



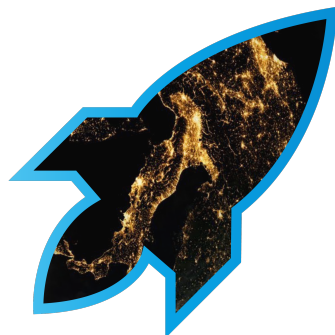
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
— ΙΔΡΥΘΕΝ ΤΟ 1837 —

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΑΛΓΟΡΙΘΜΙΚΑ ΠΡΟΒΛΗΜΑΤΑ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2020

2Η ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΑΠΟΤΙΜΗΣΗ ΝΕΥΡΩΝΙΚΟΥ
ΔΙΚΤΥΟΥ ΑΥΤΟΚΩΔΙΚΟΠΟΙΗΣΗΣ ΕΙΚΟΝΩΝ
ΑΡΙΘΜΗΤΙΚΩΝ ΨΗΦΙΩΝ



Αριθμός Μητρώου(ΑΜ):

1115201700217

1115201700203

Ονοματεπώνυμο:

Ορέστης ΣΤΕΦΑΝΟΥ

Λεωνίδας ΕΦΡΑΙΜ

ΑΚΑΔΗΜΑΪΚΗ ΧΡΟΝΙΑ 2020-2021

ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	3
2	ΜΕΡΟΣ Α' AUTOENCODER	4
3	ΜΕΡΟΣ Β' CLASSIFICATION	6

ΕΙΣΑΓΩΓΗ

Σε αυτή την εργασία κληθήκαμε να υλοποιήσουμε ένα νευρονικό δίκτυο αυτοκωδικοποίησης ψηφιακών εικόνων. Για την υλοποίηση χρησιμοποιήσαμε την γλώσσα Python με την βοήθεια των βιβλιοθηκών Keras και Tensorflow. Έχτος από αυτά χρησιμοποιήσαμε και το Google Collab το οποίο μας παρείχε επεξεργαστική ισχύ για τους μεγάλους υπολογισμούς που χρειαστήκαμε μέσω των GPU που μας παρείχε. Στο πρώτο μέρος δημιουργήσαμε τον encoder και τον εκπαιδεύσαμε ενώ στην συνέχεια στο δεύτερο μέρος υλοποιήσαμε και κατηγοριοποίηση στον encoder μας.

ΜΕΡΟΣ Α' AUTOENCODER

Για την υλοποίηση του χρησιμοποιήσαμε τις παρακάτω συναρτήσεις

- **read_data** Η συνάρτηση αυτή πέρνει ως όρισμα ένα συμπιεσμένο binary αρχείο με την μορφή που ζητά η εκφώνηση, διαβάζει τα δεδομένα από το αρχείο και στην συνέχεια τα επιστέφει. Αρχικά αποσυμπιέζει το αρχείο και επιστέφει ένα δείκτη στην αρχή του αρχείου. Στην συνέχεια διαβάζει ανα 4 bytes διαδοχικά για να διαβάσει το magic number, τον αριθμό των εικόνων, το μήκος και το πλάτος της κάθε εικόνας. Τέλος αφού γνωρίζει τον αριθμό των εικόνων που περιέχοντε στο αρχείο διαβάζει όλες τις εικόνες και τις βάζει σε μια δομή με το όνομα data και τις επιστρέφει
- **encoder** Αυτή η συνάρτηση είναι υπεύθυνή για την κωδικοποίηση του input στα convolution layers. Εδώ χρησιμοποιήσαμε την συνάρτηση από τις διάφανιες του φροντιστηρίου που μας δώθηκαν από τον Κ.Χαμόδρακα. Ο encoder δημιουργεί 4 layers όπου το πρώτο είναι ένα convolution layer με 32 φίλτρα με μέγεθος 3x3. Το επόμενο layer έχει 64 φίλτρα με μέγεθος 3x3 και τα επομένα δύο layers 128 και 256 φίλτρα με μέγεθος 3x3.
- **decoder** Η συνάρτηση αυτή κάνει την αντίστροφη διαδικασία από τον encoder δηλαδή αποκωδικοποιά το input. Πάλι εδώ χρησιμοποιήσαμε την συνάρτηση από τις διάφανιες του φροντιστηρίου που μας δώθηκαν από τον Κ.Χαμόδρακα. Στο τέλος την αποκωδικοποίησης χρησιμοποιήτε η σιγμοειδής συνάρτηση για να κανονοποιήσουμε τα τις τιμές στο διάστημα 0 με 1

Το πρόγραμμα αρχικά ελέγχει αν έχουν δοθεί σωστά τα ορίσματα. Στην συνέχεια ζητά από τον χρήστη να δώσει το batch size και τον αριθμό των epoch. Ακολούθως διαβάζει το αρχείο που δώθηκε σαν όρισμα και χωρίζει τα δεδομένα σε train set και validation set για την εκπαίδευση και στην συνέχεια την επαλήθευση του δικτύου. Τέλος δίνει την δυνατότητα στον χρήστη να επιλέξει αν θέλει να τερματίσει το πρόγραμμα ή να αποθηκεύσει τα δεδομένα ή να παρουσιάσει την γραφική παράσταση ή να επαναληφθεί ο αλγορίθμος εκμάθειας με νέες υπερπαραμέτρους

ΜΕΡΟΣ Β' CLASSIFICATION

Στο δεύτερο μέρος της εργασίας κληθήκαμε να υλοποιήσουμε ένα νευρωνικό δίκτυο κατηγοριοποίησης εικόνων βασισμένοι στην κωδικοποίηση(encoder) που κάναμε στο πρώτο μέρος της εργασίας. Το πρόγραμμα `classification.py` δέχεται συμπιεσμένα αρχεία με την μορφή που δώθηκαν στα πλέσια της εργασίας. Η συνάρτηση για την ανάγνωση των συμπιεσμένων αρχείων είναι η ίδια με το πρώτο μέρος της εργασίας (`read_data`). Αλλα χρειήστηκε ακόμα μια συνάρτηση για την ανάγνωση των αρχείων με τις ετικέτες οπότε με λίγες τροποποιήσεις δημιουργήσαμε την συνάρτηση

- **read_labels** Η συνάρτηση αυτή πέρνει ως όρισμα ένα συμπιεσμένο binary αρχείο με την μορφή που ζητά η εκφώνηση, διαβάζει τα δεδομένα από το αρχείο και στην συνέχεια τα επιστέφει. Αρχικά αποσιμπιέζει το αρχείο και επιστέφει ένα δείκτη στην αρχή του αρχείου. Στην συνέχεια διαβάζει ανα 4 bytes διαδοχικά για να διαβάσει το magic number, τον αριθμό των ετικετών. Τέλος αφού γνωρίζει τον αριθμό των ετικετών που περιέχοντε στο αρχείο διαβάζει όλες τις ετικέτες και τις βάζει σε μια δομή με το όνομα labels και τις επιστρέφει

Επίσεις χρησιμοποιούμε τις συνάρτησεις encoder από το πρώτος μέρος επομένως η περιγραφή της είναι η ίδια. Άλλες συναρτήσεις που χρησιμοποιούνται είναι οι παρακάτω

- **printHelp** Η συνάρτηση αυτή πέρνει τυπώνει ένα μήνυμα το οποίο υποδεικνεί την λάθος καταχώρηση ορισμάτων στο πρόγραμμα
- **fc** Η συνάρτηση αυτή πέρνει σαν όρισμα τον endocer και προσθέτει ένα πλήρως συνδεδεμένο στρώμα για να ενωθεί με το στρώμα εξόδου

Το πρόγραμμα αρχικά ελέγχει αν έχουν δωθεί σωστά τα ορίσματα. Στην συνέχεια ζητά από τον χρήστη να δώσει το batch size και τον αριθμό των epoch. Ακολούθως διαβάζει τα αρχεία που δώθηκαν σαν ορίσματα και χωρίζει τα δεδομένα σε train set και validation set για την εκπαίδευση. Παράλληλα διαβάζει τις ετικέτες για τα αντίστοιχα train και test δεδομένα τα οποία τα αποθηκεύει στις αντίστοιχους τύπους δεδομένων. Στην συνέχεια δημιουργεί ένα νέο μοντέλο βάση με το μοντέλο που δώκε σαν όρισμα αντιγράφοντας όλα τα layers του στο καινούριο μοντέλο και προσθέτοντας ένα πλήρως συνδεδεμένο στρώμα για να υπάρξει σύνδεση με το στρώμα εξόδου. Το καινούριο μοντέλο που δημιουργήθηκε με την κατηγοριοποίηση είναι ετοιμό για εκπαίδευση και επάληθευση. Ανάλογα με την επιλογή του χρήστη μπορούμε να επαναλάβουμε τις μετρήσεις, να παρουσιάσουμε τις γραφικές παραστάσεις με τα error και το accuracy του δικτύου ή να προχωρήσουμε με καταγοριοποίηση