Εργασία Υπολογιστική Νοημοσύνη
Μέρος Α’

Γρηγόρης Καπαδούκας (ΑΜ: 1072484)

3 Απριλίου 2023

0 Περιβάλλον Εργασίας - Σύνδεσμος GitHub με Κώδικα

Για την διεκπεραίωση αυτής της εργασίας έχω επιλέξει να χρησιμοποιήσω γλώσσα προγραμματισμού Python μαζί τις βιβλιοθήκες TensorFlow (κυρίως το API της, το Keras) για τον σχεδιασμό και την εκπαίδευση των νευρωνικών δικτύων. Επίσης χρησιμοποιώ Pandas και Scikit-Learn με σκοπό τον χειρισμό του CSV αρχείου και της προεπεξεργασίας.

Ο κώδικας που γράφτηκε για την εργασία βρίσκεται στο repository στον παρακάτω σύνδεσμο:

https://github.com/GregKapadoukas/University-Computational-Intelligence-Project-A

1 Προεπεξεργασία και Προετοιμασία Δεδομένων

1.a Κωδικοποίηση και προεπεξεργασία δεδομένων

1.a.1 Διάβασμα του CSV αρχείου και μετατροπή κατηγορικών δεδομένων σε αριθμητικά

Αρχικά, με σκοπό το διάβασμα του dataset στη μορφή του CSV αρχείου χρησιμοποιώ εντολές της βιβλιοθήκης Pandas για φορτώσω τα δεδομένα

στη μορφή ενός DataFrame. Έπειτα χωρίζω το αρχικό DataFrame σε δύο, από τα οποία το πρώτο περιέχει τα δεδομένα των αισθητήρων και τα στοιχεία του ατόμου πάνω στο οποίο έγιναν οι μετρήσεις και το δεύτερο περιέχει την κλάση δραστηριότητας στην οποία ανήκει το άτομο.

Έπειτα μετατρέπω τα κατηγορικά δεδομένα των κλάσεων του δεύτερου DataFrame σε αριθμητικά δεδομένα, τα οποία όμως είναι one-hot encoded διανύσματα μεγέθους $\mathbb{R}^{1\times5}$. Άρα οι τιμές μετατρέπονται σε διανύσματα με την εξής αντιστοίχηση:

- 'sitting': [1 0 0 0 0]
- 'sitting-down': [0 1 0 0 0]
- 'standing': [0 0 1 0 0]
- 'standing-up': [0 0 0 1 0]
- 'walking': [0 0 0 0 1]

Επέλεξα την παραπάνω one-hot encoded προσέγγιση αντί για την αριθμητική 1-5 όπως προτείνεται στην εκφώνηση, επειδή σκοπεύω να έχω 5 εξόδους στο νευρωνικό δίκτυο, όπως θα εξηγήσω στο κεφάλαιο 2.

Έπειτα μετατρέπω και τα κατηγορικά δεδομένα του δεύτερου DataFrame, δηλαδή τα series 'Name' και 'Gender' σε αριθμητικά δεδομένα, με βάση την εξής αντιστοίχηση.

Για τα ονόματα:

- 'debora': 1
- 'katia': 2
- 'wallace': 3
- 'jose_carlos': 4

Για το φύλλο:

- 'Man': 1
- 'Woman': 2

Έτσι πλέον έχω μετατρέψει όλα τα κατηγορικά δεδομένα σε αριθμητικά, το οποίο είναι αναγκαίο για να μπορέσει να γίνει η μετέπειτα προεπεξεργασία των δεδομένων και η χρήση τους για την εκπαίδευση του νευρωνικού δικτύου.

1.a.2 Εξάλειψη Ενδεχόμενης Πόλωση

Από τις μεθόδους που αναφέρονται στην εκφώνηση έχουμε δύο επιλογές διαδικασιών με σκοπό την εξάλειψη ενδεχόμενης πόλωσης. Η πρώτη επιλογή είναι να κάνουμε αρχικά κεντράρισμα και έπειτα κανονικοποίηση των δεδομένων. Με αυτή τη διαδικασία καταλήγουμε με δεδομένα στο εύρος που επιλέγουμε για την κανονικοποίηση και ταυτόχρονα τα δεδομένα σε κάθε Series έχουν μέση τιμή το μέσο του εύρους που επιλέξαμε.

Η δεύτερη επιλογή είναι να κάνουμε τυποποίηση στα δεδομένα που έχει αποτέλεσμα τη μετατροπή των δεδομένων για κάθε Series σε γκαουσιανή κατανομή με μέση τιμή 0 και διακύμανση 1. Σε αυτή τη περίπτωση δεν υπάρχει ανάγκη για κεντράρισμα ή τυποποίηση εφόσον τα δεδομένα κεντράρονται μέσω της διαδικασίας (μέση τιμή 0), και η κανονικοποίηση θα επηρέαζε τη διακύμανση, το οποίο δεν είναι επιθυμητό χαρακτηριστικό σε αυτή τη περίπτωση.

Σχετικά με τις δύο προσεγγίσεις σημειώνω ότι η τυποποίηση φέρει καλύτερα αποτελέσματα σε περιπτώσεις όπου τα δεδομένα ακολουθούν ήδη γκαουσιανή κατανομή, με διαφορετικές βέβαια μέσες τιμές και διακυμάνσεις ή σε περιπτώσεις που έχουμε outliers. Στην περίπτωση που η κατανομή δεν είναι γκαουσιανή και δεν έχουμε outliers, υπάρχει πιθανότητα το κεντράρισμα με μετέπειτα κανονικοποίηση να φέρει καλύτερο αποτέλεσμα.

Σε αυτή τη περίπτωση θέλουμε να επεξεργαστούμε τα δεδομένα του DataFrame που περιέχει τις μετρήσεις των αισθητήρων και τα στοιχεία των ατόμων, και όχι το DataFrame με τις κλάσεις δραστηριοτήτων, εφόσον αυτό έχει ήδη μετατραπεί στη μορφή που χρειάζεται μέσω του one-hot encoding. Άρα παρατηρούμε (διαισθητικά και μέσω της συνάρτησης plot.density() του Pandas) ότι τα δεδομένα σε όλες τις κλάσεις ακολουθούν κανονική κατανομή εκτός από τις κλάσεις ’User’, ’Gender’ και ’Age’.

Οπότε εφόσον έχουμε συνδυασμό κανονικών και διαφορετικών κατανομών στα δεδομένα, αποφασίζω να δοκιμάσω και τις δύο διαδικασίες στο νευρωνικό δίκτυο του κεφαλαίου 2 και παρατηρώ ότι η μετρική της ακρίβειας στη περίπτωση της τυποποίησης μόνο είναι ελάχιστα καλύτερη (0.9813 vs 0.96) μετά από 20 εποχές σε σύγκριση με τον συνδυασμό κεντραρίσματος και κανονικοποίησης (η κανονικοποίηση έγινε σε εύρος [-1,1], επειδή χρησιμοποιήθηκε ReLU συνάρτηση ενεργοποίησης, οπότε το εύρος [0,1] ήταν λιγότερο αποδοτικό).

Άρα συμπεραίνω πως επειδή τα δεδομένα ακολουθούν επί το πλείστον κανονικές κατανομές (15/18 Series), η συνολική ακρίβεια του μοντέλου με

τυποποίηση είναι υψηλότερη από ότι με κεντράρισμα και κανονικοποίηση. Μάλιστα τα δεδομένα των μη κανονικών κατανομών (3/18 Series) επιδέχονται επίσης κανονικοποίηση ως αποτέλεσμα της τυποποίησης, οπότε δεν θεώρησα αναγκαίο να χρησιμοποιήσω διαφορετική τεχνική προεπεξεργασίας μόνο σε αυτά.

1.b Διασταυρούμενη Επικύρωση (cross-validation)

Αρχικά πριν τον διαχωρισμό των δεδομένων για το 5-fold CV, θα ενώσω τα δύο DataFrame που διαχώρισα πριν για να κάνω την προεπεξεργασία, με χρήση της εντολής concat του Pandas. Άρα πλέον έχουμε πάλι ένα DataFrame με όλα τα δεδομένα, αυτή τη φορά όμως προεπεξεργασμένα.

Με σκοπό τον διαχωρισμό των δεδομένων σε σύνολο εκπαίδευσης και σύνολο ελέγχου και τη μετέπειτα χρήση 5-fold CV, θα χρησιμοποιήσω ένα object τύπου KFold της βιβλιοθήκης SKLearn. Έτσι θα αρχικοποιήσω το object χρησιμοποιώντας παραμέτρους 'nsplits = 5' και 'shuffle = true' με 'random_state = 2'. Με την παράμετρο nsplits ορίζω ότι θέλω να χωρίσω τα δεδομένα μου σε 5 μέρη, με την παράμετρο shuffle ορίζω ότι θέλω το κάθε fold να είναι ισορροπημένο ως προς τον αριθμό των δειγμάτων και με το random_state ορίζω ένα seed που χρησιμοποιείται για την τυχαιότητα για την τελική σειρά των χωρισμένων συνόλων (ορίζει έμμεσα την τυχαιότητα ως προς ποιο σύνολο επιλέγεται ως σύνολο εκπαίδευσης).

Έπειτα για να διαχωρίσω τα δεδομένα με βάση το KFold που δημιουργήθηκε χρησιμοποιώ τη συνάρτηση next, και προκύπτουν δύο σύνολα, ένα που περιέχει 4 από τα 5 folds και ένα που περιέχει το τελευταίο, τα οποία αποθηκεύονται σε μορφή πίνακα στη μεταβλητή results. Τέλος αποθηκεύω αντίστοιχα τα σύνολα σε μεταβλητές για το σύνολο εκπαίδευσης και ελέγχου, ξεχωρίζοντας πάλι τις τελικές κλάσεις από τα υπόλοιπα δεδομένα (άρα καταλήγω με τις μεταβλητές train_measurements, train_classes, test_measurements και test_classes).

2 Επιλογή Αρχιτεκτονικής