Εργασία Μηχανικής Μάθησης 2022-2023

Τσίρμπας Δημήτρης p3190205

25th January 2023

Όλα τα ερωτήματα της εργασίας έχουν υλοποιηθεί.

1 Δομή Εργασίας

Η εργασία χωρίζεται σε 10 αρχεία πηγαίου κώδικα python. Τα πιο σημαντικά είναι:

- `load_mnist.py` εκτελεί την φόρτωση και προ-επεξεργασία των δεδομένων
- `logistic_regression.py` περιέχει το μοντέλο λογιστικής παλινδρόμησης
- `run_logistic.py` περιέχει τις εντολές για την δημιουργία των γραφημάτων και αποτελεσμάτων του μέρους Β
- `mlp.py` περιέχει το μοντέλο απλού νευρωνικού δικτύου
- `run_mlp.py` περιέχει τις εντολές για την δημιουργία των γραφημάτων και αποτελεσμάτων του μέρους Γ (εκτός από το ερώτημα Ι)
- `gradcheck.py` το οποίο ελέγχει την μέθοδο back-propagation του ερωτήματος Ζ
- `sgd.py` περιέχει το μοντέλο του στοχαστικού νευρωνικού δικτύου
- `run_sgd.py` περιέχει τις εντολές για την δημιουργία των γραφημάτων και αποτελεσμάτων του ερωτήματος Ι

Περιέχονται επίσης τα αρχεία `common.py`, που περιέχει κοινές συναρτήσεις για τα παραπάνω αρχεία, και `test_load_mnist.py` το οποίο συμβάλλει στην επικύρωση των δεδομένων μας.

Η πλήρη τεκμηρίωση του κώδικα και της υλοποίησης μπορεί να βρεθεί στα παραπάνω αρχεία με την μορφή python docstrings και σχολίων.

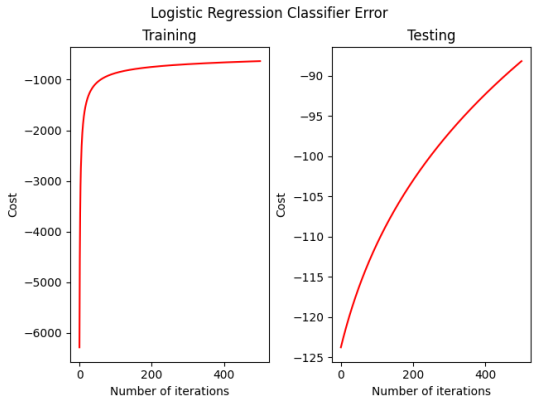

Figure 1: Τα αποτελέσματα της εκπαίδευσης και του ελέγχου στον ταξινομητή μας. Αριστερά: Το κόστος εκπαίδευσης ως συνάρτηση των επαναλήψεων του αλγορίθμου gradient ascent. Δεξία: Το αντίστοιχο κόστος ελέγχου. Υπενθυμίζουμε ότι οι κλίμακες των γραφημάτων δεν είναι ίσες, καθώς ο ήδη εκπαιδευμένος ταξινομητής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

2 Μέρος Β - Λογιστική Παλινδρόμηση

2.1 Ερώτημα Δ

Ο ταξινομητής μας έχει υλοποιηθεί στο αρχείο `logistic_regression.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 υπερπαράμετρο $\lambda = 0$, 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.

2.2 Ερώτημα Ε

Ο κώδικας παραγωγής των παραπάνω γραφημάτων και αποτελεσμάτων βρίσκεται στο αρχείο `run_logistic.py`.

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 αναζήτησης 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 προγράμ-

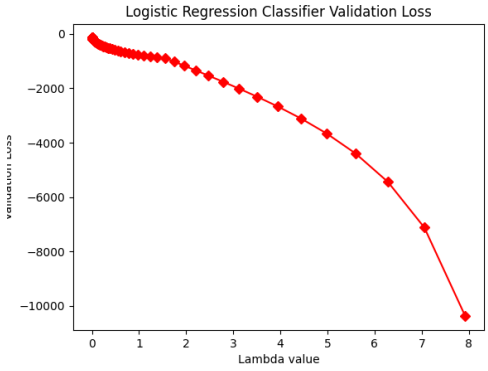

Figure 2: Τα αποτελέσματα της αναζήτησης για το βέλτιστο λ. Οι ρόμβοι αντιπροσωπεύουν τις τιμές που εξετάσαμε. Παρατηρείστε το πλήθος των τιμών που εξετάστηκαν στην αρχή συγκριτικά με το τέλος του πεδίου αναζήτησής μας.

ματος υπάρχει επίσης πιθανότητα να εμφανιστούν warnings για αριθμητική υπερχείλιση, τα οποία έχουμε απενεργοποιήσει. 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) από το διάστημα αναζήτησης μας, είτε ότι για την διαφορά ευθύνεται το στατιστικό σφάλμα.

Τα πλήρη αποτελέσματα αναζήτησης της υπερπαραμέτρου παρουσιάζονται στην Εικόνα 2.

3 ΜΕΡΟΣ Γ - Νευρωνικό Δίκτυο

3.1 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 IxH
- Ο πίνακας `o_w` (output weights) έχει μέγεθος HxO
- Ο πίνακας `h_b` (hidden bias) έχει μέγεθος $1xH$
- Ο πίνακας `o_b` (output bias) έχει μέγεθος $1xO$

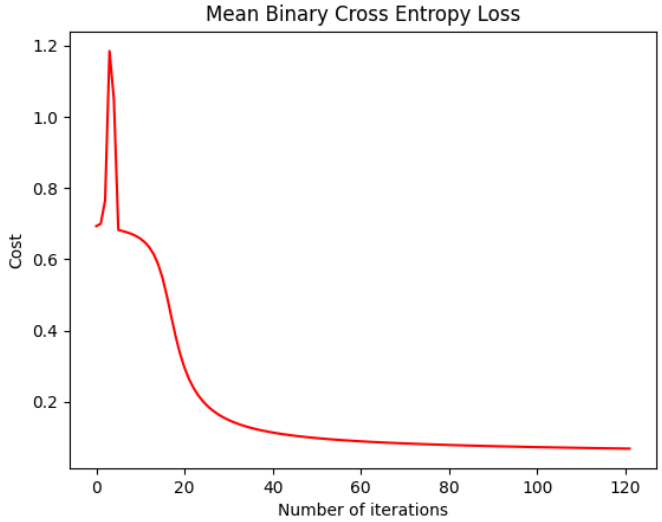

Figure 3: Το κόστος εκπαίδευσης ως συνάρτηση των επαναλήψεων του αλγορίθμου gradient descent.

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

Θα τρέξουμε το μοντέλο με υπερπαραμέτρους m=2, η=`0.2` και `tolerance`=`0.001`. Το `tolerance` είναι μια υπερ-παράμετρος 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 0.977, το μέσο κόστος επικύρωσης ίσο με 0.0634, η ακρίβεια ελέγχου ίση με 0.975 και το μέσο κόστος ελέγχου ίσο με 0.0753. Τα πλήρη αποτελέσματα της εκπαίδευσης παρουσιάζονται στην Εικόνα 3.

3.2 Ερώτημα Ζ

Ο τύπος της Δυαδικής Διασταυρούμενης Εντροπίας είναι:

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

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

Στην παραπάνω εξίσωση για να βρούμε την παραγωγό ως προς υην τελική έξοδο του δικτύου θεωρούμε ότι η τιμή t είναι πάντοτε γνωστή, και παραγωγίζουμε με βάση το $\hat{y}$. Εφόσον $y = f(x) + g(x) \iff y' = f'(x) + g'(x)$ ισχύει ότι:

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

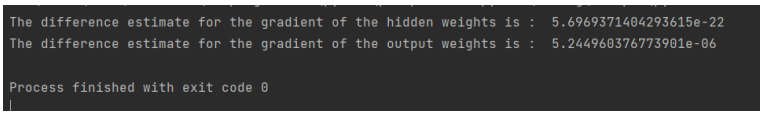

Figure 4: Το αποτέλεσμα εκτέλεσης του κώδικα ελέγχου της παραγώγου στο νευρωνικό μοντέλο μας

Αναλύουμε τις επιμέρους συναρτήσεις του αθροίσματος της παραγώγου:

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

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

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

$$\frac{\partial E_b}{\partial \hat{y}} = -(\frac{t}{\hat{y}} - \frac{1-t}{1-\hat{y}}) = \frac{1-t}{1-\hat{y}} - \frac{t}{\hat{y}} \tag{5}$$

Η εικόνα 4 δείχνει το αποτέλεσμα της εκτέλεσης του κώδικα ελέγχου της παραγώγου. Ο κώδικας επαλήθευσης του αναλυτικού τύπου βρίσκεται στο αρχείο `gradcheck.py`. Επιπλέον, εφόσον η μέθοδος που χρησιμοποιείται για τον υπολογισμό του gradient είναι *ακριβώς* η ίδια και για το στοχαστικό μοντέλο μας (στον κώδικα η μέθοδος `back_propagation`), η παραπάνω απόδειξη και αποτέλεσμα εκτέλεσης παραμένουν ίδια.

3.3 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 αποτέλεσμα για το συγκεκριμένο δίκτυο είναι το $(\eta = 0.5, m = 4, E = 139)$ με μέσο κόστος ελέγχου ίσο με 0.0669 και ακρίβεια ελέγχου 0.981.

Το πρόγραμμα εκτέλεσης του grid search βρίσκεται στο αρχείο `run_mlp.py`. Η αναζήτηση χρειάζεται περίπου 20 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 της εκφώνησης στις υπερπαραμέτρους δεν μπορούμε να μειώσουμε σημαντικά τον χρόνο εκτέλεσης.

3.4 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 κατηγοριοποιήσης 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.

3.5 Ερώτημα Ι

Ο κώδικας του ταξινομητή SGD μπορεί να βρεθεί στο αρχείο `sgd.py`.

Υλοποιούμε τον ταξινομητή στοχαστικής καθοδικής κλίσης ως υποκλάση του προηγούμενού μας ταξινομητή. Η κύρια αλλαγή είναι το πέρασμα mini-batches στην μέθοδο `back_propagation` αντί για το σύνολο των δειγμάτων εκπαίδευσης. Πιο συγκεκριμένα, αντί να περάσουμε στην `back_propagation` έναν πίνακα διαστάσεων 9072x784, περνάμε πίνακα Bx784 όπου B το μέγεθος του mini-batch. Τα παραδείγματα εκπαίδευσης και οι ετικέτες τους ανακατεύονται πριν κάθε επανάληψη του αλγορίθμου έτσι ώστε το μοντέλο μας να εξετάζει κάθε φορά διαφορετικό σύνολο παραδειγμάτων. Η ίδια η μέθοδος παραμένει ίδια.

Το πρόγραμμα εκτέλεσης του ταξινομητή καθώς και του παρακάτω grid search βρίσκεται στο αρχείο `run_sgd.py`. Τρέχοντας τον ταξινομητή μας με παρόμοιες παραμέτρους με τον προηγούμενο και με αυθαίρετο μέγεθος mini-batch=128, επιτυγχάνει ακρίβεια εκπαίδευσης 0.976 με κόστος 0.0678 και ακρίβεια ελέγχου 0.977 με κόστος 0.0793. Η ακρίβεια αυτή είναι κατά μικρό βαθμό χειρότερη από τον ταξινομητή λογιστικής παλινδρόμησης, αν και λόγω του υπολογισμού με mini-batches και του κριτηρίου early-stopping, το μοντέλο είναι σημαντικά γρηγορότερο.

Ένα πλήρες γράφημα για την εξέλιξη του κόστους εκπαίδευσης βρίσκεται στην Εικόνα 5. Παρατηρούμε ότι το κόστος του ταξινομητή μας ακολουθεί την ίδια πορεία με αυτό της Εικόνας 3, αν και πιο ανώμαλη. Αυτό οφείλεται στην τυχαιότητα του αλγόριθμου SGD, ο οποίος προσθέτει "θόρυβο" στη διαδικασία εκπαίδευσης μέσω του τυχαίου και περιορισμένου πλήθους δεδομένων που βλέπει το μοντέλο μας σε κάθε επανάληψη του SGD.

Figure 5: Το κόστος εκπαίδευσης ως συνάρτηση των επαναλήψεων του αλγορίθμου stochastic gradient descent.

Δοκιμάζοντας το μοντέλο μας με τιμές $B = 2^i, i = 1, 2, .., 8$, η βέλτιστη τιμή

μεγέθους mini-batch φαίνεται να είναι η $B = 256$, με κόστος επικύρωσης ίσο με 0.6877 και με $E = 16$ εποχές εκπαίδευσης. Τρέχοντας το μοντέλο μας στο σύνολο ελέγχου έχουμε κόστος ελέγχου ίσο με 0.0864 και ακρίβεια ελέγχου ίση με 0.970.

Χρησιμοποιώντας τη βέλτιστη τιμή B, μπορούμε να τρέξουμε την ίδια μέθοδο αναζήτησης βέλτιστων υπερπαραμέτρων όπως στο ερώτημα Η. Με την ίδια μεθοδολογία και το ίδιο χώρο αναζήτησης υπερ-παραμέτρων, το πρόγραμμα μας βρίσκει βέλτιστες υπερ-παραμέτρους ($\eta = 0.139, m = 4, E = 103$) με μέσο κόστος επικύρωσης ίσο με 0.0735. Τρέχοντας το μοντέλο μας στο σύνολο ελέγχου έχουμε κόστος ελέγχου ίσο με 0.0758 και ακρίβεια ελέγχου ίση με 0.979.

Σημειώνουμε ότι λόγω της τυχαιότητας του αλγορίθμου SGD τα αποτελέσματα της εκτέλεσης μπορεί να είναι διαφορετικά από αυτά που αναγράφονται στο έγγραφο αυτό. Υπάρχει επίσης πιθανότητα εμφάνισης gradient vanishing για μικρές τιμές του B.

References

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