Επεξεργασία Δεδομένων και Αλγόριθμοι Μάθησης
Αναφορά Σετ Ασκήσεων 1

Μακάριος Χρηστάκης

Περιεχόμενα

1 Πρόβλημα 1

1.1 Ερώτημα α

Όπως είχε αναλυθεί στη θεωρία του μαθήματος, το βέλτιστο τεστ κατά Bayes που ελαχιστοποιεί την πιθανότητα σφάλματος απόφασης είναι το τεστ λόγου πιθανοφάνειας. Για την εξέταση μεταξύ δυο υποθέσεων H_0 και H_1 σχετικά με μια τυχαία μεταβλητή x, όπου:

$$\begin{cases} H_0 : x \sim f_0(x) \\ H_1 : x \sim f_1(x) \end{cases} \tag{1}$$

και $f_0(x)$,$f_1(x)$ οι συναρτήσεις πυκνότητας πιθανότητας της τυχαίας μεταβλητής x για κάθε υπόθεση. Το τεστ λογου πιθανοφάνειας σχετικά με το ποια απόφαση θα πάρουμε είναι:

$$\frac{f_1(x)}{f_0(x)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P(H_0)}{P(H_1)} \tag{2}$$

που στην περίπτωσή μας $P(H_0) = P(H_1) = 0.5$ ισοδυναμεί με:

$$\frac{f_1(x)}{f_0(x)} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \tag{3}$$

Στο πρόβλημα αυτό εξετάζουμε τυχαίο διάνυσμα, του οποίου οι συνιστώσες x_1,x_2 είναι στατιστικώς ανεξάρτητες οπότε $f_i(x_1, x_2) = f_i(x_1) \cdot f_i(x_2)$ και έχουμε τις υποθέσεις:

$$f_0 \sim N(0, 1)$$
$$f_1 \sim 0.5\{N(-1, 1) + N(1, 1)\}$$

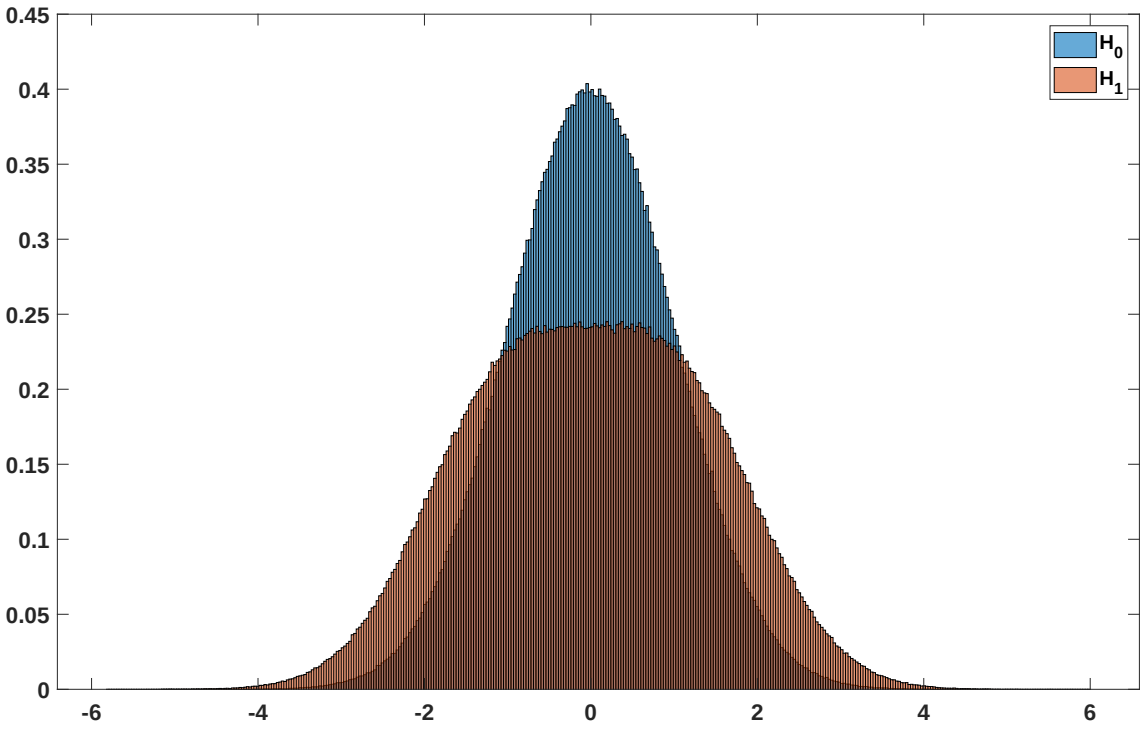
Οπότε η σχέση 3 ισοδυναμεί με:

$$r = \frac{f_1(x_1)f_1(x_2)}{f_0(x_1)f_0(x_2)} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \tag{4}$$

Κανόνας τον οποίο χρησιμοποιούμε στο επόμενο ερώτημα.

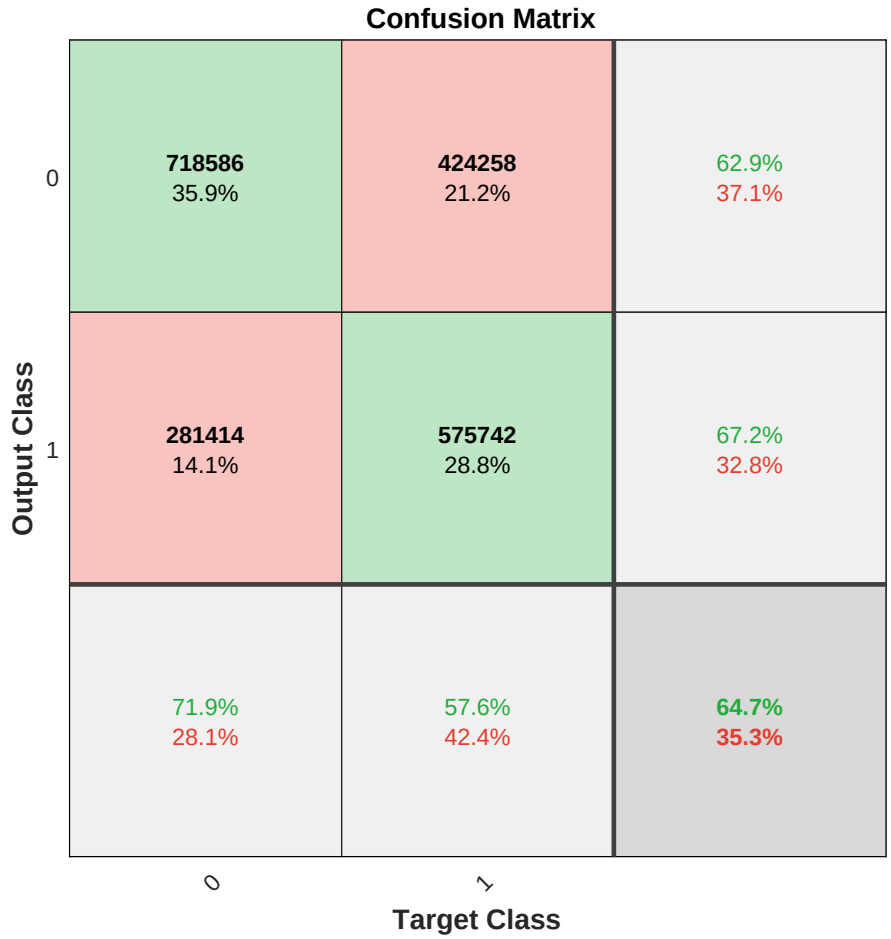
1.2 Ερώτημα β

Για το ερώτημα αυτό δημιουργήθηκαν 10^6 τυχαία διανύσματα από κάθε κατηγορία, τα ιστογράμματα του συνόλου των δεδομένων για κάθε κατηγορία με pdf κανονικοποίηση στον κατακόρυφο άξονα φαίνονται παρακάτω:



Σχήμα 1: Ιστόγραμμα δημιουργημένων δεδομένων για κάθε υπόθεση

Χρησιμοποιόντας τον κανόνα της σχέσης 4 για να κατηγοριοποιήσουμε τα δεδομένα (στην περίπτωση που ο λόγος είναι 1 αποφασίζεται υπέρ της H_1 προσεγγιστικά), παίρνουμε τον πίνακα σύγχυσης του σχήματος 2 για τον βέλτιστο κανόνα απόφασης αυτόν.



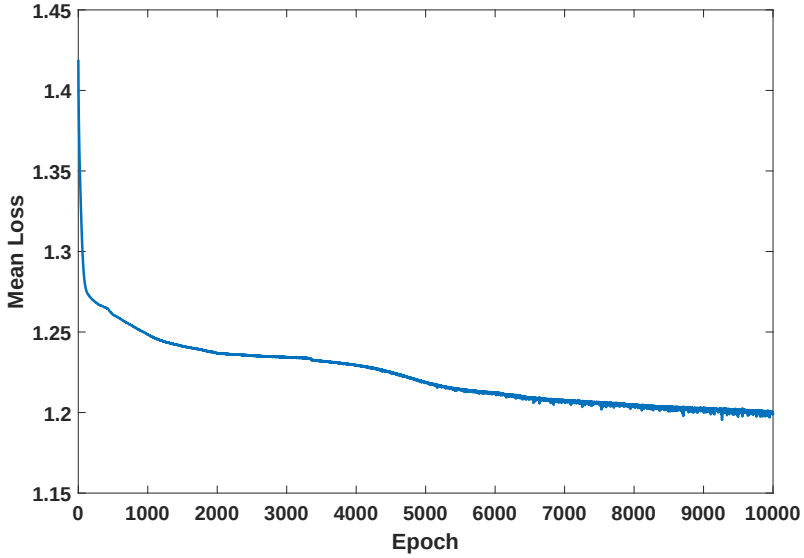
Σχήμα 2: Confusion Matrix για τον βέλτιστο Bayesian κατηγοριοποιητή

Όπως φαίνεται για το τεστ λόγου πιθανοφάνειας το συνολικό σφάλμα είναι $\underline{\mathbf{35.3\%}}$.

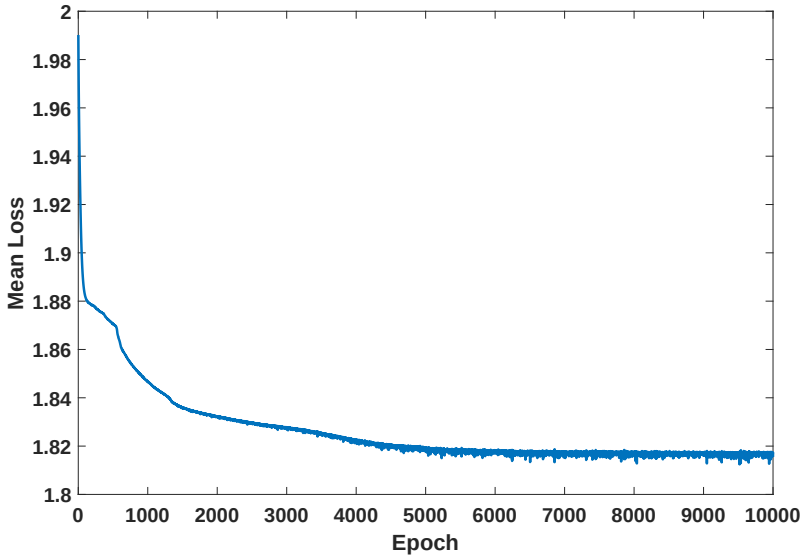
1.3 Ερώτημα γ

Υλοποιήθηκε το ζητούμενο "ρηχό" νευρωνικό δίκτυο, η αρχιτεκτονική και ο τρόπος εκπαίδευσης του οποίου αναλύονται στο κεφάλαιο 3 (Παράρτημα). Για την εκπαίδευση του νευρωνικού δικτύου δημιουργήθηκαν άλλα 200 διανύσματα από κάθε υπόθεση, τα οποία χρησιμοποιήθηκαν επανειλημμένα για να εκπαιδεύσουμε το δικτυο. Κάθε πλήρες πέρασμα από όλα τα ζεύγη διανυσμάτων αναφέρεται σαν epoch, όπως είναι η αγγλική ορολογία, στα παρακάτω διαγράμματα.

Κατά την εκπαίδευση, για κάθε epoch υπολογίστηκε ο μέσος όρος των τιμών της cost function (Σχέση 14) και έγινε plot των τιμών του μέσου Loss ανά epoch για να διαπιστώσουμε αν έχει συγκλίνει το stochastic gradient descent. Τα αποτελέσματα για την εκπαίδευση με χρήση των μεθόδων cross-entropy και exponential φαίνονται στο σχήμα 3.



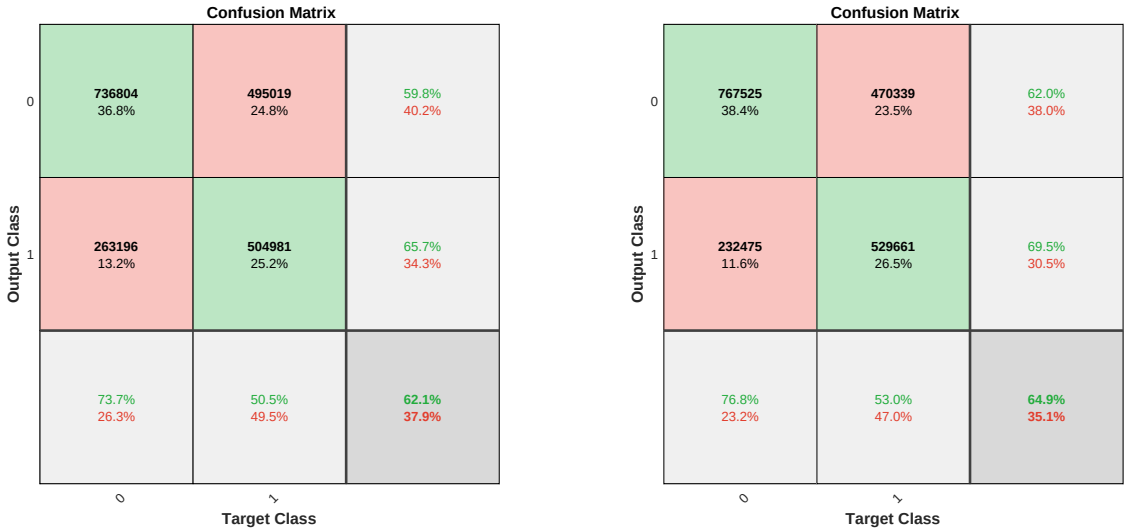
(α′) cross-entropy



(β′) exponential

Σχήμα 3: Μέσο κόστος ανά epoch κατά το training των νευρωνικών δικτύων.

Οι πίνακες σύγχυσης κατά το evaluation των δυο δικτύων αυτών με τα δεδομένα του ερωτήματος (α) φαίνονται στο σχημα 4.



(α΄) cross-entropy (β΄) exponential

Σχήμα 4: Πίνακες σύγχυσης για την αξιολόγηση των νευρωνικών δικτύων.

Συνοπτικά για την αξιολόγηση των κανόνων κατηγοριοποίησης τα ολικά σφάλματα φαίνονται στον Πίνακα 1.

	cross-entropy	exponential	likelihood ratio
σφάλμα	37.9%	35.1%	35.3%

Πίνακας 1: Συγκριση των συνολικών ποσοστιαίων σφαλμάτων ανα μέθοδο κατηγοριοποίησης.

Μπορούμε να παρατηρήσουμε πως όπως αναμένεται, μιας και το τεστ λόγου πιθανοφάνειας είναι ο βέλτιστος κατηγοριοποιητής κατά Bayes, η cross entropy μέθοδος έχει υψηλότερο σφάλμα από το βέλτιστο. Όμως παρατηρείται οτι η exponential μέθοδος έχει 0.2% μικρότερο σφάλμα από το βέλτιστο, πράγμα που είναι αδύνατο θεωρητικά, αλλά επειδή παίρνουμε ένα πεπερασμένο πλήθος υλοποιήσεων της κάθε υπόθεσης, ενδέχεται τα δεδομένα να ευνοούν κάποια μέθοδο. Στην ουσία όμως όταν το πλήθος τον υλοποιήσεων τείνει στο άπειρο ο likelihood ratio κατηγοριοποιητής έχει την ελάχιστη πιθανότητα σφάλματος.

2 Πρόβλημα 2

Τα δεδομένα του προβλήματος από την βιβλιοθήκη MNIST εισήχθηκαν στο MATLAB και κανονικοποιήθηκαν χρησιμοποιόντας τον κώδικα [1] με μια τροποποίηση για να μην κάνει trim τα γύρω πλαίσια κάθε εικόνας, οπότε προκύπτουν οι κανονικοποιημένες grayscale εικόνες 28×28.

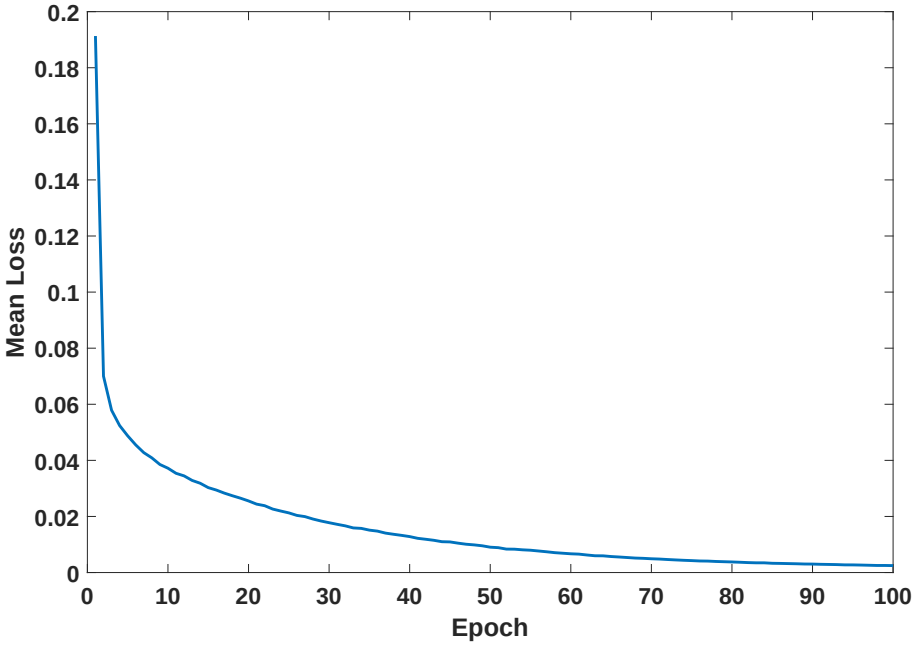
Από τις εικόνες αυτές κρατήθηκαν μόνο όσες είχαν label ψηφίου 0 και 8 και έπειτα μετατράπηκαν σε διανύσματα 784×1. Τα διανύσματα αυτά στο training dataset (και στο testing dataset) είχαν διαφορετικό πλήθος για κάθε ψηφίο, οπότε μιας και εκπαιδεύουμε ανά ζευγάρια δεδομένων το νευρωνικό, χρησιμοποιήθηκαν $min(N^8_{train}, N^0_{train})$ ζεύγη διανυσμάτων, όπου N^i_{train} το πλήθος των διαθέσιμων διανυσμάτων για εκπαίδευση που αντιστοιχούν σε ψηφίο i=0 ή 8.

Για την κατηγοριοποίηση χρησιμοποιήθηκε νευρωνικό δίκτυο $784 \times 300 \times 1$ όπως ζητήθηκε, φροντίζοντας πάντα να χρησιμοποιείται η κατάλληλη μη γραμμικότητα στο εξωτερικό layer. Όπως και στο προηγούμενο πρόβλημα το learning rate που χρησιμοποιήθηκε στον gradient descent αλγοριθμο είναι:

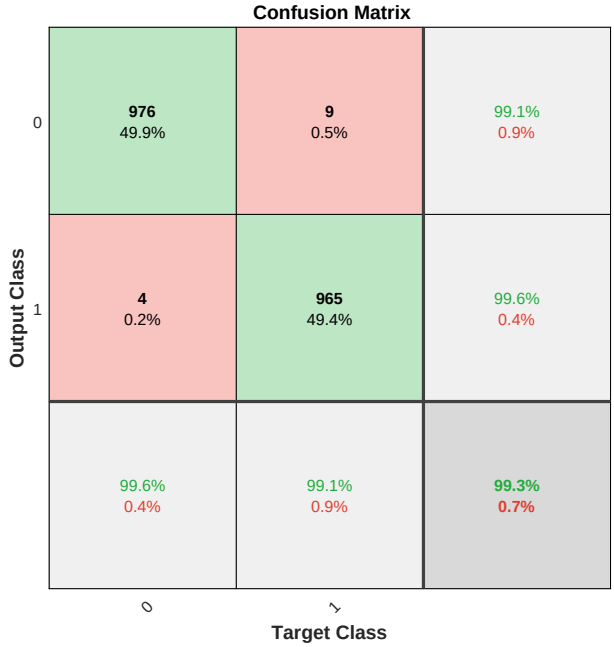
$$\mu = 1 \cdot 10^-3 \tag{5}$$

Τα αποτελέσματα των simulation για τις διάφορες μεθόδους φαίνονται στις παρακάτω σελίδες.

Για την cross entropy μέθοδο η αντίστοιχη καμπύλη εκπαίδευσης και ο πίνακας σύγχυσης για το testing set διανυσμάτων φαίνονται στο σχήμα 5.



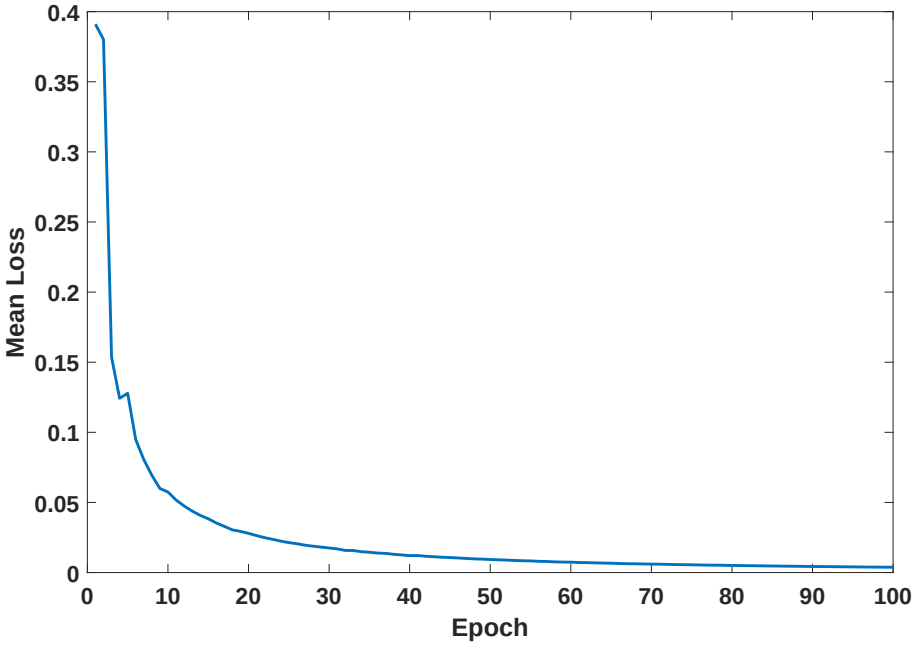
(α΄) Καμπύλη μέσου κόστους άνα epoch.



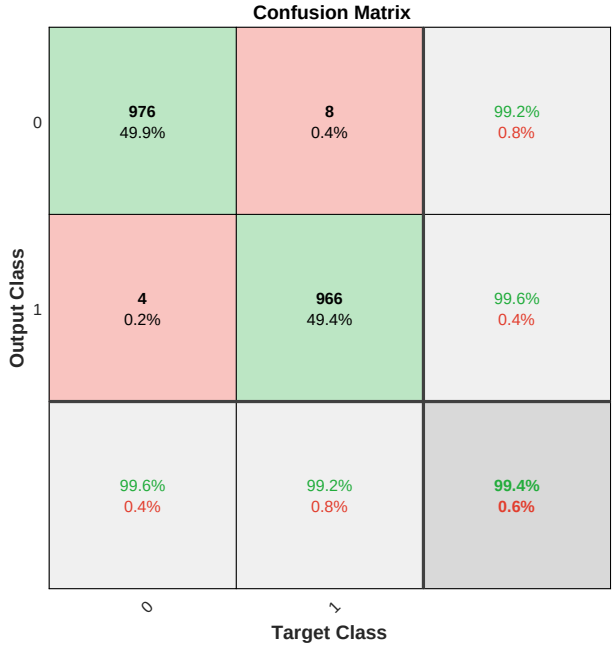
(β΄) Πίνακας σύγχυσης για την αξιολόγηση του testing set.

Σχήμα 5: Αποτελέσματα εκπαίδευσης και αξιολόγησης με τη μέθοδο Cross-Entropy για το MNIST dataset.

Για την exponential μέθοδο η αντίστοιχη καμπύλη εκπαίδευσης και ο πίνακας σύγχυσης για το testing set διανυσμάτων φαίνονται στο σχήμα 6.



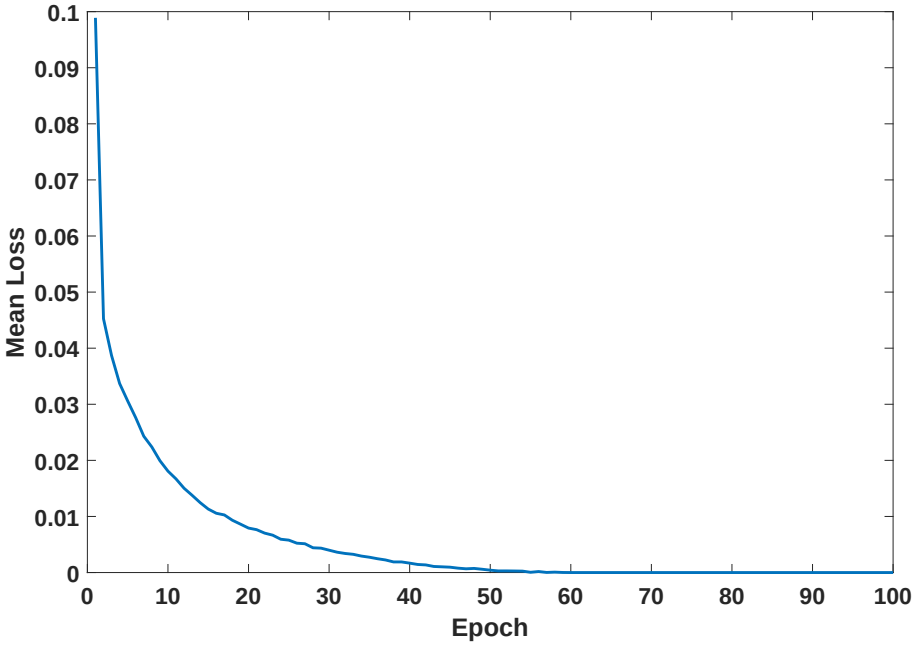
(α΄) Καμπύλη μέσου κόστους άνα epoch.



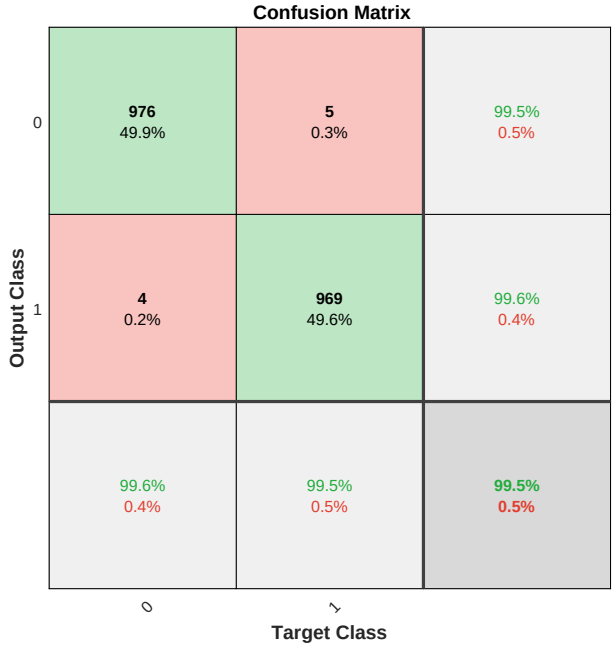
(β΄) Πίνακας σύγχυσης για την αξιολόγηση του testing set.

Σχήμα 6: Αποτελέσματα εκπαίδευσης και αξιολόγησης με την exponential μέθοδο για το MNIST dataset.

Για την hinge μέθοδο η αντίστοιχη καμπύλη εκπαίδευσης και ο πίνακας σύγχυσης για το testing set διανυσμάτων φαίνονται στο σχήμα 7.



(α΄) Καμπύλη μέσου κόστους άνα epoch.



(β΄) Πίνακας σύγχυσης για την αξιολόγηση του testing set.

Σχήμα 7: Αποτελέσματα εκπαίδευσης και αξιολόγησης με τη Hinge μέθοδο για το MNIST dataset.

Παρατηρούμε πως για να συγκλίνει το stochastic gradient descent με το dataset αυτό χρειάστηκαν πολύ λιγότερα epochs (100 σε σχέση με 5-10 χιλιάδες) μιας και το υπήρχαν περισσότερα ζεύγη παραδειγμάτων σε σχέση με αυτά του προβλήματος 1. Επιπλέον παρατηρήθηκε πως το μέσο κόστος του τελικού epoch στο training τείνει στο μηδεν για τις μεθόδους cross-entropy και exponential, ενω για την hinge μηδενίστηκε. Τέλος, στον πίνακα 2 φαίνονται τα σφάλματα ανά υπόθεση και το συνολικό ποσοστό σφάλματος για κάθε μέθοδο.

	Cross-Entropy	Exponential	Hinge
$P(D_0 \mid H_1)$	0.5%	0.4%	0.3%
$P(D_1 \mid H_0)$	0.2%	0.2%	0.2%
Συνολικό Σφάλμα	0.7%	0.6%	0.5%

Πίνακας 2: Ποσοστιαία σφάλματα κατηγοριοποίησης για όλες τις μεθόδους που χρησιμοποιήθηκαν.

3 Παράρτημα

3.1 Το νευρωνικό δίκτυο

Στην εργασία αυτή χρησιμοποιείται ένα "ρηχό" (shallow) νευρωνικό δίκτυο πλήρως συνδεδεμένο, με 1 κρυφό επίπεδο και ένα επίπεδο εξόδου. Η συμπεριφορά του δικτύου περιγράφεται από τις παρακάτω εξισώσεις, όπου N_{in} είναι η διάσταση του διανύσματος εισόδου $\mathbf{x}$ και N_n ο αριθμός των νευρώνων του κρυφού επιπέδου. Στις εξισώσεις παρακάτω διανυσματικές ποσότητες όπως και πίνακες συμβολίζονται με bold χαρακτήρες, ενώ οι scalar ποσότητες με κοινά γράμματα.

$$\mathbf{h_1} = \mathbf{A}\mathbf{x} + \mathbf{a} \tag{6}$$

,όπου $\mathbf{x} \in \Re^{N_{in}\times 1}$, $\mathbf{A} \in \Re^{N_n \times N_{in}}$, $\mathbf{a} \in \Re^{N_n \times 1}$ και $\mathbf{h1} \in \Re^{N_n \times 1}$.

$$\mathbf{z} = K(\mathbf{h_1}) \tag{7}$$

,όπου $\mathbf{z} \in \Re^{N_n \times 1}$ και K scalar συνάρτηση. Το διάνυσμα $\mathbf{z}$ αποτελεί την έξοδο του κρυφού επιπέδου. Σαν μη γραμμική συνάρτηση χρησιμοποιούμε την:

$$K(\mathbf{h_1}) = ReLU(\mathbf{h_1}) \tag{8}$$

Έπειτα για το επίπεδο εξόδου έχουμε:

$$h_2 = \mathbf{B}\mathbf{z} + b \tag{9}$$

,όπου $\mathbf{B} \in \Re^{1\times N_n}$ και $h_2, b \in \Re$.

Τελικά η έξοδος του δικτύου είναι:

$$u = g(h_2) \tag{10}$$

,όπου g κατάλληλη scalar συνάρτηση ανάλογα με τη λύση του προβλήματος βελτιστοποίησης που λύνουμε σε κάθε περίπτωση. Η συνάρτηση αυτή εξασφαλίζει ότι το πεδίο τιμών της συνάρτησης $\omega(r)$ του λόγου πιθανοφάνειας που προσεγγίζει το νευρωνικό ταυτίζεται με το πεδίο τιμών της εξόδου του.

3.2 Το πρόβλημα βελτιστοποίησης

Στόχος μας είναι να προσεγγίσουμε με το νευρωνικό δίκτυο μια συνάρτηση $\omega(r)$ του λόγου πιθανοφάνειας $r = \frac{f_1(\mathbf{x})}{f_0(\mathbf{x})}$. Αυτό μπορεί να επιτευχθεί, όπως αναφέρθηκε στο μάθημα και [3] ελαχιστοποιώντας την συνάρτηση:

$$\widetilde{J}(u(\mathbf{x},\boldsymbol{\theta})) = E_0[\Phi(u(\mathbf{x},\boldsymbol{\theta}))] + E_1[\Psi(u(\mathbf{x},\boldsymbol{\theta}))] \tag{11}$$

η οποία ορίζοντας:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_0} \\ \mathbf{x_1} \end{bmatrix} \tag{12}$$

μπορεί να γραφεί ώς:

$$\widetilde{J}(u(\mathbf{x},\boldsymbol{\theta})) = E_{\mathbf{x}}[\Phi(u(\mathbf{x^0},\boldsymbol{\theta})) + \Psi(u(\mathbf{x^1},\boldsymbol{\theta}))] \tag{13}$$

όπου $\mathbf{x^i}$ διάνυσμα από την υπόθεση H_i ,i = 0,1 και $u(\mathbf{x}, \boldsymbol{\theta})$ η έξοδος του νευρωνικού δικτύου με σύνολο παραμέτρων $\boldsymbol{\theta}$. Επειδή προσπαθούμε μέσω δεδομένων κάθε φορά να ελαχιστοποιήσουμε την συνάρτηση κόστους θα χρησιμοποιήσουμε Stochastic Gradient Descent και θα ελαχιστοποιήσουμε αντί για την $\widetilde{J}(u)$ την:

$$J(u(\mathbf{x}, \boldsymbol{\theta})) = \Phi(u(\mathbf{x^0}, \boldsymbol{\theta})) + \Psi(u(\mathbf{x^1}, \boldsymbol{\theta})) \tag{14}$$

Συνεπώς η σχέση ανανέωσης των παραμέτρων θ του δικτύου είναι:

$$\begin{aligned} \boldsymbol{\theta}_t =& \boldsymbol{\theta}_{t-1} - \mu \nabla J(u(\mathbf{x}, \boldsymbol{\theta})) \\ =& \boldsymbol{\theta}_{t-1} - \mu \nabla_{\boldsymbol{\theta}} \{\Phi(u(\mathbf{x^0}, \boldsymbol{\theta})) + \Psi(u(\mathbf{x^1}, \boldsymbol{\theta}))\} \end{aligned} \tag{15}$$

3.3 Υπολογισμός των Gradient

Δεδομένου οτι το σύνολο παραμέτρων του δικτύου είναι:

$$\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{a}, \mathbf{B}, b\} \tag{16}$$

τα αντίστοιχα gradient που θα πρέπει να υπολογιστούν για το δίκτυο είναι:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{B}} =& \frac{\partial J}{\partial u} \frac{\partial u}{\partial h_2} \frac{\partial h_2}{\partial \mathbf{B}} \\ \frac{\partial J}{\partial b} =& \frac{\partial J}{\partial u} \frac{\partial u}{\partial h_2} \frac{\partial h_2}{\partial b} \\ \frac{\partial J}{\partial \mathbf{A}} =& \frac{\partial J}{\partial u} \frac{\partial u}{\partial h_2} \frac{\partial h_2}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{h_1}} \frac{\partial \mathbf{h_1}}{\partial \mathbf{A}} \\ \frac{\partial J}{\partial \mathbf{a}} =& \frac{\partial J}{\partial u} \frac{\partial u}{\partial h_2} \frac{\partial h_2}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{h_1}} \frac{\partial \mathbf{h_1}}{\partial \mathbf{a}} \end{aligned} \tag{17}$$

όπου ορίζοντας για ευκολία:

$$\boldsymbol{\delta_1} = \frac{\partial J}{\partial \mathbf{h_1}} = \frac{\partial J}{\partial u} \frac{\partial u}{\partial h_2} \frac{\partial h_2}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{h_1}} = \delta_2 \frac{\partial h_2}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{h_1}} \tag{18}$$

$$\delta_2 = \frac{\partial J}{\partial h_2} = \frac{\partial J}{\partial u} \frac{\partial u}{\partial h_2} \tag{19}$$

γίνονται:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{B}} =& \delta_2 \frac{\partial h_2}{\partial \mathbf{B}} \\ \frac{\partial J}{\partial b} =& \delta_2 \frac{\partial h_2}{\partial b} \\ \frac{\partial J}{\partial \mathbf{A}} =& \boldsymbol{\delta_1} \frac{\partial \mathbf{h_1}}{\partial \mathbf{A}} \\ \frac{\partial J}{\partial \mathbf{a}} =& \boldsymbol{\delta_1} \frac{\partial \mathbf{h_1}}{\partial \mathbf{a}} \end{aligned} \tag{20}$$

Όπου χρησιμοποιώντας τις σχέσεις διαφόρισης διανυσματικών ποσοτήτων [2],

φροντίζοντας οι διαστάσεις του προκύπτοντος gradient να είναι ίδιες με αυτές της εκάστοτε παραμέτρου, παίρνουμε:

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{B}} =& \delta_2 \cdot \mathbf{z}^T \\ \frac{\partial J}{\partial b} =& \delta_2 \\ \frac{\partial J}{\partial \mathbf{A}} =& \boldsymbol{\delta_1}^T \cdot \mathbf{x}^T \\ \frac{\partial J}{\partial \mathbf{a}} =& \boldsymbol{\delta_1}^T\end{aligned} \tag{21}$$

Μένει μόνο να υπολογίσουμε τα $\boldsymbol{\delta_1}$ και δ_2. Για το $\boldsymbol{\delta_1}$ έχουμε:

$$\begin{aligned}\boldsymbol{\delta_1} =& \delta_2 \frac{\partial h_2}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{h_1}} \\ =& \delta_2 \cdot \mathbf{B} \cdot diag(K'(\mathbf{h_1})) \\ =& \delta_2 \cdot \mathbf{B} \cdot diag(step(\mathbf{h_1}))\end{aligned} \tag{22}$$

Οπου step βηματική συνάρτηση Heaviside:

$$step(x) = \frac{\partial\{ReLU(x)\}}{\partial x} = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{23}$$

Η παράμετρος δ_2 εξαρτάται από την cost function:

$$J(u(\mathbf{x}, \boldsymbol{\theta})) = \begin{cases} \Phi(u(\mathbf{x}, \boldsymbol{\theta})) & , \mathbf{x} \in H_0 \\ \Psi(u(\mathbf{x}, \boldsymbol{\theta})) & , \mathbf{x} \in H_1 \end{cases} \tag{24}$$

Οπότε από τη σχέση 19 αρκεί να υπολογίσουμε τα:

$$\frac{\partial J}{\partial u} = \begin{cases} \frac{\partial \Phi(u(\mathbf{x}, \boldsymbol{\theta}))}{\partial u} & , \mathbf{x} \in H_0 \\ \frac{\partial \Psi(u(\mathbf{x}, \boldsymbol{\theta}))}{\partial u} & , \mathbf{x} \in H_1 \end{cases} \tag{25}$$

$$\frac{\partial u}{\partial h_2} = \frac{\partial g(h_2)}{\partial h_2} = g'(h_2) \tag{26}$$

Τελικά, για τις διάφορες μεθόδους που χρησιμοποιήθηκαν έχουμε:

1. **Cross Entropy**: Εκτιμά το $\omega(r) = \frac{r}{1+r}$, οπότε χρησιμοποιήθηκε η σιγμοειδής activation function $g(h_2)$ και σαν κατώφλι τέθηκε το $\frac{1}{2}$.

$$\begin{aligned}\Phi(u) = -log(1-u) \Rightarrow & \frac{\partial \Phi(u)}{\partial u} = \frac{1}{1-u} \\ \Psi(u) = -log(u) \Rightarrow & \frac{\partial \Psi(u)}{\partial u} = -\frac{1}{u} \\ g(h_2) = \frac{1}{1+e^{-h_2}} \Rightarrow & g'(h_2) = \frac{e^{h_2}}{(1+e^{h_2})^2} = g(h_2) \cdot g(-h_2)\end{aligned} \tag{27}$$

2. **Exponential**: Εκτιμά το $\omega(r) = log(r)$, οπότε χρησιμοποιήθηκε η γραμμική activation function $g(h_2)$ και σαν κατώφλι τέθηκε το 0.

$$\begin{aligned} \Phi(u) = e^{0.5u} &\Rightarrow \frac{\partial \Phi(u)}{\partial u} = \frac{e^{0.5u}}{2} \\ \Psi(u) = e^{-0.5u} &\Rightarrow \frac{\partial \Psi(u)}{\partial u} = -\frac{e^{-0.5u}}{2} \\ g(h_2) = h_2 &\Rightarrow g'(h_2) = 1 \end{aligned} \tag{28}$$

3. **Hinge**: Εκτιμά το $\omega(r) = sign(log(r))$ [3], οπότε χρησιμοποιήθηκε η γραμμική activation function $g(h_2)$ και σαν κατώφλι τέθηκε το 0.

$$\begin{aligned} \Phi(u) = max\{1+u, 0\} &\Rightarrow \frac{\partial \Phi(u)}{\partial u} = \begin{cases} 1 & , u \geq -1 \\ 0 & , u < -1 \end{cases} \\ \Psi(u) = max\{1-u, 0\} &\Rightarrow \frac{\partial \Psi(u)}{\partial u} = \begin{cases} -1 & , u \leq 1 \\ 0 & , u > 1 \end{cases} \\ g(h_2) = h_2 &\Rightarrow g'(h_2) = 1 \end{aligned} \tag{29}$$

Βιβλιογραφία

[1] Sid H (2020). *Read digits and labels from MNIST database*. URL: https://www.mathworks.com/matlabcentral/fileexchange/27675-read-digits-and-labels-from-mnist-database.

[2] Kevin Clark. *Computing Neural Network Gradients*. URL: https://web.stanford.edu/class/cs224n/readings/gradient-notes.pdf.

[3] George V. Moustakides and Kalliopi Basioti. *Training Neural Networks for Likelihood/Density Ratio Estimation*. 2019. arXiv: 1911.00405 [eess.SP].