των αρχείων κώδικα και περιγραφή τους, γ) οδηγίες χρήσης του προγράμματος και δ) πλήρη στοιχεία των φοιτητών που το ανέπτυξαν.
3. Η υλοποίηση του προγράμματος θα πρέπει να γίνει με τη χρήση συστήματος διαχείρισης εκδόσεων λογισμικού και συνεργασίας (Git ή SVN) [ομάδες 2 ατόμων].