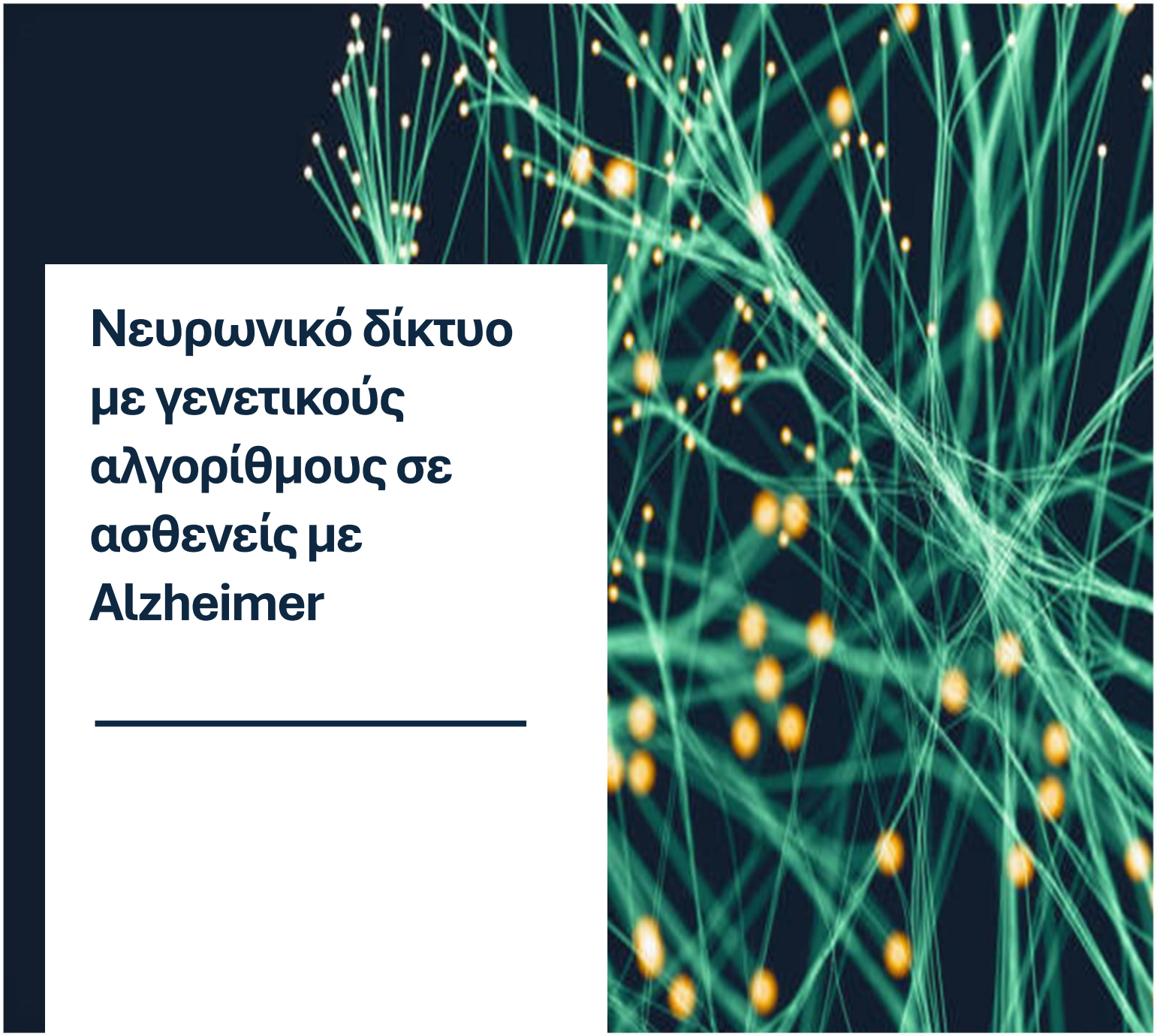


Νευρωνικό δίκτυο με γενετικούς αλγορίθμους σε ασθενείς με Alzheimer

Ζουμπουλάκης Ιωάννης

ΑΜ: 1093365
Φοιτητής 4ου έτους

Pythoncodelink:
https://github.com/Zoubou/Neural-Network

Β1. Σχεδιασμός Γενετικού Αλγορίθμου (ΓΑ)

α) Κωδικοποίηση

Για την κωδικοποίηση των ατόμων του πληθυσμού, επιλέχθηκε μια δυαδική αναπαράσταση, όπου κάθε άτομο είναι ένας δυαδικός διάνυσμα μήκους 34. Κάθε θέση στο διάνυσμα αντιστοιχεί σε μία από τις 34 δυνατές εισόδους του νευρωνικού δικτύου.

- Η τιμή 1 στη θέση i σημαίνει ότι το αντίστοιχο χαρακτηριστικό (είσοδος) είναι ενεργό και χρησιμοποιείται από το νευρωνικό δίκτυο.
- Η τιμή 0 σημαίνει ότι το χαρακτηριστικό είναι απενεργοποιημένο και δεν λαμβάνεται υπόψη ως είσοδος.

Με αυτή την κωδικοποίηση, κάθε άτομο αναπαριστά την επιλογή ενός υποσυνόλου εισόδων για το ίδιο αρχικό νευρωνικό δίκτυο. Η δομή και οι παράμετροι του δικτύου παραμένουν σταθερές, ενώ η είσοδος αλλάζει ανάλογα με το ποια χαρακτηριστικά ενεργοποιούνται.

β) Αρχικός πληθυσμός

Για τη δημιουργία του αρχικού πληθυσμού, χρησιμοποιήθηκε τυχαία γεννήτρια δυαδικών ατόμων, όπου κάθε άτομο είναι ένα διάνυσμα μήκους 34 με τιμές 0 ή 1.
Η διαδικασία έχει ως εξής:

- Για κάθε άτομο του πληθυσμού (π.χ. 34 άτομα), κάθε θέση στο διάνυσμα ανατίθεται τυχαία σε 0 ή 1, με ίση πιθανότητα (π.χ. 50%).
- Κάθε άτομο έτσι αναπαριστά έναν τυχαίο συνδυασμό ενεργοποιημένων και απενεργοποιημένων εισόδων, δηλαδή διαφορετικό υποσύνολο χαρακτηριστικών (features) για το νευρωνικό δίκτυο.
- Με αυτήν τη διαδικασία εξασφαλίζεται αρχική ποικιλία (diversity) στον πληθυσμό, απαραίτητη για αποτελεσματική αναζήτηση στο χώρο λύσεων.

Αυτή η τυχαία και ομοιόμορφη αρχικοποίηση επιτρέπει στον Γενετικό Αλγόριθμο να εξερευνήσει ευρύ φάσμα πιθανών συνδυασμών εισόδων από την αρχή.

```
population = np.random.randint(0, 2, (POP_SIZE, N_FEATURES))
```

γ) Συνάρτηση καταλληλότητας

Η συνάρτηση καταλληλότητας που χρησιμοποιήθηκε συνδυάζει δύο ανταγωνιστικά κριτήρια: την ακρίβεια ταξινόμησης και την πολυπλοκότητα του μοντέλου. Συγκεκριμένα, κάθε άτομο αξιολογείται ως προς την ακρίβεια του ήδη εκπαιδευμένου νευρωνικού δικτύου στο σύνολο ελέγχου, χρησιμοποιώντας μόνο τα χαρακτηριστικά που έχει επιλέξει. Παράλληλα, επιβάλλεται ποινή ανάλογη με τον αριθμό επιλεγμένων χαρακτηριστικών, ώστε να αποφεύγονται πολύπλοκα μοντέλα με περιττές εισόδους. Η τελική τιμή της καταλληλότητας υπολογίζεται ως: fitness = accuracy – 0.01 * num_features, με το '0.01' να μετατρέπει την τιμή της ποινής στην ίδια κλίμακα με την τιμή της ακρίβειας.

```
# === 5. Συνάρτηση αξιολόγησης ατόμου ===
def evaluate_individual(individual, model):
    masked_X_test = X_test * individual
    loss, accuracy = model.evaluate(masked_X_test, y_test, verbose=0)
    penalty = 0.01 * np.sum(individual)
    fitness = accuracy - penalty
    return fitness
```

δ) Γενετικοί Τελεστές

i. Για τη συγκεκριμένη κωδικοποίηση, όπου κάθε άτομο αναπαρίσταται ως δυαδικό διάνυσμα (0/1) μήκους ίσου με τον αριθμό χαρακτηριστικών, χρησιμοποιήθηκε τουρνουά επιλογή δύο ατόμων, καθώς είναι απλή και αποδοτική, διατηρώντας ισορροπία μεταξύ εκμετάλλευσης και εξερεύνησης του διαστήματος αναζήτησης. Η επιλογή με ρουλέτα (με βάση το κόστος ή την κατάταξη) απορρίφθηκε λόγω ευαισθησίας σε κλιμάκωση των τιμών της συνάρτησης καταλληλότητας και κινδύνου πρόωρης σύγκλισης.

```
# Tournament selection
selected = []
for _ in range(POP_SIZE):
    i1, i2 = np.random.choice(range(POP_SIZE), 2)
    winner = population[i1] if fitness_scores[i1] > fitness_scores[i2] else population[i2]
    selected.append(winner)
```

ii. Για τη διασταύρωση επιλέχθηκε η ομοιόμορφη διασταύρωση, καθώς είναι πιο κατάλληλη για δυαδικά χρωμοσώματα, επιτρέποντας τυχαία ανταλλαγή χαρακτηριστικών μεταξύ των γονέων και αυξάνοντας τη γενετική ποικιλία. Η μονού ή πολλαπλού σημείου διασταύρωση δεν αξιοποιεί πλήρως την ανεξαρτησία μεταξύ των χαρακτηριστικών.

```
# Crossover & Mutation
new_population = []
for i in range(0, POP_SIZE, 2):
    p1 = selected[i]
    p2 = selected[i + 1]

    if np.random.rand() < CROSSOVER_RATE:
        mask = np.random.randint(0, 2, size=N_FEATURES)
        child1 = np.where(mask == 1, p1, p2)
        child2 = np.where(mask == 1, p2, p1)
    else:
        child1 = np.copy(p1)
        child2 = np.copy(p2)
```

iii. Για τη μετάλλαξη εφαρμόστηκε χαμηλή πιθανότητα bitwise αναστροφής σε κάθε γονίδιο.

```
# Mutation
for child in [child1, child2]:
    for j in range(N_FEATURES):
        if np.random.rand() < MUT_RATE:
            child[j] = 1 - child[j]

    new_population.append(child)
```

Τέλος, παρότι η χρήση του ελιτισμού μπορεί να επιβραδύνει την πρόοδο του fitness σε κάποιες γενιές λόγω στασιμότητας, προσφέρει το σημαντικό πλεονέκτημα της διατήρησης των καλύτερων ατόμων μέσα στον πληθυσμό. Αυτό αποτρέπει την απώλεια των ήδη βελτιστοποιημένων λύσεων και εξασφαλίζει ότι η απόδοση δεν χειροτερεύει, ενώ τελικά οδηγεί σε καλύτερα συνολικά αποτελέσματα του αλγορίθμου. Συνεπώς, παρά τις αρχικές δυσκολίες στην ταχύτητα σύγκλισης(διπλάσιο χρόνο από το να μην είχα ελιτισμό), ο ελιτισμός διασφαλίζει σταθερότητα και βελτιωμένη απόδοση, γι' αυτό και αποφάσισα να τον υιοθετήσω.

```
# === Ελιτισμός ===
new_population = np.array(new_population)
new_fitness_scores = [evaluate_individual(ind, model) for ind in new_population]
worst_idx = np.argmin(new_fitness_scores)
new_population[worst_idx] = best_individual  # Replace worst with best from previous gen

population = new_population
```

B2. Υλοποίηση ΓΑ

Στην υλοποίηση του Γενετικού Αλγορίθμου (ΓΑ) για την επιλογή χαρακτηριστικών, ξεκίνησα από το Μέρος Α όπου εκπαίδευσα ένα νευρωνικό δίκτυο χρησιμοποιώντας το πλήρες σύνολο των δεδομένων εκπαίδευσης. Από εκεί προέκυψε το καλύτερο μοντέλο μαζί με τα αντίστοιχα σταθερά βάρη του, καθώς και ο scaler που εφαρμόζει την κατάλληλη κλίμακα στα χαρακτηριστικά. Αποθήκευσα αυτά τα αρχεία — το μοντέλο σε μορφή .keras και τον scaler σε μορφή .pkl — ώστε να μπορώ να τα φορτώνω στο Μέρος Β. Στο Μέρος Β, όπου υλοποιείται ο ΓΑ, τα φορτωμένα βάρη παραμένουν σταθερά και δεν επανεκπαιδεύονται, ενώ η αξιολόγηση κάθε ατόμου γίνεται πάνω στο σύνολο ελέγχου με χρήση του ήδη εκπαιδευμένου μοντέλου και του scaler. Αυτό διασφαλίζει σταθερότητα και ταχύτητα στην αξιολόγηση, καθώς αποφεύγεται η επανεκπαίδευση σε κάθε γενιά και επιτρέπει τη δίκαιη σύγκριση των διαφόρων επιλογών χαρακτηριστικών με βάση τα ίδια σταθερά δεδομένα και μοντέλο.

```
# === 2. Φόρτωση scaler από το Μέρος Α ===
scaler = joblib.load('/content/drive/MyDrive/ColabNotebooks/scaler.pkl')
X_scaled = scaler.transform(X)

# === 3. Φόρτωση μοντέλου από το Μέρος Α ===
model = load_model('/content/drive/MyDrive/ColabNotebooks/best_model.keras')
```

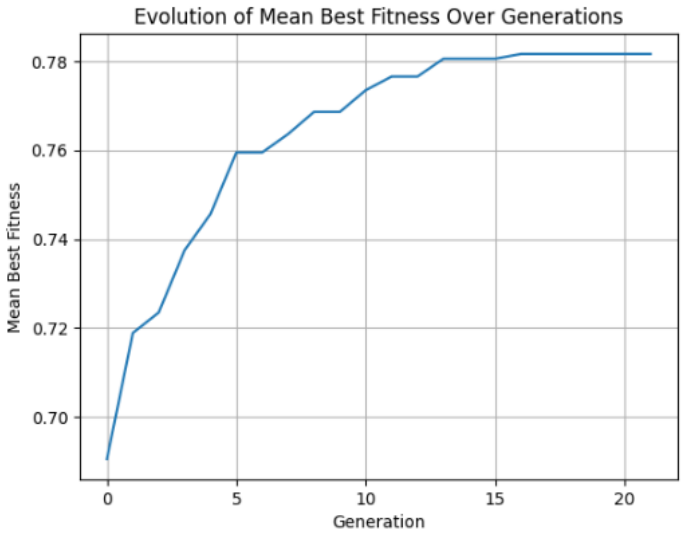
B3. Αξιολόγηση και Επίδραση Παραμέτρων

α) Τα κριτήρια τερματισμού που χρησιμοποίησα στον γενετικό αλγόριθμο είναι ο μέγιστος αριθμός γενεών (100), η υπομονή (6 συνεχόμενες γενιές χωρίς σημαντική βελτίωση) και η ελάχιστη αποδεκτή βελτίωση στο fitness (0.005). Όταν δεν επιτυγχάνεται βελτίωση πάνω από το όριο αυτό για 6 γενιές ή φτάσουμε στις 100 γενιές, ο αλγόριθμος σταματά, εξασφαλίζοντας αποδοτική και έγκαιρη σύγκλιση.

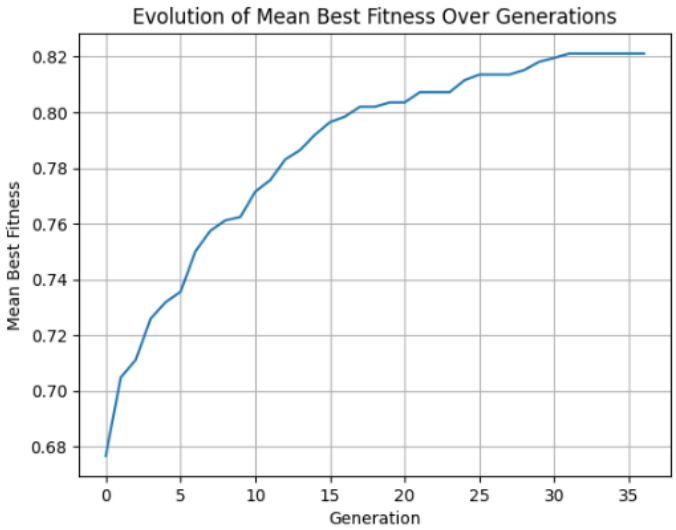
Παρακάτω φαίνεται ο ζητούμενος πίνακας:

Α/Α	ΜΕΓΕΘΟΣ ΠΛΗΘΥΣΜΟΥ	ΠΙΘΑΝΟΤΗΤΑ ΔΙΑΣΤΑΥΡΩΣΗΣ	ΠΙΘΑΝΟΤΗΤΑ ΜΕΤΑΛΛΑΞΗΣ	ΜΕΣΗ ΤΙΜΗ ΒΕΛΤΙΣΤΟΥ	ΜΕΣΟΣ ΑΡΙΘΜΟΣ ΓΕΝΕΩΝ
1	20	0.6	0.00	0.7816	16.8
2	20	0.6	0.01	0.8219	27.8
3	20	0.6	0.10	0.7858	16
4	20	0.9	0.01	0.8332	25
5	20	0.1	0.01	0.8060	27.2
6	200	0.6	0.00	0.8360	18.5
7	200	0.6	0.01	0.8365	20
8	200	0.6	0.10	0.8037	16.2
9	200	0.9	0.01	0.8345	18
10	200	0.1	0.01	0.8346	27.6

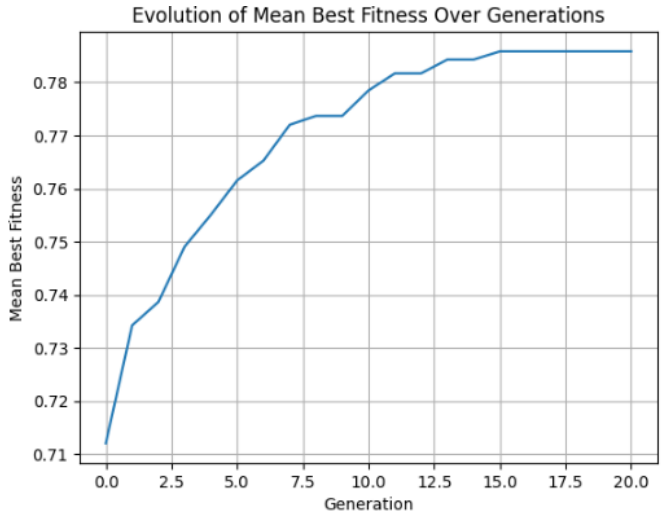
β) 1.



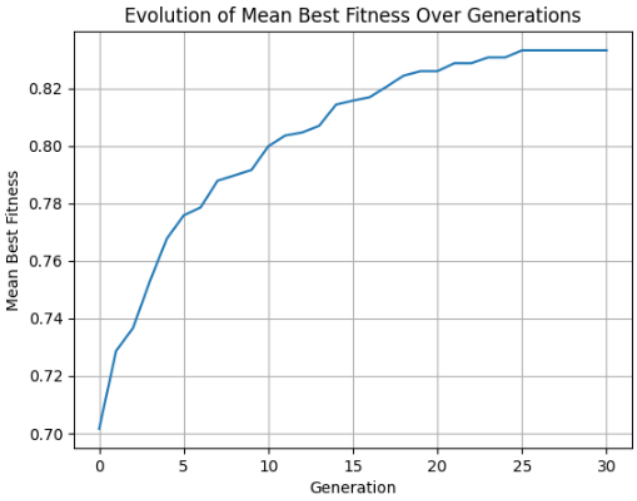
2.


Evolution of Mean Best Fitness Over Generations
Mean Best Fitness
Generation
0.82
0.80
0.78
0.76
0.74
0.72
0.70
0.68
0
5
10
15
20
25
30
35

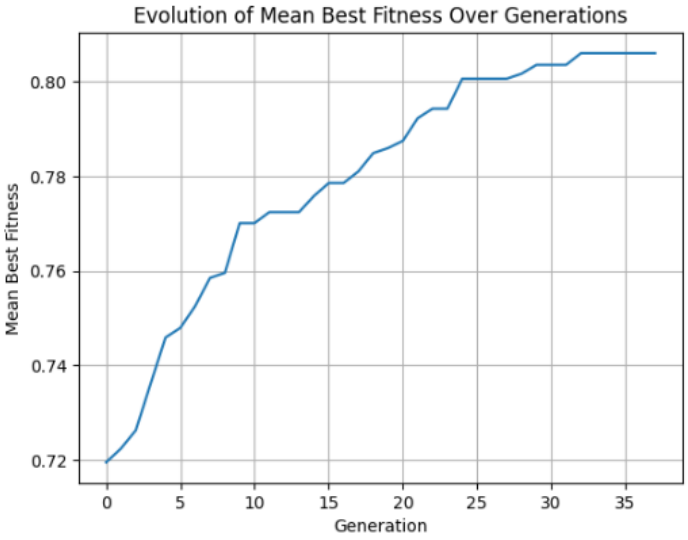
3.


Evolution of Mean Best Fitness Over Generations
Mean Best Fitness
Generation
0.78
0.77
0.76
0.75
0.74
0.73
0.72
0.71
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

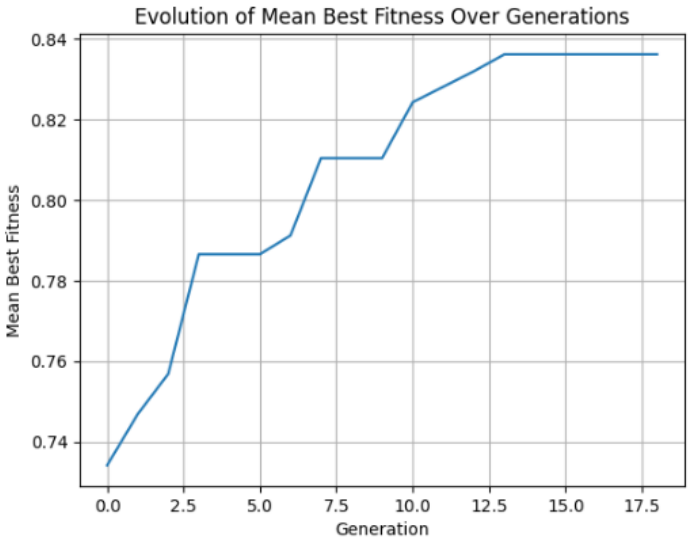
4.


Evolution of Mean Best Fitness Over Generations
Mean Best Fitness
Generation
0.82
0.80
0.78
0.76
0.74
0.72
0.70
0
5
10
15
20
25
30

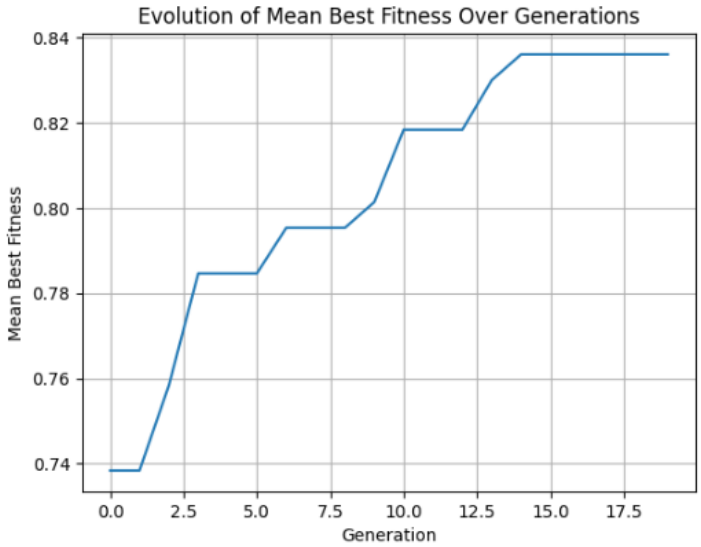
5.


Evolution of Mean Best Fitness Over Generations
Mean Best Fitness
0.72
0.74
0.76
0.78
0.80
0
5
10
15
20
25
30
35
Generation

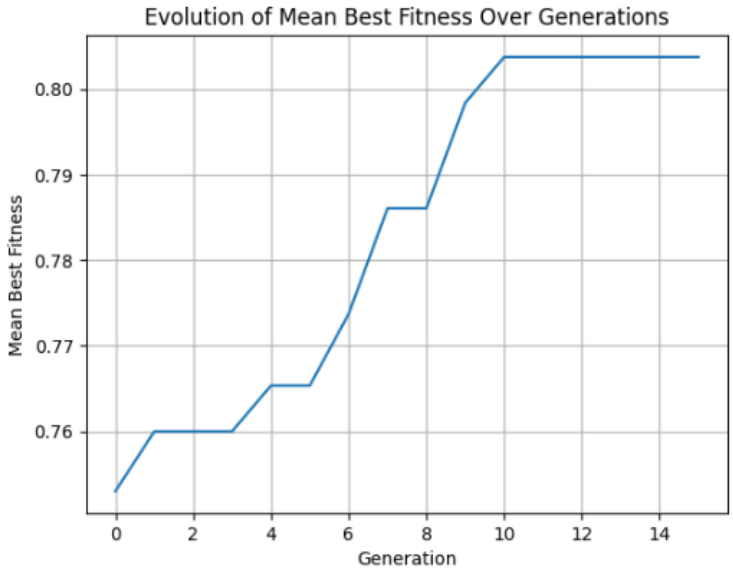
6.


Evolution of Mean Best Fitness Over Generations
Mean Best Fitness
0.74
0.76
0.78
0.80
0.82
0.84
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Generation

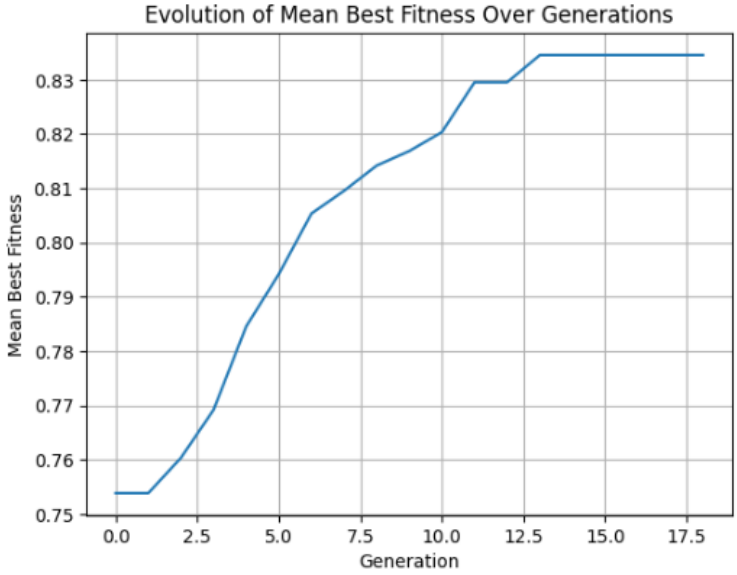
7.


Evolution of Mean Best Fitness Over Generations
Mean Best Fitness
0.74
0.76
0.78
0.80
0.82
0.84
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Generation

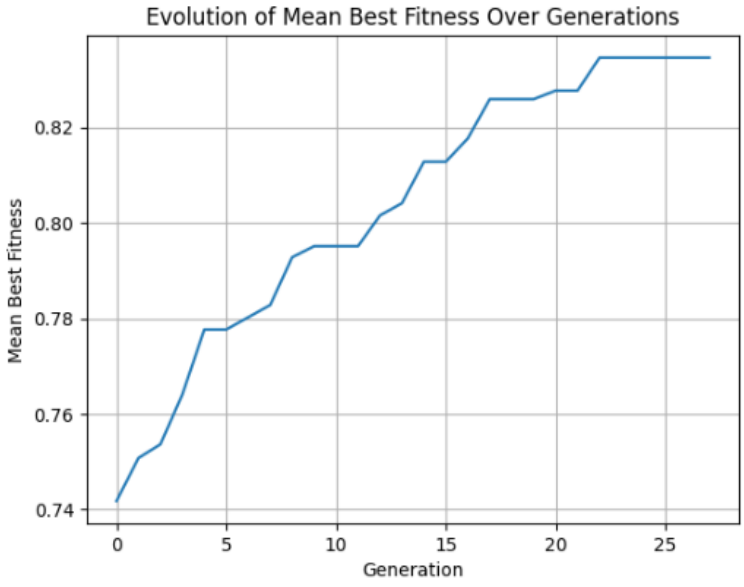
8.


Evolution of Mean Best Fitness Over Generations
Mean Best Fitness
0.80
0.79
0.78
0.77
0.76
0
2
4
6
8
10
12
14
Generation

9.


Evolution of Mean Best Fitness Over Generations
Mean Best Fitness
0.83
0.82
0.81
0.80
0.79
0.78
0.77
0.76
0.75
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Generation

10.


Evolution of Mean Best Fitness Over Generations
Mean Best Fitness
0.82
0.80
0.78
0.76
0.74
0
5
10
15
20
25
Generation

γ) 1. Μέγεθος πληθυσμού

| Τιμές: 20 vs 200 |

- Γενικά παρατηρείται ότι οι δοκιμές με πληθυσμό 200 είχαν υψηλότερες μέσες τιμές βέλτιστου (π.χ. 0.8360, 0.8365, 0.8345, 0.8346) από τις αντίστοιχες με πληθυσμό 20.
- Επίσης, ο μέσος αριθμός γενεών είναι λίγο αυξημένος (18–27.6), αλλά χωρίς υπερβολικές καθυστερήσεις.

Συμπέρασμα:

Η αύξηση του μεγέθους πληθυσμού βελτιώνει τη ποιότητα του τελικού αποτελέσματος (μέγιστο fitness), πιθανώς λόγω καλύτερης ποικιλομορφίας του πληθυσμού, που οδηγεί σε αποδοτικότερη εξερεύνηση του χώρου λύσεων.

---

2. Πιθανότητα διασταύρωσης (Crossover probability)

| Τιμές: 0.1, 0.6, 0.9 |

- Στο μέγεθος πληθυσμού 20, η υψηλή τιμή 0.9 έδωσε την καλύτερη επίδοση (0.8332) και αρκετά καλό ρυθμό σύγκλισης (25 γενεές).
- Στο μέγεθος 200, οι διαφορές είναι μικρότερες, αλλά η τιμή 0.9 (γραμμή 9) έδωσε 0.8345, εφάμιλλο με τις υπόλοιπες.

Συμπέρασμα:

Η αύξηση της πιθανότητας διασταύρωσης βελτιώνει τα αποτελέσματα όταν ο πληθυσμός είναι μικρός, βοηθώντας την ανασυνδυαστική εξερεύνηση. Σε μεγάλους πληθυσμούς, η επίδραση είναι λιγότερο σημαντική αλλά ακόμα θετική.

---

3. Πιθανότητα μετάλλαξης (Mutation probability)

| Τιμές: 0.00, 0.01, 0.10 |

- Χαμηλές τιμές γύρω στο 0.01 φαίνεται να είναι ιδανικές:
  - Για παράδειγμα, γραμμή 2 (0.01) → 0.8219
  - Γραμμή 4 (0.01) → 0.8332
  - Γραμμή 10 (0.01) → 0.8346
- Αντίθετα, πολύ χαμηλή (0.00) ή πολύ υψηλή (0.10) τιμή οδηγεί σε χαμηλότερες επιδόσεις (π.χ. 0.7816 και 0.7858 αντίστοιχα).

Συμπέρασμα:

Η κατάλληλη ποσότητα μετάλλαξης (π.χ. 0.01) βοηθά στην αποφυγή τοπικών ακροτάτων, διατηρώντας την ποικιλία χωρίς να διαταράσσει τη σύγκλιση. Πολύ υψηλή ή μηδενική μετάλλαξη έχει αρνητική επίδραση.

Β4. Αξιολόγηση ΤΝΔ

α) Το τεχνητό νευρωνικό δίκτυο (ΤΝΔ) που προέκυψε με τη χρήση γενετικού αλγορίθμου (Μέρος Β) εμφάνισε τελική ακρίβεια στο test set περίπου 90.47%, ενώ το αντίστοιχο μοντέλο του Μέρους Α είχε ακρίβεια περίπου 85.63%. Η σημαντική αυτή διαφορά δείχνει ότι το μοντέλο του Μέρους Β έχει καλύτερη γενικευτική ικανότητα, δηλαδή είναι πιο ικανό να προβλέπει σωστά νέα, άγνωστα δεδομένα.
Η βελτίωση αυτή οφείλεται κατά κύριο λόγο στην επιλογή χαρακτηριστικών που έγινε μέσω του γενετικού αλγορίθμου. Αν και το πλήθος των χαρακτηριστικών μειώθηκε, το μοντέλο όχι μόνο δεν έχασε ακρίβεια, αλλά παρουσίασε και καλύτερη απόδοση. Αυτό σημαίνει ότι τα χαρακτηριστικά που επιλέχθηκαν ήταν τα πιο αντιπροσωπευτικά και περιείχαν τις σημαντικότερες πληροφορίες για την πρόβλεψη, ενώ απομακρύνθηκαν περιττά ή θορυβώδη δεδομένα.
Επιπλέον, η διαφορά στην απόδοση δείχνει πως το αρχικό μοντέλο του Μέρους Α ίσως εμφάνιζε υπερπροσαρμογή στα δεδομένα εκπαίδευσης και επαλήθευσης. Το μοντέλο του Μέρους Β, έχοντας λιγότερα αλλά πιο χρήσιμα χαρακτηριστικά, φαίνεται να γενικεύει καλύτερα και να αποφεύγει την υπερπροσαρμογή, προσφέροντας πιο σταθερά και αξιόπιστα αποτελέσματα στο test set.
Συνολικά, η χρήση γενετικού αλγορίθμου για επιλογή χαρακτηριστικών αποδείχθηκε ιδιαίτερα αποτελεσματική, καθώς οδήγησε σε καλύτερη απόδοση, μικρότερη πολυπλοκότητα και βελτιωμένη ικανότητα γενίκευσης του μοντέλου.

β) Για να εξεταστεί η πλήρης απόδοση του βέλτιστου τεχνητού νευρωνικού δικτύου (ΤΝΔ) που προέκυψε από τον γενετικό αλγόριθμο (ΓΑ), το δίκτυο επανεκπαιδεύτηκε χρησιμοποιώντας ολόκληρο το διαθέσιμο σύνολο εκπαίδευσης (δηλαδή και τα δεδομένα εκπαίδευσης και τα δεδομένα επικύρωσης). Ο στόχος αυτής της επανεκπαίδευσης είναι να εκμεταλλευτούμε όλη τη διαθέσιμη πληροφορία ώστε το τελικό μοντέλο να έχει τη μέγιστη δυνατή απόδοση όταν εφαρμοστεί στο test set.
Μετά την επανεκπαίδευση, το μοντέλο αξιολογήθηκε ξανά στο test set και η τελική ακρίβεια παρέμεινε περίπου στο ίδιο υψηλό επίπεδο (90.15%), δείχνοντας ότι η χρήση περισσότερων δεδομένων στην εκπαίδευση δεν οδήγησε σε υπερπροσαρμογή αλλά ενίσχυσε περαιτέρω την απόδοση. Αυτό επιβεβαιώνει τη σταθερότητα και τη γενικευτική ικανότητα του μοντέλου.

Σε σύγκριση με το αρχικό πείραμα (πριν την επανεκπαίδευση), δεν παρατηρείται σημαντική διαφοροποίηση στην ακρίβεια, κάτι που υποδηλώνει ότι το μοντέλο έχει ήδη καλή συμπεριφορά και ότι τα επιλεγμένα χαρακτηριστικά είναι πράγματι επαρκή για τη συγκεκριμένη ταξινόμηση.