



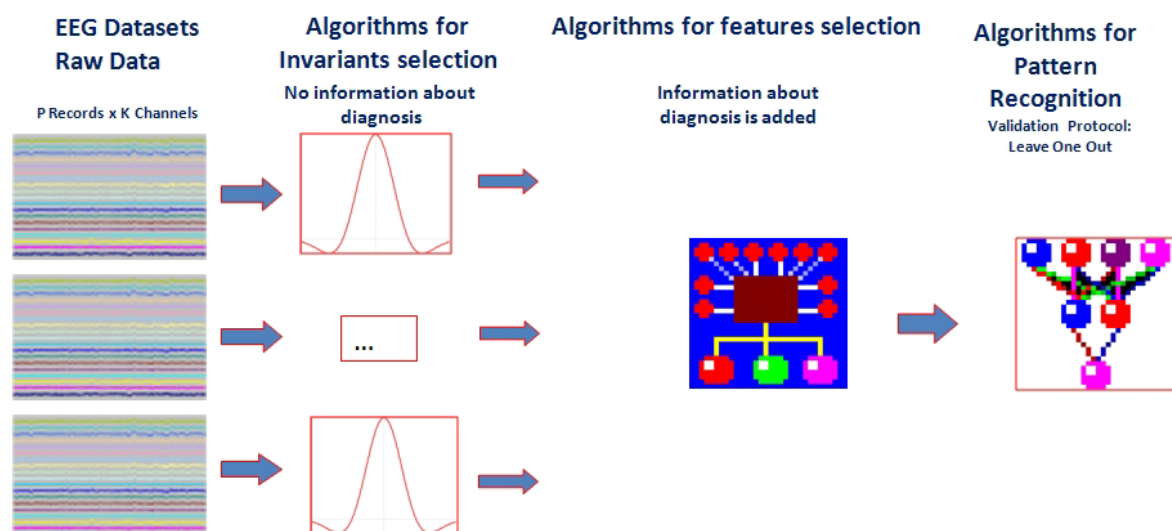
#### Οδηγίες:

- 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:
  - Οι απαντήσεις στις εργασίες, τα quiz και τις εξετάσεις, ο κώδικας και γενικά οτιδήποτε αφορά τις εργασίες θα είναι προϊόν δικής μου δουλειάς.
  - Δεν θα διαθέσω κώδικα, απαντήσεις και εργασίες μου σε κανέναν άλλο.
  - Δεν θα εμπλακώ σε άλλες ενέργειες με τις οποίες ανέντιμα θα βελτιώνω τα αποτελέσματα μου ή ανέντιμα θα αλλάζω τα αποτελέσματα άλλων.
- Η εργασία είναι ατομική
- Ημερομηνία παράδοσης: **Κυριακή, 28/6/2020 στις 23:00**
- Παραδοτέα:** α) Κώδικας και β) Αναφορά με τις απαντήσεις, παρατηρήσεις, πειράματα, αποτελέσματα και οδηγίες χρήσης του κώδικα.

#### Θέμα 1: 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 μεγάλου αριθμού των χαρακτηριστικών οι ερευνητές διάλεξαν τα 100 χαρακτηριστικά που διαφοροποιούν καλύτερα τους ασθενείς από τους φυσιολογικούς. Επίσης λόγω του μικρού αριθμού των δειγμάτων χρησιμοποίησαν την τεχνική leave-one-out (cross validation). Σύμφωνα με αυτή την τεχνική αν έχουμε  $N$  δείγματα, τα χωρίζουμε σε δύο ομάδες  $N-1$  δειγμάτων και ενός δείγματος. 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.

Οι συγγραφείς του άρθρου οργάνωσαν την έρευνά τους σύμφωνα με το παρακάτω διάγραμμα.



Ένας από τους κριτές διαμαρτυρήθηκε ότι το cross-validation πρέπει να περιλαμβάνει και το feature selection (diagnosis added). Στην άσκηση αυτή θα δούμε αν είχε δίκαιο, συμπληρώνοντας τον κώδικα overfittingTest.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%.

Τρέξτε το τμήμα του κώδικα «Classify without feature selection» για να επιβεβαιώσετε τα παραπάνω. Αν δεν είναι διαθέσιμα τα Support Vector Machines (svmtrain, svmclassify) στον υπολογιστή που εργάζεστε τότε χρησιμοποιήστε ένα οποιονδήποτε άλλο ταξινομητή (π.χ. Linear and Quadratic Discriminant Analysis

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

Τώρα θα δοκιμάσουμε να κάνουμε feature selection. Μια απλή μέθοδος είναι να διαλέξουμε τα χαρακτηριστικά που σχετίζονται καλύτερα με τα labels.

```
% 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);
```

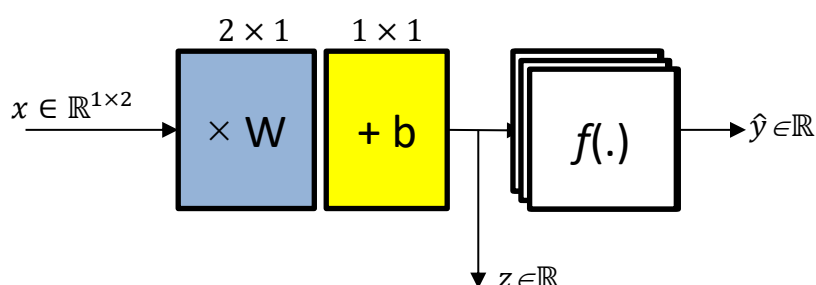
α) Συμπληρώστε τον κώδικα όπου το feature selection γίνεται 25 φορές μέσα στο cross-validation (με τα 24 δείγματα εκπαίδευσης).

β) Συμπληρώστε τον κώδικα όπου το feature selection γίνεται μία φορά πριν το cross-validation (με όλα τα 25 δείγματα).

## Θέμα 2: Υλοποίηση ενός απλού νευρωνικού δικτύου.

Σ' αυτή την άσκηση ο στόχος είναι να υλοποιηθεί και εκπαιδευτεί ένα νευρωνικό δίκτυο χωρίς την χρήση έτοιμων προγραμμάτων. Η υλοποίηση μπορεί να γίνει είτε σε matlab είτε σε python και numpy.

Δίνεται ένα σύνολο δεδομένων εκπαίδευσης  $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$  με  $x^{(i)} \in \mathbb{R}^{1 \times 2}$  και  $y^{(i)} \in \mathbb{R}$ .



Το νευρωνικό δίκτυο όπως φαίνεται πιο πάνω ορίζεται από την σχέση:

$$\hat{y}^{(i)} = f(x^{(i)}W + b)$$

όπου  $x^{(i)} \in \mathbb{R}^{1 \times 2}$  είναι ένα δείγμα,  $W \in \mathbb{R}^{2 \times 1}$ ,  $b \in \mathbb{R}^{1 \times 1}$ , και

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

Το σφάλμα ανάμεσα στην πρόβλεψη  $\hat{y}^{(i)}$  του νευρωνικού δικτύου και στην τιμή  $y^{(i)}$  που αντιστοιχεί στα δεδομένα εισόδου  $x^{(i)}$ , το μετράμε και σε αυτή την άσκηση με τη συνάρτηση cross-entropy

$$J(y^{(i)}, \hat{y}^{(i)}; W, b) = -y^{(i)} \ln(\hat{y}^{(i)}) - (1 - y^{(i)}) \ln(1 - \hat{y}^{(i)})$$

Για να έχουμε αριθμητική ευστάθεια, συνήθως υπολογίζουμε την cross-entropy για ένα σύνολο  $B$  δειγμάτων και παίρνουμε τον μέσο όρο. (Το σύνολο  $B$  δειγμάτων ονομάζεται batch).

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

α) Αν θέσουμε  $z^{(i)} = x^{(i)}W + b$ , οπότε  $\hat{y}^{(i)} = f(z^{(i)})$ , να δείξετε ότι

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

β) Να δείξετε ότι

$$\frac{\partial J}{\partial z^{(i)}} = -y^{(i)} + \hat{y}^{(i)}, \quad i \in \text{batch}$$

γ) Με τον κανόνα της αλυσίδας να υπολογίσετε τις μερικές παραγώγους

$$\frac{\partial J}{\partial W}, \quad \frac{\partial J}{\partial b}$$

Ο κανόνας παραγώγισης με τον πίνακα  $W$  είναι:  $\frac{\partial(xW)}{\partial W} = x^T$

δ) Να συμπληρώσετε τον κώδικα της άσκησης (είτε σε matlab είτε σε python) και να εκπαιδεύσετε το νευρωνικό δίκτυο.

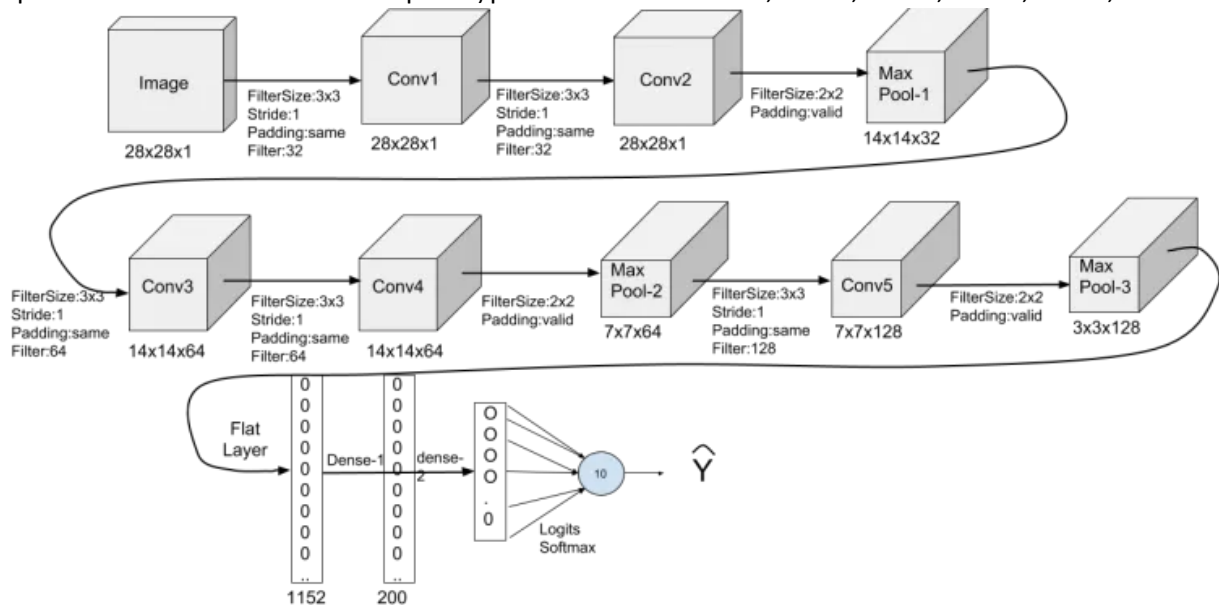
### Θέμα 3: Convolutional Neural Networks for Image Recognition

Σε αυτή την άσκηση θα δοκιμάσουμε διάφορους ταξινομητές νευρωνικών δικτύων χρησιμοποιώντας το σύνολο δεδομένων Fashion-MNIST [1]. Οι εικόνες του Fashion-MNIST χωρίζονται σε 10 κατηγορίες {'T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot'}.

Στα αρχεία της άσκησης θα βρείτε τον κώδικα fashion.py που υλοποιεί έναν ταξινομητή με dense neural networks. Σε όλες τις παρακάτω ερωτήσεις τρέξτε το πρόγραμμα και δημιουργήστε ένα γράφημα με τις τιμές της ακρίβειας (accuracy) στα δεδομένα εκπαίδευσης (training data) και στα δεδομένα επικύρωσης (validation data). Τι παρατηρείτε;

1. Τρέξτε το πρόγραμμα με την αρχιτεκτονική που δίνεται και epoch = 400. Τι παρατηρείτε με το training και validation accuracy;
2. Δοκιμάστε τους αλγορίθμους βελτιστοποίησης adam, sgd, rmsprop, nadam, adamax, και ftrl. Ποιος δίνει το καλύτερο accuracy;

3. Αλλάξτε την αρχιτεκτονική του δικτύου ώστε να υλοποιεί το παρακάτω σχήμα. Θα χρειαστεί να μετασχηματίσετε τις διαστάσεις των δεδομένων σας από  $\#images \times height \times width$  σε  $\#images \times height \times width \times \#channels$ , όπου  $\#images$  είναι ο αριθμός των εικόνων και  $\#channels=1$  είναι ο αριθμός των καναλιών. Επίσης αλλάξτε το epoch από 400 σε 50 για να τελειώνει πιο γρήγορα η εκπαίδευση και χρησιμοποιήστε τον adam optimizer.
4. Προσθέστε Batch Normalization αμέσως μετά τα επίπεδα conv1, conv2, conv3, conv4, conv5, dense1.



Δοκιμάστε να τοποθετήσετε το Batch Normalization πριν την συνάρτηση ReLU (Δηλαδή Conv χωρίς ReLU -> BatchNormalization -> ReLU).

5. Προσθέστε dropout(0.2) αμέσως μετά το πρώτο maxPooling, dropout(0.3) αμέσως μετά το δεύτερο maxPooling, dropout(0.4) αμέσως μετά το τρίτο maxPooling και dropout(0.5) αμέσως πριν το τελευταίο dense(10) layer.
1. Han Xiao, Kashif Rasul, Roland Vollgraf, Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, arXiv:1708.07747v1