

# ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ

Διδάσκοντες: Σ. Λυκοθανάσης, Δ. Κουτσομητρόπουλος  
Ακαδημαϊκό Έτος 2024-2025

## Εργαστηριακή Άσκηση Μέρος Α΄

**Στοιχεία Φοιτητή:**

**Ον/μο:** ΚΟΥΤΡΟΥΜΠΕΛΑΣ ΒΑΣΙΛΕΙΟΣ

**ΑΜ:** 1093397

**Εξάμηνο:** 8ο

**email:** [up1093397@ac.upatras.gr](mailto:up1093397@ac.upatras.gr)

**Code Repository**

<https://github.com/vasiliskoutroumpelas/computational-intelligence>

<b>A1. Προεπεξεργασία και Προετοιμασία δεδομένων.....</b>	<b>2</b>
α).....	2
β).....	2
<b>A2. Επιλογή αρχιτεκτονικής.....</b>	<b>3</b>
α).....	3
β).....	4
γ).....	4
δ).....	4
ε).....	5
στ).....	8
<b>A3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής.....</b>	<b>9</b>
<b>A4: Ομαλοποίηση.....</b>	<b>12</b>

## A1. Προεπεξεργασία και Προετοιμασία δεδομένων

α)

Για την φόρτωση και προεπεξεργασία των δεδομένων χρησιμοποιήθηκαν οι βιβλιοθήκες pandas και sklearn. Μέσω του pandas φορτώθηκε το dataset από το csv αρχείο, αφαιρέθηκαν οι αχρείαστες στήλες PatientID και DoctorInCharge και χωρίσαμε τις στήλες σε είσοδο και έξοδο (στήλη Diagnosis). Από τις αναφερόμενες τεχνικές χρησιμοποιήθηκε το One-hot encoding για τις στήλες Gender, Ethnicity και EducationLevel - μιας και παίρνουν ακέραιες τιμές οι οποίες δεν ταξινομούνται με σειρά (κατηγορικές τιμές). Επίσης, χρησιμοποιήθηκε η Τυποποίηση στις συνεχές τιμές για κανονικοποίηση και πιο γρήγορη και σταθερή σύγκλιση του μοντέλου. Το Κεντράρισμα δεν χρησιμοποιήθηκε αφού αφαιρούμε ήδη τον μέσο όρο μέσω της Τυποποίησης, ενώ η Κανονικοποίηση min-max δεν χρησιμοποιήθηκε καθώς δεν υπάρχει κάποια στήλη με εύρος για κανονικοποίηση, ενώ έχει χρησιμοποιηθεί ήδη και η Τυποποίηση.

β)

Για τον διαχωρισμό του dataset σε 5-fold CV χρησιμοποιείται η συνάρτηση StratifiedKFold η οποία κάνει ισορροπημένο, ως προς τον αριθμό των δειγμάτων κάθε κλάσης, διαχωρισμό. Το split γίνεται βάση του πίνακα X όπου είναι τα δεδομένα εισόδου της εκπαίδευσης και βάση του πίνακα Y ο οποίος περιέχει την αναμενόμενη έξοδο.

```
# Split the data into a balanced 5-Fold
kfold = StratifiedKFold(n_splits=5, shuffle=True)
...
for j, (train, test) in enumerate(kfold.split(X, Y)):
    ...
```

## A2. Επιλογή αρχιτεκτονικής

α)

Για το συγκεκριμένο πρόβλημα η εκπαίδευση είναι προτιμότερο να γίνει με **Cross-Entropy**. Η διασταυρούμενη εντροπία χρησιμοποιείται συνήθως για προβλήματα ταξινόμησης (classification) όπως αυτό. Όταν το μοντέλο προβλέπει πιθανότητες για κάθε κατηγορία, η Cross-Entropy μετρά την απόσταση μεταξύ των προβλέψεων του μοντέλου και των πραγματικών ετικετών. Η τιμή της είναι μικρότερη όταν το μοντέλο είναι πιο σίγουρο και πιο ακριβές στις προβλέψεις του.

Το **Μέσο Τετραγωνικό Σφάλμα** χρησιμοποιείται κυρίως σε προβλήματα παλινδρόμησης (regression), όπου το μοντέλο προσπαθεί να προβλέψει συνεχείς τιμές. Μετρά τη διαφορά μεταξύ της πραγματικής τιμής και της προβλεπόμενης τιμής του μοντέλου, και στη συνέχεια υπολογίζει το μέσο όρο των τετραγώνων αυτών των διαφορών. Όσο μικρότερη είναι η τιμή του MSE, τόσο καλύτερα είναι τα αποτελέσματα του μοντέλου.

Η **ακρίβεια** μετρά πόσες φορές το μοντέλο προβλέπει σωστά τις κατηγορίες. Είναι το ποσοστό των σωστών προβλέψεων σε σχέση με το σύνολο των προβλέψεων. Είναι μια απλή μετρική και χρησιμοποιείται για προβλήματα ταξινόμησης.

β)

Για το επίπεδο εξόδου αρκεί ένας νευρώνας αφού για το συγκεκριμένο πρόβλημα γίνεται ταξινόμηση σε δύο κλάσεις και μπορεί να προβλέψει την πιθανότητα ένα δείγμα να ανήκει στην μια κλάση (ασθένεια) ή στην άλλη (μη-ασθένεια).

γ)

Ως συνάρτηση ενεργοποίησης για το κρυφό επίπεδο επιλέγεται η ReLU. Αυτό γιατί είναι απλή συνάρτηση και γρήγορη στον υπολογισμό που σημαίνει ότι το δίκτυο εκπαιδεύεται πιο γρήγορα. Αποφεύγει το πρόβλημα των “vanishing gradients” που σημαίνει ότι η εκπαίδευση του δικτύου δεν επιβραδύνεται από τις μικρές τιμές παραγώγων όπως την tanh, ενώ το πρόβλημα των Dying ReLU δεν μας επηρεάζει ιδιαίτερα αφού το δίκτυο δεν είναι αρκετά βαθύ ώστε να προκληθούν προβλήματα σε μετέπειτα επίπεδα. Η επιλογή της συνάρτησης ενεργοποίησης SiLU δεν έγινε καθώς είναι αρκετά πολύπλοκη για το πρόβλημα μας και προτιμάται σε βαθιά νευρωνικά δίκτυα.

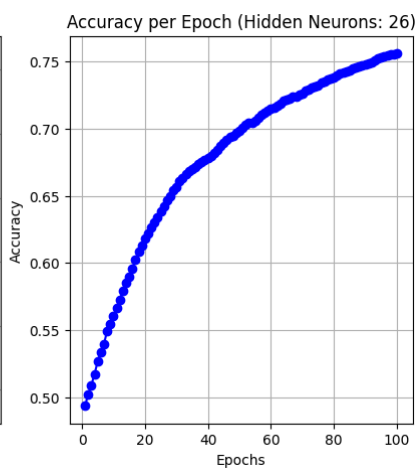
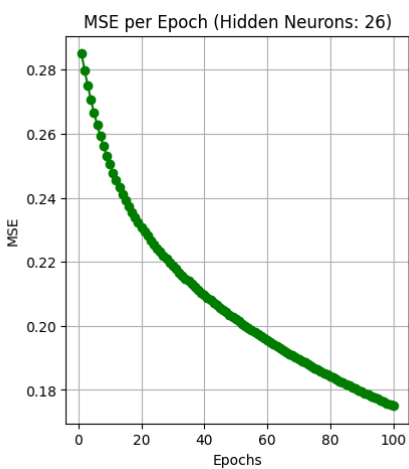
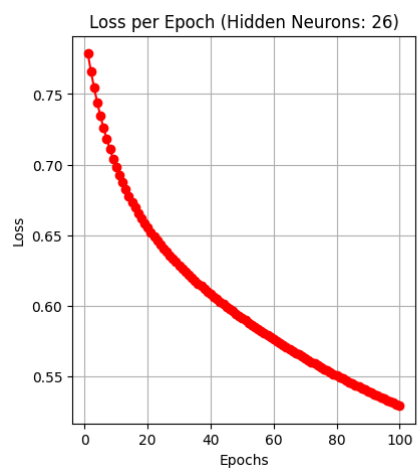
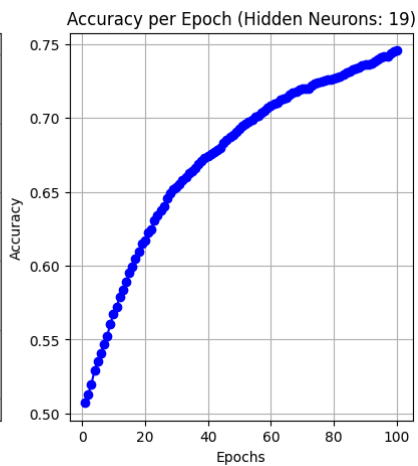
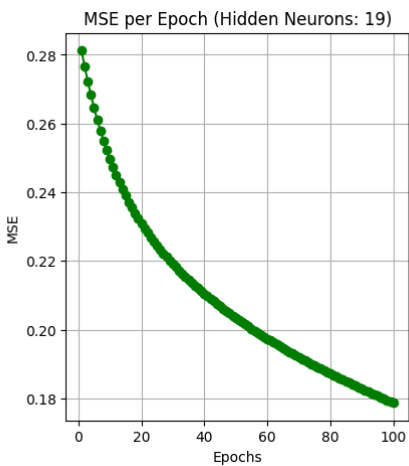
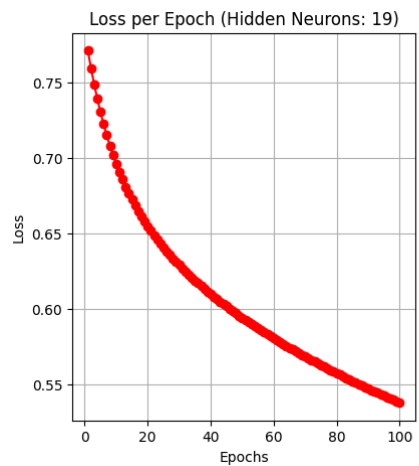
δ)

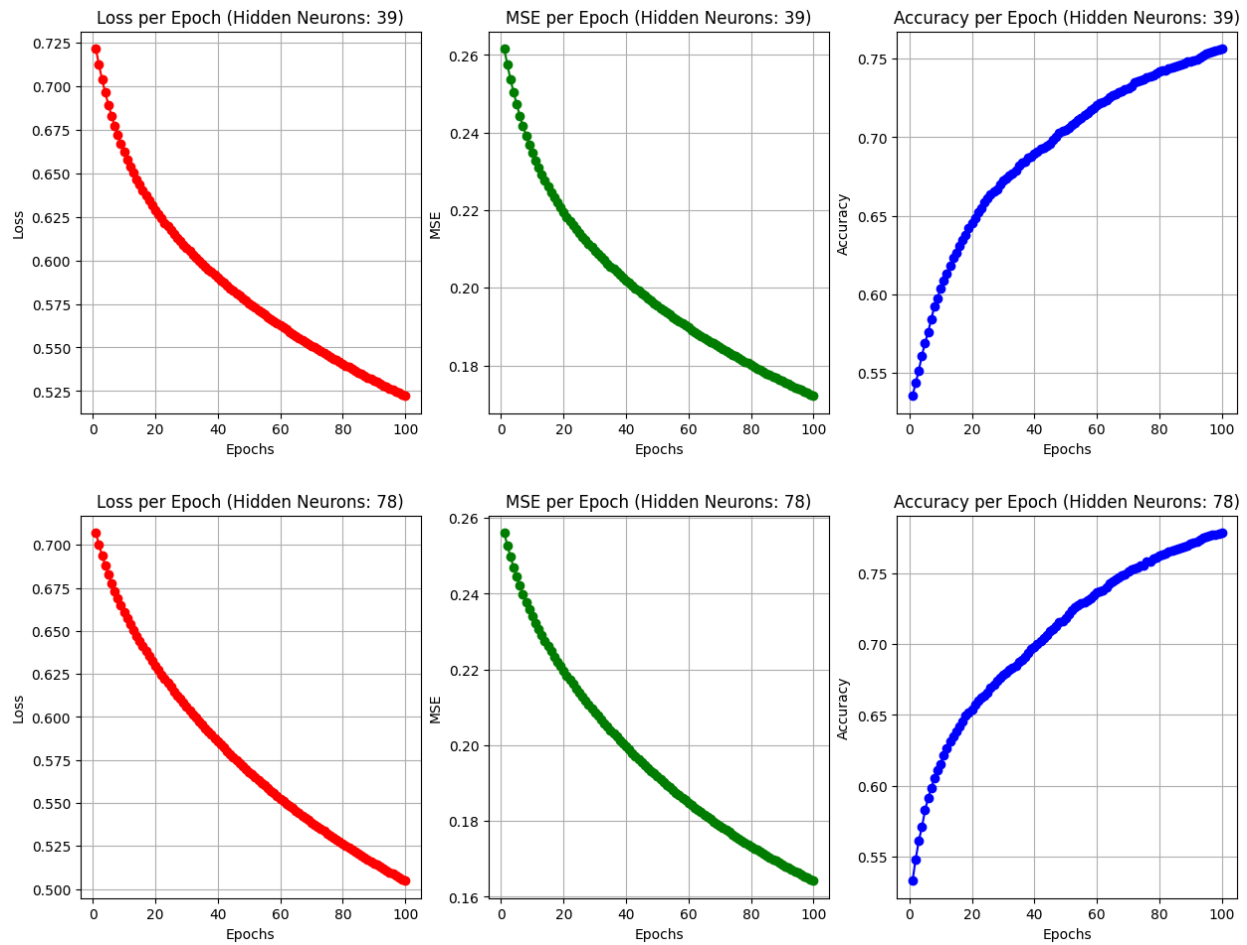
Για το επίπεδο εξόδου κατάλληλη συνάρτηση ενεργοποίησης είναι η σιγμοειδή γιατί ξεχωρίζει βάση των δεδομένων του κρυφού επιπέδου μεταξύ δύο κλάσεων (binary classification) με πιθανότητες και λειτουργεί σωστά με την loss function Binary Cross-Entropy. Η γραμμική δεν είναι κατάλληλη γιατί δεν δίνει πιθανότητες, υπολογίζει απλά το άθροισμα των εξόδων του κρυφού επιπέδου και το αποτέλεσμα μπορεί να είναι οποιοσδήποτε αριθμός. Η Softmax δεν είναι αναγκαία γιατί έχει σχεδιαστεί για classification πολλών κλάσεων και στο παρόν πρόβλημα αρκεί το binary classification.

ε)

Για το παρόν ερώτημα χρησιμοποιήθηκε batch size 50 και 100 εποχές εκπαίδευσης.

Αριθμός νευρώνων στο κρυφό επίπεδο	CE loss	MSE	Acc
$H = I/2$	0.5456	0.1825	0.7413
$H = 2I/3$	0.5304	0.1752	0.7538
$H = I$	0.5326	0.1759	0.7562
$H = 2I$	0.5164	0.1689	0.7739





(i)

Σχετικά με τον αριθμό των κρυφών κόμβων παρατηρούμε πως με την αύξηση τους αυξάνεται το accuracy και μειώνονται το μέσο τετραγωνικό σφάλμα και το CE loss.

(ii)

Σχετικά με την συνάρτηση κόστους, χρησιμοποιήθηκε η Binary Cross-Entropy, κατάλληλη για δυαδική ταξινόμηση. Άλλες συναρτήσεις όπως MSE δεν ήταν κατάλληλες καθώς δεν εκμεταλλεύονται πλήρως τη λογική της πιθανοκρατικής εξόδου.

(iii)

Ως συνάρτηση ενεργοποίησης χρησιμοποιήθηκε η ReLU, που αποδείχθηκε αποδοτική. Η ReLU επιταχύνει τη σύγκλιση και αποφεύγει προβλήματα όπως vanishing gradients που συναντάμε με Tanh ή Sigmoid.

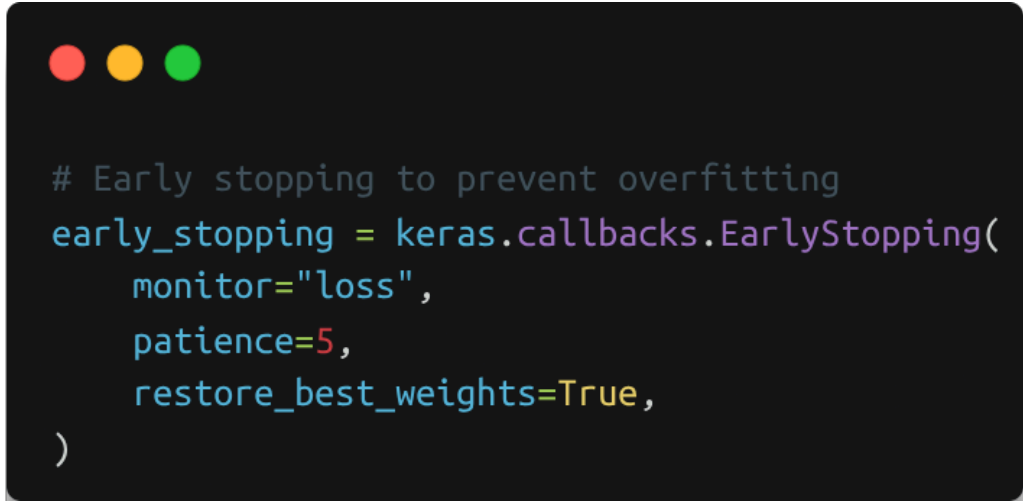
(iv)

Από τα διαγράμματα φαίνεται μια μικρή βελτίωση στην ταχύτητα σύγκλισης ως προς τις εποχές με την αύξηση του αριθμού κρυφών νευρώνων. Με περισσότερους κόμβους, το δίκτυο μαθαίνει γρηγορότερα, αλλά με κόστος πιθανής υπερεκπαίδευσης.



στ)

Ως κριτήριο τερματισμού εκπαίδευσης για κάθε fold έχει οριστεί το Early Stopping με τις παρακάτω παραμέτρους.



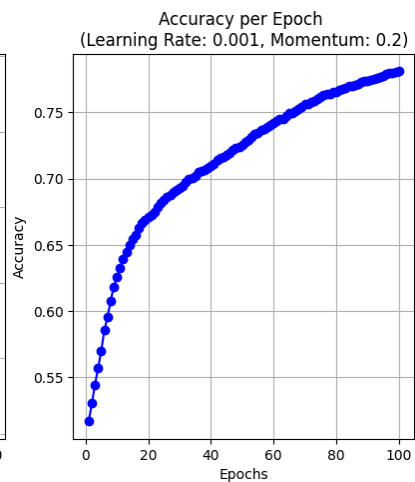
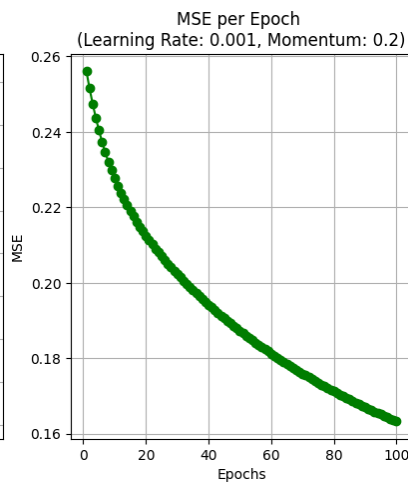
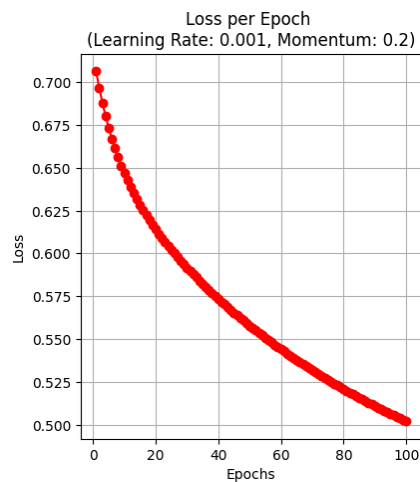
```
# Early stopping to prevent overfitting
early_stopping = keras.callbacks.EarlyStopping(
    monitor="loss",
    patience=5,
    restore_best_weights=True,
)
```

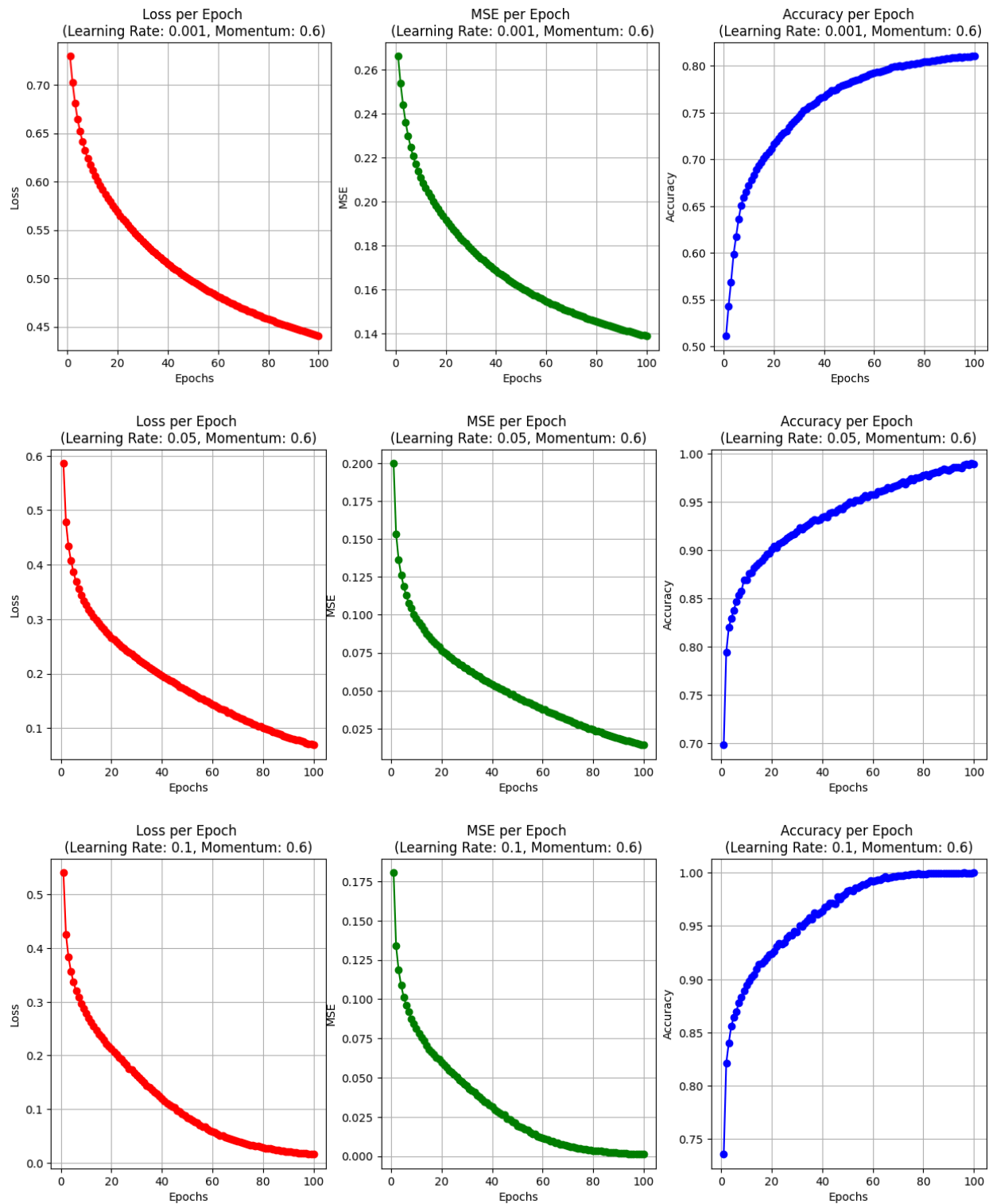
Στην παράμετρο monitor, με το loss, το μοντέλο παρακολουθεί το loss (το σφάλμα στο training set). Αν το loss σταματήσει να βελτιώνεται, τότε αρχίζει να σκέφτεται να σταματήσει. Με την παράμετρο patience=5, επιτρέπει 5 διαδοχικές εποχές (epochs) χωρίς βελτίωση πριν σταματήσει τελείως την εκπαίδευση. Τέλος, με την παράμετρο restore\_best\_weights=True, όταν σταματήσει η εκπαίδευση, φορτώνει αυτόματα τα βάρη από την καλύτερη εποχή (εκεί που το loss ήταν το χαμηλότερο).

### Α3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

Για το παρόν πρόβλημα χρησιμοποιήθηκε batch size 50 και 100 εποχές εκπαίδευσης, αριθμός νευρώνων ίσο με 21 μιας και έδωσε τα καλύτερα αποτελέσματα προηγούμενως.

$\eta$	m	CE loss	MSE	Acc
0.001	0.2	0.5101	0.1670	0.7902
0.001	0.6	0.4670	0.1493	0.7957
0.05	0.6	0.5451	0.1329	0.8315
0.1	0.6	0.8189	0.1548	0.8159





Το momentum πρέπει να είναι μικρότερο από 1 ώστε να επιταχύνει τη σύγκλιση χωρίς να προκαλεί αστάθεια. Τιμές  $\geq 1$  ενισχύουν υπερβολικά τις αλλαγές στα βάρη, με αποτέλεσμα πιθανή εκτροπή από τη βέλτιστη λύση.

Η αύξηση του momentum ( $m$ ) από 0.2 σε 0.6 με σταθερό learning rate ( $\eta = 0.001$ ) βελτιώνει όλα τα metrics — χαμηλότερο CE loss & MSE και ελαφρώς υψηλότερη ακρίβεια.

Η αύξηση του learning rate σε 0.05 (με  $m = 0.6$ ) οδηγεί σε καλύτερη απόδοση (υψηλότερη ακρίβεια 0.8315) και χαμηλότερο MSE, αλλά στο CE loss υπάρχει μια μικρή αύξηση.

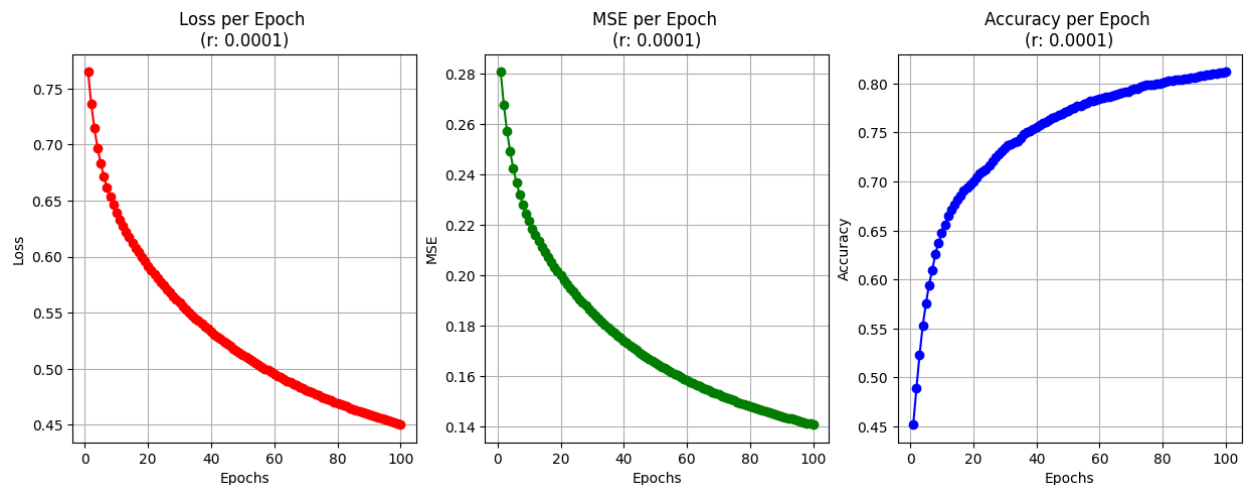
Πολύ μεγάλο learning rate ( $\eta = 0.1$ ) υποβαθμίζει το CE loss σημαντικά (0.8189), δείχνοντας ότι το μοντέλο δυσκολεύεται να συγκλίνει σωστά, αν και η ακρίβεια παραμένει σχετικά υψηλή.

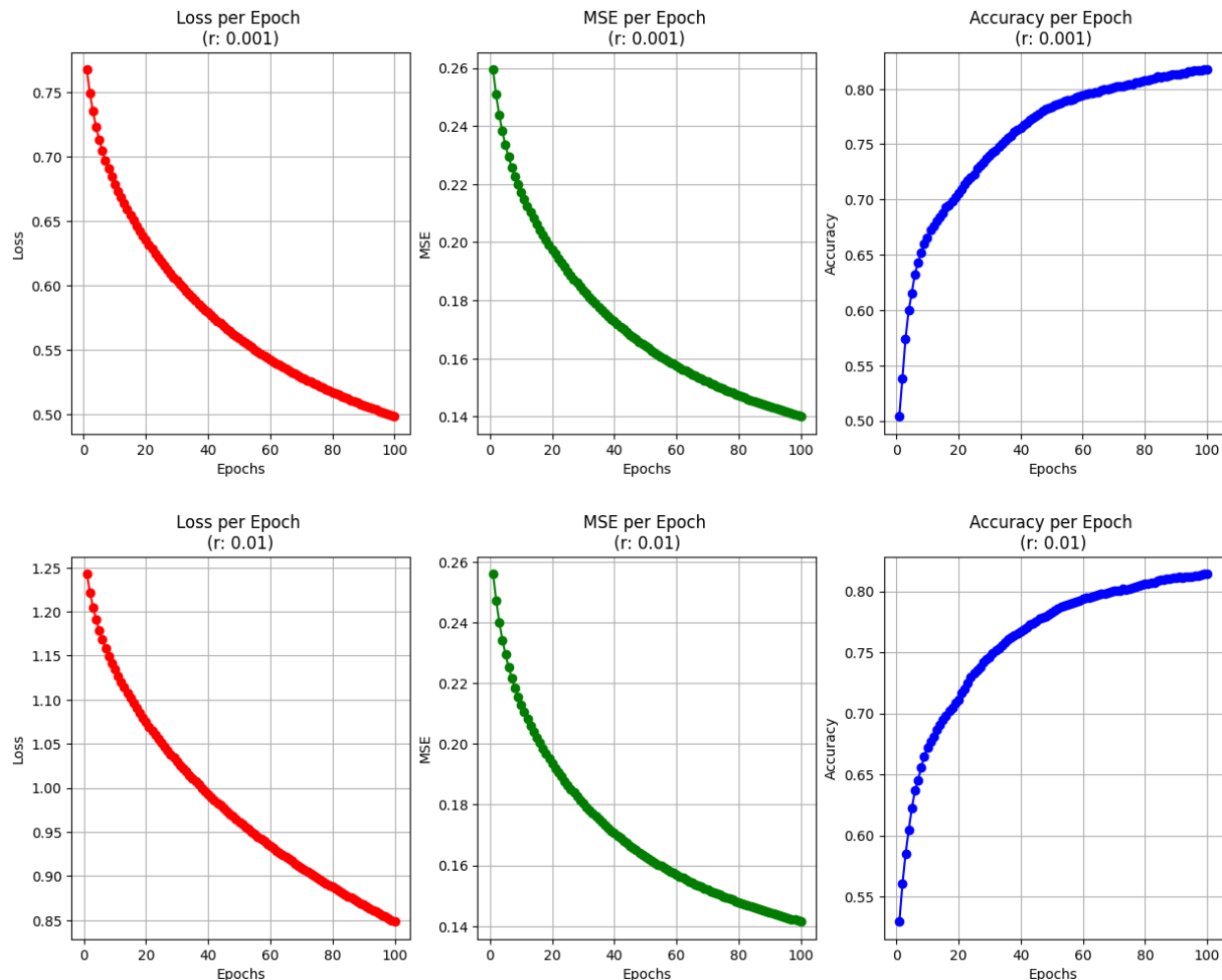
## A4: Ομαλοποίηση

Για το παρόν πρόβλημα χρησιμοποιήθηκε batch size 50 και 100 εποχές εκπαίδευσης, αριθμός νευρώνων ίσο με 2I,  $\eta = 0.001$  και  $m = 0.6$  μιας και έδωσαν τα καλύτερα αποτελέσματα προηγουμένως.

Επιλέγουμε την κανονικοποίηση L2 γιατί είναι προτιμότερη σε προβλήματα δυαδικής ταξινόμησης καθώς περιορίζει την υπερπροσαρμογή χωρίς να μηδενίζει τα βάρη, διατηρώντας έτσι τη συνεισφορά όλων των εισόδων. Προσφέρει πιο ομαλή μάθηση και σταθερότερη σύγκλιση, σε αντίθεση με την L1, η οποία προκαλεί αραιά δίκτυα και είναι λιγότερο κατάλληλη όταν όλα τα χαρακτηριστικά είναι δυνητικά σημαντικά.

Συντελεστής r	CE loss	MSE	Acc
0.0001	0.4693	0.1479	0.7948
0.001	0.5192	0.1483	0.7915
0.01	0.8753	0.1537	0.7976





Η εφαρμογή της L2 regularization στο δίκτυο έδειξε ότι η κατάλληλη επιλογή του συντελεστή  $r$  είναι κρίσιμη για τη γενικευτική ικανότητα του μοντέλου. Για πολύ μικρή τιμή του  $r$  (0.0001), παρατηρήθηκε η καλύτερη συνολική απόδοση, με χαμηλό CE loss και MSE και ικανοποιητική ακρίβεια. Αντίθετα, για μεγαλύτερες τιμές του  $r$  (0.001 και 0.01), το CE loss αυξήθηκε, γεγονός που υποδηλώνει ότι η ισχυρή ομαλοποίηση εμπόδισε το δίκτυο να μάθει σωστά τα δεδομένα, οδηγώντας σε υποεκπαίδευση. Συμπερασματικά, η χρήση L2 με πολύ μικρό συντελεστή βελτιώνει τη γενίκευση, ενώ υπερβολικά μεγάλη τιμή του  $r$  μειώνει την αποτελεσματικότητα της εκπαίδευσης.