ΚΩΝΣΤΑΝΤΙΝΟΣ ΚΟΚΟΝΟΣ 1115201400287 sdi1400287@di.uoa.gr	ΑΝΑΣΤΑΣΙΟΣ ΠΑΝΤΑΖΟΠΟΥΛΟΣ 1115201500119 sdi1500119@di.uoa.gr

Τα δύο μέρη της εργασίας 2 έτρεξαν σε linux 20.04 με έκδοση python 3.8, 8GB ram και SSD, χωρίς να χρησιμοποιηθεί η βιβλιοθήκη cuda.
Github: https://github.com/konkos2810/Projectv2

Α Μέρος
python3 autoencoder.py -d train-images-idx3-ubyte

Για το α μέρος της εργασίας εχουμε φτιάξει τα εξής αρχεία που περιέχουν τις συναρτήσεις μας:

1)autoencoder.py
Περιέχει την main(),γίνεται σπάσιμο του dataset σε train και validation,φτιάχνεται το μοντέλο μας,για κάθε μοντέλο που δημιουργούμε φτιάχνουμε και ένα γράφημα που μας δέιχνει το loss kai το validation loss για να είμαστε σίγουροι ότι δεν κάνουμε under ή overfitting,και τέλος ρωτάμε απο τον χρήστη αν θέλει να επαναλάβει την διαδικασία από την αρχή,αν θέλει να εμφανιστούν τα γραφήματα για κάθε μοντέλο καθώς και αν θέλει να αποθηκεύσει το τελευταίο μοντέλο.

2)dataset.py
Περιέχει την συνάρτηση dataset η οποία παίρνει από την γραμμή εντολών το αρχείο και διαβάζει τον magic number, number of images, number of rows και number of colunns και δημιουργεί μία λίστα, την dataset η οποία περιέχει όλες τις εικόνες.Επίσης γίνεται έλεγχος αν δίνουμε το σωστό αρχείο ελέγχοντας το flag.

3)parametroi.py
Περιέχει την συνάρτηση parameters() η οποία ζητά από τον χρήστη όλες τις υπερπαραμέτρους,καθώς και το number of layers για τον autoencoder μας.Αποδεκτές τιμές για τον αριθμό των layers είναι οι περιττοί αριθμοί ξεκινώντας από το 5.

4)encoder.py
Περιέχει την συνάρτηση encoder,η οποία μας δημιουργεί τον encoder μας.

5)decoder.py
Περιέχει την συνάρτηση dencoder,η οποία μας δημιουργεί τον dencoder μας.

6)modelo.py
Περιέχει την συνάρτηση modelo η οποία μας δημιουργεί ουσιαστικά το μοντέλο μας χρησιμοποιώντας τις παραπάνω συναρτήσεις.

7)graphs.py
Βρίσκεται η συνάρτηση graph_for_loss η οποία μας επιστρέφει το plt,για να φτιάχνουμε τα γραφήματα για τα μοντέλα μας.

Μέσα στον φάκελο έχουμε δημιουργήσει και έναν φάκελο(Graphs) όπου αποθηκεύσαμε το γράφημα μας.

Το μοντέλο καθώς και τα βάρη που υπάρχουν επίσης στον φάκελο έχουν δημιουργηθεί με τις εξής τιμές υπερπαραμέτρων: number_of_layers = 7, filter_size = 3, number_of_layers για το 1 layer = 4, number_of_layers για το 2 layer =8 , number_of_layers για το 3 layer = 16, number_of_layers για

το 4 layer =32 , number_of_layers για το 5 layer =16 , number_of_layers για το 6 layer = 8, number_of_layers για το 7 layer =4 , epochs_number = 10, batch_size = 128.
Β μέρος
python3 classification.py -d train-images-idx3-ubyte -dl train-labels-idx1-ubyte -t t10k-images-idx3-ubyte -tl t10k-labels-idx1-ubyte -model autoencoder_model.h5

Για το β μέρος της εργασίας εχουμε χρησιμοποιήσει πολλές απο τις συναρτήσεις του α μέρους και έχουμε δημιουργήσει και κάποια καινιούργια αρχεία,τα οποία είναι τα εξής:

1)dataset.py
Ίδιο με το α μερος

2)dataset_labels.py
Περιέχει την συνάρτηση dataset_labels η οποία διαβάζει από τα αρχεία των labels όλα τα labels και τα αποθηκεύει σε μία λίστα

3)encoder.py
Ιδιο με το α μέρος

4)parametroi.py
Ιδιο με το α μέρος

5)graphs.py
Ιδιο με το α μέρος

6)fc.py
Χρησιμοποιούμε την συνάρτηση fc για να κάνουμε flattening των στοιχείων που μας επιστρέφει ο encoder

7)modelo.py
Ουσιαστικά εδώ δημιουργούμε και κάνουμε training το μοντέλο μας χρησιμοποιώντας τα layers απο τον encoder του α μέρους της εργασίας.Στην αρχή κάνουμε training χωρίς αυτά τα layers και στην συνέχεια τα προσθέτουμε και αυτά ώστε να βγεί το τελικό μας μοντέλο.

8)classification.py
Σε αυτό το αρχείο γίνεται η σύνθεση όλων των παραπάνω συναρτήσεων ώστε να κάνουμε ένα classification. Αρχικά ελέγχουμε αν έχουμε τα σωστά αρχεία κάνοντας έλεγχο των flags.Στην συνέχεια διαχωρίζουμε το dataset σε train kai validation.Ακολουθούμε την ίδια διαδικασία με το α μέρος,δηλαδή φτίαχουμε το μοντέλο μαζί με το γράφημα,ρωτάμε τον χρήστη αν θέλει να φτιάξει καινιούργιο μοντέλο,αν θέλει να εμφανιστούν τα γραφήματα από τα μοντέλα μας,αν θέλει να κάνει predict τις εικόνες που υπάρχουν στο test dataset και του εμφανίζονται και κάποια predictions σε φωτογραφία,η οποία μαζί με τα γραφήματα αποθηκεύονται στο φάκελο Graphs.

Το μοντέλο καθώς και τα βάρη που υπάρχουν επίσης στον φάκελο έχουν δημιουργηθεί με τις εξής τιμές υπερπαραμέτρων: number_of_layers = 7, filter_size = 3, number_of_layers για το 1 layer = 4, number_of_layers για το 2 layer =8 , number_of_layers για το 3 layer = 16, number_of_layers για το 4 layer =32 , number_of_layers για το 5 layer =16 , number_of_layers για το 6 layer = 8, number_of_layers για το 7 layer =4 , epochs_number = 10, batch_size = 128.Θα πρέπει να βάλουμε ακριβώς τις ίδιες υπερπαραμέτρους με το α μερος με εξαίρεση το epochs_number και το batch_size.

Παρατηρήσεις για την εργασία συνολικά:

Στο α μέρος το loss και το validation loss μετά από κάποια epochs συγκλίνουν,στο β μέρος της εργασίας η διαφορά τους είναι πολύ μεγάλη όπως θα δείτε και από το γράφημα.Βγάλαμε το συμπέρασμα οτί ίσως φταίει ο χαμηλός αριθμός των epochs που χρησιμοποιούμε στο β μέρος.