Project "Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα" 2023-2024

(Χειμερινό εξάμηνο 2023-2024)

ΕΡΓΑΣΙΑ 3

Σαρλάς Λεωνίδας-Μάρκελλος (ΑΜ: 1115201700135)

Βεκρής Δημήτριος-Αλέξανδρος (ΑΜ: 1115202000021)

Github repo Εργασίας 3:

https://github.com/LeoSarlas/Software-Development-for-Algorithmic-Problems-Project3

ΠΕΡΙΓΡΑΦΗ ΑΡΧΕΙΩΝ/ΣΥΝΑΡΤΗΣΕΩΝ

Το βασικά αρχεία της Εργασίας 3 είναι:

reduce.py	Είναι το κυρίως αρχείο κώδικα που περιλαμβάνει την main() συνάρτηση. Περιλαμβάνει το import section στην αρχή του, με όλες τις χρησιμοποιούμενες βιβλιοθήκες/packages. Επίσης, περιέχει τις δηλώσεις όλων των συναρτήσεων που χρησιμοποιούνται για την υλοποίηση των ζητούμενων
1- **save_mnist_images(images, filename,rows,cols)**	Συνάρτηση που αποθηκεύει τις MNIST εικόνες σε ένα dataset σε binary μορφή.
2- **load_mnist_images(filename)**	Συνάρτηση ανάγνωσης εικόνων από ένα MNIST-like dataset.
3- **build_convolutional_autoencoder(conv_layers, filter_size, filters_per_layer, epochs, batch_size, latent_dim, dataset_filename)**	Συνάρτηση δημιουργίας και εκπαίδευσης του νευρωνικού δικτύου, με παραμέτρους που δίδονται από την main() ως ορίσματα. Μέσω αυτής της συνάρτησης και επαναλαμβανόμενων εκτελέσεων με διαφορετικούς συνδυασμούς τιμών παραμέτρων, προσδιορίστηκε το τελικό και βέλτιστο σετ τιμών παραμέτρων για το νευρωνικό δίκτυο. Επιστρέφει κάθε φορά με μορφή dictionary το συνδυασμό

	παραμέτρων που δοκιμάστηκε. Αυτός, αποθηκεύεται σε μια λίστα λεξικών στην main(), ώστε στη συνέχεια να εξαχθεί το βέλτιστο μοντέλο.
4- **main()**	Είναι η κύρια συνάρτηση του κώδικα, η οποία αρχικά διαβάζει τα ορίσματα από το terminal command, φορτώνει τις MNIST εικόνες και τις μετατρέπει σε numpy arrays για καλύτερο χειρισμό και επεξεργαία. Στη συνέχεια, το σχολιασμένο τμήμα κώδικα υλοποιεί τη ρουτίνα εκτέλεσης επαναλαμβανόμενων δοκιμών δημιουργίας και εκπαίδευσης του νευρωνικού δικτύου με διάφορους συνδυασμούς παραμέτρων. Έτσι, εξάγεται στη συνέχεια το "βέλτιστο" setup νευρωνικού μοντέλου.
MNIST_60k.idx3-ubyte	Είναι το dataset εικόνων χειρόγραφων ψηφίων, το οποίο δίδεται ως παράμετρος στο νευρωνικό δίκτυο προς μείωση της διάστασης των εικόνων από 28X28 σε vector (1,1,10). Το νέο παραγόμενο αρχείο 'NN_MNIST_60k.idx3-ubyte' είναι μειωμένης διάστασης και δίδεται ως είσοδος στους αλγορίθμους των Project 1 & 2, ώστε να δημιουργηθούν οι δομές LSH, HC, καθώς και οι αντίστοιχοι γράφοι αναζήτησης.
MNIST_10k.idx3-ubyte	Είναι το dataset εικόνων χειρόγραφων ψηφίων, το οποίο δίδεται ως παράμετρος στο νευρωνικό δίκτυο προς μείωση της διάστασης των εικόνων από 28X28 σε vector (1,1,10). Το νέο παραγόμενο αρχείο 'NN_MNIST_10k.idx3-ubyte' είναι μειωμένης διάστασης και δίδεται ως είσοδος στους αλγορίθμους των Project 1 & 2. Εκεί χρησιμοποιηθεί ως query file, από το οποίο αντλούνται τυχαία τα query search points για να ξεκινήσει η διαδικασία αναζήτησης κοντινότερων γειτόνων, σύμφωνα με τις εκφωνήσεις των Project 1 & 2.

Τα αρχεία των Εργασιών 1 & 2 έχουν τροποποιηθεί/επεκταθεί ώστε να υπολογίζουν επιπλέον όσα ζητούνται στο Β' υποερώτημα της Εργασίας. Αυτά βρίσκονται στους φακέλους '**Project-1-modified**' και '**Project-2-modified**'.

Output / Log files

Μετά από πολύωρες δοκιμές συνδυασμών παραμέτρων επάνω στο νευρωνικό δίκτυο, ο κώδικας 'reduce.py' εκτελείται κατευθείαν με εκπαίδευση του νευρωνικού με χρήση των καταλληλότερων παραμέτρων. Ωστόσο, παραμένει σχολιασμένο το τμήμα κώδικα που χρησιμοποιήθηκε για τον υπολογισμό αυτών των καταλληλότερων παραμέτρων, χρησιμοποιώντας κάθε φορά διαφορετικούς συνδυασμούς τιμών παραμέτρων μοντέλου.

Το printing της Εργασίας 3 γίνεται στο terminal, όσον αφορά στην εύρεση των καλύτερων παραμέτρων μοντέλου NN. Το printing των Εργασιών 1 & 2 γίνεται σε log files σύμφωνα με τις αντίστοιχες εκφωνήσεις.

Πειραματικά, καταλήξαμε σε αυτό τον συνδυασμό παραμέτρων δημιουργίας και εκπαίδευσης του νευρωνικού δικτύου:

```
# Best parameters combination
best_conv_layers=2
best_filter_size=8
best_filters_per_layer=8
best_epochs=45
best_batch_size=70
latent_dim=10
```

όπου το latent_dim είναι προκαθορισμένο ούτως ή άλλως από την εκφώνηση της Εργασίας 3.

Στον φάκελο 'resources' υπάρχουν τα αρχεία 'log.txt' και 'finder.py'. Το πρώτο περιέχει τα αποτελέσματα εκπαίδευσης του νευρωνικού δικτύου με διαφορετικούς συνδυασμούς παραμέτρων, ενώ το δεύτερο είναι script που δημιουργήσαμε ώστε να βρούμε γρήγορα τους 5 καλύτερους συνδυασμούς (ως προς validation loss) παραμέτρων για το νευρωνικό δίκτυο.

Στον φάκελο 'Run results' βρίσκονται ενδεικτικά αποτελέσματα εκτέλεσης των αλγορίθμων clustering επάνω στο reduced MNIST 10k dataset:

[1] **NN_Classic_output_file.txt**

[2] **NN_Reverse_search_LSH_output_file.txt**

[3] **NN_Random_projection_output_file.txt**

Επίσης, στον ίδιο φάκελο βρίσκονται και ενδεικτικά log files των εκτελέσεων των αλγορίθμων LSH, HC, GNNS και MRNG επάνω στο reduced MNIST 10k dataset:

[1] **lsh_results.txt**

[2] **cube_results.txt**

[3] **gnns_results.txt**

[4] **mrng_results.txt**

Αν παρατηρήσουμε κάθε log file, θα δούμε ότι για κάθε query εμφανίζεται ένα ζεύγος αποτελεσμάτων. Από αυτό το ζεύγος, το πρώτο κομμάτι αφορά τα αποτελέσματα σοτν αρχικό χώρο (28X28), ενώ το δεύτερο κομμάτι αφορά τα αποτελέσματα στον νέο χώρο (reduced) μετά το conversion του νευρωνικού δικτύου στο οποίο καταλήξαμε πειραματικά. Επίσης, στο τέλος κάθε log file, εχει υπολογιστεί το MAF συγκριτικά για την παλιά και για τη reduced διάσταση.

Makefile και εκτέλεση κώδικα

Εκτελούμε το 'reduce.py' μέσω της εντολής

" **python3 reduce.py -d MNIST_60k.idx3-ubyte -q MNIST_10k.idx3-ubyte -od NN_MNIST_60k.idx3-ubyte -oq NN_MNIST_10k.idx3-ubyte** "

ή

" python reduce.py -d MNIST_60k.idx3-ubyte -q MNIST_10k.idx3-ubyte -od NN_MNIST_60k.idx3-ubyte -oq NN_MNIST_10k.idx3-ubyte "

ανάλογα της Python edition μας.

Αφού παραχθούν τα νέα datasets εικόνων από το νευρωνικό δίκτυο (NN_MNIST_60k.idx3-ubyte & NN_MNIST_10k.idx3-ubyte), εκτελούμε κατά τα γνωστά τους κώδικες των Εργασιών 1 & 2, χρησιμοποιώντας το dataset των 10k εικόνων για λόγους εξοικονόμησης χρόνου. Με παρόμοια λογική, όμως, ο κώδικας εκτελείται και για το dataset των 60k εικόνων:

Εργασία 1

ΠΡΟΣΟΧΗ: το αρχείο 'input.idx3-ubyte' που βρίσκεται στο Project-1 (modified) folder, αντιστοιχεί στο 'NN_MNIST_10k.idx3-ubyte'. Έχει μετονομαστεί ακριβώς διότι ο αρχικός κώδικας της Εργασίας 1 δεχόταν ως terminal όρισμα ένα αρχείο "input.idx3-ubyte". Οπότε:

Random projections

- Για το LSH κομμάτι

" **make LSH** "

" **./lsh -d NN_MNIST_60k.idx3-ubyte -q NN_MNIST_10k.idx3-ubyte -k <int> -L <int> -o <output file> -N <number of nearest> -R <radius>** "

- Για το HC κομμάτι

" **make HC** "

" **./cube –d <input file> -q <query file> -k <int> -M <int> -probes <int> -o <output file> -N <number of nearest> -R <radius>** "

Σε κάθε διαφορετικό αλγόριθμο, παράγονται τα γνωστά (σύμφωνα με την Εργασία 1) outpout files, που περιλαμβάνουν τα αποτελέσματα clustering για το MNIST dataset **μειωμένης διάστασης**:

[1] **lsh_results.txt**

[2] **cube_results.txt**

, όπου σε κάθε ζεύγος query βλέπουμε το πρώτο μέρος να αφορά την παλιά διάσταση 28X28 και το δεύτερο μέρος τη νέα μειωμένη διάσταση μετά το conversion του νευρωνικού δικτύου.

Clustering

Με την εντολή <**make cluster**> γίνεται linking και compilation όλων των πηγαίων αρχείων και των header files.

Με την εντολή **" ./cluster -i NN_MNIST_60k.idx3-ubyte -c cluster.conf -o Classic_output_file.txt -m Classic -complete "** εκτελείται ο αλγόριθμος Lloyd's για το clustering με command line παραμέτρους που ρυθμίζουν τις παραμέτρους εκτέλεσης των συναρτήσεων και το output. (*)

Με την εντολή **" ./cluster -i NN_MNIST_60k.idx3-ubyte -c cluster.conf -o LSH_output_file.txt -m LSH -complete** " εκτελείται ο αλγόριθμος Reverse search by LSH για το clustering με command line παραμέτρους που ρυθμίζουν τις παραμέτρους εκτέλεσης των συναρτήσεων και το output. (*)

Με την εντολή **" ./cluster -i NN_MNIST_60k.idx3-ubyte -c cluster.conf -o Random_projection_output_file.txt -m Hypercube -complete "** εκτελείται ο αλγόριθμος Reverse search by random projection (Hypercube) για το clustering με command line παραμέτρους που ρυθμίζουν τις παραμέτρους εκτέλεσης των συναρτήσεων και το output. (*)

(*) Σε κάθε περίπτωση, η τελευταία παράμετρος [-complete] καθορίζει το αν θα εκτυπωθούν τα πλήρη αποτελέσματα στο output file (δηλαδή όλα τα σημεία σε κάθε cluster).

Σε κάθε διαφορετικό αλγόριθμο, παράγονται τα γνωστά (σύμφωνα με την Εργασία 1) outpout files, που περιλαμβάνουν τα αποτελέσματα clustering για το MNIST dataset **μειωμένης διάστασης**:

[1] **NN_Classic_output_file.txt**

[2] **NN_Reverse_search_LSH_output_file.txt**

[3] **NN_Random_projection_output_file.txt**

Εργασία 2

ΠΡΟΣΟΧΗ: το αρχείο 'input.idx3-ubyte' που βρίσκεται στο Project-2 (modified) folder, αντιστοιχεί στο 'NN_MNIST_10k.idx3-ubyte'. Έχει μετονομαστεί ακριβώς διότι ο αρχικός κώδικας της Εργασίας 2 δεχόταν ως terminal όρισμα ένα αρχείο "input.idx3-ubyte". Οπότε:

Για το compilation του κώδικα γράφων

" **make graph_search** "

Και για την εκτέλεσή του, αναλόγως του -m (= 1 ή 2) που καθορίζει αν θα εκτελεστεί ο GNNS ή ο MRNG αλγόριθμος:

" **./graph_search -d <input file> -q <query file> -k <int> -E <int> -R <int> -N <int> -l <int, only for Search-on-Graph> -m <1 for GNNS, 2 for MRNG> -o <output file>** "

Σε κάθε διαφορετικό αλγόριθμο, παράγονται τα γνωστά (σύμφωνα με την Εργασία 2) outpout files, που περιλαμβάνουν τα αποτελέσματα clustering για το MNIST dataset **μειωμένης διάστασης**:

[1] **gnns_results.txt**

[2] **mrng_results.txt**

, όπου σε κάθε ζεύγος query βλέπουμε το πρώτο μέρος να αφορά την παλιά διάσταση 28X28 και το δεύτερο μέρος τη νέα μειωμένη διάσταση μετά το conversion του νευρωνικού δικτύου.