

Classification Model Based on Kohonen Maps

Jiří Jelínek

Institute of Applied Informatics, Faculty of Science, University of South Bohemia, CZECH REPUBLIC, Ceske Budejovice, 1760
Branisovska St., email: jjelinek@prf.jcu.cz

Abstract: The standard Kohonen map uses unsupervised learning and single Kohonen layer, which allows the usage for clustering and visualization. The number of model parameters is relatively small and their settings are therefore not so complicated. The aim of this paper is to introduce three modifications of this basic model so that it can be used for classification tasks. The first change is the transition to supervised learning by adding input data about the required outputs. The second modification is the implementation of the hierarchical model structure to improve the classification results. The third extension is the implementation of an optimization mechanism for setting the parameters of the model because the number of model parameters was extended and their adjustment was more difficult. The results of the experiments with modified model will be presented too.

Keywords: Kohonen map, neural networks, pattern classification.

I. INTRODUCTION

A variety of methods is available for advanced data processing, often from the field of artificial intelligence. Their use then depends on what data we have and what tasks we want to solve.

One group of possible tools for the implementation of machine learning or data analysis has quite a long been a neural network based on an artificial neuron model [4]. Probably the greatest attention was given to models of multi-layered perceptron [5] and models based on this principle and using different methods of learning (e.g., back propagation of error). However, other models are also available (but the structure of the model described below, Kohonen maps, is similar to the multilayer network as well).

We use unsupervised and supervised learning methods for learning neural networks. When unsupervised learning is used, we only have unrated input data that we intend to analyze in some way. A typical example of unsupervised learning is the ART algorithm and model [1], which is capable to solve cluster analysis task.

When supervised learning is used, we train the model not only with the input patterns but also with the required output. These examples of the $R^n > R^m$ transformation are used to form internal rules or model parameter settings. A typical example is a neural network with the algorithm of learning by error propagation [6].

The whole process of using the neural network model can be divided into two main phases - the setting phase (learning) and the production phase (recall). The production phase is the very reason for the existence of the model. In it the learned

settings are used for processing of previously unseen (test) input data.

A very interesting model is the so-called Kohonen map primarily designed for unsupervised learning and therefore cluster data analysis and visualization. However, the results of its use by the author on previously solved tasks [2] revealed that the use of the standard model and learning did not always lead to the desired results and it was necessary to solve also the classification tasks with a predetermined classification of inputs in addition to the clustering.

Therefore, a modified learning algorithm and a multi-level model structure based on Kohonen maps were designed. The very basic model was at first used for economic data processing [7]. The aim of this paper is to present actual state of the model with key changes in the hierarchical learning and recall and also in modifying the learning process. These changes were supposed to improve the quality of the model and its generalization capability, which was tested in experiments as well.

The following chapters of the paper are organized as follows. Chapter 2 focuses on a brief description of the standard Kohonen map model and shows the key parameters of it. Chapter 3 then concentrates on the description of the modifications that were made, and Chapter 4 focuses on experiments with the model conducted primarily to verify the benefits of the proposed modifications.

II. RELATED WORK

The Kohonen map [3] was first introduced in the 1980s as most models of neural networks of different types.

In some ways, Kohonen map can remind us of the ART2 model [1]. The similarity lies in the same requirements on input data (number vectors). Similar is also the two-layer structure of the network. However, Kohonen map is strongly focused on the visual interpretation of the output and is therefore useful both for the better understanding of the task and for use in an online dynamic environment. An example of such usage can be monitoring of the state of the system [2].

The core model activity is basically the same as ART2 - assigning input patterns to the cells of the second layer (i.e. the output clusters representing by these cells) based on the similarity of patterns.

Basic Model Functionality

The input layer of the Kohonen map is composed of the same number of cells as the dimension n of the input space R^n is. The output layer is two-dimensional and is also referred to as the Kohonen layer. The input and output layers are fully interconnected from the input to the output one with links whose weights are interpretable as the centroid of the input

patterns cluster represented by the cell of the output layer. The number of output layer cells is the model parameter.

In the production phase, test patterns are submitted at the input of the model. Their distance (here the Euclidean one) is calculated from the output layer cell centroids and the input pattern is assigned to the output cell, from which it has the smallest distance. Assigning the input pattern to the output cell can be visualized in the output layer as shown in Fig. 1.

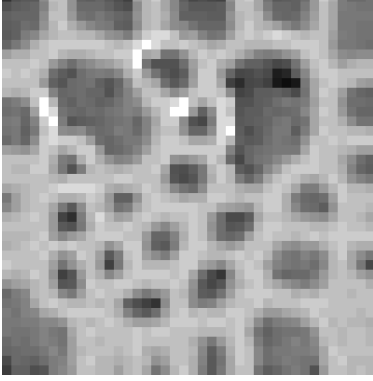


Fig. 1. Kohonen layer visualized according to the number of patterns represented by cells (the more patterns, the darker color).

The learning of the Kohonen map is iterative and is the extension of the production phase. During it, the values of the weights leading to the cell representing the pattern are modified (the winning cell, the pattern was assigned to it) so as the weights to the cells in its neighborhood. The centroids of all these cells are moved towards the vector representing the pattern according to the Eq. (1).

$$c_i^{new} = (1 - \alpha)c_i^{old} + \alpha s_i \quad (1)$$

In it the α is the learning coefficient for the winning cell, usually with the value from the interval (0,1), c_i is the i -th coordinate of the cell's centroid and s_i is the i -th coordinate of the given input pattern.

The values of weights leading to the neighbor cells are modified according to the same formula, but with a different α_{ij} learning coefficient value adjusted to respect the distance of a particular cell from the winning one:

$$\alpha_{ij} = \frac{\alpha}{(1 + d_{ij})} \quad (2)$$

Coordinates are taken relative to the winning cell with position [0, 0]. The distance d_{ij} from the winning cell in these relative coordinates is then calculated as Euclidean one. The neighborhood of the winning cell is then defined by the limit value of $d_{ij} \leq d_{max}$.

The Kohonen map also includes a mechanism to equalize the frequency of cell victory in the output layer. For each of them, the normalized frequency of their victories in the representation of the training models f_q is calculated with the normalized value in the interval (0, 1). This is then used to modify the distance of the pattern from the centroid.

$$w_{pq} = d_{pq}(f_q + K) \quad (3)$$

In the formula, w_{pq} is the modified pattern distance of pattern p from centroid q , d_{pq} the original Euclidean distance of them

and K global model parameter to limit the effect of the equalization mechanism. The calculated distance w_{pq} is then used in the learning process. The described mechanism ensures that during the learning the weights of the whole Kohonen layer will gradually be adjusted. The size of this layer together with the number P of input patterns actually determines the sensitivity of the network to the differences between the input patterns.

It was also necessary to choose the appropriate criterion for determining the end of learning. This is based on the average normalized distance v in the 2D layer through which the pattern "shifts" between the two iterations as shown in the Eq. (4).

$$v = \frac{1}{\sqrt{2NP}} \sum_P \sqrt{\Delta i^2 + \Delta j^2} \quad (4)$$

The distance is calculated on the square-shaped Kohonen layer (with N cells on the side) between the two iterations of the input set P . The v value is compared to the maximum allowable value v_{max} determining the average maximum shift of patterns allowed in one iteration.

The main parameters of the Kohonen maps are the learning coefficient α , the way of setting the decrease of this coefficient for the cells around the winning cell, the size of the neighborhood given by d_{max} and the number of cells in the two-dimensional output layer. In addition, the behavior of the model also influences the value v_{max} and the coefficient K in Eq. (3) and its possible change over time.

III. MODEL MODIFICATIONS

The modified learning algorithm and the multilevel structure of the model were designed together with optimization mechanism for model settings.

Modified Learning

The goal of learning is to get an adjusted model that is capable of certain generalization, i.e. adequate responses to patterns that have not been trained. The basic Kohonen model described above used unsupervised learning. The modification using supervised learning for solving classification tasks will be introduced.

At the learning stage, the first change was in the fact that after finding the winning cell in the Kohonen layer, the model also stores the classification of the training patterns that were assigned to it. With the help of a known classification of the training set, we can determine the probability rate of output categories in every cell, which can be used in the production phase to evaluate the test patterns in one of these two ways:

- *Probability evaluation.* We select the most likely category, so that the test pattern is always classified. The assumption here is the same distribution of *a priori* probabilities of output categories.
- *Absolute evaluation.* We only evaluate a test pattern if it is assigned to a winning cell representing only one category. A significant percentage of patterns can thus be unclassified.

So the modified model already uses at the production stage additional output information.

In the course of model experiments it has also been shown,

that classification task on data with complex transformation $R^n > R^m$ tend to a state where the same classified patterns are assigned to output layer cells that are often very distant from each other. This affects the overall efficiency of the model that must respect this fragmentation.

The aim was to limit this phenomenon by using the output categorization directly in the model's learning phase. In this case, the model is trained on data that are a conjunction of the original input and the desired output (classification). For example, if we have a classification task performing the transformation $R^d > B^l$, where B^l represents a one-dimensional binary space (one binary coordinate), the model will be taught on the input set $R^d \cap B^l$. In the production phase, the last coordinate b_l is not used in the calculations because the input test vectors will not contain it (their classification is not known).

The described modification significantly changes the model settings, but it has turned out to be a positive change under certain conditions. The key is to what extent the output (often binary) classification should be projected into the training input. If this projection is in full binary value and inputs from R^d are normalized to a range (0; 1), the model settings are distorted too much and model is not capable to generalize. Therefore, the new reduction factor u was implemented to limit this projection according to Eq. (5)

$$inp_{x+|R|} = ub_x, \quad (5)$$

where $inp_{x+|R|}$ is the value of the input of the model (preceded by the coordinates of the original input from the R space) and b_x is the original classification ($b_x = 0$ or $b_x = 1$ for pure binary classification). The factor u has a value from the interval (0; 1) and represents next parameter of the model.

Hierarchy Structure

The fundamental change in the work with Kohonen map is its repeated use with a different training set. This set can be e. g. quite uneven in terms of the representation of output categories or too large for the actual size of Kohonen layer. The modified model addresses this problem by gradually reducing this set by eliminating properly classified training patterns at the end of each learning iteration. In the next step, a new instance of the map is already learned with a training set containing only problematic (not yet categorized) patterns. The underlying idea of this approach is to use Kohonen map internal mechanisms so that the map in every step refines its classification capabilities.

Thus, in each model step, a separate Kohonen map is used. After learning, it is examined whether only patterns of one output category are assigned to the given cell. If this is the case, we can say that the map can correctly classify these patterns in accordance with the desired output and they can be excluded from the training set (Fig. 2). The successful classification of input pattern is considered as:

- *Basic classification.* Assignment to a cell representing only patterns of the same category.
- *Strict classification.* Assignment of the pattern to a cell according to the previous bullet and additionally adjacent to only the same (representing the same category) or empty cells (not representing any pattern).

The selection of one of the above classification methods is a parameter of the model.

Two criteria are crucial for the real use of the proposed model. The first one is the criterion of learning termination in each step (level) of hierarchical model learning. The criterion of the maximum average shift distance between the 2D layer cells according Eq. (4) was used.

The second criterion is that of the overall ability of the whole set of learned sub-models to correctly classify the training and later the test set of patterns. In the model was used the minimum size of the training set, which still makes sense for learning. Its higher value reduces the number of hierarchical classification steps but also limits the sensitivity of the model.

In the production phase, the classification method differs depending on whether we are in the last hierarchical step or not. For the last model in the structure, the probability evaluation is always used, where the pattern is assigned to the most likely category resulted from the learning process.

The modified model was set up using 11 parameters including both the Kohonen map original ones (used in every iteration) and the other ones characterizing the hierarchical model's operation.

IV. EXPERIMENTS

The experiments carried out were aimed at confirming the preliminary hypothesis that both modifications of the Kohonen map model are beneficial to the generalization quality and hence the classification of the test set of patterns. The test data were artificially created to represent a complex nonlinear transformation from the input space R^d to the space B^l . 10,000 training and test patterns were generated with random coordinate values in the interval (0; 1) from R^d space.

One test set and two training sets were created. From the training sets one was for the classical learning of the model (only inputs from R^d) and the other one extended with the output b_l (the network input dimension increased to R^5 where the fifth coordinate was created from b_l). Experiments have been optimized for maximizing the number of properly classified test patterns.

As mentioned above, the model has a number of adjustable parameters that significantly affect its results. Searching for optimal setup manually would be a lengthy process, and therefore, a superstructure of the model was used implementing an optimization mechanism based on genetic algorithms. The possible values (for categorical data) or limit minimal and maximal values (for number data) were inserted into the optimization system and the optimal values were searched through crossings and mutations of coordinates of settings chromosome. The chromosome evaluation function was defined as maximum correctly classified input vectors.

Two variants of model's setting have been examined. The first one focused on modified model learning and worked with a single level of the hierarchical model (as if the model was not hierarchically modified). The second variant involved a hierarchical modification of the model structure with the maximum number of levels limited to 10.

The best results are presented in Table 1. These are the best from 406 experimental settings of the model calculated by the

genetic algorithm for each variant.

It is clear from the table that the use of the modified learning process brings a significant improvement in the classification capabilities of the network. The key output is the finding that to achieve better results with normalized data, the output values used for learning (originally 0 and 1) must be reduced by the reduction factor. Its appropriate setting was found by genetic algorithm and was 0.2 or 0.3 (see the Reduction factor row in Table 1).

TABLE 1. BEST RESULTS FOR DIFFERENT MODEL SETTINGS

Parameter	Max. 1 level		Max. 10 levels	
	classic	extended	classic	extended
Number of 2D cells	40*40	35*35	40*40	35*35
Classification criterion	basic	strict	strict	strict
Reduction factor	-	0.2	-	0.3
Real used levels	1	1	10	3
Total learning iterations	33	118	238	515
Test patterns correctly classified	7998	8472	7985	8643
Model success [%]	79.98	84.72	79.85	86.43

The visual outputs of the 2D network for the setting from the last two columns of Table 1 in level 1 are shown in Fig. 2 to demonstrate the effect of modified learning. The cells representing the patterns rated 0 are green, the patterns rated 1 are red. We get "clean" colors for cells containing input patterns included in only one category, for cells containing patterns of different categories the color is mixed. This respects the number of patterns in a cell with different output categories. The influence of additional output information on the final network setting is quite obvious (right) and the fragmentation when using the classical learning algorithm (left) almost did not occur.

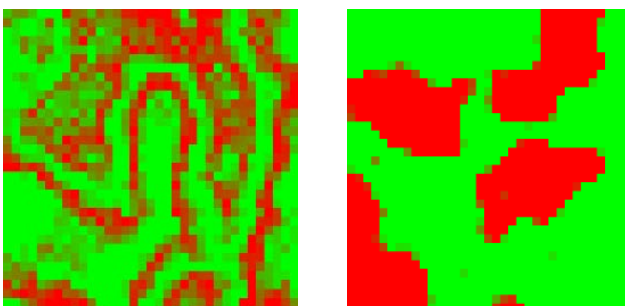


Fig. 2. Influence of modified learning algorithm.

The results can still be improved by using a hierarchical modification of the model, but for the selected transformation $R^n > R^m$ the added value is not so high (quality improvement 1.71 %). In this case, the training set was evenly generated, but the benefit will be more significant on data with

unequally represented output categories or different *a priori* probabilities of them.

V. CONCLUSION

This paper focuses on introducing a modified learning algorithm for the Kohonen map, which enables solving of the classification tasks. The core of the modification is the use of training set extended with the output categorization of the training patterns. Modifications were made even in setting of the criteria for completing model learning (criterion of minimal pattern shift in the 2D layer). A visual superstructure of the model was also developed to allow a detailed study of the dynamics of the model setup process, and the obtained knowledge could be used to better understanding of the learning process and the nature of input data.

The second important modification is the design and description of the behavior of a hierarchical classification model based on modified Kohonen maps. The training set is gradually reduced during the process of learning. This increases the sensitivity of the network to differences in input data. The algorithm for the production phase of the model was developed, based on the learned Kohonen map sub-models.

The behavior of the hierarchical model was described by a series of input parameters, whose values had to be empirically determined. Therefore, to find optimal values, optimization system based on genetic algorithms has been used.

With the modified model, experiments were conducted to verify the benefits of the proposed modifications. They confirmed the positive influence of the extended training set and the hierarchical structure of the model for better classification performance. The overall classification quality was improved by 6.58% on the generated data with nonlinear randomly selected transformation function $R^t > B^t$. The benefit of the hierarchical structure would be greater when using data unevenly covering the input space.

Future work on the model will focus on further examining the benefits of proposed modifications to the quality of the classification process. Attention will also be paid to an optimization mechanism that could include more data characterizing the model's activity.

REFERENCES

- [1] G. A. Carpenter and S. Grossberg, "ART 2: Self-organization of stable category recognition codes for analog input patterns", in *Applied optics*, Vol. 26(23), 1987, pp. 4919-4930.
- [2] J. Jelínek, *Exceptional working states of power grids*. Qualification work. CTU - FEE, Prague, 1992. (in Czech)
- [3] T. Kohonen, "Self-organized formation of topologically correct feature maps", in *Biological cybernetics*, Vol. 43(1), 1982, pp. 59-69.
- [4] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", in *The bulletin of mathematical biophysics*, Vol. 5(4), 1943, pp. 115-133.
- [5] F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Labs., 1952.
- [6] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors", in *Nature*, Vol. 323.9, 1986.
- [7] M. Vochozka, J. Jelínek, J. Váchal, J. Straková and V. Stehel, *Using of neural networks for comprehensive business evaluation*. H. C. Beck, Prague, 2017. (in Czech).