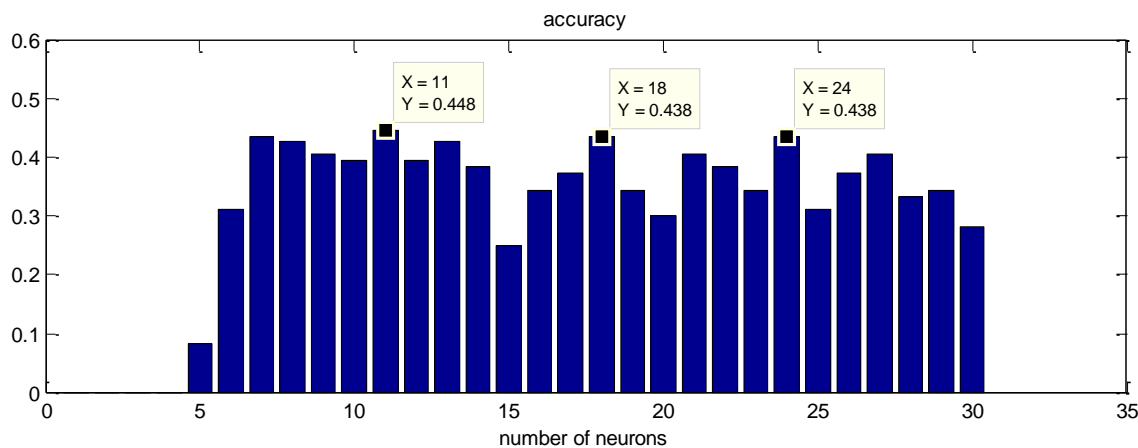


## Μέρος 1: Υλοποίηση

Σε αυτό το μέρος θα γίνει παρουσίαση των αποτελεσμάτων της άσκησης μετά απο υλοποίηση των βημάτων όπως αναφέρονται στην εκφώνησή της. Για την υλοποίηση αυτή κατασκευάστηκαν 4 mfiles και 2 συναρτήσεις matlab. Για το πρώτο μέρος το οποίο αφορά την προεπεξεργασία των δεδομένων κατασκευάστηκε το mfile **data\_preprocessing.m** το οποίο τροποποιεί τους πίνακες TrainData , TrainDataTargets, TestData και TestDataTargets με τον τρόπο που ζητείται. Στη συνέχεια το mfile **create\_the\_net.m** εκπαιδεύει και κατασκευάζει το νευρωνικό δίκτυο με συναρτήσεις εισόδου που καθορίζονται απο την εντολή newff κάνοντας στο τέλος έναν έλεγχο με το testing set. Για επανεξέταση και αξιολόγηση του δικτύου κατασκευάστηκε το mfile **test\_the\_net.m** με το οποίο μπορούμε έχοντας αποθηκεύσει κάποιο δίκτυο και έπειτα απο την κατάλληλη προεπεξεργασία των δεδομένων να το ανακαλέσουμε και να το εξετάσουμε για πιθανή σύγκριση με κάποιο άλλο δίκτυο. Τέλος κατασκευάστηκε το mfile **evaluate\_all.m** για να εξεταστούν όλα τα δίκτυα με μία και με δύο βαθμίδες, ξεχωριστά κάθε φορά κάτι που καθορίζεται απο τη συνάρτηση **BEST.m** που καλείται απο το εν λόγω m-file. Περισσότερες και αναλυτικές οδηγίες για το πώς θα χρησιμοποιήσετε τα παραπάνω mfiles θα βρείτε στο readme αρχείο που βρίσκεται εντός του παραδωταίου κώδικα.

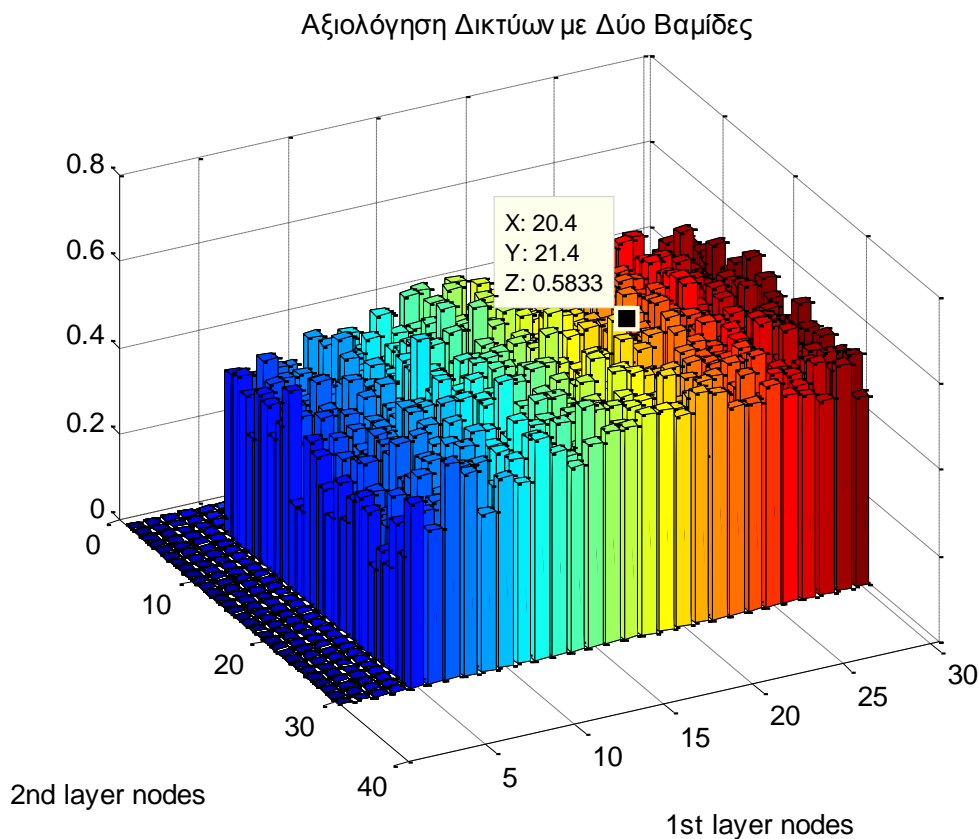
Βάσει όλων των παραπάνω έγινε έλεγχος για δίκτυα με μια και δύο κρυφές βαθμίδες ξεχωριστά και για διαφορετικές συναρτήσεις εκπαίδευσης και μάθησης. Αρχικά χρησιμοποιήθηκε η default συναρτήσεις εκπαίδευσης και μάθησης trainlm και traingdm για μια κρυφή βαθμίδα με 5 εως 30 νευρώνες. Τα αποτελέσματα της ακρίβειας κάθε παραγόμενου δικτύου φαίνονται στο *Σχήμα 1*.



Σχήμα 1 : Ακρίβεια δικτύων με μια κρυφή βαθμίδα για 5 εως 30 νευρώνες

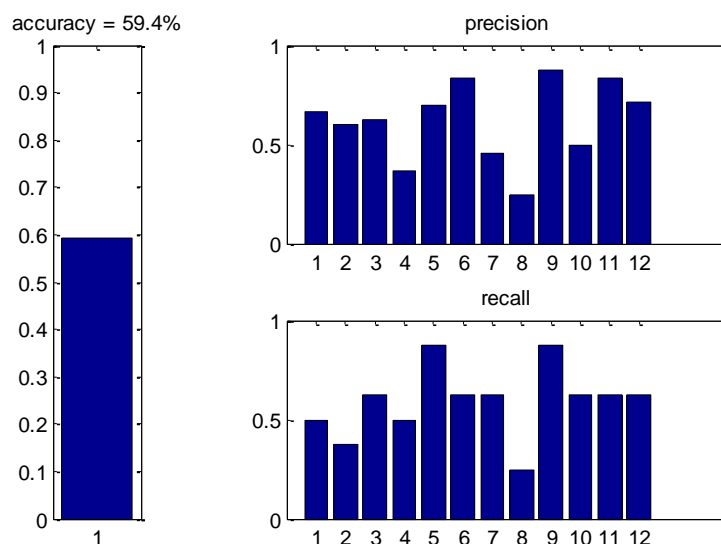
Στο παραπάνω σχήμα φαίνεται οτι το δίκτυο με την καλύτερη ακρίβεια είναι αυτό με 11 νευρώνες. Αξίζει να αναφέρουμε εδώ οτι κάθε φορά που εκπαιδεύεται ένα δίκτυο τα αποτελέσματα απο το testing set μπορεί να είναι διαφορετικά ακόμα και αν το δίκτυο αυτό έχει του ίδιους αριθμούς επιπέδων και νευρώνων σε κάθε επίπεδο. Αυτό ωφείλεται στην διαφορετική και τυχαία αρχικοποίηση των βαρών καθώς και στον τυχαίο διαχωρισμό του training set σε trainRatio και valRatio. Παρόλο που το trainRatio ήταν πάντα το 80% των trainData και το valRatio το υπόλοιπο 20% ,κάθε φορά ο διαχωρισμός γίνεται με τυχαίο τρόπο και σ'αυτό ωφείλεται η κατασκευή διαφορετικού δικτύου. Ωστόσο έγιναν αρκετές δοκιμές για να κατασκευαστεί τελικά το βέλτιστο δυνατό δίκτυο.

Στη συνέχεια με τις ίδιες συναρτήσεις εκπαίδευσης και μάθησης έγινε εκπαίδευση και έλεγχος ακρίβειας για δίκτυα με δύο κρυφές βαθμίδες και για κάθε συνδιασμό νευρώνων (απο 5 και πάνω σε κάθε βαθμίδα). Τα αποτελέσματα φαίνονται στο *Σχήμα 2*.



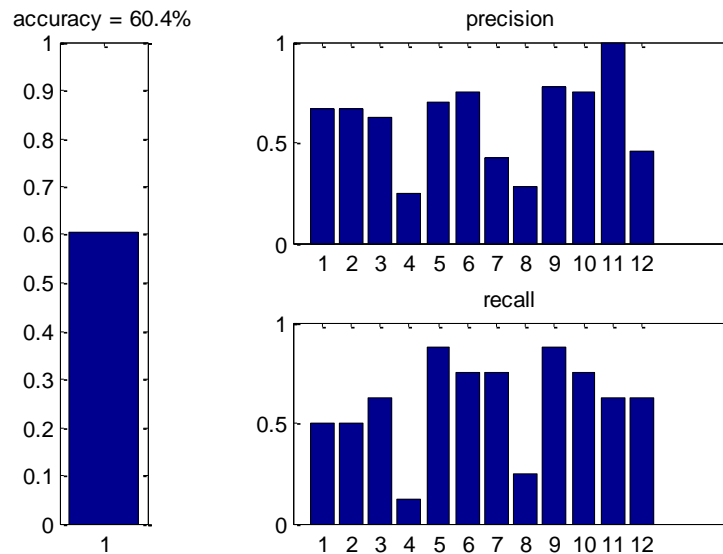
**Σχήμα 2 :** Ακρίβεια δικτύων με δυο κρυφές βαθμίδες με 5 εως 30 νευρώνες έκαστη

Έπειτα απο έλεγχο των αποτελεσμάτων της ακρίβειας των δικτύων που δημιουργήθηκαν και την απεικόνιση των δεδομένων με τη συνάρτηση *bar3* του MatLab βρέθηκε οτι το βέλτιστο δίκτυο παρουσίασε ακρίβεια 58.3% και επιτεύχθει για 20 νευτώνες στην 1η βαθμίδα και 21 νευρώνες στην επόμενη. Για το δίκτυο αυτό έγιναν στη συνέχεια αρκετές δοκιμές για όλες τις συναρτήσεις εκπαίδευσης και μάθησης. Η βέλτιστη ακρίβεια που επιτεύχθηκε ήταν 59.4% με συναρτήσεις *traingdx* και *learnqdm* και τα αντίστοιχα αποτελέσματα precision και recall φαίνονται στο *Σχήμα 3*.



**Σχήμα 3 :** accuracy,precision και recall για δυο κρυφές βαθμίδες με 20 και 21 νευρώνες

Το δίκτυο που κατασκευάστηκε αποθηκεύτηκε ως MAT-fille και βρίσκεται εντός του παραδωταίου κώδικα με όνομα **NET\_20\_21\_gdx\_gdm2(59.4%).mat**. Εν συνεχεία των δοκιμών επανεξετάστηκαν με όλες τις συναρτήσεις τα δίκτυα με μια κρυφή βαθμίδα και τελικά κατασκευάστηκε δίκτυο με μια μονο κρυφή βαθμίδα και 28 νευρώνες όπου η ακρίβεια που επιτεύχθη ήταν 60.4%. Τα αποτελέσματα μαζί με precision και recall φαίνονται στο *Σχήμα 4*.



**Σχήμα 4 :** accuracy,precision και recall για μια κρυφή βαθμίδα με 28 νευρώνες

Το δίκτυο αυτό αποθηκεύτηκε και βρίσκεται εντός του παραδωταίου κώδικα με όνομα **NET\_28\_gdx\_gdm.mat**. Αν και η ακρίβεια αυτή είναι αρκετά καλή επιτεύχθη μόνο για μια εκ των εκπαιδεύσεων ενώ για τις υπόλοιπες ήταν σχετικά μικρή.

Τα επόμενα βήματα περιγράφονται μέσω των ερωτημάτων (κυρίως 6 και 7) προς αποφυγή επαναλήψεων.

## Μέρος 2: Απαντήσεις στα ερωτήματα

### Ερώτημα 1

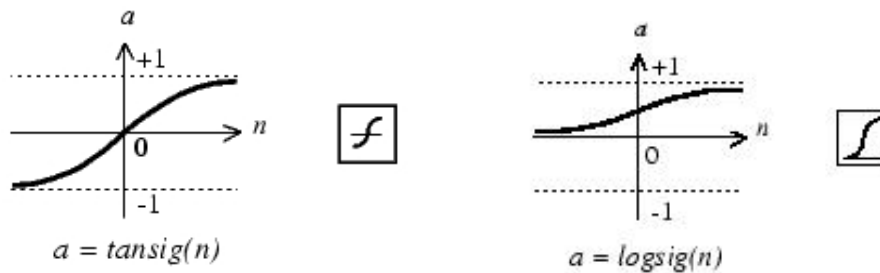
Στην περίπτωση που κατα την προεπεξεργασία των δεδομένων δεν κρατούσαμε το ίδιο πλήθος τμημάτων ανα κατηγορία τότε η εκπαίδευση θα ήταν μεροληπτική δηλαδή το τελικό δίκτυο θα έδειχνε μια προτίμηση στην κατηγορία της οποίας τα τμήματα ήταν πολύ περισσότερα κατα την εκπαίδευση.

### Ερώτημα 2

Κατα την προεπεξεργασία των δεδομένων χρησιμοποιήσαμε τη συνάρτηση `processprca` με την οποία ουσιαστικά κρατήσαμε τις πιο ασυσχέτιστες συνιστώσες του διανύσματος εισόδου. Επίσης με την `removeconstantrows` απορρίψαμε όσες συνιστώσες δεν είχαν να προσφαιρουν απολύτως τίποτα καθώς η τιμή τους δεν μεταβάλεται απο δείγμα σε δείγμα και απο κατηγορία σε κατηγορία. Η εκπαίδευση του δικτύου με αυτόν τον τρόπο απλοποιείται κατα πολύ. Σε αντίθετη περίπτωση και πάλι το δίκτυο θα μπορούσε να εκπαιδευτεί αλλά η πολυπλοκότητά του θα ήταν πολύ μεγάλη χωρίς ωστόσο αυτό να μας παράγει ένα καλύτερο μοντέλο.

### Ερώτημα 3

Η συνάρτηση ενεργοποίησης στο στάδιο εξόδου θα πρέπει να είναι η  $\text{tansig}$  δηλαδή η συνάρτηση αντίστροφης εφαπτομένης. Ο λόγος είναι ότι το δίκτυο κάνει απο μόνο του μια κανονικοποίηση τόσο στα δεδομένα εισόδου όσο και στα δεδομένα εξόδου απο -1 έως 1. Έτσι θέλουμε η έξοδος του δικτύου να είναι 1 στον νευρώνα ο οποίος αντιστοιχεί στη σωστή κατηγορία και -1 στους υπόλοιπους νευρώνες. Κάτι τέτοιο μπορεί να υλοποιηθεί με τη συνάρτηση αντίστροφης εφαπτομένης καθώς το πεδίο τιμών της είναι απο -1 έως 1. 'Αν δεν γινόταν αυτή η by default κανονικοποίηση των δεδομένων εξόδου (data targets) και παρέμεναν οι τιμές απο 0 και 1 όπως περιγράφεται στην άσκηση τότε η ιδανική συνάρτηση θα ήταν η  $\text{logsig}$  της οποίας το πεδίο τιμών είναι απο 0 έως 1.



Σχήμα 5 : Η ιδανική συνάρτηση ενεργοποίησης είναι η  $\text{tansig}$

### Ερώτημα 4

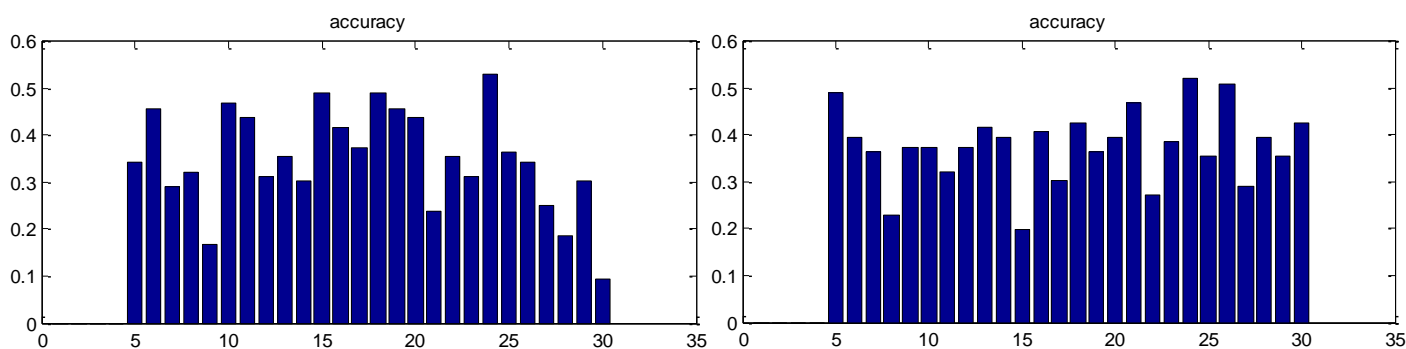
Στην αρχή του 5ου βήματος ξεκινήσαμε με 1 κρυφό επίπεδο και εξετάσαμε τα αποτελέσματα για 5 έως 30 νευρώνες με τις by default συναρτήσεις εκπαίδευσης και μάθησης  $\text{trainlm}$  και  $\text{learngdm}$  αντίστοιχα. Τα αποτελέσματα της ακρίβειας κάθε νευρωνικού είναι εμφανή στο Σχήμα 1 του 1ου Μέρους. Απο το σχήμα αυτό παρατηρούμε ότι η βέλτιστη ακρίβεια επιτεύχθηκε για 11 νευρώνες στην κρυφή βαθμίδα ενώ αρκετά καλά αποτελέσματα είχαμε και για 7, 18 και 24 νευρώνες. Όπως αναφέραμε και στο 1ο Μέρος βέβαια κάθε φορά που γίνεται εκ νέου μια εκπαίδευση το δίκτυο μπορεί να έχει διαφορετική ακρίβεια και αυτό λόγω της τυχαιότητας με την οποία γίνεται η αρχικοποίηση των βαρών αλλά και ο διαχωρισμός στο train και valuation set. Έτσι μπορεί ο εγκλωβισμός σε ένα τοπικό ελάχιστο να οδηγήσει σε ένα κατά πολύ χειρότερο μοντέλο παρόλο που για τον ίδιο αριθμό νευρώνων μπορεί να έχουμε και τη βέλτιστη ακρίβεια σε μια άλλη εκπαίδευση.

Στη συνέχεια εκπαιδεύτηκαν δίκτυα με δύο κρυφές βαθμίδες και τα αποτελέσματα της ακρίβειας είναι εμφανή στο Σχήμα 2 του 1ου Μέρους. Απο το σχήμα αυτό, στο οποίο φαίνονται οι ακρίβειες για κάθε δίκτυο και είναι αποθηκευμένες στο **AC\_all.mat** του παραδωταίου κώδικα, φαίνεται ότι η καλύτερη ακρίβεια που επιτεύχθηκε είναι για 20 και 21 νευρώνες στα κρυφά επίπεδα και ίση με 59,4%. Και στις δύο περιπτώσεις, με ένα και δύο κρυφά επίπεδα, όταν οι νευρώνες είναι πολύ λίγοι τότε το μοντέλο είναι σχετικά "φτωχό", δεν έχει δηλαδή την πολυπλοκότητα που χρειάζεται για την κατηγοριοποίηση των δεδομένων. Στην αντίθετη περίπτωση όπου οι νευρώνες είναι πάρα πολλοί το δίκτυο ουσιαστικά γίνεται υπερβολικά επιλεκτικό χάνοντας την ικανότητα γενίκευσής του καθώς τείνει να μοντελοποιήσει και τον θόρυβο που υπεισέρχεται στις μετρήσεις ή εγλωβίζεται εύκολα σε τοπικά ελάχιστα της συνάρτησης σφάλματος.

### Ερώτημα 5

Στην περίπτωση που χρησιμοποιείται η συνάρτηση μάθησης  $\text{learngd}$  παρατηρούμε ότι τα αποτελέσματα της ακρίβειας είναι αρκετά μικρότερα για κάποια δίκτυα. Τα αποτελέσματα φαίνονται στο Σχήμα 6. Ο λόγος είναι ότι κατά την μέθοδο κατάβασης χωρίς τον όρο ορμής είναι πολύ πιθανό να εγκλωβιστούμε σε ένα τοπικό ελάχιστο και ο αλγόριθμος να τερματίσει εκεί παράγοντας ένα σχετικά

κακό μοντέλο. Σε αντίθετη περίπτωση η προσθήκη όρου ορμής με την `learnsgdm` μας δίνει τη δυνατότητα να ξεφύγουμε από το τοπικό αυτό ακρότατο και να συνεχίσουμε προς έυρεση του ολικού ελαχίστου της συνάρτησης σφάλματος.

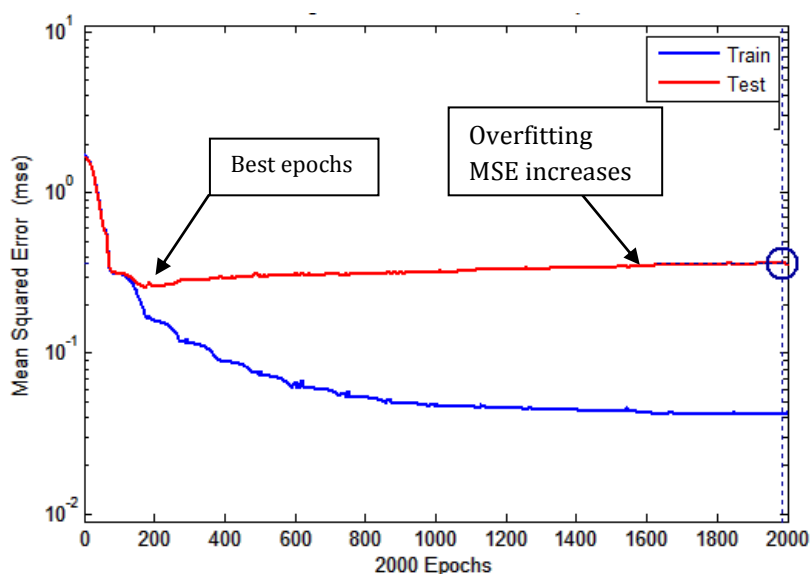


Σχήμα 6 : Ακρίβεια δικτύων χωρίς χρήση όρου ορμής (αριστερά) και με όρο ορμής (δεξιά)

Βέβαια σε περίπτωση μη εγλωβισμού σε ένα τοπικό ελάχιστο και οι δύο μέθοδοι θα παρήγαν τα ίδια αποτελέσματα. Κάτι τέτοιο ίσως δεν είναι εμφανές στο παραπάνω σχήμα και αυτό συμβαίνει λόγω των διαφορετικών αρχικών σε κάθε δίκτυο όπως αναφέραμε σε προηγούμενα ερωτήματα.

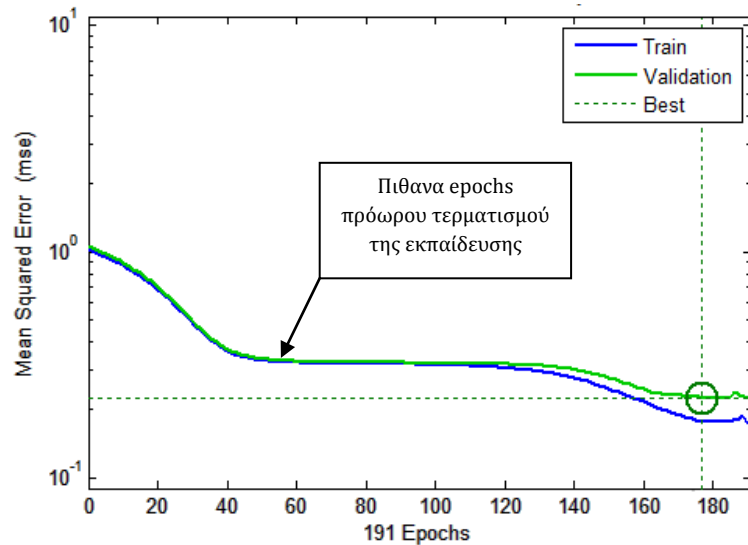
## Ερώτημα 6

Με τη χρήση ενός μέρους των δεδομένων ως validation set κάθε φορά γίνεται έλεγχος της αποτελεσματικότητας του δικτύου. Έτσι ενώ όπως είναι λογικό η συνάρτηση σφάλματος υπολογιζόμενη από το ίδιο το training set θα είναι φθίνουσα καθώς αυξάνουν οι εποχές, παρόλα αυτά δεν παρατηρείται και το ίδιο πάνω στο testing set. Συγκεκριμένα το σφάλμα φτάνει σε ένα ελάχιστο σε κάποιο epoch και στη συνέχεια αρχίζει και αυξάνεται καθώς το μοντέλο παρουσιάζει overfitting όπου ουσιαστικά μοντελοποιείται ο "θόρυβος" των δεδομένων εισόδου. Έτσι με χρήση ενός μέρους του training set ως validation set (στην προκειμένη περίπτωση το 20%) μπορούμε έγκαιρα να σταματήσουμε την εκπαίδευση όταν αυξανόμενων των εποχών αυξάνεται και το σφάλμα πάνω στο validation set. Το ζητούμενο είναι λοιπόν να σταματήσει η εκπαίδευση στην εποχή στην οποία η συνάρτηση σφάλματος πάνω σε δεδομένα αξιολόγησης του δικτύου παρουσιάζει ελάχιστο. Κάτι τέτοιο φαίνεται στο Σχήμα 7 όπου για την απεικόνιση του σχήματος χρησιμοποιήσαμε το 80% ως training set και το υπόλοιπο 20% ως testing set.



Σχήμα 7 : Overfitting και Early Stopping Method of Training

Στο παραπάνω σχήμα εκπαίδευσάμε το δίκτυο εως 2000 εποχές για να είναι εμφανή τα αποτελέσματα. Στο Σχήμα 8 φαίνεται ένα πρόβλημα που μπορεί να προκύψει απο τη μέθοδο.

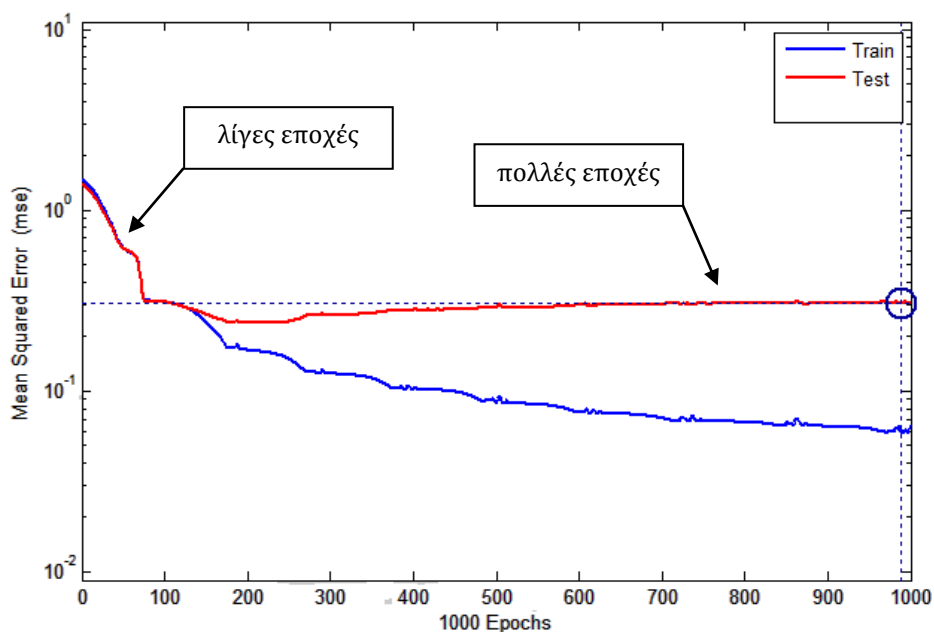


Σχήμα 8 : Πιθανό πρόβλημα πρόωρου τερματισμού της εκπαίδευσης

Στο παραπάνω σχήμα φαίνεται ο τι υπάρχει κίνδυνος ο αλγόριθμος να τερματίσει πρόωρα αν παρουσιαστεί τοπικό ελάχιστο στη συνάρτηση σφάλματος με το validation set κάτι το οποίο συμβαίνει συχνά στην εν λόγω εργασία, κυρίως αν χρησιμοποιήσουμε ως συνάρτηση εκπαίδευσης την traingdx η οποία ωστόσο μπορεί να μας δώσει και ένα πολύ καλό μοντέλο όπως είδαμε στο 1ο Μέρος.

## Ερώτημα 7

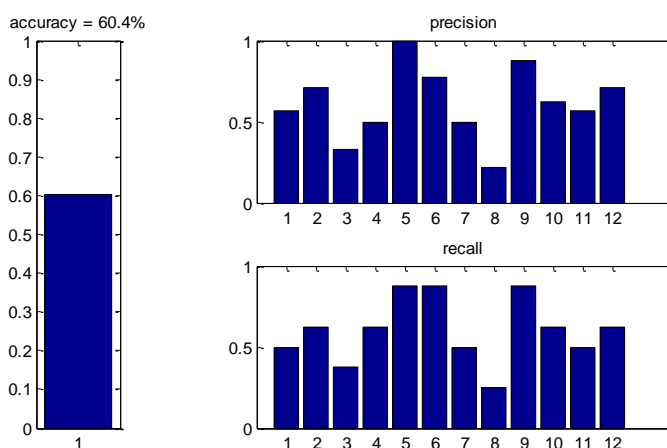
Η συγκεκριμένη ερώτηση έχει νόημα μόνο εαν δεν χρησιμοποιήσουμε τη μέθοδο Early Stopping Method of Training με validation set. Η απάντηση είναι εμφανής απο το Σχήμα 9.



Σχήμα 9 : MSE για λίγες και για πολλές εποχές

Απο το σχήμα αυτό είναι φανερό οτι για πολύ λίγες εποχές το μοντέλο παρουσιάζει μεγάλο σφάλμα και ο λόγος είναι οτι δεν έχει προλάβει να εκπαιδευτεί επαρκώς. Αντίθετα για πάρα πολλές εποχές το σφάλμα έχει αυξηθεί αρκετά απο τη βέλτιστη τιμή του καθώε το μοντέλο έχει υπερεκπαιδευτεί και παρουσιάζεται overfitting οπου το μοντέλο στην ουσία μοντελοποιεί ακόμα και τον θόρυβο των δεδομένων εισόδου. Έτσι η ακρίβεια κατηγοριοποίησης και στις δύο αυτές περιπτώσεις παρουσιάζει αισθητά μεγάλη μείωση.

Απαντώντας σ' αυτό το ερώτημα και στο Βήμα 8 της άσκησης μπορέσαμε να παρατηρήσουμε οτι σε κάθε περίπτωση με χρήση του αλγορίθμου εκπαίδευσης `trainidx` και μάθησης του `learnidx` οι εποχές για τις οποίες παρουσιάστηκε το ελάχιστο σφάλμα στο validation set και κατα συνέπεια τερματίστηκε η εκπαίδευση, ήταν 185 εποχές για τις οποίες στη συνέχεια εκπαιδεύσαμε δίκτυο με δυο κρυφές βαθμίδες απο 20 και 21 νευρώνες αντίστοιχα και τα αποτελέσματα ήταν πάρα πολύ καλά σε όλες τις περιπτώσεις με βέλτιστη ακρίβεια 60.4%



**Σχήμα 10 :** Αποτελέσματα Δικτύου με εκπαίδευση χωρίς validation set για 185 epochs

Αυτό μας δίνει παράλληλα και την γενίκευση που ζητείται στην εν λόγω άσκηση. Ένα μοντέλο με δύο κρυφές βαθμίδες απο 20 και 21 νευρώνες αντίστοιχα το οποίο θα εκπαιδευτεί με συνάρτηση εκπαίδευσης την `trainidx` και μάθησης `learnidx` για ακριβώς 185 εποχές θα είναι πάντα ενα αρκετά ικανοποιητικό μοντέλο. Τα αποτελέσματα του σχήματος 10 επιβεβαιώνουν τον κανόνα αυτό οτυλάχιστον για μια περίπτωση. Και πάλι είναι πιθανό να υπάρχουν αποκλίσεις απο εκπαίδευση σε εκπαίδευση αν και κά'τι τέτοιο θα μπορούσε πλέον να διορθωθεί με συγκεκριμένο και μη τυχαίο permutation των δεδομένων εισόδου και αρχικοποίηση των βαρών.

## Ερώτημα 8

Απο τα σχήματα 1, 4 και 10 παρατηρούμε διαφορετική απόδοση σε κάποιες κατηγορίες (π.χ αρκετά μικρό precision και recall στην κατηγορία 8 ). Αυτό μπορεί να ωφείλεται σε διάφορες λεπτομέρειες κατα την εκπαίδευση του δικτύου. Κατα πρώτον η σειρά με την οποία παρουσιάζονται τα train data στην εκπαίδευση έχει άμεση συνέπεια στον τρόπο με τον οποίο ανανεώνονται τα βάρη του δικτύου και συνεπώς το εξαγόμενο δίκτυο θα παρουσιάζει διαφορετική απόδοση ανα κατηγορία. Επίσης μετά την απόρριψη των συσχετιζόμενων συνιστωσών η διάσταση του διανύσματος εισόδου μειώθηκε κατα πολύ ενώ στη συνέχεια κρατήσαμε ενα πολύ μικρό σύνολο δεδομένων εκπαίδευσης ωστε να έχουμε ίσο πλήθος δειγμάτων ανα κατηγορία. Τέλος στο testing set το πλήθος των δειγμάτων δεν είναι ισο σε κάθε κατηγορία το οποίο αλλάζει τα στατιστικά εξαγόμενα αποτελέσματα κάθε κατηγορίας. Διορθώνοντας τα παραπάνω θα σίγουρα μια πιο ομαλή συμπεριφορά κατα στη

απόδοση. Ωστόσο για να έχουμε βέλτιστη απόδοση σε κάθε κατηγορία μια προτεινόμενη διαδικασία είναι κατά τη διαδικασία της εκπαίδευσης να παρατηρείται συνεχώς η επιλεκτικότητα του δικτύου σε κάθε κατηγορία μέσω του validation set. Έτσι το επόμενο διάνυσμα εισόδου προς ανανέωση των βαρών θα επιλέγουμε κάθε φορά να ανήκει στην κατηγορία για την οποία παρατηρείται η μικρότερη απόδοση. Αυτό σε συνδιασμό με την παρατήρηση του δικτύου ως προς τη συνολική του απόδοση για τον έγκαιρο τερματισμό της εκπαίδευσης και αποφυγή του overfitting θα μας έδινε στο τέλος ένα δίκτυο το οποίο θα παρουσίαζε βέλτιστη απόδοση σε κάθε κατηγορία.