Module Human Psychophysics: Homework 1

Kohonen network training and classification

Sepideh Hadadi, MSCV 2016, Le Creusot, France

1. Introduction:

A self-organizing map (SOM) or better saying self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning approach to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to do dimensionality reduction. This way of dimensionality reduction makes SOMs useful for visualizing low-dimensional views of high-dimensional data. This artificial neural network introduced by the Finnish professor Teuvo Kohonen in the 1980s, that is the reason why it is sometimes called a Kohonen map or network. The Kohonen net is a computationally convenient abstraction building on work on biologically neural models from the 1970s and morphogenesis models dating back to Alan Turing in the 1950s.[1]

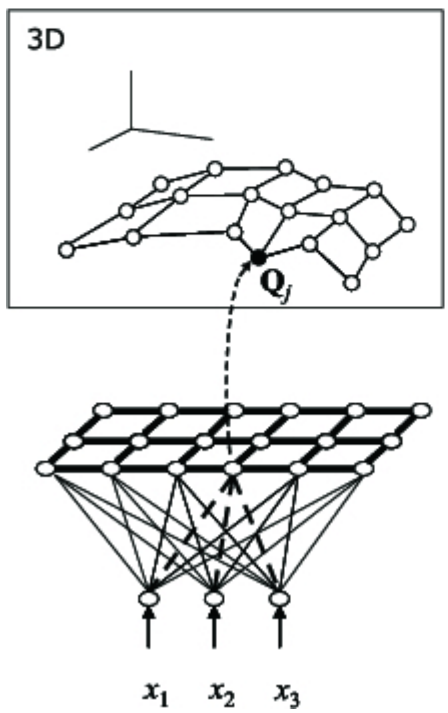

Fig.1 example of Kohonen network [2]

Like most artificial neural networks such feed forward, perceptron and many others, SOMs operate in two steps: training and mapping. "Training" builds the map and calculates the network weighting matrix using input example, while "mapping" automatically classifies a given input vector. A self-organizing map consists of components called nodes or neurons, similar to other ANNs. Associated with each node there are a weight vector of the same dimension as the input data vectors, and a position in the map space. There are three topologies for Kohonen network, i.e., grid-top, hex-top and rand-top as shown in Fig.2.a-c.

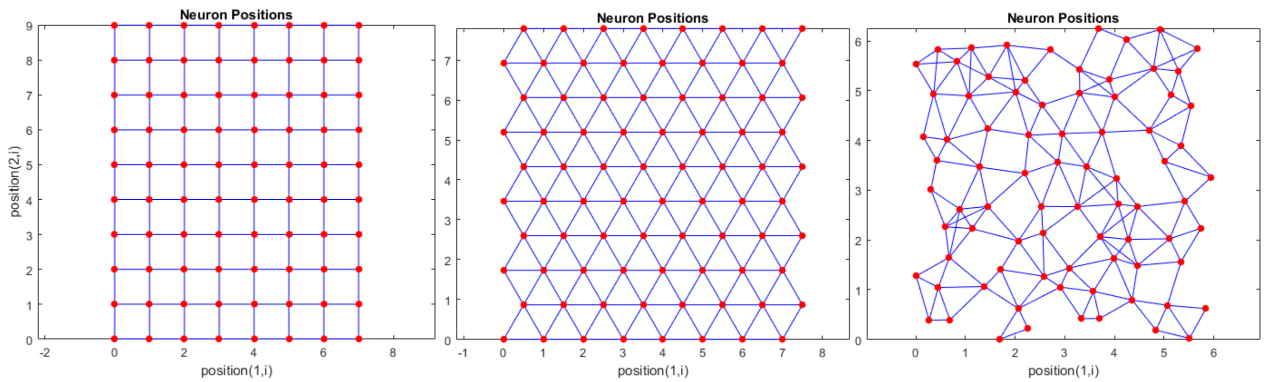

Fig.2.a) the neurons in the grid-top topology do indeed lay on a grid b) in hex-top configuration the positions of the neurones in a hexagonal arrangement c) the rand-top topologie assignes a random location to the neurone position [3]

The procedure for placing a vector from data space onto the map is to find the node with the closest (smallest distance metric) weight vector to the data space vector. Useful extensions include using toroidal grids where opposite edges are connected and using large numbers of nodes. It is very common to use the U-Matrix [4]. The U-Matrix value of a particular node by definition is the average distance between the node's weight vector and that of its closest neighbours [5]. In a square grid, for instance, we might consider the closest 4 or 8 nodes (the Von Neumann and Moore neighbourhoods, respectively), or six nodes in a hexagonal grid.

2. Implementation part I:

1) Construct a Kohonen network in order to carry out the classification of the vectors
(1 1 0 0)
(1 0 0 0)
(0 0 0 1)
(0 0 1 1)
Construct a network that is flexible in terms of the size of the input vector. This will permit you to easily utilize the patient and healthy subject data.

A 4x4 Kohonen network were established to train above data. The initial topology of the network and the final topology of the network are shown in Fig.3, Fig.4 respectively.

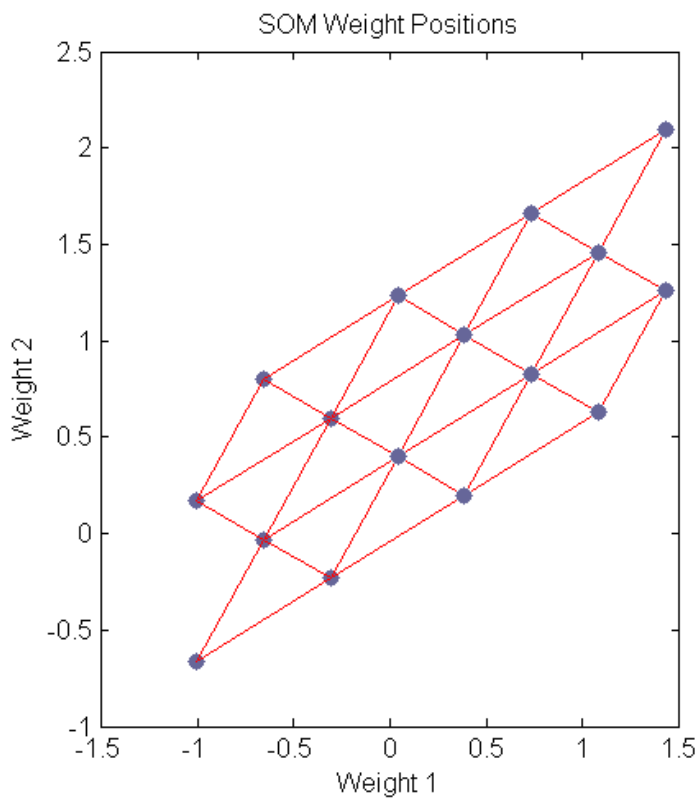

Fig.3 initial sate of the map topology

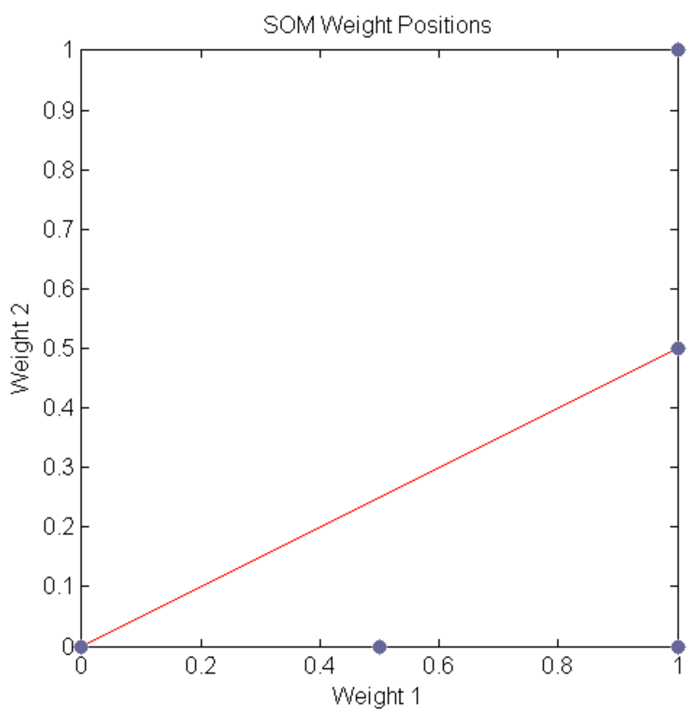

Fig.4 final topology of the network

Pay attention to the adjustment in the learning rate. You may have to find the value that allows a convergence (i.e. synaptic weights that converge).

The vectors (1 1 0 0) and (1 0 0 0) should fall in one class I while the remaining two vectors should fall in class II.
2) Once the training is completed carry out a test with the vectors
(0, 0, 0, 0.9)
(0, 0, 0.8, 0.9)
(0.7, 0, 0, 0)
(0.7, 0.9, 0, 0)
As you might expect, the vectors (0, 0, 0, 0.9) and (0, 0, 0.8, 0.9) should fall in class II while the vectors (0.7, 0, 0, 0) and (0.7, 0.9, 0, 0) should fall in class I.
The above vectors were classified and the following results were achieved.

```
Command Window

Y1 =

     0     0     1     1
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     1     1     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

Fig.5 the result of the vector classification

Note: All the above data and code are located in "homework_part1" folder. Besides, in this implementation, training function was adapted from MATLAB NNTOOL toolbox.

3. Implementation Part II: Training and test of the classifier with real data:

A new code was developed for training of the classifier using "Control" and "Patient" data set. In fact, these two data sets were shown to the classifier and the weights were calculated to have a classifier with two output "Class I, Label =0" (Control) and "Class II, Label=1" (Patient). In fact, If a set of data is recorded from control subject and the second data is recorded from a patient subject. The classifier must be able to say which one is patient and which one is control subject.

Four sets of unknown data have given to the classifier as an input (assigned to each person by professor: for me it was "test_sepideh.text") and results as shown in the second raw of table I was achieved. A second set of data were received along the homework named “solene.tex” (renamed to “test_data.txt” in the repository) that data also were inputted to the classifier and the result is presented in the third raw of table I. (these two test can be re-examined in the code by adjusting “verif” parameter in the “Kohonen_network.m” file).

Table I: result of classification for unknown data

subject	Subject 1	Subject 2	Subject 3	Subject 4
Classification result(Sepideh)	Control	Control	Patient	Patient
Classification result(Solene)	Patient	Control	Patient	Control

Note: this implementation and the rest data exist in the second folder named “homework part 2”.

4. Complementary explanation (optional)

Self-organizing maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as back-propagation with gradient descent), and in the sense that self-organizing map uses a neighbourhood function to preserve the topological properties of the input space. Besides, as shown in part 1 of the homework, self-organizing maps with a small number of nodes behave in a way that is similar to K-means; larger self-organizing maps rearrange data in a way that is fundamentally topological in character (part e of the homework).

5. Reference:

[1] https://en.wikipedia.org/wiki/Self-organizing_map

[2] Hoffmann, Miklós. "Numerical control of Kohonen neural network for scattered data approximation." Numerical Algorithms 39.1 (2005): 175-186.

[3] https://uk.mathworks.com/help/nnet/ug/cluster-with-self-organizing-map-neural-network.html

[4] Ultsch, Alfred; Siemon, H. Peter (1990). "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis". In Widrow, Bernard; Angeniol, Bernard. Proceedings of the International Neural Network Conference (INNC-90), Paris, France, July 9–13, 1990. 1. Dordrecht, Netherlands: Kluwer. pp. 305–308. ISBN 978-0-7923-0831-7.

[5] Ultsch, Alfred (2003); U*-Matrix: A tool to visualize clusters in high dimensional data, Department of Computer Science, University of Marburg, Technical Report Nr. 36:1-12