

## **ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ**

**Διδάσκοντες: Σ. Λυκοθανάσης, Δ. Κουτσομητρόπουλος**

**Ακαδημαϊκό Έτος 2022-2023**

### **Εργαστηριακή Άσκηση**

#### **Μέρος Α'**

**Σούρλας Ζήσης**

**Αρ. Μητρώου: 1072477**

**sourlas.zisis@upnet.gr**

## Κώδικας áσκησης

Ο κώδικας υλοποίησης της áσκησης βρίσκεται στο ακόλουθο github repo.

<https://github.com/zisisour/CEID-Computational-Intelligence-Projects-2023/>

### A1. Προεπεξεργασία και Προετοιμασία δεδομένων

- a) Για την προεπεξεργασία των δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη `sklearn`.

#### Μετατροπή κατηγορικών σε αριθμητικές τιμές

Για να μετατρέψουμε τις κατηγορικές τιμές της εισόδου σε αριθμητικές χρησιμοποιήθηκε ο `OrdinalEncoder` της `sklearn` ο οποίος κωδικοποιεί τις κατηγορικές τιμές σε ακέραιες τιμές στο διάστημα [0, πλήθος\_κατηγοριών-1]. Για την κωδικοποίηση των κλάσεων χρησιμοποιήθηκε ο `LabelEncoder` της `sklearn` ο οποίος κωδικοποιεί τις κλάσεις σε ακέραιες τιμές στο διάστημα [0, πλήθος\_κλάσεων-1] και προτείνεται από το documentation της `scikit learn` για κωδικοποίηση κλάσεων.

#### Κεντράρισμα (Centering)

Το κεντράρισμα των δεδομένων είναι μια μεθοδολογία που χρησιμοποιείται πολύ συχνά στην προπεξεργασία δεδομένων. Αφαιρώντας το μέσο όρο από τα δεδομένα δημιουργούμε δεδομένα με μηδενική μέση τιμή. Αυτό σημαίνει ότι κατά τη διαδικασία μάθησης μειώνεται η επίδραση των ακραίων τιμών που μπορεί να οδηγήσουν σε καθυστέρηση στη σύγκλιση και σε μικρότερη ικανότητα γενίκευσης. Συνεπώς κρίθηκε σκόπιμο να χρησιμοποιηθεί στην παρούσα εργασία και υλοποιήθηκε με χρήση του `StandardScaler` της `sklearn`.

#### Κανονικοποίηση (Min-Max Scaling)

Το min-max scaling είναι μια πολύ χρήσιμη τεχνική όταν τα δεδομένα βρίσκονται σε διαφορετικές κλίμακες. Με αυτή την τεχνική μεταφέρονται όλες οι τιμές σε μια συγκεκριμένη κλίμακα ([0,1]) έτσι ώστε να διατηρούνται μεν οι διαφορές (αναλογικά) μεταξύ των δειγμάτων ενός τύπου εισόδου αλλά μεταξύ δειγμάτων διαφορετικών τύπων εισόδων που βρίσκονται αρχικά σε διαφορετικά εύρη τιμών (π.χ. φύλο και βάρος) να υπάρχει εν τέλει μία ενιαία κλίμακα τιμών ώστε να μην δίνεται αρχικά περισσότερο βάρος σε κάποιες εισόδους από άλλες. Στη συγκεκριμένη εργασία το dataset που δίνεται περιλαμβάνει τιμές που βρίσκονται σε πολύ διαφορετικές κλίμακες και κατά συνέπεια κρίθηκε απαραίτητο να χρησιμοποιηθεί min-max scaling (υλοποιήθηκε με τον `MinMaxScaler` της `sklearn`). Ωστόσο η τεχνική αυτή είναι ευαίσθητη στην παρουσία ακραίων τιμών (outliers) καθώς τείνει να συμπιέσει τις συνηθισμένες τιμές (inliers) σε μικρό κομμάτι του εύρους τιμών. Αυτός είναι ακόμα ένας λόγος για τον οποίο πρέπει να εφαρμοστεί κεντράρισμα στα δεδομένα.

#### Τυποποίηση (Standarization)

Η τυποποίηση χρησιμοποιείται για να μετατρέψει τα δεδομένα ώστε αυτά να έχουν μέση τιμή μηδέν και διακύμανση ένα. Ωστόσο η τυποποίηση προϋποθέτει ότι τα δεδομένα έχουν Gaussian κατανομή ή κάτι κοντά σε αυτήν. Από την στιγμή που δεν γνωρίζουμε τίποτα για την κατανομή των δεδομένων μας δεν κρίνεται σκόπιμο να χρησιμοποιήσουμε αυτή την τεχνική ειδικά καθώς και αυτή είναι ευαίσθητη στην παρουσία outliers.

- b) Για την δημιουργία των 5 folds που απαιτούνται χρησιμοποιήθηκε η **StratifiedKFold** της **sklearn**. Η συγκεκριμένη συνάρτηση σε αντίθεση με την **KFold** διατηρεί σε κάθε fold την αρχική αναλογία δειγμάτων ανά κλάση.

## A2. Επιλογή αρχιτεκτονικής

- a) Επιλογή συνάρτησης κόστους

### Διεντροπία (Cross Entropy)

Η διεντροπία μετράει την διαφορά μεταξύ της κατανομής πιθανότητας των παραγόμενων αποτελεσμάτων του νευρωνικού δικτύου και της κατανομής πιθανότητας των επιθυμητών αποτελεσμάτων. Ελαχιστοποιώντας την διεντροπία το μοντέλο αυξάνει την πιθανότητα ανάθεσης σωστής κλάσης στα δεδομένα εισόδου. Αυτό είναι ένα πολύ χρήσιμο χαρακτηριστικό για προβλήματα ταξινόμησης τα οποία απαιτούν ακριβώς αυτό, δηλαδή την ανάθεση ετικετών σε σετ δεδομένων εισόδου. Για αυτόν το λόγο η διεντροπία χρησιμοποιείται κατά βάση σε προβλήματα ταξινόμησης και για αυτόν τον λόγο χρησιμοποιήθηκε και στην παρούσα εργασία. Συγκεκριμένα χρησιμοποιείται **Sparse Categorical Crossentropy** επειδή χρησιμοποιούμε **integer encoding** για τις κλάσεις.

### Μέσο Τετραγωνικό Σφάλμα (MSE)

Το Μέσο Τετραγωνικό Σφάλμα υπολογίζεται παίρνοντας το μέσο όρο των τετραγώνων των διαφορών των πραγματικών με των επιθυμητών τιμών. Δείχνει επί της ουσίας πόσο απέχουν οι προβλεπόμενες τιμές από τις πραγματικές. Χρησιμοποιείται κυρίως σε προβλήματα παλινδρόμησης.

### Ακρίβεια ταξινόμησης (Accuracy)

Η ακρίβεια ταξινόμησης δείχνει απλώς το ποσοστό σωστών προβλέψεων ενός μοντέλου. Είναι χρήσιμη μετρική για να μετρήσουμε την απόδοση ενός μοντέλου σε προβλήματα ταξινόμησης αφού μας δείχνει ακριβώς πόσο καλά ταξινομήθηκαν τα δεδομένα εισόδου ωστόσο δεν είναι κατάλληλη για συνάρτηση κόστους καθώς δεν είναι διαφορίσιμη πράγμα απαραίτητο για τις συναρτήσεις κόστους. Επιπλέον η ακρίβεια ταξινόμησης δε δίνει κάποια πληροφορία σχετικά με τη “σιγουριά” με την οποία ταξινομούνται τα δεδομένα εισόδου. Π.χ. Ας πάρουμε την περίπτωση που ένα μοντέλο ταξινομεί τη φωτογραφία ενός σκύλου στην κλάση “σκύλος” με πιθανότητα 70% και στην κλάση “γάτα” με πιθανότητα 30% ενώ ένα άλλο μοντέλο ταξινομεί την ίδια φωτογραφία στην κλάση “σκύλος” με 90% και στην κλάση “γάτα” με 10%. Τα δύο μοντέλα θα έχουν την ίδια ακρίβεια ταξινόμησης αφού και τα δύο εν τέλει ταξινομούν σωστά την φωτογραφία όμως το δεύτερο είναι πολύ πιο “σίγουρο” για την ταξινόμηση που έκανε κάτι το οποίο φαίνεται μόνο στην διεντροπία του και όχι στην ακρίβεια ταξινόμησης.

- b) Επιλογή αριθμού νευρώνων στο επίπεδο εξόδου

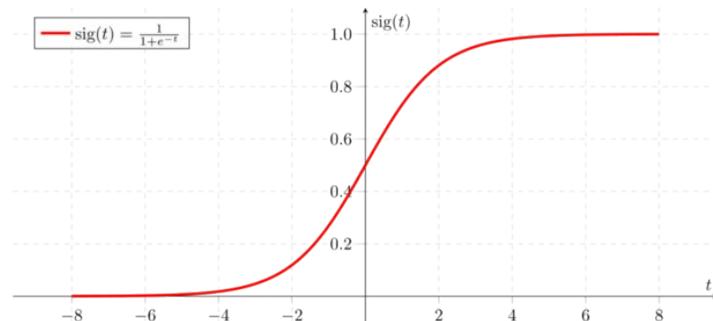
Λόγω του ότι έχουμε πρόβλημα **Multiclass Classification** δηλαδή θέλουμε να ταξινομήσουμε τα δεδομένα εισόδου σε μια σειρά από αμοιβαία αποκλειόμενες ετικέτες θα χρησιμοποιήσουμε ένα νευρώνα ανά κλάση (άρα το πλήθος των νευρώνων ισούται

με το πλήθος των κλάσεων δηλαδή 5) . Ο κάθε νευρώνας επιστρέφει την πιθανότητα με την οποία ταξινομείται η είσοδος στην κλάση στην οποία αντιστοιχεί.

### c) Επιλογή συνάρτησης ενεργοποίησης για τους κρυφούς κόμβους

Η συνάρτηση ενεργοποίησης των κρυφών νευρώνων πρέπει να είναι μη γραμμική καθώς μια γραμμική συνάρτηση ενεργοποίησης θα σήμαινε απλώς ότι οι έξοδοι των νευρώνων είναι ένας γραμμικός συνδυασμός των εισόδων τους καθιστώντας αχρείαστη την ύπαρξη κρυφών επιπέδων νευρώνων. Βασική προϋπόθεση της συνάρτησης ενεργοποίησης είναι να είναι διαφορίσιμη καθώς η παράγωγός της χρησιμοποιείται από τον Αλγόριθμο Πίσω Διάδοσης για τον υπολογισμό των βαρών. Συνήθως για τους κρυφούς κόμβους χρησιμοποιούνται τρεις συναρτήσεις:

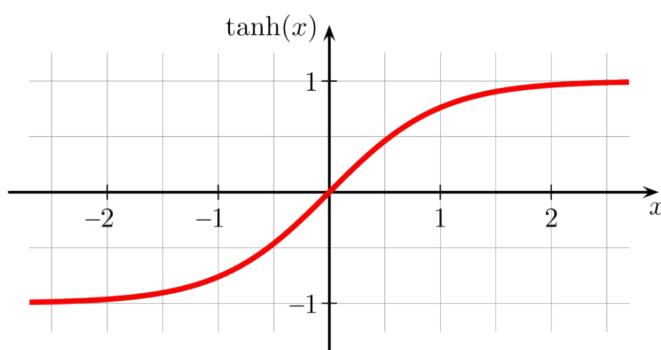
#### Λογιστική συνάρτηση



Πηγή: <https://www.aitude.com/>

Η λογιστική συνάρτηση χρησιμοποιείται ως συνάρτηση ενεργοποίησης για τους κρυφούς νευρώνες καθώς πληροί τα κριτήρια που περιγράψαμε παραπάνω ωστόσο παρουσιάζει δύο βασικά προβλήματα. Πρώτον ότι δεν έχει κέντρο το μηδέν πράγμα το οποίο μπορεί να οδηγήσει σε ταλαντώσεις στη μεταβολή των βαρών. Δεύτερον ότι όταν η έξοδος του νευρώνα βρίσκεται στα όρια της συνάρτησης η παράγωγος τείνει στο μηδέν με αποτέλεσμα ο νευρώνας να μη συνεισφέρει στην πίσω διάδοση του λάθους. Το φαινόμενο αυτό ονομάζεται gradient kill.

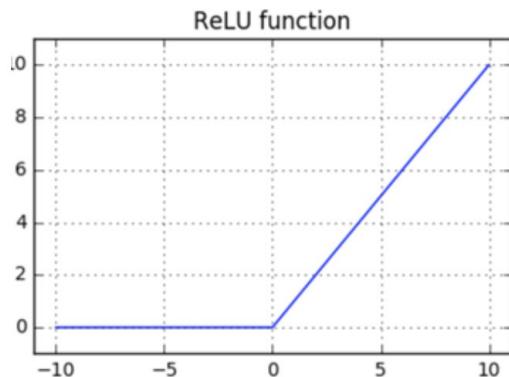
#### Συνάρτηση Υπερβολικής Εφαπτομένης (Tanh)



Πηγή: <https://www.aitude.com/>

Πρόκειται για μαθηματικό μετασχηματισμό της λογιστικής. Προτιμάται έναντι της τελευταίας λόγω του ότι έχει κέντρο το μηδέν ωστόσο εμφανίζεται και πάλι το φαινόμενο gradient kill.

### ReLU (Rectified Linear Unit)



Πηγή: <https://www.aitude.com>

Η συγκεκριμένη συνάρτηση δεν παρουσιάζει το φαινόμενο gradient kill καθώς έχει εύρος τιμών  $[0, \infty)$ . Λόγω του ότι για αρνητικές εισόδους επιστρέφει μηδενική έξοδο, ορισμένοι νευρώνες δεν ενεργοποιούνται πάντα με αποτέλεσμα καλύτερη απόδοση και ευκολότερους υπολογισμούς. Επιπλέον επειδή πρόκειται για απλούστερη συνάρτηση μαθηματικά είναι και υπολογιστικά φθηνότερη. Για αυτούς τους λόγους χρησιμοποιείται γενικώς περισσότερο από τις προαναφερθείσες και επιλέχθηκε και για τη συγκεκριμένη εργασία.

#### d) Επιλογή συνάρτησης ενεργοποίησης για τους κόμβους εξόδου

Για το επίπεδο εξόδου επιλέχθηκε η συνάρτηση **Softmax**. Ο λόγος για αυτό είναι όπως προαναφέρθηκε ότι έχουμε πρόβλημα **Multiclass Classification** και κάθε κόμβος εξόδου επιστρέφει την πιθανότητα η είσοδος να ανήκει σε μια κλάση. Συνεπώς θέλουμε μια συνάρτηση ενεργοποίησης η οποία να μας δίνει αυτές τις πιθανότητες οι οποίες να αθροίζουν στο 1. Αυτήν ακριβώς τη λειτουργία επιτελεί η **Softmax** και για αυτό το λόγο χρησιμοποιείται σε τέτοιου είδους προβλήματα.

#### e) Πειράματα με τον αριθμό των κρυφών νευρώνων

Για την εκπαίδευση του νευρωνικού δικτύου χρησιμοποιήθηκε batch μεγέθους 128 και ο μέγιστος αριθμός κύκλων εκπαίδευσης ανά fold ορίστηκε στις 600. Ως μέθοδος βελτιστοποίησης χρησιμοποιήθηκε το **gradient descent** με ρυθμό μάθησης  $\eta=0.001$  και σταθερά ορμής  $t=0$  (Γενικευμένος Κανόνας Δέλτα). Εφαρμόστηκε επιπλέον η μέθοδος πρώτου τερματισμού με κατάλληλο κριτήριο.

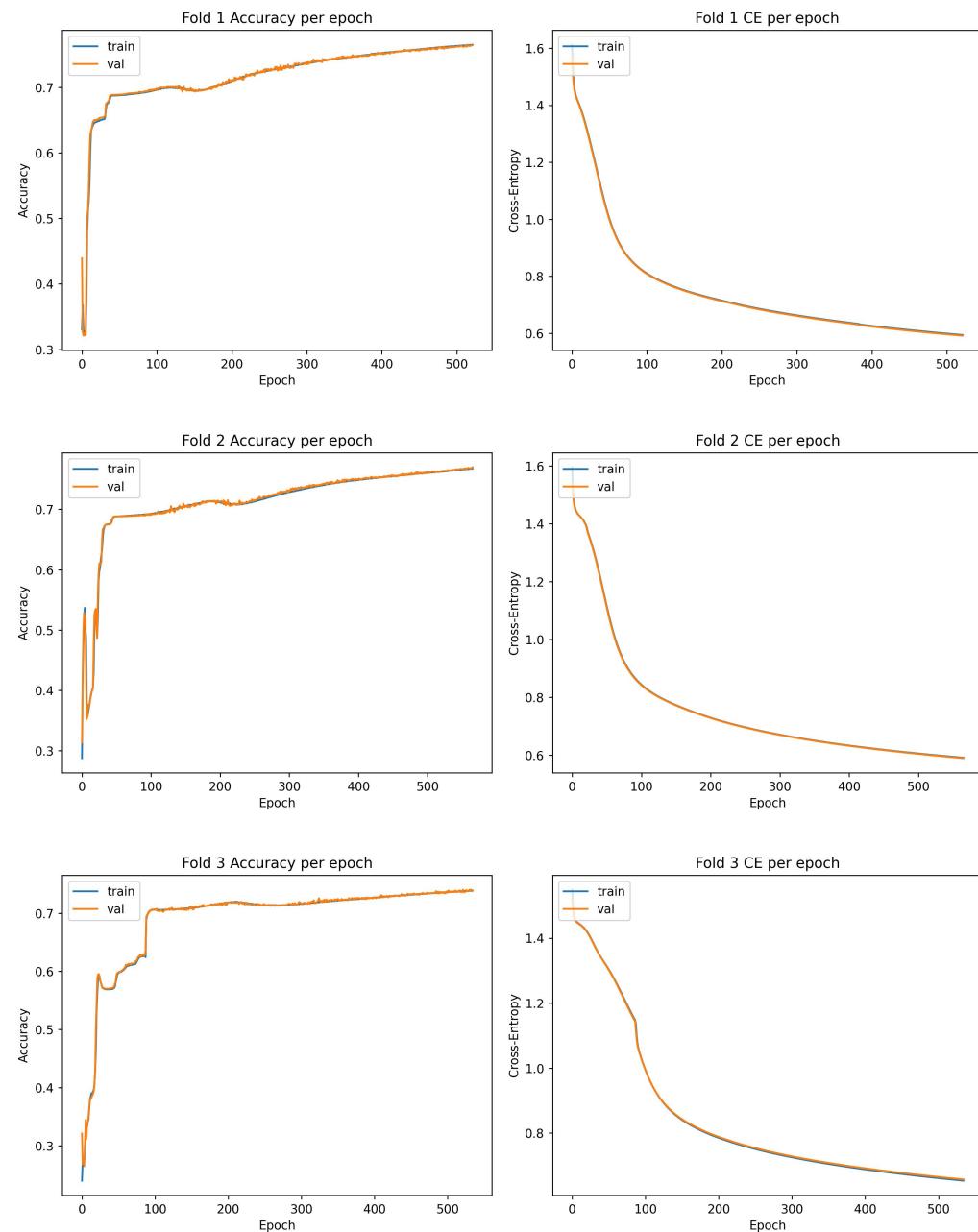
#### Αποτελέσματα πειραμάτων

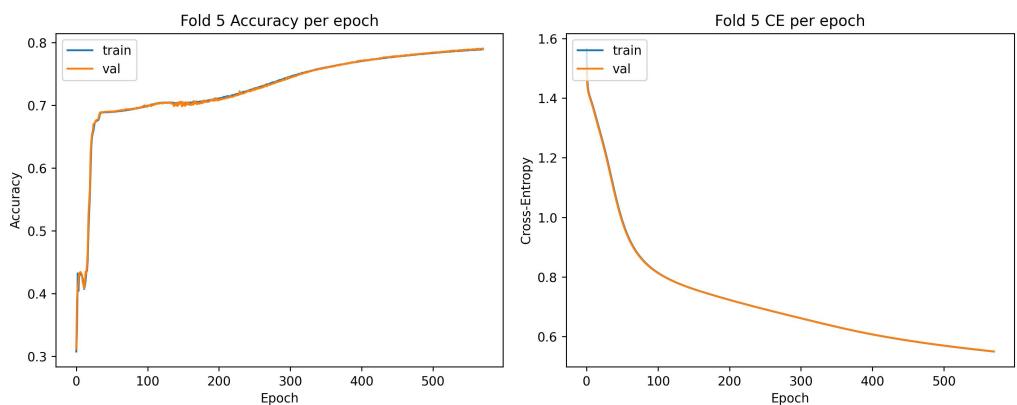
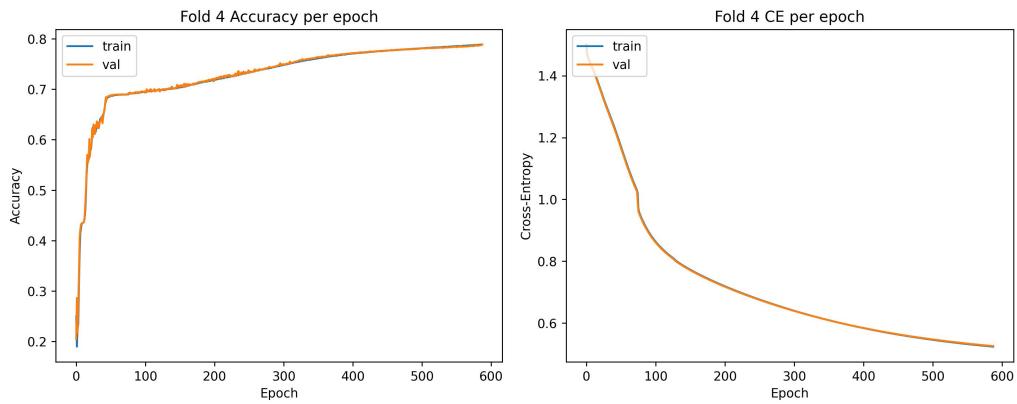
**Γίνακας μέσω τιμών μετρικών**

Αριθμός νευρώνων στο κρυφό επίπεδο	CE Loss	MSE	Accuracy
$H_1 = 5$	0.584	5.442	0.769
$H_1 = 12$	0.503	5.450	0.806
$H_1 = 23$	0.482	5.451	0.812

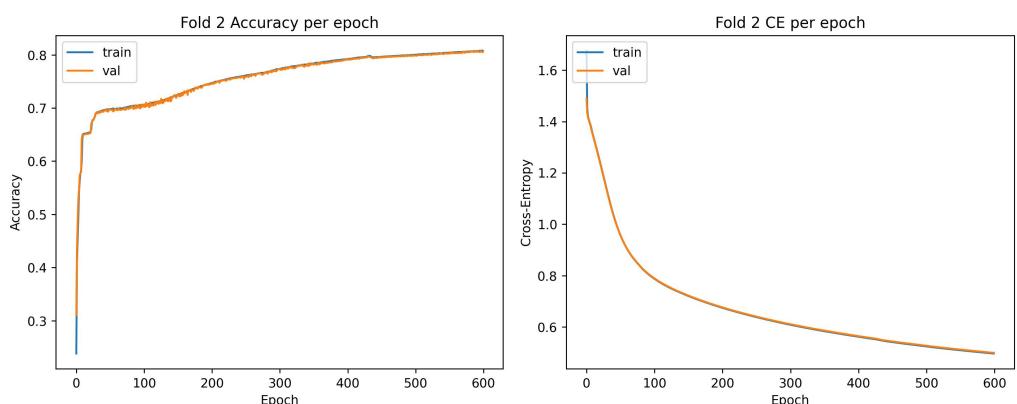
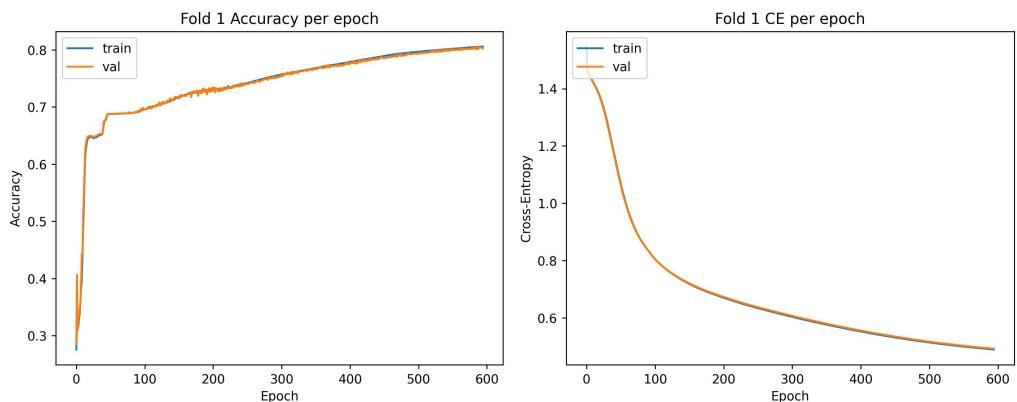
**Γραφικές παραστάσεις**

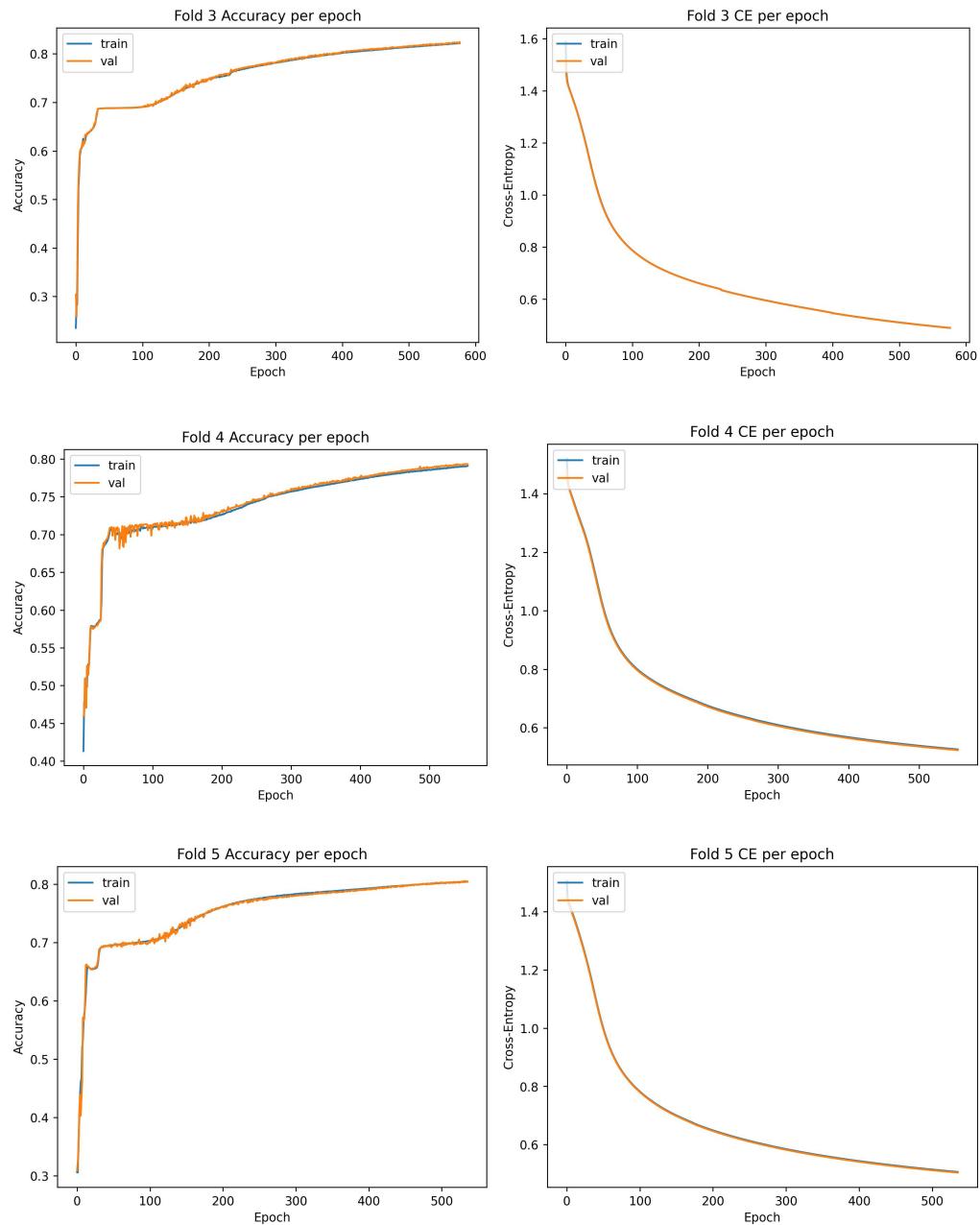
**5 νευρώνες στο κρυφό επίπεδο**



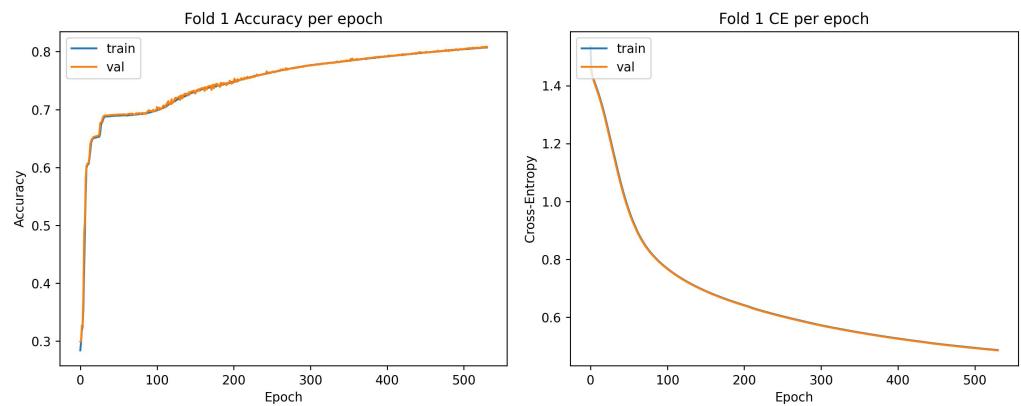


## 12 νευρώνες στο κρυφό επίπεδο

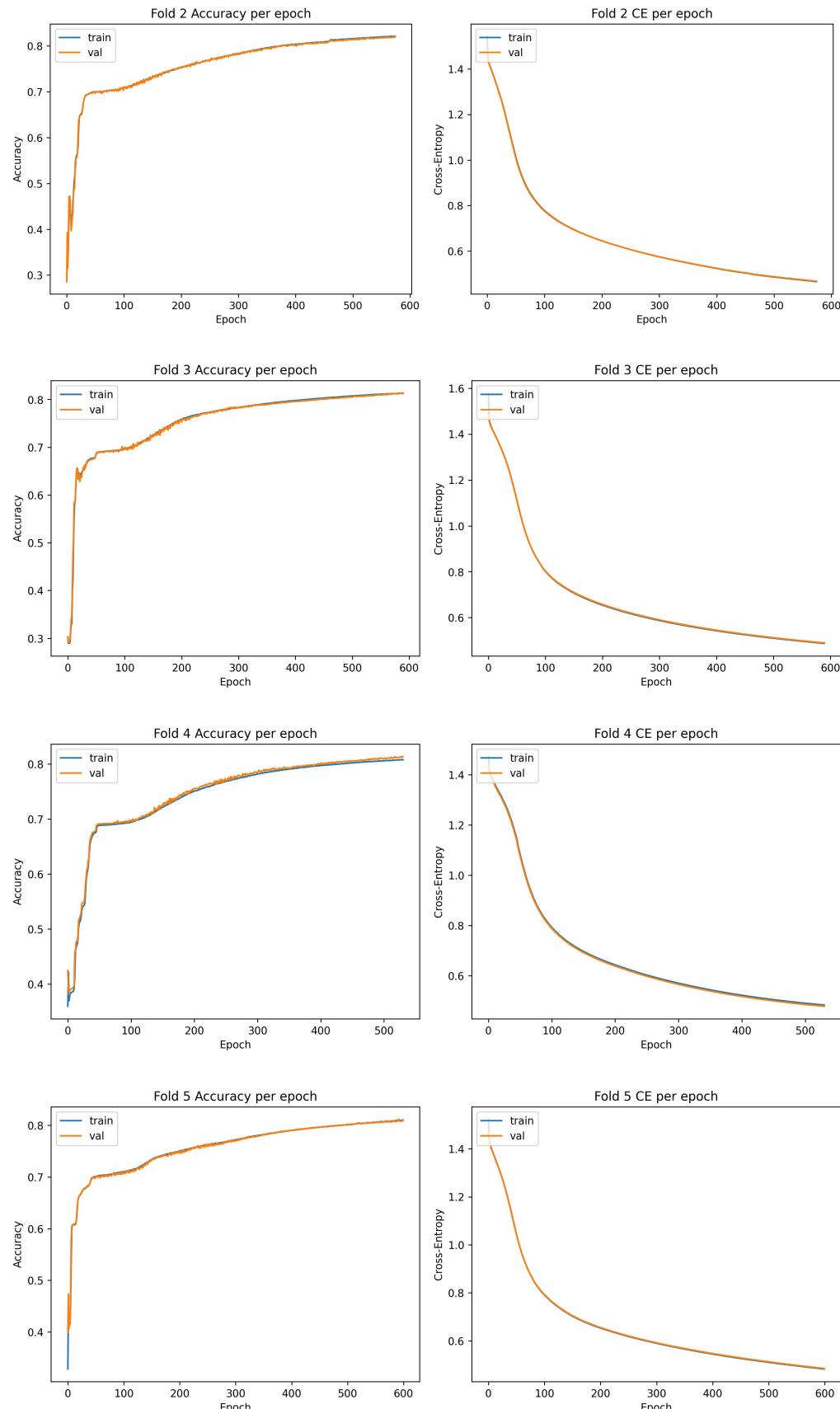




### 23 νευρώνες στο κρυφό επίπεδο



**Παν. Πατρών, ΤΜΗΥΠ**  
**CEID\_NE5218: Υπολογιστική Νοημοσύνη 2022-23**



## **Συμπεράσματα**

### **i. Σχετικά με τον αριθμό των κρυφών κόμβων**

Οι κρυφοί κόμβοι βοηθούν το νευρωνικό δίκτυο να “ανακαλύψει” χαρακτηριστικά μεταξύ των δεδομένων της εισόδου που το βοηθούν να ταξινομήσει καλύτερα τα δεδομένα στην έξοδο. Ωστόσο η προσθήκη υπερβολικά πολλών κρυφών κόμβων μπορεί να οδηγήσει (εκτός από την αύξηση της πολυπλοκότητας του μοντέλου) σε μείωση της ικανότητας γενίκευσης, πράγμα μη επιθυμητό.

Από τα πειράματά μας παρατηρούμε ότι η αύξηση στους 12 από τους 5 νευρώνες ευνοεί πράγματι το δίκτυο. Φαίνεται πως η προσέγγιση μιας ακρίβειας γύρω στο 70% γίνεται ταχύτερα ενώ η μέση ακρίβεια αυξάνεται περίπου 4-5%. Παράλληλα η συνάρτηση κόστους φαίνεται να συγκλίνει σε μικρότερες τιμές με μεγαλύτερη ταχύτητα.

Αντιθέτως η αύξηση στους 23 νευρώνες δεν φαίνεται να προσφέρει τίποτα ιδιαίτερο. Δεν παρατηρείται αύξηση στην ταχύτητα σύγκλισης ούτε της ακρίβειας ταξινόμησης ούτε της συνάρτησης κόστους ενώ και οι μέσες τιμές παρουσιάζουν πολύ μικρές βελτιώσεις (στα όρια του λάθους). Συνεπώς το μόνο που προσφέρει αυτή η αύξηση είναι αύξηση πολυπλοκότητας και ενδεχομένως απώλεια ικανότητας γενίκευσης, κάτι το οποίο δεν παρατηρείται στα πειράματα ωστόσο. Κατά συνέπεια για τα επόμενα πειράματα επιλέχθηκε το μοντέλο με τους 12 κρυφούς νευρώνες

### **ii. Σχετικά με την επιλογή της συνάρτησης κόστους**

Η επιλογή της διεντροπίας ως συνάρτησης κόστους αποδεικνύεται σωστή επιλογή. Παρατηρούμε ότι η μείωση της διεντροπίας συνοδεύεται από αύξηση της ακρίβειας ταξινόμησης πράγμα που είναι το ζητούμενο. Επίσης κατά την εκπαίδευση παρατηρήθηκε το ενδιαφέρον φαινόμενο της μικρής αύξησης της ακρίβειας παράλληλα με τη σχετικά ταχύτερη μείωση της διεντροπίας. Αυτό μας δείχνει ότι ενώ το μοντέλο φαίνεται να μη γίνεται καλύτερο όσον αφορά την ορθή ταξινόμηση, γίνεται εν τούτοις καλύτερο ως προς την σιγουριά με την οποία ταξινομεί τα δείγματα, γεγονός που ενισχύει την ικανότητα γενίκευσης.

Το μέσο τετραγωνικό σφάλμα παρατηρείται να μένει σχεδόν στάσιμο καθ' όλη τη διάρκεια της εκπαίδευσης (για αυτό το λόγο δε συμπεριλήφθηκαν και οι σχετικές γραφικές παραστάσεις). Αυτό δεν είναι ανησυχητικό καθώς δεν χρησιμοποιείται ως συνάρτηση κόστους και κατά συνέπεια το μοντέλο δε στοχεύει στην ελάττωσή του. Επιπλέον επειδή ο υπολογισμός του διαφέρει εξ ολοκλήρου με αυτόν της διεντροπίας είναι απόλυτα λογικό να παρατηρούμε στασιμότητα στο MSE και μείωση της διεντροπίας.

### **iii. Σχετικά με την ταχύτητα σύγκλισης ως προς τις εποχές εκπαίδευσης**

Παρατηρούμε ότι το νευρωνικό δίκτυο προσεγγίζει μια ακρίβεια ταξινόμησης κοντά στο 70% αρκετά γρήγορα (το πολύ 100 εποχές) ενώ μετά η ταχύτητα σύγκλισης μειώνεται αισθητά. Επίσης παρατηρούμε μικρές αυξομειώσεις (ταλαντώσεις στο γράφημα) της ακρίβειας ταξινόμησης του validation set οι οποίες στην πορεία εξομαλύνονται. Αυτές μας δείχνουν την σταδιακή βελτίωση

της γενικευτικής ικανότητας του δικτύου. Καθώς το δίκτυο εκπαιδεύεται πάνω στα δεδομένα εκπαίδευσης αυξάνει την ακρίβεια ταξινόμησής τους αλλά δεν τα καταφέρνει εξίσου καλά σε δεδομένα που δεν έχει ξαναδεί (δεδομένα ελέγχου). Σταδιακά βελτιώνει αυτή την αδυναμία του (γενίκευση) και αυτό φαίνεται στο γράφημα.

**f) Επιλογή κριτηρίου τερματισμού**

Ως κριτήριο τερματισμού επιλέχθηκε η παρακολούθηση της συνάρτησης κόστους στα δεδομένα ελέγχου. Αν για διάστημα μεγαλύτερο των 5 κύκλων εκπαίδευσης δεν παρουσιάσει κάποια βελτίωση, η εκπαίδευση τερματίζεται πρόωρα. Ως βελτίωση θεωρείται μια μείωση (minimum delta) κατά τουλάχιστον 0.001 στις διαδοχικές μετρήσεις της διεντροπίας.

Αντί αυτού του κριτηρίου θα μπορούσε να χρησιμοποιηθεί ένα κριτήριο μη βελτίωσης ακρίβειας ταξινόμησης για ένα διάστημα καθώς τελικός σκοπός του μοντέλου είναι να πετύχει τη μεγαλύτερη δυνατή ακρίβεια. Ωστόσο επειδή μια σχετική στασιμότητα στην ακρίβεια ταξινόμησης συνοδεύομενη από μείωση της διεντροπίας σημαίνει πιο “σίγουρες” ταξινομήσεις επιλέχθηκε το προαναφερθέν κριτήριο.

**A3. Μεταβολές στο ρυθμό εκπαίδευσης και σταθερά ορμής**

Η τροποποίηση του κανόνα Δέλτα ώστε να περιλαμβάνει σταθερά ορμής δίνεται από τον εξής τύπο (όπου α ή σταθερά ορμής):

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) y_i(n)$$

Η σταθερά ορμής προσδιορίζει τη συμμετοχή των προηγούμενων μεταβολών των βαρών στη διαμόρφωση της τρέχουσα μεταβολής. Συνεπώς είναι εύλογο το α να είναι μικρότερο του 1 ώστε σταδιακά να μειώνεται το βάρος των παλιών μεταβολών (εφόσον η κάθε μεταβολή περιλαμβάνει τη σταθερά ορμής ως όρο, ο πολλαπλασιασμός όρων μικρότερων από 1 δίνει όλο και μικρότερα αποτελέσματα).

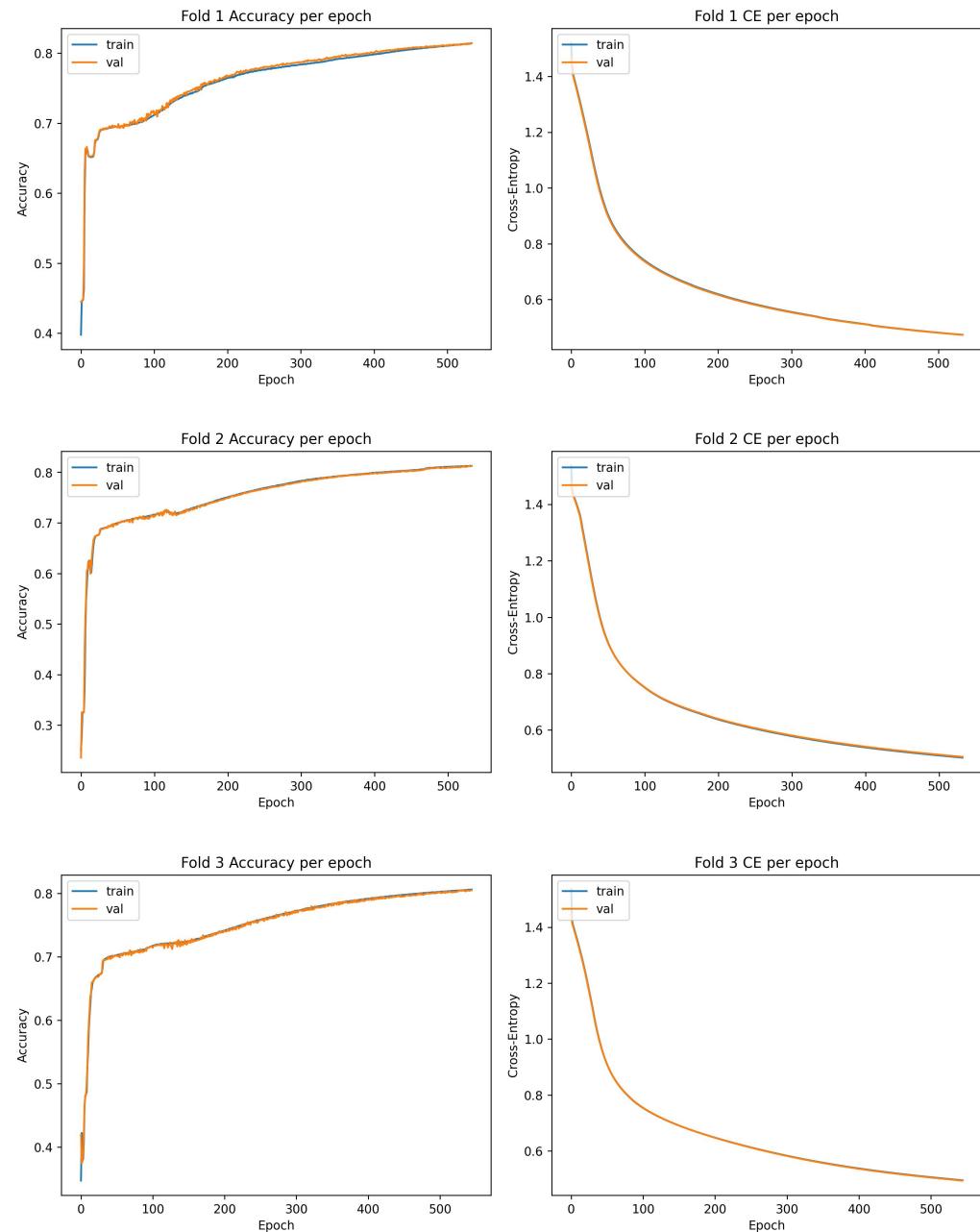
**Αποτελέσματα πειραμάτων**

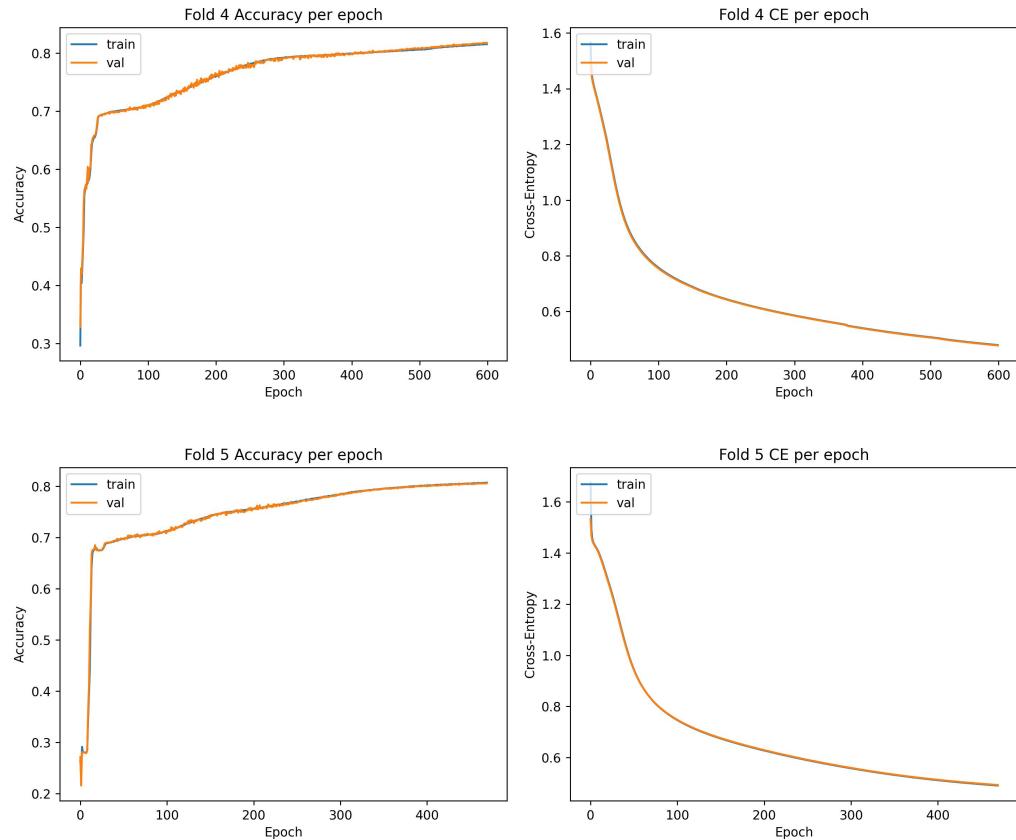
**Γίνακας μέσω τιμών μετρικών**

<b>η</b>	<b>m</b>	<b>CE loss</b>	<b>MSE</b>	<b>Acc</b>
0.001	0.2	0.489	5.451	0.81
0.001	0.6	0.434	5.459	0.836
0.05	0.6	0.198	5.498	0.936
0.1	0.6	0.183	5.5	0.942

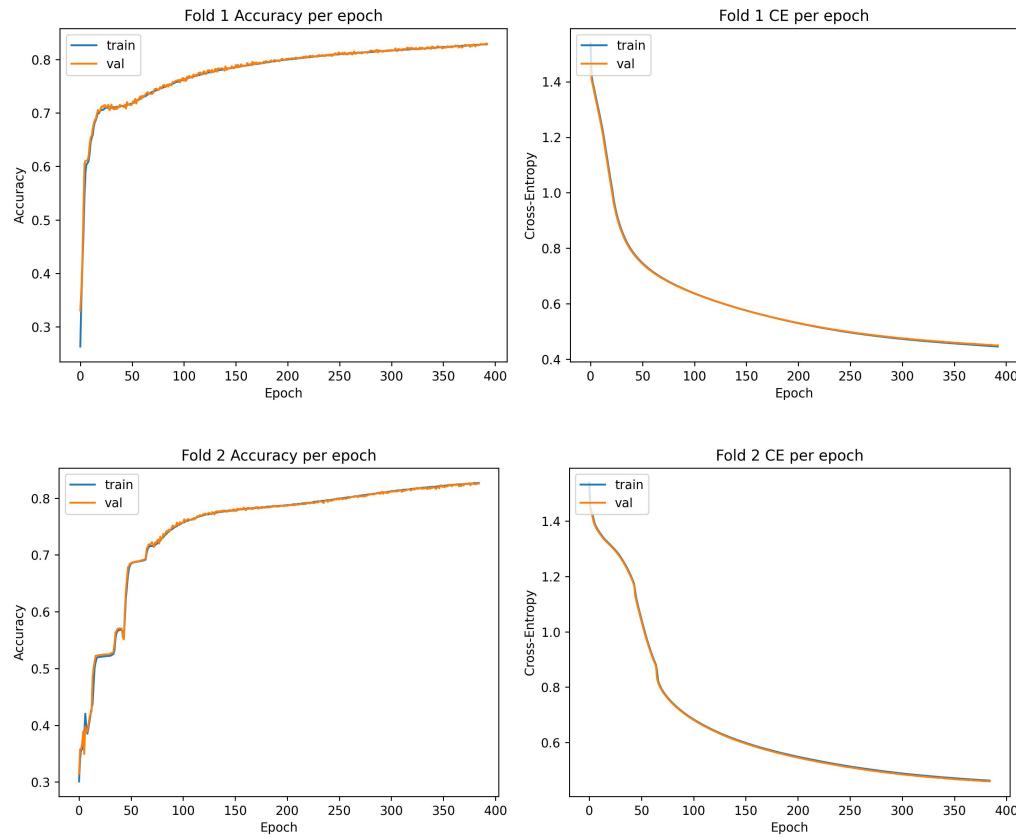
## Γραφικές παραστάσεις

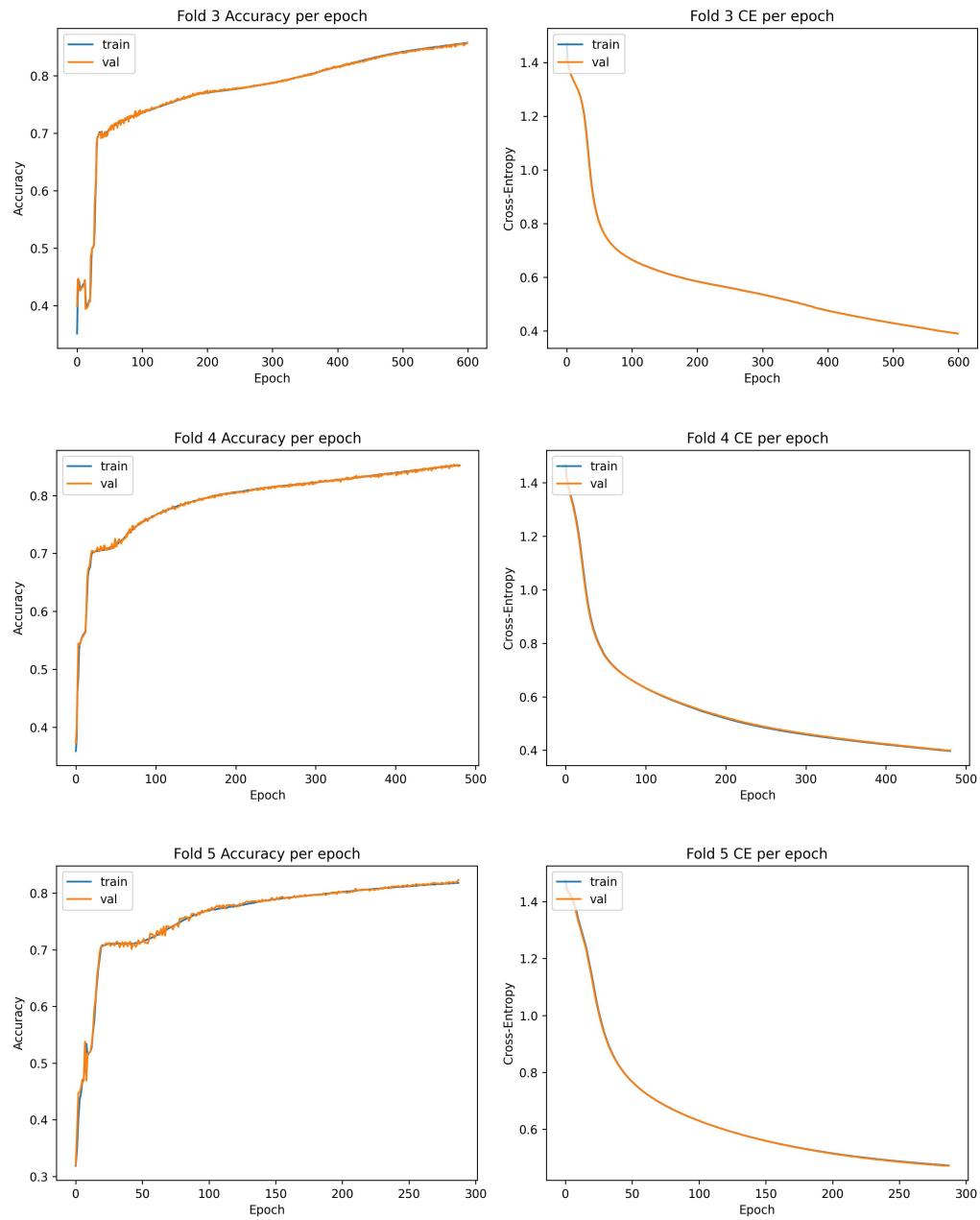
$\eta=0.001$   $m=0.2$



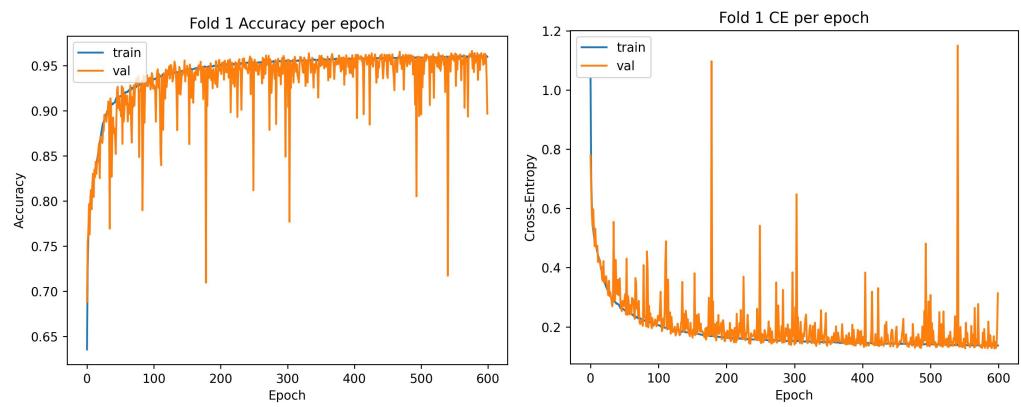


$\eta=0.001$   $m=0.6$

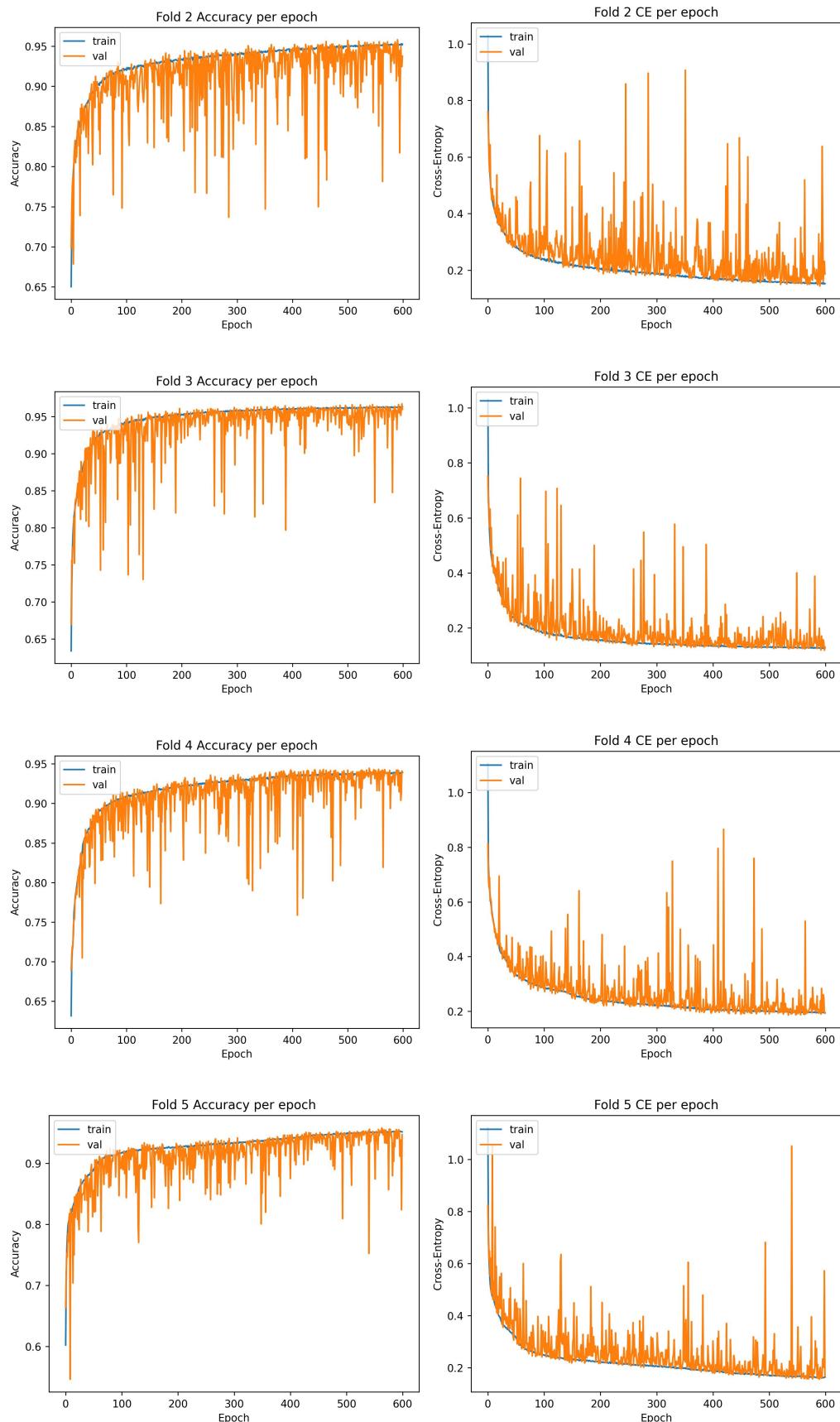




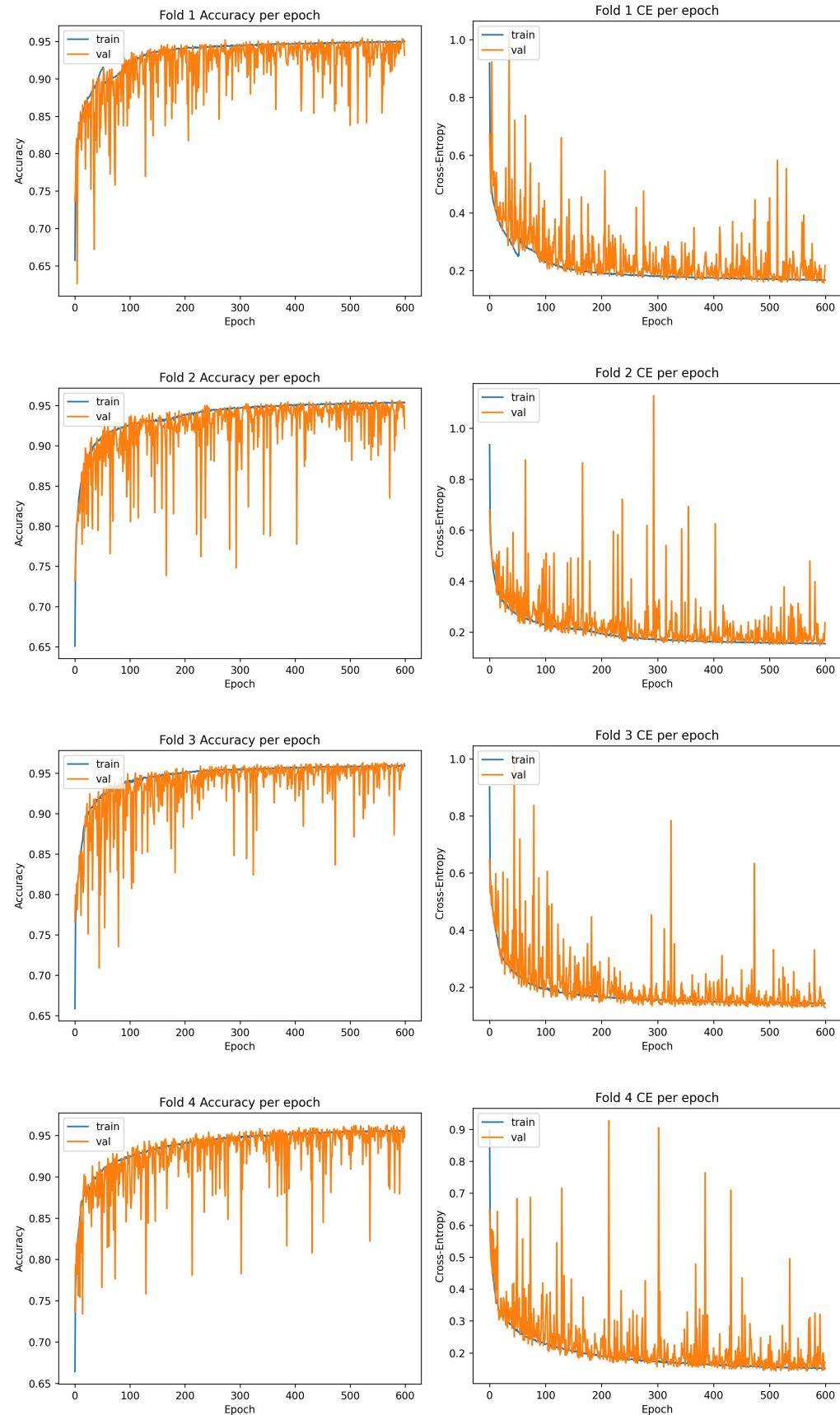
$\eta=0.05$   $m=0.6$

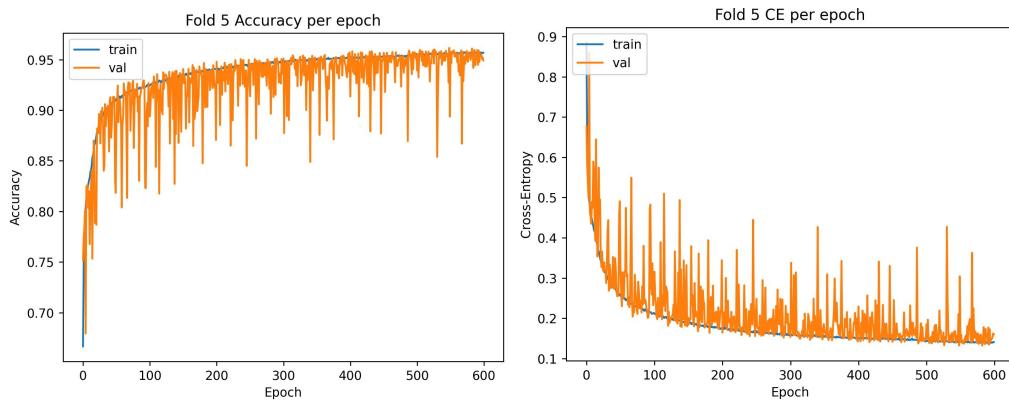


**Παν. Πατρών, ΤΜΗΥΠ**  
**CEID\_NE5218: Υπολογιστική Νοημοσύνη 2022-23**



$\eta=0.1$   $m=0.6$





## Συμπεράσματα

Η υπερπαράμετρος ρυθμού εκπαίδευσης η (learning rate) επηρεάζει σημαντικά τη διαδικασία μάθησης ενός νευρωνικού δικτύου. Χρησιμοποιώντας μικρό ρυθμό η εκπαίδευση είναι πιο ακριβής καθώς δεν κινδυνεύει να “χάσει” κάποιο τοπικό ελάχιστο ωστόσο η σύγκλιση σε ένα ολικό ή τοπικό ελάχιστο είναι αργή καθώς το βήμα που εκτελείται ανά κύκλο εκπαίδευσης είναι μικρότερο. Επιπλέον η διαδικασία εκπαίδευσης μπορεί να “κολλήσει” λόγω του μικρού ρυθμού μάθησης. Για μεγάλο ρυθμό εκπαίδευσης έχουμε ταχύτερη σύγκλιση ωστόσο εισάγεται μεγαλύτερη αστάθεια και ταλαντώσεις καθώς προσεγγίζοντας ένα ελάχιστο το μοντέλο μπορεί λόγω μεγάλου βήματος να το υπερπηδήσει (overshoot) και να πρέπει να κάνει το αντίστροφο “διορθωτικό” βήμα. Επιπλέον κινδυνεύει ακριβώς λόγω μεγαλύτερου βήματος να μην εντοπίσει κάποιο βαθύτερο ελάχιστο.

Η υπερπαράμετρος της σταθεράς ορμής μπορεί να βοηθήσει να μειωθούν οι αρνητικές επιπτώσεις του ρυθμού μάθησης. Χρησιμοποιώντας την ορμή αμβλύνεται η πορεία του gradient descent με αποτέλεσμα να αποφεύγονται οι ταλαντώσεις και να υπάρχει ταχύτερη σύγκλιση.

Οι παραπάνω θεωρητικές διατυπώσεις αποτυπώνονται σε ένα βαθμό στα πειράματά μας.

Για  $\eta=0.001$  και  $t=0.2$  ή  $t=0.6$  παρατηρούμε ότι το μοντέλο συγκλίνει ταχύτερα. Για  $t=0.6$  αυτή η τάση είναι εντονότερη ενώ παρατηρείται και σχετική αύξηση στη μέση τιμή της ακρίβειας ταξινόμησης με παράλληλη μείωσης της μέσης τιμής της συνάρτησης κόστους. Συνεπώς η προσθήκη ορμής πράγματι βοήθησε την εκπαίδευση επιστρέφοντας καλύτερα αποτελέσματα σε λιγότερους κύκλους εκπαίδευσης.

Η αύξηση του ρυθμού εκπαίδευσης σε  $\eta=0.05$  και  $\eta=0.1$  ενώ φαινομενικά αυξάνει τη μέση τιμή ακρίβειας ταξινόμησης και μειώνει την μέση τιμή της συνάρτησης κόστους, δημιουργεί έντονες ταλαντώσεις όπως φαίνεται στις γραφικές παραστάσεις. Για τα δεδομένα ελέγχου παρατηρούμε έντονη αυξομείωση στην συνάρτηση κόστους και στην ακρίβεια ταξινόμησης. Αυτό πιθανώς οφείλεται στο ότι καθώς το μοντέλο προσεγγίζει ένα τοπικό ελάχιστο το υπερπηδά (overshoot) λόγω του μεγάλου βήματος που εισάγει ο ρυθμός εκπαίδευσης.

Από τα παραπάνω συμπεραίνουμε πως το πιο αποδοτικό μοντέλο και αυτό το οποίο θα χρησιμοποιηθεί στα επόμενα παραδείγματα είναι αυτό με  $\eta=0.001$  και  $t=0.6$ .

\*Να σημειωθεί πως για την παραγωγή των αποτελεσμάτων των δύο τελευταίων περιπτώσεων αφαιρέθηκε το κριτήριο τερματισμού προκειμένου να φανούν καλύτερα οι ταλαντώσεις.

## A4. Ομαλοποίηση

### L1 Ομαλοποίηση

Με την ομαλοποίηση L1 προσθέτουμε μια “τιμωρία” στην συνάρτηση κόστους η οποία βασίζεται στο άθροισμα των απόλυτων τιμών των βαρών. Επειδή βασίζεται στις απόλυτες τιμές των βαρών, βάρη τα οποία μπορεί να κριθούν άχρηστα μηδενίζονται με αποτέλεσμα να γίνεται μια επιλογή των χρήσιμων για την εκπαίδευση χαρακτηριστικών (feature selection). Αυτό είναι ένα φαινόμενο το οποίο σε κάποιες περιπτώσεις είναι επιθυμητό ενώ σε άλλες όχι. Κρίναμε ότι για την παρούσα εργασία είναι σκόπιμο να υπάρξει αυτό το φαινόμενο καθώς το dataset περιέχει τιμές που πιθανόν να μην είναι χρήσιμες για την εκπαίδευση (π.χ. User). Για αυτόν τον λόγο επιλέχθηκε η L1 ομαλοποίηση.

### L2 Ομαλοποίηση

Στην ομαλοποίηση L2 προστίθεται το άθροισμα των τετραγώνων των βαρών στην συνάρτηση κόστους. Αυτό δεν οδηγεί σε feature selection δηλαδή σε μηδενισμό βαρών αλλά σε καλύτερη κατανομή των βαρών. Αυτή η ιδιότητα είναι ιδιαίτερα σημαντική όταν πρόκειται για σύνολα δεδομένων με χαρακτηριστικά που έχουν πολλές συσχετίσεις μεταξύ τους. Ένα μειονέκτημα αυτής της μεθόδου είναι ότι είναι ευαίσθητη στις ακραίες τιμές (outliers) λόγω της ύψωσης των βαρών στο τετράφωνο.

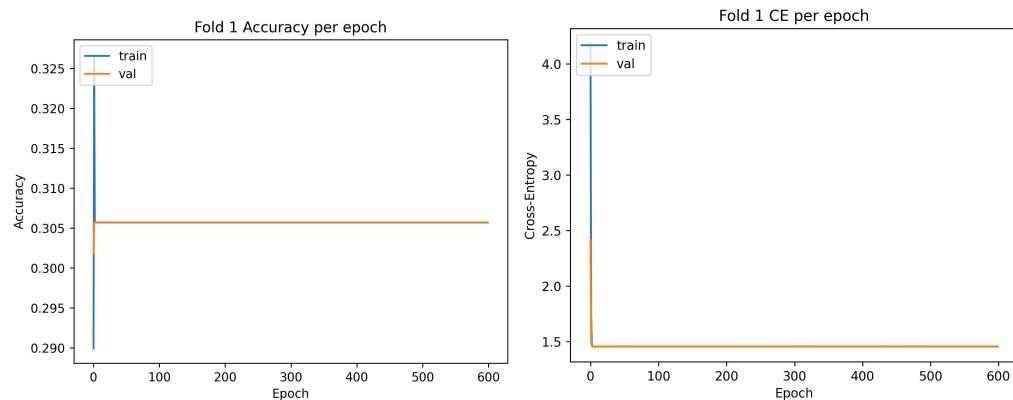
### Αποτελέσματα πειραμάτων

#### Πίνακας μέσων τιμών μετρικών

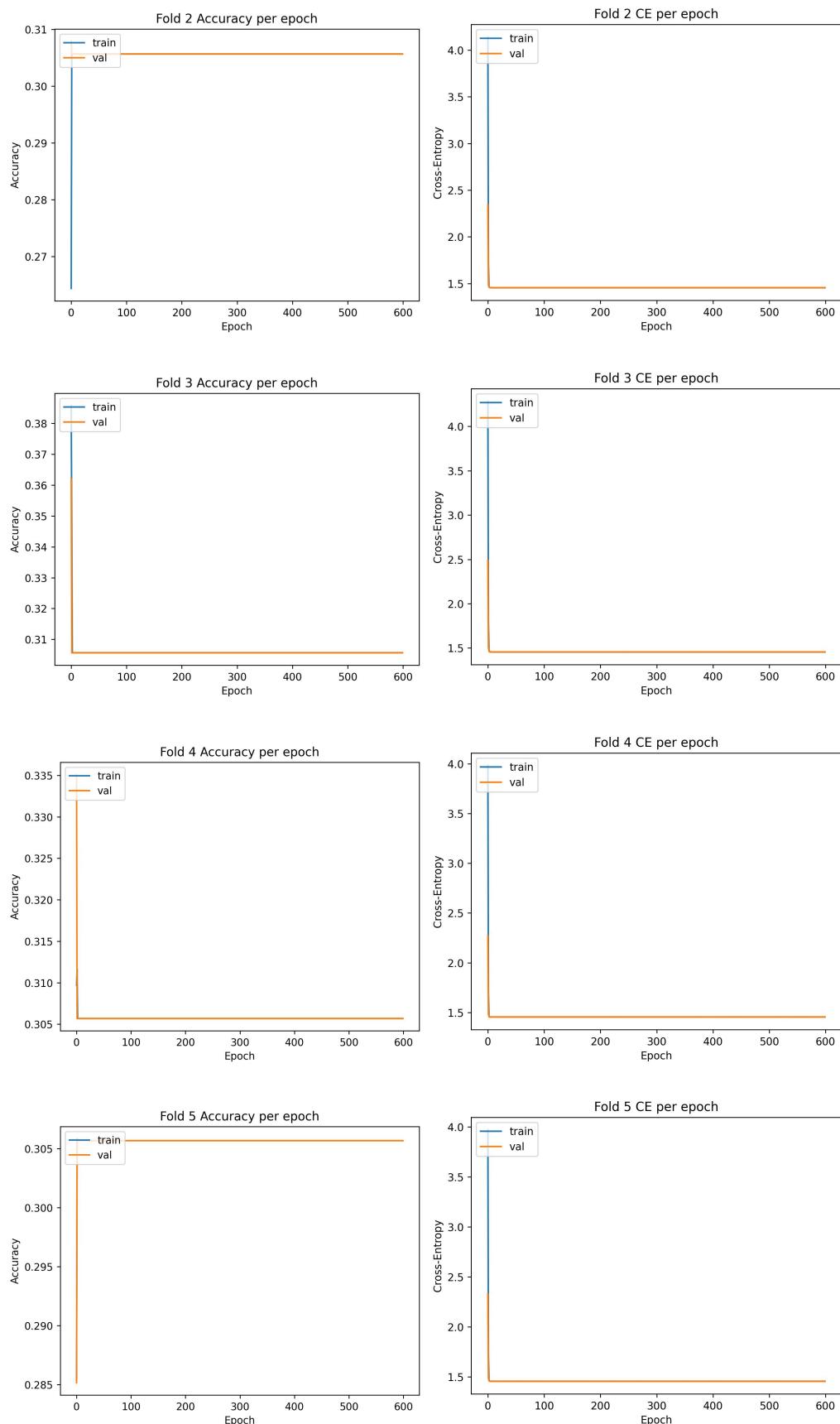
Συντελεστής $r$	CE loss	MSE	Acc
0.1	1.455	5.366	0.306
0.5	1.485	5.366	0.306
0.9	1.554	5.366	0.306

#### Γραφικές παραστάσεις

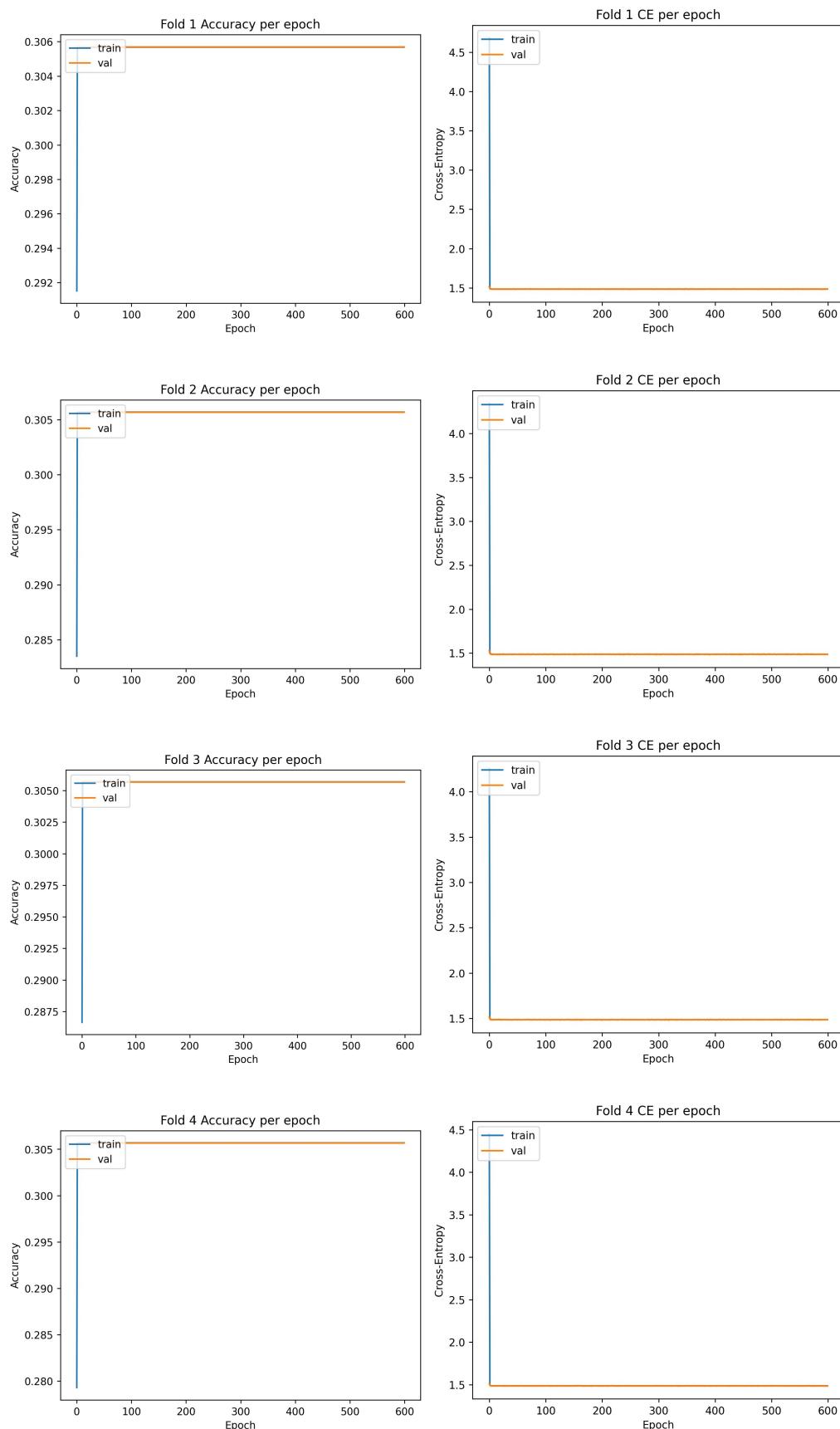
$r=0.1$

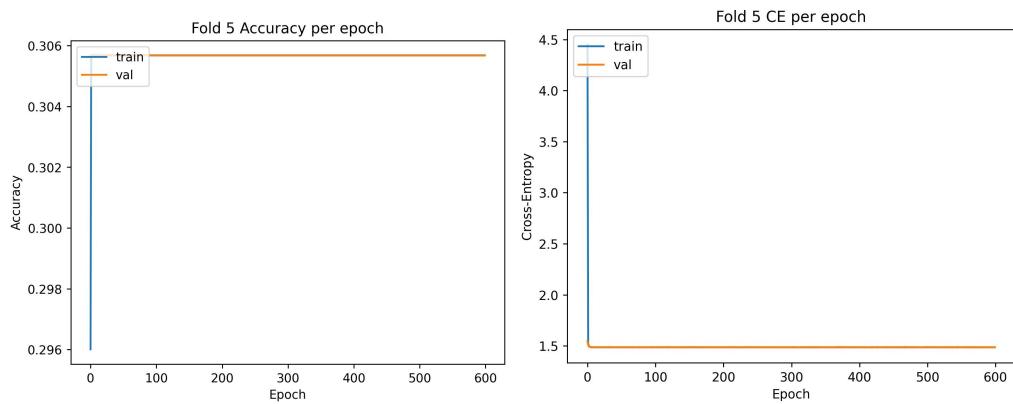


**Παν. Πατρών, ΤΜΗΥΠ**  
**CEID\_NE5218: Υπολογιστική Νοημοσύνη 2022-23**

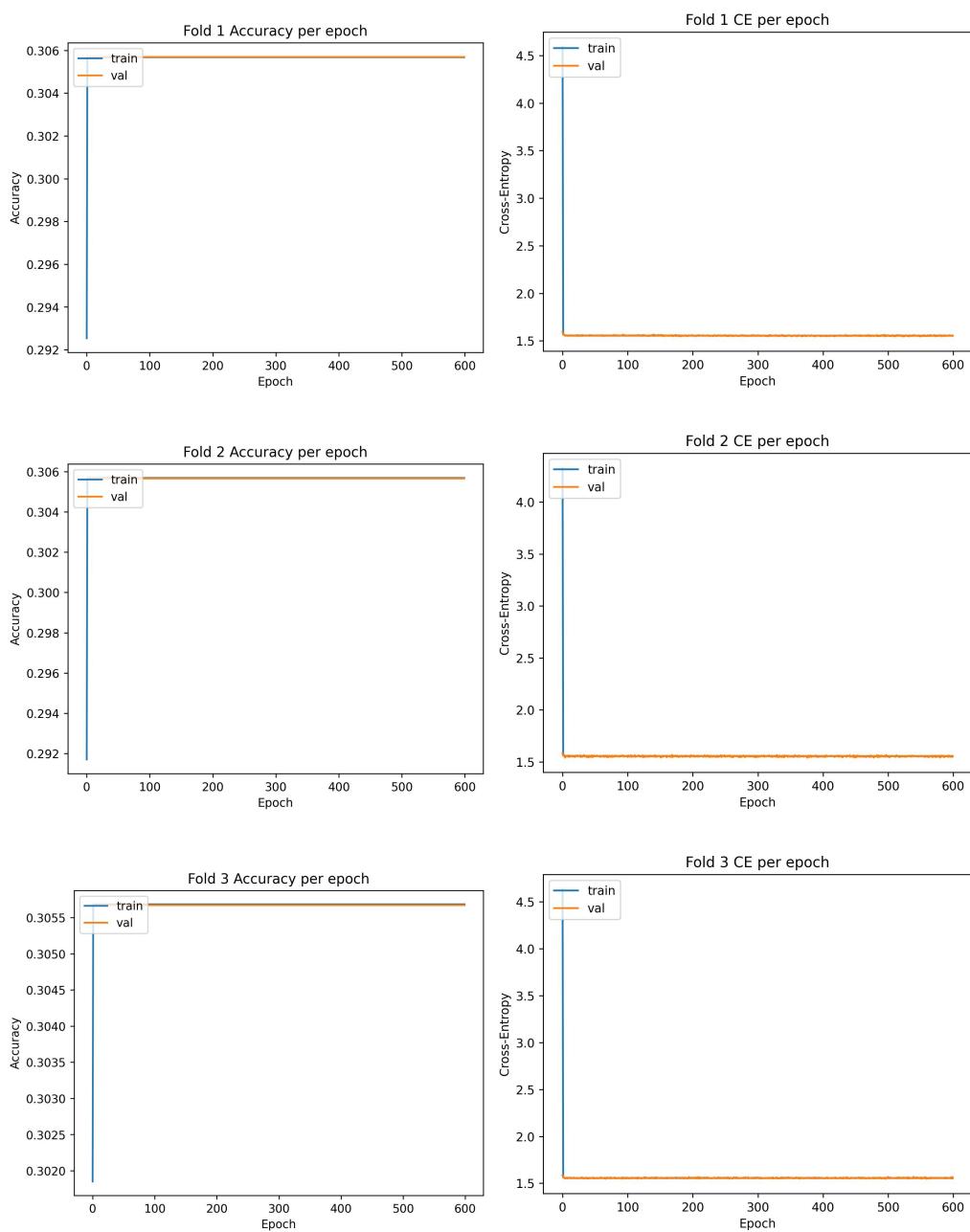


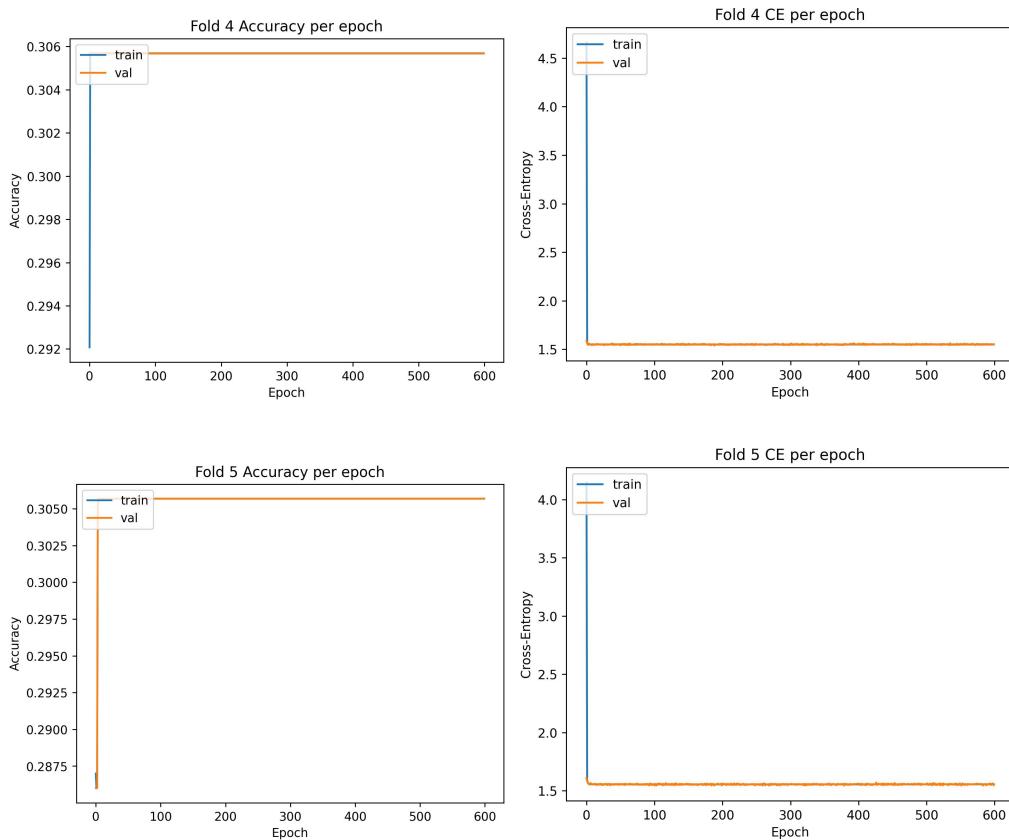
**r=0.5**





**r=0.9**





Παρατηρούμε ότι για όλες τις περιπτώσεις τόσο η ακρίβεια ταξινόμησης όσο και η συνάρτηση κόστους συγκλίνουν άμεσα σε μία τιμή (πολύ χειρότερη των προηγούμενων μετρήσεων) και παραμένουν στάσιμες σε αυτή. Αυτό πιθανώς οφείλεται σε υπερβολικά μεγάλη παράμετρο ομαλοποίησης για το συγκεκριμένο πρόβλημα. Ειδικά εφόσον προηγουμένως το μοντέλο μας δεν παρουσίαζε πρόβλημα στη γενικευτική του ικανότητα (overfitting), η προσθήκη ομαλοποίησης μάλλον χειροτερεύει την κατάσταση και οδηγεί σε underfitting.

\*Να σημειωθεί ότι για αυτά τα πειράματα δε χρησιμοποιήθηκε *early stopping* ώστε να δειχθεί η στασιμότητα των αποτελεσμάτων.

## Πηγές

Σ. Λυκοθανάσης & Ε. Γεωργόπουλος, *Υπολογιστική Νοημοσύνη Πανεπιστημιακές Παραδόσεις*,  
Πάτρα, Ελλάδα, 2014

S. Hoykin, *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*, Αθήνα, Ελλάδα: Εκδ. Παπασωτηρίου, 2010

T. A. Team and T. A. Team, "How, When, and Why Should You Normalize / Standardize / Rescale Your Data? – Towards AI – The Best of Tech, Science, and Engineering."  
<https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff>

"Compare the effect of different scalers on data with outliers – scikit-learn 0.20.3 documentation," *Scikit-learn.org*, 2018. [https://scikit-learn.org/stable/auto\\_examples/preprocessing/plot\\_all\\_scaling.html](https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html)

M. Riva, "Interpretation of Loss and Accuracy for a Machine Learning Model | Baeldung on Computer Science," *www.baeldung.com*, Jan. 19, 2021. <https://www.baeldung.com/cs/ml-loss-accuracy>

A. Kumar, "Mean Squared Error vs Cross entropy loss function," *Data Analytics*, Aug. 31, 2021. <https://vitalflux.com/mean-squared-error-vs-cross-entropy-loss-function/>

jcsladcik, "Configuring a Neural Network Output Layer," *Enthought, Inc.*, May 03, 2022. <https://www.enthought.com/blog/neural-network-output-layer/>.

S. Tiwari, "Activation functions in Neural Networks - GeeksforGeeks," *GeeksforGeeks*, Feb. 06, 2018. <https://www.geeksforgeeks.org/activation-functions-neural-networks/>

D. Goswami, "Comparison of Sigmoid, Tanh and ReLU Activation Functions," *AITUDE*, Aug. 19, 2020. <https://www.atitude.com/comparison-of-sigmoid-tanh-and-relu-activation-functions/>

J. Jordan, "Setting the learning rate of your neural network.," *Jeremy Jordan*, Mar. 02, 2018. <https://www.jeremyjordan.me/nn-learning-rate/>

Anuja Nagpal, "L1 and L2 Regularization Methods," *Medium*, Oct. 13, 2017. <https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c>

"Fighting Overfitting With L1 or L2 Regularization: Which One Is Better?," *neptune.ai*, Apr. 04, 2021. <https://neptune.ai/blog/fighting-overfitting-with-l1-or-l2-regularization>

R. Andreoni, "Regularization Techniques for Neural Networks," *Medium*, Aug. 24, 2022. <https://towardsdatascience.com/regularization-techniques-for-neural-networks-379f5b4c9ac3>

A. D'Agostino, "L1 vs L2 Regularization in Machine Learning: Differences, Advantages and How to Apply Them in...," *Medium*, Feb. 25, 2023. <https://towardsdatascience.com/l1-vs-l2-regularization-in-machine-learning-differences-advantages-and-how-to-apply-them-in-72eb12f102b5>