Νευρωνικά Δίκτυα – Βαθιά Μάθηση

1η Ενδιάμεση Εργασία

Νοέμβριος 2023

Γρηγορίου Στέργιος 9564 (ΤΗΜΜΥ)

Σύνολο Δεδομένων και Προεπεξεργασία:

Στην εργασία χρησιμοποιήθηκε η βάση δεδομένων CIFAR-10 [1] που αποτελείται από 60.000 εικόνες 32x32x3 uint8 (RGB). Οι πρώτες 50.000 εικόνες απαρτίζουν το σύνολο εκπαίδευσης και οι υπόλοιπες 10.000 το σύνολο δοκιμής. Τα δεδομένα (οι εικόνες) χωρίζονται σε 10 κλάσεις {airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck} και γενικά η βάση είναι απολύτως ισορροπημένη (κάθε κλάση έχει 5000 παρατηρήσεις στο σύνολο εκπαίδευσης και 1000 στο σύνολο δοκιμής). Σε αυτή την εργασία, δεν διενεργήθηκε κάποια προεπεξεργασία των δεδομένων, πέρα από την μετατροπή των τιμών των πίξελ σε float στο [0,1].

Κώδικας:

Αρχικά υλοποιήθηκαν δύο συναρτήσεις για την ανάγνωση των δεδομένων. Η πρώτη *unpickle(file)* βασίζεται (αντιγράφηκε από) στις οδηγίες χρήσης που συνοδεύουν την βάση [1] με μόνη αλλαγή στην παράμετρο encoding, για ευχρηστία στο υπόλοιπο πρόγραμμα, από ‘bytes’ σε ‘latin1’. Αυτό που επιτυγχάνει ουσιαστικά είναι να μετατρέπει τα δεδομένα από pickle αρχεία σε λεξικά. Επειδή όμως η βάση αποτελείται από 6 batches από 10.000 παρατηρήσεις, χρειάστηκε και άλλη μία συνάρτηση *load_data(path)* που χρησιμοποιεί την *unpickle* για να διαβάσει τα δεδομένα. Μετά την κλήση της έχουμε 5 NumPy.arrays [4]: τα δεδομένα εκπαίδευσης, τις ετικέτες τους, τα δεδομένα δοκιμής, τις ετικέτες τους και τα ονόματα των κλάσεων.

Έπειτα υλοποιήθηκε μία συνάρτηση *train_model(Xs,ys,model_type,k=0)* για την εκπαίδευση των μοντέλων με ορίσματα: τα δεδομένα εκπαίδευσης, τους στόχους εκπαίδευσης, τον τύπο του μοντέλου (k-NN ή nearest centroid) και την υπερπαράμετρο k. Η συνάρτηση χρησιμοποιεί κυρίως την βιβλιοθήκη sklearn.neighbors [2], για να προσαρμόσει τα ζητούμενα μοντέλα στα δεδομένα της εισόδου. Επίσης υπολογίζει και εκτυπώνει τον χρόνο που απαιτείται για αυτήν την διαδικασία. Τέλος επιστρέφει το προσαρμοσμένο μοντέλο, καθώς και το όνομα του που θα χρησιμοποιηθεί στην οπτικοποίηση των αποτελεσμάτων.

Η τέταρτη και τελευταία συνάρτηση που υλοποιήθηκε *evaluate_model(model,Xs,ys,model_name,label_names)* χρησιμοποιεί το προσαρμοσμένο μοντέλο της προηγούμενης για να το αξιολογήσει σε δεδομένα που δεν έχει ξαναδεί. Στην υλοποίηση αυτής της συνάρτησης, γίνεται πάλι χρήση της sklearn [2]. Αφού υπολογίσει τις εκτιμώμενες κλάσεις του νέου δείγματος και εκτυπώσει τον χρόνο που απαιτείται για αυτούς τους υπολογισμούς, υπολογίζει μια σειρά από μετρικές: precision, recall, f1_score και accuracy

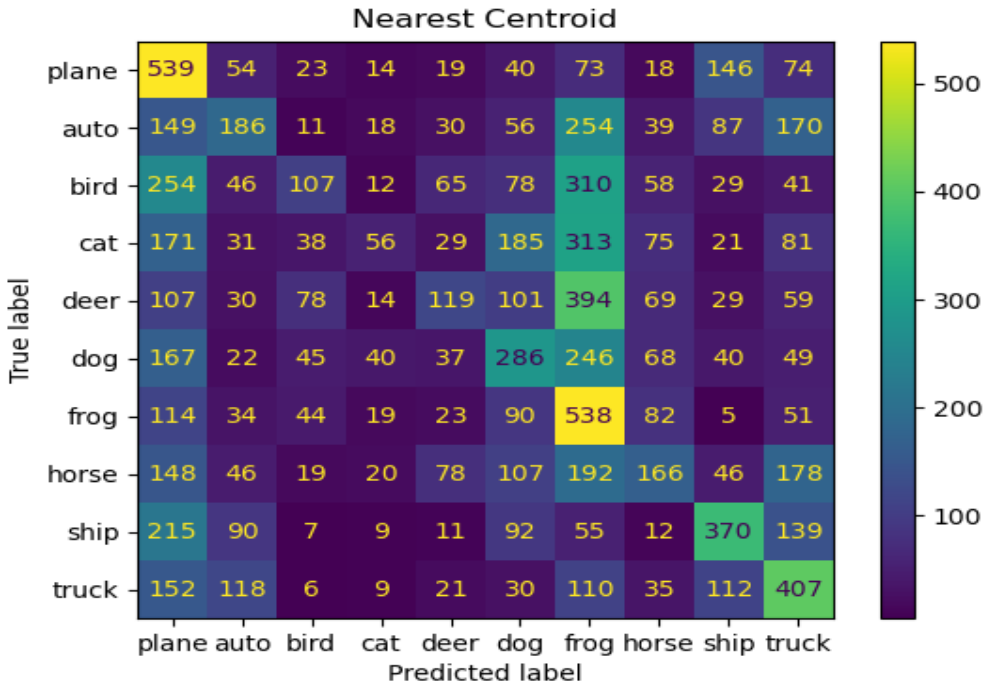
(ή μέσο recall στην προκειμένη) για κάθε κλάση ως προς της άλλες καθώς και τον μέσο όρο όλων των κλάσεων. Έπειτα τα δεδομένα αυτά εκτυπώνονται ως ένα panda.DataFrame [3]. Τέλος υπολογίζει και εκτυπώνει των πίνακα σύγχυσης του μοντέλου.

Στο κυρίως κομμάτι του κώδικα, ορίζεται η μεταβλητή path (σειρά 106) ως η διεύθυνση του φακέλου στον οποίο είναι αποθηκευμένη η CIFAR-10 στον υπολογιστή (για να «τρέξει» το πρόγραμμα πρέπει να αλλάξει αυτή η τιμή κατάλληλα). Διαβάζονται τα δεδομένα, γίνεται η μετατροπή των εικόνων σε float στο [0,1] κι έπειτα εκπαιδεύονται (στο train set) και αξιολογούνται (στο test set) τα 3 μοντέλα που ζητάει η άσκηση (nearest centroid ,1-NN,3-NN).

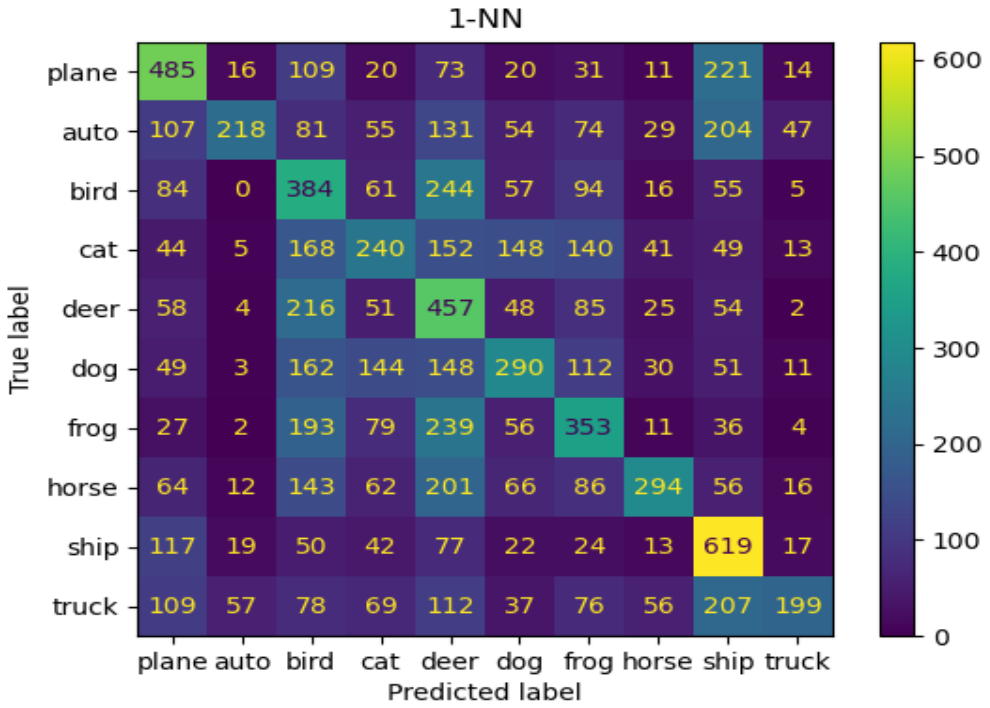
Αποτελέσματα:

Γενικά μετά από εκπαίδευση σε όλο το train set και επαλήθευση στο test set τα τρία μοντέλα είχαν μέτρια έως κακή απόδοση. Και τα τρία έχουν ακρίβεια πάνω από 10% που σημαίνει ότι είναι καλύτερα από το να μαντεύουν απλά την κλάση, αλλά είναι πολύ μακριά από το να μπορούμε να πούμε ότι δουλεύουν ικανοποιητικά. Συγκεκριμένα και βάσει ακρίβειας αλλά και βάσει μέσου f1_score το καλύτερο είναι το μοντέλο του πλησιέστερου γείτονα με 35,39% και 0,3495 αντίστοιχα, ακολουθεί το μοντέλο των 3 πλησιέστερων γειτόνων με 33,03% και 0,3192 και τελευταίο είναι το μοντέλο πλησιέστερου κέντρου με 27,74% και 0,2541 αντίστοιχα. Βάσει χρόνου εκπαίδευσης όμως το μοντέλο πλησιέστερου κέντρου είναι πάνω από 30 φορές πιο γρήγορο από τα άλλα δύο, ενώ το μοντέλου του πλησιέστερου γείτονα είναι περίπου 9% πιο γρήγορο από αυτό με τους 3 γείτονες. Συγκεκριμένα στο δικό μου σύστημα, το πρώτο τρέχει σε κάτω από 1 δευτερόλεπτο ενώ τα άλλα δύο χρειάζονται περίπου μισό λεπτό.

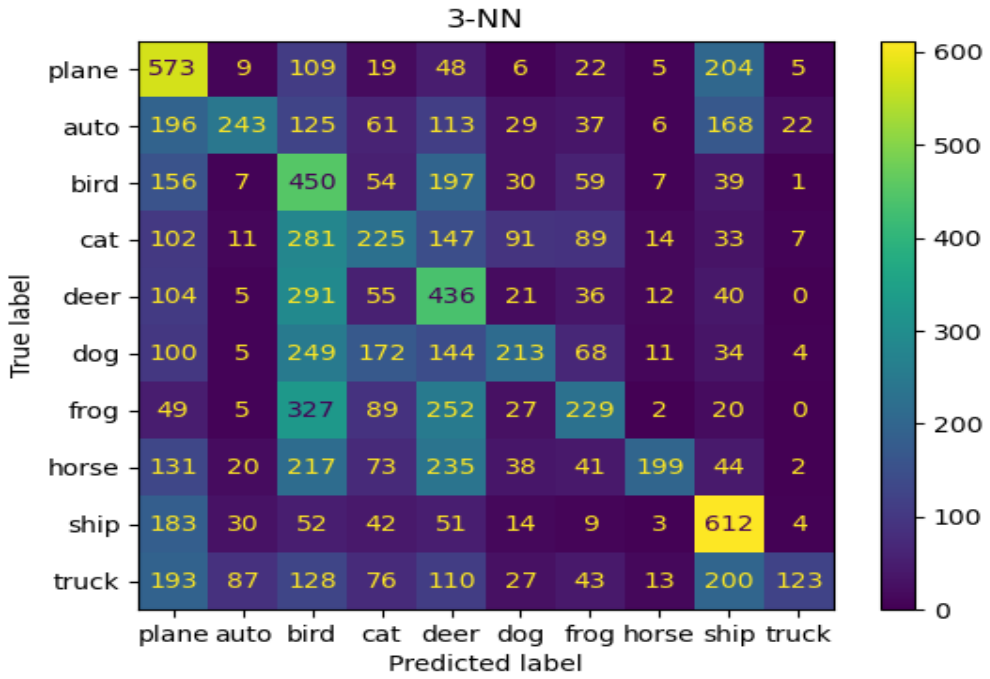
Όσον αφορά την κατηγοριοποίηση ανά κλάση τα αποτελέσματα φαίνονται και στις εκτυπώσεις του προγράμματος που χάριν συντομίας παραλείπονται (έχουν την ίδια πληροφορία με τους πίνακες), αλλά και στους αντίστοιχους πίνακες σύγχυσης που φαίνονται παρακάτω. Συγκεκριμένα το nearest centroid μοντέλο φαίνεται να έχει υψηλό bias ως προς τις κλάσεις plane και frog όπου ναι μεν πετυχαίνει πάνω από 50% ακρίβεια αλλά με πολύ χαμηλό precision. Το καλύτερο precision το παρουσιάζει στην κλάση ship αλλά με οριακά πάνω από 1 στα 3 σε ακρίβεια. Τα μοντέλα πλησιέστερων γειτόνων μοιάζουν αρκετά μεταξύ τους στον πίνακα σύγχυσης κάτι που ίσως να περιμέναμε. Το μοντέλο του ενός γείτονα έχει υψηλή μεροληψία ως προς τις κλάσεις bird, deer και ship, αν και στη ship πετυχαίνει ακρίβεια >60% αλλά με χαμηλό precision. Έχει σχετικά υψηλό precision στην κλάση truck αλλά μάταια αφού σπανίως κατηγοριοποιεί κάτι ως τέτοιο. Το μοντέλο των τριών γειτόνων έχει υψηλή μεροληψία ως προς τις κλάσεις bird, deer, ship, και plane (ειδικότερα στις 2 πρώτες >38%) όπου κατηγοριοποιεί πάνω από το 70% του συνόλου δοκιμής. Αυτό έχει ως αποτέλεσμα να κατηγοριοποιεί με σχετικά υψηλό precision 2 από τις υπόλοιπες (horse και truck). Η τάση αυτή των 2 μοντέλων γείτονα φαίνεται να γενικεύεται στο συγκεκριμένο dataset καθώς παρόμοια αποτελέσματα προέκυψαν από εκπαίδευση και επαλήθευση σε υποσύνολο του train set, καθώς και σε μοντέλα με κ από 5 έως 13. (βλ. prints.txt)



Εικόνα 1 Πίνακας σύγχυσης για πλησιέστερο κέντρο.



Εικόνα 2 Πίνακας σύγχυσης για πλησιέστερο γείτονα.



Εικόνα 3 Πίνακας σύγχυσης για 3 πλησιέστερους γείτονες.

Συμπεράσματα:

Στο πρόβλημα κατηγοριοποίησης της CIFAR-10 το καλύτερο από τα 3 μοντέλα βάσει ακρίβειας και f1_score είναι το μοντέλο του πλησιέστερου γείτονα, που απαιτεί ωστόσο αρκετά περισσότερο χρόνο εκπαίδευσης από αυτό του πλησιέστερου κέντρου και δεν είναι σημαντικά καλύτερο. Συμπερασματικά σε ένα πρόβλημα με αρκετά παραπάνω δεδομένα ίσως να αρκούμασταν στο πλησιέστερο κέντρο. Παρόλα αυτά κανένα από τα μοντέλα δεν επιτυγχάνει σημαντική ακρίβεια κατηγοριοποίησης στη συγκεκριμένη βάση δεδομένων και αν θέλαμε να επιτύχουμε κάτι τέτοιο θα πρέπει να καταφύγουμε σε άλλους τύπους μοντέλων.

Βιβλιογραφία:

[1] CIFAR-10 https://www.cs.toronto.edu/~kriz/cifar.html

[2] scikit-learn https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[3] Pandas https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf

[4] Matplotlib https://doi.org/10.1109/MCSE.2007.55

[5] NumPy https://doi.org/10.1038/s41586-020-2649-2

Παράρτημα:

Στο .zip αρχείο υπάρχει αυτή η αναφορά, το αρχείο κώδικα grigoriou_9564.py καθώς κι ένα αρχείο prints.txt στο οποίο υπάρχουν οι εκτυπώσεις του κώδικα για έξτρα πειράματα (όπως άλλα k και εκπαίδευση-επαλήθευση μόνο στο train set).