

Νευρωνικά Δίκτυα

3^η Εργασία

Σε αυτή την εργασία υλοποιείται ένα νευρωνικό δίκτυο RBF, που επιλύει το πρόβλημα κατηγοριοποίησης της βάσης δεδομένων MNIST, σε μονά και ζυγά ψηφία. Όπως και στην προηγούμενη εργασία, αφού φορτώσει το σετ δεδομένων, οι κλάσεις γίνονται δύο και οι ετικέτες 0 και 1, για τα ζυγά και τα μονά ψηφία αντίστοιχα. Κρατώνται τα 3000 πρώτα δείγματα για την εκπαίδευση και τα 600 πρώτα για το τεστ.

Περνώντας στα RBF, είναι δίκτυα που ξεχωρίζουν λόγω της ταχύτητας μάθησης που αποφέρουν. Είναι ένας τύπος feed forward νευρωνικού δικτύου που αποτελείται από 3 στρώματα, το στρώμα εισόδου, το κρυμμένο στρώμα και αυτό της εξόδου. Στο πρώτο στρώμα συνδέσεις χωρίς βάρος περνάν τις εισόδους στο κρυμμένο στρώμα. Οι κρυμμένοι κόμβοι αποτελούν ακτινικές μονάδες. Η συνάρτηση που χρησιμοποιείται ως συνάρτηση μεταφοράς είναι η γκαουσιανή και εξαρτάται από την ευκλείδεια απόσταση των δειγμάτων με τα RBF κέντρα μονάδων που βρίσκουμε με κάποιον αλγόριθμο όπως η k μέσοι, ή ακόμα και τυχαία, και την παράμετρο σίγμα. Στο δεύτερο επίπεδο παίρνουμε την RBF της εισόδου x και δίνονται βάρη με τη χρήση ελαχίστων τετραγώνων. Οι κόμβοι εξόδου επιλέγονται με βάση την μεγαλύτερη υπολογισμένη πιθανότητα. Όσον αφορά την το σίγμα, για την rbf, υπάρχουν δύο περιπτώσεις. Είτε διαλέγεται να είναι ίσο με 1 προς την τυπική απόκλιση κάθε cluster, είτε με βάση μία φόρμουλα που αντιστοιχεί στην ρίζα του 2 επί το k , προς τη μέγιστη απόσταση 2 οποιωνδήποτε κέντρων.

Ο λόγος που τα δίκτυα αυτά επιλέγουν κεντρικά διανύσματα είναι επειδή θα ήταν αδύνατο να φτιαχτεί ένας rbf κόμβος για κάθε δείγμα στο σετ εκπαίδευσης, όπως κάνουν για παράδειγμα οι kernels.

Έτσι, επιλέγουμε 2 μεθόδους για την επιλογή των κέντρων, η πρώτη είναι απλά τυχαία επιλογή k σημείων από το σετ δεδομένων, και η δεύτερη είναι με τον αλγόριθμο των k μέσων.

Ο αλγόριθμος των k μέσων ανήκει στη μάθηση χωρίς επίβλεψη, δηλαδή δεν χρειάζεται ετικέτες. Ωστόσο είναι αναγκαία η δήλωση του αριθμού των clusters, δηλαδή του k .

Αρχικά επιλέγει τυχαία k κέντρα για κάθε cluster, βρίσκει την απόσταση όλων των σημείων από αυτά τα κέντρα και αναθέτει τα σημεία στο πλησιέστερο τους κέντρο. Έπειτα σε επαναληπτικό ρυθμό για κάποιον αριθμό επαναλήψεων, βρίσκει τα νέα κέντρα του κάθε cluster χρησιμοποιώντας το μέσο όλων των σημείων ενός cluster, μέχρι όλα τα σημεία να συγκλίνουν (ή τα κέντρα να μην αλλάζουν).

Δοκιμάζοντας να τρέξουμε τον αλγόριθμο, πρέπει να ρυθμίσουμε τις διάφορες παραμέτρους, δηλαδή, την μέθοδο επιλογής κέντρου, τον αριθμό k των clusters, τις επαναλήψεις του k - μέσων, το αν θα έχουμε τυπική απόκλιση από τα clusters (μεταβλητή `std_from_clusters`).

Να σημειωθεί ότι δεν έχει γίνει κανονικοποίηση των δεδομένων και δεν έχει χρησιμοποιηθεί PCA. Δοκιμάζοντας αρχικά την μέθοδο επιλογής κέντρων με k means, με $k=10$, για 100

επαναλήψεις, και διαλέγοντας `std_from_clusters` αληθές, το αποτέλεσμα που παίρνουμε είναι 78% στο τεστ σε 16 δευτερόλεπτα, και στην εκπαίδευση 81% σε 8 δευτερόλεπτα.

Αλλάζοντας μόνο τον αριθμό των k σε 20, στην εκπαίδευση έχουμε 85% ενώ στο τεστ 82% σε 42 δευτερόλεπτα. Βλέποντας πως η ακρίβεια μεγαλώνει αυξάνοντας το k , το βάζω ίσο με 50 ώστε να δω το αποτέλεσμα, το οποίο όντως βγάζει υψηλότερη ακρίβεια στο 89% για την εκπαίδευση και στο 86.1% για το τεστ στα 51 δευτερόλεπτα.

Οπότε τώρα δοκιμάζω ένα αρκετά μικρό $k=2$, ώστε να δω πως θα αλλάξει η κατάσταση. Όντως πέφτει η ακρίβεια στο 66% για την εκπαίδευση και το τεστ, σε πολύ λιγότερα δευτερόλεπτα, μόνο 2.

Κρατώντας το $k=10$ δοκιμάζω να αλλάξω τις επαναλήψεις του `kmeans` πρώτα σε πολλές δηλαδή 1000, και έπειτα σε αρκετά λιγότερες ίσες με 20.

Για 1000 επαναλήψεις λοιπόν, η ακρίβεια στην εκπαίδευση πιάνει το 82.3% σε 10 δεύτερα και το 78.6% στο τεστ σε 8 δεύτερα. Για 20 επαναλήψεις, οι επιδόσεις είναι 83% και 80% αντίστοιχα, σε 6 δεύτερα. Φαίνονται να είναι καλύτερες δηλαδή από ότι στις 1000 επαναλήψεις.

Κρατώντας τις επαναλήψεις σε 20 και το k σε 10, αλλάζουμε απο αληθή σε ψευδή τη μεταβλητή `std_from_clusters`. Η ακρίβεια γίνεται 83.6% σε εκπαίδευση και 80.6% σε τεστ στα 6 πάλι δευτερόλεπτα. Φαίνεται πως η αλλαγή δεν επέφερε μεγάλη επίδραση στην ακρίβεια.

Αφήνοντας τη μεταβλητή ψευδή, δοκιμάζουμε $k=50$ και παίρνουμε 90.4% στην εκπαίδευση και 87% στο τεστ.

Προχωρώντας απο τους k means, σε τυχαία επιλογή των κέντρων, ξεκινάμε με $k=10$ και μεταβλητή `std_from_clusters` αληθή. Η ακρίβεια είναι 77.3% σε εκπαίδευση και 57% σε τεστ σε μόλις 1 και 2 δευτερόλεπτα αντίστοιχα. Βλέπουμε πως στο τεστ έχει πέσει αρκετά η επίδοση.

Ανεβάζοντας τα k σε 50, η ακρίβεια σε εκπαίδευση είναι 83.1% και σε τεστ 82.5%. Το χάσμα ανάμεσα τους έχει μειωθεί σημαντικά σε σχέση με πριν.

Βάζοντας τη μεταβλητή `std_from_clusters` ψευδή και αφήνοντας το k ίσο με 50, η ακρίβεια είναι 84.8% και 83.1% σε εκπαίδευση και τεστ αντίστοιχα, αρκετά γρηγορότερα από τις ίδιες παραμέτρους για τον k means. Ωστόσο οι επιδόσεις κυμαίνονται σε παρόμοια ποσοστά.

Δοκιμάζοντας και μία χαμηλή μεταβλητή για το $k=2$, όπως ήταν αναμενόμενο σε γρήγορο πάλι χρόνο του 0.1 δευτερολέπτου, η ακρίβεια είναι 69.8% και 52% αντίστοιχα.

Φαίνεται λοιπόν πως ένα μεγάλο k είναι χρονοβόρο, ωστόσο εγγυάται υψηλές επιδόσεις σε σχέση με ένα χαμηλότερο, και στους k means αλλά και στην τυχαία επιλογή.

Οι παραπάνω δοκιμές αντιστοιχούν σε δεδομένα που δεν έχουν υποστεί κανονικοποίηση. Παρατήρησα πως αν εφαρμόσω τον αλγόριθμο με τις ίδιες παραμέτρους αλλά σε κανονικοποιημένα δεδομένα, οι επιδόσεις πέφτουν πάρα πολύ.

Έτσι δοκίμασα να αφαιρέσω τη μεταβλητή `std_from_clusters` και να βάλω χειροκίνητα μία τιμή στη παράμετρο σίγμα του `rbf`. Έκανα λοιπόν κάποιες δοκιμές με βάση αυτό και τα αποτελέσματα φαίνονται παρακάτω, είτε με το κανονικό σετ δεδομένων είτε μετά από PCA.

Χωρίς PCA: Για kmeans με $k=20$ στην εκπαίδευση: 79.3% στο τεστ: 75.1%

με $k=2$ στην εκπαίδευση: 62.1% στο τεστ: 61.6%

Για τυχαία επιλογή με $k=20$ στην εκπαίδευση: 78.9% στο τεστ: 65%

Για τυχαία επιλογή με $k=2$ στην εκπαίδευση: 59% στο τεστ: 48%

Με PCA: Για kmeans με $k=20$ στην εκπαίδευση: 77.6% στο τεστ: 76.1%

με $k=2$ στην εκπαίδευση: 61.8% στο τεστ: 62%

Για τυχαία επιλογή με $k=20$ στην εκπαίδευση: 76.9% στο τεστ: 78.16%

Για τυχαία επιλογή με $k=2$ στην εκπαίδευση: 50.8% στο τεστ: 48%

Φαίνεται πως χωρίς την PCA τα αποτελέσματα είναι ελάχιστα υψηλότερα, ενώ γενικά ο αλγόριθμος k μέσων έχει καλύτερη επίδοση από την επιλογή τυχαίων κέντρων.

Όσον αφορά τους αλγορίθμους των k κοντινότερων γειτόνων και των πλησιέστερων κέντρων, τα αποτελέσματα περιγράφονται παρακάτω.

Χωρίς PCA: KNN Για $k=1$ ακρίβεια 93.83% σε 19.4 δευτερόλεπτα. Για $k=3$ 93.66% σε 19.5 δεύτερα. Nearest Centroid 80.3% σε 0.02 δεύτερα.

Με PCA: KNN Για $k=3$ ακρίβεια 94% σε 17.2 δευτερόλεπτα. Για $k=1$ 93.5% σε 17.1 δεύτερα. Nearest Centroid 80.3% σε 0.03 δεύτερα.

Ο αλγόριθμος των k κοντινότερων γειτόνων παρουσιάζει υψηλή ακρίβεια, περισσότερη από το rbf δίκτυο, ωστόσο εκείνος των πλησιέστερων κέντρων είναι κοντά στις περισσότερες περιπτώσεις.

Η εργασία υλοποιήθηκε στο σύνδεσμο:
<https://colab.research.google.com/drive/1KsnQpgSrADU9OEGltVdkbrrOy0EiZmGV#scrollTo=2etRMbHxpBb>

Η δεύτερη εκδοχή του RBF:
https://colab.research.google.com/drive/17nJpoMxvHyR3CGgmVzzqVGO9_GKz-VJ1#scrollTo=HVnO5ATOsC5F

Κώστογλου Σοφία