Ερώτημα α- Ποιό είναι το βέλτιστο τεστ κατά Bayes

που ελαχιστοποιεί την πιθανότητα σφάλματος απόφασης

Γνωρίζουμε ότι το βέλτιστο τεστ κατά Bays είναι ο λόγος πιθανοφάνειας:

$$\frac{f_1(X)}{f_0(X)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P(H_0)}{P(H_1)} \Leftrightarrow \frac{f_1(X)}{f_0(X)} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \quad \textbf{(1)} \quad \text{αφού } P(H_0) = P(H_1) = 0.5$$

Επίσης όμως έχουμε ότι $f_0(X) = f_0(x_1)\, f_0(x_2)$ και $f_1(X) = f_1(x_1)\, f_1(x_2)$, αφού τα x1, x2 είναι ανεξάρτητα. Για την περίπτωση της H_0 έχουμε: $f_0(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ **(2)** και για την H_1 αντίστοιχα $f_1(x) = 0.5\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{(x-1)^2}{2}} + \frac{1}{\sqrt{2\pi}} e^{-\frac{(x+1)^2}{2}}\right)$ **(3)**, χρησιμοποιώντας την συνάρτηση πυκνότητας πιθανότητας της κανονικής κατανομής $\mathcal{N}(\mu,\sigma^2)$:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\cdot\left(\frac{x-\mu}{\sigma}\right)^2}$$

Μπορούμε επίσης να λογαριθμίσουμε και τα 2 μέλη της ανίσωσης (1), αφού η συνάρτηση ω(x)=ln(x) είναι γνησίως αύξουσα, οπότε προκύπτει το ισοδύναμο βέλτιστο τεστ:

$$\ln\left(\frac{f_1(x_1)\cdot f_1(x_1)}{f_0(x_1)\cdot f_0(x_1)}\right) \underset{H_0}{\overset{H_1}{\gtrless}} 0$$

Κάνοντας πράξεις λογαρίθμων στο αριστερό μέλος της ανίσωσης έχουμε:

$$\ln\left(\frac{f_1(x_1)\cdot f_1(x_2)}{f_0(x_1)\cdot f_0(x_2)}\right) \Leftrightarrow ln\left(\frac{f_1(x_1)}{f_0(x_1)}\right) + \ln\left(\frac{f_1(x_2)}{f_0(x_2)}\right) \Leftrightarrow \ln(f_1(x_1)) - \ln(f_0(x_1)) + \ln(f_1(x_2)) - \ln(f_0(x_2)) \overset{(2),(3)}{\Longleftrightarrow}$$

$$ln\left(0.5\left(e^{-\frac{(x_1-1)^2}{2}} + e^{-\frac{(x_1+1)^2}{2}}\right)\right) - ln\left(e^{-\frac{x_1^2}{2}}\right) + ln\left(0.5\left(e^{-\frac{(x_2-1)^2}{2}} + e^{-\frac{(x_2+1)^2}{2}}\right)\right) - ln\left(e^{-\frac{x_2^2}{2}}\right) \quad (\mathbf{4})$$

με τους σταθερούς όρους $\frac{1}{\sqrt{2\pi}}$ να έχουν απαλειφθεί από αριθμητή και παρονομαστή, πριν την διάσπαση των λογαρίθμων.

Έτσι, $(4) \Leftrightarrow$ $\boxed{\ln(0.5) + \ln\left(e^{-\frac{(x_1-1)^2}{2}} + e^{-\frac{(x_1+1)^2}{2}}\right) + \frac{x_1^2}{2} + \ln(0.5) + \ln\left(e^{-\frac{(x_2-1)^2}{2}} + e^{-\frac{(x_2+1)^2}{2}}\right) + \frac{x_2^2}{2}}$ **(5)**

Μπορούμε μάλιστα να ορίσουμε την συνάρτηση $h(x) = \ln(0.5) + \ln\left(e^{-\frac{(x-1)^2}{2}} + e^{-\frac{(x+1)^2}{2}}\right) + \frac{x^2}{2}$, οπότε τελικά η

(5) γράφεται ως $h(x_1) + h(x_2)$ και το βέλτιστο τεστ κατά τον κανόνα του Bayes γίνεται:

$$h(x_1) + h(x_2) \underset{H_0}{\overset{H_1}{\gtrless}} 0,$$

σημειογραφία που χρησιμοποιείται και στον κώδικα, ο οποίος παρατίθεται στο τέλος της αναφοράς.

Ερώτημα β - Υπολογισμός σφάλματος με προσωμοιώσεις

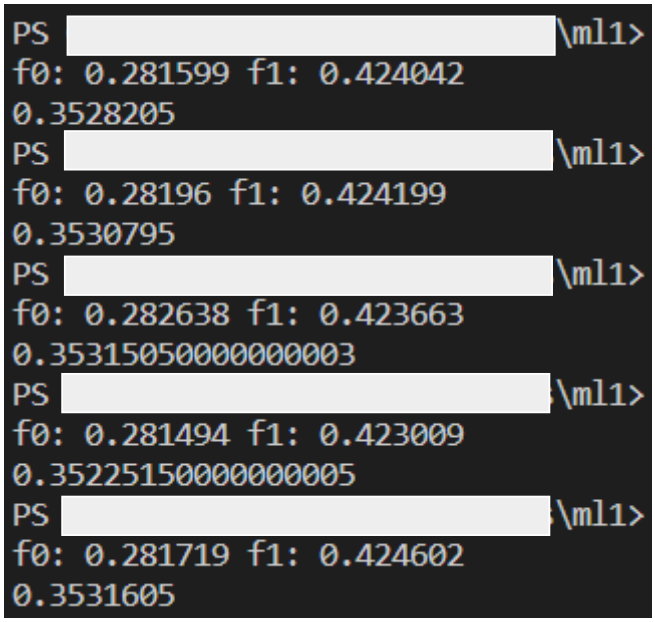
Δοθέντος, λοιπόν ενός διανύσματος X=[x1,x2] αρκεί να υπολογίσουμε το $h(x_1) + h(x_2)$. Αν αυτό είναι θετικό (>0) η μέθοδος αποφαίνεται ότι το δεδομένο προκύπτει από την υπόθεση H_1, αν είναι αρνητικό (<0) η μέθοδος αποφαίνεται ότι το δεδομένο προκύπτει από την υπόθεση H_0, ενώ τέλος αν είναι ίσο με το 0 κατατάσσουμε το X είτε κάτω από την υπόθεση H_0 είτε κάτω από την H_1 με πιθανότητα 0.5.

Για να πραγματοποιήσουμε τα παραπάνω, στο ερώτημα αυτό χρησιμοποιούμε simulations: κατασκευάζουμε 10^6 δεδομένα (ζεύγη [x1,x2]) χρησιμοποιώντας την κατανομή κάτω από την H_0 και άλλα τόσα υπό την υπόθεση H_1. Υπολογίζουμε έπειτα το $h(x_1) + h(x_2)$ όπως αναφέρθηκε στο ερώτημα 1α και μετράμε το πλήθος σφαλμάτων, δηλαδή των περιπτώσεων εκείνων που δεδομένο υπό την H_0 είχε θετικό αποτέλεσμα (f0_errors) και εκείνων που δεδομένο υπό την H_1 είχε αρνητικό αποτέλεσμα (f1_errors). Τέλος, αρκεί να υπολογίσουμε το συνολικό σφάλμα ως

$$\varepsilon = 0.5\left(\frac{f_{0_{errors}}}{K} + \frac{f_{1_{errors}}}{K}\right), \text{όπου } K \text{ το πλήθος των συνολικών δεδομένων}, K = 10^6$$

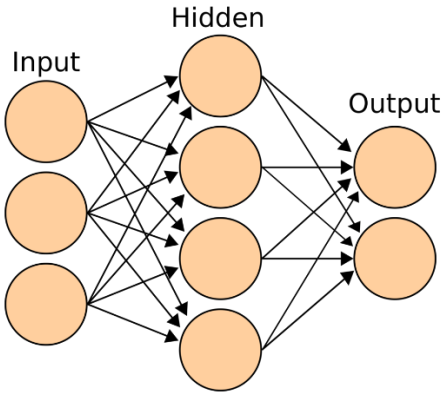
Διευκρινιστικά, η προγραμματιστική υλοποίηση πραγματοποιήθηκε με χρήση της γλώσσας προγραμματισμού python (έκδοση 3.8) για όλα τα ερωτήματα. Χρησιμοποιήθηκε επίσης η βιβλιοθήκη numpy που επιτρέπει την ταχεία εκτέλεση πράξεων πινάκων. Η συγκεκριμένη βιβλιοθήκη περιέχει και στοιχεία πιθανοτήτων, επιτρέποντας την χρήση των κανονικών κατανομών κλπ. Η προαναφερθείσα συνάρτηση h παραδείγματος χάριν, πραγματοποιεί πράξεις με πίνακες κατευθείαν και όχι στοιχείο προς στοιχείο.

Η εκτέλεση του προγράμματος εκτυπώνει στην οθόνη τα σφάλματα f_{0_errors}, f_{1_errors} καθώς και το συνολικό βέλτιστο σφάλμα, το οποίο ανέρχεται περίπου σε 35%. Το πρόγραμμα εκτελέστηκε αρκετές φορές και κάθε μία από αυτές το σφάλμα κατά bays διαφέρει στο 3ο δεκαδικό ψηφίο.



Ερώτημα γ - Classification με νευρωνικό Δίκτυο

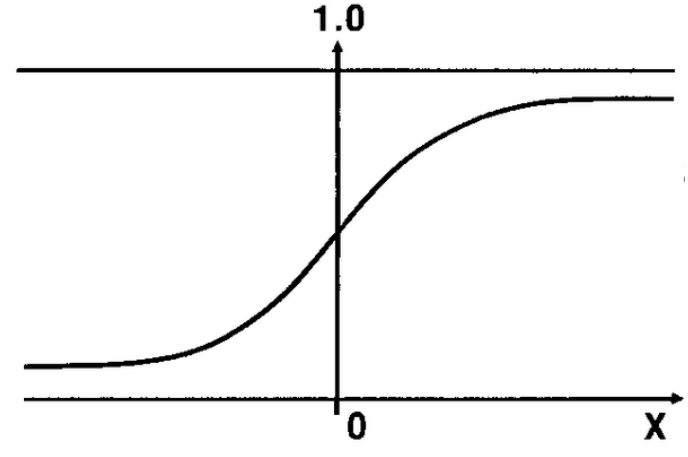
Στο ερώτημα αυτό θα επαναλάβουμε την προηγούμενη άσκηση, χρησιμοποιώντας όμως ένα νευρωνικό δίκτυο μεγέθους 2x20x1, το οποίο σημαίνει ότι το δίκτυο δέχεται δύο εισόδους: x1, x2, έχει ένα κρυφό επίπεδο με 20 νευρώνες και τέλος έχει μία μοναδική έξοδο (που αντιστοιχεί στην κατηγοριοποίηση της εισόδου ως προκύπτουσα υπό την υπόθεση H_0 ή την H_1. Τούτο συμβαίνει καθώς με κατάλληλο νευρωνικό δίκτυο μπορούμε να προσεγγίσουμε οποιαδήποτε συνάρτηση. Το ακόλουθο νευρωνικό δίκτυο είναι ενδεικτικό και δεν ανταποκρίνεται στις πραγματικές διαστάσεις του δικτύου της άσκησης.



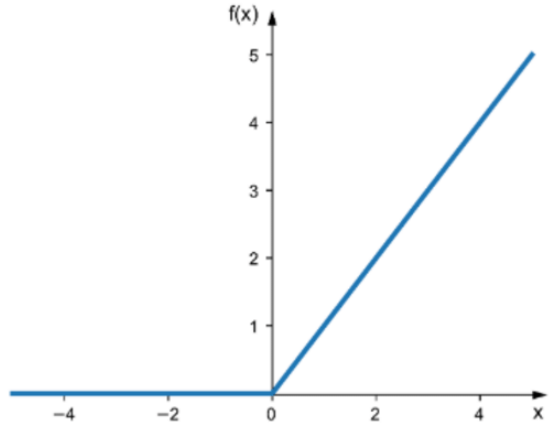
Η διαδικασία feed forward (προς τα εμπρός προώθηση των δεδομένων) είναι αρκετά απλή και συνίσταται στην πράξη πινάκων. Στην υλοποίηση έχει υιοθετηθεί η ακόλουθη σημειογραφία για τους πίνακες (κάθε πίνακας έχει ως εκθέτης την διάστασή του):

$z_1^{20x1} = W_1^{20x2} * x^{2x1} + b_1^{20x1}$	z_1 η έξοδος του πρώτου επιπέδου, W_1 τα βάρη για το 1ο επίπεδο, b_1 τα biases του 1ου επιπέδου.
$\alpha_1^{20x1} = \max(0, z_1^{20x1})$	α_1 η έξοδος μετά την συνάρτηση ενεργοποίησης ReLU
$z_2^{1x1} = W_2^{1x20} * \alpha_1^{20x1} + b_1^{1x1}$	z_2 η έξοδος του 2ου επιπέδου, W_2 τα βάρη για το 2ο επίπεδο, b_2 τα biases του 2ου επιπέδου.

Τέλος, στην περίπτωση της χρήσης της cross entropy μεθόδου ως loss function, χρειάζεται να περάσουμε το z_2 από την σιγμοειδή συνάρτηση ενεργοποίησης, οπότε $\hat{y} = \frac{1}{1+e^{-z_2}}$, ενώ για την εκθετική μέθοδο $\hat{y} = z_2$. Με $\hat{y}$ συμβολίζουμε την «πρόβλεψη» (predicted output) του δικτύου.



Σιγμοειδής συνάρτηση ενεργοποίησης.



Συνάρτηση ενεργοποίησης ReLU.

Μεγαλύτερο ενδιαφέρον παρουσιάζει η εκπαίδευση του νευρωνικού δικτύου (διαδικασία γνωστή ως backpropagation). Σύμφωνα με την διαδικασία αυτή οι παράμετροι του δικτύου (βάρη και biases) αλλάζουν σύμφωνα με τις αντίστοιχες μερικές παραγώγους της συνάρτησης κόστους. Ας εξετάσουμε την περίπτωση της cross entropy μεθόδου, καθώς παρόμοια διαδικασία εκτελείται και στην exponential μέθοδο.

Αν η κατηγορία της εισόδου είναι πραγματικά η H_0, τότε η συνάρτηση κόστους είναι η $\varphi(\hat{y}) = -\ln(1-\hat{y})$, ενώ αντίθετα αν βρίσκεται υπό την H_1 το κόστος είναι $\psi(\hat{y}) = -\ln(\hat{y})$. Ακολουθούν οι μερικές παραγωγίσεις ως προς όλες τις μεταβλητές, οι οποίες χρησιμοποιούνται για την βελτίωση των παραμέτρων θ του δικτύου.

$$Y = 0$$

$$L = -\log(1 - \hat{y}_0)$$

$$\frac{\partial L}{\partial \hat{y}_0} = \frac{1}{1-\hat{y}_0}$$

$$\frac{\partial L}{\partial z_2} = \frac{\partial L}{\partial \hat{y}_0} * \frac{\partial \hat{y}_0}{\partial z_2} = \hat{y}_0$$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial z_2} * \frac{\partial z_2}{\partial W_2} = \hat{y}_0 * \alpha_1$$

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial z_2} * \frac{\partial z_2}{\partial b_2} = \hat{y}_0$$

$$\frac{\partial L}{\partial \alpha_1} = \frac{\partial L}{\partial z_2} * \frac{\partial z_2}{\partial \alpha_1} = \hat{y}_0 * W_2$$

$$\frac{\partial L}{\partial z_1} = \hat{y}_0 * W_2 * (z_1 > 0)$$

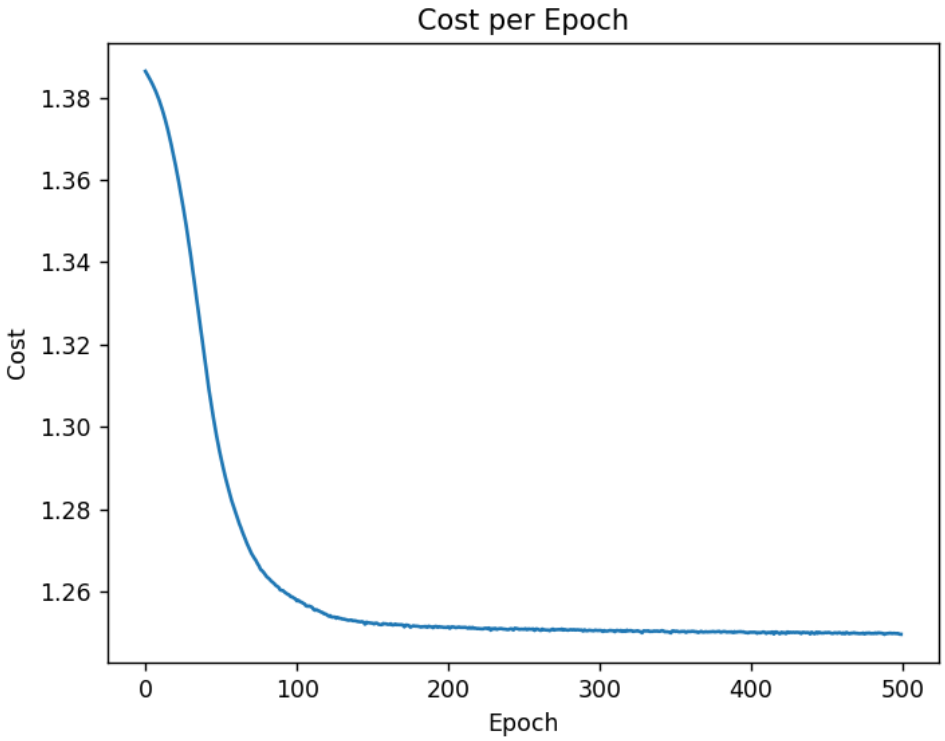
$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial z_1} * \frac{\partial z_1}{\partial b_1} = \hat{y}_0 * W_2 * (z_1 > 0)$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial z_1} * \frac{\partial z_1}{\partial W_1} = \hat{y}_0 * W_2 * (z_1 > 0) * X$$

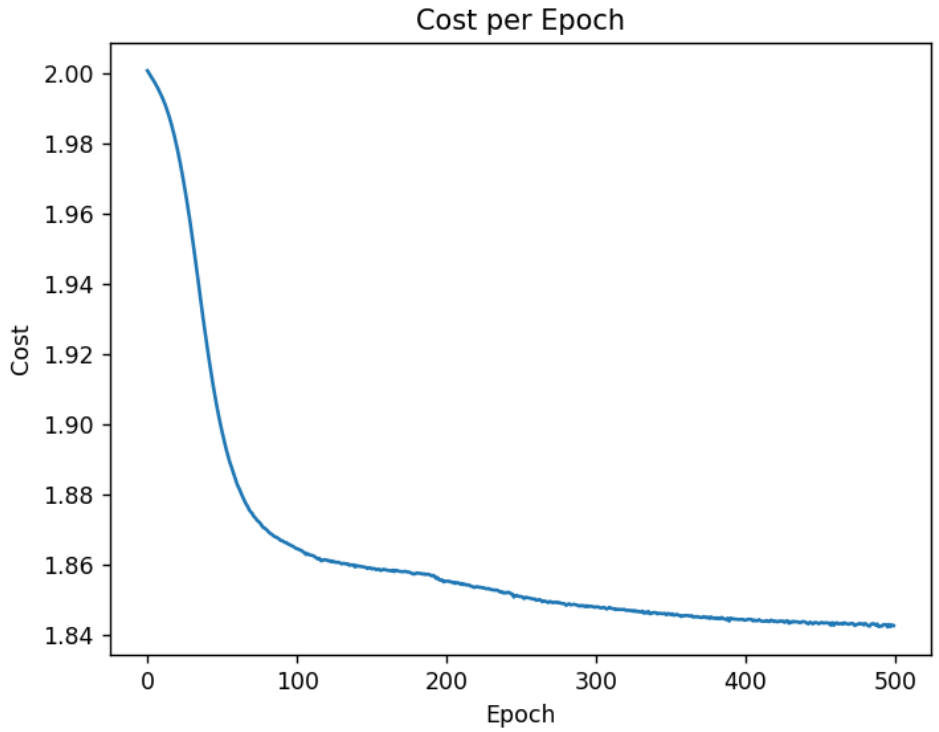
Οι παράμετροι εκπαίδευσης (learning rate=0.003 και epochs=500) επιλέχθηκαν αυθαίρετα μετά από μερικές δοκιμές. Ως μία εποχή ορίζουμε το πέρασμα όλων των 2*200 δεδομένων από το δίκτυο.

Έτσι, η διαδικασία εκπαίδευσης έχει ως ακολούθως: Σε κάθε εποχή ανακατεύουμε αρχικά τα δεδομένα που είναι αποθηκευμένα σε χωριστές δομές. Έπειτα κάνουμε πρώτα feedforward ένα δεδομένο από την H_0 και ένα από την H_1 και υπολογίζουμε το κόστος σε κάθε περίπτωση. Μετά από κάθε feedforward ο αλγόριθμος πραγματοποιεί αυτόματα backpropagation με σκοπό την βελτίωση των παραμέτρων. Το κόστος της επανάληψης είναι το άθροισμα των δύο σφαλμάτων, ενώ στο τέλος της εποχής υπολογίζουμε το μέσο κόστος της εποχής.

Κόστος ανά εποχή για την cross entropy μέθοδο.



Κόστος ανά εποχή για την exponential μέθοδο.



Ακολουθεί μια σύγκριση των σφαλμάτων του Bayes και των 2 νευρωνικών δικτύων:

```
Bays test:
f0: 0.282141 f1: 0.423691
0.352916

cross entropy:
data under H0 misclassified as H1 (e0):  0.358316
data under H1 misclassified as H0 (e1):  0.359674
Total error (e): 0.358995

exponential:
data under H0 misclassified as H1 (e0):  0.13529
data under H1 misclassified as H0 (e1):  0.616679
Total error (e): 0.3759845
```

Την ίδια φιλοσοφία ακριβώς ακολουθούμε και στην περίπτωση αναγνώρισης και διαφοροποίησης δύο χειρόγραφων χαρακτήρων (π.χ. '0' και '8').