University of Ljubljana
Faculty of Mechanical Engineering

Neural Networks course

Report

“SONN application”.

Student: Marina Gergesova
Mentor: Primož Potočnik, assist. prof.
Date: 16.01.2009
Semester: 2008/2009

Ljubljana 2009

Content

1. Introduction

Self-organization is the spontaneous often seemingly purposeful formation of spatial, temporal, spatio-temporal structures or functions in systems composed of few or many components. In physics, chemistry and biology self-organization occurs in open systems driven away from thermal equilibrium. The process of self-organization can be found in many other fields also, such as economy, sociology, medicine, technology.

It is commonly known that the cortex of the human brain is subdivided in different regions, each responsible for certain functions. The neural cells are organizing themselves in groups, according to incoming information. This incoming information is not only received by a single neural cell, but also influences other cells in its neighborhood. This organization results in some kind of a map, where neural cells with similar functions are arranged close together.

This self-organization process can also be performed by a neural network. Those neural nets are mostly used for classification purposes, because similar input values are represented in certain areas of the net's map.

2. Self-organization in NN. Kohonen networks

The essence of **self-organization** is that system structure often appears without any involvement from outside the system.

Self-organizing networks could be both **supervised** or **unsupervised**, and have 4 additional properties:

- Each weight is representative of a certain input.
- Input patterns are shown to all neurons simultaneously.
- Competitive learning: the neuron with the largest response is chosen.
- A method of reinforcing the competitive learning.

The self-organizing neural networks consist usually of 2 layers:

1) an **input layer** that receives the data
2) a **competitive layer** (or **Kohonen layer**) of neurons that compete with each other to determine to which cluster the given input belongs

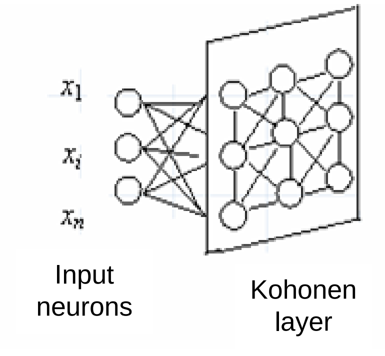


Fig. 2.1
Structure of self-organizing NN

There are quite a few <u>types of self-organizing networks</u>:

1. the Instar - Outstar networks
2. the ART-series
3. the Kohonen networks

Kohonen networks

The term Kohonen network is a slightly misleading one, because the researcher Teuvo Kohonen in fact researched many kinds of network, but only a small number of these are called Kohonen networks. The most well-known among them are:

- **Self-Organizing Map (SOM)** – is a special class of artificial NNs based on competitive unsupervised learning. These are networks whose neurons are allocated into one (ore two) dimensional lattice structure.
- **Learning Vector Quantization (LVQ)** – is supervised NN that attempts to define “decision boundaries” in the input space; each decision could, e.g., be a correct class label.

3. Real-world applications of Kohonen networks

Reduction of multi-dimensional input data

The objective of a Kohonen network is to map input vectors (patterns) of arbitrary dimension N onto a discrete map with 1 or 2 dimensions. Patterns close to one another in the input space should be close to one another in the map: they should be topologically ordered. A Kohonen network is composed of a grid of output units and N input units.

Example 1. Every object can be characterized with set of different parameters. For the simplicity let’s take only three of them. For example, industrial plant can be described with capitalization, volume of production and profit. Let’s normalize these parameters to plot them in a unit cube. Each point corresponds to one plant.

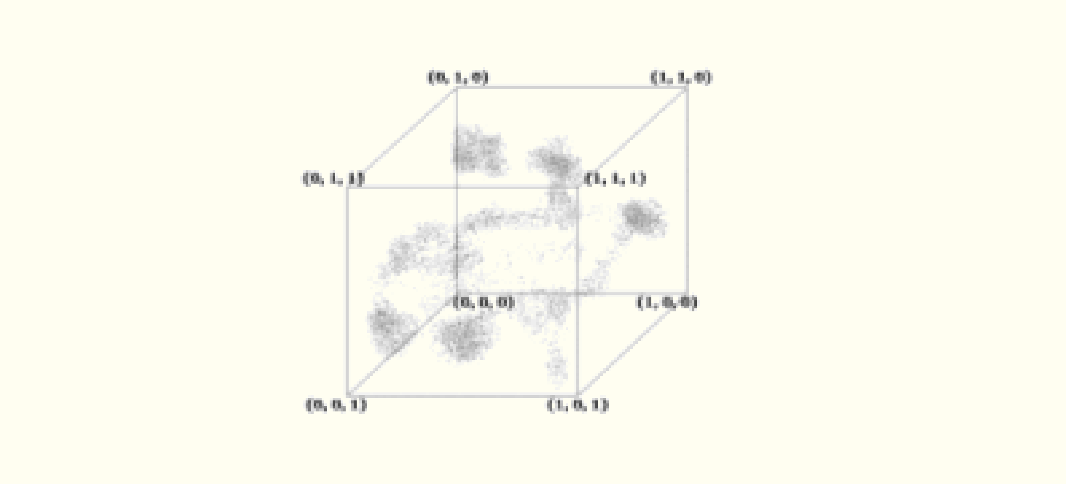


Fig. 3.1
Objects arrangement in a 3D-space

On the Fig. 3.1 one can easily determine the objects with similar properties. It is evident that this could be done only in the case of low dimensionality (n<= 3, n – dimension of space).

The way how to present multi-dimensional data in 2D-diagram is SOM – Self-Organizing Map.

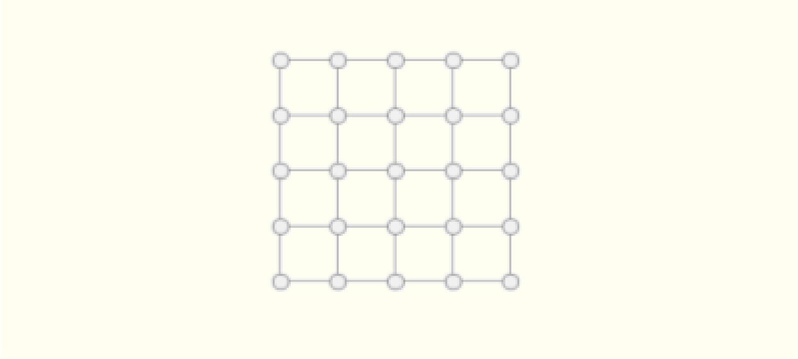


Fig. 3.2
Kohonen map

The nodes of such map coincide with location of main objects accumulations in the initial space. Also the similar objects correspond to the neighbor nodes in lattice.

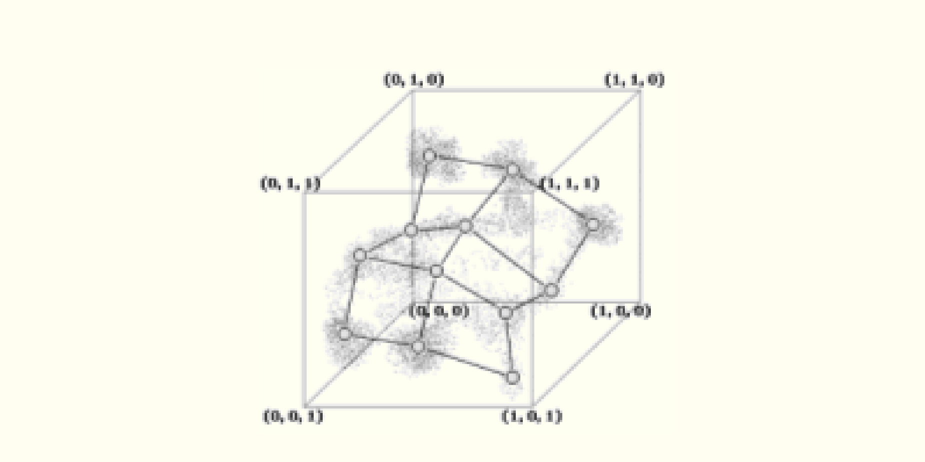


Fig. 3.3
Initial space after mapping

<u>Example 2</u>. Let's visualize colors on a two-dimensional plane. Color is normally 3-dimensional, represented as red, green and blue components (Fig. 3.4).

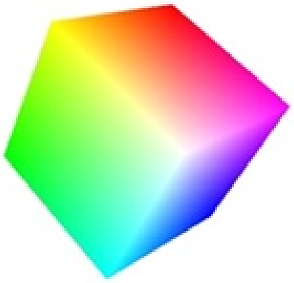


Fig. 3.4
Color

Representing this on two dimensions is harder to do, but is possible with teaching a Kohonen network to self-organize across the RGB space. There are two approaches, either by presenting a set number of colors repeatedly or by presenting a randomly selected color.

Pictures below show the initial color spread and the colors after the training algorithm. The circles in the images show the exact neurons that fire for each color the network was trained on.

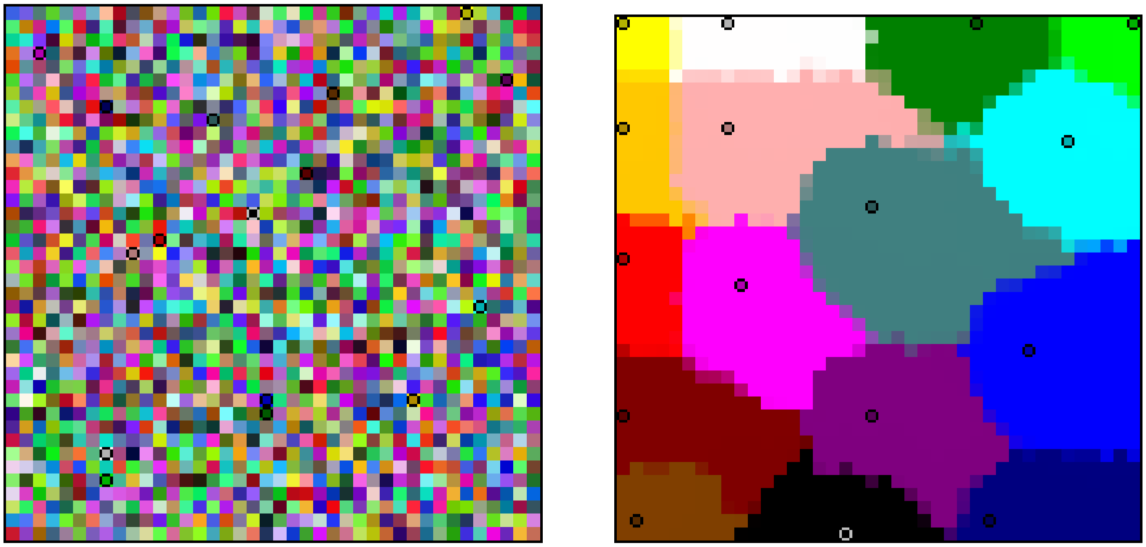


Fig. 3.5
1st approach – the network was trained on 16 colors

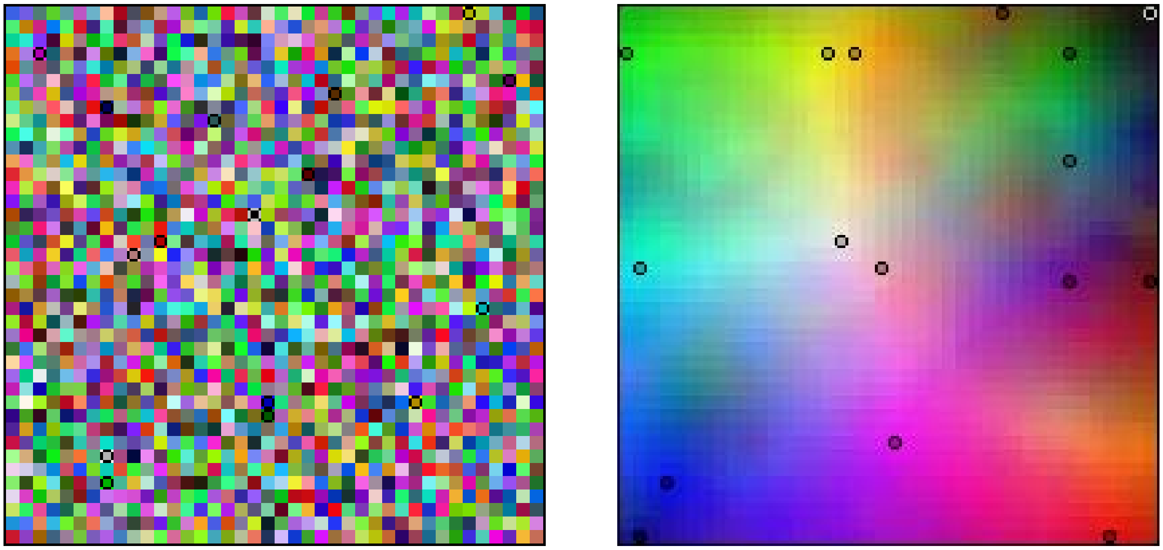


Fig. 3.6
2nd approach - random selection of a color for each training the network

Image analysis

The author of application (James Matthews) chose pictures of some cars, since many automotive images are quite similar in their features and presentation. After downloading about 200 images (and turning them into thumbnails of about 200x150 pixels each), he used the Generation5 JDK to create a Kohonen network with 80-feature input vector from each image.

Each image has its histogram mean and standard deviations (for each color component) calculated and added to the feature vector. The image is then run through a Sobel edge detector, and has its histogram calculated. The mean and standard deviation of this histogram is also added, making a total of 8 features. The image is then broken up into 9 sub regions, which has the same features calculated and added to the feature vector, making a total of 80 features.

The Kohonen network is then trained on these 200-odd training vectors. He took the size of the network relatively small (about 6x6) to force multiple images to be mapped on to the same node, helping to find similar images.

Visualizing the network is harder, since image data itself isn't stored (features of images are). Therefore, to visualize the image, the program runs each training vector through the Kohonen network to see which neuron fires. The associated image is then rendered in that position. Remember though that the network can be deliberately "constricted" to force similar images to fire the same neuron, so the program renders up to four images per neuron.

Below is example output from the program. The central image is the output from the program, with certain areas blown up:

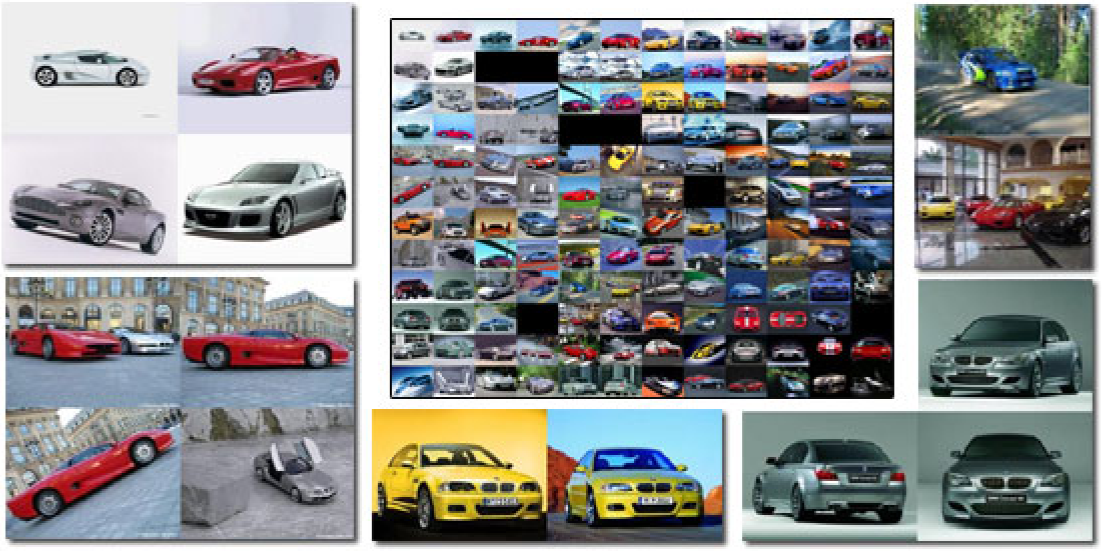


Fig. 3.7
Image classification

Anti-clockwise from the top-left: images with a strong white background group together, similar images of a red Jaguar XJ220 group together (strangely, along with an image of an SLR), these two pictures of a yellow BMW M3 are from different image sets but group together nicely nonetheless, these three similar images of an M6 despite being from different angles, and finally not all associations seem immediately obvious (last group).

While the results aren't perfect, one can see how the network can find easily arrange itself across highly multidimensional data.

Classifying New Images

While grouping similar images is interesting, it is even more interesting to see how the network classifies new images. The example program (for image analysis) also has a method to take a new image and find any similar images using the network. It does this by creating a feature vector for the new image, presenting it to the network, then looping through the training examples to find any images that fire the same neuron. For example, given this new image of a black BMW M6:

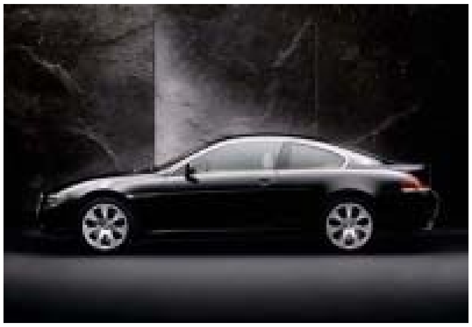


Fig. 3.8
New image

The network finds 9 similar images, 8 of which are shown below. You can see how the network is picking up on the strong black background present in the images:

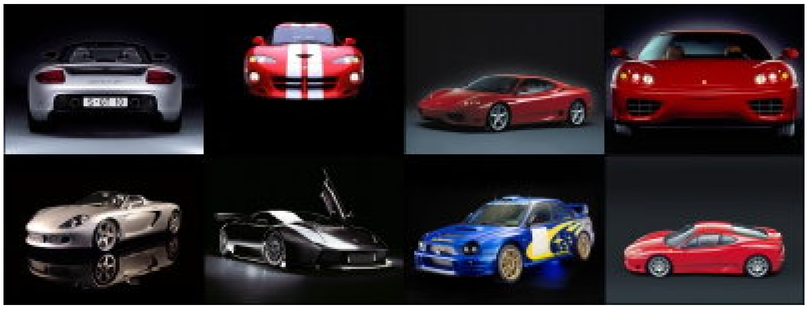
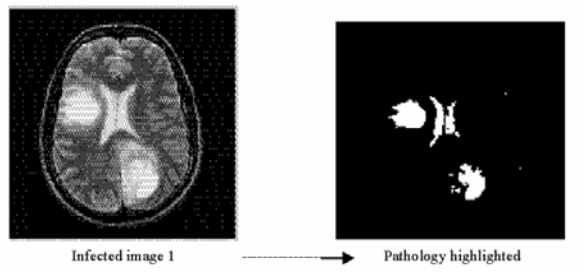


Fig. 3.9
Group of images picked up by network

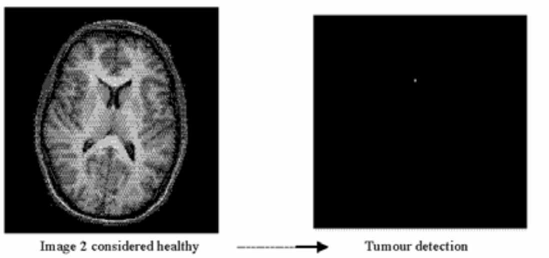
Classification

The cerebral IMR (Image by Magnetic Resonance) is an imagery technique used to follow-up the evolution of various brain pathologies. The detection and the characterization of the cerebral structures and lesions are significant diagnosis elements.

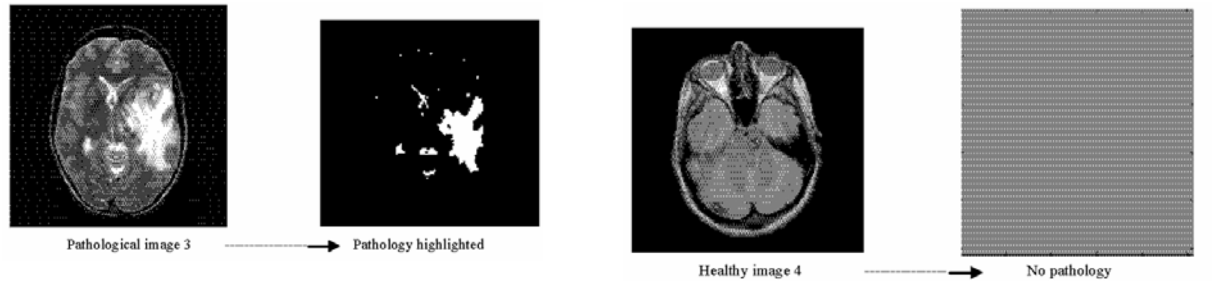
For classification of such images was used combination of SOM and LVQ. The authors investigated brain IMR to know if the brain is healthy or infected, in order to highlight pathology. They partitioned the image into 2x2 size blocks to obtain vectors with 4 elements. Subsequently, by the decision, they assign to each vector a class, i.e. vector takes the class of the nearest neuron to him. They binaries each pixel so that it takes value 0 if it belongs to a vector of healthy class (class 1) and a value 1 if it belongs to a vector of pathological class (class 2). Then they rebuild the image and see whether it is infected or not. The test was done on four images demonstrated on Fig. 3.10.



(1)



(2)



(3) (4)

Fig. 3.10
Brain IMR and classification result

Pathologies a well located on images (1) and (3). Image (4) being healthy did not reveal any pathology. Image (2) represents visually the case of a healthy image, but in reality the tumour is evolving.

4. Problem statement and results

The task was to create 1D and 2D SOM for different types of input data and to solve classification problem with LVQ.

<u>1. Uniform distribution in square</u>

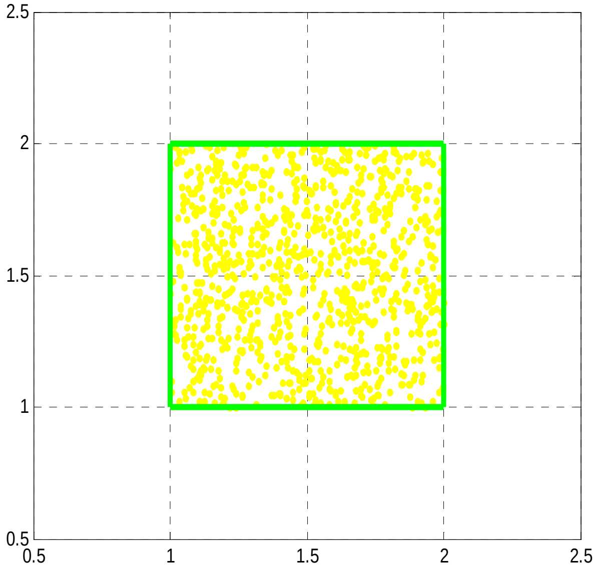


Fig. 4.1.1
1000 input points

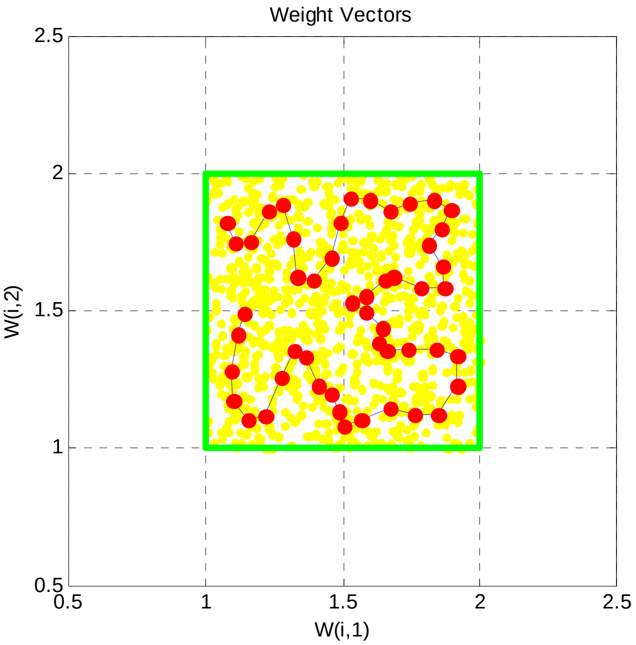


Fig. 4.1.2
Net - 50 neurons, 100 epochs, "grid top"

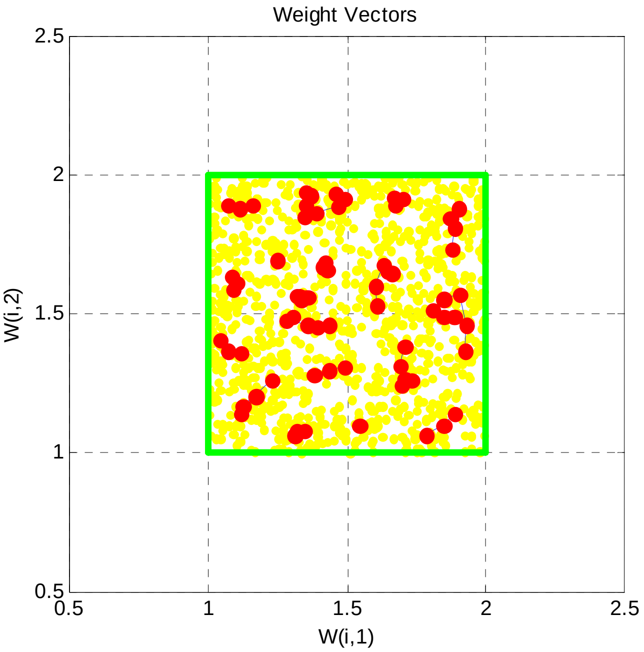


Fig. 4.1.3
Net -100 neurons, 100 epochs, "rand top"

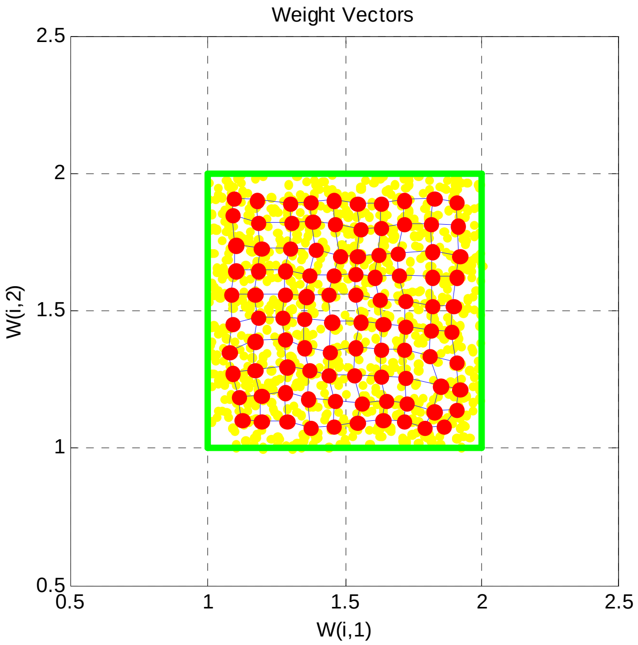


Fig. 4.1.4
Net - 10x10 neurons, 100 epochs, "grid top"

2. Uniform distribution in circle (uniform on radius)

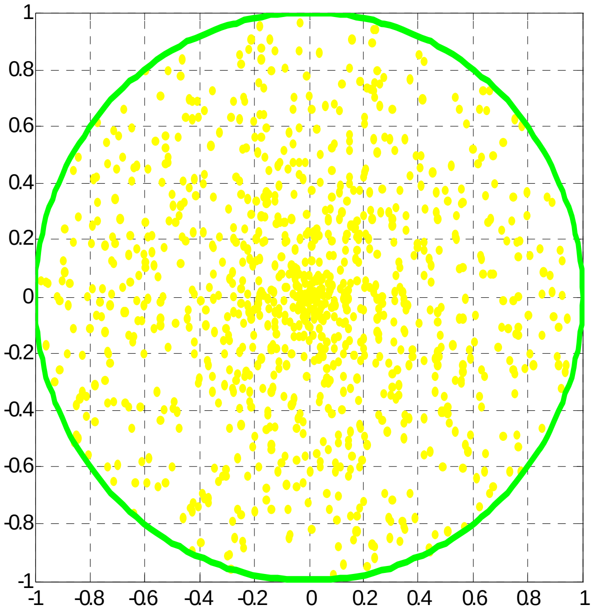


Fig. 4.2.1
1000 input points

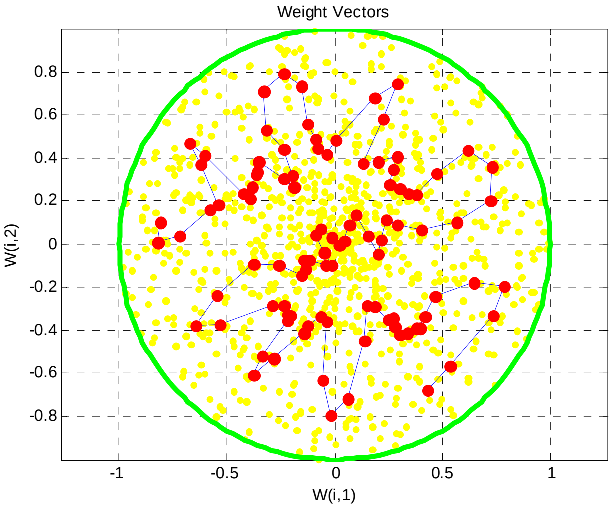


Fig. 4.2.2
Net – 100 neurons, 100 epochs, "grid top"

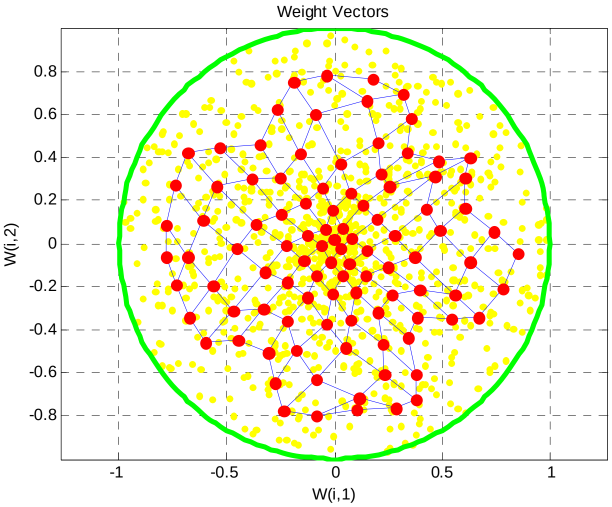


Fig. 4.2.3
Net – 10x10 neurons, 100 epochs, "grid top"

3. Uniform distribution in a circle

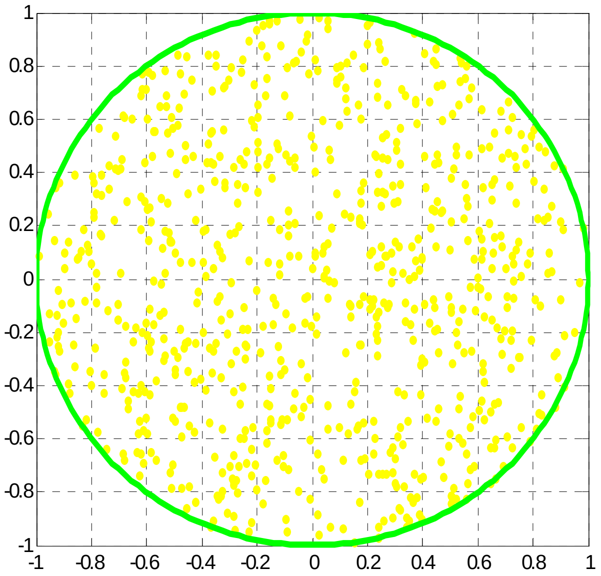


Fig. 4.3.1
1000 input points

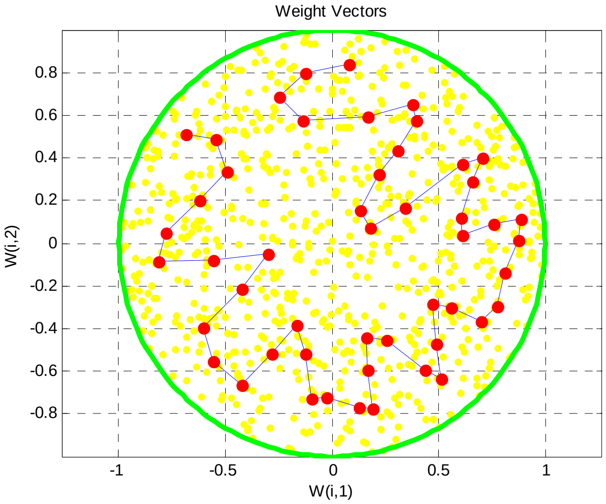


Fig. 4.3.2
Net - 50 neurons, 100 epochs, "grid top"

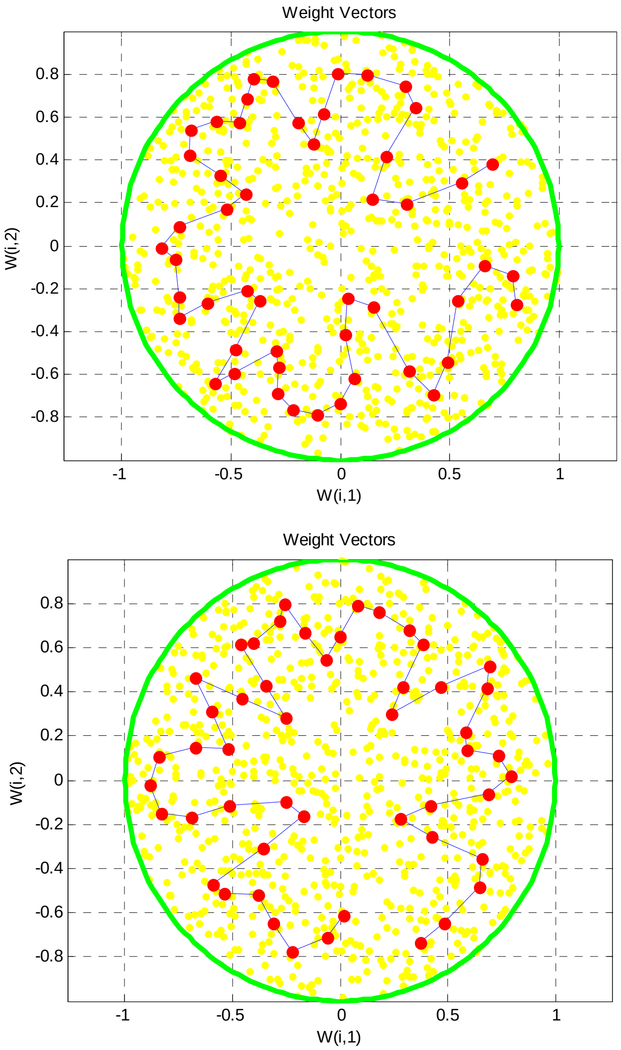


Fig. 4.3.3 a, b

Net - 50 neurons, 100 epochs, “grid top”, a – “dist”, b – “box dist”

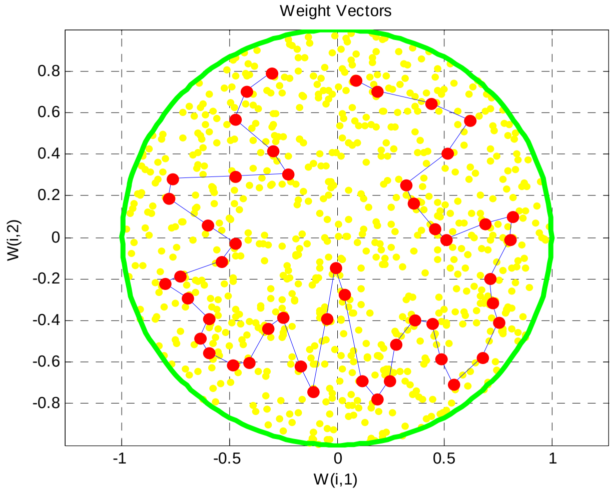


Fig. 4.3.4

Net - 50 neurons, 500 epochs, “grid top”

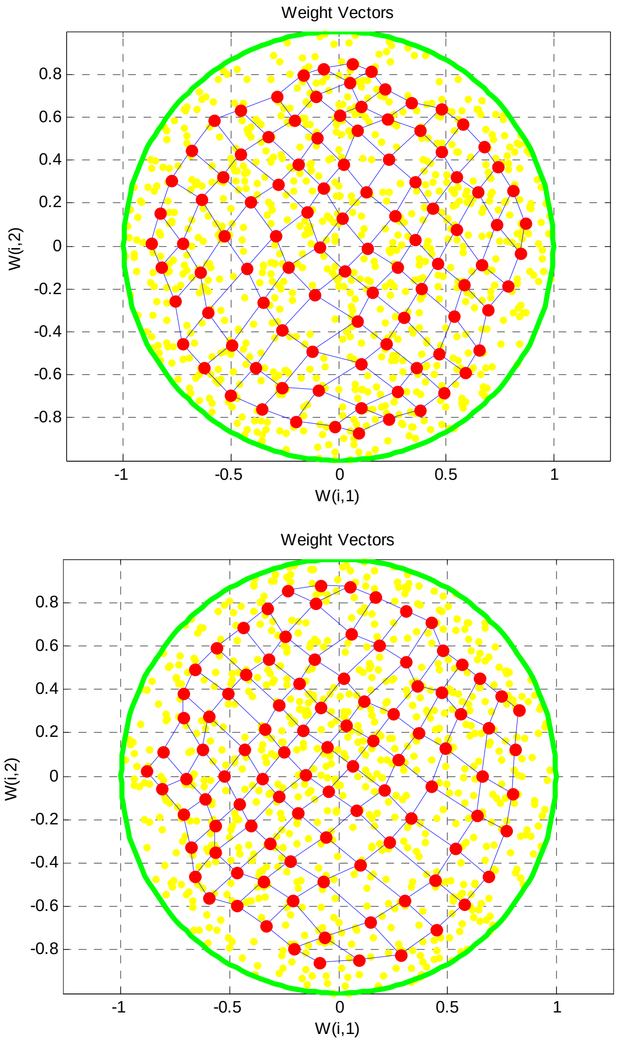


Fig. 4.3.5 a, b
Net - 10x10 neurons, 100 epochs, “grid top”, a – “linkdist”, b – “dist”

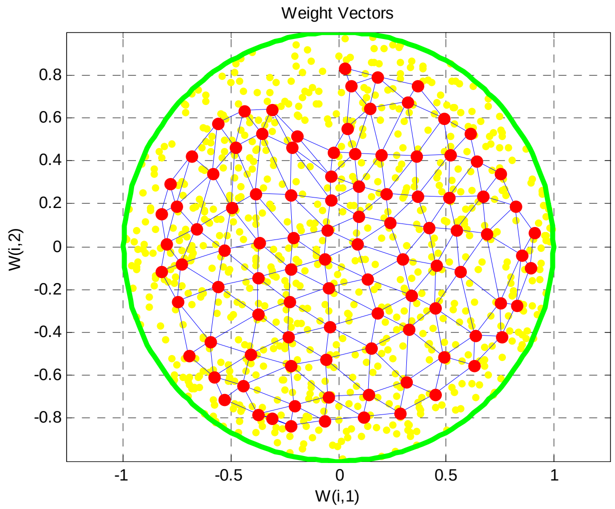


Fig. 4.3.6
Net - 10x10 neurons, 100 epochs, default

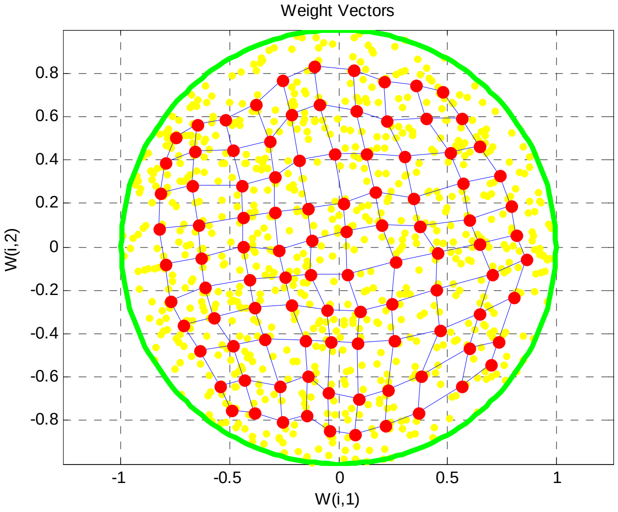


Fig. 4.3.7
Net - 10x10 neurons, 500 epochs, "grid top"

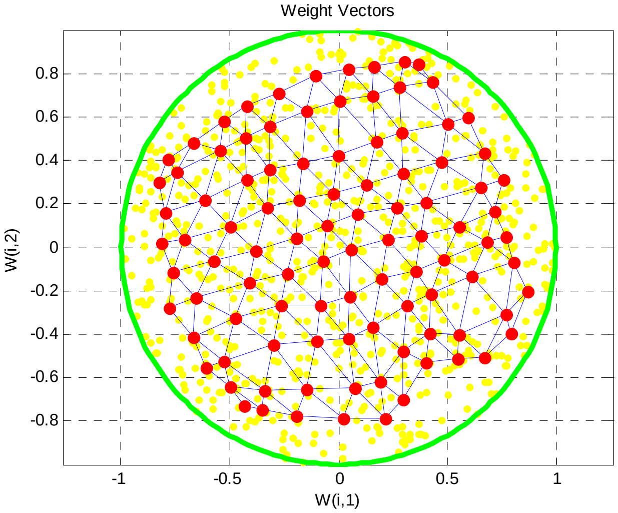


Fig. 4.3.8
Net - 10x10 neurons, 500 epochs, default

4. Normal distribution in square

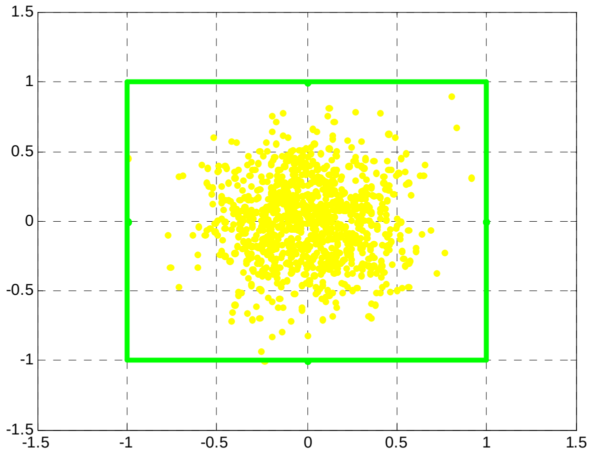


Fig. 4.4.1
1000 input points

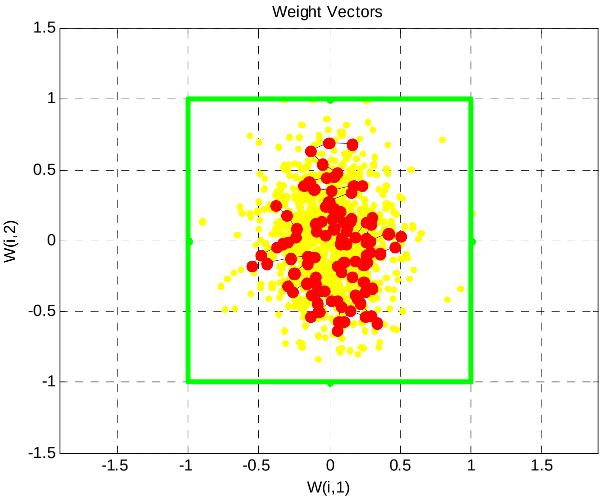


Fig. 4.4.2
Net – 100 neurons, 100 epochs, default

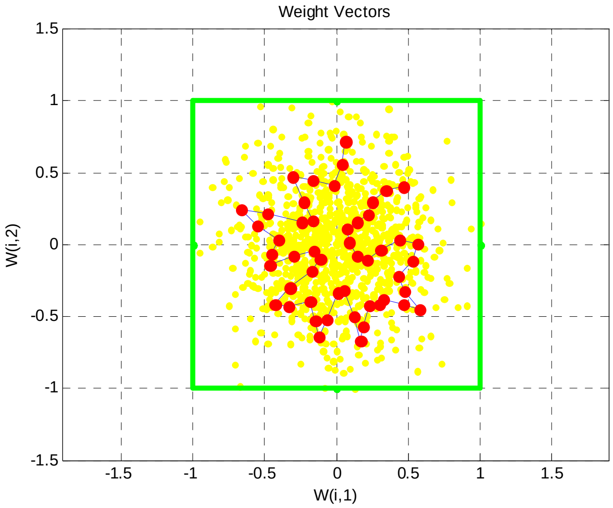


Fig. 4.4.3
Net – 50 neurons, 200 epochs, "grid top"

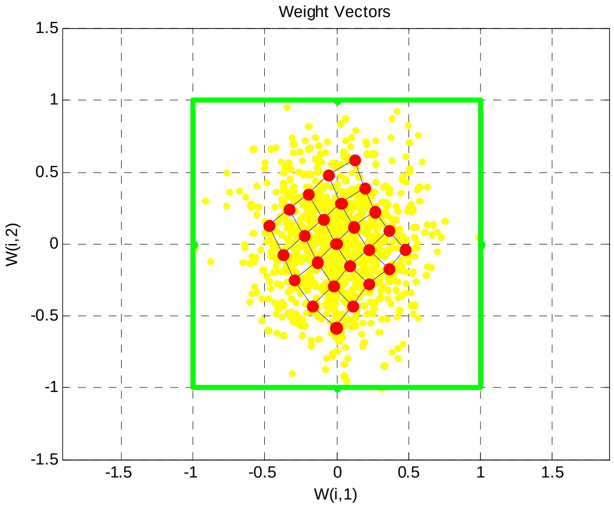


Fig. 4.4.4
Net – 50 neurons, 100 epochs, “grid top”

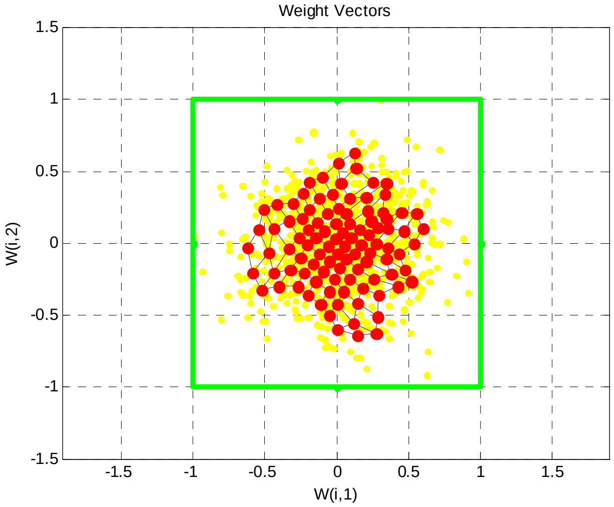


Fig. 4.4.5
Net – 10x10 neurons, 100 epochs, "grid top"

5. Complex distribution - uniform distribution in two areas divided by hyperbole in square

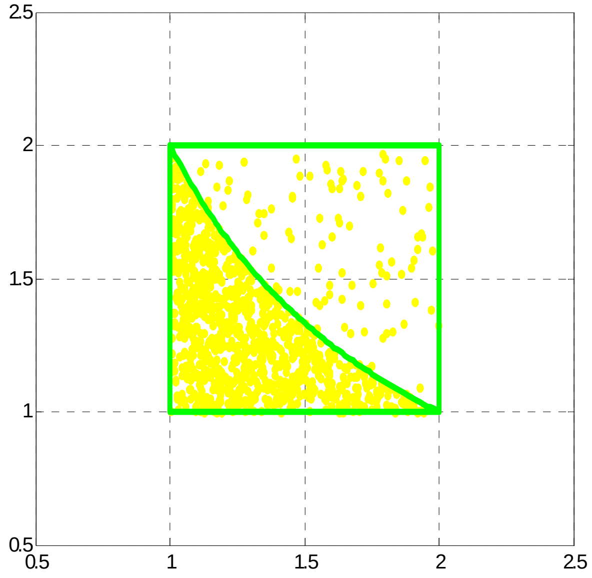


Fig. 4.5.1
1000 + 100 input points

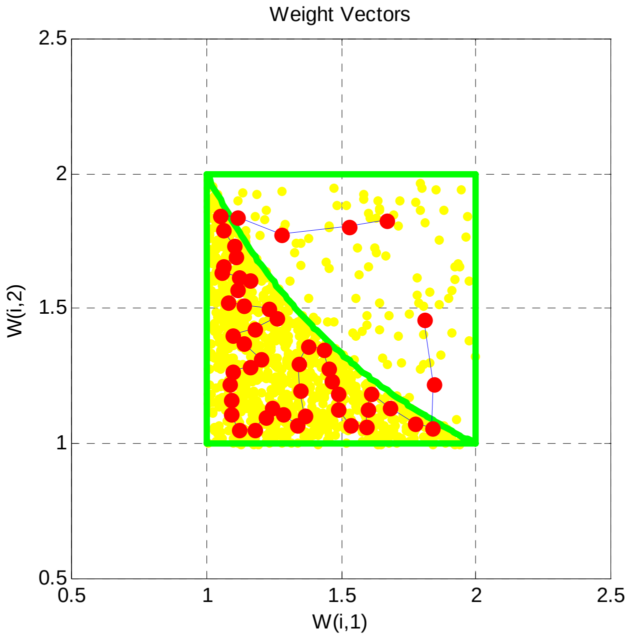


Fig. 4.5.2
Net - 50 neurons, 100 epochs, "grid top"

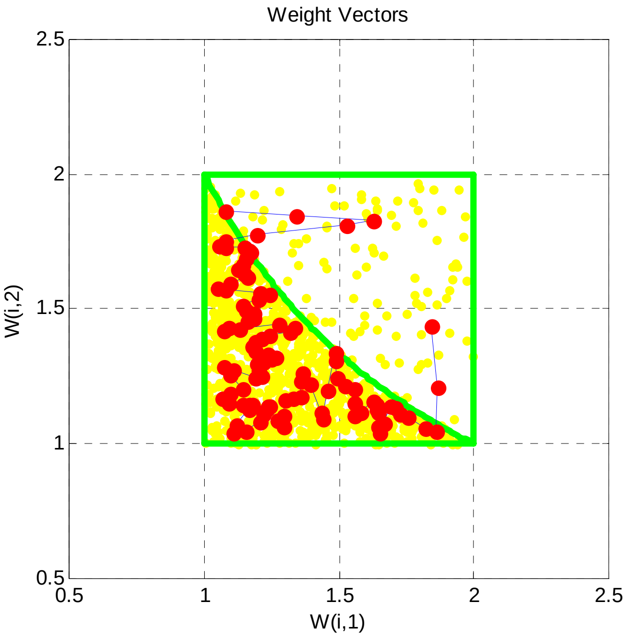


Fig. 4.5.3
Net - 100 neurons, 100 epochs, "grid top"

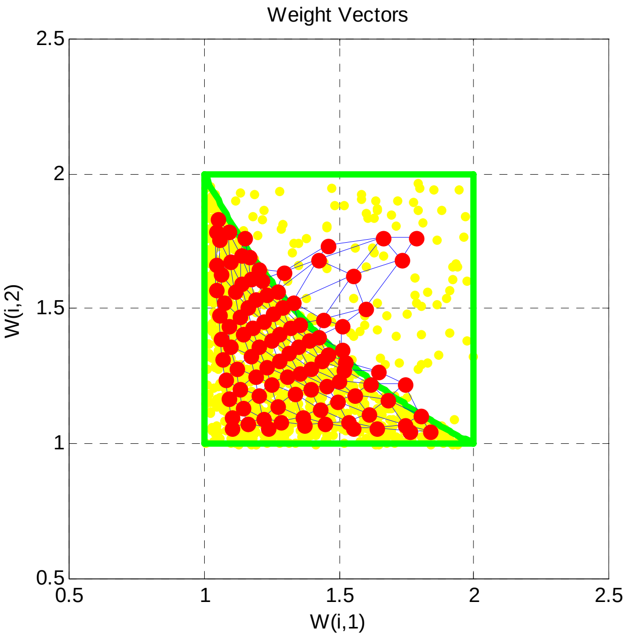


Fig. 4.5.4
Net - 10x10 neurons, 100 epochs, default

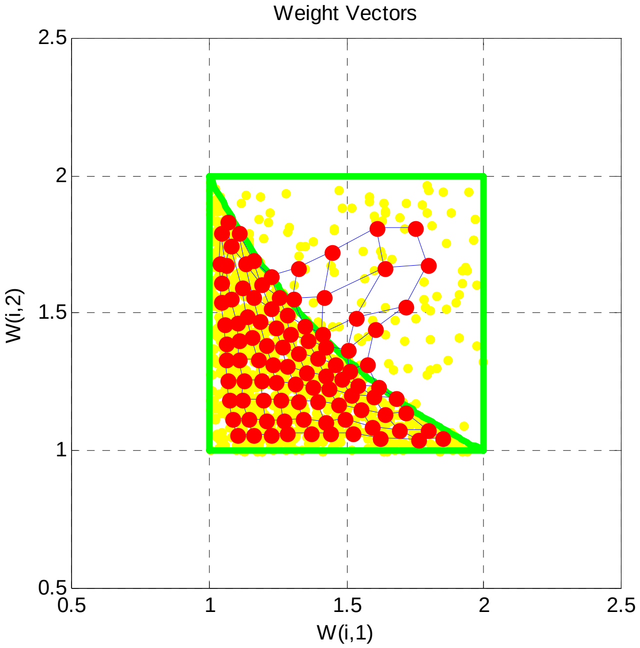
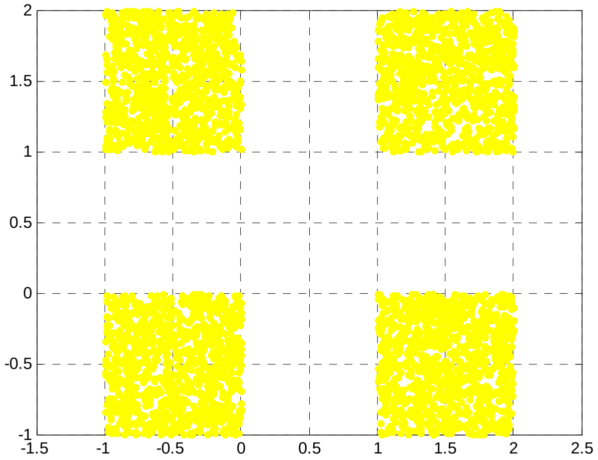


Fig. 4.5.5
Net - 10x10 neurons, 100 epochs, "grid top"

6. Four classes - uniform distribution



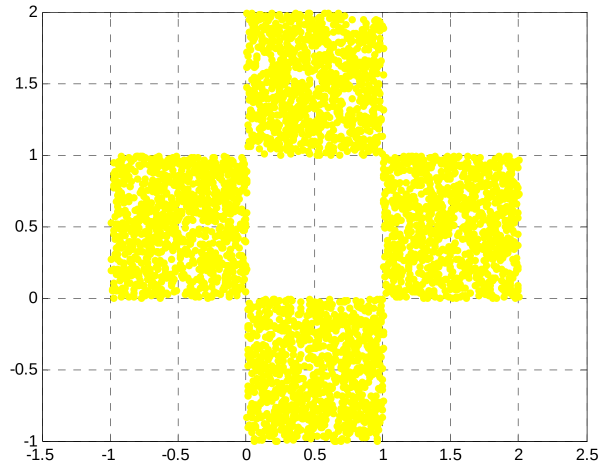


Fig. 4.6.1
4x1000 input points

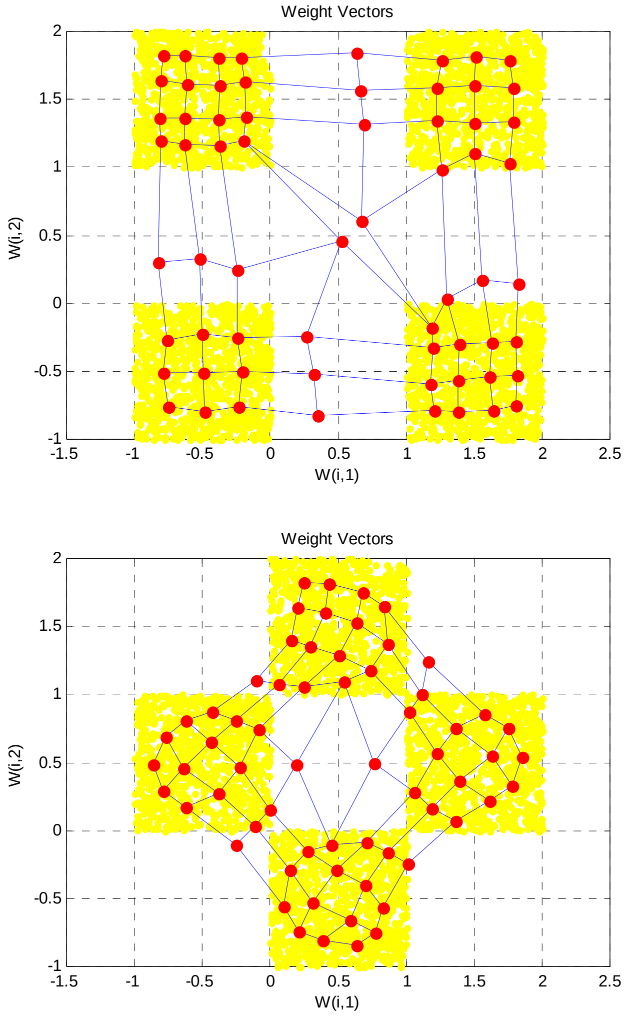


Fig. 4.6.2
Net - 8x8 neurons, 100 epochs, "grid top"

7. Classification with LVQ

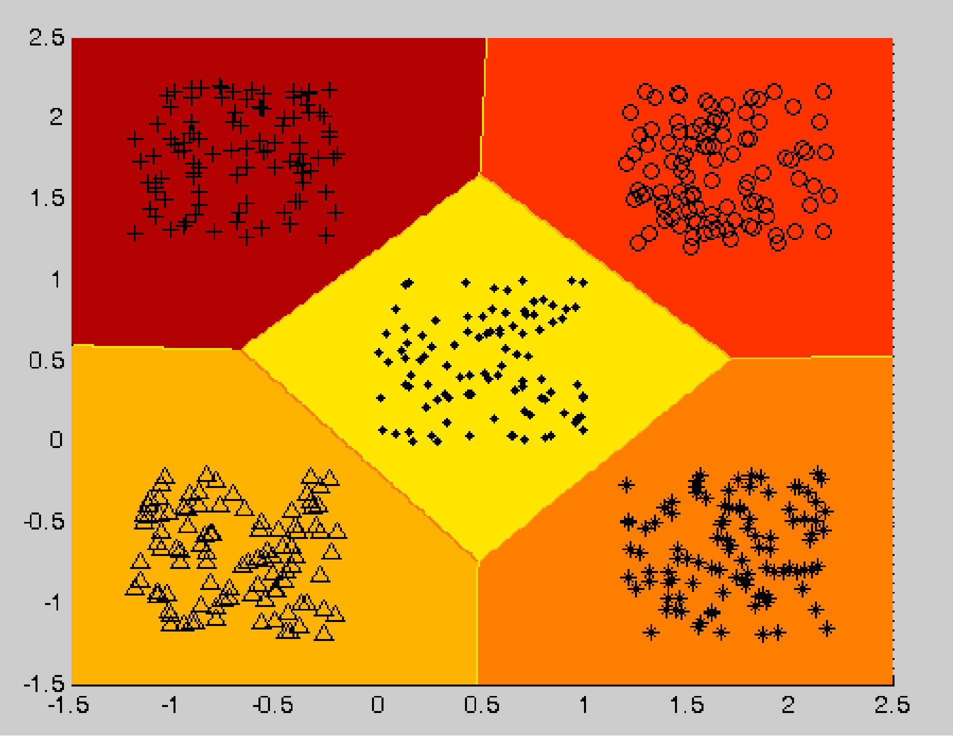


Fig. 4.7.1
5 classes, 100 points in each class, uniform distribution

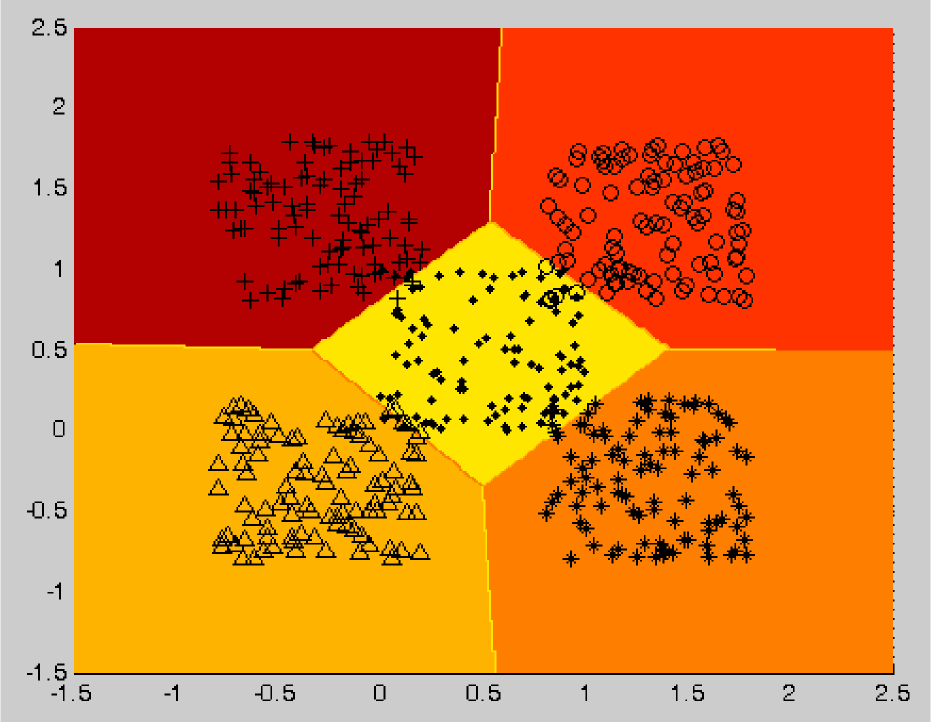


Fig. 4.7.2
5 classes with intersection, 100 points in each class, uniform distribution

5. Conclusions

- Kohonen networks have a huge field of application – in scientific, financial, medical and others areas. They are ideal for finding relationships among complex sets of data and for reducing the dimensionality of data.
- Kohonen networks give quite good results even with a small amount of neurons and iterations.
- But! They tend to take a long time to be built.

6. Literature

1. www.generation5.org
2. www.webstarstudio.com (website on Russian language)
3. Z. Chalabi, N. Berrached, N. Kharchouche, Yu. Ghellemallah, M. Mansour, H. Mouhadjer *Classification of the medical images by the Kohonen networks SOM and LVQ* // Journal of applied sciences. – 2008. – Vol. 8, No. 7. – pp. 1149–1158.
4. Ben Krose, Patrick van der Smagt *An introduction to Neural Networks. - 1996*