

# ***Υπολογιστική Νοημοσύνη***

*3η Εργασία στα Νευρωνικά Δίκτυα*

Επίλυση προβλήματος ταξινόμησης με χρήση Multi-layer Perceptron δικτύου

Κωνσταντίνος Κωνσταντινίδης, ΑΕΜ: 9162

Στοιχεία Επικοινωνίας:

- email: [konkonstantinidis@ece.auth.gr](mailto:konkonstantinidis@ece.auth.gr)

Για παράδοση κατά την πτυχιακή εξεταστική του Φεβρουαρίου 2022

## 1. Διερεύνηση απόδοσης μοντέλου με διαφοροποιήσεις στο σχεδιασμό και τη διαδικασία εκπαίδευσης

Κατόπιν της διερεύνησης των παραμέτρων του MLP δικτύου επάνω στο MNIST dataset, έχουμε τα εξής:

*Εκπαίδευση ενός default δικτύου με τα ακόλουθα minibatch μεγέθη: {1, 256, Ntrain} - ((online, minibatch, batch)), όπου Ntrain το συνολικό πλήθος των δεδομένων εκπαίδευσης*

Όσον αφορά τους χρόνους εκπαίδευσης και για τις τρεις μεθόδους:

| Μέθοδος   | Χρόνος (sec)       |
|-----------|--------------------|
| online    | 3729.1867599487305 |
| minibatch | 44.31820797920227  |
| batch     | 23.505635023117065 |

*Πίνακας 1, Χρόνος εκπαίδευσης MLK συναρτήσει batch size*

Είναι προφανές ότι όσο μεγαλύτερο είναι το batch size, τόσο γρηγορότερα εκπαιδεύεται το MLP δίκτυο, καθώς τόσο περισσότερες εγγραφές θα επεξεργαστεί πριν μπει στην διαδικασία να ενημερώσει τις εσωτερικές παραμέτρους του.

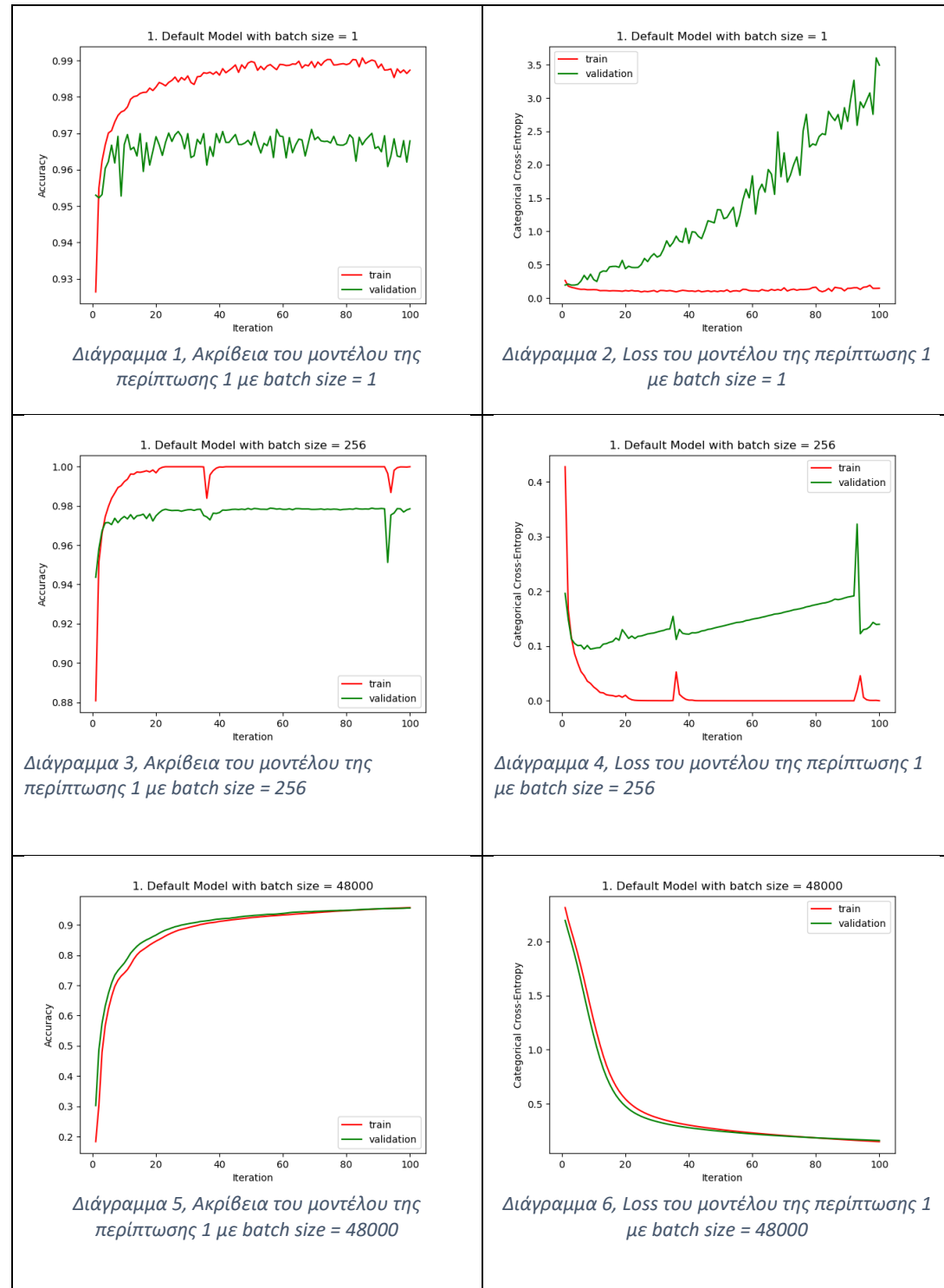
Για το σύνολο όλων μοντέλων, έχουμε τα αποτελέσματα τους επάνω στο σετ ελέγχου:

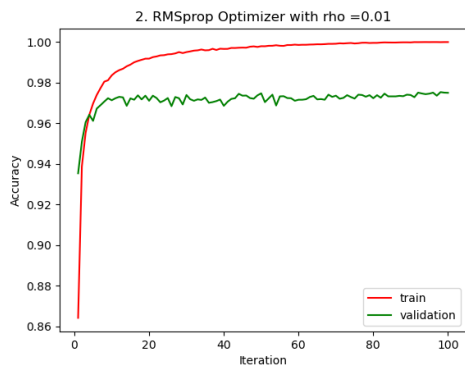
| Μοντέλο   | Χρόνος Εκτέλεσης (sec) | Accuracy |
|---|------------------------|----------|
| 1. Default Model - Batch size: 1                | 3729                   | 0.966    |
| 1. Default Model - Batch size: 256              | 44                     | 0.981    |
| 1. Default Model - Batch-size: 48000            | 23                     | 0.953    |
| 2. RMSprop Optimizer - rho: 0.01                | 50                     | 0.976    |
| 2. RMSprop Optimizer - rho: 0.99                | 47                     | 0.976    |
| 3. SGD Optimizer                                | 40                     | 0.113    |
| 4. SGD Optimizer-L2 Regularization - a: 0.1     | 42                     | 0.113    |
| 4. SGD Optimizer - L2 Regularization - a: 0.01  | 42                     | 0.113    |
| 4. SGD Optimizer - L2 Regularization   a: 0.001 | 42                     | 0.113    |
| 5. SGD Optimizer - L1 Regularization            | 52                     | 0.113    |

Είναι φανερό πως, ενώ τα μοντέλα των περιπτώσεων 1-3 τα πηγαίνουν πάρα πολύ καλά, τα μοντέλα 3-5, με την εισαγωγή κανονικοποιητών και τυχαίας

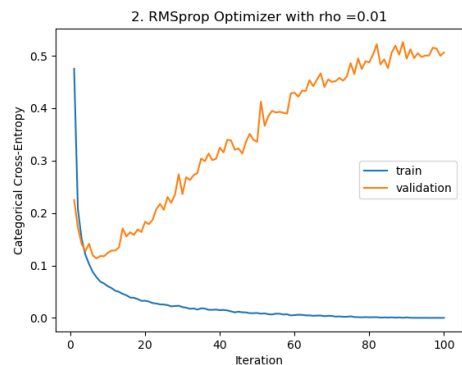
αρχικοποίησης βαρών, υποφέρουν τρομερά στην προβλεπτική τους ικανότητα, σημαντική ένδειξη της λανθασμένης πρακτικής.

Έπειτα, για το σύνολο των μοντέλων, έχουμε τις εξής καμπύλες ακρίβειας και κόστους:

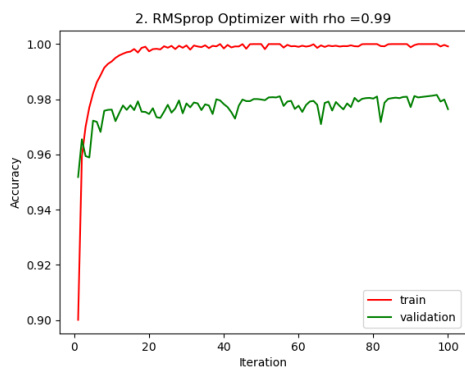




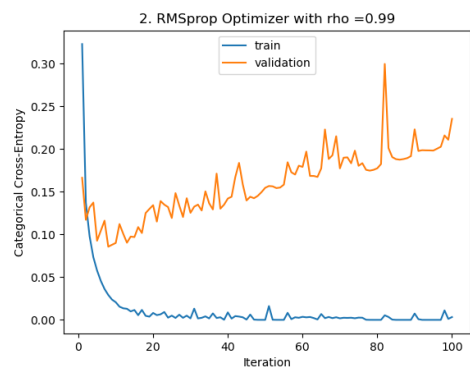
Διάγραμμα 7, Ακρίβεια του μοντέλου της περίπτωσης 2 με  $\rho=0.01$



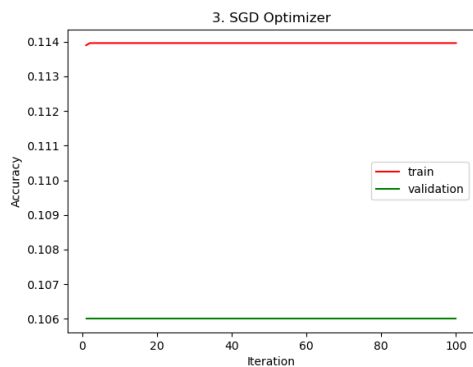
Διάγραμμα 8, Loss του μοντέλου της περίπτωσης 2 με  $\rho=0.01$



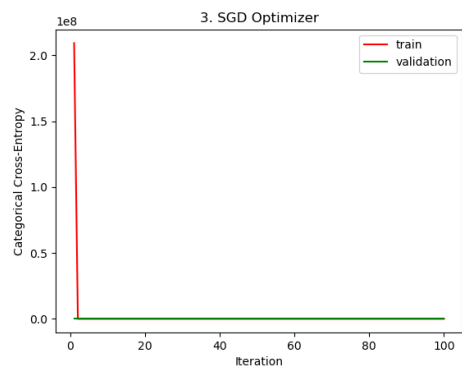
Διάγραμμα 9, Ακρίβεια του μοντέλου της περίπτωσης 2 με  $\rho=0.99$



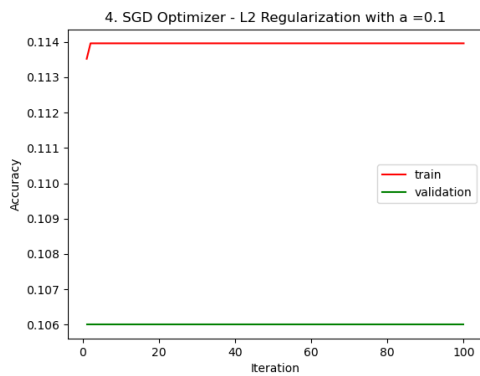
Διάγραμμα 10, Loss του μοντέλου της περίπτωσης 2 με  $\rho=0.99$



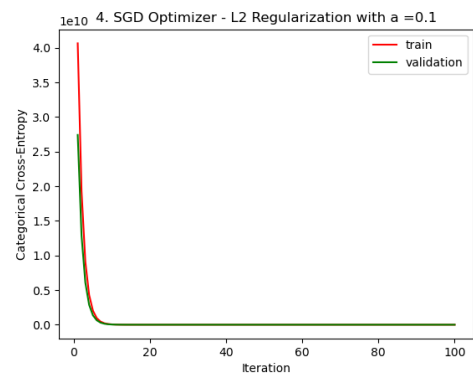
Διάγραμμα 11, Ακρίβεια του μοντέλου της περίπτωσης 3



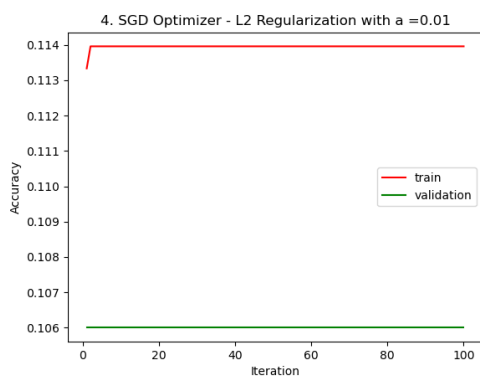
Διάγραμμα 12, Loss του μοντέλου της περίπτωσης 3



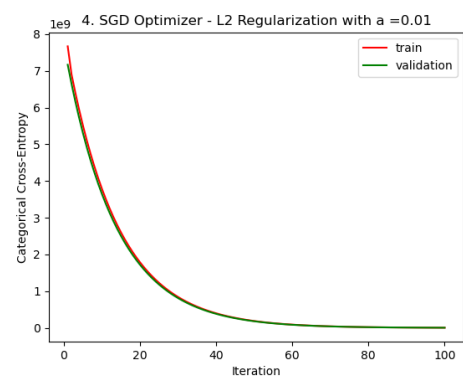
Διάγραμμα 13, Ακρίβεια του μοντέλου της περίπτωσης 4 με  $\alpha = 0.1$



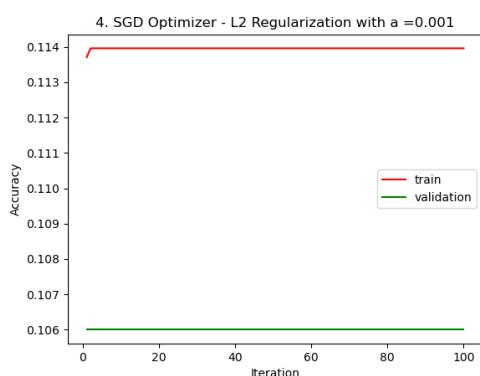
Διάγραμμα 14, Loss του μοντέλου της περίπτωσης 4 με  $\alpha = 0.1$



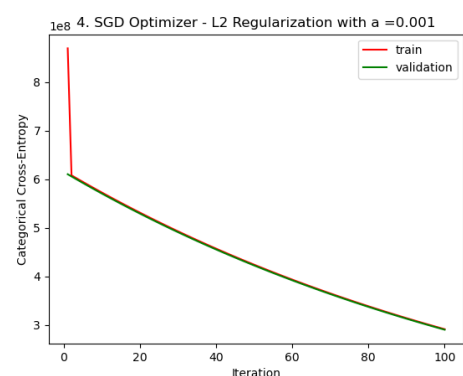
Διάγραμμα 15, Ακρίβεια του μοντέλου της περίπτωσης 4 με  $\alpha = 0.01$



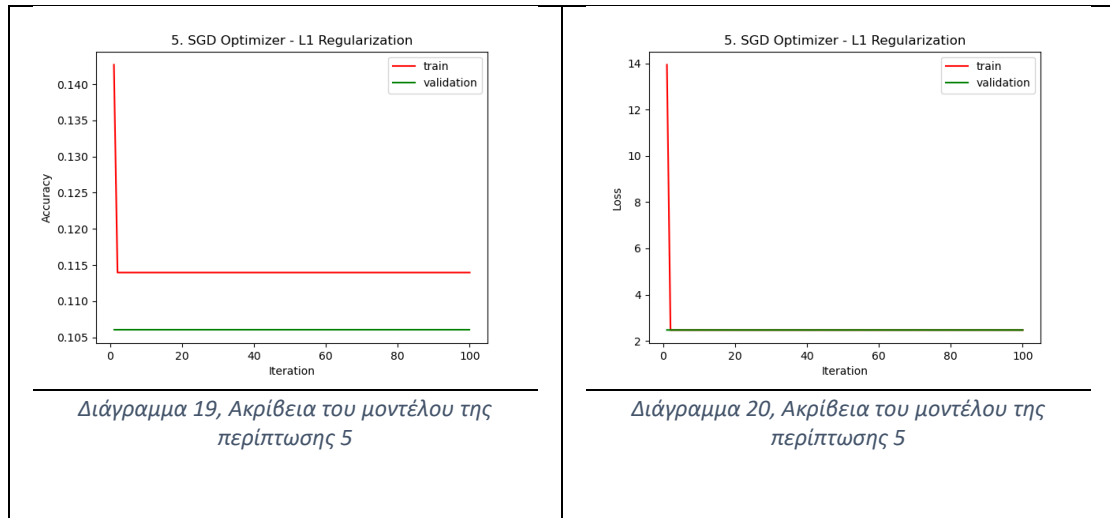
Διάγραμμα 16, Loss του μοντέλου της περίπτωσης 4 με  $\alpha = 0.01$



Διάγραμμα 17, Ακρίβεια του μοντέλου της περίπτωσης 4 με  $\alpha = 0.001$



Διάγραμμα 18, Loss του μοντέλου της περίπτωσης 4 με  $\alpha = 0.001$



### Παρατηρήσεις / Συμπεράσματα:

Τέλος, παρατηρείται πως υπάρχει overfitting στα μοντέλα των περιπτώσεων 1 με batch size = 1 (online) και (λιγότερο) με batch size = 256, εφόσον το μοντέλο ενημερώνεται (υπερβολικά) συχνά και χάνει την εποπτικότητα του επάνω στο σετ εκπαίδευσης και γίνεται ευάλωτο στο θόρυβο. Ενδείξεις overfitting έχουμε επίσης και για τα μοντέλα της περίπτωσης 2 (με βάση το loss), περισσότερο μάλιστα όταν  $\rho = 0.01$ . Τέλος, έχουμε σημαντικό underfitting για τα μοντέλα των 3-5, όπως ήδη αναφέρθηκε παραπάνω.

## 2. Fine tuning δικτύου

Εφαρμόζοντας και πάλι αναζήτηση πλέγματος επάνω στο χώρο των παραμέτρων, έχουμε ότι το καλύτερο μοντέλο έχει F-Measure = 0.182 με παραμέτρους:

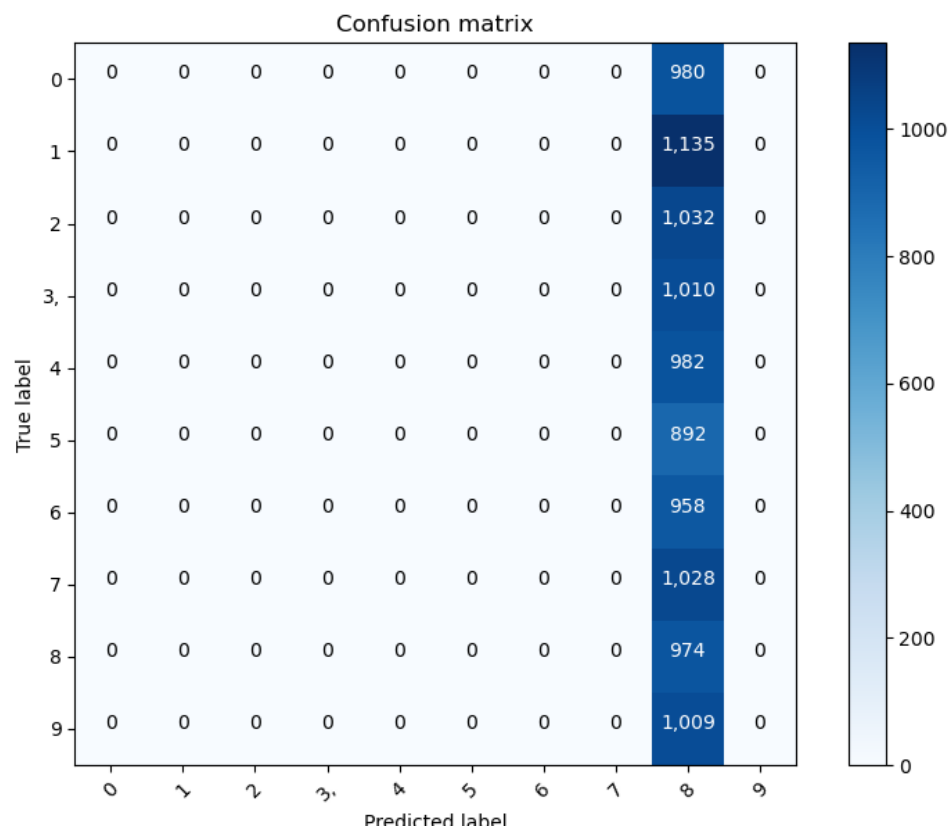
- 64 νευρώνες στο 1<sup>ο</sup> κρυφό στρώμα
- 512 νευρώνες στο 2<sup>ο</sup> κρυφό στρώμα
- 0.000001 παράμετρο κανονικοποίησης
- 0.01 ρυθμό εκμάθησης

Η απόδοση του επάνω στο σετ ελέγχου περιγράφεται ως εξής:

| Παράμετροι/Μετρικές    | Accuracy | Recall | Precision | F-Measure |
|------------------------|----------|--------|-----------|-----------|
| {64, 512, 1e-06, 0.01} | 0.097    | 0.1    | 1         | 0.182     |

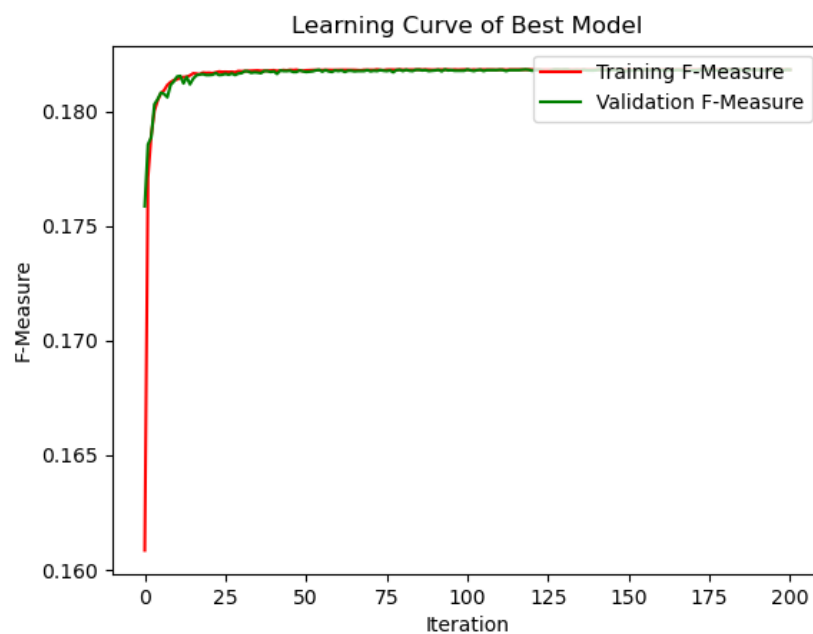
Πίνακας 2, Μετρικές του μοντέλου με τις καλύτερες παραμέτρους

Ο πίνακας σύγχυσης του μοντέλου (ώστε να φανεί εποπτικά η απόδοση του και να αιτιολογηθεί γιατί έχουμε άριστο precision):



Πίνακας 3, Πίνακας σύγχυσης του μοντέλου με τις καλύτερες παραμέτρους

Και τέλος, οι καμπύλες εκμάθησης του μοντέλου:



Εικόνα 4, Καμπύλες εκμάθησης του μοντέλου

### Παρατηρήσεις / Συμπεράσματα:

Παρότι το μοντέλο έχει κακή απόδοση (ταξινομεί όλες τις εγγραφές σε μία κατηγορία), από τις καμπύλες εκμάθησης δεν φαίνεται να παρουσιάζει overfitting. Υπάρχει μονάχα μία υποψία ότι υπολογίζονται λάθος οι μετρικές απόδοσης λόγω κάποιας παράλειψης στον κώδικα, αλλά δυστυχώς πρόλαβα να το διερευνήσω περεταίρω.