Πολυτεχνείο Κρήτης
Σχολή ΗΜΜΥ
Στατιστική Μοντελοποίηση και Αναγνώριση Προτύπων
Φυλλάδιο Ασκήσεων 2
Ομάδα Χρηστών 90

Επώνυμο	**Όνομα**	**Α.Μ.**
Νικολός	Κωνσταντίνος	2019030096
Μπέκος	Κωνσταντίνος	2019030082

Θέμα 1: Αλγόριθμος Perceptron

a) Καταρχάς, σχεδιάζονται τα δείγματα των τεσσάρων (4) κλάσεων, στο επίπεδο, με διαφορετικό χρώμα για κάθε κλάση.

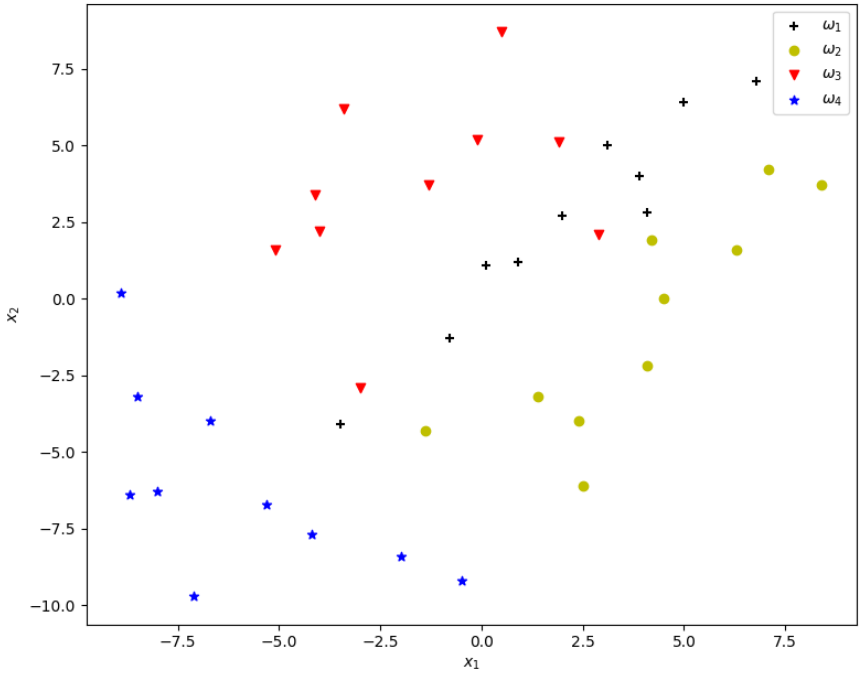


Figure 1: Samples (2D) of 4 classes

b) Στη συγκεκριμένη υλοποίηση του αλγορίθμου *Batch Perceptron*, το διάνυσμα των βαρών ($\mathbf{w}$) αρχικοποιείται με μηδενικές τιμές. Μόλις ολοκληρωθεί η εκτέλεσή του, τυπώνεται ο αριθμός των επαναλήψεων που πραγματοποιήθηκαν.

c) Ο αριθμός των επεναλήψεων, για τρεις περιπτώσεις προβλημάτων ταξινόμησης, διαμορφώνεται ως εξής:

1. για τις κλάσεις ω_1 και ω_2: 24
2. για τις κλάσεις ω_2 και ω_3: 17
3. για τις κλάσεις ω_3 και ω_4: 40

Παρατηρούμε, ότι η διαφορά στον αριθμό των επαναλήψεων, που χρειάζεται, για να συγκλίνει ο αλγόριθμος, στις παραπάνω περιπτώσεις, θα μπορούσε να εξαρτάται από τον τρόπο που είναι διεσπαρμένα τα σημεία των δύο κλάσεων στο επίπεδο. Πιο συγκεκριμένα, τα δείγματα των κλάσεων ω_2 και ω_3 είναι με παρόμοιο τρόπο διασκορπισμένα και απαιτούν το μικρότερο αριθμό επαναλήψεων, ενώ αυτά των κλάσεων ω_3 και ω_4 απλώνονται σε ευθείες με διαφορετικές κλίσεις και αναγκάζουν τον αλγόριθμο να κάνει τις περισσότερες επαναλήψεις.

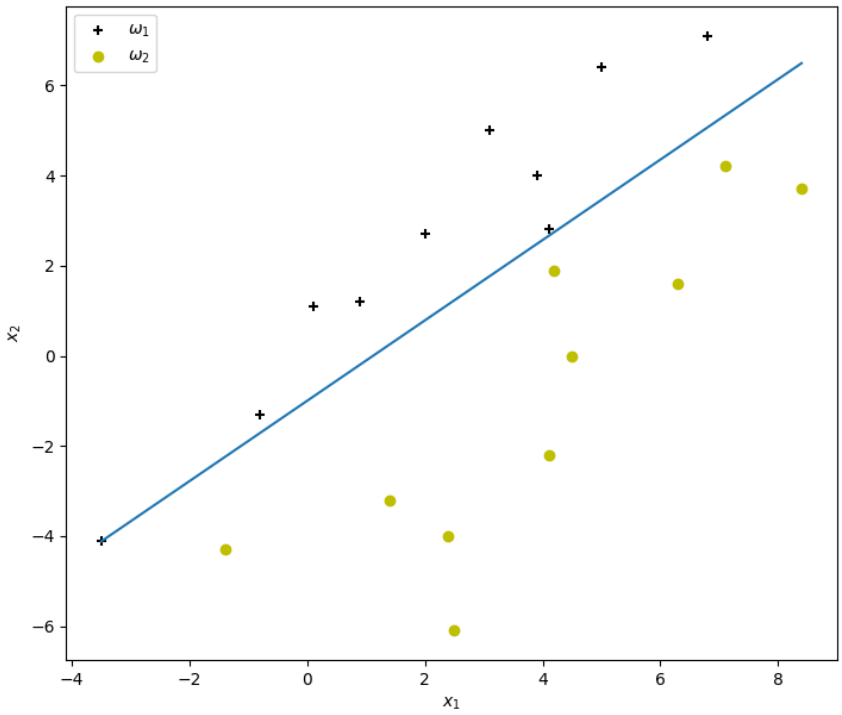


Figure 2: Decision Boundary for classes ω_1 and ω_2 (Perceptron algorithm)

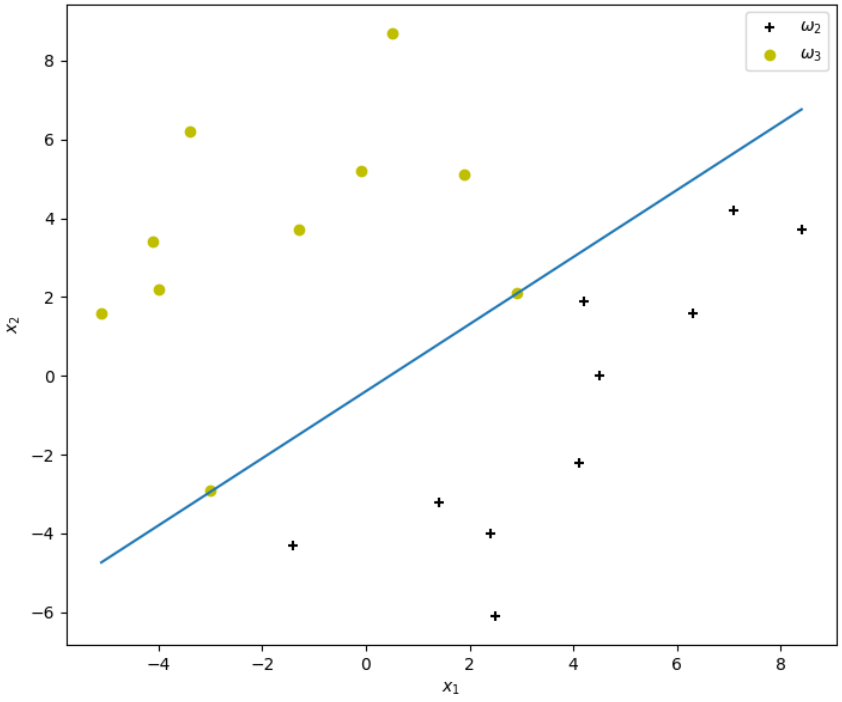


Figure 3: Decision Boundary for classes ω_2 and ω_3 (Perceptron algorithm)

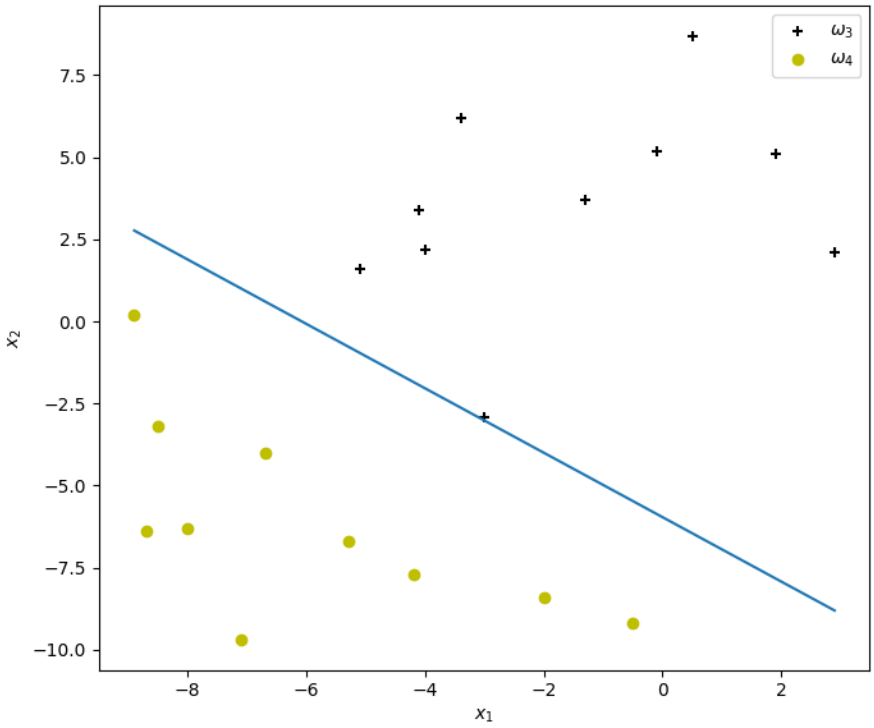


Figure 4: Decision Boundary for classes ω_3 and ω_4 (Perceptron algorithm)

Θέμα 2: Λογιστική Παλινδρόμηση: Αναλυτική εύρεση κλίσης (Gradient)

Εξισώσεις της συνάρτησης λογιστικής παλινδρόμησης, της λογιστικής συνάρτησης και της εκτίμησης για την κλάση κάθε δείγματος:

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = f(\boldsymbol{\theta}^T \mathbf{x})$$
$$f(z) = \frac{1}{1 + 2e^{-z}}$$
$$\hat{y}^{(i)} = h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$$

Συνάρτηση κόστους (cross-entropy):

$$
\begin{aligned}
J(\boldsymbol{\theta}) &= \frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\ln(\hat{y}^{(i)}) - (1-y^{(i)})\ln(1-\hat{y}^{(i)}) \\
&= \frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\ln(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) - (1-y^{(i)})\ln(1-h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) \\
&= \frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\ln(f(\boldsymbol{\theta}^T\mathbf{x}^{(i)})) - (1-y^{(i)})\ln(1-f(\boldsymbol{\theta}^T\mathbf{x}^{(i)})) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\ln\left(\frac{1}{1+2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}}\right) - (1-y^{(i)})\ln\left(1-\frac{1}{1+2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}}\right)\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\ln(1+2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}) + (y^{(i)}-1)\ln\left(\frac{2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}}{1+2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}}\right)\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\ln(2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}) - \ln\left(\frac{2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}}{1+2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}}\right)\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}((y^{(i)}-1)\ln(2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}) + \ln(1+2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}})) \\
&= \frac{1}{m}\sum_{i=1}^{m}((y^{(i)}-1)(\ln(2)-\boldsymbol{\theta}^T\mathbf{x}^{(i)}) + \ln(1+2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}))
\end{aligned}
$$

Το j-στοιχείο της κλίσης του σφάλματος είναι:

$$
\begin{aligned}
\frac{\partial J(\boldsymbol{\theta})}{\partial\theta_j} &= \frac{1}{m}\sum_{i=1}^{m}\left((y^{(i)}-1)\frac{\partial}{\partial\theta_j}\left(\ln(2)-\boldsymbol{\theta}^T\mathbf{x}^{(i)}\right) + \frac{\partial}{\partial\theta_j}\left(\ln(1+2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}})\right)\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left((y^{(i)}-1)(-x_j^{(i)}) - x_j^{(i)}\left(\frac{2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}}{1+2e^{-\boldsymbol{\theta}^T\mathbf{x}^{(i)}}}\right)\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}\left((y^{(i)}-1)(-x_j^{(i)}) - x_j^{(i)}(1-h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}))\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}x_j^{(i)}(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})-y^{(i)})
\end{aligned}
$$

- Η λογιστική παλινδρόμηση χρησιμοποιείται, για να προβλέψουμε, αν ένας φοιτητής θα γίνει δεκτός, σε ένα πανεπιστήμιο, με βάση τους βαθμούς του, σε δύο εξετάσεις.
- Τα δεδομένα του προβλήματος σχεδιάζονται στο επίπεδο. Οι συντεταγμένες του κάθε σημείου είναι ένα ζεύγος βαθμών, στις δύο εξετάσεις, κάποιου φοιτητή. Τα δύο διαφορετικά χρώματα αντιστοιχούν στην κλάση του αντίστοιχου δείγματος (αποδοχή ή απόρριψη).
- Με την ολοκλήρωση της εκτέλεσης του αλγορίθμου, έχει υπολογιστεί το όριο απόφασης, για τις δύο κλάσεις. Τα δείγματα δεν είναι γραμμικά διαχωρίσιμα, αλλά ο αλγόριθμος πετυχαίνει ακρίβεια εκπαίδευσης 92%.
- Για τον υπολογισμό της ακρίβειας εκπαίδευσης, καλείται η συνάρτηση `predict`, ώστε να υπολογιστεί η πρόβλεψη της κλάσης κάθε δείγματος ($\hat{y}^{(i)}$) και να συγκριθεί με την πραγματική κλάση του ($y^{(i)}$). Η συνάρτηση `predict` υπολογίζει την τιμή της σιγμοειδούς (λογιστικής) συνάρτησης, μέσω της συνάρτησης `sigmoid`, χρησιμοποιώντας την τιμή της παραμέτρου $\boldsymbol{\theta}$ που υπολογίζεται από τον αλγόριθμο, καθώς και ένα διάνυσμα χαρακτηριστικών ($\mathbf{x}^{(i)}$). Κατόπιν, ταξινομεί το αντίστοιχο δείγμα στην κλάση αποδοχής (1), αν η τιμή που επιστρέφεται είναι μεγαλύτερη από $\frac{1}{3}$. Η τιμή αυτή του κατωφλίου ισούται με την τιμή της συγκεκριμένης συγμοειδούς συνάρτησης, στο σημείο 0 ($f(0) = \frac{1}{1+2e^{-0}} = \frac{1}{3}$).
- Η πιθανότητα να γίνει δεκτός ένας φοιτητής, με βαθμολογίες 45 και 85, στις δύο εξετάσεις, υπολογίζεται, μέσω της συνάρτησης `sigmoid`, ίση με: 0.7762933285009056.

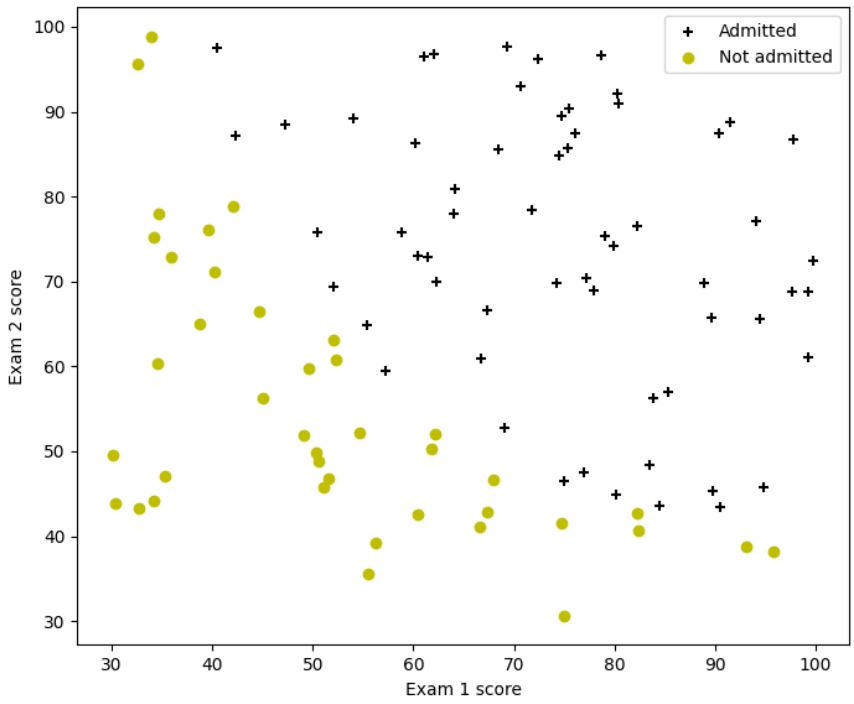


Figure 5: Samples of 2 classes

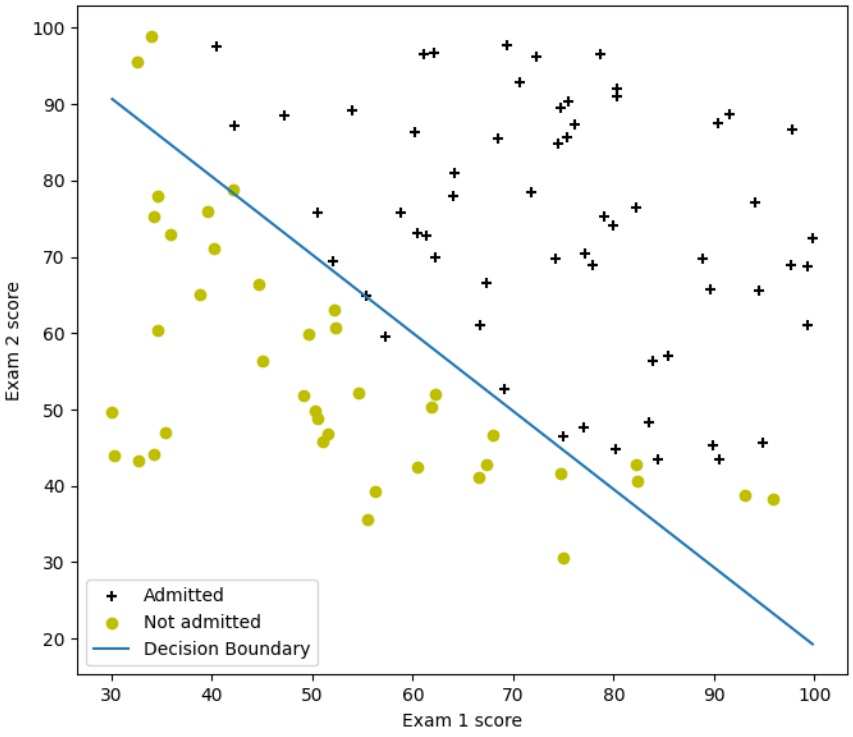


Figure 6: Decision Boundary for 2 classes (Logistic Regression)

Θέμα 3: Εκτίμηση Παραμέτρων με Maximum Likelihood

1. Ο τύπος για τον υπολογισμό της μέσης τιμής μ_{MLE}, για 1-D κατανομή, είναι:

$$\mu_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

ενώ, για τη διασπορά, είναι:

$$\sigma^2_{\text{MLE}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{\text{MLE}})^2$$

Χωρίζουμε τα δεδομένα της κλάσης ω_1 στα 3 διαφορετικά χαρακτηριστικά x_i και υπολογίζουμε για το κάθε ένα ξεχωριστά:

   (a) Για το δείγμα x_1:
      i. $\mu_{\text{MLE}} = -0.070899$
      ii. $\sigma^2_{\text{MLE}} = 0.906177$
   (b) Για το δείγμα x_2:
      i. $\mu_{\text{MLE}} = -0.6047$
      ii. $\sigma^2_{\text{MLE}} = 4.2007148$
   (c) Για το δείγμα x_3:
      i. $\mu_{\text{MLE}} = -0.9109999$
      ii. $\sigma^2_{\text{MLE}} = 4.5419949$

2. Χωρίζουμε τα δεδομένα σε ζευγάρια δύο χαρακτηριστικών και υπολογίζουμε για το κάθε ένα τις παραμέτρους:
Μέση τιμή

$$\boldsymbol{\mu}_{\text{MLE}} = \begin{bmatrix}\mu_1 & \mu_2\end{bmatrix}^T$$

Πίνακας συνδιακύμανσης

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \boldsymbol{\mu}_{\text{MLE}})(\mathbf{x}^{(i)} - \boldsymbol{\mu}_{\text{MLE}})^T$$

Για τον υπολογισμό του πίνακα συνδιακύμνασης Σ, χρησιμοποιούνται οι διακυμάνσεις

$$\text{var}(x_j) = \frac{1}{N}\sum_{i=1}^{N}(x_j^{(i)} - \mu_j)^2$$

και η συνδιακύμανση μεταξύ δύο μεταβλητών

$$\text{cov}(x_1, x_2) = \frac{1}{N}\sum_{i=1}^{N}(x_1^{(i)} - \mu_1)(x_2^{(i)} - \mu_2)$$

   (a) Για τα δείγματα $x_1,\ x_2$:

$$\boldsymbol{\mu}_{\text{MLE}} = \begin{bmatrix}\mu_{x_1} & \mu_{x_2}\end{bmatrix}^T = \begin{bmatrix}-0.709 & -0.6047\end{bmatrix}^T$$

   (b) Για τα δείγματα $x_1,\ x_3$:

$$\boldsymbol{\mu}_{\text{MLE}} = \begin{bmatrix}\mu_{x_1} & \mu_{x_3}\end{bmatrix}^T = \begin{bmatrix}-0.709 & -0.911\end{bmatrix}^T$$

   (c) Για τα δείγματα $x_2,\ x_3$:

$$\boldsymbol{\mu}_{\text{MLE}} = \begin{bmatrix}\mu_{x_2} & \mu_{x_3}\end{bmatrix}^T = \begin{bmatrix}-0.6046 & -0.911\end{bmatrix}^T$$

   (d) Ενώ οι διακυμάνσεις τους προκύπτουν:

$$\text{var}(x_1) = 0.906177$$
$$\text{var}(x_2) = 4.2007148$$
$$\text{var}(x_3) = 4.5419949$$

3. Πλέον χρησιμοποιούμε και τα τρία χαρακτηριστικά που δίνονται στα δεδομένα της κλάσης ω_1, υπολογίζοντας της παραμέτρους της 3-D κατανομής, όπως και στο προηγούμενο ερώτημα:

$$\boldsymbol{\mu}_{\text{MLE}} = \begin{bmatrix}\mu_{x_1} & \mu_{x_2} & \mu_{x_3}\end{bmatrix}^T = \begin{bmatrix}-0.709 & -0.6047 & -0.911\end{bmatrix}^T$$
$$\text{var}(x_1) = 0.90617729$$
$$\text{var}(x_2) = 4.20071481$$
$$\text{var}(x_3) = 4.541949000000001$$

4. Εκτελούμε την ίδια διαδικασία, για τα δεδομένα της κλάσης ω_2, υποθέτοντας, ότι το 3-D μοντέλο είναι διαχωρίσιμο και τα χαρακτηριστικά x_i δεν συσχετίζονται μεταξύ τους. Ως αποτέλεσμα, ο πίνακας συνδιακύμανσης Σ προκύπτει διαγώνιος, με τις τιμές $\text{var}(x_i)$ ενώ οι υπόλοιπες τιμές, που αφορούν τις συνδιακυμάνσεις μεταξύ δύο τιμών $\text{cov}(x_1, x_2)$, γίνονται ίσες με μηδέν.

$$\boldsymbol{\mu}_{\text{MLE}} = \begin{bmatrix}\mu_{x_1} & \mu_{x_2} & \mu_{x_3}\end{bmatrix}^T = \begin{bmatrix}-0.1126 & 0.4299 & 0.00372\end{bmatrix}^T$$
$$\text{var}(x_1) = 0.05392$$
$$\text{var}(x_2) = 0.04597$$
$$\text{var}(x_3) = 0.00726$$

5. Παρατηρούμε, ότι, και στα τρία πρώτα ερωτήματα, που χρησιμοποιούμε τα δεδομένα της κλάσης ω_1, παρά την διαφορά της διάστασης της κατανομής, η μέγιστη πιθανοφάνεια της μέσης τιμής και της διασποράς είναι (αναμενόμενα) η ίδια, για κάθε χαρακτηριστικό x_i, που υπολογίζεται και στις τρεις περιπτώσεις, αφού χρησιμοποιούνται οι ίδιοι τύποι για τον υπολογισμό τους. Όσον αφορά το 4ο ερώτημα, όπου τα χαρακτηριστικά είναι ανεξάρτητα, χρησιμοποιούμε διαφορετικά δεδομένα (κλάση ω_2), επομένως τα αποτελέσματα δεν είναι συγκρίσιμα με τις προηγούμενες περιπτώσεις. Όμως, αν και σε αυτή την περίπτωση χρησιμοποιούσαμε τα δεδομένα της κλάσης ω_1, πάλι θα προκύπταν κοινά αποτελέσματα με τα προηγούμενα ερωτήματα.

Θέμα 4: Ομαδοποίηση (Clustering) με K-means και GMM

Ο αλγόριθμος *K-means* εφαρμόστηκε σε δύο εικόνες, για τρεις (3) διαφορετικούς αριθμούς κλάσεων (χρωμάτων): 16, 32 και 64. Φυσικά, η αύξηση του αριθμού των κλάσεων οδηγεί σε αύξηση της ποιότητας της εικόνας, ενώ η μείωσή του έχει ως αποτέλεσμα τη μείωση του μεγέθους της συμπιεσμένης εικόνας. Πιο συγκεκριμένα, αν ο αριθμός κλάσεων είναι k, τότε, για κάθε pixel της συμπιεσμένης εικόνας, μπορούμε να αποθηκεύσουμε ένα μοναδικό αριθμό (δείκτη), που να αντιστοιχεί στην κλάση (χρώμα) του συγκεκριμένου pixel. Συνεπώς, αυτός ο μοναδικός δείκτης θα έχει μέγεθος σε bits: $\log_2 k$. Για τους αριθμούς των κλάσεων που αναφέρθηκαν προηγουμένως και χρησιμοποιούνται στην παρούσα άσκηση, ο αριθμός των bits που απαιτούνται, για την αποθήκευση ενός pixel μιας συμπιεσμένης εικόνας, είναι: 4 bits/pixel, 5 bits/pixel και 6 bits/pixel. Παρατηρούμε, ότι, με το διπλασιασμό του πλήθους των κλάσεων (χρωμάτων), που χρησιμοποιούνται για τη συμπίεση, ο όγκος της πληροφορίας ανά pixel αυξάνεται κατά ένα μόνο bit, ενώ, παράλληλα, είναι εμφανής η βελτίωση της ποιότητας του αποτελέσματος.

Θέμα 5α: Υλοποίηση ενός απλού νευρωνικού δικτύου

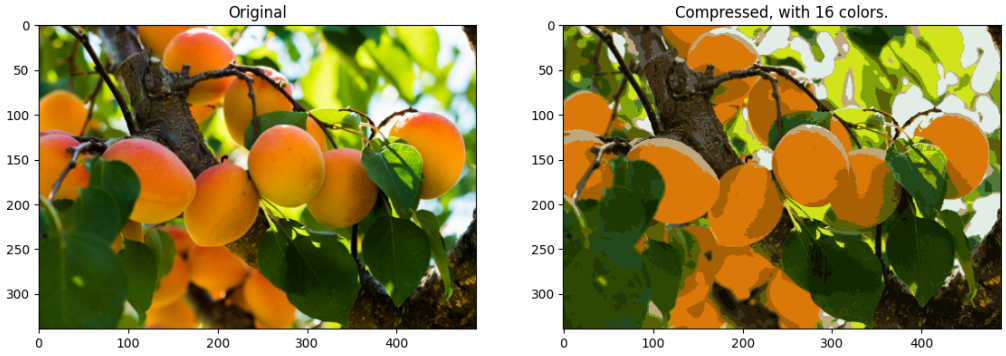
Μέρος Α

α) Γνωρίζουμε ότι

$$\hat{y}^{(i)} = \frac{1}{1 + e^{-z^{(i)}}} = f(z^{(i)})$$

Αντικαθιστούμε στην αρχική συνάρτηση της cross-entropy την

$$1 - \hat{y}^{(i)} = \frac{e^{-z^{(i)}}}{1 + e^{-z^{(i)}}}$$

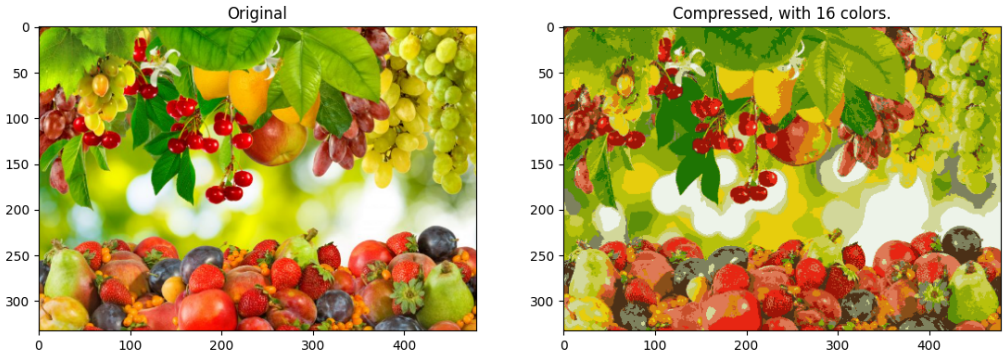


(a) Compression using 16 colors

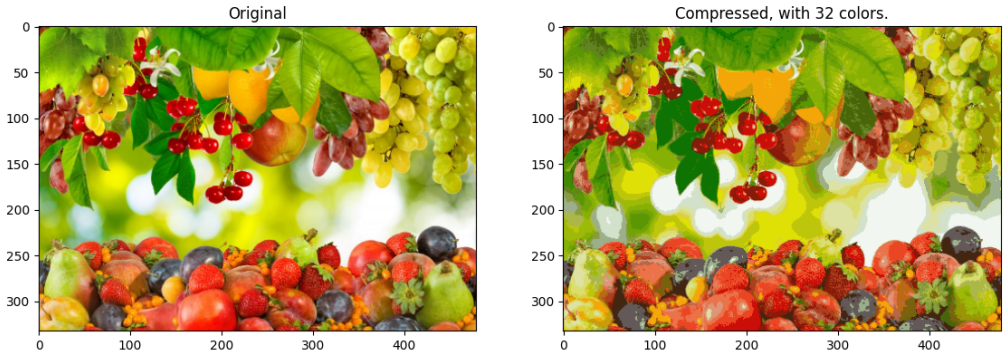
(b) Compression using 32 colors

(c) Compression using 64 colors

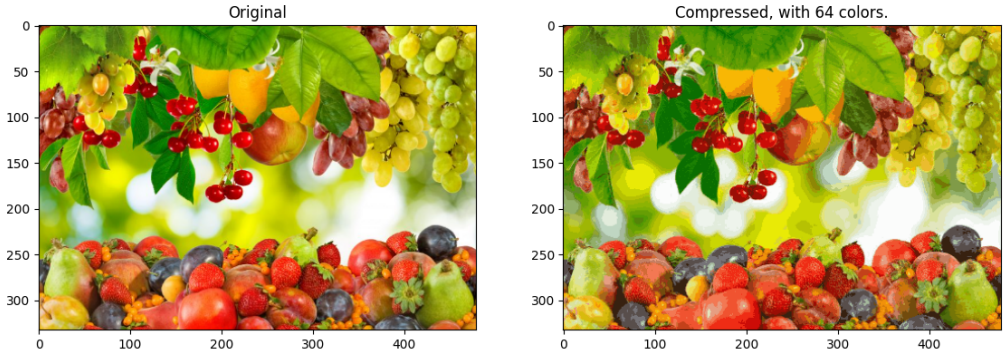
Figure 7: Applying *K-means* algorithm for Image Compression



(a) Compression using 16 colors



(b) Compression using 32 colors



(c) Compression using 64 colors

Figure 8: Applying *K-means* algorithm for Image Compression

και προκύπτει:

$$J(y^{(i)}, \hat{y}^{(i)}; W, b) = -y^{(i)} \ln\left(\frac{1}{1+e^{-z^{(i)}}}\right) - (1-y^{(i)}) \ln\left(\frac{e^{-z^{(i)}}}{1+e^{-z^{(i)}}}\right)$$
$$= y^{(i)} \ln(1+e^{-z^{(i)}}) + (1-y^{(i)})(z^{(i)} + \ln(1+e^{-z^{(i)}}))$$
$$= z^{(i)} - y^{(i)} z^{(i)} + \ln(1+e^{-z^{(i)}}))$$

Επομένως

$$J(Y, \hat{Y}; W, b) = \frac{1}{B} \sum_i (z^{(i)} - y^{(i)} z^{(i)} + \ln(1+e^{-z^{(i)}})))$$

β) Σύμφωνα με τον κανόνα της αλυσίδας:

$$\frac{\partial J}{\partial z^{(i)}} = \frac{\partial J}{\partial \hat{y}^{(i)}} \frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}}$$

Όπου

$$\frac{\partial J}{\partial \hat{y}^{(i)}} = -\frac{y^{(i)}}{\hat{y}^{(i)}} + \frac{1-y^{(i)}}{1-\hat{y}^{(i)}}$$

και

$$\frac{\partial \hat{y}^{(i)}}{\partial z^{(i)}} = \hat{y}^{(i)}(1-\hat{y}^{(i)})$$

Άρα

$$\frac{\partial J}{\partial z^{(i)}} = \left(-\frac{y^{(i)}}{\hat{y}^{(i)}} + \frac{1-y^{(i)}}{1-\hat{y}^{(i)}}\right) \hat{y}^{(i)}(1-\hat{y}^{(i)})$$
$$= -(1-\hat{y}^{(i)}) y^{(i)} + \hat{y}^{(i)}(1-y^{(i)})$$
$$= -y^{(i)} + \hat{y}^{(i)}$$

γ) Σύμφωνα με τον κανόνα της αλυσίδας:

$$\frac{\partial J}{\partial W} = \frac{\partial J}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial W}$$

Όπου

$$\frac{\partial J}{\partial z^{(i)}} = -y^{(i)} + \hat{y}^{(i)}$$

και

$$\frac{\partial z^{(i)}}{\partial W} = (x^{(i)})^T$$

Επομένως

$$\frac{\partial J}{\partial W} = (-y^{(i)} + \hat{y}^{(i)})(x^{(i)})^T$$

Σύμφωνα με τον κανόνα της αλυσίδας:

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial b}$$

Όπου

$$\frac{\partial z^{(i)}}{\partial b} = 1$$

Επομένως

$$\frac{\partial J}{\partial b} = -y^{(i)} + \hat{y}^{(i)}$$

δ) Για την διαδικασία backpropagation:
Αρχικά, εκτελείται ένα forward pass, υπολογίζοντας την τιμή της εξόδου $\hat{y}^{(i)}$. Ξεκινώντας από το τελευταίο επίπεδο, υπολογίζεται η συνάρτηση σφάλματος (cross-entropy), για το σφάλμα στην έξοδο. Έχοντας υπολογίσει το σφάλμα, υπολογίζονται οι μερικές παράγωγοι $\frac{\partial J}{\partial W}$, $\frac{\partial J}{\partial b}$, οι οποίες χρησιμοποιούνται για την ενημέρωση των βαρών στο τελευταίο επίπεδο L. Στη συνέχεια, κινούμαστε προς τα πίσω, στο προτελευταίο επίπεδο, υπολογίζοντας το σφάλμα,χρησιμοποιώντας τα σφάλματα του προηγούμενου επιπέδου και υπολογίζουμε τις μερικές παραγώγους, ώστε να ενημερώσουμε τα βάρη του προτελευταίου επιπέδου. Η ίδια διαδικασία επαναλαμβάνεται μέχρι να φτάσουμε στο πρώτο επίπεδο.
Για να ενημερωθούν τα βάρη σε ένα επίπεδο χρησιμοποιούνται οι παρακάτω σχέσεις:

$$W^{\text{new}} = W^{\text{old}} - \rho \frac{\partial J}{\partial W}$$
$$b^{\text{new}} = b^{\text{old}} - \rho \frac{\partial J}{\partial b}$$

Όπου ρ = learning rate, $\frac{\partial J}{\partial W} = (-y^{(i)} + \hat{y}^{(i)})(x^{(i)})^T$, $\frac{\partial J}{\partial b} = -y^{(i)} + \hat{y}^{(i)}$, όπως υπολογίστηκαν πριν.

Μέρος Β

Υλοποίηση Νευρωνικού Δικτύου με 2 Layers

Για την υλοποίηση του νευρωνικού δικτύου, χρησιμοποιούνται δύο dense layers, σύμφωνα με τον κώδικα της άσκησης που δόθηκε. Ως Activation Function του πρώτου Layer, χρησιμοποιείται η λογική συνάρτηση, ενώ, για το δεύτερο layer εξόδου, χρησιμοποιείται η Softmax συνάρτηση, καταλήγοντας σε πιθανότητες που αθροίζουν στο 1. Για τον υπολογισμό του σφάλματος και την ενημέρωση των βαρών, μέσω της διαδικασίας backpropagaton, χρησιμοποιείται η συνάρτηση Loss Function Cross-Entropy. Αφού υλοποιήθηκε το δίκτυο, ακολούθησε η αλλαγή ορισμένων παραμέτρων, με στόχο την εκτίμησή τους, όσον αφορά το ratio.
Αποτελέσματα:

Learning Rate	#Epochs	Batch Size	Act. Func.	Loss Func.	#Layers	Output Function	Ratio
0.1	100	128	Sigmoid	Cross-Entr.	50	Softmax	0.75
0.1	100	128	Sigmoid	Cross-Entr.	200	Softmax	0.55
0.1	100	128	Sigmoid	Cross-Entr.	30	Softmax	0.8
0.05	100	128	Sigmoid	Cross-Entr.	50	Softmax	0.6
0.5	100	128	Sigmoid	Cross-Entr.	50	Softmax	0.8
0.1	300	128	Sigmoid	Cross-Entr.	50	Softmax	0.85
0.1	50	128	Sigmoid	Cross-Entr.	50	Softmax	0.75
0.1	100	228	Sigmoid	Cross-Entr.	50	Softmax	0.65
0.1	100	68	Sigmoid	Cross-Entr.	50	Softmax	0.8

Παρατηρήσεις:

- Αύξηση του αριθμού των Νευρώνων στα Layers:
Παρατηρείται, με την αύξηση των νευρώνων στους 200 ανά layer, η μείωση του learning rate, που οφείλεται πιθανά σε overfitting του μοντέλου και αποτυχία γενίκευσης και αποτελεσματικότητας στα διαφορετικά test data.

- Αύξηση του learning rate:
Παρατηρείται αύξηση του ratio, με την αύξηση του learning rate, γεγονός που ίσως να σημαίνει, ότι, στην εκκίνηση του training του μοντέλου, είναι μεγάλη η τιμή της loss function και οδηγείται γρηγορότερα σε σύγκλιση ή καλύτερη προσαρμογή στο συγκεκριμένο πρόβλημα.

- Αλλαγή αριθμού των epochs:
Όσον αφορά την αύξηση του αριθμού των ecpochs, παρατηρείται βελτίωση του ratio στα 0.8 από 0.75, αλλά και σημαντική αύξηση του χρόνου εκτέλεσης. Ενώ, με την μείωση του αριθμού των epochs, το ratio παραμένει σταθερό στα 0.75.

- Αλλαγή του Batch size:
Με την αύξηση του batch size στα 128, παρατηρείται μείωση του learning rate, γεγονός που ίσως οφείλεται στην μικρότερη ενημέρωση των βαρών, λόγω του μεγαλύτερου μεγέθους του. Ενώ, με την μείωση του batch size στα 68, παρατηρείται αύξηση στο learning rate (0.8).

Χρήση Tanh ως Activation Function – Χρήση MSE ως Loss Function

Στη συνέχεια, ως Activation Function του δεύτερου output Layer, χρησιμοποιήθηκε η συνάρτηση Tanh, ενώ, ως Loss Function, η συνάρτηση MSE.
Αποτελέσματα:

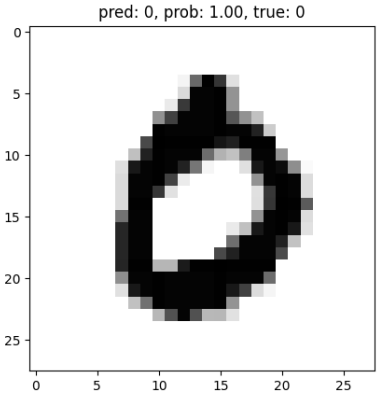
Learning Rate	#Epochs	Batch Size	Act. Func.	Loss Func.	#Layers	Output Function	Ratio
0.1	100	128	Sigmoid	Cross-Entr.	50	Softmax	0.75
0.1	100	128	Sigmoid	MSE	50	Softmax	0.45
0.1	100	128	Sigmoid	MSE	50	Tanh	0.65
0.5	100	128	Sigmoid	MSE	50	Tanh	0.65

Παρατηρήσεις:

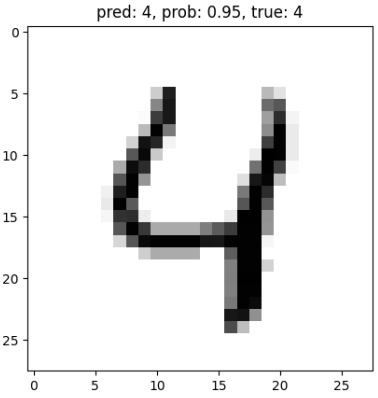
- Χρήση MSE Loss Function:
  Με την χρήση MSE, ως Loss Function, παρατηρείται σημαντική μείωση, στην απόδοση του δικτύου, με το ratio να μειώνεται στα 0.45.
- Χρήση Tanh Activation Function και MSE Loss Function:
  Με την αλλαγή και της Activation Function σε Tanh, παρατηρείται βελτίωση του ratio στα 0.65, όμως, ο συνδυασμός της χρήσης Activation Function Sigmoid και Loss Function Cross-Entropy συνεχίζει να παραμένει καλύτερος.
- Χρήση Tanh Activation Function και MSE Loss Function με Αύξηση του Learning Rate:
  Με την αύξηση του learning rate, παρατηρείται αύξηση του ratio στα 0.75.

Τέλος, ως Activation Function του πρώτου Layer, χρησιμοποιήθηκε η συνάρτηση Tanh, ως Loss Function η συνάρτηση MSE και ως Activation Function του δεύτερου output Layer η Softmax.

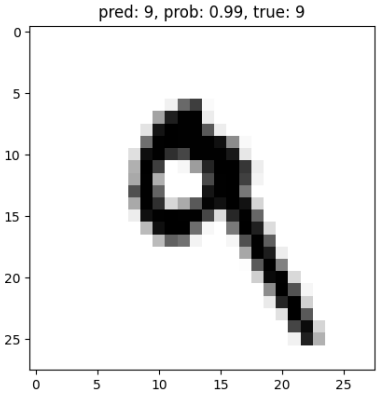
Learning Rate	#Epochs	Batch Size	Act. Func.	Loss Func.	#Layers	Output Function	Ratio
0.1	100	128	Sigmoid	Cross-Entr.	50	Softmax	0.75
0.1	100	128	Tanh	MSE	50	Softmax	0.65



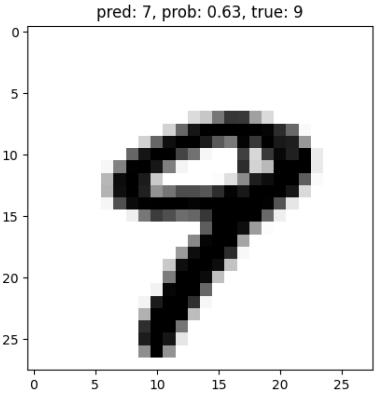
(a) Class 0



(b) Class 4



(c) Class 9



(d) Incorrect Prediction for true class 9

Figure 9: Label (Class) Predictions for four samples