ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Αναφορά 2ης Εργασίας

Authors
Θωμάς Λάγκαλης, 2021030079
Ιωάννης Λεμονάκης, 2021030138

ΤΗΛ311 - Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πολυτεχνείο Κρήτης

Ιούλιος 2023

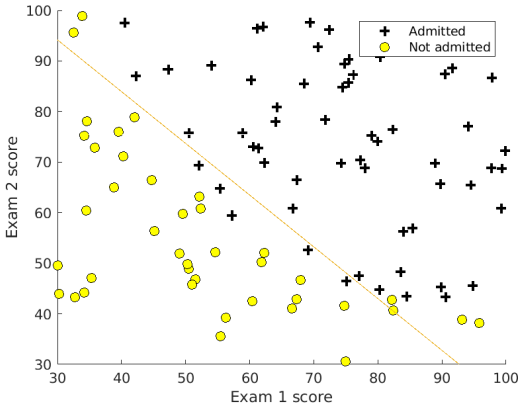
Περιεχόμενα

1 Εισαγωγή

Το δεύτερο σετ ασκήσεων καταπιάνεται με διάφορες τεχνικές και αλγορίθμους μηχανικής μάθησης. Στην παρούσα αναφορά παρουσιάζονται τα απαραίτητα σχόλια για τα θέματα του δεύτερου σετ ασκήσεων, όπου αυτό ζητείται, και το μαθηματικό αποτέλεσμα ή η συνοπτική περιγραφή/λύση, όπου αυτό δεν ζητείται. Πιο συγκεκριμένα, το πρώτο θέμα αφορούσε την λογιστική παλινδρόμηση και την αναλυτική εύρεση κλίσης. Το δεύτερο θέμα αφορούσε την λογιστική παλινδρόμηση με ομαλοποίηση. Το τρίτο θέμα σχετιζόταν με τους εκτιμητές MAP και ML. Το τέταρτο θέμα αφορούσε την ομαδοποίηση δεδομένων κατά K-Means. Το πέμπτο θέμα σχετιζόταν με την υλοποίηση ενός απλού νευρωνικού δικτύου. Τέλος, στο έκτο θέμα σκοπός ήταν η υλοποίηση ενός συνελικτικού νευρωνικού δικτύου για αναγνώριση διάφορων εικόνων.

2 Θέμα 1ο - Λογιστική Παλινδρόμηση : Αναλυτική εύρεση κλίσης (Gradient)

Στο παρόν θέμα έγιναν οι απαραίτητοι υπολογισμοί για το υποερώτημα (α) οι οποία περιλαμβάνονται στο αρχείο computations.pdf. Στο ερώτημα (β) υλοποιήθηκε ο κώδικας Matlab που βρίσκεται στο αρχείο My_logisticRegression.m. Η ακρίβεια της εκπαίδευσης είναι 89% λίγο χαμηλότερη από το αναμενόμενο. Συγκεκριμένα, σε μοντέλα παλινδρόμησης μία ικανοποιητική ακρίβεια πρόβλεψης είναι σίγουρα μεγαλύτερη από 90% (εξαρτάται φυσικά και από το πρόβλημα).



Σχήμα 1

Μερικές ανακρίβειες φαίνονται και στο σχήμα 1, όπου μερικά δείγματα (+) - Admitted - βρίσκονται εντός του σύνορου απόφασης (o) - Not admitted - και το αντίστροφο.

Η μειωμένη ακρίβεια του μοντέλου λογιστικής παλινδρόμησης μπορεί να συμβαίνει για δύο λόγους:

1. Λίγα χαρακτηριστικά: Αν το σύνολο δεδομένων έχει λίγα χαρακτηριστικά, το γραμμικό μοντέλο μπορεί να μην είναι αρκετά πλούσιο για να αναπαραστήσει την πολυπλοκότητα των δεδομένων.

2. Λίγα δείγματα: Με μόνο 100 δείγματα, η ποσότητα των δεδομένων είναι περιορισμένη και η ακρίβεια του γραμμικού μοντέλου μπορεί να επηρεαστεί από την ανεπαρκή πληροφορία που δίνουν τα δείγματα.

Γενικά, η μειωμένη ακρίβεια πρόβλεψης σε ένα γραμμικό μοντέλο μπορεί να οφείλεται σε αδυναμίες στη μοντελοποίηση των σχέσεων μεταξύ των μεταβλητών εισόδου και εξόδου, τέτοιες αδυναμίες μπορεί να πηγάζουν από πολλές επιπλέον αιτίες όπως η παρουσία θορύβου ή η απουσία συσχέτισης μεταξύ των χαρακτηριστικών. Τα αίτια, όμως, εν προκειμένω είναι η απουσία επαρκών χαρακτηριστικών και δεδομένων.

3 Θέμα 2ο - Λογιστική Παλινδρόμηση με Ομαλοποίηση

Στη δοθείσα άσκηση το ζητούμενο είναι η υλοποίηση ενός μοντέλου λογιστικής παλινδρόμησης **με ομαλοποίηση** για την πρόβλεψη επιτυχίας ορισμένων μικροτσίπ σε έναν έλεγχο ποιότητας (QA). Η ομαλοποιημένη συνάρτηση κόστους δίνεται από την σχέση:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}ln(h_\theta(x^{(i)})) - (1-y^{(i)})ln(1-h_\theta)) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^i$$

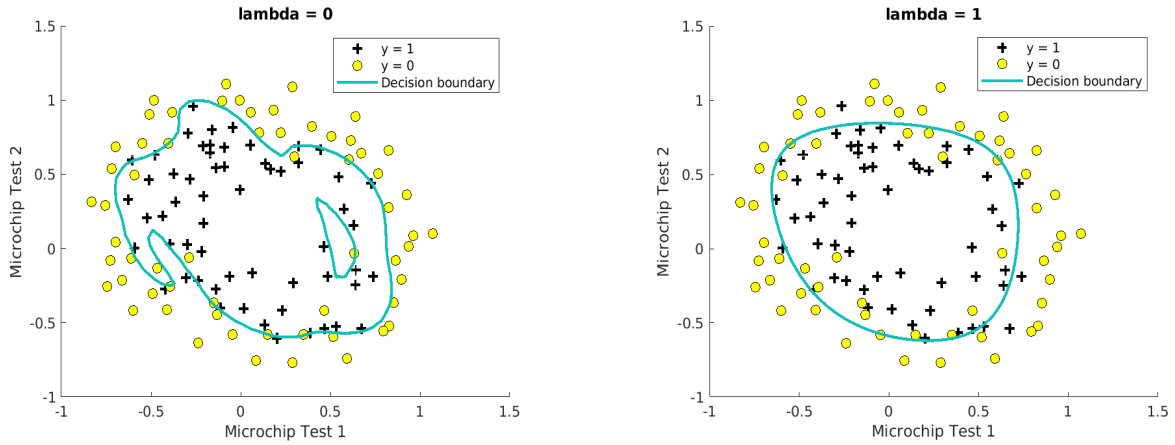
Με το λ να είναι ο παράγοντας ομαλοποίησης (regularization parameter). Στον πίνακα 1 και σχήμα 3 φαίνονται τα αποτελέσματα τις εκτέλεσης του προγράμματος Matlab (λογιστικής παλινδρόμησης με ομαλοποίηση) για διάφορες τιμές του λ.

Γενικά αναμένεται, όταν η τιμή του λ είναι πολύ μικρή ή μηδενική, ο όρος ομαλοποίησης έχει ελάχιστη επίδραση στη συνάρτηση κόστους. Σε αυτήν την περίπτωση, το μοντέλο εστιάζει περισσότερο στην ελαχιστοποίηση του σφάλματος εκπαίδευσης, και μπορεί να οδηγήσει σε υπερεκτίμηση (overfitting) των δεδομένων εκπαίδευσης. Αυτό σημαίνει ότι το μοντέλο θα προσαρμοστεί υπερβολικά στα δεδομένα εκπαίδευσης και δεν θα γενικεύει καλά σε νέα δεδομένα.

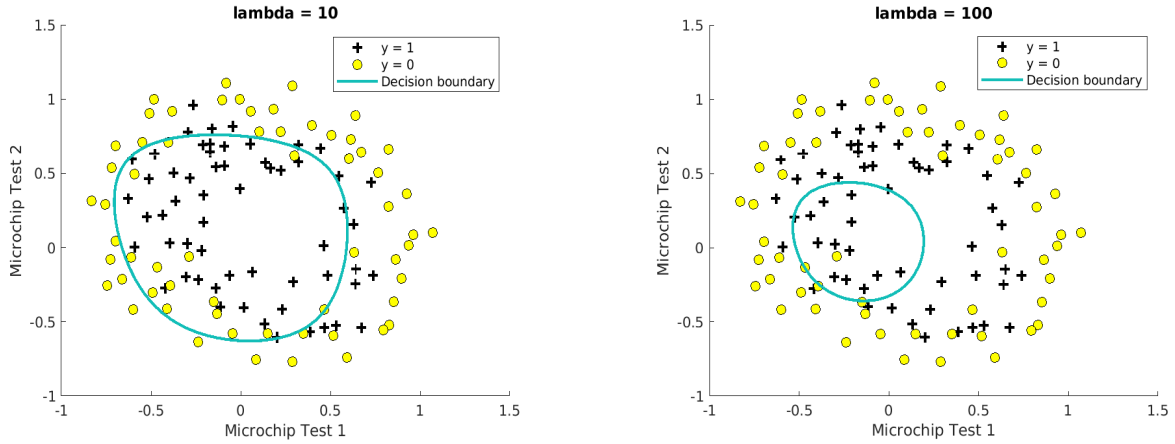
Αντίθετα, όταν η τιμή του λ είναι πολύ μεγάλη, ο όρος ομαλοποίησης έχει σημαντική επίδραση στη συνάρτηση κόστους. Σε αυτήν την περίπτωση, το μοντέλο προτιμά να έχει μικρότερες τιμές για τις παραμέτρους, περιορίζοντας την πολυπλοκότητα του μοντέλου. Αυτό μπορεί να βοηθήσει στην αποφυγή υπερεκτίμησης και να βελτιώσει τη γενίκευση σε νέα δεδομένα. Ωστόσο, μια πολύ μεγάλη τιμή του λ μπορεί να οδηγήσει σε υποεκπαίδευση (underfitting), όπου το μοντέλο δεν μπορεί να αναπαραστήσει επαρκώς την πολυπλοκότητα των δεδομένων.

lamda (λ)	0	1	10	100
Train accuracy	88.98	81.35	74.57	60.16

Πίνακας 1: Ακρίβεια στα δεδομένα εκπαίδευσης ανάλογα με το λ



Τα αποτελέσματα δεν είναι αντιπροσωπευτικά του ρόλου που επιτελεί το λ, διότι θα έπρεπε για λ=0 να παρατηρείται το overfitting ενώ για πολύ μεγάλες τιμές του λ (λ=100) να φαίνεται το underfitting. Τα δύο αυτά φαινόμενα δεν μπορούν να παρατηρηθούν άμεσα, γιατί τα αποτελέσματα είναι βασισμένα στα δεδομένα εκπαίδευσης, ενώ χρειάζονται και τα δεδομένα ελέγχου (validation data). Αυτό που μπορεί να γίνει αντιληπτό όμως, είναι πως για λ=0 υπάρχει μεγάλη προσαρμογή (ακρίβεια) στα



Σχήμα 3: Διαγράμματα των δεδομένων με τα σύνορα απόφασης για διαφορετικές τιμές των λ - δύο στην σελίδα 3 και δύο στην 4.

δεδομένα εκπαίδευσης (χαρακτηριστικό του overfitting) ενώ για μεγάλες τιμές του λ υπάρχει χαμηλή προσαμογή στο training set (χαρακτηριστικό του underfitting). Συνεπώς, δεν καθίσταται δυνατόν να παρατηρηθούν τα φαινόμενα overfitting και underfitting μόνο με τα δεδομένα εκπαίδευσης.

4 Θέμα 3ο - Εκτίμηση Παραμέτρων με Maximum Likelihood και MAP

Σημείωση: Οι μαθηματικές πράξεις για τον υπολογισμό του εκτιμητή MLE και MAP βρίσκοτναι στο αρχείο Computations.pdf στο .zip παραδοτέο. Εδώ αναγράφονται μόνο τα αποτελέσματα.

Σε αυτό το θέμα, σκοπός ήταν η εύρεση του εκτιμητή ML όσο και του MAP μαθηματικά. Υποτέθηκε ότι υπάρχουν n δείγματα $D = \{x_1, \ldots, x_n\}$ όπου παράγονται ανεξάρτητα από μια κατανομή Poisson παραμέτρου λ.

Στο πρώτο ερώτημα, έπειτα από τις μαθηματικές πράξεις, βρέθηκε ότι ο εκτιμητής μέγιστης πιθανοφάνειας (MLE) της παραμέτρου $\hat{\lambda}_{ML}$ είναι ο μέσος όρος των n δειγμάτων, $\boxed{\hat{\lambda}_{ML} = \frac{1}{N}\sum_{k=1}^{N} x_k}$.

Στο δεύτερο ερώτημα, έπειτα από τις μαθηματικές πράξεις, η τιμή της παραμέτρου $\hat{\lambda}_{MAP}$ (Maximum A-Posteriori) που μεγιστοποιεί την πιθανότητα $\mathcal{P}(\lambda|D)$ είναι ο κανονικοποιημένος μέσος όρος των n δειγμάτων, $\boxed{\hat{\lambda}_{MAP} = \frac{1}{N+1}\sum_{k=1}^{N} x_k}$.

5 Θέμα 4ο - Ομαδοποίηση (Clustering) με K-means

Σε αυτή την άσκηση υλοποιήθηκε ο αλγόριθμος K-Means και χρησιμοποιήθηκε για την συμπίεση μίας εικόνας. Ο αλγόριθμος υλοποιήθηκε σε Matlab σύμφωνα με την εκφώνηση της άσκησης. Ο κώδικας βρίσκεται στο παραδοτέο αρχείο (ex24_kmeans.m).

5.1 Πως επηρεάζει ο αριθμός των κλάσεων την συμπίεση της εικόνας;

Κάθε κλάση αντιστοιχεί σε ένα σύνολο χρωμάτων που χρησιμοποιούνται για να αναπαραστήσουν τμήματα της εικόνας. Όταν χρησιμοποιούμε περισσότερες κλάσεις, έχουμε μεγαλύτερο αριθμό χρωμάτων για την αναπαράσταση της εικόνας. Αυτό σημαίνει ότι μικρές λεπτομέρειες και αποχρώσεις στην αρχική εικόνα μπορεί να αναπαρασταθούν με μεγαλύτερη ακρίβεια στη συμπιεσμένη εικόνα. Ωστόσο, αν αυξηθεί υπερβολικά ο αριθμός των κλάσεων, μπορεί να παρατηρηθεί αυξημένο μέγεθος αρχείου για τη συμπιεσμένη εικόνα.

Αν χρησιμοποιηθεί λιγότερος αριθμός κλάσεων, το αποτέλεσμα της συμπίεσης ενδέχεται να μην είναι ικανοποιητικό. Η εικόνα μπορεί να φαίνεται πιο απλοποιημένη και να χάνει λεπτομέρειες. Αυτό οφείλεται στο γεγονός ότι ο αλγόριθμος δεν μπορεί να αναπαραστήσει απόλυτα τις πολύπλοκες λεπτομέρειες της εικόνας με λιγότερες κλάσεις.

Συνεπώς, ο αριθμός των κλάσεων που χρησιμοποιείται στον αλγόριθμο k-Means επηρεάζει την ποιότητα της συμπιεσμένης εικόνας. Πρέπει να επιλεγεί ένας κατάλληλος αριθμός κλάσεων που εξασφαλίζει ισορροπία μεταξύ της συμπίεσης και της ποιότητας της αναπαράστασης της εικόνας.

5.2 Αλγόριθμος K-Means

Ακολουθεί η συνοπτική περιγραφή του αλγορίθμου K-Means, σε ψευδογλώσσα (βάσει του αντίστοιχου σε matlab από την εκφώνηση).

Algorithm 1: K-Means algorithm

Data: $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}, x^{(i)} \in \mathbb{R}^2$
$K \leftarrow numOfClasses$
Result: $centroids \in \mathbb{R}^{K \times 2}$
$centroids \leftarrow kMeansInitCentroids(X, K)$;
for $iter = 1$ **to** $iterations$ **do**
  $idx \leftarrow findClosestCentroids(X, centroids)$;
  $centroids \leftarrow computeMeans(X, idx, centroids$;
end

Σημείωση: Οι συναρτήσεις $kMeansInitCentroids(X, K)$, $findClosestCentroids(X, centroids)$, $computeMeans(X, idx, K)$ υλοποιούνται στα αντίστοιχα .m αρχεία, εδώ απλά παρατίθενται σαν τα βήματα του αλγορίθμου (black boxes). Ο αλγόριθμος τερματίζει όταν υπάρχει σύγκλιση δηλαδή οι επαναλήψεις του αλγορίθμου από ένα σημείο και μετά παράγουν τα ίδια κέντρα και ομαδοποιούν τα δεδομένα στις ίδιες κλάσεις.

6 Θέμα 5ο - Υλοποίηση ενός απλού νευρωνικού δικτύου

6.1 Μέρος Α

Σημείωση: Για το μέρος Α τα υποερωτήματα α,β,γ έχουν αποδειχθεί στο αρχείο Computations.pdf. Ακολουθεί η λύση του υποερωτήματος δ.

Ο αλγόριθμος οπισθοδιάδοσης (Back Propagation) χρησιμοποιείται για την εκπαίδευση ενός νευρωνικού δικτύου πολλαπλών επιπέδων. Κατά τη διάρκεια της οπισθοδιάδοσης, το σφάλμα υπολογίζεται ανάστροφα από το τελευταίο επίπεδο προς τα πίσω, επιτρέποντας την ενημέρωση των παραμέτρων (βάρη και bias) κάθε επιπέδου. Για ένα μόνο επίπεδο του νευρωνικού δικτύου, ο αλγόριθμος οπισθοδιάδοσης εκτελεί τα παρακάτω βήματα:

1. Τυχαία αρχικοποίηση των παραμέτρων. Συνήθως τα biases αρχικοποιούνται στο 0 και τα βάρη (weight) επιλέγονται τυχαία από μία κανονικοποιημένη γκαουσιανή κατανομή.

2. Εκτέλεση fowrard pass των δεδομένων από την είσοδο μέχρι την έξοδο.

3. Υπολογισμός της παραγώγου του κόστους ως προς την παράμετρος προς ενημέρωση. Στην προκείμενη περίπτωση η συνάρτηση κόστους είναι η **cross entropy** και συνάρτηση ενεργοποίησης είναι η **sigmoid**.

   Από τον κανόνα της αλυσίδας προκύπτει:

$$\frac{\partial J}{\partial W_L} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial W_L} \tag{1}$$

   ή για την ενημέρωση του βάρους

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial b} \tag{2}$$

   Όπου W_L τα βάρη του layer L, J η συνάρτηση κόστους cross entropy, $\hat{y}$ η έξοδος του προηγούμενου νευρώνα μετά την συνάρτηση ενεργοποίησης $\hat{y} = \Phi(z)$ και $z = W_i^L x + b$ με x την είσοδο του νευρώνα. Οπότε, πρέπει να υπολογιστούν οι επιμέρους παράγωγοι.

$$\frac{\partial J}{\partial \hat{y}} = \frac{\partial}{\partial \hat{y}}(-yln(\hat{y}) - (1-y)ln(1-\hat{y})) = -\frac{y}{\hat{y}} + (1-y)\frac{1}{1-\hat{y}} = \frac{\hat{y}-y}{(1-\hat{y})\hat{y}}$$

$$\frac{\partial \hat{y}}{\partial z} = \frac{d\Phi}{dz} = \frac{d}{dz}(\frac{1}{1+e^{-z}}) = -\frac{e^{-z}}{(1+e^{-z})^2} = -\frac{e^{-z}+1-1}{1+e^{-z}}\frac{1}{1+e^{-z}} = (\Phi(z)-1)\Phi(z)$$

$$\frac{\partial z}{\partial W_L} = \frac{\partial}{\partial W_L}(W_L x + b) = x = \hat{y_{L-1}}$$

   (Η είσοδος του παρόντος επιπέδου είναι η έξοδος του προηγούμενου, για hidden layers)

$$\frac{\partial z}{\partial b} = \frac{\partial}{\partial b}(W_L x + b) = 1$$

   Συνεπώς, οι σχέσεις (1) και (2) γίνονται:

$$\boxed{\frac{\partial J}{\partial W_L} = \frac{\hat{y}-y}{(1-\hat{y})\hat{y}}(\Phi(z)-1)\Phi(z)\hat{y_{L-1}}}$$

   και

$$\boxed{\frac{\partial J}{\partial b} = \frac{\hat{y}-y}{(1-\hat{y})\hat{y}}(\Phi(z)-1)\Phi(z)}$$

4. Τέλος, ενημερώνονται οι παράμετροι με τον Gradient Decent αλγόριθμο (ή κάποιον άλλον όπως adam, adamax κτλ):

$$W_L^{new} = W_L - \rho \frac{\partial J}{\partial W_L}$$

και

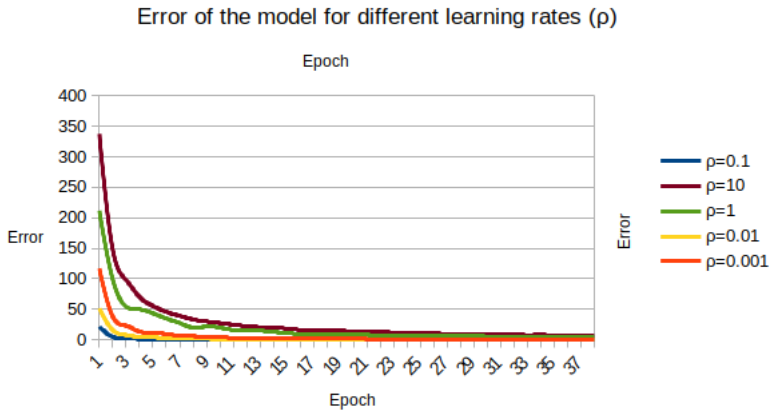
$$b^{new} = b - \rho \frac{\partial J}{\partial b}$$

όπου ρ είναι το learning rate (ρυθμός μάθησης).

6.2 Μέρος Β

Αρχικά, συμπληρώθηκε ο κώδικας (ex2_5.py) και έπειτα έγιναν μερικές πειραματικές δοκιμές με διαφορετικές ρυθμίσεις του ANN και διαφορετκές αρχιτεκτονικές.

Όπως παρουσιάζεται στο σχήμα 4 για διαφορετικές τιμές ρυθμού μάθησης έχουμε διαφορετικές συγκλίσεις. Η καλύτερη σύγκλιση του μοντέλου παρατηρείται για $\rho = 0.1$ ενώ όσο οι τιμές του learning rate αποκλίνουν (μεγαλύτερες ή μικρότερες) το μοντέλο συγκλίνει πιο αργά προς μηδενικές, λανθασμένες, προβλέψεις.



Σχήμα 4: Σφάλμα πρόβλεψης κατά την εκπαίδευση για διαφορετικά learning rates

Τα συμπεράσματα που αντλούνται από τον πειραματισμό με όλες τις ζητούμενες ρυθμίσεις είναι τα ακόλουθα

1. Ρυθμός Μάθησης (Learning Rate): Ο ρυθμός μάθησης επηρεάζει το πώς ενημερώνονται οι παράμετροι του μοντέλου κατά τη διάρκεια της εκπαίδευσης. Ένας υψηλός ρυθμός μάθησης μπορεί να οδηγήσει σε ασταθείς ενημερώσεις και να καθυστερήσει τη σύγκλιση. Από την άλλη πλευρά, ένας πολύ χαμηλός ρυθμός μάθησης μπορεί να οδηγήσει σε αργή σύγκλιση ή να κολλήσει σε τοπικά ελάχιστα. Επομένως, είναι σημαντικό να γίνει σωστή επιλογή ρυθμού μάθησης που εξασφαλίζει γρήγορη και σταθερή σύγκλιση του μοντέλου.

   Αριθμός Epochs: Ο αριθμός epochs καθορίζει πόσες φορές το μοντέλο θα εκπαιδευτεί στο σύνολο των δεδομένων εκπαίδευσης. Ένας μεγαλύτερος αριθμός epochs μπορεί να επιτρέψει στο μοντέλο να μάθει περισσότερες λεπτομέρειες και παραμέτρους, αλλά μπορεί επίσης να οδηγήσει σε υπερπροσαρμογή (overfitting) στα δεδομένα εκπαίδευσης. Είναι σημαντικό να

βρεθεί ένας ισορροπημένος αριθμός epochs που εξασφαλίζει τη σύγκλιση και τη γενίκευση του μοντέλου.

Batch Size: Το batch size καθορίζει πόσα δείγματα θα χρησιμοποιηθούν για κάθε ενημέρωση των παραμέτρων του μοντέλου. Ένα μικρό batch size μπορεί να οδηγήσει σε ασταθείς ενημερώσεις, ενώ ένα μεγάλο μέγεθος παρτίδας μπορεί να απαιτήσει περισσότερη μνήμη και υπολογιστική ισχύ. Επιλέγοντας ένα κατάλληλο batch size, εξισορροπείται η απόδοση και η αποδοτικότητα του μοντέλου.

2. Συνάρτηση Ενεργοποίησης (Activation Function) και Συνάρτηση Σφάλματος (Loss Function): Η επιλογή της συνάρτησης ενεργοποίησης εξαρτάται από το πρόβλημα που επιδιώκεται να λυθεί. Η συνάρτηση tanh (υπερβολική εφαπτομένη) είναι μια συνάρτηση ενεργοποίησης που παράγει έξοδο στο διάστημα [-1, 1]. Χρησιμοποιείται συχνά σε προβλήματα που απαιτούν κοινωνικοποίηση των δεδομένων.

   Η επιλογή της συνάρτησης σφάλματος εξαρτάται επίσης από το πρόβλημα και τον τύπο της εξόδου που επιδιώκεται να προβλεθεί. Η συνάρτηση Mean Square Error (MSE) είναι μια συνάρτηση σφάλματος που υπολογίζει το τετραγωνικό μέσο του σφάλματος μεταξύ των προβλέψεων και των πραγματικών τιμών. Είναι κατάλληλη για προβλήματα που η ακρίβεια απόδοσης μετριέται με βάση την απόκλιση ανάμεσα στην πρόβλεψη και την πραγματική τιμή.

3. Διαφορετικές Αρχιτεκτονικές Δικτύου: Η αρχιτεκτονική αφορά τον αριθμό και τη διάταξη των επιπέδων (layers) και των νευρώνων σε ένα μοντέλο ANN. Η επιλογή διαφορετικών αρχιτεκτονικών, όπως περισσότερα layers ή/και διαφορετικός αριθμός νευρώνων στο hidden layer, μπορεί να επηρεάσει την απόδοση του μοντέλου. Πειραματισμός με διάφορες αρχιτεκτονικές μπορεί να αποκαλύψει ποια αρχιτεκτονική παρέχει την καλύτερη απόδοση για το συγκεκριμένο πρόβλημα.

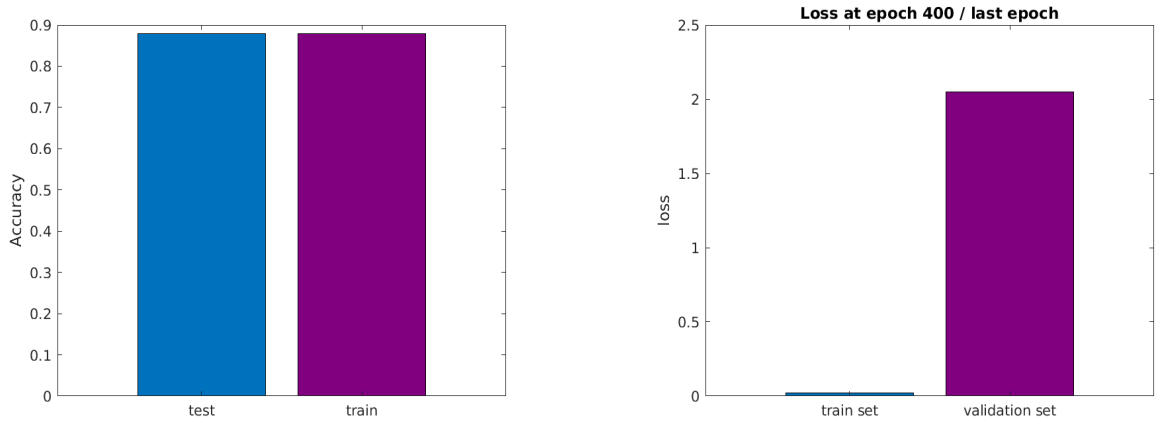
Επομένως, η επιλογή των παραπάνω παραμέτρων είναι εξαιρετικά σημαντική και εξαρτάται από τη φύση του προβλήματος, τον τύπο των δεδομένων και τις απαιτήσεις απόδοσης. Ο πειραματισμός με διάφορες παραμέτρους και αρχιτεκτονικές είναι σημαντικός για τη βελτιστοποίηση και την επίτευξη των επιθυμητών αποτελεσμάτων.

7 Θέμα 6ο - Convolutional Neural Networks for Image Recognition

7.1 Ερώτηση 1

Στο παρόν ερώτημα ρυθμίστηκαν τα **epochs = 400** και έτρεξε το μοντέλο. Για κάθε epoch υπάρχουν τα δεδομένα εκπαίδευσης (training data) καθώς και ένα σύνολο για επαλήθευση (validation data). Κατά την εκπαίδευση του μοντέλου παρουσιάζονται το κόστος (loss) και η ακρίβεια (accuracy) για τα δεδομένα εκπαίδευσης και επαλήθευσης. Στο σχήμα 5 παρουσιάζεται η ακρίβεια των προβλέψεων (5a) και το αντίστοιχο κόστος (5b) για δεδομένα εκπαίδευσης και επαλήθευσης. Επισημαίνεται πως το train accuracy είναι ο μέσος όρος των τιμών για κάθε epoch.

Γίνεται αντιληπτό πως ενώ υπάρχουν πολύ μικρές διαφορές στην ακρίβεια των προβλέψεων μεταξύ των δειγμάτων εκπαίδευης και επαλήθευσης (Test accuracy = 0.8796 και Train accuracy = 0.9353), παρατηρείται μία μία μεγάλη διαφορά στο κόστος μεταξύ των ίδιων δειγμάτων. Αυτό το σενάριο μπορεί να συμβεί όταν η συνάρτηση απώλειας (κόστους) που χρησιμοποιείται για αξιολόγηση διαφέρει από τη συνάρτηση ακρίβειας.



(a) Ακρίβεια των προβλέψεων

(b) Κόστος των προβλέψεων

Σχήμα: 5

Η ακρίβεια είναι μια συνάρτηση που μετρά το ποσοστό σωστών προβλέψεων. Υπολογίζεται διαιρώντας τον αριθμό των σωστών προβλέψεων με τον συνολικό αριθμό δειγμάτων. Από την άλλη πλευρά, η συνάρτηση απώλειας μετρά την απόκλιση μεταξύ των προβλεπόμενων εξόδων και των πραγματικών ετικετών, παρέχοντας ένα συνεχή μέτρο της απόδοσης του μοντέλου. Η συνάρτηση κόστους του συγκεκριμένου μοντέλου είναι η cross entropy.

Όπως φαίνεται κυρίως στο σχήμα 5b, το μοντέλο παρουσιάζει το πρόβλημα του overfitting. Το πρόβλημα αυτό συμβαίνει όταν ένα μοντέλο μάθει να αποδίδει εξαιρετικά καλά στα δεδομένα εκπαίδευσης, αλλά αποτυγχάνει να τα γενικεύσει σε νέα δεδομένα. Ουσιαστικά, το μοντέλο γίνεται υπερβολικά προσαρμοσμένο στο σετ εκπαίδευσης και χάνει την ικανότητά του να προβλέπει με ακρίβεια τα νέα παραδείγματα.

Το μοντέλο της δοθείσας άσκησης αρχίζει να καταγράφει τον θόρυβο και τις τυχαίες διακυμάνσεις στα δεδομένα εκπαίδευσης αντί να μάθει τα πραγματικά μοτίβα και τις σχέσεις που υπάρχουν. Συγκεκριμένα, το κόστος αρχικά είναι μικρό και στα πρώτα epochs μειώνεται μέχρι ένα σημείο απ' όπου ξεκινάει να αυξάνεται συνεχώς.

7.2 Ερώτηση 2

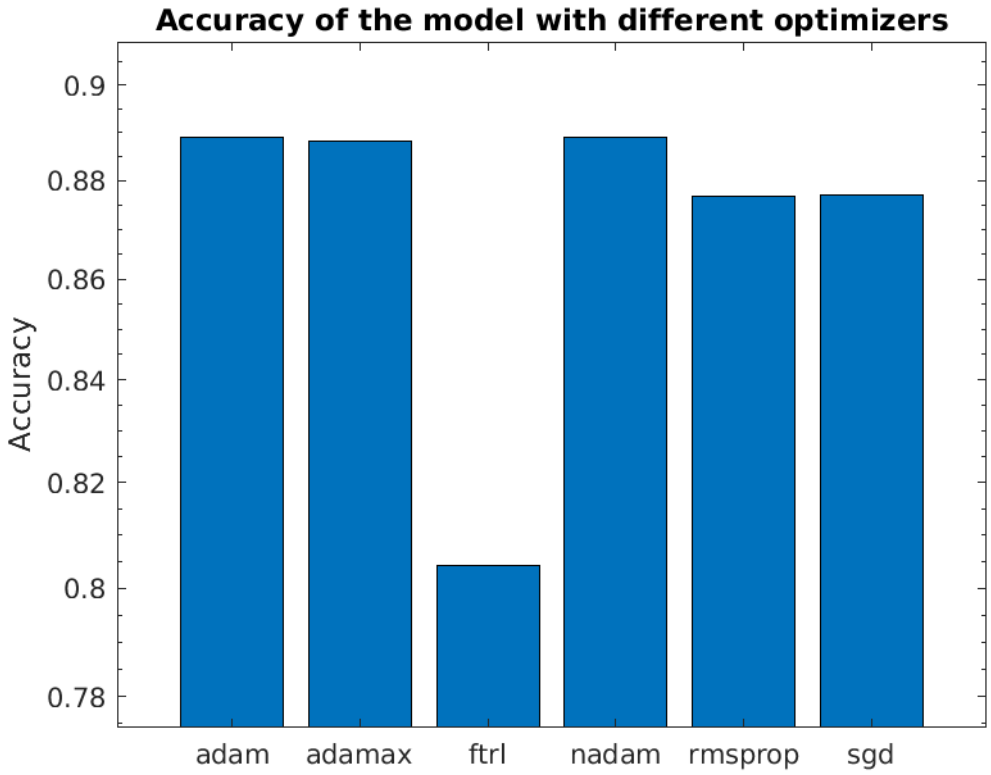
Ο optimizer για ένα νευρωνικό δίκτυο εξαρτάται από πολλούς παράγοντες, όπως το πρόβλημα που αντιμετωπίζεται, το μέγεθος των δεδομένων εκπαίδευσης, την αρχιτεκτονική του μοντέλου και άλλες παραμέτρους της εκπαίδευσης. Κάθε optimizer έχει τις δικές του ιδιαιτερότητες και πλεονεκτήματα.

Όπως προκύπτει από τα αποτελέσματα που παρουσιάζονται στο σχήμα 6 η ταξινόμηση των αλγορίθμων με φθίνουσα ακρίβεια είναι η ακόλουθη:

$$adam > nadam > adamax > sgd > rmsprop > ftrl$$

7.3 Ερώτηση 3,4,5

Αρχικά, προστέθηκε Batch Normalization αμέσως μετά τα επίπεδα conv1, conv2, conv3, conv4, conv5, dense1. Με αυτόν τον τρόπο αυξήθηκε η ταχύτητα σύγκλισης δηλαδή, στο CNN με Batch



Σχήμα: 6: Ακρίβεια προβλέψεων για διαφορετικούς optimizers

Normalization το κόστος εκπαίδευσης (trainig loss) συγκλίνει πιο γρήγορα από το νευρωνικό χωρίς Batch Normalization.

Αυτό συμβαίνει διότι με το Batch Normalization κοινωνικοποιείται η κατανομή της εισόδου κάθε του layer. Δηλαδή, η είσοδος μετά την κοινωνικοποίηση έχει μοναδιαία τυπική απόκλιση και μηδενική μέση τιμή. Με αυτόν τον τρόπο εξαλείφονται τυχαίες (μεγάλες) διακυμάνσεις οι οποίες καθυστερούν τη σύγκλιση του μοντέλου. Παρ' όλη τη γρήγορη σύγκλιση το overfitting συνέχισε να υφίσταται.

Έπειτα, προστέθηκε Drop Out αμέσως μετά τα Max Pooling layers. Αυτό είχε ως αποτέλεσμα την αποφυγή του overfitting. Ο αλγόριθμος Dropout επιτυγχάνει τη μείωση του overfitting εφαρμόζοντας την απόρριψη (dropout) τυχαίων νευρώνων κατά την εκπαίδευση. Με αυτόν τον τρόπο κάθε νευρώνας μαθαίνει ανεξάρτητα από τους υπόλοιπους και είναι πιο ικανός στο να γενικεύσει τα δεδομένα εκπαίδευσης σε νέα δεδομένα.

8 Πηγές

- www.tensorflow.org κατά την διάρκεια υλοποίησης του κώδικα Στο θέμα 6ο.
- en.wikipedia.org/wiki/K-means_clusteringApplications Κατα την επίλυση του θέματος 5.
- www.overleaf.com/learn/latex/Algorithms Template για τον ψευδοκώδικα K-Means.