



# ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

Διπλωματική Εργασία

“Υπολογισμός μετατόπισης στην τοποθέτηση ολοκληρωμένου  
κυκλώματος με χρήση συσχετιστικών μνημών”

Ελένη Καλύβα

©Χανιά, Σεπτέμβριος 2001

*Στους γονείς μου Ιωάννα και Θανάση*

## Περίληψη

Σκοπός της εργασίας αυτής είναι η σχεδίαση και υλοποίηση ενός συστήματος μηχανικής όρασης, για τον υπολογισμό της μετατόπισης του *chip* (component) ενός ολοκληρωμένου κυκλώματος κατά την τοποθέτηση του πάνω στην πλακέτα του κυκλώματος, σε πραγματικό χρόνο, με χρήση συσχετιστικών μνημών. Η είσοδος είναι η ψηφιακή εικόνα του ολοκληρωμένου κυκλώματος. Το σύστημα σχεδιάστηκε να δέχεται ως είσοδο τις περιοχές ενδιαφέροντος της αρχικής εικόνας, δηλαδή τις εικόνες των ακροδεκτών (*leads*) του. Η συνολική μετατόπιση θα προκύψει από τη μέση τιμή των μετατοπίσεων των *leads* για κάθε πλευρά του ολοκληρωμένου κυκλώματος. Το σύστημα περιλαμβάνει ένα αρχικό στάδιο προεπεξεργασίας των εικόνων (με μεθόδους επεξεργασίας εικόνας), καθώς και τεχνητά νευρωνικά δίκτυα που υλοποιούν τις συσχετιστικές μνήμες (τα δίκτυα *Hopfield* και *Hamming*).

# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή.....</b>	<b>7</b>
1.1	Γενική εισαγωγή.....	7
1.2	Ορισμός του προβλήματος .....	9
1.3	Επεξεργασία του εικόνas του PCB.....	10
<b>2</b>	<b>Θεωρία των δικτύων.....</b>	<b>12</b>
2.1	Εισαγωγή στα νευρωνικά δίκτυα .....	12
2.1.1	Ορισμός .....	12
2.1.2	Λόγοι χρησιμοποίησης των ΤΝΔ .....	12
2.1.3	Δομή και λειτουργία του ΤΝΔ .....	13
2.1.3.1	Βιολογικό μοντέλο.....	13
2.1.3.2	Μοντέλο ΤΝΔ .....	14
2.1.3.3	Λειτουργία δομικών στοιχείων του ΤΝΔ .....	14
2.1.4	Αρχιτεκτονική ΤΝΔ.....	16
2.1.5	Εκπαίδευση ΤΝΔ .....	16
2.2	Συσχετιστικές μνήμες (associative memories).....	17
2.3	Το δίκτυο Hopfield.....	17
2.3.1	Τοπολογία του δικτύου.....	18
2.3.2	Λειτουργία του δικτύου.....	19
2.3.2.1	Φάση αποθήκευσης.....	19
2.3.2.2	Φάση ανάκλησης.....	19
2.3.3	Θεωρητική βάση.....	20
2.4	Προβλήματα των δικτύων Hopfield .....	22
2.5	Τροποποιήσεις του αρχικού μοντέλου .....	23
2.5.1	Επιλογή του κατάλληλου learning rule.....	23
2.5.1.1	Projection (or inverse) learning rule .....	23
2.5.1.2	Learning in bounds .....	24
2.5.2	Επιλογή του κατάλληλου threshold .....	25
2.5.2.1	Adjustable thresholds .....	25
<b>3</b>	<b>Προτεινόμενη διαδικασία .....</b>	<b>26</b>
3.1	Προεπεξεργασία .....	26

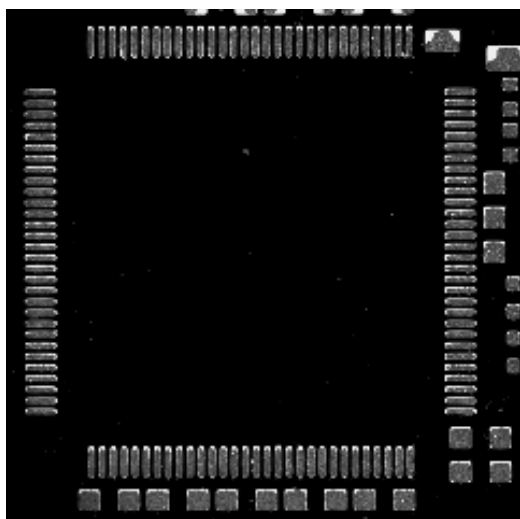
3.2	Επιλογή των associative memories .....	29
3.2.1	Διαδικασία 1.....	30
3.2.2	Διαδικασία 2.....	33
3.2.3	Διαδικασία 3.....	34
3.3	Λειτουργία του δικτύου Hopfield .....	36
3.4	Classification των εικόνων μετά την έξοδο τους από το δίκτυο Hopfield .....	38
3.4.1	Δίκτυο Hamming .....	39
3.4.1.1	Τοπολογία του δικτύου .....	40
3.4.1.2	Λειτουργία .....	41
3.5	Συνολικό σύστημα.....	42
4	Αποτελέσματα και συμπεράσματα .....	43
4.1	Απόδοση του δικτύου.....	43
4.1.1	Αποτελέσματα για σύστημα με την πρώτη διαδικασία επιλογής μνημών .....	43
4.1.2	Αποτελέσματα για σύστημα με τη δεύτερη διαδικασία επιλογής μνημών .....	45
4.1.3	Αποτελέσματα για σύστημα με την τρίτη διαδικασία επιλογής μνημών .....	46
4.1.4	Παρατηρήσεις .....	48
4.2	Απόδοση του δικτύου για μετατόπιση ανά 2 pixels.....	49
4.2.1	Αποτελέσματα για σύστημα με την πρώτη διαδικασία επιλογής μνημών .....	49
4.2.2	Αποτελέσματα για σύστημα με τη δεύτερη διαδικασία επιλογής μνημών .....	50
4.2.3	Αποτελέσματα για σύστημα με την τρίτη διαδικασία επιλογής μνημών .....	51
4.3	Απόδοση του δικτύου για μετατόπιση ανά 3 pixels.....	52
4.3.1	Αποτελέσματα για σύστημα με την πρώτη διαδικασία επιλογής μνημών .....	52
4.3.2	Αποτελέσματα για σύστημα με τη δεύτερη διαδικασία επιλογής μνημών .....	53
4.3.3	Αποτελέσματα για σύστημα με την τρίτη διαδικασία επιλογής μνημών .....	54
4.4	Μέσοι αριθμοί επαναλήψεων .....	55
4.5	Παραδείγματα διάδοσης στο δίκτυο .....	56
4.6	Αναλύοντας την ποιότητα του συστήματος .....	61
5	Αξιολόγηση – επεκτάσεις .....	65
5.1	Αξιολόγηση .....	65
5.2	Επεκτάσεις.....	66
6	Παράρτημα 1 .....	67
7	Παράρτημα 2 .....	69
8	Βιβλιογραφία .....	71



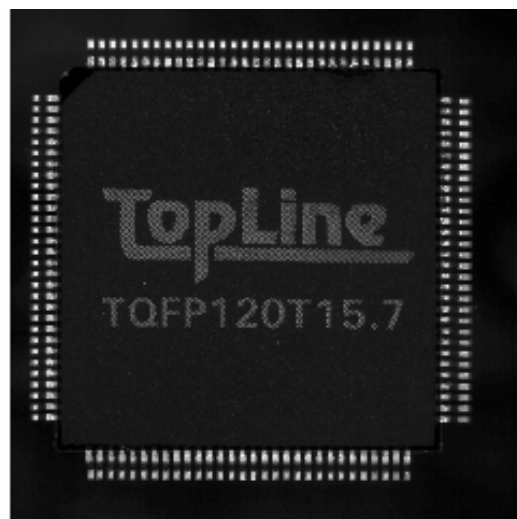
# 1 Εισαγωγή

## 1.1 Γενική εισαγωγή

Η εργασία αυτή είναι μέρος του HIPER project, το οποίο διαπραγματεύεται την τοποθέτηση SMD (Surface Mounted Device) πάνω σε PCB (printed circuit board). Σκοπός αυτού του project είναι να εξεταστεί η ποιότητα τοποθέτησης ενός group από SMD πάνω στο PCB. Η μορφή του SMD καθώς και του PCB φαίνονται στο σχήμα 1.1.



PCB



SMD component

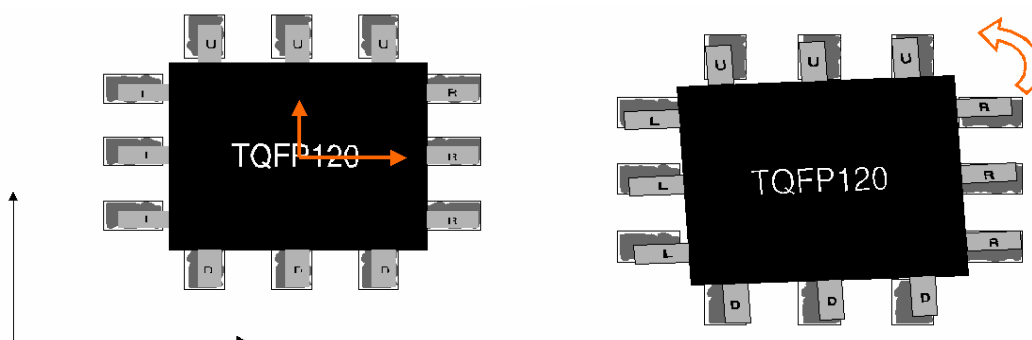
**Σχήμα 1.**Error! Unknown switch argument. **Μορφή του PCB και του SMD component**

Η διαδικασία που ακολουθείται είναι η εξής : πρώτα τοποθετείται το SMD component (το οποίο είναι το Topline TQFP120t15.7) πάνω στο PCB (το οποίο περιέχει υγρή κόλλα (solder paste), καθώς και τη θέση (pad) πάνω στην οποία θα πρέπει να τοποθετηθεί το ‘ποδαράκι’ (lead) του component). Στη συνέχεια μια κάμερα (η HIPER camera) παίρνει φωτογραφίες του ολοκληρωμένου κυκλώματος. Οι εικόνες που παίρνουμε είναι 16- bit και έχουν τη γενική μορφή που φαίνεται στο σχήμα 1.2 .



**Σχήμα 1.**Error! Unknown switch argument. **Μορφή του ολοκληρωμένου κυκλώματος που λαμβάνουμε από την κάμερα**

Κατά την τοποθέτηση του component πάνω στο PCB μπορούμε να έχουμε λανθασμένη τοποθέτηση του component πάνω στο pad. Η λανθασμένη αυτή τοποθέτηση αφορά είτε την μετατοπισμένη τοποθέτηση του component, στον οριζόντιο ή στον κατακόρυφο άξονα (σχήμα 1.3), είτε στην τοποθέτηση του με κάποια γωνία στροφής (σχήμα 1.3 επίσης).



**Σχήμα 1.**Error! Unknown switch argument. **Περιπτώσεις λανθασμένης τοποθέτησης του component πάνω στο PCB**

Ανάλογα με το μέγεθος της μετατόπισης ή της γωνίας στροφής το σύστημα θα πρέπει να πραγματοποιεί έναν έλεγχο ποιότητας και να αποφασίζει εάν το ολοκληρωμένο κύκλωμα είναι 'καλό', έχει δηλαδή μετατόπιση και γωνία στροφής μικρότερες από κάποιες προδιαγραφές. Εάν η τοποθέτηση είναι έξω από τις προδιαγραφές, τότε το σύστημα θα απορρίψει το ολοκληρωμένο. Σημειώνεται ότι όλη η επεξεργασία θα πρέπει να γίνεται σε πραγματικό χρόνο.



## 1.2 Ορισμός του προβλήματος

Όπως έχει αναφερθεί το component του PCB έχει διάφορες μετατοπίσεις. Σκοπός της εργασίας αυτής είναι ο υπολογισμός της μετατόπισης του component χρησιμοποιώντας νευρωνικά δίκτυα συσχετιστικών μνημών. Θα μελετήσουμε μόνο την μετατόπιση στον κατακόρυφο άξονα, γιατί από την μετατόπιση αυτή κρίνεται η ποιότητα της τοποθέτησης. Οι πιθανές μετατοπίσεις του component είναι μέχρι 6 pixel πάνω και κάτω στον κατακόρυφο άξονα, καθώς και μέχρι 6 pixel δεξιά και αριστερά στον οριζόντιο άξονα από την θέση που πρέπει κανονικά να τοποθετηθεί (στη θέση αυτή έχει 0 pixel μετατόπιση). Όποτε, οι συνολικές μετατοπίσεις θα είναι 25( 1..6 pixel πάνω, 1..6 pixel κάτω, 1..6 pixel δεξιά, 1..6 pixel αριστερά, 0 pixel ).

Όμως, επειδή είναι εξαιρετικά δύσκολο να πραγματοποιηθεί αυτός ο υπολογισμός για ολόκληρη την εικόνα του κυκλώματος, θα προχωρήσουμε στη διαδικασία που περιγράφεται στην παρακάτω παράγραφο, έτσι ώστε να δημιουργήσουμε εικόνες που θα περιέχουν μόνο το κάθε 'ποδαράκι' (lead) του κυκλώματος και θα υπολογίσουμε τη μετατόπιση για κάθε lead χωριστά ( η διαδικασία που ακολουθείται περιγράφεται αναλυτικότερα στην παράγραφο 1.3).

Η συνολική μετατόπιση του component πάνω στο PCB θα προκύψει από τη μέση μετατόπιση των lead σε κάθε πλευρά του κυκλώματος.

Όταν έχω μετατόπιση κατά  $p$  pixel ( το  $p$  ανήκει στο ακέραιο σύνολο  $[-6, \dots, +6]$  ) στον άξονα  $y$  τότε σε κάθε πλευρά του ολοκληρωμένου θα έχω τις εξής μετατοπίσεις (στον κατακόρυφο άξονα):

left side	+p pixels
right side	+p pixels
up side	0 pixels
down side	0 pixels

**Πίνακας 1.**Error! Unknown switch argument. Μετατοπίσεις σε κάθε πλευρά του ολοκληρωμένου για μετατόπιση κατά  $p$  pixel στον άξονα  $y$

Αντίστοιχα, όταν έχω μετατόπιση κατά  $p$  pixel (επίσης το  $p$  ανήκει στο ακέραιο σύνολο  $[-6, \dots, +6]$ ) στον άξονα  $x$  τότε σε κάθε πλευρά του ολοκληρωμένου θα έχω τις εξής μετατοπίσεις (στον κατακόρυφο άξονα):

left side	0 pixels
right side	0 pixels
up side	+p pixels
down side	+p pixels

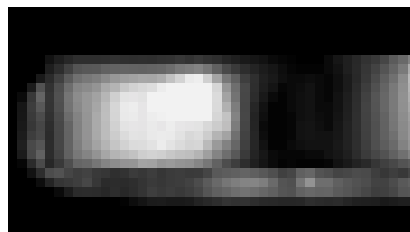
**Πίνακας 1.**Error! Unknown switch argument. **Μετατοπίσεις σε κάθε πλευρά του ολοκληρωμένου για μετατόπιση κατά p pixel στον άξονα x**

Παρατηρούμε ότι υπολογίζοντας ξεχωριστά την μετατόπιση για κάθε lead, οι πιθανές μετατοπίσεις που μπορεί αυτό να έχει περιορίζονται μόνο σε μετατοπίσεις στον κατακόρυφο άξονα. Οπότε, οι πιθανές μετατοπίσεις του κάθε lead θα είναι συνολικά 13 (1..6 pixel πάνω, 1..6 pixel κάτω και 0 pixel).

### 1.3 Επεξεργασία του εικόνας του PCB

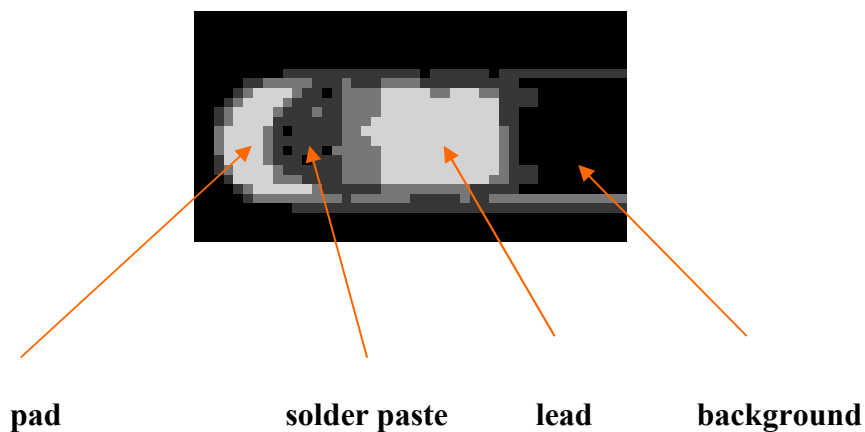
Αμέσως μετά την απόκτηση της παραπάνω εικόνας από την HIPER camera, προχωρούμε στην εξής διαδικασία :

Κόψιμο των lead από την εικόνα του σχήματος 1.2, έτσι ώστε να έχουμε εικόνες που αναπαριστούν ένα lead. Το μέγεθος των εικόνων είναι 24x44 pixels και η μορφή των εικόνων μετά από τη διαδικασία αυτή φαίνονται στο σχήμα 1.4.



**Σχήμα 1.**Error! Unknown switch argument. **Εικόνα που παριστάνει ένα lead του component**

Στη συνέχεια προχωρούμε σε τμηματοποίηση της εικόνας μας έτσι ώστε να πάρουμε εικόνα με 4 διαφορετικά χρωματικά επίπεδα. Το καθένα από τα επίπεδα αυτά θα αναπαριστά τις εξής περιοχές: lead, pad, solder paste και background. Ο αλγόριθμος που χρησιμοποιείται είναι ο αλγόριθμος του Otsu για τμηματοποίηση σε 4 επίπεδα. Οι εικόνες που παίρνουμε μετά τη διαδικασία αυτή φαίνονται στο σχήμα 1.5.



**Σχήμα 1.**Error! Unknown switch argument. **To lead μετά το segmentation**

Τέλος, εφαρμόζουμε στις εικόνες αλγορίθμους ταυτοποίησης (labeling) για απομάκρυνση μικρών περιοχών από το background, καθώς και για ομαλοποίηση των υπόλοιπων περιοχών της εικόνας. Οι τελικές εικόνες είναι αυτές που φαίνονται στο σχήμα 1.6.



**Σχήμα 1.**Error! Unknown switch argument. **To lead μετά το labeling**

Αυτή ακριβώς τη μορφή των εικόνων θα χρησιμοποιήσουμε για την εκπαίδευση και την επικύρωση των δικτύων που υλοποιούνται στην εφαρμογή αυτή.

Αυτό γίνεται γιατί , όπως αναφέρεται και στο κεφάλαιο 2 , τα δίκτυα συσχετιστικών μνημών που χρησιμοποιούμε απαιτούν δυαδικές εικόνες, όποτε θα είναι πιο εύκολη η μετατροπή εικόνων από 4 χρωματικά επίπεδα σε δυαδική.

## 2 Θεωρία των δικτύων

Στο κεφάλαιο αυτό παρουσιάζεται μια σύντομη αναφορά στα νευρωνικά δίκτυα και ειδικότερα στο δίκτυο Hopfield, το οποίο χρησιμοποιούμε στην εφαρμογή μας.

### 2.1 Εισαγωγή στα νευρωνικά δίκτυα

#### 2.1.1 Ορισμός

Ένα τεχνητό νευρωνικό δίκτυο (ΤΝΔ) είναι ένα σύστημα επεξεργασίας πληροφορίας, εμπνευσμένο από τον τρόπο με τον οποίο τα βιολογικά νευρικά συστήματα, όπως ο ανθρώπινος εγκέφαλος, επεξεργάζονται πληροφορίες. Αποτελείται από ένα σύνολο πολλαπλά διασυνδεδεμένων βασικών μονάδων επεξεργασίας (κόμβων), οι οποίοι προσομοιώνουν τους νευρώνες του ανθρώπινου εγκεφάλου, και που δουλεύουν σε πλήρη συμφωνία μεταξύ τους. Τα τεχνητά νευρωνικά δίκτυα, όπως και ο άνθρωπος, έχουν τη δυνατότητα μάθησης μέσα από παραδείγματα. Στα βιολογικά συστήματα αυτό επιτυγχάνεται με την τροποποίηση των (συνοπτικών) συνδέσεων μεταξύ των νευρώνων. Η ίδια διαδικασία εφαρμόζεται και στα νευρωνικά δίκτυα.

#### 2.1.2 Λόγοι χρησιμοποίησης των ΤΝΔ

Λόγω της ικανότητας των νευρωνικών δικτύων να βρίσκουν λύσεις σε περιπτώσεις με πολύπλοκα ή αποσπασματικά δεδομένα, χρησιμοποιούνται για να παρατηρήσουν δεδομένα και να εξάγουν συμπεράσματα εξαιρετικά πολύπλοκα για να παρατηρηθούν από ανθρώπους ή με τη χρήση άλλων τεχνικών.

Τα πλεονεκτήματα ενός νευρωνικού δικτύου περιλαμβάνουν:

- *Δυνατότητα προσαρμοστικής μάθησης.* Το δίκτυο είναι σε θέση να μάθει να εκτελεί διαδικασίες βασισμένες στα δεδομένα που του δίνονται.
- *Αυτοοργάνωση.* Ένα νευρωνικό δίκτυο είναι σε θέση να οργανώνεται με την πάροδο του χρόνου και με βάση τα δεδομένα που δέχεται από τις διάφορες εισόδους του.
- *Επεξεργασία σε συνθήκες πραγματικού χρόνου.* Οι υπολογισμοί ενός νευρωνικού δικτύου γίνονται ιδιαίτερα γρήγορα, αφού η σχεδιάσή τους είναι βασισμένη σε παράλληλη αρχιτεκτονική των επεξεργαστών του συστήματος.
- *Δυνατότητα διόρθωσης λαθών.* Μερική καταστροφή του δικτύου έχει ως αποτέλεσμα τη μείωση της ταχύτητας επεξεργασίας. Τα μερικώς κατεστραμμένα νευρωνικά δίκτυα είναι σε θέση να αναδιατάσσουν τις

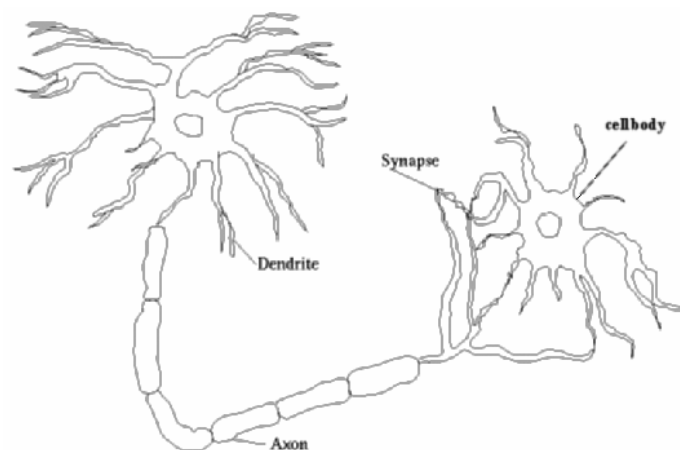
διεργασίες επεξεργασίας του δικτύου διοχετεύοντας όλη τη διαδικασία επεξεργασίας πληροφοριών στο υγιές τμήμα του νευρωνικού δικτύου.

### 2.1.3 Δομή και λειτουργία του ΤΝΔ

Τα ΤΝΔ είναι μοντέλα εμπνευσμένα από βιολογικά νευρικά συστήματα. Για να κατανοήσουμε λοιπόν η λειτουργία των ΤΝΔ καλό θα ήταν μελετηθεί πρώτα η δομή και η λειτουργία ενός βιολογικού συστήματος.

#### 2.1.3.1 Βιολογικό μοντέλο

Το δομικό στοιχείο των συστημάτων αυτών είναι ο νευρώνας (σχήμα 2.1). Το κεντρικό μέρος ονομάζεται σώμα του κύτταρου (cell body). Από εκεί εξέρχονται κάποιες επεκτάσεις οι οποίες ονομάζονται δενδρίτες (dendrites), όπως επίσης και ένα επιμήκη, σωληνοειδές νεύρο ο άξονας (axon), ο οποίος διαχωρίζεται στο τέλος του σε έναν αριθμό κλάδων. Οι δενδρίτες λειτουργούν ως λήπτες σημάτων από διπλανούς νευρώνες, ενώ ο σκοπός του άξονα είναι η μετάδοση της δραστηριότητας του νευρώνα σε άλλα νευρικά κύτταρα.



Σχήμα 2. Error! Unknown switch argument. **Συστατικά στοιχεία ενός νευρώνα**

Η ένωση μεταξύ του τέλους ενός αξονικού κλάδου και ενός άλλου νευρώνα ονομάζεται σύναψη (synapse). Οι συνάψεις μπορεί να βρίσκονται είτε στο σώμα του κύτταρου, είτε στους δενδρίτες των επόμενων νευρώνων. Η επίδραση γενικά ελαττώνεται όσο αυξάνεται η απόσταση από το σώμα.

Στον ανθρώπινο εγκέφαλο, ένας τυπικός νευρώνας συλλέγει σήματα από άλλους νευρώνες μέσω των δενδριτών. Ο νευρώνας στέλνει παλμούς ηλεκτρικής δραστηριότητας μέσω του άξονα, ο οποίος διαχωρίζεται σε πολλούς κλάδους. Στο τέλος καθενός από τους κλάδους αυτούς βρίσκεται η σύναψη, η οποία μετατρέπει τη δραστηριότητα από τον άξονα σε ηλεκτρικά αποτελέσματα, τα οποία αποτρέπουν ή

διεγείρουν τη δραστηριότητα στους διασυνδεδεμένους νευρώνες. Όταν ένας νευρώνας δεχθεί είσοδο μεγαλύτερη από μια συγκεκριμένη τιμή κατωφλίου, τότε διεγείρεται και στέλνει παλμό ηλεκτρικής δραστηριότητας μέσω του άξονα του. Η εκπαίδευση πραγματοποιείται τροποποιώντας δυναμικά τη δομή των συνάψεων, έτσι ώστε να αλλάζει η επίδραση ενός νευρώνα σε κάποιον άλλο.

### 2.1.3.2 Μοντέλο TNA

Καθ' ομοίωση λοιπόν με τα βιολογικά συστήματα αναπτύχθηκαν και τα TNA τα οποία αποτελούνται από ένα σύνολο απλών μονάδων επεξεργασίας (νευρώνων), οι οποίες επικοινωνούν μεταξύ τους μέσω μεγάλου αριθμού συνδέσεων.

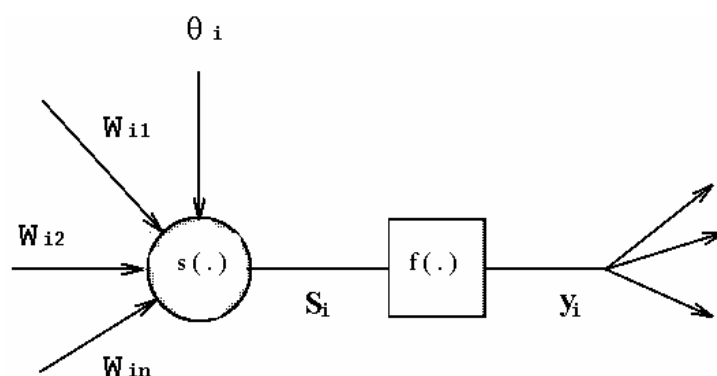
Τα κύρια στοιχεία του δικτύου είναι :

- Ένα σύνολο μονάδων επεξεργασίας (νευρώνες).
- Μια κατάσταση διέγερσης  $y_i$  για κάθε μονάδα επεξεργασίας  $i$ , η οποία είναι ισοδύναμη με την έξοδο της μονάδας.
- Συνδέσεις μεταξύ των μονάδων. Για κάθε σύνδεση ορίζεται ένα βάρος  $w_{ij}$ , το οποίο καθορίζει πως το σήμα της μονάδας  $j$  επηρεάζει το σήμα της μονάδας  $i$ .
- Ένα κανόνα διάδοσης, ο οποίος καθορίζει την ενεργή είσοδο  $s_i$  της μονάδας από τις εξωτερικές εισόδους.
- Μια εξωτερική πόλωση ( threshold)  $\theta_i$  για κάθε μονάδα  $i$ .
- Μια συνάρτηση διέγερσης  $f_i$ , η οποία καθορίζει τη νέα κατάσταση της μονάδας συναρτήσει της ενεργής εισόδου και της τρέχουσας κατάστασης της μονάδας.
- Ένα περιβάλλον στο οποίο πρέπει το TNA να λειτουργεί και το οποίο παρέχει τα σήματα εισόδου.

### 2.1.3.3 Λειτουργία δομικών στοιχείων του TNA

Η λειτουργία του κάθε νευρώνα είναι να δέχεται τις εισόδους των γειτονικών νευρώνων, καθώς και τις εξωτερικές εισόδους (αν υπάρχουν) και να υπολογίζει το σήμα εξόδου το οποίο μεταδίδεται στους άλλους νευρώνες. Επίσης, εκτελεί (αν χρειάζεται) την προσαρμογή βαρών.

Κατά τη λειτουργία του ΤΝΔ οι νευρώνες μπορούν να ενημερώνονται σύγχρονα ή ασύγχρονα. Στη σύγχρονη ενημέρωση, όλες οι μονάδες ενημερώνουν ταυτόχρονα την κατάσταση διέγερσης τους, ενώ στην ασύγχρονη μια μόνο μονάδα ενημερώνει την κατάσταση διέγερσης σε κάθε χρονική στιγμή  $t$ .



Σχήμα 2. Error! Unknown switch argument. Διάγραμμα ενός τεχνητού νευρώνα

Υποθέτουμε ότι στην είσοδο κάθε νευρώνα  $i$  (σχήμα 2.2) υπάρχει αθροιστής ο οποίος αθροίζει τα εισερχόμενα στη μονάδα σήματα. έτσι, η συνολική είσοδος  $s_i$  της μονάδας είναι το σταθμισμένο άθροισμα των εξόδων των μονάδων με τις οποίες είναι συνδεδεμένη, συν την εξωτερική πόλωση  $\theta_i$  της μονάδας :

$$s_i(t) = \sum_j w_{ij}(t) y_j(t) - \theta_i(t)$$

Στη συνέχεια υπολογίζεται η έξοδος του νευρώνα εφαρμόζοντας κάποιον κανόνα ενεργοποίησης, ο οποίος δίνει το αποτέλεσμα της επίδρασης όλων των εισόδων στην κατάσταση διέγερσης του νευρώνα. Αυτό γίνεται χρησιμοποιώντας τη συνάρτηση διέγερσης  $f_i$  η οποία δέχεται τη συνολική είσοδο  $s_i$  και την τρέχουσα κατάσταση διέγερσης  $y_i$  και παράγει τη νέα κατάσταση διέγερσης του νευρώνα :

$$y_i(t+1) = f_i \left( s_i(t), y_i(t) \right)$$

#### 2.1.4 Αρχιτεκτονική ΤΝΔ

Τα ΤΝΔ αποτελούνται από στρώματα (layers) τεχνητών νευρώνων τα οποία μπορούν να θεωρηθούν ως σύνολα νευρώνων που ανήκουν σε ένα επίπεδο. Τα βασικά είδη των layers είναι :

- Το στρώμα εισόδου (input layers), οι οποίες δέχονται δεδομένα έξω από το ΤΝΔ
- Το στρώμα εξόδου (output layers), οι οποίες στέλνουν δεδομένα έξω από το ΤΝΔ
- τα κρυμμένα στρώματα (hidden layers), των οποίων οι είσοδοι και εξοδοι παραμένουν μέσα στο ΤΝΔ.

Τα ΤΝΔ μπορούν να θεωρηθούν ως κατευθυνόμενοι γράφοι, στους οποίους οι τεχνητοί νευρώνες είναι οι κομβοί οι οποίοι συνδέονται μεταξύ τους με κατευθυνόμενες ακμές στις οποίες ανατίθεται ένα βάρος. Οι συνδέσεις επιτρέπονται μόνο από ένα layer στο layer του αμέσως επόμενου επιπέδου .

Ανάλογα με το είδος της σύνδεσης μεταξύ των layers των νευρώνων τα ΤΝΔ διακρίνονται σε:

- *Feed-forward*, όπου τα δεδομένα διαδίδονται μόνο από την είσοδο προς την έξοδο του δικτύου. Η επεξεργασία των δεδομένων μπορεί να εκτείνεται σε πολλά επίπεδα μονάδων επεξεργασίας, αλλά δεν υπάρχουν συνδέσεις ανατροφοδότησης, οι οποίες να τροφοδοτούν μονάδες προηγούμενου επιπέδου με εξόδους μονάδων επόμενου επιπέδου.
- *Recurrent*, τα οποία σε αντίθεση με τα feed-forward περιέχουν συνδέσεις ανατροφοδότησης από υψηλότερα σε χαμηλότερα επίπεδα.

#### 2.1.5 Εκπαίδευση ΤΝΔ

Ένα ΤΝΔ πρέπει να εκπαιδευτεί, έτσι ώστε η εφαρμογή ενός συνόλου πρότυπων δεδομένων εισόδου να παράγει το επιθυμητό σύνολο εξόδου. υπάρχουν δυο βασικοί τύποι εκπαίδευσης: είτε μπορούμε να υπολογίσουμε απ' ευθείας τις τιμές των βαρών, είτε να εκπαιδεύσουμε το δίκτυο και να το αφήσουμε να μεταβάλλει τα βάρη του σύμφωνα με κάποιον κανόνα εκμάθησης.

Οι μέθοδοι εκμάθησης χωρίζονται σε 2 βασικούς τύπους :

- *Επιβλεπόμενη εκμάθηση* (supervised learning) όπου το ΤΝΔ εκπαιδεύεται με συγκεκριμένες εισόδους και τις αντίστοιχες εξόδους , οι οποίες συσχετίζονται με τις εισόδους. Τα ζεύγη εισόδων – εξόδων δίνονται εξωτερικά στο ΤΝΔ .
- *Μη επιβλεπόμενη εκμάθηση* ή αυτοοργάνωση (unsupervised learning or self-organization), όπου μια μονάδα εξόδου εκπαιδεύεται να ανταποκρίνεται σε ομάδες πρότυπων που υπάρχουν στην είσοδο. Σε αντίθεση με την



επιβλεπόμενη μάθηση εδώ δεν υπάρχουν σύνολα κατηγοριών στα οποία θα ταξινομηθούν τα πρότυπα, αλλά το σύστημα πρέπει να αναπτύξει τη δική του αναπαράσταση των διεγέρσεων εισόδου.

Έτσι, τα ΤΝΔ χωρίζονται στις κατηγορίες:

- *Επιβλεπόμενα ΤΝΔ*, όπου εφαρμόζουμε επιβλεπόμενη εκμάθηση
- *Μη επιβλεπόμενα ΤΝΔ*, όπου εφαρμόζουμε μη επιβλεπόμενη εκμάθηση
- *ΤΝΔ με σταθερά βάρη*, όπου δεν εφαρμόζεται κανένας κανόνας εκμάθησης αφού τα βάρη των συνδέσεων υπολογίζονται εκ των πρότερων και παραμένουν σταθερά κατά τη λειτουργία του δικτύου.

## 2.2 Συσχετιστικές μνήμες (associative memories)

Οι συσχετιστικές μνήμες βασίζονται σε μια από τις βασικότερες ιδιότητες του ανθρώπινου εγκεφάλου: αυτή να μπορεί να αναγνωρίζει ένα αντικείμενο από τα χαρακτηριστικά του (π.χ. ένας άνθρωπος μπορεί να αναγνωρίσει έναν αριθμό γραμμένο με διαφορετικούς γραφικούς χαρακτήρες ή να αναγνωρίσει μια εικόνα από μια εκφυλισμένη έκδοση της ).

Κατ' αναλογία με τα παραπάνω, η λειτουργία των συστημάτων αυτών είναι να συσχετίζουν ένα set από patterns εισόδου με ένα set από patterns εξόδου.

Τα μοντέλα αυτά χρησιμοποιούνται κυρίως για την αναπαραγωγή σημάτων από ατελείς ή αλλοιωμένες από θόρυβο εκδοχές τους.

Το πιο γνωστό ΤΝΔ που υλοποιεί την ιδέα των συσχετιστικών μνημών είναι το δίκτυο Hopfield.

## 2.3 Το δίκτυο Hopfield

Το δίκτυο αυτό πρωτοπαρουσιάστηκε το 1982 από τον J. Hopfield. Η κύρια λειτουργία τους είναι να αποθηκεύουν ένα set από  $M$  patterns, τα οποία ονομάζονται συσχετιστικές μνήμες (associative memories) ή ελκυστές (attractors). Όταν ένα νέο pattern  $X$  διαδίδεται στο δίκτυο, τότε η έξοδος του δικτύου θα πρέπει είναι κάποιο από τα αποθηκευμένα patterns το οποίο μοιάζει περισσότερο με το  $X$ .

Η δομή των μονάδων (νευρώνων) του δικτύου είναι ίδια με αυτήν του σχήματος 2.2, δηλαδή του προτύπου του στοιχειώδους νευρώνα. Κάθε νευρώνας χαρακτηρίζεται από μία δυαδική κατάσταση, δηλαδή μπορεί να έχει μία από δύο δυνατές τιμές. Συνήθως οι τιμές αυτές είναι 1 και -1, και αντιστοιχούν στο αν κάποιος νευρώνας είναι ενεργοποιημένος ή όχι .

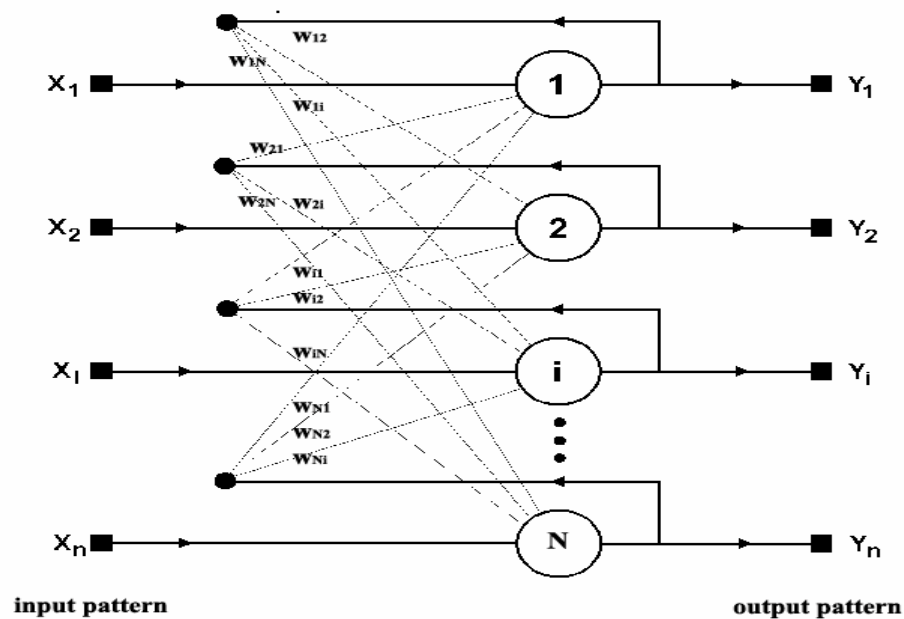
### 2.3.1 Τοπολογία του δικτύου

Τα δίκτυα τύπου Hopfield είναι ανατροφοδοτούμενα και αποτελούνται από ένα μόνο επίπεδο (layer) νευρώνων (σχήμα 2.3). Οι μονάδες του δικτύου έχουν πλήρη συνδεσμολογία, δηλαδή κάθε μονάδα συνδέεται (ανατροφοδοτεί) με κάθε άλλη μονάδα στο σύστημα. Καμία μονάδα δεν ανατροφοδοτεί τον εαυτό της, δεν υπάρχουν δηλαδή αυτό-συνδέσεις. Έτσι, για ένα δίκτυο με  $N$  μονάδες αυτό σημαίνει ότι θα έχουμε  $N(N-1)$  συνδέσεις. Οι συνδέσεις έχουν και συγκεκριμένη κατεύθυνση. Σε κάθε ζευγάρι μονάδων που συνδέονται υπάρχει σύνδεση και προς τις δύο κατευθύνσεις, δηλαδή μεταξύ των μονάδων  $i$  και  $j$  υπάρχει η σύνδεση  $w_{ij}$  και η

σύνδεση  $w_{ji}$ . Όμως, το δίκτυο είναι συμμετρικό, οπότε θα έχουμε πάντοτε

$$w_{ij} = w_{ji}.$$

Το μέγεθος  $N$  του δικτύου (ο αριθμός των νευρώνων δηλαδή) ισούται με το μέγεθος των αποθηκευμένων patterns. Στο σχήμα μας το  $X_i$  αντιπροσωπεύει τα στοιχεία του pattern που διαδίδεται στο δίκτυο (δηλαδή την είσοδο του δικτύου), ενώ το  $Y_i$  αντιπροσωπεύει τα στοιχεία της εξόδου του δικτύου, όπου  $i = 1..N$ .



Σχήμα 2. Error! Unknown switch argument. Τοπολογία του δικτύου Hopfield

### 2.3.2 Λειτουργία του δικτύου

Η λειτουργία χωρίζεται σε 2 φάσεις : τη φάση αποθήκευσης και τη φάση ανάκλησης.

#### 2.3.2.1 Φάση αποθήκευσης

Κατά τη φάση αυτή το δίκτυο αποθηκεύει ένα με ένα set από  $M$  μνήμες (τις associative memories ) με  $N$  στοιχεία η καθεμία. Θα συμβολίζουμε τις associative memories με  $p_i^k$ , όπου  $i = 1..N$  και  $k = 1..M$ .

Η εκπαίδευση του δικτύου, δηλαδή ο υπολογισμός των βαρών των συνδέσεων, γίνεται σύμφωνα με το Hebbian learning rule .

Η σύνδεση  $w_{ij}$  μεταξύ των μονάδων  $i$  και  $j$  δίνεται από τον τύπο :

$$w_{ij} = \frac{1}{N} \sum_{k=1}^M p_j^k p_i^k \quad \text{για } i \neq j$$

Ο πίνακας είναι συμμετρικός αφού ισχύει ότι :

$$w_{ij} = \frac{1}{N} \sum_{k=1}^M p_i^k p_j^k = w_{ji}$$

Τέλος, δεν υπάρχει ‘αυτοσύνδεση’ μεταξύ των κόμβων του δικτύου

$$w_{ii} = 0$$

Σημειώνεται εδώ ότι ο παραπάνω κανόνας δουλεύει ικανοποιητικά για περιπτώσεις όπου οι associative memories είναι ασυσχέτιστες (uncorrelated) ή έχουν χαμηλό συντελεστή συσχέτισης.

#### 2.3.2.2 Φάση ανάκλησης

Για να περιγράψει τη διαδικασία που ακολουθείται κατά τη φάση αυτή θα χρησιμοποιηθούν οι συμβολισμοί της παραγράφου 2.1.3.2.

Έστω ότι στο δίκτυο εμφανίζεται ένα pattern  $X$  με στοιχεία  $\left( x_1, \dots, x_N \right)$ .

Το δίκτυο ακολουθώντας μια δυναμική διαδικασία θα διαδώσει το pattern  $X$  και θα αρχίσει να ανανεώνει επαναληπτικά τους νευρώνες του ασύγχρονα και με τυχαία σειρά. Αφού το δίκτυο είναι ανατροφοδοτούμενο ανάμεσα σε 2 διαδοχικές ανανεώσεις το δίκτυο θα τροφοδοτεί συνεχώς τα σήματα εισόδου με τα σήματα εξόδου έως ότου επιτευχθεί σύγκλιση. Αυτό θα γίνει όταν η έξοδος του δικτύου θα σταθεροποιηθεί σε μια κατάσταση, η οποία συνήθως συμπίπτει με κάποια από τις associative memories.

Αρχικά, οι μονάδες εισόδου θέτονται ίσες με τα στοιχεία του pattern  $X$  και στη συνέχεια αρχίζει η ανανέωση της κατάστασης των νευρώνων του δικτύου σύμφωνα με την εξίσωση :

$$y_i(t+1) = \text{sgn} \left\{ \sum_{j=1, j \neq i}^N w_{ij} y_j(t) - \theta_i \right\}$$

με συνάρτηση ενεργοποίησης την  $\text{sgn}(x) = \begin{cases} +1, x \geq 0 \\ -1, x < 0 \end{cases}$

Το δίκτυο θα συνεχίσει την ανανέωση έως ότου επιτευχθεί σύγκλιση. Αυτό συμβαίνει όταν ανάμεσα σε δυο διαδοχικές ανανεώσεις (όλων των νευρώνων) κανένας από αυτούς δεν αλλάξει κατάσταση.

### 2.3.3 Θεωρητική βάση

Όλη η λειτουργία των δικτύων βασίζεται στην ελαχιστοποίηση της συνάρτησης ενέργειας τους. Αυτή η ιδιότητα, μας εγγυάται και ότι πάντα τα δίκτυα θα συγκλίνουν σε μια κατάσταση μετά από έναν πεπερασμένο αριθμό επαναλήψεων, καθώς επίσης και ότι η κατάσταση αυτή θα αντιπροσωπεύει μια συσχετιζόμενη μνήμη.

Αυτό αποδεικνύεται ως εξής :

Το δίκτυο Hopfield χαρακτηρίζεται από την ενέργεια  $E$  η οποία δίνεται από:

$$E = -\frac{1}{2} \sum_j \sum_i w_{ji} y_j y_i \quad i \neq j$$

Με τις ανανεώσεις που πραγματοποιούνται στα  $\mu$ (καταστάσεις των νευρώνων) , το  $E$  αλλάζει με τον χρόνο Σε κάθε ανανέωση η ενέργεια  $E$  ή ελαττώνεται ή παραμένει η ίδια.

Όταν αναπροσαρμόζεται ο νευρώνας  $j$  τότε ισχύει :

$$\begin{aligned} E_j &= -\frac{1}{2} \sum_i w_{ji} y_j y_i \\ &= -\frac{1}{2} y_j \sum_i w_{ji} y_i, \quad i \neq j \end{aligned}$$

Η αλλαγή στην ενέργεια θα είναι :

$$\Delta E_j = E_j(\nu\epsilon\alpha) - E_j(\pi\alpha\lambda\iota\alpha) = -\frac{1}{2} \Delta y_j \sum_i w_{ji} y_i$$

όπου  $\Delta y_j = y_j(\nu\epsilon\alpha) - y_j(\pi\alpha\lambda\iota\alpha)$

Ανάλογα με τις τιμές του αθροίσματος  $\sum_i w_{ji} y_i$  έχουμε τις περιπτώσεις :

αν  $\sum_i w_{ji} y_i \geq 0$  τότε  $y_j(\nu\epsilon\alpha) = 1$  και  $\Delta y_j > 0$ , όποτε  $\Delta E_j < 0$

αν  $\sum_i w_{ji} y_i < 0$  τότε  $y_j(\nu\epsilon\alpha) = -1$  και  $\Delta y_j < 0$  όποτε  $\Delta E_j < 0$

Παρατηρούμε ότι η αλλαγή του E είναι αρνητική ή μηδέν, για οποιοδήποτε αλλαγή της κατάστασης του j.

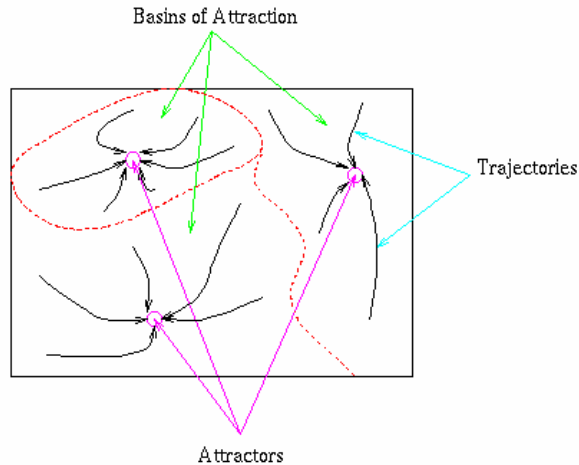
Αυτό σημαίνει πως κατά τη λειτουργία του δικτύου το pattern το οποίο διαδίδεται στο δίκτυο θα βαίνει προς μια κατάσταση, η οποία αντιπροσωπεύει ένα τοπικό ελάχιστο της συνάρτησης ενέργειας και βρίσκεται πιο κοντά στην αρχική κατάσταση του pattern.

Επίσης, το γεγονός ότι μετά από κάθε επανάληψη η ενέργεια του δικτύου μειώνεται μας εξασφαλίζει ότι το δίκτυο πάντοτε θα συγκλίνει σε μια σταθερή κατάσταση μετά από έναν πεπερασμένο αριθμό επαναλήψεων. Αυτό προφανώς συμβαίνει γιατί εάν το δίκτυο φτάσει σε ένα τοπικό ελάχιστο τότε η ενέργεια του δεν θα μπορεί να μπορεί να μειωθεί άλλο, οπότε μετά από δεν θα ξεφύγει ποτέ από την κατάσταση αυτή. Τα ελάχιστα της συνάρτησης ενέργειας ονομάζονται και *attractors*, αφού έλκουν τα patterns που παρουσιάζονται στο δίκτυο.

Οι κανόνες εκπαίδευσης που χρησιμοποιούμε, μας εξασφαλίζουν ότι οι συσχετιστικές μνήμες του δικτύου είναι τοπικά ελάχιστα της συνάρτησης ενέργειας.

Έτσι, κάθε pattern που παρουσιάζεται στο δίκτυο θα ακολουθεί μια συγκεκριμένη διαδρομή (από κατάσταση σε κατάσταση) μέσα σε αυτό, η οποία ονομάζεται τροχιά (*trajectory*), μέχρι να φτάσει σε μια σταθερή κατάσταση (*attractor*). Ο χώρος στον οποίο ένα pattern που θα βρεθεί μέσα σε αυτόν θα συγκλίνει σε μια συγκεκριμένη σταθερή κατάσταση (δηλαδή σε έναν *attractor*) ονομάζεται περιοχή σύγκλισης (*basin of attractor*). Τα παραπάνω φαίνονται πιο ξεκάθαρα στο σχήμα 2.4.

Όμως συχνά το δίκτυο συγκλίνει σε έναν *attractor* ο οποίος δεν συμπίπτει με καμία από τις συσχετιστικές μνήμες. Οι *attractors* αυτοί ονομάζονται *spurious attractors*.



Σχήμα 2. Error! Unknown switch argument. Διάγραμμα στο χώρο της ενέργειας

## 2.4 Προβλήματα των δικτύων Hopfield

- *Capacity (χωρητικότητα)*: Αναφέρεται στην ικανότητα του δικτύου να μπορεί να αποθηκεύει και να ανακαλεί αποτελεσματικά έναν αριθμό  $M$  συσχετιστικών μνημών. Έχει αποδειχθεί ότι στο δίκτυο μας η χωρητικότητα είναι :  $M < 14\%N$ .
- Επίσης, για να έχουμε σωστά recalls θα πρέπει :  $M < \frac{N}{2 \ln N}$ .
- *Spurious attractors*: Αναφέρεται στην ύπαρξη καταστάσεων όπου το δίκτυο σταθεροποιείται χωρίς αυτές να συμπίπτουν με κάποια associative memory. Spurious attractors έχουμε όταν οι μνήμες που αποθηκεύονται στο δίκτυο δεν είναι ορθογώνιες μεταξύ τους (οι καταστάσεις αυτές είναι γνωστές ως mixture states). Τότε, η έξοδος του δικτύου θα είναι κάποιος γραμμικός συνδυασμός περιττού αριθμού μνημών. Άλλη μια περίπτωση είναι όταν ο αριθμός των συσχετιστικών μνημών είναι αρκετά μεγάλος (οι καταστάσεις αυτές είναι γνωστές ως spin-glass states). Τότε, η έξοδος του δικτύου δεν θα συσχετίζεται με καμία από τις μνήμες.
- *Basins of attractions*: Αναφέρεται στο μέγεθος του basin ενός attractor. Αυτό που θέλουμε είναι οι μνήμες μας να παρουσιάζουν basins μεγαλύτερα από αυτά των spurious attractors.
- *Μνήμες με υψηλό ποσοστό συσχέτισης*. Πολύ σημαντικό στοιχείο είναι η ικανότητα του δικτύου να αποθηκεύει και να ανακαλεί ικανοποιητικά τις correlated memories. Ο κανόνας εκπαίδευσης του Hebb που προτάθηκε για το αρχικό δίκτυο δεν είναι καθόλου αποτελεσματικός για την περίπτωση αυτή. Εάν υπάρχει έστω και μικρό ποσοστό συσχέτισης, τότε η απόκριση του δικτύου θα περιέχει ένα ποσοστό αλλοιωμένων στοιχείων, τα οποία

αντιστοιχούν στα στοιχεία εκείνα για τα οποία συμβαίνει η συσχέτιση. Αυτό το φαινόμενο ονομάζεται cross talk. Όσο μεγαλύτερο είναι το ποσοστό συσχέτισης, τόσο μεγαλύτερος είναι και ο παράγοντας cross talk.

- *Επιλογή κατάλληλων thresholds:* Χωρίς thresholds ένας σημαντικός αριθμός test patterns θα τείνει στο global min της energy function. Με την εισαγωγή thresholds σε κάθε μονάδα του δικτύου πετυχαίνουμε να μειώσουμε την ευαισθησία (σε αλλαγές καταστάσεων) των νευρώνων οι οποίοι παρουσιάζουν υψηλή δραστηριότητα. Για κάθε κανόνα εκπαίδευσης απαιτείται διαφορετική επιλογή thresholds.
- *Σειρά ανανέωσης των νευρώνων κατά τη φάση ανάκλησης:* Το trajectory που θα ακολουθήσει ένα pattern μέχρι να φτάσει σε έναν attractor εξαρτάται από τη Σειρά με την οποία ανανεώνονται οι νευρώνες του δικτύου, αφού κάθε αλλαγή στην κατάσταση ενός νευρώνα θα τροφοδοτεί τις εισόδους όλων των υπόλοιπων (αφού το δίκτυο είναι ανατροφοδοτούμενο). Έτσι διαφορετική Σειρά ανανέωσης συνεπάγεται διαφορετικό trajectory που στις περισσότερες περιπτώσεις οδηγεί σε διαφορετικό attractor.
- *Learning rules:* Αναφέρεται στην εύρεση κανόνων να δουλεύουν ικανοποιητικά για correlated memories, να αυξάνουν τη χωρητικότητα του δικτύου και γενικά να επιλύουν ικανοποιητικά όλα τα παραπάνω προβλήματα.

## 2.5 Τροποποιήσεις του αρχικού μοντέλου

Για να λυθούν τα παραπάνω προβλήματα έχουν διατυπωθεί πολλές Τροποποιήσεις του αρχικού μοντέλου, που αφορούν κυρίως την αύξηση της χωρητικότητας, την μείωση του αριθμού των spurious attractors και τη αύξηση της απόδοσης όταν οι μνήμες παρουσιάζουν κάποιο ποσοστό συσχέτισης.

### 2.5.1 Επιλογή του κατάλληλου learning rule

Στην παράγραφο αυτή παρουσιάζονται δυο από τους γνωστότερους κανόνες εκμάθησης στα δίκτυα Hopfield, τους οποίους χρησιμοποιούμε στην εφαρμογή μας.

#### 2.5.1.1 Projection (or inverse) learning rule

Όπως αναφέρθηκε παραπάνω, όταν οι μνήμες οι οποίες αποθηκεύονται στο δίκτυο δεν είναι ορθογώνιες μεταξύ τους, τότε ο κανόνας Hebb αδυνατεί να τις αποθηκεύσει αποτελεσματικά. Έτσι αυξάνεται κατά πολύ ο αριθμός των spurious states του δικτύου. Για την επίλυση του προβλήματος αυτού, αναπτύχθηκε ο projection learning rule ο οποίος δίνεται από τις σχέσεις :

$$w_{ij} = \frac{1}{N} \sum_{\mu=1}^M \sum_{\nu=1}^M p_i^{\mu} Q_{\mu\nu}^{-1} p_{\nu}^{\nu}$$

όπου

$$Q_{\mu\nu} = \frac{1}{N} \sum_{i=1}^N p_i^\mu p_i^\nu$$

Όταν οι αποθηκευμένες στο δίκτυο μνήμες είναι ορθογώνιες, τότε ο projection learning rule συμπίπτει με το Hebbian learning Rule, αφού:

$$Q_{\mu\nu} = \frac{1}{N} \sum_{i=1}^N p_i^\mu p_i^\nu = I$$

Για τις περιπτώσεις που ο πίνακας  $Q$  δεν είναι αντιστρέψιμος, έχουν αναπτυχθεί διάφορες επαναληπτικές μέθοδοι (γνωστότερος ο pseudo-inverse learning rule) που υπολογίζουν τον πίνακα  $Q^{-1}$  αρκετά ικανοποιητικά.

#### 2.5.1.2 Learning in bounds

Αρκετό ενδιαφέρον παρουσιάζει ο κανόνας αυτός ο οποίος είναι μια παραλλαγή του Hebbian learning rule. Κύρια ιδιότητα του είναι να συσχετίζει κάποια pattern  $x_i$  με

μια μνήμη  $p_j^m$  (ιδιότητα ετεροσυσχετιζόμενων μνημών). Σημειώνεται επίσης ότι η μέθοδος αυτή απαιτεί το δίκτυο να ανανεώνει τους νευρώνες του παράλληλα (μόνο για την περίπτωση αυτή έχει αποδειχθεί ότι το δίκτυο συγκλίνει σε σταθερή κατάσταση).

Σύμφωνα με τον κανόνα αυτό, αφού αρχικά έχει υπολογιστεί ο πίνακας βαρών (από τον κανόνα Hebb) στη συνέχεια αναπροσαρμόζεται σύμφωνα με τον τύπο :

$$\Delta w_{ij} = \alpha x_i p_j^m$$

όπου  $\alpha$  : το ποσοστό μάθησης που ορίζεται από το χρήστη, συνήθως παίρνουμε

$$\alpha = \frac{1}{N}$$

$p$ : κάποια από τις μνήμες

και  $x$  : κάποιο pattern που θέλουμε να συσχετίσουμε με τη μνήμη  $p$ .



## 2.5.2 Επιλογή του κατάλληλου threshold

Εξίσου σημαντική είναι και η επιλογή κατάλληλων thresholds τα οποία σε συνδυασμό με κατάλληλους κανόνες μάθησης θα αυξάνουν την απόδοση του δικτύου. Το μοντέλο που χρησιμοποιήθηκε είναι :

### 2.5.2.1 Adjustable thresholds

Αρχικά υπολογίζουμε τον πίνακα βαρών του δικτύου. Στη συνέχεια εισάγουμε στο δίκτυο τις μνήμες και για καθεμία από αυτές υπολογίζουμε την έξοδο τους για κάθε νευρώνα από τον τύπο :

$$h_i = \sum_j w_{ij} y_j$$

Τα thresholds για κάθε μονάδα  $i$  υπολογίζονται ως εξής:

$$\theta_i = \frac{h_i^+ + h_i^-}{2}$$

$$\text{με } h_i^+ = \min\{h_i^m : h_i^m \geq 0\}$$

$$\text{και } h_i^- = \max\{h_i^m : h_i^m < 0\}$$

### 3 Προτεινόμενη διαδικασία

Όπως αναφέρθηκε και στο κεφάλαιο 1, σκοπός μας είναι ο υπολογισμός της κατακόρυφης μετατόπισης του κάθε lead. Το πρόβλημα αυτό θα το προσεγγίσουμε χρησιμοποιώντας συσχετιστικές μνήμες. Εφόσον οι πιθανές μετατοπίσεις είναι 13 (1..6 pixel πάνω, 1..6 pixel κάτω και 0 pixel) θα δημιουργήσουμε 13 διαφορετικές μνήμες που η καθεμία θα αντιστοιχεί σε μια πιθανή μετατόπιση. Στη συνέχεια, θα συσχετίζουμε την κάθε εικόνα με κάποια από τις μνήμες αυτές. Θα μπορούσαμε λοιπόν να ισχυριστούμε πως το σύστημα μας κατηγοριοποιεί την κάθε εικόνα σε μια από 13 διαφορετικές κλάσεις με την κάθε κλάση να αντιστοιχεί σε μια από τις μνήμες μας.

Συνεπώς, το σύστημα το οποίο προτείνουμε είναι ένας classifier, που στηρίζεται στο ΤΝΔ Hopfield. Στο κεφάλαιο αυτό περιγράφουμε τη διαδικασία που ακολουθούμε για την κατηγοριοποίηση των εικόνων στις αντίστοιχες κλάσεις και η οποία χωρίζεται στα εξής μέρη : *την προεπεξεργασία που απαιτούν οι εικόνες έτσι ώστε να μπορούν να εισαχθούν στο δίκτυο Hopfield, την επιλογή των συσχετιστικών μνημών με βάση τις οποίες θα εκπαιδευτεί το δίκτυο, την επιλογή των κατάλληλων αλγορίθμων για εκπαίδευση και λειτουργία του δικτύου και τέλος την επιλογή κριτηρίου με βάση την οποία θα γίνει η κατηγοριοποίηση της εξόδου του δικτύου Hopfield σε κάποια από τις κλάσεις.* Αναλυτικότερα, τα βήματα αυτά θα είναι :

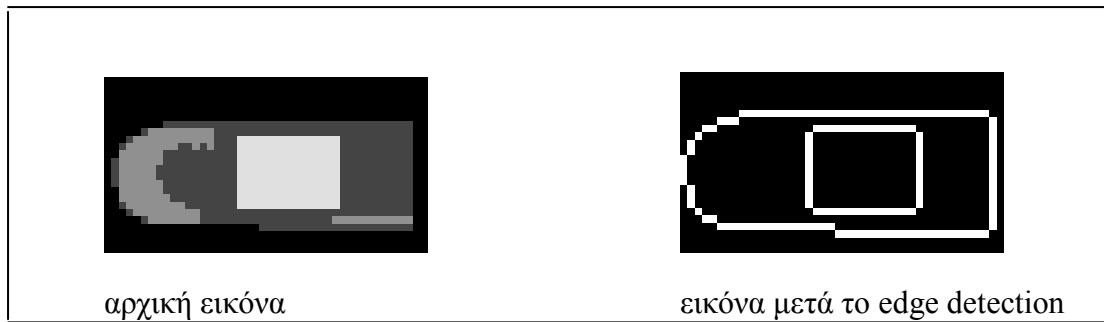
#### 3.1 Προεπεξεργασία

Η φάση της προεπεξεργασίας περιλαμβάνει μια ειδική επεξεργασία των εικόνων πριν την είσοδο τους στο δίκτυο Hopfield.

Πρώτο μέλημα μας είναι η μετατροπή των εικόνων μας σε δυαδικές εικόνες αφού όπως έχει αναφερθεί το δίκτυο απαιτεί τέτοιες εικόνες.

Θα πρέπει να βρούμε μια διαδικασία μετατροπής των εικόνων μας από εικόνες με 4 περιοχές φωτεινότητας (lead, pad, solder και paste background) σε εικόνες με 2 περιοχές φωτεινότητας, διατηρώντας παράλληλα τις σημαντικές πληροφορίες που αυτές περιέχουν , αυτές δηλαδή με βάση τις οποίες θα τις κατηγοριοποιήσουμε στη συνέχεια.

Αυτή ακριβώς η μετατροπή μπορεί να γίνει εφαρμόζοντας στις εικόνες μια μάσκα ανίχνευσης ακμών. Το αποτέλεσμα θα είναι το περίγραμμα του αντικειμένου της κάθε εικόνας (σχήμα 3.1) .



**Σχήμα 3.**Error! Unknown switch argument. **Παράδειγμα εικόνας πριν και μετά το edge detection**

Για να δουλέψει καλύτερα η ανίχνευση ακμών συνενώνουμε αρχικά τις περιοχές pad και solder paste γιατί περιέχουν πληροφορίες που δεν μας είναι απαραίτητες.

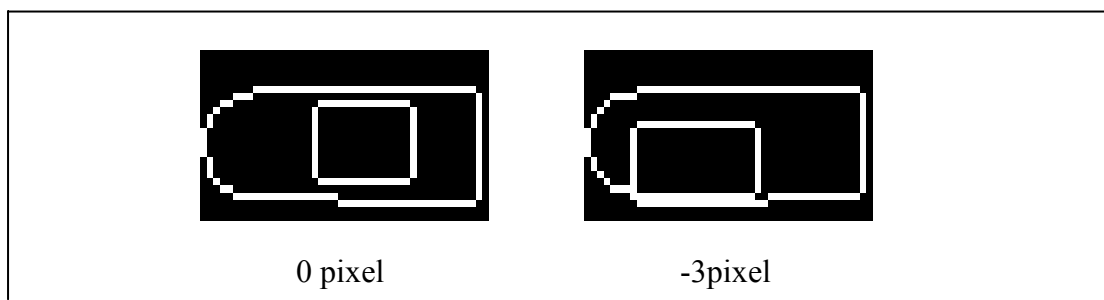
Η ανίχνευση ακμών γίνεται χρησιμοποιώντας τη μάσκα Laplace η οποία είναι :

0	-1	0
-1	4	-1
0	-1	0

Όμως, το αποτέλεσμα της εφαρμογής της μάσκας αυτής δεν δίνει δυαδικές εικόνες. Για να κάνουμε την εικόνα δυαδική προχωρούμε σε thresholding (με  $T=0$ ). Έτσι η εικόνα έχει μόνο 2 περιοχές φωτεινότητας ( 0 και 255).

Το αποτέλεσμα της διαδικασίας αυτής φαίνεται στο σχήμα 3.1 .

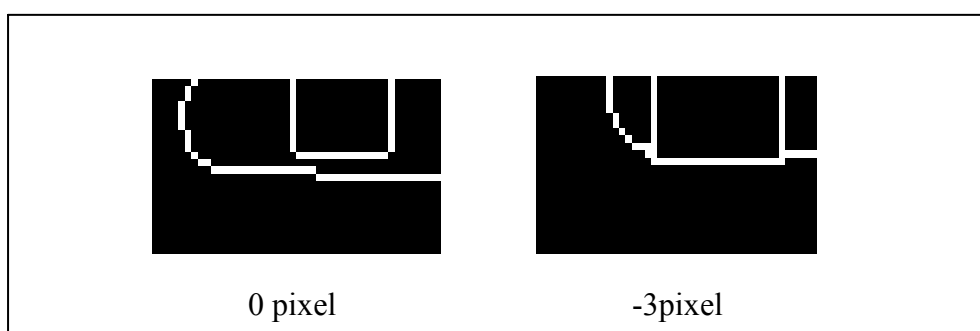
Στο σχήμα 3.2 βλέπουμε 2 εικόνες που ανήκουν σε διαφορετικές κλάσεις, στα 0 και στα -3 pixels αντίστοιχα. Παρατηρούμε ότι το περίγραμμα της περιοχής που προκύπτει από τη συνένωση του pad και του solder paste (που έχει τα περισσότερα pixels) είναι σε σταθερή θέση στην εικόνα, ενώ το περίγραμμα της περιοχής του lead (που έχει λιγότερα pixels) είναι αυτό που διαφοροποιείται.



**Σχήμα 3.**Error! Unknown switch argument. **Παράδειγμα εικόνων διαφορετικών κλάσεων μετά το edge detection**

Πολύ καλό θα ήταν να βρούμε έναν τρόπο να διακριτοποιήσουμε όσο περισσότερο μπορούμε τις εικόνες της κάθε κλάσης. Όπως βλέπουμε και στο σχήμα 3.2 οι εικόνες διαφορετικών κλάσεων, πέρα από τις κατακόρυφες μετατοπίσεις, διαφοροποιούνται μόνο στη θέση του lead. Για να πετύχουμε λοιπόν το σκοπό μας θα πρέπει να βρούμε έναν τρόπο έτσι ώστε η περιοχή του lead (που έχει λιγότερα pixels) να βρίσκεται σε σταθερή θέση στην εικόνα, ενώ η περιοχή που προκύπτει από τη συνένωση του pad και του solder paste (που έχει τα περισσότερα pixels) να διαφοροποιείται.

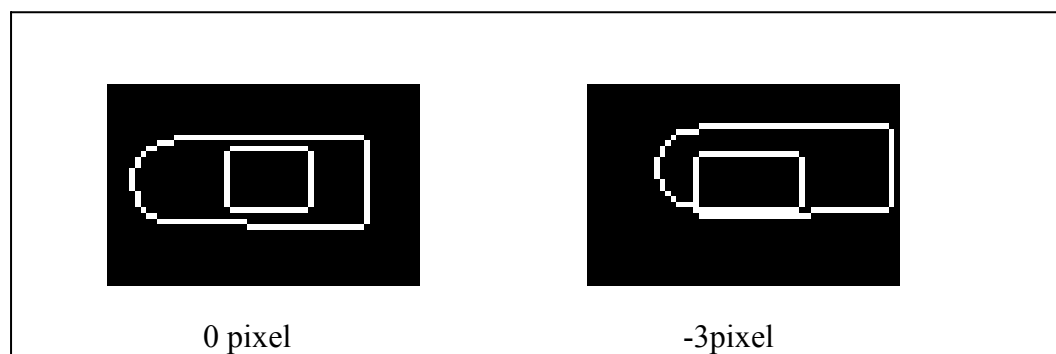
Αυτό μπορούμε να το πετύχουμε μετατοπίζουμε το κέντρο βάρους του κάθε lead στο κέντρο της εικόνας. Έτσι, η περιοχή του lead θα βρίσκεται σε σταθερή θέση στην εικόνα, ενώ η περιοχή που προκύπτει από τη συνένωση του pad και του solder paste θα διαφοροποιείται (σχήμα 3.3).



**Σχήμα 3.** Παράδειγμα εικόνων διαφορετικών κλάσεων μετά τη μετατόπιση του κέντρου βάρους του lead στο κέντρο της εικόνας

Όπως φαίνεται και στο σχήμα 3.3 πράγματι οι εικόνες διαφοροποιούνται μεταξύ τους σε μεγαλύτερο αριθμό pixels, αλλά κάποιες περιοχές των εικόνων με σημαντική πληροφορία μένουν εκτός των ορίων τους. Για την αποφυγή τέτοιων καταστάσεων μεγαλώνουμε το μέγεθος των εικόνων μας. Μετά από πειράματα, καταλήξαμε στο συμπέρασμα πως πρέπει να αυξήσουμε τις διαστάσεις των εικόνων κατά 12 pixels σε μήκος και κατά αλλά τόσα σε πλάτος. Έτσι ώστε σε καμία περίπτωση σημεία που ανήκουν στο περίγραμμα του αντικειμένου μας να μένει εκτός ορίων της εικόνας.

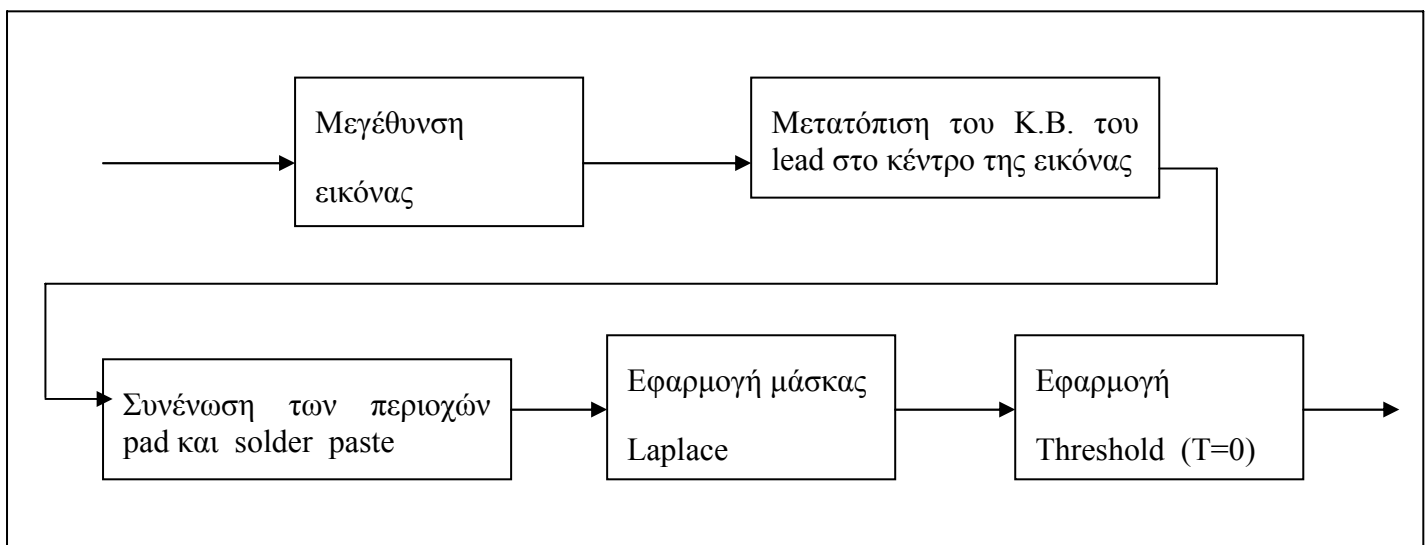
Στο σχήμα 3.4 φαίνονται οι τελικές εικόνες, έχοντας υποστεί όλη τη διαδικασία που περιγράφηκε παραπάνω. Αυτές θα είναι και οι εικόνες που θα εισαχθούν στο δίκτυο Hopfield.



**Σχήμα 3.**Error! Unknown switch argument. **Παράδειγμα εικόνων διαφορετικών κλάσεων μετά την μεγέθυνση τους**

Τελικά λοιπόν , η προεπεξεργασία είναι μια διαδικασία που δέχεται σαν είσοδο μια εικόνα μεγέθους 24x44 pixels με 4 επίπεδα φωτεινότητας και δίνει ως έξοδο μια εικόνα μεγέθους 36x56 pixels με 2 επίπεδα φωτεινότητας έτοιμη να εισαχθεί στο δίκτυο Hopfield.

Η διαδικασία που ακολουθούμε συνοψίζεται στο σχήμα 3.5.



**Σχήμα 3.**Error! Unknown switch argument. **Διάγραμμα της διαδικασίας που ακολουθείται κατά την προεπεξεργασία**

### 3.2 Επιλογή των associative memories

Η επιλογή μνημών είναι ένα πολύ σημαντικό ζήτημα στα δίκτυα των associative memories μια και αυτές καθορίζουν τη λειτουργία του δικτύου.

Στην εφαρμογή μας έχουμε 13 διαφορετικές μετατοπίσεις (0,+1..+6,-1..-6 pixels) και συνεπώς 13 διαφορετικές κλάσεις. Οπότε και ο αριθμός των associative memories θα είναι επίσης 13.

Η μνήμη που αντιπροσωπεύει κάθε κλάση θα πρέπει να περιέχει τα στοιχεία εκείνα που διαχωρίζουν τις κλάσεις μεταξύ τους καθώς επίσης και να είναι ανεξάρτητη των

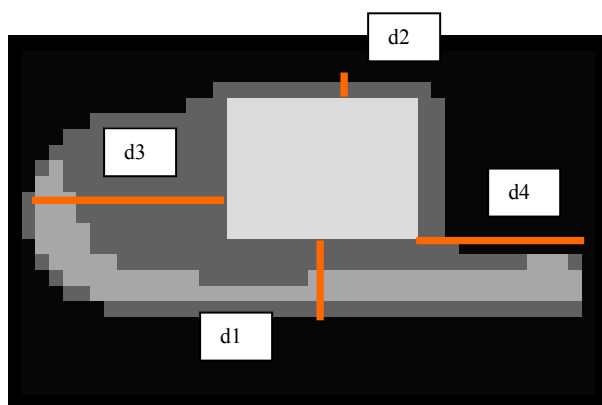
διαφοροποιήσεων που μπορούν να εμφανίζονται στις εικόνες της κάθε κλάσης. Τα στοιχεία τα οποία διαφοροποιούν τις κλάσεις είναι οι κατακόρυφες αποστάσεις του lead από τα όρια του solder paste. Τα στοιχεία από τα οποία θα πρέπει οι κλάσεις να είναι ανεξάρτητες είναι οι μετατοπίσεις του lead από τα όρια του solder paste στον οριζόντιο άξονα και το μέγεθος του lead.

Εννοείται βεβαίως πως οι μνήμες μας θα είναι δυαδικές εικόνες.

Η διαδικασία που προτείνουμε είναι η εξής :

### 3.2.1 Διαδικασία 1

Λαμβάνοντας υπ' όψη τα στοιχεία αυτά υπολογίζουμε τις μέσες τιμές της κάθε κλάσης για τις κατακόρυφες αποστάσεις του lead από τα όρια του solder paste (αποστάσεις d1 και d2 στο σχήμα 3.6). Επίσης, υπολογίζουμε τη μέση τιμή του μεγέθους του lead όπως και τη μέση τιμή των οριζόντιων αποστάσεων του lead από τα όρια του solder paste (αποστάσεις d3 και d4 στο σχήμα 3.6) όλων των κλάσεων .

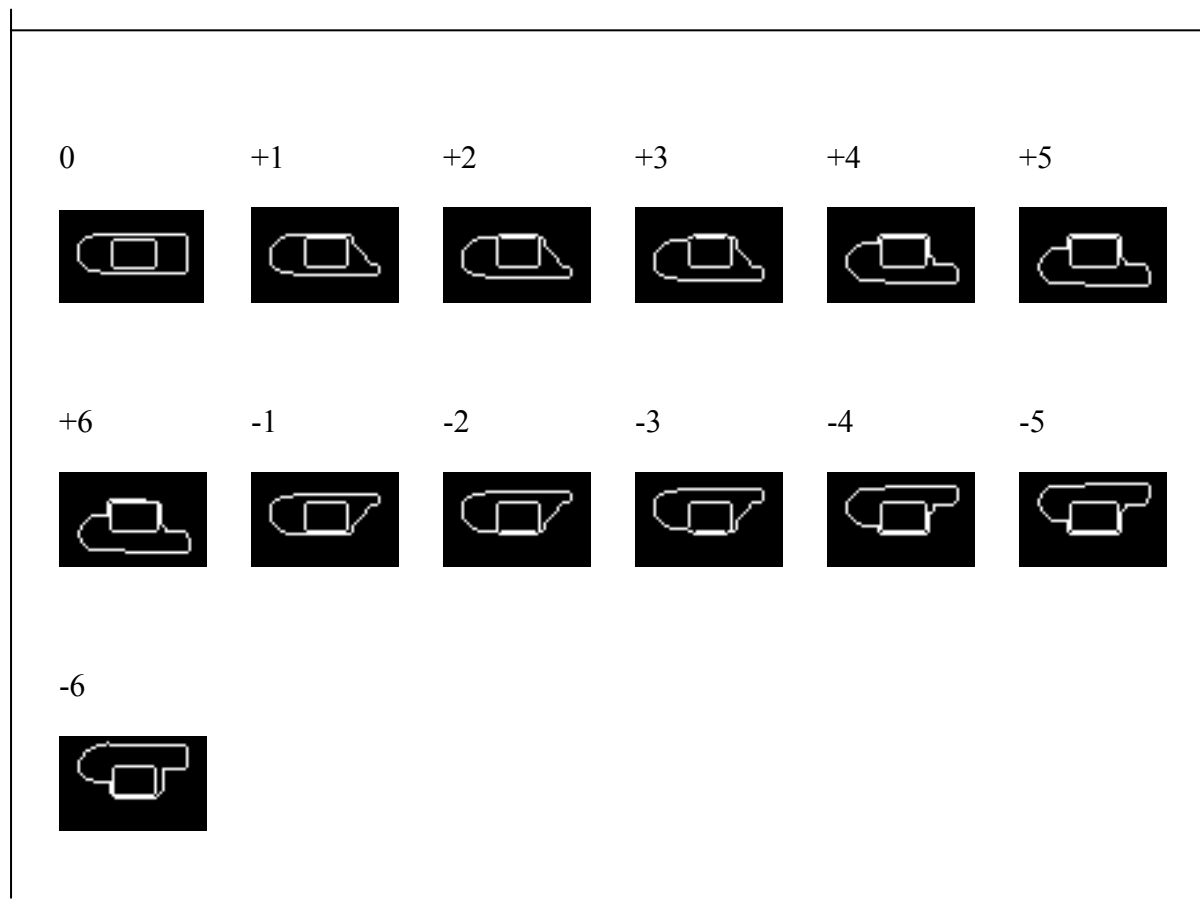


**Σχήμα 3.**Error! Unknown switch argument. **Οι αποστάσεις που λαμβάνουμε υπ' όψη μας κατά τη διαδικασία επιλογής μνημών**

Επιλέγουμε ως μνήμη την εικόνα εκείνη από κάθε κλάση η οποία έχει μικρότερη διασπορά από τη συνολική μέση τιμή των lead, d3, d4 καθώς και τη μικρότερη διασπορά από τη μέση τιμή των στοιχείων d1, d2 της κλάσης στην οποία ανήκει .

Με τον τρόπο αυτό, επιλέγουμε ως μνήμες εικόνες από τις διαφορετικές κλάσεις με τις οποίες πριν εκπαιδεύσουμε το δίκτυο θα πρέπει να τις κάνουμε δυαδικές. Αυτό γίνεται ακολουθώντας τα βήματα της προεπεξεργασίας που περιγράφονται στην προηγούμενη παράγραφο.

Οι μνήμες που παίρνουμε ακολουθώντας τη διαδικασία αυτή φαίνονται στο σχήμα 3.7.



**Σχήμα 3.**Error! Unknown switch argument. **Οι μνήμες που προκύπτουν σύμφωνα με τη διαδικασία 1**

Για καλή λειτουργία του δικτύου θα πρέπει οι μνήμες να έχουν μεταξύ τους μικρό συντελεστή συσχέτισης.

Πιο συγκεκριμένα, ο συντελεστής συσχέτισης μεταξύ 2 pattern X,Y δίνεται από τη σχέση:

$$r_{xy} = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}}$$

Οι πιθανές τιμές του παραπάνω κλάσματος είναι:

$0 < r_{xy} \leq 1$	αν $X$ και $Y$ είναι θετικά συσχετιζόμενα ( $r_{xy} = 1$ αν $X=Y$ )
$r_{xy} = 0$	αν $X$ και $Y$ δεν είναι συσχετιζόμενα ( είναι ορθογώνια )
$-1 \leq r_{xy} < 0$	αν $X$ και $Y$ είναι αρνητικά συσχετιζόμενα ( $r_{xy} = -1$ αν $X=-Y$ )

**Πίνακας 3.**Error! Unknown switch argument. **πιθανές τιμές του συντελεστή συσχέτισης**

Οι συσχετίσεις μεταξύ των μνημών είναι :

	0	1	2	3	4	5	6	-1	-2	-3	-4	-5	-6
0	1	0,79	0,79	0,79	0,80	0,77	0,77	0,81	0,79	0,79	0,80	0,77	0,77
1	0,79	1	0,86	0,85	0,84	0,81	0,79	0,82	0,82	0,81	0,78	0,76	0,80
2	0,79	0,86	1	0,87	0,83	0,80	0,81	0,82	0,81	0,81	0,78	0,76	0,80
3	0,79	0,85	0,87	1	0,84	0,81	0,81	0,81	0,81	0,80	0,78	0,76	0,03
4	0,80	0,84	0,83	0,84	1	0,86	0,83	0,78	0,78	0,78	0,80	0,78	0,77
5	0,77	0,81	0,80	0,81	0,86	1	0,86	0,76	0,76	0,76	0,78	0,79	0,76
6	0,77	0,79	0,81	0,81	0,83	0,86	1	0,76	0,76	0,77	0,79	0,80	0,76
-1	0,81	0,82	0,82	0,81	0,78	0,76	0,76	1	0,86	0,85	0,84	0,81	0,85
-2	0,79	0,82	0,81	0,81	0,78	0,76	0,76	0,86	1	0,87	0,83	0,80	0,85
-3	0,79	0,81	0,81	0,80	0,78	0,76	0,77	0,85	0,87	1	0,84	0,81	0,85
-4	0,80	0,78	0,78	0,78	0,80	0,78	0,79	0,84	0,83	0,84	1	0,86	0,83
-5	0,77	0,76	0,76	0,76	0,78	0,79	0,80	0,81	0,80	0,81	0,86	1	0,83
-6	0,77	0,80	0,80	0,80	0,77	0,76	0,76	0,85	0,85	0,85	0,83	0,83	1

**Πίνακας 3.**Error! Unknown switch argument. **Συσχετίσεις μεταξύ των μνημών που προκύπτουν σύμφωνα με τη διαδικασία 1**

Με αυτή τη διαδικασία επιλογής έχουμε μεν διαφοροποίηση στα στοιχεία d1 και d2 αλλά επίσης και στα στοιχεία lead , d1 και d2. Επίσης, έχουμε σημαντική διαφοροποίηση στην περιοχή εκείνη της εικόνας που βρίσκεται μετά το τέλος του



lead και η οποία δεν δίνει σημαντική πληροφορία για την κλάση στην οποία ανήκει η εικόνα.

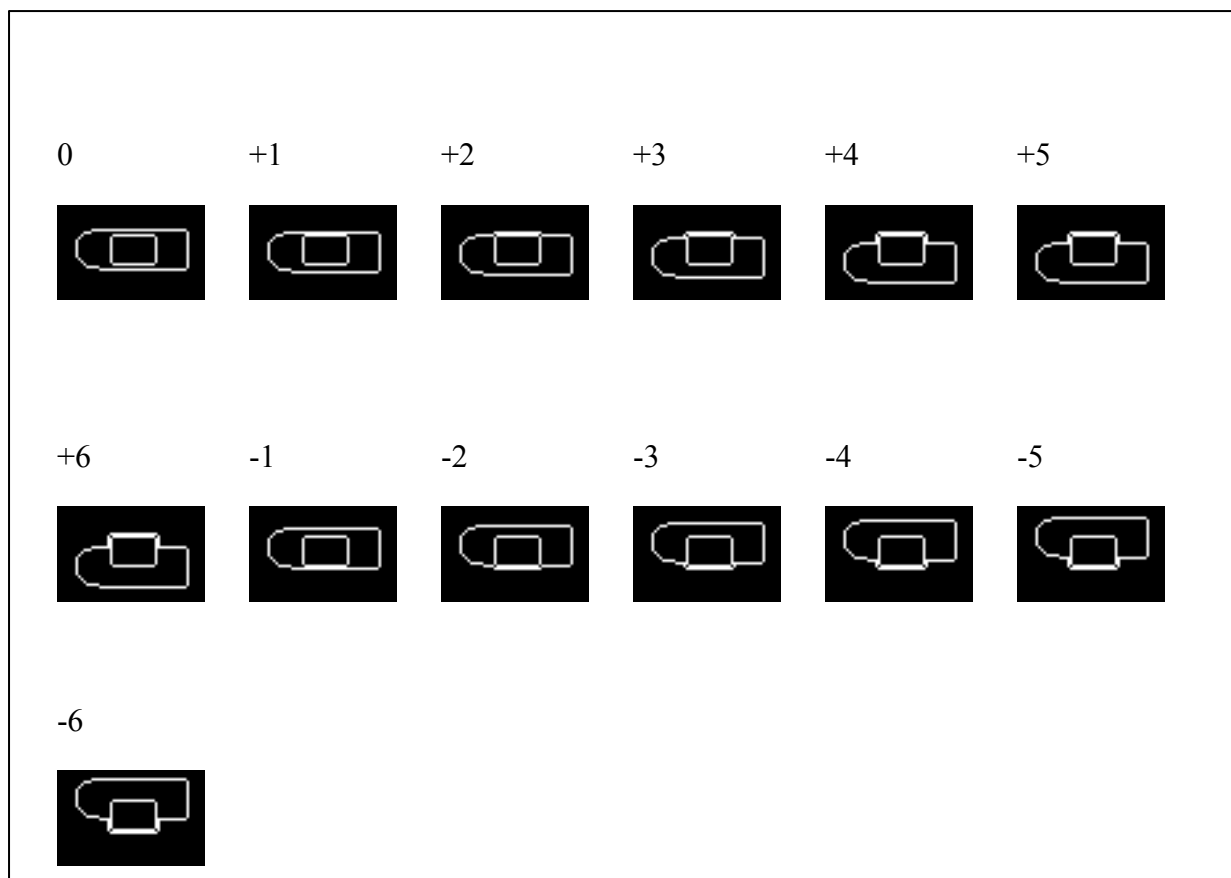
Όλα αυτά έχουν ως αποτέλεσμα την λάθος κατηγοριοποίηση εικόνων οι οποίες είτε έχουν μέγεθος lead , είτε απόσταση d3 είτε ακόμη η περιοχή μετά το τέλος του lead να είναι περισσότερο παρόμοια με αυτή κάποιας λάθος μνήμης.

Για αποφυγή τέτοιων περιπτώσεων καταλήξαμε στο συμπέρασμα πως οι μνήμες μας θα πρέπει να έχουν τα στοιχεία lead, d3, d4 ίδια μεταξύ τους. Ακολουθούμε στην περίπτωση αυτή την εξής διαδικασία:

### 3.2.2 Διαδικασία 2

Επιλέγουμε μια εικόνα από το set των 0 pixel μετατόπιση σύμφωνα με τη διαδικασία 1 και στη συνέχεια μετατοπίσαμε το lead στον κατακόρυφο άξονα, έτσι ώστε να πάρουμε μνήμες για όλες τις μετατοπίσεις.

Με τον τρόπο αυτό οι μνήμες που παίρνουμε φαίνονται στο σχήμα 3.8.



**Σχήμα 3.**Error! Unknown switch argument. **Οι μνήμες που προκύπτουν σύμφωνα με τη διαδικασία 2**

Οι συσχετίσεις μεταξύ των μνημών είναι :

	0	1	2	3	4	5	6	-1	-2	-3	-4	-5	-6
0	1,000	0,833	0,829	0,823	0,817	0,812	0,806	0,833	0,829	0,823	0,817	0,812	0,806
1	0,833	1,000	0,869	0,863	0,857	0,851	0,845	0,829	0,825	0,819	0,813	0,808	0,802
2	0,829	0,869	1,000	0,871	0,865	0,859	0,853	0,825	0,821	0,815	0,810	0,804	0,802
3	0,823	0,863	0,871	1,000	0,871	0,865	0,859	0,819	0,815	0,810	0,804	0,802	0,796
4	0,817	0,857	0,865	0,871	1,000	0,871	0,865	0,813	0,810	0,804	0,802	0,796	0,794
5	0,812	0,851	0,859	0,865	0,871	1,000	0,871	0,808	0,804	0,802	0,796	0,794	0,788
6	0,806	0,845	0,853	0,859	0,865	0,871	1,000	0,802	0,802	0,796	0,794	0,788	0,786
-1	0,833	0,829	0,825	0,819	0,813	0,808	0,802	1,000	0,869	0,863	0,857	0,851	0,845
-2	0,829	0,825	0,821	0,815	0,810	0,804	0,802	0,869	1,000	0,871	0,865	0,859	0,853
-3	0,823	0,819	0,815	0,810	0,804	0,802	0,796	0,863	0,871	1,000	0,871	0,865	0,859
-4	0,817	0,813	0,810	0,804	0,802	0,796	0,794	0,857	0,865	0,871	1,000	0,871	0,865
-5	0,812	0,808	0,804	0,802	0,796	0,794	0,788	0,851	0,859	0,865	0,871	1,000	0,871
-6	0,806	0,802	0,802	0,796	0,794	0,788	0,786	0,845	0,853	0,859	0,865	0,871	1,000

**Πίνακας 3.**Error! Unknown switch argument. **Συσχετίσεις μεταξύ των μνημών που προκύπτουν σύμφωνα με τη διαδικασία 2**

Παρατηρούμε ότι η συσχέτιση μεταξύ των μνημών αυξήθηκε στην περίπτωση αυτή (αλλά όπως παρατηρήσαμε τα ποσοστά επιτυχίας έχουν αυξηθεί).

Τα μεγάλα ποσοστά συσχέτισης οφείλονται στο ότι όλα τα pixels των εικόνων μας είναι ίδια, εκτός από αυτά που συνθέτουν το περίγραμμα του solder paste. Έτσι, αυτό που στην ουσία γίνεται είναι σύγκριση μεταξύ σχημάτων τοποθετημένων σε διαφορετικές περιοχές της εικόνας .

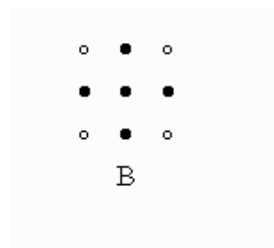
Για να αυξήσουμε τη διαφοροποίηση μεταξύ των μνημών και συνεπώς για μειώσουμε τη συσχέτιση μεταξύ τους ακολουθούμε την εξής διαδικασία:

### 3.2.3 Διαδικασία 3

Το ζητούμενο στην περίπτωση αυτή είναι να μειώσουμε τη συσχέτιση μεταξύ των μνημών που αποθηκεύονται στο δίκτυο. Ένας από τους πιο αποτελεσματικούς τρόπους για να το πετύχουμε αυτό είναι να αλλάξουμε τη μορφολογία του αντικειμένου που απεικονίζεται στις μνήμες, επεκτείνοντας το περίγραμμα του και διατηρώντας παράλληλα τα στοιχεία που τις διαφοροποιούν μεταξύ τους. Μια από τις διαδεδομένες διαδικασίες για επέκταση αντικειμένων σε δυαδικές εικόνες είναι η εφαρμογή του μορφολογικού τελεστή dilation. Το σχήμα και η ακτίνα του δομικού στοιχείου που θα χρησιμοποιούμε για το dilation θα πρέπει να είναι τέτοια έτσι ώστε να διατηρούνται τα στοιχεία εκείνα που διαφοροποιούν τις μνήμες μεταξύ τους

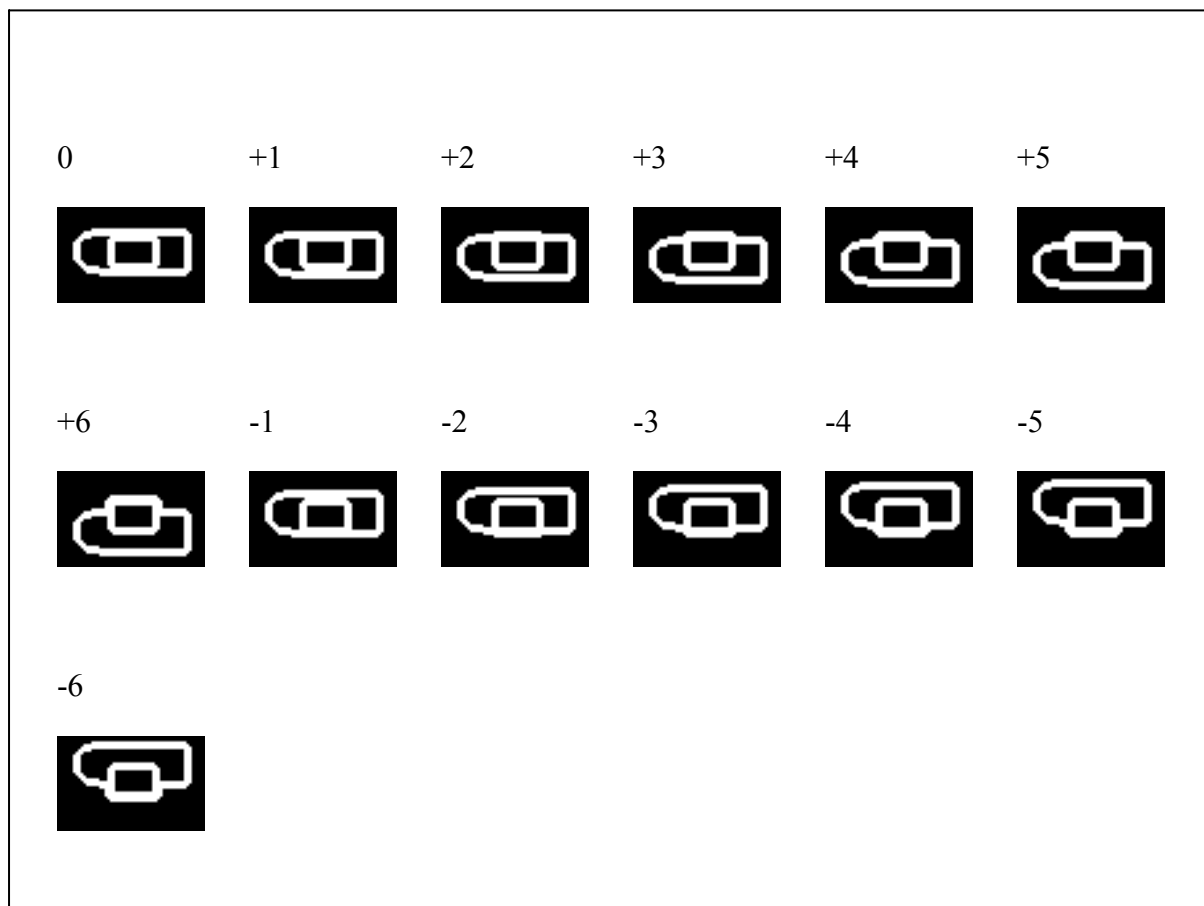
(δηλαδή οι αποστάσεις του περιγράμματος του lead από το περίγραμμα της περιοχής του pad και solder paste). Μετά από πειράματα καταλήξαμε πως το καταλληλότερο δομικό στοιχείο είναι σχήματος ρόμβου και ακτίνας 1 pixel. Η διαδικασία που ακολουθούμε λοιπόν στην περίπτωση αυτή είναι:

Στις μνήμες που προέκυψαν από τη διαδικασία 2 εφαρμόζουμε το μορφολογικό τελεστή dilation με structure element σχήματος ρόμβου και ακτίνας 1 pixel (σχήμα 3.9).



**Σχήμα 3.** το δομικό στοιχείο που χρησιμοποιείται για dilation

Με τον τρόπο αυτό οι μνήμες που παίρνουμε φαίνονται στο σχήμα 3.10.



**Σχήμα 3.**Error! Unknown switch argument. **Οι μνήμες που προκύπτουν σύμφωνα με τη διαδικασία 3**

Οι συσχετίσεις μεταξύ των μνημών είναι :

	0	1	2	3	4	5	6	-1	-2	-3	-4	-5	-6
0	1,000	0,867	0,738	0,611	0,597	0,583	0,569	0,867	0,738	0,611	0,597	0,583	0,569
1	0,867	1,000	0,871	0,744	0,619	0,605	0,591	0,734	0,605	0,593	0,579	0,565	0,552
2	0,738	0,871	1,000	0,873	0,748	0,623	0,609	0,605	0,560	0,548	0,534	0,520	0,510
3	0,611	0,744	0,873	1,000	0,875	0,750	0,625	0,593	0,548	0,536	0,522	0,512	0,502
4	0,597	0,619	0,748	0,875	1,000	0,875	0,750	0,579	0,534	0,522	0,512	0,502	0,496
5	0,583	0,605	0,623	0,750	0,875	1,000	0,875	0,565	0,520	0,512	0,502	0,496	0,490
6	0,569	0,591	0,609	0,625	0,750	0,875	1,000	0,552	0,510	0,502	0,496	0,490	0,492
-1	0,867	0,734	0,605	0,593	0,579	0,565	0,552	1,000	0,871	0,744	0,619	0,605	0,591
-2	0,738	0,605	0,560	0,548	0,534	0,520	0,510	0,871	1,000	0,873	0,748	0,623	0,609
-3	0,611	0,593	0,548	0,536	0,522	0,512	0,502	0,744	0,873	1,000	0,875	0,750	0,625
-4	0,597	0,579	0,534	0,522	0,512	0,502	0,496	0,619	0,748	0,875	1,000	0,875	0,750
-5	0,583	0,565	0,520	0,512	0,502	0,496	0,490	0,605	0,623	0,750	0,875	1,000	0,875
-6	0,569	0,552	0,510	0,502	0,496	0,490	0,492	0,591	0,609	0,625	0,750	0,875	1,000

**Πίνακας 3.**Error! Unknown switch argument. **Συσχετίσεις μεταξύ των μνημών που προκύπτουν σύμφωνα με τη διαδικασία 3**

Παρατηρούμε λοιπόν ότι πράγματι οι συσχετίσεις έχουν μειωθεί σημαντικά, κάτι που θα βοηθήσει στη λειτουργία του δικτύου.

Σημειώνεται ότι στην περίπτωση αυτή, κατά τη διάρκεια της προεπεξεργασίας στις εικόνες εφαρμόζεται dilation με το ίδιο structure element.

### 3.3 Λειτουργία του δικτύου Hopfield

Όπως έχει αναφερθεί, οι εικόνες οι οποίες θα εισάγονται στο δίκτυο είναι μεγέθους  $36 \times 56 = 2016$  pixels. Όποτε και το δίκτυο μας θα αποτελείται από  $N=2016$  νευρώνες καθένας από τους οποίους θα αντιπροσωπεύει ένα pixel της εικόνας.

Πρόβλημα χωρητικότητας δεν έχουμε, αφού ο αριθμός των μνημών που αποθηκεύονται στο δίκτυο είναι  $M = 13$ , και συνεπώς τηρείται ο περιορισμός

$$M < 14\%N \text{ καθώς και ο } M < \frac{N}{2 \ln N} .$$

Για την αναπαράσταση των εικόνων πριν την εισαγωγή τους στο δίκτυο θεωρούμε πως τα άσπρα pixels των εικόνων έχουν τιμή +1 , ενώ τα μαύρα pixels τιμή -1.

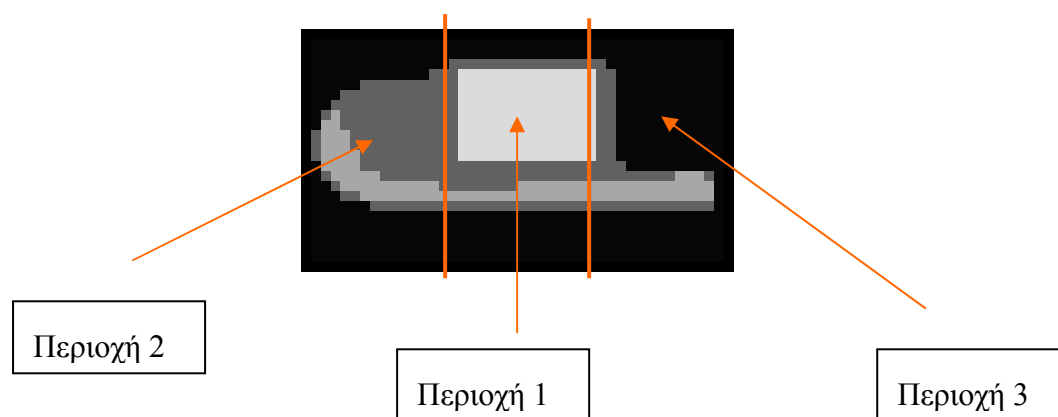
Για την εκπαίδευση του δικτύου χρησιμοποιήσαμε και τους 3 κανόνες που αναφέρονται στο κεφάλαιο 2 (Hebbian learning rule, learning in bounds, projection learning rule).

Για τη μέθοδο learning in bounds, χρησιμοποιήσαμε για την εκπαίδευση (εκτός φυσικά από τις μνήμες) 30 εικόνες από κάθε κλάση.

Επειδή οι μνήμες μας και στις 3 περιπτώσεις που αναφέρθηκαν στην προηγούμενη παράγραφο παρουσιάζουν υψηλό ποσοστό συσχέτισης, θα περιμένουμε πως ο projection learning rule θα έχει την καλύτερη απόδοση, κάτι που όπως θα δούμε και στο κεφάλαιο 4 πράγματι συμβαίνει.

Αναγκαία είναι η εισαγωγή thresholds σε κάθε νευρώνα του δικτύου, γιατί παρατηρήθηκε πως χωρίς αυτά ένας σημαντικός αριθμός εικόνων κάθε κλάσης έτεινε στο ολικό ελάχιστο της συνάρτησης ενέργειας του δικτύου. συνεπώς με την εισαγωγή thresholds, αυξάνεται η απόδοση του δικτύου. Στην εφαρμογή μας χρησιμοποιήσαμε το μοντέλο των adjustable thresholds, που αναφέρεται αναλυτικότερα στο κεφάλαιο 2.

Μια διαφοροποίηση έγινε στον τρόπο με τον οποίο το δίκτυο ανανεώνει τους νευρώνες του. Όπως έχει αναφερθεί, οι νευρώνες ανανεώνονται με τυχαία σειρά (εκτός από την περίπτωση του learning in bounds που έχουμε παράλληλη ανανέωση). Μετά από πειράματα καταλήξαμε στο συμπέρασμα πως η απόδοση του δικτύου αυξάνεται όταν αρχίζει η ανανέωση πρώτα από την περιοχή που βρίσκεται ανάμεσα στις 2 κατακόρυφες ευθείες που ορίζουν τα όρια του lead (σχήμα 3.11), μια και στην περιοχή αυτή βρίσκονται οι κατακόρυφες μετατοπίσεις του lead, δηλαδή τα στοιχεία εκείνα που διαφοροποιούν τις κλάσεις.



Σχήμα 3. Error! Unknown switch argument. Περιοχές ανανέωσης

Αναλυτικότερα η διαδικασία ανανέωσης θα είναι η εξής :

Πρώτα, θα υπολογίζεται η περιοχή ανάμεσα στα όρια του lead. Στη συνέχεια θα ανανεώνονται πρώτα όλοι οι νευρώνες μέσα στην περιοχή αυτή (περιοχή 1 στο σχήμα 3.11), μετά θα ανανεώνονται όλοι οι νευρώνες της περιοχής 2 και τέλος οι νευρώνες της περιοχής 3, μια και η περιοχή αυτή περιέχει τα λιγότερο σημαντικά στοιχεία της εικόνας.

Με τη μέθοδο αυτή δεν αλλάζουν τα basins of attractors, απλά τα trajectories των test patterns και φυσικά, τα σημεία όπου αυτά σταθεροποιούνται.

### 3.4 Classification των εικόνων μετά την έξοδο τους από το δίκτυο

#### Hopfield

Η έξοδος του δικτύου Hopfield θα είναι μια εικόνα η οποία θα προσεγγίζει κάποια από τις associative memories. Όμως, όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο, ένα από τα προβλήματα των δικτύων Hopfield είναι η ύπαρξη των spurious states, δηλαδή των καταστάσεων όπου η έξοδος του δικτύου δεν συμπίπτει με καμία από τις μνήμες που το δίκτυο έχει αποθηκεύσει. Στην εφαρμογή μας, λόγω των υψηλών ποσοστών συσχετίσεων μεταξύ των μνημών, το ποσοστό των εικόνων που φτάνουν σε τέτοιες καταστάσεις είναι υψηλό.

Αναλυτικότερα, τα ποσοστά των εικόνων που σταθεροποιούνται σε κάποιο spurious state για κάθε μια από τις διαδικασίες επιλογής μνημών που περιγράφηκε στην παράγραφο 3.2, καθώς και για κάθε κανόνα εκμάθησης που αναφέρεται στην παράγραφο 3.3 φαίνονται στον πίνακα 3.5.

	Hebbian Learning	Learning in bounds	Projection learning
διαδικασία1	74%	66%	62%
διαδικασία2	76%	69%	66%
διαδικασία3	86%	80%	78%

**Πίνακας 3.5.** Ποσοστά εικόνων που φτάνουν σε spurious states

Θα πρέπει λοιπόν να επιλέξουμε ένα κριτήριο με βάση το οποίο οι εικόνες που προκύπτουν από την έξοδο του δικτύου μας να κατηγοριοποιούνται σε μια από τις 13 διαφορετικές κλάσεις. Το κριτήριο αυτό είναι η *ευκλείδεια απόσταση* (ή η απόσταση Hamming) της εξόδου του δικτύου Hopfield από τις συσχετιστικές μνήμες. Δηλαδή, το classification γίνεται υπολογίζοντας τις ευκλείδειες αποστάσεις της εξόδου του δικτύου από όλες της μνήμες και επιλέγοντας αυτή από την οποία απέχει τη μικρότερη απόσταση.

Επειδή όπως έχει αναφερθεί και στο κεφάλαιο 1 η εφαρμογή πραγματοποιείται σε πραγματικό χρόνο, μας ενδιαφέρει ιδιαίτερος ο χρόνος που απαιτεί η εφαρμογή μας. Μετά από πειράματα καταλήξαμε στο συμπέρασμα πως ένα σύστημα άμεσου

υπολογισμού της ευκλείδεια απόστασης της εξόδου του δικτύου Hopfield από όλες τις μνήμες απαιτεί περισσότερο χρόνο από ότι ένα TND το οποίο θα εκπαιδευτεί αρχικά και στη συνέχεια θα είναι σε θέση να υπολογίσει τη μικρότερη απόσταση από τις μνήμες. Αναλυτικότερα, είδαμε ότι ένα σύστημα άμεσου υπολογισμού απαιτεί περίπου το 150% του χρόνου που απαιτεί η διάδοση στο δίκτυο που περιγράφεται παρακάτω (ο άμεσος υπολογισμός πραγματοποιήθηκε σε 0,6 secs ενώ η διάδοση στο παρακάτω δίκτυο πραγματοποιήθηκε σε 0,42 secs).

Για τον υπολογισμό της απόστασης αυτής επιλέξαμε το *TND Hamming*, το οποίο υπολογίζει τις παραπάνω αποστάσεις αυτή και δίνει σαν αποτέλεσμα την κλάση στην οποία η εικόνα μας ανήκει. Η λειτουργία του περιγράφεται παρακάτω :

### 3.4.1 Δίκτυο Hamming

Το μοντέλο αυτό, που διατυπώθηκε από τον Lippman το 1984, υλοποιεί έναν κατηγοριοποιητή (classifier) που βασίζεται στην ελάχιστη απόσταση Hamming. Όπως ακριβώς και το δίκτυο Hopfield αποθηκεύει ένα set από M patterns (τα οποία επίσης ονομάζονται associative memories). Όταν ένα νέο pattern  $X$  διαδίδεται στο δίκτυο, τότε υπολογίζεται την απόσταση Hamming του pattern αυτού από όλες τις συσχετιστικές μνήμες και Στη συνέχεια χρησιμοποιώντας ένα ανταγωνιστικό TND (συνήθως το maxnet) θα επιλέγεται η μνήμη αυτή η οποία απέχει λιγότερο από το pattern αυτό.

Το δίκτυο βασίζεται στην εξής ιδέα :

Η απόσταση Hamming μεταξύ 2 διανυσμάτων  $x$  και  $y$  μεγέθους  $N$  είναι ο αριθμός των στοιχείων που διαφέρουν μεταξύ τους. Δηλαδή:

$$HD = (\text{αριθμός στοιχείων των } x \text{ και } y \text{ που διαφέρουν}).$$

Όμως, επειδή έχουμε bipolar διανύσματα θα ισχύει ότι:

$$\sum_{i=1}^N \left( x_i \cdot y_i \right) = (\text{αριθμός στοιχείων που τα } x \text{ και } y \text{ ταιριάζουν}) - HD.$$

Συνδυάζοντας τις δυο παραπάνω σχέσεις παίρνουμε ότι:

$$HD = \frac{1}{2} \left[ N - \sum_{i=1}^N \left( x_i \cdot y_i \right) \right].$$

Όποτε , για να ελαχιστοποιηθεί η απόσταση, θα πρέπει να μεγιστοποιηθεί ο όρος

$$\sum_{i=1}^N \left( x_i \cdot y_i \right).$$

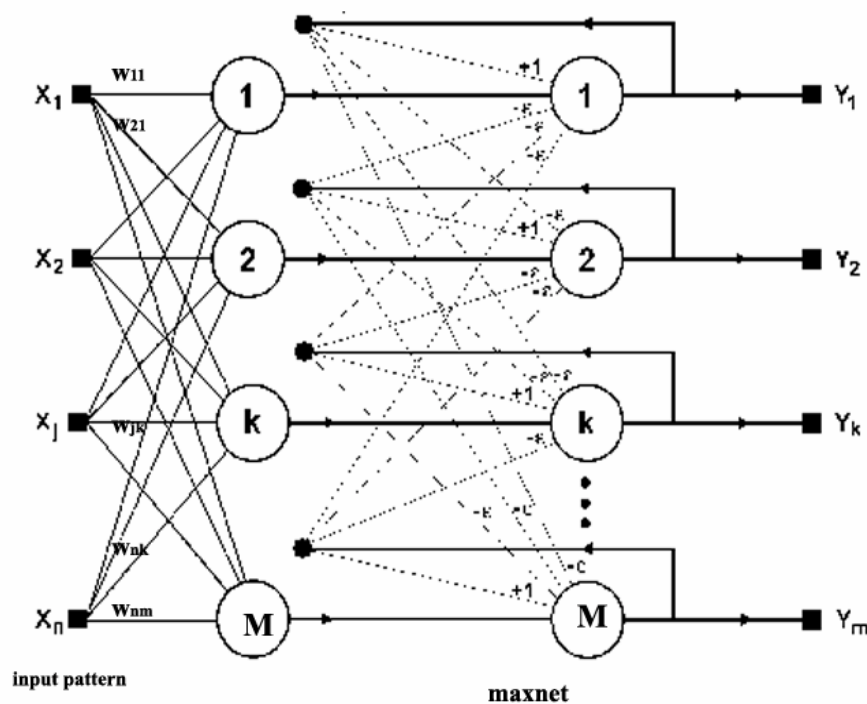
Έτσι, για να βρούμε την ελάχιστη απόσταση Hamming μεταξύ του διανύσματος που διαδίδεται στο δίκτυο και των μνημών μας, χρησιμοποιούμε πρώτα το δίκτυο Hamming που υπολογίζει τον παραπάνω παράγοντα για κάθε μνήμη και στη συνέχεια το ανταγωνιστικό δίκτυο maxnet που μας δίνει τον μεγαλύτερο όρο της παραπάνω συνάρτησης.

### 3.4.1.1 Τοπολογία του δικτύου

Τα δίκτυα τύπου Hamming είναι feed-forward και αποτελούνται από ένα δυο επίπεδα (layer) νευρώνων (σχήμα 3.12). Οι μονάδες του πρώτου επιπέδου συνδέονται με κάθε μονάδα του δεύτερου επιπέδου. Ο αριθμός των μονάδων του πρώτου επιπέδου ισούται με το μέγεθος  $N$  των αποθηκευμένων μνημών, ενώ ο αριθμός των μονάδων του δεύτερου επιπέδου ισούται με τον αριθμό  $M$  των αποθηκευμένων μνημών. Όποτε, το δίκτυο θα έχει  $MN$  συνδέσεις. Στη συνέχεια χρησιμοποιείται το ανταγωνιστικό δίκτυο maxnet, το οποίο είναι ένα ανατροφοδοτούμενο δίκτυο, που διαλέγει τη μονάδα με τη μεγαλύτερη έξοδο, η οποία θα είναι και ο νικητής του δικτύου. Τα βάρη των συνδέσεων του δικτύου αυτού είναι 1 για την αυτό-ανατροφοδότηση και  $-ε$  (σταθερά), για τις υπόλοιπες συνδέσεις. Το maxnet έχει  $M^2$  συνδέσεις. Όποτε, το συνολικό σύστημα θα έχει  $MN + M^2$  συνδέσεις.

Στο σχήμα μας το  $X_i$ , όπου  $i = 1..N$ , αντιπροσωπεύει τα στοιχεία του pattern που

διαδίδεται στο δίκτυο (δηλαδή την είσοδο του δικτύου), ενώ το  $Y_j$  αντιπροσωπεύει τα στοιχεία της εξόδου του δικτύου, όπου  $j = 1..M$ .



Σχήμα 3. Error! Unknown switch argument. Τοπολογία του δικτύου Hamming



### 3.4.1.2 Λειτουργία

- **Φάση αποθήκευσης**

Θα συμβολίζουμε τις associative memories με  $p_i^k$ , όπου  $i = 1..N$  και  $k = 1..M$ , όπου  $N$  το μέγεθος των μνημών και  $M$  ο αριθμός τους.

Κατ' αρχήν εκπαιδεύουμε το δίκτυο υπολογίζοντας τα βάρη των συνδέσεων σύμφωνα με τον τύπο:

$$w_{ij} = \frac{p_i^j}{2}$$

καθώς και τα thresholds για κάθε μονάδα από τον τύπο:

$$\theta_j = \frac{N}{2}$$

- **Διάδοση**

Έστω ότι στο δίκτυο εμφανίζεται ένα pattern  $X$  με στοιχεία  $\left(x_1, \dots, x_N\right)$ .

Αρχικά, οι μονάδες του πρώτου επιπέδου θέτονται ίσες με τις τιμές του  $X$  και κατόπιν υπολογίζουμε τις τιμές για κάθε μονάδα του δεύτερου επιπέδου:

$$y_i = \sum_{j=1}^N w_{ij} x_j(t) + \theta_i$$

Στη συνέχεια το δίκτυο maxnet θα επιλέξει το νικητή από τις μονάδες του δεύτερου επιπέδου, ακολουθώντας την εξής διαδικασία :

Θα ανανεώσει τους νευρώνες αυτούς σύμφωνα με τον τύπο:

$$y_i(t) = f(y_i(t-1) - \varepsilon \sum_{k \neq i} y_k(t-1))$$

Με συνάρτηση ενεργοποίησης την: 
$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

Και όπου  $\varepsilon$  παράμετρος που εμείς ορίζουμε και η οποία πρέπει να είναι  $0 < \varepsilon < \frac{1}{M}$ . Σημειώνεται εδώ ότι για την εφαρμογή μας επιλέξαμε  $\varepsilon = \frac{1}{2} \left( \frac{1}{M} \right)$ , δηλαδή  $\varepsilon = 0,0385$ .

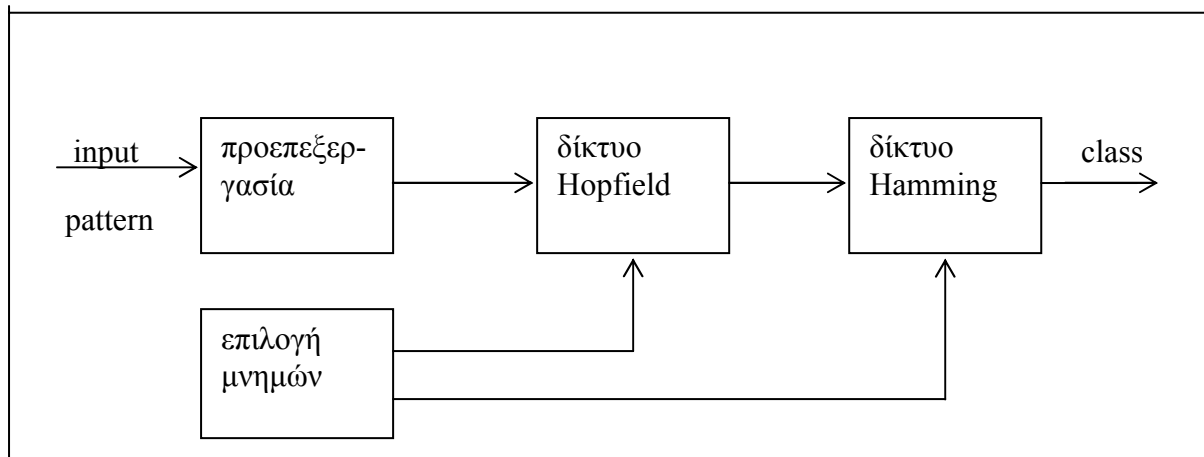
Το maxnet σταματά την ανανέωση όταν μόνο ένας νευρώνας έχει μη-μηδενική έξοδο. Αυτός ο νευρώνας είναι και ο νικητής. Ο αριθμός του νευρώνα μας δίνει και την κλάση στην οποία ανήκει το διάνυσμα εισόδου μας.

### 3.5 Συνολικό σύστημα

Το σύστημα το οποίο προτείνουμε είναι ένας classifier, που στηρίζεται σε ΤΝΔ συσχετιζόμενων μνημών (associative memories).

Η είσοδος είναι μια εικόνα μεγέθους 24x44 pixels με 4 επίπεδα φωτεινότητας, η οποία αφού υφίσταται την προεπεξεργασία που περιγράφεται στην παραγραφο3.1, διοχετεύεται στο δίκτυο Hopfield, η έξοδος του οποίου είναι η είσοδος του δικτύου Hamming, το οποίο δίνει ως έξοδο την κλάση στην οποία η εικόνα ανήκει.

Η διαδικασία που ακολουθούμε συνοψίζεται στο σχήμα 3.13.



**Σχήμα 3.**Error! Unknown switch argument. **Η διαδικασία που ακολουθεί το συνολικό σύστημα**

## 4 Αποτελέσματα και συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα του συστήματος σε ένα set εικόνων με όλες τις μετατοπίσεις. Πιο συγκεκριμένα, παρουσιάζεται η απόδοση του δικτύου καθώς και παραδείγματα διάδοσης εικόνων στο σύστημα και φυσικά, τα συμπεράσματα που προκύπτουν από τα αποτελέσματα αυτά.

### 4.1 Απόδοση του δικτύου

Οι παρακάτω πίνακες παρουσιάζουν τα αποτελέσματα του συστήματος κατηγοριοποίησης μας (σε ποσοστά %). Οι οριζόντιες γραμμές δείχνουν τις κλάσεις στις οποίες οι εικόνες μας ανήκουν και οι κατακόρυφες τις κλάσεις στις οποίες το σύστημα τις κατατάσσει.

#### 4.1.1 Αποτελέσματα για σύστημα με την πρώτη διαδικασία επιλογής μνημών

- *Hebbian learning rule*

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	60,8%	31,7%	2,5%	0,0%	0,8%	0,8%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%	1,7%
-5	16,7%	38,3%	25,8%	10,8%	2,5%	3,3%	0,8%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%
-4	0,0%	2,5%	65,8%	25,0%	4,2%	0,0%	1,7%	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%
-3	0,0%	1,7%	2,5%	52,5%	37,5%	3,3%	0,8%	0,0%	0,0%	0,0%	0,0%	1,7%	0,0%
-2	0,0%	0,8%	0,0%	7,5%	65,8%	23,3%	0,8%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%
-1	0,0%	1,7%	0,0%	0,0%	6,7%	73,3%	15,0%	0,0%	0,0%	0,8%	0,0%	0,0%	0,0%
0	0,0%	0,0%	0,0%	0,0%	0,0%	4,2%	61,7%	31,7%	1,7%	0,0%	0,0%	0,8%	0,0%
1	0,0%	1,7%	0,0%	0,8%	0,0%	0,0%	3,3%	69,2%	20,8%	2,5%	0,0%	1,7%	0,0%
2	0,0%	1,7%	0,0%	0,0%	0,0%	0,8%	0,0%	3,3%	70,0%	20,8%	2,5%	0,8%	0,0%
3	0,0%	4,2%	0,0%	0,8%	0,8%	0,0%	0,8%	2,5%	8,3%	67,5%	12,5%	1,7%	0,8%
4	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	1,7%	0,0%	0,8%	5,0%	79,2%	11,7%	1,7%
5	0,0%	2,5%	0,0%	0,0%	0,0%	2,5%	0,0%	5,0%	0,0%	5,0%	16,7%	46,7%	21,7%
6	0,0%	0,8%	3,3%	0,0%	2,5%	1,7%	0,0%	5,0%	4,2%	0,0%	5,8%	12,5%	64,2%

Πίνακας 4. Error! Unknown switch argument. Απόδοση δικτύου με κανόνα εκμάθησης τον hebbian learning rule και με την πρώτη διαδικασία επιλογής μνημών

- *Learning in bounds*

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	60,8%	31,7%	2,5%	0,0%	0,8%	0,8%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%	1,7%
-5	15,8%	41,7%	25,8%	10,8%	2,5%	0,8%	0,8%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%
-4	0,0%	2,5%	65,8%	25,0%	5,0%	0,0%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-3	0,0%	1,7%	2,5%	52,5%	37,5%	3,3%	0,8%	0,0%	0,0%	0,0%	0,0%	1,7%	0,0%
-2	0,0%	0,8%	0,0%	7,5%	66,7%	22,5%	0,8%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%
-1	0,0%	1,7%	0,0%	0,0%	6,7%	73,3%	15,0%	0,0%	0,0%	0,8%	0,0%	0,0%	0,0%
0	0,0%	0,8%	0,0%	0,0%	0,0%	4,2%	64,2%	29,2%	0,8%	0,0%	0,0%	0,8%	0,0%
1	0,0%	2,5%	0,0%	0,8%	0,0%	0,0%	4,2%	68,3%	20,8%	2,5%	0,0%	0,0%	0,8%
2	0,0%	2,5%	0,0%	0,0%	0,0%	0,8%	0,0%	3,3%	69,2%	20,8%	2,5%	0,8%	0,0%
3	0,0%	5,8%	0,0%	0,8%	0,8%	0,0%	0,8%	2,5%	8,3%	65,8%	12,5%	1,7%	0,8%
4	0,0%	0,0%	0,0%	1,7%	0,8%	0,0%	1,7%	0,0%	0,0%	5,0%	78,3%	10,8%	1,7%
5	0,0%	2,5%	0,0%	0,0%	2,5%	0,0%	0,0%	3,3%	0,0%	3,3%	15,8%	50,0%	22,5%
6	0,0%	5,8%	3,3%	0,0%	2,5%	1,7%	0,0%	2,5%	1,7%	0,0%	5,8%	10,8%	65,8%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου με κανόνα εκμάθησης το *learning in bounds* και με την πρώτη διαδικασία επιλογής μνημών

- *Projection learning rule*

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	63,3%	32,5%	2,5%	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,8%
-5	16,7%	43,3%	25,8%	8,3%	1,7%	2,5%	0,0%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%
-4	0,0%	3,3%	65,8%	25,0%	4,2%	0,0%	0,8%	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%
-3	0,0%	0,8%	2,5%	59,2%	32,5%	3,3%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
-2	0,0%	0,0%	0,0%	8,3%	65,8%	23,3%	1,7%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%
-1	0,0%	0,8%	0,0%	0,0%	7,5%	74,2%	16,7%	0,0%	0,0%	0,0%	0,0%	0,0%	0,8%
0	0,0%	0,0%	0,0%	0,0%	0,0%	5,8%	59,2%	33,3%	1,7%	0,0%	0,0%	0,0%	0,0%
1	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	4,2%	68,3%	21,7%	4,2%	0,8%	0,8%	0,0%
2	0,0%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%	3,3%	69,2%	20,8%	4,2%	0,8%	0,0%
3	0,0%	1,7%	0,0%	0,0%	0,8%	0,0%	0,0%	2,5%	8,3%	69,2%	15,8%	1,7%	0,0%
4	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%	0,8%	5,0%	79,2%	12,5%	1,7%
5	0,0%	0,8%	0,0%	0,0%	0,8%	0,0%	0,0%	3,3%	0,0%	5,0%	14,2%	51,7%	24,2%
6	0,0%	0,0%	0,8%	0,0%	1,7%	0,8%	0,0%	2,5%	1,7%	0,0%	7,5%	15,8%	69,2%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου με κανόνα εκμάθησης τον *projection learning rule* και με την πρώτη διαδικασία επιλογής μνημών

#### 4.1.2 Αποτελέσματα για σύστημα με τη δεύτερη διαδικασία επιλογής μνημών

- *Hebbian learning rule*

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	65,0%	25,0%	6,7%	0,0%	0,8%	0,0%	2,5%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-5	14,2%	43,3%	24,2%	12,5%	1,7%	0,8%	1,7%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%
-4	0,0%	0,0%	62,5%	29,2%	3,3%	0,8%	4,2%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-3	0,0%	0,0%	2,5%	54,2%	38,3%	3,3%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
-2	0,0%	0,0%	0,0%	7,5%	65,8%	20,8%	4,2%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%
-1	0,0%	0,0%	0,0%	0,0%	8,3%	69,2%	21,7%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
0	0,0%	0,0%	0,0%	0,0%	0,0%	2,5%	71,7%	24,2%	0,8%	0,0%	0,0%	0,8%	0,0%
1	0,0%	0,8%	0,0%	0,8%	0,8%	0,0%	5,8%	67,5%	21,7%	2,5%	0,0%	0,0%	0,0%
2	0,0%	0,8%	0,0%	0,8%	0,8%	0,0%	0,0%	4,2%	70,0%	20,8%	2,5%	0,0%	0,0%
3	0,0%	0,8%	0,0%	0,0%	0,0%	0,8%	0,8%	2,5%	11,7%	70,0%	13,3%	0,0%	0,0%
4	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	3,3%	0,0%	0,8%	4,2%	80,0%	11,7%	0,0%
5	0,0%	0,0%	0,0%	1,7%	2,5%	2,5%	0,0%	0,8%	0,0%	4,2%	12,5%	49,2%	26,7%
6	0,0%	0,0%	0,0%	0,8%	5,8%	5,0%	1,7%	3,3%	1,7%	2,5%	5,8%	14,2%	59,2%

Πίνακας 4. Error! Unknown switch argument. Απόδοση δικτύου με κανόνα εκμάθησης τον hebbian learning rule και με τη δεύτερη διαδικασία επιλογής μνημών

- *Learning in bounds*

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	65,0%	25,8%	6,7%	0,0%	0,8%	0,0%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-5	14,2%	43,3%	24,2%	12,5%	1,7%	0,8%	1,7%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%
-4	0,0%	0,0%	62,5%	29,2%	3,3%	0,8%	3,3%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%
-3	0,0%	0,0%	2,5%	54,2%	38,3%	3,3%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
-2	0,0%	0,0%	0,0%	7,5%	65,8%	21,7%	3,3%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%
-1	0,0%	0,0%	0,0%	0,0%	8,3%	69,2%	20,8%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%
0	0,0%	0,0%	0,0%	0,0%	0,0%	1,7%	69,2%	26,7%	1,7%	0,0%	0,0%	0,8%	0,0%
1	0,8%	0,8%	0,0%	0,8%	0,0%	0,8%	4,2%	67,5%	22,5%	2,5%	0,0%	0,0%	0,0%
2	0,0%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%	4,2%	70,8%	20,8%	2,5%	0,0%	0,0%
3	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	2,5%	11,7%	70,0%	14,2%	0,0%	0,0%
4	0,0%	0,0%	0,0%	0,0%	0,0%	0,8%	1,7%	0,0%	0,8%	4,2%	80,0%	12,5%	0,0%
5	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%	4,2%	11,7%	50,8%	27,5%
6	0,0%	0,0%	0,0%	0,0%	6,7%	5,0%	1,7%	2,5%	0,8%	2,5%	3,3%	12,5%	64,2%

Πίνακας 4. Error! Unknown switch argument. Απόδοση δικτύου με κανόνα εκμάθησης το learning in bounds και με τη δεύτερη διαδικασία επιλογής μνημών

- *Projection learning rule*

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	64,2%	31,7%	2,5%	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,8%
-5	15,0%	45,8%	25,8%	7,5%	1,7%	2,5%	0,0%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%
-4	0,0%	3,3%	65,8%	25,0%	4,2%	0,0%	0,8%	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%
-3	0,0%	0,8%	2,5%	59,2%	32,5%	3,3%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
-2	0,0%	0,0%	0,0%	7,5%	67,5%	23,3%	0,8%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%
-1	0,0%	0,8%	0,0%	0,0%	6,7%	75,8%	15,8%	0,0%	0,0%	0,0%	0,0%	0,0%	0,8%
0	0,0%	0,0%	0,0%	0,0%	0,0%	3,3%	65,8%	30,0%	0,8%	0,0%	0,0%	0,0%	0,0%
1	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	4,2%	72,5%	21,7%	1,7%	0,0%	0,0%	0,0%
2	0,0%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%	3,3%	71,7%	20,8%	2,5%	0,0%	0,0%
3	0,0%	1,7%	0,0%	0,0%	0,8%	0,0%	0,0%	2,5%	8,3%	72,5%	12,5%	1,7%	0,0%
4	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%	0,8%	5,0%	80,0%	11,7%	1,7%
5	0,0%	0,8%	0,0%	0,0%	0,8%	0,0%	0,0%	3,3%	0,0%	5,0%	12,5%	53,3%	24,2%
6	0,0%	0,0%	0,8%	0,0%	1,7%	0,8%	0,0%	2,5%	1,7%	0,0%	6,7%	15,0%	70,8%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου με κανόνα εκμάθησης τον **projection learning rule** και με τη δεύτερη διαδικασία επιλογής μνημών

#### 4.1.3 Αποτελέσματα για σύστημα με την τρίτη διαδικασία επιλογής μνημών

- *Hebbian learning rule*

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	63,3%	29,2%	4,2%	0,8%	0,8%	0,0%	1,7%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-5	15,0%	42,5%	27,5%	12,5%	1,7%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-4	0,0%	0,0%	60,8%	30,8%	5,0%	0,8%	2,5%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-3	0,0%	0,0%	2,5%	54,2%	38,3%	3,3%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
-2	0,0%	0,0%	0,0%	6,7%	65,8%	19,2%	6,7%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%
-1	0,0%	0,0%	0,0%	0,0%	7,5%	65,8%	25,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	72,5%	25,8%	0,8%	0,0%	0,0%	0,8%	0,0%
1	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%	4,2%	71,7%	20,8%	2,5%	0,0%	0,0%	0,0%
2	0,0%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%	5,8%	70,0%	20,0%	2,5%	0,0%	0,0%
3	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	2,5%	11,7%	70,8%	13,3%	0,0%	0,0%
4	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	2,5%	0,0%	0,8%	5,0%	79,2%	12,5%	0,0%
5	0,0%	0,0%	0,0%	0,0%	1,7%	0,0%	0,0%	0,8%	0,0%	3,3%	13,3%	51,7%	29,2%
6	0,0%	0,0%	0,0%	0,0%	1,7%	0,0%	0,0%	0,0%	0,8%	0,0%	3,3%	14,2%	80,0%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου με κανόνα εκμάθησης τον **hebbian learning rule** και με την τρίτη διαδικασία επιλογής μνημών

- *Learning in bounds*

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	66,7%	27,5%	4,2%	0,8%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-5	15,0%	46,7%	27,5%	12,5%	1,7%	5,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-4	0,0%	0,0%	60,8%	30,8%	5,0%	0,8%	2,5%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-3	0,0%	0,0%	2,5%	54,2%	38,3%	3,3%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
-2	0,0%	0,0%	0,0%	6,7%	65,8%	19,2%	6,7%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%
-1	0,0%	0,0%	0,0%	0,0%	7,5%	65,8%	25,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
0	0,0%	0,0%	0,0%	0,0%	0,0%	4,2%	69,2%	25,8%	0,0%	0,0%	0,0%	0,8%	0,0%
1	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%	4,2%	71,7%	20,8%	2,5%	0,0%	0,0%	0,0%
2	0,0%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%	5,8%	70,0%	20,0%	2,5%	0,0%	0,0%
3	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	2,5%	11,7%	70,8%	13,3%	0,0%	0,0%
4	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	2,5%	0,0%	0,8%	5,0%	79,2%	12,5%	0,0%
5	0,0%	0,0%	0,0%	0,0%	1,7%	0,0%	0,0%	0,0%	0,0%	3,3%	10,0%	54,2%	29,2%
6	0,0%	0,0%	0,0%	0,0%	1,7%	0,0%	0,0%	0,0%	0,8%	0,0%	3,3%	14,2%	80,0%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου με κανόνα εκμάθησης το learning in bounds και με την τρίτη διαδικασία επιλογής μηνμών

- *Projection learning rule*

	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6
-6	67,5%	28,3%	2,5%	0,8%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-5	13,3%	53,3%	26,7%	5,8%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-4	0,0%	0,8%	60,8%	32,5%	5,0%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-3	0,0%	0,0%	5,8%	60,0%	30,8%	2,5%	0,8%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
-2	0,0%	0,0%	0,0%	7,5%	66,7%	19,2%	5,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
-1	0,0%	0,0%	0,0%	0,0%	7,5%	65,8%	25,8%	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%
0	0,0%	0,0%	0,0%	0,0%	0,0%	4,2%	73,3%	20,8%	0,8%	0,0%	0,0%	0,8%	0,0%
1	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%	5,8%	71,7%	20,8%	0,8%	0,0%	0,0%	0,0%
2	0,0%	0,8%	0,0%	0,0%	0,0%	0,8%	0,0%	5,8%	70,8%	19,2%	2,5%	0,0%	0,0%
3	0,0%	0,8%	0,0%	0,0%	0,0%	0,0%	0,8%	2,5%	11,7%	70,8%	13,3%	0,0%	0,0%
4	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	2,5%	5,8%	79,2%	12,5%	0,0%
5	0,0%	0,0%	0,0%	0,0%	0,8%	0,0%	0,0%	0,8%	0,0%	3,3%	15,0%	55,8%	24,2%
6	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	5,0%	14,2%	80,8%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου με κανόνα εκμάθησης τον projection learning rule και με την τρίτη διαδικασία επιλογής μηνμών

#### 4.1.4 Παρατηρήσεις

- Για όλες τις διαδικασίες επιλογής μνημών το σύστημα έχει καλύτερη απόδοση όταν χρησιμοποιείται ο projection learning rule. Κάτι τέτοιο είναι αναμενόμενο, αφού σε όλες τις περιπτώσεις οι συσχετίσεις μεταξύ των μνημών είναι ιδιαίτερα υψηλές.
- Οι εικόνες κάθε κλάσης που κατατάσσονται λάθος, κατατάσσονται κατά μεγαλύτερο ποσοστό στις κλάσεις εκείνες των οποίων οι μνήμες παρουσιάζουν υψηλή συσχέτιση με τη μνήμη της κλάσης που εξετάζουμε.
- Με τη μέθοδο learning in bounds τα αποτελέσματα βελτιώνονται, αφού το δίκτυο εκπαιδεύεται με νέα patterns τα οποία συσχετίζει με τη μνήμη στην οποία ανήκουν. Η ανανέωση των νευρώνων του δικτύου γίνεται με σύγχρονο τρόπο (όλοι οι νευρώνες ανανεώνονται ταυτόχρονα). Έτσι, το δίκτυο παγιδεύεται λιγότερο συχνά σε τοπικά ελάχιστα του energy function. Αυτό έχει σαν αποτέλεσμα πολλές φορές, το δίκτυο να μην συγκλίνει σε κάποια κατάσταση κοντά στην κοντινότερη μνήμη αλλά σε καταστάσεις μακριά από αυτή. Γι'αυτό και παρατηρούμε πως τα αποτελέσματα δεν είναι συγκεντρωμένα γύρω από την κλάση την οποία εξετάζουμε ,αλλά παρατηρούμε μεγαλύτερη διασπορά στις 'μακρινές' κλάσεις .
- Με τη μέθοδο του projection τα αποτελέσματα σε όλες τις περιπτώσεις βελτιώνονται αφού δουλεύουμε με highly correlated memories. Η πιο σημαντική βελτίωση με τη με τη χρήση της μεθόδου αυτής είναι ότι τα αποτελέσματα είναι συγκεντρωμένα γύρω από την κλάση την οποία εξετάζουμε. Κάτι τέτοιο είναι λογικό, αφού ο κανόνας αυτός συσχετίζει τις μνήμες μεταξύ τους, λαμβάνοντας υπ'όψην το συντελεστή συσχέτισης τους. Έτσι, οι εικόνες οι οποίες κατατάσσονται σε λάθος κλάση, ανήκουν κυρίως σε κλάσεις που παρουσιάζουν υψηλή συσχέτιση σε σχέση με την κλάση που εξετάζουμε. Τα αποτελέσματα βελτιώνονται περισσότερο στην περίπτωση που χρησιμοποιούμε ιδανικές μνήμες και λιγότερο στην περίπτωση των dilated μνημών. Αυτό συμβαίνει γιατί στην πρώτη περίπτωση οι συσχετίσεις είναι πολύ υψηλές ενώ στη δεύτερη χαμηλότερες. Όμως υπάρχει σε όλες τις περιπτώσεις βελτίωση, αφού παντού έχουμε σημαντικές συσχετίσεις.
- Παρατηρούμε ακόμη πως σε όλες τις περιπτώσεις λάθος κατηγοριοποίησης το σύστημα κατατάσσει τις συγκεκριμένες εικόνες στις κοντινότερες κλάσεις αυτής της οποίας εξετάζουμε. Το ποσοστό των εικόνων το οποίο κατατάσσεται σε 'μακρινές' κλάσεις είναι πολύ χαμηλό. Επίσης, σχεδόν σε καμία περίπτωση το σύστημα δεν κατατάσσει εικόνες που έχουν μετατόπιση προς τα επάνω σε κλάσεις με μετατόπιση προς τα κάτω. Το σύστημα μπορεί να μην έχει ιδιαίτερα υψηλή απόδοση, όμως οι εικόνες πάντα κατηγοριοποιούνται σε κλάσεις κοντινές σε αυτήν στην οποία πραγματικά ανήκουν.



Θα είχε ενδιαφέρον λοιπόν να εξετάσουμε την απόδοση του δικτύου για περιπτώσεις που έχουμε μετατόπιση ανά 2 ή ανά 3 pixels. Αυτό σημαίνει πως στην πρώτη περίπτωση θα έχουμε μόνο τις μνήμες  $0, \pm 2, \pm 4, \pm 6$  και στη δεύτερη περίπτωση τις μνήμες  $0, \pm 3, \pm 6$ . Η απόδοση του δικτύου για τις περιπτώσεις αυτές φαίνεται στις 2 επόμενες παραγράφους (4.2 και 4.3).

## 4.2 Απόδοση του δικτύου για μετατόπιση ανά 2 pixels

Οι παρακάτω πίνακες παρουσιάζουν τα αποτελέσματα του συστήματος κατηγοριοποίησης (σε ποσοστά %) για μετατοπίσεις ανά 2 pixels. Όπως και στην προηγούμενη παράγραφο, οι οριζόντιες γραμμές δείχνουν τις κλάσεις στις οποίες οι εικόνες μας ανήκουν και οι κατακόρυφες τις κλάσεις στις οποίες το σύστημα τις κατατάσσει.

### 4.2.1 Αποτελέσματα για σύστημα με την πρώτη διαδικασία επιλογής μνημών

- *Hebbian learning rule*

	-6	-4	-2	0	2	4	6
-6	76,67%	17,50%	0,83%	3,33%	0,00%	0,00%	1,67%
-4	0,00%	89,17%	4,17%	6,67%	0,00%	0,00%	0,00%
-2	0,00%	11,67%	76,67%	9,17%	0,00%	0,00%	2,50%
0	0,00%	0,00%	1,67%	88,33%	3,33%	1,67%	5,00%
2	0,83%	0,00%	0,83%	0,83%	72,50%	19,17%	5,83%
4	0,00%	0,00%	0,00%	3,33%	0,00%	85,00%	11,67%
6	0,00%	2,50%	1,67%	0,00%	0,83%	10,83%	84,17%

Πίνακας 4. Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 2 pixel με κανόνα εκμάθησης τον hebbian learning rule και με την πρώτη διαδικασία επιλογής μνημών

- *Learning in bounds*

	-6	-4	-2	0	2	4	6
-6	80,00%	13,33%	2,50%	3,33%	0,00%	0,83%	0,00%
-4	0,00%	86,67%	9,17%	3,33%	0,00%	0,83%	0,00%
-2	0,00%	10,83%	83,33%	5,00%	0,83%	0,00%	0,00%
0	0,83%	3,33%	8,33%	73,33%	10,83%	1,67%	1,67%
2	1,67%	2,50%	2,50%	2,50%	85,00%	4,17%	1,67%
4	0,00%	0,00%	1,67%	3,33%	13,33%	79,17%	2,50%
6	0,00%	2,50%	2,50%	4,17%	1,67%	9,17%	80,00%

Πίνακας 4. Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 2 pixel με κανόνα εκμάθησης το learning in bounds και με την πρώτη διαδικασία επιλογής μνημών

- *Projection learning rule*

	-6	-4	-2	0	2	4	6
-6	77,50%	19,17%	3,33%	0,00%	0,00%	0,00%	0,83%
-4	0,00%	89,17%	6,67%	4,17%	0,00%	0,00%	0,00%
-2	0,00%	11,67%	78,33%	9,17%	0,00%	0,00%	0,83%
0	0,00%	0,00%	3,33%	88,33%	5,83%	1,67%	0,83%
2	0,83%	0,00%	0,00%	2,50%	75,83%	18,33%	3,33%
4	0,00%	0,00%	0,00%	0,83%	2,50%	85,00%	11,67%
6	0,00%	0,83%	0,00%	0,00%	0,83%	13,33%	85,00%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 2 pixel με κανόνα εκμάθησης τον projection learning rule και με την πρώτη διαδικασία επιλογής μνημών

#### 4.2.2 Αποτελέσματα για σύστημα με τη δεύτερη διαδικασία επιλογής μνημών

- *Hebbian learning rule*

	-6	-4	-2	0	2	4	6
-6	85,00%	7,50%	3,33%	4,17%	0,00%	0,00%	0,00%
-4	0,83%	78,33%	13,33%	7,50%	0,83%	0,00%	0,00%
-2	0,00%	5,83%	85,00%	8,33%	0,00%	0,00%	0,83%
0	0,00%	0,83%	5,83%	83,33%	10,00%	0,00%	0,00%
2	0,83%	0,00%	2,50%	1,67%	85,00%	10,00%	0,83%
4	0,00%	0,00%	3,33%	3,33%	2,50%	82,50%	8,33%
6	0,00%	1,67%	12,50%	3,33%	2,50%	5,83%	74,17%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 2 pixel με κανόνα εκμάθησης τον hebbian learning rule και με τη δεύτερη διαδικασία επιλογής μνημών

- *Learning in bounds*

	-6	-4	-2	0	2	4	6
-6	80,00%	12,50%	3,33%	4,17%	0,00%	0,00%	0,00%
-4	0,83%	75,83%	13,33%	9,17%	0,83%	0,00%	0,00%
-2	0,00%	1,67%	88,33%	9,17%	0,00%	0,00%	0,83%
0	0,00%	0,00%	1,67%	90,00%	8,33%	0,00%	0,00%
2	0,83%	0,00%	2,50%	1,67%	82,50%	11,67%	0,83%
4	0,00%	0,00%	3,33%	3,33%	2,50%	83,33%	7,50%
6	0,00%	1,67%	12,50%	4,17%	4,17%	6,67%	70,83%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 2 pixel με κανόνα εκμάθησης το learning in bounds και με τη δεύτερη διαδικασία επιλογής μνημών

- *Projection learning rule*

	-6	-4	-2	0	2	4	6
-6	80,00%	13,33%	6,67%	0,00%	0,00%	0,00%	0,00%
-4	1,67%	78,33%	13,33%	6,67%	0,00%	0,00%	0,00%
-2	0,00%	1,67%	86,67%	10,83%	0,00%	0,00%	0,83%
0	0,00%	0,00%	2,50%	89,17%	8,33%	0,00%	0,00%
2	0,83%	0,00%	0,83%	4,17%	82,50%	10,83%	0,83%
4	0,00%	0,00%	3,33%	3,33%	2,50%	83,33%	7,50%
6	0,00%	0,00%	0,00%	3,33%	7,50%	13,33%	75,83%

Πίνακας 4. Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 2 pixel με κανόνα εκμάθησης τον projection learning rule και με τη δεύτερη διαδικασία επιλογής μνημών

#### 4.2.3 Αποτελέσματα για σύστημα με την τρίτη διαδικασία επιλογής μνημών

- *Hebbian learning rule*

	-6	-4	-2	0	2	4	6
-6	82,50%	15,00%	0,00%	2,50%	0,00%	0,00%	0,00%
-4	0,00%	79,17%	18,33%	2,50%	0,00%	0,00%	0,00%
-2	0,00%	3,33%	70,83%	25,00%	0,00%	0,00%	0,83%
0	0,00%	0,00%	0,00%	96,67%	2,50%	0,00%	0,83%
2	0,00%	0,83%	0,00%	4,17%	82,50%	12,50%	0,00%
4	0,00%	0,00%	0,00%	3,33%	2,50%	86,67%	7,50%
6	0,00%	0,00%	1,67%	0,83%	0,00%	6,67%	90,83%

Πίνακας 4. Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 2 pixel με κανόνα εκμάθησης τον hebbian learning rule και με την τρίτη διαδικασία επιλογής μνημών

- *Learning in bounds*

	-6	-4	-2	0	2	4	6
-6	83,33%	14,17%	0,83%	1,67%	0,00%	0,00%	0,00%
-4	0,00%	82,50%	15,83%	1,67%	0,00%	0,00%	0,00%
-2	1,67%	3,33%	76,67%	17,50%	0,00%	0,00%	0,83%
0	0,00%	3,33%	13,33%	80,00%	2,50%	0,00%	0,83%
2	0,00%	0,83%	1,67%	3,33%	86,67%	7,50%	0,00%
4	0,00%	0,00%	0,83%	3,33%	3,33%	85,83%	6,67%
6	0,00%	0,00%	1,67%	1,67%	1,67%	10,00%	85,00%

Πίνακας 4. Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 2 pixel με κανόνα εκμάθησης το learning in bounds και με την τρίτη διαδικασία επιλογής μνημών

- *Projection learning rule*

	-6	-4	-2	0	2	4	6
-6	85,00%	14,17%	0,83%	0,00%	0,00%	0,00%	0,00%
-4	0,00%	84,17%	15,83%	0,00%	0,00%	0,00%	0,00%
-2	0,00%	1,67%	77,50%	20,00%	0,00%	0,00%	0,83%
0	0,00%	0,00%	0,00%	92,50%	0,83%	6,67%	0,00%
2	0,00%	0,00%	0,00%	5,00%	82,50%	12,50%	0,00%
4	0,00%	0,00%	0,00%	0,83%	8,33%	82,50%	8,33%
6	0,00%	0,00%	0,00%	1,67%	3,33%	7,50%	88,33%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 2 pixel με κανόνα εκμάθησης τον projection learning rule και με την τρίτη διαδικασία επιλογής μνημών

### 4.3 Απόδοση του δικτύου για μετατόπιση ανά 3 pixels

Οι παρακάτω πίνακες παρουσιάζουν τα αποτελέσματα του συστήματος κατηγοριοποίησης (σε ποσοστά %) για μετατοπίσεις ανά 3 pixels. Όπως και στην προηγούμενη παράγραφο, οι οριζόντιες γραμμές δείχνουν τις κλάσεις στις οποίες οι εικόνες μας ανήκουν και οι κατακόρυφες τις κλάσεις στις οποίες το σύστημα τις κατατάσσει.

#### 4.3.1 Αποτελέσματα για σύστημα με την πρώτη διαδικασία επιλογής μνημών

- *Hebbian learning rule*

	-6	-3	0	3	6
-6	87,50%	6,67%	5,83%	0,00%	0,00%
-3	3,33%	93,33%	3,33%	0,00%	0,00%
0	0,00%	6,67%	85,83%	5,00%	2,50%
3	2,50%	4,17%	0,83%	88,33%	4,17%
6	3,33%	6,67%	1,67%	6,67%	81,67%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 3 pixel με κανόνα εκμάθησης τον hebbian learning rule και με την πρώτη διαδικασία επιλογής μνημών

- *Learning in bounds*

	-6	-3	0	3	6
-6	85,00%	6,67%	5,83%	0,00%	2,50%
-3	4,17%	89,17%	5,83%	0,00%	0,83%
0	0,83%	9,17%	67,50%	20,83%	1,67%
3	9,17%	2,50%	2,50%	87,50%	6,67%
6	3,33%	0,83%	1,67%	5,00%	89,17%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 3 pixel με κανόνα εκμάθησης το *learning in bounds* και με την πρώτη διαδικασία επιλογής μνημών

- *Projection learning rule*

	-6	-3	0	3	6
-6	89,17%	7,50%	3,33%	0,00%	0,00%
-3	3,33%	93,33%	3,33%	0,00%	0,00%
0	0,00%	6,67%	86,67%	6,67%	0,00%
3	0,00%	0,00%	8,33%	87,50%	4,17%
6	0,00%	1,67%	8,33%	7,50%	82,50%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 3 pixel με κανόνα εκμάθησης τον *projection learning rule* και με την πρώτη διαδικασία επιλογής μνημών

#### 4.3.2 Αποτελέσματα για σύστημα με τη δεύτερη διαδικασία επιλογής μνημών

- *Hebbian learning rule*

	-6	-3	0	3	6
-6	87,50%	4,17%	8,33%	0,00%	0,00%
-3	1,67%	86,67%	10,83%	0,00%	0,83%
0	0,83%	0,00%	96,67%	2,50%	0,00%
3	1,67%	0,00%	9,17%	89,17%	0,00%
6	1,67%	0,83%	15,00%	5,83%	76,67%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 3 pixel με κανόνα εκμάθησης τον *hebbian learning rule* και με τη δεύτερη διαδικασία επιλογής μνημών

- *Learning in bounds*

	-6	-3	0	3	6
-6	87,50%	3,33%	5,00%	2,50%	1,67%
-3	3,33%	86,67%	8,33%	0,83%	0,83%
0	0,83%	3,33%	81,67%	10,83%	3,33%
3	1,67%	0,00%	7,50%	89,17%	1,67%
6	1,67%	0,83%	13,33%	5,83%	78,33%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 3 pixel με κανόνα εκμάθησης το *learning in bounds* και με τη δεύτερη διαδικασία επιλογής μνημών

- *Projection learning rule*

	-6	-3	0	3	6
-6	88,33%	4,17%	7,50%	0,00%	0,00%
-3	1,67%	86,67%	10,83%	0,00%	0,83%
0	0,83%	0,00%	96,67%	2,50%	0,00%
3	0,00%	0,00%	9,17%	90,00%	0,83%
6	0,00%	0,83%	8,33%	7,50%	83,33%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 3 pixel με κανόνα εκμάθησης τον *projection learning rule* και με τη δεύτερη διαδικασία επιλογής μνημών

#### 4.3.3 Αποτελέσματα για σύστημα με την τρίτη διαδικασία επιλογής μνημών

- *Hebbian learning rule*

	-6	-3	0	3	6
-6	86,67%	10,00%	3,33%	0,00%	0,00%
-3	0,00%	76,67%	22,50%	0,00%	0,83%
0	0,00%	0,00%	95,00%	4,17%	0,83%
3	0,83%	0,00%	5,00%	93,33%	0,83%
6	0,83%	0,00%	3,33%	5,00%	90,83%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 3 pixel με κανόνα εκμάθησης τον *hebbian learning rule* και με την τρίτη διαδικασία επιλογής μνημών

- *Learning in bounds*

	-6	-3	0	3	6
-6	68,33%	31,67%	0,00%	0,00%	0,00%
-3	4,17%	93,33%	1,67%	0,00%	0,83%
0	0,00%	1,67%	95,00%	2,50%	0,83%
3	0,83%	2,50%	11,67%	70,83%	14,17%
6	0,00%	0,83%	7,50%	0,83%	90,83%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 3 pixel με κανόνα εκμάθησης το learning in bounds και με την τρίτη διαδικασία επιλογής μνημών

- *Projection learning rule*

	-6	-3	0	3	6
-6	86,67%	10,00%	3,33%	0,00%	0,00%
-3	0,00%	79,17%	20,00%	0,00%	0,83%
0	0,00%	0,83%	95,00%	4,17%	0,00%
3	0,83%	0,00%	5,00%	92,50%	1,67%
6	0,00%	0,00%	1,67%	5,00%	93,33%

**Πίνακας 4.**Error! Unknown switch argument. Απόδοση δικτύου για μετατοπίσεις ανά 3 pixel με κανόνα εκμάθησης τον projection learning rule και με την τρίτη διαδικασία επιλογής μνημών

#### 4.4 Μέσοι αριθμοί επαναλήψεων

Όπως, έχει αναφερθεί εκτενέστερα και στο κεφάλαιο 3, όταν μια εικόνα εμφανίζεται στο δίκτυο Hopfield, τότε αρχίζει επαναληπτικά η ασύγχρονη ανανέωση της, μέχρι να συγκλίνει σε κάποια από τις αποθηκευμένες μνήμες ή σε κάποιο spurious state, δηλαδή σε μια σταθερή κατάσταση. Ο μέσος αριθμός των επαναλήψεων που απαιτούνται μέχρι το σύστημα να σταθεροποιηθεί σε μια κατάσταση είναι :

	Hebbian Learning	Learning in bounds	Projection learning
διαδικασία1	7	11	14
διαδικασία2	9	12	16
διαδικασία3	12	14	18

**Πίνακας 4.**Error! Unknown switch argument. Μέσοι αριθμοί επαναλήψεων μέχρι το σύστημα να φτάσει σε σταθερή κατάσταση

Λαμβάνοντας υπ'όψη και τον πίνακα 3.5 που δείχνει τα ποσοστά εικόνων που φτάνουν σε spurious states μπορούμε να συμπεράνουμε πως για κάθε διαδικασία όσο μειώνονται τα ποσοστά των spurious states τόσο αυξάνονται οι επαναλήψεις που συμβαίνουν για να φτάσει το δίκτυο σε σταθερή κατάσταση. Αυτό είναι αναμενόμενο, αφού για να σταθεροποιηθεί το δίκτυο σε μια κατάσταση που συμπίπτει με κάποια μνήμη απαιτούνται γενικά περισσότερες επαναλήψεις από το να σταθεροποιηθεί σε κάποια spurious κατάσταση.

#### 4.5 Παραδείγματα διάδοσης στο δίκτυο

Παρουσιάζεται στη συνέχεια ένα παράδειγμα διάδοσης στο δίκτυο.

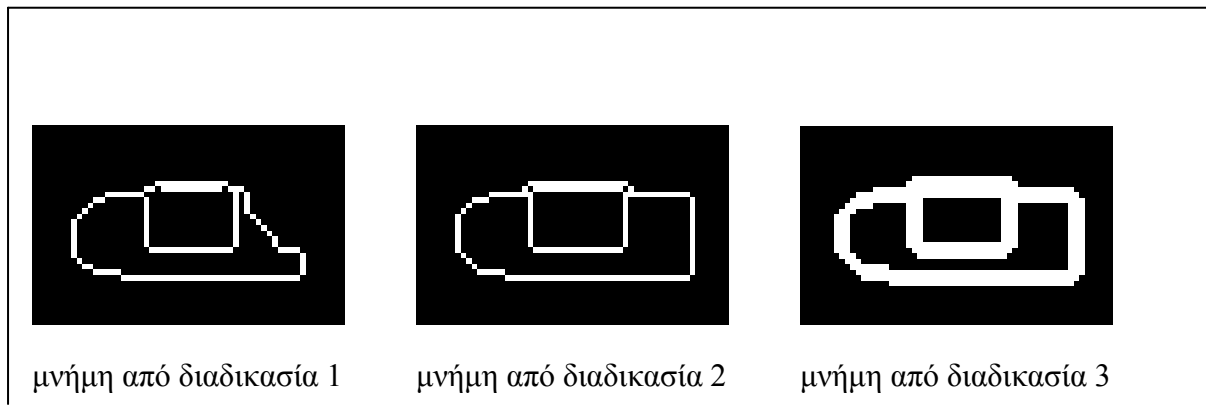
Έχουμε μια εικόνα που ανήκει στην κλάση των +3 pixels μετατόπιση. Στο σχήμα 4.1 φαίνεται η αρχική εικόνα καθώς και η ίδια εικόνα μετά την προεπεξεργασία και μετά την εφαρμογή του τελεστή dilation.



Σχήμα 4. Error! Unknown switch argument. Εικόνα προς διάδοση στο σύστημα

Στο σχήμα 4.2 φαίνονται οι μνήμες των +3 pixels μετατόπιση. Σε αυτές τις εικόνες θέλουμε οι εικόνες του σχήματος 4.1 να συγκλίνουν.

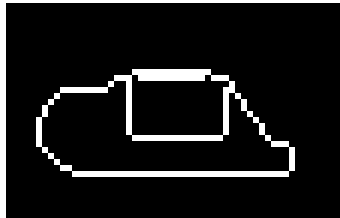




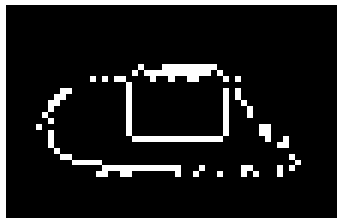
**Σχήμα 4.**Error! Unknown switch argument. **Μνήμες για την κλάση των +3 pixels μετατόπιση**

Παρακάτω παρουσιάζεται η επαναληπτική διαδικασία ανακατασκευής των εικόνων του σχήματος και για τις 3 διαδικασίες επιλογής μνημών. Ο κανόνας εκμάθησης στην περίπτωση που παρουσιάζονται είναι ο κανόνας Hebb. Σημειώνεται ότι και στις τρεις περιπτώσεις που παρουσιάζονται το δίκτυο συγκλίνει τελικά στη σωστή εικόνα.

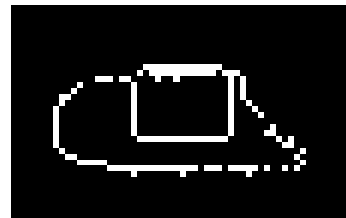
Στο σχήμα 4.3 παρουσιάζεται η πορεία της εικόνας στο δίκτυο (trajectory), με το δίκτυο να έχει εκπαιδευτεί με μνήμες που έχουν επιλεγεί σύμφωνα με την πρώτη διαδικασία. Οι εικόνες που παρουσιάζονται είναι η έξοδος του δικτύου μετά από κάθε επανάληψη, μέχρι το δίκτυο να σταθεροποιηθεί σε κάποια κατάσταση.



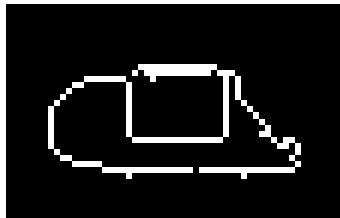
αρχική εικόνα



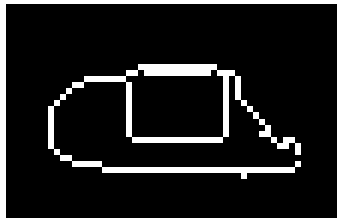
πρώτη επανάληψη



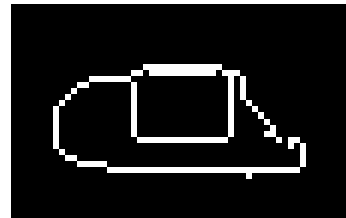
δεύτερη επανάληψη



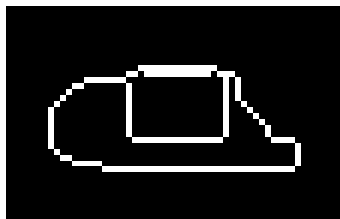
τρίτη επανάληψη



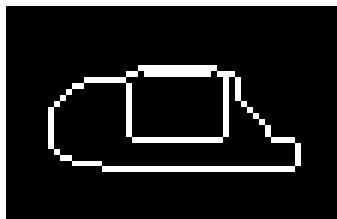
τέταρτη επανάληψη



πέμπτη επανάληψη



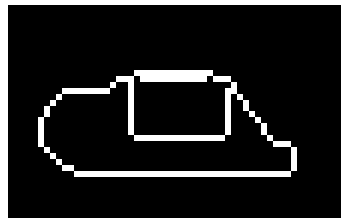
έκτη επανάληψη



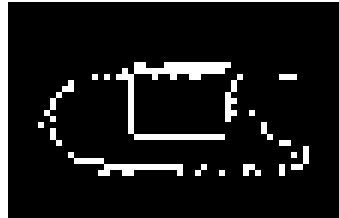
τελική εικόνα ( έξοδος δικτύου)

**Σχήμα 4.**Error! Unknown switch argument. Trajectory της εικόνας του σχήματος 4.1 στο δίκτυο στο οποίο οι μνήμες έχουν επιλέγει από τη διαδικασία 1

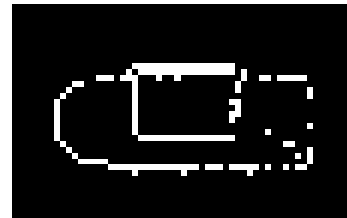
Στο σχήμα 4.4 παρουσιάζεται η πορεία της εικόνας στο δίκτυο (trajectory), με το δίκτυο να έχει εκπαιδευτεί με μνήμες που έχουν επιλέγει σύμφωνα με τη δεύτερη διαδικασία.



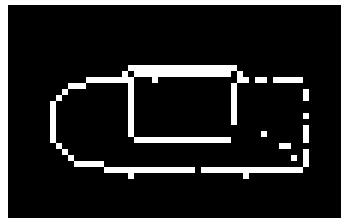
αρχική εικόνα



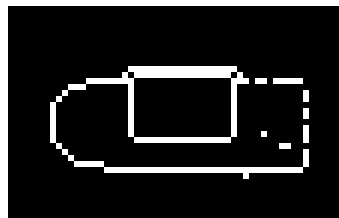
πρώτη επανάληψη



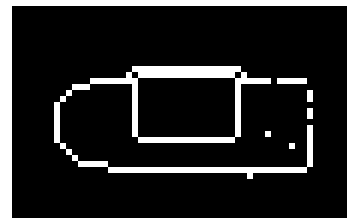
δεύτερη επανάληψη



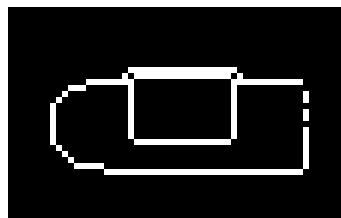
τρίτη επανάληψη



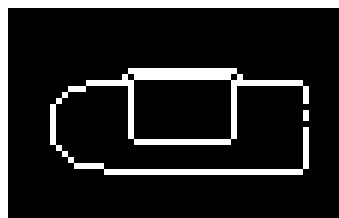
τέταρτη επανάληψη



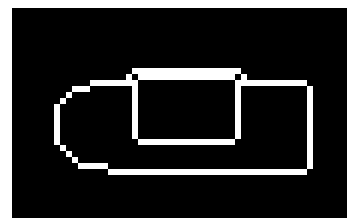
πέμπτη επανάληψη



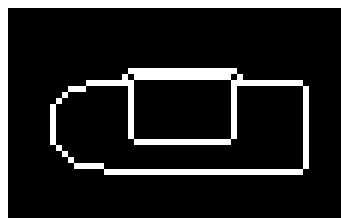
έκτη επανάληψη



έβδομη επανάληψη



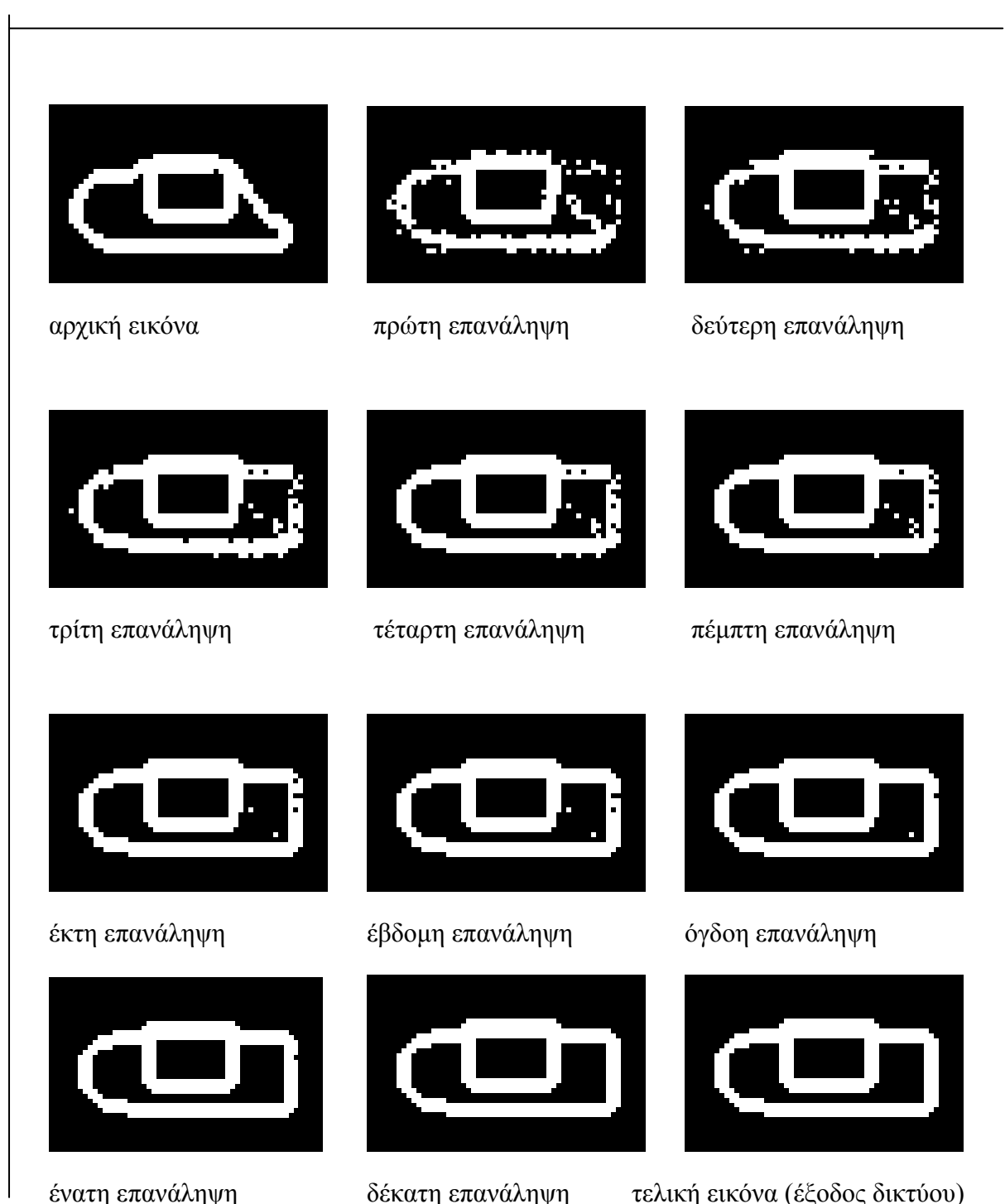
όγδοη επανάληψη



τελική εικόνα ( έξοδος δικτύου)

**Σχήμα 4.**Error! Unknown switch argument. Trajectory της εικόνας του σχήματος 4.1 στο δίκτυο στο οποίο οι μνήμες έχουν επιλέγει από τη διαδικασία 2

Στο σχήμα 4.5 παρουσιάζεται η πορεία της εικόνας στο δίκτυο (trajectory), με το δίκτυο να έχει εκπαιδευτεί με μηνύες που έχουν επιλέγει σύμφωνα με την τρίτη διαδικασία.



**Σχήμα 4.**Error! Unknown switch argument. Trajectory της εικόνας του σχήματος 4.1 στο δίκτυο στο οποίο οι μηνύες έχουν επιλέγει από τη διαδικασία 3

### Παρατηρήσεις:

Κατά τις πρώτες επαναλήψεις αλλάζουν πολλοί περισσότεροι κόμβοι του δικτύου απ' ότι κατά τις τελευταίες επαναλήψεις. Άρα, μπορούμε να συμπεράνουμε ότι όσο το δίκτυο πλησιάζει σε ένα τοπικό ελάχιστο της συνάρτησης ενέργειας του, η ενέργεια του μειώνεται κατά μικρότερο ποσό από όσο όταν βρίσκεται μακριά από ένα τέτοιο σημείο.

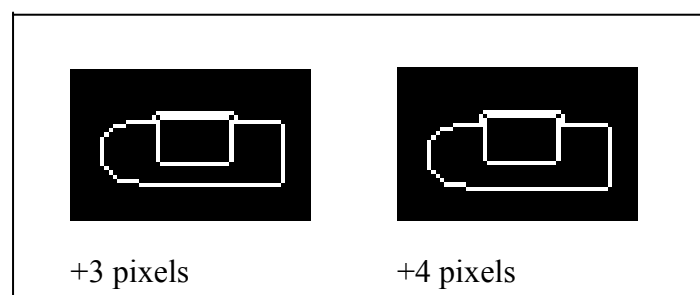
## 4.6 Αναλύοντας την ποιότητα του συστήματος

Είναι ιδιαίτερα δύσκολο να συμπεράνουμε οτιδήποτε για τις εικόνες οι οποίες κατηγοριοποιούνται λάθος. Τα αποτελέσματα της κατηγοριοποίησης εξαρτώνται από τον τρόπο με τον οποίο έχουν δημιουργηθεί οι περιοχές σύγκλισης των μνημών (basins of attractors). Επειδή στο δίκτυο Hopfield διαδίδεται όλη η εικόνα, το κάθε στοιχείο της είναι δυνατόν να επηρεάσει το αποτέλεσμα της σύγκλισης. Παρακάτω παρουσιάζουμε ένα τέτοιο παράδειγμα. Είναι μια εικόνα της κλάσης των +3 pixels up (σχήμα 4.6), η οποία με τον κανόνα εκπαίδευσης του Hebb βρίσκεται ότι ανήκει στην κλάση των +4 pixels up, ενώ με τον κανόνα εκπαίδευσης του projection βρίσκεται ότι ανήκει στην σωστή κλάση.



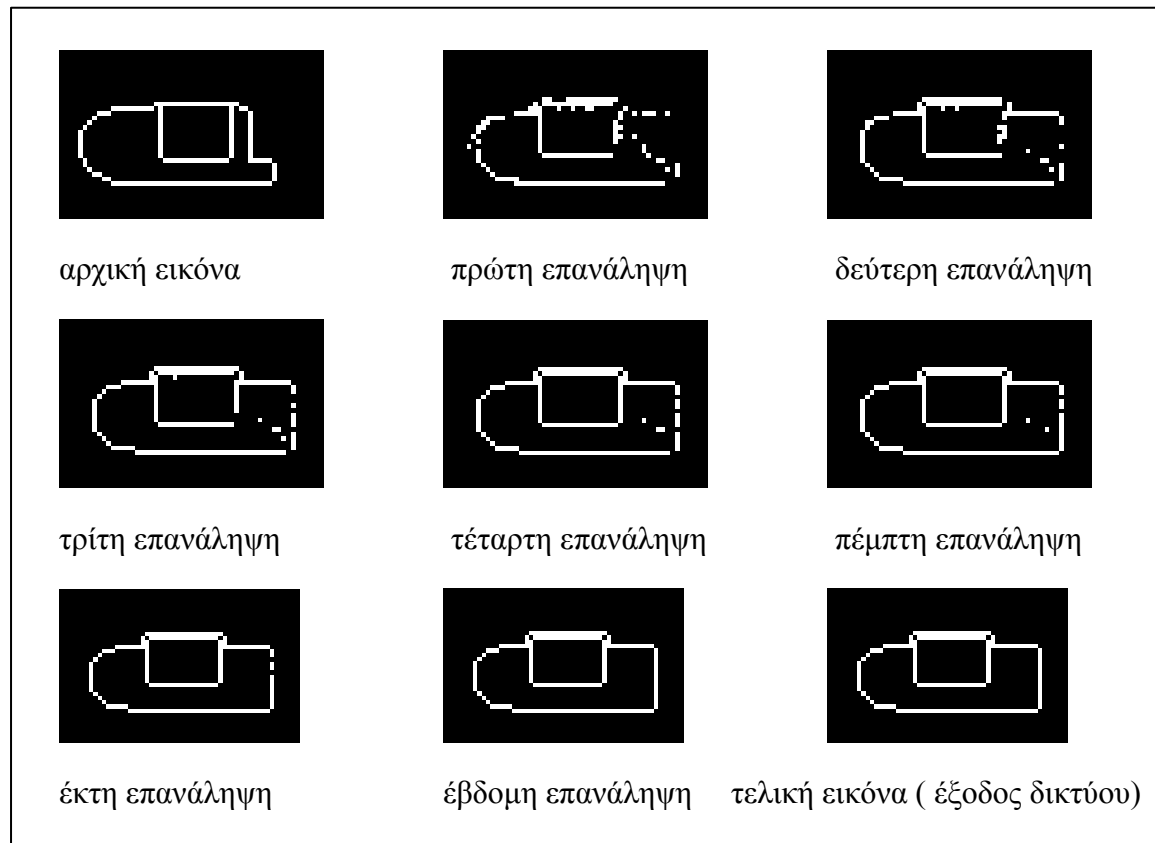
Σχήμα 4. Error! Unknown switch argument. Εικόνα προς διάδοση στο σύστημα

Στο σχήμα 4.7 φαίνονται οι μνήμες των +3 και +4 pixels μετατόπιση, που έχουν επιλέγει από τη διαδικασία 2. Σε αυτές τις εικόνες η εικόνα του σχήματος 4.6 συγκλίνει.



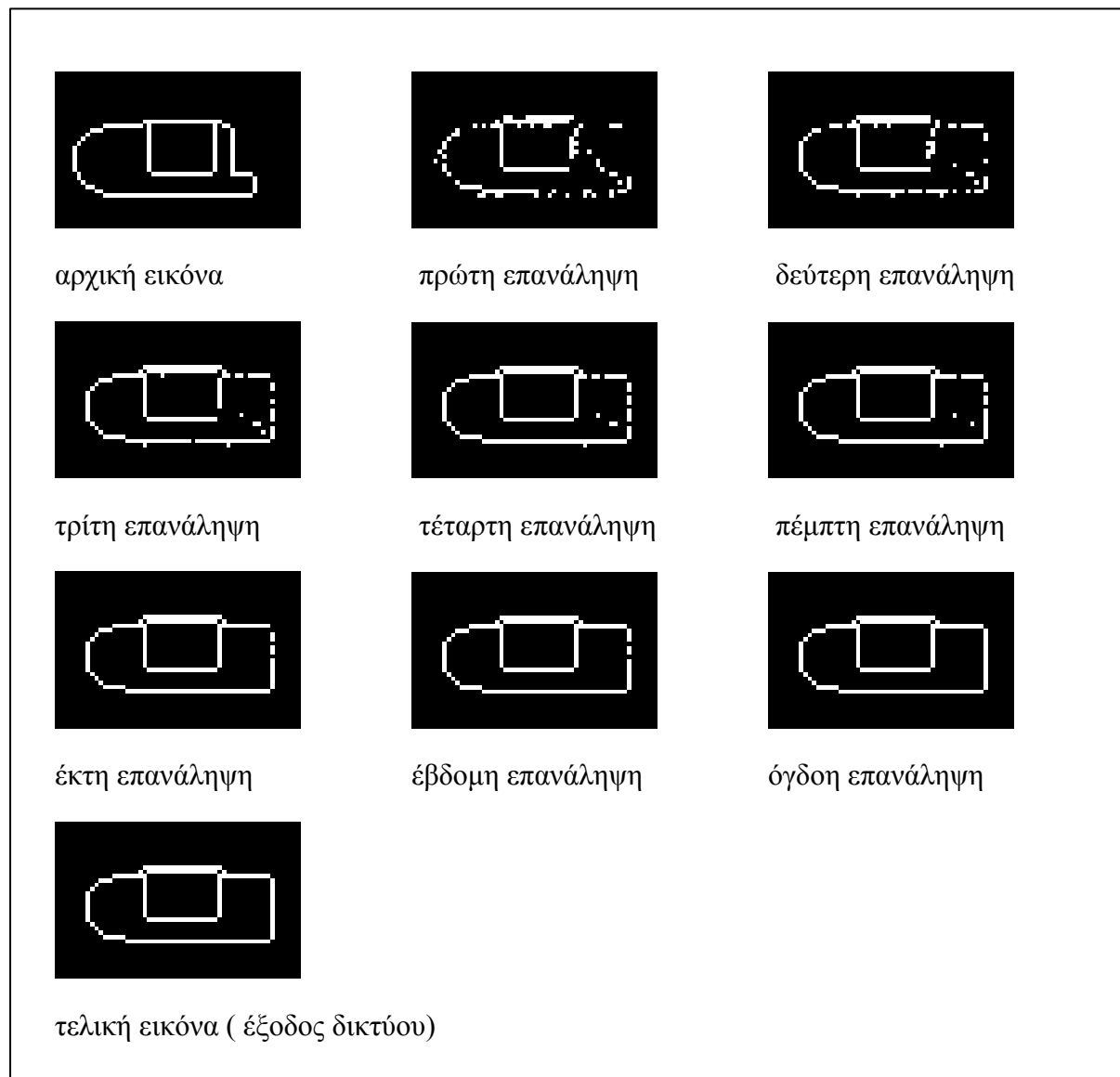
**Σχήμα 4.**Error! Unknown switch argument. **Μνήμες στις οποίες συγκλίνει η εικόνα του σχήματος 4.6**

Στο σχήμα 4.8 παρουσιάζεται η πορεία της εικόνας στο δίκτυο (trajectory), με το δίκτυο να έχει εκπαιδευτεί με τον κανόνα Hebb και μνήμες που έχουν επιλέγει σύμφωνα με τη δεύτερη διαδικασία.



**Σχήμα 4.**Error! Unknown switch argument. **Trajectory της εικόνας του σχήματος 4.6 στο δίκτυο το οποίο έχει εκπαιδευτεί σύμφωνα με τον κανόνα Hebb**

Στο σχήμα 4.9 παρουσιάζεται η πορεία της εικόνας στο δίκτυο ( trajectory) , με το δίκτυο να έχει εκπαιδευτεί με τον κανόνα projection και μνήμες που έχουν επιλέγει σύμφωνα με τη δεύτερη διαδικασία.



**Σχήμα 4.**Error! Unknown switch argument. Trajectory της εικόνας του σχηματος4.6 στο δίκτυο το οποίο έχει εκπαιδευτεί σύμφωνα με τον κανόνα projection





## 5 Αξιολόγηση – επεκτάσεις

Στο κεφάλαιο αυτό παρουσιάζεται η αξιολόγηση του συστήματος κατηγοριοποίησης καθώς και πιθανές μελλοντικές του επεκτάσεις.

### 5.1 Αξιολόγηση

Στα πλεονεκτήματα του συστήματος μας συγκαταλέγεται το γεγονός ότι για την εκπαίδευση των δικτύων που χρησιμοποιούνται απαιτείται μόνο η γνώση των μνημών και όχι δείγματα εικόνων, όπως συμβαίνει σε αλλά νευρωνικά δίκτυα. Επίσης, είναι ανεξάρτητο της θέσης του αντικειμένου στην εικόνα, μια και κατά τη φάση της προεπεξεργασίας αυτό μετατοπίζεται στο κέντρο της εικόνας. Επιπλέον, τα αποτελέσματα της κατηγοριοποίησης, εάν είναι λανθασμένα, είναι πάντα οι κοντινότερες κλάσεις σε αυτή την οποία εξετάζουμε. Γι' αυτό και η απόδοση του συστήματος για μετατοπίσεις ανά 2 ή 3 pixels είναι ιδιαίτερα υψηλή.

Το γεγονός ότι το σύστημα κατατάσσει τις εικόνες μας, αν όχι στη σωστή, σε κλάσεις κοντινές σε αυτήν είναι ιδιαίτερα σημαντικό, αφού όπως έχει αναφερθεί και στην παράγραφο 1.2 το ζητούμενο είναι ο υπολογισμός της συνολικής μετατόπισης του component πάνω στο PCB και η οποία θα προκύψει από τη μέση μετατόπιση των lead σε κάθε πλευρά του κυκλώματος (αναλυτικότερα πίνακες 1.1 και 1.2). Εφόσον λοιπόν είμαστε σε θέση να κατηγοριοποιήσουμε το κάθε lead, αν όχι στη σωστή, σε κλάση κοντινή στην πραγματική του, μπορούμε να υπολογίσουμε ικανοποιητικά τη συνολική μετατόπιση του component. Αναλυτικότερα, παραδείγματα υπολογισμού της συνολικής μετατόπισης του component σε ολοκληρωμένα κυκλώματα φαίνονται στο παράρτημα 2 (σημειώνεται εδώ ότι για την εξαγωγή αυτών των αποτελεσμάτων ακολουθήσαμε τη διαδικασία που μεγιστοποιεί την απόδοση του συστήματος μας, δηλαδή αυτή με κανόνα εκπαίδευσης τον projection learning rule και με την τρίτη διαδικασία επιλογής μνημών).

Το κυριότερο μειονέκτημα του συστήματος είναι το υψηλό κόστος του σε υπολογιστικό χρόνο. Αναλυτικότερα, στο παράρτημα 1 φαίνονται οι συνολικοί χρόνοι που απαιτούνται για τη διάδοση μιας εικόνας στο σύστημα.

Η απόδοση του συστήματος είναι δυνατόν να βελτιωθεί ακόμη περισσότερο εάν πραγματοποιηθούν κάποιες επεκτάσεις οι οποίες αναφέρονται αναλυτικότερα στην επόμενη παράγραφο.

## 5.2 Επεκτάσεις

Η απόδοση του συστήματος μπορεί να βελτιωθεί ακόμη περισσότερο, εάν γίνουν τα εξής :

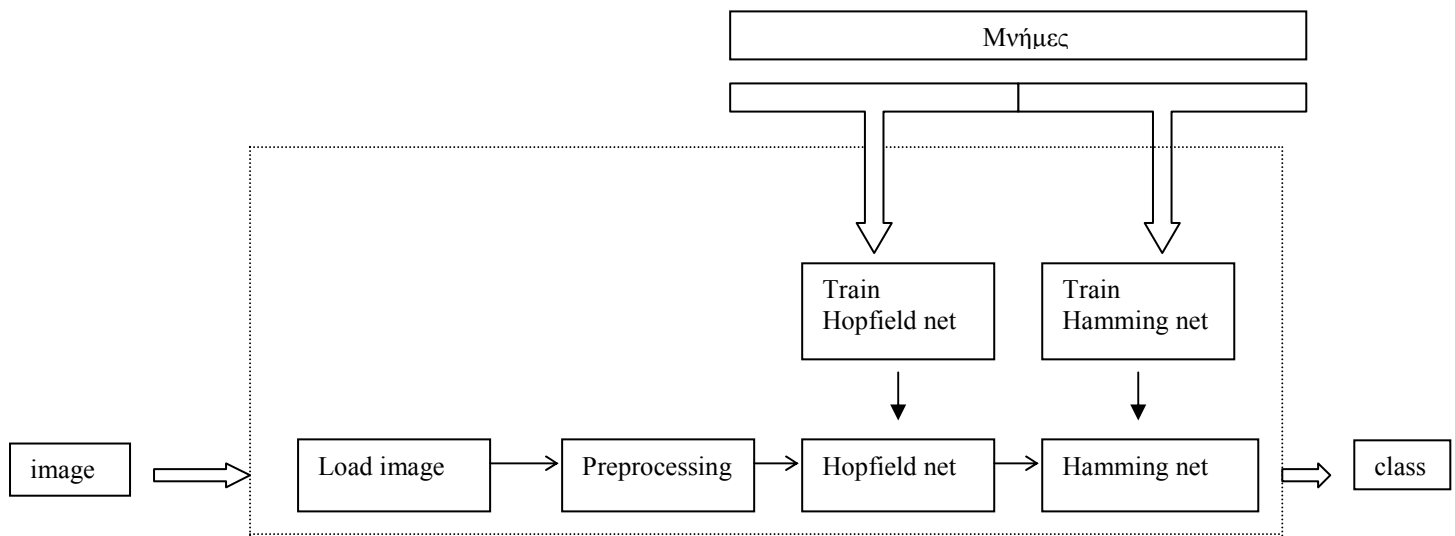
- Εύρεση νέας διαδικασίας επιλογής μνημών έτσι ώστε οι μεταξύ τους συσχετίσεις να είναι χαμηλές.
- Εφαρμογή νέων κανόνων εκμάθησης οι οποίοι είναι αποδοτικοί για περιπτώσεις μνημών με υψηλές συσχετίσεις.
- Εφαρμογή νέου κανόνα κατωφλίας (threshold) σε κάθε νευρώνα που θα αποτρέπει το δίκτυο από το να σταθεροποιείται συχνά σε spurious states.
- Εφαρμογή νέου κανόνα ανανέωσης των νευρώνων κατά τη διάρκεια λειτουργίας του δικτύου, έτσι ώστε η ανανέωση να αρχίζει από τους 'σημαντικότερους' νευρώνες, αυτούς δηλαδή που καθορίζουν την κλάση της εικόνας που διαδίδεται στο δίκτυο.
- Εύρεση νέου κριτηρίου κατηγοριοποίησης των εικόνων σε περιπτώσεις που φτάνουν σε spurious states.
- Εύρεση ενός συστήματος μετατροπής των αρχικών εικόνων μας σε δυαδικές, λαμβάνοντας υπ'όψη και τις 4 περιοχές των αρχικών εικόνων, μια και η προεπεξεργασία που προτείναμε απαλείφει την περιοχή του pad.

## 6 Παράρτημα 1

Στο παράρτημα αυτό αναλύουμε τον κώδικα της εφαρμογής μας, καθώς και τους χρόνους που απαιτούνται για το συνολικό σύστημα.

Η εφαρμογή έχει υλοποιηθεί στη γλώσσα προγραμματισμού C.

Το flow chart της εφαρμογής είναι :



Επειδή όπως έχει αναφερθεί και στο κεφάλαιο 1, η εφαρμογή πρέπει να πραγματοποιείται σε πραγματικό χρόνο, έχει σημασία να δούμε και τους χρόνους που χρειάζονται οι παραπάνω εφαρμογές. Οι μετρήσεις έγιναν σε υπολογιστή PENTIUM II 350MHz σε περιβάλλον Windows NT.

Training of Hopfield network	3.81 sec
Training of Hamming network	0.66 sec
Load image	0.11 sec
Preprocessing	0.02 sec
Propagation in Hopfield network	0.688 sec (for one iteration)
Propagation in Hamming network	0.42 sec

Οπότε, οι συνολικοί χρόνοι για την προετοιμασία του συστήματος (εύρεση μνημών και εκπαίδευση δικτύων) καθώς και για την επεξεργασία κάθε εικόνας είναι :

Total time for preparation	4.47 sec
Total time for processing (net)	7.43 sec (estimated 10 iterations for the Hopfield net)

Η προεπεξεργασία (εκπαίδευση δικτύων) πραγματοποιείται μόνο μια φορά.

Κατά τον υπολογισμό του συνολικού χρόνου επεξεργασίας θεωρήσαμε το δίκτυο Hopfield εκτελεί 10 επαναλήψεις (που είναι ο μέσος όρος επαναλήψεων).

Ο χρόνος επεξεργασίας είναι ιδιαίτερα υψηλός (γίνεται υψηλότερος εάν συνυπολογιστεί και ο χρόνος που απαιτείται για το κόψιμο και την τμηματοποίηση της κάθε εικόνας), γεγονός που καθιστά το σύστημα μας σχεδόν απαγορευτικό για εφαρμογές πραγματικού χρόνου. Αυτό είναι και το κύριο μειονέκτημα του.

## 7 Παράρτημα 2

Στο παράρτημα αυτό παρουσιάζονται παραδείγματα υπολογισμού συνολικής μετατόπισης του component σε ολοκληρωμένα κυκλώματα. Η πρώτη στήλη δείχνει το είδος του ολοκληρωμένου, η δεύτερη τις πραγματικές μετατοπίσεις του component και οι υπόλοιπες τις μετατοπίσεις (σε pixels) για κάθε πλευρά. Σημειώνεται εδώ ότι τα νούμερα πάνω και κάτω από τις μετατοπίσεις κάθε ολοκληρωμένου, δείχνουν τις πραγματικές τους μετατοπίσεις (επίσης σε pixels).

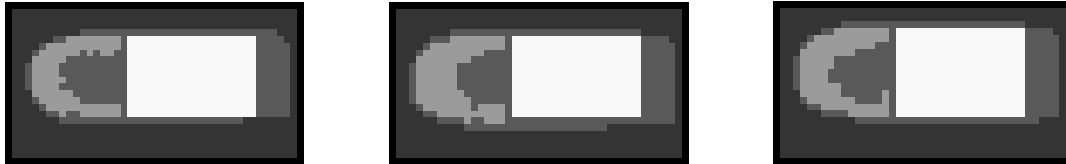
Boardnbr	Board Mount Description	Left Side	Right Side	Bottom Side	Top Side	Horizontal Sides	Vertical Sides
1	Golden board; no shifts	0.1	-2.5	2.5	0.5	-1.2	1.5
h01_1QF		0.033333	-2.166667	2.633333	1.166667	-1.066667	1.900000
h01_2QF		2.600000	3.800000	0.586207	2.866667	3.200000	1.683334
		3	3.5	1.5	3	3.25	2.25
2	20 micron right shift in Y direction	-2	-2.5	2	1.5	-2.25	1.75
h02_1QF		-2.100000	-3.133333	2.266667	1.933333	-2.616667	2.100000
h02_2QF		2.166667	2.533333	2.466667	2.666667	2.350000	2.566667
		2.5	2	2.5	2.5	2.25	2.5
3	60 micron right shift in Y-direction	-4.5	-5	1.5	0	-4.75	0.75
h03_1QF		-4.300000	-5.533333	1.600000	-0.633333	-4.916667	0.483334
h03_2QF		-0.500000	-0.800000	0.133333	1.800000	-0.650000	0.966667
		-0.5	-0.5	-0.1	1	-0.5	0.45
4	100 micron right shift in Y-direction	-5.5	-6	2.5	0.5	-5.75	1.5
h04_1QF		-4.800000	-5.766667	2.866667	0.100000	-5.283334	1.483334
h04_2QF		-1.766667	-2.586207	2.300000	2.333333	-2.176437	2.316667
		-1.5	-1.5	1.5	1.5	-1.5	1.5
5	40 micron right shift in X-direction & 60 micron left shift in Y-direction'	1.1	0.5	0.1	0.1	0.8	0.1
h05_1QF		2.066667	0.566667	0.500000	0.766667	1.316667	0.633334
h05_2QF		6.000000	4.633333	0.233333	-0.400000	5.316667	-0.083334
		7	5	0.5	-1.5	6	-0.5
6	20 micron right shift in X-direction & 20 micron left shift in Y-direction'	0.5	-1.5	-0.5	-0.5	-0.5	-0.5
h06_1QF		0.633333	-1.933333	-0.033333	0.103448	-0.650000	0.035058
h06_2QF		4.100000	4.133333	0.333333	0.133333	4.116667	0.233333
		5	4	0	0	4.5	0
7	120 micron left shift in X-direction & 60 micron right shift in Y-direction'	-3	-6	6	2.5	-4.5	4.25
h07_1QF		-2.366667	-5.366667	5.966667	3.366667	-3.866667	4.666667
h07_2QF		-0.666667	-0.620690	5.933333	5.933333	-0.643679	5.933333
		-0.5	0.1	6	6.5	-0.2	6.25
8	0.2 rotation angle ( 3 pixels shift) & 50 micron paste shift	-0.5	-5	0.5	-3	-2.75	-1.25
h08_1QF		-0.533333	-4.566667	0.366667	-3.266667	-2.550000	-1.45
h08_2QF		3.466667	1.666667	-0.566667	-2.033333	2.566667	-1.3
		3.5	1	-0.5	-2.5	2.25	-1.5
9	0.2 rotation angle ( 3 pixels shift)	1.5	-5.5	4	-2	-2	1
h09_1QF		1.733333	-4.466667	4.266667	-2.066667	-1.366667	1.100000
h09_2QF		5.130435	-0.733333	5.566667	-3.333333	2.198551	1.116667
		10	-0.5	7.5	-3.5	4.75	2
10	0.6 rotation angle ( 9 pixels shift)	3	-9.5	7	-5.5	-3.25	0.75
h10_1QF		2.800000	-5.233333	5.933333	-5.366667	-1.216667	0.283333
h10_2QF		5.666667	-2.700000	5.800000	-2.533333	1.483334	1.633334
		8	-2.5	8	-2.5	2.75	2.75

Όπως φαίνεται και στον παραπάνω πίνακα υπάρχουν περιπτώσεις μεγάλης διαφοράς στην πραγματική και μετρούμενη μετατόπιση και κυρίως αφορούν τα εξής boards:

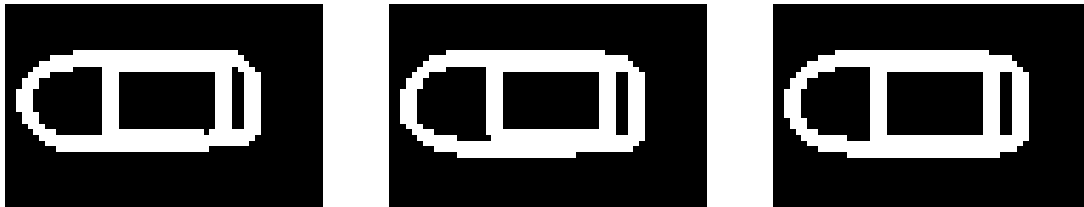
- H01\_2QF Bottom side

Πραγματική μετατόπιση	Μετρούμενη μετατόπιση
1.5	0.586207

Η διάφορα αυτή οφείλεται στο ότι μεγάλο πλήθος εικόνων κατατάσσονται στα 0 pixels. Κάποια παραδείγματα τέτοιων εικόνων φαίνονται παρακάτω:



Οι παραπάνω εικόνες μετά την προεπεξεργασία θα είναι:



Λαμβάνοντας υπόψη και τις μνήμες που χρησιμοποιούμε στο δίκτυο μας (σχήμα 3.10), θα μπορούσαμε να ισχυριστούμε πως το σύστημα μας κατατάσσει στη σωστή κλάση τις παραπάνω εικόνες. Η μεγάλη απόκλιση της μετρούμενης από την πραγματική μετατόπιση είναι απόρροια της ύπαρξης σημαντικού αριθμού τέτοιων εικόνων ανά πλευρά.

Παρόμοιοι είναι και οι λόγοι της μεγάλης διαφοροποίησης της μετατόπισης και στα παρακάτω boards.

- H04\_2QF Right side

Πραγματική μετατόπιση	Μετρούμενη μετατόπιση
-1.5	-2.586207

Η διάφορα αυτή οφείλεται στο ότι σημαντικός αριθμός εικόνων κατατάσσονται στα -3 pixels.

- H05\_2QF Top side

Πραγματική μετατόπιση	Μετρούμενη μετατόπιση
-1.5	-0.4

Η διάφορα αυτή οφείλεται στο ότι σημαντικός αριθμός εικόνων κατατάσσονται στα 0 pixels.

- H07\_2QF Right side

Πραγματική μετατόπιση	Μετρούμενη μετατόπιση
0.1	-0.620690

Η διάφορα αυτή οφείλεται στο ότι σημαντικός αριθμός εικόνων κατατάσσονται στα +1 pixels.

## 8 Βιβλιογραφία

- [1] HIPER Test bed specification, CTB596-98-2229
- [2] Sun Yuan Kung: Digital Neural Network, PTR Prentice Hall Inc, US, 1993
- [3] James P. Coughlin, Robert H. Baran: Neural Computation in Hopfield Networks and Boltzmann Machines, Associated University Press Inc, US, 1995
- [4] Shigeo Abe, Neural Networks and Fuzzy Systems, Kluwer Academic Publishers, US, 1997
- [5] J. J. Hopfield: Neural Network and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Science, US, 1982
- [6] N. Davey, S. P. Hunt: A Comparative Analysis of High Performance Associative Memory Models
- [7] N. Davey, S. P. Hunt: The Capacity and Attractor Basins of Associative Memory Models, Proceedings of the 5<sup>th</sup> International Work Conference on Artificial and Natural Neural Networks, IWANN 99, Vol 1, pp 330-339, 1999
- [8] S. Dierkerich, M. Oppen: Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules, Physical Review Letters 58, 1987
- [9] I. Kanter, H. Sompolinsky: Associative Recall of Memory Without Errors, Physical Review A 35, 1987
- [10] L. F. Abbott: Learning in Neural Network Memories, Network: Comp. Neural Sys., 1990
- [11] W. Krauth, M. Mezard: Learning algorithms with optimal stability for neural networks, J. Phys. 20, 1987

- [12] L. Personnaz, I. Guyon, G. Dreyfous: Collective Computational Properties of Neural Networks : New Learning Mechanisms, Physical Review A(34), 1986
- [13] A. Schultz: Five Variations of Hopfield Associative Network, Journal of Artificial Neural Networks 2, 1995
- [14] A. Storkey, R. Valabregue: Hopfield Learning Rule with High Capacity Storage of Correlated Patterns, Electronic Letters 33(21), 1997
- [15] A. Storkey, R. Valabregue: The basins of attraction of a new Hopfield learning rule, Neural Networks 12, 1999
- [16] R. P. Lipmann: An Introduction to Computing with Neural Nets, IEEE ASSPPM Magazine, 1987
- [17] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck: Machine Vision, McGraw-Hill, New York, 1995
- [18] Gonzalez/Woods: Digital Image Processing, Addison-Wesley Publishing Company, US, 1993