ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

# ΥΠΟΛΟΓΙΣΤΙΚΗ ΝΟΗΜΟΣΥΝΗ

## Επίλυση προβλήματος ταξινόμησης με χρήση Multi-layer Perceptron δικτύου

ΕΙΣΗΓΗΤΗΣ : ΘΕΟΧΑΡΗΣ ΙΩΑΝΝΗΣ
ΚΩΣΤΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ
kostakonst@ece.auth.gr
ΑΕΜ : 9209

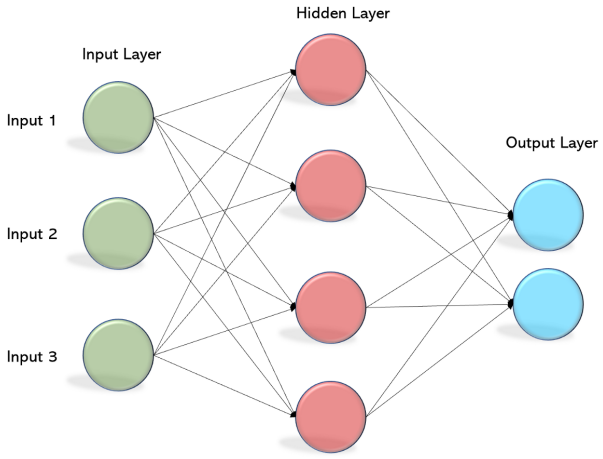
Ιανουάριος 2022

# Contents

# 1 Εισαγωγή

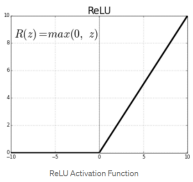
Η παρούσα εργασία, όπως αναφέρεται και στον τίτλο, πραγματεύεται την επίλυση ενός προβήματος ταξινόμησης με χρήση νευρωνικών δικτύων. Η κατασκευή των δικτύων αυτών καθίσταται εφικτή με την χρήση βιβλιοθηκών tensorflow και keras της Python. Η βάση δεδομένων στην οποία εφαρμόζονται τα μοντέλα είναι η MNIST, βάση που περιέχει χειρόγραφα ψηφία απο το 0 έως το 9 και χρησιμοποιείται για την εκπαίδευση νευρωνικών δικτύων. Σύμφωνα με την εκφώνηση, ένα 20% του συνόλου των δεδομένων εκπαίδευσης παρακρατείται για να χρησιμοποιηθεί ως σύνολο επικύρωσης. Έτσι τα δεδομένα είναι χωρισμένα σε 3 sets $(training, validation, testing) = (48000, 12000, 10000)$.

Figure 1: Δίκτυο Multi-layer Perceptron




Για την κατασκευή των δικτύων, εκτός απο το validation set πρέπει να οριστούν μερικές ακόμη υπερ-παράμετροι. Αρχικά, δίνεται οτι τα μοντέλα θα εκπαιδευτεούν όλα για 100 εποχές (epochs = 100). Δηλαδή για τον καθορισμό των άγνωστων παραμέτρων θα δοθούν στο δίκτυο 100 φορές όλα τα δείγματα εκπαίδευσης. Επιλέον, θα περιλαμβάνουν δύο κρυφά στρώματα με 128 και 256 νευρώνες αντίστοιχα, και συνάρτηση ενεργοποίησης την ReLU. Δηλαδή η έξοδος κάθε νευρώνα των κρυφών στρωμάτων, για κάθε άθροισμα βαρών(weighted sum), θα καθορίζεται απο την συνάρτηση ReLU της εικόνας 2. Αντίστοιχα, η συνάρτηση ενεργοποίησης του στρώματος εξόδου είναι η

Softmax.

['ReLU'] 

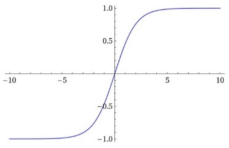
['Softmax'] 

Figure 2: Συναρτήσεις Ενεργοποίησης

Επίσης, η συνάρτηση κόστους που η εκφώνηση υποδεικνύει είναι η Categorical Cross entropy και η μετρική αξιολόγησης του μοντέλου η ακρίβεια [accuracy].

Εκτός απο τις παραμέτρους αυτές που είναι κοινές, ζητούνται να κατασκευαστούν δύο διαφορετικά μοντέλα νευρωνικών δικτύων ως προς τους αλγοριθμους βελτιστοποίησης που χρησιμοποιούν.Ένα με RMSprop (Root Mean Square Propogation) και ένα με SGD (Stochastic gradient descent) opimizer. Το δεύτερο μοντέλο, που χρησιμοποιεί για optimize τον SGD, χρησιμοποιεί $lr = 0.01$ (learning rate) και έναν initializer για την αρχικοποίηση των συναπτικών βαρών κάθε στρώματος (hidden layers) με βάση μια κανονική κατανομή με $mean = 10$. Για τον RMSprop optimizer δημιουργούνται δύο optimizers με βάση την παραμέτρο ρ. Ένας για $\rho = 0.01$ και $lr = 0.001$ και ένας για $\rho = 0.99$ και $lr = 0.001$.

# 2 Διερεύνηση απόδοσης μοντέλου με διαφοροποιήση στο σχεδιασμό και τη διαδικασία εκπαίδευσης

## 2.1 Batch size

Μία απο τις υπερ-παραμέτρους που είναι σημαντικό να προσδιοριστούν ανεξάρτητα του μοντέλου είναι το Batch Size. Αποτελεί τον αριθμό των δειγμάτων που εισάγοται πριν απο κάθε ανανέωση των παραμέτρων. Η επιλογή του είναι καθοιστική για την ταχύτητα εκπαίδευσης και την ποιότητα του τελικού νευρωνικού δικτύου. Όσο μεγαλύτερο το batch size τόσο ταχύτερη η εκπαίδευση καθώς η υπολογιστική ισχύς και η μνήμη επιτρέπουν την ταυτόχρονη επεξεργασία περισσότερων του ενός δείγματος κάθε φορά. Παρόλα αυτά η αύξησή

του σε μεγάλες τιμές υποβαθμίζει σταδιακά το μοντέλο. Συνεπώς, για batch size = 256, παρατηρήθηκε πως η εκπαίδευση είναι αρκετά ικανοποιητική και η παράμετρος θα διατηρηθεί σε όλους τους συνδιασμούς μοντέλων που ακολουθούν. Ενδεικτικά, οι χρόνοι εκτέλεσης κάθε epoch για τις τιμές του batch size απεικονίζονται στον ακόλουθο πίνακα.

| Batch Size | response time (sec) |
|---|---|
| 1 | 147.2 |
| 256 | 1 |
| 48000 | $< 1$ |

Table 1: Ταχύτητα εκπαίδευσης για Batch Sizes

## 2.2 Καμπύλες ακρίβειας και κόστους για training και validation sets

Με την εκπαίδευση των μοντέλων στα δεδομένα της βάσης MNIST προκύπτουν τα ακόλουθα διαγράμματα. Το μοντέλο με αλγόριθμο βελτιστοποίησης τον SGD φαίνεται στην εικόνα 3 ενώ αυτά με RMSProp και $rho = 0.01$ και $rho = 0.99$ στις εικόνες 4 και 5 αντίστοιχα. Σε κάθε γράφημα απεικονίζονται τα δεδομένα εκπαίδευσης με μπλέ και τα δεδομένα επικύρωσης με πορτοκαλί χρώμα.

Figure 3: SGD optimizer performance

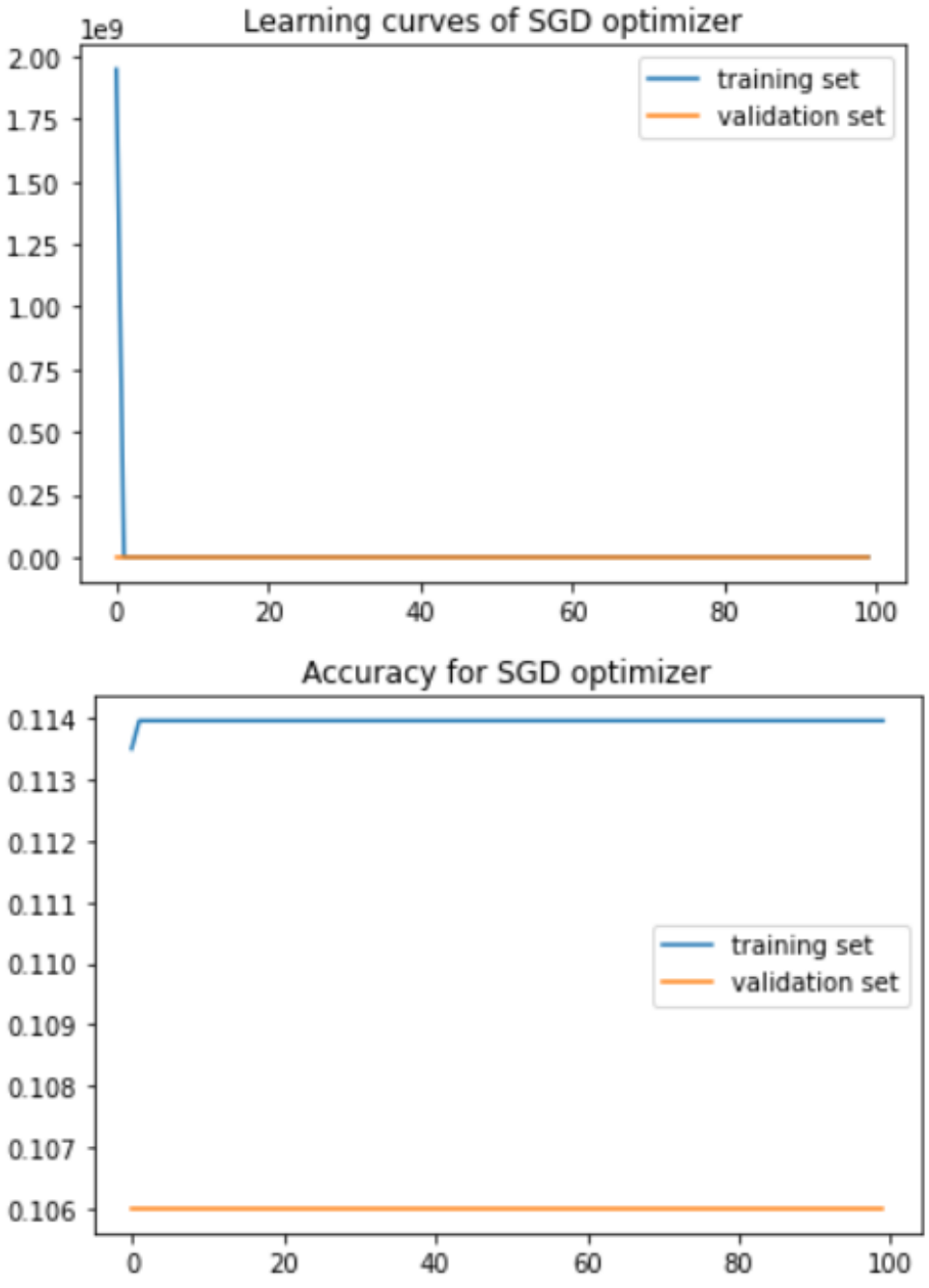


Figure 4: RMSprop optimizer performance, $rho = 0.01$

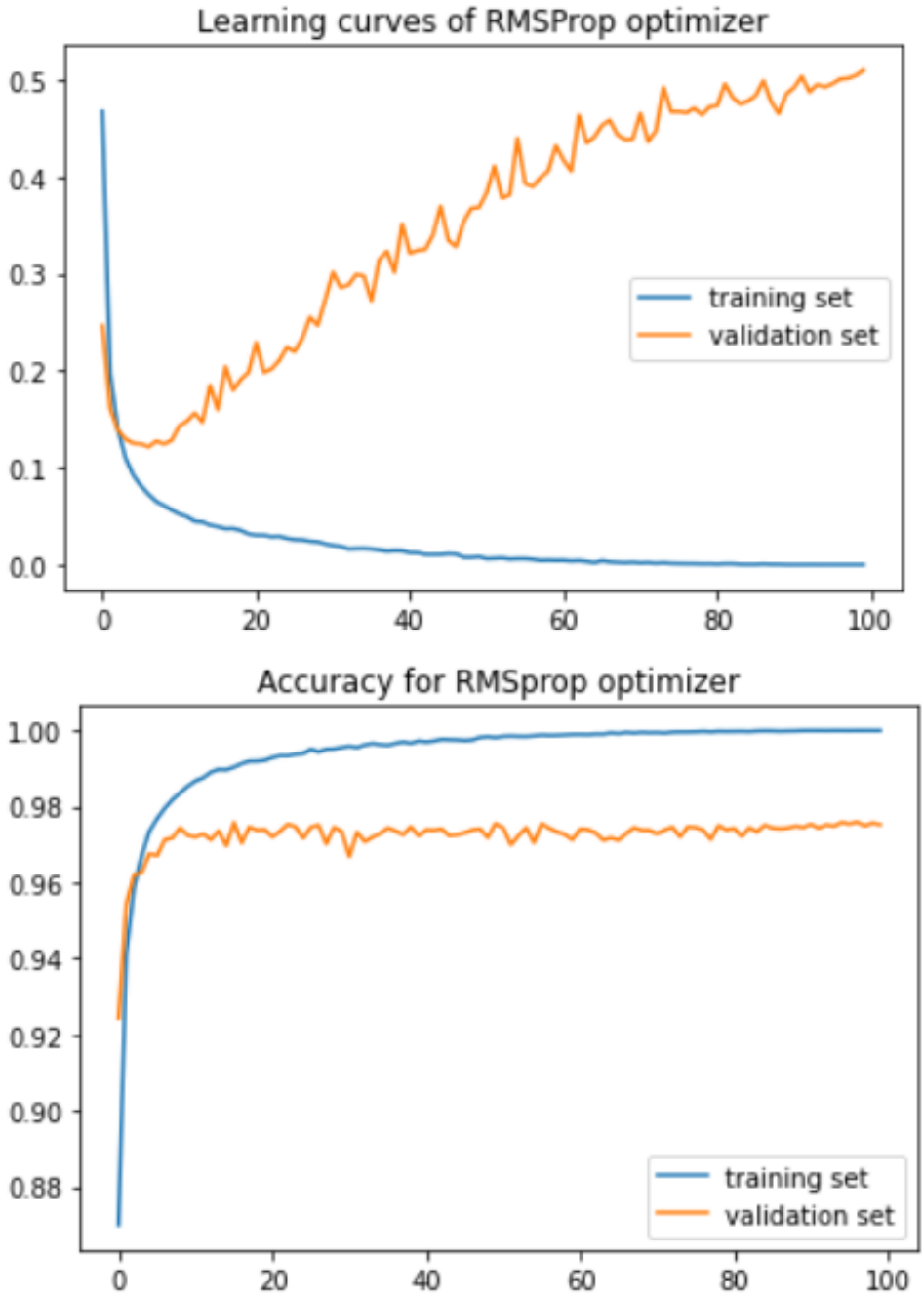
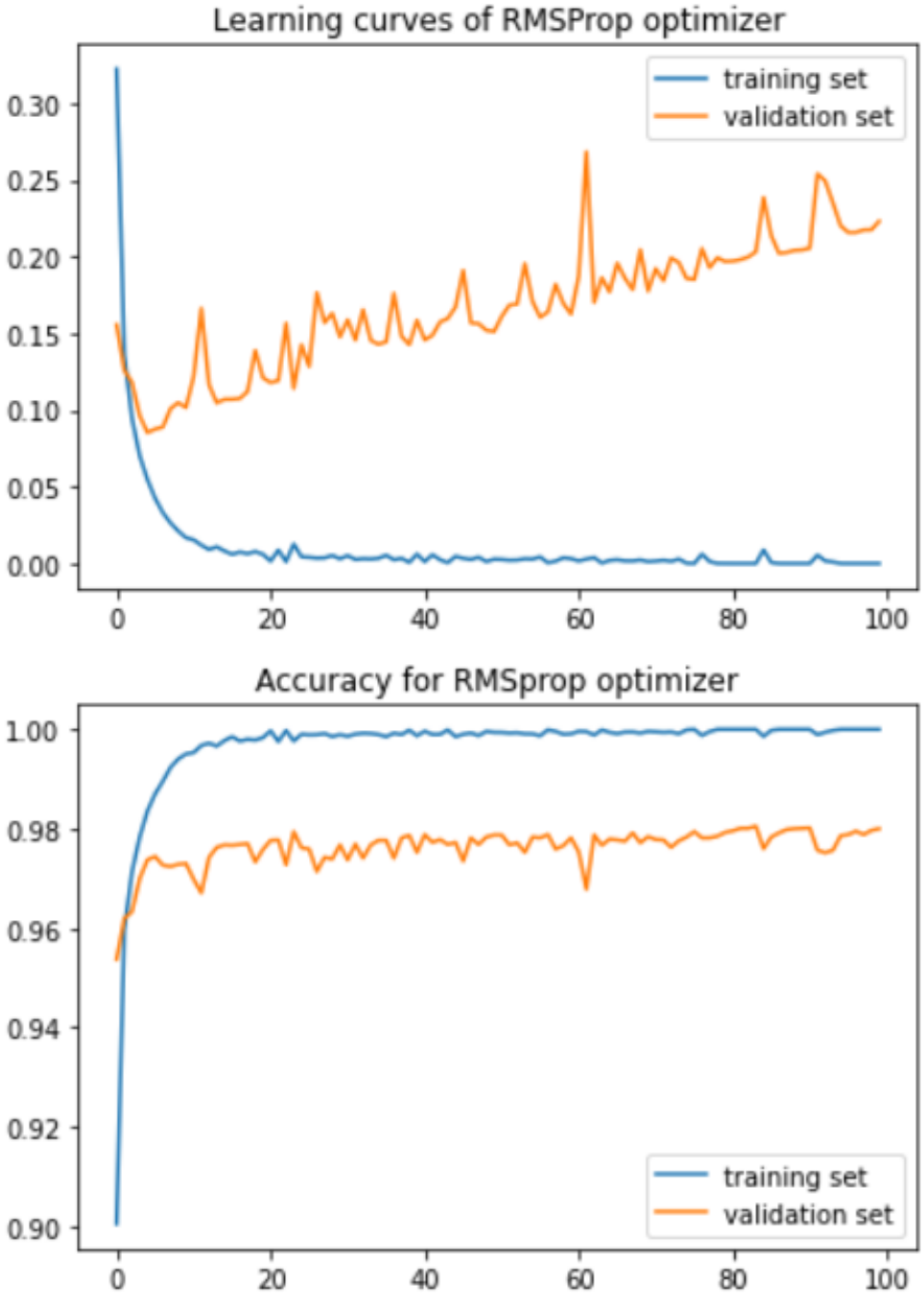


Figure 5: RMSprop optimizer performance, $rho = 0.99$



## 2.3 Σχολιασμός των αποτελεσμάτων

Αρχικά, μία απο τις σημαντικότερες παραμέτρους που πρέπει να διερευνηθούν για την εξήγηση των αποτελεσμάτων είναι η αρχικοποίηση των βαρών του μοντέλου. Μπορεί να θεωρηθεί ως τον παράγοντα που δείχνει το πόσο θα επιρρεάζεται η έξοδος απο την είσοδο. Μια λάθος αρχικοποίηση εύκολα οδηγεί σε καταστροφικά φαινόμενα όπως η εξαφάνιση (vanishing) και η εκτόξευση (exploding). Όπως είναι γνωστό απο την κατασκευή τους, τα νευρωνικά δίκτυα λαμβάνουν αποφάσεις με χρήση βαρών. Τα βάρη αυτά κατα την εκπαίδευση ρυθμίζονται ετσι ώστε να λαμβάνονται, με την μεγαλύτερη δυνατή πιθανότητα, σωστές προβλέψεις απο το μοντέλο. Η λειτουργία της εκπαίδευσης περιλαμβάνει την διάδοση προς τα εμπός (forward propagation) και την οπισθοδιάδοση (back propagation). Η διάδοση προς τα εμπρός, όπως φανερώνει και το όνομα της, είναι η διάδοση της πληροφορίας απο την είσοδο στην έξοδο. Κάθε κόμβος τότε δέχεται σαν είσοδο το βεβαρημένο άθροισμα των βαρών κάθε σύνδεσης με την αντίστοιχη έξοδο του προηγούμενου κόμβου. Το άθροισμα αυτό με την σειρά του οδηγείται σε μια συνάρτηση ενεργοποίησης και δημιουργείται η έξοδος του κόμβου. Με την ίδια διαδικασία η είσοδος οδηγείται σε όλα τα επίπεδα του μοντέλου και καταλήγει στην έξοδο οπου και υπολογίζεται η απόκλιση απο το επιθυμητό αποτέλεσμα. Στην συνέχεια, ο αλγόριθμος οπισθοδιάδοσης(Back propagation) αποτελεί την πορεία για την βελτίωση των βαρών. Με δεδομένο ένα τεχνητό νευρωνικό δίκτυο και μια συνάρτηση σφάλματος, η μέθοδος υπολογίζει την κλίση της συνάρτησης σφάλματος σε σχέση με τα βάρη του νευρωνικού δικτύου. Η διαδικασία αυτή περιλαμβάνει μια αλυσιδωτή παραγώγιση η οποία για μικρές τιμές αρχικών βαρών οδηγεί στο Vanishing δηλαδή τα βάρη των πρώτων στρωμάτων δεν επιρρεάζονται επαρκώς ανεξάρτητα της εισόδου, ενώ αντίστοιχα μεγάλες αρχικές τιμές προκαλούν μεγάλες μεταβολές βαρών που δεν βοηθούν στην τελική σύγκλιση του μοντέλου.

Στην εικόνα 3 φαίνεται το μοντέλο να διατηρεί μια σταθερά μικρή τιμή για την ακρίβεια επικύρωσης και ένα σφάλμα σταθερό στο πέρασμα των εποχών. Με βάση την ανάλυση που προηγήθηκε και παρουσιάζει την εκπαίδευση του νευρωνικού δικτύου, γίνεται αντιλυπτό οτι το νευρωνικό με τον SGD optimizer(εικόνα 3) οδηγείται σε exploding. Η αρχικοποίηση των βαρών με τιμές κοντά στο 10 (πολυ μεγάλες) καθιστά το μοντέλο δυσλειτουργικό. Οι ενημερώσεις των βαρών που συμβαίνουν λόγω της οπισθοδιάδοσης δεν αποδεικνύονται αρκετά αποτελεσματικές για να συγκλίνει το μοντέλο. Είναι συνεπώς μια περίπτωση underfitting καθώς δεν καταφέρνει να εκπαιδευτεί ουτε στο training

set.

Τέλος, το μοντέλο με τον RMSProp optimizer που παρουσιάζεται στις εικόνες 4 και 5 φαίνεται να αντιμετωπίζει το πρόβλημα του overfitting ανεξάρτητα απο την τιμή του rho. Το σφάλμα στα δεδομένα εκπαίδευσης μειώνεται συνεχώς και γρήγορα τείνει στο μηδέν ενω στα δεδομένα επικύρωσης, το σφάλμα μειώνεται μέχρι ένα σημείο απο το οποίο και μετά αυξάνεται συνεχώς. Απο εκείνο το σημείο και μετά το μοντέλο έχει περάσει στο overfitting. Αυτό επιβεβαιώνεται και απο την καμπύλη ακρίβειας του μοντέλου όπου για τα δεδομένα εκπαίδευσης φτάνει το 1 γρήγορα ,ενώ για τα δεδομένα επικύρωσης απο το αντίστοιχο πάλι σημείο και μετά παύει να αυξάνεται.

## 2.4 Προσθήκη κανονικοποίησης με L2-νόρμα

Η χρήση του κανονικοποιητή L2 συμβάλει στην αποφυγή του overfitting. Βοηθά δηλαδή το μοντέλο να γενικεύει καλύτερα και να λειτουργεί στα άγνωστα για αυτό δεδομένα απροβλημάτιστα. Όπως συνεπώς είναι λογικό, σε ένα μοντέλο που αντιμετοπίζει πρόβλημα ακόμη και στα δεδομένα εκπαίδευσης, η προσθήκη κανονικοποιητή δεν επιδιορθώνει το πρόβλημα. Αυτό αποδεικνύεται και στην εικόνα 6.a οπου το μοντέλο με τον SGD optimizer εξακολουθεί να μην έχει καλή ακρίβεια εξαιτίας της κακής αρχικοποίησης του.

Για το μοντέλο με τον RMSprop optimizer, παρατηρείται πως η λειτουργία της κανονικοποίησης κατάφερε να εξαλείψει το φαινόμενο του overfitting με καλύτερα αποτελέσματα για $\alpha = 0.01$ (εικόνα 6.c) . Η καμπύλη κόστους των δεδομένων επικύρωσης τώρα ακολουθεί αυτήν των δεδομένων εκπαίδευσης. Αυτό επιβεβαιώνεται και απο τις καμπύλες ακρίβειας, όπου η ακρίβεια επικύρωσης φαίνεται να έχει μικρές απόκλισεις.

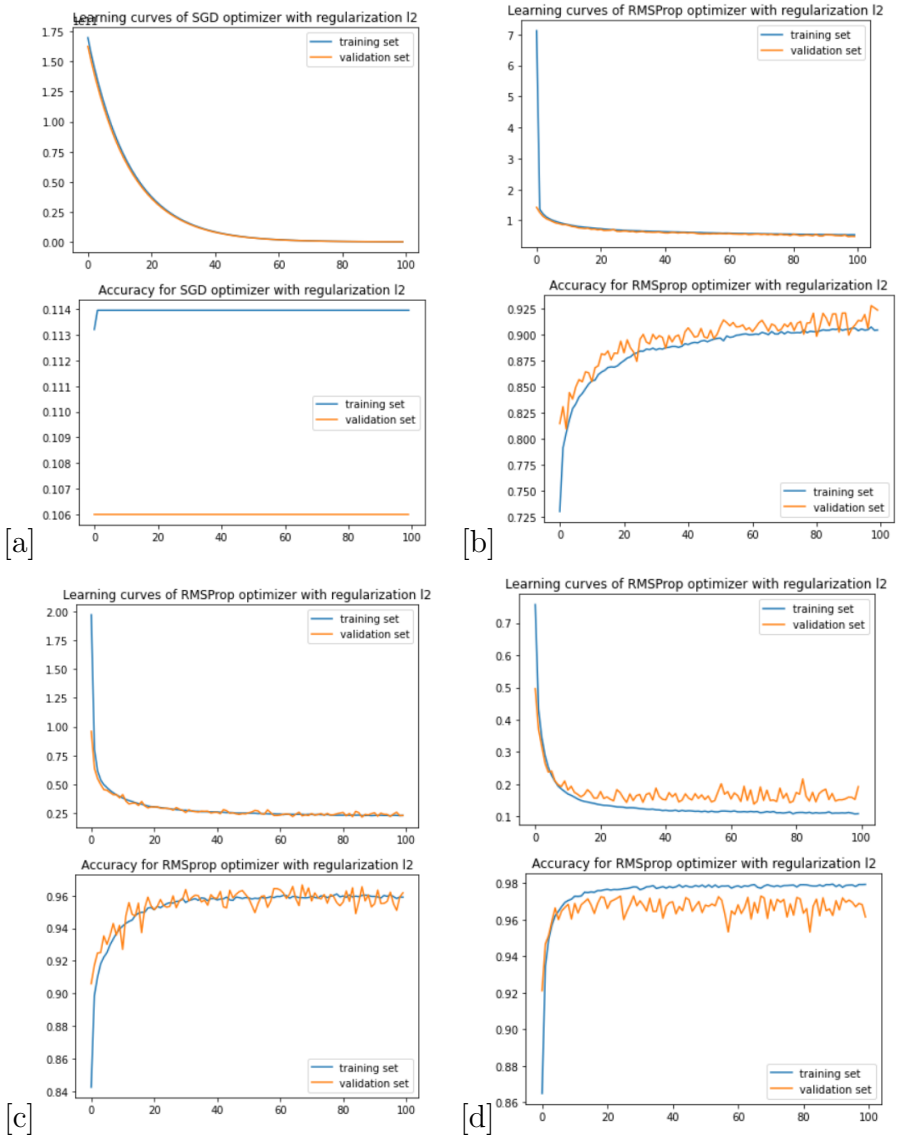



Figure 6: (a) SGD (b)$a = 0.1$ (c) $a = 0.01$ (d) $a = 0.001$

## 2.5 Κανονικοποίηση με L1-νόρμα και dropout

Στην συνέχεια της άσκησης ζητείται η προσθήκη κανονικοποίησης L1-νόρμας για τα συναπτικά βάρη των στρωμάτων του δικτύου ($\alpha = 0.01$) και ταυτόχρονη χρήση dropout με dropout probability 0.3. Και οι δύο αυτές τεχνικές αποτελούν τρόπους γενίκευσης του μοντέλου ωστε να ανταποκρίνεται στα νέα δεδομένα που θα δεχθεί. Αποτελούν δηλαδή χρήσιμα εργαλεία για τα μοντέλα που εμφανίζουν overfitting. Συνεπώς, η χρήση του στο Μοντέλο με τον SGD optimizer που δεν οδηγείται σε αποτελεσματική εκπαίδευση δεν έχει ιδιαίτερη σημασία.

Αναλυτικότερα, το Dropout είναι μια μέθοδος κανονικοποίησης που τυχαία επιλεγμένοι κόμβοι αγνοούνται κατα την εκπαίδευση. Αυτό έχει ως αποτέλεσμα το στρώμα να μοιάζει και να αντιμετωπίζεται όπως ένα στρώμα με διαφορετικό αριθμό κόμβων και συνδεσιμότητα σε σχέση με το προηγούμενο στρώμα. Όταν οι νευρώνες απομακρυνθούν τυχαία από το δίκτυο κατά τη διάρκεια της εκπαίδευσης, ότι άλλοι νευρώνες θα πρέπει να παρέμβουν και να χειριστούν την αναπαράσταση που απαιτείται για να γίνουν προβλέψεις για τους νευρώνες που λείπουν. Αυτό πιστεύεται ότι έχει ως αποτέλεσμα πολλαπλές ανεξάρτητες εσωτερικές αναπαραστάσεις που μαθαίνονται από το δίκτυο. Το αποτέλεσμα είναι ότι το δίκτυο γίνεται λιγότερο ευαίσθητο στα ειδικά βάρη των νευρώνων. Αυτό με τη σειρά του οδηγεί σε ένα δίκτυο που είναι ικανό για καλύτερη γενίκευση και είναι λιγότερο πιθανό να υπερπροσαρμόσει τα δεδομένα εκπαίδευσης. Με την μέθοδο εισάγεται μια νέα υπερπαράμετρος που καθορίζει την πιθανότητα με την οποία οι έξοδοι του στρώματος εγκαταλείπονται, ή αντίστροφα, την πιθανότητα με την οποία οι έξοδοι του στρώματος διατηρούνται. Στην άσκηση δίνεται ίση με 0.3.

Ενδιαφέρον παρουσιάζει η εφαρμογή τους στο δεύτερο μοντέλο. Για τιμές $(dropoutpropability, alpha) = (0.3, 0.01)$ τα αποτελέσματα της εκπαίδευσης φαίνονται στις εικόνες 7.α και 7.b. Παρατηρώντας την πολύ μικρη τιμή ακρίβειας (accuracy) τόσο στο σετ εκπαίδευσης όσο και στο σετ επικύρωσης γίνεται αντιλυπτό οτι το σύστημα οδηγήθηκε σε underfitting. Με την ανάθεση του $alpha = 0.001$ το πρόβλημα λύνεται και τελικά το δίκτυο εκπαιδεύεται ικανοποιητικά (εικόνες 7.c,7.d). Στο τελικό μοντέλο, το κόστος φαίνεται να έχει και στα δεδομένα εκπαίδευσης και στα δεδομένα επικύρωσης αρκετά μεγάλη ταχύτητα σύγκλισης στο μηδέν ,ενώ η ακρίβεια φαίνεται να είναι αρκετά μεγάλη με αυτή του validation set να είναι μεγαλύτερη. Παρόλα αυτά, στο γράφημα της ακρίβειας υπάρχει μια αστάθεια που οφείλεται στην πολυπλοκότητα που

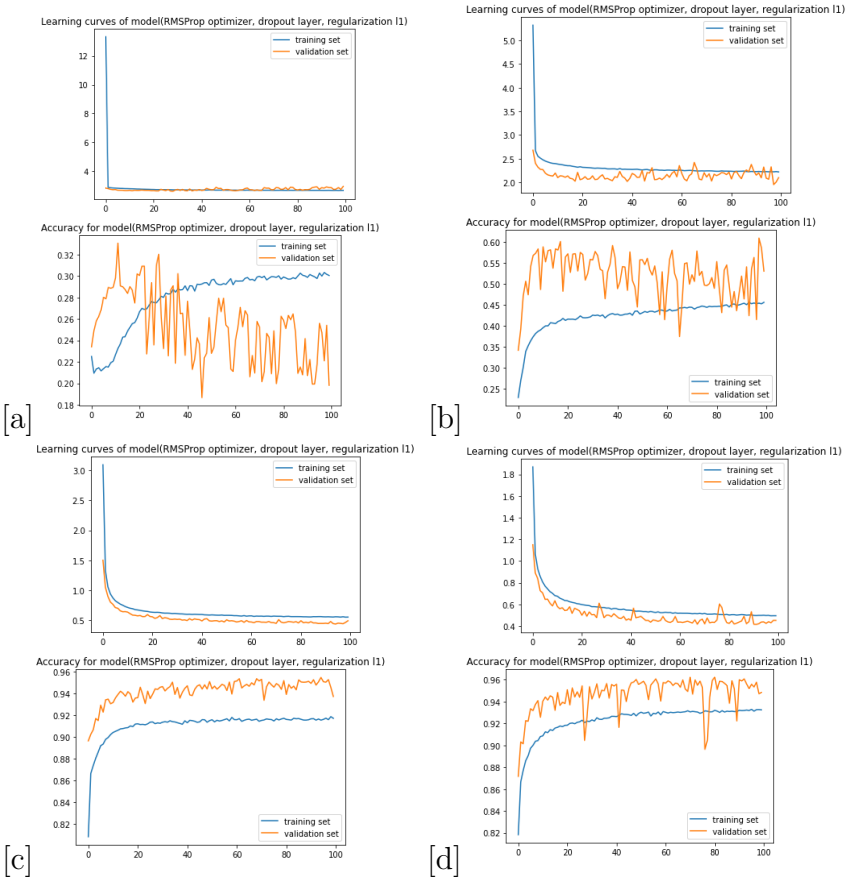



Figure 7: (a) $a = 0.01$, RMSProp optimizer $\rho = 0.01$ (b)$a = 0.01$, RMSProp optimizer με $\rho = 0.99$ (c) $a = 0.001$, RMSProp optimizer με $\rho = 0.01$(d) $a = 0.001$, RMSProp optimizer με $\rho = 0.99$

προστέθηκε στο μοντέλο.

# 3 Fine tuning δικτύου

Στο συγκεκριμένο κομμάτι της εργασίας, σκοπός είναι η εύρεση των βέλτιστων τιμών για μερικές υπερπαραμέτρους του δικτύου και η τελική εκπαίδευση και αξιολόγηση ενός μοντέλου με βάση τις επιλεγμένες τιμές.

Οι υπερπαράμετροι του δικτύου προς εξέταση, καθώς και το εύρος αναζήτησης

για κάθε παράμετρο, θα είναι:
1. αριθμός νευρώνων πρώτου κρυφού στρώματος $n_{h1} \in \{64, 128\}$
2. αριθμός νευρώνων δεύτερου κρυφού στρώματος $n_{h2} \in \{256, 512\}$
3. παράμετρος κανονικοποίησης $\alpha \in \{0.1, 0.001, 0.000001\}$
4. ρυθμός εκμάθησης $lr \in \{0.1, 0.01, 0.001\}$

Ορίζεται στην συνέχεια με βάση το $keras_tuner$, ένας tuner με βάση την κλάση Hyperband, καθώς και μια μέθοδος Early stopping με patience=200. Έχοντας ακόμη ορίσει όλες τις παραμέτρους της εκφώνησης προκύπτει οτι το τελικό μοντέλο με τα καλύτερα αποτελέσματα δίνεται απο $(n_{h1}, n_{h2}, \alpha, lr) = (128, 512, 0.000001, 0.001)$. Οι υπερπαράμετροι χρησιμοποιούνται για να χτίσουμε το τελικό μοντέλο που θα εκπαιδευτεί. Το μοντέλο εκπαιδεύεται για 100 εποχές, καθώς η εκπαίδευση διαρκεί αρκετά.

## 3.1 Πίνακας σύγχυσης, learning curves και μετρικές αξιολόγησης

Μόλις κατασκευαστεί ο πίνακας σύγχυσης του τελικού μοντέλου ( υπολογίζεται με την βοήθεια των metrics της scikit-learn βιβλιοθήκης) γίνεται χρήση της heatmap() μεθόδου που παρέχεται στην βιβλιοθήκη seaborn ωστε να αναπαρασταθεί. Στην αναπαράσταση αυτή (εικόνα 8), όσο μεγαλύτερες είναι οι τιμές τόσο μιο σκούρο είναι το μπλε που τις απεικονίζει. Στην προκυμένη περίπτωση, οι μεγαλύτερες τιμές εμφανιζονται στην κύρια διαγώνιο.

Έτσι λοιπον προκύπτει το συμπλερασμα οτι το προβλεπόμενο απο το μοντέλο ψηφίο είναι ίδιο με το πραγματικό. Αυτό σημαίνει πως το μοντέλο καταφέρνει αρκετά ικανοποιητικά να προβλέψει τα ψηφία που του δίνονται. Παρά την πολύ καλύ πρόβλεψη που δίνει το τελικο μοντέλο, είναι φανερό απο την καμπύλη εκμάθησης(εικόνα 9) οτι έχει οδηγηθεί σε overfitting. Τέλος, οι μετρικές αξιολόγησης του μοντέλου που ζητήθηκαν είναι
$[accuracy, Fmeasure, Precision, Recall] = [0.97, 0.96, 0.91]$ Αυτό δείχνει οτι το μοντέλο μπορεί να αξιολογηθεί ικανοποιητικά απο όλες τις μετρικές.

Figure 8: confusion matrix

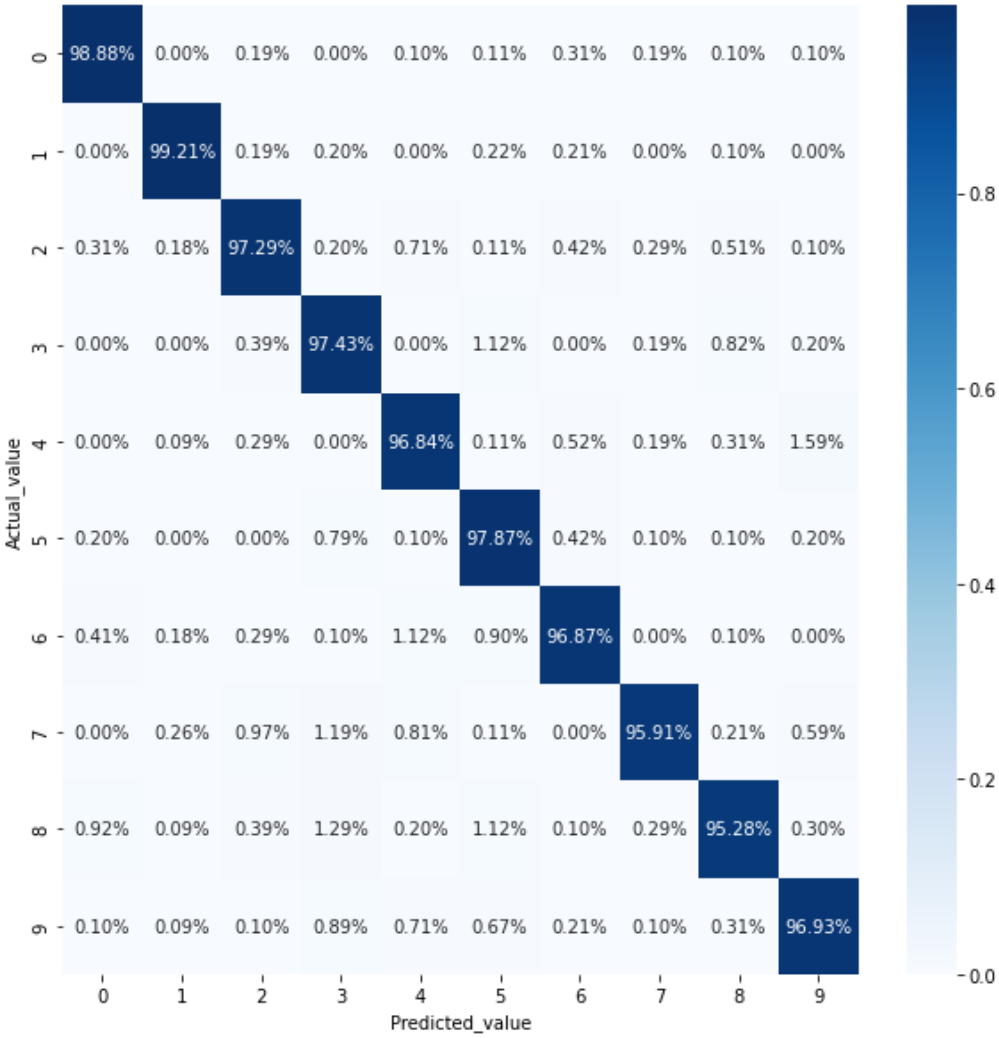


Figure 9: learning curves