**ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ**

*Διδάσκων: Δ. Κουτσομητρόπουλος*

*Ακαδημαϊκό Έτος 2021-2022*

***Εργαστηριακή Άσκηση***

***Μέρος Α'***

Φοιτητής: Τσιαμήτρος Κωνσταντίνος ΑΜ 235913

# Α. Αναγνώριση κειμένου με Χρήση Νευρωνικών Δικτύων

Το αντικείμενο της άσκησης είναι η ανάθεση μια ή περισσότερων ετικετών σε κάποια δεδομένα εισόδου (κείμενα εγγράφων) – δηλαδή multilabel multiclass classification με multilayered perceptron (MLP).

## Α1. Προεπεξεργασία και Προετοιμασία δεδομένων

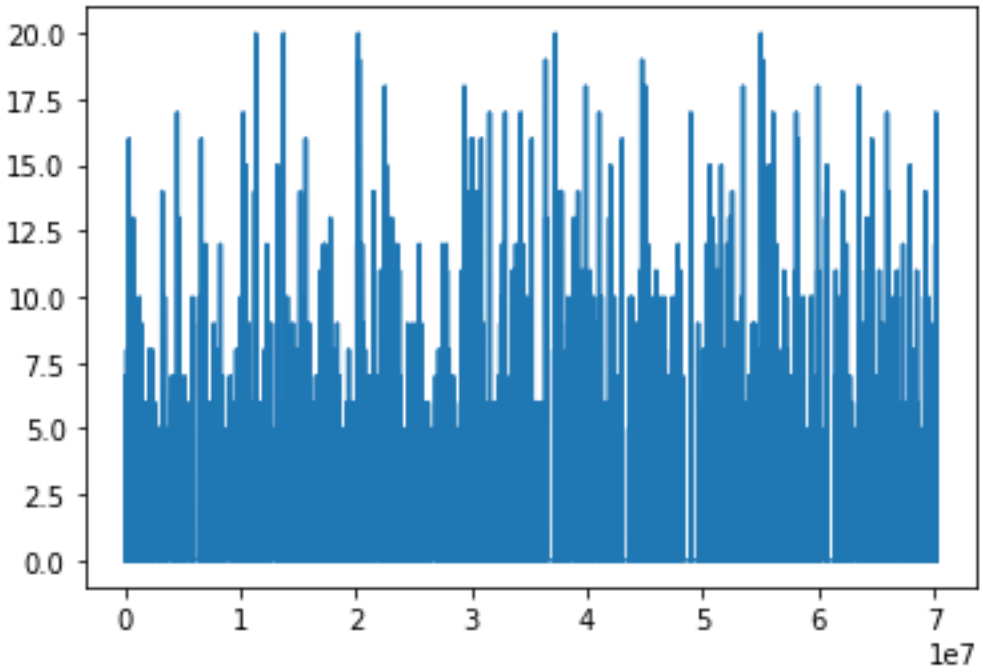
a) Τα δεδομένα εισόδου κωδικοποιήθηκαν σύμφωνα με το μοντέλο Bag of words. (αρχείο Bag_of_words.py).

   Αρχικά, γίνεται ανάγνωση του dataset με μια κλήση στη συνάρτηση read_file() (αρχείο load_data.py). Η συνάρτηση αυτή επιστρέφει τους όρους ανά πρόταση (η πληροφορία αυτή αποθηκεύεται στη μεταβλητή words), το πλήθος των προτάσεων ανά κείμενο (αποθηκεύεται στη μεταβλητή sent_num) και το πλήθος των λέξεων ανά πρόταση - ανά κείμενο (αποθηκεύεται στην μεταβλητή word_num).

   Έπειτα, γίνεται μια κλήση στη συνάρτηση count_freqs(), η οποία επιστρέφει τις συχνότητες εμφάνισης των όρων ανά κείμενο.

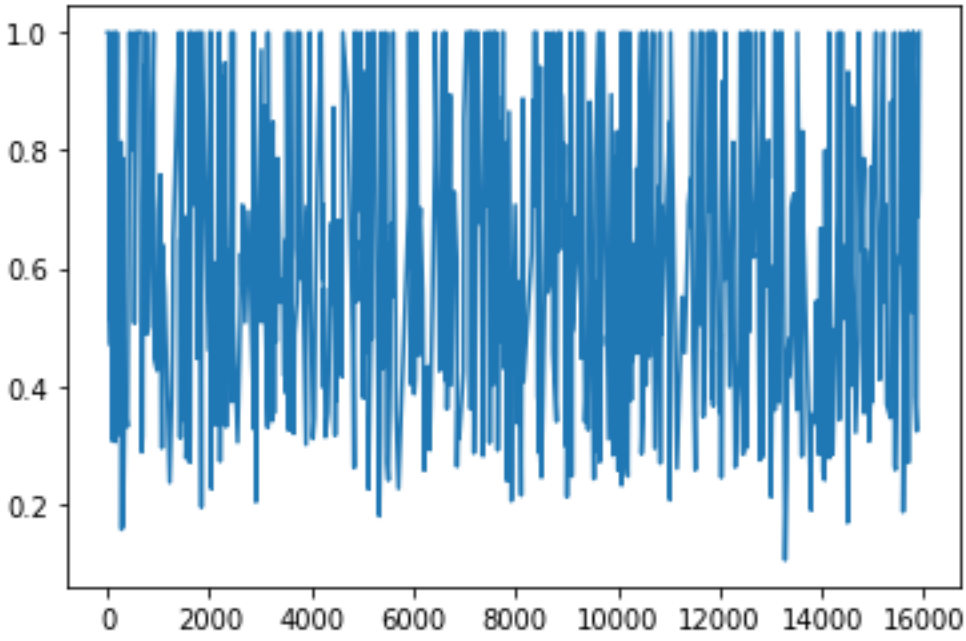
   Τέλος, αρχικοποιείται ένα sparse COO μητρώο για ελαχιστοποίηση του δεσμευόμενου χώρου καθώς και του χρόνο προσπέλασης των δεδομένων.

b) Για την προεπεξεργασία του dataset έχουν εξεταστεί τα παρακάτω ενδεχόμενα:
   a. **Κεντράρισμα (Centering):**
      Αφαιρούμε το μέσο όρο από κάθε δείγμα.

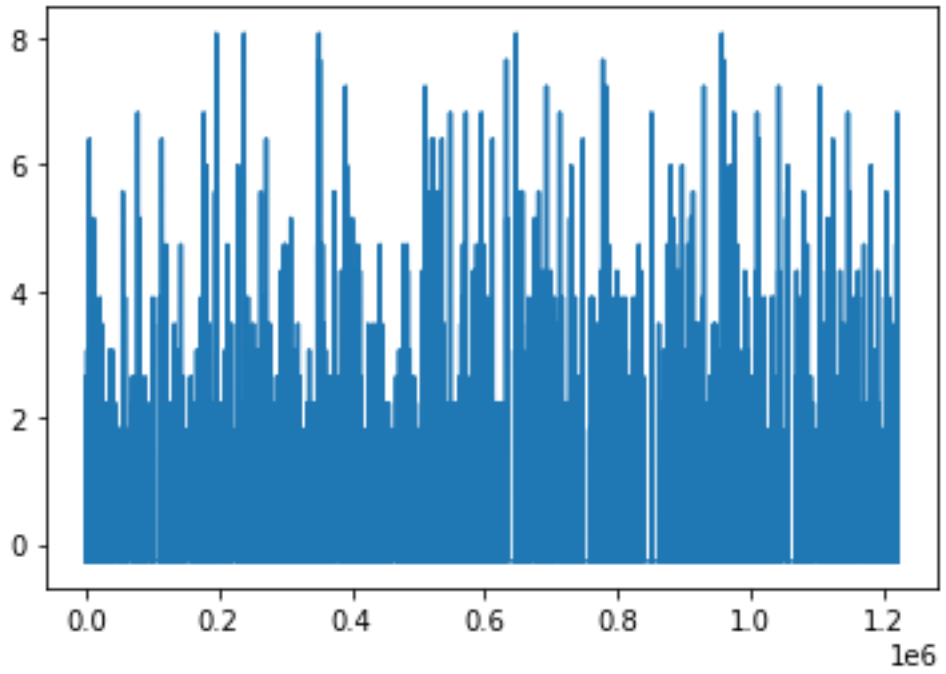



**b. Κανονικοποίηση (Normalization):**

Μεταφέρουμε το εύρος τιμών στο εύρος [0,1].



c. **Τυποποίηση (Standardization):**
Παρέχουμε στο dataset μέση μηδενική τιμή και μοναδιαία διακύμανση.




Δεδομένου ότι εξετάζουμε την εφαρμογή του dataset σε ένα MLP νευρωνικό δίκτυο, έκρινα σκόπιμο να εφαρμόσω την τυποποίηση, λόγω των ιδιοτήτων που προσδίδει στο dataset.

c) Για όλα τα πειράματα χρησιμοποιήθηκε 5-fold cross validation (αυτό φαίνεται στον κώδικα σε κάποια σημεία, ενδεικτικά, στο αρχείο MLP_multilabel.py στις γραμμές 50-51)

## Α2. Επιλογή αρχιτεκτονικής

**a) Cross-entropy:**

$$CE = \frac{1}{n}\sum_{x}\sum_{i=1}^{k} p_i(x)\log\left(q_i(x)\right)$$

Η μετρική αυτή, υποδεικνύει ομοιότητα για μικρές τιμές. Στο συγκεκριμένο πρόβλημα, δείχνει πόσο όμοια είναι η πρόβλεψη του νευρωνικού δικτύου, με την γνωστή ετικέτα, για κάθε δείγμα.

**MSE (Mean Squared Error):**

$$MSE = \frac{1}{2n}\sum_{x}\sum_{i=1}^{k} (p_i(x) - q_i(x))^2$$

Το ελάχιστο τετραγωνικό σφάλμα αποτελεί ένα μέτρο για την ποιότητα του ταξινομητή. (Anon., n.d.). Όσο πιο κοντά στο μηδέν πλησιάζει η μετρική αυτή, τόσο πιο ακριβής είναι ο ταξινομητής.

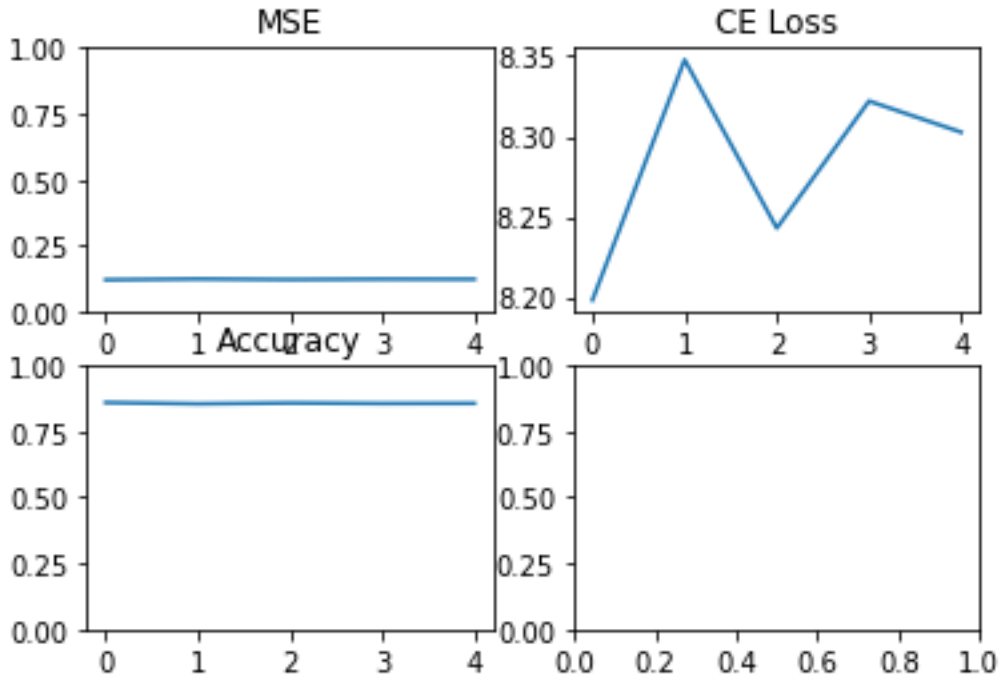
**Accuracy (Ακρίβεια):**
Η ποσότητα αυτή κυμαίνεται στο [0,1] και υποδηλώνει το ποσοστό των σωστών αποκρίσεων του ταξινομητή επί του dataset (δηλαδή αντιπροσωπεύει το ποσοστό των σωστών ταξινομήσεων των δειγμάτων)

b) Για το επίπεδο εξόδου θα χρειαστούν 20 νευρώνες, δεδομένου ότι υπάρχουν 20 πιθανά labels τα οποία μπορεί να ανατεθούν σε κάθε δείγμα.

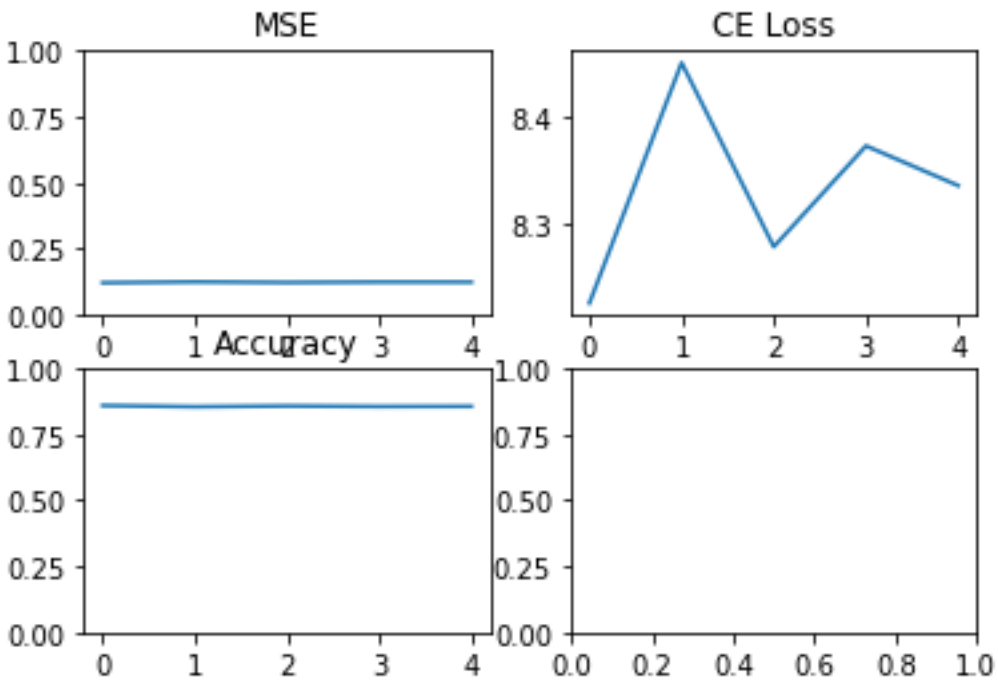
c) Για το κρυφό επίπεδο επιλέχθηκε η συνάρτηση ενεργοποίηση leaky ReLU για να αποφευχθεί το φαινόμενο νεκρών νευρώνων (σε σύγκριση με την απλή ReLU).

d) Για το επίπεδο εξόδου επιλέχθηκαν 20 διαφορετικές sigmoid συναρτήσεις ενεργοποίησης, δεδομένου ότι έχουμε πρόβλημα multilabel multiclass classification (η sigmoid παράγει μια τιμή στο [0,1], η οποία υποδηλώνει την πιθανότητα να ανήκει το δείγμα στη συγκεκριμένη ετικέτα)

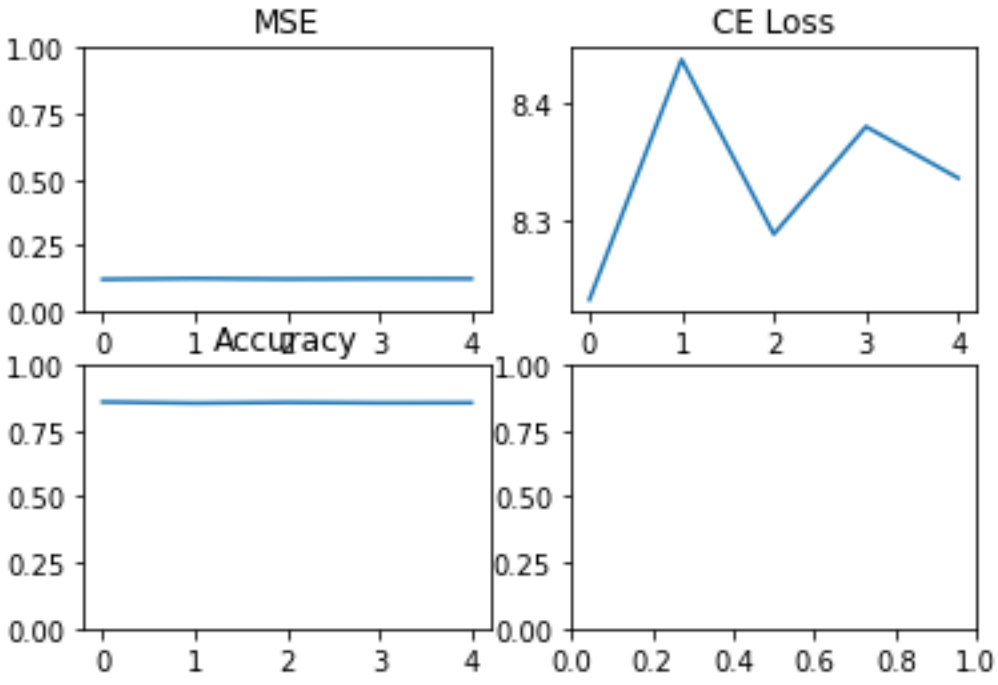
e) Οι γραφικές παραστάσεις σύγκλισης ανά κύκλο εκπαίδευσης (Μ.Ο.):




*Εικόνα 1 - 1 hidden layer με μέγεθος Ο*




*Εικόνα 2 - 1 hidden layer με μέγεθος (I + O)/2*




*Εικόνα 3 - 1 hidden layer με μέγεθος I+O*

Για κάθε configuration έχει γίνει hyperparameter tuning με τη βοήθεια του keras tuner (αρχείο MLP_multilabel_hptuning.py – οι τιμές των παραμέτρων για κάθε configuration φαίνονται στα σχόλια του κώδικα)

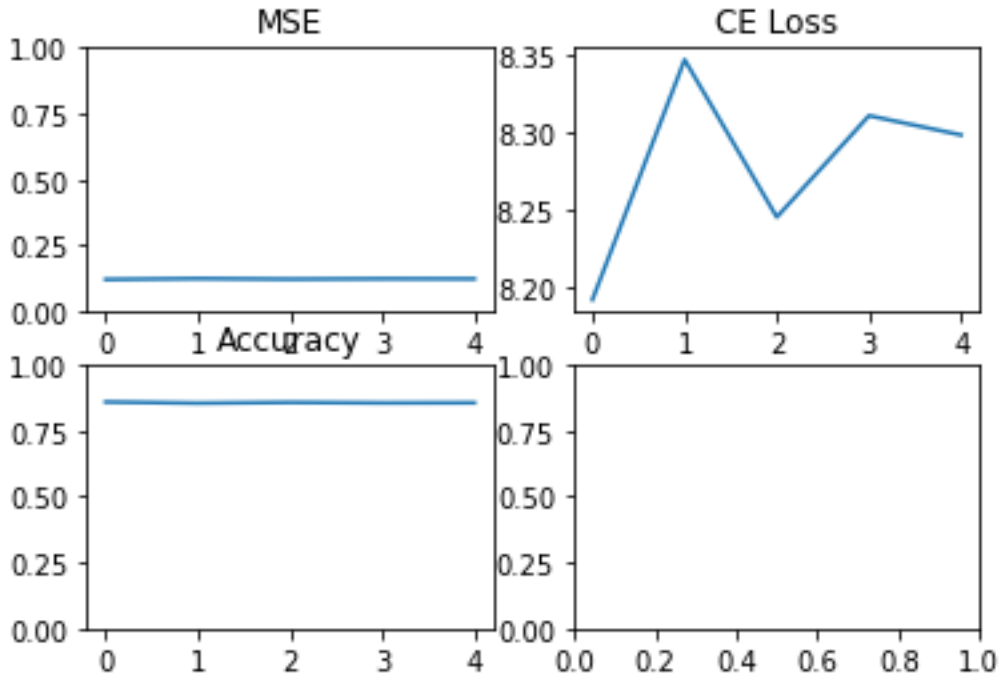
| Αριθμός νευρώνων στο κρυφό επίπεδο | CE loss | MSE | Accuracy |
|---|---|---|---|
| H1 = O | ~8.34 | ~0.20 | ~0.85 |
| H1 = (I + O)/2 | ~8.35 | ~0.20 | ~0.85 |
| H1 = I + O | ~8.35 | ~0.20 | ~0.85 |

i) Βλέπουμε ότι η καλύτερη απόκριση (και η πιο γρήγορη από άποψη χρόνου εκτέλεσης) για 1 hidden layer είναι με μέγεθος O (όσο και οι έξοδοι του MLP).

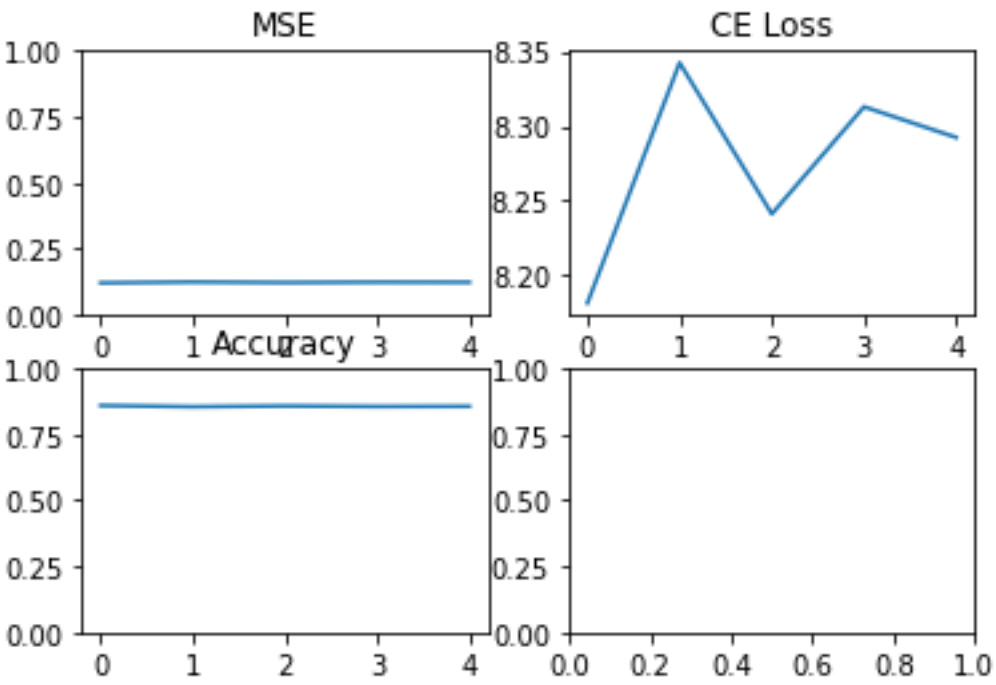
ii) Ως συνάρτηση κόστους επιλέχθηκε η binary crossentropy διότι μας ενδιαφέρει η ομοιότητα της εξόδου με τα Label (και Binary γιατί έχουμε binary classification – ή ανήκει μια ετικέτα σε ένα δείγμα ή δεν ανήκει σε αυτό).

iii) Την γρηγορότερη ταχύτητα σύγκλισης ως προς τις εποχές εκπαίδευσης την είχε το μικρότερο μέγεθος δικτύου, και ο χρόνος απόκρισης αυξάνεται πολύ γρήγορα σε συνάρτηση με το μέγεθος του hidden layer.

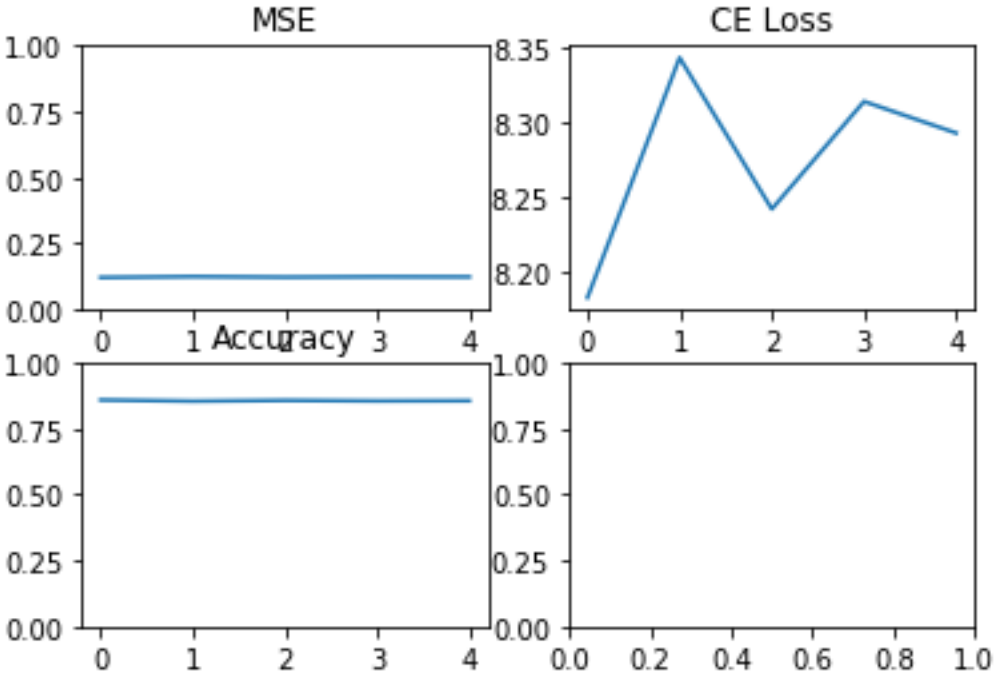
f) Οι γραφικές παραστάσεις σύγκλισης ανά κύκλο εκπαίδευσης (Μ.Ο.):




*Εικόνα 4 - H1: O, H2: O*




*Εικόνα 5 - H1: O, H2: (I + O)/4*




*Εικόνα 6 - H1: O, H2: I + O*

Για κάθε configuration έχει γίνει hyperparameter tuning manually (αρχείο MLP_multilabel_manual_hptuning.py – οι τιμές των παραμέτρων για κάθε configuration φαίνονται στα σχόλια του κώδικα)

Ακολουθώντας το παράδειγμα του προηγούμενου ερωτήματος σχετικά με την επιλογή του μεγέθους του δεύτερου hidden layer, με την ίδια λογική επιλέχθηκαν τα εξής μεγέθη:

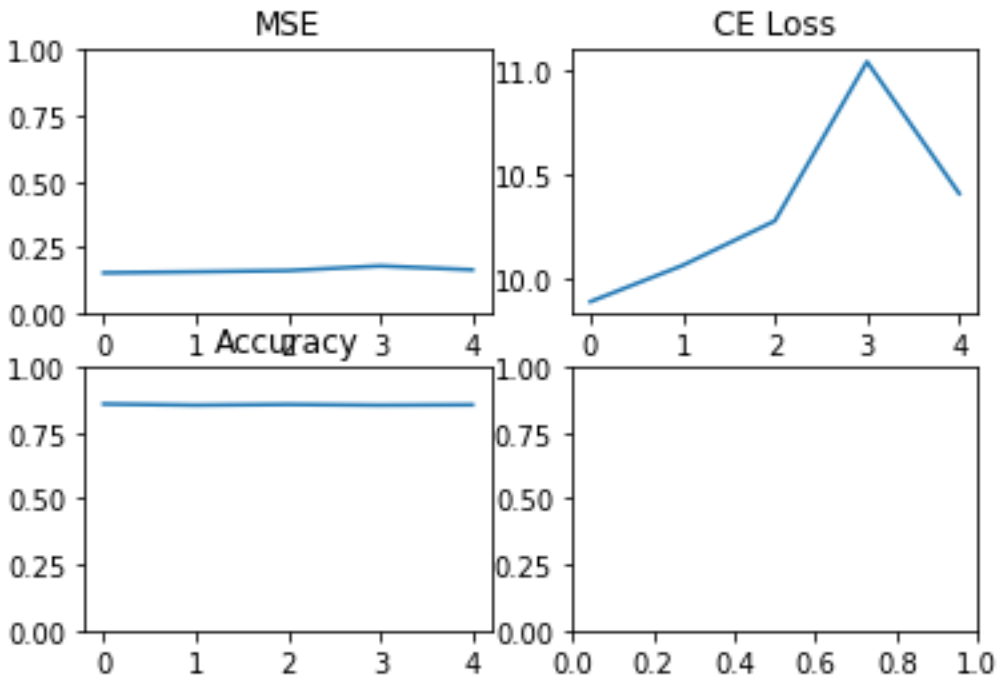
| Αριθμός νευρώνων στο κρυφό επίπεδο | CE loss | MSE | Accuracy |
|---|---|---|---|
| **H2 = O** | ~8.3 | ~0.20 | ~0.85 |
| **H2 = (I + O)/4** | ~8.3 | ~0.20 | ~0.85 |
| **H2 = (I + O)/2** | ~8.3 | ~0.20 | ~0.85 |

Η αποδόσεις ήταν παραπλήσιες (μάλλον λόγω του hp tuning και της ευκρίνειας των διαγραμμάτων). Με την προσθήκη hidden layers, το MLP «μαθαίνει» τα features του dataset (feature extraction). Τα layers για feature extraction καλό είναι να έχουν μέγεθος ~100 (σύμφωνα με πηγές), διότι όπως φαίνεται και από τα πειραματικά δεδομένα, για πιο μεγάλο μέγεθος δεν υπάρχει ουσιαστικό κέρδος στην ακρίβεια του MLP, και ταυτόχρονα ο χρόνος εκπαίδευσης αυξάνεται – και τα δύο ενδεχόμενα δεν είναι επιθυμητά.

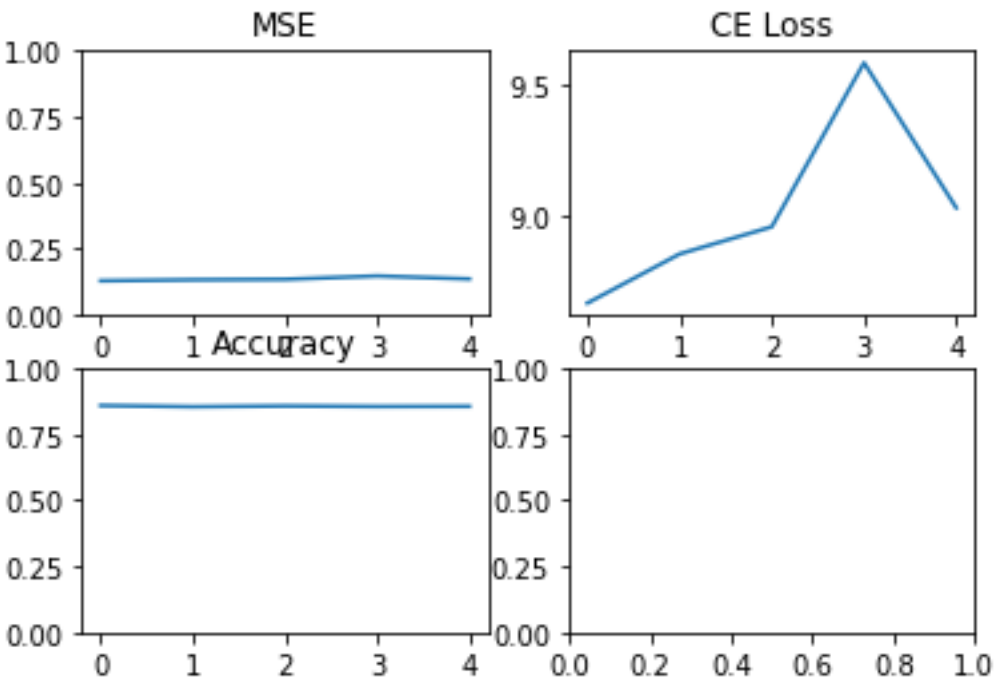
g) Για κριτήριο τερματισμού, επιλέχθηκε το όριο επαναλήψεων ανά δείγμα (epoch). Προκειμένου να αποφευχθεί το φαινόμενο του Overfitting μπορεί να χρησιμοποιηθεί η τεχνική του early stopping. Δεδομένου ότι η ακρίβεια δεν έπεφτε (συγκρίνοντας train-test data) δεν παρατηρήθηκε Overfitting, κι για αυτό δεν χρησιμοποιήθηκε η μέθοδος αυτή. (Anon., n.d.)

## A3. Μεταβολές στο ρυθμό εκπαίδευσης

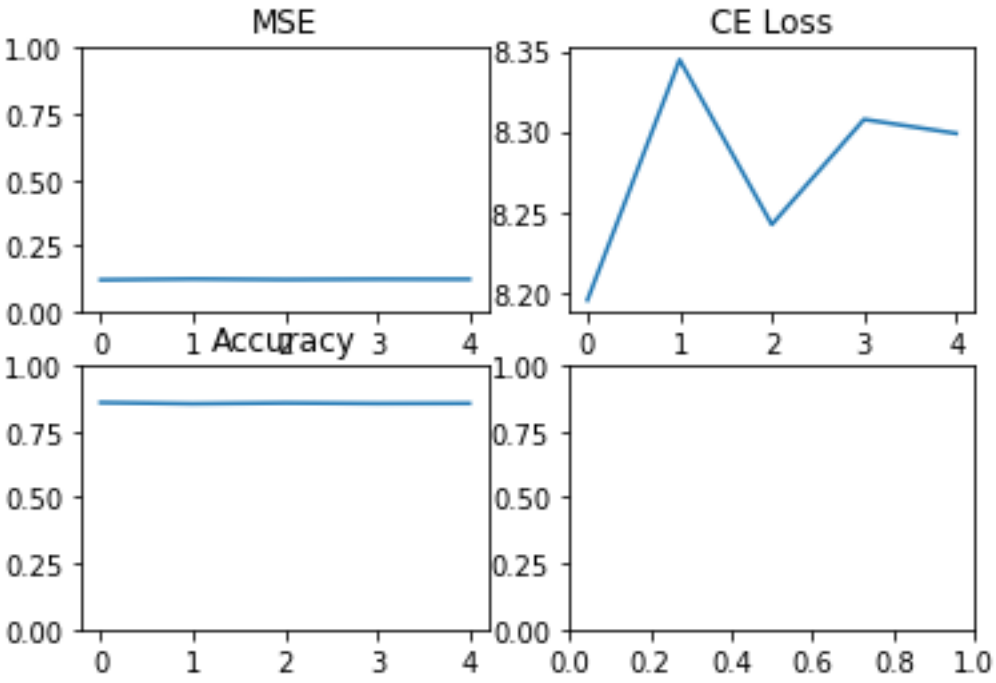
Οι γραφικές παραστάσεις σύγκλισης ανά κύκλο εκπαίδευσης (Μ.Ο.):



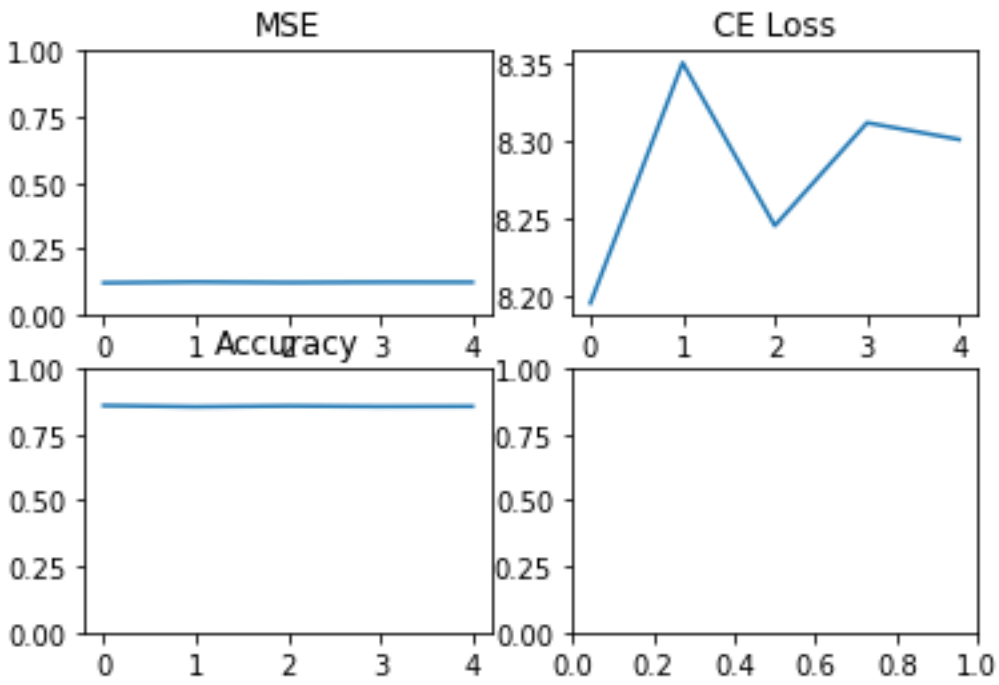

*Εικόνα 7 - H1: O, H2: O, n: 0.001, m: 0.2*




*Εικόνα 8 - H1:O, H2: O, n: 0.001, m: 0.6*




*Εικόνα 9 - H1: O, H2: O, n: 0.01, m: 0.6*




*Εικόνα 10 - H1: O, H2: O, n: 0.1, m: 0.6*

| n | m | CE loss | MSE | Accuracy |
|---|---|---|---|---|
| 0.001 | 0.2 | ~10.5 | ~0.21 | ~0.78 |
| 0.001 | 0.6 | ~9.3 | ~0.21 | ~0.8 |
| 0.05 | 0.6 | ~8.3 | ~0.20 | ~0.85 |
| 0.1 | 0.6 | ~8.3 | ~0.20 | ~0.85 |

Βλέπουμε ότι για τις μικρότερες τιμές του ρυθμού μάθησης και της ορμής το CE loss αυξάνεται λίγο (σε σχέση με τα προηγούμενα ερωτήματα, καθώς και με τις μεγαλύτερες τιμές των παραμέτρων αυτών), συνεπώς το μοντέλο έχει ελαφρώς χειρότερη απόδοση.

Από τη θεωρία ξέρω ότι ο γενικευμένος κανόνας δέλτα ορίζεται ως εξής (πανεπιστημιακές σημειώσεις εξίσωση (71) ):

$$\Delta w_{ji}(n) = -n \sum_{t=0}^{n} a^{n-1} \frac{\partial E(t)}{\partial w_{ji}(t)}$$

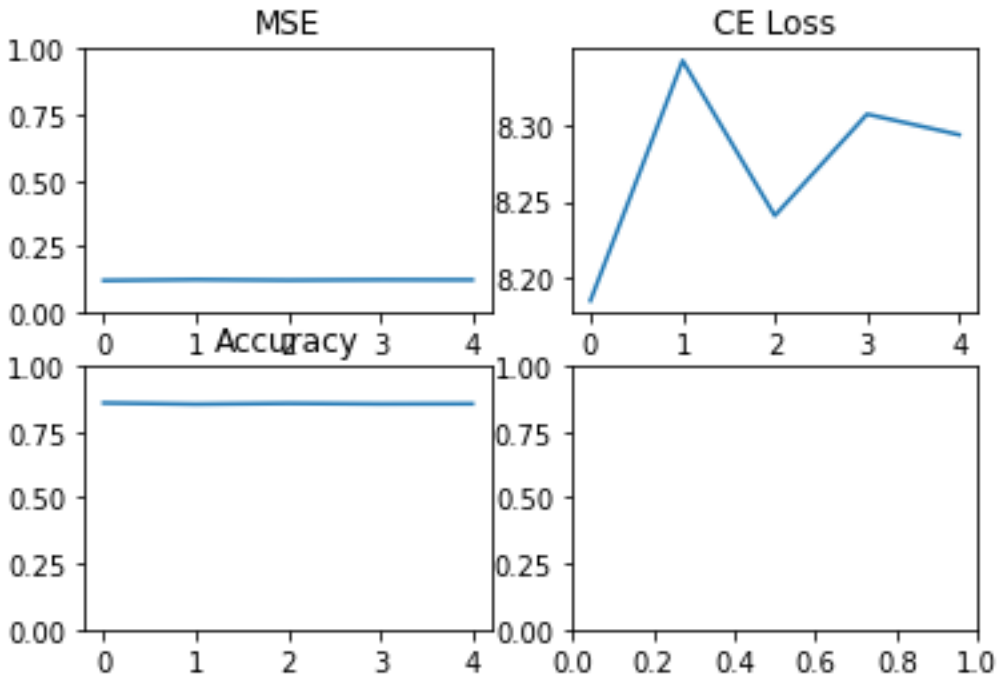
Όπου n: ρυθμός μάθησης και α: σταθερά ορμής

Από αυτή προκύπτει ότι ο παράγοντας διόρθωσης βάρους $\Delta w_{ji}(n)$ αποτελείται από άθροισμα εκθετικών όρων. Για να συγκλίνει αυτή η σειρά πρέπει να ισχύει $0 < |\alpha| < 1$. Εάν $\alpha = 0$ τότε ο

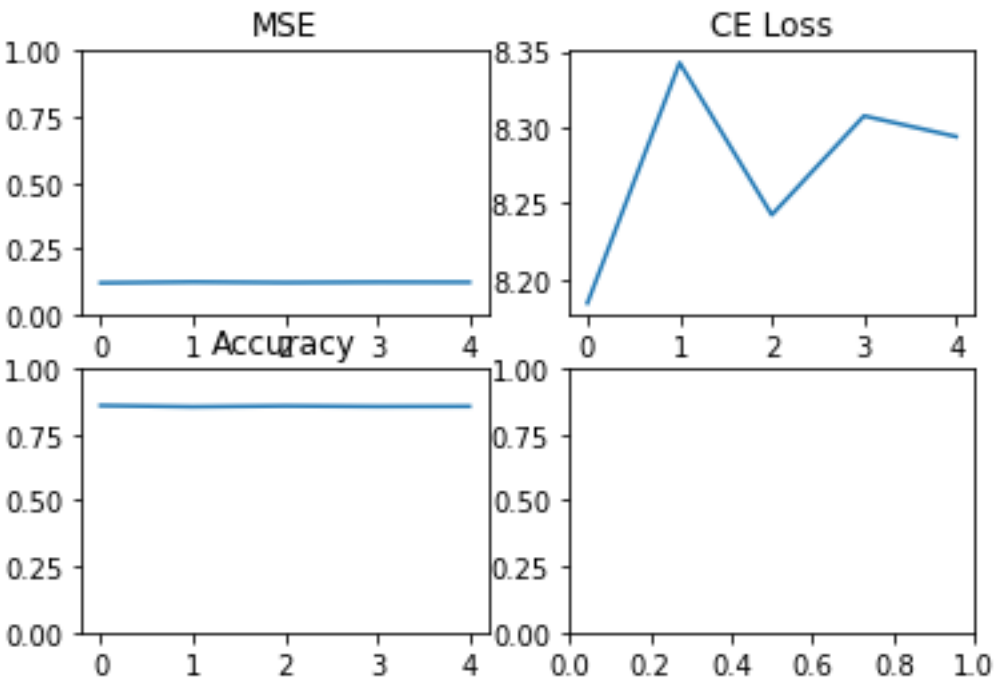
αλγόριθμος πίσω διάδοσης σφάλματος δεν χρησιμοποιεί σταθερά ορμής. Θεωρητικά μπορούν να χρησιμοποιηθούν και τιμές α < 0, αλλά αυτό στην πράξη δεν γίνεται.

## Α4. Ομαλοποίηση

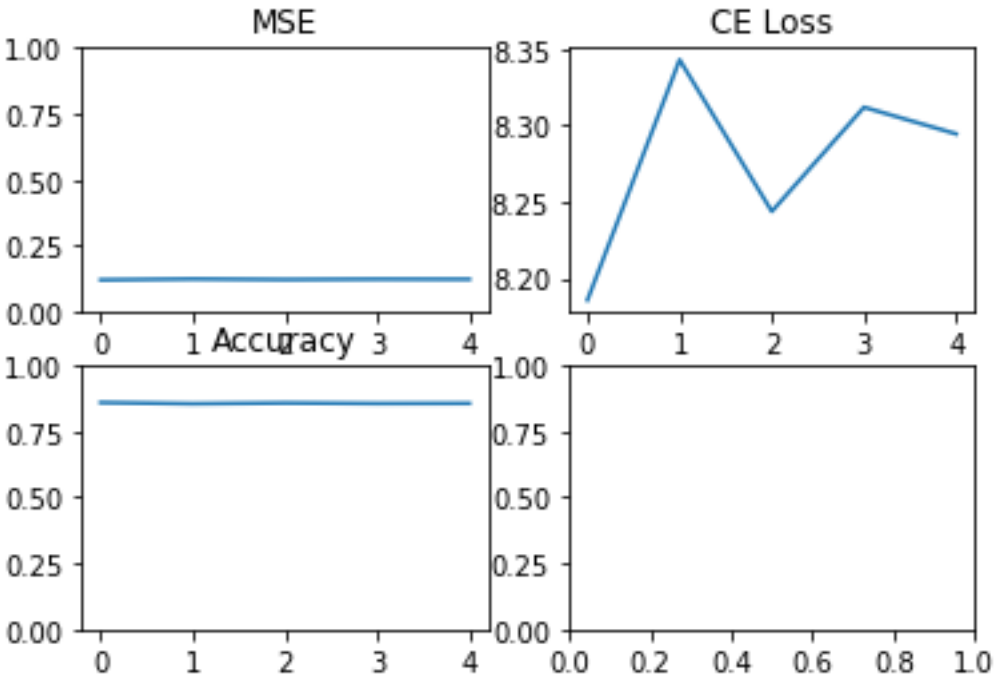
Οι γραφικές παραστάσεις σύγκλισης ανά κύκλο εκπαίδευσης (Μ.Ο.):




*Εικόνα 11 - H1: O, H2: O, d: 0.1*




*Εικόνα 12 - H1:O, H2:O, d:0.5*




*Εικόνα 13 - H1: O, H2: O, d: 0.9*

| m | CE loss | MSE | Accuracy |
|---|---|---|---|
| **0.1** | ~8.18 | ~0.12 | ~0.85 |
| **0.5** | ~8.18 | ~0.12 | ~0.85 |
| **0.9** | ~8.18 | ~0.12 | ~0.85 |

Βλέπουμε ότι δεν υπάρχει βελτίωση μεταξύ των διαφορετικών τιμών της παραμέτρου decay, πράγμα που είναι λογικό, διότι δεν παρατήρησα σε κανένα από τα προηγούμενα ερωτήματα φαινόμενο overfitting.

# Bibliography

Anon., n.d. *wikipedia.* [Online]
Available at: en.wikipedia.org/wiki/Mean_squared_error

Anon., n.d. *wikipedia.* [Online]
Available at: en.wikipedia.org/wiki/Early_stopping