Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων (ΤΗΛ311)

Αναφορά 4ης Σειράς Ασκήσεων

Ανδρεαδάκης Αντώνης 2013030059

1. Για την άσκηση αυτή, καλούμαστε να απαντήσουμε στο ερώτημα: Κατά τη διαδικασία της ταξινόμησης, σε ποιο σημείο είναι ιδανικό να εφαρμοστεί επιλογή των χαρακτηριστικών (feature selection) ? Για τα δεδομένα προσομοίωσης του πειράματός μας (ασθενείς με αυτισμό ή όχι), δοκιμάστηκαν οι εξής τεχνικές:

   i) Leave one out χωρίς feature selection. Ταξινομούμε τα δεδομένα μας με χρήση SVM και με την κλασσική τεχνική Leave one out έχουμε ένα δεδομένο test και όλα τα υπόλοιπα δεδομένα εκπαίδευσης. Αρκετά αποτελεσματική μέθοδος και εύκολη στην υλοποίηση, πετυχαίνει ακρίβεια περίπου 50%.

   ii) Modified Leave one out χρησιμοποιώντας έναν αλγόριθμο σύγκρισης με χρήση επιλογής χαρακτηριστικών, βασισμένος στο συντελεστή συσχέτισης. Σε κάθε επανάληψη του Leave one out, εφαρμόζουμε feature selection. Η τεχνική αυτή, επίσης πετυχαίνει ακρίβεια περίπου 50% (αναμενόμενο όπως η προηγούμενη), καθώς τα τεχνητά μας δεδομένα δεν δίνουν πληροφορίες για τους ασθενείς.

   iii) Modified Leave one out με επιλογή των χαρακτηριστικών μια φορά στην αρχή. Δυστυχώς, στην περίπτωση αυτή έχουμε overfitting και για αυτό το λόγο έχουμε σταθερή ακρίβεια 100%.

   Ενδεικτικά:

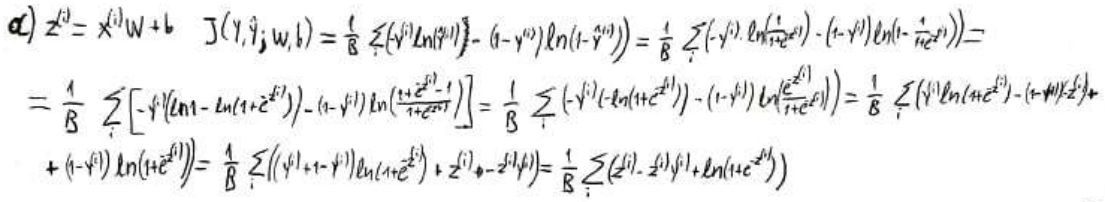
```
Classify without feature selection:
accuracy: 56.00%

Classify with feature selection inside the cross validation:
accuracy: 52.00%

Classify with feature selection outside the cross validation:
accuracy: 100.00%
```

2. Στην άσκηση αυτή, σκοπός ήταν η δημιουργία του πιο απλού νευρωνικού δικτύου που μπορεί να υπάρξει. Δηλαδή ένα δίκτυο μόνο με είσοδο και έξοδο, χωρίς κρυφό επίπεδο. Ως συνάρτηση ενεργοποίησης δικτύου, χρησιμοποιήθηκε η sigmoid συνάρτηση (την οποία έχουμε συναντήσει σε παλαιότερο σετ ασκήσεων) και ορίζεται ως $f(z) = \frac{1}{1+e^{-z}}$. Για τον υπολογισμό του σφάλματος, μεταξύ της πρόβλεψης και της πραγματικής τιμής, χρησιμοποιήθηκε η συνάρτηση cross-entropy που ορίζεται ως:

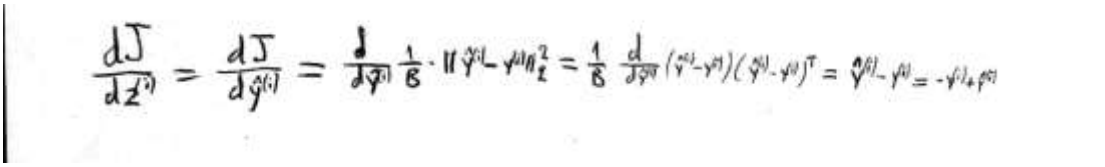
$$J(y^{(i)}, \hat{y}^{(i)}; W, b) = -y^{(i)} ln(\hat{y}^{(i)}) - (1 - y^{(i)}) ln(1 - \hat{y}^{(i)})$$

Όλη η υλοποίηση του κώδικα έγινε σε matlab (θα μπορούσε και σε python).

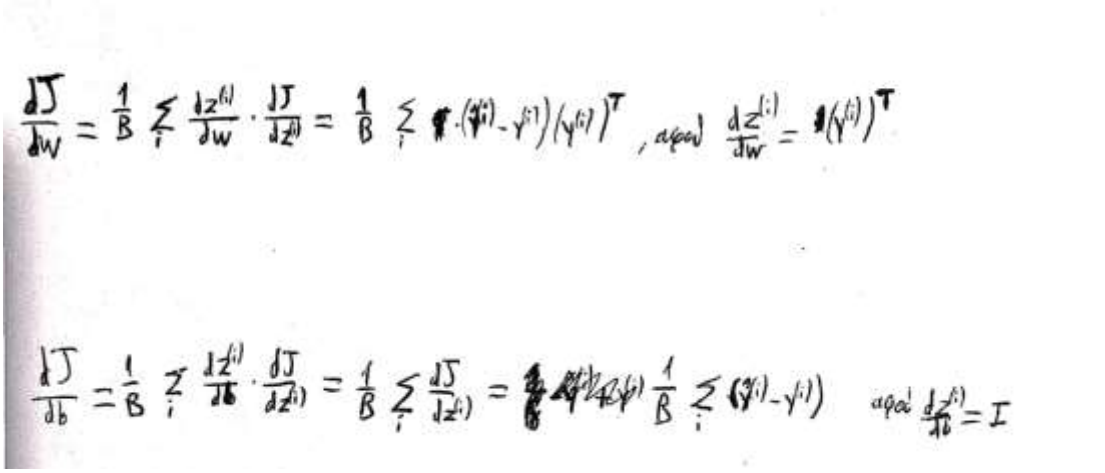
a)



b)



c)



d) Για το forward propagation του NN, χρειάστηκε να υπολογιστεί η είσοδος πολλαπλασιασμένη με τα βάρη και να προστεθεί το bias. Έπειτα, το αποτέλεσμα αυτό δίνεται ως είσοδος στην sigmoid συνάρτηση κι έτσι βρίσκουμε την έξοδο του δικτύου μας. Στη συνέχεια, αξιολογούμε το αποτέλεσμα με χρήση της cross-entropy. Για να ελαχιστοποιήσουμε αυτές

τις απώλειες και να μπορέσουν να ανανεωθούν οι τιμές των βαρών και το bias του δικτύου, γίνεται μια «ανατροφοδότηση» της προβλεπόμενης εξόδου στο δίκτυο. Η διαδικασία αυτή λέγεται back propagation. Με την εύρεση των επιθυμητών παραγώγων, ανανεώνονται τα βάρη και το bias όπως φαίνεται στον κώδικα. Στην αρχή, με δεδομένο το num_epochs = 55 έχουμε τα εξής αποτελέσματα που φαίνονται παρακάτω:

```
epoch_loss =9.7159      4.1386
epoch_loss =9.5832      4.1158
epoch_loss =9.4528      4.0935
epoch_loss =9.3248      4.0717
epoch_loss =9.1991      4.0503
epoch_loss =9.0757      4.0295
epoch_loss =8.9546      4.0091
epoch_loss =8.8358      3.9891
epoch_loss =8.7193      3.9696
epoch_loss =8.6051      3.9505
epoch_loss =8.4931      3.9317
epoch_loss =8.3833      3.9134
epoch_loss =8.2757      3.8954
epoch_loss =8.1703      3.8778
epoch_loss =8.067      3.8605
epoch_loss =7.9659      3.8436
epoch_loss =7.8668       3.827
epoch_loss =7.7698      3.8107
epoch_loss =7.6749      3.7947
epoch_loss =7.582      3.7791
epoch_loss =7.4911      3.7637
epoch_loss =7.4021      3.7486
epoch_loss =7.315      3.7337
epoch_loss =7.2298      3.7191
epoch_loss =7.1465      3.7048
epoch_loss =7.065      3.6907
epoch_loss =6.9852      3.6769
epoch_loss =6.9072      3.6633
epoch_loss =6.831      3.6499
epoch_loss =6.7564      3.6368
epoch_loss =6.6834      3.6238
epoch_loss =6.6121      3.6111
epoch_loss =6.5423      3.5986
epoch_loss =6.4741      3.5863
epoch_loss =6.4074      3.5741
epoch_loss =6.3422      3.5622
epoch_loss =6.2784      3.5504
epoch_loss =6.216      3.5389
epoch_loss =6.155      3.5275
epoch_loss =6.0953      3.5162
epoch_loss =6.037      3.5051
epoch_loss =5.9799      3.4942
epoch_loss =5.9241      3.4835
```

```
epoch_loss =5.9241      3.4835
epoch_loss =5.8695      3.4729
epoch_loss =5.8161      3.4624
epoch_loss =5.7639      3.4521
epoch_loss =5.7128       3.442
epoch_loss =5.6628       3.432
epoch_loss =5.6138      3.4221
epoch_loss =5.5659      3.4123
epoch_loss =5.5191      3.4027
epoch_loss =5.4732      3.3932
epoch_loss =5.4284      3.3839
epoch_loss =5.3844      3.3746
epoch_loss =5.3414      3.3655

Predicting the probabilities of example [45, 85]:
Propability = 47.69%
Propability = 65.26%
Propability = 17.37%
Propability = 88.81%

Accuracy = 74.00%
```

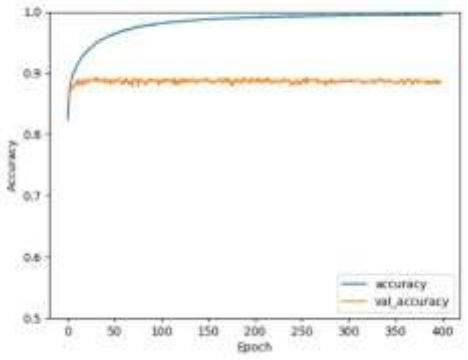
Για να δούμε την απόδοση του δικτύου, αλλάζουμε το num_epochs από 55 σε 500 έτσι, ώστε το epoch_loss να είναι το ελάχιστο δυνατό και πετυχαίνουμε accuracy 90%. Τα αποτελέσματα:

```
epoch_loss =6.4439      4.4503
epoch_loss =6.3747      4.4222
epoch_loss =6.3072      4.3948
epoch_loss =6.2413       4.368
epoch_loss =6.177      4.3418
epoch_loss =6.1143      4.3163
epoch_loss =6.0531      4.2913
epoch_loss =5.9933      4.2668
epoch_loss =5.935      4.2429
epoch_loss =5.8781      4.2195
epoch_loss =5.8225      4.1966
epoch_loss =5.7682      4.1742
epoch_loss =5.7152      4.1523
epoch_loss =5.6635      4.1308
epoch_loss =5.6129      4.1097
epoch_loss =5.5636      4.0891
epoch_loss =5.5154      4.0688
epoch_loss =5.4682       4.049
epoch_loss =5.4222      4.0296
epoch_loss =5.3772      4.0105
epoch_loss =5.3332      3.9918
epoch_loss =5.2902      3.9734
```

Παραλείφθηκαν οι ενδιάμεσες τιμές του epoch_loss (καθώς φθίνει) για χάρην απλότητας, οπότε τελικά έχουμε:

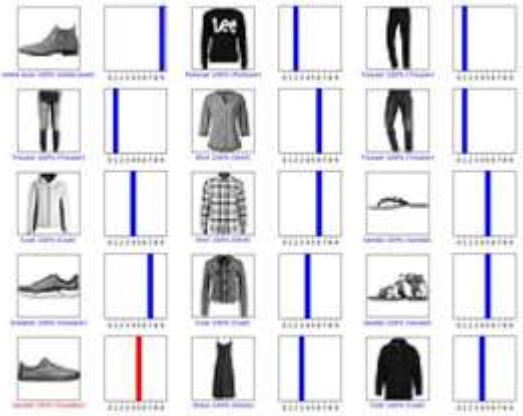
```
epoch_loss =2.311      2.2659
epoch_loss =2.31      2.2651
epoch_loss =2.309      2.2642
epoch_loss =2.3081      2.2634
epoch_loss =2.3071      2.2626
epoch_loss =2.3062      2.2618
epoch_loss =2.3052       2.261
epoch_loss =2.3043      2.2602
epoch_loss =2.3033      2.2594
epoch_loss =2.3024      2.2586
epoch_loss =2.3015      2.2579
epoch_loss =2.3005      2.2571
epoch_loss =2.2996      2.2563

Predicting the probabilities of example [45, 85]:
Propability = 4.29%
Propability = 98.79%
Propability = 3.79%
Propability = 98.98%

Accuracy = 90.00%
```

3. Στην τελευταία άσκηση του σετ, κατασκευάζουμε ένα Convolutional Neural Network για την αναγνώριση φωτογραφιών ρούχων. Πρώτα, παρακολουθούμε την εξέλιξη των δεικτών accuracy πάνω στα δεδομένα εκπαίδευσης και τον δείκτη validation accuracy στα δεδομένα επικύρωσης. Όταν ο αριθμός epochs είναι 400, βλέπουμε στο παρακάτω γράφημα την κλιμάκωση των δεικτών:

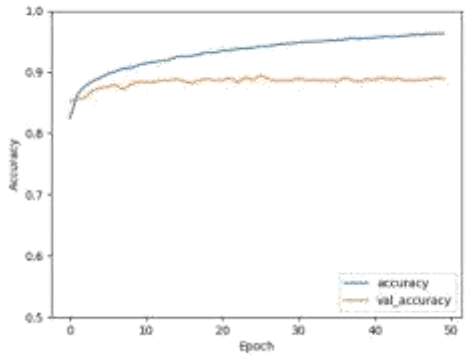


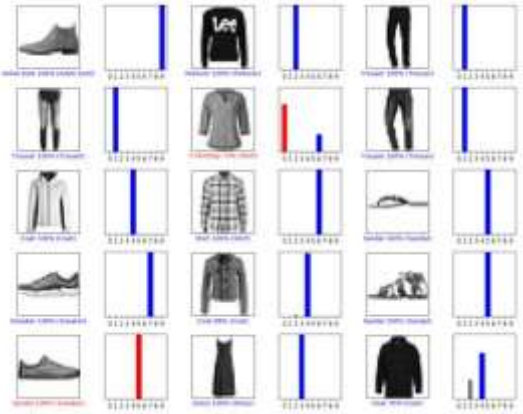
Παρατηρούμε ότι το accuracy τείνει στη μονάδα, ενώ το validation διατηρείται σταθερό κοντά στο 0,9.

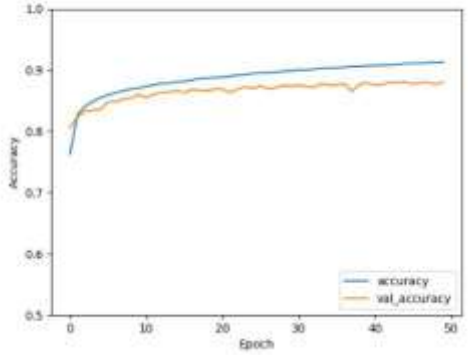
Παρακάτω βλέπουμε την πρόβλεψη του μοντέλου:

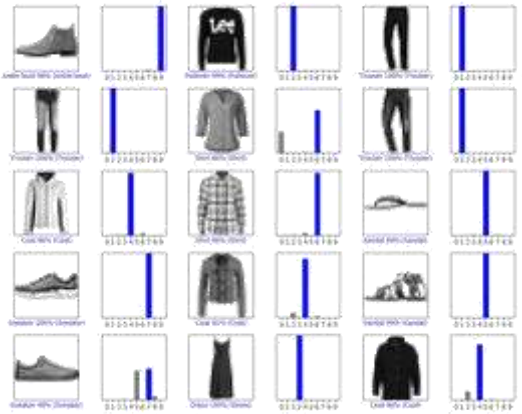


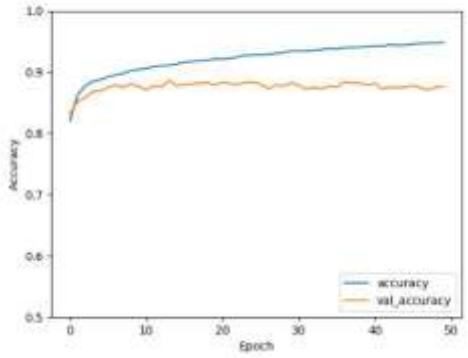
Μειώνοντας τον αριθμό του epochs απο 400 σε 50 και δοκιμάζοντας διάφορους optimizers, οι οποίοι παρέχονται από το keras api (αλγόριθμος adam, sgd, rmsprop, nadam, adamax, ftrl), παραθέτουμε τις παρακάτω εικόνες στις οποίες παρουσιάζεται το accuracy και το validation για κάθε έναν από τους αλγορίθμους και τις προβλέψεις τους.

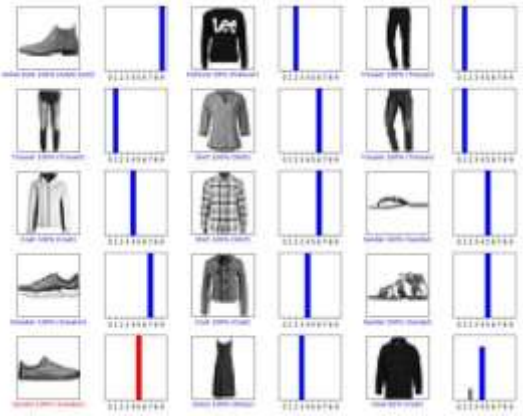


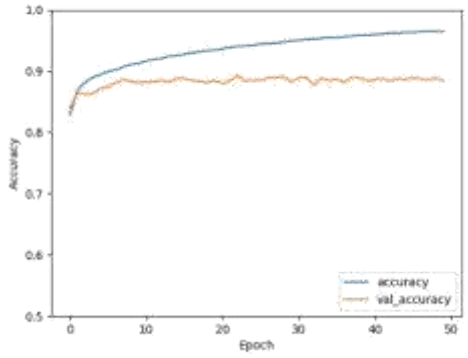


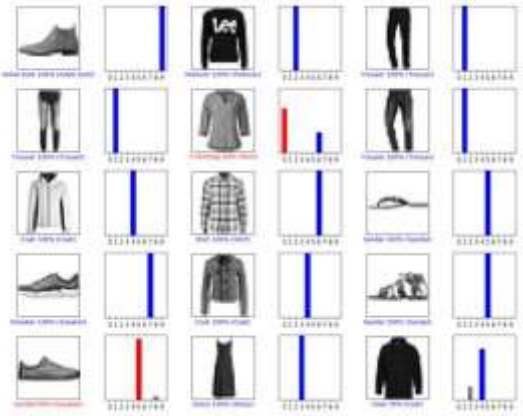


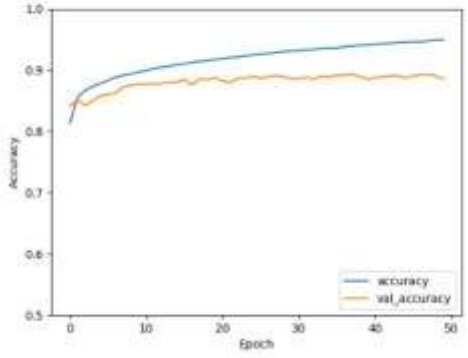


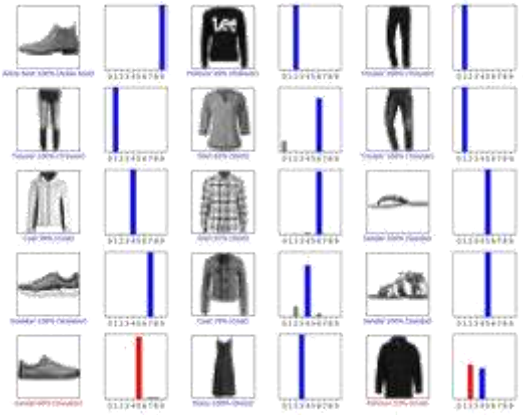

Accuracy
Epoch
accuracy
val_accuracy

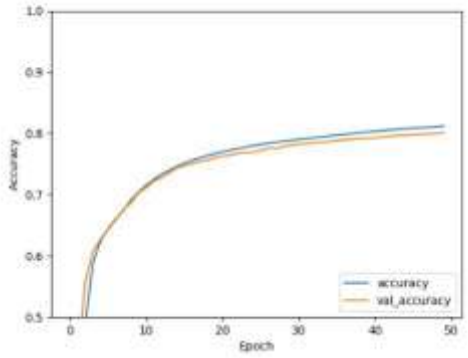


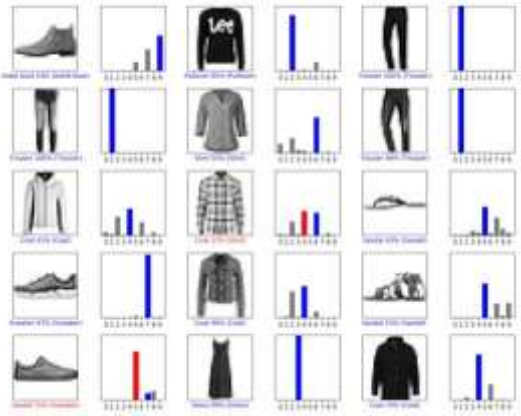

Accuracy
Epoch
accuracy
val_accuracy




Accuracy
Epoch
accuracy
val_accuracy

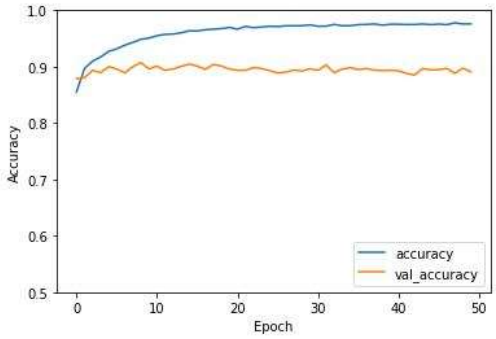


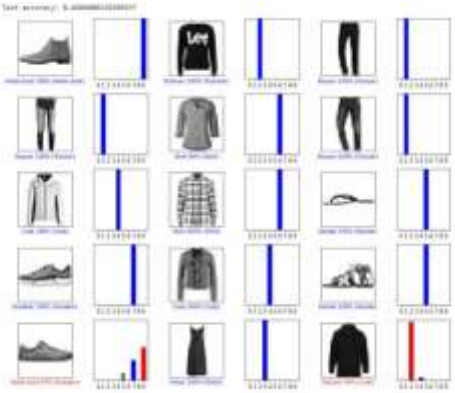




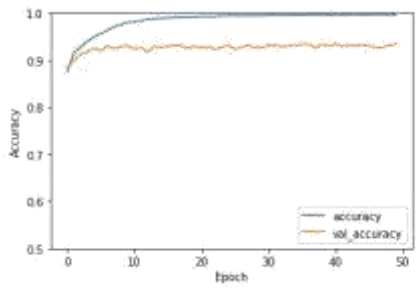
Με παρατήρηση όλων των παραπάνω, καταλήγουμε στο συμπέρασμα ότι ο adam είναι ο ιδανικότερος για το πείραμά μας, καθώς έχει το μεγαλύτερο accuracy και validation accuracy.

Στη συνέχεια, κατασκευάζουμε ένα NN σύμφωνα με τις οδηγίες της εκφώνησης. Αρχικά, υλοποιούμε το μοντέλο χωρίς batch normalization και τα αποτελέσματα της ακρίβειας και της πρόβλεψης του μοντέλου φαίνονται παρακάτω:

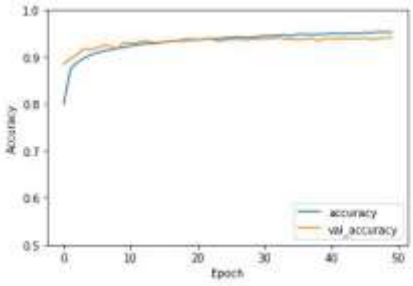


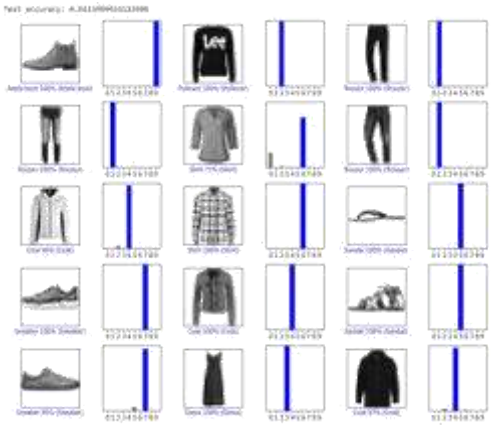


Με batch normalization:



Το ολοκληρωμένο μοντέλο:





Το validation accuracy αυξάνεται, με την προσθήκη επιπλέον επιπέδων στο μοντέλο και τείνει να ταυτιστεί με το accuracy.