



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΚΑΙ ΕΥΦΥΗ
ΥΠΟΛΟΓΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

3^η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

Αυτο-οργανούμενοι χάρτες

ΟΜΑΔΑ ΕΡΓΑΣΙΑΣ:

ΑΡΓΥΡΙΟΥ ΣΤΕΦΑΝΟΣ – Α.Μ.: 03112006

ΒΑΜΒΟΥΚΑΚΗΣ ΓΕΩΡΓΙΟΣ – Α.Μ.: 03112906

1. Υλοποίηση ενός SOM

Σε πρώτη φάση, σκοπός της παρούσας εργαστηριακής άσκησης είναι κυρίως η υλοποίησης ενός –απλού μεν, αρκετά γενικού δε– αυτο-οργανούμενου χάρτη. Ακολουθώντας τις διεξοδικές προδιαγραφές που αναφέρονται στην εκφώνηση της παρούσας άσκησης, διαμορφώσαμε τον κώδικα που υπολειπόταν ώστε να υλοποιήσουμε ένα SOM με μονοδιάστατο ή δισδιάστατο πλέγμα νευρώνων καθώς και διάφορες επιλογές ως προς τον τύπο πλέγματος και τοπολογικής γειτονιάς. Ο αλγόριθμος περιλαμβάνει τόσο το στάδιο διάταξης (ordering), όσο και το στάδιο ρύθμισης (tuning), καθένα εκ των οποίων υποδιαιρείται στις τρεις φάσεις ανταγωνισμού, συνεργασίας και ανταμοιβής. Οι συναρτήσεις **somCreate.m** και **somTrainParameters.m** μας δίνονται ολοκληρωμένες και εμείς υλοποιήσαμε κατά αντιστοιχία με αυτές τις δικές μας συναρτήσεις **somActivation.m**, **somOutput.m**, **somTrain.m** και **somUpdate.m** οι οποίες εσωκλείονται στο **.zip** αρχείο της παρούσας εργασίας και παρουσιάζονται ακολούθως:

```
function [ a ] = somActivation(pattern, neighborDist)
    global distances Winners;

    fire = find(somOutput(pattern)); % Find winner neuron (value = 1)
    Winners(fire) = Winners(fire) + 1 ;

    a = distances(fire, :) <= neighborDist;
    %Set a = 1 within neighborhood, a = 0 otherwise

    a = a .* 0.5;      %For neurons in neighborhood a = 0.5
    a(fire) = 1;      %For winner neuron a = 1
```

```
function [ output ] = somOutput(pattern)
    global IW;

    % IW -> Nx D
    % pattern -> Dx 1, contains D-characteristics of a single pattern
    % output -> Nx 1, contains output value of each neuron

    output = compet(negdist(IW, pattern));

    %negdist -> compute negative euclidean distance
    %compet -> find winner neuron and set its value 1
```

```
function somTrain(patterns)
```

```

% patterns -> DxP, each column is a pattern x
global maxNeighborDist tuneND orderLR orderSteps tuneLR P;

% Ordering Phase %

%Create matrices which contain exponentially declining values
from:
% i) maxNeighborDist to TuneND
% ii) orderLR to tuneLR
OrderingND = linspace(maxNeighborDist,tuneND,orderSteps);
OrderingLR = linspace(orderLR,tuneLR,orderSteps);

for i=1:orderSteps
    neighborDist = OrderingND(i);
    learningRate = OrderingLR(i);

    %Train for each of P - patterns
    for j=1:P
        somUpdate(patterns(:,j), learningRate, neighborDist);
    end
end

% Tuning Phase %

neighborDist = tuneND ;
learningRate = tuneLR ;

coeff = 4; %coefficient values range from 2 to 5
tuningSteps = orderSteps * coeff;

for i=1:tuningSteps

    %Train for each of P - patterns
    for j=1:P
        somUpdate(patterns(:,j), learningRate, neighborDist);
    end
end

function somUpdate(pattern, learningRate, neighborDist)
    global N IW;

    a = somActivation(pattern, neighborDist); %compute a using
    somActivation

    %update weights using:  $W_{new} = W_{old} + LR * a * (X_{pattern} - W_{old})$ 
    for j=1:N
        IW(j,:) = IW(j,:) + learningRate * a(1,j) * (pattern' -
IW(j,:));
    end
end

```

2. Μελέτη και ανάλυση SOM

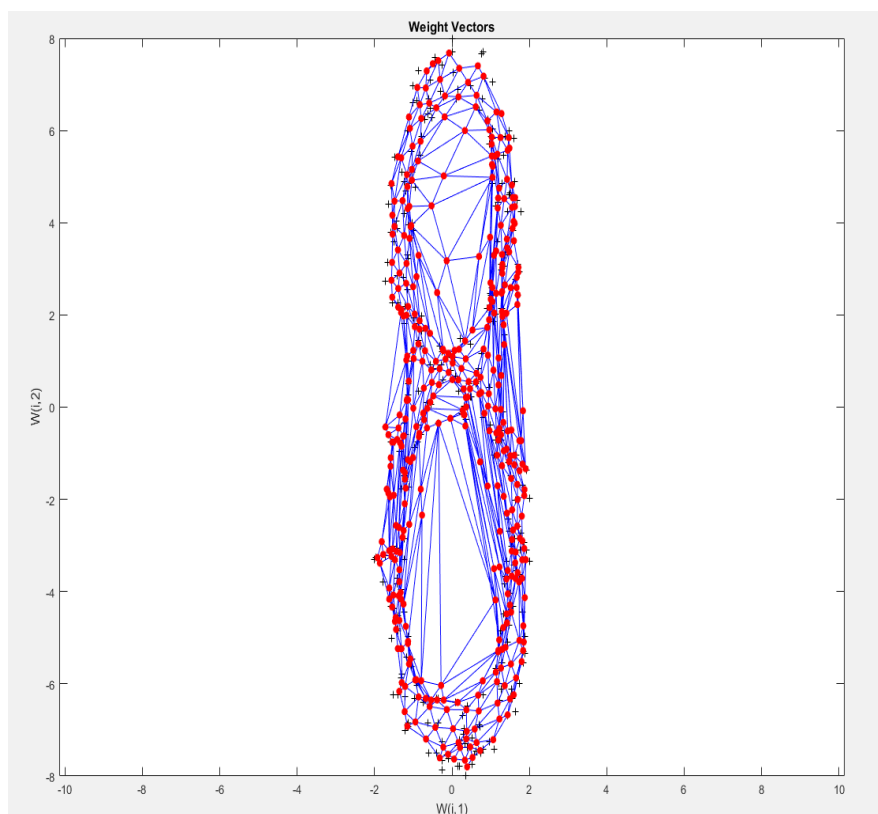
Ερώτημα 2Α

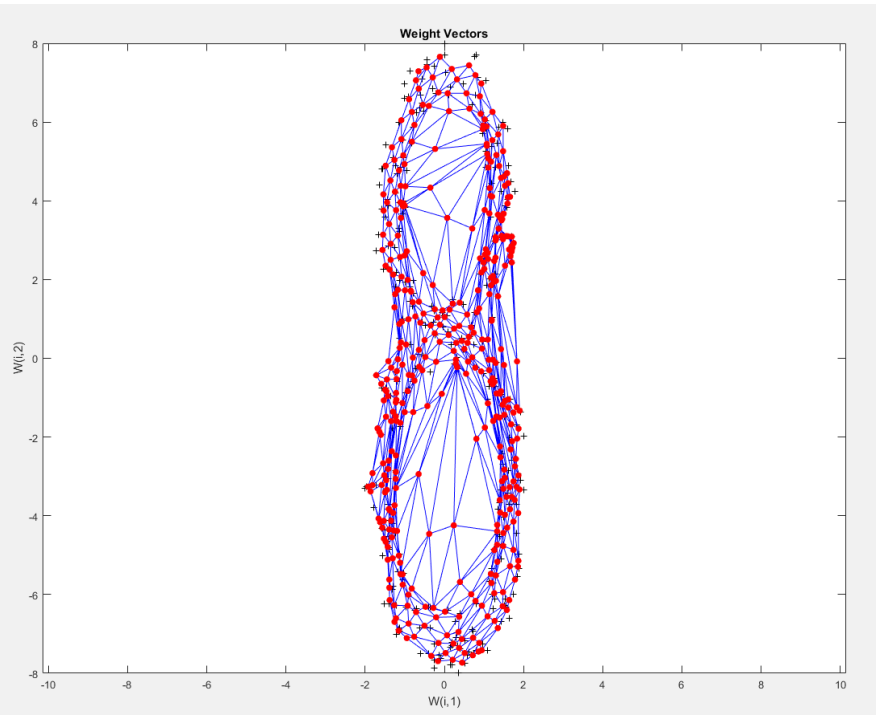
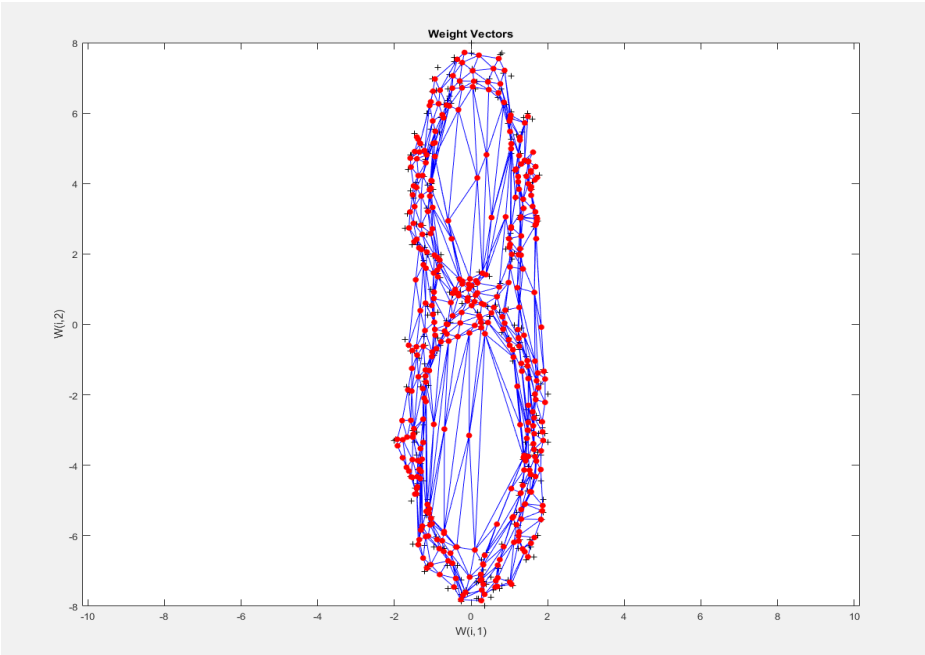
Αφού υλοποιήσαμε το SOM, επιχειρήσαμε τη μελέτη και ανάλυση των ιδιοτήτων, των δυνατοτήτων και των επιδόσεων του αυτο-οργανούμενου χάρτη. Για τα τρία σύνολα προτύπων που μας δίνονται, εκτελέσαμε scripts που κατασκευάζουν μονοδιάστατα και δισδιάστατα πλέγματα που εκπαιδεύονται με βάση αυτά τα πρότυπα. Για να τρέξουμε μαζικά τα τρία πρότυπα για όλους τους τύπους πλεγμάτων και όλα τα μέτρα απόστασης δημιουργήσαμε το αρχείο ScriptA.m που επισυνάπτεται.

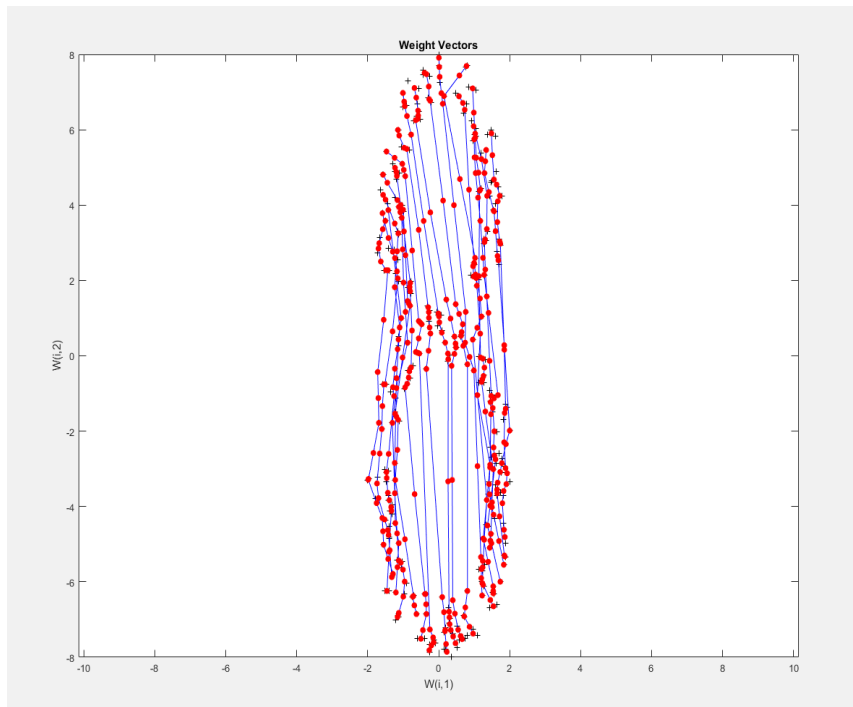
Eight Patterns

Για το συγκεκριμένο σύνολο που αποτελείται από 300 δεδομένα που απεικονίζουμε τον αριθμό 8, χρησιμοποιήσαμε δισδιάστατο πλέγμα, αρχικώς 20x20 νευρώνων και πειραματιστήκαμε για όλα τα μέτρα απόστασης και πλεγμάτων. Τα αποτελέσματα παρουσιάζονται ακολούθως:

- *gridtop*

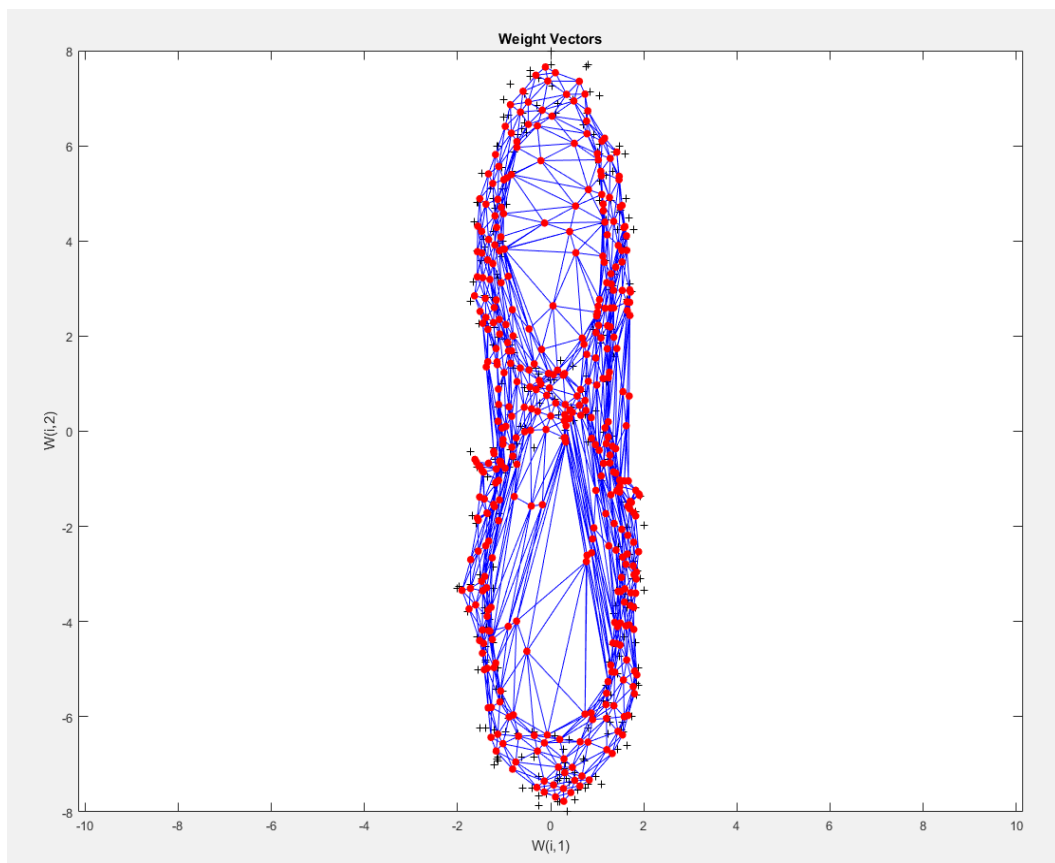


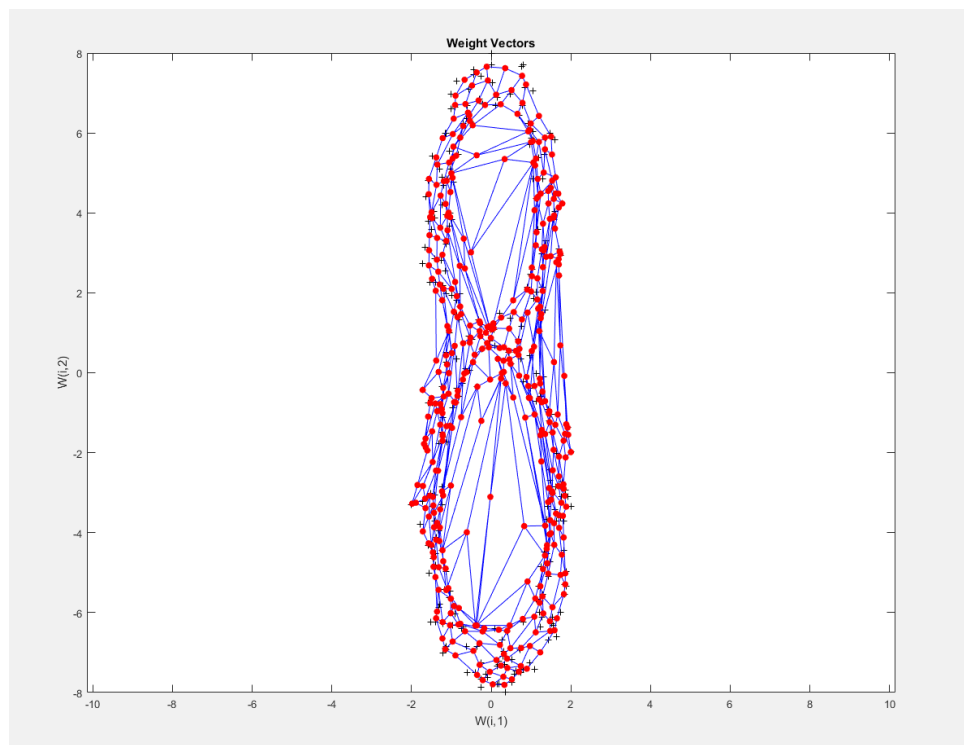
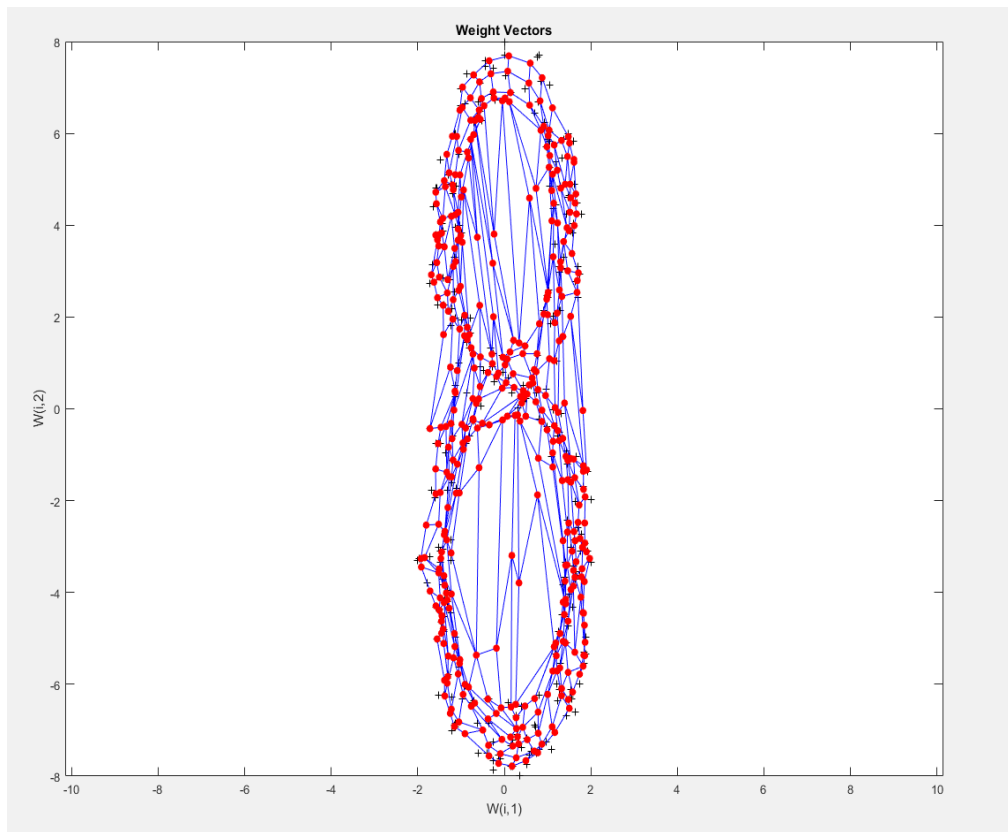


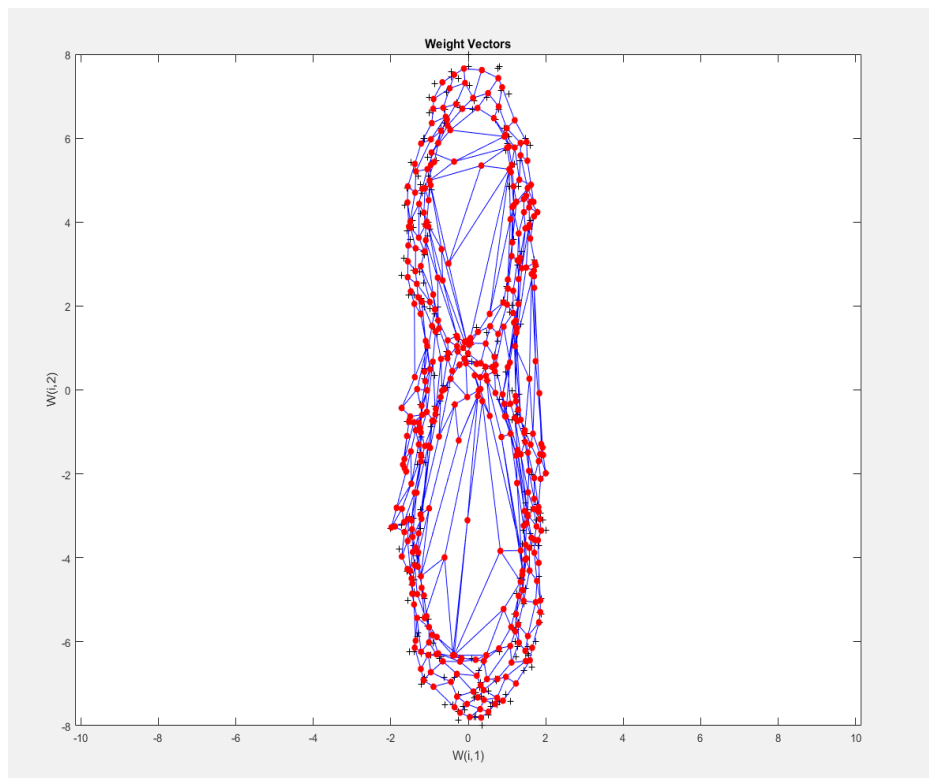


Παρατηρούμε πως την καλύτερη συμπεριφορά για τον τύπο πλέγματος gridtop την παρουσιάζουν τα μέτρα απόστασης mandist και linkdist (1η και 2η εικόνα αντίστοιχα).

- *hextop*

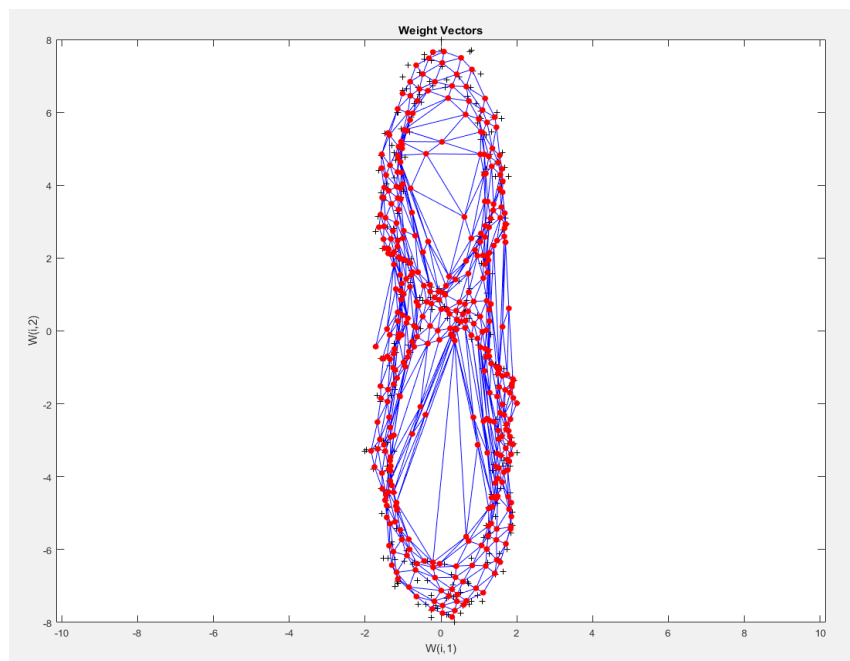


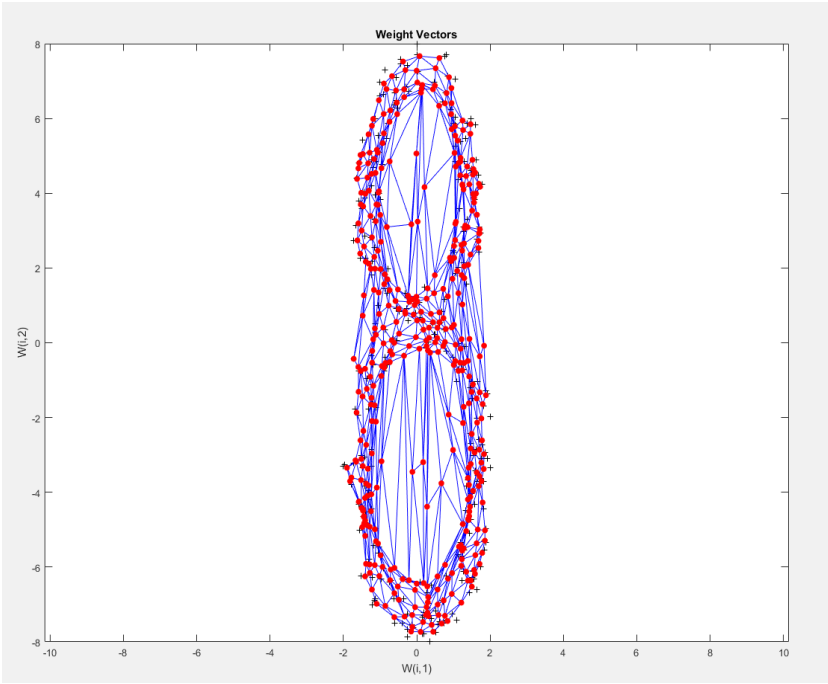
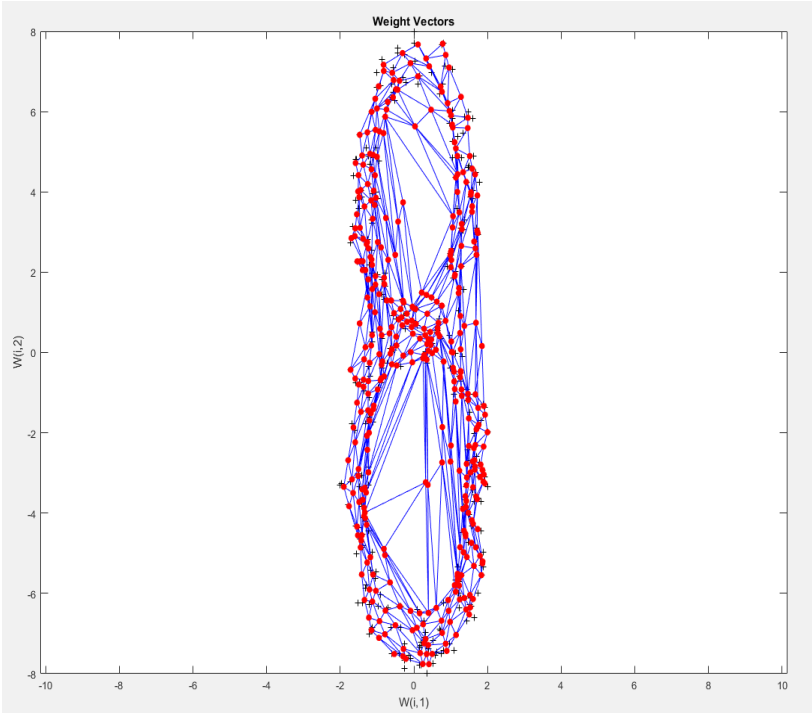


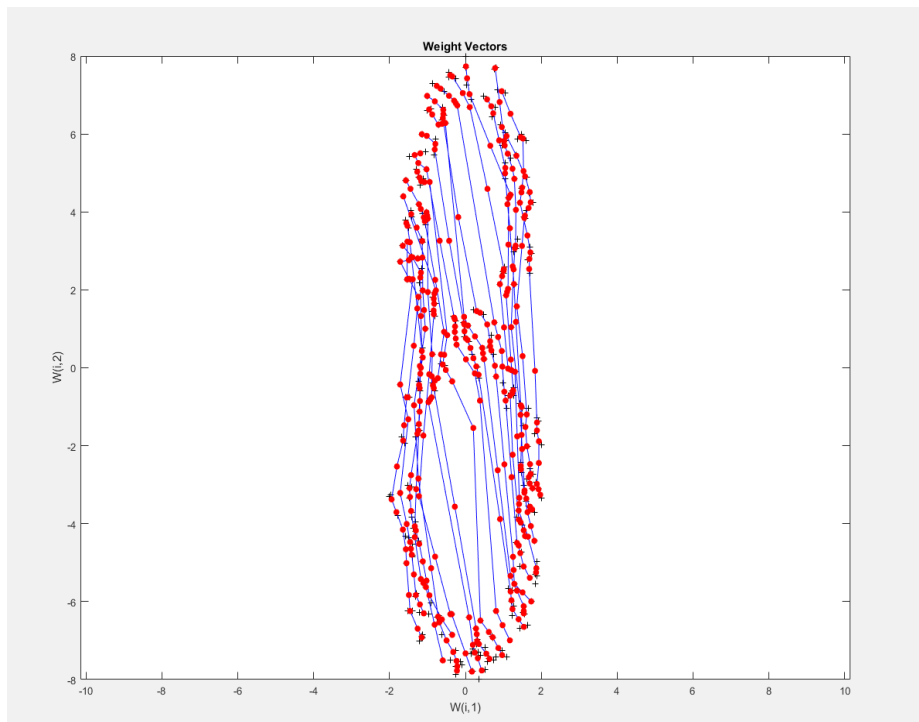


Παρατηρούμε πως την καλύτερη συμπεριφορά για τον τύπο πλέγματος hextor την παρουσιάζει το μέτρο απόστασης dist (4η εικόνα).

- *hexagonal topology*

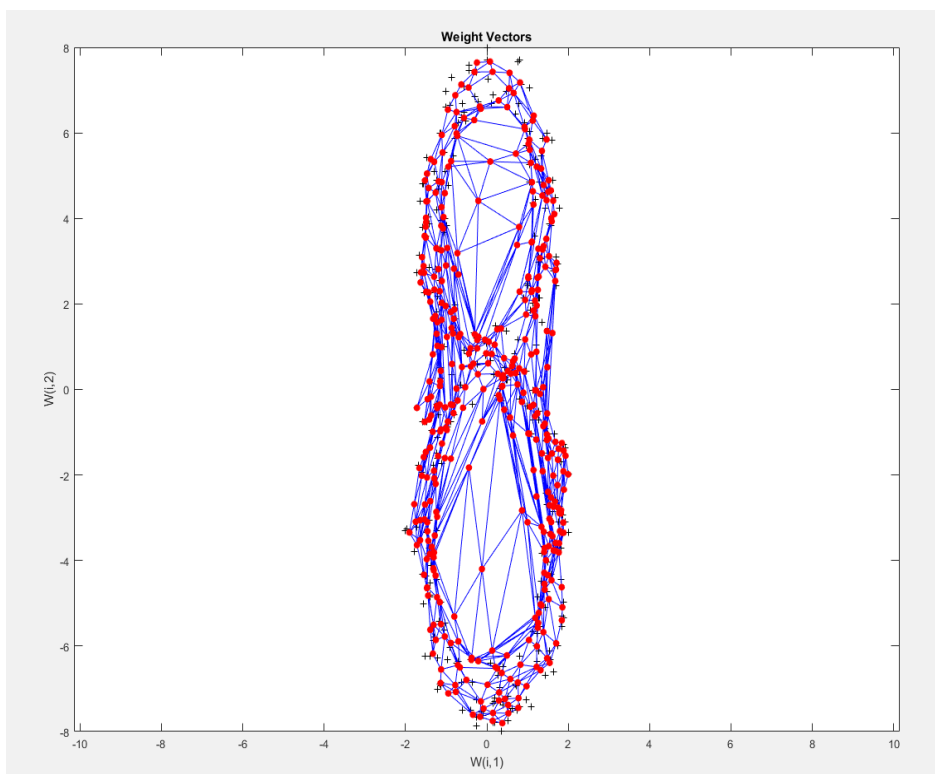


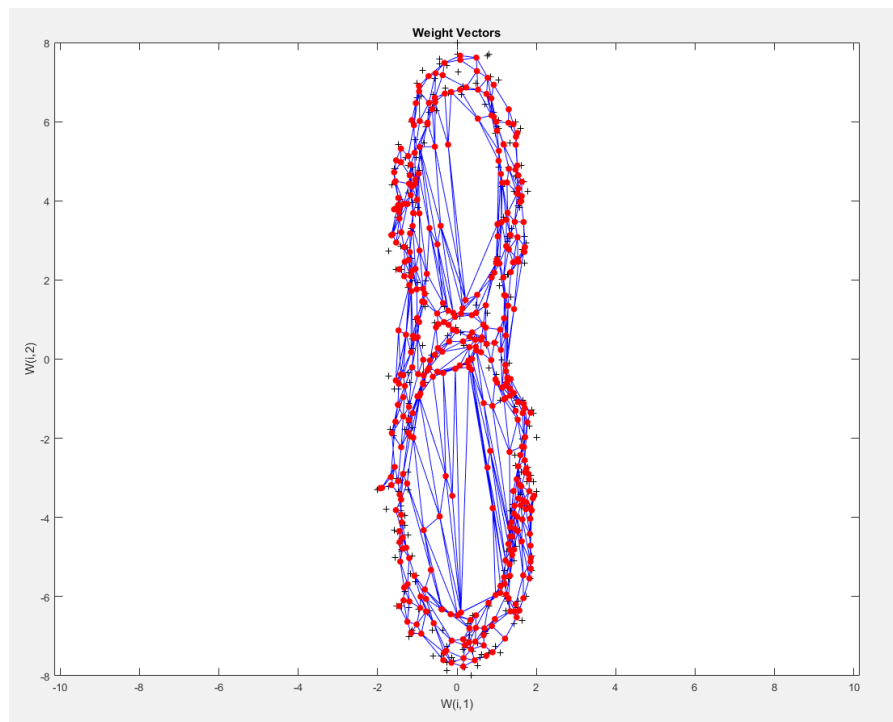
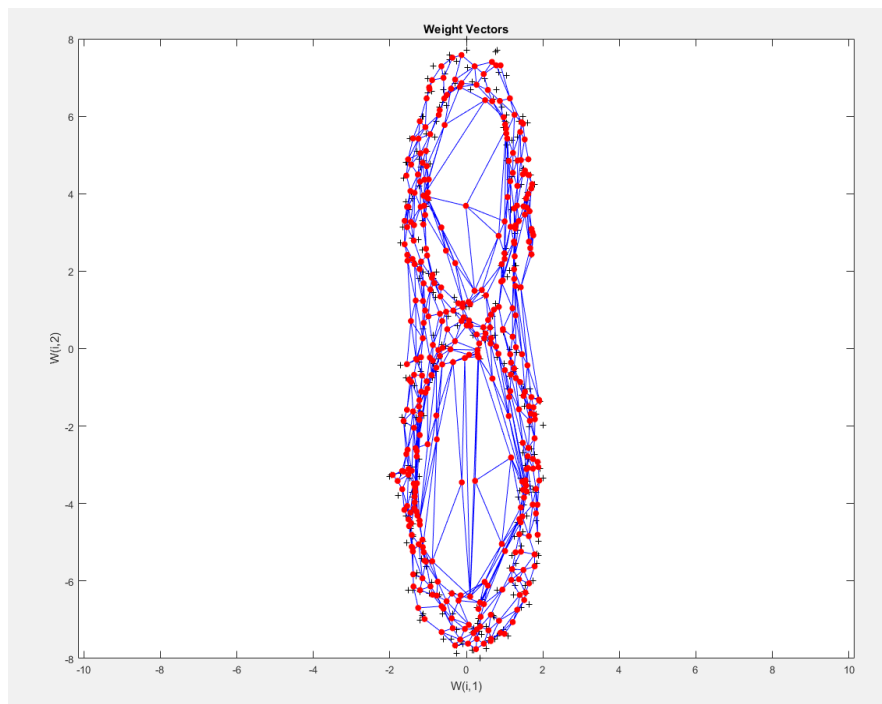


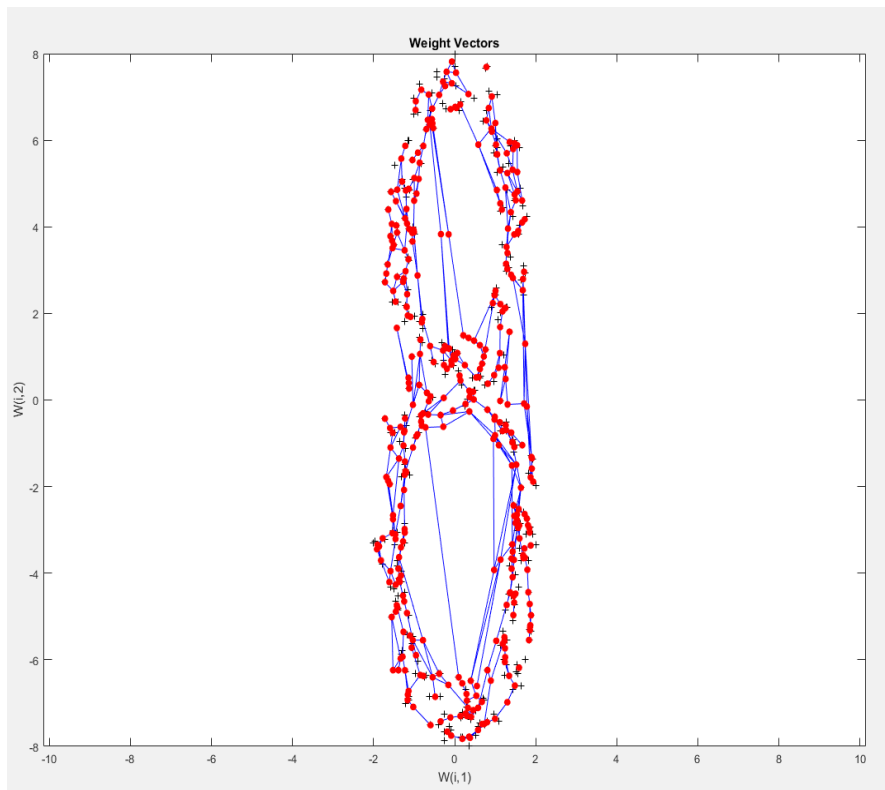


Παρατηρούμε πως την καλύτερη συμπεριφορά για τον τύπο πλέγματος hexagonal topology την παρουσιάζει το μέτρο απόστασης mandist (1η εικόνα).

- *randtop*







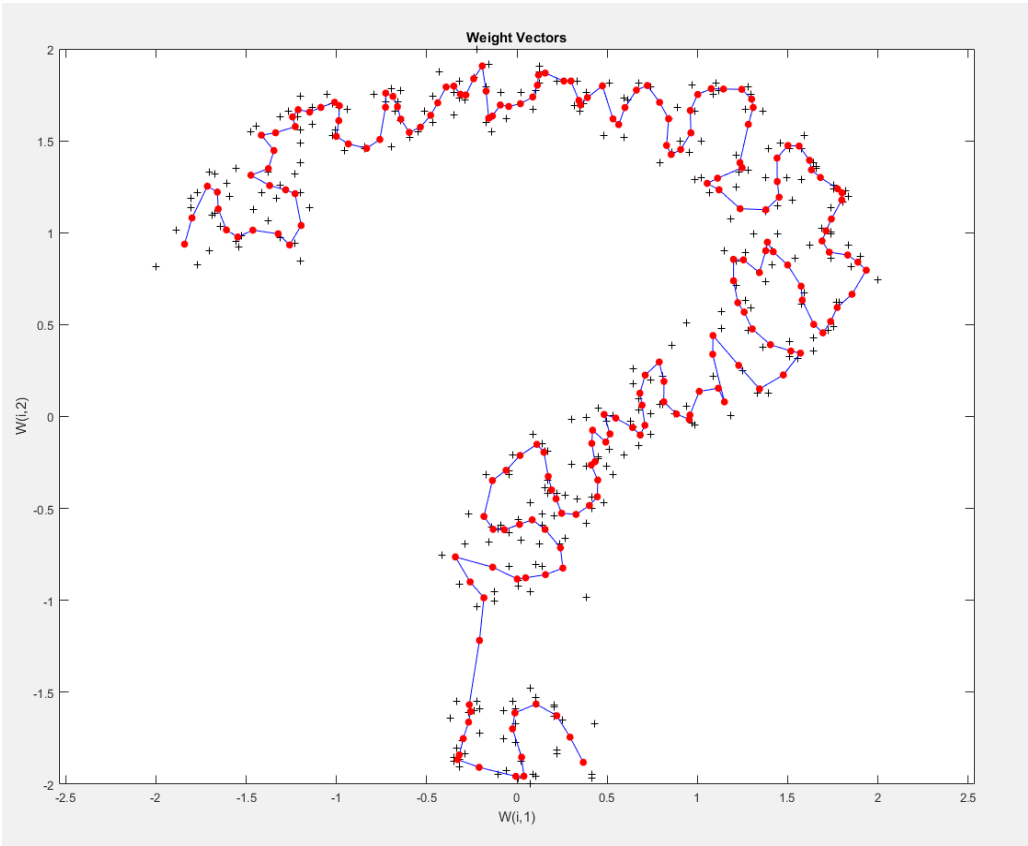
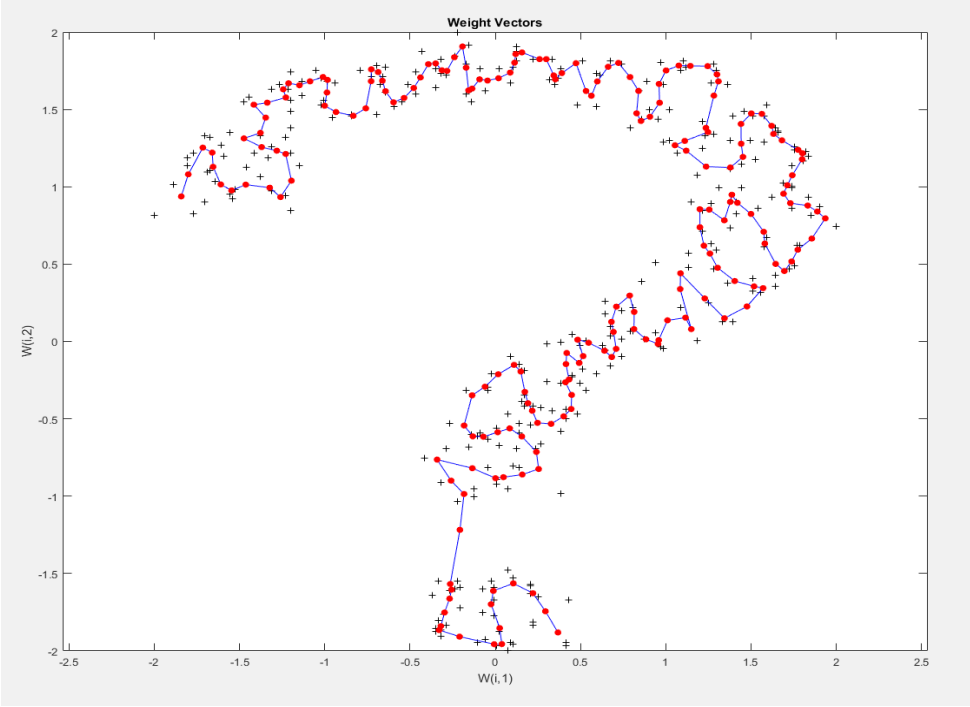
Παρατηρούμε πως την καλύτερη συμπεριφορά για τον τύπο πλέγματος `randtop` την παρουσιάζει το μέτρο απόστασης `mandist`.

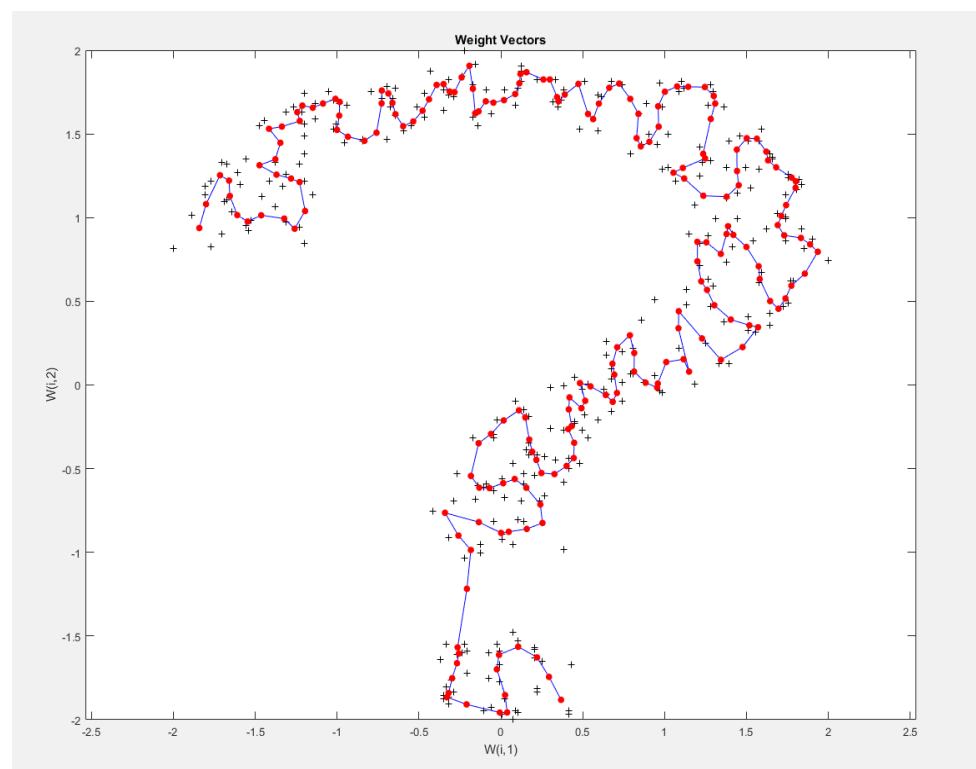
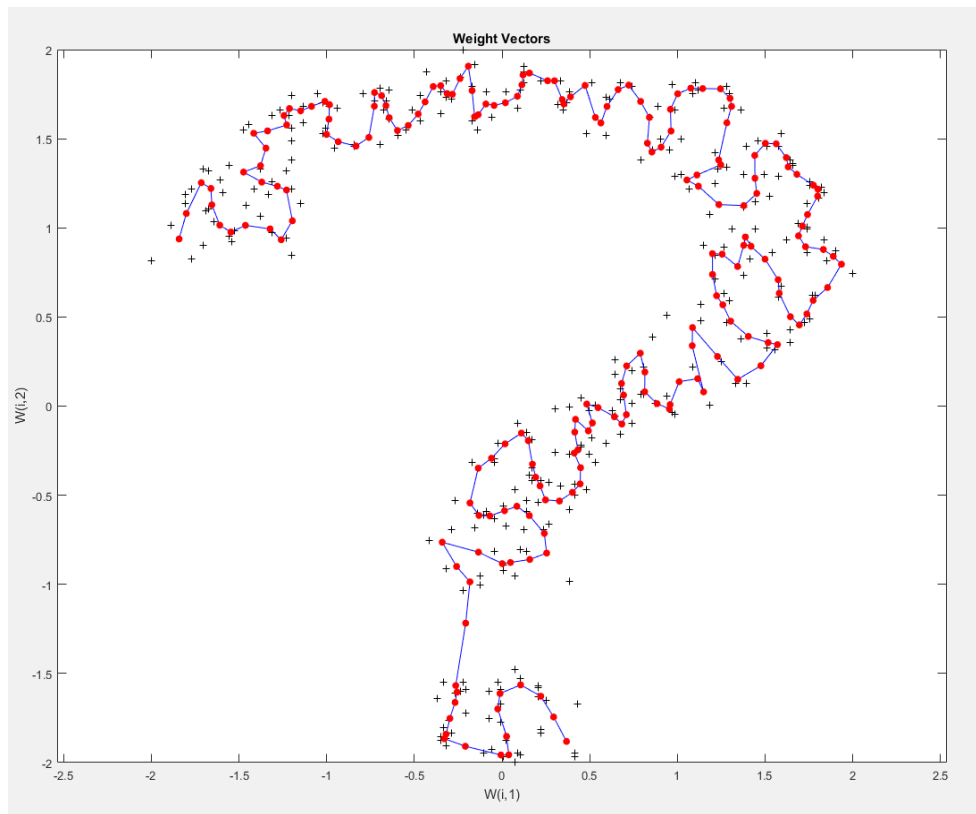
Συνοψίζοντας τα παραπάνω αποτελέσματα θεωρούμε πως αποτελεσματικότερος αυτο-οργανούμενος χάρτης είναι αυτός με τύπο πλέγματος `gridtop` και μέτρο απόστασης `mandist`.

Question Patterns

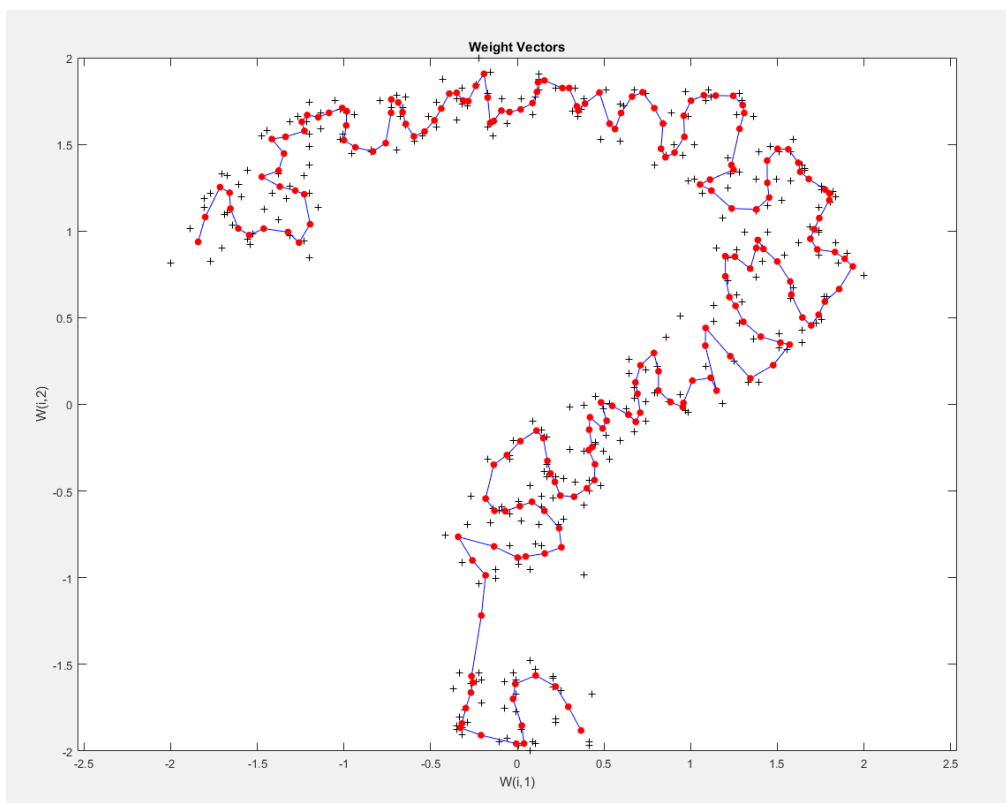
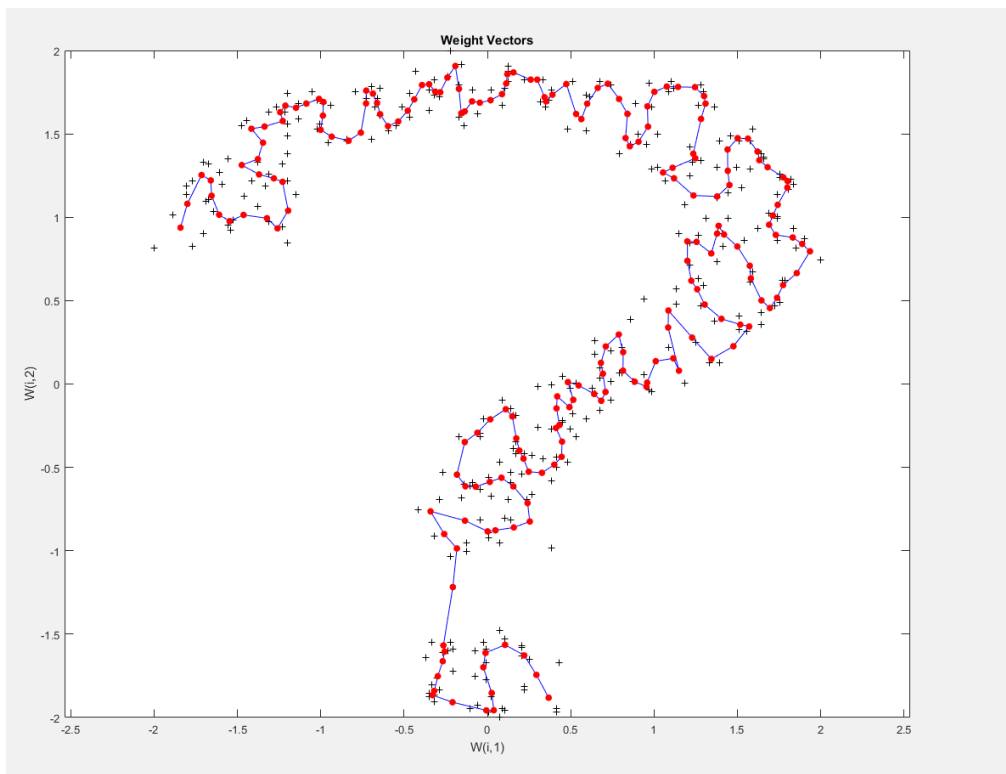
Για το συγκεκριμένο σύνολο που αποτελείται από 300 δεδομένα σε μορφή αγγλικού ερωτηματικού, χρησιμοποιήσαμε μονοδιάστατο πλέγμα, αρχικώς 200 νευρώνων και πειραματιστήκαμε για όλα τα μέτρα απόστασης και πλεγμάτων. Τα αποτελέσματα παρουσιάζονται ακολούθως:

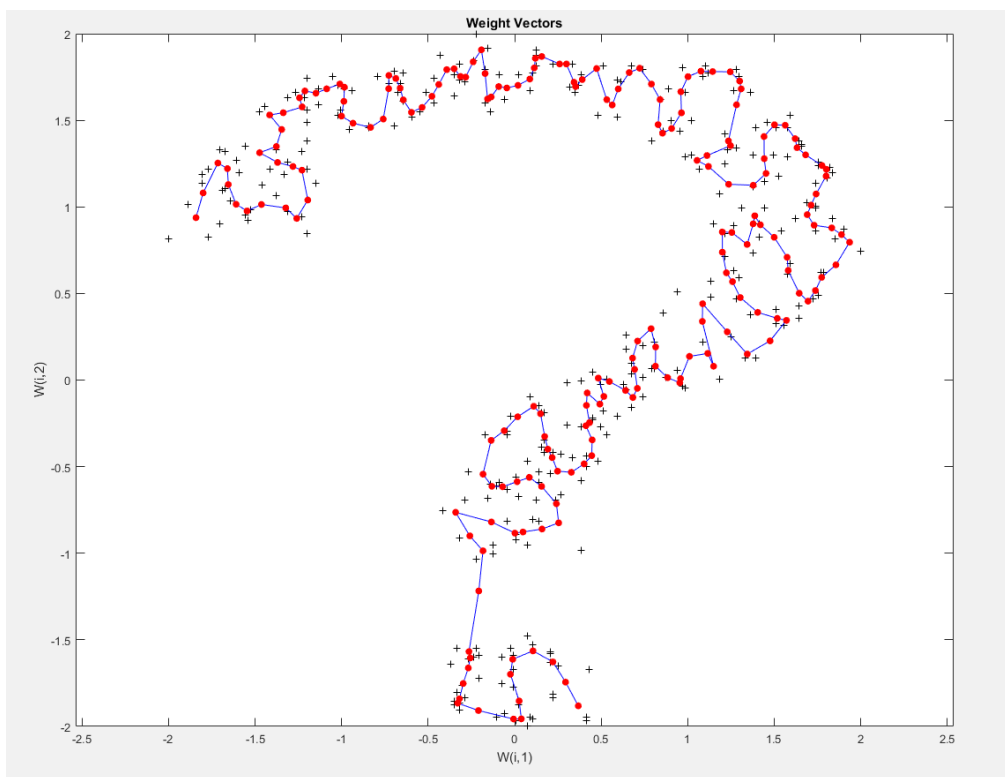
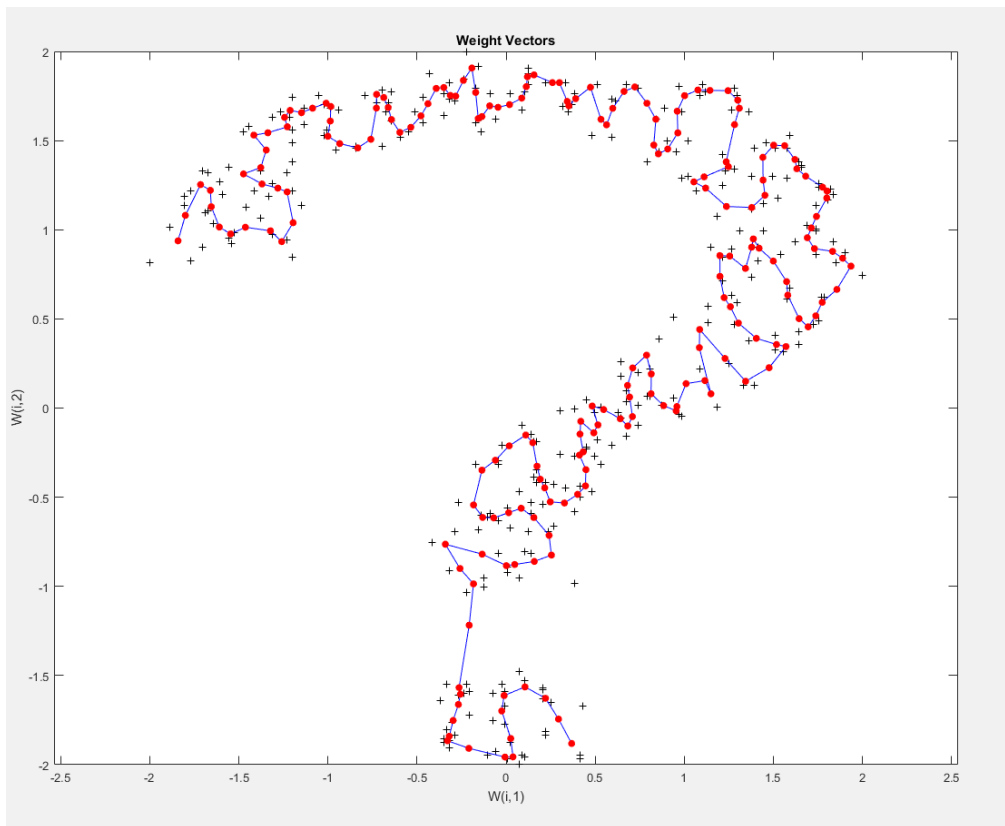
- *gridtop*



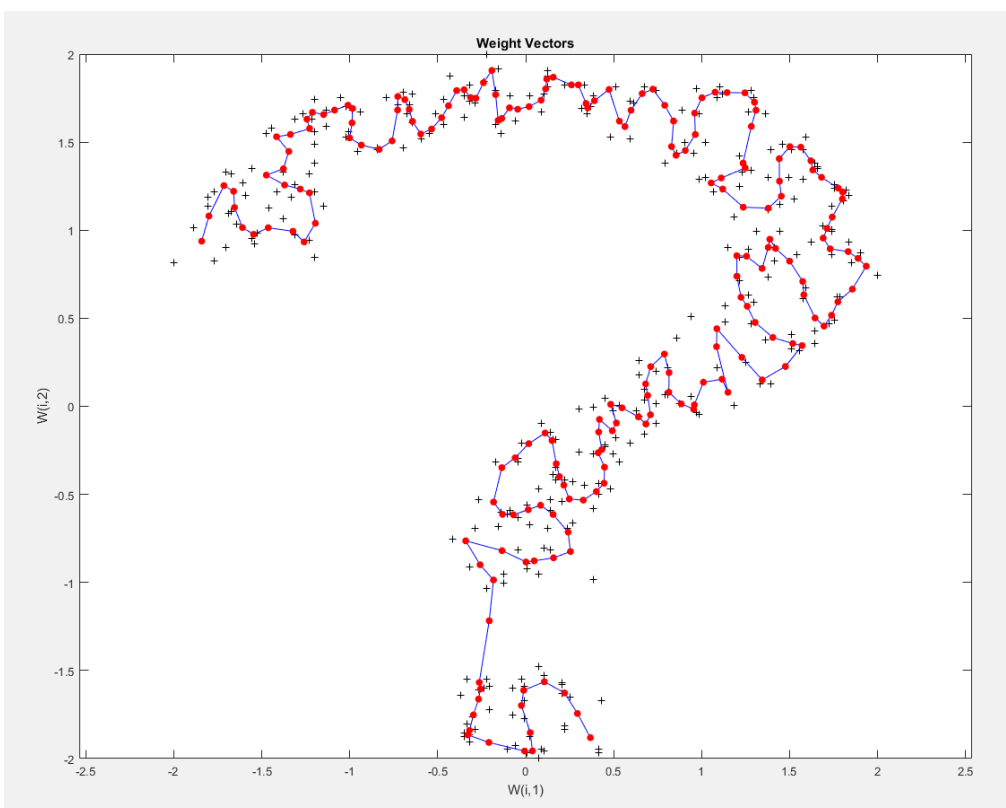
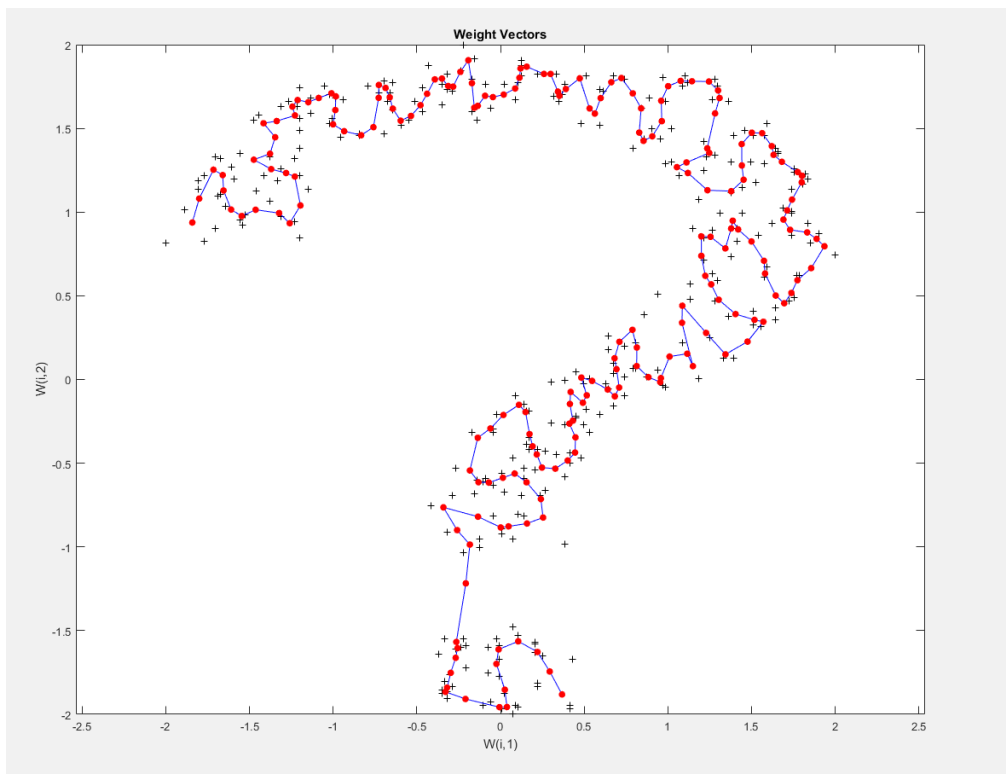


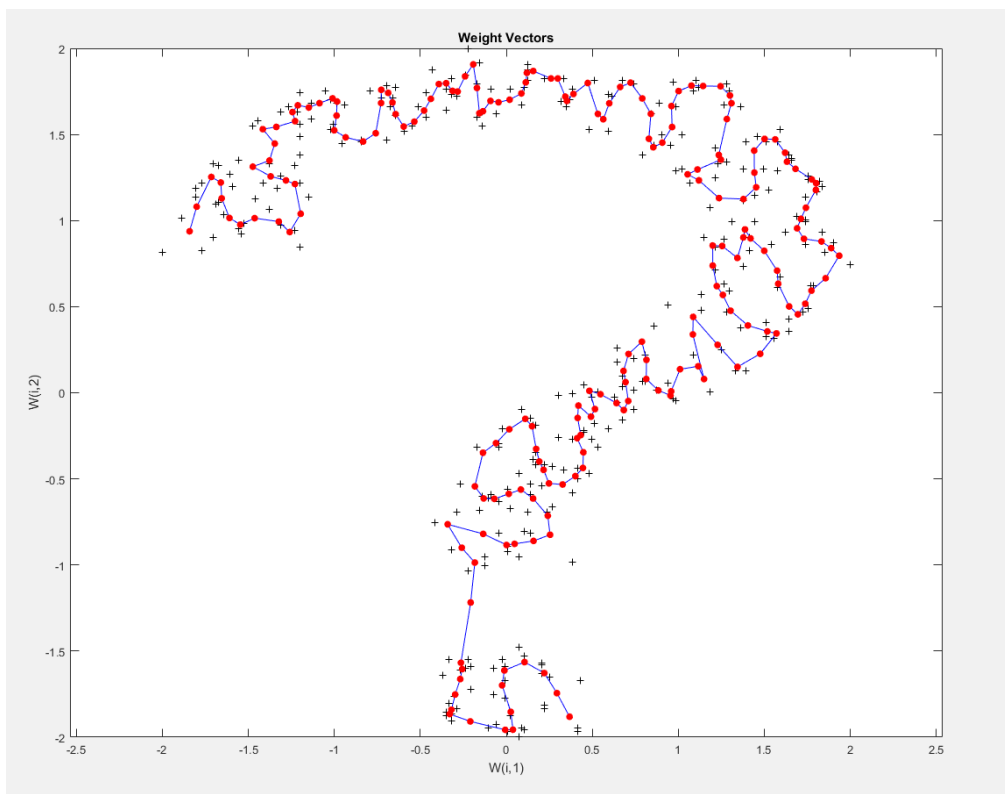
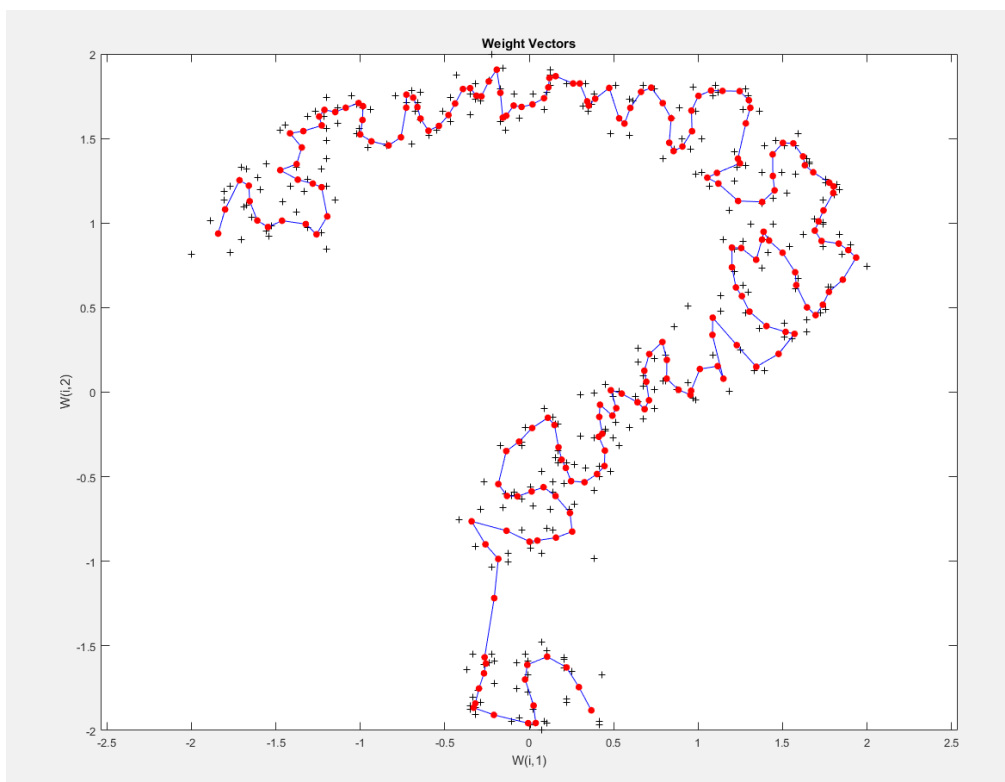
- *hextop*





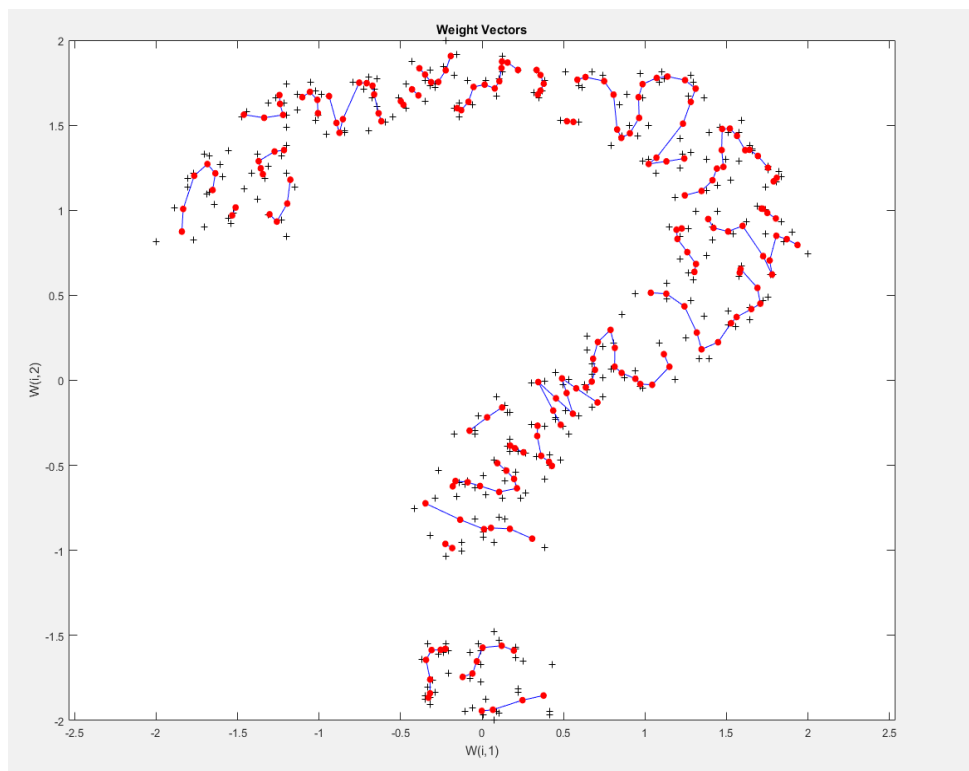
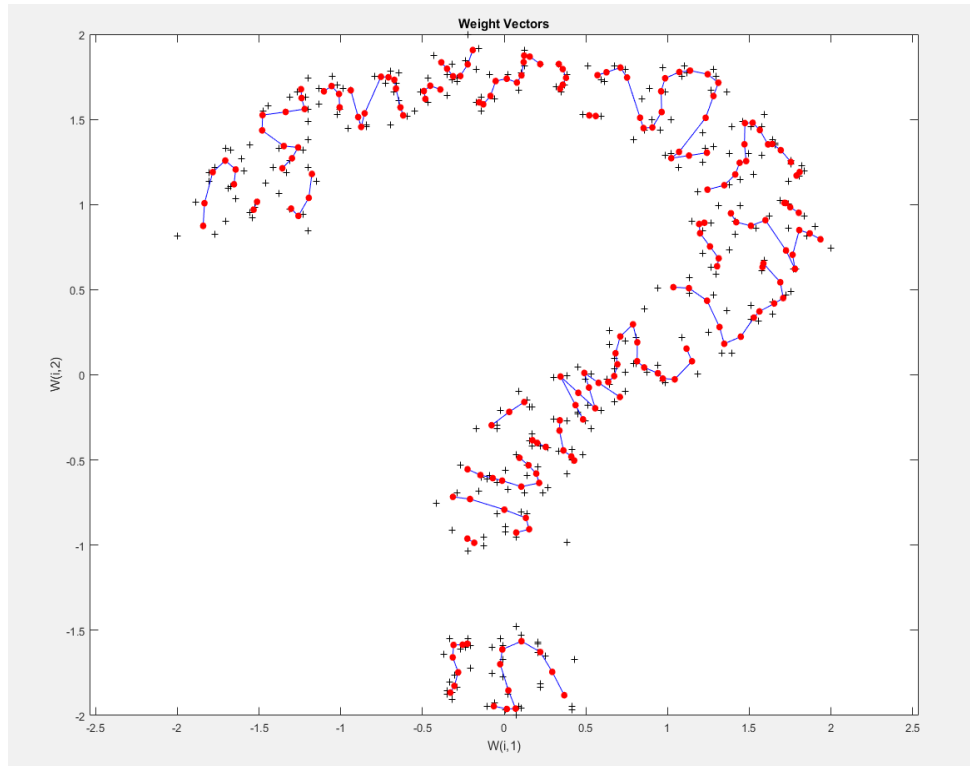
- *hexagonal topology*

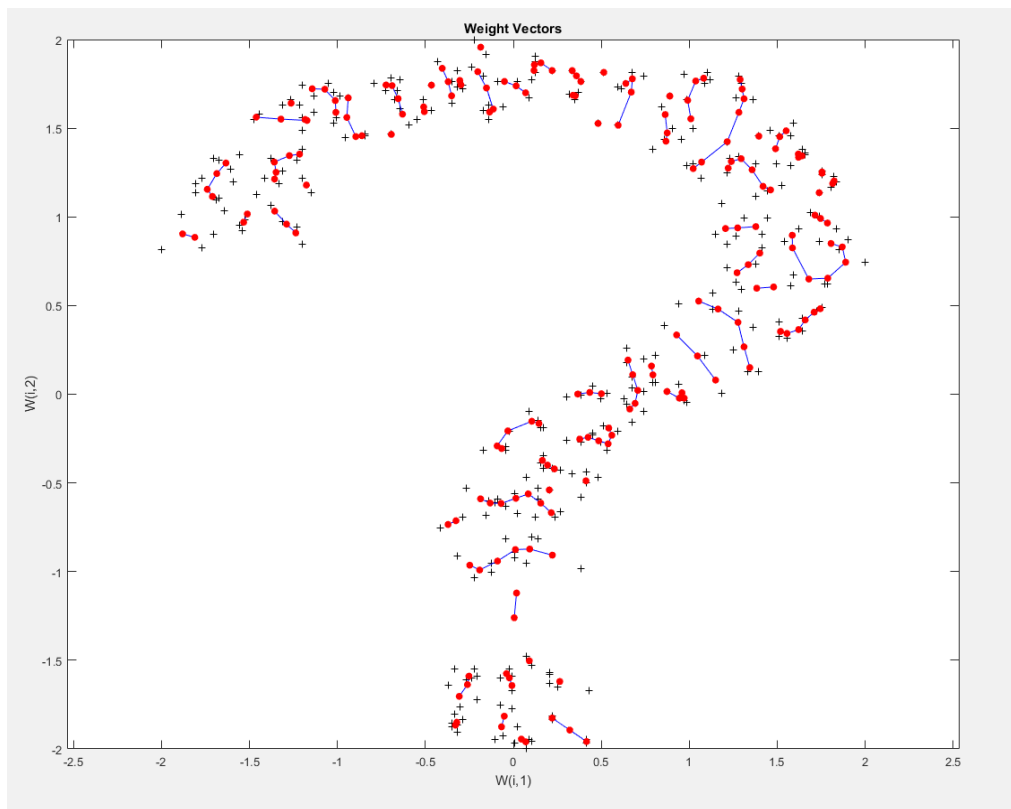
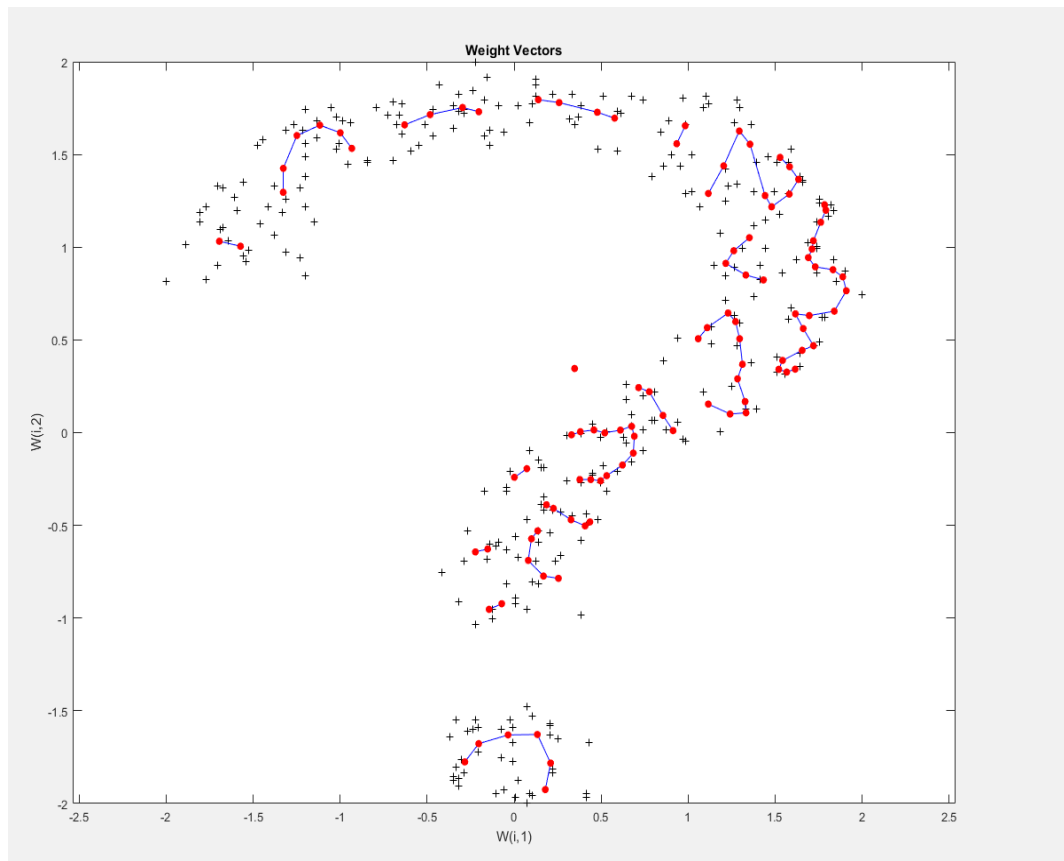




Όπως γίνεται εμφανές, όλοι οι παραπάνω τύποι πλέγματος μας οδηγούν στο ίδιο ακριβώς αποτέλεσμα ανεξαρτήτως του μέτρου αποστάσεως που χρησιμοποιείται.

- *randtop*



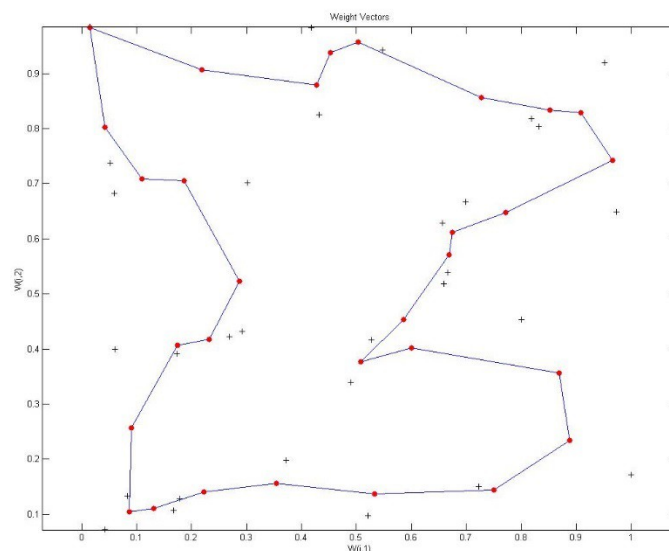


Παρατηρούμε πως την καλύτερη συμπεριφορά για τον τύπο πλέγματος randtor την παρουσιάζει το μέτρο απόστασης mandist (1η εικόνα).

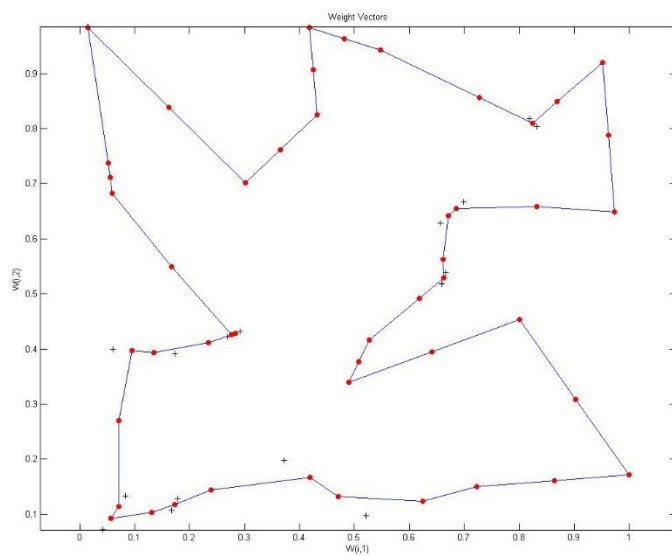
Συνοψίζοντας τα παραπάνω αποτελέσματα θεωρούμε πως αποτελεσματικότερος αυτο-οργανούμενος χάρτης είναι αυτός με τύπο πλέγματος randtor και μέτρο απόστασης mandist.

Ερώτημα 2B

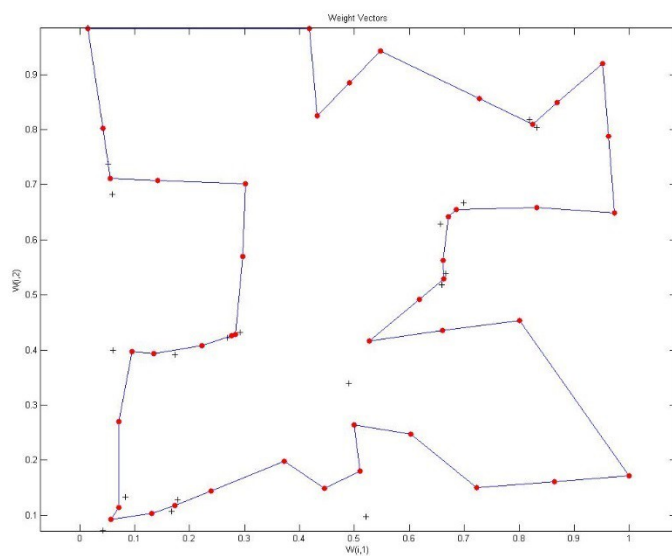
Μια ειδική εφαρμογή του αυτο-οργανούμενου χάρτη είναι η επίλυση του προβλήματος του περιοδεύοντος πωλητή (Travelling Salesman Problem), το οποίο ανήκει στην κλάση των υπολογιστικά δύσκολων προβλημάτων (NP-complete) συνδυαστικής βελτιστοποίησης. Θεωρούμε N πόλεις τοποθετημένες σε ένα δισδιάστατο επίπεδο, με δεδομένες τις μεταξύ τους αποστάσεις. Ζητείται να βρεθεί η συντομότερη κλειστή διαδρομή που επισκέπτεται όλες τις πόλεις από μια μόνο φορά. Έτσι, κατασκευάζουμε ένα δίκτυο Kohonen με εισόδους τις συντεταγμένες των πόλεων και έξοδο αποτελούμενη από N κόμβους σε σύνδεση δακτυλίου. Επιλέχθηκε μονοδιάστατο πλέγμα με μέτρο απόστασης linkdist, το οποίο μέσω της συνάρτησης `ring_distances` μετατράπηκε σε μορφή δακτυλίου. Με πειραματισμούς στο πλήθος των νευρώνων και τον αριθμό των εποχών, τα αποτελέσματα προκύπτουν ακολούθως και λήφθηκαν χρησιμοποιώντας το `ScriptB.m` με κατάλληλη επιλογή δεδομένων `DataPatterns`:



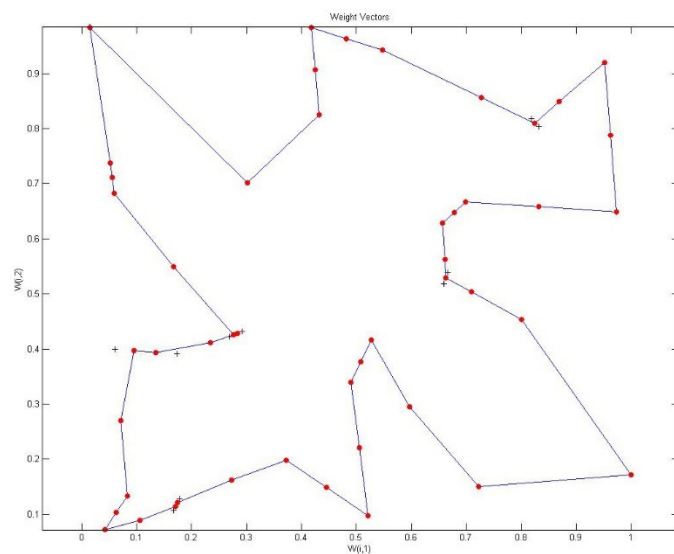
30 νευρώνες – 200 εποχές



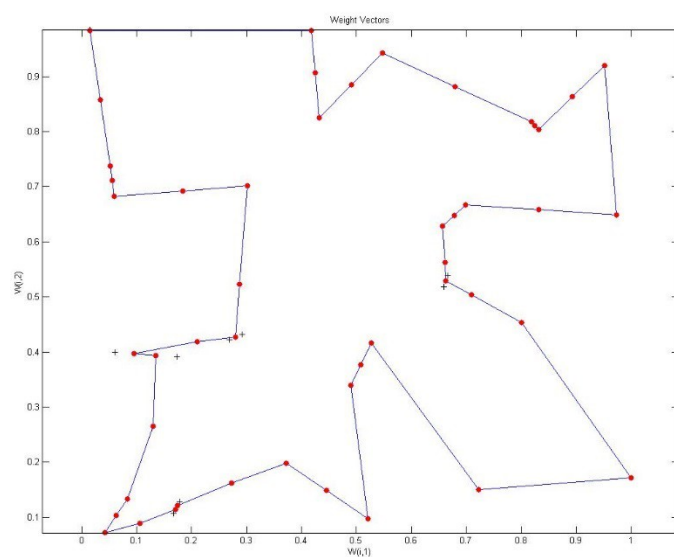
50 νευρώνες – 200 εποχές



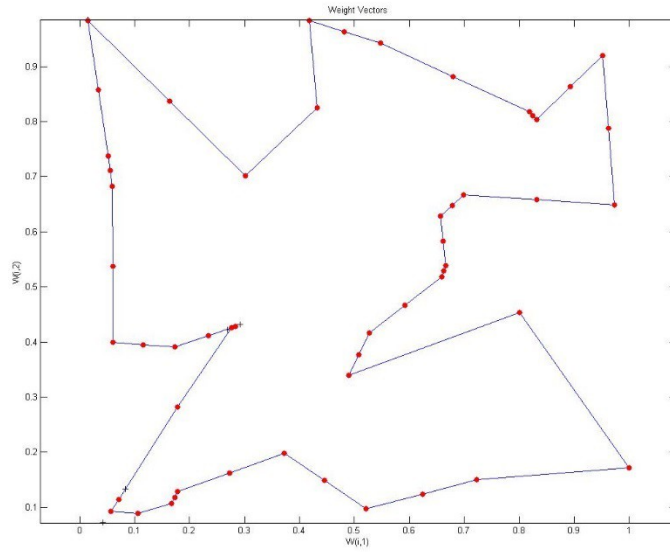
50 νευρώνες – 600 εποχές



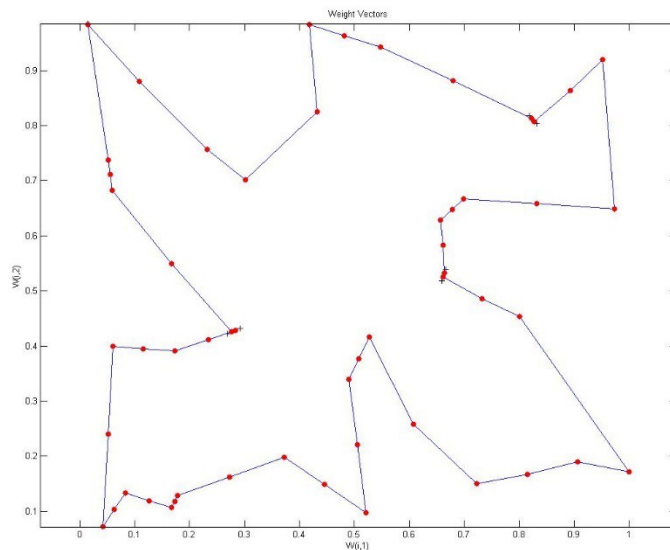
60 νευρώνες – 200 εποχές



60 νευρώνες – 500 εποχές



65 νευρώνες – 500 εποχές



70 νευρώνες – 200 εποχές

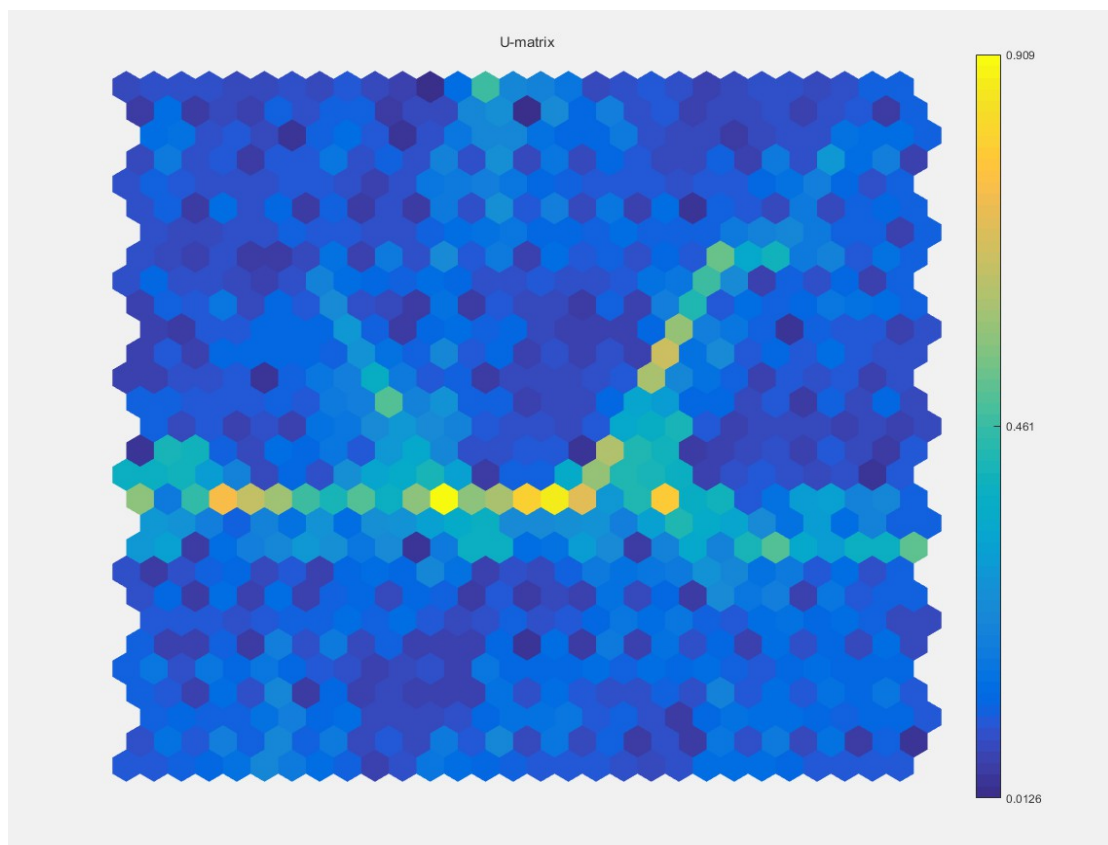
Παρατηρούμε πως ακόμα και αν έχουμε να απεικονίσουμε μόλις τριάντα πόλεις, ο αριθμός των νευρώνων που χρειάζεται είναι τουλάχιστον ο διπλάσιος. Από εκεί και πέρα, επαρκή αναπαράσταση προσφέρουν οι 65 νευρώνες με μια εκπαίδευση 500 εποχών, ωστόσο έχουμε ακόμα καλύτερο αποτέλεσμα με 70 νευρώνες και 200 εποχές. Αυτό σημαίνει πως μια μικρή αύξηση του αριθμού των νευρώνων μας δίνει καλύτερο αποτέλεσμα και μάλιστα σε αρκετά μικρό αριθμό εποχών, μειώνοντας τελικά το υπολογιστικό κόστος.

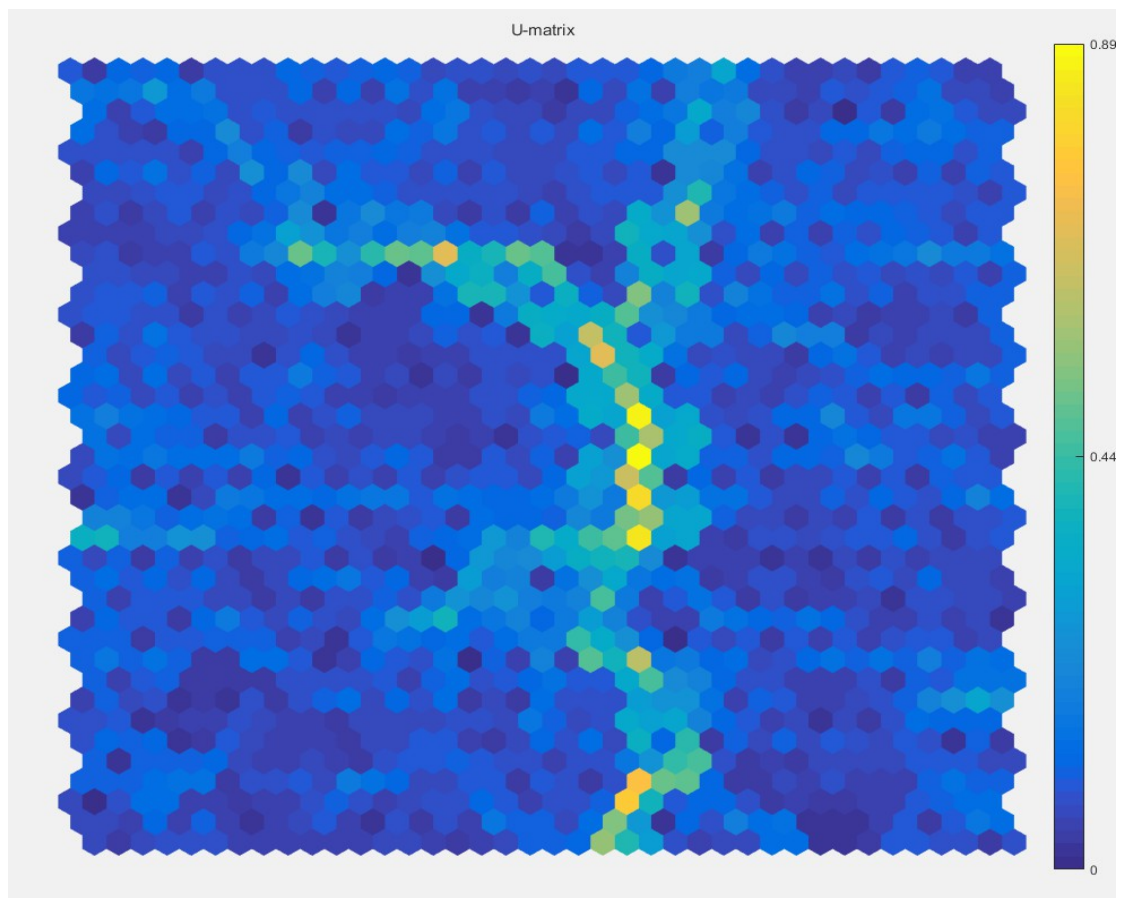
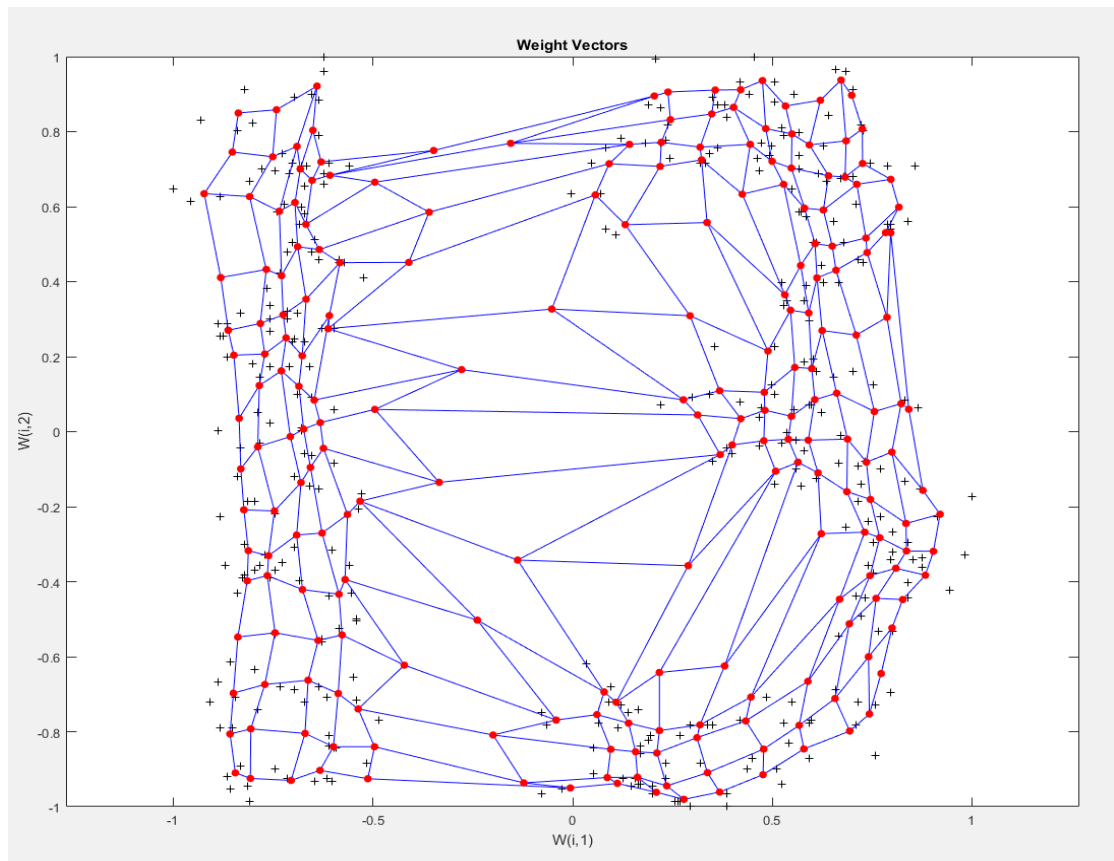
Ερώτημα 2Γ

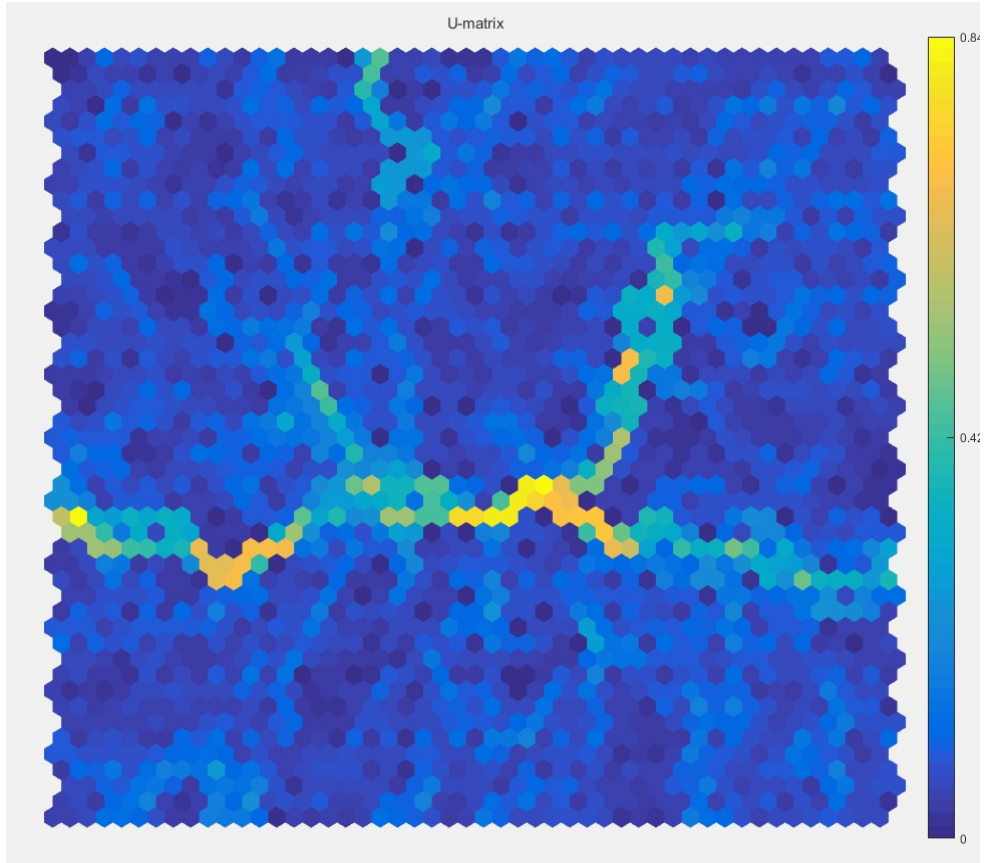
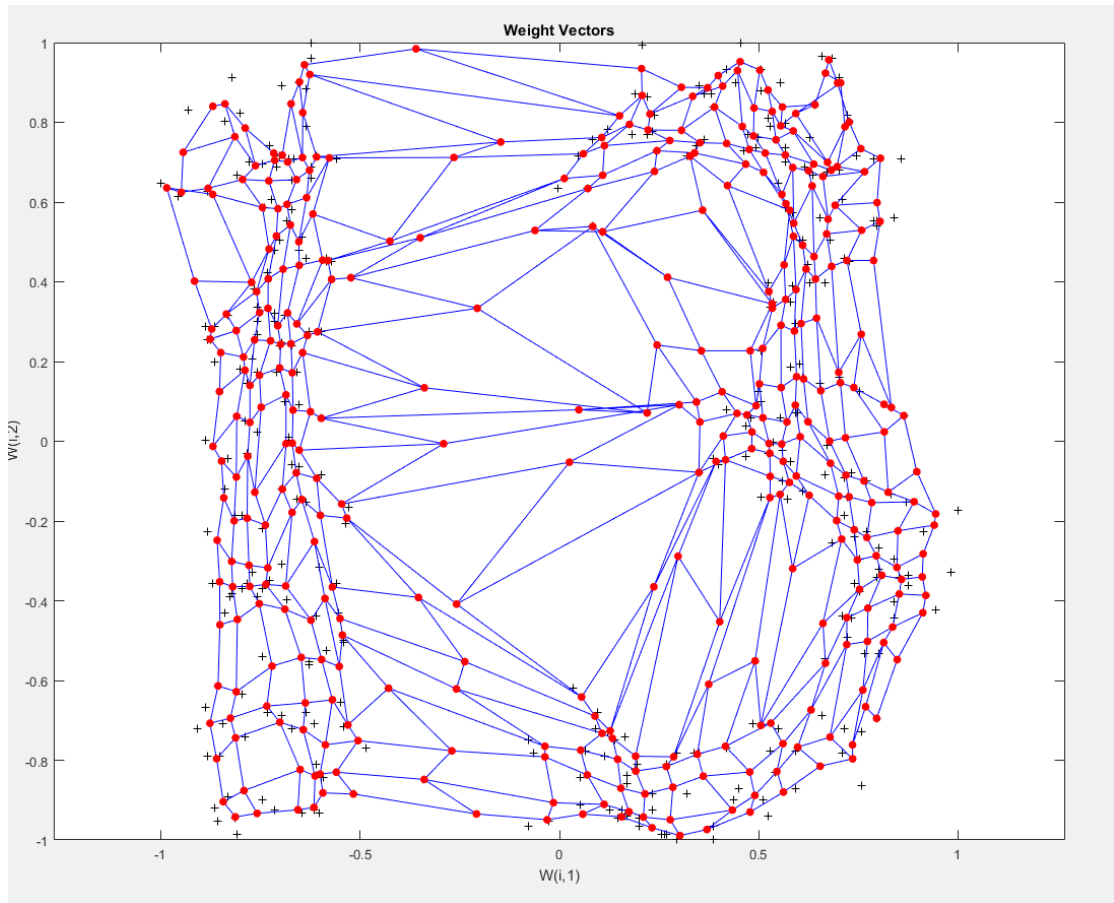
Σε αυτό το ερώτημα θα δημιουργήσουμε ένα SOM με σκοπό την ομαδοποίηση προτύπων. Εξετάζουμε λοιπόν ένα σύνολο δεδομένων που αποτελείται από 350 πρότυπα τα οποία χωρίζονται σε 2 ομάδες (0 ή 1).

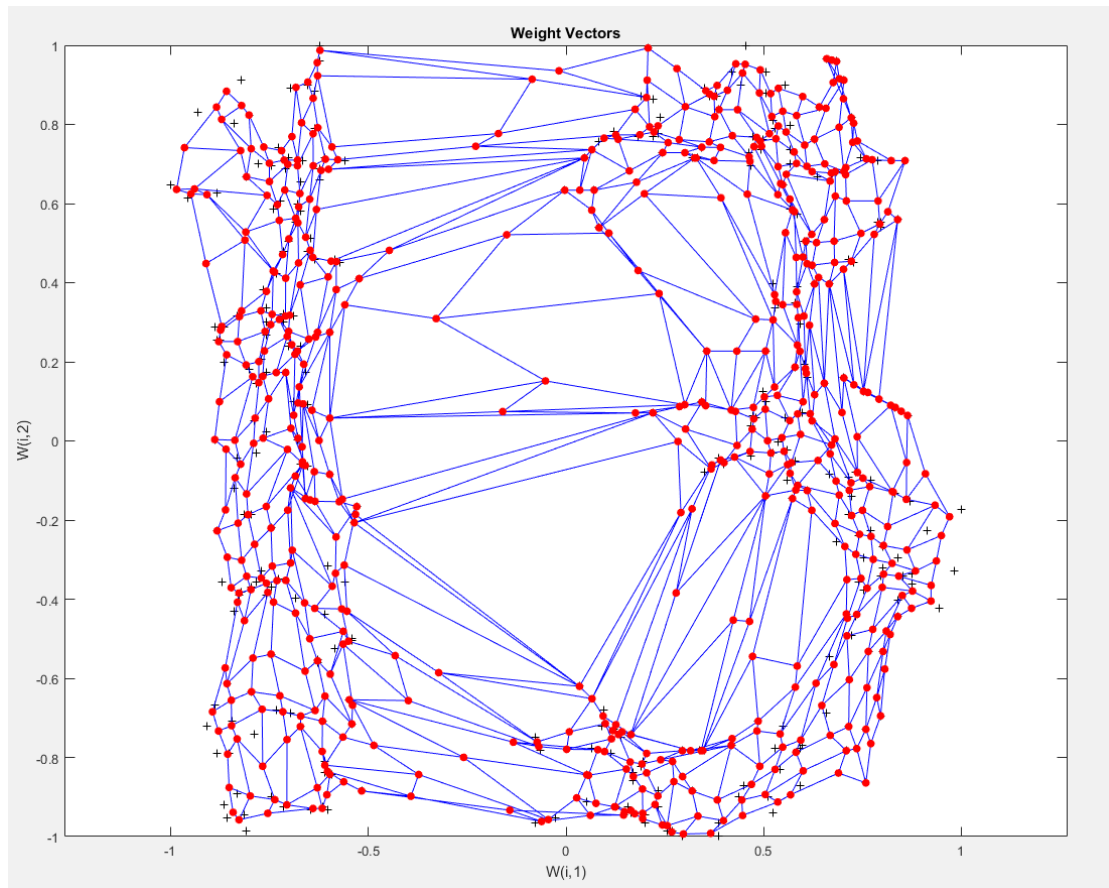
Σε αυτό το σημείο θα πρέπει να σημειωθεί ένα σημαντικό στοιχείο που κατηύθυνε τις κινήσεις που πραγματοποιήσαμε για την επίλυση του προβλήματος. Παρατηρώντας το αρχείο GroupData.m φαίνεται εξαρχής ότι το classification του προβλήματος δίνεται ήδη από τα δεδομένα. Κληθήκαμε λοιπόν να αποφασίσουμε αν θα έπρεπε να αξιοποιήσουμε αυτή την πληροφορία εξαρχής ή όχι. Η επίλυση της άσκησης λοιπόν έγινε με δύο τρόπους. Στον έναν τρόπο αξιοποιούσαμε το βάρος που προέκυψε από τους νευρώνες που κρίνουν το τρίτο χαρακτηριστικό (κατηγοριοποίηση), ενώ στο δεύτερο τρόπο αγνοήσαμε την τρίτη στήλη του GroupData.m, εκπαιδεύσαμε το σύστημα με βάση τις δύο πρώτες στήλες και υλοποιήσαμε μόνοι μας το classification. Όπως αναμενόταν λόγω του απλού συνόλου δεδομένων που λάβαμε, τα αποτελέσματα ήταν πανομοιότυπα και στις δύο περιπτώσεις.

Ακολουθώς παρουσιάζονται οι U-matrices που προέκυψαν για διαφορετικά πλέγματα νευρώνων και 500 εποχές (10x10, 15x15 και 17x17 αντίστοιχα). Και σε αυτό το ερώτημα χρησιμοποιήσαμε το αρχείο ScriptC.m με κατάλληλη είσοδο για το SOM:









Παρατηρούμε πως όσο περισσότεροι είναι οι νευρώνες τόσο καλύτερος είναι ο διαχωρισμός των προτύπων, ωστόσο ακόμα και για πλέγμα 10x10 βλέπουμε μια ικανοποιητική προσέγγιση.

Η σχέση μεταξύ του μεγέθους μια ομάδας όπως εμφανίζεται στον πίνακα U-matrix και του πλήθους προτύπων που περιέχει είναι χωρική, δηλαδή όσο περισσότερα πρότυπα περιέχει μια ομάδα τόσο πιο εμφανής θα είναι η περιοχή της στον πίνακα. Αυτό δηλώνεται στον πίνακα από νευρώνες ίδιου χρώματος (μπλε σκούρο) που σημαίνει πως έχουν κοντινές αποστάσεις από τους γείτονές τους. Ο διαχωρισμός των ομάδων φαίνεται στον πίνακα από νευρώνες που λειτουργούν ως σύνορα (το χρώμα τους δείχνει μεγάλη μέση απόσταση από γειτονικούς νευρώνες).

Στο αρχείο matlab `ClassificationResults.m` (`ClassificationResultsBetter.m` αν αγνοήσουμε την τρίτη στήλη του `GroupData.m`) που δημιουργήσαμε, παρέχουμε κώδικα που με βάση τις τιμές των νευρώνων καταγράφει τον αριθμό των νευρώνων που ανήκουν σε κάθε ομάδα, καθώς και τον αριθμό των νευρώνων που αποτελούν “ενδιάμεσους νευρώνες” (ανενεργούς νευρώνες).

Ιδανικά η ποσοστιαία συσχέτιση μεταξύ του πλήθους των προτύπων κάθε ομάδας και του πλήθους των νευρώνων που ανατίθενται σε κάθε ομάδα είναι σχεδόν ίδιες. Πρακτικά όμως παρατηρούμε διαφοροποιήσεις καθώς δεν υπάρχει 1-1 αντιστοίχιση μεταξύ προτύπων και νευρώνων με αποτέλεσμα κάποιοι νευρώνες να μην

αντιστοιχίζονται με ξεκάθαρο τρόπο σε μια ομάδα. Ακόμη, ίσως και ο αριθμός των εποχών εκπαίδευσης του δικτύου να μην είναι επαρκής ώστε να προσεγγίσουμε την ιδανική συσχέτιση.

Παρακάτω παρουσιάζονται ενδεικτικά αποτελέσματα του τρεξίματος του ClassificationResults.m με SOM 15x15 νευρώνων, τύπο πλέγματος gridtop και μέτρο απόστασης mandist:

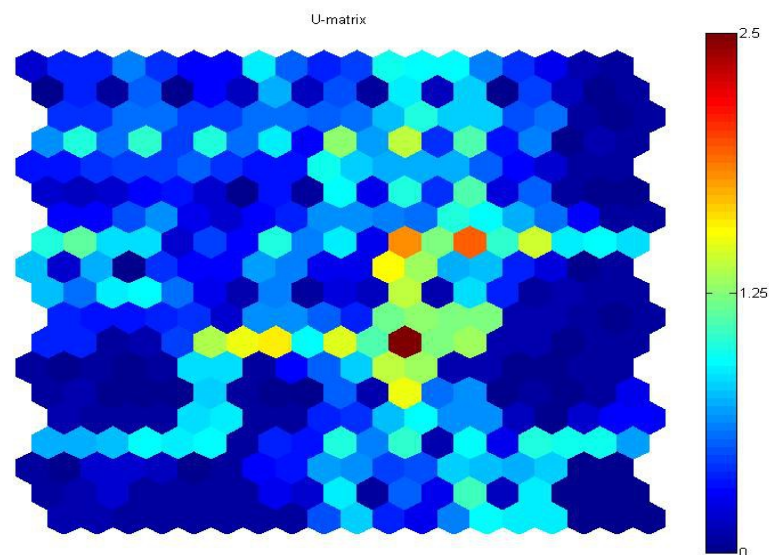
```
>> ClassificationResultsBetter
```

```
Group 0: 132 neurons and on the edge there are: 14 neurons
```

```
Group 1: 79 neurons
```

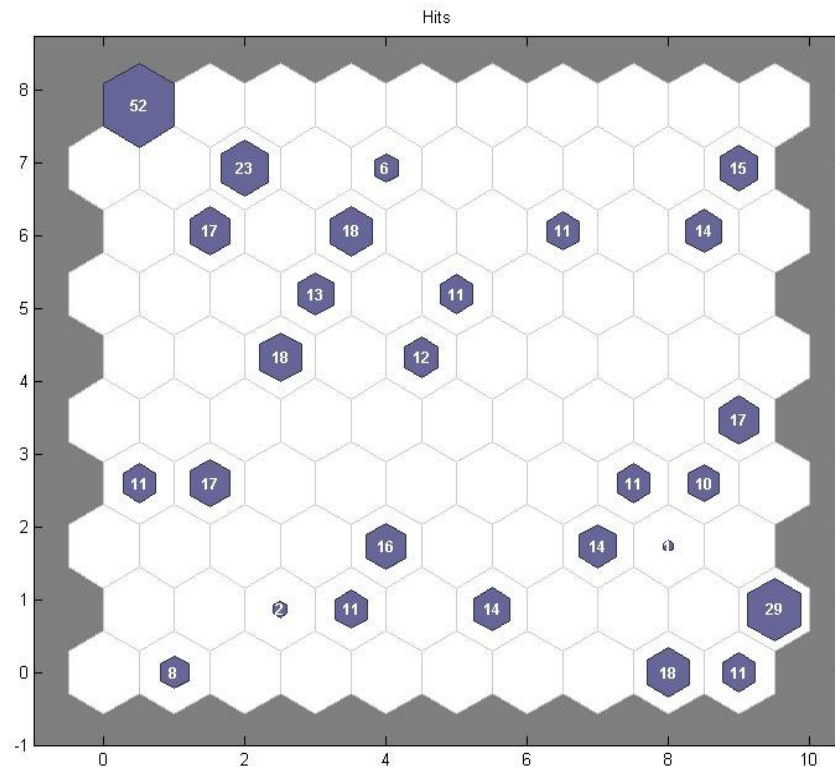
Ερώτημα 3

Σε αυτό το ερώτημα θα επαναλάβουμε την διαδικασία της ομαδοποίησης, αυτή τη φορά δημιουργώντας τον αυτό-οργανούμενο χάρτη μας μέσω της εγκατεστημένης εργαλειοθήκης της Matlab. Σύμφωνα με την εκφώνηση κατασκευάσαμε ένα πλέγμα εξαγωνικής μορφής, μεγέθους 10x10, χρησιμοποιώντας την ευκλείδεια απόσταση και με τις τιμές των βαρών των νευρώνων στο διάστημα [0, 1]. Παρουσιάζουμε τον U-matrix για σύγκριση με το προηγούμενο ερώτημα καθώς και ένα γράφημα με τον αριθμό προτύπων ανά νευρώνα:



Self-Organizing Map

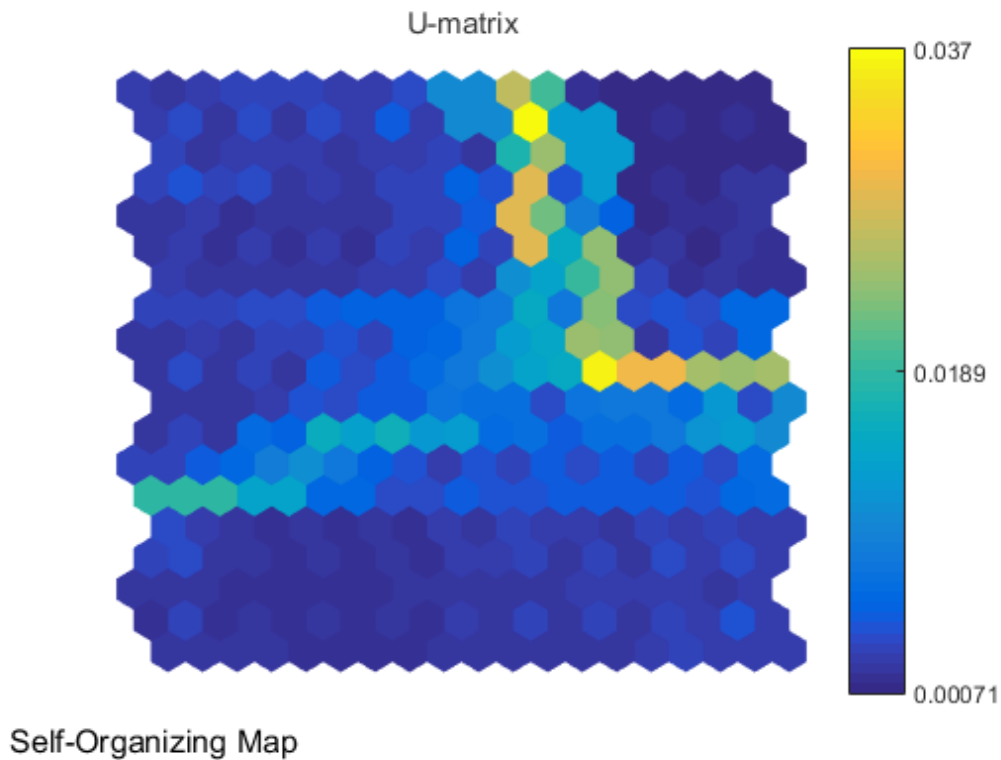
Συγκριτικά με το προηγούμενο ερώτημα, παρατηρούμε μια πιο ξεκάθαρη ομαδοποίηση στον U-matrix, σε σχέση με το 10x10 πλέγμα που είχαμε παρουσιάσει.



Σε αυτό το γράφημα με τη βοήθεια της `plotsomhits` φαίνεται (με λίγη καλή θέληση) ο αριθμός προτύπων ανά νευρώνα. Στο αρχείο `3_10x10.mat` οι παραπάνω αριθμοί περιέχονται στον πίνακα `HitsPerNeuron`, ενώ ο κώδικας επισυνάπτεται στο αρχείο `Master3.m`.

Ερώτημα 3

Σε αυτό το ερώτημα θέλουμε να κάνουμε ομαδοποίηση 500 εγγράφων με 8296 λέξεις το καθένα με ένα SOM 10x10. Εφαρμόζοντας τη συνάρτηση `tfidf1.m` υπολογίσαμε τον τελικό πίνακα με τα ανηγμένα βάρη. Μέσω των συναρτήσεων δημιουργήσαμε κατάλληλο SOM. Ο U-matrix μετά την προσομοίωση παρουσιάζεται ακολούθως:



Επειδή το μέγεθος των δεδομένων εισόδου (500x8296) είναι συγκριτικά πολύ μεγαλύτερο από το SOM που χρησιμοποιούμε (10x10) ο U-matrix δεν μας δίνει αρκετά ξεκάθαρη εικόνα, αλλά μια ισχνή εκτίμηση για το πλήθος των ομάδων το οποίο εκτιμάται κοντά στις 3 ομάδες.

Στο αρχείο `neural23.m` περιέχεται ο κώδικας για όλα τα ζητούμενα και στο `ask3resultsfinal.mat` όλα τα αποτελέσματα:

- i. Το πλήθος εγγραφών για κάθε νευρώνα (περιέχεται στον πίνακα `RecordsPerNeuron`) παρουσιάζεται ακολούθως:

Neuron 1 has 16 records

Neuron 2 has 9 records

Neuron 3 has 10 records

Neuron 4 has 11 records

Neuron 5 has 6 records

Neuron 6 has 8 records

Neuron 7 has 7 records

Neuron 8 has 5 records

Neuron 9 has 6 records

Neuron 10 has 8 records

Neuron 11 has 3 records

Neuron 12 has 6 records

Neuron 13 has 10 records

Neuron 14 has 8 records

Neuron 15 has 5 records

Neuron 16 has 1 records

Neuron 17 has 7 records

Neuron 18 has 3 records

Neuron 19 has 2 records

Neuron 20 has 8 records

Neuron 21 has 5 records

Neuron 22 has 4 records

Neuron 23 has 6 records

Neuron 24 has 6 records

Neuron 25 has 6 records

Neuron 26 has 5 records

Neuron 27 has 3 records

Neuron 28 has 1 records

Neuron 29 has 6 records

Neuron 30 has 6 records

Neuron 31 has 7 records

Neuron 32 has 0 records

Neuron 33 has 0 records

Neuron 34 has 3 records

Neuron 35 has 4 records

Neuron 36 has 7 records

Neuron 37 has 3 records

Neuron 38 has 5 records

Neuron 39 has 6 records

Neuron 40 has 4 records

Neuron 41 has 7 records

Neuron 42 has 6 records

Neuron 43 has 7 records

Neuron 44 has 2 records

Neuron 45 has 0 records

Neuron 46 has 0 records

Neuron 47 has 1 records

Neuron 48 has 0 records

Neuron 49 has 0 records

Neuron 50 has 0 records

Neuron 51 has 8 records

Neuron 52 has 8 records

Neuron 53 has 4 records

Neuron 54 has 8 records

Neuron 55 has 3 records

Neuron 56 has 3 records

Neuron 57 has 0 records

Neuron 58 has 2 records

Neuron 59 has 3 records

Neuron 60 has 1 records

Neuron 61 has 10 records

Neuron 62 has 8 records

Neuron 63 has 6 records

Neuron 64 has 4 records

Neuron 65 has 6 records

Neuron 66 has 3 records

Neuron 67 has 0 records

Neuron 68 has 4 records

Neuron 69 has 4 records

Neuron 70 has 5 records

Neuron 71 has 5 records

Neuron 72 has 7 records

Neuron 73 has 10 records

Neuron 74 has 4 records

Neuron 75 has 4 records

Neuron 76 has 4 records

Neuron 77 has 0 records

Neuron 78 has 3 records

Neuron 79 has 3 records

Neuron 80 has 2 records
Neuron 81 has 6 records
Neuron 82 has 4 records
Neuron 83 has 2 records
Neuron 84 has 5 records
Neuron 85 has 4 records
Neuron 86 has 2 records
Neuron 87 has 0 records
Neuron 88 has 6 records
Neuron 89 has 9 records
Neuron 90 has 1 records
Neuron 91 has 14 records
Neuron 92 has 6 records
Neuron 93 has 8 records
Neuron 94 has 5 records
Neuron 95 has 11 records
Neuron 96 has 0 records
Neuron 97 has 5 records
Neuron 98 has 7 records
Neuron 99 has 13 records
Neuron 100 has 11 records

- ii. Για κάθε νευρώνα βρέθηκε ο τίτλος του κειμένου που έχει τη μικρότερη απόσταση από αυτόν τον νευρώνα (στον πίνακα er2 βρίσκονται τα indexes των κειμένων για τον πίνακα titles για τον αντίστοιχο νευρώνα):

Neuron 1 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 2 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 3 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 4 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 5 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 6 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 7 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 8 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 9 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 10 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 11 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 12 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 13 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 14 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 15 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 16 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 17 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 18 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 19 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 20 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 21 has smallest distance from Mean FieM Methods for Classification with Gaussian Processes,

Neuron 22 has smallest distance from Mean FieM Methods for Classification with Gaussian Processes,

Neuron 23 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 24 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 25 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 26 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 27 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 28 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 29 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 30 has smallest distance from Spatial Decorrelation in Orientation Tuned Cortical Cells,

Neuron 31 has smallest distance from Invariant Feature Extraction and Classification in Kernel Spaces,

Neuron 32 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 33 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 34 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 35 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 36 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 37 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 38 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 39 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 40 has smallest distance from Spatial Decorrelation in Orientation Tuned Cortical Cells,

Neuron 41 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 42 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 43 has smallest distance from Online Independent Component Analysis with Local Learning Rate Adaptation,

Neuron 44 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 45 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 46 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 47 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 48 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 49 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 50 has smallest distance from A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-test Split

Neuron 51 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 52 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 53 has smallest distance from Online Independent Component Analysis with Local Learning Rate Adaptation,

Neuron 54 has smallest distance from Modeling Acoustic Correlations by Factor Analysis,

Neuron 55 has smallest distance from Modeling Acoustic Correlations by Factor Analysis,

Neuron 56 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 57 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 58 has smallest distance from Recurrent Cortical Competition: Strengthen or Weaken?,

Neuron 59 has smallest distance from Predictive Sequence Learning in Recurrent Neocortical Circuits,

Neuron 60 has smallest distance from Predictive Sequence Learning in Recurrent Neocortical Circuits,

Neuron 61 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 62 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 63 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 64 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 65 has smallest distance from Robot Docking Using Mixtures of Gaussians,

Neuron 66 has smallest distance from Learning Temporally Persistent Hierarchical Representations,

Neuron 67 has smallest distance from Learning Temporally Persistent Hierarchical Representations,

Neuron 68 has smallest distance from Variational Inference for Bayesian Mixtures of Factor Analysers,

Neuron 69 has smallest distance from Variational Inference for Bayesian Mixtures of Factor Analysers,

Neuron 70 has smallest distance from Variational Inference for Bayesian Mixtures of Factor AnaIysers,

Neuron 71 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 72 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 73 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 74 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 75 has smallest distance from Learning Temporally Persistent Hierarchical Representations,

Neuron 76 has smallest distance from Learning Temporally Persistent Hierarchical Representations,

Neuron 77 has smallest distance from Image Representations for Facial Expression Coding,

Neuron 78 has smallest distance from Variational Inference for Bayesian Mixtures of Factor AnaIysers,

Neuron 79 has smallest distance from Variational Inference for Bayesian Mixtures of Factor AnaIysers,

Neuron 80 has smallest distance from Learning Factored Representations for Partially Observable Markov Decision Processes,

Neuron 81 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 82 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 83 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 84 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 85 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 86 has smallest distance from Learning Temporally Persistent Hierarchical Representations,

Neuron 87 has smallest distance from Image Representations for Facial Expression Coding,

Neuron 88 has smallest distance from Variational Inference for Bayesian Mixtures of Factor Analysers,

Neuron 89 has smallest distance from Variational Inference for Bayesian Mixtures of Factor Analysers,

Neuron 90 has smallest distance from Learning Factored Representations for Partially Observable Markov Decision Processes,

Neuron 91 has smallest distance from Gradient Descent for General Reinforcement Learning,

Neuron 92 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 93 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 94 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 95 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 96 has smallest distance from Learning Curves for Gaussian Processes,

Neuron 97 has smallest distance from Variational Inference for Bayesian Mixtures of Factor AnaIysers,

Neuron 98 has smallest distance from Variational Inference for Bayesian Mixtures of Factor AnaIysers,

Neuron 99 has smallest distance from Variational Inference for Bayesian Mixtures of Factor AnaIysers,

Neuron 100 has smallest distance from Learning Factored Representations for Partially Observable Markov Decision Processes,

- iii. Για κάθε νευρώνα βρέθηκαν οι 3 όροι που έχουν το μεγαλύτερο βάρος σε αυτόν τον νευρώνα (στους πίνακες max1, max2 και max3 βρίσκονται ο πρώτος, ο δεύτερος και ο τρίτος μεγαλύτερος όρος αντίστοιχα):

Neuron 1 has max weight in the following terms: tree , teacher , student

Neuron 2 has max weight in the following terms: queri , tree , teacher

Neuron 3 has max weight in the following terms: kernel , loss , queri

Neuron 4 has max weight in the following terms: ica , kurtosi , pca

Neuron 5 has max weight in the following terms: style , pca , bind

Neuron 6 has max weight in the following terms: bind , ensembl , schedul

Neuron 7 has max weight in the following terms: hmm , word , ensembl

Neuron 8 has max weight in the following terms: speech , saccad , hmm

Neuron 9 has max weight in the following terms: motion , saccad , movement

Neuron 10 has max weight in the following terms: motion , spike , circuit

Neuron 11 has max weight in the following terms: tree , queri , teacher

Neuron 12 has max weight in the following terms: queri , mse , momentum

Neuron 13 has max weight in the following terms: kalman , ekf , outcom

Neuron 14 has max weight in the following terms: pca , latent , style

Neuron 15 has max weight in the following terms: style , em , bind

Neuron 16 has max weight in the following terms: expert , hmm , ensembl

Neuron 17 has max weight in the following terms: rout , polici , stock

Neuron 18 has max weight in the following terms: saccad ,
motor , movement

Neuron 19 has max weight in the following terms: spike ,
motion , neuron

Neuron 20 has max weight in the following terms: spike ,
synaps , motion

Neuron 21 has max weight in the following terms: wind ,
hyperparamet , bayesian

Neuron 22 has max weight in the following terms: bayesian ,
hyperparamet , momentum

Neuron 23 has max weight in the following terms: convolut ,
polynomi , cliqu

Neuron 24 has max weight in the following terms: slice , som ,
retriev

Neuron 25 has max weight in the following terms: cluster , som
, latent

Neuron 26 has max weight in the following terms: expert ,
cluster , hme

Neuron 27 has max weight in the following terms: hme ,
expert , drift

Neuron 28 has max weight in the following terms: hme ,
polici , reward

Neuron 29 has max weight in the following terms: neuron ,
popul , cortex

Neuron 30 has max weight in the following terms: spike ,
neuron , cell

Neuron 31 has max weight in the following terms: kernel ,
pca , svm

Neuron 32 has max weight in the following terms: kernel ,
ica , pca

Neuron 33 has max weight in the following terms: decod , ica ,
eeg

Neuron 34 has max weight in the following terms: decod , bit ,
languag

Neuron 35 has max weight in the following terms: face ,
interpol , imag

Neuron 36 has max weight in the following terms: face ,
interpol , skill

Neuron 37 has max weight in the following terms: chess ,
game , wavelet

Neuron 38 has max weight in the following terms: wavelet ,
textur , contour

Neuron 39 has max weight in the following terms: contour ,
cell , wavelet

Neuron 40 has max weight in the following terms: dispar , cell
, head

Neuron 41 has max weight in the following terms: kernel ,
svm , smo

Neuron 42 has max weight in the following terms: kernel ,
shrinkag , ica

Neuron 43 has max weight in the following terms: ica , eeg ,
yellow

Neuron 44 has max weight in the following terms: hmm , decod ,
languag

Neuron 45 has max weight in the following terms: hmm , decod ,
word

Neuron 46 has max weight in the following terms: document ,
face , word

Neuron 47 has max weight in the following terms: skill ,
document , face

Neuron 48 has max weight in the following terms: wavelet ,
skill , textur

Neuron 49 has max weight in the following terms: wavelet ,
contour , textur

Neuron 50 has max weight in the following terms: cell , dispar , contour

Neuron 51 has max weight in the following terms: cia , posterior , svm

Neuron 52 has max weight in the following terms: ica , ergod , blind

Neuron 53 has max weight in the following terms: hmm , ica , speech

Neuron 54 has max weight in the following terms: hmm , speech , triphon

Neuron 55 has max weight in the following terms: hmm , word , document

Neuron 56 has max weight in the following terms: document , word , matlab

Neuron 57 has max weight in the following terms: skill , document , matlab

Neuron 58 has max weight in the following terms: exc , citatori , nize

Neuron 59 has max weight in the following terms: fire , spike , synapt

Neuron 60 has max weight in the following terms: fire , cell , synaps

Neuron 61 has max weight in the following terms: transduct , loss , margin

Neuron 62 has max weight in the following terms: node , expert , imag

Neuron 63 has max weight in the following terms: imag , fusion , committe

Neuron 64 has max weight in the following terms: cluster , imag , user

Neuron 65 has max weight in the following terms: cluster , manifold , hofmann

Neuron 66 has max weight in the following terms: cluster ,
face , manifold

Neuron 67 has max weight in the following terms: face , imag ,
manifold

Neuron 68 has max weight in the following terms: deflect , itc
, visuo

Neuron 69 has max weight in the following terms: cit , deflect
, itc

Neuron 70 has max weight in the following terms: cue ,
phenomenolog , jin

Neuron 71 has max weight in the following terms: margin , node
, adaboost

Neuron 72 has max weight in the following terms: node , imag ,
segment

Neuron 73 has max weight in the following terms: imag , motion
, scene

Neuron 74 has max weight in the following terms: cluster ,
salienc , attent

Neuron 75 has max weight in the following terms: cluster ,
face , attractor

Neuron 76 has max weight in the following terms: face , imag ,
emot

Neuron 77 has max weight in the following terms: face ,
deflect , crop

Neuron 78 has max weight in the following terms: deflect ,
occipit , itc

Neuron 79 has max weight in the following terms: realism , lam
, jin

Neuron 80 has max weight in the following terms: brows ,
mcreynold , phenomenolog

Neuron 81 has max weight in the following terms: polici ,
adaboost , boost

Neuron 82 has max weight in the following terms: polici ,
action , reinforc

Neuron 83 has max weight in the following terms: cell , coin ,
robot

Neuron 84 has max weight in the following terms: neuron ,
motion , cell

Neuron 85 has max weight in the following terms: neuron , cell
, synapt

Neuron 86 has max weight in the following terms: neuron ,
synapt , face

Neuron 87 has max weight in the following terms: face , love ,
imag

Neuron 88 has max weight in the following terms: inen ,
logit , ascertain

Neuron 89 has max weight in the following terms: stdev ,
backer , realism

Neuron 90 has max weight in the following terms: boyen , stdev
, undiscount

Neuron 91 has max weight in the following terms: polici ,
action , mdp

Neuron 92 has max weight in the following terms: polici ,
action , reinforc

Neuron 93 has max weight in the following terms: chip , coin ,
cell

Neuron 94 has max weight in the following terms: neuron , iiii
, cell

Neuron 95 has max weight in the following terms: neuron ,
synapt , cell

Neuron 96 has max weight in the following terms: neuron ,
synapt , cell

Neuron 97 has max weight in the following terms: attractor ,
rota , attrac

Neuron 98 has max weight in the following terms: nick ,
stdev , chair

Neuron 99 has max weight in the following terms: boyen , stdev
, dial

Neuron 100 has max weight in the following terms: boyen ,
undiscount , elf

.

- iv. Οι νευρώνες του χάρτη για τους οποίους οι όροι «network» και «function» ταυτόχρονα έχουν τιμή μεγαλύτερη του 30% της μέγιστης τιμής τους σύμφωνα με το τρέξιμο του κώδικά μας προκύπτουν (πίνακας cool):

Neuron 1 satisfies

Neuron 2 satisfies

Neuron 3 satisfies

Neuron 4 satisfies

Neuron 5 satisfies

Neuron 6 satisfies

Neuron 7 satisfies

Neuron 8 satisfies

Neuron 11 satisfies

Neuron 12 satisfies

Neuron 13 satisfies

Neuron 14 satisfies

Neuron 15 satisfies

Neuron 16 satisfies

Neuron 17 satisfies

Neuron 18 satisfies

Neuron 19 satisfies

Neuron 21 satisfies

Neuron 22 satisfies

Neuron 23 satisfies

Neuron 24 satisfies

Neuron 25 satisfies

Neuron 26 satisfies

Neuron 27 satisfies

Neuron 28 satisfies

Neuron 29 satisfies

Neuron 30 satisfies

Neuron 36 satisfies

Neuron 37 satisfies

Neuron 38 satisfies

Neuron 46 satisfies

Neuron 47 satisfies

Neuron 48 satisfies

Neuron 61 satisfies

Neuron 62 satisfies

Neuron 71 satisfies

Neuron 72 satisfies

- v. Η μέση τιμή του βάρους κάθε όρου για τους νευρώνες του προηγούμενου ερωτήματος ως ποσοστό της μέγιστης τιμής για όλους τους νευρώνες του χάρτη περιέχεται στον πίνακα AvG του ask3resultsfinal.mat.