

# Εργαστηριακή Άσκηση Μέρος Α'

## Υπολογιστική Νοημοσύνη

Όνομα: Δήμτσας Ιωάννης    ΑΜ:1054423

Γλώσσα προγραμματισμού: Python

Άλλα εργαλεία: Sci-kit learn, Keras

Github: [link](#)

### Αναγνώριση χειρόγραφων χαρακτήρων με Χρήση Νευρωνικών Δικτύων

#### A1. Προεπεξεργασία και προετοιμασία δεδομένων

a.

- Κεντράρισμα: Αφαιρείται ο μέσος όρος των δεδομένων από όλα τα δεδομένα, με σκοπό ο μέσος όρος των δεδομένων να είναι 0.
- Κανονικοποίηση: Κλιμακώνει τα δεδομένα μεταξύ του 0 και 1. Για το συγκεκριμένο ζήτημα χρησιμοποιείται η *utils.normalize* του keras.
- Τυποποίηση: Θέτει τα δεδομένα γύρω από το 0 και τα κλιμακώνει με βάση την τυπική απόκλιση.

b. Διασταυρούμενη Επικύρωση (cross-validation): Για την υλοποίηση του cross-validation χρησιμοποιήθηκε η βιβλιοθήκη KFold του Sci-kit. Τα δεδομένα χωρίστηκαν σε 5 set train-test και η διαδικασία της μάθησης γίνεται 5 φορές.

#### A2. Επιλογή αρχιτεκτονικής

A) Η εκπαίδευση και η αξιολόγηση των μοντέλων θα γίνει με:

- **Cross-Entropy**: Χρησιμοποιείται για να ποσοτικοποιήσει την διαφορά ανάμεσα σε δύο κατανομές πιθανότητας. Δηλαδή το πόσο κοντά είναι στην

πραγματική κατανομή είναι το αποτέλεσμα που προβλέφθηκε. Όσο πιο μικρή είναι αυτή η ποσότητα, τόσο καλύτερο είναι το μοντέλο.

- **MSE**: Είναι το άθροισμα όλων των τετραγωνικών διαφορών μεταξύ της προβλεπόμενης και πραγματικής τιμής διαιρούμενο με το σύνολο των δεδομένων

Β) Θα χρειαστούμε μια **είσοδο** για κάθε pixel και εφόσον οι εικόνες είναι 28x28 θα χρειαστούμε **784 εισόδους**.

Γ) Θα χρειαστούμε **10 νευρώνες για την έξοδο**, όσο και το πλήθος των διαφορετικών ψηφίων.

Δ) Για το κρυφό επίπεδο του μοντέλου χρησιμοποιούμε την **relu** ως συνάρτηση ενεργοποίησης.

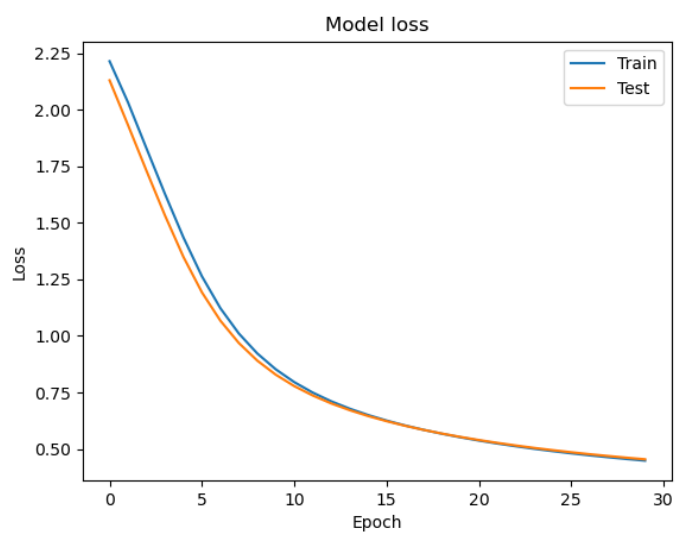
Ε) Για το επίπεδο εξόδο χρησιμοποιούμε την **softmax** ως συνάρτηση ενεργοποίησης, καθώς αυτή παράγει μία τιμή για κάθε κόμβο στο επίπεδο εξόδου, πράγμα το οποίο θέλουμε στην συγκεκριμένη περίπτωση. Δηλαδή να έχουμε 1 έξοδο για καθένα από τα 10 διαφορετικά πιθανά ψηφία.

Στ)

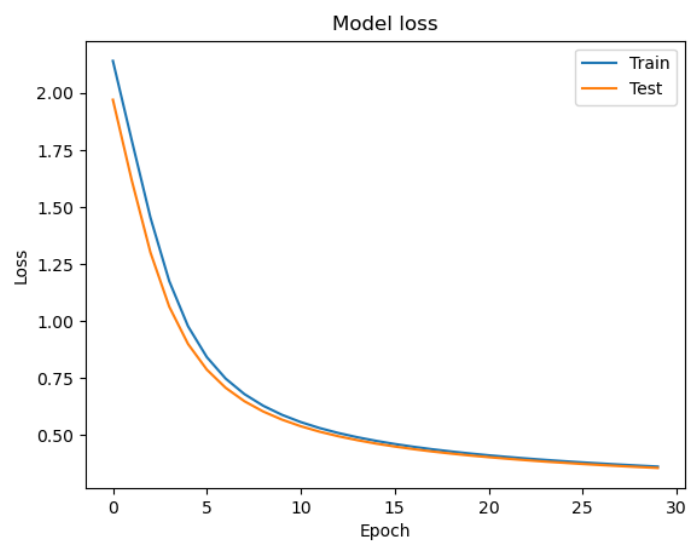
Παρατηρούμε ότι με 397 νευρώνες στο κρυφό επίπεδο πετυχαίνουμε καλή απόδοση. Δεν χρειάζεται παραπάνω γιατί μετά το δίκτυο γίνεται όλο και πιο αργό.

Αριθμός νευρώνων στο κρυφό επίπεδο	CE loss	MSE
$H_1 = O = 10$	0.45	27.38
$H_1 = (I + O)/2 = 397$	0.35	27.36
$H_1 = I = 784$	0.35	27.40

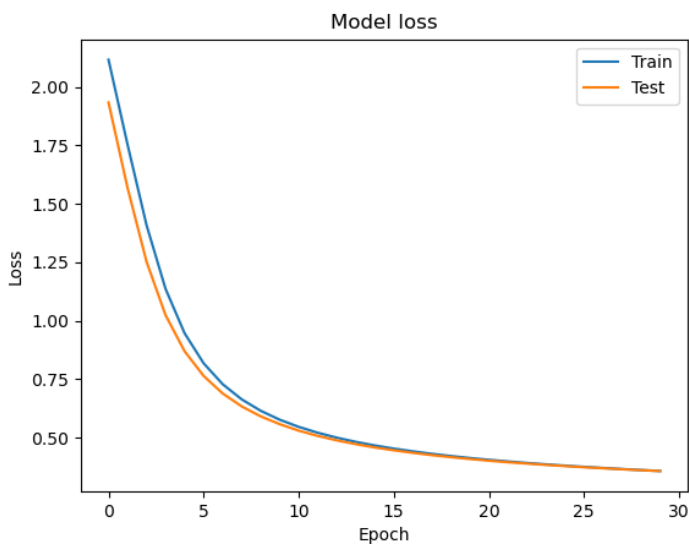
Για  $H_1=10$



$\Gamma\alpha H_1=397$



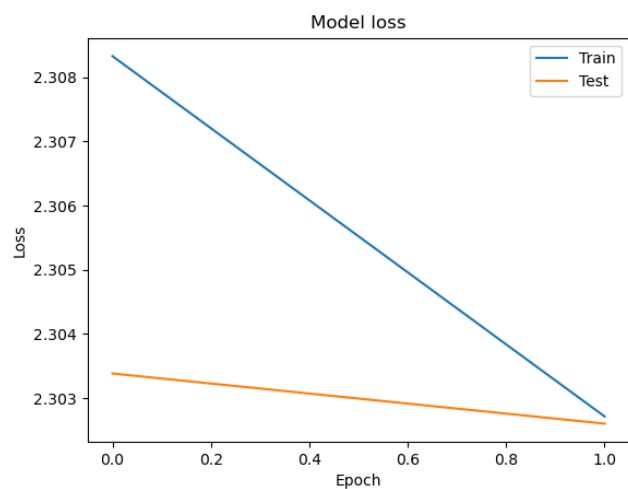
$\Gamma\alpha H_1 = 784$



Z)

Αριθμός νευρώνων στο κρυφό επίπεδο	<i>CE loss</i>	<i>MSE</i>
$H2=10$	0.11	27.43
$H2=397$	0.085	27.25
$H2=784$	0.10	26.94

Η) Το Early Stopping εφαρμόζεται όταν υπάρχει σημείο της γραφικής παράστασης του train που είναι πάνω από το validation. Στην δική μας περίπτωση αυτό συμβαίνει στο δεύτερο epoch το οποίο είναι αρκετά νωρίς, οπότε δεν είναι χρήσιμο.



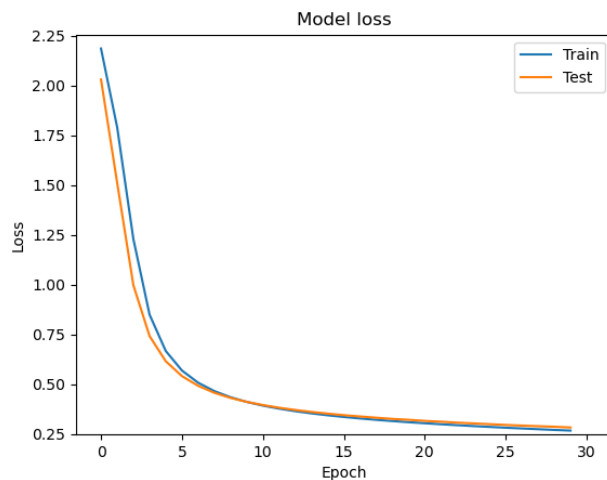
### A3. Μεταβολές στον ρυθμό εκπαίδευσης και σταθεράς ορμής

Η ορμή βοηθάει στην επιτάχυνση της μάθησης, οδηγώντας σε ταχύτερη σύγκλιση. Για να εξασφαλιστεί η σύγκλιση, η σταθερά ορμής πρέπει να είναι  $m < 1$ , ώστε οι παλιότερες μεταβολές να βαρύνουν λιγότερο.

$\eta$	$m$	CE loss	MSE
0.001	0.2	0.26	27.37
0.001	0.6	0.20	27.009
0.05	0.6	0.09	27.23
0.1	0.6	0.10	26.83

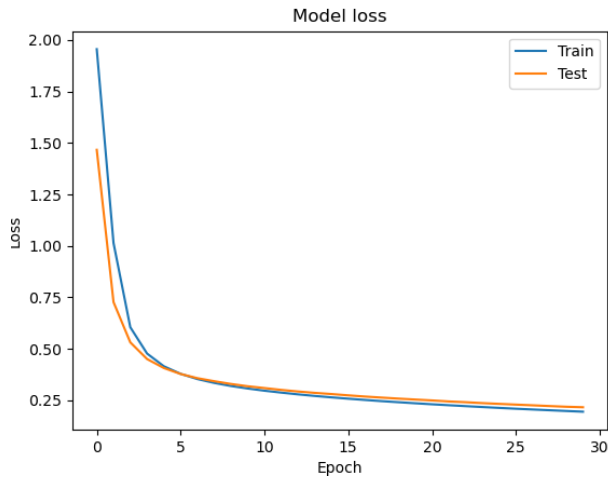
- Για  $\eta=0.001$  και  $m=0.2$

Το μοντέλο με αυτές τις υπερπαραμέτρους δεν έχει καλή απόδοση. Παρατηρούμε ότι έχει μεγάλο σφάλμα.



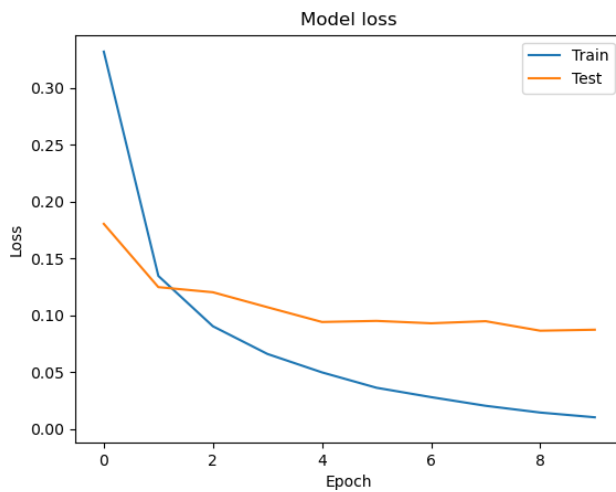
- Για  $\eta=0.001$  και  $m=0.6$

Εδώ με την αύξηση της ορμής παρατηρούμε κάπως καλύτερο σφάλμα αλλά όχι αρκετά ικανοποιητικό.



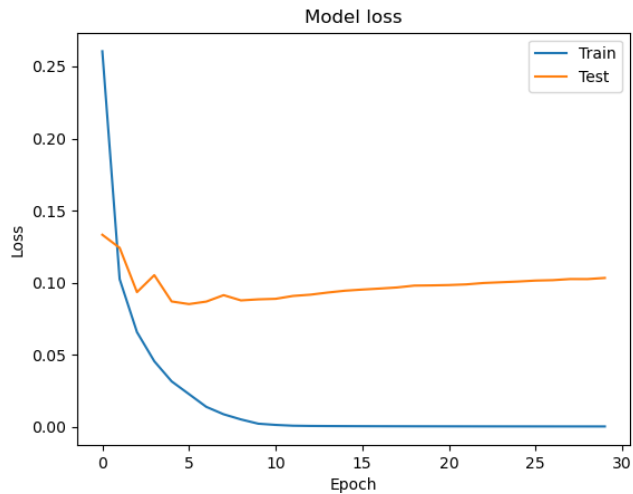
- Για  $\eta=0.05$  και  $m=0.6$

Με την αύξηση του ρυθμού μάθησης πετυχαίνουμε μεγάλη μείωση σφάλματος, όμως δεν έχουμε καλή σύγκλιση του test loss. Παρατηρούμε ότι η γραφική παράσταση του test ανεβαίνει πάνω από του train το οποίο δεν είναι καλό



- Για  $\eta=0.1$  και  $m=0.6$

Το ίδιο συμβαίνει και με αυτές τις υπερπαραμέτρους.

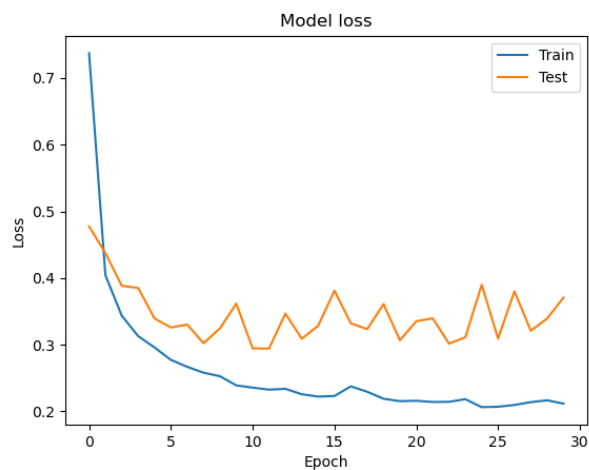


#### A4. Ομαλοποίηση

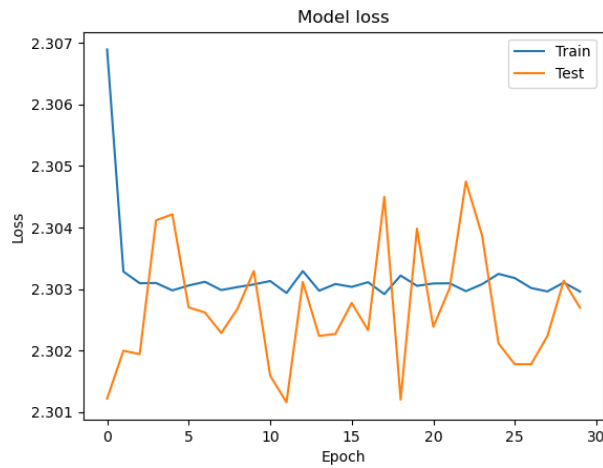
Με την ομαλοποίηση αποθαρρύνεται η εκμάθηση ενός περίπλοκου μοντέλου για την αποφυγή της υπερπροσαρμογής. Εδώ χρησιμοποιείται η L2 ομαλοποίηση ή αλλιώς Ridge, η οποία προσθέτει τετραγωνική τιμή στον συντελεστή  $\beta$  της συνάρτησης κόστους.

Συντελεστής Φθοράς	CE loss	MSE
0.1	0.21	27.46
0.5	2.3	27.62
0.9	2.3	27.24

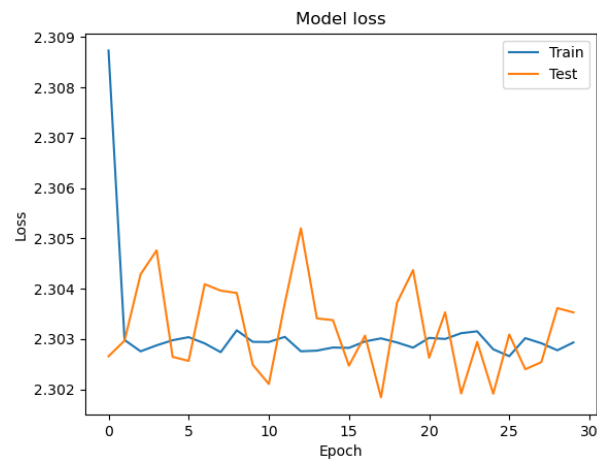
Για  $r=0.1$



Για  $r=0.5$



Για  $r=0.9$



## A5. CNN

Για την αρχιτεκτονική του μοντέλου χρησιμοποιήθηκε ένα conv2d layer με `kernel_size(3,3)`, ακολουθούμενο από ένα MaxPooling2D layer με `pool_size(2,2)`. Στη συνέχεια προστίθεται ένα Flatten layer για να “επιπεδοποιηθούν” τα χαρακτηριστικά που θα φτάσουν στον classifier. Τέλος προστέθηκαν δύο Dense layers, ένα κρυφό με relu συνάρτηση ενεργοποίησης και το output με softmax συνάρτηση ενεργοποίησης, καθώς και ένα Dropout layer το οποίο θέτει τυχαία



input units στο 0 με rate 0.2 προκειμένου να αποφευχθεί το overfitting, ενώ τα υπόλοιπα input units θέτονται στο  $1/(1-0.2)$ .

Ως optimizer χρησιμοποιήθηκε ο SGD με learning\_rate=0.1 και momentum=0.6.

Τέλος, έγινε χρήση του 5-fold Cross-Validation με κάθε test set να είναι το 20% επί του συνολικού train dataset, ενώ το μοντέλο προπονείται για 10 εποχές.

Τα τελικά αποτελέσματα είναι 0.08 για το Cross-Entropy και 27.34 για το MSE, αποτελέσματα τα οποία δεν φαίνεται να διαφέρουν σε σχέση με το προηγούμενο μοντέλο. Το accuracy του μοντέλου είναι 98.18%.