Σύγκριση Extreme Learning Machine και Νευρωνικού Δικτύου Στο FisherIris Benchmark

Κωνσταντίνος Καλότσι, Καπαγερίδης Ιωάννης

Δημοκρίτειο Πανεπιστήμιο Θράκης

Οκτώβριος 2025

Abstract

Στην παρούσα εργασία πραγματοποιείται συγκριτική ανάλυση μεταξύ ενός νευρωνικού δικτύου (MLP) και ενός δικτύου Extreme Learning Machine (ELM) στο κλασικό σύνολο δεδομένων Fisher Iris.Το σύνολο δεδομένων αποτελείται από 150 δείγματα τριών ειδών ίριδας (Iris setosa, Iris versicolor Iris virginica),με τέσσερις μετρήσεις χαρακτηριστικών για κάθε δείγμα. Η αξιολόγηση των δύο αρχιτεκτονικών πραγματοποιείται για διάφορο αριθμό νευρώνων, χρησιμοποιώντας την τεχνική k-fold cross validation για την εξασφάλιση αξιόπιστων αποτελεσμάτων. Συγκρίνονται οι επιδόσεις και η γενίκευση των δύο μοντέλων, με στόχο την αναγνώριση των πλεονεκτημάτων και μειονεκτημάτων κάθε προσέγγισης στο συγκεκριμένο πρόβλημα ταξινόμησης.

1 Εισαγωγή

Το σύνολο δεδομένων Φισηερ Ιρις αποτελεί ένα από τα πιο γνωστά και ευρέως χρησιμοποιούμενα σύνολα δεδομένων στον τομέα της μηχανικής μάθησης και της αναγνώρισης προτύπων [1]. Παρουσιάστηκε για πρώτη φορά από τον Βρετανό στατιστικολόγο Ronald Fisher το 1936 στην κλασική του εργασία “The use of multiple measurements in taxonomic problems”, όπου χρησιμοποιήθηκε ως παράδειγμα για την ανάλυση γραμμικού διαχωρισμού (linear discriminant analysis) [1].

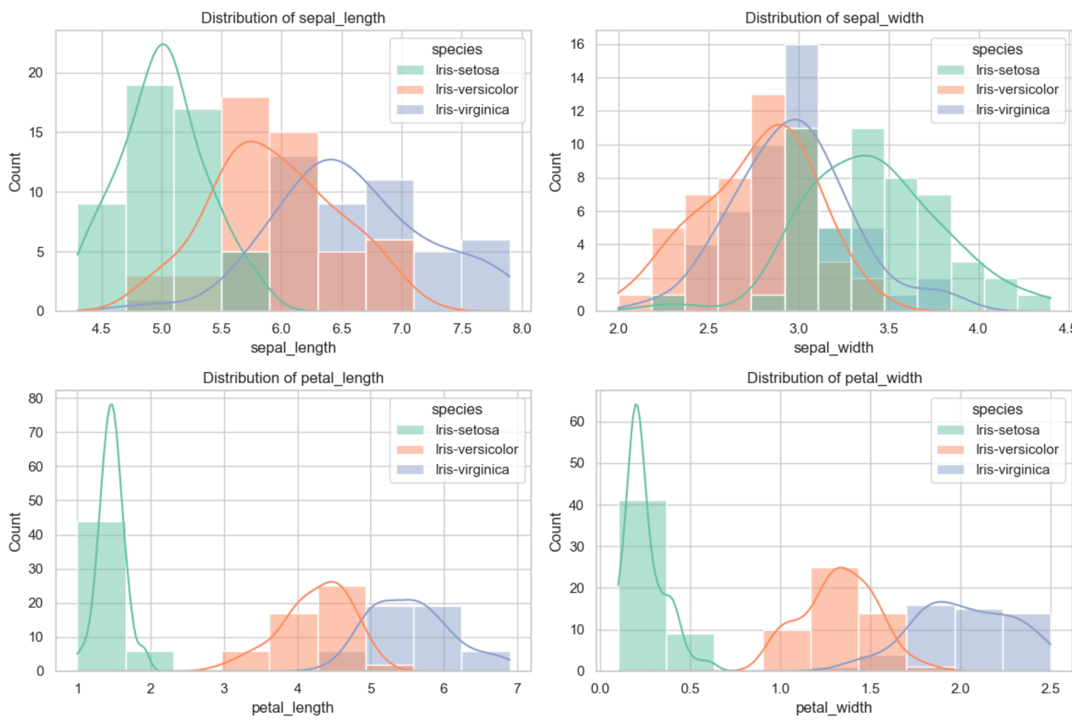
Τα δεδομένα συλλέχθηκαν από τον Edgar Anderson και περιλαμβάνουν μετρήσεις από 150 δείγματα λουλουδιών ίριδας, που ανήκουν σε τρία διαφορετικά είδη: Iris setosa, Iris versicolor και Iris virginica [2].Για κάθε δείγμα καταγράφονται τέσσερα χαρακτηριστικά: το μήκος και το πλάτος του σέπαλου (σεπαλ) και το μήκος και το πλάτος του πετάλου (πεταλ), μετρημένα σε εκατοστά. Κάθε είδος αντιπροσωπεύεται από 50 δείγματα.

Η σημασία του συνόλου δεδομένων Φισηερ Ιρις ως benchmark έγκειται σε διάφορους παράγοντες. Πρώτον, αποτελεί ένα σχετικά απλό πρόβλημα πολυκλασικής ταξινόμησης με τέσσερα χαρακτηριστικά και τρεις κλάσεις, το οποίο είναι ιδανικό για την αρχική αξιολόγηση και σύγκριση αλγορίθμων ταξινόμησης. Δεύτερον, παρουσιάζει ενδιαφέρουσα δομή: το είδος Iris setosa είναι γραμμικά διαχωρίσιμο από τα άλλα δύο είδη, ενώ τα Iris versicolor και Iris virginica παρουσιάζουν κάποια επικάλυψη, γεγονός που καθιστά το πρόβλημα μη τετριμμένο [3]. Τρίτον, το σύνολο δεδομένων είναι πλήρες χωρίς ελλείπουσες τιμές, γεγονός που το καθιστά ιδανικό για εκπαιδευτικούς και συγκριτικούς σκοπούς.

Λόγω της ιστορικής του σημασίας και των ιδιοτήτων του, το Φισηερ Ιρις δατασετ έχει καθιερωθεί ως πρότυπο σύνολο δεδομένων για την αξιολόγηση τεχνικών μηχανικής μάθησης και στατιστικής ταξινόμησης (για την αξιολόγιση απλούστερων μοντέλων φυσικά), και συμπεριλαμβάνεται σε πολλές βιβλιοθήκες όπως το scikit-learn της Python και το βασικό πακέτο της R [3] . Στην παρούσα εργασία, χρησιμοποιείται για τη συγκριτική αξιολόγηση δύο διαφορετικών νευρωνικών αρχιτεκτονικών, όπου συγκρίνουμε τα μοντέλα Extreme Learning Machine (ELM) και Multi-Layer Perceptron για ένα πλήθος διαφορετικών στρώσεων - παραμέτρων αντίστοιχα.

2 Ανάλυση Δεδομένων

Καθώς τα δεδομένα έχουν μόνο 4 διαστάσεις - features μπορούμε εύκολα να τα οπτικοποίησουμε και να δούμε τις ανάλογες συσχετίσεις. Αρχικά, βλέπουμε τις κατανομές των δεδομένων στο Σχήμα 1 ως ιστόγραμμα και έπειτα μπορούμε να εξάγουμε τις αντίστοιχες κατανομές. Εδώ είναι ξεκάθαρο πως τα δεδομένα είναι εύκολα διαχωρίσημα όσων αναφορά τα χαρακτηριστικά *petal_length*, *pedal_width* αλλά είναι δυσκολότερο να διακριθούν, κατά μέσο όρο στις άλλες δύο κατηγορίες.



Σχήμα 1: Διάγραμμα κατανομών ανά κατηγορία (ιστόγραμμα και κατανομές).

Μπορούμε επίσης να δούμε τα δεδομένα από μια άλλη σκοπία μέσω ενός διαγράμματος *pairplot* στο Σχήμα 2. Εδώ στα διαγράμματα που βρίσκονται στην κύρια διαγώνιο έχουμε τις κατανομές όπως στο Σχήμα 1, ωστόσο στα υπόλοιπα υπο-διαγράμματα βλέπουμε τα χαρακτηριστικά ως *Scatter plot* , δηλαδή ως σημεία με το χρώμα να αναπαριστά την αντίστοιχη κλάση. Ξανά, είναι ξεκάθαρο πως τα πράσινα σημεία (κλάσης Iris Setosa) είναι διαχωρίσημα από τις άλλες δύο κλάσεις οι οποίες δεν είναι εύκολα ευδιάκριτες, σχεδόν για όλα τα ζευγάρια χαρακτηριστικών.

Σχήμα 2: *Pairplot* των ζευγαρίων ανά κατηγορία.

3 Εκπαίδευση Μοντέλων

Όπως αναφέρθηκε παραπάνω, το συγκεκριμένο σύνολο δεδομένων είναι αρκετά απλό με μόνο 4 χαρακτηριστικά ανά εντρψ και αποτελείται από μόνο 150 εντριες, 50 για κάθε κλάση.
Εδώ είναι σημαντικό να αναφερθεί πως τα δεδομένα είναι πλήρη, δηλαδή δεν έχουμε κενές τιμές σε κάποια κελιά - χαρακτηριστικά, συνεπώς δεν χρειάζεται να ¨γεμίσουμε’ τεχνητά κάποιες τιμές. Όσον αναφορά την διαχείρηση των δεδομένων, απλά κανονικοποιήσαμε τα δεδομένα με βάση την (τυπική) κανονική κατανομή.

Καθώς αναφερόμαστε σε ένα απλό πρόβλημα, ο σκοπός της παρούσας εργασίας είναι όχι μόνο να συγκρίνει τα δύο προαναφερθέντα μοντέλα, αλλά και να συγκρίνει διάφορες παραμέτρους για το κάθε μοντέλο. Συγκεκριμένα, για το MLP ξεκινάμε με μόνο 1 νευρώνα (φυσικά δεν μετράμε τους 4 εισόδου και 3 εξόδου) και προσθέτουμε έναν νευρώνα και εκπαιδεύομε ξεχωριστά, μέχρι να φτάσουμε τους 5 ανα στρώση, για 5 συνολικές στρώσεις. Με άλλα λόγια, εκπαιδεύουμε 25 ξεχωριστά μοντέλα με 1 έως 25 κρυφούς νευρώνες.

Λόγω του μικρόυ μεγέθους του συνόλου δεδομένων και της δυσκολίας στην ερμηνέια των κλάσεων όπως φαίνεται στα Σχήματα 1, 2 είναι λογικό πως εάν ακολουθειθεί μια κλασσική προσσέγγιση διαχωρισμού των δεδομένων, όπως 80/20 τότε υπάρχει καλή πιθανότητα να έχουμε ¨ελλατωματικά’ δεδομένα στο test κομμάτι και το μοντέλο να δίνει εσφαλμένα accuracies.Για να αντιμετωπίσουμε αυτό το πρόβλημα, και για τα δύο μοντέλα όσον αναφορά το train-test split χρησιμοποιούμε K-Fold Cross Validation , δηλαδή τα μοντέλα θα δούν, **ξεχωριστά και ανεξάρτητα**, όλα τα δεδομένα και θα μας δώσουν το μέσο λάθος των προβλέψεων ως τελικό λάθος. Στον αντίποδα φυσικά, κάθε μοντέλο θα πρέπει να εκπαιδευτέι 5 φόρές αντί για 1 και συνεπώς ο συνολικός χρόνος είναι ελαφρώς μεγαλύτερος.

Φυσικά, χρησιμοποιούμε την τεχνική one-hot encoding για να μετατρέψουμε τις κλάσσεις μας σε διανύσματα στον R^3.

Όλα τα πειράματα πραγματοποιήθηκαν σε υπολογιστή με επεξεργαστή Intel® Core™ i5-10400 CPU @ 2.90 GHz, ο οποίος διαθέτει 6 φυσικούς πυρήνες και 12 λογικά νήματα. Ο επεξεργαστής έχει 384 KB L1 cache, 1,5 MB L2 cache και 12 MB L3 cache.Το σύστημα ήταν εξοπλισμένο με 24 GB RAM και μια κάρτα γραφικών NVIDIA GeForce (η οποία δεν χρησιμοποιήθηκε στην εκπαίδευση με CPU).

3.1 Multi-Layer Perceptron

Για την προσέγγιση του κλασσικού νευρωνικού θα χρησιμοποιήσουμε $learning\_rate = 0.01$, καθώς θα εκπαιδεύσουμε για 500 εποχές. Πειραματικά είδαμε πως περισσότερες εποχές δεν αντιστοιχούν σε καλύτερη ακρίβεια και οι 500 είναι ήδη αρκετές. Ως αλγόριθμος οπισθοδρόμησης χρησιμοποιείται ο **Stochastic Gradient Descent**. Ως συνάρτηση κόστους χρησιμοποιούμε την Μέση Τετραγωνική διαφορά (ή αλλιώς **MSE**) . Χρησιμοποιούμε την σιγμοιδή συνάρτηση

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

ως συνάρτηση ενεργοποίησης για όλες τις στρώσεις, εκτός από την τελευταία. Στην τελευταία στρώση (εξόδου) χρησιμοποιούμε την συνάρτηση **Softmax** η οποία ορίζεται ώς εξής

$$\text{Σοφτμαξ}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad i = 1, \ldots, K$$

και μετατρέπει τις εξόδους σε μια **κατανομή πιθανότητας**. Αυτό μας επιτρέπει εάν θέλουμε να ερμηνεύσουμε την έξοδο του μοντέλου ως μια πιθανότητα ή αβεβαιότητα απόφασης.
Εκπαιδεύοντας τα 25 προαναφερθέντα μοντέλα έχουμε τα εξής δεδομένα:

Αρχιτεκτονική Μοντέλου	Λάθος Εκπαίδευσης	Λάθος Επαλήθευσης
1	0.224	0.223
2x1	0.224	0.224
3x1	0.227	0.228
4x1	0.230	0.232
5x1	0.224	0.224
2	0.216	0.216
2x2	0.223	0.224
3x2	0.225	0.226
4x2	0.228	0.231
5x2	0.225	0.224
3	0.208	0.209
2x3	0.226	0.226
3x3	0.223	0.225
4x3	0.224	0.227
5x3	0.223	0.224
4	0.204	0.208
2x4	0.222	0.223
3x4	0.222	0.224
4x4	0.224	0.224
5x4	0.222	0.223
5	0.201	0.204
2x5	0.224	0.225
3x5	0.222	0.223
4x5	0.225	0.229
5x5	0.223	0.223

Πίνακας 1: Τιμές μέσου τετραγωνικού λάθους ανά μοντέλο. Αριστερά αναγράφεται το πλήθος των κρυφών στρώσεων επί το πλήθος νευρώνων ανά στρώση.

Εδώ αξίζει να αναφερθούμε στο γεγονός πως παρόλη την πολυπλοκότητα των μοντέλων (όσο πηγαίνουμε προς τα κάτω στον Πίνακα 1, βλέπουμε πως τα λάθη εκπαίδευσης και επαλύθευσης είναι σχεδόν ίδια, ενώ θα περιμέναμε τουλάχιστον το λάθος εκπαίδευσης να μικραίνει. Αυτό είναι διότι όπως αναφέραμε τα δεδομένα είναι δύσκολο να διαχωριστούν και συνεπώς το λάθος παραμένει σταθερό στο 0.22. Η εκπαίδευση για τα 25 μοντέλα πήρε συνολικά **42.85** δευτερόλεπτα.

3.2 Extreme Learning Machine

Για την προσέγγιση του Extreme Learning Machine ξεκινάμε από έναν κρυφό νευρώνα και σταματάμε στους 25. Ξανά χρησιμοποιούμε σιγμοειδή συνάρτηση ενεργοποίησης από την στρώση εισόδου προς την κρυμμένη, και την softmax από την κρυμμένη προς την στρώση εξόδου. Τα δεδομένα είναι:

Πλήθος κρυφών νευρώνων	Λάθος Επαλύθευσης
1	0.206
2	0.193
3	0.158
4	0.148
5	0.146
6	0.137
7	0.120
8	0.127
9	0.118
10	0.124
11	0.114
12	0.117
13	0.119
14	0.106
15	0.111
16	0.111
17	0.107
18	0.108
19	0.105
20	0.103
21	0.105
22	0.104
23	0.107
24	0.103
25	0.104

Πίνακας 2: Λάθος επαλύθευσης ανά πλήθος κρυφών νευρώνων στο ELM.

με συνολικό χρόνο ̈εκπαίδευσης' μόλις 0.15 δευτερόλεπτα.

4 Τελική Σύγκριση Μοντέλων

Αφότου εκπαιδεύσαμε τα δύο μοντέλα με διάφορες παραμέτρους μπορούμε συνοπτικά σε έναν πίνακα να παρουσιάσουμε τα αποτελέσματα στον Πίνακα 3.

Αρχιτεκτονική Μοντέλου MLP	Λάθος Εκπαίδευσης MLP	Λάθος Επαλήθευσης MLP	Πλήθος κρυφών νευρώνων ELM	Λάθος Επαλήθευσης ELM
1	0.224	0.224	1	**0.206**
2x1	0.224	0.224	2	**0.193**
3x1	0.227	0.228	3	**0.158**
4x1	0.230	0.232	4	**0.148**
5x1	0.224	0.224	5	**0.146**
2	0.216	0.216	6	**0.137**
2x2	0.223	0.224	7	**0.120**
3x2	0.225	0.226	8	**0.127**
4x2	0.228	0.231	9	**0.118**
5x2	0.225	0.224	10	**0.124**
3	0.208	0.209	11	**0.114**
2x3	0.226	0.226	12	**0.117**
3x3	0.223	0.225	13	**0.119**
4x3	0.224	0.227	14	**0.106**
5x3	0.223	0.224	15	**0.111**
4	0.204	0.208	16	**0.111**
2x4	0.222	0.223	17	**0.107**
3x4	0.222	0.224	18	**0.108**
4x4	0.224	0.224	19	**0.105**
5x4	0.222	0.223	20	**0.103**
5	0.201	0.204	21	**0.105**
2x5	0.224	0.225	22	**0.104**
3x5	0.222	0.223	23	**0.107**
4x5	0.225	0.229	24	**0.103**
5x5	0.223	0.223	25	**0.104**

Πίνακας 3: Τιμές μέσου τετραγωνικού λάθους ανά αρχιτεκτονική στο MLP (αριστερά) και λάθος επαλύθευσης ανά πλήθος κρυφών νευρώνων στο ELM (δεξιά).

Από τα αποτελέσματα στον Πίνακα 3 παρατηρούμε ότι το μέσο τετραγωνικό λάθος (MSE) για το MLP παραμένει σχετικά σταθερό καθώς αλλάζει η αρχιτεκτονική του δικτύου, με μικρές διακυμάνσεις ανάμεσα σε διαφορετικούς συνδυασμούς στρώσεων και νευρώνων. Αντίθετα, το ELM παρουσιάζει σταδιακή μείωση του λάθους καθώς αυξάνεται ο αριθμός των κρυφών νευρώνων, υποδεικνύοντας ότι η απόδοσή του βελτιώνεται με μεγαλύτερη χωρητικότητα. Συνολικά, και οι δύο μέθοδοι επιτυγχάνουν χαμηλά σφάλματα επαλήθευσης στο σύνολο δεδομένων Φισηερ Ιρις, με το ΕΛΜ να εμφανίζει ελαφρώς καλύτερη προσαρμοστικότητα σε υψηλότερους αριθμούς νευρώνων.

Ωστόσο όπως αναφέραμε, για την προσέγγιση του MLP ακολουθούμε ¨παραδοσιακές' μεθόδους (MSE, Stochastic Gradient Descent) και για αυτό το δεδομένο αδυνατεί να κάνει υπερπροσαρμογή εύκολα. Παρόλα αυτά, τα δεδομένα παραμένουν ελλατωματικά και αδιάκριτα, ακόμα και για το ELM. Τέλος, το ELM κατάφερει να εκπαιδευτεί σε μόλις 0.15 δευτερόλεπτα, δηλαδή περίπου 300 φορές γρηγορότερα από το MLP!

Αναφορές

Ο κώδικας είναι υλοποιημένος σε Python 3.11 σε υπολογιστή με χαρακτηριστικά που αναφέρθηκαν παραπάνω. Για τον έλεγχο εκδώσεων χρησιμοποιήθηκε Github . Ο πλήρης κώδικας μπορεί να βρεθεί στον σύνδεσμο https://github.com/kw5t45/fisheriris-benchmark

[1] Fisher, R. A. (1936). *The use of multiple measurements in taxonomic problems*. Annals of Eugenics, 7(2), 179-188.

[2] Anderson, E. (1936). *The species problem in Iris*. Annals of the Missouri Botanical Garden, 23(3), 457-509.

[3] UCI Machine Learning Repository. *Iris Data Set*. Available at: https://archive.ics.uci.edu/ml/datasets/iris