Εργασία στο μάθημα Μηχανικής Μάθησης (2024-2025)

Στόχος της εργασίας είναι η υλοποίηση και εφαρμογή μεθόδων που παρουσιάστηκαν στις διαλέξεις και στα εργαστήρια σε ένα classification task της επιλογής σας. Το dataset που θα επιλέξετε να χρησιμοποιήσετε θα πρέπει να περιλαμβάνει τουλάχιστον 3 ξεχωριστές κλάσεις (multiclass classification). Το ίδιο θα πρέπει να ισχύει και για το πλήθος των features. Από τα datasets που μπορείτε να χρησιμοποιήστε αποκλείονται το Iris dataset και τα MNIST datasets (π.χ. handwritten digits, fashion MNIST κλπ.) μερικά από τα οποία έχουμε χρησιμοποιήσει και στα εργαστήρια.

Πιο συγκεκριμένα καλείστε να υλοποιήστε τις εξής μεθόδους :

1. Τον αλγόριθμο Principal Component Analysis (PCA), για την μείωση των διαστάσεων των δεδομένων σας. Αν n είναι το πλήθος των features των δεδομένων σας θα πρέπει να εκτελέσετε τον αλγόριθμο για ένα m ε [2, n) της επιλογής σας. Αρκεί η επίδειξη των αποτελεσμάτων στα δεδομένα, ενώ η εφαρμογή του στις υπόλοιπες μεθόδους δεν είναι απαραίτητη.
2. Τον αλγόριθμο ελαχίστων τετραγώνων (least squares). Στα πλαίσια των φροντιστηρίων αναπτύξαμε τον αλγόριθμο σε ένα regression task. Καλείστε να προσαρμόσετε τον αλγόριθμο στο classification task που επιλέξατε.
3. Τον αλγόριθμο λογιστικής παλινδρόμησης (logistic regression). Θα πρέπει να εκπαιδεύσετε το μοντέλο σας με τη χρήση του αλγορίθμου Stochastic Gradient Descent. Ως loss function θα πρέπει να χρησιμοποιήσετε το Cross Entropy Loss.
4. Τον αλγόριθμο K κοντινότερων γειτόνων (K Nearest Neighbors). Θα πρέπει να βρείτε τη βέλτιστη τιμή της υπερπαραμέτρου K στο διάστημα [1, 10]. Παρουσιάστε τα αποτελέσματα σας για κάθε τιμή της K.
5. Τον αλγόριθμο Naïve Bayes. **ΠΡΟΣΟΧΗ!** Η μετατροπή των features σε binary δεδομένα (τιμές 0 ή 1), όπως παρουσιάστηκαν στα φροντιστήρια δεν θα γίνει αποδεκτή. Αντίθετα, μπορείτε να εφαρμόσετε κανονικές κατανομές με διαγώνιους πίνακες συμμεταβλητότητας (όχι κατ' ανάγκη κοινούς για κάθε κλάση), που είναι ο Naïve Bayes για δεδομένα στο R^d (δεν χρειάζεται να αποδείξετε κάτι).
6. Έναν Multilayer Perceptron (νευρωνικό δίκτυο με πολλαπλά γραμμικά επίπεδα σε συνδυασμό με μη γραμμικές συναρτήσεις ενεργοποίησης) μέσω του Pytorch framework. Είστε ελεύθεροι να επιλέξετε το πλήθος των επιπέδων, τον τύπο των συναρτήσεων ενεργοποίησης, το learning rate καθώς επίσης και οποιοδήποτε άλλη υπερπαράμετρο. Ισχύουν οι ίδιοι περιορισμοί με τη λογιστική παλινδρόμηση.
7. Τον αλγόριθμος SVM (Support Vector Machines). Στα εργαστήρια το πρόβλημα που αντιμετωπίσαμε αφορούσε binary classification. Στη περίπτωση σας μπορείτε να αντιμετωπίσετε το πρόβλημα με πολλαπλά One-vs-Rest SVMs (η θετική κλάση ως θετική και οι υπόλοιπες ως αρνητικές). Εκπαιδεύστε ένα SVM για κάθε κλάση (με κατάλληλες τροποποιήσεις στα δεδομένα σας) και θεωρήστε ως τελικά label το label του SVM με το υψηλότερο score.

8. Τον αλγόριθμο K-Means. Ο αλγόριθμος αφορά την περίπτωση συσταδοποίησης (clustering), οπότε, μόνο γι' αυτή τη περίπτωση, θεωρείστε ότι η κλάση του κάθε παραδείγματος των δεδομένων σας είναι άγνωστος. Εκτελέστε τον αλγόριθμο για έναν αριθμό συστάδων ίσο με τον πλήθος των διαφορετικών κλάσεων του dataset που επιλέξατε.

Επιπλέον παρατηρήσεις/tips.

- Ο κώδικας σας θα πρέπει να είναι γραμμένος σε python notebook. Μπορείτε να χρησιμοποιήσετε το Google Colab, το Jupyter Notebook, ή οποιαδήποτε άλλη πλατφόρμα που τα υποστηρίζει. Στο notebook προσθέστε cells με μια σύντομη περιγραφή της εκάστοτε υλοποίησης που αναπτύξατε, ιδανικά πριν από την κάθε υλοποίηση. Το παραδοτέο θα είναι ένα ή περισσότερα αρχεία τύπου .ipynb.
- Στην αρχή του notebook συμπεριλάβετε μια περιγραφή του task που επιλέξατε. Περιγράψτε ποιο είναι το πρόβλημα που θα λύσετε, πόσα δεδομένα έχετε, πόσα features υπάρχουν, ποια είναι τα labels κλπ. Μια γραφική αναπαράσταση των δεδομένων σας (π.χ. μέσω της βιβλιοθήκης matplotlib) θα ήταν ιδιαίτερα βοηθητική τόσο για εσάς όσο και για τους διορθωτές.
- Το dataset που θα επιλέξετε, πέραν των περιορισμών που αναφέρθηκαν ήδη, μπορεί να περιλαμβάνει κατηγορικές ή συνεχείς μεταβλητές καθώς επίσης και μία μίξη αυτών. Προσέξτε τον τύπο των δεδομένων σας, καθώς μπορεί να χρειαστεί να κάνετε μετατροπές τόσο στα δεδομένα σας όσο στις υλοποιήσεις σας. Για παράδειγμα ο υπολογισμός των παραμέτρων του Naïve Bayes είναι διαφορετικός στην περίπτωση κατηγορικών μεταβλητών σε σχέση με την περίπτωση συνεχών μεταβλητών. Αντίστοιχα, datasets με κατηγορικές μεταβλητές μπορούν να χρησιμοποιηθούν και στα νευρωνικά δίκτυα με κατάλληλη επεξεργασία των δεδομένων (κάθε feature θα πρέπει να μετατραπεί σε one-hot vector). Αν σας διευκολύνει μπορείτε να χρησιμοποιήσετε διαφορετικά datasets ανά υλοποίηση.
- Χωρίστε τα δεδομένα σας σε training και test set, αν δεν έχουν διαχωριστεί ήδη. Χρησιμοποιήστε το training set σας για την εκπαίδευση και το testing set για την αξιολόγηση των μεθόδων σας. Προς διευκόλυνσή σας μπορείτε να αγνοήσετε το development/validation set.
- Για κάθε μέθοδο παρουσιάστε το accuracy των μεθόδων σας τόσο στο train όσο και στο test set. Ειδικά για την περίπτωση της λογιστικής παλινδρόμησης και του νευρωνικού δικτύου, παρουσιάστε επιπλέον, σε ένα plot το Cross entropy loss των μεθόδων σε κάθε βήμα (epoch) του αλγορίθμου (για το train και το test set).
- **Δεν επιτρέπεται** η χρήση έτοιμων υλοποιήσεων των ζητούμενων αλγορίθμων (π.χ. μέσω του scikit-learn library). **Όμως**, αφού αναπτύξετε τις μεθόδους, (**προαιρετικά**) μπορείτε να τις συγκρίνετε με έτοιμες υλοποιήσεις και να παρουσιάσετε τα ευρήματα σας (το sklearn προσφέρει μέχρι και έτοιμη υλοποίηση MLP). Μπορείτε

να χρησιμοποιήσετε έτοιμες μεθόδους που διευκολύνουν στην παρουσίαση ή την προεπεξεργασία των δεδομένων σας (π.χ. για τα γραφήματα ή τον διαχωρισμό των δεδομένων σας σε train/test sets).

- Σε καμία περίπτωση δε θα ληφθεί υπόψη η τελική απόδοση των αλγορίθμων σας. Στόχος δεν είναι η επίλυση του προβλήματος αλλά η εμπέδωση, υλοποίηση και εφαρμογή των μεθόδων που αναφέρθηκαν. Μπορείτε γι' αυτό το λόγο να περιοριστείτε σε μικρότερο αριθμό δεδομένων από αυτόν που προσφέρει το dataset σας (εντός λογικών πλαισίων), για παράδειγμα στην περίπτωση που η εκπαίδευση των μοντέλων σας είναι ασύμφορη από άποψη χρόνου ή αποθηκευτικού χώρου.