

Εργασία Μηχανικής Μάθησης 2022

Tsirmpas Dimitris

16th January 2023

1 Λογιστική Παλινδρόμηση

1.1 Ερώτημα Δ

Ο ταξινομητής μας έχει υλοποιηθεί στο αρχείο `LogisticRegClassifier.py`. Η πλήρης τεκμηρίωση του μοντέλου, των μεθόδων του και των υπερπαραμέτρων βρίσκεται εκεί με τη μορφή docstrings. Η κανονικοποίηση L_2 έχει ήδη υλοποιηθεί σε αυτό το αρχείο, αλλά για τους σκοπούς αυτής της ερώτησης θα θέσουμε την υπερπαραμέτρο λ ως 0, παρακάμπτοντας την.

Ο κώδικας για την εκτέλεση του μοντέλου βρίσκεται στο αρχείο `run_logistic.py`. Θα τρέξουμε το μοντέλο με υπερπαραμέτρους `iter = 500` και `alpha = 0.2`. Το αποτέλεσμα είναι η ακρίβεια εκπαίδευσης να είναι ίση με 0.982 και η ακρίβεια ελέγχου ίση με 0.981. Τα πλήρη αποτελέσματα της εκπαίδευσης και του ελέγχου παρουσιάζονται στην Εικόνα 1.

1.2 Ερώτημα Ε

Επιλέγουμε το διάστημα των λ τιμών μας λογαριθμικά, εφόσον η βέλτιστη τιμή κανονικοποίησης είναι πολύ πιο πιθανό να βρίσκεται αρκετά κοντά στο 0. Η λογαριθμική κλίμακα μας επιτρέπει να ψάξουμε πιο πολλές τιμές του λ όσο πιο κοντά φτάνουμε στο κάτω όριο αναζήτησής μας, το 10^4 . Η προσέγγιση αυτή χρησιμοποιείται και στην πράξη για κανονικοποίηση L^2 [1]

Σημειώνουμε ότι λόγω υπολογιστικών απαιτήσεων, ο αριθμός επαναλήψεων των μοντέλων μας μειώνεται στις 250 επαναλήψεις. Κατά την εκτέλεση του προγράμματος υπάρχει επίσης πιθανότητα να εμφανιστούν ειδοποιήσεις για αριθμητική υπερχείλιση. Αυτό είναι αποτέλεσμα της επιλογής πολύ μεγάλης τιμής του λ , κυρίως στο διάστημα $[8, 10]$. Σε αυτό το σημείο η κανονικοποίηση είναι τόσο ισχυρή που αποτρέπει το μοντέλο μας από το να μάθει, και έτσι αυτό μαντεύει την ίδια κατηγορία για κάθε παράδειγμα ελέγχου.

Στην δική μας περίπτωση το μοντέλο, κρατώντας τις υπόλοιπες υπερπαραμέτρους ίσες με το προηγούμενο υποερώτημα, προτιμά την ελάχιστη τιμή κανονικοποίησης $\lambda = 10^{-4}$ με ακρίβεια ελέγχου ίση με 0.981. Παρατηρούμε ότι η ακρίβεια ελέγχου με την επιλεγμένη τιμή λ είναι μικρότερη από την αντίστοιχη στο υποερώτημα Δ. Αυτό μας υποδεικνύει είτε ότι η βέλτιστη τιμή βρίσκεται έξω (και πιο συγκεκριμένα πριν)

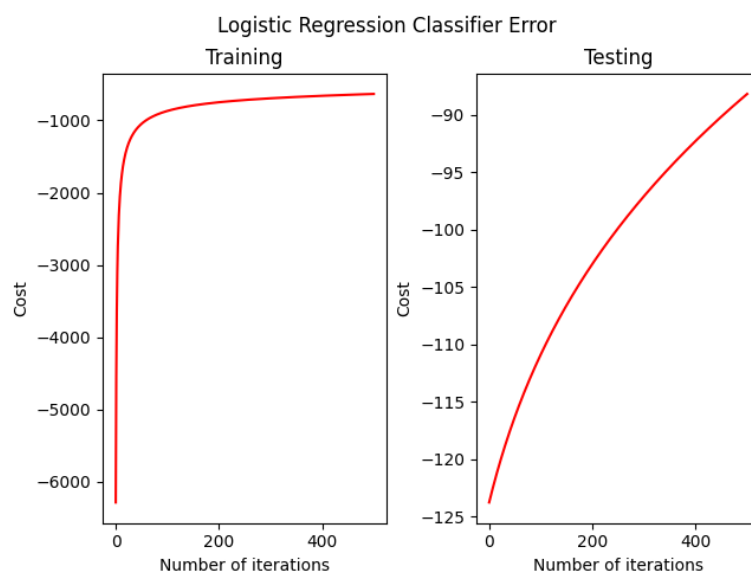


Figure 1: Τα αποτελέσματα της εκπαίδευσης και του ελέγχου στον ταξινομητή μας. Αριστερά: Το κόστος εκπαίδευσης ως συνάρτηση των επαναλήψεων του αλγορίθμου gradient ascent. Δεξιά: Το αντίστοιχο κόστος ελέγχου. Υπενθυμίζουμε ότι οι κλίμακες των γραφημάτων δεν είναι ίσες, καθώς ο ήδη εκπαιδευμένος ταξινομητής αρχίζει με πολύ μικρότερο κόστος.

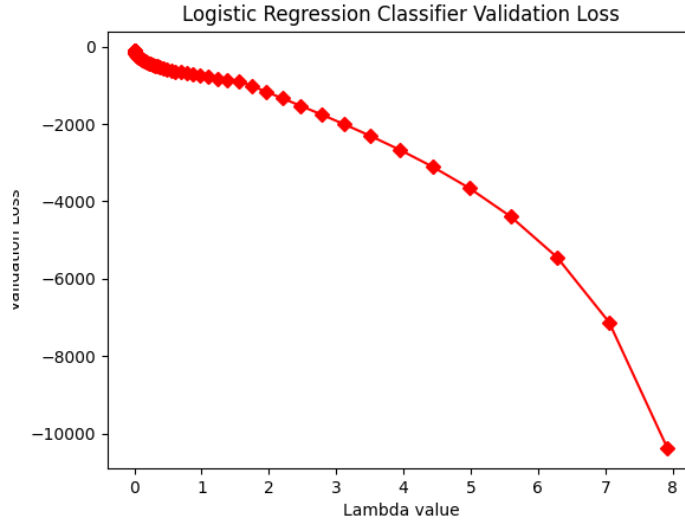


Figure 2: Τα αποτελέσματα της αναζήτησης για το βέλτιστο λ . Οι ρόμβοι αντιπροσωπεύουν τις τιμές που εξετάσαμε. Παρατηρήστε το πλήθος των τιμών που εξετάστηκαν στην αρχή συγκριτικά με το τέλος του πεδίου αναζήτησής μας.

από το διάστημα αναζήτησης μας, είτε ότι για την διαφορά ευθύνεται το στατιστικό σφάλμα.

Ο κώδικας παραγωγής των παραπάνω γραφημάτων και αποτελεσμάτων βρίσκεται στο αρχείο `run_logistic.py`. Τα πλήρη αποτελέσματα αναζήτησης της υπερπαραμέτρου παρουσιάζονται στην Εικόνα 2.

2 ΜΕΡΟΣ Γ

2.1 Ερώτημα ΣΤ

Το νευρωνικό μας δίκτυο έχει υλοποιηθεί στο αρχείο `mlp.py`. Αποτελείται από δύο πίνακες βαρών και δύο πίνακες bias:

- Ο πίνακας `h_w` (hidden weights) έχει μέγεθος $I \times H$
- Ο πίνακας `o_w` (output weights) έχει μέγεθος $H \times O$
- Ο πίνακας `h_b` (hidden bias) έχει μέγεθος $1 \times H$
- Ο πίνακας `o_b` (output bias) έχει μέγεθος $1 \times O$

όπου $I = \text{input_size}$, $H = \text{hidden_layer_size}$, $O = \text{output_size}$. Ο κώδικας για την εκτέλεση του δικτύου βρίσκεται στο αρχείο `run_mlp.py`, το οποίο εκτελεί κώδικα και για τα ερωτήματα H , Θ .

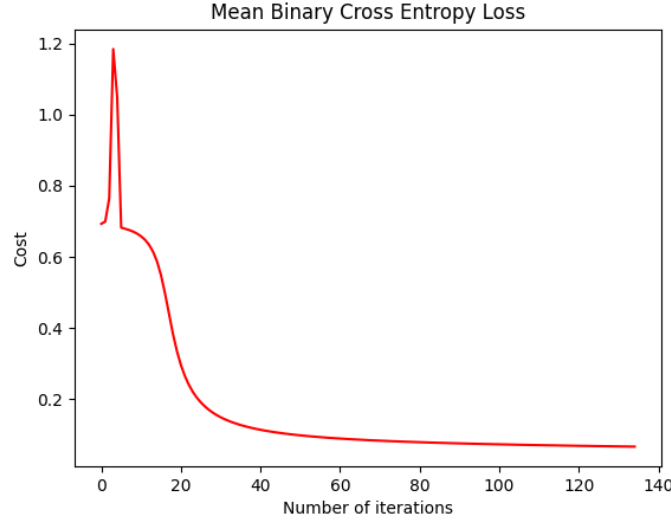


Figure 3: Το κόστος εκπαίδευσης ως συνάρτηση των επαναλήψεων του αλγορίθμου gradient descent.

Θα τρέξουμε το μοντέλο με υπερπαραμέτρους $m=2$, $\eta=0.2$ και $\text{tolerance}=0.001$. Το tolerance είναι μια υπερ-παραμέτρος απαραίτητη για το early stopping, και η οποία καθορίζει πόσο το κόστος πρέπει να έχει μειωθεί για να θεωρείται η τρέχουσα εποχή "βελτίωση". Το αποτέλεσμα είναι η ακρίβεια εκπαίδευσης να είναι ίση με 0.978, το τελικό κόστος εκπαίδευσης ίσο με 0.0671, η ακρίβεια ελέγχου ίση με 0.974 και το μέσο κόστος ελέγχου ίσο με 0.0734. Τα πλήρη αποτελέσματα της εκπαίδευσης παρουσιάζονται στην Εικόνα 3.

2.2 Ερώτημα Z

Ο τύπος της Διαδικιακής Διασταυρούμενης Εντροπίας είναι:

$$E_b = -(t \ln \hat{y} + (1 - t) \ln(1 - \hat{y})) \quad (1)$$

όπου t το διάνυσμα των ετικετών δεδομένων και \hat{y} το της εκτιμώμενης πιθανότητας.

Στην παραπάνω εξίσωση παρατηρούμε ότι η τιμή t είναι πάντοτε γνωστή, αντίθετα με τη \hat{y} , επομένως θα παραγωγίσουμε με βάση το \hat{y} . Εφόσον $y = f(x) + g(x) \iff y' = f'(x) + g'(x)$ ισχύει ότι:

$$\frac{\partial E_b}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}}(t \ln \hat{y}) + \frac{\partial}{\partial \hat{y}}((1 - t) \ln(1 - \hat{y})) \quad (2)$$

Αναλύουμε τις επιμέρους συναρτήσεις του αθροίσματος της παραγώγου:

$$\frac{\partial}{\partial \hat{y}}(t \ln \hat{y}) = \frac{\partial}{\partial \hat{y}}(t \ln \hat{y}) + \ln \hat{y} \frac{\partial}{\partial \hat{y}} t = \frac{t}{\hat{y}} + \ln \hat{y} \cdot 0 = \frac{t}{\hat{y}} \quad (3)$$

$$\frac{\partial}{\partial \hat{y}}((t-1)\ln(1-\hat{y})) = (1-t)\frac{\partial}{\partial \hat{y}}(\ln(1-\hat{y})) + \ln(1-\hat{y})\frac{\partial}{\partial \hat{y}}(1-t) = \frac{1-t}{1-\hat{y}} + \ln(1-\hat{y}) \cdot 0 = \frac{1-t}{1-\hat{y}} \quad (4)$$

Επομένως, αθροίζοντας τις 3, 4, η 2 γίνεται:

$$\frac{\partial E_b}{\partial \hat{y}} = -\left(\frac{t}{\hat{y}} - \frac{1-t}{1-\hat{y}}\right) = \left(\frac{t}{\hat{y}} + \frac{1-t}{1-\hat{y}}\right) \quad (5)$$

Ο κώδικας επαλήθευσης του αναλυτικού τύπου βρίσκεται στο αρχείο `test_mlp.py`.

2.3 Ερώτημα Η

Εκτελούμε `grid search` για τις παραμέτρους m, η . Επιλέγουμε το διάστημα των η τιμών μας στο χώρο αναζήτησης $[0.5, 10^{-5}]$, εκθετικά κοντά στο 0.5 εφόσον εμπειρικά αναμένουμε ότι η βέλτιστη τιμή της θα βρίσκεται κοντά στις τιμές 0.5, 0.01, 0.01. το οποίο για το συγκεκριμένο δίκτυο είναι το ($\eta = 0.5, m = 0.5, E = 172$) με κόστος επικύρωσης ίσο με 0.0423.

Το πρόγραμμα εκτέλεσης του `grid search` βρίσκεται στο αρχείο `run_mlp.py`. Η αναζήτηση χρειάζεται περίπου 10 λεπτά για να βγάλει το βέλτιστο συνδυασμό υπερπαραμέτρων, το οποίο οφείλεται σχεδόν αποκλειστικά στην εκπαίδευση και επαλήθευση μοντέλων με $M \in 256, 512, 1024$. Λόγω των περιορισμών στις υπερπαραμέτρους δεν μπορούμε να μειώσουμε σημαντικά τον χρόνο εκτέλεσης.

2.4 Ερώτημα Θ

Χρησιμοποιούμε την μέθοδο `predict` του μοντέλου μας για να αποκτήσουμε τις προβλεπόμενες ετικέτες του. Η μέθοδος αυτή χρησιμοποιεί μεθόδους της `numpy`, ισοδύναμες της κατηγοριοποίησης με βρόγχο. Στο σύνολο ελέγχου το μοντέλο με τις βελτιστοποιημένες παραμέτρους επιτυγχάνει ακρίβεια ελέγχου ίση με 0.979 και κόστος ελέγχου ίσο με 0.0569.

References

- [1] Jerome Friedman, Trevor Hastie and Robert Tibshirani. ‘Regularization Paths for Generalized Linear Models via Coordinate Descent’. In: *Journal of Statistical Software* (2010).