Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Τομέας Ηλεκτρονικής και Υπολογιστών

9ο Εξάμηνο
Φθινόπωρο 2019

Classification στην R: kNN και ANN

1 Εισαγωγή

1.1 Εισαγωγή στους αλγορίθμους kNN και ANN

Ο k-Nearest Neighbors (kNN) είναι ένας αλγόριθμος μηχανικής μάθησης για τον οποίο η πρόβλεψη για ένα νέο δείγμα βασίζεται στα k κοντινότερα training δείγματα στο συγκεκριμένο δείγμα. Έτσι, η τιμή κλάσης για ένα νέο δείγμα αποφασίζεται κατά πλειοψηφία (majority vote) με βάση τις τιμές κλάσης των k κοντινότερων δειγμάτων του.

Ο αλγόριθμος perceptron είναι ένα είδος τεχνητού νευρωνικού δικτύου που προσεγγίζει γραμμικά προβλήματα 2 κλάσεων. Δεδομένων των εισόδων x και της εξόδου y, το perceptron εκπαιδεύεται υπολογίζοντας τα βάρη w που είναι τέτοια ώστε η παρακάτω συνάρτηση να είναι θετική για τη μία κλάση (κλάση C1) και αρνητική για την άλλη (κλάση C2):

$$d(x) = w^T x = [w_1 \quad w_2 \quad w_3] \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

Κατά την εκπαίδευση τα βάρη ανανεώνονται επαναληπτικά με βάση το σετ εκπαίδευσης. Ένα νευρωνικό δίκτυο (Artificial Neural Network – ANN) μπορεί να αποτελείται από περισσότερα από ένα perceptrons ενώ υπάρχουν διάφοροι τύποι εκπαίδευσης (π.χ. back propagation).

1.2 kNN στην R

Για τον αλγόριθμο kNN στην R, μπορούμε να χρησιμοποιήσουμε τη βιβλιοθήκη class:

```
> library(class)
```

Για να ταξινομήσουμε μια νέα τιμή τρέχουμε την εντολή knn:

```
> knn(X_train, X_test, Y_train, k = 1, prob = TRUE)
```

όπου δηλώνουμε τα δεδομένα εκπαίδευσης, το νέο δείγμα και την τιμή του k, ενώ προσθέτοντας την παράμετρο prob = TRUE δίνονται επιπλέον οι πιθανότητες για την κλάση.

1.3 ANN στην R

Για νευρωνικά δίκτυα στην R, μπορούμε να χρησιμοποιήσουμε τη βιβλιοθήκη neuralnet:

```
> library(neuralnet)
```

Για να κατασκευάσουμε ένα μοντέλο τρέχουμε την εντολή neuralnet:

```
> model = neuralnet(Y ~ X1 + X2, data, hidden = c(2,2))
```

όπου με το hidden επιλέγουμε τον αριθμό των στρωμάτων (layers) και των νευρώνων (neurons). Μπορούμε ακόμα να επιλέξουμε παραμέτρους όπως τον αλγόριθμο εκμάθησης (algorithm), το μέγιστο επιτρεπτό σφάλμα (threshold), τον μέγιστο αριθμό επαναλήψεων (stepmax), κτλ. Για να δούμε αυτές τις επιλογές εκτελούμε την εντολή `?neuralnet`.

Έχοντας ένα μοντέλο, για να προβλέψουμε μια νέα τιμή τρέχουμε την εντολή

```
> compute(model, test)
```

όπου επιστρέφεται μια τιμή ανάμεσα στις δύο κλάσεις. Π.χ. για κλάσεις -1, 1 μπορούμε να επιλέξουμε την κλάση εκτελώντας την παρακάτω εντολή:

```
> ifelse(compute(model, test)$net.result > 0, 1, -1)
```

Μπορούμε επιπλέον να εμφανίσουμε το δίκτυο με την εντολή:

```
> plot(model)
```

2 Κατασκευή Μοντέλου kNN και Κατάταξη Τιμών

Για την κατασκευή ενός μοντέλου kNN θα χρησιμοποιήσουμε ως εφαρμογή τα δεδομένα εκπαίδευσης του παρακάτω πίνακα για ένα πρόβλημα δυαδικής ταξινόμησης.

X1	**X2**	**Y**
0.7	0.7	A
0.7	0.8	A
0.6	0.6	A
0.5	0.5	A
0.5	0.6	A
0.5	0.7	A
0.5	0.8	A
0.7	0.5	B
0.8	0.7	B
0.8	0.5	B
0.8	0.6	B
1.0	0.3	B
1.0	0.5	B
1.0	0.6	B

Θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Σχεδιάστε τα δεδομένα με διαφορετικά χρώματα/σύμβολα για κάθε κλάση.

β) Χρησιμοποιώντας τον kNN με k = 1, σε ποια κλάση θα κατατάσσατε μία νέα παρατήρηση με τιμές (X1, X2) = (0.7, 0.4);

γ) Επαναλάβετε το ερώτημα (β) χρησιμοποιώντας k = 5.

δ) Σε ποια κλάση θα κατέτασσε ο kNN με k = 5 μία παρατήρηση με τιμές (X1, X2) = (0.7, 0.6);

2.1 Εισαγωγή Δεδομένων και Βιβλιοθηκών

Αρχικά διαβάζουμε τα δεδομένα και φορτώνουμε τις απαραίτητες βιβλιοθήκες:

```
> knndata = read.csv("knndata.txt")
> library(class)
```

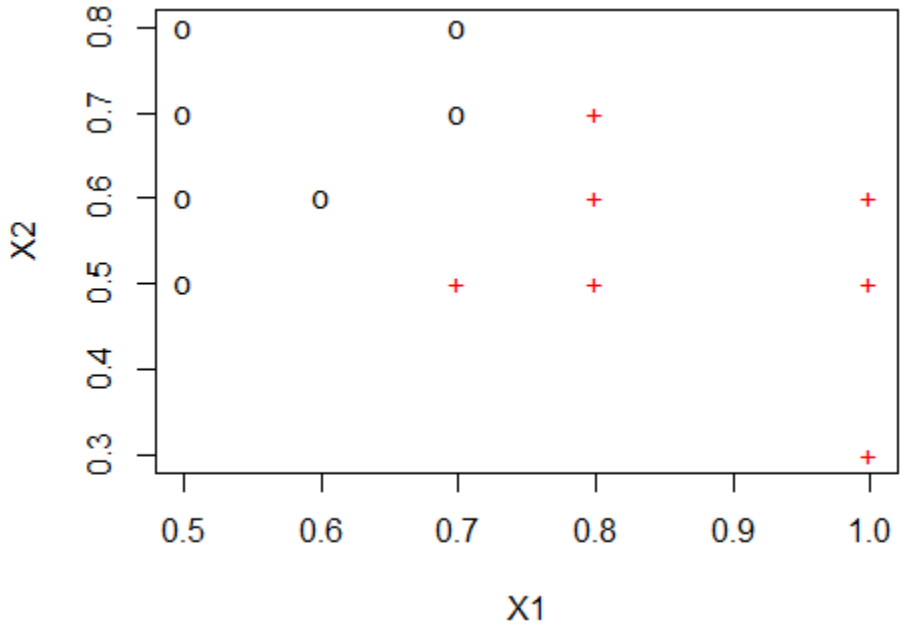
2.2 Εφαρμογή Αλγορίθμου kNN

Χωρίζουμε αρχικά το dataset όπως παρακάτω:

```
> X_train = knndata[,c("X1","X2")]
> Y_train = knndata$Y
```

Για το ερώτημα (α) εκτελούμε :

```
> plot(X_train, col = Y_train, pch = c("o","+")[Y_train])
```



Για k = 1 (ερώτημα (β)), αρκεί να βρούμε την κοντινότερη παρατήρηση (ευκλείδεια απόσταση) στην (0.7, 0.4), η οποία είναι η (0.7, 0.5) που ανήκει στην κλάση B. Οπότε εκτελώντας την εντολή:

```
> knn(X_train, c(0.7, 0.4), Y_train, k = 1, prob = TRUE)
```

προκύπτει ότι η νέα παρατήρηση ανήκει στην κλάση B με πιθανότητα 1.

Για k = 5 (ερώτημα (γ)), θα βρούμε τις 5 πιο κοντινές παρατηρήσεις, που είναι οι (0.7, 0.5), (0.8, 0.5), (0.6, 0.6), (0.8, 0.6), (0.7, 0.7) που ανήκουν στις κλάσεις B, B, A, B, A αντίστοιχα. Άρα αφού τα 3/5 των παρατηρήσεων αυτών ανήκουν στην κλάση B εκτελώντας την εντολή:

```
> knn(X_train, c(0.7, 0.4), Y_train, k = 5, prob = TRUE)
```

προκύπτει ότι η νέα παρατήρηση ανήκει στην κλάση B με πιθανότητα 0.6.

Για το ερώτημα (δ), θα θέλαμε να βρούμε τις 5 πιο κοντινές παρατηρήσεις στο σημείο (0.7, 0.6). Παρατηρούμε όμως ότι οι παρατηρήσεις (0.8, 0.7) και (0.8, 0.5) (που ανήκουν στην κλάση B) ισαπέχουν από το συγκεκριμένο σημείο. Έτσι (και επειδή η παράμετρος use.all του kNN είναι TRUE), ο αλγόριθμος θα χρησιμοποιήσει όλες αυτές τις παρατηρήσεις. Άρα αφού τα 4/6 των παρατηρήσεων ανήκουν στην κλάση B εκτελώντας την εντολή:

```
> knn(X_train, c(0.7, 0.6), Y_train, k = 5, prob = TRUE)
```

προκύπτει ότι η νέα παρατήρηση ανήκει στην κλάση B με πιθανότητα 0.66.

3 Κατασκευή Μοντέλου Perceptron και ANN

Για την κατασκευή ενός μοντέλου ANN θα χρησιμοποιήσουμε ως εφαρμογή τα δεδομένα εκπαίδευσης του παρακάτω πίνακα για ένα πρόβλημα δυαδικής ταξινόμησης.

X1	X2	Y
0	0	1
0	1	1
1	0	-1
1	1	-1

Θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Σχεδιάστε τα δεδομένα με διαφορετικά χρώματα/σύμβολα για κάθε κλάση.

β) Να εφαρμοστεί ο αλγόριθμος Perceptron για την εύρεση γραμμικής συνάρτησης απόφασης που διαχωρίζει τις δύο κλάσεις. Να χρησιμοποιηθεί w(1) = [0, 0, 0] και learning rate c = 1.

γ) Επαναλάβετε το ερώτημα (β) χρησιμοποιώντας την R και εμφανίστε το νευρωνικό δίκτυο που προκύπτει.

3.1 Κατασκευή Δεδομένων και Εισαγωγή Βιβλιοθηκών

Αρχικά κατασκευάζουμε τα δεδομένα και φορτώνουμε τις απαραίτητες βιβλιοθήκες:

```
> anndata = data.frame(X1 = c(0,0,1,1), X2 = c(0,1,0,1), Y = c(1,1,-1,-1))
> library(neuralnet)
```

3.2 Εφαρμογή Αλγορίθμου Perceptron

Για το ερώτημα (α) εκτελούμε:

```
> Y12 = ifelse(anndata$Y > 0, 1, 2)
> plot(anndata[,c("X1","X2")], col = Y12, pch = c("o","+")[Y12])
```

Για τα ερωτήματα (β), (γ), αρκεί να βρούμε τα βάρη του perceptron ώστε η παρακάτω συνάρτηση να είναι θετική για Y = 1 και αρνητική για Y = -1:

$$d(x) = w^T x = [w_1 \quad w_2 \quad w_3] \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

Αρχικά, τα βάρη είναι w1 = 0, w2 = 0, w3 = 0. Εκτελούμε επαναλήψεις και σε κάθε επανάληψη προσαρμόζουμε τα βάρη ανάλογα με το αν το σημείο θα έπρεπε να ανήκει στην κλάση 1 ή στην κλάση -1. Αν ένα σημείο x(i) ανήκει στην κλάση 1 ενώ θα έπρεπε να ανήκει στην -1, τότε προσθέτουμε στα βάρη τον πίνακα c·x(i). Αν ανήκει στην κλάση -1 ενώ θα έπρεπε να ανήκει στην 1, τότε αφαιρούμε από τα βάρη τον πίνακα c·x(i).

1η επανάληψη

1. $w^T(1)x(1) = [0 \quad 0 \quad 0] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$, άρα $w(2) = w(1) + x(1) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

2. $w^T(2)x(2) = [0 \quad 0 \quad 1] \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1 > 0$, άρα $w(3) = w(2) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

3. $w^T(3)x(3) = [0 \quad 0 \quad 1] \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 1 > 0$, άρα $w(4) = w(3) - x(3) = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$

4. $w^T(4)x(4) = [-1 \quad 0 \quad 0] \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = -1 < 0$, άρα $w(5) = w(4) = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$

2η επανάληψη

5. $w^T(5)x(1) = [-1 \quad 0 \quad 0] \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$, άρα $w(6) = w(5) + x(1) = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

6. $w^T(6)x(2) = [-1 \quad 0 \quad 1] \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = 1 > 0$, άρα $w(7) = w(6) = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

7. $w^T(7)x(3) = [-1 \quad 0 \quad 1]\begin{bmatrix}1\\0\\1\end{bmatrix} = 0$, άρα $w(8) = w(7) - x(3) = \begin{bmatrix}-2\\0\\0\end{bmatrix}$

8. $w^T(8)x(4) = [-2 \quad 0 \quad 0]\begin{bmatrix}1\\1\\1\end{bmatrix} = -2 < 0$, άρα $w(9) = w(8) = \begin{bmatrix}-2\\0\\0\end{bmatrix}$

3η επανάληψη

9. $w^T(9)x(1) = [-2 \quad 0 \quad 0]\begin{bmatrix}0\\0\\1\end{bmatrix} = 0$, άρα $w(10) = w(9) + x(1) = \begin{bmatrix}-2\\0\\1\end{bmatrix}$

10. $w^T(10)x(2) = [-2 \quad 0 \quad 1]\begin{bmatrix}0\\1\\1\end{bmatrix} = 1 > 0$, άρα $w(11) = w(10) = \begin{bmatrix}-2\\0\\1\end{bmatrix}$

11. $w^T(11)x(3) = [-2 \quad 0 \quad 1]\begin{bmatrix}1\\0\\1\end{bmatrix} = -1 < 0$, άρα $w(12) = w(11) = \begin{bmatrix}-2\\0\\1\end{bmatrix}$

12. $w^T(12)x(4) = [-2 \quad 0 \quad 1]\begin{bmatrix}1\\1\\1\end{bmatrix} = -1 < 0$, άρα $w(13) = w(12) = \begin{bmatrix}-2\\0\\1\end{bmatrix}$

4η επανάληψη

13. $w^T(13)x(1) = [-2 \quad 0 \quad 1]\begin{bmatrix}0\\0\\1\end{bmatrix} = 1 > 0$, άρα $w(14) = w(13) = \begin{bmatrix}-2\\0\\1\end{bmatrix}$

14. $w^T(14)x(2) = [-2 \quad 0 \quad 1]\begin{bmatrix}0\\1\\1\end{bmatrix} = 1 > 0$, άρα $w(15) = w(14) = \begin{bmatrix}-2\\0\\1\end{bmatrix}$

15. $w^T(15)x(3) = [-2 \quad 0 \quad 1]\begin{bmatrix}1\\0\\1\end{bmatrix} = -1 < 0$, άρα $w(16) = w(15) = \begin{bmatrix}-2\\0\\1\end{bmatrix}$

16. $w^T(16)x(4) = [-2 \quad 0 \quad 1]\begin{bmatrix}1\\1\\1\end{bmatrix} = -1 < 0$, άρα $w(17) = w(16) = \begin{bmatrix}-2\\0\\1\end{bmatrix}$

Ο αλγόριθμος έχει συγκλίνει διότι πλέον δεν συμβαίνει καμία διόρθωση στα βάρη. Η συνάρτηση απόφασης που προκύπτει είναι:

$d(x) = w^T x = [-2 \quad 0 \quad 1]\begin{bmatrix}x_1\\x_2\\1\end{bmatrix} = -2x_1 + 1$ Ή αλλιώς η ευθεία $x_1 = \frac{1}{2}$.

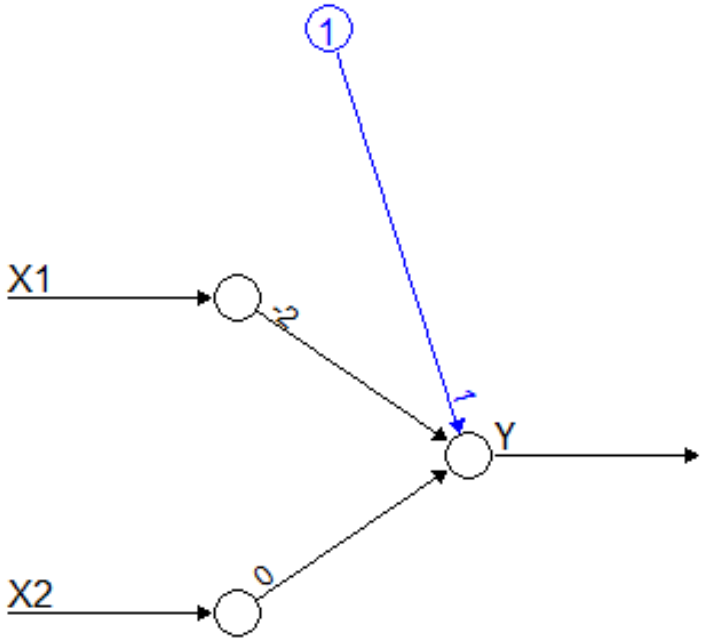
Επαναλαμβάνουμε τη λύση με την R (ερώτημα (γ)) εκτελώντας την παρακάτω εντολή:

```
> model = neuralnet(Y ~ X1 + X2, anndata, hidden = 0, threshold = 0.000001)
```

Μπορούμε επιπλέον να εμφανίσουμε το δίκτυο που προκύπτει και να εκτυπώσουμε τα βάρη εκτελώντας τις παρακάτω εντολές:

```
> plot(model)
```

```
> print(model$weights)
```



4 Κατασκευή Μοντέλου ANN

Για την εκπαίδευση ενός νευρωνικού δικτύου (ANN) θα χρησιμοποιήσουμε ως εφαρμογή τα δεδομένα εκπαίδευσης του αρχείου που δίνεται (alldata.txt) για ένα πρόβλημα δυαδικής ταξινόμησης.

Ένα summary των δεδομένων με την R είναι το παρακάτω:

```
       X1                    X2                    y
 Min.   :-3.25322007   Min.   :-4.664161   Min.   :1.0
 1st Qu.:-0.70900886   1st Qu.: 0.625361   1st Qu.:1.0
 Median :-0.01162501   Median : 1.641370   Median :1.5
 Mean   : 0.03493570   Mean   : 1.939284   Mean   :1.5
 3rd Qu.: 0.73337179   3rd Qu.: 3.115350   3rd Qu.:2.0
 Max.   : 3.63957363   Max.   : 9.784076   Max.   :2.0
```

Αρχικά, εισάγουμε τα δεδομένα και τα διαχωρίζουμε σε training set και test set:

```
alldata = read.csv("alldata.txt")

trainingdata = alldata[1:600, ]

testdata = alldata[601:800, ]
```

Στη συνέχεια, θα απαντήσουμε στα παρακάτω ερωτήματα:

α) Σχεδιάστε τα training data με διαφορετικά χρώματα/σύμβολα για κάθε κλάση.

β) Κατασκευάστε ένα νευρωνικό δίκτυο με 2 hidden layers και 2 hidden nodes ανά layer και συνθήκη τερματισμού 0,01 (threshold = 0.01). Οπτικοποιήστε το νευρωνικό σας δίκτυο που κατασκευάσατε.

γ) Υπολογίστε το σφάλμα εκπαίδευσης (training error) και το σφάλμα ελέγχου (testing error) στο training set και στο test set αντίστοιχα.

δ) Επαναλάβετε τα ερωτήματα (β) και (γ) για 3 hidden layers και 2 hidden nodes ανά layer και συνθήκη τερματισμού 0,01 (threshold = 0.01). Τι παρατηρείτε;

4.1 Εισαγωγή Βιβλιοθηκών

Αρχικά φορτώνουμε τις απαραίτητες βιβλιοθήκες:

```
library(neuralnet)
```

4.2 Κατασκευή ANN

Για το ερώτημα (α) εκτελούμε:

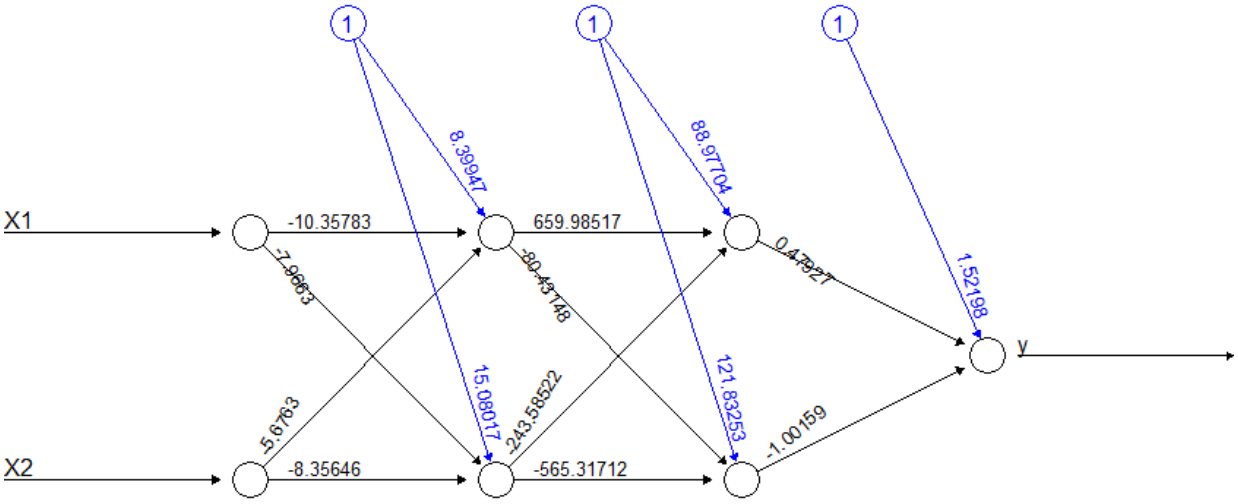
```
> plot(trainingdata[, c(1:2)], col = trainingdata$y,
                               pch = c("o","+")[trainingdata$y])
```

Για το ερώτημα (β) εκτελούμε:

```
> model <- neuralnet(y ~ X1 + X2, data = trainingdata,
                     hidden = c(2, 2), threshold = 0.01)
```

Στη συνέχεια και αφότου έχουμε εκπαιδεύσει το μοντέλο μας μπορούμε να το οπτικοποιήσουμε με την εντολή:

```
> plot(model)
```



Error: 14.642159 Steps: 31864

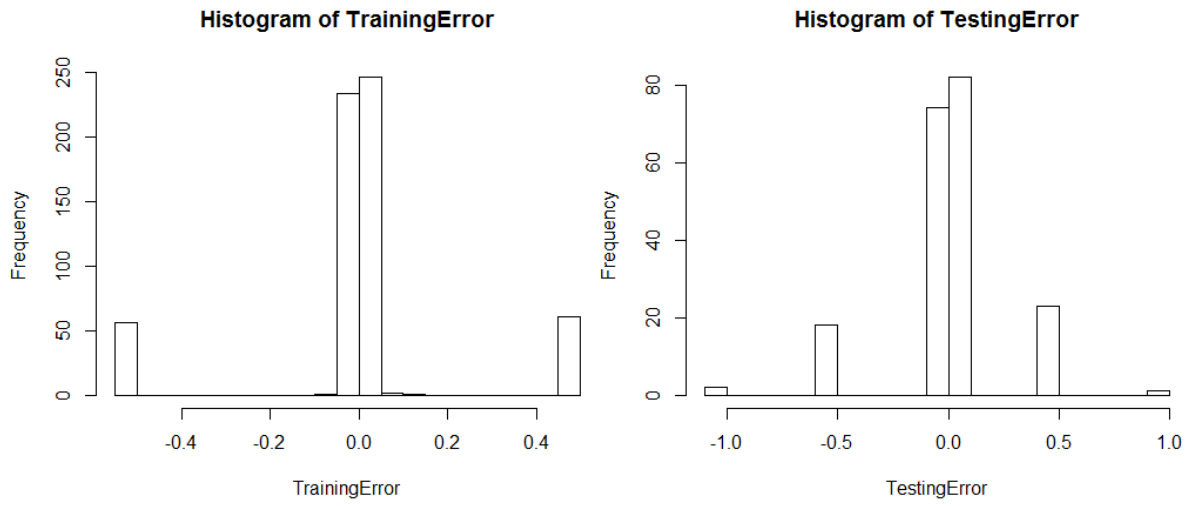
Για το ερώτημα (γ) έχουμε:

```
## Training Error

> yEstimateTrain = compute(model, trainingdata[, c(1:2)])$net.result
> TrainingError = trainingdata$y - yEstimateTrain
> MAE = mean(abs(trainingdata$y - yEstimateTrain))
> plot(hist(TrainingError, breaks = 20))

## Testing Error

> yEstimateTest = compute(model, testdata[, c(1:2)])$net.result
> TestingError = testdata$y - yEstimateTest
> MAE = mean(abs(testdata$y - yEstimateTest))
> plot(hist(TestingError, breaks = 20))
```



5 Πρόβλημα για Εξάσκηση

Δίνονται τα παρακάτω δεδομένα όπου Y είναι το διάνυσμα κλάσης:

X1	X2	Y
2	2	1
2	-2	1
-2	-2	1
-2	2	1
1	1	2
1	-1	2
-1	-1	2
-1	1	2

Μπορείτε να κατασκευάσετε τα δεδομένα στην R με τις παρακάτω εντολές:

```
> X1 = c(2, 2, -2, -2, 1, 1, -1, -1)
> X2 = c(2, -2, -2, 2, 1, -1, -1, 1)
> Y = c(1, 1, 1, 1, 2, 2, 2, 2)
> alldata = data.frame(X1, X2, Y)
```

Απαντήστε στα παρακάτω ερωτήματα:

α) Σχεδιάστε τα δεδομένα με διαφορετικά χρώματα/σύμβολα για κάθε κλάση.

β) Χρησιμοποιώντας τον kNN με k = 3, σε ποια κλάση θα κατατάσσατε μία νέα παρατήρηση με τιμές (X1, X2) = (-2, 0.5);

γ) Κατασκευάστε ένα νευρωνικό δίκτυο με 1 hidden layer και 2 hidden nodes ανά layer και συνθήκη τερματισμού 0,01 (threshold = 0.01). Υπολογίστε το σφάλμα εκπαίδευσης (training error).