

# 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 in the input and output maps for the simultaneous update of the network maps, whereas the BDK network uses this weighted similarity measure to update the input and output map in an alternating way.

It will be shown that both XYF and BDK networks yield better prediction models (expressed by the overall model accuracy) than the classical CPN and SKN networks. This study focuses solely on multi-output classification problems. Because in supervised self-organising maps (binary) class information is combined with continuous input values, we investigated the influence of two similarity measures applied to the output maps: the Euclidean and the Tanimoto distance. It will be shown that the Tanimoto distance measure yields better results. Two additional learning mechanisms will be introduced: adaptive learning and dynamical weight decay. Adaptive learning can improve network performance for difficult data sets. Inclusion of dynamical weight decay does this, too, and is especially useful for XYF and BDK networks.

Various ways to analyse the maps of the supervised Kohonen networks are introduced in this paper as well. For example, the average input profiles for each particular class membership and the visualisation of the correlation coefficients computed for all unit weights in the input and output map serve as additional tools to analyse the content of the networks and the nature of the relationship between the input and the output objects.

© 2006 Elsevier B.V. All rights reserved.

*Keywords:* Supervised Kohonen networks; Self-organising feature maps; Classification

## 1. Introduction

The basis for any fundamental or application-oriented study is a thorough analysis of the data. Traditionally, in chemometrics for such an unsupervised explorative study, linear transformations and data reduction techniques like Principal Component Analysis (PCA) are applied. The main advantage of such data reduction techniques is that, in general, high-dimensional data structures can be projected onto a low-dimensional coordinate system. The data at hand can easily be visualised and analysed by score-plots of the objects and the associated loading vectors providing information concerning the correlations present between the input variables. Moreover, performing a straight PCA analysis does not require a high level

of modelling expertise. However, PCA-like [1,2] transformation techniques have a few disadvantages as well. First of all, it is explicitly assumed that the high-dimensional structure (topology) of the input data can be reduced in a linear fashion. Secondly, these techniques perform only well if the objects in the data set do not contain oddities like outliers (which introduce a leverage effect disturbing the quality of the projection). Thirdly, the visualisation power of these transformation techniques deteriorates considerably if the number of relevant dimensions in the multivariate space (i.e., the number of significant principal components or, speaking mathematically, the actual rank of the input matrix) remains high after a PCA analysis. An alternative is formed by methods such as non-linear mapping and multidimensional scaling [3]. These explicitly aim to map objects in a low-dimensional space (usually two-dimensional) in such a way that the distances between objects are preserved in the mapping.

\* Corresponding author.

E-mail address: [W.Melssen@science.ru.nl](mailto:W.Melssen@science.ru.nl) (W. Melssen).

Approximately two decades ago, Kohonen [4] introduced an elegant mapping technique for examining the structure of high-dimensional data sets: the self-organising feature map. Such a Kohonen map incorporates in an unsupervised way the topology present in the data. Intuitively, the operation of a Kohonen network can be compared to the well-known Mercator projection which maps the three-dimensional Earth onto a flat, two-dimensional, topographical plane. Usually, a two-dimensional map is sufficient to elucidate all specific features characterizing the nature of the high-dimensional data. Moreover, many ways exist to visualise the properties of the units in a Kohonen map [4–7], which facilitates a thorough explorative analysis of the data. For example, the unsupervised self-organising feature map has been applied to explore the information contained in a large crystallographic database [8].

Usually, the follow-up of an explorative study is the model-building phase. In that case, a supervised modelling technique needs to be selected that is able to capture the relationship between the input data (measurements, observations) and output data (properties of interest). Four widely used techniques to model such relationships between input and output are Multiple Linear Regression (MLR) and Partial Least Squares (PLS) (both for regression problems) and Linear Discriminant Analysis (LDA) and *K*-Nearest Neighbours (KNN) (for handling classification purposes) [1,2]. As is the case with PCA, such techniques are quite easy to apply and are open and transparent meaning that one can look into the machinery of the particular model (inspection of regression coefficients and, for instance, visualisation of the scores and loadings of a PLS model). Nevertheless, these modelling techniques fail if a strong non-linear or topologically incoherent relationship is manifest between the input and output objects. Also problems might arise if the data contain a considerable number of outliers. Alternatives like artificial neural networks (ANNs) [9–11] and support vector machines (SVMs) [12,13] can tackle such non-linear relationships in a convincing way. However, the visualisation and interpretation of these models is severely hampered due to the fact that they are more or less ‘black-box’ techniques. The user really has to guess what is inside the predictive model.

Amongst others, Zupan [6] and Carrera [14] introduced a transparent modelling alternative in the field of chemistry and drug design: the Counter Propagation network (CPN). Many years ago, the theoretical concept of the CPN was founded by Hecht–Nielsen [15]. Although the CPN in its original appearance (a five-layered network) was a real supervised technique, later on its concept was drastically modified because it was observed that the CPN network was not able to model the inverse relationship between the output and the input [10]. So, nowadays, the CPN network simply consists of two layers: one input layer (a Kohonen network) and an associated output layer containing the values of the properties to be predicted. Consequently, the output layer of the simplified CPN model is driven exclusively by the topology present in the input space. In other words, the output properties are not involved in the formation of the directing Kohonen input map. Hence, the CPN model cannot be considered as being a true supervised method.

The transparency of a CPN network, however, is very appealing. One can easily look inside the driving Kohonen map and its associated output map layers to analyse the relationship between the input variables and the corresponding output properties.

Recently, following the example of Kohonen [4], Xiao [16] and Bayram [17] used a Supervised Kohonen network (SKN) to tackle problems in the fields of drug discovery and quantitative structure–activity relationships. Their SKN network exploits, on one hand, explicitly the (non-)linear relationship present between measurements (input) and properties (output). On the other hand, all aspects of visualisation and interpretation (transparency) of the Kohonen and CPN networks are still present in the supervised Kohonen variant. A drawback of the SKN network is that the user must determine beforehand the proper balance between the influence of the input and output objects: in general, correct scaling of the input and output variables is of utmost importance. Moreover, the ratio of the number of variables in the input and output objects determines to a high extent whether the SKN model is dominated by the input and/or the output objects. An imbalance between input and output might yield moderately performing or even wrong predictive models.

In this paper, we introduce two supervised Kohonen-based networks to overcome the problem of scaling and balancing of the input and output data. The first network, the *XY*-Fused network (XYF, where *X* refers to the input space and *Y* to the output), consists of two separate Kohonen maps: one for the input of the model (this map embeds the structure of the input space) and another map for the output (capturing the topology present in the output). Basically, the formation of each map in the XYF network is guided by the straight Kohonen training formalism [4]. To take into account the (non-linear) relationship between input and output, in the XYF model the formation of the maps is guided by a fused (shared) similarity measure. Briefly, the set of similarities obtained for an object *X* and the input map units is combined with the similarities corresponding to the output object *Y* and the output map to drive the formation process of the maps.

The second network, the Bi-Directional Kohonen network (BDK), has the same architecture as the XYF network. The scheme for updating the units in the maps, however, is different. In a BDK network the similarity between an object *Y* and the units in the output map determines to a high extent the formation of the units in the input map, whereas the similarities between an input object *X* and the units in the input map drive dominantly the adaptations of the units in the output map. Hence, in an BDK network the maps are updated in an alternating bi-directional way.

In multi-class problems the output objects usually are represented as multiplets of binary values. In SKN, XYF and BDK networks this binary information is combined in a supervised way with the input objects possessing continuously valued variables. Therefore we investigated the effect of two similarity measures applied to the output objects and the unit weights in the output map: the Euclidean distance and the Tanimoto distance [18].

For all networks, a modified learning strategy was implemented. First, we included the principle of adaptive learning (AL). The principle of AL is that well-matching objects have a higher impact to the adaptational process of the unit weights than bad matching objects. Secondly, during the modelling phase, optionally, the unit weights in the input map and output map can be decayed dynamically. In the past, such weight regularisation mechanism has been included in multi-layer feed-forward neural networks [9,19]. In this way, the risk of premature convergence of the networks was counteracted.

By invoking a straight normalisation of the set of similarity indices obtained for a  $(X, Y)$  pair and the units in both maps in the XYF and BDK network we circumvent the SKN scaling problem. Due to the dual map structure (in which each input and output entity possesses its own similarity measure), we rule out the problem of the imbalance between the number of input and output variables of the input and output maps. Accordingly, XYF and BDK are driven by the topology of the data and the relationship between individual pairs of input and output objects. Moreover, both techniques are able to model a non-linear relationship between input and output space. Last but not least, XYF and BDK networks yield transparent ‘open’ models, allowing the modeller to visualise and analyse the underlying relationship between input variables and output properties by inspecting (dis)similarities of the unit weights in the intrinsically coupled input and output maps.

The aim of this study is two-fold. First, we will focus on the quality, the coherence (i.e., units encoding similar features are directly connected in the map) and interpretability of the input and output maps of the Kohonen networks. Secondly, a comparison is made between the modelling power of the considered networks. For simplicity, in this paper we treat exclusively multi-class problems. In a forthcoming paper we will discuss the advantages of XYF and BDK networks for modelling multivariate non-linear multi-output regression problems.

This paper is organised as follows: first the basic principles of the unsupervised Kohonen and CPN networks and the supervised SKN, XYF and BDK networks are described in an intuitive way. In this section, we deliberately omitted the use of too many formula's because it was our intention to present just the basic concepts of these techniques. For readers who are interested in the mathematical background, we refer to some publications which discuss this topic in detail [4,10,16,17].

Next, we give a description of the data sets used to test the different networks. Three artificial data sets were created to investigate the quality of the input and output mappings and the modelling power of the CPN and SKN networks and the novel XYF and BDK networks: an Odd–Even recognition problem, an example of a typical three-class identification problem in which two classes are strongly overlapping and a modified four-class XOR-Cross problem. In addition, three real-world applications were selected to assess the map quality and interpretability and to compare the modelling power of all networks.

In the section Results and Discussion, the generalisation performance, applicability, advantages and drawbacks of the

four supervised Kohonen based networks will be discussed in depth. Finally, this paper will end with some conclusions.

## 2. Theory

### 2.1. Unsupervised Kohonen networks

#### 2.1.1. Kohonen self-organising feature map

A Kohonen network or self-organising feature map [4] consists of a set of non-interconnected units which are spatially ordered according to some topology; typically a two-dimensional hexagonal or rectangular grid is chosen. The units in the network are attached to the vertices of this grid. The term self-organising feature map refers to the unsupervised way the map is trained. Briefly, each unit in the map is equipped with a weight vector of which the number of elements is equal to the number of variables per input object (say, a spectrum, chromatogram or a combination of continuous, binary and ordinal variables).

Before the training of the network starts, the elements of all weight vectors need to be initialised by values in a data set specific range. Usually, for each map layer, the unit elements are initialised by small random number added to the average value of the corresponding variable obtained for the entire training set. Then, individual objects belonging to the pre-selected training set are presented in random order to all units in the network. The unit in the map possessing the weight vector most similar to the presented object is assigned to be the winner. Subsequently, the weight vectors of this unit and its closest neighbours in the map are updated by, first, calculating the difference between the actual input object and the respective weight vector and, second, adding this difference attenuated by a certain factor (i.e., the leaning rate) to the original weight vector. Thus, after a ‘match’ between input object and Kohonen unit, the weights of the winning unit and its neighbours become slightly more similar to the presented input object. This iterative process of weight updating is repeated until all objects belonging to the training set are presented a sufficient number of times to the network. Clearly, to obtain a qualitatively good Kohonen map, a proper similarity measure needs to be selected.

The size of the specific neighbourhood (with the winning unit located at the centre of it) is of vital importance to guarantee that relevant features of the multivariate input space are embedded in the weight vectors. Initially, the size of the neighbourhood is approximately equal to that of the size of the map itself. In this phase of network training, global characteristics of the data set are captured in the unit weights. During training of the network, the size of the neighbourhood is gradually decreased. This shrinkage procedure forces local clusters of units in both maps to start to represent specific combinations of features present in the data set. Finally, during the remainder of the training process, which takes most of the learning iterations, only the weights of the winning unit itself are adapted. As a consequence, such winning unit becomes specialised to those objects which are frequently mapped onto it. Usually, at the beginning the leaning rate is chosen relatively high (forcing a quick and global adaptation of the units in the

map); during the training process this rate is decreased gradually towards a small value (allowing the individual units in the Kohonen map to diversify).

### 2.1.2. Counter Propagation Network (CPN)

Fundamentals about the mathematical principles of the Counter Propagation Network (CPN) can be found in Ref. [10]. These authors called the original five-layered spider-like network a supervised neural network because of the ‘counter propagation’ way of information processing of the inputs (denoted by  $X$ ) and the outputs ( $Y$ ) throughout the hidden layer connections in this artificial neural network. Indeed, such CPN appeared to be able to model in a forward way the relation between the object pairs  $X$  and  $Y$ . However, it was observed that the CPN network was not capable to catch the inverse relationship for most practical problems, i.e., predicting  $X$  based on the properties of  $Y$ . Hence, an alternative CPN network existing of just two layers was proposed, suited to model the relationship in a unidirectional way between  $X$  and  $Y$  in which the input map drives the output map. Various interesting applications of this CPN network have been reported by [6,14].

In a nutshell, the unidirectional CPN network is actually trained in the same way as a unsupervised Kohonen network with the exception that an associated output map is trained simultaneously. An arbitrary input object  $X$  (taken from the input data set) is presented to all the units in the input map (in the remainder of this paper referred to as Xmap). 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 Xmap and the output map (Ymap) are updated simultaneously at the same spatial locations. For the update of the Xmap, the current object  $X$  is used to adapt the weight vectors of the winning unit and its neighbours, while for the update of the Ymap units the corresponding output object  $Y$  is used. 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 Xmap then is used to look-up the class membership of the corresponding unit in the Ymap. Then, the element having the maximum value of this unit’s weight vector determines the actual class membership. For example, for a three-class problem, the output vector  $(-0.99, 0.87, -1.0)$  of such winning unit will yield class 2 as output of the network.

Of course, all aspects (initialisation of weights, selection of network configuration, setting of dynamic neighbourhood shape and size, etcetera) which are required to be defined for training an unsupervised Kohonen network, are inevitably also needed in the set-up of a CPN network.

We would like to stress here, that in a CPN network the flow of information is directed from the Xmap units towards the Ymap (See also Fig. 1). For this reason, we prefer to denote the unidirectional CPN as being a pseudo-supervised or even unsupervised learning strategy, because the information present in the  $Y$  objects is not taken explicitly into account during the formation process of the driving Xmap. In fact, by applying an unsupervised Kohonen network on a particular data set, followed by a unit-wise averaging of the outputs associated

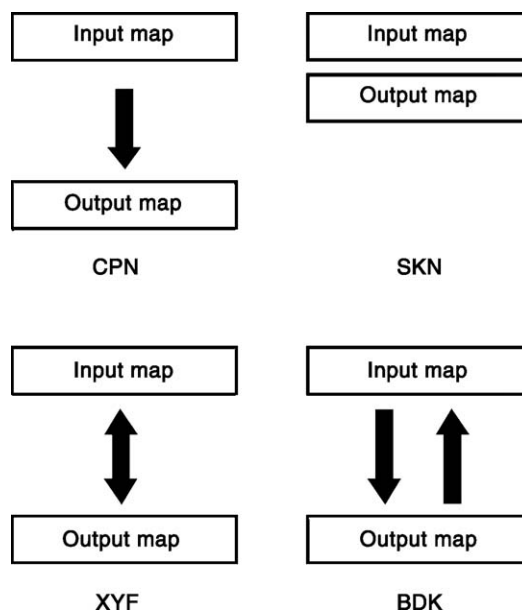


Fig. 1. At the top left: the unidirectional information flow in a Counter Propagation Network. Top right: in SKN networks the concatenated XYmap is updated according to the standard Kohonen learning strategy. Bottom left: the update of both maps in XYF networks is directed by the fused similarity measure, indicated by the two-headed arrow. Bottom right: both arrows schematically depict the bi-directional sequence for updating the units in the input and output map.

with the mapped  $X$  objects, a Ymap can be constructed which is similar to the Ymap of the CPN network. The only difference is that the averaged Kohonen output map will contain holes corresponding to units onto which none of the training objects is mapped.

## 2.2. Supervised Kohonen networks

### 2.2.1. Supervised Kohonen Network (SKN)

Previously, the SKN network was suggested by Kohonen as being a possibly more powerful modelling alternative as compared to its predecessor, the unsupervised Kohonen map [4]. In his book, Kohonen described various interesting application areas demonstrating the modelling power of the supervised Kohonen network. Recently, two applications have been described [16,17], in which such Supervised Kohonen Network (SKN) was used to model the relationship between the set of input and output patterns.

In a SKN network, the input map Xmap and the output map Ymap are ‘glued’ together, thereby forming a combined input-output map (XYmap) which is updated accordingly to the unsupervised Kohonen network training scheme. Each input  $X$  and its corresponding output  $Y$  are concatenated to serve as input for the common XYmap. 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. 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, in order to achieve an optimal embedding of the topology of the input and output space into the concatenated map of the SKN network. Moreover, 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. For example, it is not clear how a single output variable  $Y$  (in case of classification a 0/1 valued class membership variable) must be re-weighted internally in the concatenated XYmap in order to compete in a fair manner with, say, 2000 spectral variables belonging to the corresponding input object  $X$ .

### 2.2.2. $X$ - $Y$ Fused Network (XYF)

To deal with the aforementioned problems we developed two conceptually different ways of invoking a supervised Kohonen learning strategy. The first algorithm, the so-called  $X$ - $Y$  Fused network (XYF), exploits the similarities in both Xmap and Ymap in a straightforward way. By using a ‘fused’ similarity measure based on a weighted combination of the similarities between an object  $X$  and all units in the Xmap,  $S(X, Xmap)$ , and the similarities between the corresponding output object  $Y$  and the units in the Ymap,  $S(Y, Ymap)$ , the common winning unit for both maps is determined. The fused similarity measure,  $S_{Fused}(i, k)$ , for the object pair  $(X_i, Y_i)$  and unit  $k$  in the Xmap and Ymap can be expressed as:

$$S_{Fused}(i, k) = \alpha(t)S(X_i, Xmap_k) + (1-\alpha(t))S(Y_i, Ymap_k) \quad (1)$$

The common winning unit is determined by the location of the minimum in  $S_{Fused}(i, k)$ . The parameter  $\alpha(t)$  regulates the relative weight between the similarities  $S(X, Xmap)$  and  $S(Y, Ymap)$ . The argument  $t$  in  $\alpha(t)$  refers to the iteration number (expressed in epochs) during the training. One epoch equals the presentation of all the objects  $X$  (and thus  $Y$ ) contained in the training set to the XYF network. For XYF training,  $\alpha(t)$  decreases linearly in time, implying that in the initial stage of the training the similarity between the objects  $X$  and the units in the Xmap will dominate the determination of the common winning unit. At the end of the training, both similarities  $S(X, Xmap)$  and  $S(Y, Ymap)$  contribute equally to the determination of the shared winning unit.

It should be noted that throughout this paper we applied for both the input and the output map a straight Kohonen network in which the units were placed on the vertices of a two-dimensional rectangular grid. The units in the grid are numbered row-wise from the top-left position in the map (unit identifier equal to 1) to the bottom-right position (identifier  $N \times M$ ), where  $N$  denotes the number of rows and  $M$  the number of columns in the rectangular grid of units.

To deal with possible differences in magnitude of the similarity measures  $S(X, Xmap)$  and  $S(Y, Ymap)$ , both sets of similarities are re-scaled by their maximum value so that the maximal distances in both  $X$  and  $Y$  equal 1. According to Eq. (1), the common winning unit for both maps is determined, by calculating the minimum value in the set of fused Euclidean distances.

In case of the classification problems considered in this paper, it might be profitable to use the Tanimoto distance [18] instead of the Euclidean distance for determining the similarity between a  $Y$  object and the units in the Ymap. The Tanimoto distance is equal to 1 minus the number of equal (+1 or -1) values in  $Y$  and the weight values of a unit in the Ymap divided by the total number of  $Y$  variables. Hence, this yields a normalised ‘distance’ measure ranging between 0 (perfect match) and 1 (no match at all). Note that during training the Ymap weights are continuously valued and not exactly equal to +1 or -1: to overcome this problem we applied, before the Tanimoto distance was computed, a threshold function to the Ymap weights yielding a +1 for a positive weight element or a -1 otherwise. The Ymap weights are not modified by the threshold function; only a temporary binary copy is made of all the weight vectors in order to apply the Tanimoto distance.

At this point, we are in trodden turf. Both maps are simply updated simultaneously accordingly to the standard Kohonen formalism as has been outlined previously. Here, the common winner serves for both maps as the winning unit, which, on turn, defines its region of neighbouring units (depending on the actual size and shape of the neighbourhood function) that needs to be updated as well.

### 2.2.3. Bi-Directional Kohonen network (BDK)

Our second algorithm, the Bi-Directional Kohonen network (BDK) employs another approach for determining the winning unit and the adaptation of the units in the Xmap and Ymap. Instead of updating the units in both maps at once (based on a fused similarity measure, as is the case for the XYF network), in the first updating pass of a BDK network, only the weights of the units in the Xmap are adapted. The location of the winning unit is determined by the (initially) dominating similarity measure between an object  $Y$  and the Ymap units (see Eq. (2)). This procedure is repeated until all objects in the training set are presented once in random order to the BDK network.

$$S_{WinnerX}(i, k) = (1-\alpha(t))S(X_i, Xmap_k) + \alpha(t)S(Y_i, Ymap_k) \quad (2)$$

As was the case for XYF (see Eq. (1)),  $\alpha(t)$  regulates the relative weight between the similarities  $S(X, Xmap)$  and  $S(Y, Ymap)$ . Initially, a high  $\alpha(t)$  value is chosen. The winning unit  $k$  refers to the location of the minimum in the list  $S_{WinnerX}(i, k)$ .

In the reverse pass, the Ymap units of the BDK network are updated object-wise by using the winner determined by the dominating similarity between the objects  $X$  and the corresponding Xmap units according to Eq. (3).

$$S_{WinnerY}(i, k) = \alpha(t)S(X_i, Xmap_k) + (1-\alpha(t))S(Y_i, Ymap_k) \quad (3)$$

In this way the units in the Xmap and the Ymap are updated in an alternating way driven by the topology gradually embedded in the unit weight vectors located in the opposite map. Hence, the Xmap and Ymap units are updated in a bi-directional way (See also Fig. 1).

For a trained XYF or BDK network, the class membership of an unknown object is determined according to the classification procedure outlined for the CPN network.

### 2.3. Enhanced learning in supervised Kohonen networks

A problem which might be encountered during training of any Kohonen network is premature convergence (i.e., the network is stuck in a local optimum), resulting in bad or over-trained prediction models. A second problem might be caused by the random initialisation of the map weights: an unfortunate initialisation might lead to moderately performing networks. To overcome these two problems, we introduced an additional mechanism of learning which deals with the random weight initialisation and a possible premature convergence of the Xmap and Ymap: the dynamical decay of the unit weights. In other words, dynamical weight decay gives the network the opportunity to ‘recover’ during the training process from a bad weight configuration by modifying (shrinking) the unit weight vectors. In addition, we introduced the principle of adaptive learning. In extreme difficult (non-linear or degenerated) cases, this mechanism leads to an improved quality of the Xmap and Ymap, which, on turn, yields better and robust solutions.

#### 2.3.1. Dynamical weight decay (WD)

After each learning epoch, the weights in the Xmap and Ymap are rescaled according to

$$Xmap_k = Xmap_k(1-\beta(t)) \quad (4)$$

where  $\beta(t)$  is defined by

$$\beta(t) = \exp(-Decay_{begin} - (Decay_{end} - Decay_{begin})t/T) \quad (5)$$

In this formula,  $t$  refers to the epoch number and  $T$  to the maximum number of epochs. Typical values for  $Decay_{begin}$  and  $Decay_{end}$  are, respectively, 1 (implying that at the beginning of the training all map weights are epoch-wise rescaled by a factor 0.63) and 24 (no rescaling at all at the end of the training). Actually, the role of the parameter  $Decay_{end}$  is to regulate from the beginning of the training the number of epochs that WD will be effective. It should be noted that dynamical weight decay might interfere with the decay in time of the size of the neighbourhood function. For example, if the neighbourhood size shrinks too fast in time in comparison to the weight decay factor, then the growth of the weights will be counteracted too much by the WD mechanism. Obviously, WD must exert no influence any more at the end of the network training process.

#### 2.3.2. Adaptive learning (AL)

As was explained in the theory section, after the determination of a common winning unit, the weights of this unit (and its neighbours) in the Xmap and Ymap are updated. The degree of weight adaptation is controlled by the learning rate which is linearly decreased during training. In this way the maps learn to incorporate features present in the input and output space. However, it can be envisaged that a bad input and/or output

object might disturb (locally) this formation process. To prevent this, the learning rate was weighted by a factor  $F$ . In the XYF network, for a pair of input and output objects  $X$  and  $Y$ , the weighting factor is computed according to:

$$F = 2 - (\alpha(t)S(X, Xmap_k) + (1-\alpha(t))S(Y, Ymap_k)) \quad (6)$$

Here  $k$  refers to the position of the common winning unit. Because both similarity measures are normalised, theoretically the factor  $F$  can take values between 1 (no match at all between the objects and the map units) and 2 (a perfect match for both maps). Hence, the degree of adaptation of the weights is increased by a factor 2 if a perfect matching object pair is presented to the XYF network. For the BDK network the role of the similarity measures  $S(X, Xmap)$  and  $S(Y, Ymap)$  is interchanged due to its bi-directional character. In CPN only the non-weighted similarity  $S(X, Xmap)$  was used to calculate the factor  $F$ . Last, in the SKN network, the  $\alpha(t)$  weighted similarities  $S(X, Xmap)$  and  $S(Y, Ymap)$  are replaced by the single non-weighted similarity for a  $XY$  object and the winning unit in the concatenated XYmap.

## 3. Experimental

### 3.1. Simulated data sets

The first data set, referred to as *Odd–Even*, contains 400 rows (input objects) with 8 variables: each integer valued variable was varied randomly between 1 and 100. The corresponding output for each object was generated by applying the following decision rule: if per row the total number of even values was greater than the total number of odd values, the class membership for that row was assigned to be 1 (147 objects), otherwise the class membership was set to  $-1$  (253 objects). This data set was used to investigate whether the CPN, SKN, XYF and BDK networks are able to discover this quite artificial relationship between the  $X$  and  $Y$  objects. Moreover, in a qualitative sense, we examined for each network the coherence of the input and output maps. Note that in this particular case the multivariate distribution in the input space is, in a topological sense, not at all related to the distribution of the actual classes in the output space.

The *Overlap* data set consists of three normally distributed clouds of data points in three dimensions: the first 150 objects belong to a multivariate normal distribution around the origin (objects assigned to be of class 3), whereas the other two classes 1 and 2, each consisting of 150 objects as well, are normally distributed around the centroids (5, 3, 4) and (5.5, 3.5, 4.5). The scope of this simulated data set was to check whether the Kohonen networks are capable to unravel the overlap of the data points in the input space by exploiting the known class membership in the output space.

The third data set, *XOR–Cross*, contains as input objects duplets of randomly generated continuous values for two dimensions in the range of 0 up to 1. These input values (denoted by  $X_1$  and  $X_2$ ) were assigned to their class membership following the scheme graphically depicted in Fig. 2. 600

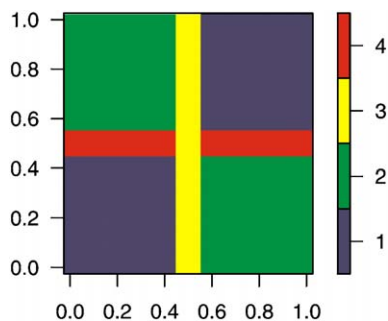


Fig. 2. Graphical representation of the *XOR-Cross* problem. The four square areas correspond to the standard XOR problem. The vertical and horizontal bar in this image correspond to the two additional classes. The class membership is indicated by the colour bar at the right side.

random samples were generated consisting of the object pairs  $X$  (continuous valued, 2-dimensional) and  $Y$  (binary valued, 4-dimensional). Obviously, classes 3 and 4 are represented by far fewer objects than class 1 and 2: 251 for class 1, 229 for class 2, 61 for class 3 and 59 for class 4. The aim of this data set was to investigate whether the (un)supervised Kohonen networks were able to deal with 1) the classical XOR problem and 2) the minority of objects, i.e., the two local disruptive data partitions (the narrow vertical and horizontal bar in Fig. 2) belonging to the classes 3 and 4. This data set was used to investigate the quality and the classification power of all (un)supervised networks.

### 3.2. Real world data sets

The first real-world data set was gathered during the EC Interpret project [20] and will be referred to in the remainder of this paper as the *Interpret* data set. This data set consists of 669 Magnetic Resonance Spectroscopy (MRS) voxel measurements and corresponding areas in Magnetic Resonance Images (MRI) of brain scans of healthy volunteers and tumour patients. Each measurement (object) is characterised by 10 metabolite concentrations derived from the MRS spectra (i.e., Creatine, Glutamate, Myo-Inositol, unresolved Glutamate and Glutamine, Choline, Creatine2, Glutamate2, *N*-Acetyl-Aspartate, Lactate and Fatty Acids) and 4 averaged pixel intensity values acquired from the proton density,  $T_1$  weighted,  $T_2$  weighted and Gadolinium enhanced MRI images. The six identified classes in this data set belong to Healthy tissue (class 1), Cerebro spinal fluid (2), Oligodendroglioma tumours of grade II (moderate stage), III (intermediate) and IV (lethal stage) (classes 3 up to 5), and Meningioma tumours (6).

The *Autosort* data set was collected during the EC project Autosort [21]. The aim of this project was to build an automated system for the recognition and robotic sorting of three demolition waste fractions: wood, paper and cardboard (class 1), plastics (class 2) and stone, glass and cement (class 3). For the initial set-up of the system 1245 NIR spectra (at 228 wavelengths) were measured: 291 for the wood fraction, 191 for the plastics and 763 for the stone fraction.

The third data set, extensively described in Ref. [13,22], hereafter referred to as the *NIR* data set, contains 95 NIR spectra

(measured in the wavelength range of 850–1049 nm) measured at different temperatures. In particular this set is interesting for exploring the modelling capabilities of the CPN, SKN, XYF and BDK networks because it is known that the NIR spectra are strongly influenced by the sample temperature in a non-linear way. We used the same division of the NIR spectra for the training set (65 objects) and test set (30 objects) as described in [22]. Instead of predicting the concentrations of water, ethanol and iso-propanol for the five sample temperatures (30, 40, 50, 60 and 70 °C), we modelled the temperature classes themselves. Class 1 corresponds to spectra measured at 30 °C, class 2 to 40 °C, etcetera. Hence, in total 5 classes are present in this data set.

### 3.3. Network configuration and parameters

For the CPN, SKN, XYF and BDK networks used throughout this paper, per data set the same network parameters were used. All networks consisted of units organised in a square grid having hard boundaries implying that no toroidal updating (wrapping around the map edges) was allowed. The number of epochs was set to 200. Convergence was checked by monitoring the sum of absolute differences between the weights of the units in the input and output map before and after an epoch. The learning rate for the units in the Xmap and Ymap was initially set to 0.1 and was linearly decreased to a value of 0.001 at the end of the training. The neighbourhood function was a square with the winning unit at the centre of it. The neighbourhood size was decreased exponentially in time during training. Initially, the size of the square was set to size of the entire network minus one. As a consequence of this, initially almost the entire maps were adapted if the winning unit was located at the centre of the map, whereas for the last 66% of the training epochs only the winning unit in both maps was updated. The learning rate for the units within the neighbourhood was weighted by a Gaussian implying that the learning rate for the winning unit gets a weight of 1 whereas units located at the edges were weighted by a factor  $\exp(-w^2)$ , where  $w$  denotes the width of the neighbourhood function.

The XYF and BDK algorithms have, in contrast to CPN and SKN, the flexibility to define different neighbourhood shapes and sizes and, moreover, different learning rate adaptation schemes for the Xmap and Ymap, respectively, to accommodate in a subtle way to the nature and complexity of the problem at hand. Nonetheless, in this paper, for the sake of simplicity, the XYF and BDK input/output maps were updated according to identical parameter settings.

For the *Overlap* data set the effect on the mapping accuracy of the Euclidean and Tanimoto distance will be investigated. For all other data sets we applied the Tanimoto distance as similarity measure for the output objects and the output map.

Throughout this paper the weighting parameter  $\alpha(t)$  was decreased linearly in time from 0.75 towards 0.5 (See Eq. (1) (XYF network) and Eqs. (2) and (3) (BDK network)).

The most crucial parameter, i.e., the size of the input and output map, was optimised for each data set. As a rule of thumb, first the number of units in the network was set to  $2N \times 2N$ ,

where  $N$  equals the number of classes, but maximally the total number of objects in the training set. Then, for a few test runs the network size was decreased step-wise. The optimal setting was achieved if 1) monitoring of the weight changes still indicated a good convergence and 2) the difference between the prediction errors for the training and test sets differed not more than 10% (this to avoid over-training). This procedure was applied to all four network types thereby selecting that minimal network size which did not deteriorate the performance of any of the networks. The input variables were not scaled unless stated otherwise.

### 3.4. Model validation

For most data sets used in this paper, the classification networks were evaluated by a 10-fold validation procedure: 10 random divisions of the data were made resulting in 10 training and 10 independent test sets. The division was done in a balanced way, implying that the ratio of the different classes appearing in the total set was preserved in the training and test sets. The training sets consisted of 67% of the data. The test sets containing the remaining 33% were used to evaluate the performance. Throughout this paper, the performance of a network is expressed in a mean and a standard deviation of the prediction accuracy. For those data sets with a predefined division in a training and a test set, we trained and tested also 10 classifier networks, now having 10 different weight initialisations, of which the performance was evaluated in the same way as has been outlined above. For the *Odd–Even* and *Overlap* data set, just one explorative model was generated using the complete data set.

### 3.5. Software

The unsupervised Kohonen network and all supervised Kohonen variants discussed in this paper were implemented in Matlab code (Matlab V6.5, The Mathworks Inc, Natick) and R [23]. Both software packages offer a lot of visualisation functions to analyse the CPN, SKN, XYF and BDK networks.

Both toolboxes are available at our web-site: <http://www.cac.science.ru.nl/software>. The R package, called kohonen, is also available from the central R package repository: <http://cran.r-project.org>.

## 4. Results and discussion

### 4.1. Simulated data sets

First, we start with exploring (without validating the models) the nature of the artificial data sets *Odd–Even* (2 classes) and *Overlap* (3 classes). All objects in the *Odd–Even* data set were presented to the CPN, SKN, XYF and BDK networks. A square grid of  $12 \times 12$  units was used in all networks. When looking to the corresponding output maps (see Fig. 3), one can observe that the map of the CPN network displays a scattered pattern of colours. Even after applying the classification procedure to the output map (as was outlined in the Theory Section), still a chess-

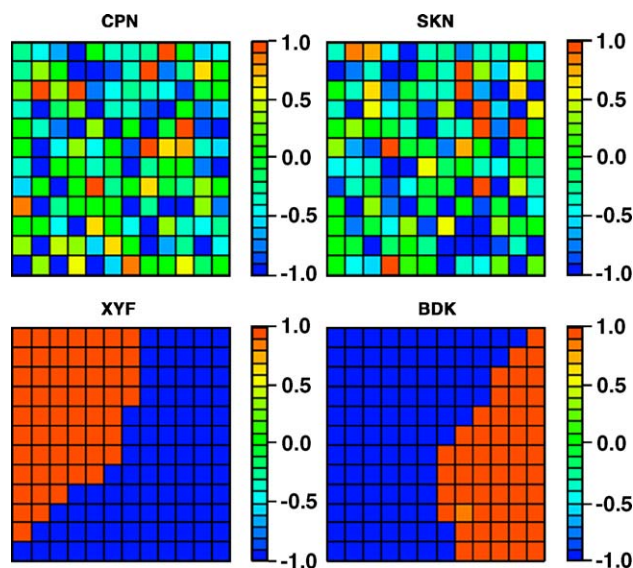


Fig. 3. Output maps of the four supervised Kohonen networks for the *Odd–Even* data set. Each Xmap and Ymap consists of  $12 \times 12$  units. The number of epochs was 200. Classes are indicated by the extremes appearing in the bar at the right hand of each map. Intermediate output values can be interpreted as being a kind of probability for belonging to one of the two classes.

board like pattern remained. This is due to the fact that there exists no real one-to-one relationship between the multivariate topological structure present in the input space and the associated class membership of the objects. Moreover, the CPN network does not take into account the  $Y$  information during the formation of the input and output maps. The Ymap of the SKN network shows a similar scattering of unit weight values. Although SKN takes the output variable into account during the formation of the maps, apparently the imbalance between the number of input (8) and output (2) variables results in such speckled incoherent map. Contrary to that, the XYF and BDK networks possess nice coherent output maps, indicating that a certain ‘hidden’ relationship must be present between the input and output space. Thereupon, the Ymap shows a crisp transition between both class areas, indicating an unambiguous class membership assignment. The XYF output map, for instance, illustrates that all units in the upper left area of the associated Xmap represent data objects having more even elements (class 1) than the units located on the right side of the map. Also the ratio between the number of units identifying class 1 and 2 in the output maps of the XYF and BDK networks corresponds well to the ratio of the number of the ‘Even’ and ‘Odd’ objects present in the data set. The mapping accuracies of the networks are summarised in Table 1. Clearly, the XYF and BDK networks are better able to capture the underlying artificial relationship between the  $X$  and  $Y$  objects.

Fig. 4 shows for both an extreme high and the default initial learning rate the map weight convergence plots for a BDK network. These curves were constructed by calculating during the training of the network the sum of the absolute difference between all elements of the unit weight vectors in the Xmap (and Ymap) before and after one epoch, respectively. At each time instance this sum of differences was divided by the actual

Table 1

For the *Odd–Even* and *Overlap* data sets, exclusively the mapping accuracies (i.e., the percentage of correct predictions) are given

	<i>Odd–Even</i> (%)	<i>Overlap</i> (%)	<i>XORC</i> train (%)	<i>XORC</i> test (%)
CPN	74	78	85±2	85±3
SKN	75	82	86±2	88±1
XYF	<b>94</b>	<b>84</b>	91±1	91±2
BDK	93	83	<b>92±1</b>	<b>92±2</b>

For the *XOR-Cross* data set (*XORC*), the accuracies are given for the training as well as the test sets. Means and standard deviations were determined by applying a ten-fold validation procedure. Bold numbers indicate for each data set the best performing network.

learning rate. The two curves at the top panel (belonging to an initial learning rate of 0.5) exhibit strong oscillations, this in particular for the Ymap curve. A sudden drop for the Ymap curve appears after 70 epochs. The Xmap curve remains flat for about 80 epochs and then starts to decay slowly. Clearly the weights in the Xmap had not converged at all after 200 epochs. This is confirmed by the prediction accuracy: this particular network predicted approximately 40% of the training objects correctly. The lower panel belongs to an initial learning rate of 0.1: both curves decay more or less smoothly in time and after some 100 epochs the unit weights in both maps become quite stable. The appearance of the Xmap and Ymap was very coherent and the accuracy of the converged network was 93%.

Fig. 5 compares the quality of the 8×8 input and output maps for the CPN and XYF networks for the *Overlap* data set. Clearly, both algorithms are able to capture the uniform distribution of the objects in the 3-dimensional input space. However, by inspecting the corresponding output maps, one can observe that the CPN network was not able to unravel the two overlapping classes 1 and 2. The upper two Ymaps, which

correspond to these classes, show quite diffuse patterns for the area where the corresponding input objects are located in the corresponding Xmap. The third isolated class is well separated by the CPN. The output maps of the SKN network were slightly more coherent. The XYF network, however, not only separates class 3 from the other two, but also orders the classes 1 and 2 in two distinct areas in the output maps. The same observation holds for the unit weights in the Xmap. The two maps depicted at the lower row of Fig. 4 represent the so-called classification maps: such integrated map summarises (according to the classification criterion used in this paper) the information contained in all individual output map layers shown in Fig. 5. Clearly, the XYF network explicitly exploits the class membership during the formation of both maps. This was also the case for the BDK network. Table 1 shows that all three supervised networks yield a comparable modelling accuracy. Interestingly, for these networks, the average of the weight vectors in the Xmap belonging to a certain class resembled closely the centroid of that particular class, which was used to generate the normally distributed objects in this data set. For example, for the XYF network the following averages of the Xmap units, based on the classification map, were calculated: (5.1, 3.0, 3.9), (5.5, 3.4, 4.5) and (0.1, 0.2, 0.1).

At this stage, we switch from exploring the underlying relationships between data structures in the multivariate input/output space towards building and validating classifier models. The *XOR-Cross* data set was modelled by the supervised Kohonen networks having a size of 6×6 units. The last two columns in Table 1 show per network type the mean mapping accuracy and the corresponding standard deviation for the randomly selected class-balanced training and test sets. As was already observed for the *Odd–Even* and *Overlap* set, the CPN network performs slightly worse than the SKN network. The

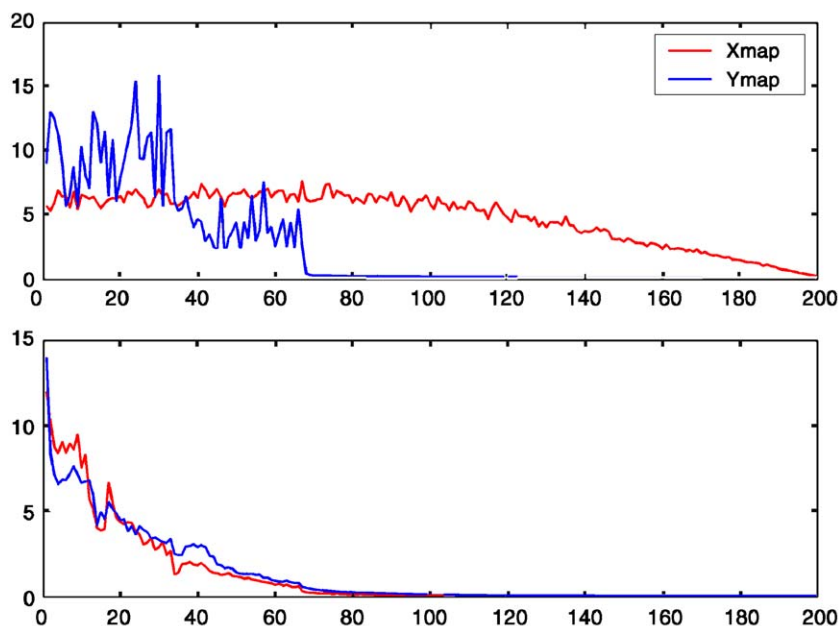


Fig. 4. Map weight convergence plots for two learning rates. In this example, a BDK network was trained for the *Odd–Even* data set. The Xmap and Ymap curve in the upper panel correspond to an extreme high initial learning rate of 0.5. The curves in the lower plot belong to the BDK network trained with the default initial learning rate of 0.1. For both cases the final learning rate was equal to 0.001.

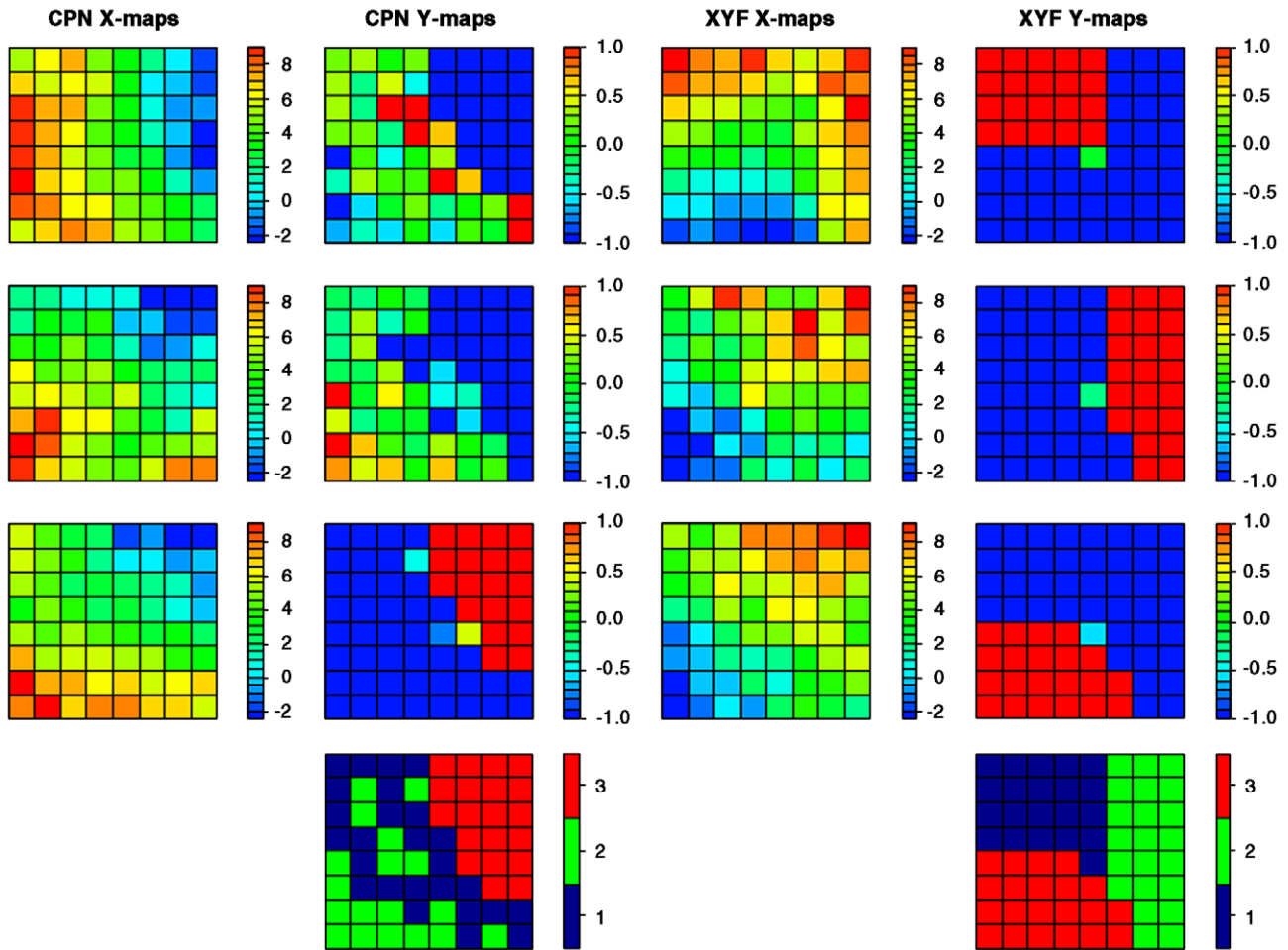


Fig. 5. Input and output maps for the *Overlap* data set. Map size is  $8 \times 8$ , number of epochs 100. On the left, the maps for the three input variables are shown for the CPN network. The output maps for the three classes are depicted in the second column. The map in the fourth row represents the classification map. This map summarises the information contained in the three maps depicted above. In the third column the input maps for the XYF network are shown. At the right, the layers of the output map. For the Xmaps and Ymaps the bars indicate the range of the unit weight values. The bar at the right of the classification maps indicates the class membership.

XYF and BDK network models yield for all simulated data sets the best classification performance. Based on one of the ten BDK networks, obtained during the 10-fold validation procedure, we outline here one of the various ways to analyse the unit weights in the Xmap and Ymap.

At the left hand of Fig. 6, it is shown that maps belonging to the two uniformly distributed input variables  $X_1$  and  $X_2$  are organised in a smooth coherent way. Now, let us reconsider the *XOR-Cross* problem (depicted schematically in Fig. 2), this with regard to the relation between the input map and the classification map. The area at the right side in the classification map shows that class 1 is related to combinations of high  $X_1$  values together with small  $X_2$  values, and the other way around. The units at the left correspond to units in the Xmaps possessing small ( $X_1, X_2$ ) values or, opposed, combinations of high values. Indeed, this is exactly conform the decision rules establishing the classical *XOR* problem. The units located at the centre of the classification map correspond to the  $2 \times 2$  square in the middle of the first Xmap layer (here, values are close to 0.5). The same holds for the units belonging to class 4 and the small L shaped region down–

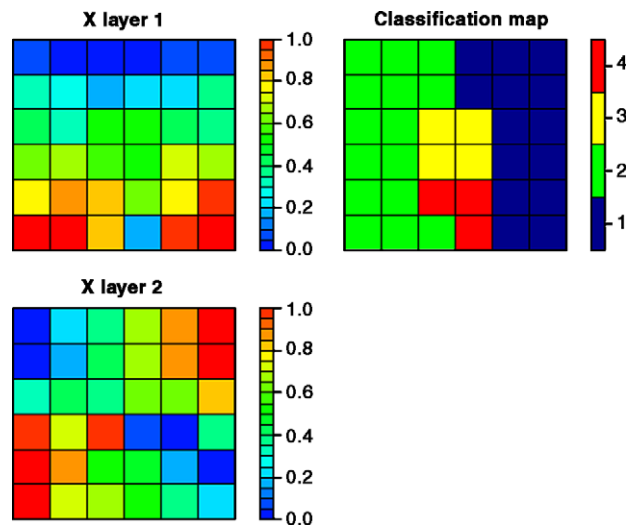


Fig. 6. Input and output maps of the BDK network for the *XOR-Cross* data set. At the left, the two Xmaps for the two input variables are shown. The image at the right depicts the classification map: the areas at the left and right hand in this map correspond to the classical XOR problem. The other two areas belong to the classes corresponding to the horizontal and vertical bar shown in Fig. 2.

middle in the  $X_2$  input map. This is in agreement with the horizontal and vertical class bars shown in Fig. 2.

#### 4.2. Real world data sets

For the *Interpret* data set, the transparent character of the Kohonen networks is illustrated in Fig. 7. Three metabolite layers and one pixel intensity layer of the trained  $12 \times 12$  Xmap are shown. Clearly, high concentrations of *N*-Acetyl-Aspartate correlate to the ‘healthy’ region in the classification map depicted in the lower section of Fig. 7. The lactate and fatty acids Xmap layers show, on one hand, that these maps are strongly correlated and, on the other hand, correspond well to the Oligodendroglioma grade IV tumours. This is in agreement with bio-medical insights: fatty acids and lactate are often present in necrotic areas of high grade tumours. The Xmap down-right shows a strong visual correlation between low valued average pixel intensities (units in the upper-right corner) of the proton density MRI image and the ‘CSF’ region in the classification map. High pixel intensities are present for the classes ‘healthy’, ‘grade II’ and ‘grade IV’. Apparently the proton density map is in some way discriminative for healthy brain tissue/fluids and some tumour grades. Indeed, in many hospitals it is common practice to measure the proton density map during a set of MRI brain scans. Table 2 shows that XYF and BDK perform on average better than the CPN and SKN networks. The prediction performance of the novel networks is even better as had been reported by Simonetti et. al who used a

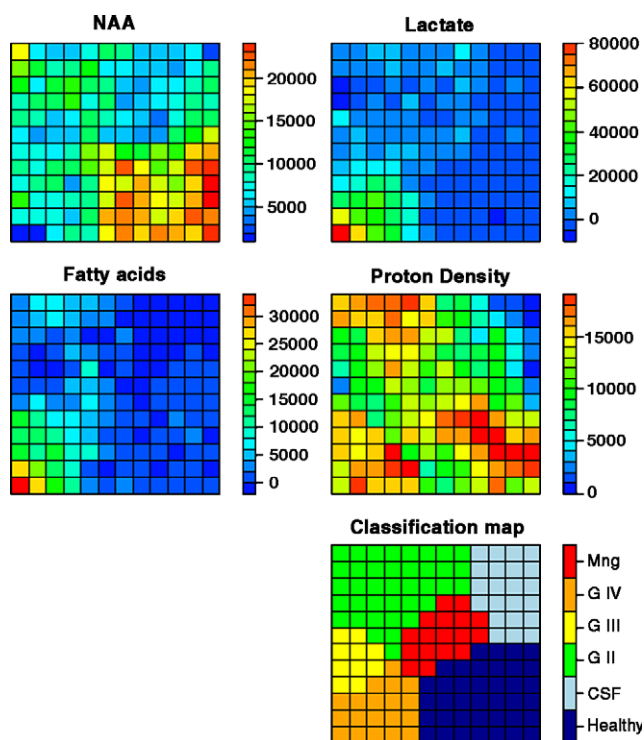


Fig. 7. The XYF maps for the *Interpret* data set. A  $12 \times 12$  network was used. At the top, four layers of the input map are depicted: *N*-Acetyl-Aspartate (NAA), Lactate (Lac), fatty acids (Fat) and the averaged pixel intensities derived from the proton density MRI image. The classification map at the bottom indicates in which region a particular brain tissue, fluid or tumour class is present.

Table 2

Performance of the four Kohonen networks obtained for the ten random test sets used during the ten-fold validation procedure

WD		<i>Interpret</i> (%)	<i>Autosort</i> (%)	<i>NIR</i> (%)
–	CPN	91 ± 2	93 ± 2	63 ± 7
–	SKN	92 ± 1	94 ± 1	69 ± 9
–	XYF	95 ± 1	<b>97 ± 0.5</b>	72 ± 7
–	BDK	<b>96 ± 1</b>	96 ± 0.5	<b>73 ± 8</b>
+	CPN	93 ± 1	94 ± 1	71 ± 8
+	SKN	93 ± 1	95 ± 1	78 ± 5
+	XYF	95 ± 1	97 ± 0.5	88 ± 3
+	BDK	<b>96 ± 1</b>	<b>98 ± 0.5</b>	<b>92 ± 2</b>

The last four rows show the effect of including the principle of dynamical weight decay (WD) in the networks. Bold numbers refer to the best performing network.

statistical decision criterion based on the Mahalanobis distance [20] yielding a prediction accuracy of 93%. On average, the classification performances of XYF and BDK are 95% and 96%, respectively. As was previously observed for the simulated *Odd–Even* data set, the CPN classification map looked very scattered (not shown here). The classification map of the SKN network also contained a few irregularities, whereas the output maps for the XYF and BDK network exhibit a coherent appearance.

The *Autosort* data set contains NIR spectra of three demolition waste fractions (wood, plastic and stone). From literature it is known that the best classification performance is achieved if the NIR spectra are Standard Normal Variate (SNV) scaled [21]. Hence, for a fair comparison, we adopted this type scaling as well. SNV scaling comes down to auto-scaling row-wise each object in the input matrix. Fig. 8 depicts an alternative way of analysing the network maps. The lower panel shows the

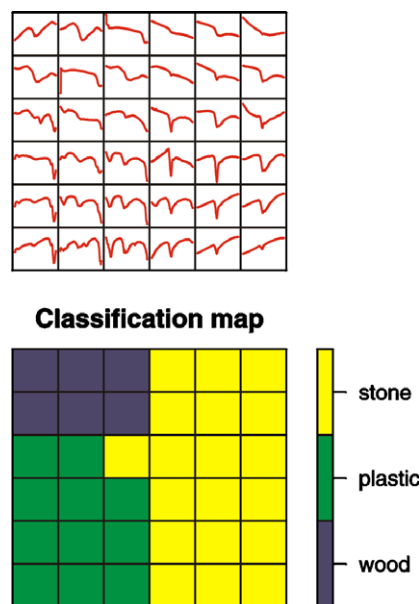


Fig. 8. Analysis of a BDK network for the *Autosort* data set. The profiles of the unit weights in the Xmap of the  $6 \times 6$  network are shown at the top. The bottom shows the classification map for wood, plastic and stone. Each material class corresponds to a set of typical spectral patterns. For example, units identifying the stone class all exhibit a bird-like SNV scaled NIR profile.

6 × 6 classification map for the three material fractions. At the top, the Xmap unit weight vectors are plotted. These profiles can be considered as being typical for the content of the *Autosort* data set. Clearly, all ‘bird-like’ spectral profiles belong to the units representing the stone class (the area at the right hand in the classification map). Also the plastic materials are characterised by typical spectral patterns (appearing down left in the Xmap). The same holds for the units belonging to the wood fraction. The weights for the units in the Xmap exhibit typical features of SNV scaled NIR spectra as could be expected from a spectroscopic point of view [21]. Again, the prediction accuracy of the XYF and BDK network appeared to be better as compared to CPN and SKN. Overall validation results are summarised in Table 2.

To quantify for the *Autosort* data set the relationship between the input and output units, we calculated for each Xmap layer the correlation coefficient with each output map layer. The result of this analysis is visualised in Fig. 9. For example, the first column in this image shows the colour-encoded correlation values between all Xmap layers and the Ymap layer corresponding to ‘wood’ class. The correlations were determined by unfolding the square matrices of the Xmap and Ymap layers to vectors. Then the standard correlation coefficient was computed for each pair of input/output vectors. Like in the analysis of DNA sequences, it can be seen that each waste material possesses its own characteristic correlation profile. In this way, specific regions in the spectra can be assigned which are important for the discrimination between wood, plastic and stone. For example, the wavelengths between 95–115 are important for the

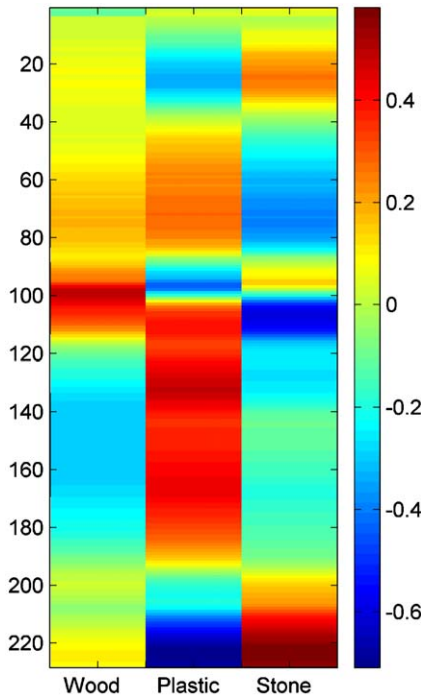


Fig. 9. Colour encoded image representing the correlation coefficients between the unit weight values in the Xmap (228 wavelengths layers) and the three corresponding Ymaps (for classifying the materials wood, plastic and stone) in a BDK network. For more details, see text.

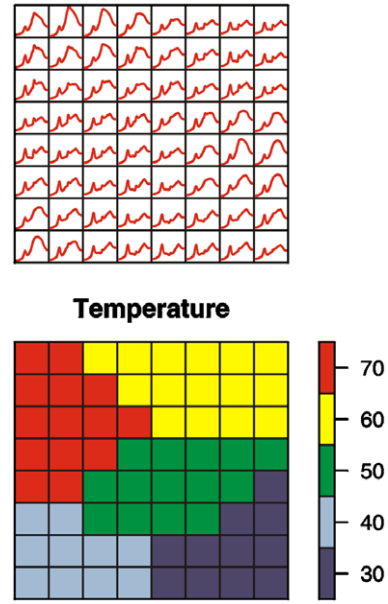


Fig. 10. For the NIR set, the unit weights of the Xmap (represented as spectral profiles) of the BDK network are depicted at the top. The 8 × 8 network was trained for 200 epochs. At the bottom, the corresponding classification map is shown. Temperature classes are indicated by the colour bar.

recognition of wood whereas various other spectral regions are necessary for the identification of plastic and stone.

Fig. 10 shows the Xmap weights of the 8 × 8 BDK network and the corresponding unit class membership in the classification map for the NIR data set. Comparable weight profiles can be observed at different locations in the Xmap. For example, the weights of the units at the upper-left, middle-right and down-left part in this profile plot are quite similar. This is due to the different concentration levels of ethanol, water and iso-propanol used in the experimental design of Wülfert et. al [22]. The

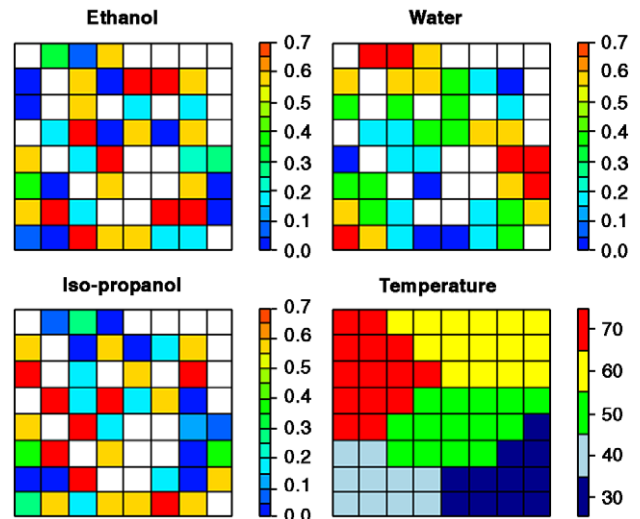


Fig. 11. Based on the trained Xmap the associated mean concentration level maps for ethanol, water and iso-propanol were computed. An uncoloured (empty) unit in each concentration map corresponds to a unit in the Xmap onto which none of the spectra was mapped. The map down-right resembles the BDK classification map. For more details, the reader is referred to the text.

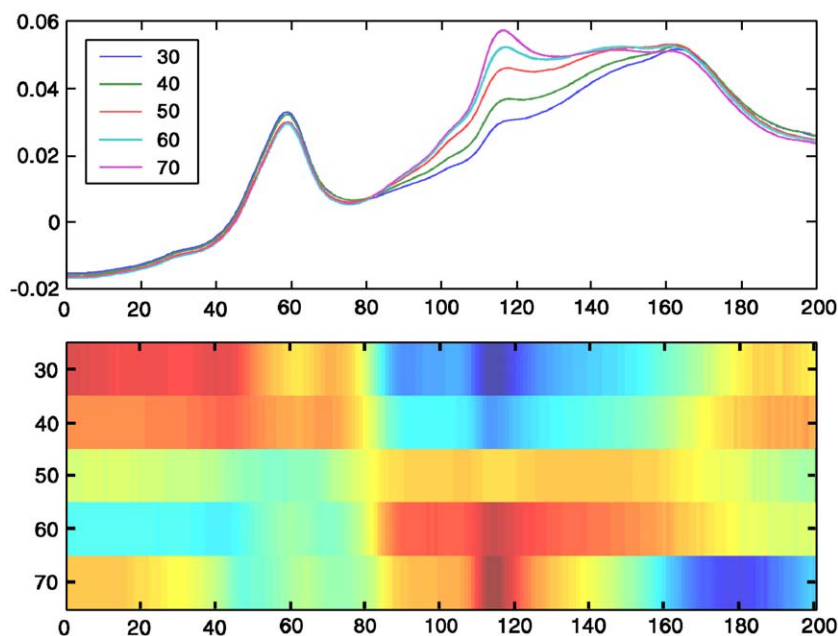


Fig. 12. At the top, the mean spectral NIR profiles for the five temperature classes in the NIR data set. The profiles were obtained by taking into account the distribution of the units in the classification map of the BDK network (see also Fig. 10). At the bottom, the correlation image for the Xmap and Ymap layers. Here, the correlation image is transposed to facilitate the comparison with the mean spectra at the top. The correlation values varied between  $-0.5$  and  $0.4$ .

driving output variable in this particular case, however, is the temperature class. Hence, units are grouped together in the Xmap in order to fit the class membership coherence in the Ymap. Note that in an unsupervised Kohonen or CPN network, or to a lesser extent in a SKN network, similar spectra would have been grouped together in the Xmap.

Fig. 11 show the mapping of the concentration levels of ethanol, water and iso-propanol. These mappings were constructed by projecting a NIR spectrum onto the trained BDK Xmap. At the location determined by the winning unit in the Xmap, we entered in the Ymap layers the associated concentration values. This procedure was repeated for all spectra used for training the network. In case more than one spectrum was projected onto a unit, then the concentration levels were averaged. A close inspection of the water map reveals that for each temperature class region (depicted in the temperature classification map) all concentration levels used in the experimental design are present. This is not the case for ethanol and iso-propanol. This indicates that for water there is for each temperature a tight one-to-one connection between the spectra and the concentrations. Indeed, in Refs. [13,22] it is shown that the concentration of water can be modelled much better than the two alcohols for the full temperature range. For the five areas in the classification map, we also computed the mean weight values of the corresponding units in the Xmap. These spectral temperature profiles are depicted at the top of Fig. 12. Clearly, the strong effect of the sample temperature is most manifest for the wavelengths located at the middle-right part of the average spectral profiles, which is in accordance with previously reported observations. Fig. 12 depicts also the correlation image for the Xmap and the five temperature layers of the Ymap. Clearly, each temperature is characterised by a typical spectral correlation profile. Note the vertical correlation

band at wavelengths close to 115. Moreover, two opposite shifts can be observed for the wavelengths in the vicinity of 80 and 180. The region in between corresponds to the area in the mean spectra where the largest temperature variations are present.

Returning to the issue of the classification performance of the networks, the upper rows in Table 2 demonstrate that XYF and BDK perform on average better than the CPN and SKN networks. Given the large standard deviations obtained from the 10-fold validation procedure, however, statistically there is no significant difference between the four classifiers.

#### 4.3. Effects of adaptive learning and dynamical weight decay

For the *Overlap* data set we investigated the influence of adaptive learning, dynamical weight decay and two similarity

Table 3  
Analysis of the influence of the adaptive learning (AL) mechanism and dynamic weight decay (WD) for the *Overlap* data set

AL	WD	XYF		BDK	
		Euclidean	Tanimoto	Euclidean	Tanimoto
–	–	79±2	82±1	78±2	80±1
+	–	80±2	83±1	79±2	82±1
–	+	79±2	82±1	78±2	81±1
+	+	<b>81±2</b>	<b>84±1</b>	<b>80±2</b>	<b>83±1</b>

In the first column a ‘+’ indicates that adaptive learning was switched on during the training phase. A ‘–’ corresponds to no adaptive learning. The same holds for the second column in the table for the weight decay. For the XYF network, the two columns depict mapping accuracies and standard deviations obtained for three different weight initialisations using the Euclidean and Tanimoto distance applied to the output objects and Ymap. The same holds for the BDK network. In each case, a  $8 \times 8$  network was used which was trained for 50 epochs. The entire *Overlap* data set was used to generate the explorative models. Bold numbers refer to the best performing network.

measures for the output map on the performance of the XYF and BDK networks. The results are summarised in Table 3. Although differences are small, for both networks the prediction performance is worst if both AL and WD are turned off, whereas the accuracy is at maximum if both mechanisms are included. On average, the performance of XYF is slightly better than the BDK network. Strikingly, for all combinations of AL and WD, the mapping accuracy for the Tanimoto distance is on average 3% higher than for the Euclidean distance measure. A possible explanation might be that the sign-dependent Tanimoto distance is a ‘harder’ similarity measure than the continuous Euclidean distance, forcing in the output map a rigorous clustering of units having identical sequences of +/- signs in their weights. This difference in performance was observed for all other data sets as well. For this reason the Tanimoto distance was used throughout this paper to express the similarity between output objects and Ymap units. Obviously, this distance measure is not applicable to the input objects because these contain for all data sets continuously valued variables.

For XYF and BDK the performance is (to some extent) improved by the mechanism of adaptive learning. In none of the cases the mapping accuracy was deteriorated by AL. Therefore, default adaptive learning was switched on during all experiments described in this paper.

The lower rows in Table 2 show that for the *Interpret* data set the performance of all networks was equal or even slightly better if dynamical weight decay was invoked. Apparently, dynamical weight decay allows the networks to rearrange in the initial phase of the training the units in such way that the input weights match at the end in a better way the corresponding class membership. Intuitively, WD can be considered as a kind of regularisation mechanism for a proper formation of the input map units. In other words, WD can be considered as a kind of temporary memory loss which enables the network to (re-)learn from possible other important features in the data.

Including the principle of dynamic weight decay yields for the *Autosort* data set comparable or better results for each network type. The performance of the BDK network (98%) can even compete with the overall performance reported in Ref. [21], in which a time-consuming wavelength selection procedure was conducted together with Kennard–Stone data selection to improve the performance of a Linear Discriminant Analysis classifier based on the Mahalanobis distance.

For the *NIR* set the performance of all network types was just moderate. By including the principle of dynamical weight decay this picture changes dramatically. Both CPN and SKN networks show an increased accuracy of approximately 10%. However, the large standard deviations remain manifest. For these networks  $\text{Decay}_{\text{end}}$  was optimised and yielded a value of 24 (or higher). The average modelling accuracy of the XYF network, however, was improved by 17%, whereas the BDK network even profited a bit more from this mechanism: the classification accuracy changed from 73% to 92%. Moreover, the accompanying standard deviations were reduced considerably, indicating that inclusion of weight decay yields for this data set much more stable models, i.e., classifier networks which are less influenced by the random initialisation of the

weights of the Xmap and Ymap units. Also here,  $\text{Decay}_{\text{end}}$  was optimised resulting in a value of 12 for XYF and BDK. Remarkably, the average performance on the test set of the CPN and SKN networks collapsed for this parameter setting to 40% and 43%, respectively. This indicates that the XYF and BDK networks are more robust regarding the particular setting of the weight decay parameters.

#### 4.4. General remarks and future perspectives

An overall inspection of Table 2 reveals that for all the used data sets XYF and BDK outperform to some extent the CPN and SKN networks. If dynamical weight decay was switched on, for all networks an improvement of the prediction accuracy could be observed. The performance gap between on one hand CPN and SKN and, on the other hand XYF and BDK, remained intact. Moreover, in general higher prediction accuracies were accompanied by smaller standard deviations indicating a higher degree of stability and robustness of the XYF and BDK networks. The XYF and BDK networks yield (within the statistical boundaries given by the standard deviations) more or less equally performing models. Only for the *NIR* data set there appears to be a tendency that a BDK network performs better than the XYF network.

Kohonen networks are intended to extract information from (very) large data bases. Strikingly, even for relatively small data sets considered in this paper (for example, the *NIR* data set contained just 65 training objects for describing the 5 temperature classes) the XYF and BDK networks performed well in a stable, robust way.

Although there are many possibilities for fine-tuning the network settings, the supervised Kohonen networks as presented here only add one extra parameter compared to the unsupervised self-organising feature maps: the weighting  $\alpha(t)$  for the common similarity measure (Eqs. (1)–(3)). This parameter was kept constant for all experiments described in this paper. Preliminary experiments have shown that this parameter does not crucially influence the outcome of training. Virtually identical results to the ones presented in Tables 1–3 have been obtained with XYF networks when  $\alpha(t)$  was set to a constant value of 0.5. Indeed, XYF and BDK are equivalent when  $\alpha(t)$  is set to this value. For different weighting values, however, differences between the behaviour and performance of the XYF and BDK network might emerge if, for example, huge data sets possessing complex non-linear or degenerated relationships will be investigated.

## 5. Conclusions

In this paper we present two novel alternatives for the supervised Kohonen learning strategy for solving typical classification problems which are encountered in the field of chemistry, bio-medicine and chemometrics: the fused similarity based XYF network and the bi-directional updating principle of the maps in the BDK network. Based on the observations made for simulated as well as real-world data sets, it was demonstrated that the supervised SKN network in general

performs better than the classical unidirectional CPN network. In comparison to the CPN and SKN networks, the XYF and BDK networks yield on average better mean classification accuracies and smaller standard deviations.

The good performance of the XYF and BDK networks for some of the data sets is mainly due to the manner the internally weighted similarities between the units in the Xmap and Ymap are handled. The separate determination of the normalised similarity between a presented input/output object and the respective units in the Xmap and Ymap followed by a unique scheme determining the shared winner for the maps results in models which are better able to capture the multivariate structure present in the input space, this in combination with a consistent association to the unit weight values located in the output maps.

All Kohonen networks allow an 'open' way to inspect the relationship embedded in the input and output maps. Such transparency facilitates in an easy way various ways of analysing and interpreting the possible dependencies between the weights of the units in the input and output maps. This analysis might elucidate known as well as unknown or unforeseen (non-linear) relationships between the input and output variables.

It is demonstrated that the Tanimoto distance is better suited than the Euclidean distance as a similarity measure when applied to binary multi-class output objects. Inclusion of adaptive learning in general increases the modelling performance by a few percent for all data sets. As was observed during the examination of the real-world data sets, the inclusion of the mechanism of dynamical weight decay might improve the modelling power of the CPN and SKN networks, and even more of the XYF and BDK networks. In particular, in presence of strong non-linear degenerated relationships (*NIR* data set), the prediction accuracy of the XYF and BDK network was boosted enormously by this dynamic weight rescaling mechanism. In addition, the formation of the XYF and BDK networks becomes more robust, as was expressed by the reduced standard deviations observed for modelling the *NIR* data set. Both networks allow a kind of 'local' modelling (this for each temperature) which preserved, on one hand, the temperature class and on the other hand, the relation to the concentrations of ethanol, iso-propanol and water.

Summarising, the XYF and BDK networks are two novel and powerful modelling alternatives for tackling high-dimensional multivariate input/multi-class output problems. Generally, these techniques yield well-performing prediction models which are characterised by a high classification accuracy in combination with a robust and stable training behaviour. Inclusion of adaptive learning and dynamical weight decay results in equal or even better prediction models. Last but not least, the supervised self-organising feature maps are transparent, allowing a researcher to visualise, analyse and interpret the relationship manifest between the input and output variables in a straightforward and elegant way.

## Acknowledgements

The authors thank Egon Willighagen and Bülent Üstün (Radboud University Nijmegen, the Netherlands) for fruitful

discussions and for providing some of the real-world data sets used in this paper.

## References

- [1] D.L. Massart, B.G.M. Vandeginste, L.M.C. Buydens, S. de Jong, P.J. Lewi, J. Smeyers-Verbeke, Handbook of Chemometrics and Qualimetrics. Part A, Elsevier, 1998.
- [2] B.G.M. Vandeginste, D.L. Massart, L.M.C. Buydens, S. de Jong, P.J. Lewi, J. Smeyers-Verbeke, Handbook of Chemometrics and Qualimetrics. Part B, Elsevier, 1998.
- [3] J.B. Kruskal, M. Wish, Multidimensional scaling, Sage University Paper Series on Quantitative Application in The Social Sciences, Sage Publications, Beverly Hills and London, 1978.
- [4] T. Kohonen, Self-organizing maps, 3rd ed., No. 30 in Springer Series in Information Sciences, 2001, Berlin: Springer.
- [5] W.J. Melssen, J.R.M. Smits, L.M.C. Buydens, G. Kateman, Using artificial neural networks for solving chemical problems: Part II. Kohonen self-organizing feature maps and Hopfield networks, Chemometr. Intell. Lab. Syst. 23 (1994) 267–291.
- [6] J. Zupan, J. Gasteiger, Neural Networks in Chemistry and Drug Design: An Introduction, 2nd ed., Wiley, New York, 1999.
- [7] Y. Vander Heyden, P. Vankeerberghen, M. Novic, J. Zupan, D.L. Massart, The application of Kohonen neural networks to diagnose calibration problems in atomic absorption spectrometry, Talanta 51 (2000) 455–466.
- [8] R. Wehrens, W.J. Melssen, L.M.C. Buydens, R. de Gelder, Representing structural databases in a self-organising map, Acta Crystallogr., B 61 (2005) 548–557.
- [9] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995.
- [10] J.A. Freeman, D.M. Skapura, Neural Networks: Algorithms, Applications and Programming Techniques, Addison & Wesley, New York, 1991.
- [11] J.R.M. Smits, W.J. Melssen, L.M.C. Buydens, G. Kateman, Using artificial neural networks for solving chemical problems: Part I. Multi-layer feed-forward networks, Chemom. Intell. Lab. Syst. 22 (1994) 165–189.
- [12] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, Singapore, 1999.
- [13] B. Üstün, W.J. Melssen, M. Oudenhuijzen, L.M.C. Buydens, Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization, Anal. Chim. Acta 544 (2005) 292–305.
- [14] G. Carrera, J. Aires-de-Sousa, Estimation of melting points of pyridinium bromides ionic liquids with decision trees and neural networks, Green Chem. 7 (1) (2005) 20–27.
- [15] R. Hecht-Nielsen, Counter propagation networks, Appl. Opt. 26 (23) (1987) 4979–4984.
- [16] Y.D. Xiao, A. Clauset, R. Harris, E. Bayram, P. Santiago II, J.D. Schmitt, Supervised self-organizing maps in drug discovery: 1. Robust behavior with overdetermined data sets, J. Chem. Inf. (2005) Model.
- [17] E. Bayram, P. Santiago II, R. Harris, Y.D. Xiao, A.J. Clauset, J.D. Schmitt, Genetic algorithms and self-organizing maps: a powerful combination for modelling complex QSAR and QSPR problems, J. Comp. Aid. Mol. Des. 18 (2004) 483–493.
- [18] D. Rogers, T. Tanimoto, Improved tools for a biological sequence comparison, Proc. Natl. Acad. Sci. U. S. A. 85 (1960) 2444–2428.
- [19] C.A. Casas, Reducing portfolio volatility with artificial neural networks, Proc. Artificial Intelligence and Applications, 2005, p. 453.
- [20] A.W. Simonetti, W.J. Melssen, F. Szabo de Edelenyi, J.J.A. van Asten, A. Heerschap, L.M.C. Buydens, Combination of feature-reduced MR Spectroscopic and MR imaging data for improved brain tumor classification, NMR Biomed. 18z (2005) 34–43.
- [21] P.J. de Groot, G.J. Postma, W.J. Melssen, L.M.C. Buydens, Selecting a representative training set for the classification of demolition waste using remote NIR sensing, Anal. Chim. Acta 392 (1999) 67–75.