**Υπολογιστική Νοημοσύνη**

****

**Α’ Μέρος**

**Γιαννακόπουλος Στέφανος (1054365)**

**Link Github:** **https://github.com/stefanos98**

**Ακ. Έτος 2020-2021**

**Α1. Προεπεξεργασία και Προετοιμασία δεδομένων**

**α)**

* Το **κεντράρισμα (centering)** και η κλιμάκωση (scaling) πραγματοποιούνται ανεξάρτητα σε κάθε χαρακτηριστικό, υπολογίζοντας τα σχετικά στατιστικά στοιχεία των δειγμάτων στο σύνολο εκπαίδευσης. Η μέση τιμή και η τυπική απόκλιση αποθηκεύονται στη συνέχεια για να χρησιμοποιηθούν σε μεταγενέστερα δεδομένα.
* Όπως γνωρίζουμε ορισμένοι αλγόριθμοι μηχανικής μάθησης δεν μπορούν να εκτελέσουν με επιτυχία συγκεκριμένες λειτουργίες χωρίς **κανονικοποίηση (normalization)** των τιμών εισόδου. Για παράδειγμα, η πλειοψηφία των ταξινομητών υπολογίζει την απόσταση μεταξύ δύο σημείων κατά την ευκλείδεια απόσταση. Εάν κάποιο από τα features έχει ένα ευρύ φάσμα τιμών, η απόσταση θα καθορίζεται από αυτό το συγκεκριμένο χαρακτηριστικό. Επομένως, το εύρος όλων των χαρακτηριστικών 81 θα πρέπει να κανονικοποιηθεί έτσι ώστε κάθε χαρακτηριστικό να συμβάλλει περίπου αναλογικά στην τελική απόσταση.
* Η **τυποποίηση (standardization)** ενός συνόλου δεδομένων είναι μια κοινή απαίτηση για πολλούς classifiers μηχανικής μάθησης. Eνδέχεται να μη λειτουργούν σωστά εάν τα μεμονωμένα χαρακτηριστικά δεν παρουσιάζουν ομοιότητες με τα τυπικά κανονικά κατανεμημένα δεδομένα (π.χ. Gaussian κατανομή με μηδενική μέση τιμή και διακύμανση).

Θα πραγματοποιήσουμε normalization στις τιμές εισόδου ώστε να μεταμορφωθούν σε συγκεκριμένο εύρος τιμών, κάτι το οποίο θα μας βοηθήσει αργότερα στα μοντέλα ως προς τη διαδικασία της πρόβλεψης. Αν δεν εφαρμοστεί η συγκεκριμένη τεχνική θα υπάρξει πρόβλημα πιθανότατα όπως για παράδειγμα μικρό εύρος τιμών, μεγάλο εύρος τιμών, αραιές τιμές ως προς το εύρος τους. Επίσης κρίνεται απαραίτητη η κανονικοποίηση όλων των χαρακτήρων διότι θα πρέπει να έχουν ένα προκαθορισμένο μέγεθος καθώς και οι νευρώνες του επιπέδου εισόδου έχουν επίσης προκαθορισμένο μέγεθος.

Κεντράρισμα δε θα χρειαστεί να εφαρμοστεί καθώς οι εικόνες είναι στάνταρ και δεν χρειάζονται κάποιου είδους πλαισιοποίηση ώστε να αναγνωριστούν. Το ίδιο και η τυποποίηση.

**β)**

Η μέθοδος CV αποφεύγει το πρόβλημα επικάλυψης των συνόλων εκπαίδευσης

Πρώτο βήμα: διάσπαση των δεδομένων σε 5 ισομεγέθη υποσύνολα  
 Δεύτερο βήμα: χρήση κάθε υποσυνόλου διαδοχικά για έλεγχο, τα υπόλοιπα δεδομένα (4 υποσύνολα) για εκπαίδευση.  
Ο μέσος όρος των 5 εκτιμήσεων για το σφάλμα αποτελεί μία αξιόπιστη εκτίμηση.

**Α2. Επιλογή αρχιτεκτονικής**

**α)** Θα μπορούσαμε να πούμε με απλά λόγια πως το Cross-Entropy (CE) το χρησιμοποιούμε για την εκπαίδευση του μοντέλου μας και για να βγάλουμε μια απλή και γρήγορη εκτίμηση για το μοντέλο μας. Επίσης το Μέσο Τετραγωνικό Σφάλμα (MSE) το χρησιμοποιούμε για τον έλεγχο του μοντέλου μας. Το μέσο τετραγωνικό σφάλμα μετράει την απόκλιση του σφάλματος υψωμένο στο τετράγωνο για όλο το σύνολο της πρόβλεψης. Εξαιτίας του τετραγώνου τα σφάλματα αντίθετου πρόσημου δεν ακυρώνονται μεταξύ τους. Ένα άλλο χαρακτηριστικό του MSE είναι ότι «τιμωρεί» τα μεγάλα σφάλματα περισσότερο από τα μικρά. Σημειώνεται ωστόσo, ότι δεν είναι εύκoλα ερμηνεύσιμo, αλλά θεωρείται καλό μέτρo όσον αναφορά τα σφάλματα.

**β)** Στην εικόνα του παραδείγματος αποτυπώνονται 100 χειρόγραφα ψηφία, με το κάθε ένα να έχει ανάλυση 28X28=784 pixels = Αντιστοιχεί ένας νευρώνας για κάθε pixel του τεμαχισμένου χαρακτήρα. Οπότε έχουμε 28X28=784 pixels=**784 νευρώνες εισόδου.**

**γ)** Οι **νευρώνες εξόδου** είναι τόσοι όσα είναι τα επιθυμητά πρότυπα διαχωρισμού. Στην περίπτωσή μας έχει **10** νευρώνες στο επίπεδο εξόδου, όσα δηλαδή και τα αριθμητικά ψηφία προς διαχωρισμό.

**δ)** Είναι γεγονός ότι η **ReLu** αποδίδει συχνά καλύτερα από άλλες συναρτήσεις ενεργοποίησης για κρυφά επίπεδα. Ο βασικός λόγος της αυξημένης απόδοσης οφείλεται στο γεγονός ότι η ReLU είναι μια γραμμική συνάρτηση μη κορεσμού. Ο κορεσμός είναι το μεγαλύτερο πρόβλημα των logistic ή tanh. Σε αντίθεση λοιπόν με αυτές, η ReLU δεν κορέζεται στο -1, 0 ή 1. Οι πιο πρόσφατες έρευνες αναφέρουν ότι τα κρυμμένα επίπεδα του ΤΝΔ πρέπει να χρησιμοποιούν την ενεργοποίηση του ReLU.

**ε)** Η συνάρτηση **Softmax** αποτελεί μία γενίκευση της σιγμοειδούς συνάρτησης για μετασχηματισμό τιμών σε πιθανότητες, όταν υπάρχουν παραπάνω από δύο κατηγορίες (κλάσεις). Εφαρμόζεται στο δίανυσμα που προκύπτει από το στρώμα εξόδου και μετασχηματίζει τις τιμές σε πιθανότητες οι οποίες αθροίζουν στη μονάδα. Αν το διάνυσμα εξόδου είναι μήκους 2, η softmax ταυτίζεται με τη σιγμοειδή συνάρτηση. Η συνάρτηση softmax ορίζεται ως εξής:

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Η softmax χρησιμοποιείται ευρέως σε προβλήματα κατηγοριοποίησης, όπου μόνο μία κατηγορία είναι σωστή.

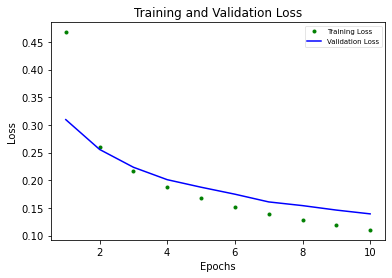
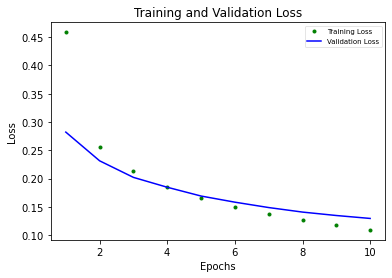
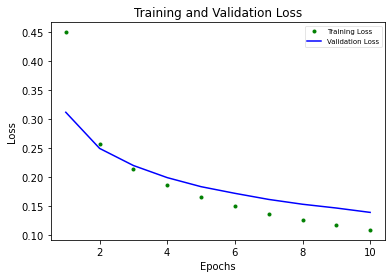
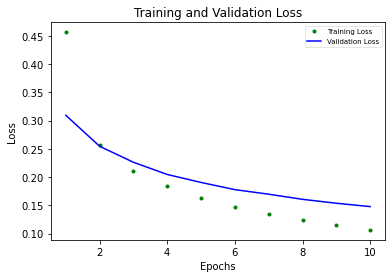
**\*Σε κάθε περίπτωση χρησιμοποιήθηκαν 10 εποχές**

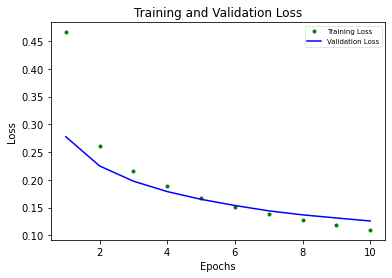
**στ)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Αριθμός νευρώνων στο κρυφό επίπεδο** | **CE loss** | **Validation CE loss** | **ΜSE loss** | **Validation MSE loss** |
| Η1 = Ο=10 | 0.2652 | 0.2518 | 1.9227 | 1.8406 |
| Η1 = (Ι+Ο)/2=397 | 0.1137 | 0.1314 | 0.6215 | 0.7515 |
| Η1 = Ι+Ο=794 | 0.1139 | 0.1259 | 0.5604 | 0.7490 |

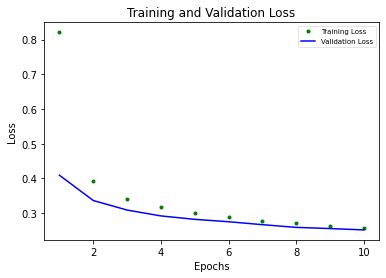
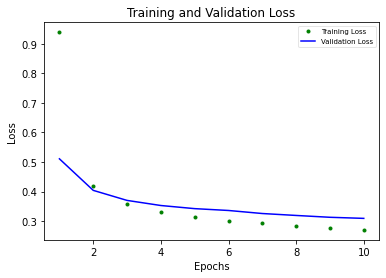
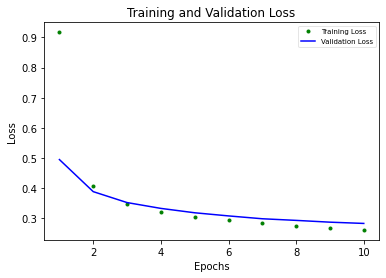
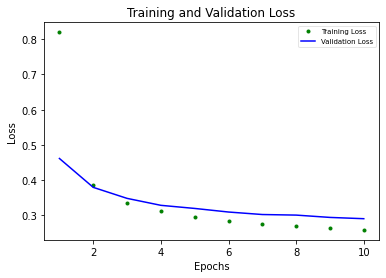
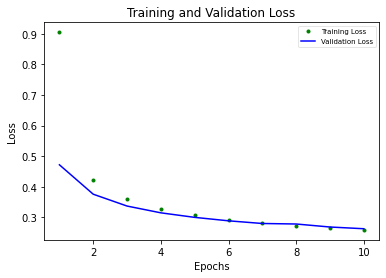
* Βάσει των πειραμάτων που πραγματοποίησα παρατηρώ ότι καλύτερη είναι η συνάρτηση κόστους CE με αριθμό κόμβων στο κρυφό επίπεδο H1=397

\***CE\_loss, H1=10**

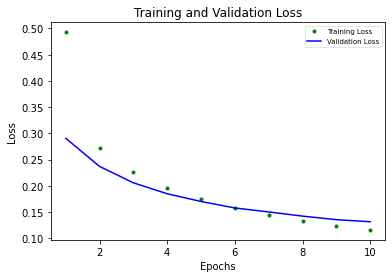
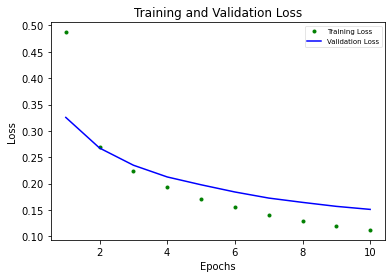
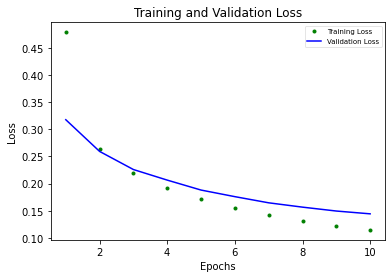
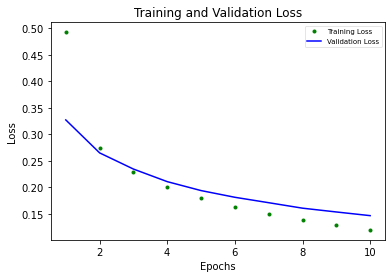
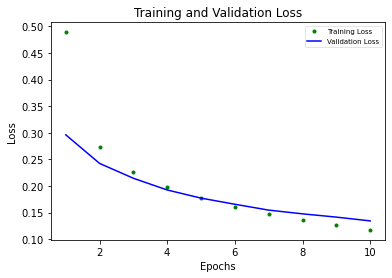
0123

4

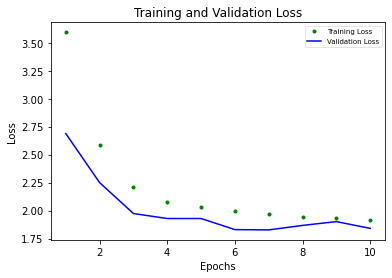
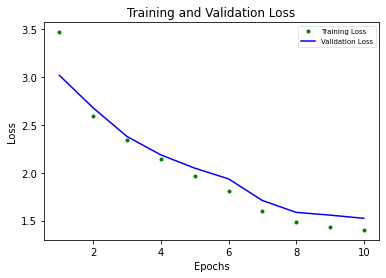
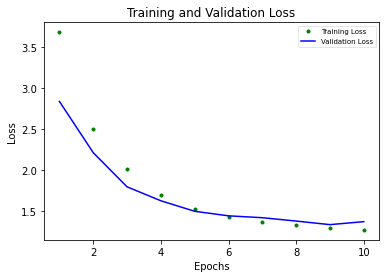
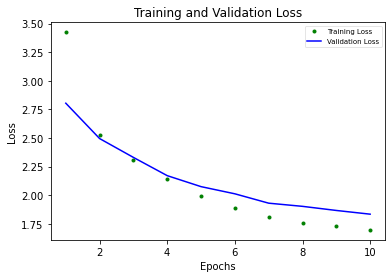
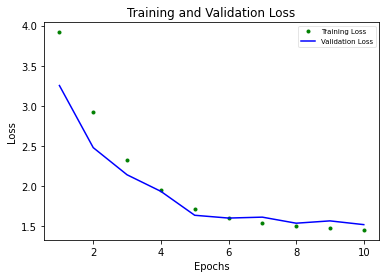
\***CE\_loss, H1=397**

****

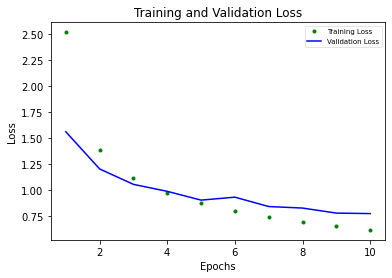
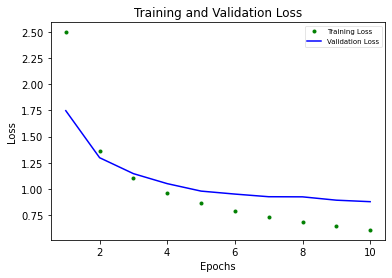
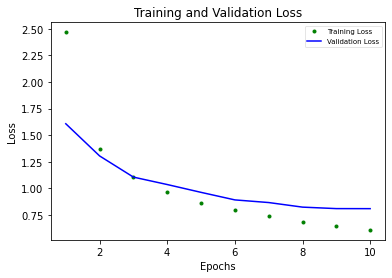
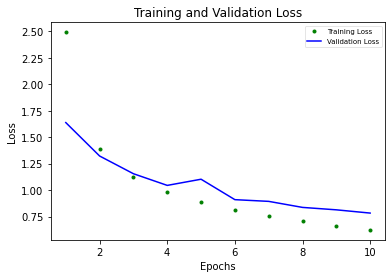
\***CE\_loss, H1=794**

****

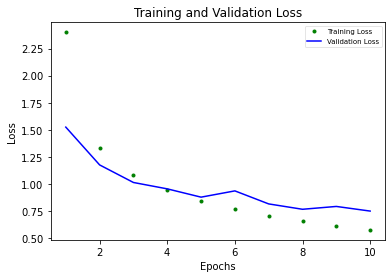
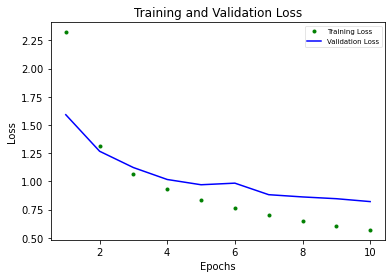
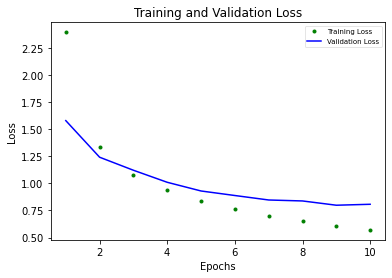
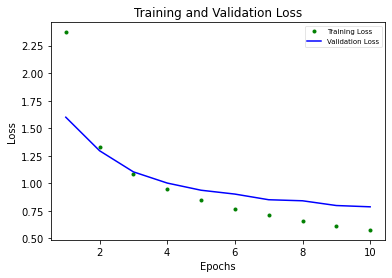
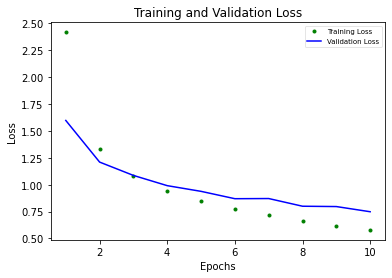
\***MSE, H1=10**

****

\***MSE, H1=397**

****

\***MSE, H1=794**

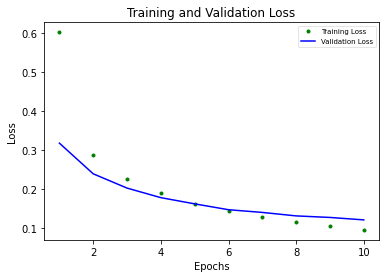
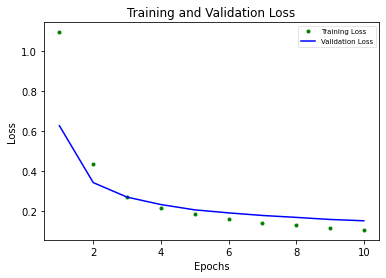
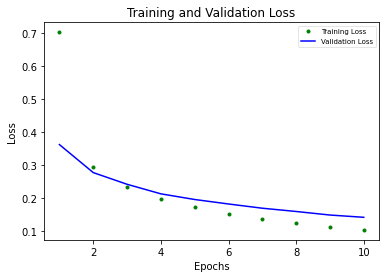
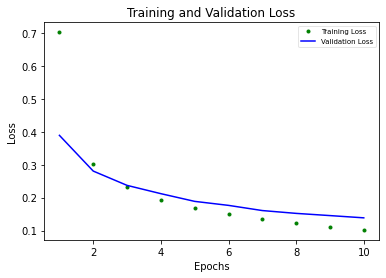
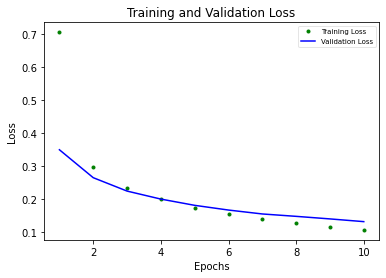
****

**ζ)** Η1=10

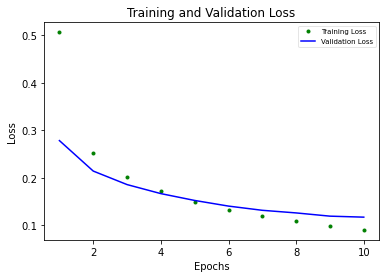
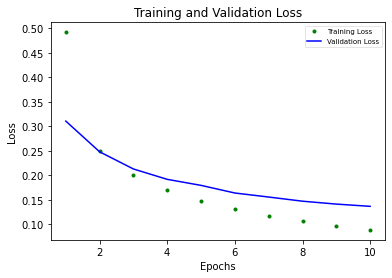
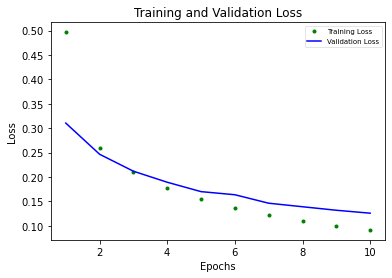
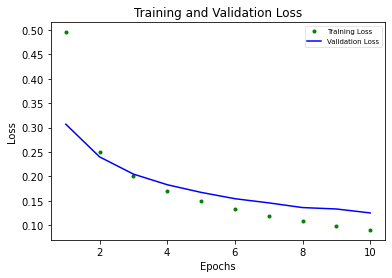
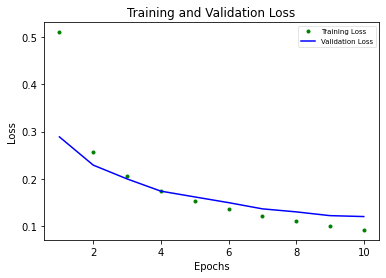
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Αριθμός νευρώνων στο κρυφό επίπεδο** | **CE loss** | **Validation CE loss** | **ΜSE** | **Validation MSE loss** |
| H2=10 | 0.0970 | 0.1217 | 0.4251 | 0.7557 |
| H2=397 | 0.0834 | 0.1179 | 0.3671 | 0.6024 |
| H2=794 | 0.0934 | 0.1192 | 0.3713 | 0.5951 |

Βάσει των πειραμάτων που πραγματοποίησα, παρατήρησα πως η απόδοση του νευρωνικού δικτύου είναι βέλτιστη όταν ο αριθμός των κόμβων στο δεύτερο κρυφό επίπεδο είναι ίσος με αυτόν στο πρώτο κρυφό επίπεδο. Το πόσο καλύτερος φαίνεται στα παραπάνω πινακάκια που έχουμε καταγράψει τις τιμές σφάλματος σε κάθε συνάρτηση κόστους στον αντίστοιχο αριθμό κόμβων. Τελικά, παρατηρούμε πως καλύτερη συνάρτηση κόστους είναι η Cross-Entropy (CE) loss και βέλτιστος αριθμός κόμβων στα κρυφά επίπεδα είναι H1=H2=397, δηλαδή προτιμητέο είναι ο αριθμός κόμβων στα 2 κρυφά επίπεδα να είναι ίδιος.

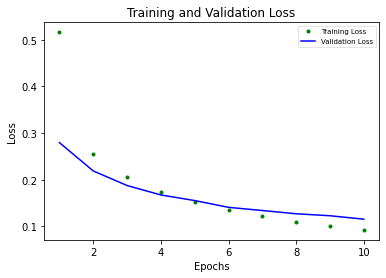
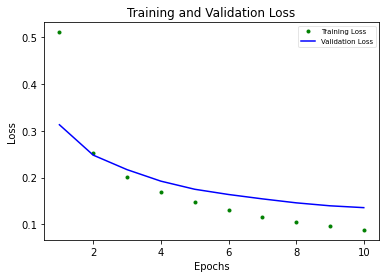
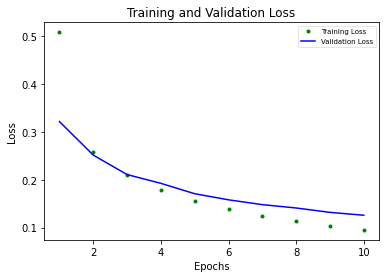
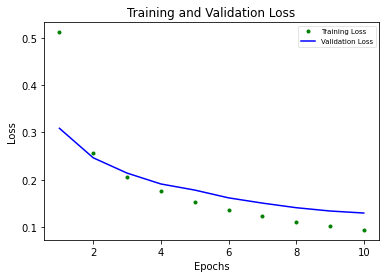
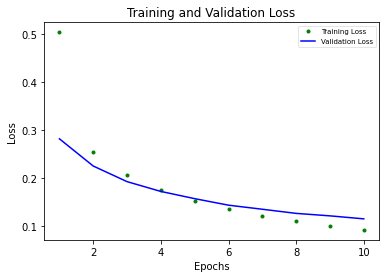
**\*CE\_loss, H1=397, H2=10**

****

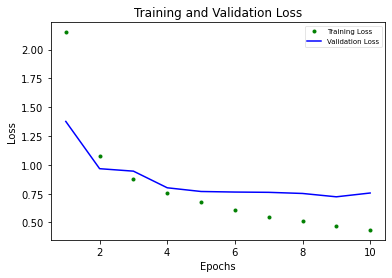
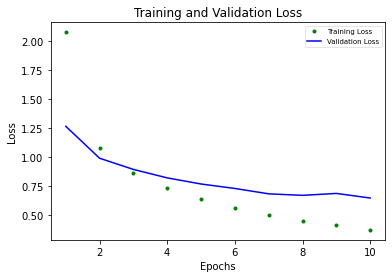
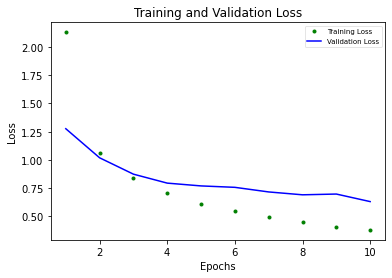
**\*CE\_loss, H1=397, H2=397**

****

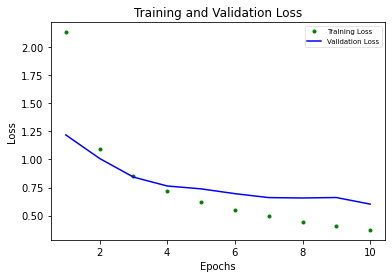
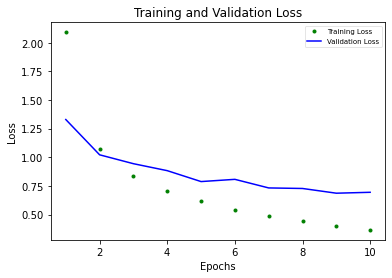
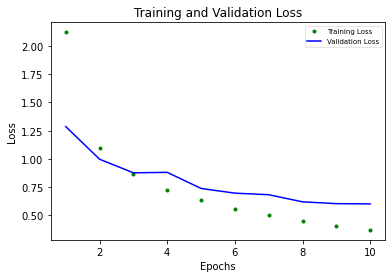
**\*CE\_loss, H1=397, H2=794**

****

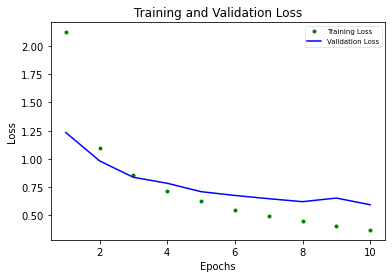
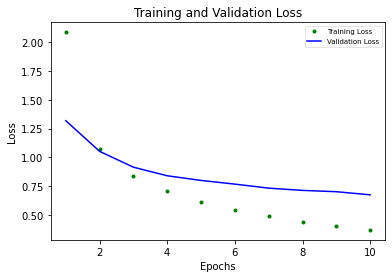
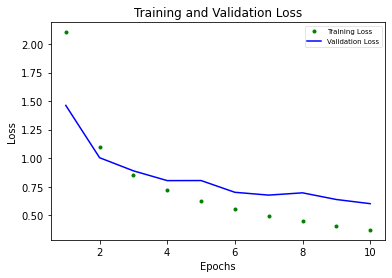
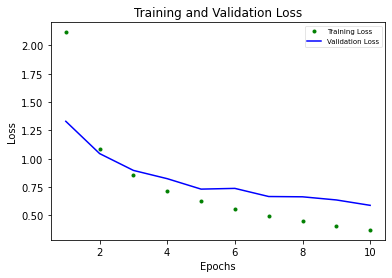
**\*MSE, H1=397, H2=10**

****

**\*MSE, H1=397, H2=397**

****

**\*MSE, H1=397, H2=794**

****

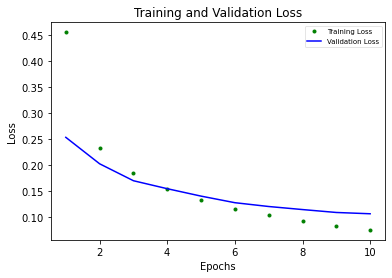
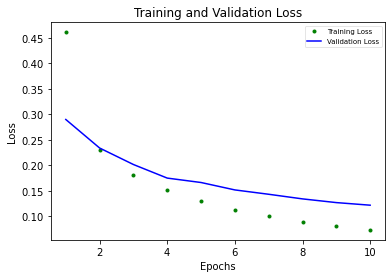
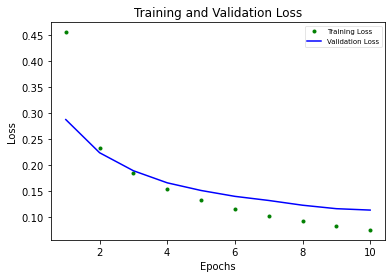
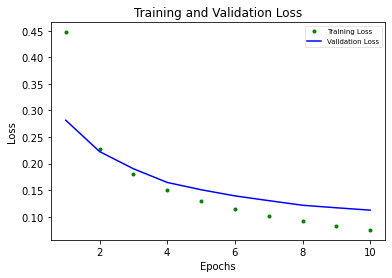
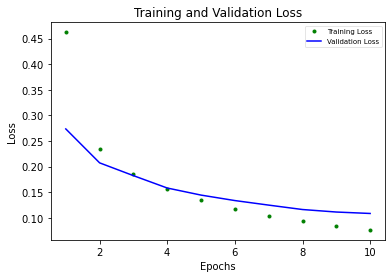
η) Ως κριτήριο τερματισμού έχει χρησιμοποιηθεί το Validation loss ώστε το κριτήριο να είναι πιο αυστηρό και ακριβές. Χρησιμοποιείται επίσης η τεχνική του πρόωρου σταματήματος.(patience = 30)

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

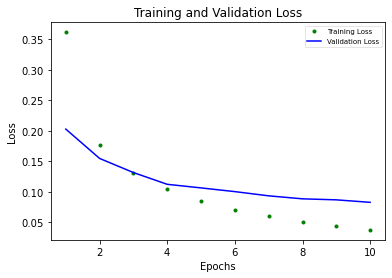
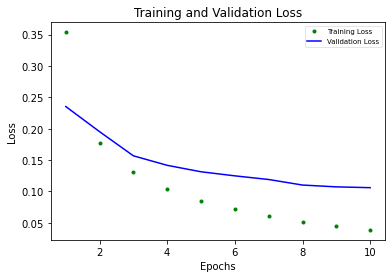
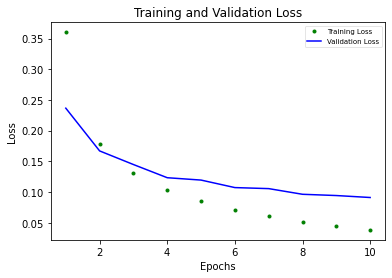
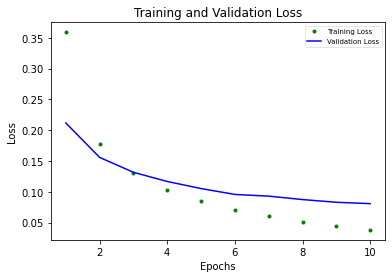
**Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα που πήραμε από κάθε περίπτωση και παρατηρούμε πως για η=0,001 και m=0.6 παίρνουμε καλύτερες επιδόσεις.** **Το CNN πρέπει να είναι σε θέση να προσαρμόζει τον ρυθμό εκπαίδευσής του, έτσι ώστε να μην ολοκληρώνει την εκπαίδευσή του πολύ γρήγορα και αποκτήσει ανολοκλήρωτη “γνώση” του περιβάλλοντος του και παράλληλα να μην καθυστερεί την ολοκλήρωση της εκπαίδευσής του εάν αυτό δεν είναι απαραίτητο. Για το σκοπό αυτό χρησιμοποιείται η momentum και η τιμή της δεν ξεπερνα την μονάδα (m<1).**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **η** | **m** | **CE loss** | **Validation CE loss** | **MSE** | **Validation MSE loss** |
| 0.001 | 0.2 | 0.0748 | 0.1069 | 0.3183 | 0.5675 |
| 0.001 | 0.6 | 0.0370 | 0.0829 | 0.2031 | 0.5093 |
| 0.05 | 0.6 | 0.1769 | 0.3646 | nan | nan |
| 0.1 | 0.6 | nan | nan | nan | nan |

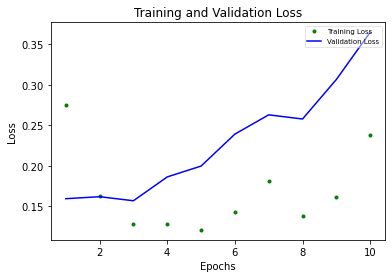
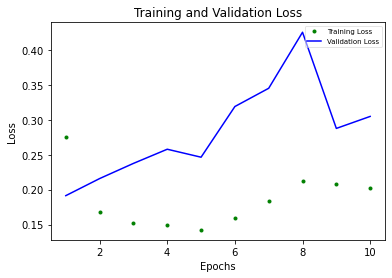
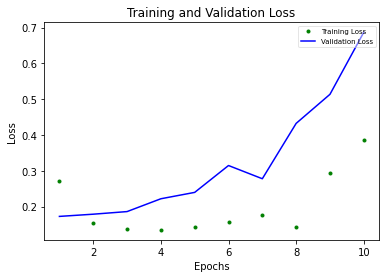
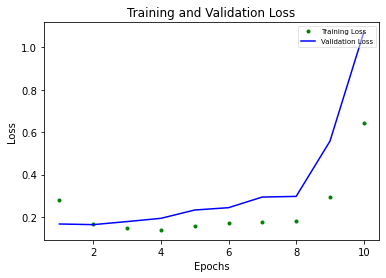
**\*CE, H1=397, H2=397, η=0.001, m=0.2**

****

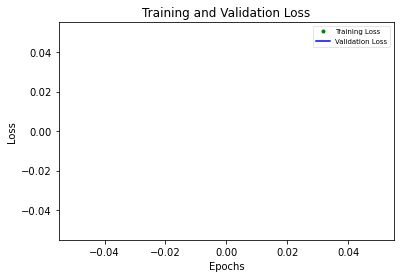
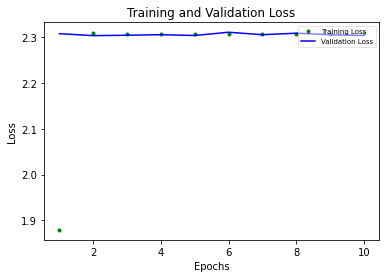
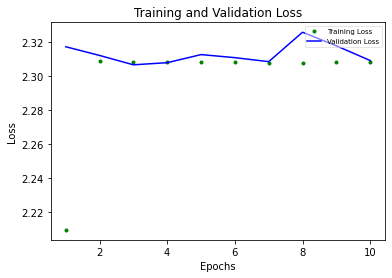
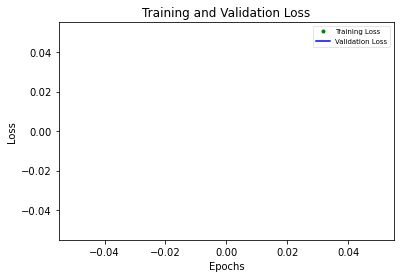
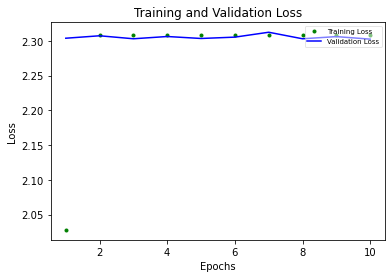
**\*CE, H1=397, H2=397, η=0.001, m=0.6**

****

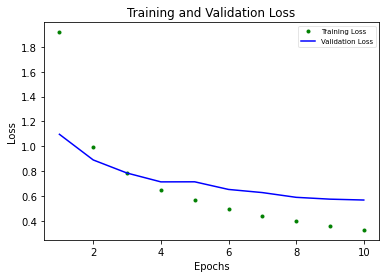
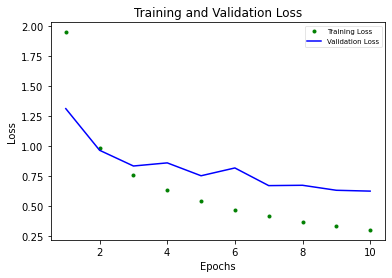
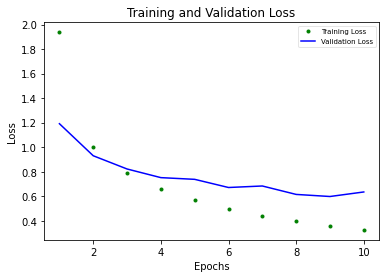
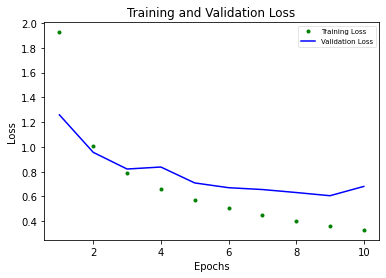
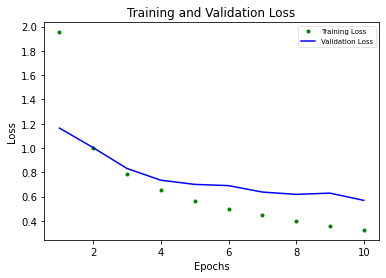
**\*CE, H1=397, H2=397, η=0.05, m=0.6**

****

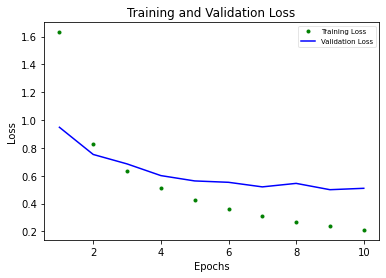
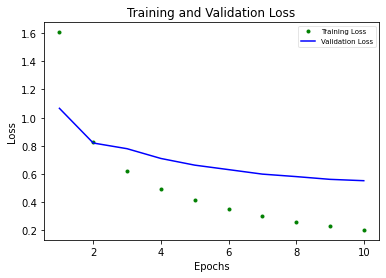
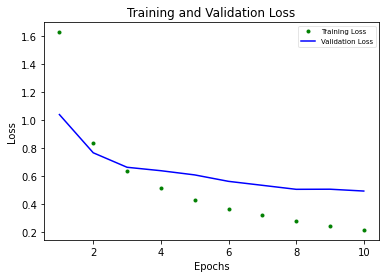
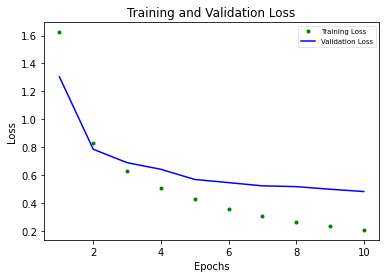
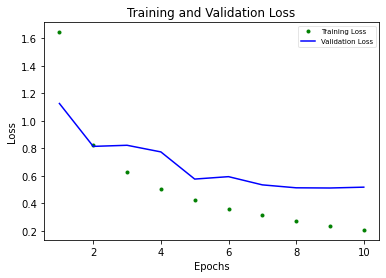
**\*CE, H1=397, H2=397, η=0.1, m=0.6**

****

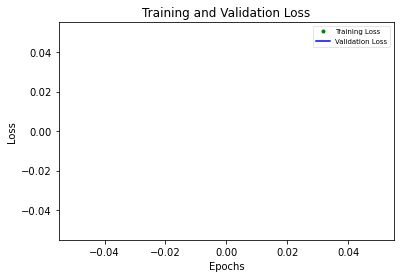
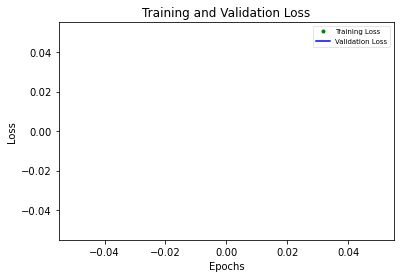
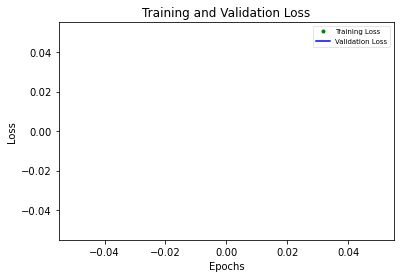
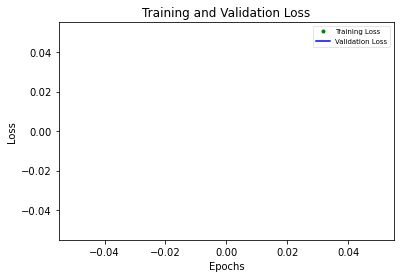
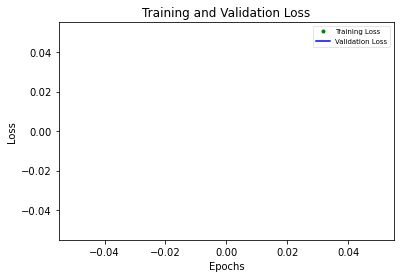
**\*MSE, H1=397, H2=397, η=0.001, m=0.2**

****

**\*MSE, H1=397, H2=397, η=0.001, m=0.6**

****

**\*MSE, H1=397, H2=397, η=0.05, m=0.6**

****

**\*MSE, H1=397, H2=397, η=0.1, m=0.6**

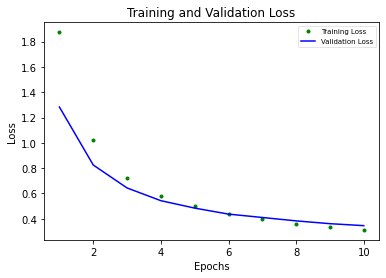
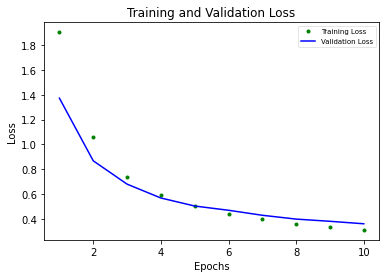
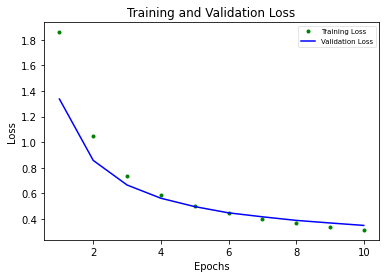
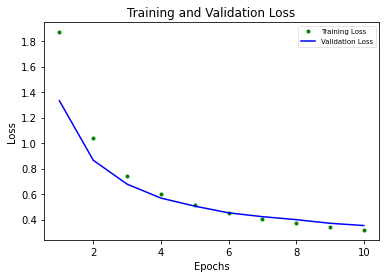
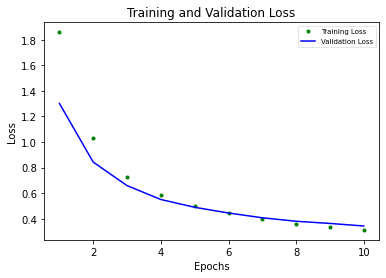
**Ομοίως με την πάνω περίπτωση.**

**Α4. Ομαλοποίηση**

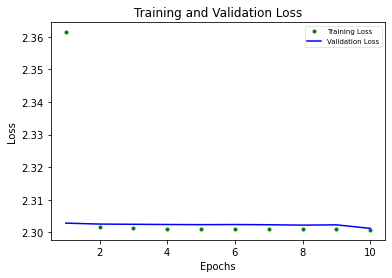
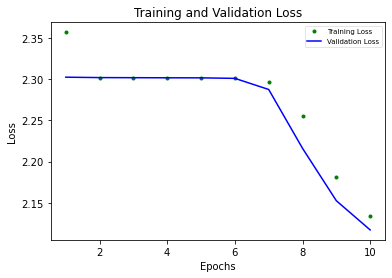
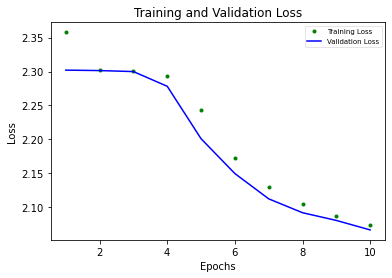
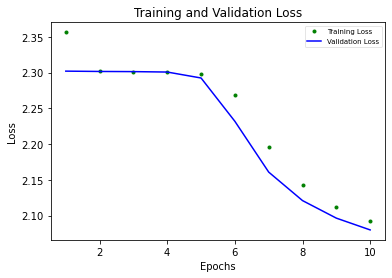
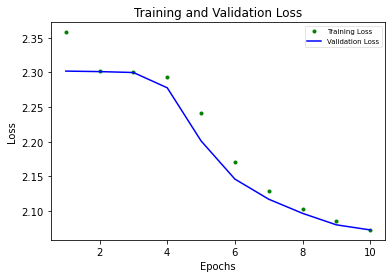
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Συντελεστής φθοράς (r)** | **CE loss** | **Validation CE loss** | **MSE** | **Validation MSE loss** |
| **0.1** | 0.3158 | 0.3450 | 0.3254 | 0.6348 |
| **0.5** | 2.3012 | 2.3012 | 0.5563 | 0.8769 |
| **0.9** | 2.3009 | 2.3025 | 0.7958 | 1.0713 |

Όπως βλέπουμε δεν επιδρά θετικά η μέθοδος στην γενικευτική ικανότητα του δικτύου καθώς για τις καλύτερες τιμές του προηγούμενου ερωτήματος (η=0.001, m=0.6) παρατηρούμε αύξηση σφάλματος.

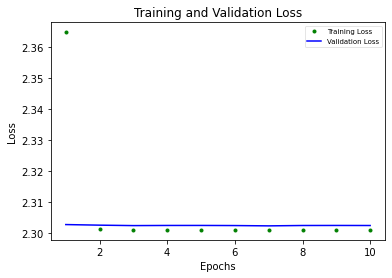
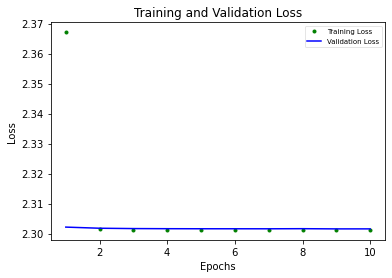
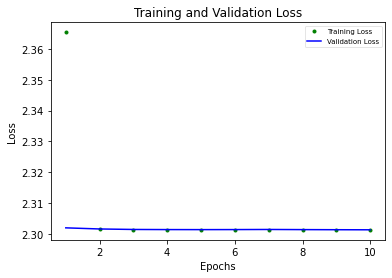
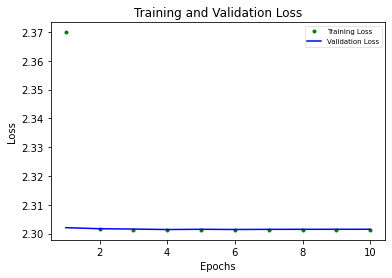
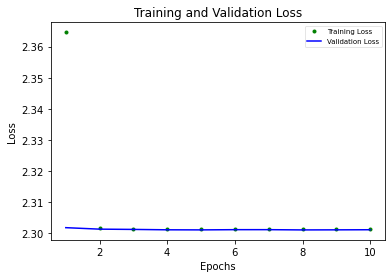
**\*CE\_loss, r=0.1**

****

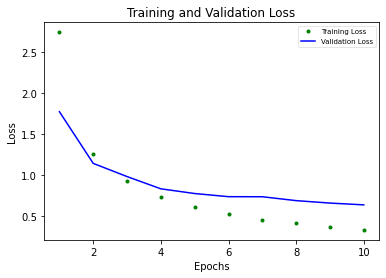
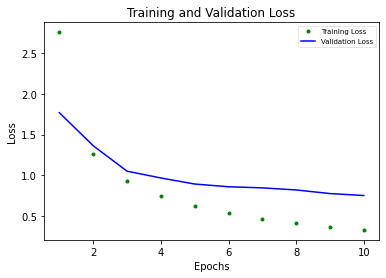
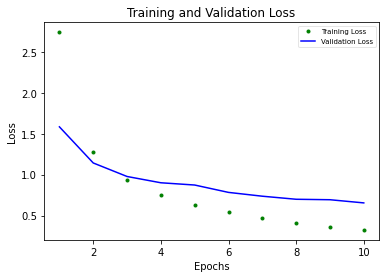
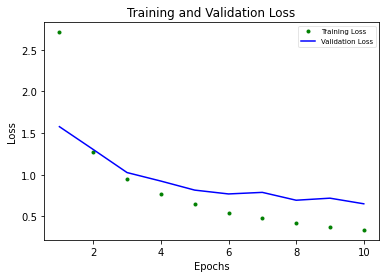
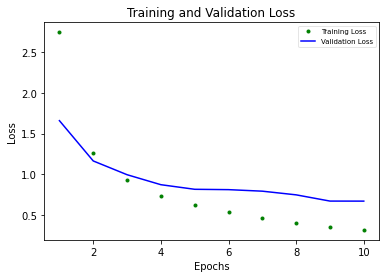
**\*CE\_loss, r=0.5**

****

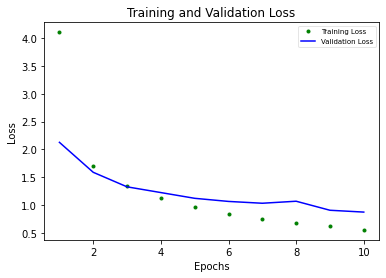
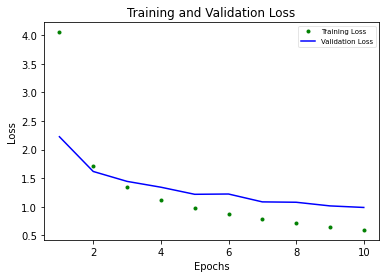
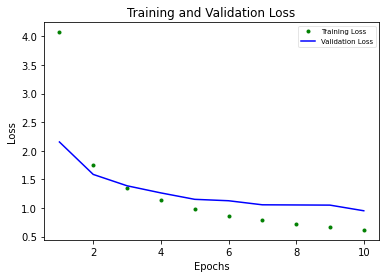
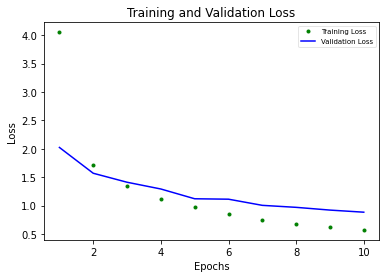
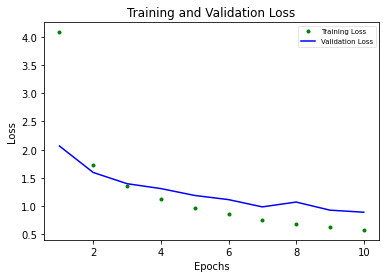
**\*CE\_loss, r=0.9**

****

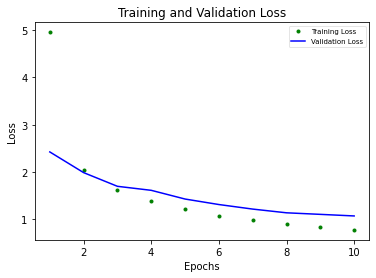
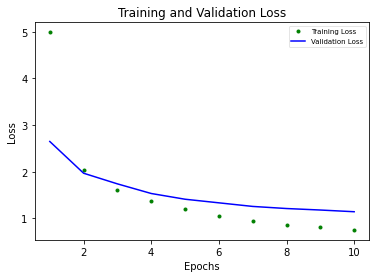
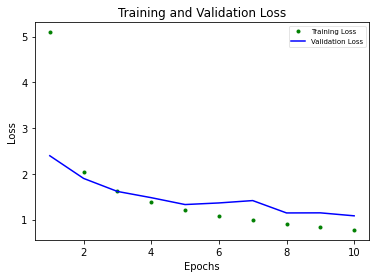
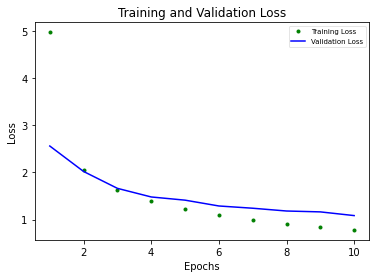
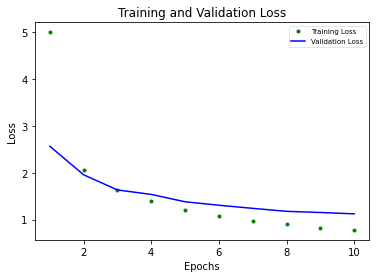
**\*MSE, r=0.1**

****

**\*MSE, r=0.5**

****

**\*MSE, r=0.9**

****

**A5. Convolutional Neural Network.**

**#compare how the model performs on the test dataset**

**#text\_images=text\_x**

**#test\_labels=test\_y**

**test\_loss, test\_acc = model.evaluate(test\_x, test\_y, verbose=2)**

**print('\nTest accuracy:', test\_acc)**

**#Make predictions-Softmax**

**probability\_model = tf.keras.Sequential([model,**

**tf.keras.layers.Softmax()])**

**predictions = probability\_model.predict(test\_x)**

**#First prediction**

**print (predictions[0])**

**#You can see which label has the highest confidence value**

**print (np.argmax(predictions[0]))**

**#test labels**

**print(test\_y [0])**