
Αναφορά αποτελεσμάτων HomeWork 3

Μπανέλας Δημήτριος: 2018030140

Μάθημα: Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων

Διδάσκων θεωρίας: Ζερβάκης Μιχάηλ

Διδάσκων εργαστηρίου: Διακολουκάς Βασίλειος

Περιεχόμενα αναφοράς και οδηγίες εκτέλεσης κώδικα

Ο κώδικας που αναπτύχθηκε για τις ανάγκες των ασκήσεων βρίσκεται στα αρχεία με κώδικα που έχουν ήδη δοθεί. Ο τρόπος εκτέλεσης έμεινε ίδιος με πριν. Πιο συγκεκριμένα, μετά απο κάθε ενέργεια, το πρόγραμμα γίνεται pause και περιμένει το πάτημα ενός πλήκτρου από τον χρήστη για να συνεχιστεί. Σημαντικό είναι, να υπάρχει παρατήρηση του console καθ' όλη τη διάρκεια εκτέλεσης του κάθε προγράμματος, καθώς εκεί εμφανίζονται σημαντικές πληροφορίες.

Σε πολλά σημεία υπάρχουν blocks κώδικα που βρίσκονται μέσα σε σχόλια. Ο κώδικας αυτός γράφτηκε για την καλύτερη κατανόηση των εννοιών και δεν είναι απαραίτητος για την ορθή λειτουργία του κάθε προγράμματος. Τέλος, ο κώδικας περιέχει επεξηγηματικά σχόλια έτσι ώστε να μπορεί να διαβαστεί με σχετική ευκολία.

Θέμα 1: Feature Selection - Classification - Cross Validation - Overfitting

Σε αυτή την άσκηση, ασχοληθήκαμε με το feature selection και το cross validation - leave one out, σαν μέθοδο αξιολόγησης του μοντέλου. Έχουμε δεδομένα για 25 άτομα, με 1000 χαρακτηριστικά. Καθώς οι τιμές των χαρακτηριστικών παράγονται τυχαία, δεν υπάρχει καμία πληροφορία για τα άτομα, κάτι το οποίο θα διαπιστώσουμε με τους παρακάτω ταξινομητές. Σε όλα τα παρακάτω πειράματα, για το feature selection έχει χρησιμοποιηθεί ως similarity measure, το Pearson correlation coefficient μεταξύ των χαρακτηριστικών και των labels.

1.α Ταξινομητής χωρίς feature selection

Όπως διαπιστώνουμε μέσω του κώδικα, τα δεδομένα δεν παρέχουν καμία πληροφορία στον ταξινομητή, σχετικά με τον διαχωρισμό των ατόμων και έτσι η ακρίβεια που επιτυγχάνεται κατά μέσο όρο είναι 50%.

1.β Ταξινομητής με feature selection εντός του cross validation

Με τη χρήση της μεθόδου cross validation - leave one out χωρίζουμε το dataset σε N-1 training δείγματα και 1 test δείγμα. Το feature selection γίνεται με τα N-1 δείγματα. Λόγω της τυχειότητας των δεδομένων το μοντέλο έχει και πάλι μέσο όρο accuracy 50%.

1.γ Ταξινομητής με feature selection εκτός του cross validation

Στο συγκεκριμένο πείραμα, το feature selection έγινε χρησιμοποιώντας όλα τα δείγματα, πριν τη διαδικασία του cross validation. Αυτό σημαίνει πως το test δείγμα του cross validation έχει συμμετάσχει στο feature selection και έτσι, το μοντέλο επιτυγχάνει κάθε φορά

ακρίβεια 100%. Αυτά τα ποσοστά μας δείχνουν ότι έχει συμβεί overfitting, ότι δηλαδή το μοντέλο έχει υπερπροσαρμοστεί στα training δεδομένα. Αυτό μας δείχνει ότι θα πρέπει να αποφύγουμε το feature selection πριν από το cross validation, και να το ενσωματώσουμε μέσα σε αυτό.

Θέμα 2: Υλοποίηση ενός απλού νευρωνικού δικτύου

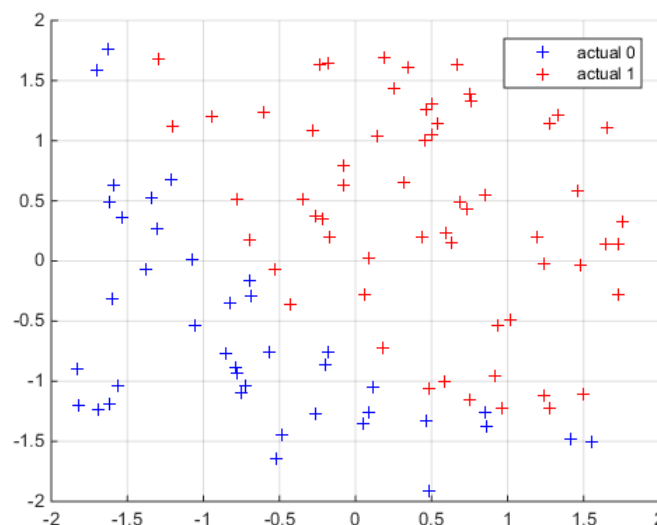
Σε αυτή την άσκηση υλοποιήσαμε ένα νευρωνικό δίκτυο και το χρησιμοποιήσαμε για ταξινόμηση. Για activation function χρησιμοποιήθηκε η λογιστική συνάρτηση

$$f(z) = \frac{1}{1 + e^{-z}}$$

ενώ για συνάρτηση κόστους χρησιμοποιήθηκε ο μέσος όρος της cross entropy loss πάνω σε B δείγματα (batch).

$$J(Y, \hat{Y}; W, b) = \frac{1}{B} \sum_i (-y^{(i)} \ln(\hat{y}^{(i)}) - (1 - y^{(i)}) \ln(1 - \hat{y}^{(i)}))$$

Το κανονικοποιημένο dataset φαίνεται στο παρακάτω σχήμα:



Στο **forward pass** υλοποιήθηκε η εξής συνάρτηση:

$$\hat{y}^{(i)} = f(x^{(i)}W + b)$$

Στο **backward pass** υλοποιήθηκαν οι μερικές παράγωγοι:

$$\frac{\partial J}{\partial W} = \frac{1}{B} \sum_i (\hat{y}^{(i)} - y^{(i)}) x^{(i)T}$$

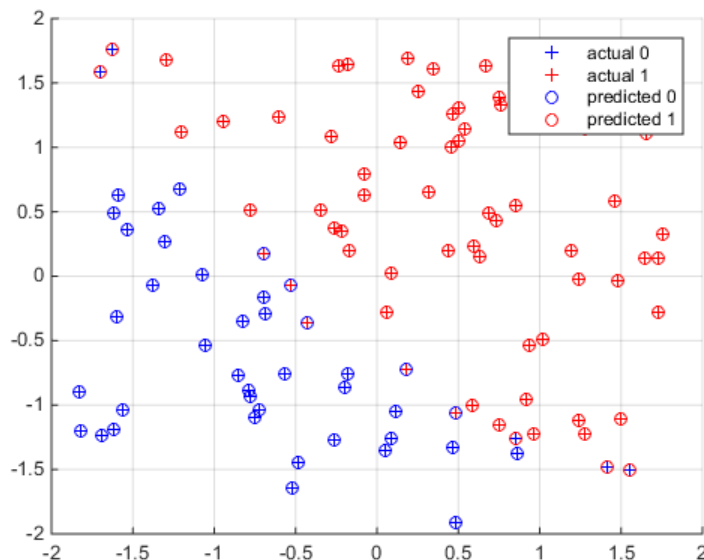
$$\frac{\partial J}{\partial b} = \frac{1}{B} \sum_i (\hat{y}^{(i)} - y^{(i)})$$

Οι παραπάνω παράγωγοι στη συνέχεια χρησιμοποιήθηκαν για την υλοποίηση του αλγορίθμου gradient descent, με στόχο την εύρεση του διανύσματος βαρών που ελαχιστοποιεί το κόστος.

$$W \leftarrow W - \rho \frac{\partial J}{\partial W}$$

$$b \leftarrow b - \rho \frac{\partial J}{\partial b}$$

Το παραπάνω νευρωνικό νευρωνικό δίκτυο έχει accuracy 88% πάνω στα training δεδομένα, για 55 epochs εκπαίδευσης. Παρατίθενται τα πραγματικά δεδομένα σε σχέση με τις προβλέψεις που έκανε το νευρωνικό.



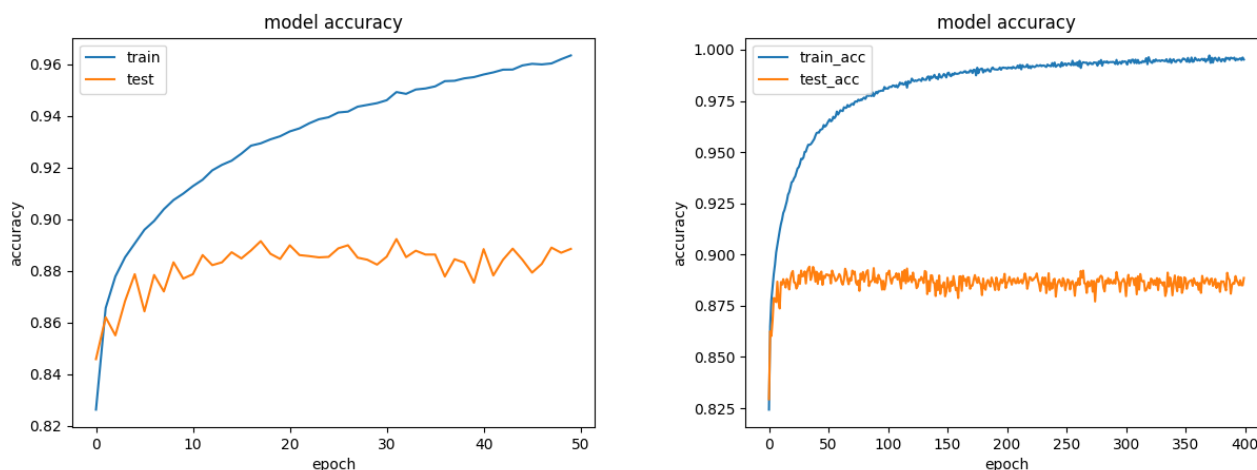
Μπορούμε να παρατηρήσουμε ότι οι λανθασμένες προβλέψεις βρίσκονται επί τω πλείστον κοντά στη νοητή γραμμή η οποία διαχωρίζει τις κλάσεις μας, κάτι το οποίο περιμέναμε, μιας και αυτές είναι οι οριακές καταστάσεις. Το να αναπτύσσαμε ένα όριο απόφασης το οποίο θα απέφευγε αυτά τα λάθη, θα είχε ως αποτέλεσμα overfitting.

Θέμα 3: Convolutional Neural Networks for Image Recognition

Σε αυτή την άσκηση θα μελετήσουμε διάφορες αρχιτεκτονικές ταξινομητών νευρωνικών δικτύων, για την ταξινόμηση του σύνολου δεδομένων Fashion - MNIST.

Στα πρώτα 2 ερωτήματα ασχοληθήκαμε με το ήδη υπάρχον νευρωνικό δίκτυο. Το διάγραμμα των τιμών του accuracy για κάθε περίπτωση παρατίθεται παρακάτω.

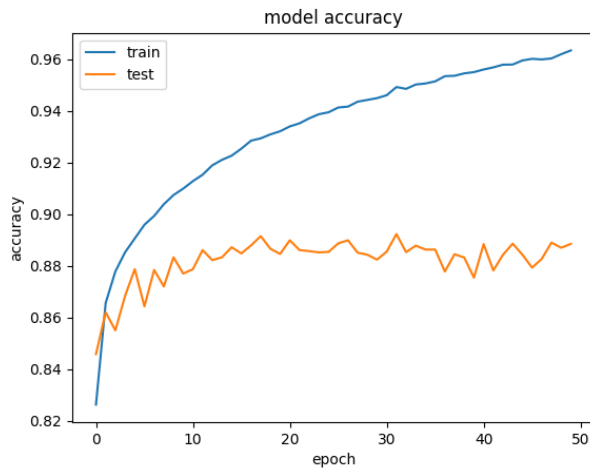
1. 400 epochs και adam optimizer



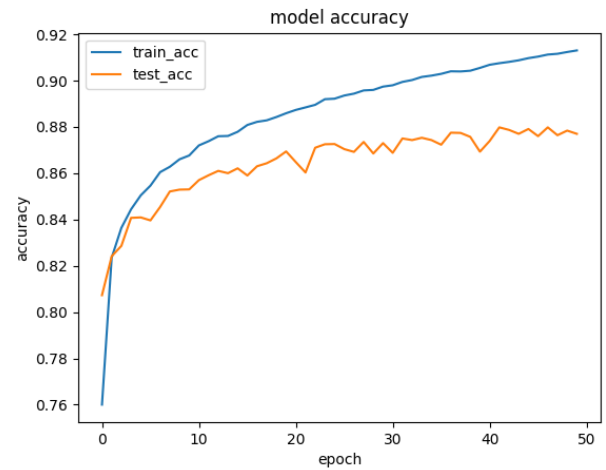
Μπορούμε να δούμε ότι όσο τα epochs αυξάνονται, το train_acc αυξάνεται, ενώ το test_acc μειώνεται ελάχιστα ($0,8895 \rightarrow 0,8817$). Αυτό συμβαίνει καθώς όσο περισσότερο χρόνο εκαιδεύεται το νευρωνικό, τόσο περισσότερο τα βάρη του γίνονται fit στα training δεδομένα.

2. Μελέτη απόδοσης για διάφορους optimizers

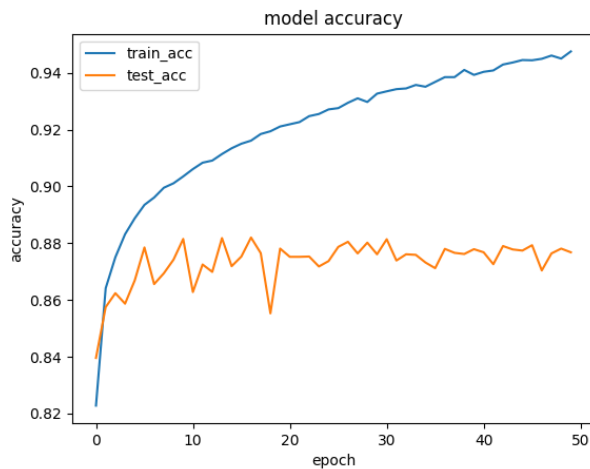
Παρακάτω φαίνονται τα διαγράμματα των τιμών accuracy για όλους τους optimizers (50 epochs).



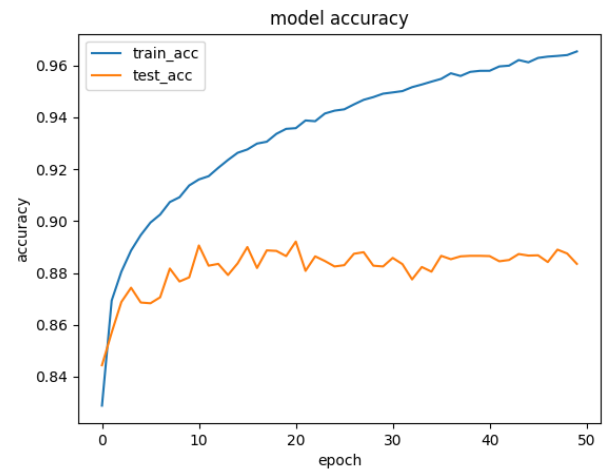
(a) adam



(b) sgd

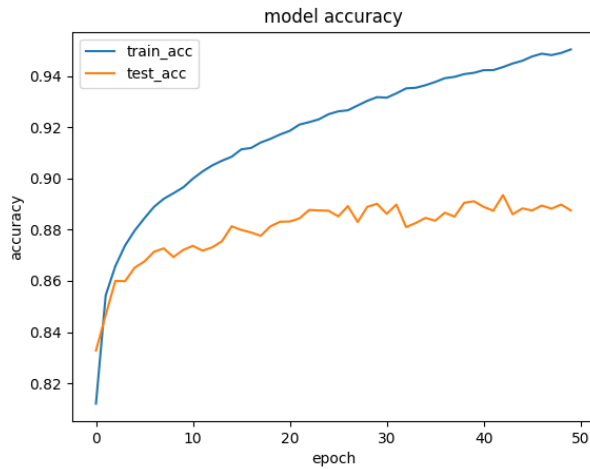


(c) rmsprop

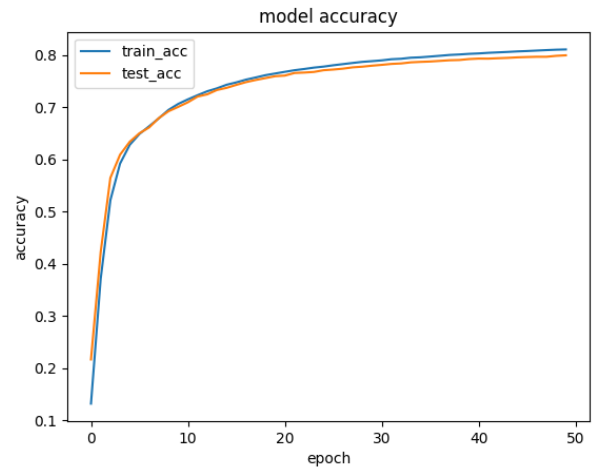


(d) nadam

Και adamax, ftrl



(a) adam



(b) sgd

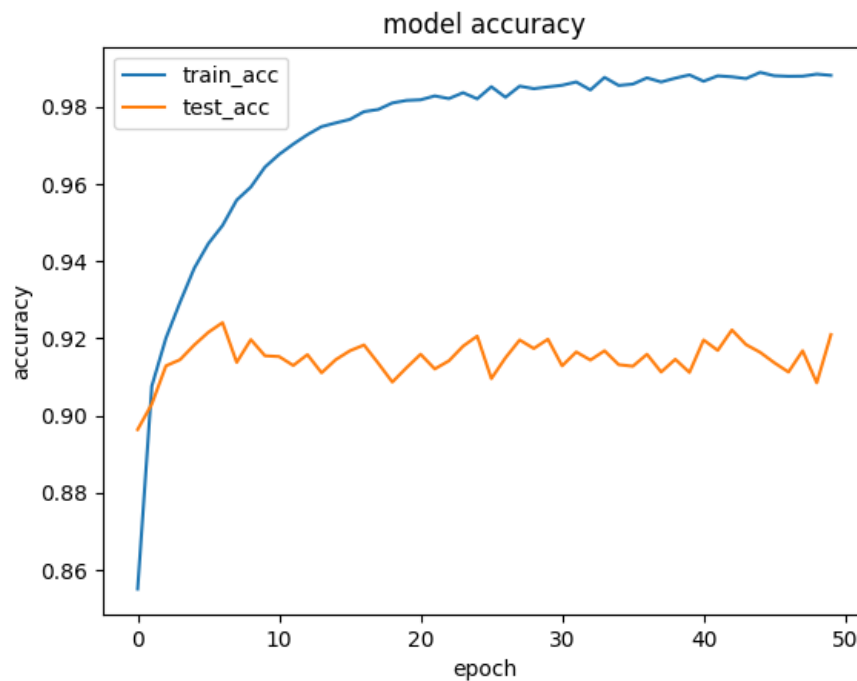
Παρατίθεται επίσης και ο πίνακας με τα τελικά accuracies των optimizers.

optimizer	train_acc	val_acc
adam	0.9622	0.8921
sgd	0.9137	0.8817
rmsprop	0.9477	0.8814
nadam	0.9654	0.8835
adamax	0.9473	0.8899
ftrl	0.8194	0.8089

Παρατηρούμε πως οι δύο καλύτεροι αλγόριθμοι βελτιστοποίησης είναι οι adam και nadam. Θα διαλέξουμε ως καλύτερο τον adam αφού πετυχαίνει καλύτερο validation accuracy και εξίσου καλό training accuracy με τον nadam.

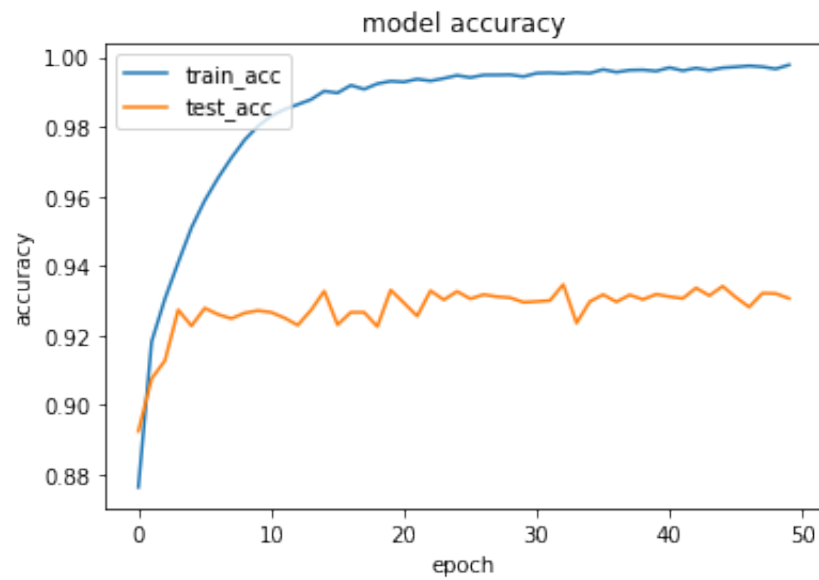
3. Κατασκευή CNN

Στη συνέχεια, θα προσεγγίσουμε το παραπάνω πρόβλημα της ταξινόμησης χρησιμοποιώντας Convolutional NNs. Αφού υλοποιήσαμε την αρχιτεκτονική της εκφώνησης, κάναμε τα παρακάτω πειράματα. Παρατίθεται το διάγραμμα των accuracies πριν οποιαδήποτε αλλαγή.

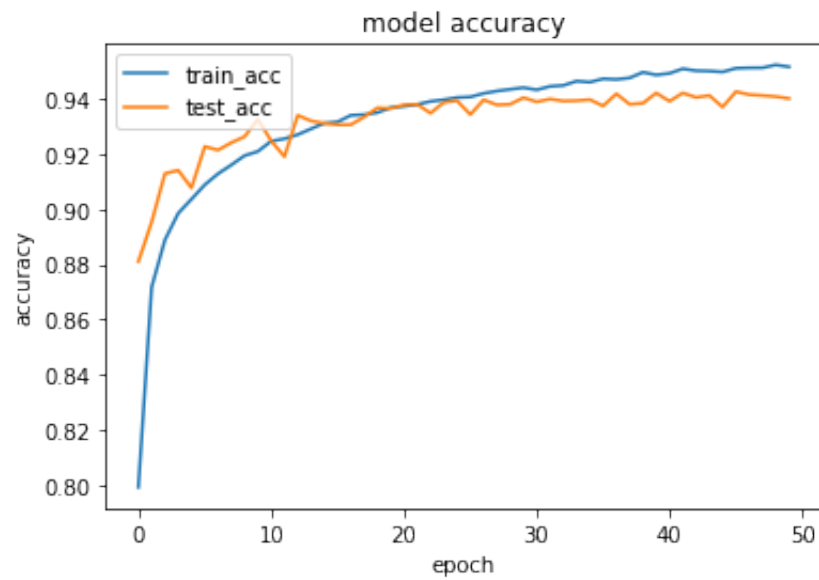


4. Προσθήκη Batch normalization

Το διάγραμμα των accuracies μετά την προσθήκη του batch normalization παρατίθεται παρακάτω:



5. Προσθήκη Dropout



Η χρήση του dropout απενεργοποιεί τυχαία εξόδους νευρώνων στο layer στο οποίο εφαρμόζεται. Αυτό έχει ως αποτέλεσμα τη μείωση του train_acc. Έτσι η τάση για overfitting μειώνεται και η απόδοση σε δεδομένα εκτός του training set αυξάνεται.