ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ

ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Επίλυση προβλήματος ταξινόμησης με χρήση Multi-layer Perceptron δικτύου

Κωνσταντίνος Ανδρέου
9521
andreouk@ece.auth.gr

Περιεχόμενα :

Στόχος και ζητούμενα εργασίας

Στόχος της εργασίας είναι ο πειραματισμός πάνω σε μια απλή αρχιτεκτονική MLP για την επίλυση ενός απλού προβλήματος ταξινόμησης. Επιλέγεται το MNIST dataset το οποίο περιλαμβάνει εικόνες χειρόγραφων ψηφίων από 28×28 pixels η κάθε μια, με στόχο τη σωστή ταξινόμηση κάθε εικόνας στην κλάση που αντιστοιχεί στο ψηφίο που αυτή απεικονίζει. Τα δεδομένα είναι χωρισμένα σε training και testing υποσύνολα, το καθένα με 60000 και 10000 δείγματα αντίστοιχα

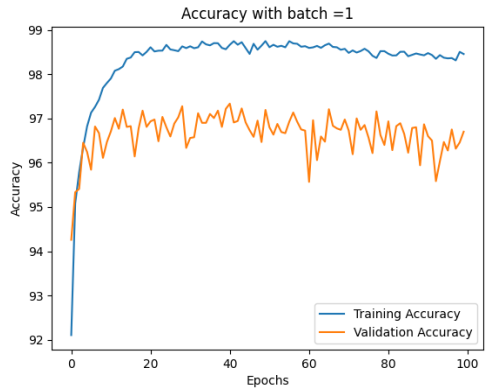
Διερεύνηση απόδοσης μοντέλου με διαφοροποίηση στο σχεδιασμό και τη διαδικασία εκπαίδευσης

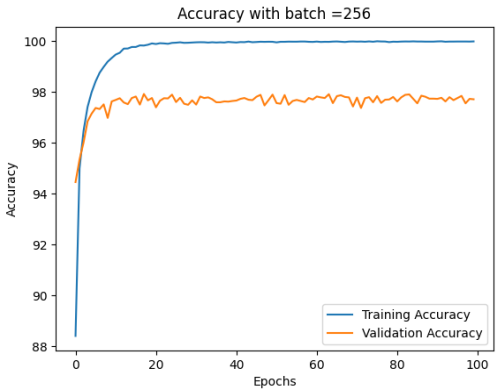
Αρχικά μελετάμε την απόδοση του δικτύου για διαφορετικές επιλογές όσον αφορά την επιλογή μεθόδου βελτιστοποίησης(optimization) , κανονικοποίησης(regularization) και αρχικοποίησης(initialization) των παραμέτρων.

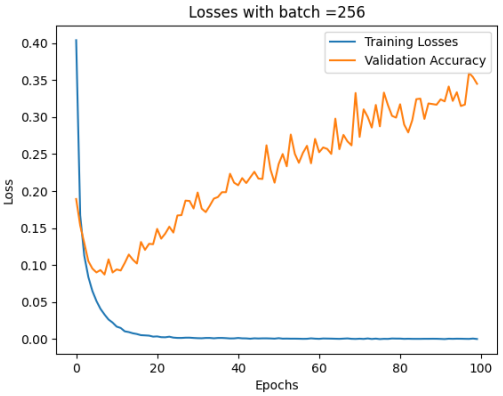
Παρατηρούμε ότι όσο μεγαλύτερο batch-size έχουμε τόσο πιο γρήγορα γίνεται η εκπαίδευση του μοντέλου για τις τρεις διαφορετικές μεθόδους. Για batch-size=1 το μοντέλο μας ανανεώνεται ξεχωριστά για κάθε δείγμα ,δηλαδή γίνονται 48000 επαναλήψεις . για batch-size=256 το μοντέλο μας τρέχει για 48000/256 = 188 επαναλήψεις , δηλαδή παίρνουμε σαν δείγματα κομμάτια του dataset και όχι ολόκληρο το dataset.Για batch-size=Ntrain κάθε εποχή περιλαμβάνει μόνο μια επανάληψη επειδή το μέγεθος του batch-size είναι ίσο με το μέγεθος των δεδομένων.

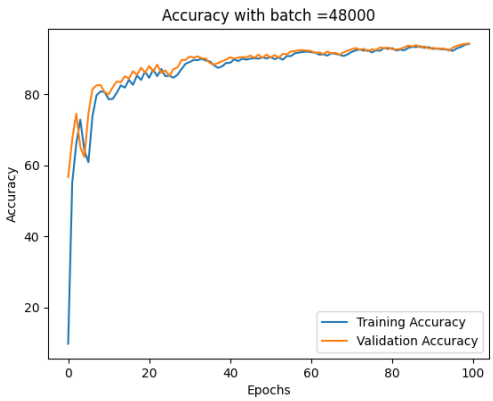
Όσο πιο μεγάλο είναι το batch-size τόσο πιο μεγάλο είναι το κόστος του μοντέλου σε θέμα μνήμης αλλα και τόσο πιο γρήγορα η εκπαίδευση του αφού κάνει πιο λίγες επαναλήψεις.

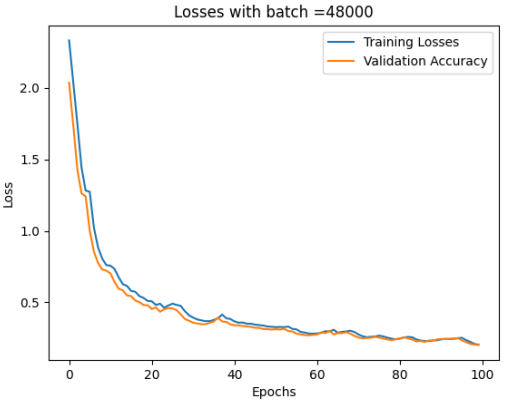
Καμπύλες ακρίβειας και κόστους για training και validation



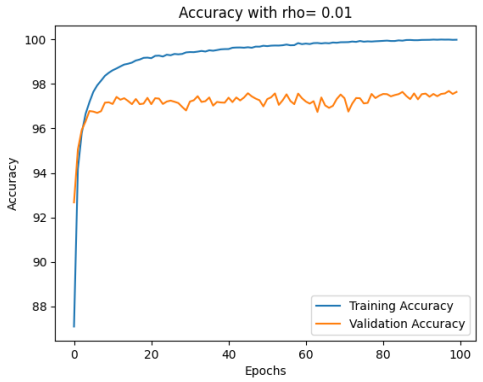


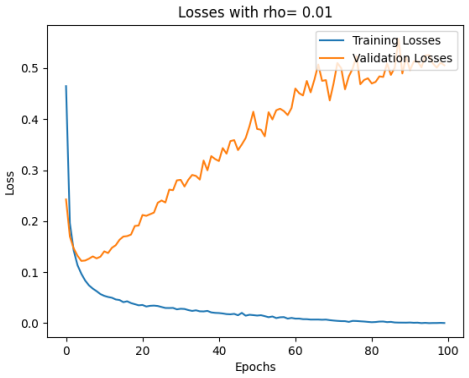


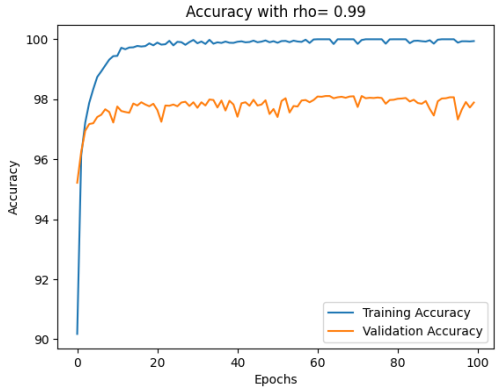


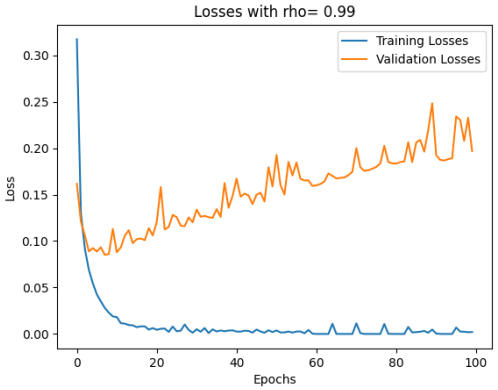


RMSProp

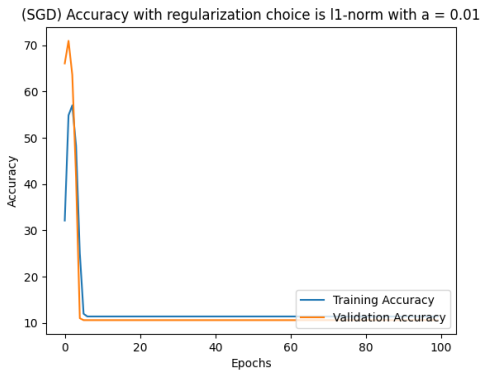

Accuracy with rho= 0.01
Accuracy
Epochs
Training Accuracy
Validation Accuracy

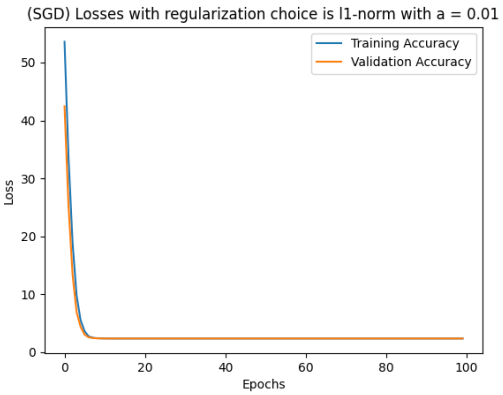

Losses with rho= 0.01
Loss
Epochs
Training Losses
Validation Losses

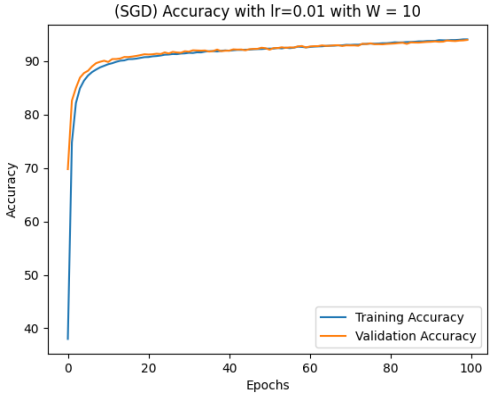

Accuracy with rho= 0.99
Accuracy
Epochs
Training Accuracy
Validation Accuracy

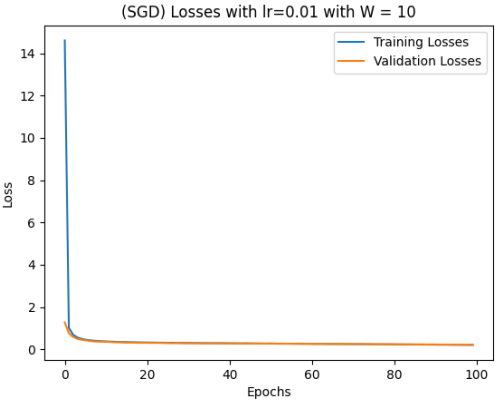

Losses with rho= 0.99
Loss
Epochs
Training Losses
Validation Losses

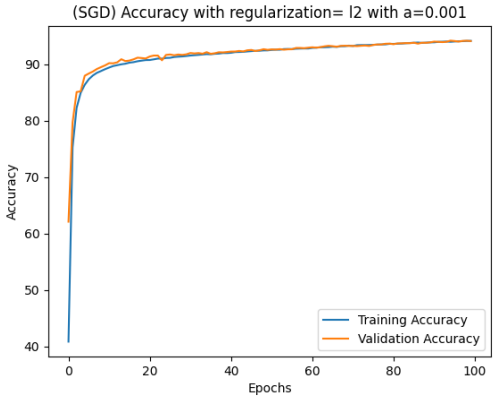
SGD

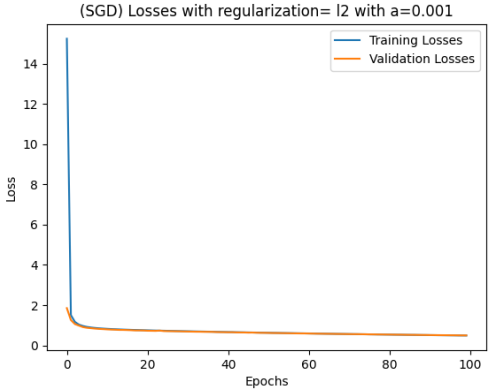

(SGD) Accuracy with regularization choice is l1-norm with a = 0.01
Accuracy
Epochs
Training Accuracy
Validation Accuracy

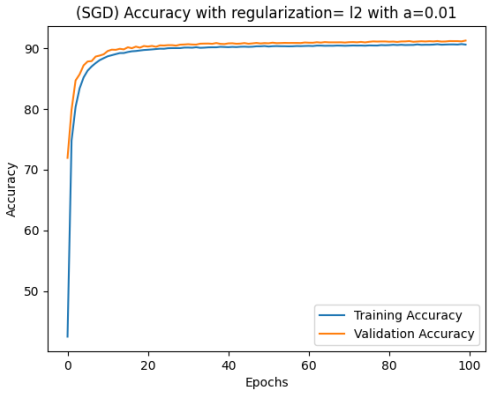

(SGD) Losses with regularization choice is l1-norm with a = 0.01
Loss
Epochs
Training Accuracy
Validation Accuracy

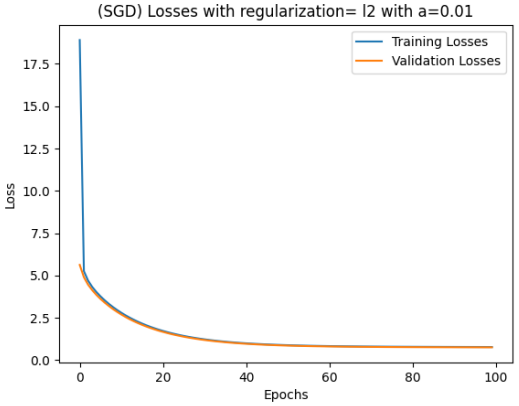

(SGD) Accuracy with lr=0.01 with W = 10
Accuracy
Epochs
Training Accuracy
Validation Accuracy

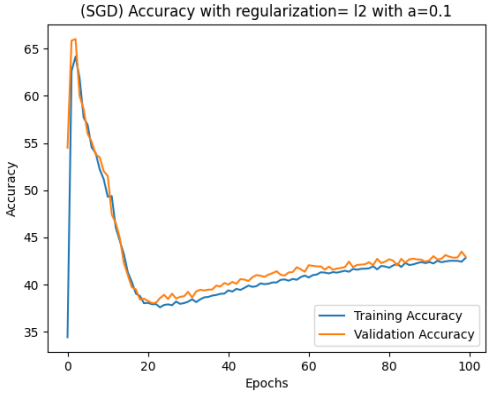

(SGD) Losses with lr=0.01 with W = 10
Loss
Epochs
Training Losses
Validation Losses

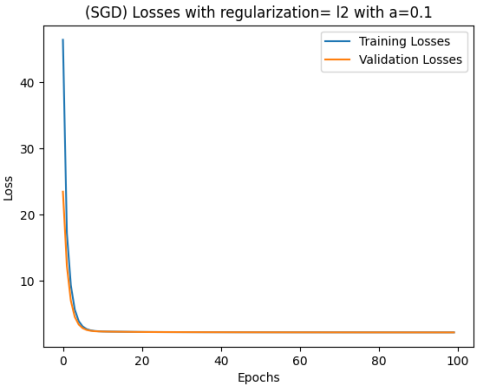












Όπως βλέπουμε στις πιο πάνω παραστάσεις ότι έχουμε πολύ διαφορετικά αποτελέσματα σε κάθε περίπτωση και μπορούμε να καταλάβουμε εύκολα ότι για l2-norm , a=0.01 , lr=0.01,W=10 έχουμε τα καλύτερα αποτελέσματα

Underfitting παρατηρούμε όταν το μοντέλο μας δεν μπορεί να συλλάβει την υποκείμενη τάση των δεδομένων ως αποτέλεσμα να καταστρέφει την ακρίβεια του μοντέλου μας. Δηλαδή εμφανίζεται όταν το μοντέλο μας δεν ταιριάζει αρκετά καλά στα δεδομένα. Συνήθως παρατηρείται το φαινόμενο αυτό όταν λιγότερα δεδομένα από αυτά που χρειαζόμαστε για την δημιουργία του

μοντέλου ή όταν προσπαθούμε να δημιουργήσουμε ένα γραμμικό μοντέλο από μη λίγα γραμμικά δεδομένα . Ο λόγος που συμβαίνει αυτό είναι γιατί οι κανόνες του μοντέλου μας παραμένουν πολύ εύκολοι και ευέλικτοι γιατί εφαρμόστηκε σε πολύ λίγα δεδομένα με αποτέλεσμα να γίνονται πολλές λάθος προβλέψεις. Μπορούμε εύκολα να το αποφύγουμε με την χρήση περισσότερων δεδομένων και με την μείωση των χαρακτηριστικών κατα την επιλογή μας. Underfitting δηλαδή σημαίνει υψηλή προκατάληψη και χαμηλή διακύμανση.

Σε αντίθεση όμως overfitting έχουμε όταν το μοντέλο μας προσαρμόζεται υπερβολικά στο training set δηλαδή του δίνουμε υπερβολικά δεδομένα εκπαίδευσης με αποτέλεσμα να μαθαίνει από τον θόρυβο και τις ανακριβείς καταχωρίσεις. Αυτό έχει ως αποτέλεσμα το μοντέλο μας να μην κατηγοριοποιεί σωστά τα δεδομένα μας λόγο υπερβολικής λεπτομέριας και θορύβου. Άλλες αιτίες που μπορεί να μας οδηγήσουν σε υπερπροσαρμογή είναι οι μη παραμετρικές και μη γραμμικές μέθοδοι γιατί οι μέθοδοι αυτοί μας δίνουν περισσότερη ελευθερία στη δημιουργία του μοντέλου με βάση το σύνολο δεδομένων με αποτέλεσμα τον κίνδυνο να δημιουργήσουμε μη ρεαλιστικά μοντέλα. Απλές λύσεις για να αποφύγουμε το φαινόμενο αυτό είναι η χρήση γραμμικών αλγορίθμων για γραμμικά δεδομένα και η επιλογή σωστών παραμέτρων όπως το μέγιστο βάθος ή ακόμα και ο καλύτερος χωρισμός των δεδομένων μας σε εκπαίδευση και έλεγχο.Υπερπροσαρμογή δηλαδή σημαίνει υψηλή διακύμανση και χαμηλή προκατάληψη.

Υπερπροσαρμογή έχουμε στα μοντέλα που παρατηρούμε το validation score , losses να μειώνεται , αυξάνεται με την αύξηση των εποχών. Υπερπροσαρμογή δεν φαίνεται να έχουμε σε κάποιο μοντέλο αλλά θεωρητικά θα έχουμε σε όλα σε πολύ μικρό αριθμό εποχών.

Fine tuning Δικτύου

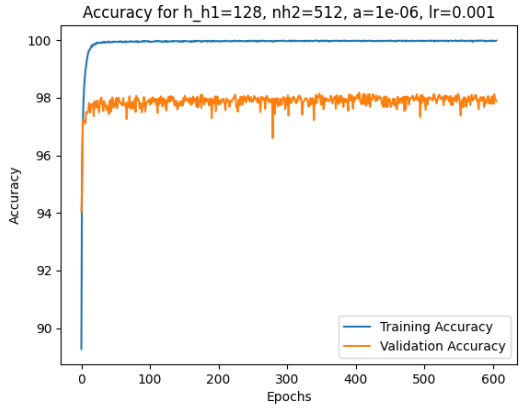
Σ Αυτό το κομμάτι της εργασίας , πρέπει να βρούμε τις βέλτιστες τιμές για μερικές υπερ παραμέτρους του δικτύου , να εκπαιδεύσουμε και να αξιολογήσουμε το μοντέλο με τις τιμές αυτές. Χρησιμοποιήσαμε τον αλγόριθμο βελτιστοποίησης RMSProp και σε κάθε στόμα εφαρμόστηκε κανονοικοποίηση με l2-norm και αρχικοποίηση των συναπτικών βαρών κάθε στώματαρο με βάση τον αλγόριθμο Henormalization.

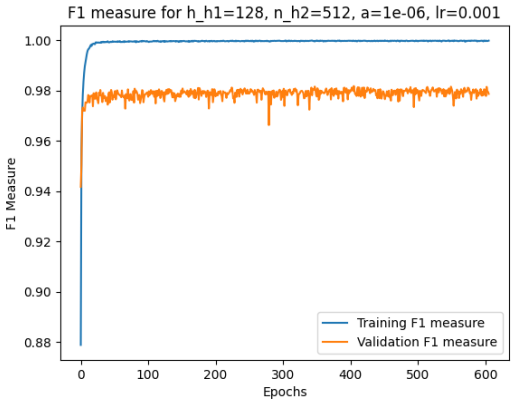
Βρίσκουμε ότι το βέλτιστο μοντέλο έχει τιμές f_measure=0.982 , nh1=128 , nh2=512 , a=10^-6 , lr=0.001

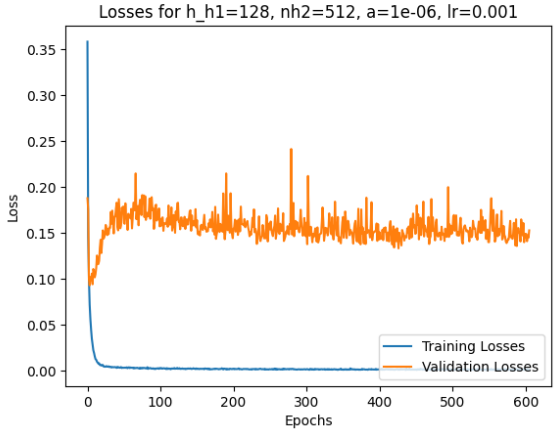
Πίνακας σύγχυσης

```
[[ 971    0    2    1    0    0    1    1    4    0]
 [   0 1130    2    1    0    0    0    1    1    0]
 [   3    2 1009    4    2    0    1    5    6    0]
 [   0    0    2  994    0    6    0    4    3    1]
 [   3    1    1    1  961    0    5    4    0    6]
 [   3    1    0   10    1  865    4    1    5    2]
 [   5    3    1    1    2    3  941    0    2    0]
 [   0    3   10    4    1    0    0 1004    5    1]
 [   2    0    2    6    3    2    1    2  955    1]
 [   4    2    0    7    8    6    2    6    1  973]]
```

Καμπύλες για accuracy , f1 measure , loss για το βέλτιστο μοντέλο







Οι μετρικές accuracy , precision , recall , f-measure υπολογίζονται πολύ εύκολα από τους πιο κάτω τύπους

PRECISION = TRUE POSITIVE/(TRUE POSITIVE + FALSE POSITIVE)

FM_1=2TRUE POSITIVE/(2TRUE POSITIVE + FALSE POSITIVE+FALSE NEGATIVE)
ACCURACY=(TRUE POSITIVE+TRUE NEGATIVE)/(TRUE POSITIVE+TRUE NEGATIVE+FALSE NEGATIVE+FALSE POSITIVE)
RECALL=TRUE POSITIVE/(TRUE POSITIVE + FALSE NEGATIVE)

Ακολουθόντας τους τύπους αυτούς στην περίπτωση μας έχουμε PR = 0.99, RE = 0.98, CA = 0.997 και = 0.987.

Όπως βλέπουμε αρχικά οι μετρικές μας είναι πολύ κοντά στην μονάδα , αυτό σημαίνει πως το grid search μας επέλεξε την βέλτιστη λύση η οποία έγινε για εκπαίδευση 200 εποχών αφού το atience που βάλαμε ήταν 200.