



# Νευρωνικά Δίκτυα

---

- **Ονοματεπώνυμο** : Απόστολος Γιαννουλίδης
- **ΑΕΜ** : 2906
- **Βάση Δεδομένων** : MNIST (χειρόγραφα ψηφία)

# Ενδιάμεση Εργασία

- **κατηγοριοποιητής πλησιέστερου γείτονα με 3 γείτονες :**

Ακρίβεια στα δεδομένα εκπαίδευσης: 0.9755

Ακρίβεια στα δεδομένα δοκιμής : 0.9235

Χρόνος : 222.16224620000003

- **κατηγοριοποιητής πλησιέστερου γείτονα με 1 γείτονα :**

Ακρίβεια στα δεδομένα εκπαίδευσης: 1.0

Ακρίβεια στα δεδομένα δοκιμής : 0.9225

Χρόνος :142.87078640000001

- **κατηγοριοποιητής πλησιέστερου κέντρου:**

Ακρίβεια στα δεδομένα εκπαίδευσης: 0.8093

Ακρίβεια στα δεδομένα δοκιμής : 0.76

Χρόνος : 0.14205130000000483

# 1<sup>η</sup> Εργασία

## Αναγνώριση ψηφίων με Νευρωνικό Δίκτυο

**Είσοδος :** διάνυσμα με 784 στοιχεία ( [0-1])

**Κρυφό Επίπεδο :** 10 νευρώνες (Dense) με χρήση bias και συνάρτηση ενεργοποίησης relu.

**Επίπεδο Εξόδου:** 10 νευρώνες , συνάρτηση ενεργοποίησης softmax.

**Τρόπος Εκπαίδευσης :** stochastic gradient descent

# Αλλαγές στο momentum και επιδώσεις

- **Momentum ίσο με 0:**

Σφάλμα και ακρίβεια μετά από τις 10 εποχές :Σφάλμα 0.9256 , ακρίβεια: 0.8205

- **Momentum ίσο με 0.5 :**

Σφάλμα και ακρίβεια μετά από τις 10 εποχές :Σφάλμα : 0.6065 , ακρίβεια: 0.8636

- **Momentum ίσο με 1:**

Σφάλμα και ακρίβεια μετά από τις 10 εποχές :Σφάλμα : 0.7412 , ακρίβεια: 0.9375

- **Momentum ίσο με 2 :**

Σφάλμα και ακρίβεια μετά από τις 10 εποχές :Σφάλμα : nan , ακρίβεια: 0.0980

## Τελικό momentum ίσο με 0.95 :

Σφάλμα και ακρίβεια μετά από τις 50 εποχές :Σφάλμα : 0.1363 , ακρίβεια: 0.9605

# Συμπέρασμα

Η επιλογή χρήσης momentum μπορεί να βοηθήσει σημαντικά στο πόσο γρήγορα και αποτελεσματικά θα συγκλίνει το νευρωνικό δίκτυο σε μια καλή κατάσταση , καθώς στο συγκεκριμένο παράδειγμα είχαμε βελτίωση στην ακρίβεια μέχρι και 10%.

**Σύγκριση δικτύου με κατηγοριοποιητή πλησιέστερου γείτονα με έναν και τρεις γείτονες :**

- **κατηγοριοποιητής πλησιέστερου γείτονα με 3 γείτονες :**

Train acc = 0.985366      Test acc = 0.9687      Time : 89.62

- **κατηγοριοποιητής πλησιέστερου γείτονα με 1 γείτονα :**

Train acc = 1.0      Test acc = 0.9638      Time : 13.796097

Τα αποτελέσματα μας δείχνουν πως οι δύο προσεγγίσεις είναι ισοδύναμες όσον αφορά την ακρίβεια , με το νευρωνικό δίκτυο να είναι πιο αργό στον χρόνο εκπαίδευσης .

Παρόλα αυτά εφόσον το δίκτυο έχει εκπαιδευτεί η διαδικασία της αναγνώρισης είναι πολύ πιο γρήγορη (  $O(1)$  ) από τους δύο κατηγοριοποιητές.

# 2<sup>η</sup> Εργασία Χρήση SVM

## Αναγνώριση άρτιων και περιττών αριθμών

- **Επιλογή C και Gamma (για rbf) :**

Η αναζήτηση των τιμών για τις μεταβλητές αυτές έγινε σε ένα τυχαίο υποσύνολο από το σύνολο των δεδομένων που διαθέτουμε και εξετάστηκαν όλοι οι συνδυασμοί τιμών για τα C ίσο με (0.01,0.1,1,10) και gamma ίσο με (0.001,0.01,0.1,1,10).

Για τις τιμές C ίσο με 10 και Gamma ίσο με 0.01 έχουμε την καλύτερη απόδοση .

- **Επιλογή Πύρινα :**

|                 |                           |                            |                        |
|-----------------|---------------------------|----------------------------|------------------------|
| <b>linear :</b> | Test acc: 0.881000        | Train acc: 0.892200        | Time : 12.828125       |
| <b>poly :</b>   | Test acc: 0.978000        | Train acc: 0.995200        | Time : 2.015625        |
| <b>rbf :</b>    | Test acc: <b>0.991000</b> | Train acc: <b>0.998400</b> | Time : <b>1.390625</b> |

- **Εφαρμογή του rbf πυρήνα σε όλο το σύνολο των δεδομένων:**

|                    |                     |                   |
|--------------------|---------------------|-------------------|
| Test acc: 0.990600 | Train acc: 0.996817 | Time : 196.265625 |
|--------------------|---------------------|-------------------|

# Σύγκριση με άλλες μεθόδους

- **SVM :**

Train acc: 0.996817      Test acc: 0.990600      Time : 193.140625

- **KNeighborsClassifier k=3:**

Train acc = 0.99375      Test acc k=3 = 0.9867      Time : 470.515625

- **KNeighborsClassifier k=1:**

Train acc = 1.0      Test acc k=1 = 0.9855      Time : 63.359375

- **Nearest Centroid:**

Train acc = 0.808      Test acc = 0.8018      Time : 0.09375

## Συμπέρασμα :

Η προσέγγιση του svm και των πλησιέστερων γειτόνων έχουν την καλύτερη ακρίβεια .

## 3<sup>η</sup> Εργασία

# Αναγνώριση άρτιων και περιττών αριθμών με Rbf δίκτυο

- **Δημιουργία Rbf :**

Υλοποίηση κλάσης για την αναπαράσταση νευρωνικού Rbf .

- **Επιλογή κ:**

Δοκιμή για διαφορετικό αριθμό κέντρων ( 2, 5 , 10 ) με τελική επιλογή  $k=5$ .

- **Επιλογή βήμα εκπαίδευσης:**

Δοκιμές για βήμα εκπαίδευσης ίσο με 0.1 και 0.01

- **Επιλογή κέντρων:**

Επιλογή κέντρων με κ-μέσους ή με τυχαία επιλογή .



# Σύγκριση με άλλες μεθόδους

- **RBF :**

Train : 0.7908833      Testing results : 0.7866      Time : 128.828125

- **KNN3:**

Train acc = 0.993933      Test acc k=3 = 0.9865      Time : 211.21875

- **KNN1:**

Train acc = 1.0      Test acc k=1 = 0.9857      Time : 27.671875

- **NearestCentroid:**

Train acc = 0.807466      Test acc = 0.8018      Time : 0.09375

**Συμπέρασμα:** Η χρήση rbf δεν δίνει ικανοποιητικά αποτελέσματα ( το συγκεκριμένο δίκτυο πάντα ) για την λύση του προβλήματος .

# Γενικά Συμπεράσματα

- **Ποιο από τα παραπάνω μοντέλα θα χρησιμοποιούσαμε για την επίλυση κάποιου προβλήματος κατηγοροποίησης;**

Εξαρτάται από την φύση του προβλήματος , τα δεδομένα και την ίδια την εφαρμογή .

- **Παράμετροι :**

Η επιλογή των παραμέτρων εκπαίδευσης καθορίζουν και την αποτελεσματικότητα της μεθόδου κάθε φορά .

- **Δεδομένα :**

Ο όγκος δεδομένα και ο τύπος των δεδομένων αποτελούν περιορισμό για τις διαθέσιμες μεθόδους που επιλύουν το πρόβλημα .

Παράλληλα η σωστή μορφή των δεδομένων ( ανάλογα το εργαλείο που χρησιμοποιείται κάθε φορά ) είναι πολύ σημαντική για τον καθορισμό της απόδοσης της εκατοστέ μεθόδου.