BONIECKI P.
Agricultural University of Poznan, Poland
Institute of Agricultural Engineering

THE *KOHONEN* NEURAL NETWORKS IN CLASSIFICATION PROBLEMS SOLVING IN AGRICULTURAL ENGINEERING

S u m m a r y

During the adaptation process of the weights vector that occurs in the iteration presentation of the teaching vector, the Kohonen type neural network attempts to learn the structure of the data. Such a network can learn to recognise aggregates of input data occurring in the input data set regardless of the assumed criteria of similarity and the quantity of the data explored. Following identification of aggregates occurring in the data set, they can be named (labelled), and as a result the Kohonen network gains the ability to classify them in compliance with the inner logic included in the data set. The Kohonen type neural network can therefore be used for classification of data also when the output classes are not known (defined) in advance.
Key words: *Kohonen* neural networks, recognition of an image

SIECI NEURONOWE TYPU *KOHONENA* W KLASYFIKACYJNYCH PROBLEMACH INŻYNIERII ROLNICZEJ

S t r e s z c z e n i e

Podczas procesu adaptacji wektora wag zachodzącego w trakcie iteracyjnej prezentacji wektora uczącego, sieć neuronowa typu Kohonena próbuje nauczyć się struktury danych. Sieć taka może nauczyć się rozpoznawania skupień występujących w zbiorze danych wejściowych bez względu na przyjęte kryteria podobieństwa oraz ilość eksplorowanych danych. Po identyfikacji skupień występujących w zbiorze danych można nadać im nazwy (zaetykietować je), skutkiem czego sieć Kohonena uzyskuje możliwość przeprowadzania ich klasyfikacji, zgodnie z wewnętrzną logiką zawarta w zbiorze danych. Sieć neuronowa typu Kohonena może zatem być użyta do klasyfikacji danych również wtedy, gdy klasy wyjściowe nie są z góry znane (zdefiniowane).
Słowa kluczowe: sieci neuronowe, rozpoznawanie obrazu

Introduction

The *Kohonen* artificial neural networks are modelled on the topological properties of the human brain. These networks are also known as self-organizing feature maps (*SOFM - **S**elf **O**rganizing **F**eature **M**aps*). The *Kohonen* neural networks are most frequently used for a widely understood classification [1]. They perform this task in a relatively untypical way. Thanks to the processing of output values carried out within the *postprocessing*, the result of the network activity is an output variable of nominal character. Each value of this variable represents one definite class. Appropriate neurons occurring in the output layer of the network correspond to particular classes. The connection of the neuron with a given class is indicated by the label with the class name attributed to it. During the action of the network, after the input signal has appeared, the winning neuron is indicated each time (that is the neuron of the lowest activation, which indicates the highest compatibility of weights and the given input pattern). The label of the neuron determines the class to which the input case is ascribed. This untypical structure enables the user to define the output layer of the *Kohonen* network as a specific two-dimensional "map" of the input multi-dimensional data set. It enables to place in it an optional number of neurons that are distinguished and established points in this map (Tadeusiewicz R., 1990).

The topology of the *Kohonen* network differs considerably from the structures of other neural networks. The structure discussed is basically a one-layer network. It consists of an input layer and an output layer which processes the data presented. The output layer is built of radial neurons. This layer is also defined as a layer forming a topological map. Neurons in the layer forming the topological map are taken into consideration as if they were placed in space according to some predetermined order - usually for convenience and better perception we imagine them to be nodes of a two-dimensional network.

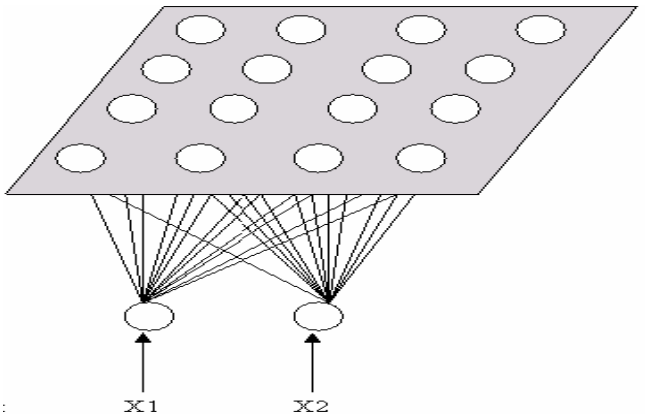

Fig. 1. The structure of the *Kohonen* type neural network

The nature of the teaching process of the *Kohonen* network

During the teaching of the self-organizing networks by competition at the input of each neuron, an n-dimension signal x from the set of teaching patterns is provided (randomly or in an established sequence). The weights of the synaptic connections create a vector $W_i = [W_{i1}, W_{i2}...., W_{in}]$. Next, input vectors are normalised and then the network is stimulated by the teaching vector *x*. In the adaptation process, this neuron whose weights differ the least from vector *x* presented at the input, wins in the competition. The winning *w*-th neuron fulfils this relation (Tadeusiewicz R., 1990):

$$d = (x, W_w) = \min_{1 \le i \le n} d(x, W_i) \tag{1}$$

where:

$d(x, W)$ - determines the measurement of distance between vector x and vector *W*,

n - number of neurons,

x - input (teaching) vector,

W_w -"winning" vector of weights.

The distance between vector *x* and weight vector of the winning neuron W_i can be calculated using different definitions e.g.:

- Euclidean measurement:

$$d(x, W_i) = \|x - W_i\| = \sqrt{\sum_{j=1}^{N} (x_j - W_{ij})^2} \tag{2}$$

- scalar product measurement:

$$d(x, W_i) = x \cdot W_i = \|x\| \cdot \|W_i\| \cos(x, W_i) \tag{3}$$

where:

$\|x\|, \|W_i\|$ - modules of vectors *x* and W_i

Around the "winning" neuron, a topological neighbourhood is assumed of a definite radius that is decreasing in the course of the teaching process. The "winning" neuron and all neurons situated within the neighbourhood area undergo a process of adaptation, changing their weights vectors toward the input vector x according to the formula:

$$W_i(k+1) = W_i(k) + \eta_i(k)(x - W_i(k)) \tag{4}$$

where:

η_i - teaching coefficient (decreasing in time),

k - number of steps (iteration).

The numbers of neurons winning in consecutive presentations of vectors x create the so-called code book. In classical coding, the algorithm k – mean values is usually applied, whose equivalent in the case of neural networks is the *WTA* (***W**inner **T**akes **A**ll)* algorithm. In this algorithm, after the presentation of vector x the activity of each neuron is calculated. The winner is the neuron of the biggest output signal. When using normalised vectors, it equals the smallest Euclidean distance between the input vector and the neuron weight vector. The winner adapts its weight in the direction of vector *x* according to the formula:

$$W_w \leftarrow W_w + \eta(x - W_w) \tag{5}$$

where:

η - teaching coefficient,

x - input vector,

W_w - weights vector of the winning neuron.

It should be noted that the remaining neurons do not undergo adaptation. The *WTA* algorithm takes into account the so-called fatigue of neurons by considering the number of victories of each of them and favouring the less active neurons to equal their chances. Such a modification is used in the initial stages of the algorithm switching it off after activation of all neurons. In the *WTA* algorithm only the "winner" modifies its weights (Ossowski S., 2000).

In the *WTM* (***W**inner **T**akes **M**ost*) algorithm, apart from the winner its neighbouring neurons also update their weights. The longer the distance of the neuron from the winner, the smaller the change of value of the neuron weights. The weights adaptation process can be described using the dependence relationship:

$$W_i \leftarrow W_i + \eta_i G(i, x)[x - W_i] \tag{6}$$

for all *i* - th neurons belonging to the "winner's" neighbourhood.

In the formula, the teaching coefficient η_i of every neuron is separated from its distance with reference to vector x included using the neighbourhood function $G(i, x)$ defined in the following way:

$$G(i, x) = \begin{cases} 1 & for\ i = n \\ 0 & for\ i \neq n \end{cases} \tag{7}$$

where:

$G(i, x)$ - function of neighbourhood,

n - number of the "winner".

There are many types of *WTM* algorithm, which differ first of all in the neighbourhood function $G(i, x)$.

In the classical algorithm put forward by the Finnish scholar *T. Kohonen* the neural network initiates by assigning to neurons definite place in space, associating it next in a specific way with neighbours. The moment the „winner" is called, the updating includes not only the weights of the „winning" neuron but also the weights of its neighbours from the closest neighbourhood. In the algorithm discussed, the function of neighbourhood $G(i, x)$ takes the following form:

$$G(i, x) = \begin{cases} 1 & for\ d(i, w \le \lambda) \\ 0 & for\ other \end{cases} \tag{8}$$

where:
$d(i,w)$ - means the Euclidean distance between the weights vectors of the "winner" neuron w and the i-th neuron or the distance measured in the number of neurons,
λ - means the radius of neighbourhood, whose value decreases in time to zero.

This type of neighbourhood is called rectangular. Another type of neighbourhood used in the *Kohonen* network is the Gaussian type, which can be described using the following formula:

$$G(i,x)=\exp\left(-\frac{d^2(i,w)}{2\lambda^2}\right) \quad (9)$$

where:
λ - is the neighbourhood radius decreasing in time,
$d(i,w)$ - measure of distance between the weights vector of the "winner" neuron w and the i-th neuron.

The adaptation degree of neurons from a neighbourhood is determined not only by the distance of the *i* - th neuron from the "winner" (*w*-th neuron), but also the neighbourhood radius (Ossowski S., 2000).
In the neighbourhood of the *Gaussian* type the degree of adaptation varies and depends on the value of the *Gaussian* function, whereas in the rectangular type neighbourhood every neuron belonging to the winner neighbourhood undergoes adaptation to an equal extent. The final formula for adaptation of neighbours weights can be presented as follows:

$$W_i(k+1)=W_i(k)+\eta\exp\left(-\frac{d^2(i,w)}{2\lambda^2}\right)\left(x-W_i(k)\right) \quad (10)$$

where:
λ - radius of the neighbourhood,
η - teaching coefficient (decreasing in time),
$d(i,w)$ - distance between the vector of weights of the winner neuron w and the i-th neuron
k - number of iteration,
W_i - weights vector of the i - th neuron.

The neighbourhood is decreased with the passage of teaching time. Initially, the neighbourhood includes a relatively large number of neurons. At the final stages, the neighbourhood has a zero range. The algorithm makes use of the time-variable coefficient of teaching that is applied for determination of the weighted sum and causes that the initially large and rapid changes become increasingly smaller during successive epochs. Teaching the *Kohonen* networks is frequently carried out in two clearly divided stages: at the beginning a relatively short phase with a large teaching coefficient and extensive neighbourhood, and a long phase of "repeated and supplementary teaching" with a small value of the teaching coefficient and a zero or almost zero neighbourhood.
The neighbourhood plays a crucial role in *Kohonen* neural network training. By updating surrounding units in addition to the winning unit, *Kohonen* training assigns related data to contiguous areas of the topological map. During training the neighbourhood size is progressively reduced, together with the learning rate, so that early on a coarse mapping is produced (with large clusters of units responding to similar cases). Finer detail is produced later (as individual units within a cluster respond to finer levels of discrimination among related cases).

Application areas of *Kohonen* neural network in agricultural engineering

Many areas of agricultural engineering lend well to the application of *Kohonen* neural networks. These include:
- predicting crop yields, particularly for forecasting the yield on the basis of previous years' results;
- spatial approximations (where physical parameters of several points in space are used to extrapolate the values in between); applications include among others geological studies of soil, thermal field analysis in heat storage units;
- classification problems of recognising selected categories; applications include the use of neural networks to entomologically classify insects based on their representative features (such as colour);
- identification problems – signal screening (the processing of high-noise data); applications include the analysis of stochastic weather data.

The above instances of *Kohonen* neural network applications are reflected in research that has been conducted by various Polish research institutes. Their findings have generated a range of specific applications.
In agricultural engineering research, *Kohonen* neural networks have proven to be of particular use in resolving a wide range of classification problems. The most common task performed by neural networks in this area is the assigning of each submitted case represented by an appropriated set of input data to one of previously selected classes (categories). Generally one may say that the network estimates the probability of a given case belonging to a given class.
Since *Kohonen* networks learn data structures during the weight adaptation process, one of the applications of such networks is exploratory data analysis (and particularly the analysis of empirical data). *Kohonen* neural networks have the capacity to learn to e.g. recognise concentrations in the input data and at the same time link similar classes of data. Notably, the network can achieve this regardless of the type of data similarity criteria used (even if algorithm designer does not know such criteria *a priori*) and the number of such criteria.
The *Kohonen* neural networks are usually better than other analytical methods, if the following conditions, regarding the empirical input data of the processing process are fulfilled. In classification problems, the purpose of the network is to assign each case to one of a number of classes (or, more generally, to estimate the probability of membership of the case in each class). One of advantages of the *Kohonen* neural network, is the ability of the discussed artificial neural network to determine the degree of similarity occurring between classes. This network can be also used to detect regularities occurring in the obtained empirical data. The desirable feature enables the *Kohonen* neural network to identify the pests correctly based on the presentation of images not originating from the teaching

set i.e.: noisy photographs taken under different light exposure conditions and using different quality of the equipment. In connection with the above, it is possible to find many areas in generally defined agricultural engineering, in which the use of *Kohonen* neural networks seems to be particularly justified.

For example category process recognition in agricultural engineering (classification problem): on the basis of the characteristic features of an insect the *Kohonen* neuron network determines to which entomological category the object belongs (Boniecki P., Piekarska-Boniecka H., 2004).

Unfortunately, there are certain limitations to the use of neuron networks. One of the problems relating to the networks is that it is not possible to use them to obtain a mathematical relation between the input and the output of neuron networks. The network creates a neuron model of the studied phenomenon, but it is not possible to determine the structure of the model - only the network structure is known. Apart from this, it is difficult to prove that a network, which has been taught, can really cope with the assigned problem. The nature of the learning process of a neuron network often determines the stochastic nature of results, which are output by the network. It is also important to realise that there is no method to teach a network to use magic to create information, which is not included in the training data.

Conclusion

The radial layer of a *Kohonen* artificial neural network, with units laid out in two-dimensions, is trained so that inter-related clusters tend to be situated close together in the layer. That's why neural network is often used for cluster analysis.

The *Kohonen* training algorithm adjusts the centres in the topological map layer to move them closer to cluster centres in the training data. During training, the algorithm selects the unit whose centre is closest to the training case. That unit and its neighbours are then adjusted to be more like the training case. *Kohonen* neural networks perform unsupervised learning: they learn to recognise clusters within a set of unlabelled training data: that is, training data which includes input variables only. This neural networks also place related-clusters close together in the output layer, forming a topological map. Labelling of cases is less frequent, but since *Kohonen* neural networks are used for unsupervised training tasks, where the clustering of data is typically not understood before training begins, sometimes as you begin to understand what clustering units represent, you may label the cases accordingly.

In a *Kohonen* neural network, the winning node in the topological map (output) layer is the one with the highest activation level (which measures the distance of the input case from the point stored by the unit). Some or all of the units in the topological map may be labelled with class names. If the distance is small enough, then the case is assigned to the class (providing that a name is given). If an input case is further than this distance away from the winning unit, or if the winning unit is unlabelled (or its label doesn't match one of the output variable's nominal values) then the case is unclassified. The reject threshold is not used in *Kohonen* networks.

Literature

[1] Tadeusiewicz R., 1990. Sieci neuronowe [Neural networks], Warsaw.

[2] Ossowski S., 2000. Sieci neuronowe do przetwarzania informacji [Neural networks for processing information], Warsaw.

[3] Boniecki P., Piekarska – Boniecka H., Neuronowa identyfikacja wybranych szkodników drzew owocowych w oparciu o cyfrowe techniki analizy obrazu [Neural identification of selected fruit trees pests based on digital techniques for the analysis of images], Journal of Research and Applications in Agricultural Engineering (3'2004), Vol. 49(3), pp. 25-30, Poznań.