Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Εργασία 3



Παπαχρήστου Δημήτρης-1115201500124

Κανταρτζής Θεόφιλος-1115201400061

Χειμερινό 2020-2021

Σύντομη Περιγραφή

1ο Ερώτημα:

Εκπαίδευση και αποτίμηση νευρωνικού δικτύου αυτοκωδικοποίησης εικόνων αριθμητικών ψηφίων.

Χρήση του κωδικοποιητή για τη δημιουργία νευρωνικού δικτύου κατηγοριοποίησης των εικόνων. Εκπαίδευση και αποτίμηση του νευρωνικού δικτύου ταξινόμησης. Εξαγωγή των δεδομένων σε νέο διανυσματικό χώρο.

Χρησιμοποιήθηκε η γλώσσα Python (3.8) και η προγραμματιστική διεπαφή Keras επί της πλατφόρμας νευρωνικών δικτύων TensorFlow.

Η εργασία αναπτύχθηκε σε περιβάλλον linux.

2ο Ερώτημα:

Σύγκριση απόδοσης lsh στον αρχικό διανυσματικό χώρο και true distance στον καινούριο μειωμένο.

3ο Ερώτημα:

Αποτίμηση και σύγκριση αποτελεσμάτων της μετρικής EMD με την μετρική manhattan. (Για emd χρησιμοποιήθηκε η βιβλιοθήκη PULP).

4ο Ερώτημα:

Συσταδοποίηση και σύγκριση shilouette και objective value στον αρχικό διανυσματικό χώρο, στον νέο μειωμένο και στα αποτελέσματα του δεύτερου παραδοτέου. (Συμπεριλάμβανεται στοω φάκελο code το αρχείο classification.py που εξάγει τα αποτελέσματα του 2ου παραδοτέου)

Κατάλογος Αρχείων

Στον φάκελο data βρίσκονται όλα τα δεδομένα που χρησιμοποιούνται και στα 4 ερωτήματα, καθώς και δεδομένα που εξάγουν τα ίδια τα ερωτήματα

Στον φάκελο output αποθηκεύονται τα αρχεία με τα αποτελέσματα των ερωτημάτων.

Στον φάκελο code :

1ο Ερώτημα:

- reduce.py

2ο Ερώτημα:

- Χρησιμοποιούνται τα ίδια αρχεία με του πρώτου παραδοτέου με τις κατάλληλες τροποποιήσεις για τα νέα ζητούμενα.Η main βρίσκεται στο lsh.cpp

3ο Ερώτημα:

- emd.py
- Επίσης χρησιμοποιούνται τα ίδια αρχεία με του πρώτου παραδοτέου με τις κατάλληλες τροποποιήσεις για τα νέα ζητούμενα.Η main βρίσκεται στο cluster.cpp

4ο Ερώτημα:

- Χρησιμοποιούνται τα ίδια αρχεία με του πρώτου παραδοτέου με τις κατάλληλες τροποποιήσεις για τα νέα ζητούμενα.Η main βρίσκεται στο cluster.cpp

Οδηγίες Χρήσης

1ο Ερώτημα:

$python reduce.py –d <dataset> -q <query set> -d <output_dataset_file> -oq <output_query_file>

Παράδειγμα για τα δεδομένα μας:

$python reduce.py -d ../data/train-images.idx3-ubyte -q ../data/t10k-images.idx3-ubyte -od train_latent -oq test_latent

2ο Ερώτημα:

$./search2 –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>

Παράδειγμα για τα δεδομένα μας:

$./search2 -d ../data/train-images.idx3-ubyte -i ../data/t10k-images.idx3-ubyte -q ../data/train_latent -s ../data/test_latent -k 14 -L 5 -o ../output/outputSearch2

3ο Ερώτημα:

$./search3 –d <input file original space> –q <query file original space> -l1

<labels of input dataset> -l2 <labels of query dataset> -o <output file> -EMD

Παράδειγμα για τα δεδομένα μας:

$./search3 -d ../data/train-images.idx3-ubyte -q ../data/t10k-images.idx3-ubyte -l1 ../data/train-labels.idx1-ubyte -l2 ../data/t10k-labels.idx1-ubyte -o ../output/outputSearch3 -EMD

4ο Ερώτημα:

$./cluster –d <input file original space> -i <input file new space>

-n <classes from NN as clusters file> –c <configuration file> -o <output file>

Παράδειγμα για τα δεδομένα μας:

./cluster -d ../data/train-images.idx3-ubyte -i ../data/train_latent -n ../data/classification_results.txt -c ./cluster.conf -o ../output/outputCluster

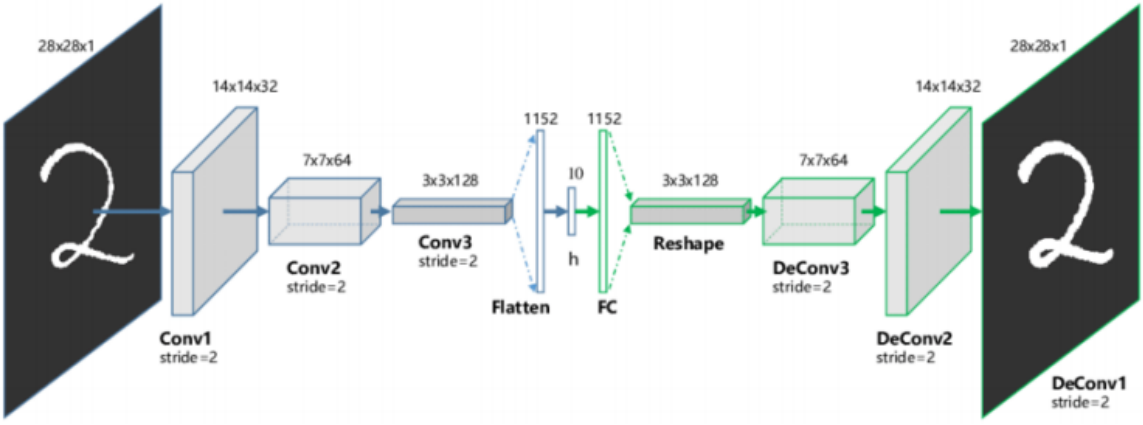
Σχολιασμός Κώδικα

Α΄ ΕΡΩΤΗΜΑ

Υλοποίηση “bottleneck”

Η υλοποίηση του autoencoder με στρώματα συμπίεσης και αποσυμπίεσης (“bottleneck”) βρίσκεται στο reduce.py. Η δημιουργία της δομής του δικτύου ακολουθεί την ίδια λογική με αυτή της δεύτερης εργασίας. Ο χρήστης επιλέγει αρχικά αριθμό στρωμάτων, μέγεθος batch size και αριθμό εποχών. Στην συνέχεια επιλέγει για κάθε layer τον αριθμό των φίλτρων και το μέγεθος αυτών. Επίσης, διαλέγει πότε θα προσθέσει pooling layers στον encoder. Για τον decoder δεν δίνουμε επιπλέον παραμέτρους καθώς εφαρμόζεται mirroring στα layers του, από τον encoder. Η προσθήκη αυτή της εργασίας είναι ότι μεταξύ encoder και decoder ο χρήστης επιλέγει την διάσταση συμπίεσης για την αναπαράσταση των εικόνων σε πολύ μικρότερη διάσταση απο την αρχική.

Σκοπός μας είναι να βρούμε τον βέλτιστο συνδυασμό υπερπαραμέτρων για την ελαχιστοποίηση του σφάλματος και την αποφυγή του overfitting.



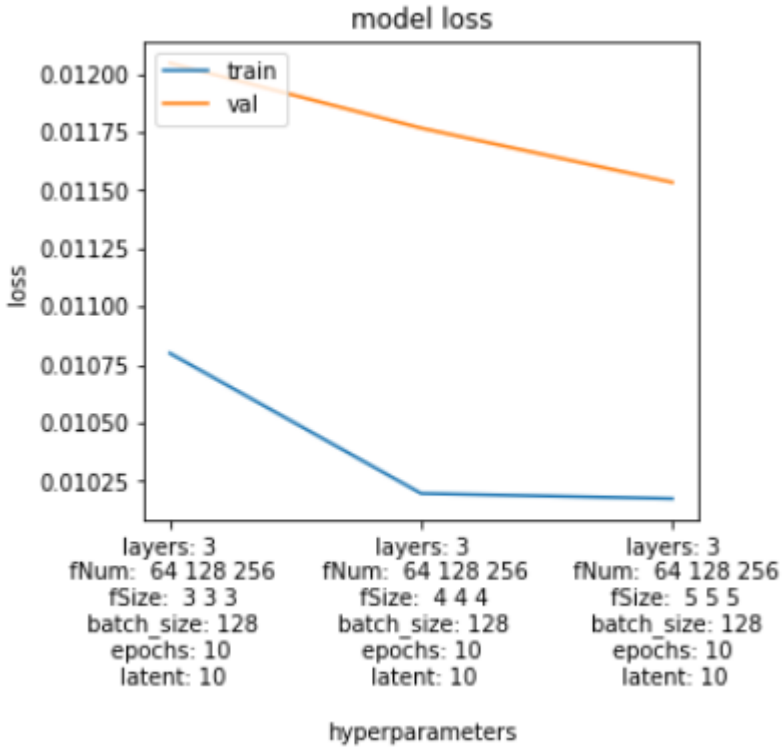
Οι υπερπαράμετροι που θα μελετηθούν είναι:

1.Μέγεθος Φίλτρων
2.Αριθμός Layers
3.Batch Size
4.Αριθμός Φίλτρων
5.Αριθμός Εποχών
6.Διάστασης Συμπίεσης

1. Μέγεθος Φίλτρων

Σε αυτό το βήμα θα μελετήσουμε την μεταβολή των loss για διάφορα μεγέθη φίλτρων. Όπως έχει αναφερθεί και παραπάνω μπορούμε να επιλέξουμε το μέγεθος των φίλτρων σε κάθε layer ξεχωριστά. Για την περιεκτικότητα των αποτελεσμάτων σε κάθε δοκιμή επιλέγουμε ένα μέγεθος το οποίο χρησιμοποιείται σε κάθε layer.

Κρατάμε fixed τις υπόλοιπες υπερπαραμέτρους (batch_size 128, εναρκτήριο αριθμό φίλτρων 64, εποχές 10, layers 3, διάσταση συμπίεσης 10) και μεταβάλουμε το μέγεθος φίλτρων από 3, 4, 5.



Σχολιασμός

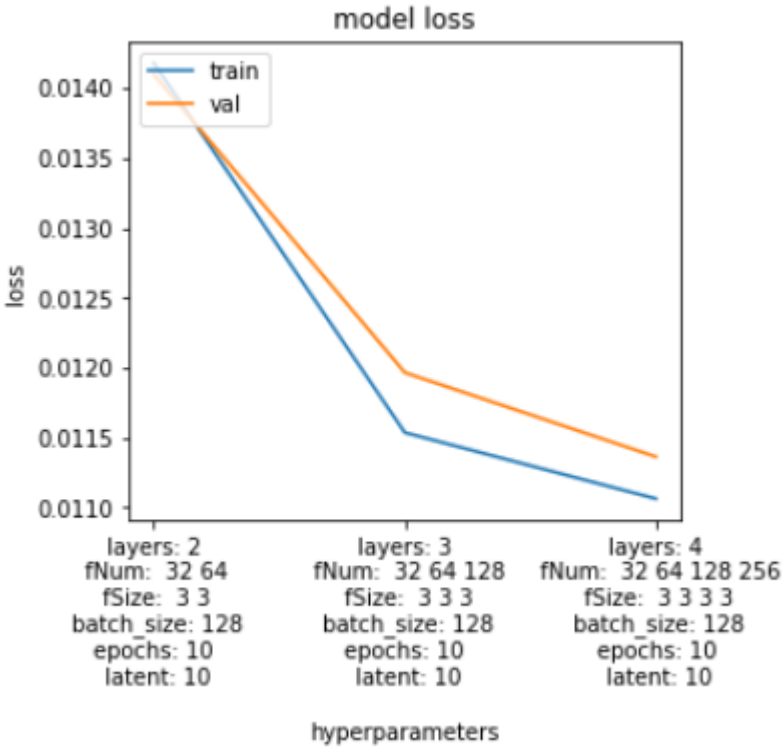
Αποτελεσμάτων:

1. Παρατηρούμε και στα τρία διαγράμματα ότι παίρνουμε τις πολύ καλες τιμές loss.
2. Ακόμα παρατηρούμε και στα τρία διαγράμματα ότι για φίλτρα μεγέθους 3x3 έχουμε και την μικρότερη διαφορά ανάμεσα στο train_loss και val_loss.

Συμπέρασμα: Επιλέγουμε φίλτρα μεγέθους 3x3.

2. Αριθμός Layers

Μελετήσαμε την μεταβολή των train_loss και val_loss με fixed τις τιμές των υπόλοιπων υπερπαραμέτρων (batch_size 128, εποχές 10, διάσταση συμπίεσης 10, εναρκτήριο αριθμό φίλτρων 32 και μέγεθος 3x3) με 2, 3, 4 layers.



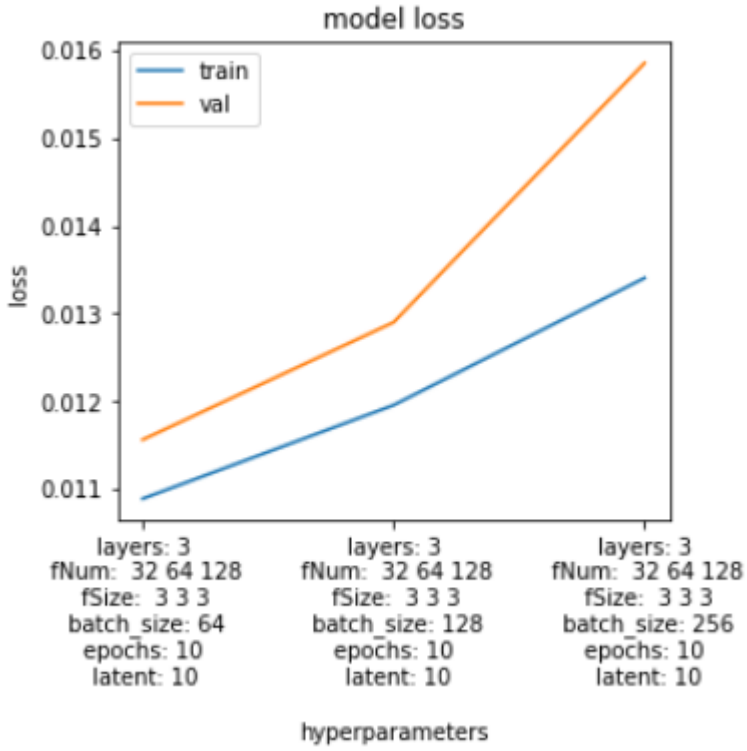
Σχολιασμός Αποτελεσμάτων:

1. Παρατηρούμε ότι όσο αυξάνονται τα layers τόσο μικρότερο loss έχουμε.
2. Ακόμα βλέπουμε η μείωση δεν είναι πολύ σημαντική από τα 3 στα 4 layers, ενώ με τα 4 έχουμε μεγαλύτερο χρόνο εκτέλεσης.

Συμπέρασμα: Τα καλύτερα αποτελέσματα στα παραπάνω πειράματα θεωρούμε ότι παρατηρούνται για 3 layers.

3. Batch Size

Μέχρι τώρα έχουμε συμπεράνει ότι ο πιο κατάλληλος αριθμός layer είναι 3 και μέγεθος φίλτρων 3x3. Θα εκτελέσουμε ένα πείραμα στο οποίο θα μεταβάλλουμε το batch size από 128, 256, 512, κρατώντας fixed τις υπόλοιπες παραμέτρους (layers 3, μέγεθος φίλτρων 3x3, εναρκτήριος αριθμός φίλτρων 32, εποχές 10, διάσταση συμπίεσης 10).



Σχολιασμός Αποτελεσμάτων:

1. Παρατηρούμε ότι όσο αυξάνεται το batch_size τόσο χειρότερες τιμές έχουμε για train_loss και val_loss.
2. Επίσης βλέπουμε ότι η διαφορά μεταξύ train_loss και val_loss για 128 είναι αρκετά μικρή παραμένοντας σε χαμηλές τιμές.

Συμπέρασμα: Τα καλύτερα αποτελέσματα στο παραπάνω πείραμα θεωρούμε ότι παρατηρούνται για batch_size 128.

4. Αριθμός Φίλτρων

Μέχρι τώρα έχουμε συμπεράνει ότι ο πιο κατάλληλος αριθμός layer είναι 3, μέγεθος φίλτρων 3x3, και batch size 128. Όπως και με το μέγεθος φίλτρων μπορούμε να επιλέξουμε τον αριθμό φίλτρων σε κάθε layer. Στα παρακάτω διαγράμματα θα μελετήσουμε τις μεταβολές των loss για διάφορες εναρκτήριες τιμές αριθμού φίλτρων στο πρώτο layer, όπου σε κάθε επόμενο layer διπλάσιαζεται αυτός ο αριθμός.

Π.χ. 1ο layer: 8 > 2ο layer: 16 > 3ο layer: 32 > 4ο layer: 64

Σχολιασμός Αποτελεσμάτων:

1. Παρατηρούμε ότι όσα περισσότερα είναι τα φίλτρα τόσο μικρότερα είναι τα train_loss και val_loss.
2. Επιπλέον, στο τρίτο πείραμα (64) η διαφορά μεταξύ train_loss και val_loss είναι αρκετά μικρότερη από τα άλλα δύο πειράματα.

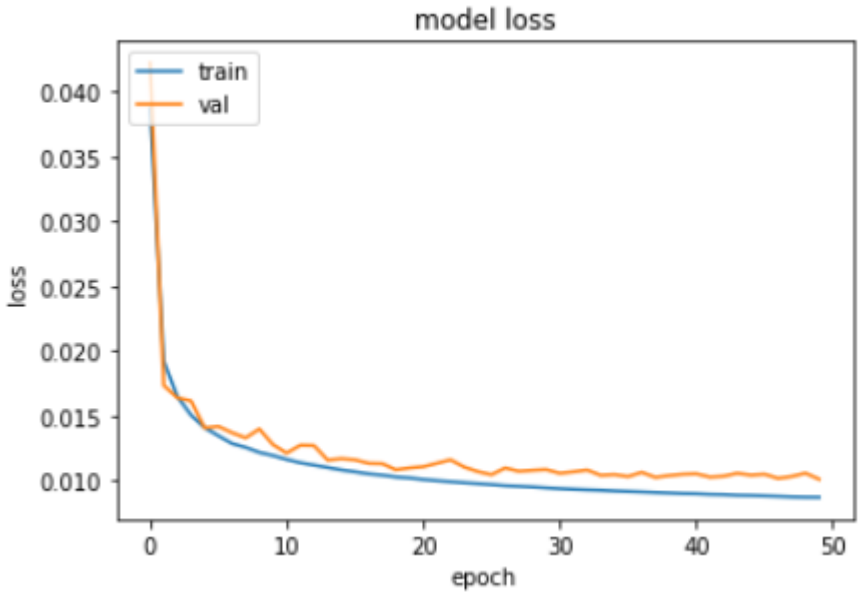
Συμπέρασμα: Τα καλύτερα αποτελέσματα στα παραπάνω πειράματα θεωρούμε ότι παρατηρούνται για αρχικό αριθμό φίλτρων 64.

5. Αριθμός Εποχών

Μέχρι τώρα οι υπερπαράμετροι που έχουμε συμπεράνει ότι δίνουν το βέλτιστο αποτέλεσμα για το ζητούμενο μας είναι οι εξής:

- Μέγεθος Φίλτρων : 3x3
- Αριθμός layers : 3
- Batch size : 128
- Αριθμός φίλτρων : 64 (στο πρώτο layer)

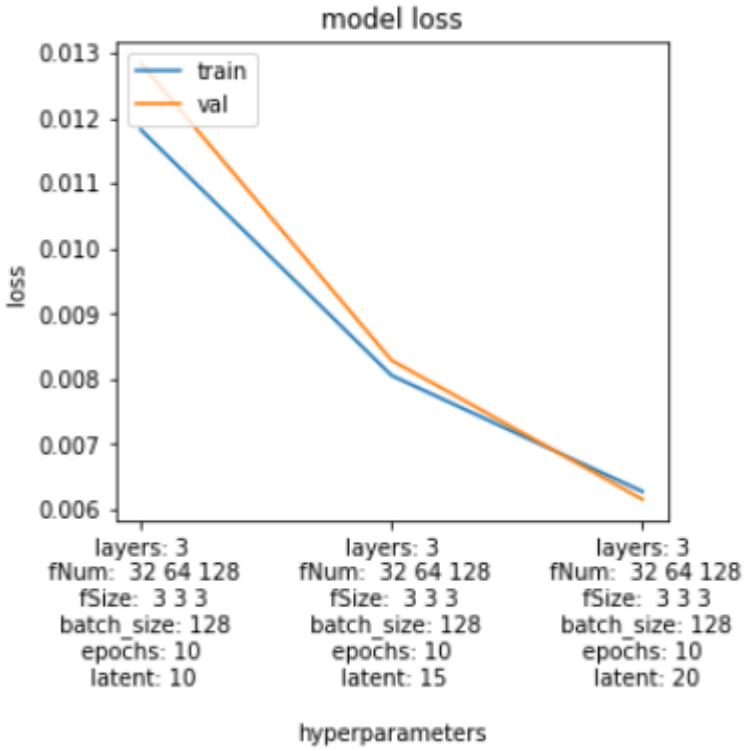
Οπότε εκπαιδεύσαμε το μοντέλο για 50 εποχές με τις παραπάνω τιμές για να δούμε που συγκλίνει.



Συμπέρασμα: Σύμφωνα με το παραπάνω διάγραμμα θεωρούμε ότι ο αριθμός εποχών για τον οποίο τα loss συγκλίνουν ικανοποιητικά είναι 10.

6. Διάσταση Συμπίεσης

Με βάση όλα τα προηγούμενα πειράματα , τώρα θα πειραματιστούμε ως προς την διάσταση συμπίεσης (bottleneck). Κρατώντας σταθερές τις υπόλοιπες υπερπαραμέτρους, δοκιμάζουμε για διάσταση 10, 15, 20.



Σχολιασμός Αποτελεσμάτων:

1. Παρατηρούμε ότι όσο μεγαλύτερη είναι η διάσταση συμπίεσης τόσο μικρότερες τιμές έχουμε στα train_loss και val_loss.
2. Το ζήτημά εδώ όμως είναι η μείωση των διαστάσεων ώστε να επιτευχθεί καλύτερη απόδοση στην αναζήτηση στην συνέχεια.

Συμπέρασμα: Με βάση τις δύο παραπάνω παρατηρήσεις κρατάμε για διάσταση συμπίεσης την τιμή 10.

Β΄ ΕΡΩΤΗΜΑ

Το εκτελέσιμο που δημιουργείται από το makefile ονομάζεται search2 για να ξεχωρίζεται με αυτό του τρίτου ερωτήματος.Οδηγίες εκτέλεσης έχουν περιγραφεί παραπάνω.

Υλοποιήθηκαν όλα τα ζητούμενα.Διαβάζονται τα αρχεία από τον φάκελο data. Διαβάζοντας τα αρχεία χρησιμοποιήσαμε το ίδιο format για την αποθήκευση των δεδομένων με αυτά του πρώτου παραδοτέου.Έτσι, χρησιμοποιήσαμε έτοιμες τις συναρτήσεις από το πρώτουπαραδοτέο για τον υπολογισμό των ζητούμενων.Όλοι οι υπολογισμοί και οι συγκρίσεις έχουν υλοποιηθεί με τον σωστό τρόπο και στου σωστούς διανυσματικούς χώρους όπως τους περιγράφει η εκφώνηση.Τα αποτελέσματα της εκτέλεσης αποθηκεύονται στον φάκελο output με ονομασία outputSearch2.

Γ΄ ΕΡΩΤΗΜΑ

Το εκτελέσιμο που δημιουργείται από το makefile ονομάζεται search3 για να ξεχωρίζεται με αυτό του δεύτερου ερωτήματος.Οδηγίες εκτέλεσης έχουν περιγραφεί παραπάνω.

Εκτός από αυτό χρησιμοποιείται το emd.py το οποίο κατά την εκτέλεση του ζητάει από τον χρήστη να δώσει πόσες εικόνες, από το train και test dataset αντίστοιχα, θα χρησιμοποιηθούν για την μετρική emd. Επίσης ζητάει να δώσει n όπου nxn το μέγεθος του κάθε cluster. Χρησιμοποιήθηκε η βιβλιοθήκη PULP. Το πρόβλημα γραμμικού προγραμματισμού λύθηκε όπως ζητείται από την εκφώνηση.Τέλος, γράφει στο αρχείο data/emd_results.txt την τιμή αποτέλεσμα καθώς και το μέγεθος των υποσυνόλων που χρησιμοποιήθηκαν από τα train και test set.

Για το δεύτερο κομμάτι η search3.cpp διαβάζει το emd_results.txt και υπολογίζει αποτελέσματα στην Manhattan μετρική (όπως έχει υλοποιηθεί από το πρώτο παραδοτέο) στο ίδιο υποσύνολο που χρησιμοποίησε η emd.Τέλος εκτυπώνει τα αποτελέσματα στο output/outputSearch3

Δ΄ ΕΡΩΤΗΜΑ

Το εκτελέσιμο που δημιουργείται από το makefile ονομάζεται cluster. Οδηγίες εκτέλεσης έχουν περιγραφεί παραπάνω.

Υλοποιήθηκαν όλα τα ζητούμενα.Διαβάζονται τα αρχεία από τον φάκελο data. Διαβάζοντας τα αρχεία χρησιμοποιήσαμε το ίδιο format για την αποθήκευση των δεδομένων με αυτά του πρώτου παραδοτέου.Έτσι, χρησιμοποιήσαμε έτοιμες τις συναρτήσεις από το πρώτο παραδοτέο για τον υπολογισμό των ζητούμενων.Όλοι οι υπολογισμοί και οι συγκρίσεις έχουν υλοποιηθεί με τον σωστό τρόπο και στου σωστούς διανυσματικούς χώρους όπως τους περιγράφει η εκφώνηση.Τα αποτελέσματα της εκτέλεσης αποθηκεύονται στον φάκελο output με ονομασία outputCluster.

Η objective function υλοποιήθηκε όπως περιγράφεται στις διαφάνειες.