



#### Οδηγίες:

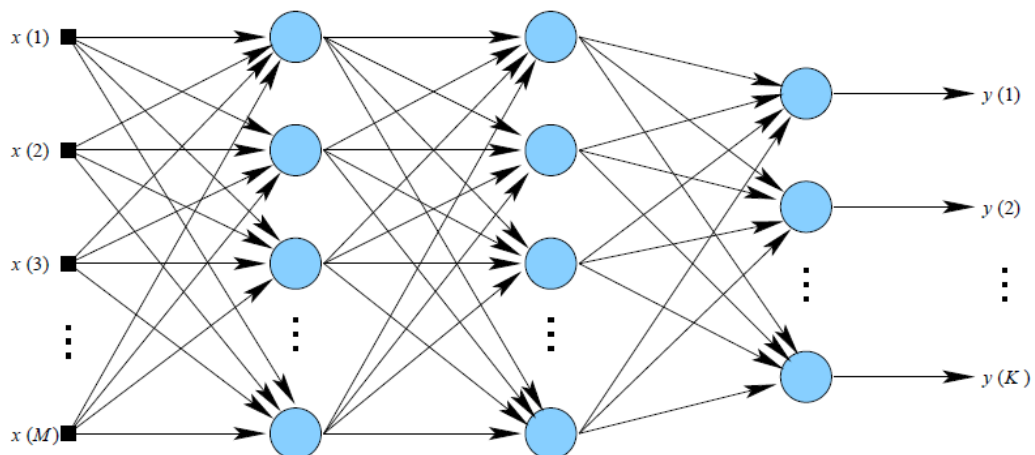
1. Σας παρακαλώ να σεβαστείτε τον παρακάτω κώδικα τιμής τον οποίον θα θεωρηθεί ότι προσυπογράφετε μαζί με τη συμμετοχή σας στο μάθημα και τις εργασίες του:
  - a) Οι απαντήσεις στις εργασίες, τα quiz και τις εξετάσεις, ο κώδικας και γενικά οτιδήποτε αφορά τις εργασίες θα είναι προϊόν δικής μου δουλειάς.
  - b) Δεν θα διαθέσω κώδικα, απαντήσεις και εργασίες μου σε κανέναν άλλο.
  - c) Δεν θα εμπλακώ σε άλλες ενέργειες με τις οποίες ανέντιμα θα βελτιώνω τα αποτελέσματα μου ή ανέντιμα θα αλλάζω τα αποτελέσματα άλλων.
2. Η εργασία είναι ατομική
3. Ημερομηνία παράδοσης: **Τετάρτη, 15/5/2019 στις 20:00**
4. **Παραδοτέα:** α) Κώδικας και β) Αναφορά με τις απαντήσεις, παρατηρήσεις, πειράματα, αποτελέσματα και οδηγίες χρήσης του κώδικα.

#### Θέμα 1: Αρχιτεκτονική Νευρωνικών δικτύων

Εξετάστε την παραδοσιακή αρχιτεκτονική ρηχών νευρωνικών δικτύων. Συγκεκριμένα, έστω το νευρωνικό δίκτυο του παρακάτω σχήματος. Ας υποθέσουμε ότι η είσοδος του είναι RGB εικόνες σημάτων κυκλοφορίας.

Η δομή του δικτύου είναι η εξής:

- Είσοδος:  $3 \times 64 \times 64 = 12288$  – διάστατη
- Στο πρώτο επίπεδο υπάρχουν 100 κόμβοι (εμφανίζονται με γαλάζιο χρώμα στο σχήμα)
- Στο δεύτερο επίπεδο υπάρχουν 100 κόμβοι (γαλάζιο χρώμα)
- Στο τρίτο επίπεδο υπάρχουν 10 κόμβοι (γαλάζιο χρώμα, ένας για κάθε κλάση)



Υπολογίστε τον συνολικό αριθμό των παραμέτρων (βάρη) στο παραπάνω δίκτυο. Αιτιολογήστε την απάντησή σας.

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

Υποθέτουμε ότι έχουμε ένα σύνολο  $m$  δεδομένων  $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(m)}, y^{(m)})\}$ , όπου  $\mathbf{x}^{(i)} \in \mathbb{R}^{n \times 1}$  είναι τα διανύσματα χαρακτηριστικών και  $y^{(i)} \in \{0, 1\}$  ορίζουν την κλάση κάθε δείγματος (labels). Θέλουμε να προβλέψουμε τις τιμές των  $y^{(i)}$  από τις αντίστοιχες τιμές  $\mathbf{x}^{(i)}$ ,  $i \in \{1, 2, \dots, m\}$ , χρησιμοποιώντας την συνάρτηση της λογιστικής παλινδρόμησης, η οποία ορίζεται ως εξής:

$$h_{\theta}(\mathbf{x}) = f(\theta^T \mathbf{x})$$

όπου  $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$  είναι οι παράμετροι του γραμμικού μοντέλου και  $f(\cdot)$  είναι η λογιστική συνάρτηση που ορίζεται ως:

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

Έστω  $\hat{y}^{(i)} = h_{\theta}(\mathbf{x}^{(i)})$  η εκτίμηση της λογιστικής συνάρτησης για το  $y^{(i)}$ . Σύμφωνα με τη θεωρία, στην περίπτωση της λογιστικής παλινδρόμησης, μπορούμε να υπολογίσουμε το σφάλμα με βάση τη συνάρτηση κόστους/σφάλματος (loss function), που ονομάζεται cross-entropy, και ορίζεται ως εξής:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (-y^{(i)} \ln(\hat{y}^{(i)}) - (1 - y^{(i)}) \ln(1 - \hat{y}^{(i)}))$$

Αντικαθιστώντας το  $\hat{y}_i$  έχουμε

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (-y^{(i)} \ln(h_{\theta}(\mathbf{x}^{(i)})) - (1 - y^{(i)}) \ln(1 - h_{\theta}(\mathbf{x}^{(i)})))$$

Για τη βελτιστοποίηση του σφάλματος χρειάζεται να υπολογίσουμε την κλίση (gradient) του σφάλματος  $J(\theta)$  η οποία θα είναι ένα διάνυσμα ίσης διάστασης με το  $\theta$ .

a) Αν  $\theta_j$  και  $x_j^{(i)}$  είναι η  $j$  συνιστώσα των διανυσμάτων  $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$  και

$\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}]^T$  αντίστοιχα, να δείξετε ότι το  $j$ -στοιχείο της κλίσης του σφάλματος είναι:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}$$

b) Θα χρησιμοποιήσουμε την λογιστική παλινδρόμηση για να προβλέψουμε αν ένας φοιτητής θα γίνει δεκτός σε ένα πανεπιστήμιο με βάση τους βαθμούς του σε δύο εξετάσεις. Στο αρχείο «exam\_scores\_data1.txt» υπάρχουν δεδομένα από παλαιότερες αιτήσεις φοιτητών στη μορφή «Exam1Score, Exam2Score, [0: απόρριψη, 1: αποδοχή]», που θα χρησιμοποιηθούν ως δεδομένα εκμάθησης της λογιστικής παλινδρόμησης. Για να διαβάσετε τα δεδομένα τρέξτε την εντολή:

```
data = load('exam_scores_data1.txt');
```

Θα χρειαστεί να συμπληρώσετε κώδικα ώστε να τρέξει το κύριο αρχείο της άσκησης: **My\_logisticRegression.m**

i) Αρχικά δείτε τα δεδομένα με την συνάρτηση plotData.m

- ii) Υλοποιήστε την σιγμοειδή συνάρτηση  $f(z)$  στο αρχείο `sigmoid.m`. Αν η είσοδος  $z$  είναι πίνακας, η `sigmoid` θα πρέπει να εφαρμόζει την  $f(z)$  σε κάθε στοιχείο του πίνακα.
- iii) Υλοποιήστε την συνάρτηση `costFunction` στο αρχείο `costFunction.m` έτσι ώστε να επιστρέφει το κόστος  $J(\theta)$  ( $J$ ) και την κλίση  $\nabla J(\theta)$  (`grad`) όπως ορίζονται στο ερώτημα (α) (**Σημείωση:** Χρησιμοποιώντας ως αρχικές τιμές του  $\theta = 0$ , το κόστος θα πρέπει να βγαίνει περίπου  $J=0.693$  και η κλίση περίπου  $[-0.1, -12.009217, -11.262842]$ ).
- iv) Η βελτιστοποίηση των παραμέτρων γίνεται με κώδικα που υπάρχει έτοιμος στην άσκηση `My_logisticRegression.m`. Εσείς απλά τρέξετε τον κώδικα και βρείτε το σύνολο απόφασης με την συνάρτηση `plotDecisionBoundary.m`. Επίσης τρέξετε την συνάρτηση `predict.m` για να προβλέψετε αν ο φοιτητής θα γίνει δεκτός με βάση διάφορες τιμές βαθμών στις δύο εξετάσεις.

### Θέμα 3: Νευρώνες Sigmoid και SoftMax: Βελτιστοποίηση κατά προσέγγιση χρησιμοποιώντας SGD

Σ' αυτή την άσκηση ο στόχος είναι και πάλι να χρησιμοποιήσουμε το ίδιο dataset με το προηγούμενο θέμα και να δημιουργήσουμε ένα σύστημα με το οποίο θα μπορούμε να προβλέπουμε αν ένας φοιτητής θα γίνει δεκτός σε ένα πανεπιστήμιο με βάση τους βαθμούς του σε δύο εξετάσεις. Η διαφορά αυτή τη φορά είναι ότι θα χρησιμοποιήσουμε προσεγγιστική μέθοδο βελτιστοποίησης του συστήματος που θα δημιουργήσουμε χρησιμοποιώντας επαναληπτικά **Stochastic Gradient Descent (SGD)** αλγόριθμο. Πιο συγκεκριμένα, χρησιμοποιώντας Python και Tensorflow θα δημιουργήσουμε δύο διαφορετικές εκδοχές του συστήματος.

- A. Στην πρώτη εκδοχή θα θεωρήσουμε ότι έχουμε έναν νευρώνα με δύο εισόδους και μία έξοδο. Η συνάρτηση ενεργοποίησης θα είναι όπως και πριν η **λογιστική συνάρτηση** (Σιγμοειδής). Ο στόχος είναι να χρησιμοποιήσουμε τα δεδομένα εκπαίδευσης για να εκτιμήσουμε τις τιμές των παραμέτρων του νευρώνα ( $W \in \mathbb{R}^{2 \times 1}$  και  $b \in \mathbb{R}$ ), έτσι ώστε να μπορούμε να κάνουμε δυαδική πρόβλεψη [**0: απόρριψη, 1: αποδοχή**], με βάση έναν οποιονδήποτε συνδυασμό επίδοσης σε δύο εξετάσεις για έναν φοιτητή (**Exam1Score, Exam2Score**).
  - a. Υπολογίστε την πρόβλεψη του συστήματος για το δείγμα ελέγχου [45, 85]. Ποια είναι η πρόβλεψη και με ποια πιθανότητα; Σχολιάστε το αποτέλεσμα
  - b. Στα πλαίσια της άσκησης μετρήστε επίσης και την ακρίβεια πρόβλεψης πάνω στα δεδομένα εκπαίδευσης. Τρέξτε πολλαπλές φορές το πείραμα και αναφέρετε πόσο πολύ αλλάζει το αποτέλεσμα. Που μπορεί να οφείλεται αυτό;
  - c. Τυπικά σε ένα τέτοιο σύστημα το κατώφλι απόφασης ορίζεται να είναι 0.5. Δοκιμάστε όμως και άλλα κατώφλια στο διάστημα [0, 1] και καταγράψτε τις παρατηρήσεις σας. Για ποιο κατώφλι προκύπτει η μεγαλύτερη ακρίβεια στα δεδομένα εκπαίδευσης;
  - d. Για την εκπαίδευση δοκιμάστε batches των δειγμάτων μεγέθους 10, 20, 50 και διαφορετικές τιμές learning rate. Καταγράψτε τις παρατηρήσεις σας.
- B. Στην δεύτερη εκδοχή η λογιστική συνάρτηση ενεργοποίησης αντικαθίσταται από τη συνάρτηση **SoftMax** και συνεπώς και οι έξοδοι του συστήματος γίνονται 2. Συγκρίνετε τα αποτελέσματα που βγάξετε με αυτά του συστήματος με τη λογιστική συνάρτηση. Πιο από τα δύο θεωρείτε καλύτερο για το συγκεκριμένο πρόβλημα;

Για την υλοποίηση της άσκησης αυτής σας δίνεται ο απαραίτητος κώδικας:

- αρχείο **mylogLR.py** για το σύστημα με τη λογιστική συνάρτηση ενεργοποίησης
- αρχείο **mySoftMax.py** για το σύστημα με την SoftMax ως συνάρτηση ενεργοποίησης

Χρησιμοποιήστε τον κώδικα αυτό σαν βάση και συμπληρώστε τον κατάλληλα ώστε να μπορεί να κάνει εκπαίδευση και έλεγχο επίδοσης για τα δύο παραπάνω συστήματα. Σημειώστε τις ακόλουθες συναρτήσεις του tensorflow τις οποίες πιθανόν να χρειαστεί να χρησιμοποιήσετε κατάλληλα για να συμπληρώσετε τον κώδικα σας:

```
tf.nn.sparse_softmax_cross_entropy_with_logits,  
tf.train.GradientDescentOptimizer, tf.nn.softmax, tf.matmul,  
tf.nn.sigmoid_cross_entropy_with_logits, tf.nn.sigmoid, tf.squeeze,  
tf.cast, tf.reduce_mean, tf.equal, tf.trainable_variables, tf.argmax
```

#### Θέμα 4: Feature Selection – Classification – Cross Validation – Overfitting

Σε επιστημονικό περιοδικό που δημοσιεύει πρωτότυπες εργασίες στον τομέα της βιοϊατρικής πληροφορικής, στάλθηκε ένα άρθρο για αυτόματη ανίχνευση του αυτισμού από δεδομένα εγκεφαλογραφήματος (EEG). Οι ερευνητές χρησιμοποίησαν EEG δεδομένα από ένα σύνολο 25 νεαρών ατόμων, 15 από τους οποίους είχαν διαγνωστεί με αυτισμό και 10 ήταν φυσιολογικοί (χωρίς την διάγνωση κάποιας πάθησης). Από τα εγκεφαλογραφήματα οι ερευνητές υπολόγισαν έναν μεγάλο αριθμό χαρακτηριστικών (περίπου 1000).

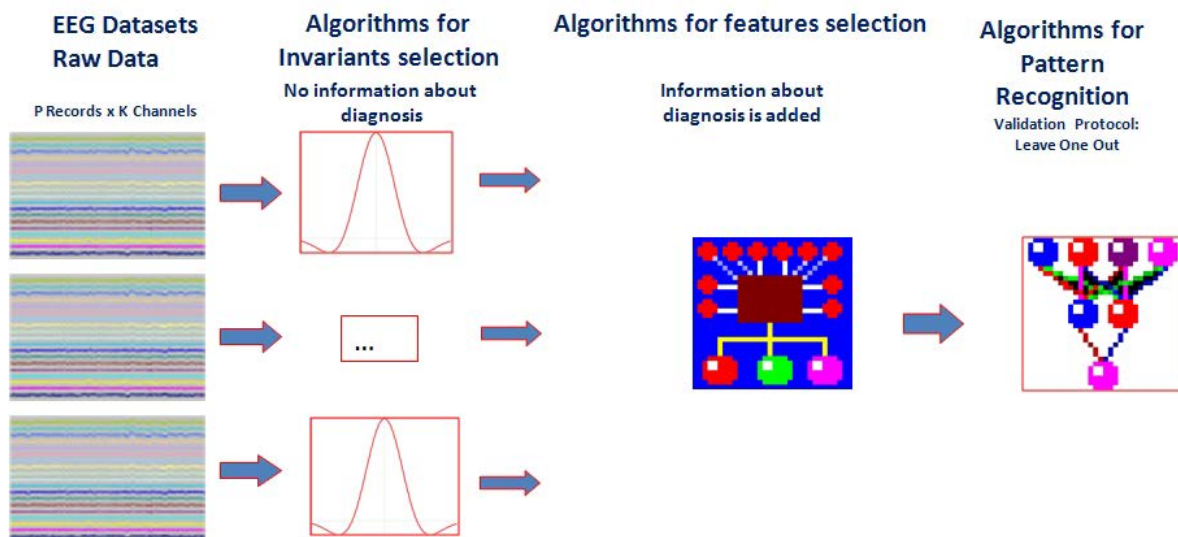
Λόγω του μικρού αριθμού των δειγμάτων χρησιμοποίησαν cross validation και πιο συγκεκριμένα την τεχνική leave-one-out. Σύμφωνα με αυτή την τεχνική αν έχουμε N δείγματα, τα χωρίζουμε σε δύο ομάδες, μία με N-1 δείγματα και μία με το ένα δείγμα που απομένει. Χρησιμοποιούμε την ομάδα με τα N-1 δείγματα για να εκπαιδεύσουμε το μοντέλο και δοκιμάζουμε το μοντέλο στο ένα δείγμα της άλλης ομάδας. Επαναλαμβάνουμε αυτή την διαδικασία N φορές και κάθε φορά αφήνουμε εκτός εκπαίδευσης ένα διαφορετικό δείγμα. Στο τέλος παίρνουμε την μέση τιμή των N αποτελεσμάτων.

Αρχικά χρησιμοποίησαν την παραπάνω διαδικασία για να μετρήσουν την επίδοση του συστήματος όμως λόγω του μεγάλου αριθμού των χαρακτηριστικών σε σχέση με τον όγκο των δεδομένων οι ερευνητές αποφάσισαν να εφαρμόσουν μια διαδικασία επιλογής χαρακτηριστικών. Μια απλή μέθοδος είναι να διαλέξουμε τα χαρακτηριστικά που σχετίζονται καλύτερα με τα labels. Συγκεκριμένα, αποφάσισαν να επιλέξουν τα 100 χαρακτηριστικά που διαφοροποιούν καλύτερα τους ασθενείς από τους φυσιολογικούς. Για να γίνει αυτό χρησιμοποιήθηκε μια μετρική συσχέτισης κάθε χαρακτηριστικού  $X$  με τιμές  $x$  με την κλάση  $C$  (labels). Τα  $X, C$  σ' αυτή την περίπτωση θεωρούνται τυχαίες μεταβλητές για τις οποίες μπορούμε να ορίσουμε τον συντελεστή συσχέτισης Pearson (Pearson's correlation coefficient) ως εξής:

$$\rho_{XC} = \frac{cov(X, C)}{\sigma_X \sigma_C} = \frac{\sum_{i=1}^m (x_i - \bar{x})(c_i - \bar{c})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2 \sum_{i=1}^m (c_i - \bar{c})^2}}$$

όπου  $\bar{x}$  και  $\bar{c}$  είναι οι δειγματικοί μέσοι των δύο τυχαίων μεταβλητών.

Οι συγγραφείς του άρθρου οργάνωσαν την έρευνά τους σύμφωνα με το παρακάτω διάγραμμα. Δηλαδή πρώτα εφαρμόστηκε η επιλογή χαρακτηριστικών πάνω στο σύνολο των δειγμάτων και ακολούθως έγινε εκπαίδευση και εκτίμηση της επίδοσης στα πλαίσια της διαδικασίας cross-validation.



Ένας από τους κριτές όμως διαμαρτυρήθηκε λέγοντας ότι το feature selection πρέπει να γίνει στα πλαίσια του cross-validation και να βασίζεται πάνω στα δεδομένα εκπαίδευσης μόνο (diagnosis added). Στην άσκηση αυτή θα δούμε αν είχε δίκαιο, συμπληρώνοντας τον κώδικα στα αρχεία `overfittingTest.m` και `similarityMeasure.m`.

Στο κώδικα αυτό παράγουμε με εντελώς τυχαίο τρόπο 1000 χαρακτηριστικά για 25 άτομα (15 ασθενείς και 10 φυσιολογικοί). Όπως είναι αναμενόμενο τα τυχαία χαρακτηριστικά δεν μας δίνουν καμία πληροφορία για ένα άτομο και ένας ταξινομητής που θα εκπαιδευτεί με αυτά τα δεδομένα θα προβλέπει την κατηγορία ενός ατόμου με πιθανότητα κοντά στο 50%.

A) Από το αρχείο `overfittingTest.m` τρέξτε το τμήμα του κώδικα «Classify without feature selection» για να υπολογίσετε την επίδοση του συστήματος χωρίς επιλογή χαρακτηριστικών. Στο συγκεκριμένο κομμάτι του κώδικα χρησιμοποιείται η μέθοδος ταξινόμησης Support Vector Machines (`svmtrain`, `svmcclassify`). Αν δεν είναι διαθέσιμες αυτές οι συναρτήσεις στον υπολογιστή που εργάζεστε, χρησιμοποιήστε έναν οποιονδήποτε άλλο ταξινομητή (π.χ. Linear and Quadratic Discriminant Analysis) όπως πιο κάτω:

```
lda = fitcdiscr(data, labels);
ldaClass = resubPredict(lda);

qda = fitcdiscr(data, labels, 'DiscrimType', 'quadratic');
qdaResubErr = resubLoss(qda);
```

B) Συμπληρώστε τον κώδικα στη συνάρτηση `similarityMeasure` (αρχείο `similarityMeasure.m`), έτσι ώστε αυτή να υπολογίζει το τετράγωνο του συντελεστή συσχέτισης Pearson ( $\rho_{XC}$ )<sup>2</sup> για ένα χαρακτηριστικό  $X$  και τις κλάσεις  $C$ .

Τώρα θα δοκιμάσουμε να κάνουμε feature selection. Χρησιμοποιήστε αν θέλετε ως υπόδειγμα τον παρακάτω κώδικα και προσαρμόστε τον κατάλληλα για τα παρακάτω ερωτήματα (C) και (D):

```
% Feature selection
r = zeros(numFeatures, 1);
for j = 1:numFeatures
    % Compute the similarity measure
    r(j) = similarityMeasure(features, labels);
```

```
end
[rSorted, sortedFeatureIndices] = sort(r, 'descend');
% The indices of the selected features.
selectedIndices = sortedFeatureIndices(1:numSelectedFeatures);
```

- C) Συμπληρώσετε τον κώδικα στο αρχείο `overfittingTest.m` έτσι ώστε το feature selection να γίνεται 25 φορές μέσα στο cross-validation (με τα 24 δείγματα εκπαίδευσης).
- D) Συμπληρώσετε τον κώδικα στο αρχείο `overfittingTest.m` έτσι ώστε το feature selection να γίνεται μία φορά πριν το cross-validation (με όλα τα 25 δείγματα).

Σημειώστε τις παρατηρήσεις σας στο παράδειγμα αυτό και δώστε μια εξήγηση για το αποτέλεσμα.