

Neural Network Assignment No3

Φωτεινή Κοκκώνη

Introduction:

Στην συγκεκριμένη εργασία επιλέχθηκε η δημιουργία Radial Basis Function Neural Network, επιλέχθηκε η ίδια βάση όπως στην δεύτερη εργασία και στην συνέχεια λύνουμε το ίδιο multi-classification πρόβλημα. Θα ακολουθήσει εξήγηση, ανάλυση και παρατήρηση της εργασίας.

Radial Basis Function NN Analysis

Τι είναι το RBF NN:

Τα συγκεκριμένα δίκτυα χρησιμοποιούνται κυρίως για εργασίες προσέγγισης συναρτήσεων καθώς είναι μία ειδική κατηγορία νευρωνικών δικτύων με εμπρόσθια ανάδραση που αποτελείται από 3 επίπεδα{input επίπεδο , hidden layer(RBF), output}. Λόγω αυτής της ξεχωριστής αρχιτεκτονικής 3ων επιπέδων και των καθολικών δυνατοτήτων προσέγγισης, έτσι προσφέρουν ταχύτερες ταχύτητες εκμάθησης και αποτελεσματική απόδοση σε προβλήματα ταξινόμησης και παλιδρόμησης

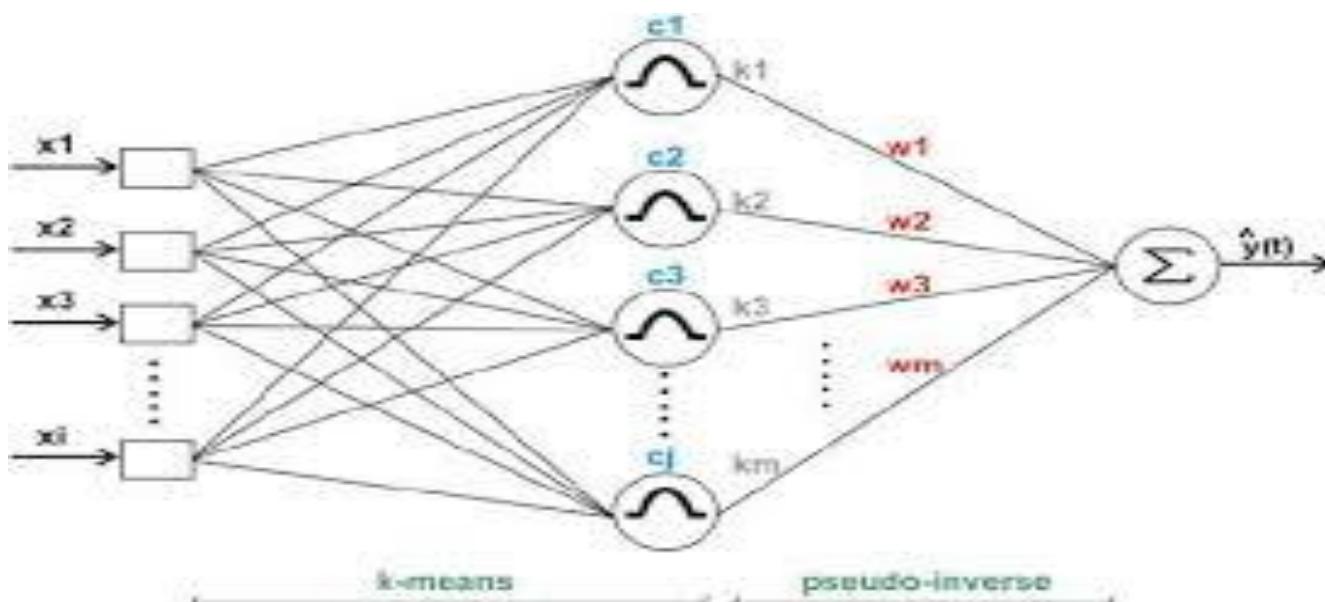
Πως λειτουργούν:

Input Vector= Το δίκτυο λαμβάνει ένα διάνυσμα διαστάσεων όσο και των δεδομένων που θέλουμε να εισάγουμε.

RBF Neurons= Κάθε νευρώνας στο hidden layer μας αντιπροσωπεύει ένα πρωτότυπο διάνυσμα από το σύνολο εκπαίδευσης. Έπειτα υπολογίζεται η Ευκλείδεια απόσταση μεταξύ του διανύσματος εισόδου και του κέντρου κάθε νευρώνα.

Activation Function= Η Ευκλείδεια απόσταση μετασχηματίζεται χρησιμοποιώντας μια Ακτινική Βασική Συνάρτηση(συνήθως Gaussian)για να υπολογιστεί η τιμή ενεργοποίησης του νευρώνα. Μειώνεται εκθετικά καθώς αυξάνεται η απόσταση.

Output Nodes=Κάθε κόμβος εξόδου υπολογίζει μία βαθμολογία με βάση ένα σταθμισμένο άθροισμα των τιμών ενεργοποίησης από όλους τους νευρώνες RBF. Όσο αφορά την ταξινόμηση επιλέγεται η κατηγορία με την υψηλότερη βαθμολογία.



3 Stages of building a RBF NN

Step 1 Selecting the Center

Μπορούν να επιλεχτούν τυχαία από το σύνολο εκπαίδευσης δεδομένων ή εφαρμόζοντας τεχνικές όπως η ομαδοποίηση k-means. K-means είναι μια εφαρμογή όπου τα κέντρα των συστάδων χρησιμοποιούνται ως κέντρα για τους νευρώνες RBF η οποία έπειτα ομαδοποιεί τα δεδομένα εισόδου σε k ομάδες

Step 2 Determining The Spread Parameters

Το σ ουσιαστικά είναι υπεύθυνο για το πλάτος της περιοχής κάθε νευρώνα RBF. Η παράμετρος εξάπλωσης μπορεί να ρυθμιστεί χειροκίνητα για κάθε νευρώνα ή να οριστεί ως σταθερά για όλους τους νευρώνες. Ο καθορισμός του σ με βάση την απόσταση μεταξύ των κέντρων είναι δυνατός συχνά με την βοήθεια μίας ευρετικής?(heuristic)όπως η διαίρεση της μεγαλύτερης απόστασης μεταξύ των κέντρων με την τετραγωνική ρίζα του διπλάσιου του αριθμού των κεντρών.

Step 3 Training the Output Weights

Linear Regression: Ελαχιστοποίηση του σφάλματος μεταξύ της αναμενόμενης εξόδου και των πραγματικών τιμών-στόχων

Pseudo-Inverse Method: Υπολογισμός βέλτιστων βαρών εξόδου μέσω Linear Regression με χρήση του αντιστρόφου πίνακα H για την ελαχιστοποίηση του τετραγωνικού σφάλματος μεταξύ προβλεπόμενων και πραγματικών τιμών.

My Implementation Analysis

Κάναμε χρήση του προαναφερόμενου dataset για τον input vector όπου εισήγαμε τα δεδομένα μας που πέρασαν από την PCA(90%) διαδικασία για να μειωθούν οι διαστάσεις τους καθώς είχαμε RGB χρώματα(3εις διαστάσεις) 32*32 bits φτάνοντας έτσι τις 3072 διαστάσεις. Με την PCA κρατήθηκαν μόνο οι «κυρίαρχες» ορίζουσες που αποτελούν και τα κύρια χαρακτηριστικά των εικόνων μας. Έπειτα χρησιμοποιήθηκε η Gaussian Radial Basis Function:

$$\varphi(x) = \exp\left(-\frac{|x - c|^2}{2\sigma^2}\right)$$

όπου

c=είναι το κέντρο των νευρώνων

σ=ελέγχει το πλάτος του RBF

Selection of RBF Centers

Χρησιμοποιήθηκε η K-Means τεχνική όπου ομαδοποιεί παρόμοια σημεία δεδομένων σε clusters χωρίζνα χρειάζεται δεδομένα με ετικέτες. Η τιμή του K αντικατοπτρίζει των αριθμό των ομάδων μας όπου στο συγκεκριμένο dataset είναι 43 όσες και οι κλάσεις μας. Έπειτα στον κώδικα εκπαιδεύουμε το μοντέλο μας με διαφορετικούς αριθμούς κέντρων για να λάβουμε στατιστικά όσο αφορά το Train/Validation accuracy στο 60/40 μοιρασμένο dataset μας. Οι νευρώνες του RBF μας εξαρτώνται από τον αριθμό των κέντρων για κάθε κλάση και τον αριθμό των κλάσεων.

$$(\text{αριθμός κλάσεων}) * (\text{αριθμός κέντρων για κάθε κλάση})$$

Estimation of our σ

Η παράμετρος σ ελέγχει το εύρος ενεργοποίησης των RBF νευρώνων. Στην περίπτωση του global sigma, χρησιμοποιείται μία κοινή τιμή σ για όλους τους νευρώνες, η οποία υπολογίζεται από τη μέση ελάχιστη απόσταση μεταξύ ΟΛΩΝ των κέντρων. Αντιθέτως στο classwise υπολογίζεται διαφορετική τιμή σ για κάθε κλάση, βάση των αποστάσεων μεταξύ των κέντρων της ίδιας κλάσης, επιτρέποντας την προσαρμογή στη διαφορετική ενδοκλασική μεταβλητότητα. Το εύρος τιμών σ που προκύπτει αντανακλά

τη διαφοροποίηση στη δομή των κλάσεων στο χώρο χαρακτηριστικών PCA. Οι μικρότερες τιμές αντιστοιχούν σε πιο συμπαγείς κλάσεις και οι μεγαλύτερες σε κλάσεις με μεγαλύτερη διασπορά, βοηθώντας έτσι στο multi-classification.

Training the Output Weights

Για την έξοδο χρησιμοποιήθηκε η μέθοδος ridge regressions όπου αποτελεί μια διαφορετική μορφή Linear Regressions όπου προσθέτει έναν όρο «τιμωρίας» στα βάρη για να περιορίσει το overfitting το οποίο αποτέλεσε μεγάλο πρόβλημα στα προηγούμενα Assignments. Έχουμε:

$$W = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T T$$

όπου

$\Phi(\phi)$ =RBF πίνακας σχεδιασμού (έξοδος του κρυφού στρώματος)

$\lambda(\lambda)$ =Παράμετρος Ρύθμισης (Ελεγχος της «τιμωρίας» στα βάρη)

I=Μοναδιαίος Πίνακας

T=Διάνυσμα Στόχων (επιθυμητές τιμές εξόδου για εκπαίδευση)

W=Διάνυσμα Βαρών

Experimentation

1st Experiment, change of sigma.method(classwise vs global)

Global=Το σ δεν έχει range αλλά είναι μια raw τιμή όπου «μοιράζονται» όλοι οι RBF νευρώνες

Με την αλλαγή της μεθόδου παρατηρούμε ότι η απόδοση του RBF μας πέφτει από ~47% σε ~41% (test accuracy) κάτι λογικό καθώς με την προηγούμενη μέθοδο είχαμε καλύτερο classification διότι το σ προσαρμοζόταν στα κέντρα με ευελιξία ενός range τιμών. Παρατηρούμε επίσης ότι αυτή η μικρή μείωση ισχύει και στο training/validation accuracy

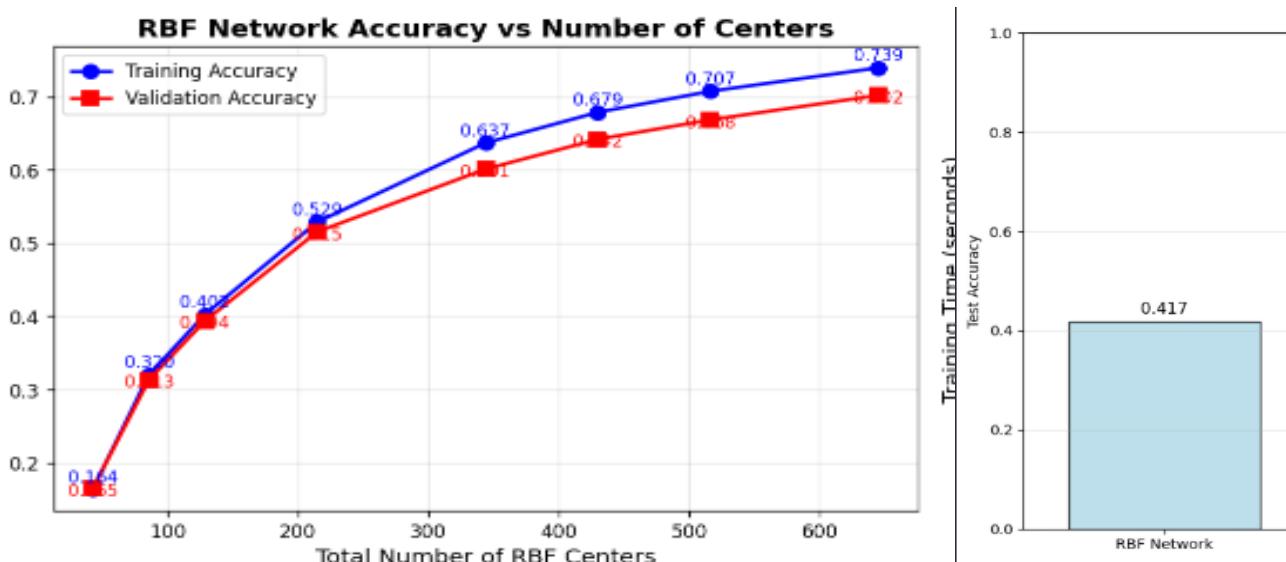
2nd Experiment, Increase and Decrease of Lambda

Με την αύξηση του λ ($\lambda = 1.0$) παρατηρούμε μείωση του train/validation accuracy, καθώς και πτώση του test accuracy, γεγονός που είναι αναμενόμενο, διότι η αύξηση της παραμέτρου αυτής ενισχύει την κανονικοποίηση και αυξάνει την «τιμωρία» που επιβάλλεται στα βάρη του μοντέλου. Ως αποτέλεσμα, τα βάρη τείνουν να λαμβάνουν μικρότερες τιμές, περιορίζοντας την ικανότητα του RBF δικτύου να προσαρμόζεται στα δεδομένα εκπαίδευσης, γεγονός που οδηγεί σε underfitting και μειωμένη γενίκευση. (bonus: παρατήρησα και αυξημένο χρόνο training)

Με μικρή μείωση του λ ($\lambda=1e-8$) ΔΕΝ παρατηρείται κάποια ειδοποιός διαφορά στην επίδοση του μοντέλου μας για αυτό ίσως θα πρέπει να δοκιμάσουμε ακόμα πιο extreme τιμές.

Για $\lambda=0$ επίσης δεν παρατηρούμε διαφορά στα στατιστικά του μοντέλου. Γιατί όμως? Αυτό μπορεί να οφείλεται στο στο οποίο στην μέθοδο classwise είναι ευέλικτο και έχει ένα range τιμών με συνήθως μια τιμή να είναι αρκετά μεγάλη. Έτσι το regularization ΔΕΝ έχει τίποτα να τιμωρήσει.

Όμως όταν αλλάζουμε το sigma.method σε global όπου έχουμε fixed τιμές σ που συνήθως είναι μικρές, έτσι οντως βλέπουμε αλλαγές στα στατιστικά με μείωση στο train/validation accuracy καθώς και στο testing accuracy



Visual results for sigma.method=global and $\lambda=0$

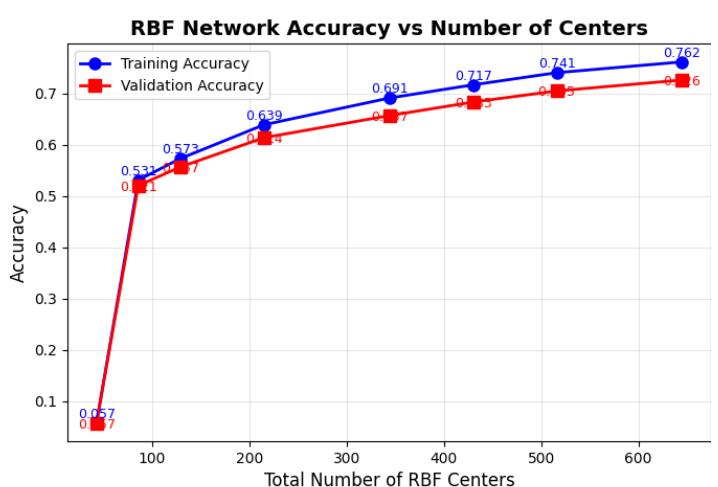
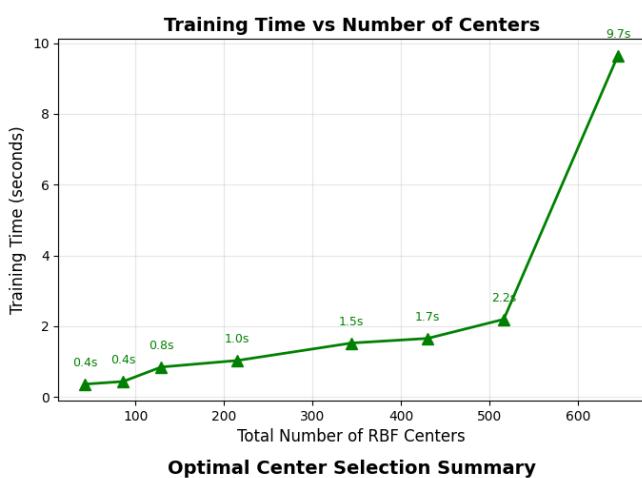
3rd Experiment, Different amount of centers PER class:

sigma.method="classwise", lambda=1e-6

Τρέχουμε το πρόγραμμα για ένα range τιμών για κάθε κλάση, centers=[1,2,3,5,8,10,12,15]

(δυστυχώς ΔΕΝ μπόρεσα να φτάσω στα 20+ διότι λογικά δεν είχα αρκετή RAM)

Από το σχήμα παρακάτω παρατηρούμε ότι για μικρό αριθμό κέντρων για κάθε κλάση έχουμε χαμηλό validation/training accuracy το οποίο είναι και ίδιο κάτι που ΔΕΝ θέλουμε. Θέλουμε ΠΑΝΤΑ να ισχύει Training Acc > Validation Acc. Επίσης σε χαμηλούς αριθμούς κέντρων σημειώνεται μεγαλύτερο εύρος τιμών σ (εκτός του center=1 όπου σ=[1.0,1.0] λόγω του TrainAcc=ValAcc) αυτό προκύπτει διότι ο κάθε RBF νευρώνας αναγκάζεται να καλύψει μεγαλύτερο τμήμα του χώρου χαρακτηριστικών καθώς οι αποστάσεις μεταξύ κάθε κέντρου της ίδιας κλάσεις είναι μεγαλύτερες. Με την αύξηση των κέντρων ανά κλάση μειώνονται οι αποστάσεις μεταξύ των νευρώνων με αποτέλεσμα το σ να παίρνει μικρότερες τιμές αλλά και να γίνεται πιο ακριβής classification κάτι που οδηγεί στο Train/Validation/Test Accuracy να αυξηθούν. Όσο αφορά τον χρόνο εκπαίδευσης είναι λογικό ότι με την αύξηση των κέντρων θα υπάρχει και αύξηση του χρόνου εκπαίδευσης κατά μερικά δευτερόλεπτα.

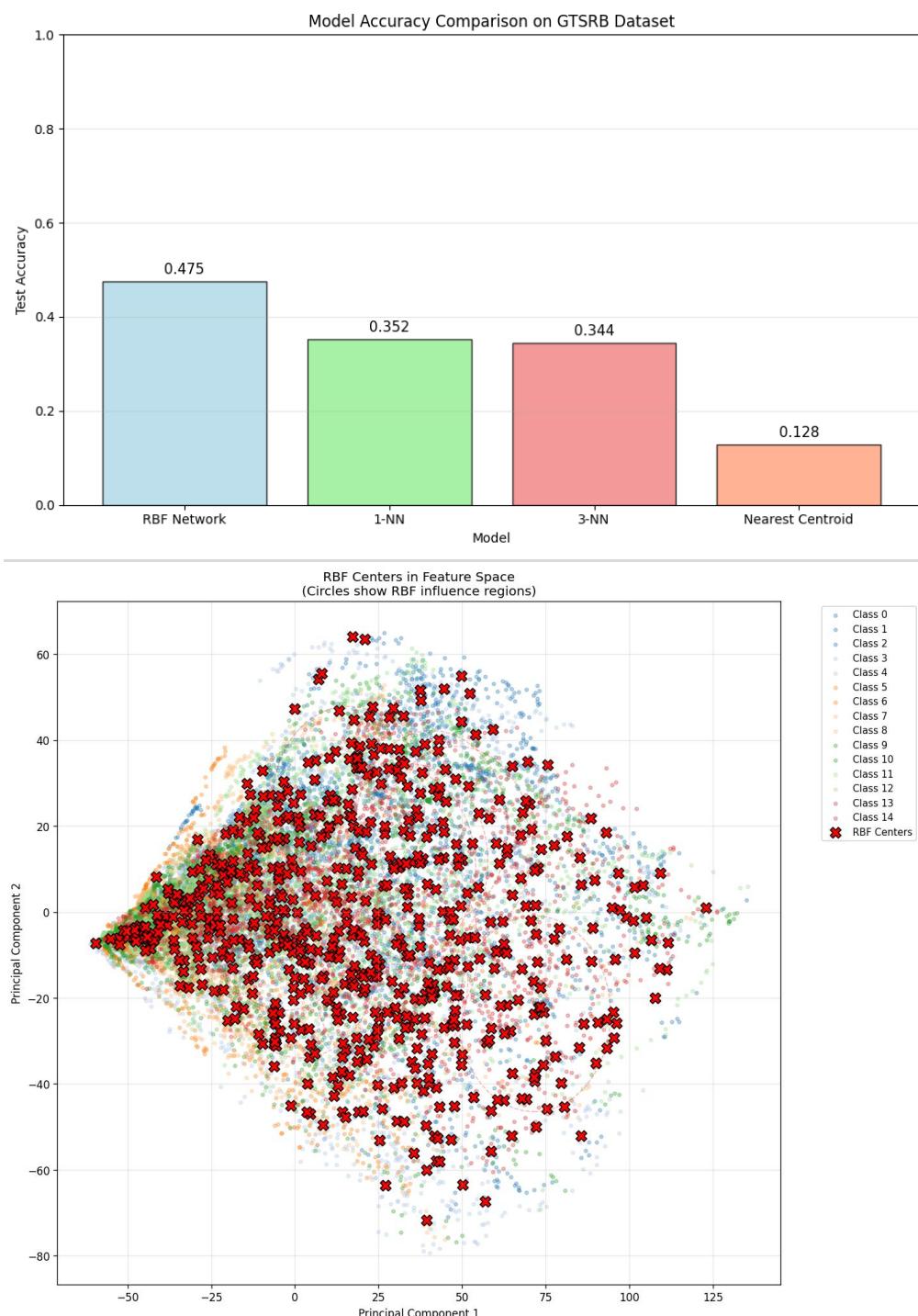


4th Experiment, change of k-Means value

Σε όλα τα παραπάνω πειράματα η τιμή init=3, με την αύξηση και μείωση του δεν παρατηρούμε κάποια διαφορά στα αποτελέσματα μονάχα στο Train/Validation Accuracy όπου με την μείωση, μειώνεται κατά ~40s ενώ με την αύξηση του αυξάνεται κατά ~2s. Το ότι δεν έχουμε κάποια άλλη διαφορά μπορεί να οφείλεται στο πολύ καλά ορισμένο πρόβλημα μας ή και ακόμα στην εφαρμογή PCA τακτικής.

Τελικές Παρατηρήσεις

Το RBF μας σε σύγκριση με τις υπόλοιπες τακτικές 1NN, 3NN, NCC είναι λογικό να έχει καλύτερες αποδόσεις με την εφαρμογή PCA για μείωση του χώρου χαρακτηριστικών. Συγκεκριμένα για την σύγκριση πήραμε το RBF με 15 κέντρα για κάθε κλάση, με sigma.method=classwise και lambda=1e-6 καθώς είχε την καλύτερη απόδοση στο Train/Validation. Ακολουθούν εικόνες



Examples of wrong and right classification:

RBF Network Prediction Examples
(Green=Correct, Red=Incorrect)



Τέλος του Report

Καλή Χρονία και καλή σας μέρα 😊