Υπολογιστική Νοημοσύνη

Β’ μέρος εργασίας

Ονοματεπώνυμο Ακαρέπης Ιωάννης

ΑΜ 1054368

Έτος 5ο

Email [akarepis@ceid.upatras.gr](mailto:akarepis@ceid.upatras.gr)

**Β – Εισαγωγή**

Στο συγκεκριμένο μέρος της εργασίας σκοπός ήταν η ελάττωση του αριθμού εισόδων του ΤΝΔ που προέκυψε από το μέρος Α με χρήση γενετικών αλγορίθμων.

Λόγω της φύσης του συγκεκριμένου προβλήματος υπήρξαν αρκετές αναλύσεις και παραλλαγές τόσο θεωρητικές όσο και τεχνικές/προγραμματιστικές οι οποίες θα αναλυθούν παρακάτω. Όσον αφορά τον κώδικα της συγκεκριμένης εργασίας, βρίσκεται στον παρακάτω σύνδεσμο.

Ο κώδικας: <https://drive.google.com/drive/folders/12iciBYSxPKnZbMC1o9aLwZO4MJT89CMk?usp=sharing>

Στο μέρος Α της εργασίας υπήρξε αρκετή εξάρτηση από βιβλιοθήκες όπως η **keras,** η **tensorflow** και η **scikit\_learn** της **Python** για την δημιουργία του ΤΝΔ καθώς και για την δοκιμή των υπερπαραμέτρων. Στο μέρος Β η υλοποίηση του γενετικού αλγορίθμου έγινε “από το 0”, δηλαδή δημιουργήθηκε μία κλάση από την αρχή που υλοποίει τον αλγόριθμο που ζητήθηκε.

Όπως είναι αναμενόμενο, η χρονική πολυπλοκότητα είναι αρκετά υψηλή, ωστόσο σε τέτοιου είδους συστήματα δεν μας ενδιαφέρει ιδιαίτερα ο χρόνος εκτέλεσης παρά μόνο το αποτέλεσμα και κατά πόσο αποδοτικό αυτό είναι.

Η κλάση που δημιουργήθηκε, ονομάζεται “ my\_Gen” και καλείται ως:

**My\_Gen(ANN\_model, input\_dataset, output\_dataset, crossover\_probability, mutation\_probability, chromosomes, population\_initializer)**

Όπυυ

* ANN\_model είναι το μοντέλο ΤΝΔ που έχει εκπαιδευτεί (Τα επιπρόσθετα αρχεία “ **model.json**” και “ **model.h5**” που εμπεριέχονται στο **google drive** παρέχουν πληροφορίες για το ΤΝΔ του μέρους Α που επιλέχθηκε καθώς και για τα βάρη του εκπαιδευμένου μοντέλου, και είναι απαραίτητα για την σωστή λειτουργία του γενετικού αλγορίθμου).
* input\_dataset είναι τα δεδομένα εισόδου.
* output\_dataset είναι τα δεδομένα εξόδου.
* crossover\_probability είναι η πιθανότητα διασταύρωσης για την οποία θα πρέπει να ισχύει ότι
* mutation\_probability είναι η πιθανότητα μετάλλαξης και ισχύει ο ίδιος περιορισμός με την διασταύρωση.
* chromosomes είναι το πλήθος των ατόμων που θα ανήκουν στον πληθυσμό.
* population\_initializer είναι προαιρετικό όρισμα και αν χρησιμοποιηθεί ορίζει περιοχές γονιδίων στις οποίες όλα τα άτομα του πληθυσμού αρχικοποιούνται με μηδέν (παρακάτω εξηγείται και με πιο σαφή τρόπο πώς αξιοποιείται αυτό το όρισμα)

Οι μέθοδοι που περιέχει η κλάση αυτή είναι οι

* **\_\_init\_\_** όπου εκεί γίνονται όλες οι αναγκαίες αρχικοποιήσεις και οι έλεγχοι ορθότητας των ορισμάτων.
* **\_\_fitness\_function\_\_** όπου περιέχει την συνάρτηση καταλληλότητας που περιγράφει την εξελικτική πορεία των ατόμων ανά γενιά
* **fitness\_eval** όπου εφαρμόζεται η διαδικασία της αξιολόγησης όλων των ατόμων του πληθυσμού (χρησιμοποίει την **\_\_fitness\_function\_\_** για τον υπολογισμό της καταλληλότητας)
* **rank\_based\_roulette\_wheel\_selection** όπου αποτελεί τον γενετικό τελεστή επιλογής (εξηγείται αναλυτικότερα στην θεωρητική ανάλυση παρακάτω)
* **\_\_swap\_\_** όπου χρησιμοποιείται στην διασταύρωση ως η πράξη ανταλλαγής γονιδίων μεταξύ του επιλεγόμενου ζευγαριού προς διασταύρωση
* **crossover\_process** όπου περιέχει τον γενετικό τελεστή διασταύρωσης (εξηγείται αναλυτικότερα στην θεωρητική ανάλυση)
* **mutation\_process** όπου περιέχει τον γενετικό τελεστή μετάλλαξης (εξηγείται αναλυτικότερα στην θεωρητική ανάλυση)
* **run** που ουσιαστικά αποτελεί την “main” της κλάσης και αυτή είναι που καλείται από τον χρήστη προκειμένου να εκτελεστεί ο γενετικός αλγόριθμος. Ο χρήστης έχει την επιλογή να θέσει και ένα όρισμα τύπου **Boolean** για το αν θέλει να δει και την γραφική παράσταση της εξελικτικής πορείας του βέλτιστου ατόμου ανά γενιά. Η χρήση του ορίσματος είναι προαιρετική και αν δεν επιλεγεί το σύστημα θα το θεωρήσει ως **False** από default. Η μέθοδος αυτή επιστρέφει το βέλτιστο χρωμόσωμα που εντοπίστηκε ως λύση.
* **best\_solution\_history** που λειτουργεί ως history verbose. Δηλαδή αν ο χρήστης θέλει να απομονώσει από την κλάση δεδομένα για την εξελικτική πορεία του γενετικού αλγορίθμου μπορεί να χρησιμοποιήσει αυτήν την μέθοδο η οποία επιστρέφει ως **list** την εξελικτική απόδοση του βέλτιστου ατόμου ανά γενιά. Και εδώ ο χρήστης έχει την επιλογή να θέσει και ένα όρισμα τύπου **Boolean** που παρουσιάζει ακριβώς την ίδια λειτουργία με την μέθοδο **run**

**Β1 – Σχεδιασμός ΓΑ (Θεωρητική ανάλυση απαιτήσεων συστήματος)**

Οι απαιτήσεις που πρέπει να καλυφθούν σε αυτό το σημείο είναι οι επιλογές

* Κωδικοποίησης
* Αρχικού πληθυσμού
* Συνάρτησης Καταλληλότητας
* Γενετικών Τελεστών
  + Τελεστής Επιλογής
  + Τελεστής Διασταύρωσης
  + Τελεστής Μετάλλαξης

**Β1.1 – Κωδικοποίηση**

Στο κομμάτι της κωδικοποίησης είναι σημαντικό να κατανοήσουμε ποια είναι η δομή του προβλήματος που πρέπει να λύσουμε. Στο συγκεκριμένο πρόβλημα θεωρούμε ότι έχουμε ένα ΤΝΔ το οποίο περιέχει 784 εισόδους και μας ζητείται να εφαρμόσουμε τον γενετικό αλγόριθμο για να μειώσουμε αυτόν τον αριθμό εισόδων. Για να μειωθεί ο αριθμός αυτός, το μόνο που πρέπει να κάνουμε είναι ορισμένες εισόδους να τις μηδενίσουμε (πιο συγκεκριμένα να μηδενίσουμε τα βάρη που αντιστοιχούν στις ακμές που έχουν ως άκρα τις εισόδους αυτές). Έτσι η κωδικοποίηση θα είναι πολύ απλή, καθώς μπορούμε να θεωρήσουμε για το κάθε άτομο του πληθυσμού μία ακολουθία από 784 δυαδικά ψηφία (δηλαδή 784 γονίδια) όπου το κάθε άτομο αντιπροσωπεύει ένα στιγμιότυπο του ΤΝΔ και ακολουθεί την παρακάτω λογική

* Αν το γονίδιο ενός ατόμου είναι 0 τότε μηδενίζουμε τις ακμές (τα βάρη) που έχουν ως άκρο την είσοδο j (όπου j η στήλη στην οποία βρίσκεται αυτό το γονίδιο), δηλαδή θεωρούμε πως η είσοδος j δεν περνάει στο ΤΝΔ
* Αν το γονίδιο ενός ατόμου είναι 1 τότε περνάει απευθείας ως είσοδος στο ΤΝΔ.

Αυτή η κωδικοποίηση είναι αρκετά ικανοποιητική για το συγκεκριμένο είδος προβλήματος καθώς διευκολύνει αρκετά την θεωρητική ανάλυση όλων των υπολοίπων παραμέτρων.

**Β1.2 – Αρχικός Πληθυσμός**

Όταν αναφερόμαστε σε αρχικό πληθυσμό, κυρίως εννοούμε τον τρόπο με τον οποίο θα αρχικοποιηθούν τα άτομα του πληθυσμού. Προφανώς, δεν θα ήταν βέλτιστη επιλογή στο συγκεκριμένο πρόβλημα, να αρχικοποιηθούν όλα τα γονίδια ενός ατόμου με 0 ή όλα τα γονίδια με 1 (θα χρειαζόταν μεγάλη πιθανότητα μετάλλαξης για να μεταβούμε σε καταστάσεις που έρχονται πιο κοντά στην λύση, όπου αυτό και πάλι θα αύξανε την υπολογιστική πολυπλοκότητα χωρίς λόγο).

Η αρχικοποίηση που επιλέχθηκε εδώ είναι η τυχαία απόδοση 0 και 1 στα γονίδια, με την συνάρτηση τυχαιότητας να ακολουθεί ομοιόμορφη κατανομή. Αποτελεί μία καλή μέση λύση αρχικού πληθυσμού και η κωδικοποίηση που επιλέχθηκε είναι αρκετά βολική, ούτως ώστε να εφαρμοστεί κάτι τέτοιο.

Μία καλή προσέγγιση στο κομμάτι της αρχικοποίησης είναι και η μελέτη της κατανομής της πληροφορίας στα δεδομένα. Ενδεχομένως, να μπορούμε να δημιουργήσουμε ένα **σχήμα** στον πληθυσμό μας προκειμένου να επιταχύνουμε την σύγκλιση του γενετικού αλγορίθμου (σαφώς αυτό μειώνει κάπως σημαντικά την πολυπλοκότητα).

Στο περιβάλλον της MATLAB ελέγχθηκε η κατανομή της πληροφορίας στο σετ ελέγχου χρησιμοποιώντας τον μέσο όρο της κάθε στήλης και εμφανίζοντας την γραφική παράσταση, η οποία φαίνεται παρακάτω

Chart, histogram

Description automatically generated

Όπως φαίνεται υπάρχει ολοένα και μεγαλύτερη συγκέντρωση πληροφορίας όσο περισσότερο κατευθυνόμαστε προς το κέντρο και σχεδόν μηδενική πληροφορία στις άκρες. Αυτό στις εικόνες σημαίνει, πως το ψηφίο εμφανίζεται στο κέντρο της εικόνας και όχι στις άκρες της.

Chart

Description automatically generated with medium confidence

Για παράδειγμα, η παραπάνω εικόνα που απεικονίζει το ψηφίο 7 βρίσκεται στο κέντρο της εικόνας και όχι αριστερότερα ή δεξιότερα. Το ίδιο ισχύει και για όλα τα υπόλοιπα ψηφία του dataset ελέγχου (και dataset εκπαίδευσης).

Μπορούμε να εκμεταλλευτούμε αυτήν την πληροφορία και πράγματι να ορίσουμε ένα σχήμα στον πληθυσμό μας. Το σχήμα αυτό, προκύπτει εύκολα καθώς θα πρέπει απλά να θεωρήσουμε, βάσει της κωδικοποίησης που δημιουργήσαμε, πως οι περιοχές 1-60 και 750-784 αρχικοποιούνται με 0.

Chart

Description automatically generated with medium confidence

Μετά την εφαρμογή του σχήματος, είναι εμφανές πως το 7 που είχαμε και προηγουμένως δεν άλλαξε καθόλου το οποίο σημαίνει πως πράγματι το σχήμα αυτό μάλλον είναι καλό (προφανώς το 7 της εικόνας είναι μία μεμονωμένη περίπτωση, δεν ισχύει κατ’ανάγκη για όλα τα ψηφία).

**Β1.3 – Συνάρτηση καταλληλότητας**

Η συνάρτηση καταλληλότητας που χρησιμοποιήθηκε για το συγκεκριμένο πρόβλημα είναι η

όπου

* x 🡪 Η απώλεια του ΤΝΔ μέσω της cross entropy (loss function)
* y 🡪 Το πλήθος των εισόδων που περιγράφει ένα άτομο του πληθυσμού
* c 🡪 σταθερά που περιγράφει τον μέγιστο αριθμό εισόδων που δίνεται στον γενετικό αλγόριθμο.

Η συνάρτηση καταλληλότητας αυτή, είναι μία γνησίως φθίνουσα συνάρτηση, κάτι το οποίο θέλαμε να επιτύχουμε, καθώς σκοπός μας είναι η μείωση του αριθμού εισόδων, και επομένως η συνάρτηση καταλληλότητας θέλουμε να έχει μεγαλύτερες τιμές για μικρότερα x και y. Η σταθερά c χρησιμοποιείται σαν “scaler”, δηλαδή φέρνει την μεταβλητή y στην ίδια κλίμακα με την μεταβλητή x, για να μην κυριαρχεί η y που είναι πολύ μεγαλύτερη.

Συγκεκριμένα, το πεδίο ορισμού της μεταβλητής x είναι το διάστημα [0,1] ενώ το πεδίο ορισμού της y είναι το (0,c]. Το σύνολο τιμών της συνάρτησης F είναι το διάστημα [0, ∞)

**Β1.4 – Γενετικοί τελεστές**

**Τελεστής Επιλογής (Επιλογή ρουλέτας με βάση την κατάταξη)**

Στο κομμάτι της επιλογής, αποφασίστηκε αρχικά να χρησιμοποιηθεί η χρήση μεθόδου ρουλέτας με βάση το κόστος, ωστόσο κατά την διάρκεια της υλοποίησης του κώδικα και των πειραμάτων, παρατηρήθηκε πως η επιλογή αυτή ήταν λανθασμένη. Ο λόγος ήταν ο τρόπος με τον οποίο αρχικοποίειται ο πληθυσμός. Όπως εξηγήθηκε και πιο πάνω, η αρχικοποίηση γίνεται με ομοιόμορφο τρόπο κάτι το οποίο σημαίνει πως η συνάρτηση καταλληλότητας δεν θα διαφέρει πολύ μεταξύ των ατόμων του πληθυσμού. Ακόμη, η μέθοδος ρουλέτας βάσει κόστους χρησιμοποίει την συνάρτηση καταλληλότητας για να υπολογίσει πιθανότητες επιλογής, ωστόσο είναι προφανές ότι με παραπλήσιες τιμές στην συνάρτηση καταλληλότητας, τα άτομα του πληθυσμού θα τείνουν να είναι ισοπίθανα. Ακόμη, η συνάρτηση καταλληλότητας δεν παρουσιάζει μεγάλες μεταβολές από γενιά σε γενιά (για παράδειγμα αν στην γενιά i η συνάρτηση καταλληλότητας είναι 1.45 στην i + 1 θα μπορούσε να είναι 1.46 ή ακόμα χειρότερα 1.456…) επομένως σε όλες τις γενιές θα καταλήξουν τα άτομα του πληθυσμού να μοιάζουν (και πρακτικά να είναι) ισοπίθανα, κάτι το οποίο μας δείχνει πως ο γενετικός αλγόριθμος δεν επιβραβεύει το καλύτερο άτομο και συνεπώς δεν μπορεί να υπάρξει βελτίωση ή και σύγκλιση του αλγορίθμου.

Η βέλτιστη επιλογή σε ένα τέτοιο πρόβλημα και με βάση την συνάρτηση καταλληλότητας που επιλέχθηκε, φαίνεται να είναι η μέθοδος της ρουλέτας με βάση την κατάταξη. Η μέθοδος αυτή χρησιμοποίει την συνάρτηση καταλληλότητας για να καθορίσει κάποια “ranks” στα άτομα του πληθυσμού με το μεγαλύτερο rank να σηματοδοτεί το καλύτερο της γενιάς και το μικρότερο rank, το χειρότερο (αν δύο άτομα έχουν την ίδια τιμή στην συνάρτηση καταλληλότητας λαμβάνουν το ίδιο rank). Αφού έχουν καταταχθεί τα άτομα του πληθυσμού, υπολογίζεται η πιθανότητα επιλογής τους με βάση την παρακάτω σχέση

όπου pop\_size το μέγεθος του πληθυσμού, Ps η πιθανότητα επιλογής του ατόμου j και r η κατάταξη που έχει το άτομο. Αυτή η μέθοδος σαφώς και επιβραβεύει το καλύτερο άτομο της γενιάς κάθε φορά, καθώς πλέον οι πιθανότητες επιλογής είναι μεγαλύτερες για άτομα που παρουσιάζουν μεγαλύτερες τιμές στην συνάρτηση καταλληλότητας.

**Τελεστής Διασταύρωσης (Ομοιόμορφη διασταύρωση)**

Η διασταύρωση είναι ένα αρκετά σημαντικό κομμάτι για τον καθορισμό τόσο της ταχύτητας σύγκλισης του αλγορίθμου όσο και τον ρυθμό βελτίωσης ανά γενιά. Λόγο του τρόπου με τον οποίο έχει δομηθεί από τα παραπάνω η αρχιτεκτονική του γενετικού αλγορίθμου, και λόγω της φύσης του προβλήματος η επιλογή ομοιόμορφης διασταύρωσης δείχνει να είναι η καλύτερη. Σκοπός του γενετικού αλγορίθμου είναι η μείωση των χαρακτηριστικών (εισόδων) του ΤΝΔ και συνεπώς η χρήση διασταύρωσης τόσο μονού όσο και πολλαπλού σημείου δεν συνίσταται καθώς η αρχικοποίηση του αρχικού πληθυσμού γίνεται με ομοιόμορφο τρόπο. Αυτό σημαίνει ότι στα σημεία κοπής υπάρχει υψηλή πιθανότητα να μεταφέρεται το ίδιο πλήθος από 0 και 1 μεταξύ δύο γονέων, το οποίο ουσιαστικά δηλώνει πως δεν θα υπάρχει βελτίωση στην γενιά ακόμα και μετά την διασταύρωση. Η επιλογή της ομοιόμορφης διασταύρωσης είναι η πιο “ασφαλής” επιλογή για το συγκεκριμένο πρόβλημα και συνεπώς επιλέγεται στην υλοποίηση. Το σχήμα του πληθυσμού δεν χαλάει με αυτή τη μέθοδο (και με καμία από τις υπόλοιπες μεθόδους) καθώς όλα τα άτομα του πληθυσμού έχουν αρχικοποιηθεί με τον ίδιο τρόπο (οι μηδενικές περιοχές απλά θα μεταφέρουν το 0 από τον έναν γονέα στον άλλον).

**Τελεστής Μετάλλαξης (Χρήση ελιτισμού)**

Η μετάλλαξη αποτελεί ένα πολύ σημαντικό κομμάτι στην ανάλυση αυτή, καθώς μπορούμε πολύ εύκολα να οδηγηθούμε σε παγίδα. Έτσι όπως έχει δομηθεί ο γενετικός αλγόριθμος η θεωρία λέει πως καλό θα ήταν να μην εισάγουμε κάποια πιθανότητα μετάλλαξης (Pm = 0), καθώς αν είχαμε μετάλλαξη θα καταστρεφόταν το αρχικό σχήμα που ορίστηκε. Από την άλλη όμως η χρήση της μετάλλαξης μας βοηθάει να “ξεκολλήσουμε” από πιθανές γνωστές καταστάσεις και να εξερευνούμε τελείως διαφορετικές καταστάσεις στις επόμενες γενιές. Αυτό έχει πρακτική χρήση, επειδή ο αλγόριθμός μας είναι πιο πιθανό να ξεφύγει από κάποιο τοπικό βέλτιστο και να συνεχίσει να εντοπίζει “καλύτερα” βέλτιστα.

Η χρήση Ελιτισμού μας εγγυάται πως το βέλτιστο άτομο της κάθε γενιάς (ή τα βέλτιστα αν έχουν την ίδια τιμή συνάρτησης καταλληλότητας) θα παραμένει αναλλοίωτο (ή αναλλοίωτα) και επομένως δεν θα έχουμε κατάρρευση της εξελικτικής πορείας του γενετικού αλγορίθμου. Συνεπώς η χρήση της συγκεκριμένης μεθόδου είναι απαραίτητη.

**Β1.4 – Συμπεράσματα (μόνο θεωρητικά)**

Όλα τα παραπάνω, καλύπτουν την θεωρητική ανάλυση του συστήματος που πρέπει να δημιουργήσουμε. Ωστόσο, για να είναι πλήρης η περιγραφή πρέπει να μελετήσουμε και πώς αναμένουμε να λειτουργήσει βέλτιστα αυτός ο αλγόριθμος. Αυτό σημαίνει πως πρέπει να μελετήσουμε τις παραμέτρους (θεωρητικά) για τις οποίες το σύστημά μου θα αποκριθεί με τον καλύτερο τρόπο.

Για το συγκεκριμένο πρόβλημα η ανάλυση αυτή είναι σχετικά εύκολη, καθώς έχουμε καλύψει παραπάνω όλα τα “εργαλεία” που θα χρησιμοποιήσουμε. Αρχικά πρέπει να ελέγξουμε την σχέση του πλήθους των ατόμων σε σχέση με την απόκριση του παραπάνω συστήματος. Για να το ελέγξουμε αυτό θα πρέπει να μελετήσουμε τους γενετικούς τελεστές που ορίστηκαν παραπάνω. Οι τελεστές αυτοί ορίζονται ως εξής

* Για επιλογή 🡪 Rank based roulette wheel selection (Χρήση ρουλέτας βάσει κατάταξης)
* Για διασταύρωση 🡪 Ομοιόμορφη Διασταύρωση
* Για Μετάλλαξη 🡪 Ελιτισμός

Επίσης, έχουμε δημιουργήσει ένα αρχικό σχήμα το οποίο θα θέλαμε όσο είναι δυνατόν να μην καταστρέψουμε καθώς αποτελεί ένα ισχυρό σχήμα. Γνωρίζοντας αυτό, θα θέλαμε να επιλέξουμε μηδενική πιθανότητα μετάλλαξης (ή έστω υπερβολικά μικρή), το οποίο όμως σημαίνει ότι το σύστημα μας, κατά την διασταύρωση είναι σχετικά πιθανό να επανέρχεται σε “γνωστές” καταστάσεις (για παράδειγμα ένα άτομο στην γενιά 1 που έχει 500 εισόδους θα μπορούσε στην γενιά 100 να έχει και πάλι 500 εισόδους το οποίο σαφώς δείχνει πως δεν υπάρχει βελτίωση). Η Επιλογή της μεθόδου ομοιόμορφης διασταύρωσης, προσπαθεί να επιλύσει το πρόβλημα αυτό με την υπόθεση όμως ότι τα άτομα του πληθυσμού είναι πολλά (είναι διαφορετικό να επιλέγεις από 20 άτομα για διασταύρωση και διαφορετικό από 200, πόσο μάλλον όταν αυτή η ανταλλαγή γενετικού υλικού γίνεται με ομοιόμορφο τρόπο). Επομένως, εύκολα παρατηρούμε πως για να λειτουργήσει αυτό το σύστημα αποδοτικά, θα χρειαστούμε μία πιθανότητα διασταύρωσης, σχετικά υψηλή (), μηδενική πιθανότητα μετάλλαξης και ένα μεγάλο πλήθος ατόμων για να αντιμετωπίσουμε το πρόβλημα της μηδενικής πιθανότητας μετάλλαξης.

Η επαλήθευση σε αυτό, έρχεται από τον τελεστή επιλογής ο οποίος χρησιμοποίει την ρουλέτα βάσει κατάταξης, το οποίο σημαίνει πως αν έχω αρκετά μεγάλο πληθυσμό, θα πρέπει να επιλέξω τα καλύτερα άτομα αρκετές φορές (σαφώς υπάρχει και η πιθανότητα η ρουλέτα να επιλέξει ένα άτομο με μέση απόδοση και όχι βέλτιστη, αλλά δεν πειράζει καθώς στην επόμενη γενιά πολύ πιθανόν να καταλήξει στο τέλος της κατάταξης) και αυτά τα άτομα θα διασταυρωθούν (κάποια από αυτά).

Έτσι στα πειράματα που αναφέρονται στο ερώτημα **B3** μπορούμε να αναμένουμε εκ των προτέρων πως το καλύτερο πείραμα για το σύστημά μας θα μας το δώσει το πείραμα 6 και το χειρότερο το πείραμα 3.

**Β2 – Υλοποίηση ΓΑ**

Η υλοποίηση του γενετικού αλγορίθμου ακολούθησε πιστά την δομή που αναλύθηκε παραπάνω. Το πρόγραμμα δημιουργήθηκε σε γλώσσα **Python** και όπως αναφέρθηκε και στην εισαγωγή η κλάση my\_Gen δημιουργήθηκε από την αρχή, το οποίο σημαίνει πως όλος ο πηγαίος κώδικας που στέλνεται περιέχει σαφώς ορισμένα (και σχολιασμένα) όλα όσα εξηγήθηκαν παραπάνω.

**Β3.1 – Αξιολόγηση και επίδραση παραμέτρων**

Σε αυτό το σημείο παρουσιάζεται απευθείας ο πίνακας και παρακάτω οι γραφικές παραστάσεις που ζητούνται και στο τέλος θα διατυπωθεί το συμπέρασμα.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Α/Α | Μέγεθος Πληθυσμού | Πιθανότητα διασταύρωσης | Πιθανότητα μετάλλαξης | Μέση τιμή βέλτιστου | Μέσος αριθμός γενεών |
| 1 | 20 | 0.6 | 0.00 | 1.6227123419875684 | 18 |
| 2 | 20 | 0.6 | 0.01 | 1.7267480974003812 | 51 |
| 3 | 20 | 0.6 | 0.10 | 1.5470737314224243 | 37 |
| 4 | 20 | 0.9 | 0.01 | 1.680994706458311 | 57 |
| 5 | 20 | 0.1 | 0.01 | 1.6532978741213732 | 33 |
| 6 | 200 | 0.6 | 0.00 | 2.9768879133770723 | 107 |
| 7 | 200 | 0.6 | 0.01 | 2.1161049313717576 | 120 |
| 8 | 200 | 0.6 | 0.10 | 1.590174662537045 | 35 |
| 9 | 200 | 0.9 | 0.01 | 2.0756721139443627 | 91 |
| 10 | 200 | 0.1 | 0.01 | 1.6356151175485466 | 32 |

Chart, line chart

Description automatically generated

Πείραμα 1

Chart

Description automatically generated

Πείραμα 2

Chart, line chart, histogram

Description automatically generated

Πείραμα 3

Chart, scatter chart

Description automatically generated

Πείραμα 4

Chart, line chart

Description automatically generated

Πείραμα 5

Chart, line chart

Description automatically generated

Πείραμα 6

Chart, line chart, scatter chart

Description automatically generated

Πείραμα 7

Chart

Description automatically generated

Πείραμα 8

Chart, line chart, scatter chart

Description automatically generated

Πείραμα 9

Chart, line chart, histogram

Description automatically generated

Πείραμα 10

**Β3.2 – Συμπεράσματα (Πρακτικά και θεωρητικά)**

Με βάση τα αποτελέσματα που παίρνουμε από κάθε πείραμα μπορούμε να αξιολογήσουμε την ποιότητα του γενετικού αλγορίθμου στην πράξη ως προς την θεωρητική ανάλυση που κάναμε παραπάνω.

Σύμφωνα με την θεωρητική προσέγγιση του προβλήματος αυτού, αναμέναμε το πείραμα 6 να είναι το καλύτερο και πράγματι έχει την μέγιστη μέση τιμή απόδοσης. Ομοίως, αναμέναμε το πείραμα 3 να έχει την χειρότερη τιμή, και πράγματι στον ανωτέρω πίνακα φαίνεται αυτό ακριβώς.

Τα πειράματα αυτά επαληθεύουν τα συμπεράσματα της θεωρητικής ανάλυσης γιατί πράγματι, για μεγάλο πληθυσμό, πιθανότητα διασταύρωσης μεγαλύτερη ή ίση του 0.5 και πιθανότητα μετάλλαξης 0 ή πολύ κοντά στο 0, το σύστημα φαίνεται να εμφανίζει τις καλύτερες τιμές βέλτιστης απόδοσης (μέσης τιμής αυτών). Πιο συγκεκριμένα το πείραμα 6 έφτασε μέσο αριθμό εισόδων ΤΝΔ ίσο με 247 (και σε μεμονωμένες εκτελέσεις έχει φτάσει και 190 εισόδους περίπου).

Οι γραφικές παραστάσεις αποτυπώνουν πλήρως όλα όσα προαναφέρθηκαν και στην θεωρητική ανάλυση και στα ανωτέρω συμπεράσματα. Οι γραφικές παραστάσεις των πειραμάτων 6, 7 και 9 δείχνουν να είναι οι πιο ευσταθείς και ομαλές, έναντι όλων των υπολοίπων. Πράγματι, αν προσέξουμε τα συγκεκριμένα πειράματα οι παράμετροι που χρησιμοποιήθηκαν βρίσκονται εντός των προβλέψεων που έγιναν πριν την πλήρη υλοποίηση. Παρομοίως, μπορούμε να παρατηρήσουμε πως τα πειράματα 3 και 8 είναι τα χειρότερα ως προς την ευστάθεια (υπάρχουν αρκετές αυξομειώσεις και ακόμα χειρότερα υπάρχει διάστημα εντός των γενεών όπου η βέλτιστη συνάρτηση καταλληλότητας ελαττώνεται).

**Β4 – Αξιολόγηση ΤΝΔ**

Σε αυτό το σημείο επιλέγουμε ως βέλτιστες παραμέτρους αυτές του πειράματος 6 και επομένως συνεχίζουμε με πληθυσμό 200 ατόμων, πιθανότητα διασταύρωσης 0.6 και πιθανότητα μετάλλαξης 0. Τρέχοντας πλέον το αντίστοιχο κομμάτι κώδικα που δημιουργήθηκε για αυτόν τον σκοπό μπορούμε παρακάτω να δούμε απευθείας το στιγμιότυπο εκτέλεσης και να το αναλύσουμε περαιτέρω.

Text

Description automatically generated

Όπως μπορούμε να παρατηρήσουμε η γενικευτική ικανότητα των δύο δικτύων δεν επηρεάζεται δραματικά στην περίπτωση χωρίς επανεκπαίδευση. Ήταν αναμενόμενο η απόδοση να μειωθεί με λιγότερες εισόδους αλλά αυτό δεν επηρέασε δραματικά την γενικευτική ικανότητα του δικτύου συνεπώς μπορούμε ακόμα να το θεωρήσουμε βέλτιστο.

Σαφώς, στην περίπτωση επανεκπαίδευσης θα λάβουμε μεγαλύτερη απόδοση γιατί πλέον το ΤΝΔ του μέρους Α επανεκπαιδεύεται πάνω σε όλα τα δεδομένα εκπαίδευσης τα οποία όλα προσαρμόστηκαν να αντιπροσωπεύουν τις νέες εισόδους. Επομένως, η γενικευτική ικανότητα, φαίνεται να είναι καλύτερη αν το νευρωνικό δίκτυο επανεκπαιδευτεί.

Όσον αφορά την συνολική απόδοση του δικτύου, είναι φαινομενικά καλύτερη η περίπτωση με επανεκπαίδευση έναντι της περίπτωσης χωρίς. Αυτό συμβαίνει επειδή στην περίπτωση με επανεκπαίδευση το ΤΝΔ έχει υψηλότερη πιθανότητα να παρουσιάσει υπερπροσαρμογή στα δεδομένα ελέγχου. Όλος ο ΓΑ δομήθηκε γύρω από την μείωση του αριθμού εισόδων του ΤΝΔ βάσει του σετ ελέγχου και συνεπώς είναι αναμενόμενο ακόμα και μετά την επανεκπαίδευση να αποκρίνεται περίφημα σε αυτό. Γνωρίζουμε πως όσο πιο πολύ ελαττώνουμε τα χαρακτηριστικά τόσο περισσότερο μειώνουμε την πολυπλοκότητα, αλλά αυξάνουμε την πιθανότητα υπερπροσαρμογής (κατάρα της διαστατικότητας). Το ίδιο ισχύει άλλωστε και στην πραγματική ζωή. Περισσότερες πηγές πληροφορίας, ισοδυναμούν με καλύτερη μάθηση και συνεπώς καλύτερη γενίκευση.

Τέλος, αξίζει να αναφερθεί και η σχέση που έχουν τα χαρακτηριστικά που αποκόπτονται με την θέση των pixel στην εικόνα. Ας πάρουμε για παράδειγμα πάλι τον αριθμό 7 που είχαμε στην αρχή

Chart

Description automatically generated with medium confidence

Εφαρμόζοντας τον ΓΑ και ελέγχοντας το αποτέλεσμα σε μορφή εικόνας (μέσω MATLAB) λαμβάνουμε το εξής αποτέλεσμα

Chart

Description automatically generated

Τα χαρακτηριστικά που αποκόπτει ο ΓΑ τείνουν να είναι στην μέση των pixel που παρέχουν την πληροφορία. Για παράδειγμα στην παρακάτω εικόνα μπορούμε να δούμε πως στην περιοχή εντός του κόκκινου κύκλου τα pixels που ήταν στις άκρες πλέον αποκόπτονται. Το ίδιο ισχύει και για τις περισσότερες άκρες του ψηφίου.

Graphical user interface

Description automatically generated with low confidence

Η χρήση ΓΑ, πράγματι φάνηκε αποδοτική και ελάττωσε κατά πολύν τον αριθμό εισόδων. Για το συγκεκριμένο πρόβλημα ίσως να είναι καλύτερη στρατηγική να μην επανεκπαιδεύσουμε το ΤΝΔ, εκτός και αν θέλουμε να ασχοληθούμε μόνο με το σετ ελέγχου. Ακόμα και χωρίς επανεκπαίδευση, η μείωση εισόδων οδηγεί σε μείωση πολυπλοκότητας. Αν η εικόνα ήταν μεγαλύτερων διαστάσεων (περισσότερα pixels), η μείωση της πολυπλοκότητας θα είχε πολύ μεγάλη σημασία, καθώς επιτυγχάνουμε μία μέση λύση μεταξύ απόδοσης και ταχύτητας.