

K23γ: Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Χειμερινό εξάμηνο 2023-24

3^η 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 Παρασκευή 12/1/2024 στις 23.59.

Περιγραφή της εργασίας

A) Κατασκευάστε νευρωνικό δίκτυο αυτοκωδικοποίησης (autoencoder) εικόνων το οποίο θα περιλαμβάνει στρώματα συμπίεσης και αποσυμπίεσης. 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.

B) Επεκτείνετε και χρησιμοποιείτε τα παραδοτέα της πρώτης και της δεύτερης εργασίας για την εύρεση του πλησιέστερου γείτονα των εικόνων του συνόλου αναζήτησης στον **νέο** διανυσματικό χώρο (εξαντλητική αναζήτηση, GNNS, MRNG στον **νέο** χώρο) καθώς και του πραγματικού (true) και του προσεγγιστικού (LSH, Hypercube, GNNS, MRNG) πλησιέστερου γείτονα στον αρχικό διανυσματικό χώρο. Τα αποτελέσματα συγκρίνονται ως προς τον χρόνο αναζήτησης και το μέσο κλάσμα προσέγγισης, δηλ. τη μέση απόσταση προσεγγιστικού (NeuralNet ή LSH, Hypercube, GNNS, MRNG) / πραγματικού πλησιέστερου γείτονα από το διάνυσμα επερώτησης στον **αρχικό** χώρο.

Γ) Συσταδοποίηση **Σ1** k-means των εικόνων του συνόλου εισόδου στον νέο χώρο και **Σ2** στον αρχικό χώρο. Χρήση του παραδοτέου της 1^{ης} εργασίας για σύγκριση των συσταδοποιήσεων ως προς το silhouette και την αποτίμηση της συνάρτησης-στόχου.

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

ΕΚΤΕΛΕΣΗ

A) 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> -od <output_dataset_file> -oq <output_query_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. Οι συντεταγμένες των διανυσμάτων κανονικοποιούνται σε ακέραιες τιμές στο σύνολο {0,1, ..., 255} και φυλάσσονται σε διαδοχικά bytes.

Β και **Γ)** Είσοδος και έξοδος ακολουθούν τα πρότυπα της 1^{ης} και της 2^{ης} εργασίας. Στο ερώτημα Β υπολογίζεται για όλους τους αλγόριθμους αντί του μέγιστου, το μέσο κλάσμα προσέγγισης και τυπώνεται.

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.