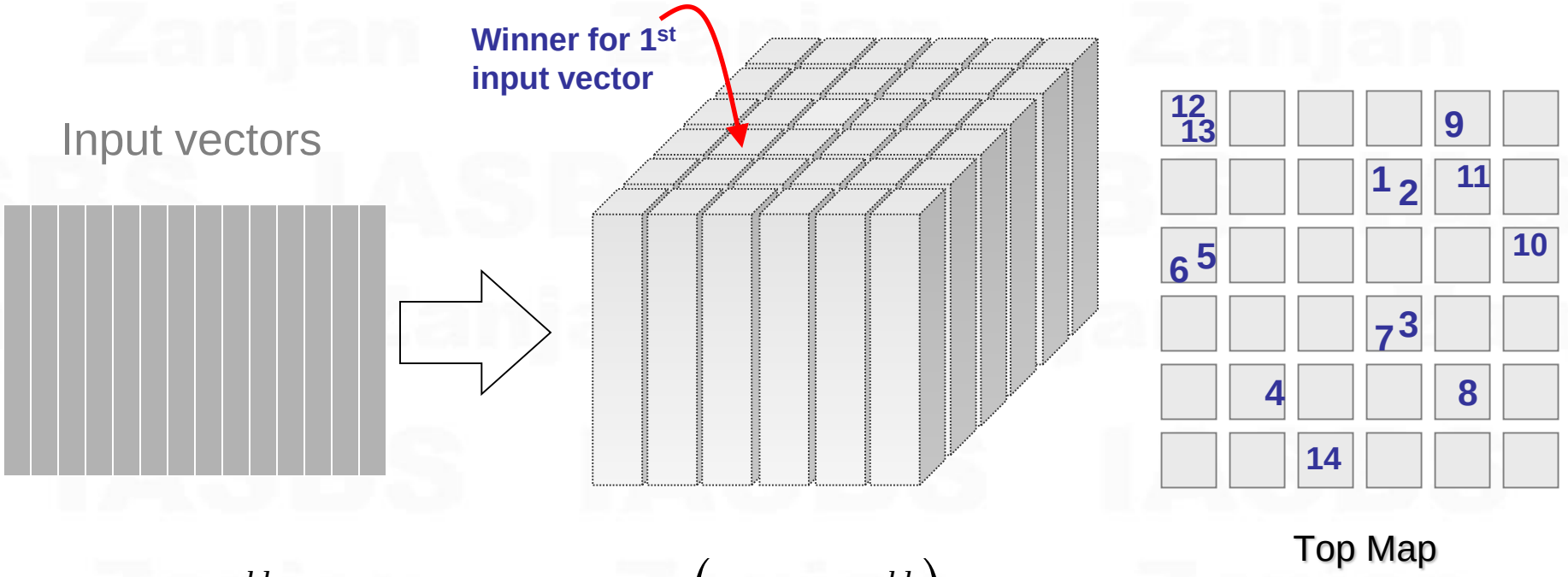
Institute for Advanced Studies in Basic Sciences – Zanjan
Department of Computer Science and Information Technology
By: Mahdi Vasighi

Artificial Neural Networks

Part 13

Self Organizing Maps



$$w_{ji}^{new} = w_{ji}^{old} + \eta(t)\ d(c-j,t)\ \left(X_{Si} - w_{ji}^{old}\right)$$

Self Organizing Maps

Inside feature space

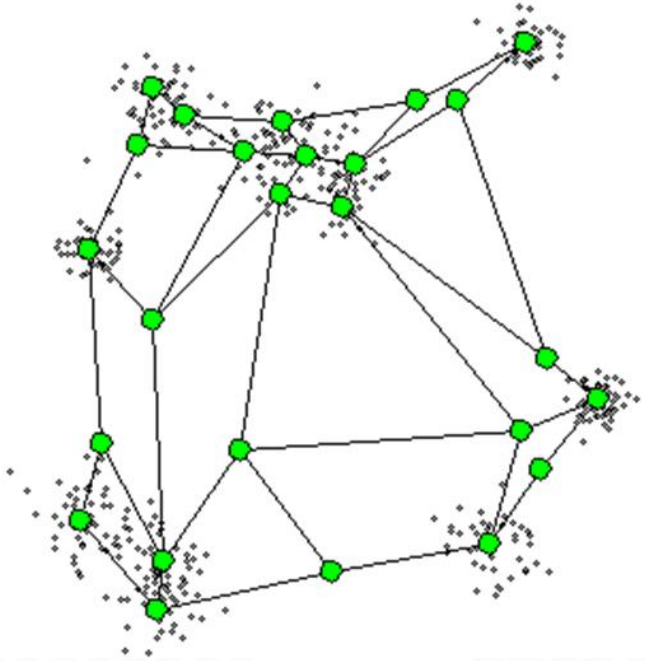
The aim of SOMs is to learn a feature map from the spatially continuous input space to the low dimensional spatially discrete output space, which is formed by arranging the computational neurons into a grid. The feature map has some important properties:

Approximation of the Input Space

The feature map represented by the set of weight vectors in the output space, provides a good approximation to the input space.

the goodness of the approximation is given by the total squared distance which we wish to minimize.

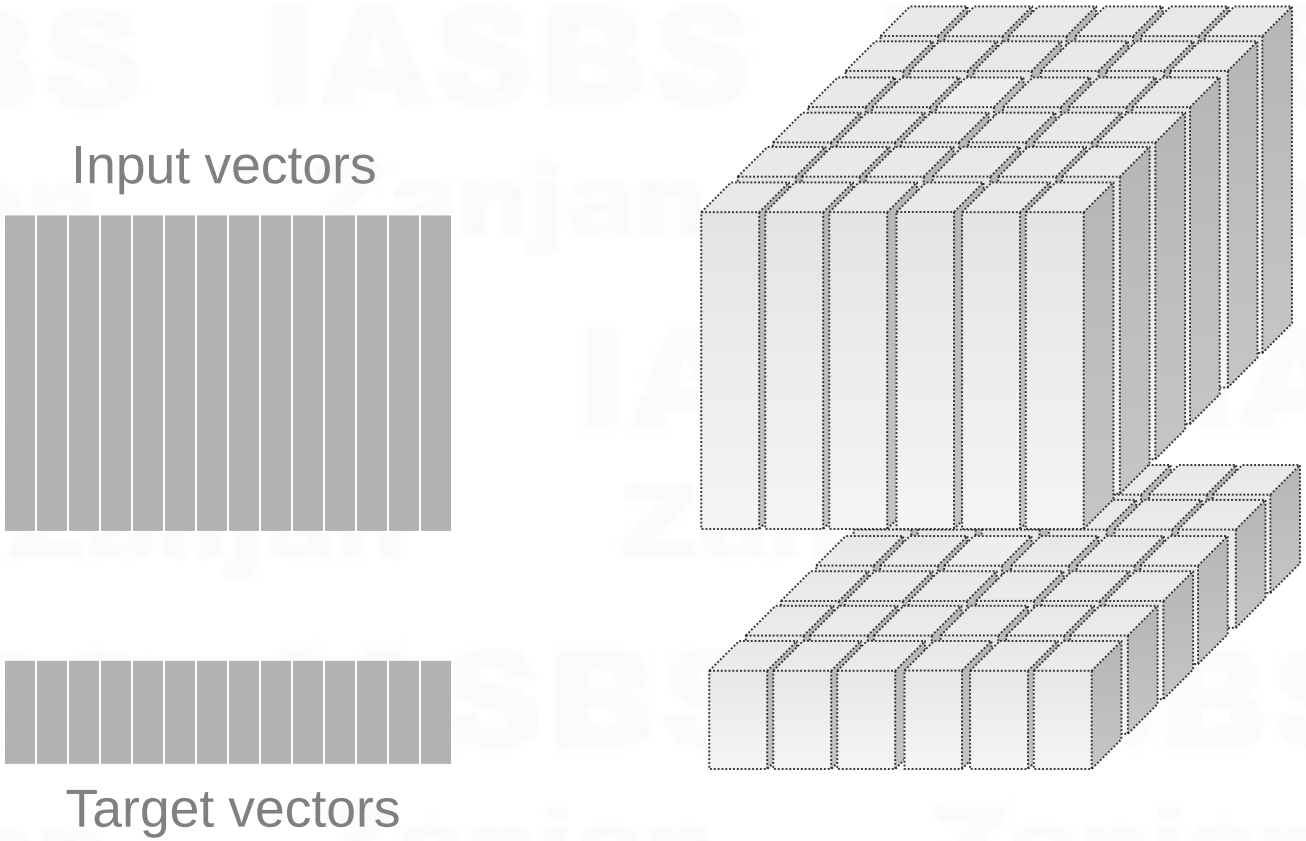
$$D = \sum_i \left\| \mathbf{x}_i - \mathbf{w}_{\text{winner}(i)} \right\|$$



Self Organizing Maps

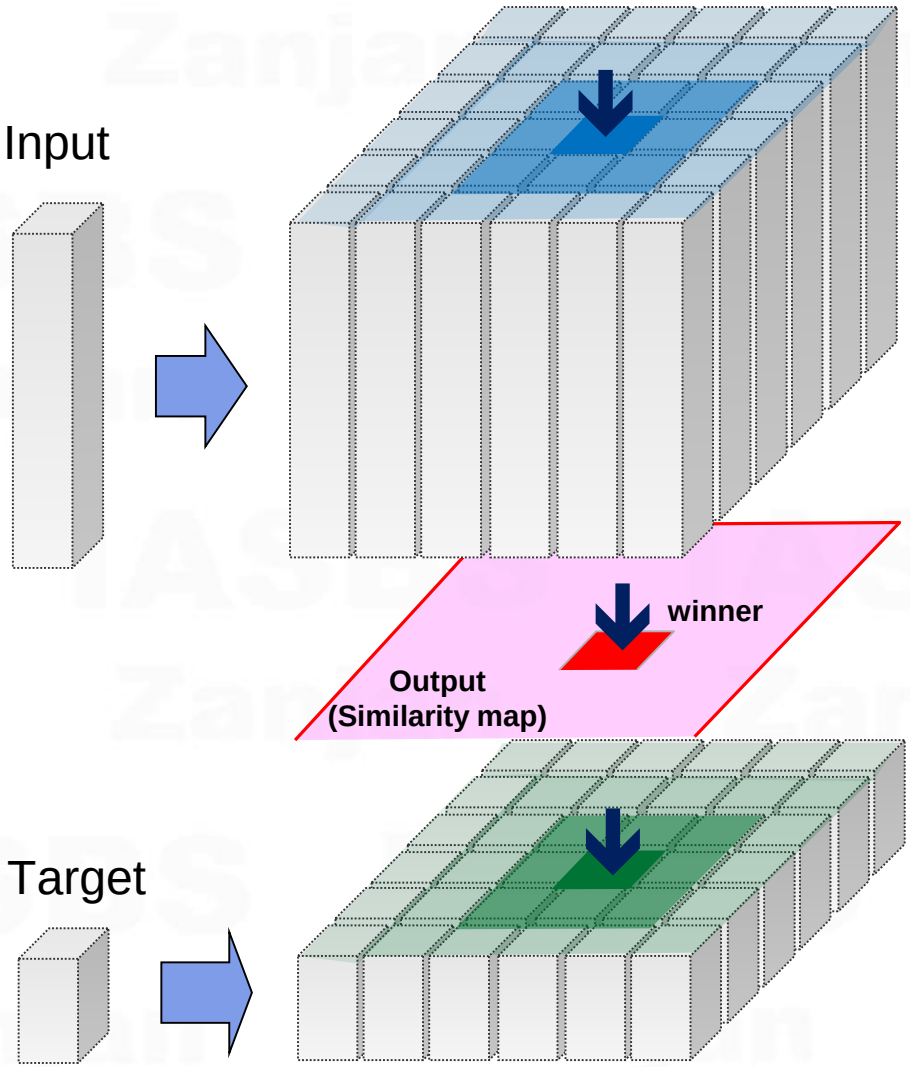
Counter Propagation ANNs

- CPANN is a supervised version of kohonen networks.
- It has the same structure as Kohonen network with an additional output layer with same layout as input layer.



Self Organizing Maps

Counter Propagation ANNs



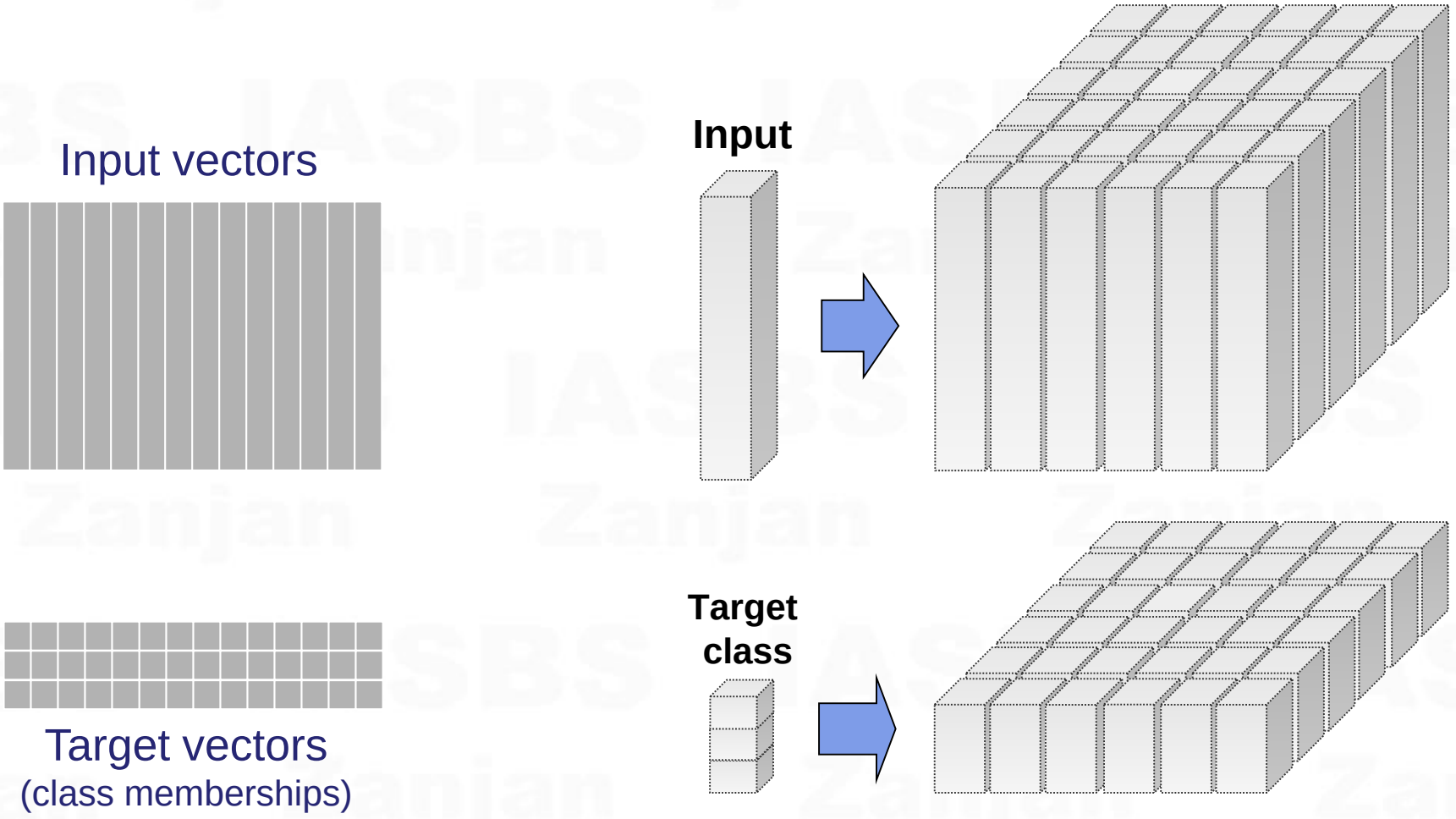
Based on the **location of** the **winning** unit in the input map (i.e., the unit which is most similar or closest to the presented object X), the **input map** and the **output map** are **updated** simultaneously at the same spatial locations.

If the CPN network is **trained**, it can be **used for prediction**. Simply, an **unknown** input object is presented to the network. The **position** of the winning unit **in the input map** then is used to look-up the **predicted value** of the corresponding unit in the **output** map.

Self Organizing Maps

Counter Propagation ANNs

As an example, For solving a classification problem (three classes):

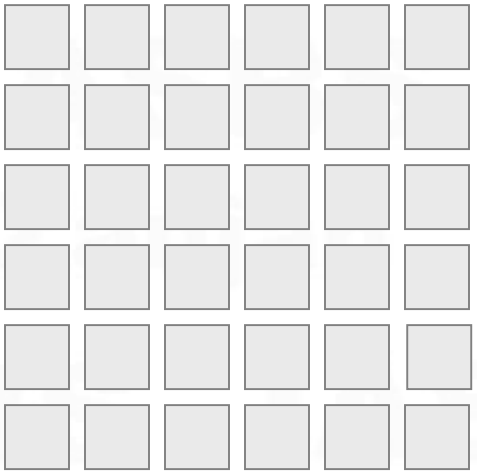


Self Organizing Maps

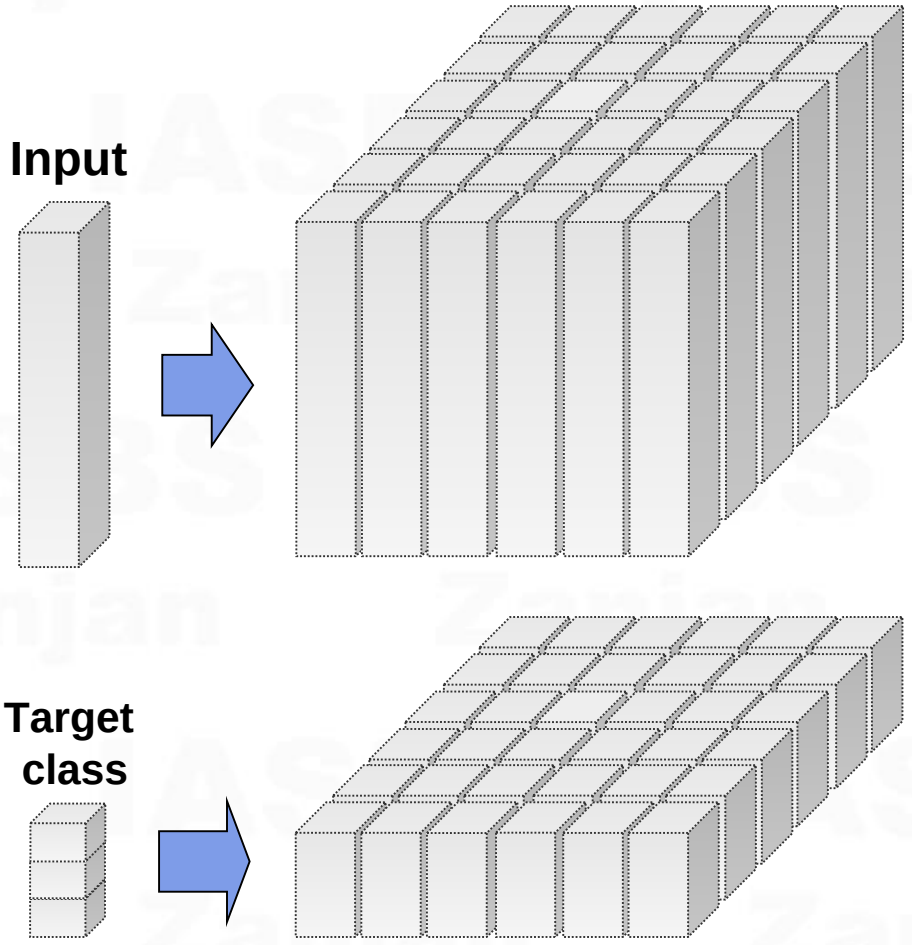
Counter Propagation ANNs

As an example, For solving a classification problem (three classes):

After completion of training process, weight vectors of the **output layer** can help us to assign each neuron to a class and build an **assignation map**.

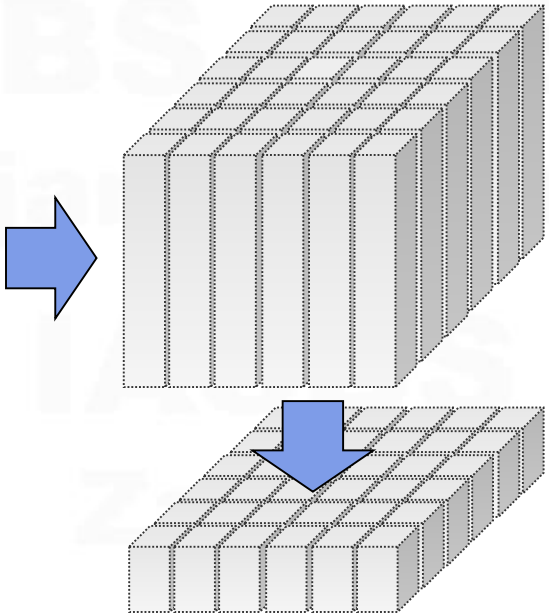


Assignation Map



Self Organizing Maps

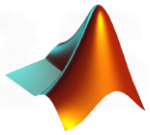
Counter Propagation ANNs



in a **CPN** network the **flow of information** is directed from the **input** layer units **towards** the **output** layer. For this reason, we prefer to denote the CPN as being a **pseudo-supervised** strategy.

The **targets** are **not involved** in the formation of the Kohonen input map. Hence, the CPN model cannot be considered as being a true supervised method.

Self Organizing Maps

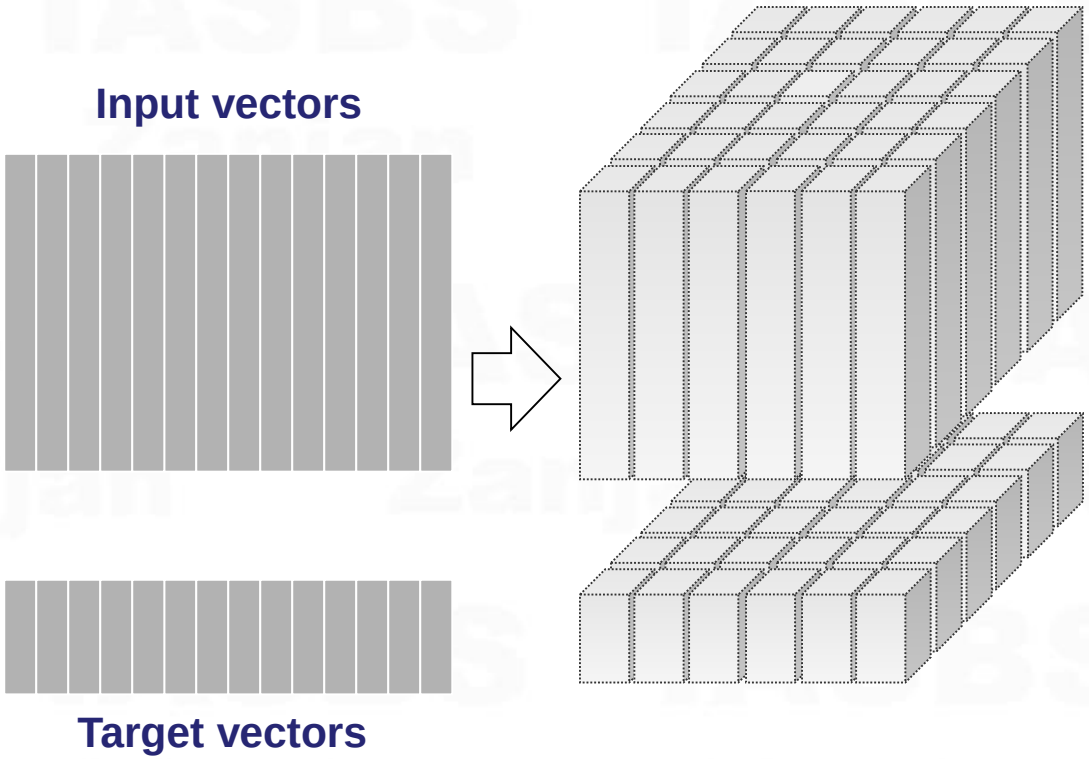


Write a Matlab function(s) to design a Kohonen Network.

- **Size of the network (n×m) and learning rate should be tunable.**
- **A Gaussian neighbor function should be used.**
- **Designed network should be checked using data points like Iris data (first 2 dimensions) and synthetic example data**
- **Training should be visualized in two dimensional data space**

Self Organizing Maps

Supervised Kohonen Networks



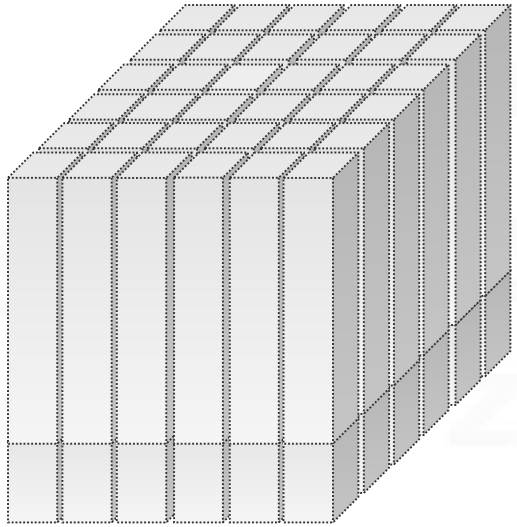
In a **SKN network**, the **input** layer and the **output** layer are '**glued**' together, thereby forming a **combined** input-output **layer**.

Because in a SKN **information** present in the objects **X** and **Y is used explicitly** during the update of the units in the map, the **topological formation** of the concatenated map is driven by X and Y in a **truly supervised way**.

Self Organizing Maps

Supervised Kohonen Networks

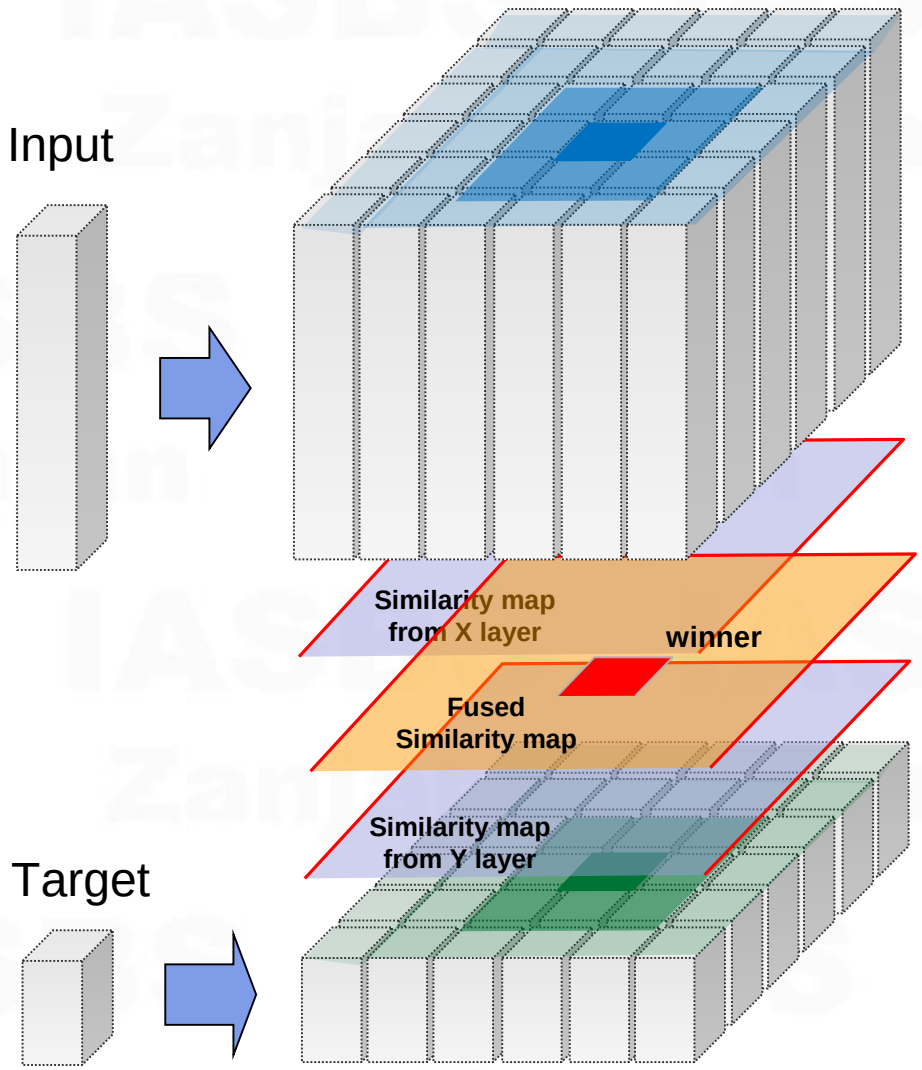
After training, the input and output maps are **decoupled**. Then, for a **new input** object its class membership is **estimated** according to the procedure outlined for the **CPN** network.



The **variables** of the objects X and Y in the **training** set must be **scaled** properly , but it is not trivial **how to deal** with the relative weight of the number of variables in X and the number of variables in Y during the training of a SKN network.

Self Organizing Maps

X-Y Fused Networks



By using a '**fused**' **similarity** measure based on a **weighted combination** of the similarities between an object X and all units in the input layer, and the similarities between the corresponding target and the units in the output layer, the common **win**ning unit **for both maps** is determined.

α(t) **decreases** linearly in epochs

$$S_{\text{Fused}}(i,k) = \alpha(t)S(X_i, \text{Xmap}_k) + (1-\alpha(t))S(Y_i, \text{Ymap}_k)$$

Self Organizing Maps

Bi-Directional Kohonen Networks

IASBS IASBS IASBS

ELSEVIER

Available online at www.sciencedirect.com
ScienceDirect

Chemometrics and Intelligent Laboratory Systems 83 (2006) 99–113

Chemometrics and intelligent laboratory systems

www.elsevier.com/locate/chemolab

Supervised Kohonen networks for classification problems

Willem Melssen *, Ron Wehrens, Lutgarde Buydens

Institute for Molecules and Materials, Radboud University Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands

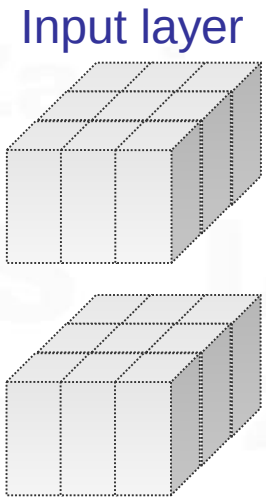
Received 22 December 2005; received in revised form 6 February 2006; accepted 8 February 2006
Available online 6 March 2006

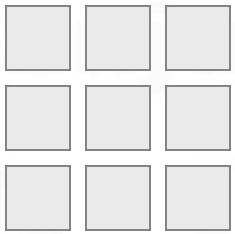
Abstract

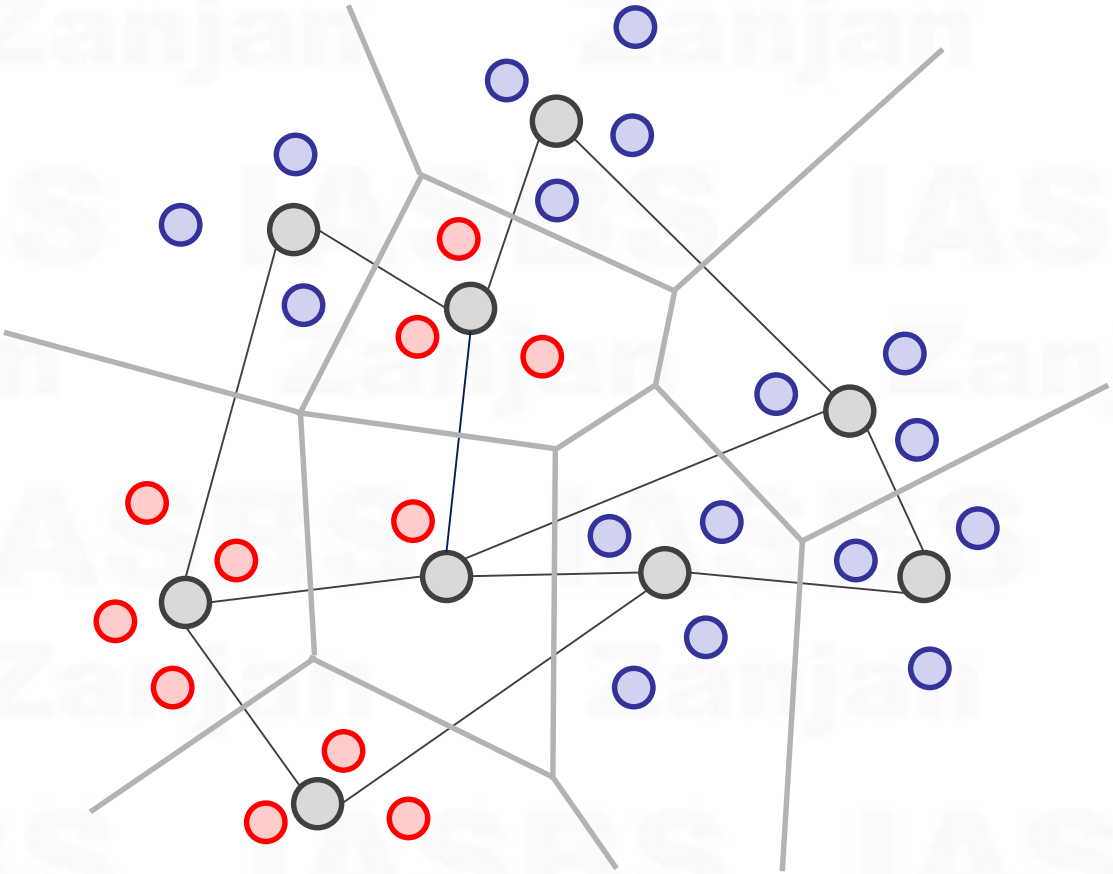
In this paper the transparency of the Counter Propagation Network (CPN) and the modelling power of the supervised Kohonen network (SKN) is combined. Two alternative supervised Kohonen networks are introduced: the *XY*-fused (XYF) and the Bi-Directional Kohonen (BDK) network. Both networks have in common that they deal in a straightforward and concise way with the (non-linear) relationship between the topology of the data and the corresponding class membership. The XYF network exploits a weighted and normalised similarity between a data object and the units

Self Organizing Maps

Classification Boundary

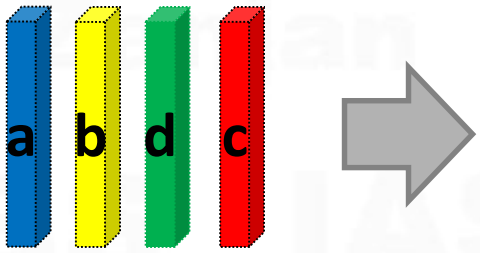




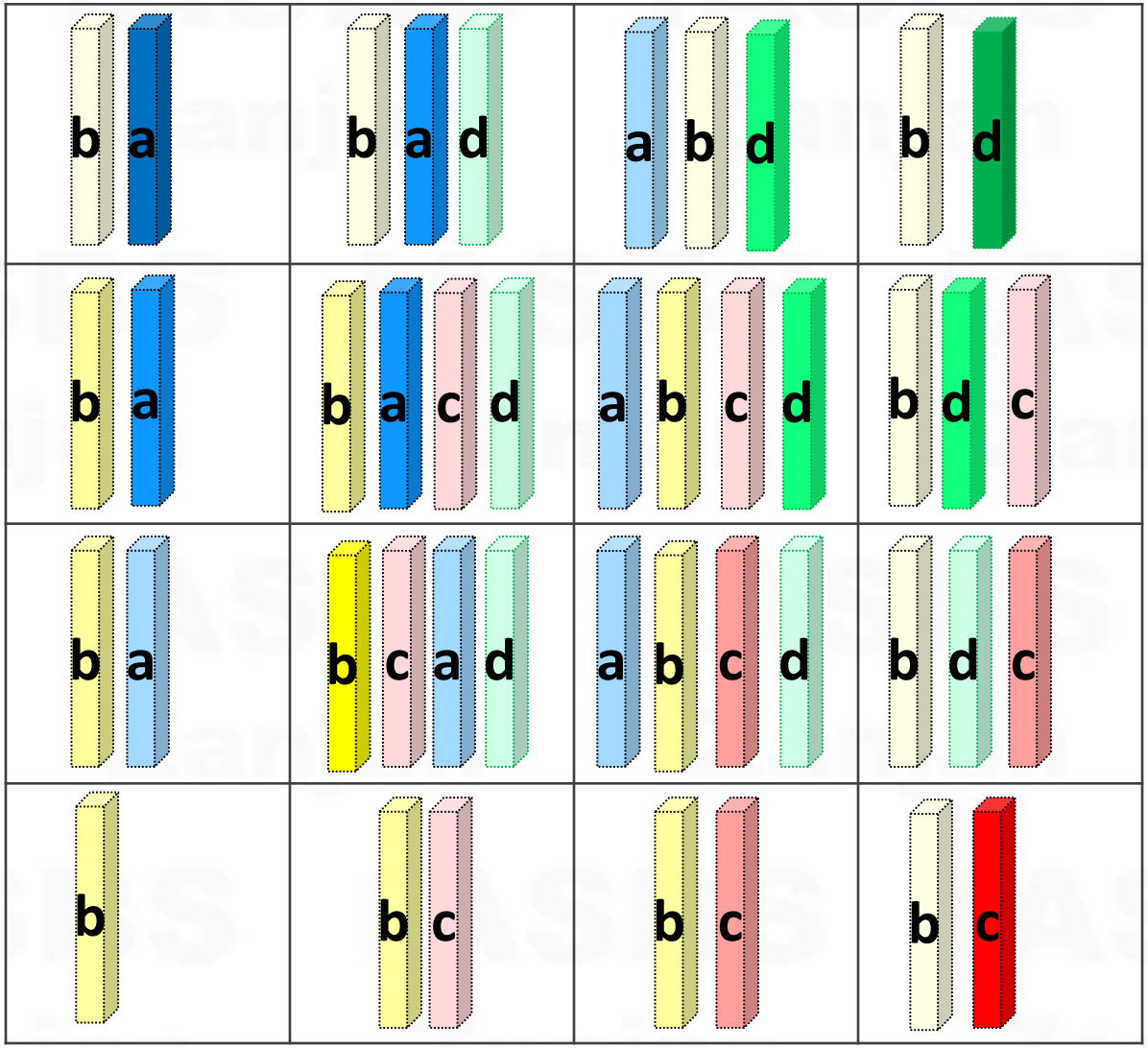


Self Organizing Maps

Batch Learning

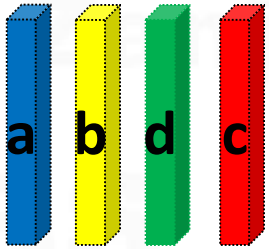


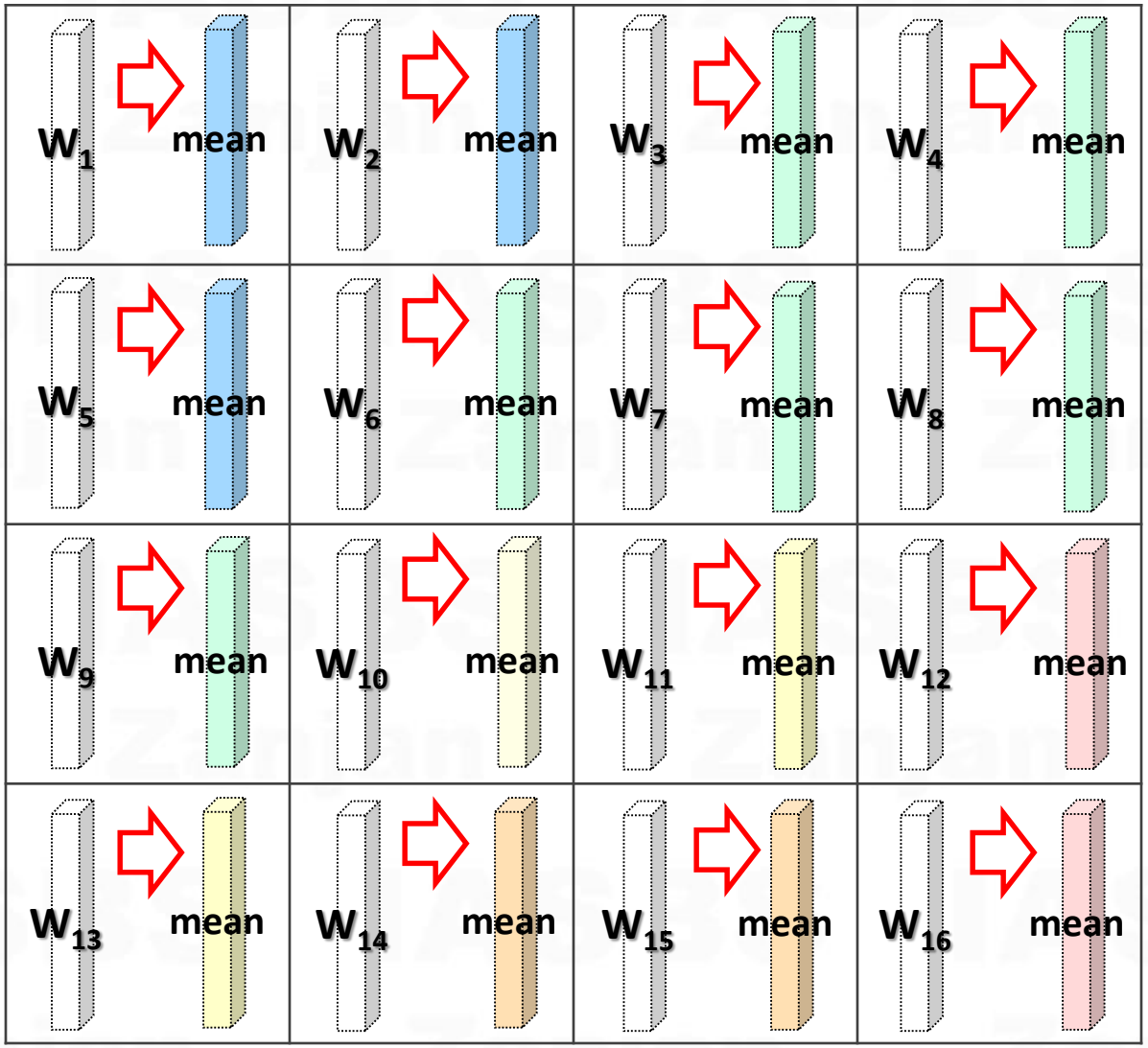
the whole set of samples is presented to the network and winner neurons are found.



Self Organizing Maps

Batch Learning



the whole set of samples is presented to the network and winner neurons are found

after this, the map weights are updated with the effect of all the samples: